

*** 실험 3, 4주차 수업은 라이브러리 함수나 구조체를 이용하지 말고 레지스터 어드레스를 직접 접근하여 제어/조작해야 합니다. ***

- 각 레지스터의 필드값은 초기화 후 원하는 값을 대입해야 합니다.
- 기존 reset value가 이미 들어있기 때문에 그냥 |=로 대입하면 다른 값이 될 수 있습니다.

예) GPIO 포트 B 0번핀 input with pull-up / pull-down 설정 (GPIOB_CRL 레지스터 이용)

```
*((volatile unsigned int *)0x40010C00) &= ~0x0000000F; // 0번핀 관련 필드만 0값으로 초기화
*((volatile unsigned int *)0x40010C00) |= 0x00000008; // 0번핀 input with pull-up/pull-down 모드 설정
```

이번 주 예비 발표 영상 시청

- PLATO에 업로드 된 조별 발표 동영상 각자 시청 (시청 체크하여 성적에 반영)
- 시청 기한 ~09.28

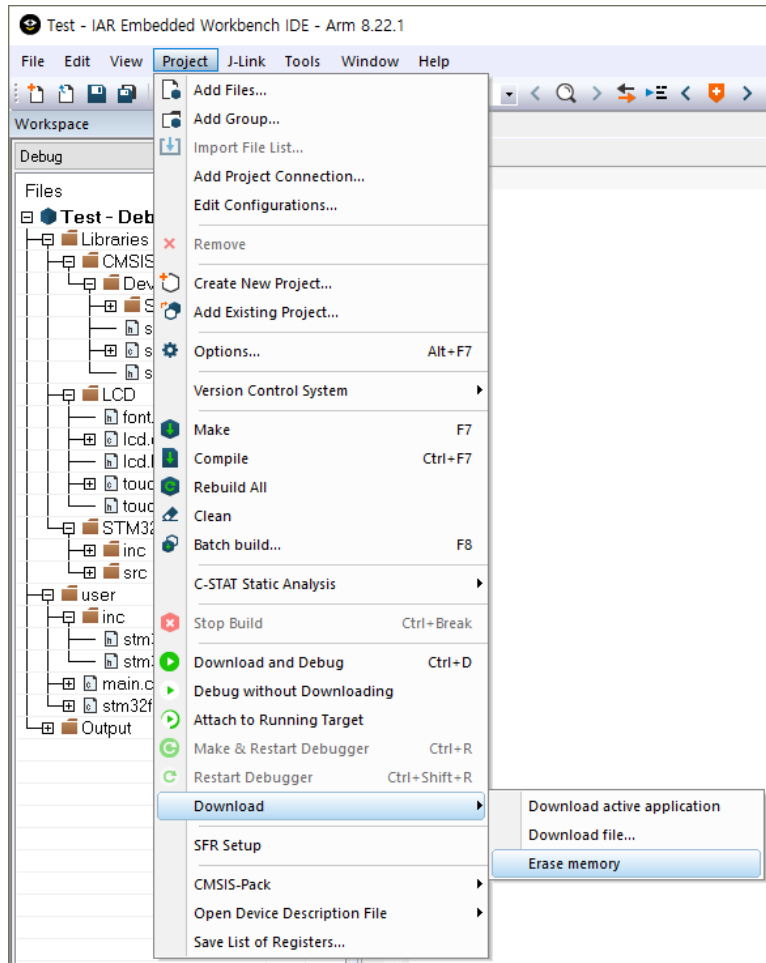
다음 주 (9월 29일) 실험 휴강

- 명절 잘 보내세요~

10월 6일 (2조)	10월 13일 (2조)	10월 27일 (2조)	11월 3일 (2조)	11월 10일 (2조)	11월 17일
7조 10조	11조 6조	1조 8조	3조 2조	5조 4조	9조

