

앞		
1조	배준석 이승원 장원석 조성환	2조
		나상진 송민재 사르센바예바 누리야 이슬람 살리흐
3조	김 에 밀 박재홍 이민경 최지광	4조
		박준형 이경원 이성훈 조유진
5조	강등원 김이경 윤전형 조주은	6조
		박지환 이창욱 이한홍 차기은
7조	이영민 강찬석 누가예바 알트나이 케네스 예라슬	8조
		김윤하 성가빈 이은진 조찬우
9조	이승호 주소미 조용진	10조
		이서진 이종민 진예규 리 알렉세이
11조	김범수 박정우 이승민 이종일	12조
		구지원 김현태 정재영

조장 선정 및 연락처 교환

단톡방 생성

예비 발표

- 해당 주차 실습 내용 및 이론적 배경 등을 약 10분 발표
- 발표 자료는 수업 하루 전(화요일) 23:59 까지 PLATO '예비 발표 자료실'에 PDF 업로드 (늦으면 감점)

발표 일정

9월 27일	10월 11일	10월 18일	11월 1일	11월 8일	11월 15일	11월 22일
3조, 11조	6조, 10조	5조, 2조	8조, 12조	4조, 9조	1조	7조

4주차 예비 발표 준비 내용

- Scatter file 정의 및 사용 목적
- Scatter file 구조 분석
- 버튼의 floating, Pull up, Pull down 방식의 차이
- 인터럽트 / 폴링 방식의 차이
- 릴레이 모듈



Pusan
National
University



September 15, 2023

조교
안현기

tykr0001@pusan.ac.kr

임베디드 시스템 설계 및 실험

수요일 분반

3주차 GPIO 제어

실험 목적

- 임베디드 시스템 설계의 기본 원리 습득
- 디버깅 툴 사용방법 습득 및 레지스터 제어를 통한 임베디드 펌웨어 개발

세부 목표

- 개발 환경 구축
- IAR Embedded Workbench에서 프로젝트 생성 및 설정
- Datasheet 및 Reference Manual을 참고하여 해당 레지스터 및 주소에 대한 설정 이해
- GPIO(general-purpose input/output)를 사용하여 LED제어
- 오실로스코프에 대한 이해와 DebugPin설정

Contents

IAR EW 설치
(실험실에서는 필요 없음)

IAR Embedded Workbench 설치

- 실험실에는 라이선스 및 프로그램이 이미 설치되어 있음
 - 개인 혹은 조별로 사용하기 위한 방법
 - <https://www.iar.com/kr/products/architectures/arm/iar-embedded-workbench-for-arm/iar-embedded-workbench-for-arm---free-trial-version/>
- > IAR Embedded Workbench for Arm 무료 버전 다운로드

IAR Embedded Workbench for Arm 무료 버전

평가판 라이선스는 무료이며 14일 동안 소프트웨어를 사용해 효율성과 사용 편의성을 평가할 수 있습니다. 다운로드를 클릭하면 라이선스를 얻기 위해 등록하라는 메시지가 표시됩니다.

14일 기간 제한의 평가용 라이선스와 관련하여 다음 사항에 유의하십시오:

- 평가용 라이선스 기간은 14일입니다.
- 제품 개발 또는 상업적 용도로 사용할 수 없습니다.
- 런타임 라이브러리의 소스 코드는 포함되어 있지 않습니다.
- C-RUN은 상수 데이터를 제외하고, 12Kbyte의 코드로 사이즈가 제한됩니다.
- 기술 지원이 제한적입니다.

다운로드

IAR Embedded Workbench 설치

- 아래 양식 모두 작성 후 제출

Register for Evaluation

Please complete the following registration in order to complete the activation of your product.

Product

IAR Embedded Workbench for Arm, v. 9.40, Evaluation Edition

First name *

Last name *

Job title

Email *

Phone *

 Extension:

Company *

Department / Division

- 수신한 메일에서 링크 클릭



noreply.www1@iar.com

나에게

Dear Developer,

Please confirm your web registration for the product

IAR Embedded Workbench for Arm, v. 9.40, Evaluation Edition

using this link

<https://register.iar.com/confirm?lang=en&key=2ce2a944-50ce-4dbd-9a26-7351d3c60309>

Unconfirmed registrations are erased from our system after 14 days.

You cannot reply to this email. Please use the Contact page on our website (<http://www.iar.com/contact/>) if you have any comments or questions.

Best regards,

IAR

IAR Embedded Workbench 설치

- 라이선스 번호 확인
- 설치파일 다운로드 후 실행

Registration Complete

Thank you for your registration!


You have been assigned the following license number:

[Redacted license number]

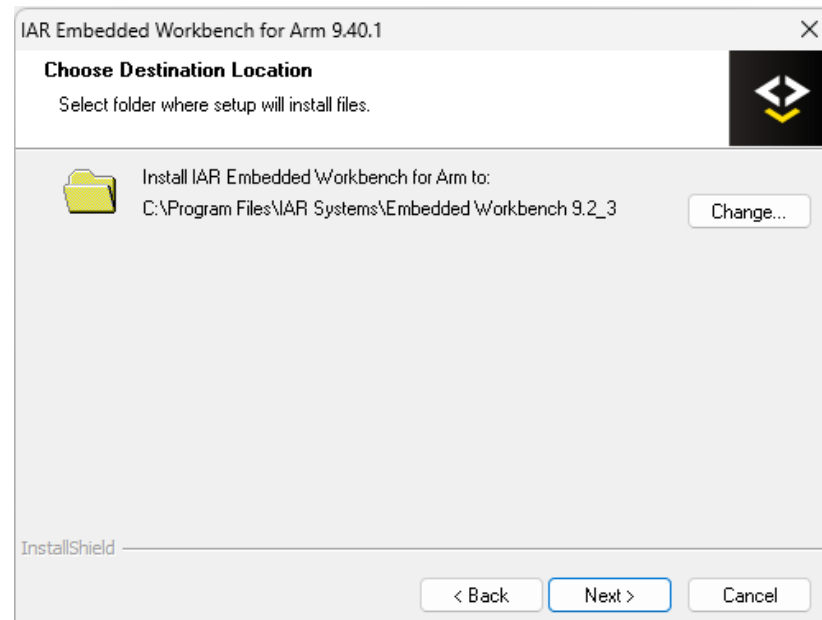
When you launch the application for the first time, you will be asked to specify this license number in the IAR License Manager. If the IAR License Manager does not pop-up automatically you can start it either from the Windows Start menu or from the Help menu in your IAR product.

[Download software](#) (Windows, 1.6 GB)

