

앞	
1조	3조
5조	7조
9조	11조

앞	
2조	4조
6조	8조
10조	

주차	날짜	실험내용
1	09.02	분반편성 및 Overview
2	09.08	개발 환경 구축 및 개발 장비 교육
3	09.15	디버깅 툴(J-Link) 및 레지스터와 주소 제어를 통한 임베디드 펌웨어 개발
4	09.22	휴강 (추석)
5	09.29	스캐터 로딩 파일 및 플래시 메모리 이해
6	10.06	Polling 방식을 이용한 UART 통신 및 Clock control
7	10.13	Interrupt 방식을 활용한 GPIO 제어 및 UART 통신
8	10.20	휴강 (중간고사 없음)
9	10.27	Bluetooth 및 납땜
10	11.03	TFT-LCD 제어 및 ADC 구현
11	11.10	Timer 및 PWM 구현
12	11.17	DMA 구현
13~	11.24~12.24	텀 프로젝트 진행 (최종 검사일은 미정)
15~16	12.08~12.19	기말고사 (시험일은 미정)

차후에 변경 가능

예비 발표 방법 변경

- 발표 영상을 녹화하여 조교 이메일 (chrismail@naver.com) 로 제출
- 모든 학생은 조교가 PLATO에 업로드한 동영상 각자 시청 (시청 체크하여 성적에 반영)

