

# Eclipse Embedded C/C++ Developer

## 설치 및 예제

Application-aware Computing for Embedded Systems Lab.  
School of Electronics Engineering, KNU, KOREA



# 1. GNU ARM Embedded Toolchain 설치

1. 아래의 링크에서 최신 버전의 GNU ARM Embedded Toolchain 다운로드 (작성 시점에는 10.3-2021.10이 최신 버전)

<https://developer.arm.com/tools-and-software/open-source-software/developer-tools/gnu-toolchain/gnu-arm-eabi/10-2021-10>

## What's new in 10.3-2021.10

This release provides mitigation for the [VLLDM instruction security vulnerability](#).

### In this release:

1

[gcc-arm-none-eabi-10.3-2021.10-win32.exe](#)

Windows 32-bit Installer (Signed for Windows 10 and later) (Formerly SHA2 signed binary)

MD5: 8d0f75f33f9e3d5f9600197626297212

2

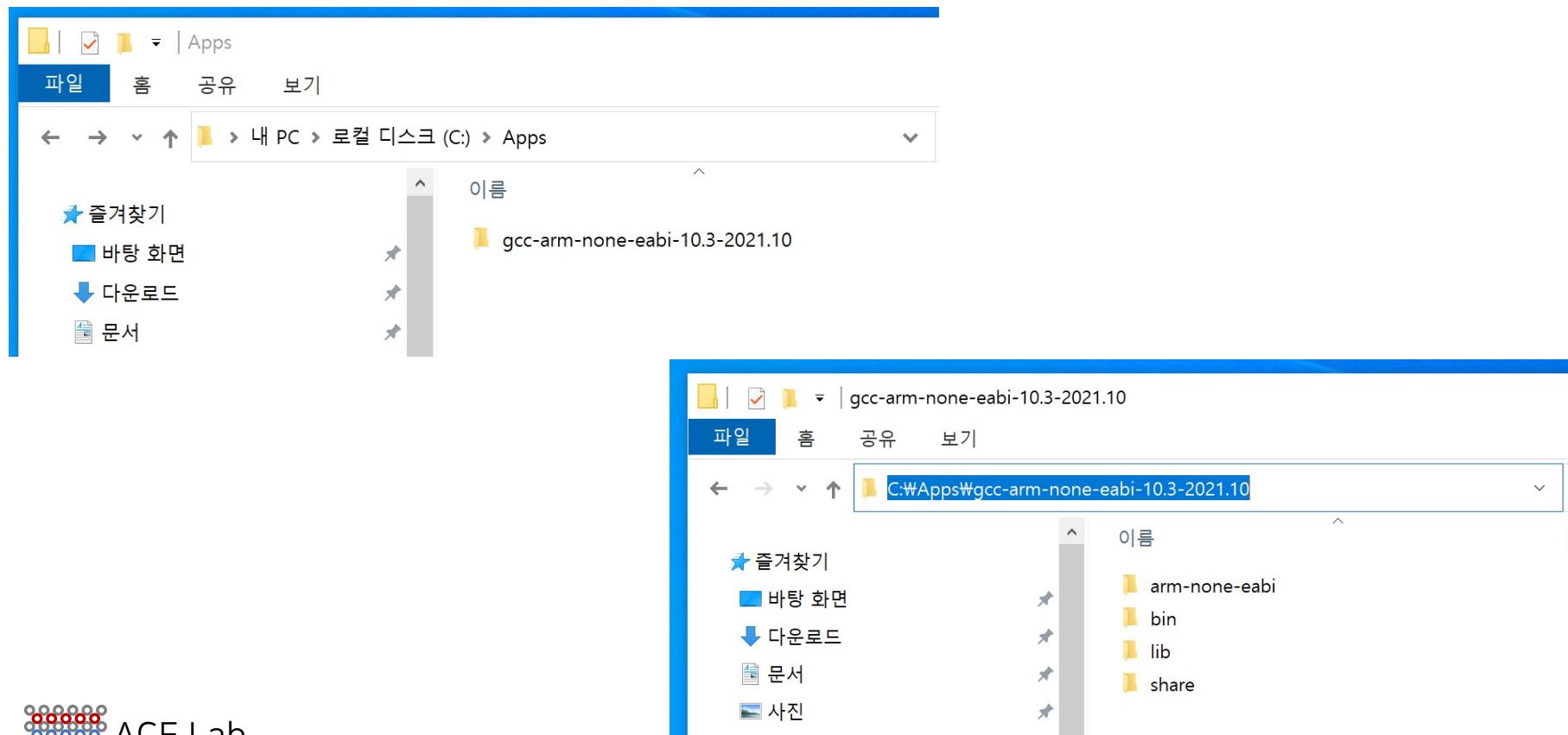
[gcc-arm-none-eabi-10.3-2021.10-win32.zip](#)

Windows 32-bit ZIP package

MD5: 2bc8f0c4c4659f8259c8176223eeafc1

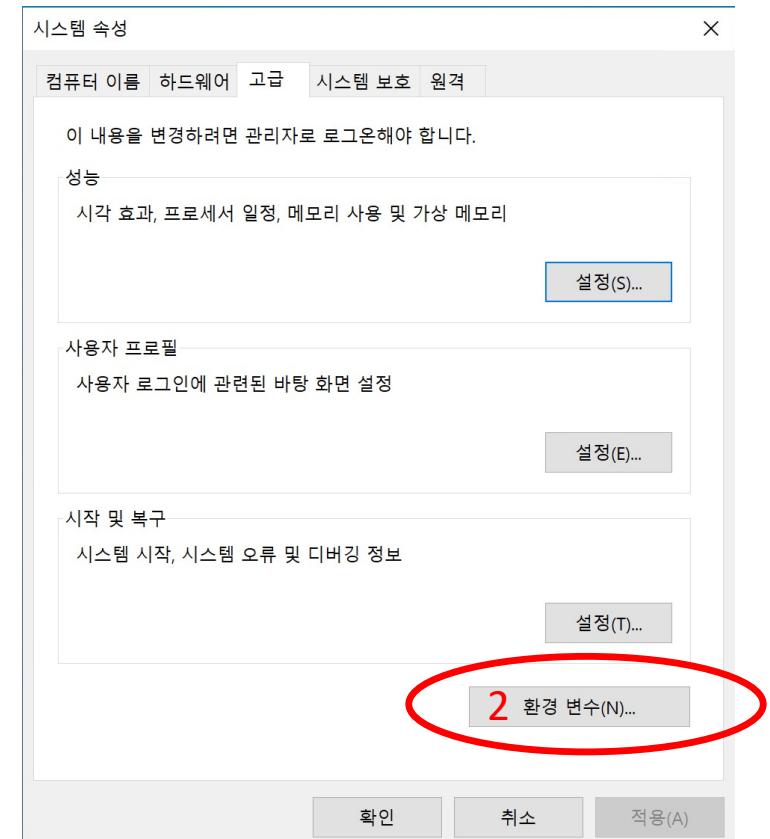
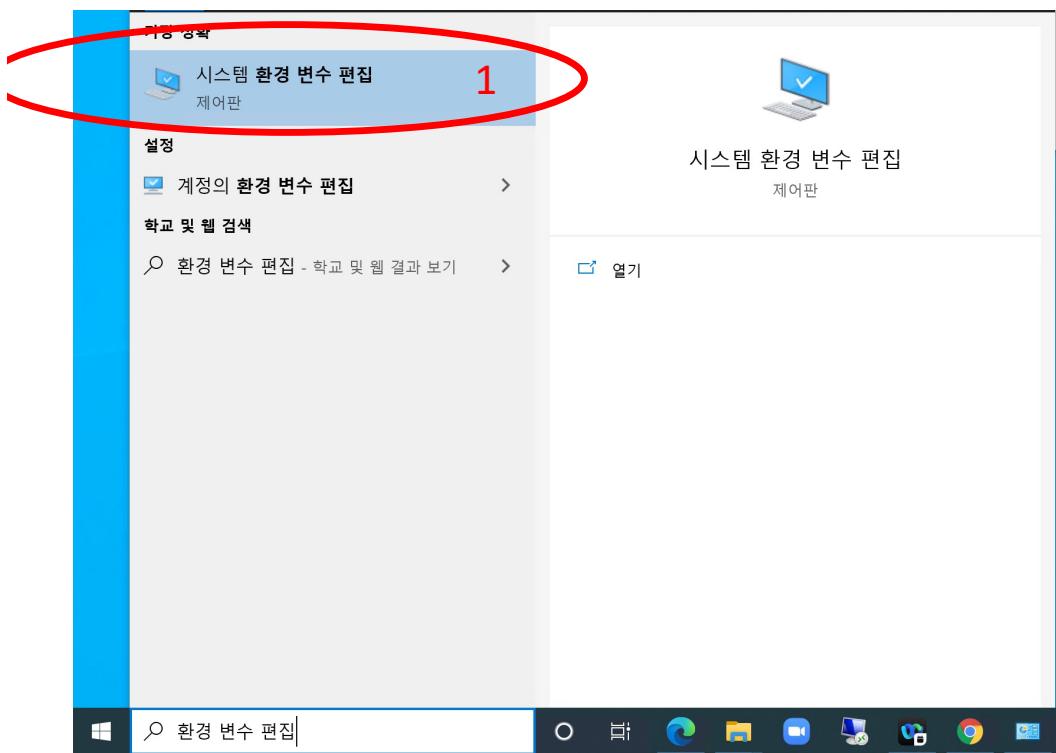
# 1. GNU ARM Embedded Toolchain 설치

