

앞	
	1조
3조	
	5조
7조	
	9조
11조	

앞	
2조	
	4조
6조	
8조	
10조	

조장 선정 및 연락처 교환

단톡방 생성

예비 발표 방법 변경

- 발표 영상을 녹화하여 조교 이메일 (min7ae@gmail.com) 로 제출
- 모든 학생은 조교가 PLATO에 업로드한 동영상 각자 시청 (시청 체크하여 성적에 반영)

예비 발표 순서 정하기 – A class(한 주 씩), B class (11월 17일 제외 한 주 씩)

10월 6일 (2조)	10월 13일 (2조)	10월 27일 (2조)	11월 3일 (2조)	11월 10일 (2조)	11월 17일



Pusan
National
University



September 15, 2020

조교
김민재

min7ae@gmail.com

임베디드 시스템 설계 및 실험

화요일 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/iar-embedded-workbench2/#!?currentTab=free-trials>
 - > IAR Embedded Workbench for Arm 다운로드

ARM

Download Software

(Version 8.50, 1467.44 MB)

IAR Embedded Workbench for Arm

The evaluation license is completely free of charge and allows you to try the integrated development environment and evaluate its efficiency and ease of use. When you start the product for the first time, you will be asked to register to get your evaluation license.

After download and installation, you have the following evaluation options to choose from:

- a 30-day time-limited but fully functional license
- a size-limited Kickstart license without any time limit

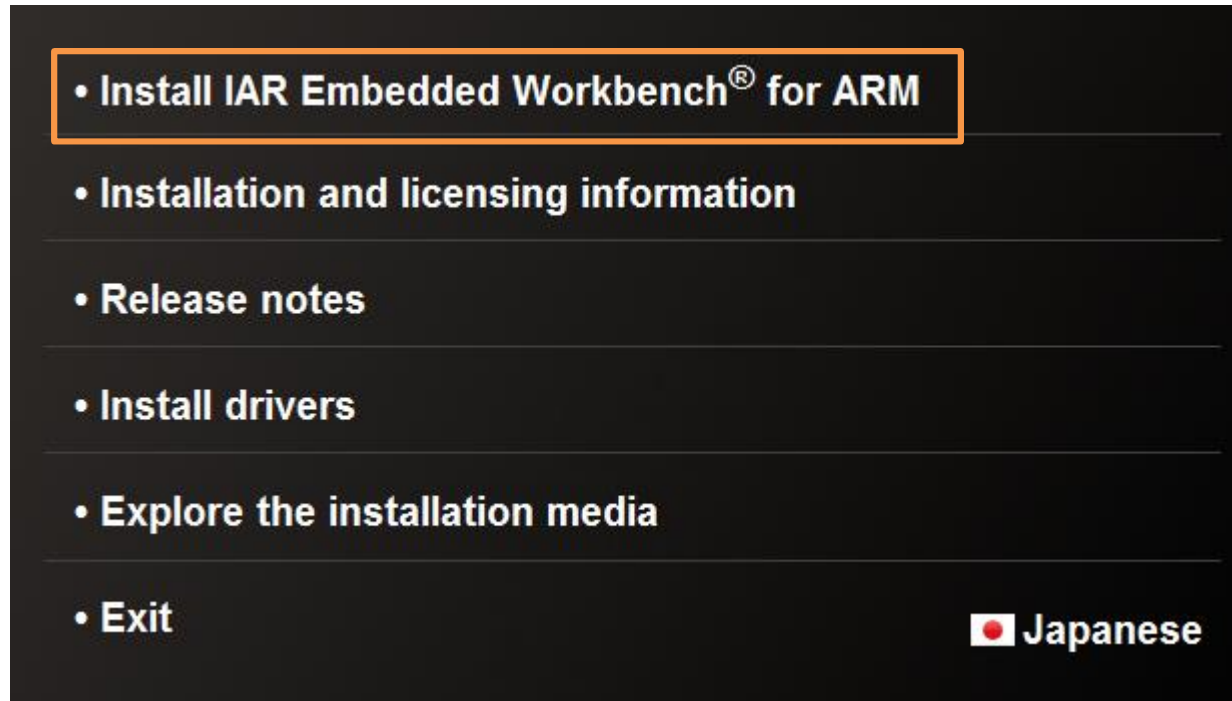
Restrictions to the 30-day time-limited evaluation

- A 30-day time limitation.
- Source code for runtime libraries is not included.
- No support for MISRA C.
- C-RUN is size-limited to 12 Kbytes of code, excluding constant data.
- Limited technical support.
- Must not be used for product development or any other kind of commercial use.

Restrictions to the Kickstart, size-limited evaluation

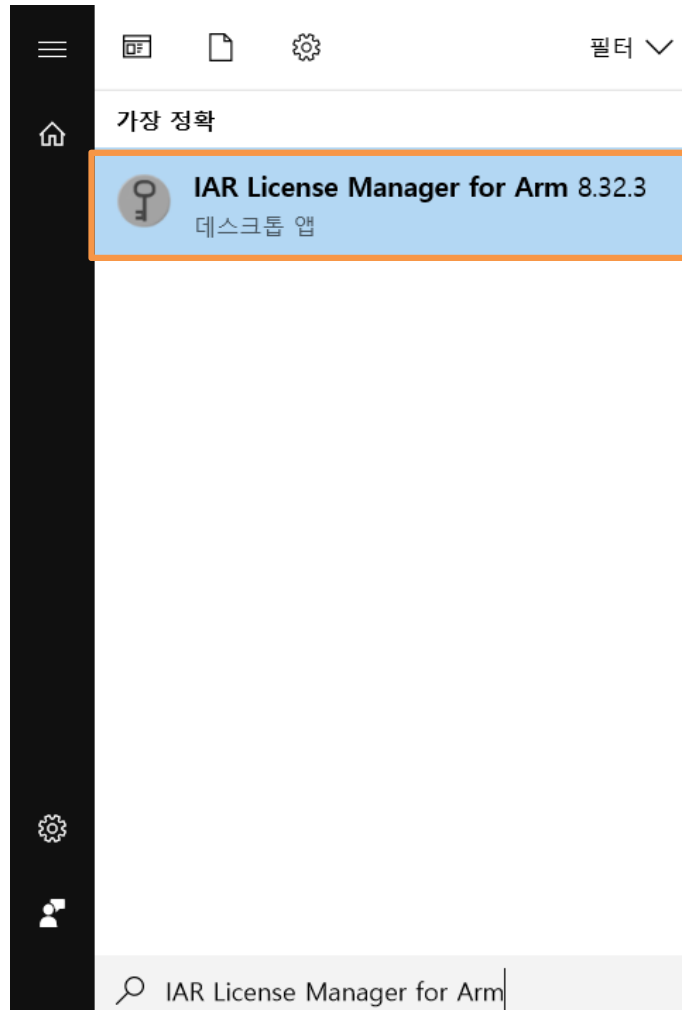
- A 32 Kbyte code size limitation
- Source code for runtime libraries is not included.
- No support for MISRA C.
- C-RUN is not available.
- Limited technical support.

IAR Embedded Workbench 설치



- 설치 파일 실행
- Install IAR Embedded Workbench for ARM 클릭해서 IAR 툴 및 모든 드라이버 설치

IAR Embedded Workbench 설치



- 설치된 IAR License Manager for Arm 실행

IAR Embedded Workbench 설치

License Wizard

✕

Welcome



This wizard will help you to activate your IAR Embedded Workbench for Arm license.