제출 기한

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



Pusan
National
University



September 27, 2021

조교
김준명

chrismail@naver.com

임베디드 시스템 설계 및 실험

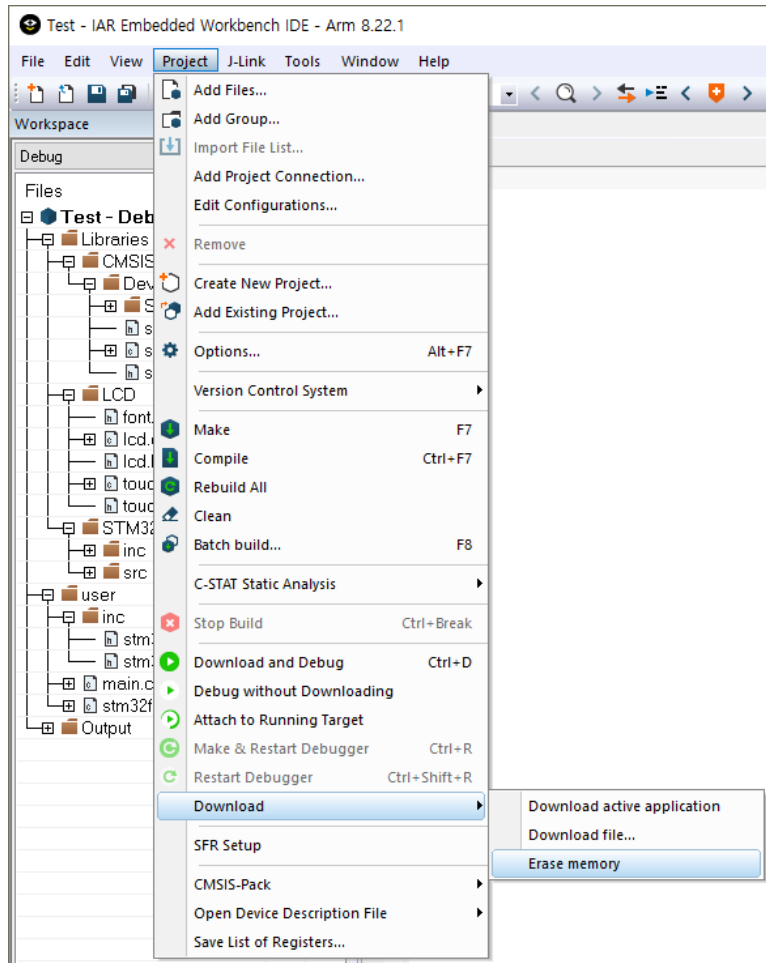
수요일 분반

4주차 Scatter file 이해

오실로스코프 사용법 실습



오실로스코프로 조이스틱 중 버튼 한 개와 LED 한 개 pin을 연결하여 조이스틱 누르고 LED 켜질 때까지의 시간 측정하기



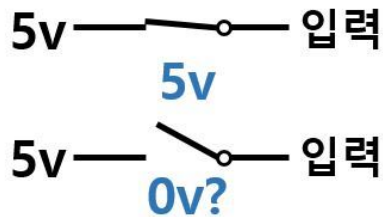
간혹 이전에 잘못 짠 코드를 올린 것이 그대로 레지스터에
남아서 원하는 대로 동작이 안 될 경우가 있습니다.
왼쪽과 같이 memory를 지운 뒤 작성한 코드를 올리시기 바
랍니다.

Contents

4주차 실험 내용

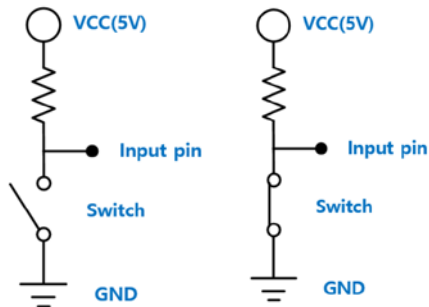
- 스케터 파일의 이해 및 플래시 프로그래밍
- 릴레이 모듈의 이해 및 임베디드 펌웨어를 통한 동작
- 센싱에서 폴링 방식의 이해

• 플로팅 (Floating)



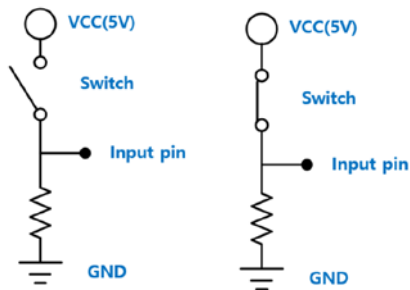
- 전압을 High / Low 로 보기 힘든 상태
- 아주 작은 노이즈만으로도 High와 low 사이를 빠르게 이동하여 오동작 유발
- 따라서 풀업 저항 또는 풀다운 저항을 사용

• 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]

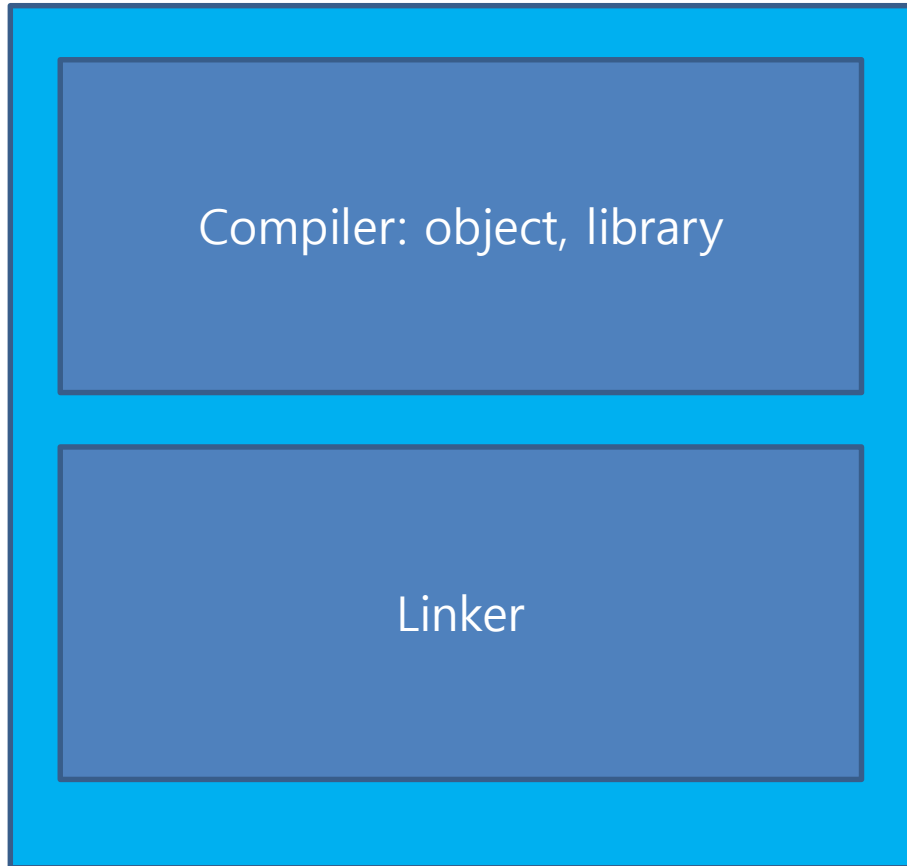
꺼내기의 한 형식으로 판독 모듈의 제어 섹션을 주기억 장치 가운데 각각의 장소에 적재하는 것.