2. 압축 해제 후 C:\Apps 디렉토리에 아래와 같이 복사



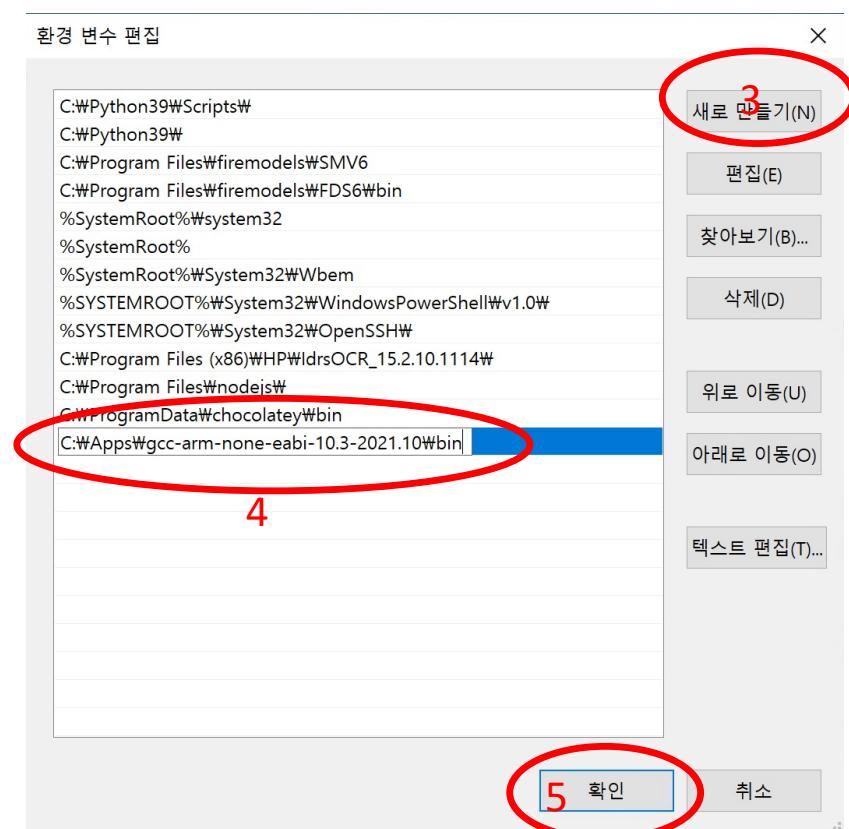
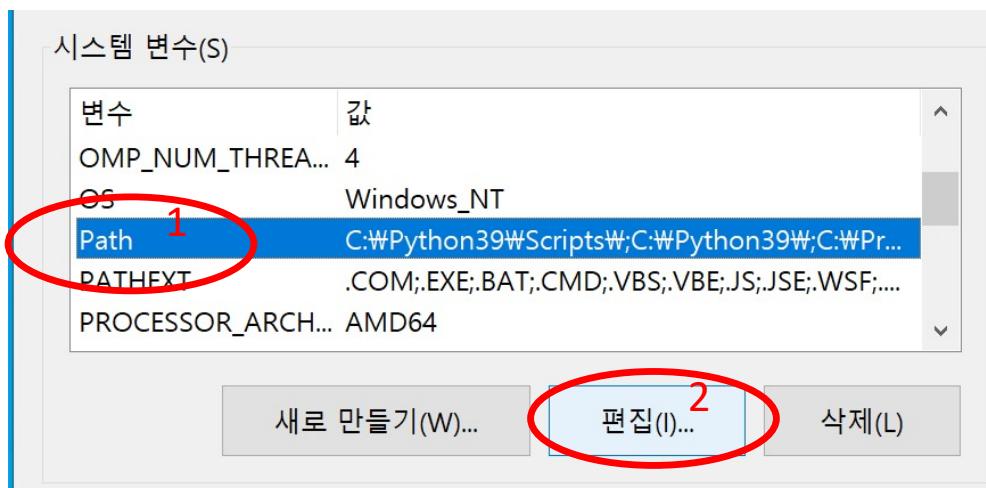
# 1. GNU ARM Embedded Toolchain 설치