IAR

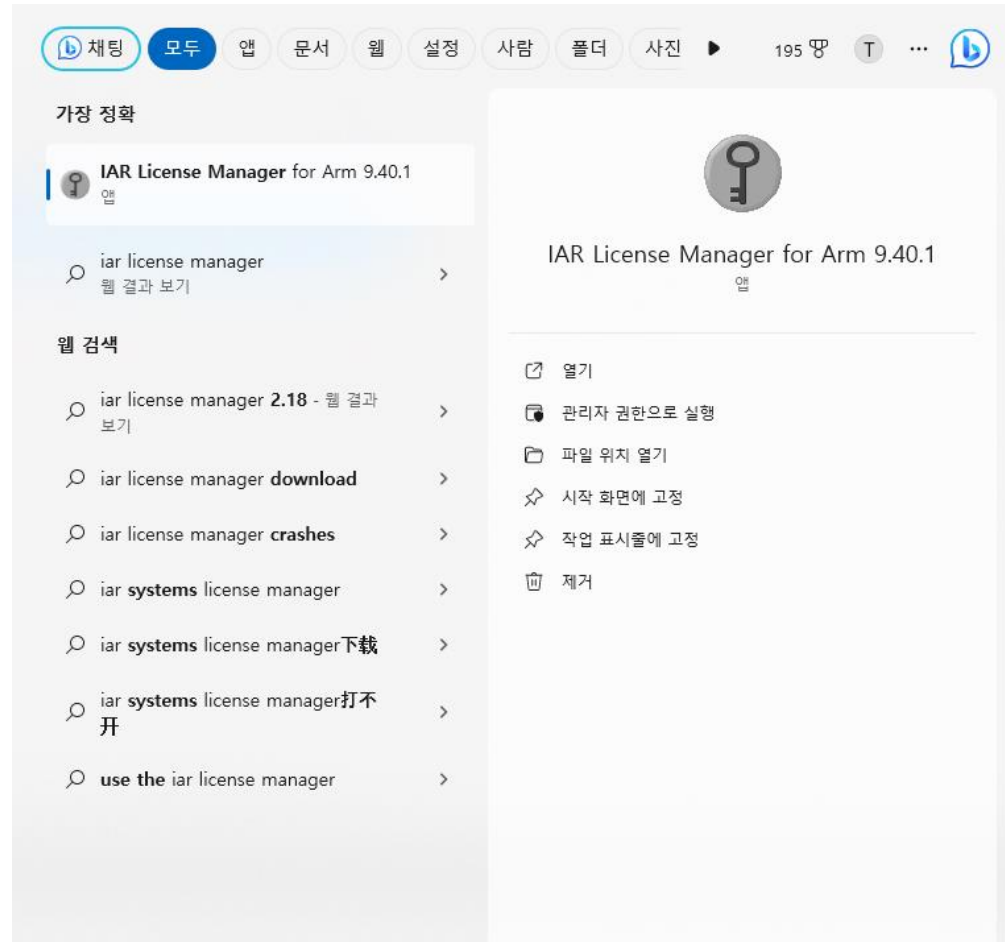
 EWARM-9401-63915

- IAR 툴 및 모든 드라이버 기본 경로로 설치



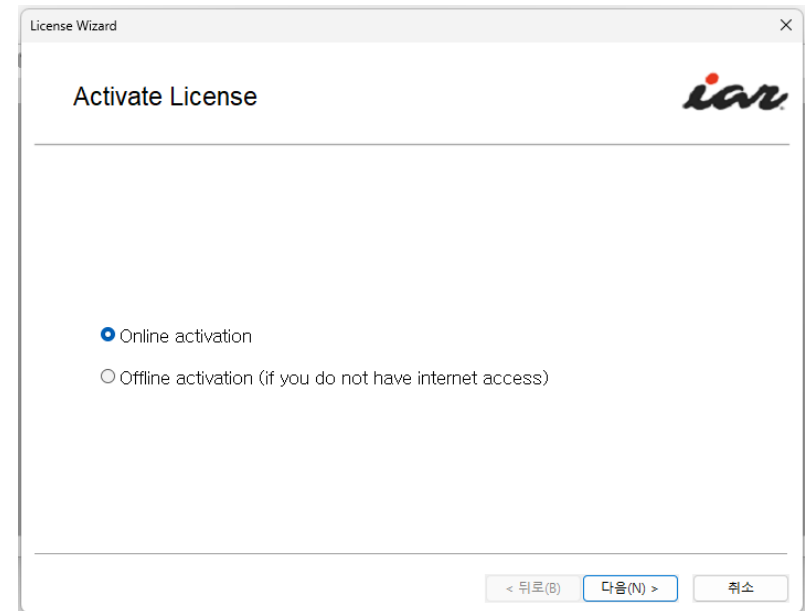
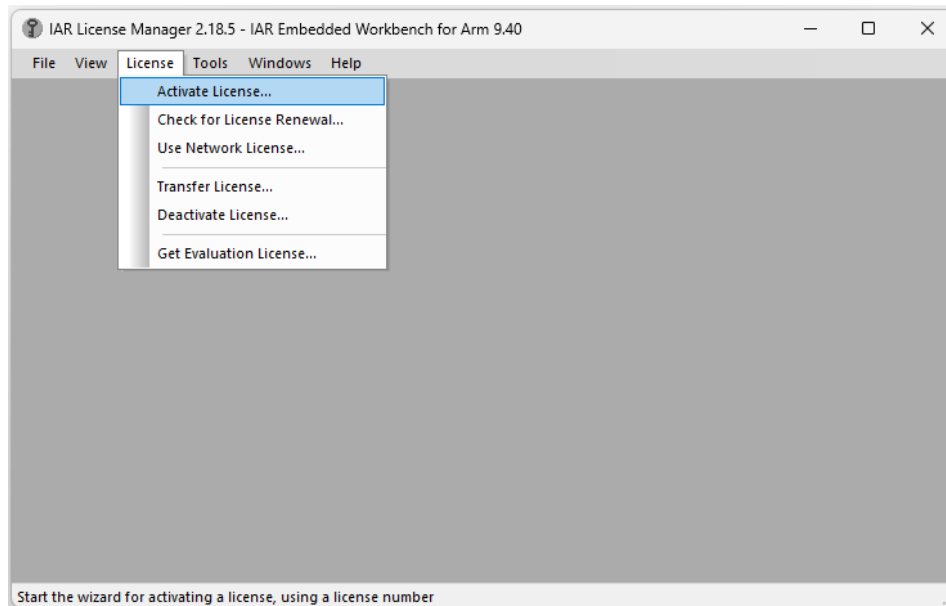
IAR Embedded Workbench 설치