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

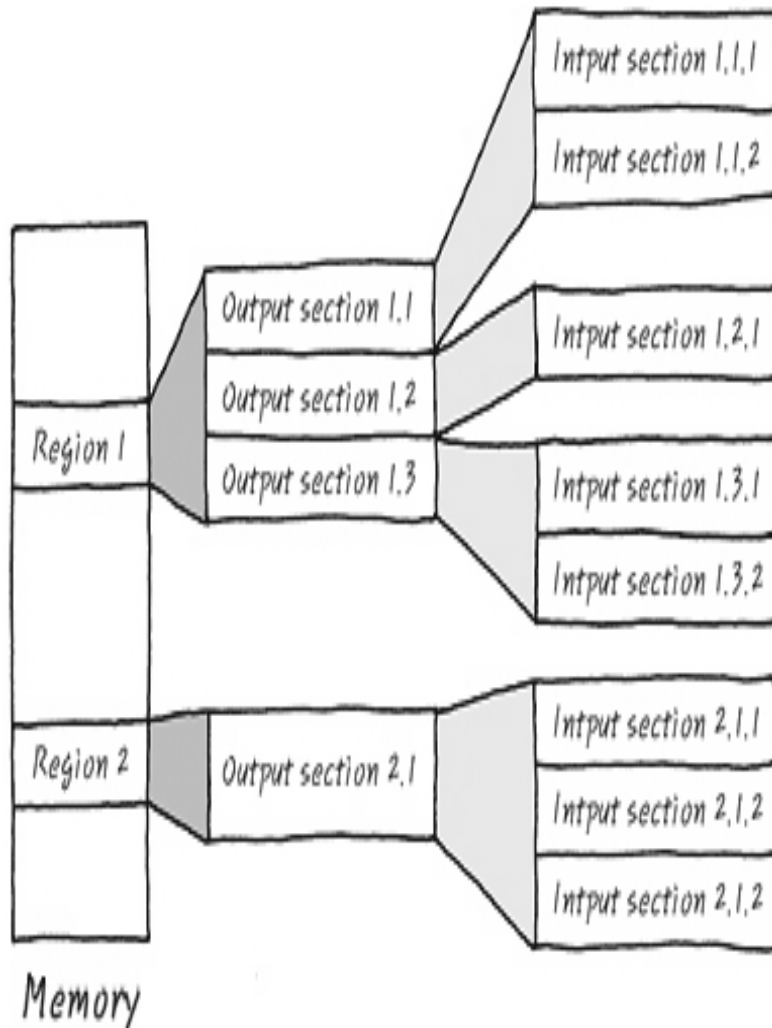
Scatter File이 필요한 이유?

- 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.