3. GNU ARM Embedded Toolchain의 경로를 설정하기 위해 시스템 속성의 '환경 변수'를 선택한다.



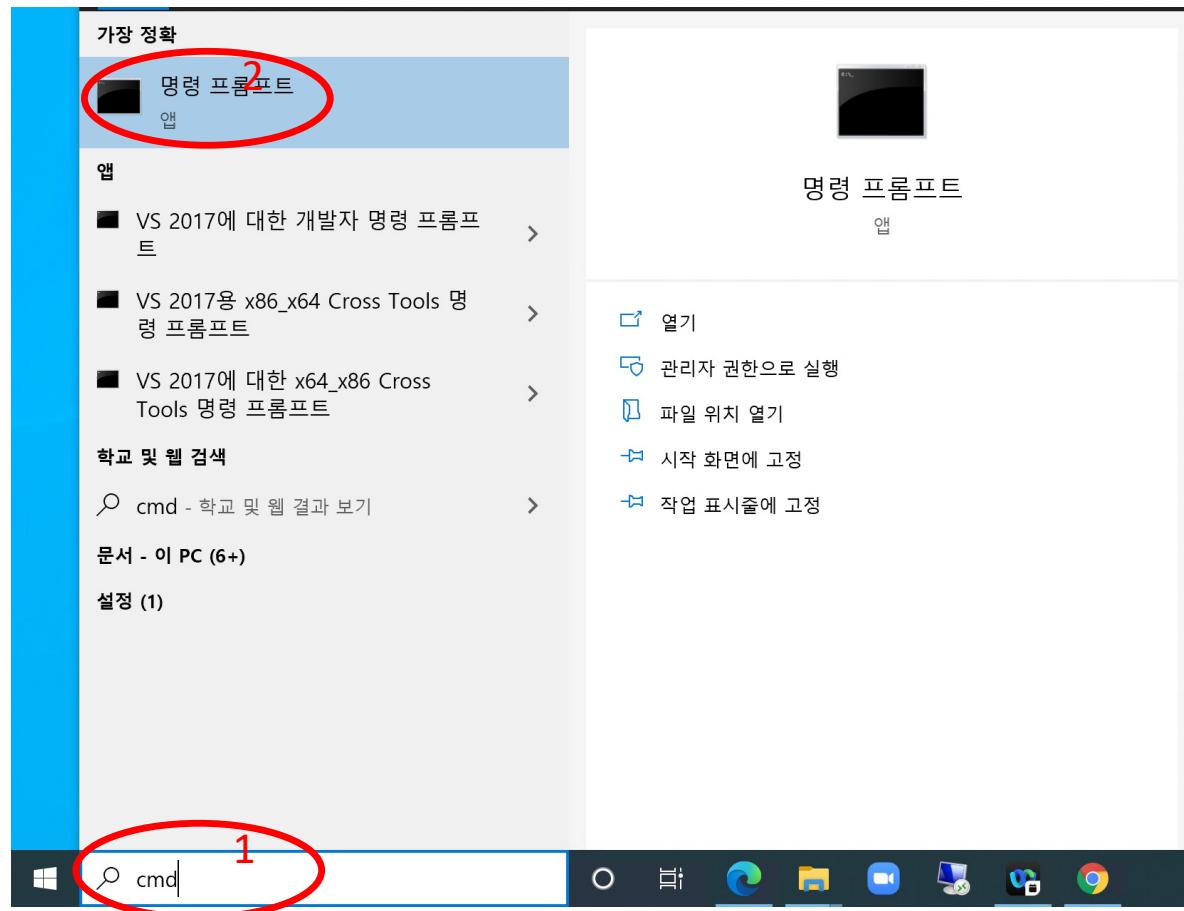
# 1. GNU ARM Embedded Toolchain 설치

4. GNU ARM Embedded Toolchain의 경로를 설정하기 위해 시스템 속성의 '환경 변수'를 선택한다.



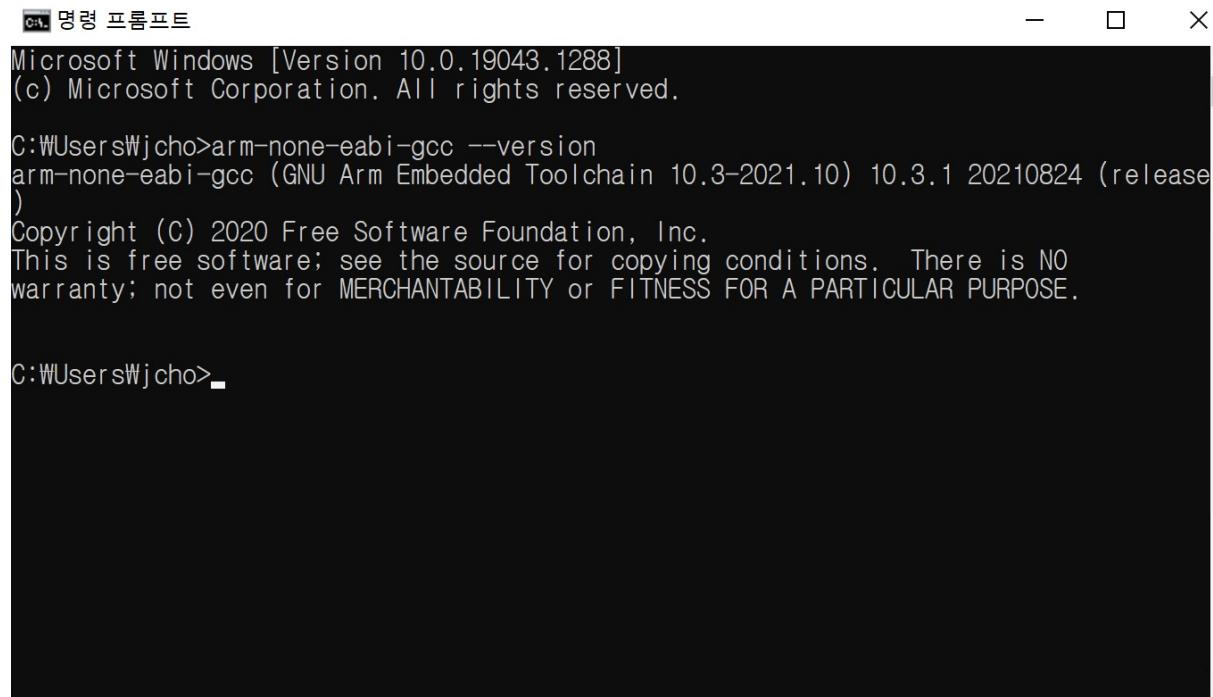
# 1. GNU ARM Embedded Toolchain 설치

5. GNU ARM Embedded Toolchain의 경로를 확인하기 위해 검색창에 cmd를 입력하고 “명령 프롬프트”를 선택한다.



# 1. GNU ARM Embedded Toolchain 설치

## 6. arm-none-eabi-gcc --version 명령어를 입력하여 툴체인의 정상 설치 확인



C:\ 명령 프롬프트

```
Microsoft Windows [Version 10.0.19043.1288]
(c) Microsoft Corporation. All rights reserved.

C:\Users\jcho>arm-none-eabi-gcc --version
arm-none-eabi-gcc (GNU Arm Embedded Toolchain 10.3-2021.10) 10.3.1 20210824 (release)
Copyright (C) 2020 Free Software Foundation, Inc.
This is free software; see the source for copying conditions. There is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

C:\Users\jcho>
```

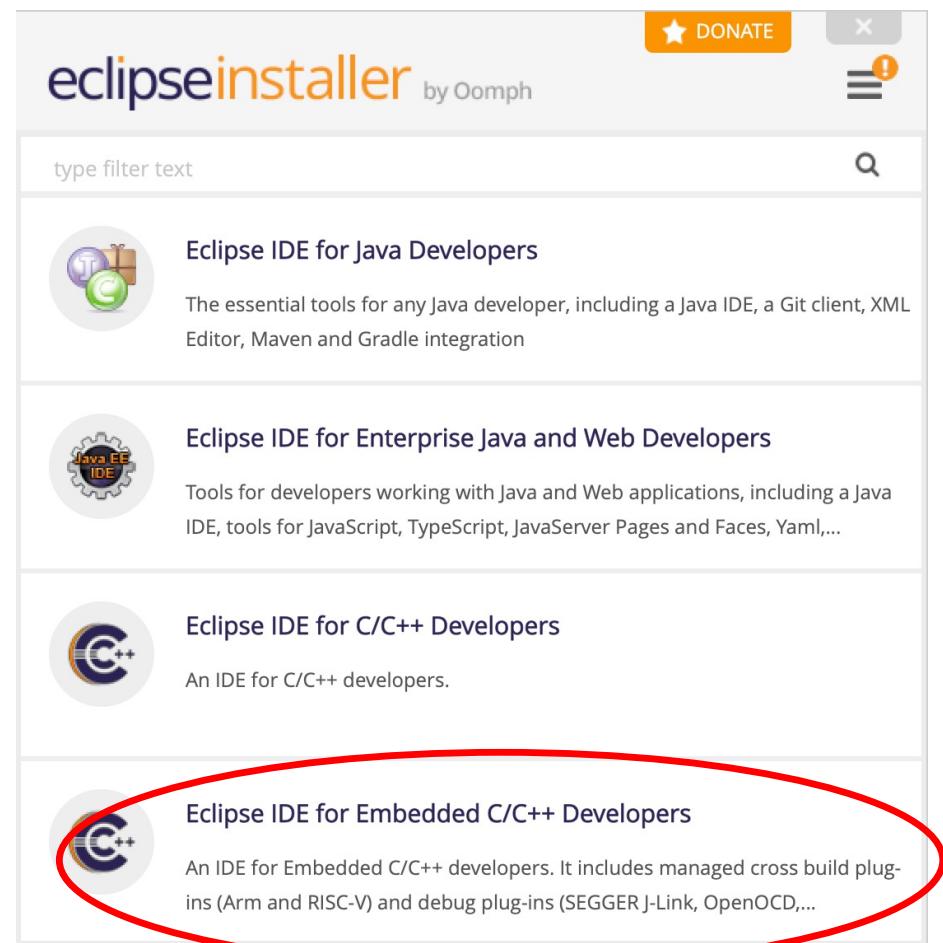
# 2. Eclipse 설치

1. <http://eclipse.org/downloads>에서 Eclipse installer 다운로드

The screenshot shows the Eclipse Foundation website's download section. At the top, there's a navigation bar with links for 'Log in', 'Manage Cookies', 'Projects', 'Working Groups', 'Members', and 'More'. Below the navigation, a large orange banner on the left says 'Download Eclipse Technology that is right for you'. To the right of the banner is a sponsored advertisement for Red Hat Developer, which includes a call to action 'Join the DevNation!'. Below the ad, it says 'Sponsored Ad'. In the center, there's a dark blue box containing the text 'The Eclipse Installer 2021-09 R now includes a JRE for macOS, Windows and Linux.' Below this box is the Eclipse logo. To the right of the logo, there's a section titled 'Tool Platforms' featuring the Eclipse Che logo and a brief description: 'Eclipse Che is a developer workspace server and cloud IDE.' Further down, there's a button labeled 'Get Eclipse IDE 2021-09' with the sub-instruction 'Install your favorite desktop IDE packages.' and a 'Download x86\_64' button. At the bottom of the main content area, there are links for 'Download Packages' and 'Need Help?'. The overall background of the main content area is light gray.