☐ If you have a license number, enter it here:

☐ Use a network license

☒ Register with IAR Systems to get an evaluation license

☐ Don't run the Wizard for this product at startup.

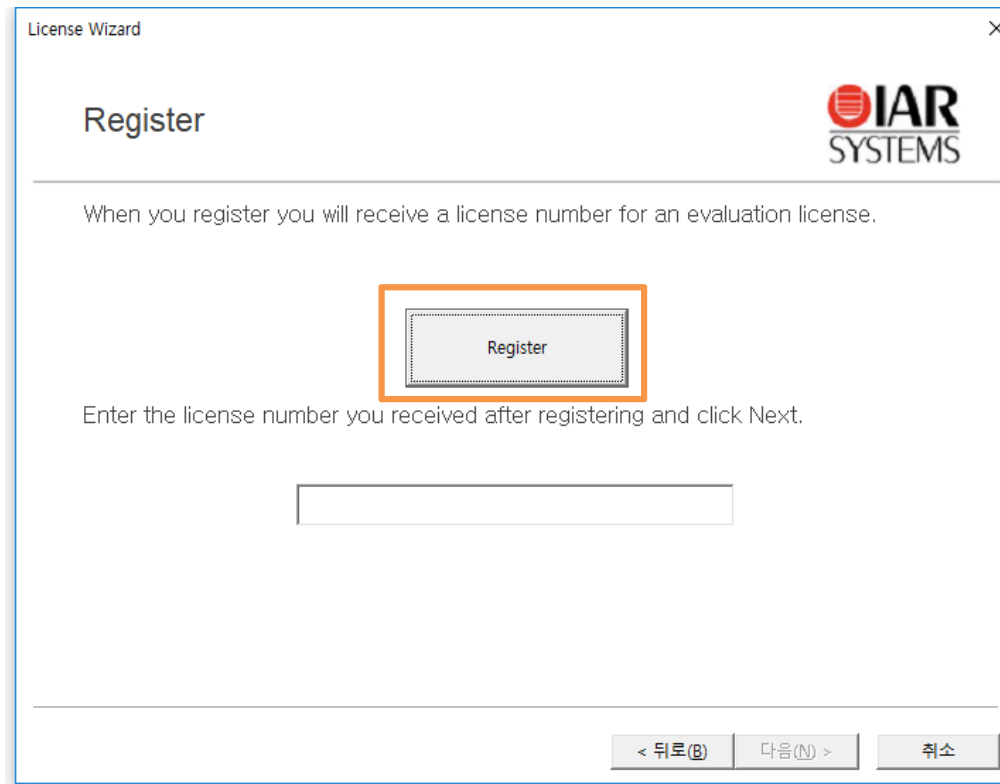
< 뒤로(B)

다음(N) >

취소

- Register with IAR Systems to get an evaluation license 선택

IAR Embedded Workbench 설치



- Register 클릭

IAR Embedded Workbench 설치

- 라이선스 종류 두가지

1. 코드 제한 없이 30일
2. 코드 제한으로 무제한

– 수업 과제는 코드 제한으로도 가능

- 차후 텀 프로젝트는 새로 설치해서 다른 이메일을 이용하여 30일 Time limited 사용

Select the manufacturer of the processor you intend to run your code on: *

STMicroelectronics ▼

- Manufacturer는 STMicroelectronics 선택

English • 日本語

IAR SYSTEMS

Register for Evaluation

Evaluation license type *

☐ Time limited (30 days)
IAR Embedded Workbench for Arm, v. 8.22, Evaluation Edition

☒ Code size limited
IAR Embedded Workbench for Arm, v. 8.22, 32K Kickstart Edition

First name *

Last name *

Job title

Email *

Phone *

Extension:

Company *

Department

Country (and State/Province) *

Please specify: ▼

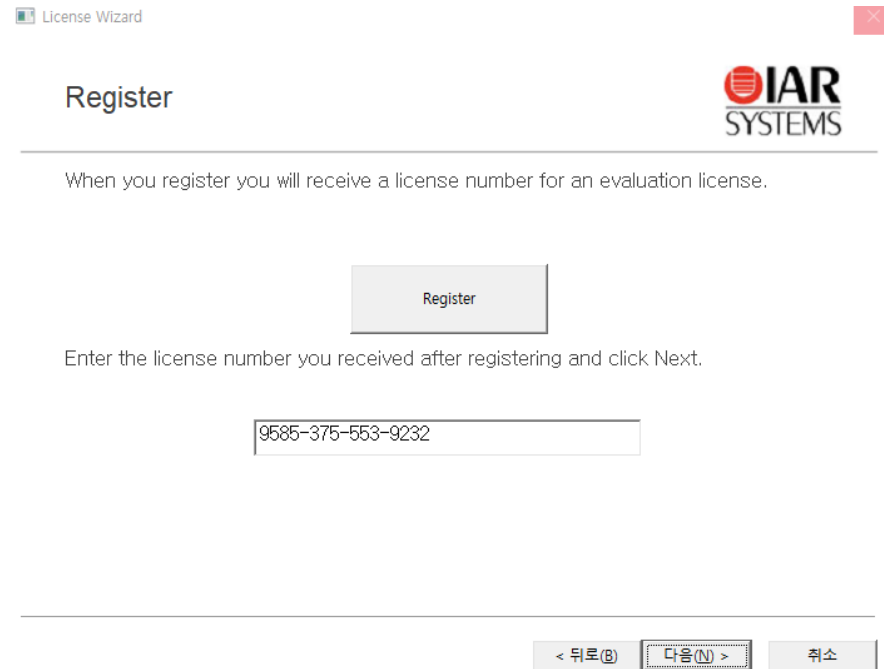
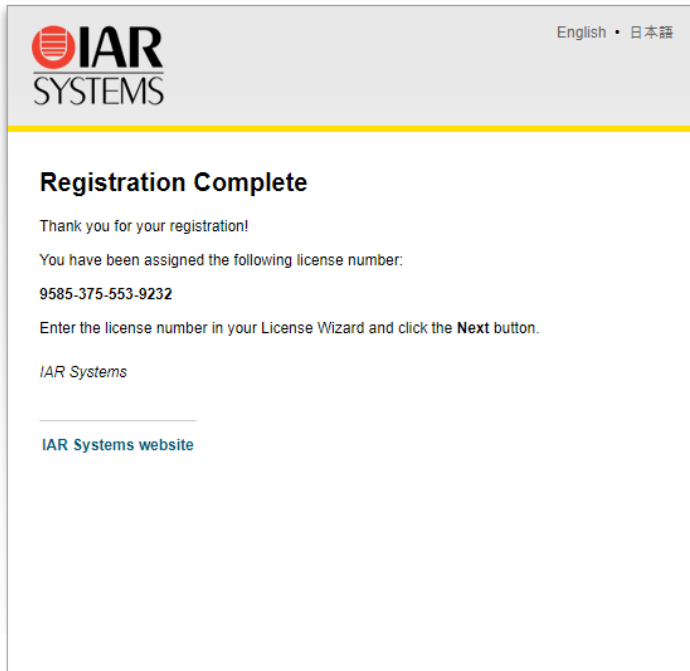
Street address / P.O. box *