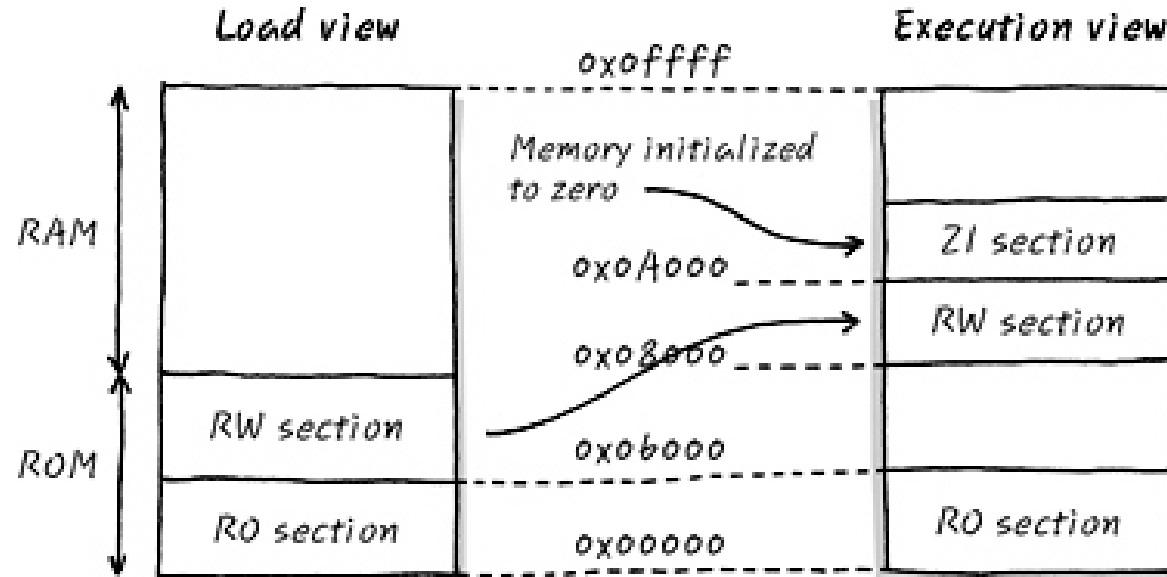
Compile(or Build)



*.icf: Linker configuration file

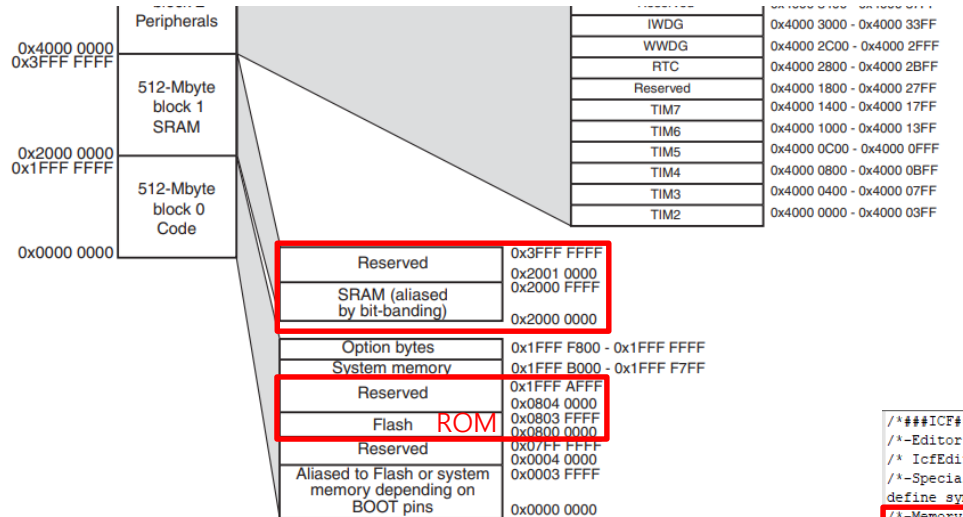


- **Input Section**
 - RO (code, constant data)
 - RW (global data)
 - ZI(zero initialized)
 - 중 하나의 속성을 갖는 집합
- **Output Section**
 - Input section들 중에 같은 속성을 갖는 것들을 묶어 놓은 것
- **Region**
 - Output section을 묶어 놓은 것



- 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__ { };