## 2. Eclipse 설치

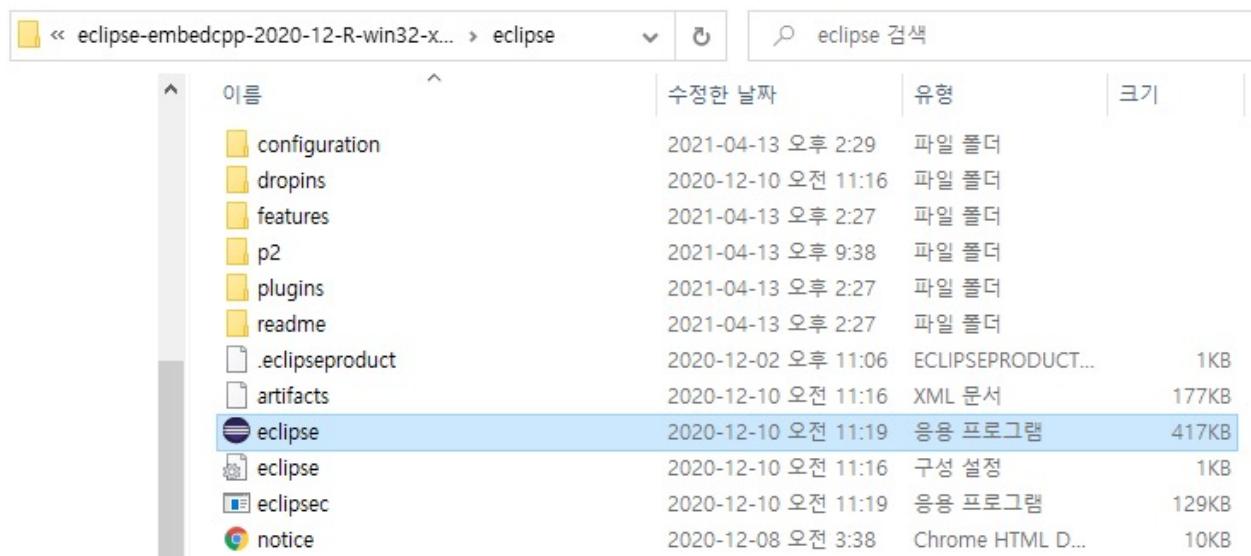
2. Eclipse installer를 실행하고 Eclipse IDE for Embedded C/C++ Developers 선택