Zip code *

City *

Company website *

IAR Embedded Workbench 설치



- 왼쪽과 같은 메일이 오면 라이선스 번호 확인 후

오른쪽과 같이 하단에 입력

IAR Embedded Workbench 설치

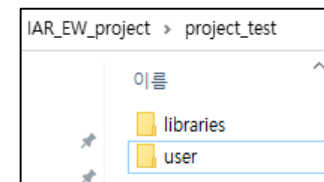
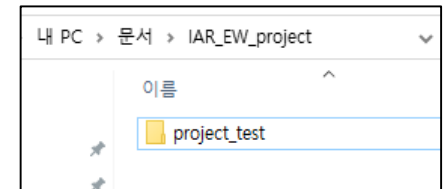
- trial 버전과 실험실에 설치된 버전은 서로 프로젝트 호환이 안 되므로 프로젝트를 새로 만들고 코드를 복사해야 함

Contents

프로젝트 및 실험 장비 설정

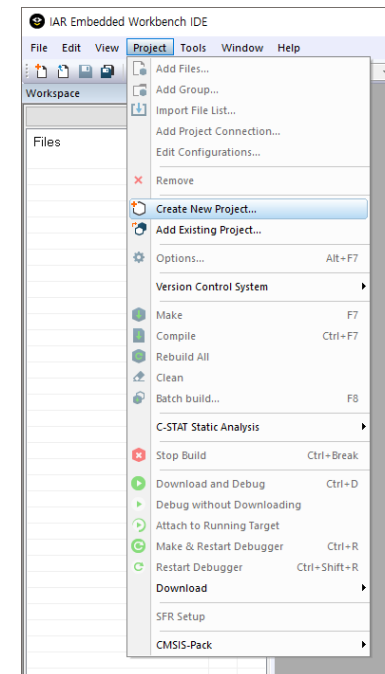
프로젝트 생성 및 설정

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



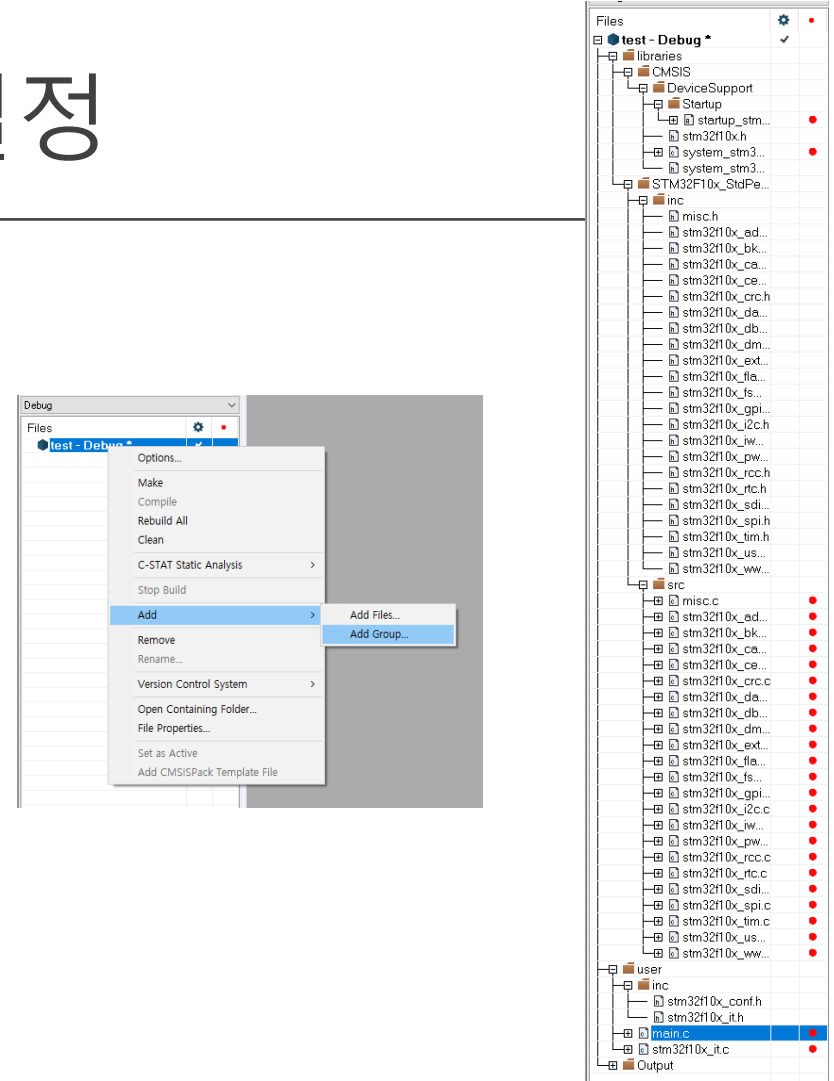
프로젝트 생성 및 설정

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



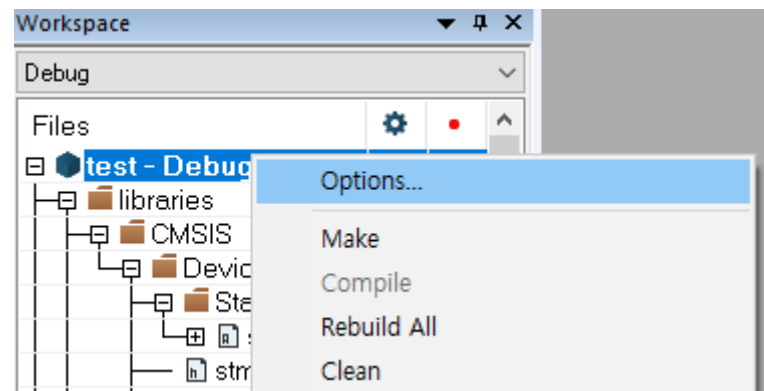
프로젝트 생성 및 설정