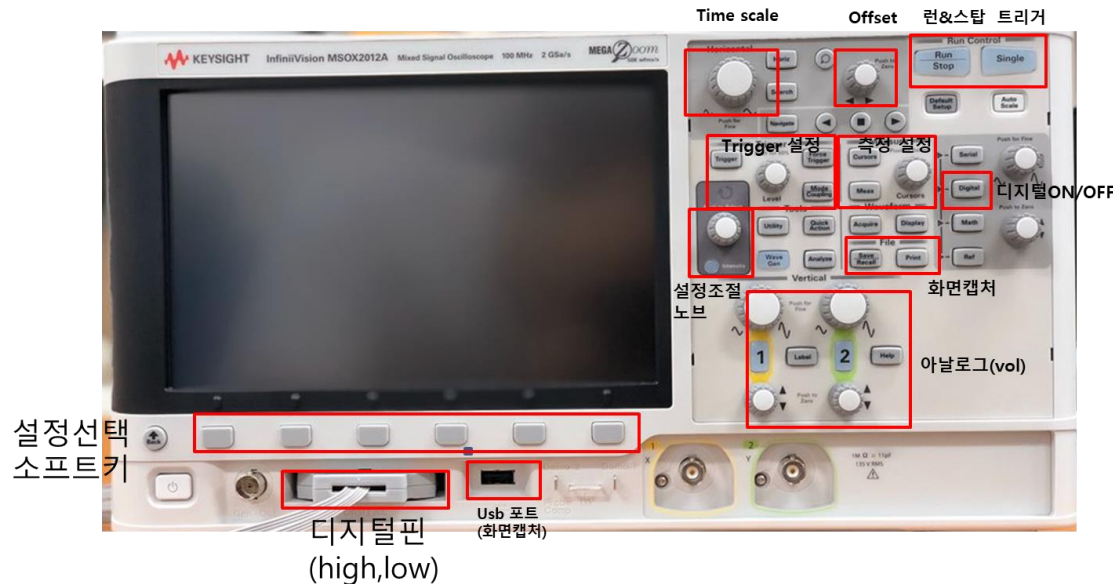
예비 발표 영상 제출 기한 재 안내: 10월 5일 23시 59분까지, 6, 11조 영상 제출 해주세요!

월 28	화 29	수 30	목 10월 1일	금 2	토 3	일 4
5 7주차 예비 실험 제출!	6 6주차 실험	7	8	9	10	11
12 6주차 결과보고서 제출 8주차 예비 실험 제출	13 7주차 실험	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30	31	11월 1일



간혹 이전에 잘못 짠 코드를 올린 것이 그대로 레지스터에
남아서 원하는 대로 동작이 안 될 경우가 있습니다.
왼쪽과 같이 memory를 지운 뒤 작성한 코드를 올리시기 바
랍니다.

오실로스코프 사용법 실습 및 3주차 실험 검사



각 조 3주차 실험 검사하는 동안
보드 뒷쪽 핀 소켓에 핀으로
오실로스코프로 조이스틱 중 버튼 한 개와 LED 한 개 pin을 연결하여
조이스틱 누르고 LED 켜질 때까지의 시간 측정하기



Pusan
National
University



September 22, 2020

조교
김민재

min7ae@gmail.com

임베디드 시스템 설계 및 실험

화요일 분반

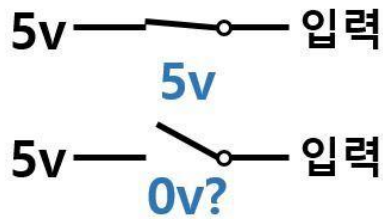
4주차

- 풀업 / 풀다운 저항
- 스캐터 파일의 이해 및 플래시 프로그래밍
- 릴레이 모듈의 이해 및 임베디드 펌웨어를 통한 동작
- 센싱에서 폴링 방식의 이해

Contents

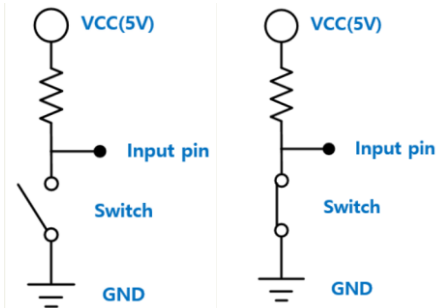
실험 내용

• 플로팅 (Floating)



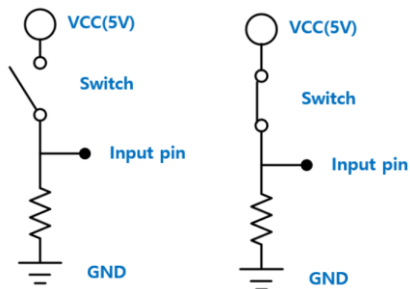
- 전압을 High / Low 로 보기 힘든 상태
- 아주 작은 노이즈만으로도 High와 low 사이를 빠르게 이동하여 오동작 유발
- 따라서 풀업 저항 또는 풀다운 저항을 사용

• Pull Up



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

• Pull Down



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

Scatter File 이란?