# 3. LED\_ON 예제

## 1. Eclipse 실행

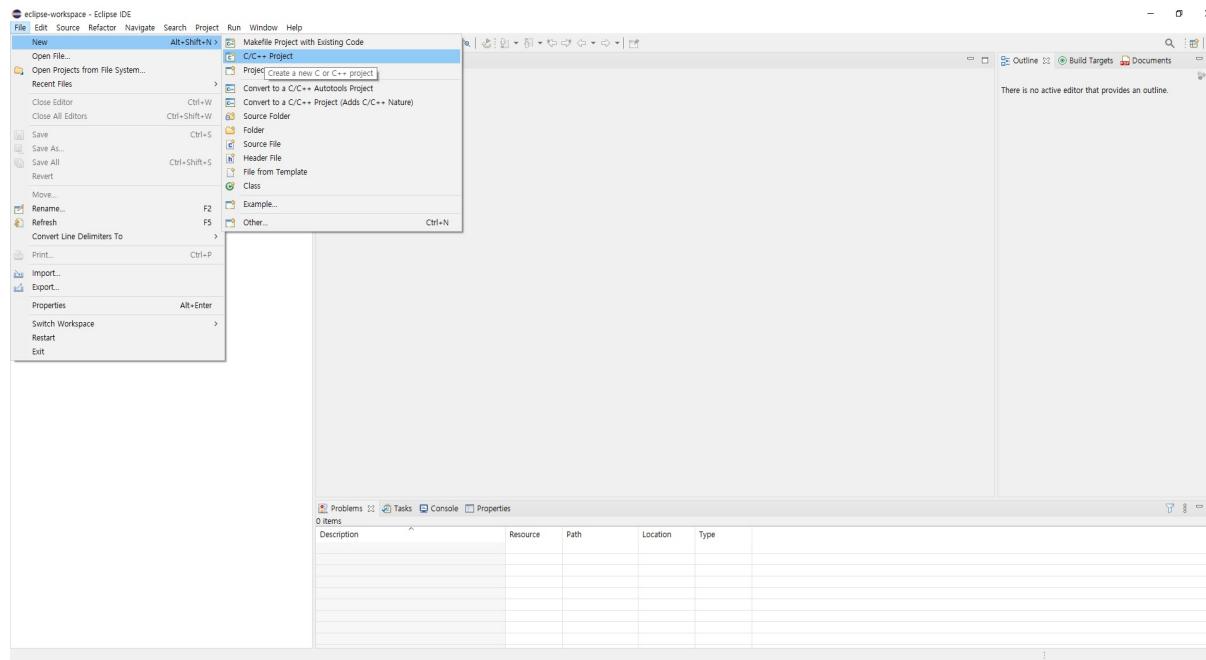
- ✓ 설치한 경로에서 Eclipse를 실행



# 3. LED\_ON 예제

## 2. 프로젝트 생성

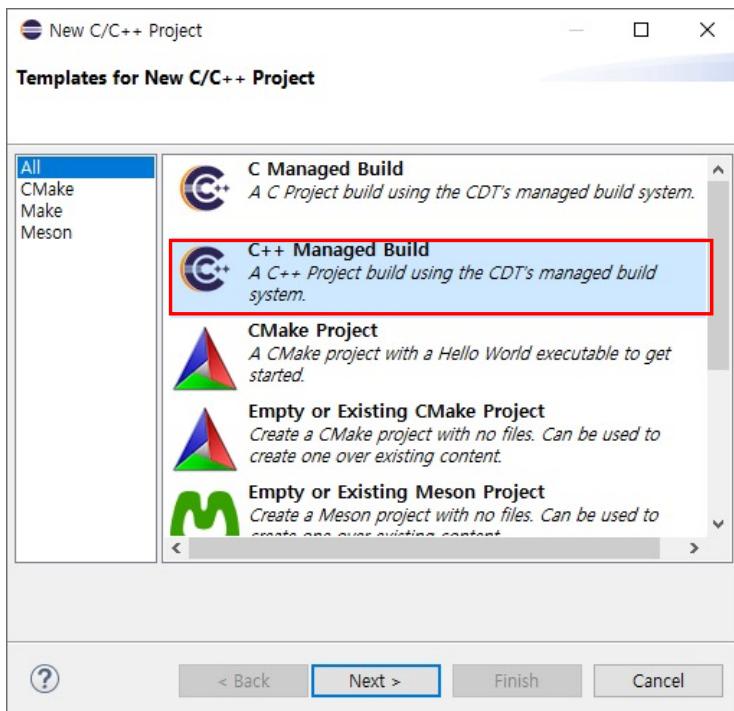
- ✓ File->new->C/C++ Project를 실행



# 3. LED\_ON 예제

## 2. 프로젝트 생성

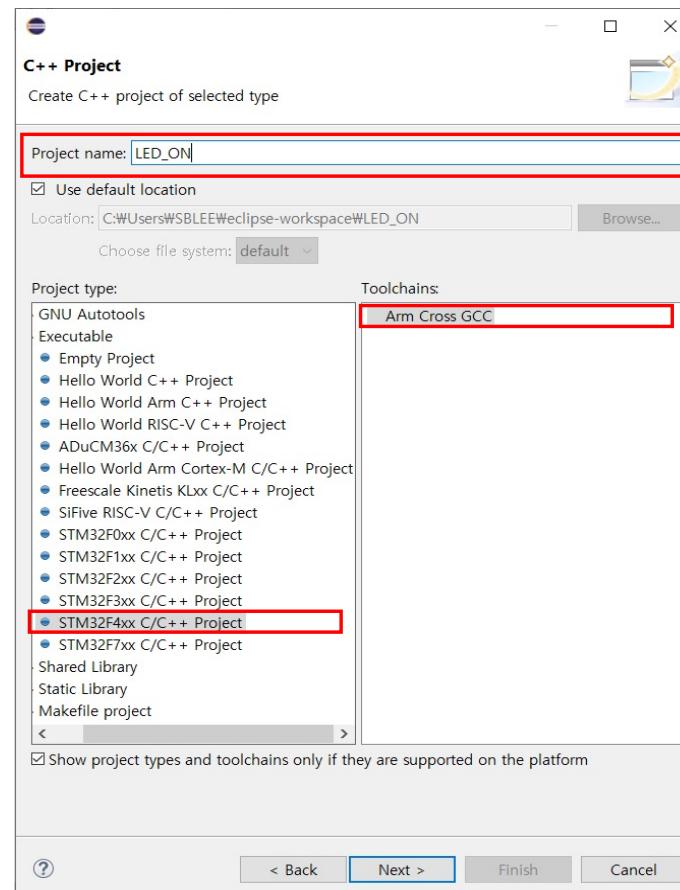
- ✓ C++ Managed Build 선택하고 Next



# 3. LED\_ON 예제

## 2. 프로젝트 생성

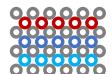
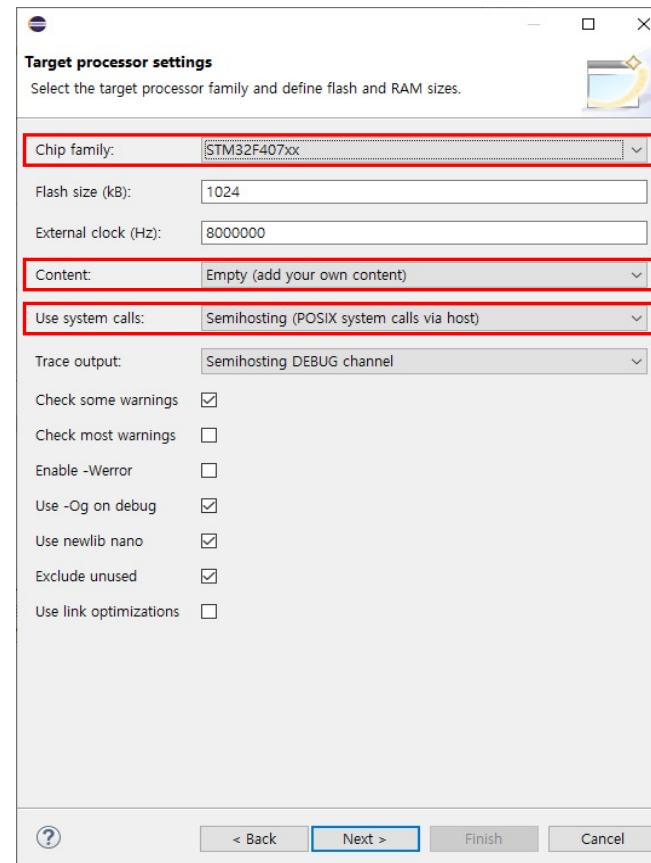
- ✓ 프로젝트 이름에 LED\_EX를 입력
- ✓ Project type을 STM32F4xx C/C++ Project로 선택
- ✓ Toolchains를 Arm Cross GCC로 선택



# 3. LED\_ON 예제

## 2. 프로젝트 생성

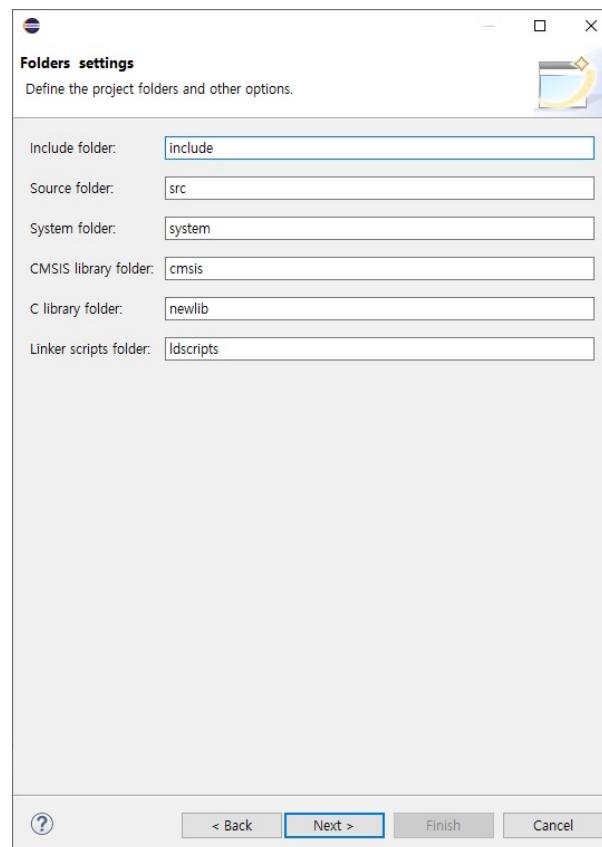
- ✓ Chip family : STM32F407xx
- ✓ Content : Empty
- ✓ Use system calls : Semihosting



# 3. LED\_ON 예제

## 2. 프로젝트 생성

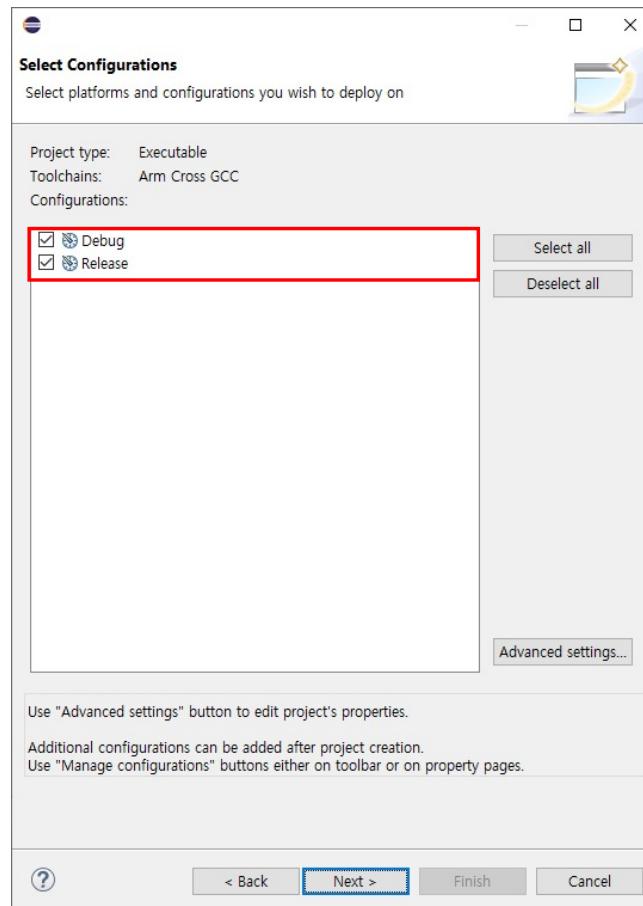
- ✓ 건드리지 않고 넘어간다.



# 3. LED\_ON 예제

## 2. 프로젝트 생성

- ✓ Debug/Release를 모두 선택한다.

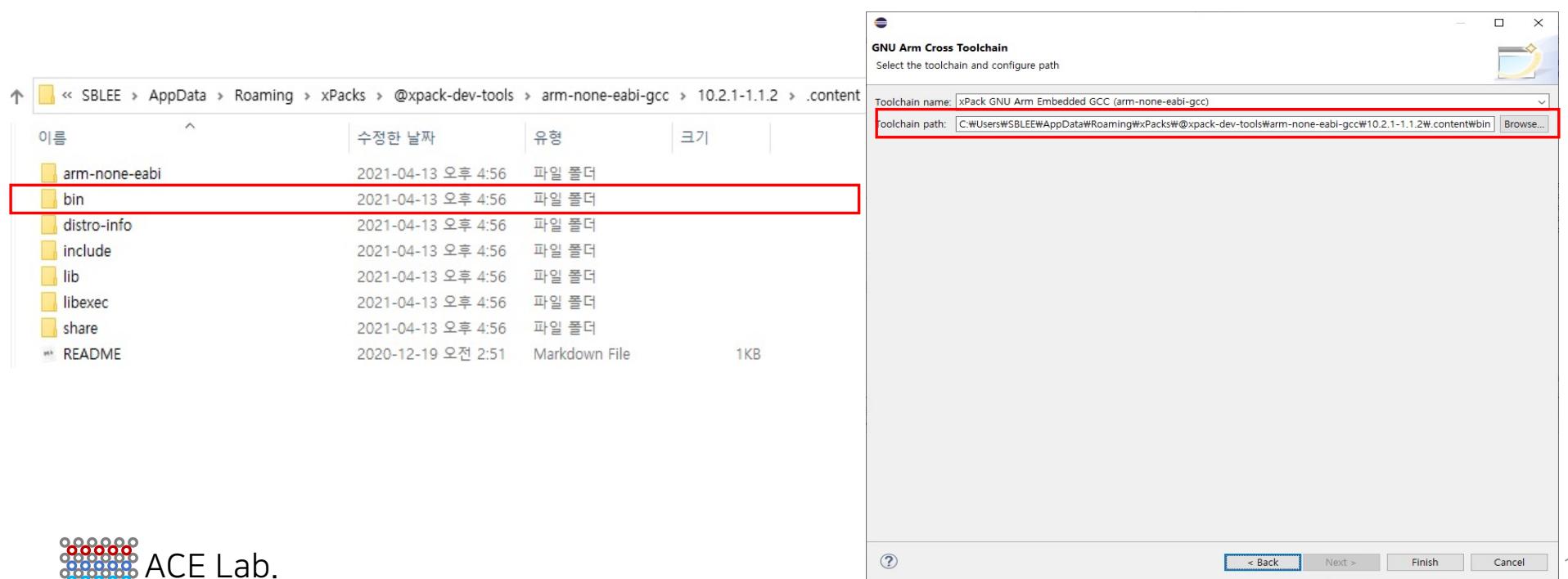


# 3. LED\_ON 예제

## 2. 프로젝트 생성

- ✓ 툴 체인 경로는 자동으로 등록됨.
- ✓ 등록이 안된 경우 수동으로 등록할 툴 체인 경로

C:\Users\AppBarData\Roaming\xPacks@\xpck-dev-tools\arm-none-eabi-gcc\10.2.1-1.1.2\content\bin