- 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



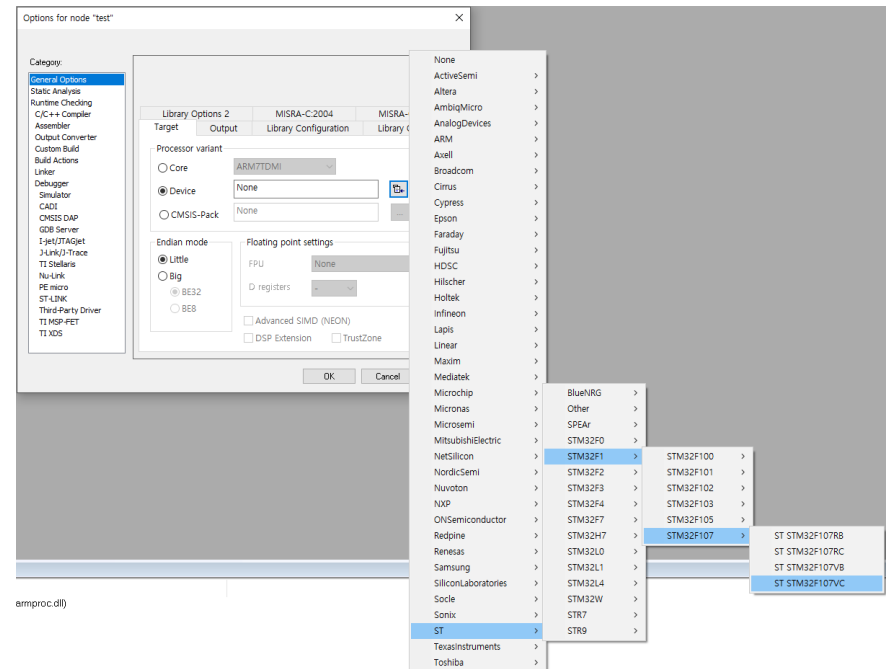
프로젝트 생성 및 설정

- 프로젝트 오른쪽 클릭 - Options



프로젝트 생성 및 설정

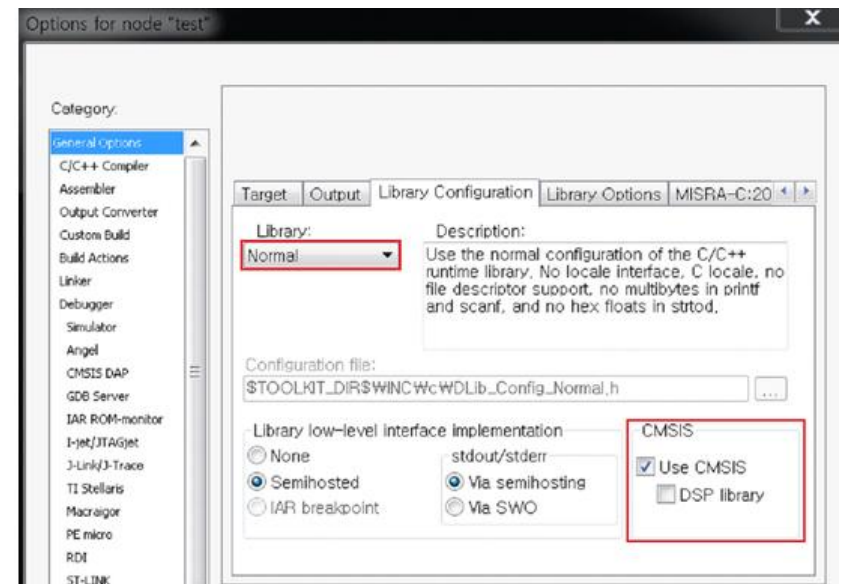
- General Options – Target 탭 – Device
 - ST – STM32F1 – STM32F107 – ST STM32F107VC



프로젝트 생성 및 설정

• General Options – Library Configuration 탭

- Library
 - Normal
- CMSIS
 - Use CMSIS

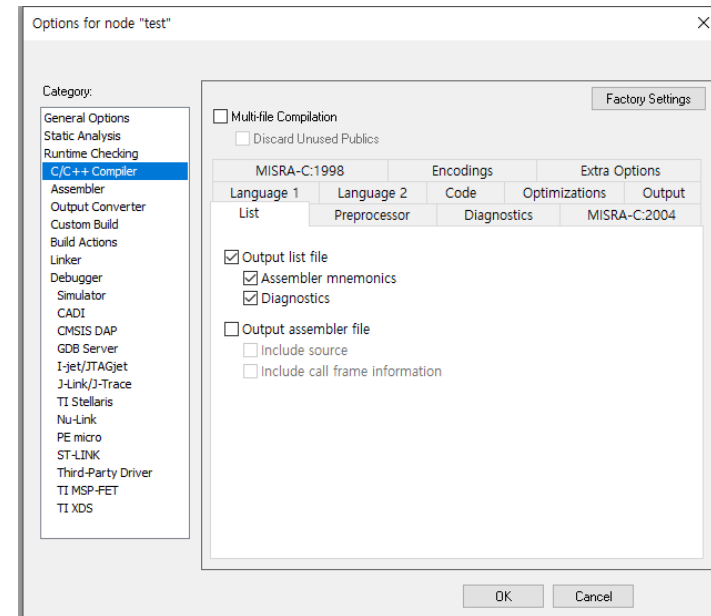


프로젝트 생성 및 설정

- 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

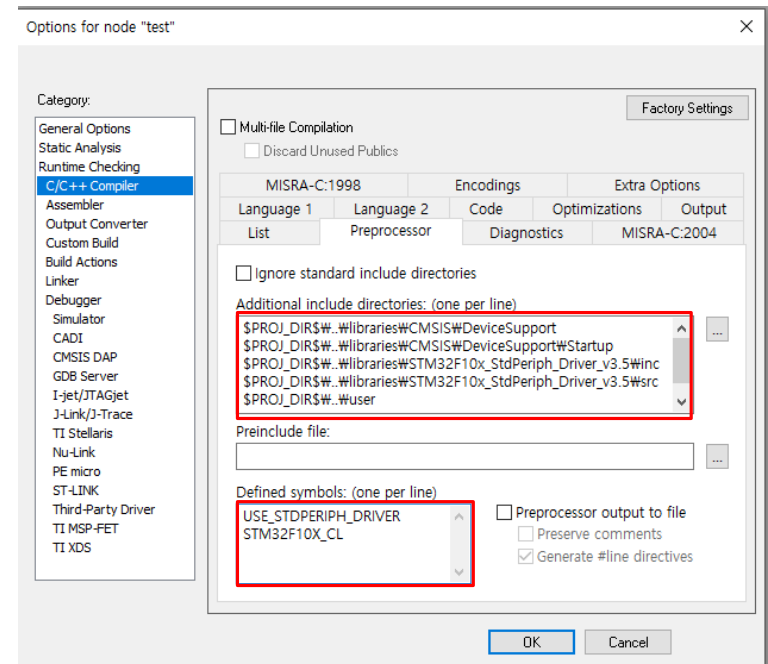
\$PROJ_DIR\$\user

\$PROJ_DIR\$\user\inc

- Defined symbols: (one per line)