- 설치된 IAR License manager for Arm 9.40.1 실행



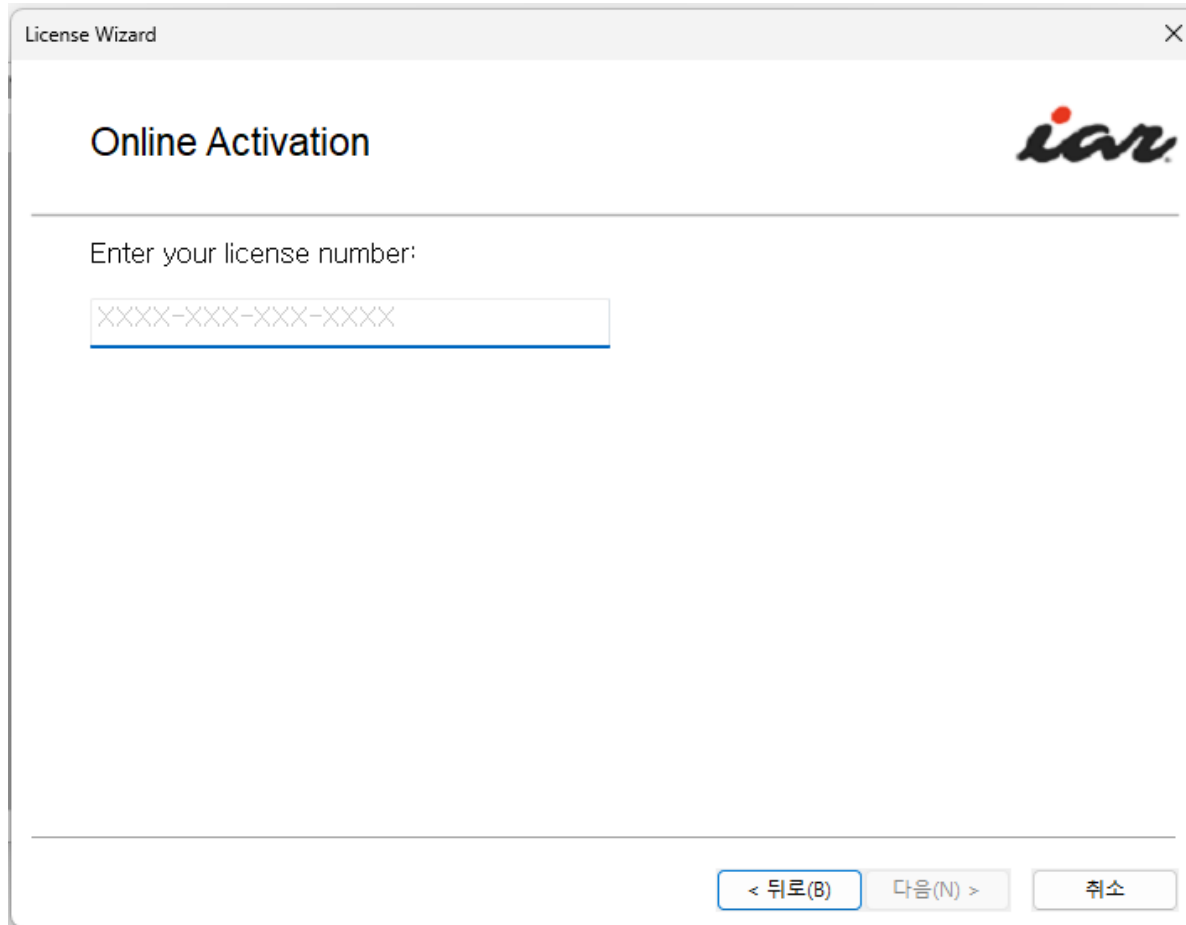
IAR Embedded Workbench 설치

- License 탭에서 Activate License 클릭
- Online activation 선택 후 다음 클릭



IAR Embedded Workbench 설치

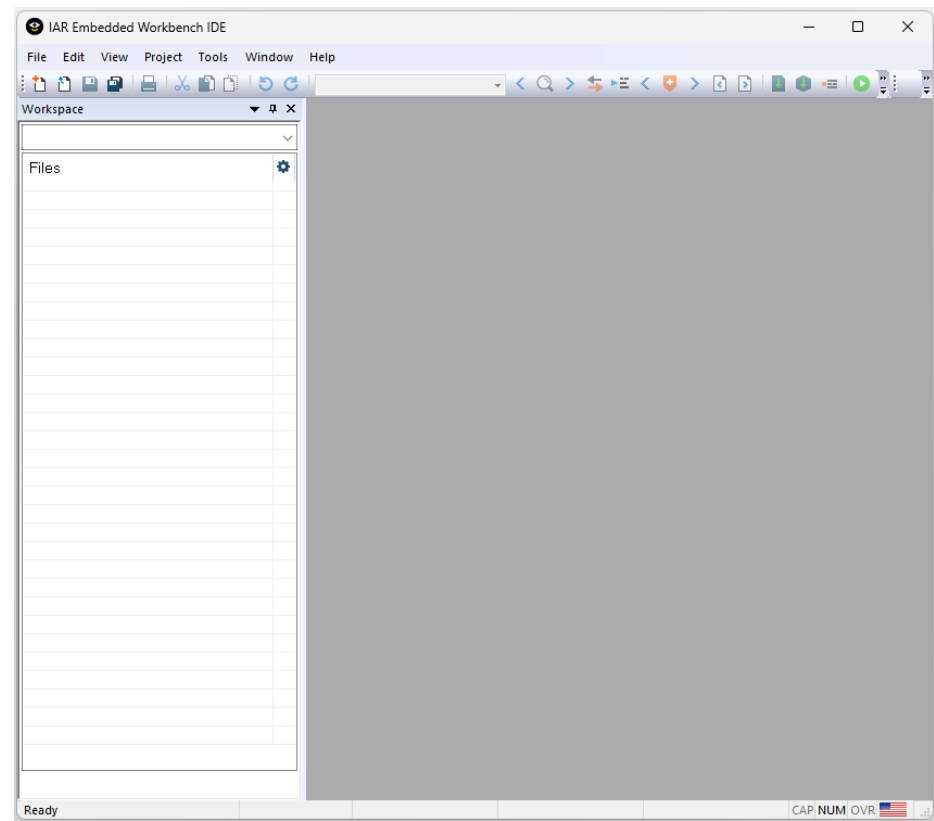
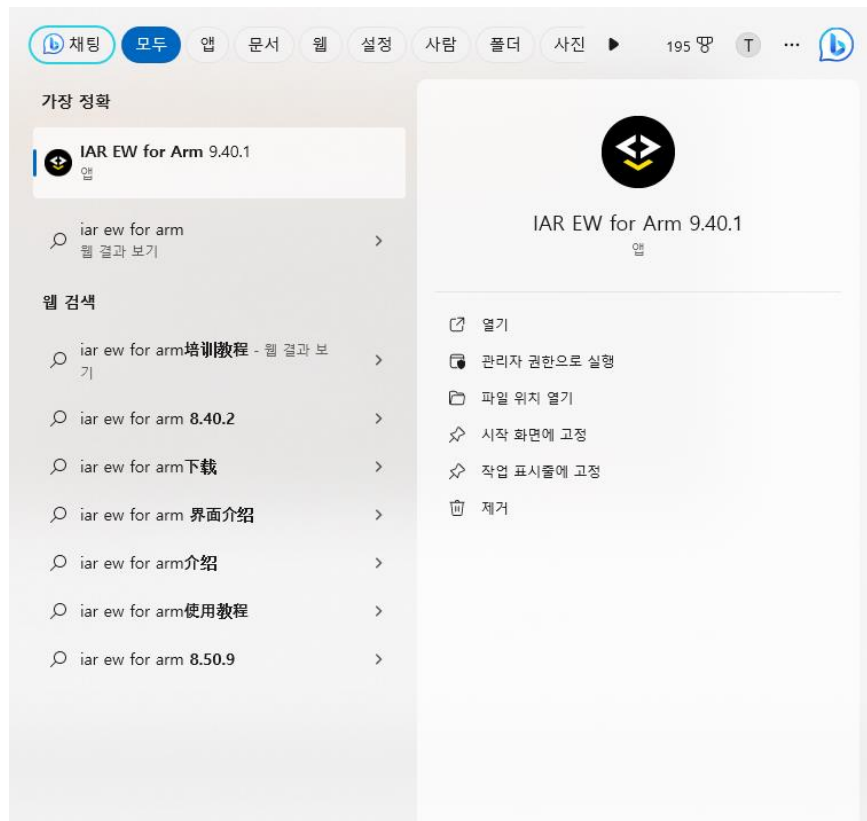
- 라이선스 번호 입력하여 등록



The image shows a screenshot of the 'License Wizard' window for IAR Embedded Workbench. The window has a title bar with 'License Wizard' and a close button. The main content area is titled 'Online Activation' and features the IAR logo in the top right corner. Below the title, there is a text prompt 'Enter your license number:' followed by a text input field containing the placeholder 'XXXX-XXX-XXX-XXXX'. At the bottom of the window, there are three buttons: '< 뒤로(B)' (Back), '다음(N) >' (Next), and '취소' (Cancel).

IAR Embedded Workbench 설치

- IAR EW for Arm 실행



IAR Embedded Workbench 설치

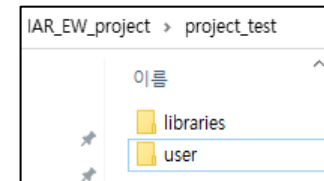
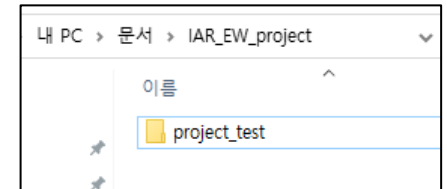
- trial 버전과 실험실에 설치된 버전은 서로 프로젝트 호환이 안 되므로 프로젝트를 새로 만들고 코드를 복사해야 함

Contents

프로젝트 및 실험 장비 설정

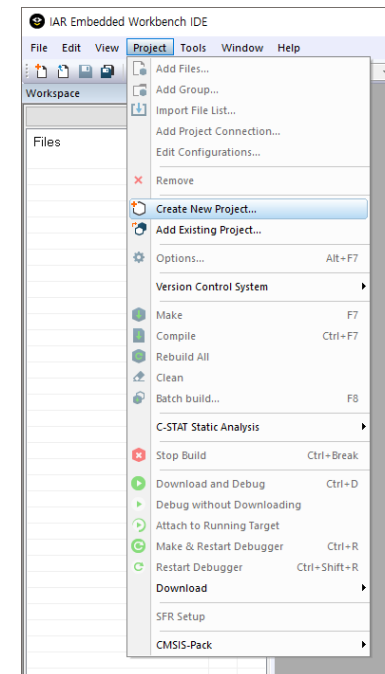
프로젝트 생성 및 설정

- 원하는 경로에 원하는 이름으로 폴더 생성 (ex: project_test)
 - 단, 한글경로는 포함되지 않도록 유의
- project_test/ 폴더에 실험 제공 파일 (CoreSupport, libraries, user 등) 복사



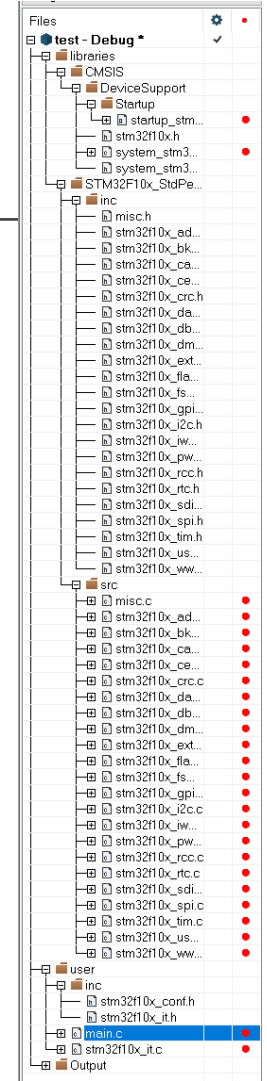
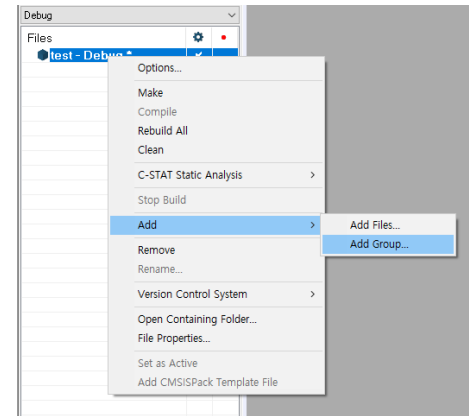
프로젝트 생성 및 설정

- IAR Embedded Workbench IDE 실행
- Project – Create New Project
- Empty project – OK
- 생성한 project_test 폴더에 원하는 프로젝트 명 (ex: test) 으로 저장



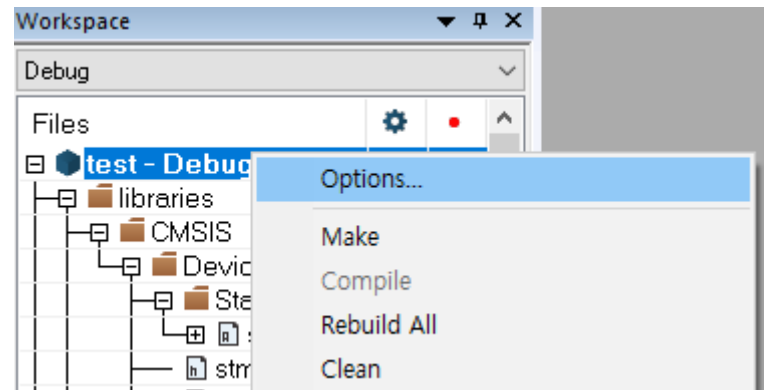
프로젝트 생성 및 설정

- 프로젝트 오른쪽 클릭 - Add - Add Group으로 project_test 폴더의 구조와 같이 만들기
 - 필수는 아니지만 보기 좋게 하기 위함
 - 하나하나 추가해야 함
- 프로젝트 오른쪽 클릭 - Add - Add Files으로 .c, .h, .s 파일 모두 추가
 - 이걸 필수