컴퓨터인터넷IT용어대사전

분산 적재

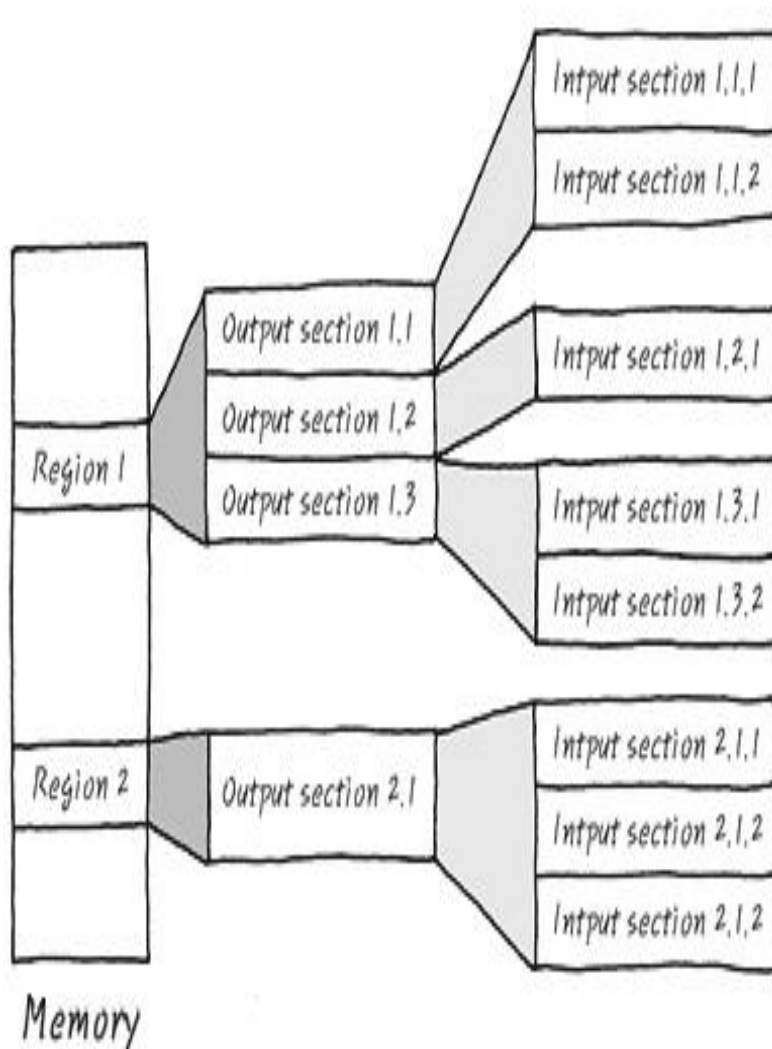
[scatter loading]

꺼내기의 한 형식으로 판독 모듈의 제어 섹션을 주기억 장치 가운데 각각의 장소에 적재하는 것.

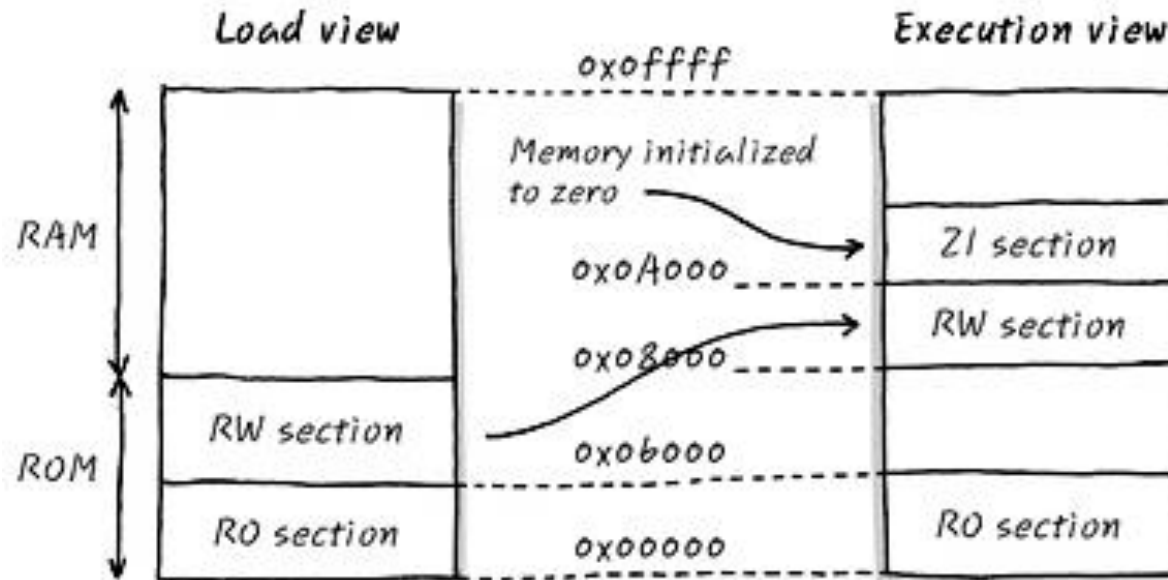
실행시킬 바이너리 이미지가 메모리에 로드될 때,
바이너리 이미지의 어떤 영역이 어느 주소에 어느
크기만큼 배치되어야 할 지 작성한 파일.