USE_STDPERIPH_DRIVER

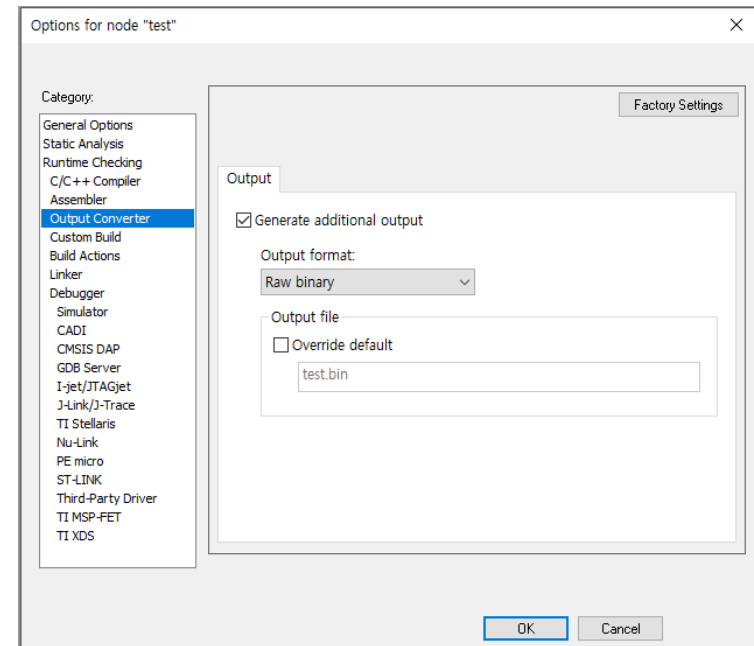
STM32F10X_CL



프로젝트 생성 및 설정

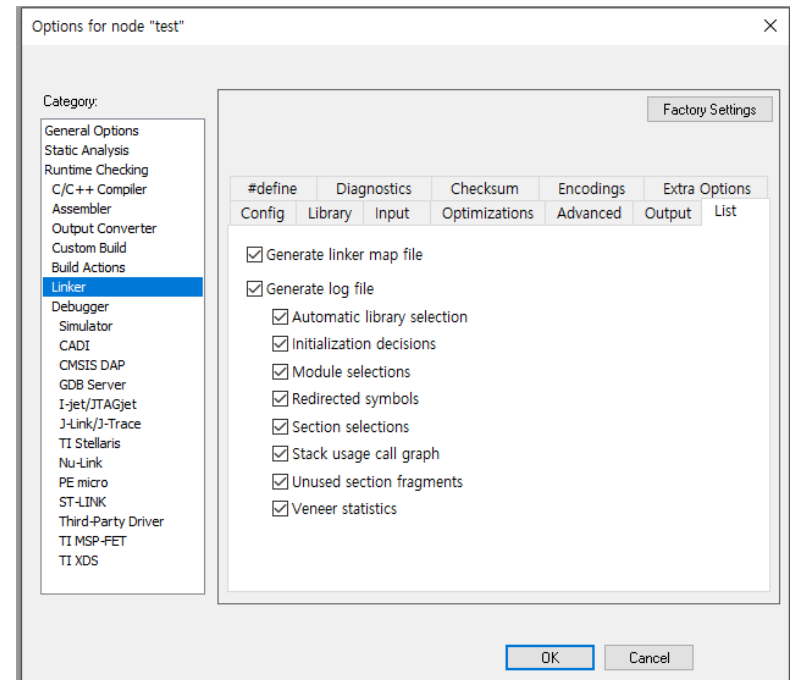
- Runtime Checking – Output Converter

- Generate additional output
 - Output format – Raw binary



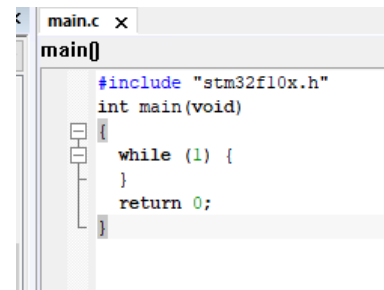
프로젝트 생성 및 설정

- 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 뜸
 - 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

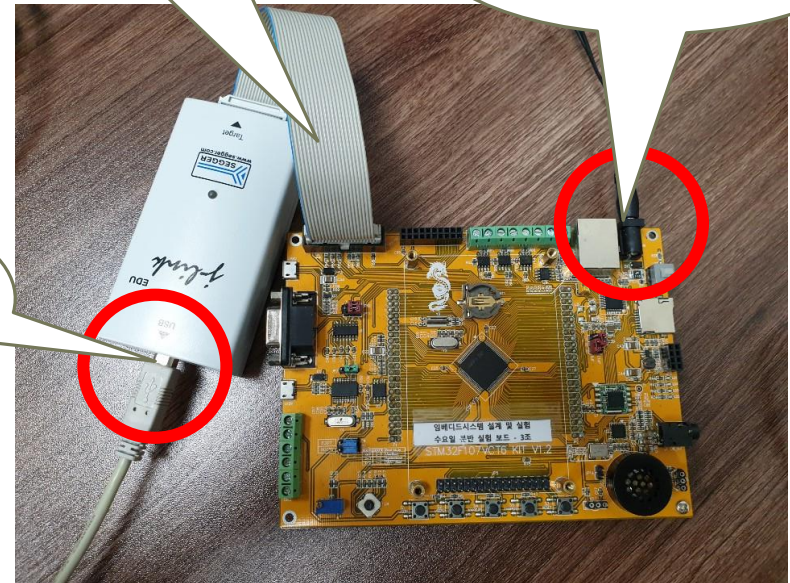
- JTAG

- 보드와 연결
- PC와 연결

보드 - JTAG 연결
플랫케이블 조심히
다뤄주세요

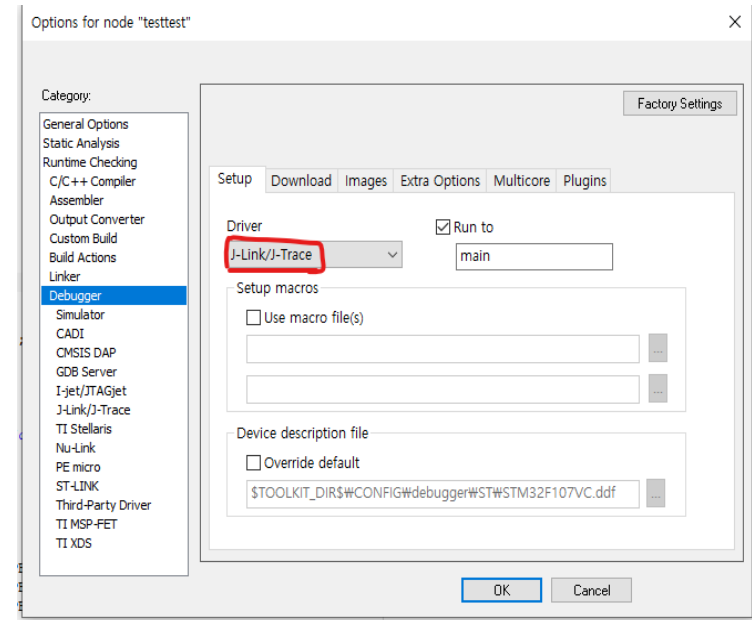
5V 1A 전원 연결
(5V 5A 전원 절대 연결 금지,
같은 규격이므로 주의!!)

PC와 연결



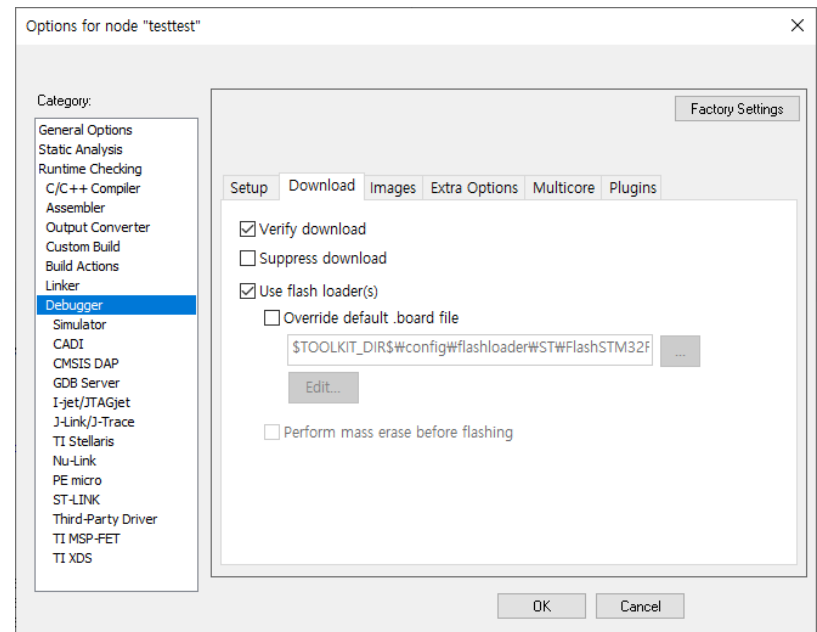
보드에 포팅

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



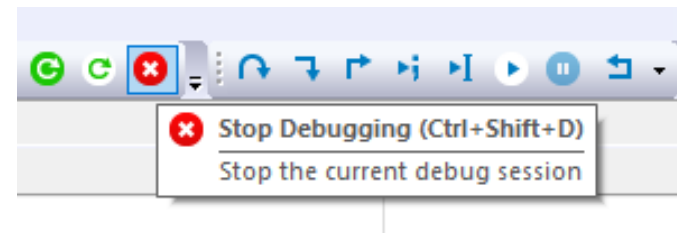
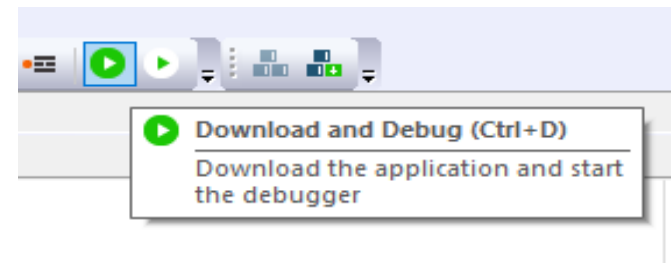
보드에 포팅