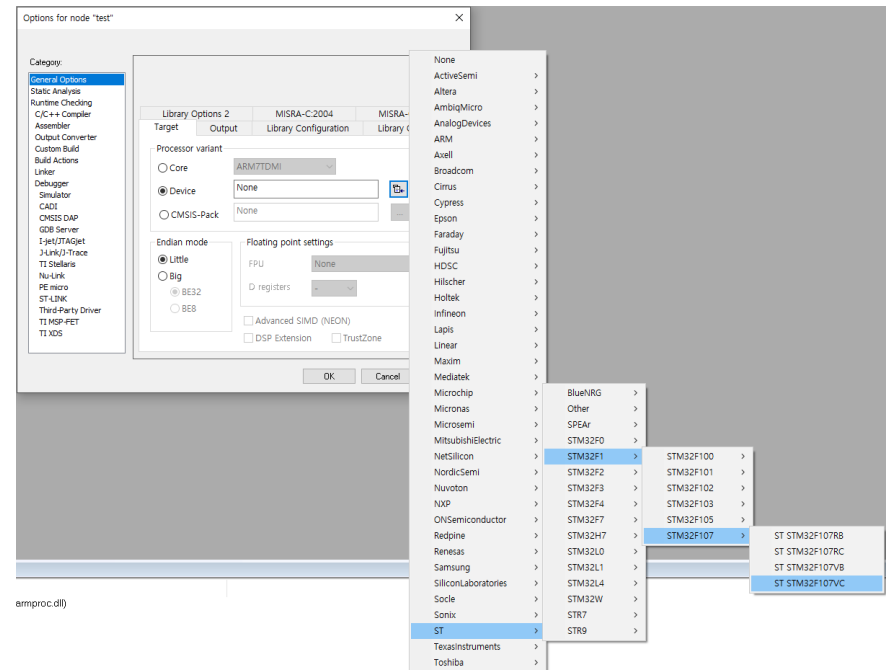
프로젝트 생성 및 설정

- 프로젝트 오른쪽 클릭 - Options



프로젝트 생성 및 설정

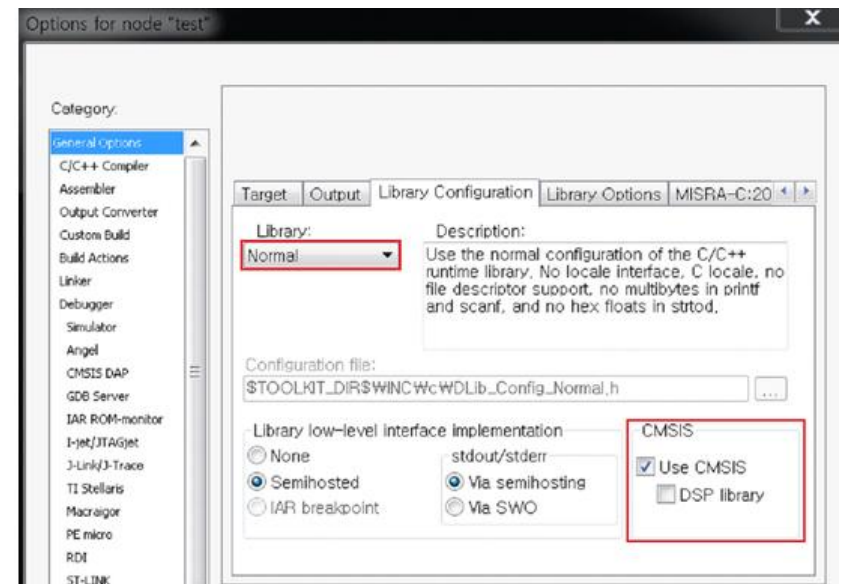
- General Options – Target 탭 – Device
 - ST – STM32F1 – STM32F107 – ST STM32F107VC



프로젝트 생성 및 설정

• General Options – Library Configuration 탭

- Library
 - Normal
- CMSIS
 - Use CMSIS

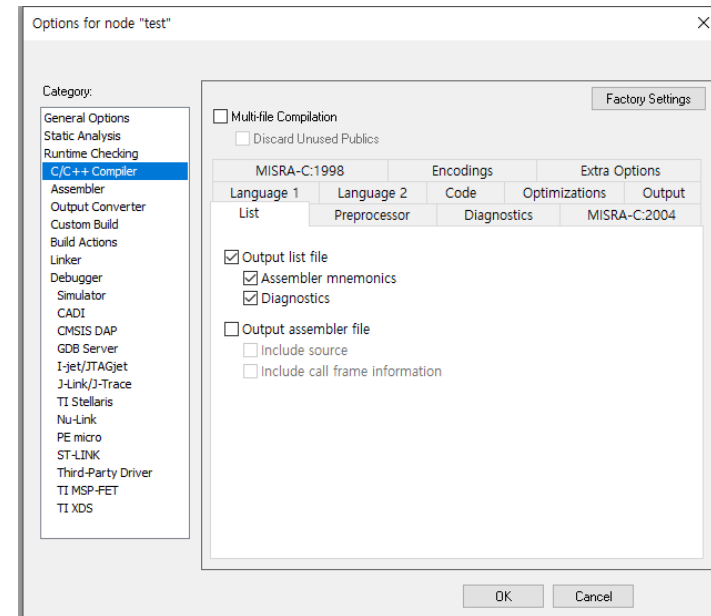


프로젝트 생성 및 설정

- Runtime Checking – C/C++ Compiler

- List 탭

- Output list file
 - Assembler mnemonics
 - Diagnostics



프로젝트 생성 및 설정

• Runtime Checking – C/C++ Compiler

• Preprocessor 탭

- Additional include directories: (one per line)

\$PROJ_DIR\$\libraries\CMSIS\DeviceSupport

\$PROJ_DIR\$\libraries\CMSIS\DeviceSupport\Startup

\$PROJ_DIR\$\libraries\STM32F10x_StdPeriph_Driver_v3.5\inc

\$PROJ_DIR\$\libraries\STM32F10x_StdPeriph_Driver_v3.5\src

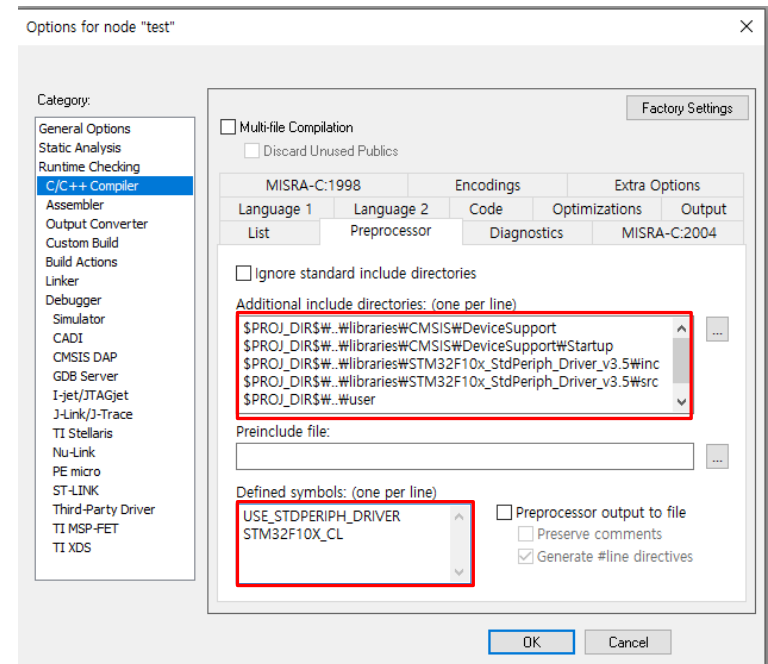
\$PROJ_DIR\$\user

\$PROJ_DIR\$\user\inc

- Defined symbols: (one per line)