Scatter File이 필요한 이유?

- 1) 바이너리의 여러 부분을 각각 별개의 메모리 영역에 로드해야 될 때
- 2) 자주 사용되거나 빠른 실행을 요구하는 코드영역을 접근 시간이 빠른 메모리에 우선 배치하도록 설정할 수 있음.

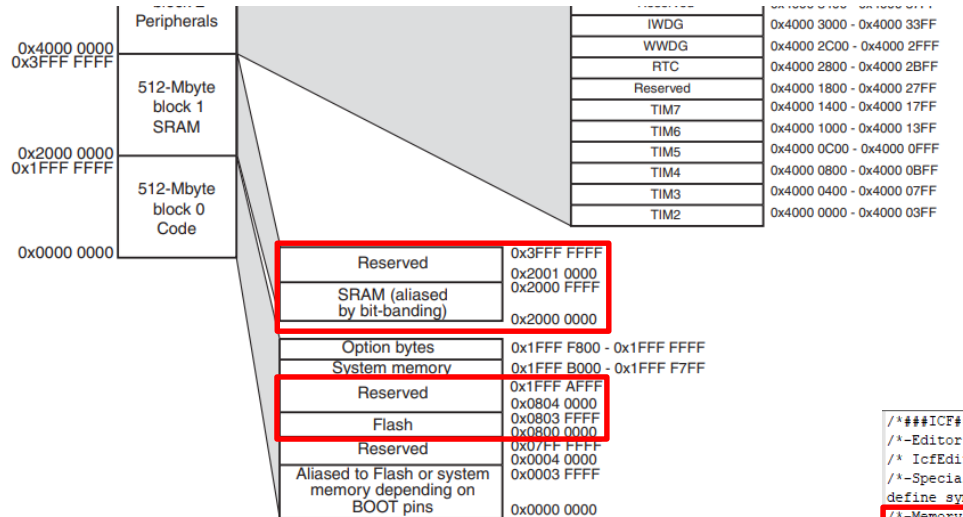


- **Input Section**
 - RO (code, constant data)
 - RW (global data)
 - ZI(zero initialized)
 - 중 하나의 속성을 갖는 집합
- **Output Section**
 - Input section들 중에 같은 속성을 갖는 것들을 묶어 놓은 것
- **Region**
 - Output section을 묶어 놓은 것