- 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

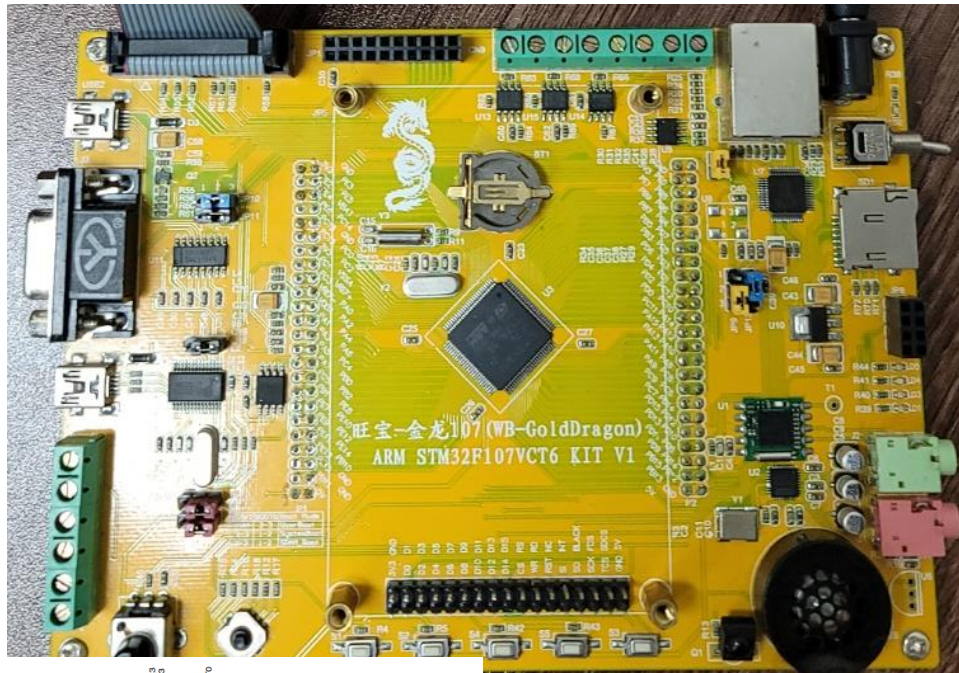
실험 내용

데이터시트와 레퍼런스 문서를 통하여 임베디드 보드를 제어하는 능력향상

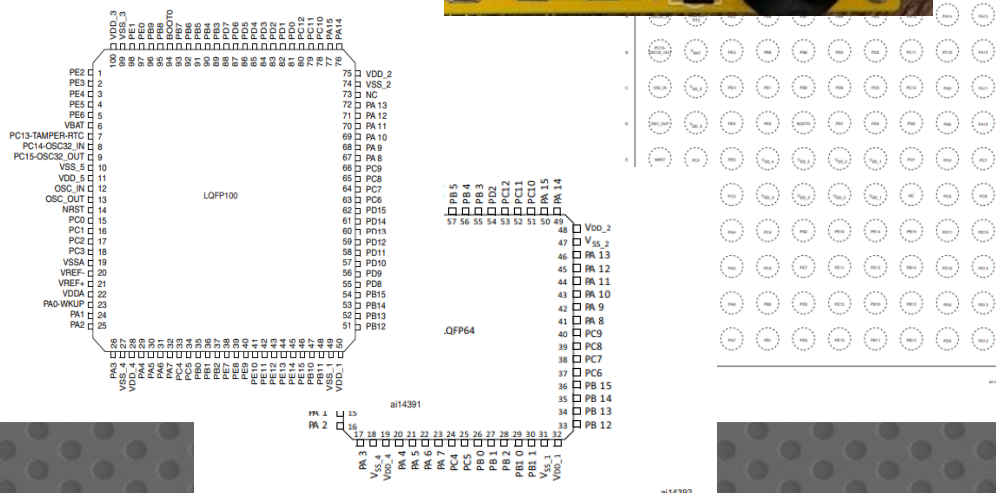
데이터시트: stm32_Datasheet.pdf

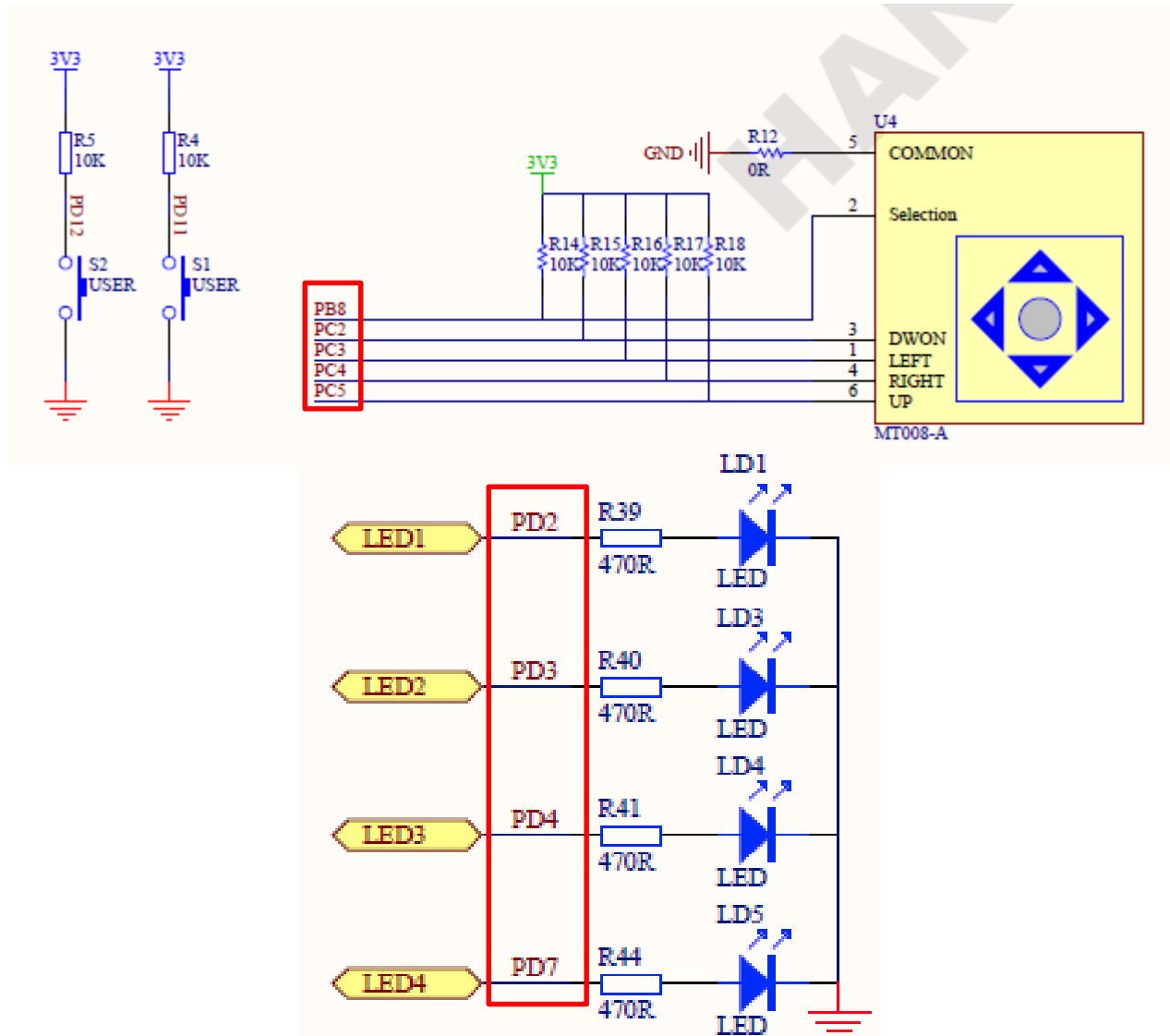
레퍼런스 매뉴얼: stm32_ReferenceManual.pdf

STM32 보드 회로도: STM32F107VCT6_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 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

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 IDRx: 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;
```

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

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.

(아두이노)

- 버튼은 10번 핀
- LED는 11번 핀
- 버튼을 누르면 LED가 켜진다.
- 버튼은 풀업 저항에 연결되어있음.

(아두이노 코드(간소화))

```
pinMode (10, INPUT);
pinMode (11, OUTPUT);

While (1) {
    if (digitalRead(10) == LOW) {
        digitalWrite(11, HIGH);
    }
}
```

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.

(아두이노)

- 버튼은 10번 핀
- LED는 11번 핀
- 버튼을 누르면 LED가 켜진다.
- 버튼은 풀업 저항에 연결되어있음.

(아두이노 코드(간소화))

```
pinMode (10, INPUT);
pinMode (11, OUTPUT);