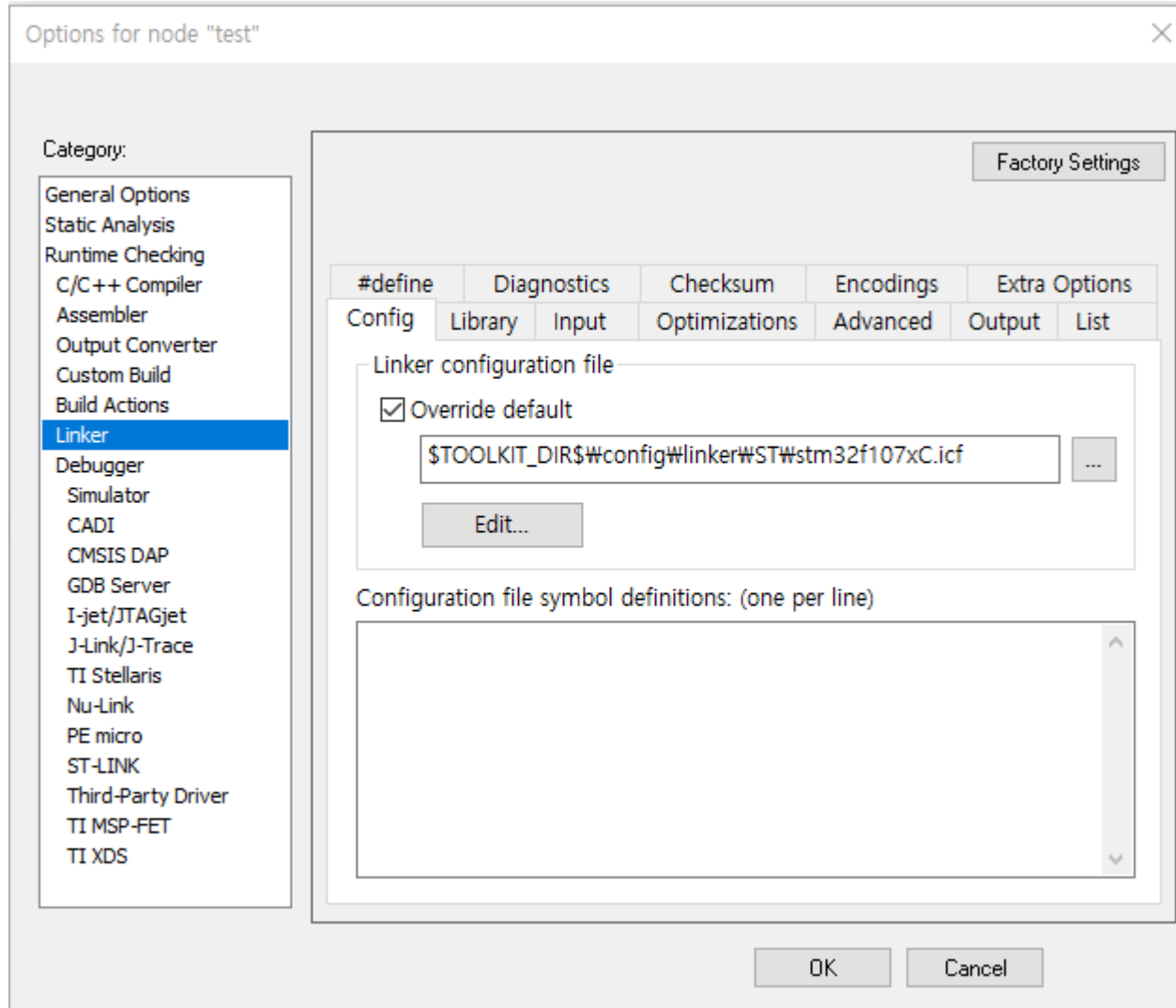
initialize by copy { readwrite };
do not initialize { section .noinit };

