
개발환경 구축

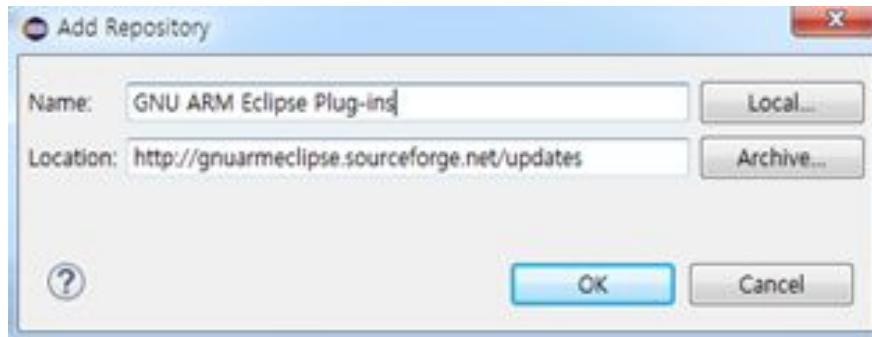
Hancheol Cho

개발환경 설치 순서

- JAVA SE(JDK) 설치
- Eclipse IDE for C/C++ Developers 설치
- GNU ARM Eclipse Plug-in 설치
- Windows Build Tools 설치
- GNU ARM GCC 설치
- OpenOCD 설치
- STLink-V2 드라이버 설치
- USB Serial 드라이버 설치
- DFU 드라이버 설치

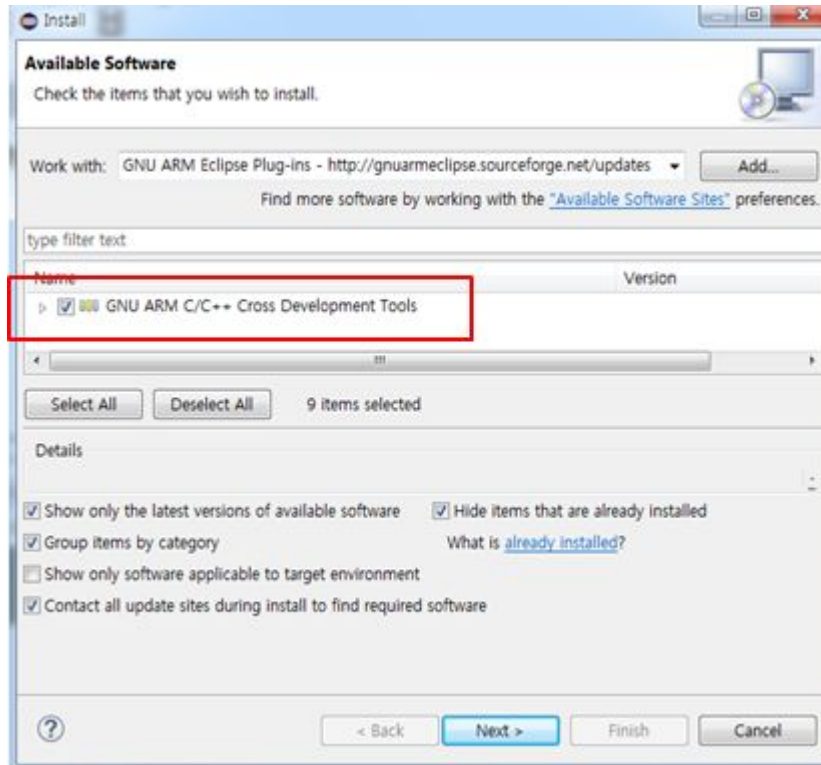
개발환경 설치

- Java SE (JDK) 설치
 - <http://www.oracle.com/technetwork/java/javase/downloads/index.html>
- Eclipse IDE for C/C++ Developers 설치
 - <https://eclipse.org/downloads/>
- GNU ARM Eclipse Plug-in 설치
 - Eclipse 실행 후 Help->Install New Software->Add 선택 후 내용 입력
 - Name : GNU ARM Eclipse Plug-ins
 - Loc : <http://gnuarmclipse.sourceforge.net/updates>