While (1) {
    if (digitalRead(10) == LOW) {
        digitalWrite(11, HIGH);
    }
}
```

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

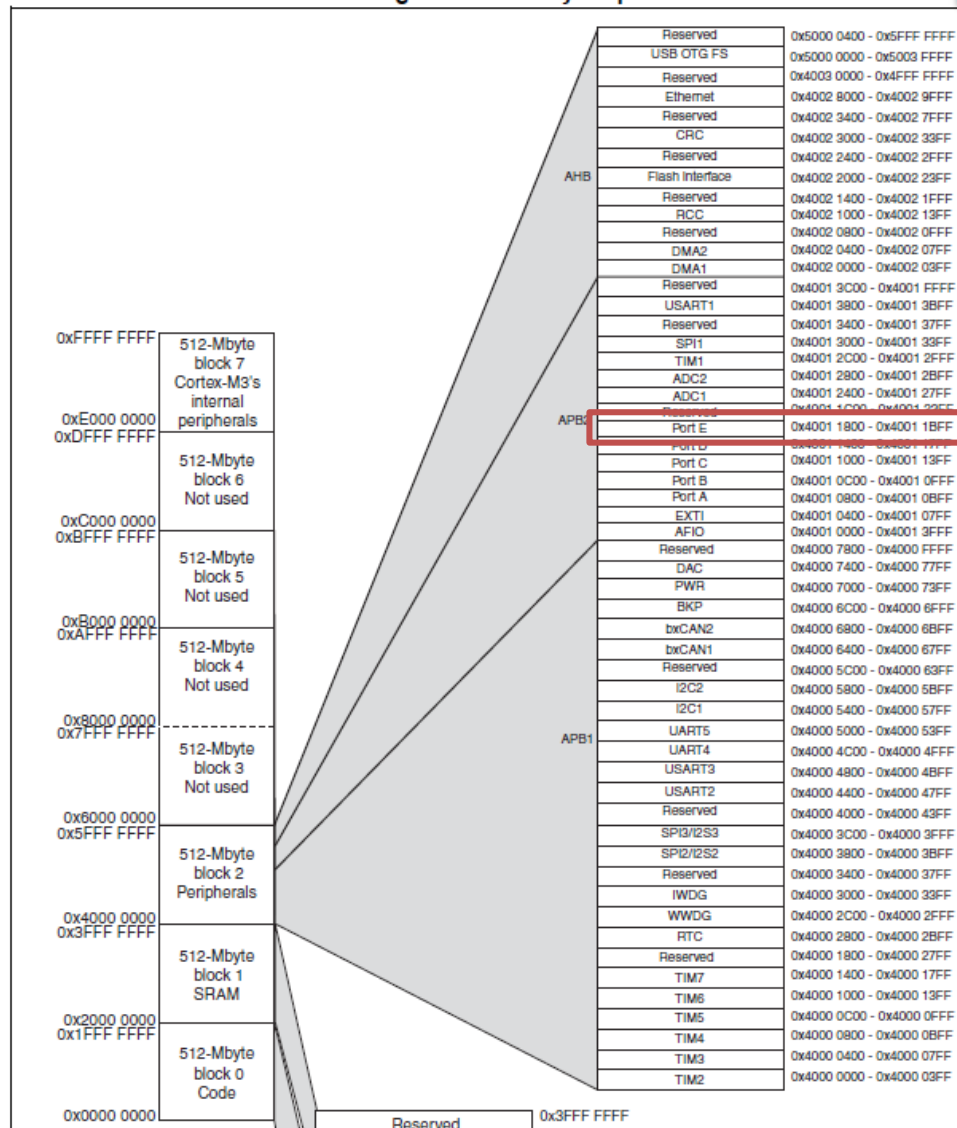
원하는 포트와 핀(GPIO)

사용하려는 GPIO Port, Pin의 input/

레퍼런스 메뉴얼 확인

- GPIO E의 BASE는
- 0x4002 1800
- GPIO E 5번 핀을 이용한다 가정
- (Set / Reset 할 수 있는데, Set 하겠다.)

Figure 5. Memory map



원하는 포트와 핀(GPIO)을 제어하기 위해 해야할 일들

사용하려는 GPIO Port, Pin의 input/output 설정 (Port Configuration)

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

레퍼런스 메뉴얼 확인

- GPIO E의 BASE는
- 0x4002 1800
- BSRR (Bit Set/Reset Register) Offset 은
- 0x10

- GPIO E 5번 핀을 이용한다 가정
- (Set / Reset 할 수 있는데, Set 하겠다.)

- 0x40021800
- + 0x00000010
- 0x40021010

- *((volatile unsigned int *)0x40021010) |= 0x0020; //(== 0b10 0000)

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

원하는 포트와 핀(GPIO)을 제어하기 위해 해야할 일들

사용하려는 GPIO Port, Pin의 input/output 설정 (Port Configuration)

- `*((volatile unsigned int *)0x40021010) |= 0x0020;`
- 0x40021010 일반 숫자
- (volatile unsigned int *)0x40021010 주소로 변환
- `*((volatile unsigned int *)0x40021010)` 주소에 있는 값