# 3. LED\_ON 예제

## 3. 소스코드(main.cpp) 작성

- ✓ RCC/GPIO 설정관련 레지스터 주소 및 비트 필드 정의

```
// RCC Registers
#define RCC_BASE      (0x40023800)
#define RCC_AHB1      *((volatile unsigned*)(RCC_BASE + 0x30))

#define GPIOD_EN_BIT  3

// GPIO Registers
#define GPIOD_BASE    (0x40020C00)
#define GPIOD_MODER   *((volatile unsigned*)(GPIOD_BASE + 0x00))
#define GPIOD_OTYPER   *((volatile unsigned*)(GPIOD_BASE + 0x04))
#define GPIOD_OSPEEDR  *((volatile unsigned*)(GPIOD_BASE + 0x08))
#define GPIOD_PUPDR   *((volatile unsigned*)(GPIOD_BASE + 0x0C))
#define GPIOD_BSRR   *((volatile unsigned*)(GPIOD_BASE + 0x18))

#define LED  GREEN
#define GREEN (12)
#define ORANGE (13)
#define RED   (14)
#define BLUE  (15)
```

GREEN / ORANGE / RED / BLUE LED를 선택하여 ON시킬 수 있음



ACE Lab.

- ✓ RCC/GPIO 설정 코드

```
void PORT_init(void)
{
    RCC_AHB1 |= (1<<GPIOD_EN_BIT);           /* PORTA Clock enable */

    GPIOD_MODER &= ~((0b11)<<(LED*2));       /* Output MODE */
    GPIOD_MODER |= ((0b01)<<(LED*2));

    GPIOD_OTYPER &= ~(1<<LED);                /* Output push-pull */
    GPIOD_OSPEEDR &= ~((0b11)<<(LED*2));       /* High speed */
    GPIOD_OSPEEDR |= ((0b10)<<(LED*2));

    GPIOD_PUPDR &= ~((0b11)<<(LED*2));       /* High speed */
    GPIOD_PUPDR |= ((0b01)<<(LED*2));

    GPIOD_BSRR |= (1<<(LED+16));             /* LED OFF */
}
```

- ✓ main 함수

```
int main(int argc, char* argv[])
{
    PORT_init();                  /* PORT configure */
    GPIOD_BSRR |= (1<<LED); /* LED ON */

    for(;;{})

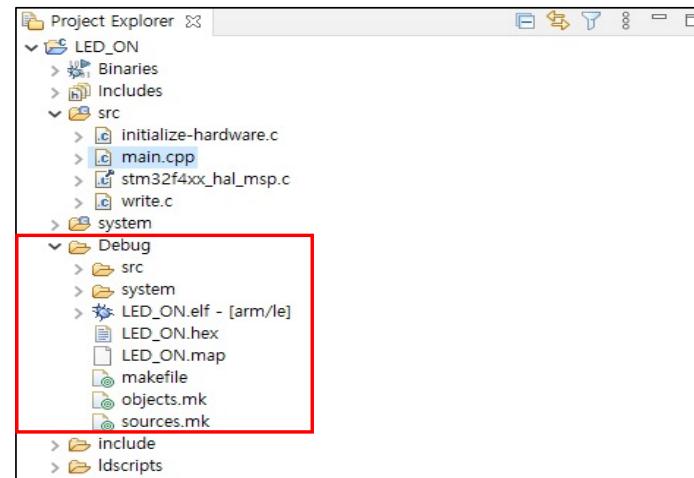
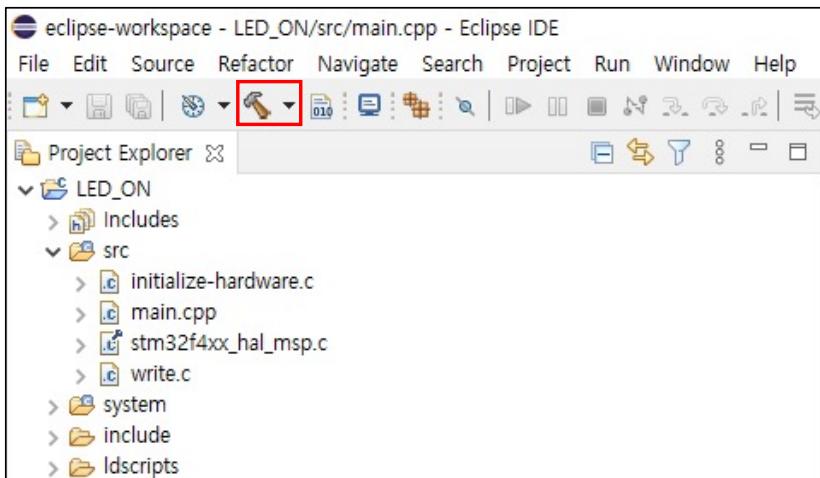
    return 0;
}
```

# 3. LED ON 예제

## 4. 프로젝트 빌드

- ✓ 망치 아이콘을 클릭하여 프로젝트를 빌드한다.
- ✓ 빌드가 완료되면 콘솔창을 통해 확인할 수 있다.
- ✓ 빌드의 결과물이 Debug 폴더 아래에 생성된다.

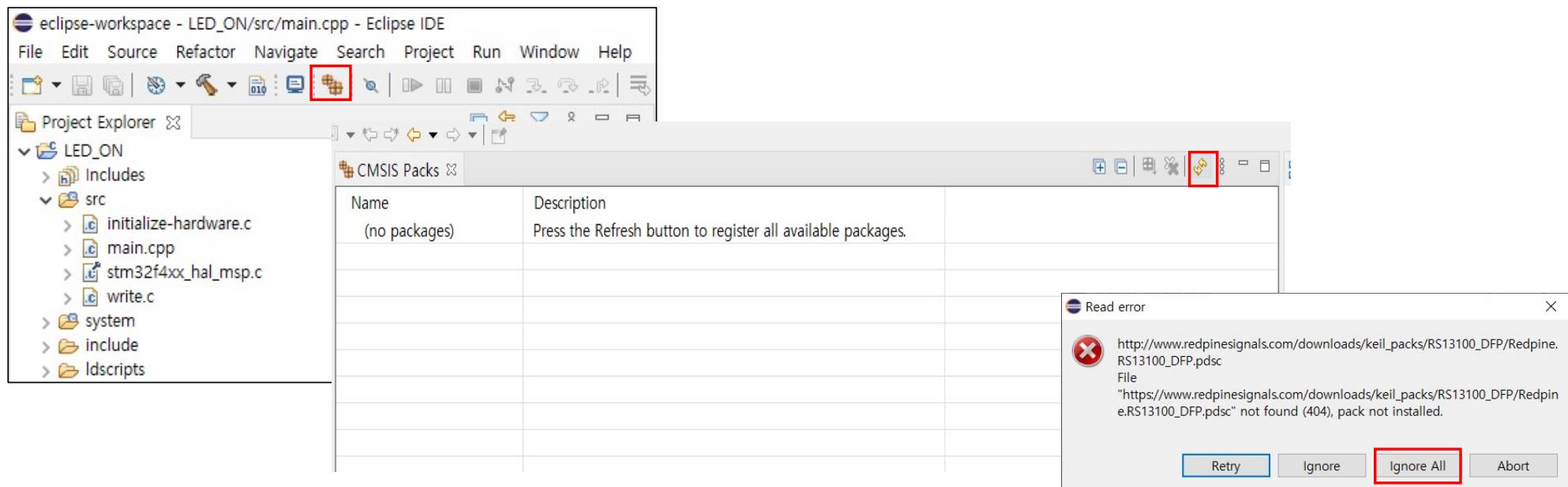
The screenshot shows the Eclipse CDT Build Console window. It displays the command: "Invoking: GNU Arm Cross Print Size arm-none-eabi-size --format=berkeley \"LED\_ON.elf\"". Below it, the size output is shown: "text data bss dec hex filename 9199 160 708 10067 2753 LED\_ON.elf". At the bottom, it says "Finished building: LED\_ON.siz". The status bar at the bottom indicates "13:29:10 Build Finished. 0 errors, 0 warnings. (took 1s.987ms)".