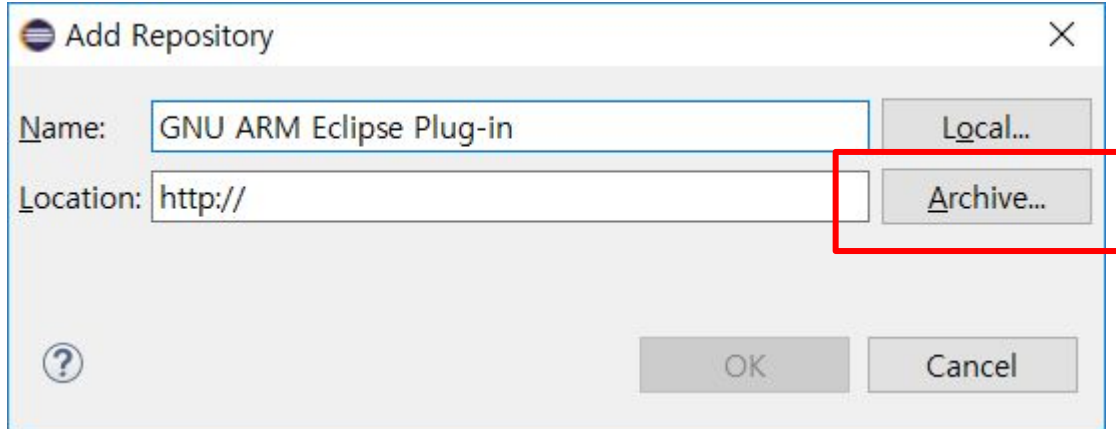
개발환경 설치

- GNU ARM Eclipse Plug-in 설치
 - GNU ARM C/C++ Cross Development Tools 선택 후 설치함



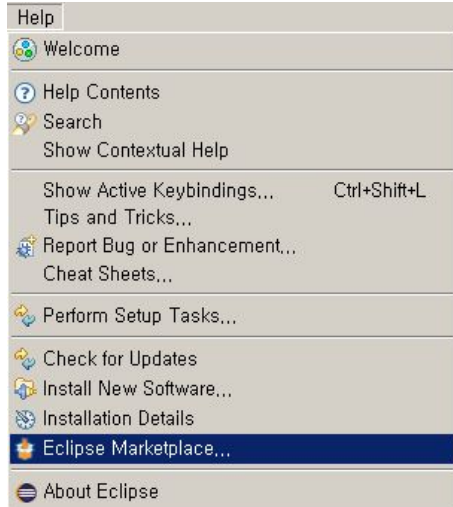
개발환경 설치

- GNU ARM Eclipse Plug-in 설치 (ZIP 파일을 이용한 설치)
 - <https://github.com/gnuarmeclipse/plugin/releases> 에서 최신 zip 파일을 다운로드
 - Help->Install New Software->Add 선택 후 Archive에 다운받은 zip파일을 선택하고 Name에 GNU ARM Eclipse Plug-in



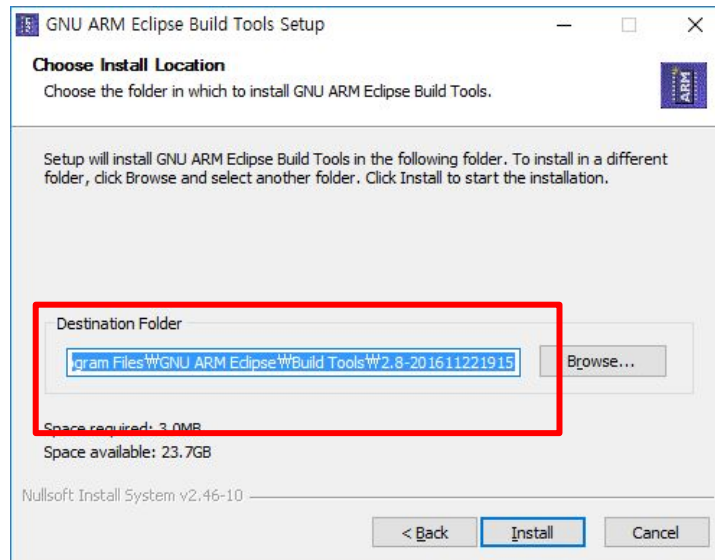
개발환경 설치

- GNU ARM Eclipse Plug-in 설치 (Marketplace 이용한 설치)
 - Help->Eclipse Marketplace 선택
 - Find->gnu arm 입력 후 검색
 - GNU ARM Eclipse 설치



개발환경 설치

- Windows Build Tools 설치
 - GCC로 빌드하기 위해서는 `make`와 `rm` 실행파일이 필요한데 윈도우에서는 기본 내장되어 있지 않기 때문에 설치 필요
 - <https://github.com/gnuarmclipse/windows-build-tools/releases> 에서 윈도우 버전에 맞게 설치
 - 설치 폴더는 향후에 Eclipse에서 설정시 필요함으로 어느 폴더 인지 확인 필요함



개발환경 설치

- GNU ARM GCC 설치
 - <https://launchpad.net/gcc-arm-embedded/5.0/5-2016-q2-update> 각 플랫폼에 맞는 버전의 zip 파일로 다운로드
 - Zip 파일로 다운로드 하고 원하는 폴더에 압축을 푼다.