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

IAR EW는 .icf 파일을 스캐터 파일로 이용



```

/****ICF**** Section handled by ICF editor, don't touch! ****/
/*-Editor annotation file-*/
/* IcfEditorFile="$TOOLKIT_DIR$\config\ide\IcfEditor\cortex_vl_0.xml" */
/*-Specials-*/
define symbol __ICFEDIT_intvec_start__ = 0x08000000;
/*-Memory Regions-*/
define symbol __ICFEDIT_region_ROM_start__ = // TODO
define symbol __ICFEDIT_region_ROM_end__ = // TODO
define symbol __ICFEDIT_region_RAM_start__ = // TODO
define symbol __ICFEDIT_region_RAM_end__ = // TODO
/*-Sizes-*/
define symbol __ICFEDIT_size_cstack__ = 0x1000;
define symbol __ICFEDIT_size_heap__ = 0x1000;
/**** End of ICF editor section. ****ICF****/

define memory mem with size = 4G;
define region ROM_region = mem:[from __ICFEDIT_region_ROM_start__ to __ICFEDIT_region_ROM_end__];
define region RAM_region = mem:[from __ICFEDIT_region_RAM_start__ to __ICFEDIT_region_RAM_end__];

define block CSTACK with alignment = 8, size = __ICFEDIT_size_cstack__ { };
define block HEAP with alignment = 8, size = __ICFEDIT_size_heap__ { };

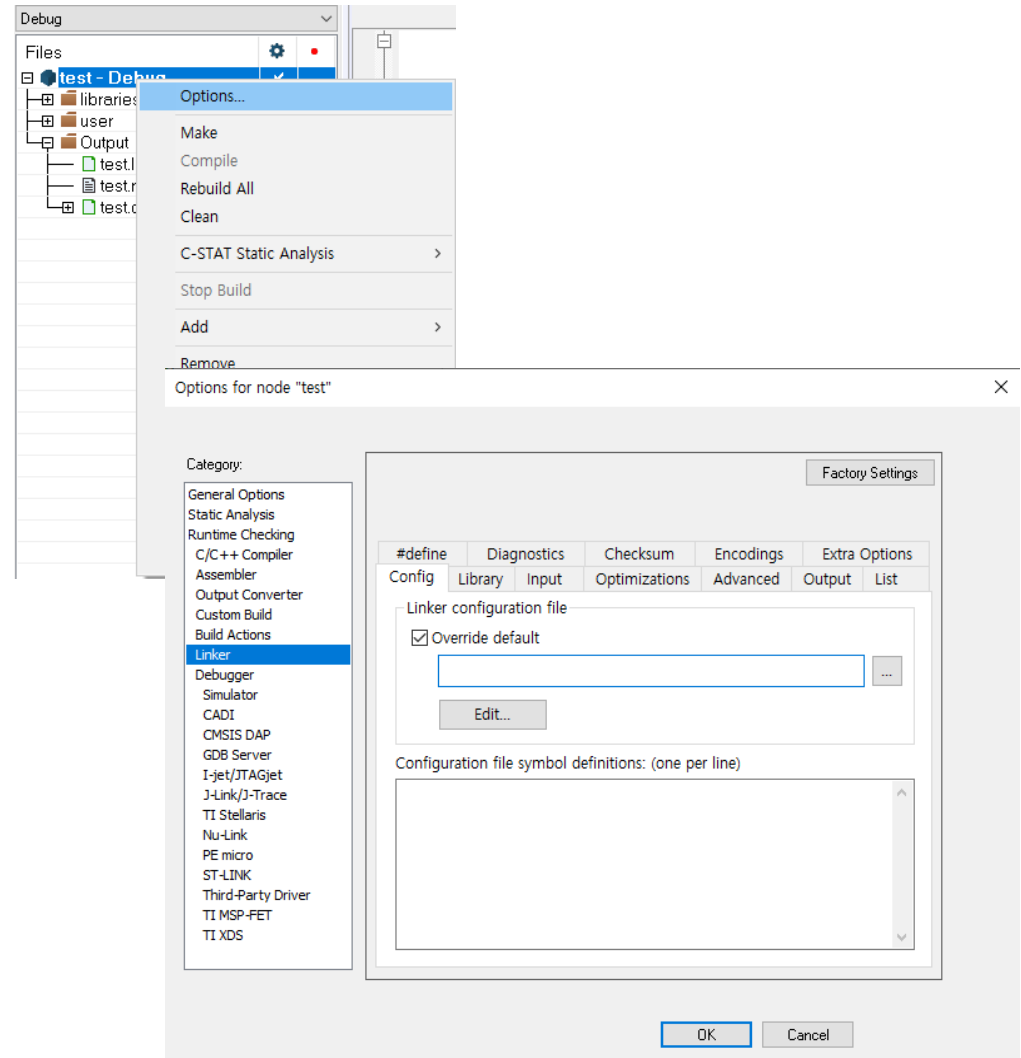
initialize by copy { readwrite };
do not initialize { section .noinit };

place at address mem:__ICFEDIT_intvec_start__ { readonly section .intvec };

place in ROM_region { readonly };
place in RAM_region { readwrite,
block CSTACK, block HEAP };
    
```