- `*((volatile unsigned int *)0x40021010) |= 0x0020;`
- |= 연산에서, b == 0 원래 있던 값은 상관하지 않겠다.

a	b	a b
0	0	0
1	0	1
0	1	1
1	1	1

7	6	5	4	3	2	1	0
0	0	1	0	0	0	0	0

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

원하는 포트와 핀(GPIO)을 제어하기 위해 해야할 일들

사용하려는 GPIO Port, Pin의 input/output 설정 (Port Configuration)

- `*((volatile unsigned int *)0x40021010) |= 0x0020;`
- 0x40021010 일반 숫자
- `(volatile unsigned int *)0x40021010` 주소로 변환
- `*((volatile unsigned int *)0x40021010)` 주소에 있는 값
- `*((volatile unsigned int *)0x40021010) |= 0x0020;`
- `|=` 연산에서, `b == 1` 원래 있던 값에 상관없이 1을 쓰겠다.
- (원래 있던 값과 상관없이) bit 5 만 1로 set하겠다.

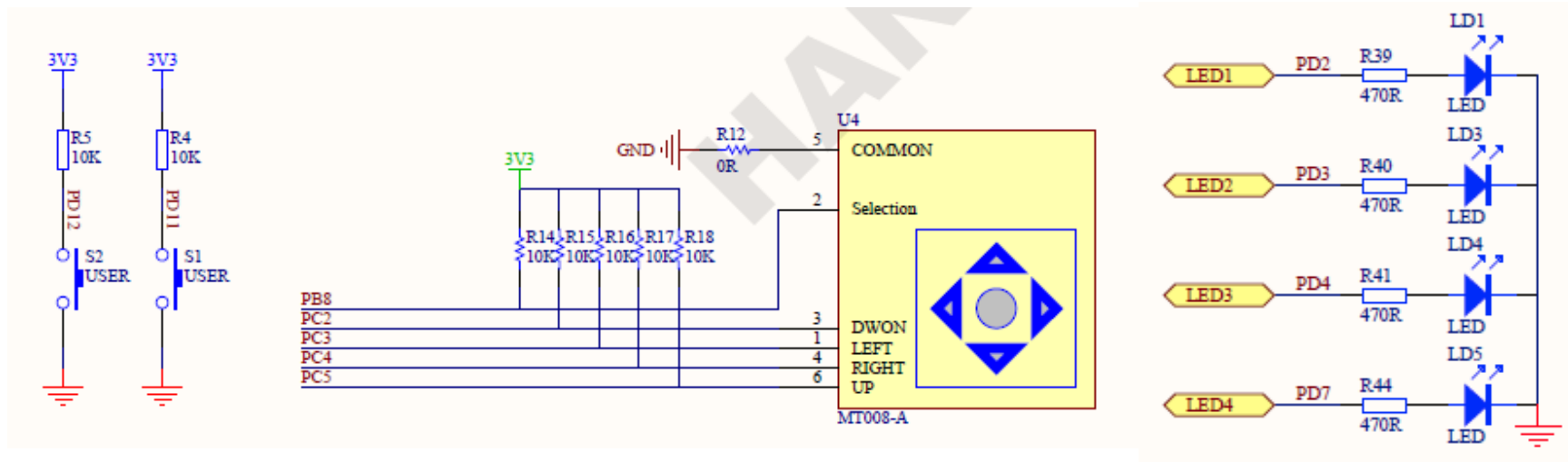
a	b	a b
0	0	0
1	0	1
0	1	1
1	1	1

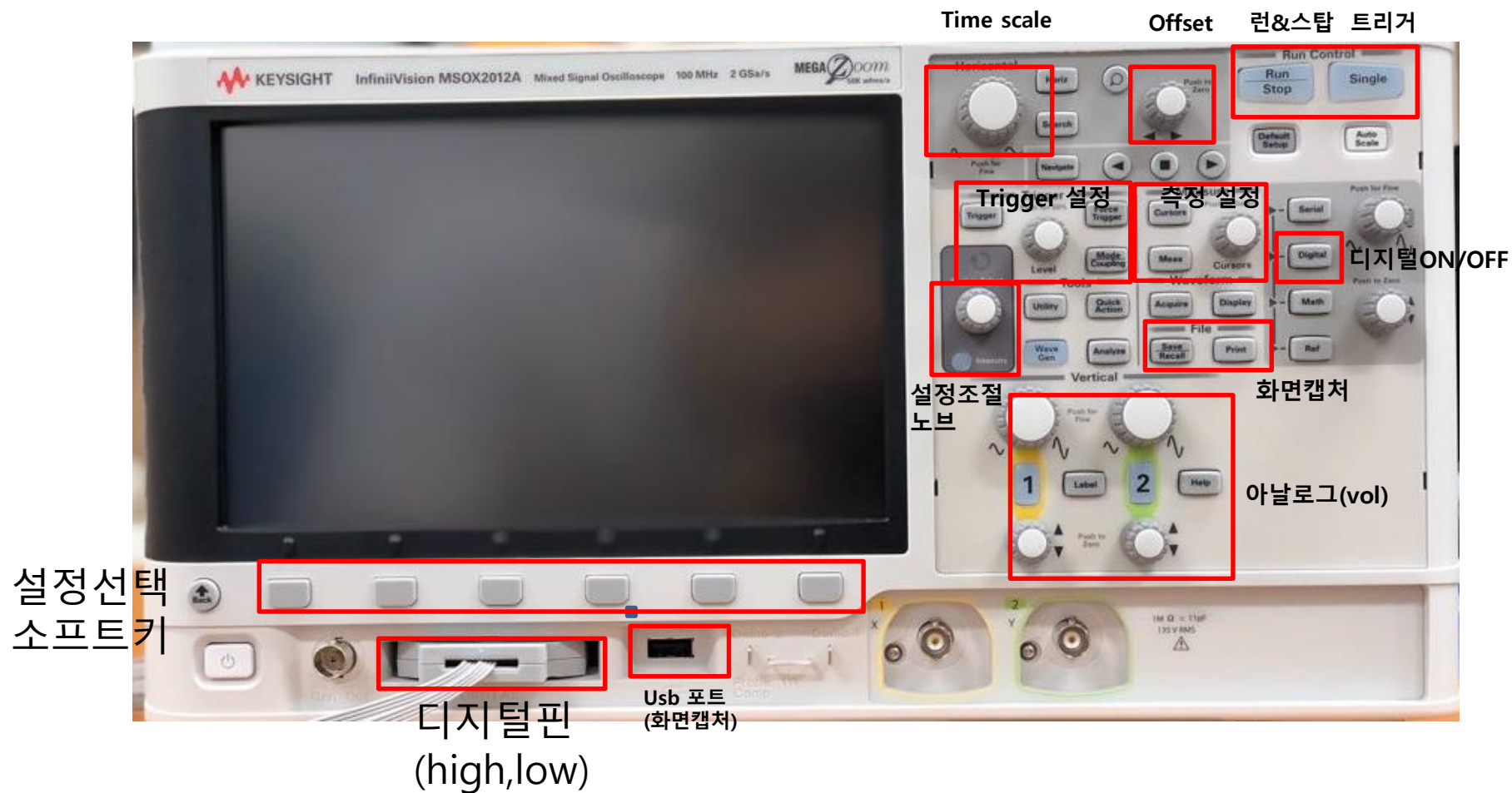
7	6	5	4	3	2	1	0
0	0	1	0	0	0	0	0

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

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.







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

이번 주 실험 결과 보고서

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

예비 발표 조는 발표 자료(영상) 만들어서
일요일 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!!!