USE_STDPERIPH_DRIVER

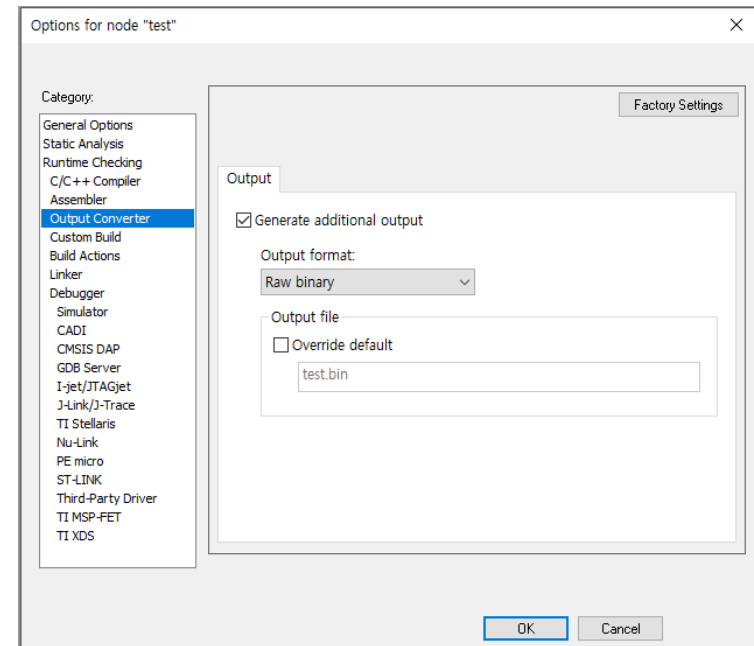
STM32F10X_CL



프로젝트 생성 및 설정

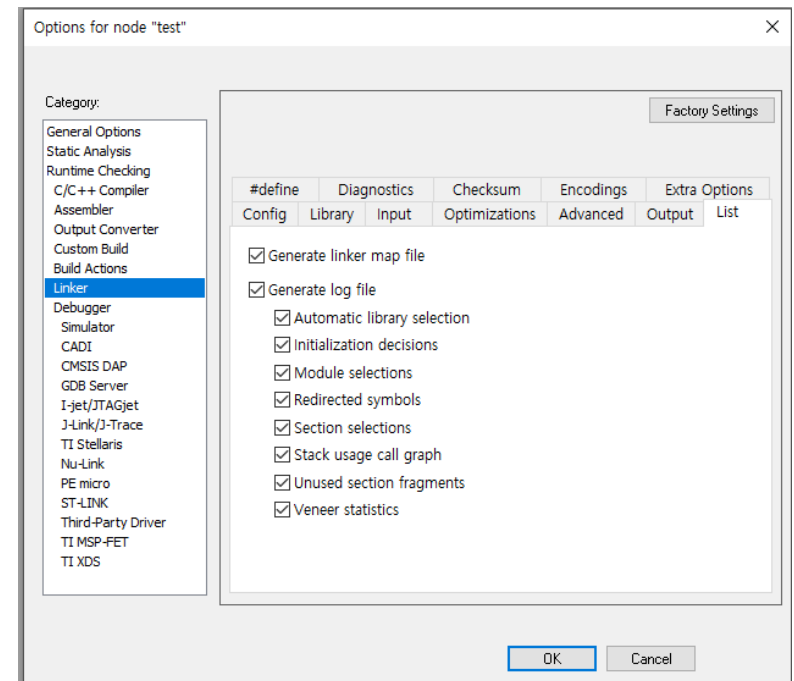
- Runtime Checking – Output Converter

- Generate additional output
 - Output format – Raw binary



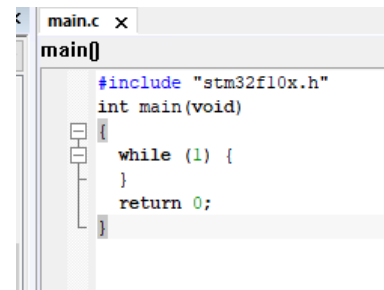
프로젝트 생성 및 설정

- Runtime Checking – Linker – List 탭
 - 모두 체크



프로젝트 생성 및 설정

- user/main.c에 원하는 코드 작성
- F7 (Make) 로 컴파일
- 첫 컴파일 시 저장하는 창이 뜬
 - project_test/ 폴더에 프로젝트 명으로 저장



```
main.c x
main()
#include "stm32f10x.h"
int main(void)
{
    while (1) {
    }
    return 0;
}
```

보드에 포팅

- 보드

- 5V 1A 전원 연결
- JTAG 연결

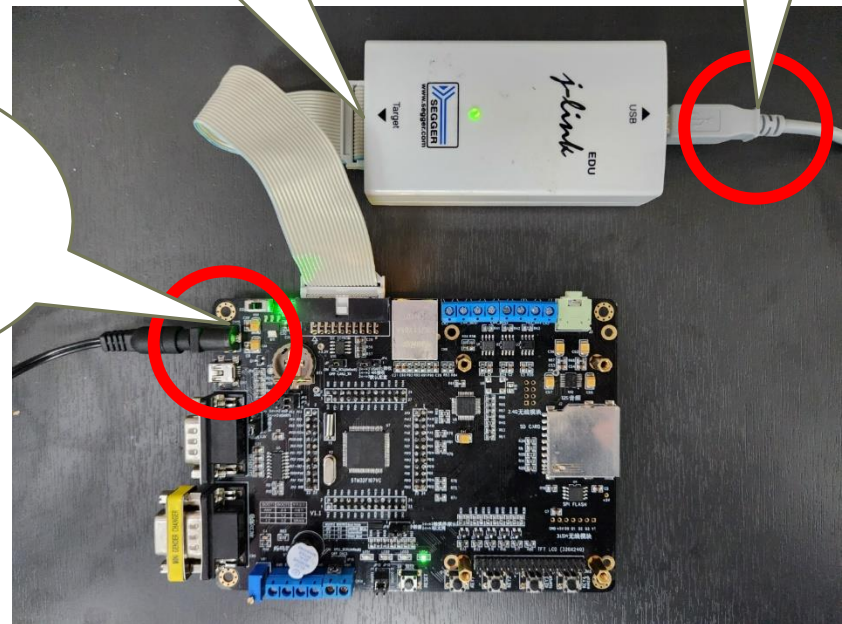
- JTAG

- 보드와 연결
- PC와 연결

보드 - JTAG 연결
플랫케이블 조심히
다뤄주세요

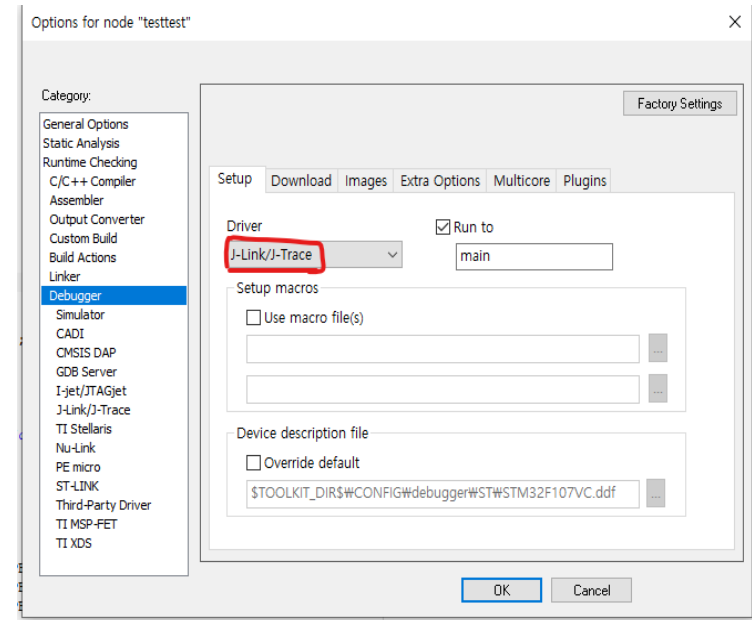
PC와 연결

5V 1A 전원 연결
(5V 5A 전원 절대 연결 금지,
같은 규격이므로 주의!!)



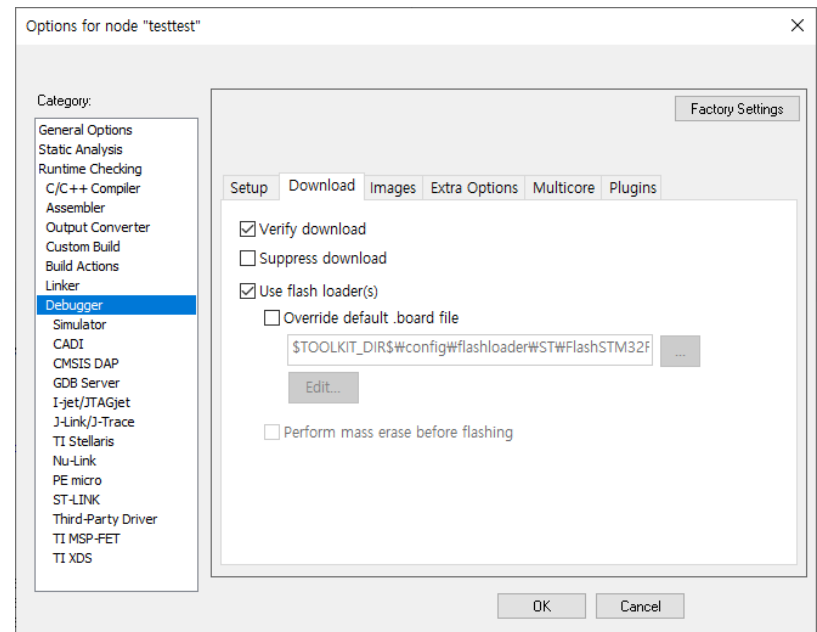
보드에 포팅

- options - Runtime checking – Debugger – Setup 탭
 - Driver – J-link/J-Trace



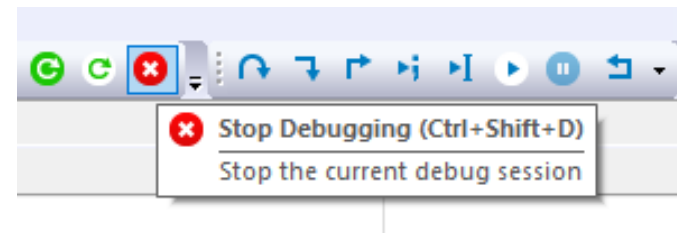
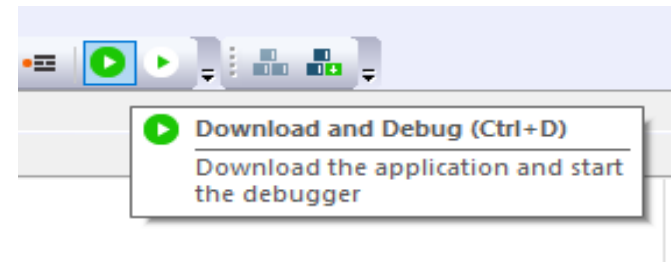
보드에 포팅

- options - Runtime checking – Debugger – Download
 - Verify download
 - Use flash loader(s)



보드에 포팅

- Download and Debug
 - 동의 묻는 창 뜨면 Accept
 - 보드 flash에 프로그램 다운로드 하면서 디버깅 모드로 들어감
- Stop Debugging
 - 디버깅 모드에서 나오기
- 보드 전원 껏다 켜기
- 원하는 대로 동작하는지 확인