원하는 만큼 메모리 영역을 할당 가능

- 작성한 *.icf 파일
- project 오른쪽 클릭 – Options..
- Linker – Config – Override default
- ... 을 눌러 업로드

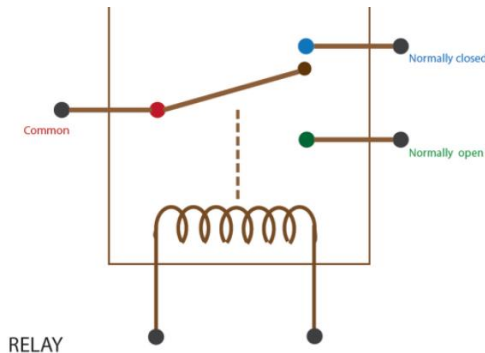
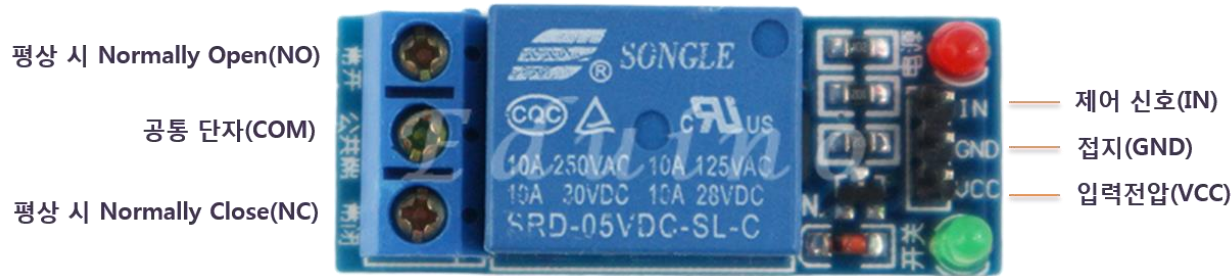


Interrupt

- Hardware의 변화를 감지해 외부로부터 전기신호 입력을 CPU가 알아채는 방법
- CPU 마다 다른 방식으로 동작
- 진행 중인 일을 잠시 멈추고 인터럽트 처리 루틴을 실행하여 신호를 처리함

Polling

- Hardware의 변화를 지속적으로 읽어들이며 변화를 알아채는 방법
- 신호를 판단하기 위해 지속적으로 확인해야 함
- 다른 일을 하는 중에 신호를 읽을 수 없음



Relay Module

- : 릴레이를 제어하는 모듈
- : 전자기유도원리를 이용하여 스위치 역할
- : 릴레이에 신호를 가하면 출력 상태(ON/OFF)가 변경된다

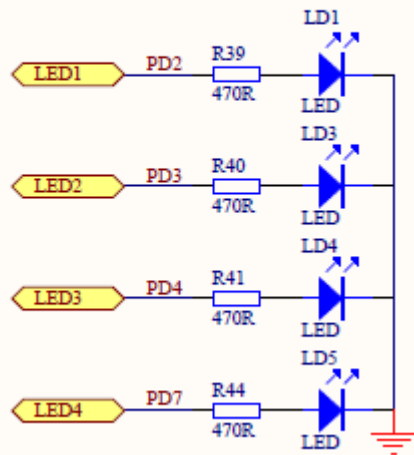
릴레이 모듈에 **3.3v** 전원 인가해서 사용 (5V 는 작동 안 할 수 있음)