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

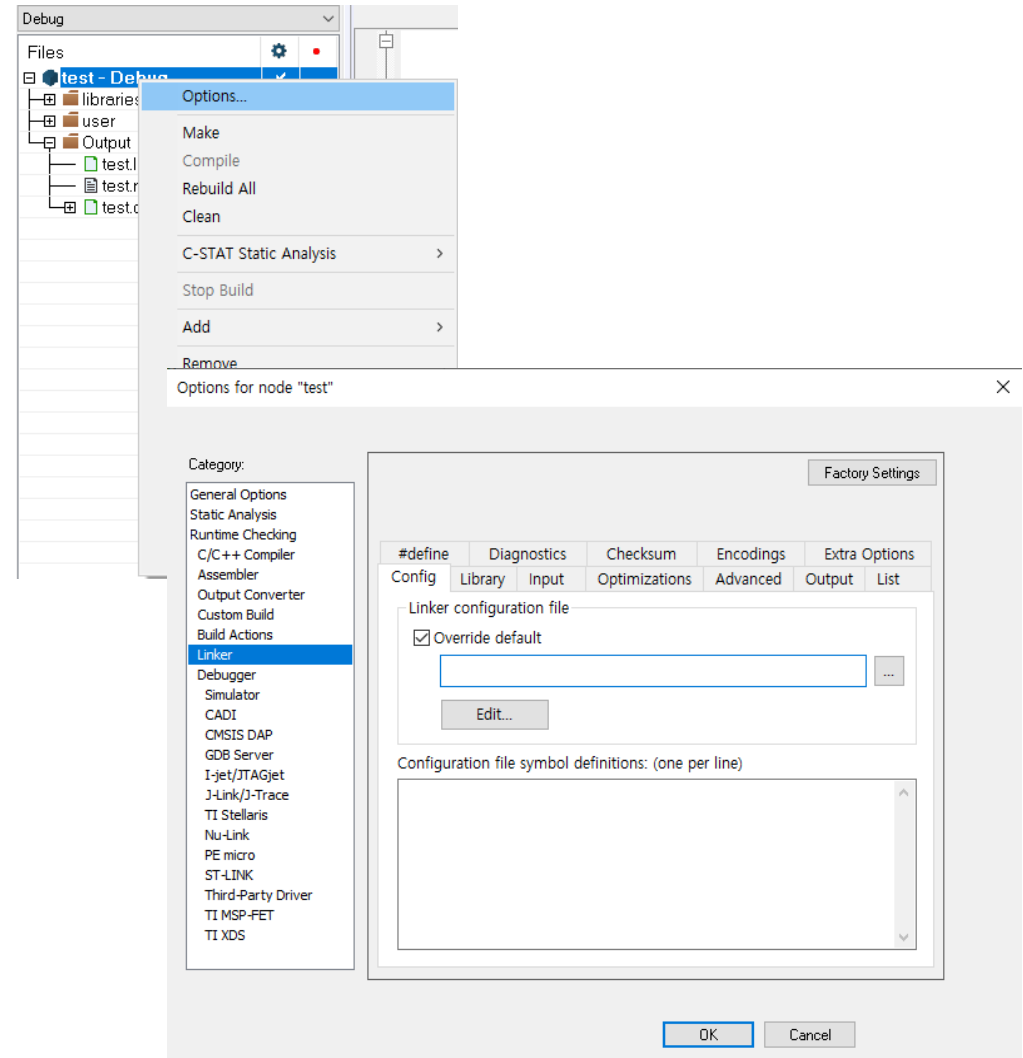
우리 보드 모델명인 "STM32F107VCT6" 에 대한 스캐터파일을 확인할 수 있음

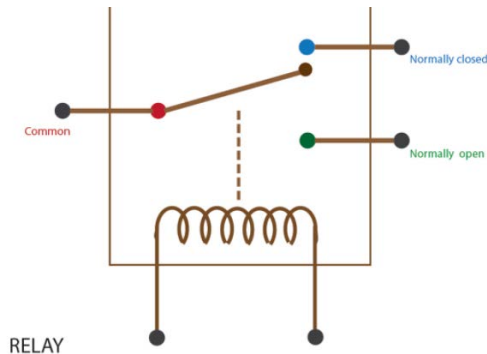


Features

- Core: ARM® 32-bit Cortex®-M3 CPU
 - 72 MHz maximum frequency, 1.25 DMIPS/MHz (Dhrystone 2.1) performance at 0 wait state memory access
 - Single-cycle multiplication and hardware division
- Memories
 - 64 to 256 Kbytes of Flash memory
 - 64 Kbytes of general-purpose SRAM
- Clock, reset and supply management
 - 2.0 to 3.6 V application supply and I/Os
 - POR, PDR, and programmable voltage detector (PVD)
 - 3-to-25 MHz crystal oscillator
 - Internal 8 MHz factory-trimmed RC
 - Internal 40 kHz RC with calibration
 - 32 kHz oscillator for RTC with calibration
- Low power