- 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

Contents

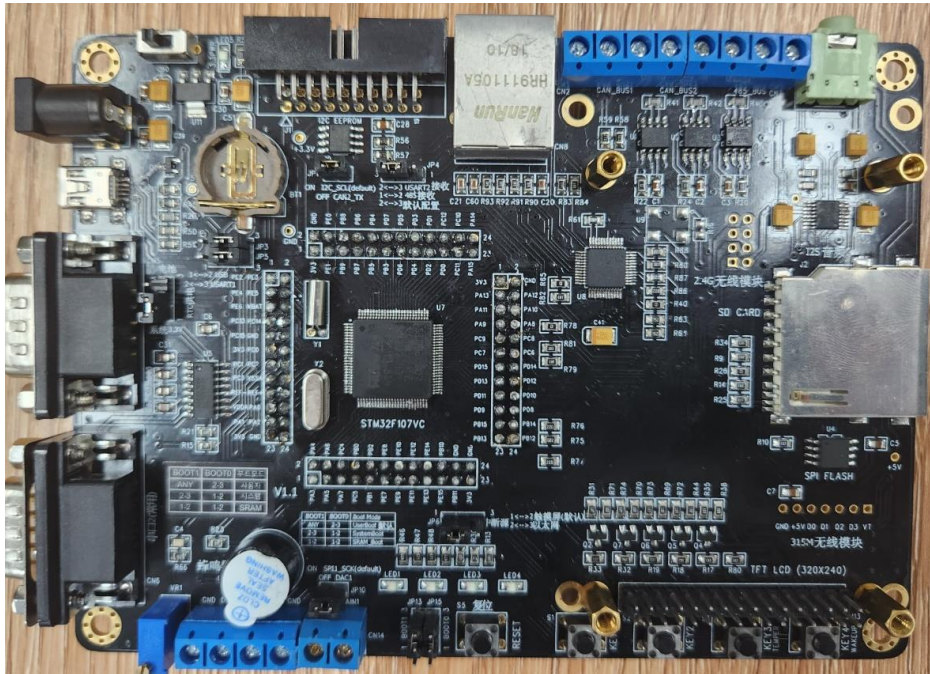
실험 내용

데이터시트와 레퍼런스 문서를 통하여 임베디드 보드를 제어하는 능력향상

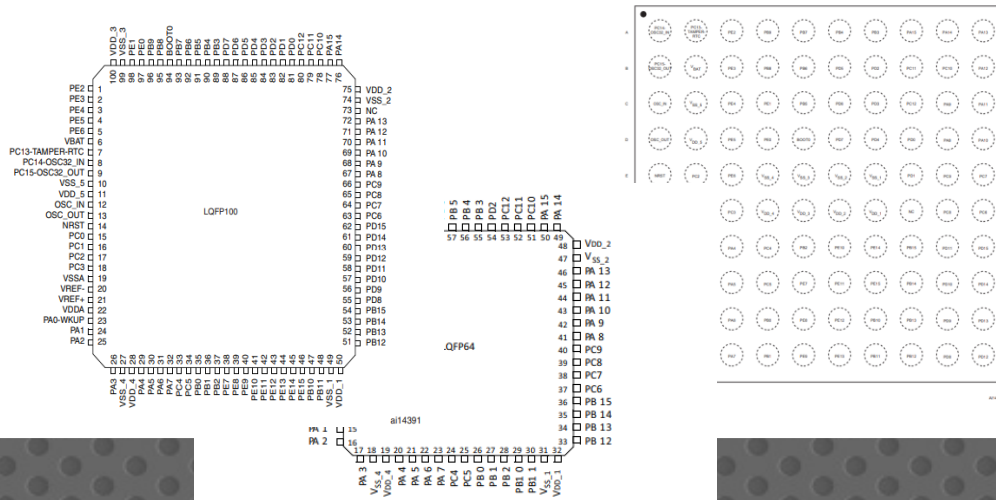
데이터시트: stm32_Datasheet.pdf

레퍼런스 매뉴얼: stm32_ReferenceManual.pdf

STM32 보드 회로도: STM32107VCT6_schematic.pdf



- 모든 외부 장치는 Port, Pin 을 통해 MCU와 연결
- 원하는 Port/Pin에 clock을 부여, 포트 설정을 한 뒤 입출력 가능



데이터시트와 레퍼런스 문서를 통하여 임베디드 보드를 제어하는 능력향상

4 Memory mapping

The memory map is shown in [Figure 5](#).

- 데이터시트에 각 레지스터에 대한 메모리맵핑 주소가 나와 있음
- APB2에 각 GPIO포트가 할당되어 있는 것을 확인할 수 있음
- **각 할당된 주소에 offset을 더하면 해당 레지스터 주소가 됨**

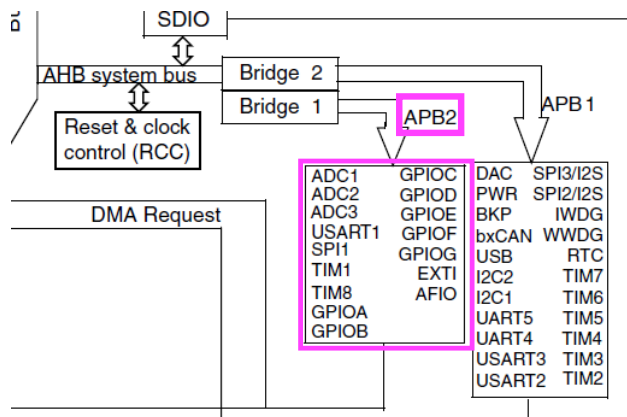
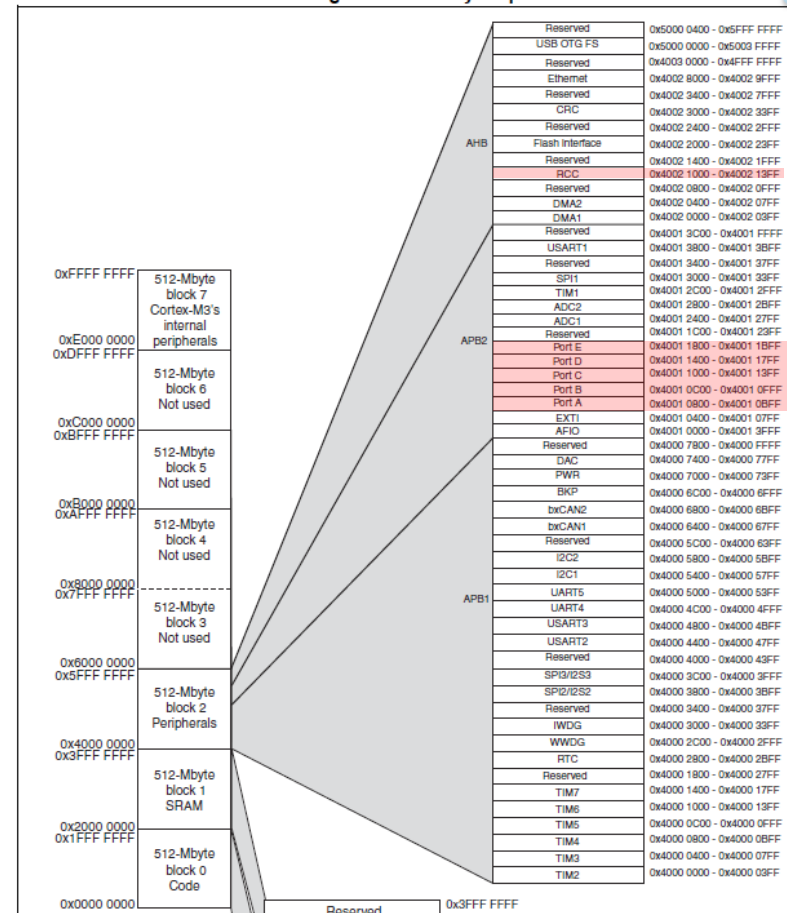


Figure 5. Memory map



데이터시트와 레퍼런스 문서를 직접리딩하여 임베디드 보드를 제어하는 능력향상

7.3.7 APB2 peripheral clock enable register (RCC_APB2ENR)

Address: 0x18

Reset value: 0x0000 0000

Access: word, half-word and byte access

No wait states, except if the access occurs while an access to a peripheral in the APB2 domain is on going. In this case, wait states are inserted until the access to APB2 peripheral is finished.

Note: When the peripheral clock is not active, the peripheral register values may not be readable by software and the returned value is always 0x0.

31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16

레퍼런스 문서에

각 레지스터 설정값과 설명들이
나와있음

General-purpose and alternate-function I/Os (GPIOs and AFIOs)

RM0008

9.2 GPIO registers

Refer to [Section 2.1 on page 47](#) for a list of abbreviations used in register descriptions.

The peripheral registers have to be accessed by words (32-bit).

9.2.1 Port configuration register low (GPIOx_CRL) (x=A..G)

Address offset: 0x00

Reset value: 0x4444 4444

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CNF7[1:0]	MODE7[1:0]	CNF6[1:0]	MODE6[1:0]	CNF5[1:0]	MODE5[1:0]	CNF4[1:0]	MODE4[1:0]	CNF3[1:0]	MODE3[1:0]	CNF2[1:0]	MODE2[1:0]	CNF1[1:0]	MODE1[1:0]	CNF0[1:0]	MODE0[1:0]
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CNF7[1:0]	MODE7[1:0]	CNF6[1:0]	MODE6[1:0]	CNF5[1:0]	MODE5[1:0]	CNF4[1:0]	MODE4[1:0]	CNF3[1:0]	MODE3[1:0]	CNF2[1:0]	MODE2[1:0]	CNF1[1:0]	MODE1[1:0]	CNF0[1:0]	MODE0[1:0]
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits 31:30, 27:26, 23:22, 19:18, 15:14, 11:10, 7:6, 3:2

CNFy[1:0]: Port x configuration bits (y= 0 .. 7)
These bits are written by software to configure the corresponding I/O port.