# 3. LED\_ON 예제

## 5. Boards/Devices 선택

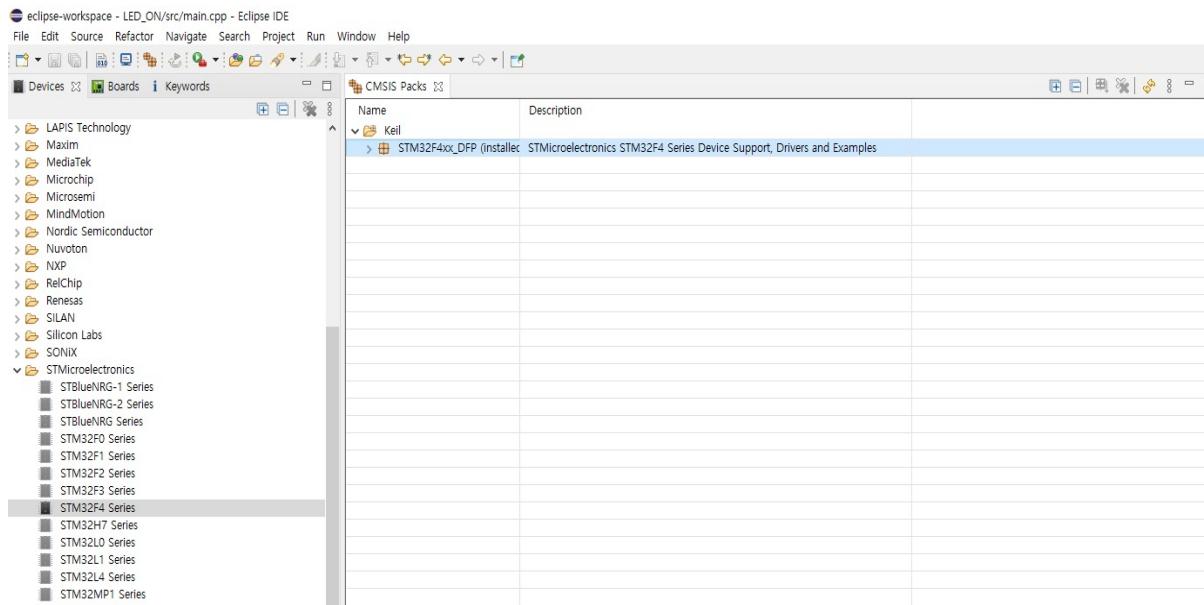
- ✓ CMSIS PACK을 클릭한다.
- ✓ CMSIS PACK을 업데이트 한다.
- ✓ 오른쪽 아래와 같은 에러가 발생하면 Ignore All을 클릭한다.



# 3. LED\_ON 예제

## 5. Boards/Devices 선택

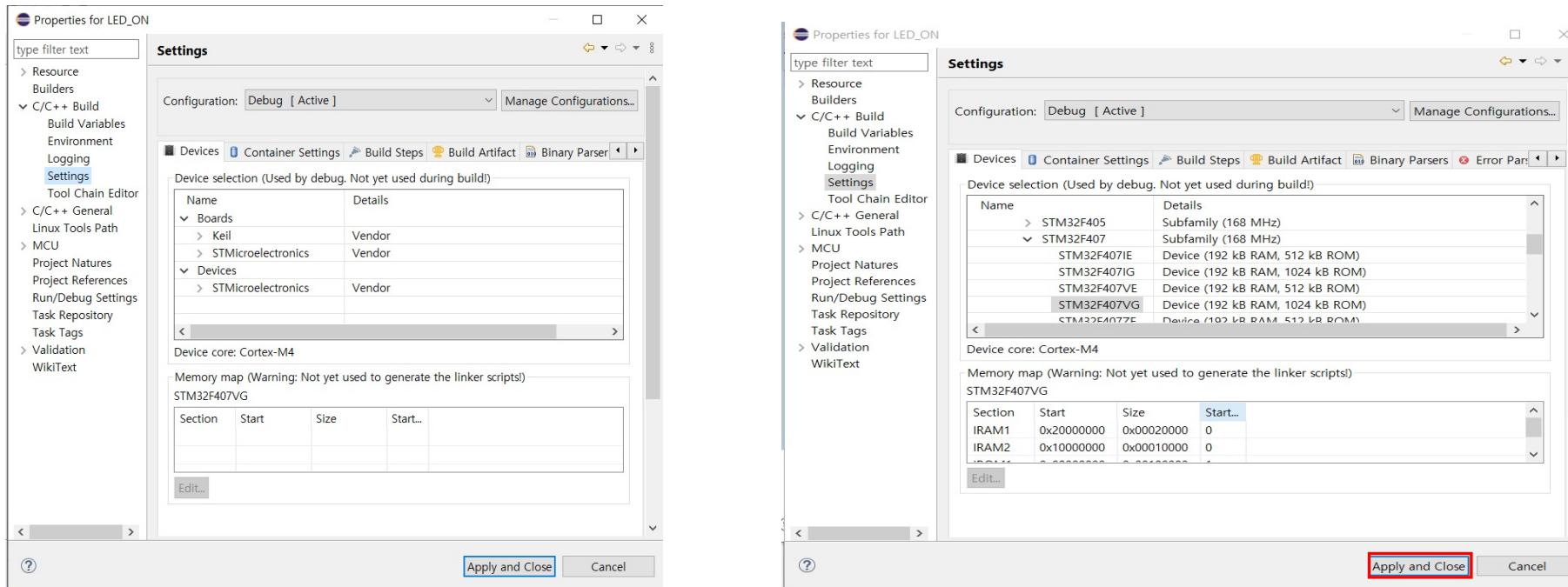
- ✓ 업데이트가 완료되면 install 할 수 있는 Device/Board가 보이게 된다.
- ✓ Devices 탭에서 STMicroelectronics 폴더 내 STM32F4 Series를 클릭한다.
- ✓ STM32F4xx\_DFP를 install 한다.



# 3. LED\_ON 예제

## 5. Boards/Devices 선택

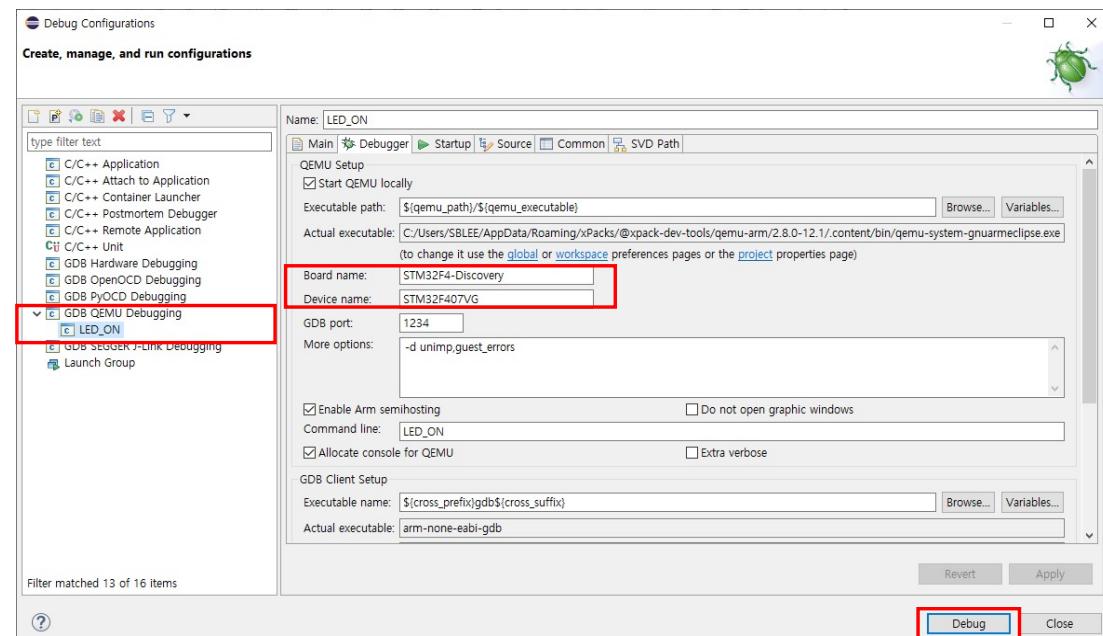
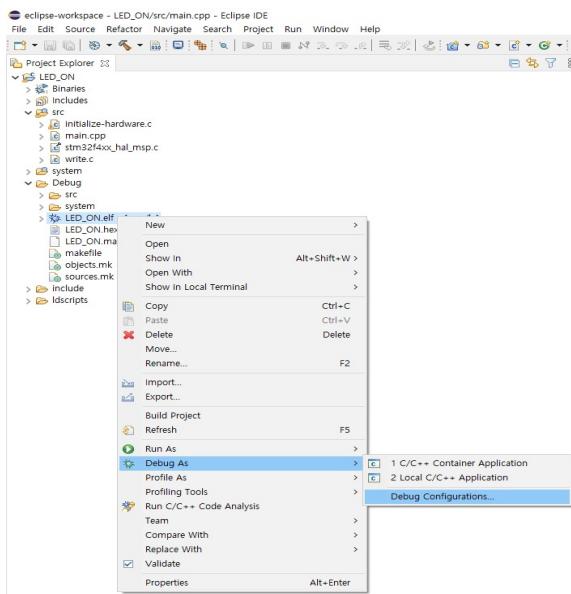
- ✓ 프로젝트 우 클릭 -> Properties -> C/C++ Build -> Settings 탭 선택
- ✓ Devices -> STMicroelectronics -> STM32F4 Series -> STM32F407 -> STM32F407VG 선택