COM은 제어 신호(IN)에 따라 NO 또는 NC로 붙는다

NO: 평소에 open, high 신호가 들어오면 close

NC: 평소에 close, high 신호가 들어오면 open

LED

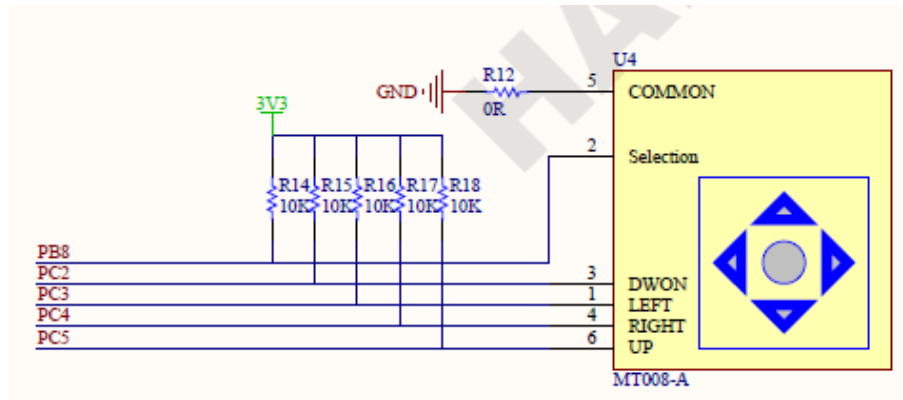
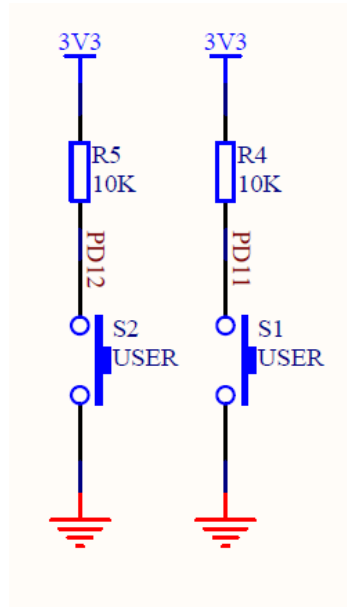


LED

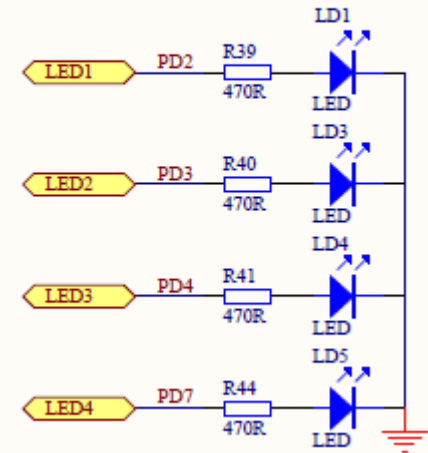
- 1 : PD2
- 2 : PD3
- 3 : PD4
- 4 : PD7

PD2, 3, 4, 7 LED 점멸 시 다음과 같은 Delay 함수 사용

```
void delay() {  
    int i;  
    for (i = 0; i < 10000000; i++) {}  
}
```



LED



User Button : PD11, PD12

Up : PC5, Down : PC2, Left : PC3, Right : PC4

릴레이 모듈 사용을 위한 GPIO : PC8, PC9

LED

- 1 : PD2
- 2 : PD3
- 3 : PD4
- 4 : PD7

- 실험 장비들을 연결 및 분리할 때 반드시 모든 전원을 끄고 연결해주세요.
- 장비사용시 충격이 가해지지 않도록 주의해주세요.
- 자리는 항상 깔끔하게 유지하고 반드시 정리 후 퇴실해주세요.
- 실험 **소스 코드와 프로젝트 폴더**는 **백업** 후 반드시 **삭제**해주세요.
- 장비 관리, 뒷정리가 제대로 되지 않을 경우 해당 조에게 감점이 주어집니다.

- **동작 중 케이블 절대 뽑지말것**
- **보드는 전원으로 USBPort나 어댑터(5V,1A)를 사용할것 (5V 5A 어댑터(비슷하게 생김)와 혼동하지 말 것, 사용시 보드가 타버림 -> 감점)**
- **디버깅 모드 중에 보드 전원을 끄거나 연결 케이블을 분리하지 말 것!!!**

- **-> 지켜지지 않을 시 해당 조 감점**

미션 ! 별도 미션지 참고

실험 검사

1. 정확한 장비 설정 유무 확인
2. 레지스터 및 주소 설정 이해 확인
3. 스캐터 파일, 릴레이 모듈 이해 확인
4. 오실로스코프 디지털 핀 사용법 이해

이번 주 실험 결과 보고서

- A. 이론부터 실습까지 전반적인 내용을 포함하도록 작성 (실험 과정 사진 찍으시면 좋아요)
- B. main.c, pdf로 변환한 결과보고서 파일
- C. 다음 실험시간 전까지 PLATO 제출

예비 발표 조는 발표 자료(영상) 만들어서
월요일 24시까지 조교 이메일로 제출

나가실 때, 만드신 코드 및 프로젝트 폴더는 모두 백업하시고 삭제해주세요.
다른 분반 파일은 만지지 마시고 조교에게 알려주세요.
자리 정리정돈 안 되어 있으면 **감점**합니다!!!