Refer to [Table 20: Port bit configuration table on page 161](#).

In input mode (MODE[1:0]=00):

- 00: Analog mode
- 01: Floating input (reset state)
- 10: Input with pull-up / pull-down
- 11: Reserved

In output mode (MODE[1:0] > 00):

- 00: General purpose output push-pull
- 01: General purpose output Open-drain
- 10: Alternate function output Push-pull
- 11: Alternate function output Open-drain

Bits 29:28, 25:24, 21:20, 17:16, 13:12, 9:8, 5:4, 1:0

MODEy[1:0]: Port x mode bits (y= 0 .. 7)
These bits are written by software to configure the corresponding I/O port.

Refer to [Table 20: Port bit configuration table on page 161](#).

- 00: Input mode (reset state)
- 01: Output mode, max speed 10 MHz.
- 10: Output mode, max speed 2 MHz.
- 11: Output mode, max speed 50 MHz.

데이터시트와 레퍼런스 문서를 직접리딩하여 임베디드 보드를 제어하는 능력향상

General-purpose and alternate-function I/Os (GPIOs and AFIOs)

RM0008

9.2 GPIO registers

Refer to [Section 2.1 on page 47](#) for a list of abbreviations used in register descriptions.
The peripheral registers have to be accessed by words (32-bit).

9.2.1 Port configuration register low (GPIOx_CRL) (x=A..G)

Address offset: 0x00

Reset value: 0x4444 4444

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CNF7[1:0]	MODE7[1:0]	CNF6[1:0]	MODE6[1:0]	CNF5[1:0]	MODE5[1:0]	CNF4[1:0]	MODE4[1:0]	CNF3[1:0]	MODE3[1:0]	CNF2[1:0]	MODE2[1:0]	CNF1[1:0]	MODE1[1:0]	CNF0[1:0]	MODE0[1:0]
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CNF3[1:0]	MODE3[1:0]	CNF2[1:0]	MODE2[1:0]	CNF1[1:0]	MODE1[1:0]	CNF0[1:0]	MODE0[1:0]								
rw	rw	rw	rw	rw	rw	rw	rw								

Bits 31:30, 27:26, CNFy[1:0]: Port x configuration bits (y= 0 .. 7)

23:22, 19:18, 15:14, 11:10, 7:6, 3:2 These bits are written by software to configure the corresponding I/O port.
Refer to [Table 20: Port bit configuration table on page 161](#).

In input mode (MODE[1:0]=00):

00: Analog mode

01: Floating input (reset state)

10: Input with pull-up / pull-down

11: Reserved

In output mode (MODE[1:0] > 00):

00: General purpose output push-pull

01: General purpose output Open-drain

10: Alternate function output Push-pull

11: Alternate function output Open-drain

Bits 29:28, 25:24, MODEy[1:0]: Port x mode bits (y= 0 .. 7)

21:20, 17:16, 13:12, 9:8, 5:4, 1:0 These bits are written by software to configure the corresponding I/O port.
Refer to [Table 20: Port bit configuration table on page 161](#).

00: Input mode (reset state)

01: Output mode, max speed 10 MHz.

10: Output mode, max speed 2 MHz.

11: Output mode, max speed 50 MHz.

포트의 핀 설정시 주의사항

레지스터의 사용하려는 부분을 0으로 초기화 후 사용

0011(output push-pull 50MHz) = 0x3
|= 0x30000000 (X)

Reset value |= 0x3;

0100(2) -> 0111(2)(X) (output Open-drain 50MHz)

데이터시트와 레퍼런스 문서를 직접리딩하여 임베디드 보드를 제어하는 능력향상

9.2.3 Port input data register (GPIOx_IDR) (x=A..G)

Address offset: 0x08h

Reset value: 0x0000 XXXX

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
IDR15	IDR14	IDR13	IDR12	IDR11	IDR10	IDR9	IDR8	IDR7	IDR6	IDR5	IDR4	IDR3	IDR2	IDR1	IDR0
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

Bits 31:16 Reserved, must be kept at reset value.

Bits 15:0 **IDRy**: Port input data (y= 0 .. 15)

These bits are read only and can be accessed in Word mode only. They contain the input value of the corresponding I/O port.

레퍼런스 문서에

각 레지스터설명과, 원하는 설정을 위한 Bit Setting 값이 나와있음.

General-purpose and alternate-function I/Os (GPIOs and AFIOs)

RM0008

9.2.4 Port output data register (GPIOx_ODR) (x=A..G)

Address offset: 0x0C

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
ODR15	ODR14	ODR13	ODR12	ODR11	ODR10	ODR9	ODR8	ODR7	ODR6	ODR5	ODR4	ODR3	ODR2	ODR1	ODR0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits 31:16 Reserved, must be kept at reset value.

Bits 15:0 **ODRy**: Port output data (y= 0 .. 15)

These bits can be read and written by software and can be accessed in Word mode only.
Note: For atomic bit set/reset, the ODR bits can be individually set and cleared by writing to the GPIOx_BSRR register (x = A .. G).

9.2.5 Port bit set/reset register (GPIOx_BSRR) (x=A..G)

Address offset: 0x10

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
BSR15	BSR14	BSR13	BSR12	BSR11	BSR10	BSR9	BSR8	BSR7	BSR6	BSR5	BSR4	BSR3	BSR2	BSR1	BSR0
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BS15	BS14	BS13	BS12	BS11	BS10	BS9	BS8	BS7	BS6	BS5	BS4	BS3	BS2	BS1	BS0
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w

Bits 31:16 **BSRy**: Port x Reset bit y (y= 0 .. 15)

These bits are write-only and can be accessed in Word mode only.

0: No action on the corresponding ODRx bit

1: Reset the corresponding ODRx bit

Note: If both BSx and BRx are set, BSx has priority.

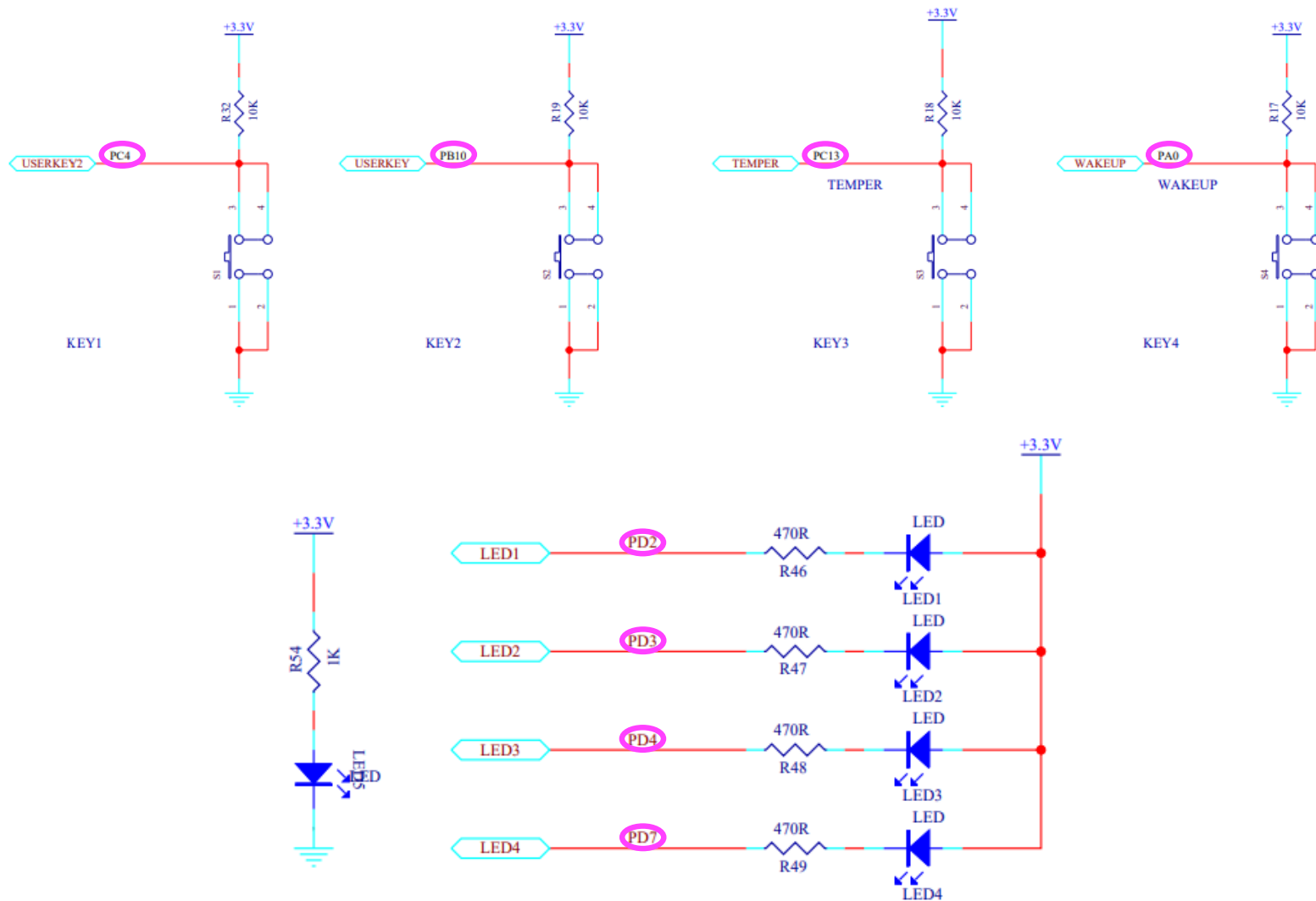
Bits 15:0 **BSy**: Port x Set bit y (y= 0 .. 15)

These bits are write-only and can be accessed in Word mode only.