# 3. LED\_ON 예제

## 6. 디버깅

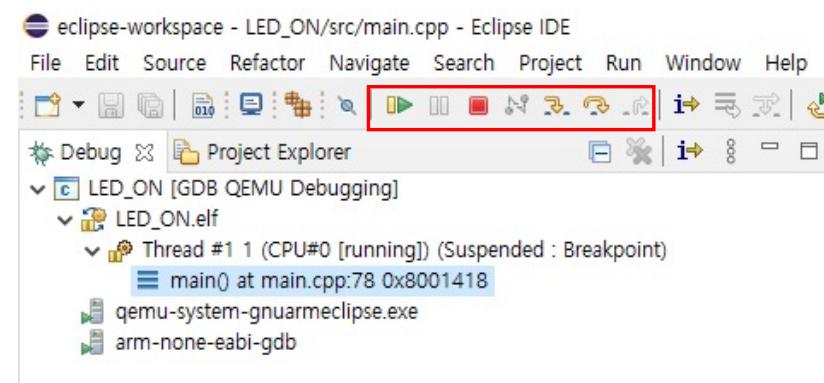
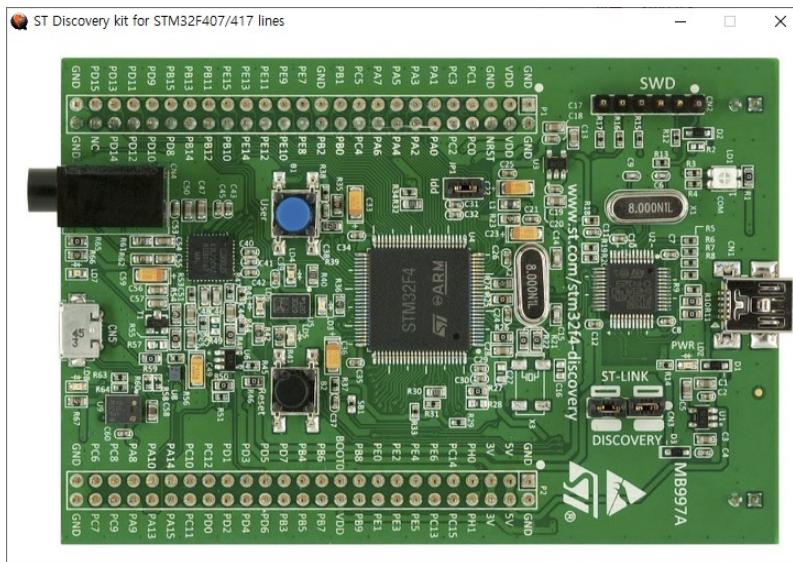
- ✓ LED\_ON.elf 오른쪽으로 클릭 -> Debug As -> Debug Configurations..
- ✓ GDB QEMU Debugging 더블 클릭
- ✓ Board name : STM32F4-Discovery / Device name : STM32F407VG 이 입력되어 있지 않은 경우 입력
- ✓ Debug 클릭



# 3. LED\_ON 예제

## 6. 디버깅

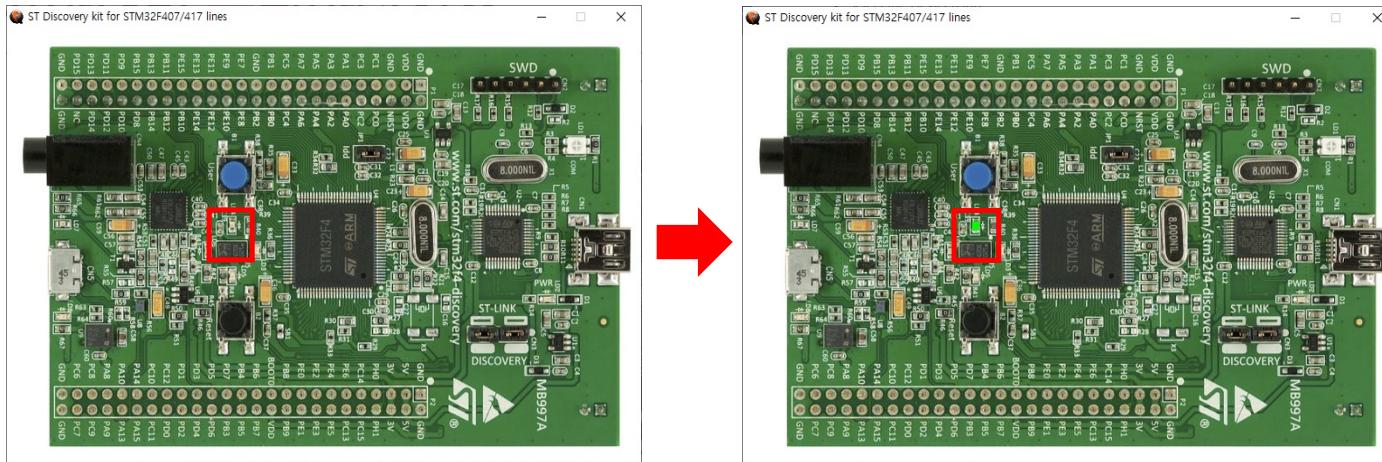
- ✓ 그래픽 윈도우가 생성된다.
- ✓ Run 버튼을 누르면 동작이 수행된다.



# 3. LED\_ON 예제

## 7. 실행결과 확인

- ✓ 초록 LED가 ON된 것을 확인할 수 있음



# 4. hello\_world 예제

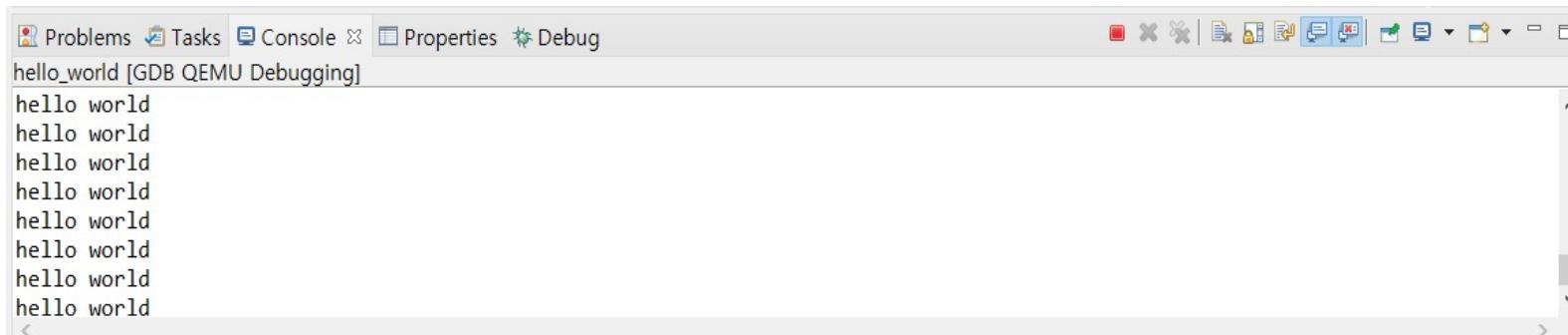
1. LED\_ON과 동일한 방식으로 hello\_world 프로젝트를 생성

✓ 소스코드

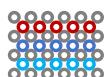
```
int main(int argc, char* argv[])
{
    for (;;)
    {
        fprintf(stdout, "hello world \n");
    }
    return 0;
}
```

✓ 빌드 / 디버깅

✓ 실행 결과

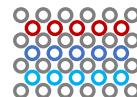


The screenshot shows a software interface with a toolbar at the top and several tabs: Problems, Tasks, Console, Properties, and Debug. The 'Console' tab is active, displaying the text 'hello world' repeated eight times. The interface has a light gray background with dark gray toolbars and tabs.



# Q & A

Thank you for your attention



Application-aware Computing  
for Embedded Systems Lab.

School of Electronics Engineering, KNU  
ACES Lab