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





Relay Module

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

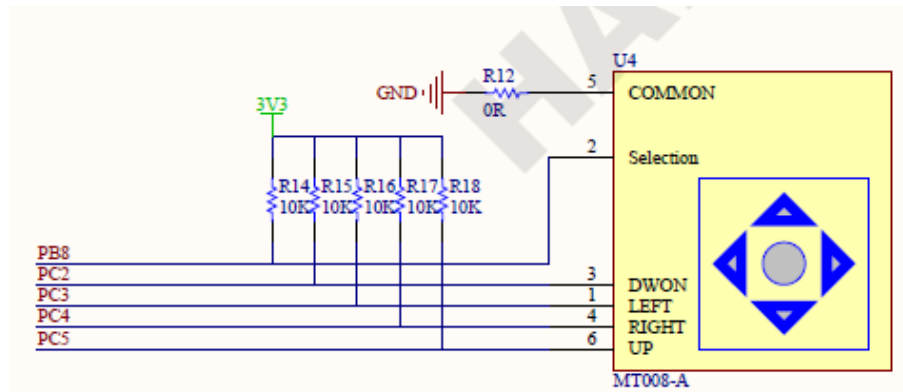
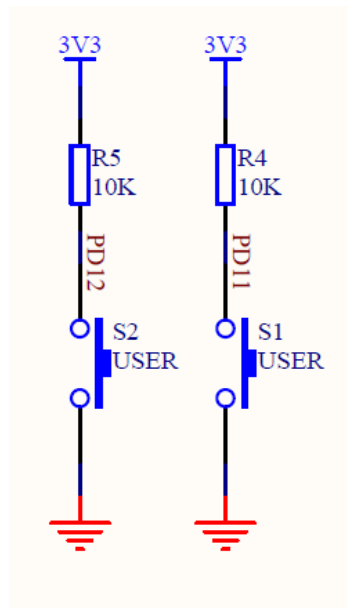
릴레이 모듈에 전원 인가해서 사용

COM은 제어 신호(IN)에 따라 NO 또는 NC로 붙는다

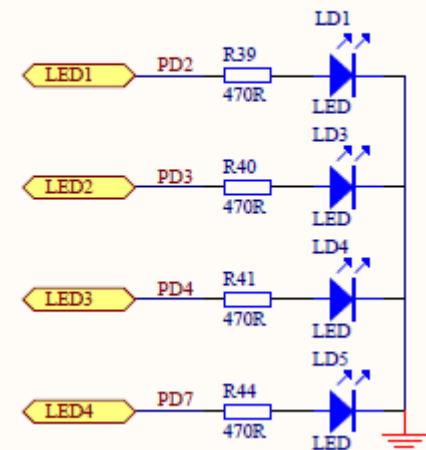
NO: 평소에 open, high 신호가 들어오면 close

NC: 평소에 close, high 신호가 들어오면 open

릴레이 모듈은 스위치 일뿐입니다. 릴레이 모듈이 켜졌다고 전원이 나오진 않아요



LED



4주차 실험 릴레이 모듈 사용을 위한 GPIO 자유선택

단, 이미 default 사용중인지 주의해서 고르세요

- 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

릴레이 모듈과 연결된 pin 을 set 한 뒤 delay() 함수 호출하고 다시 reset 하는 방식으로 구현

```
int i=8000000;  
while(i>0)  
{  
  i--;  
}
```

1초 딜레이

For문으로 사용시 컴파일러상 계산 문제인지 1초 초과

실험 검사

오늘 검사 받을 수 있는 조는 오늘 받고 못 받는 조는 따로 미션 수행 후 다음 주 수업 시작할 때 검사

이번 주 실험 결과 보고서 및 소스 코드

- A. 이론부터 실습까지 전반적인 내용을 포함하도록 작성 (실험 과정 사진 찍으시면 좋아요)
- B. 다음 실험시간 전까지 PLATO 제출 (보고서, 동작 영상)
- C. 소스 코드는 직접 작성 및 수정한 파일만 제출

예비 발표 조는 발표 자료(영상) 만들어서
화요일 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!!!