0: No action on the corresponding ODRx bit

1: Set the corresponding ODRx bit

두 문서를 참고하여 레지스터를 제어하여 펌웨어를 보드에 올리면 동작



Bitwise Operation

- AND 연산 : $\&$
 - $0b0011 \& 0b0101 == 0b0001$
 - $X \& 0 == 0$
 - $X \& 1 == X$
- OR 연산 : $|$
 - $0b0011 | 0b0101 == 0b0111$
 - $X | 0 == X$
 - $X | 1 == 1$
- XOR 연산 : \wedge
 - $0b0011 \wedge 0b0101 == 0b0110$
 - $X \wedge 0 == X$
 - $X \wedge 1 == \sim X$
- Not 연산 : \sim
 - $\sim 0b01 == 0b10$
 - $\sim 0 = 1$
 - $\sim 1 = 0$
- Shift 연산 : \ll, \gg
 - $0b01011 \ll 1 == 0b10110$
 - $0b01011 \gg 1 == 0b00101$

Volatile Keyword

- volatile 선언된 변수는 컴파일러가 최적화를 수행하지 않음.

(접근 시 항상 메모리 참조)

```
*(unsigned int *)0x8C0F = 0x8001  
*(unsigned int *)0x8C0F = 0x8002;  
*(unsigned int *)0x8C0F = 0x8003;  
*(unsigned int *)0x8C0F = 0x8004;  
*(unsigned int *)0x8C0F = 0x8005;
```

VS.

```
*(volatile unsigned int *)0x8C0F = 0x8001  
*(volatile unsigned int *)0x8C0F = 0x8002;  
*(volatile unsigned int *)0x8C0F = 0x8003;  
*(volatile unsigned int *)0x8C0F = 0x8004;  
*(volatile unsigned int *)0x8C0F = 0x8005;
```

*0x8C0F = 0x8005만 수행하면 만족

vs

5개의 instruction 모두 수행

원하는 포트와 핀(GPIO)을 제어하기 위해 해야할 일들

1. RCC (reset and clock control)를 사용하여 사용하고자 하는 GPIO에 clock을 인가 (peripheral clock enable)
2. 사용하려는 GPIO Port, Pin의 input/output 설정 (Port Configuration)
3. GPIO의 Input (Port input data), output (Port output data) <- Port bit set/reset으로 제어를 통하여 센서 및 액추에이터를 제어하고 오실로스코프로 확인한다.

• 레지스터에 들어있는 값을 읽거나 레지스터에 씴으로써 보드 제어하는 방법

예시) GPIO PE5 reset

APB2	Reserved	0x4001 1C00 - 0x4001 23FF
	Port E	0x4001 1800 - 0x4001 1BFF
	Port D	0x4001 1400 - 0x4001 17FF
	Port C	0x4001 1000 - 0x4001 13FF
	Port B	0x4001 0C00 - 0x4001 0FFF
	Port A	0x4001 0800 - 0x4001 0BFF
	EXTI	0x4001 0400 - 0x4001 07FF

GPIO Port E의 BSRR 레지스터 주소 계산:

(Port E base) + (GPIOx_BSRR address offset) => 0x40011800 + 0x10 = 0x40011810

해당 bit에 대입: *((volatile unsigned int *)0x40011810) |= 0x00200000;

9.2.5 Port bit set/reset register (GPIOx_BSRR) (x=A..G)

Address offset: 0x10

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
BR15	BR14	BR13	BR12	BR11	BR10	BR9	BR8	BR7	BR6	BR5	BR4	BR3	BR2	BR1	BR0
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BS15	BS14	BS13	BS12	BS11	BS10	BS9	BS8	BS7	BS6	BS5	BS4	BS3	BS2	BS1	BS0
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w

Bits 31:16 **BRy**: Port x Reset bit y (y= 0 .. 15)

These bits are write-only and can be accessed in Word mode only.

0: No action on the corresponding ODRx bit

1: Reset the corresponding ODRx bit

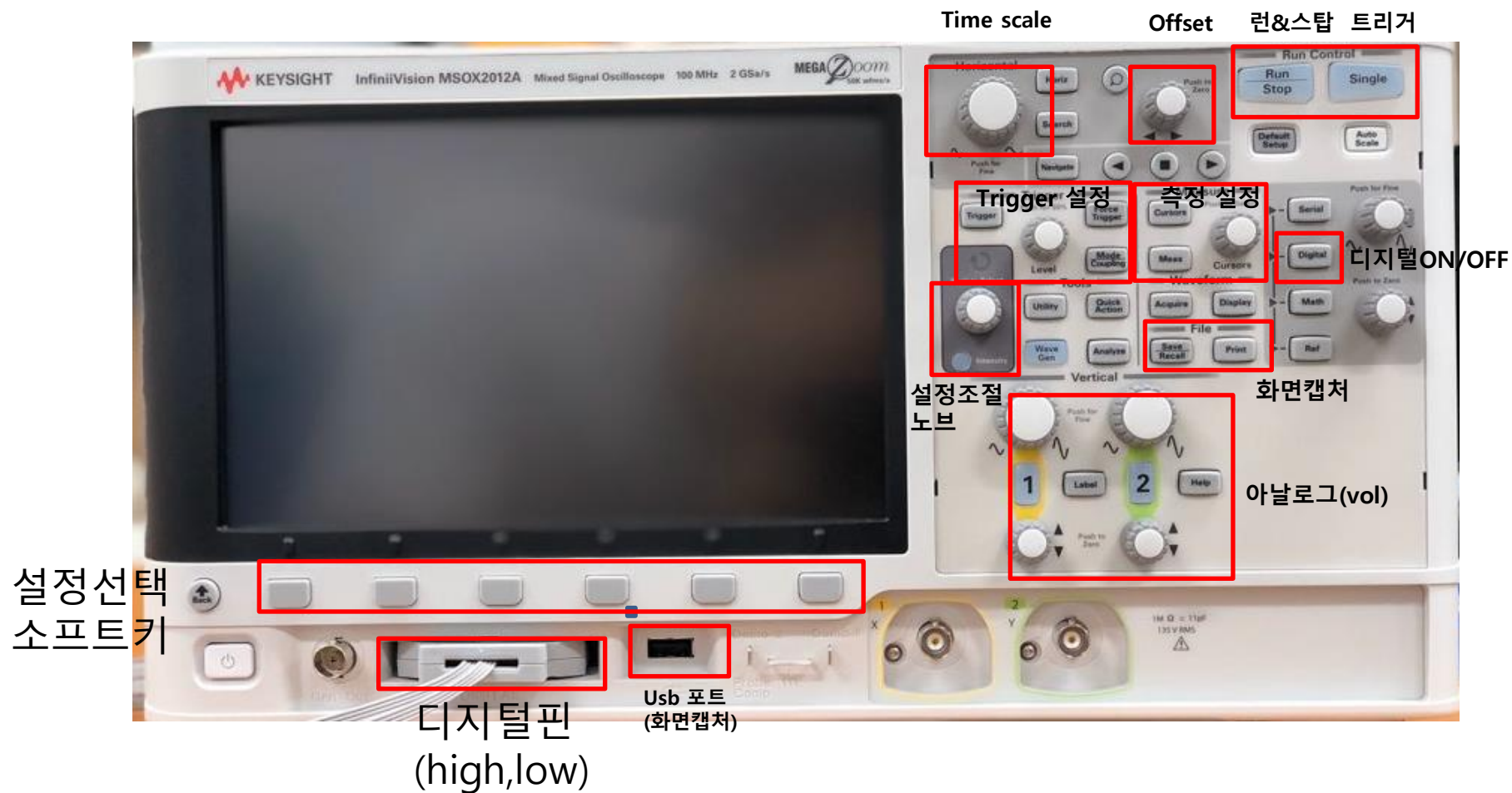
Note: If both BSx and BRx are set, BSx has priority.

Bits 15:0 **BSy**: Port x Set bit y (y= 0 .. 15)

These bits are write-only and can be accessed in Word mode only.

0: No action on the corresponding ODRx bit

1: Set the corresponding ODRx bit





미션 ! 별도 미션지 참고

실험 검사

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

1. 정확한 장비 설정 유무 확인
2. 레지스터 및 주소 설정 이해 확인
3. LED제어 원리 이해 및 동작 확인
4. 오실로스코프 디지털 핀 사용법 이해

실험 결과 보고서

- A. 이론부터 실습까지 전반적인 내용을 포함하도록 작성 (실험 과정 사진 찍으시면 좋아요)
- B. 다음 실험시간 전까지 PLATO 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!!!