 gcc-arm-none-eabi-5_4-2016q2-20160622-win32.exe (md5)

 gcc-arm-none-eabi-5_4-2016q2-20160622-win32.zip (md5)

개발환경 설치

- OpenOCD 설치
 - <https://github.com/gnuarmclipse/openocd/releases> 에서 윈도우 버전에 맞는 실행파일을 다운로드 후 설치

 [gnuarmclipse-openocd-win32-0.10.0-201701241841-setup.exe](#)

 [gnuarmclipse-openocd-win32-0.10.0-201701241841-setup.md5](#)

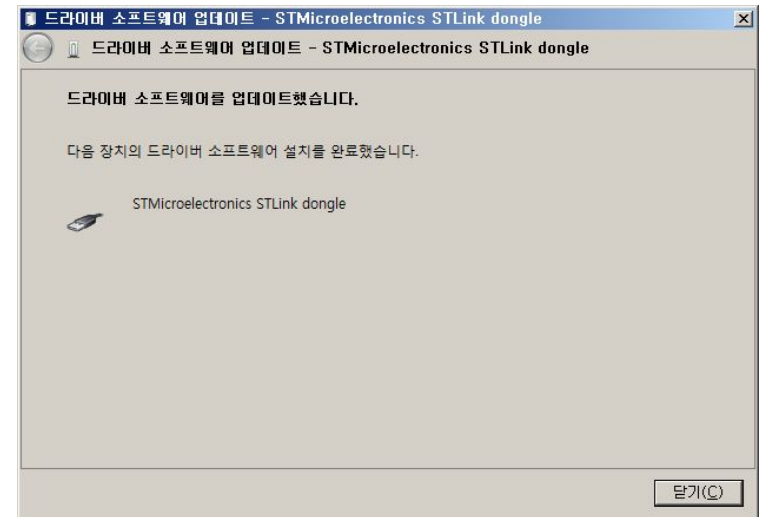
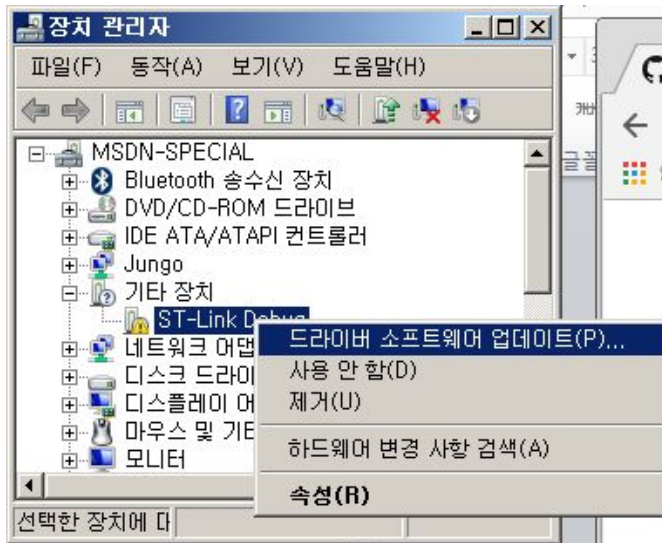
 [gnuarmclipse-openocd-win64-0.10.0-201701241841-setup.exe](#)

 [gnuarmclipse-openocd-win64-0.10.0-201701241841-setup.md5](#)

개발환경 설치

- STLink-V2 드라이버 설치

- http://www.st.com/content/st_com/en/products/embedded-software/development-tool-software/stsw-link009.html 에서 드라이버를 다운로드
- STLink-V2를 PC에 연결하고 장치 관리자에 ST-Link Debug장치가 보이고 드라이버 소프트웨어 업데이트를 선택하면 자동으로 설치됨



개발환경 설치

- ST Virtual COM 드라이버 설치
 - USB를 가상의 시리얼포트로 인식하게 해주는 드라이버를 <http://www.st.com/en/development-tools/stsw-stm32102.html> 에서 다운로드 후 설치
 - 드라이버를 다운로드 하기위해서는 간단히 성명과 이메일주소가 필요

First Name:

Last Name:

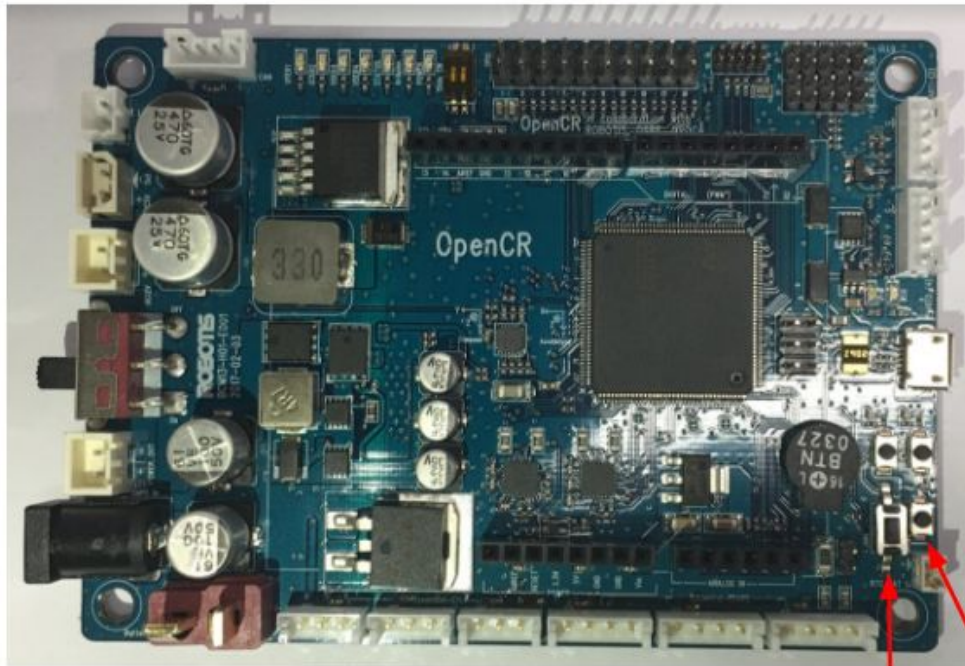
E-mail address:

☐ I would like to stay up to date with ST's latest products and subscribe to the ST newsletters.

Download

개발환경 설치

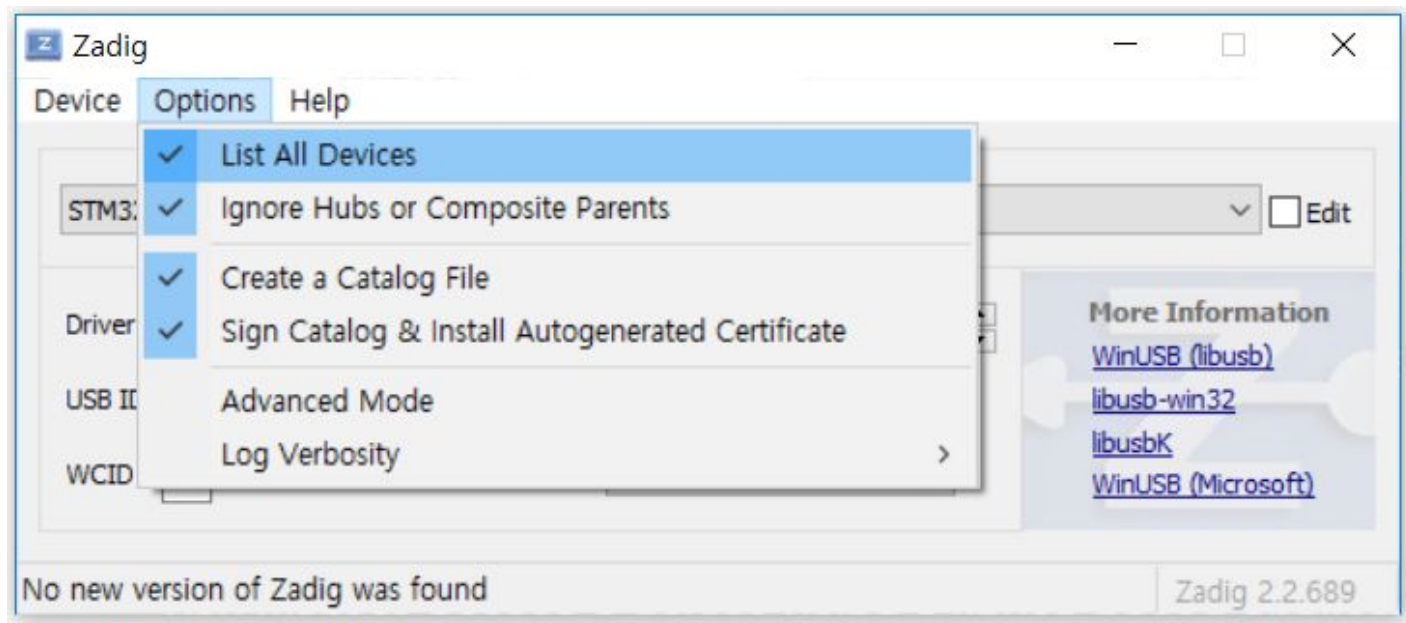
- DFU 모드 실행
 - OpenCR보드의 BOOT0핀을 누른 상태에서 RESET을 누르면 DFU모드로 실행됨



RESET BOOT0

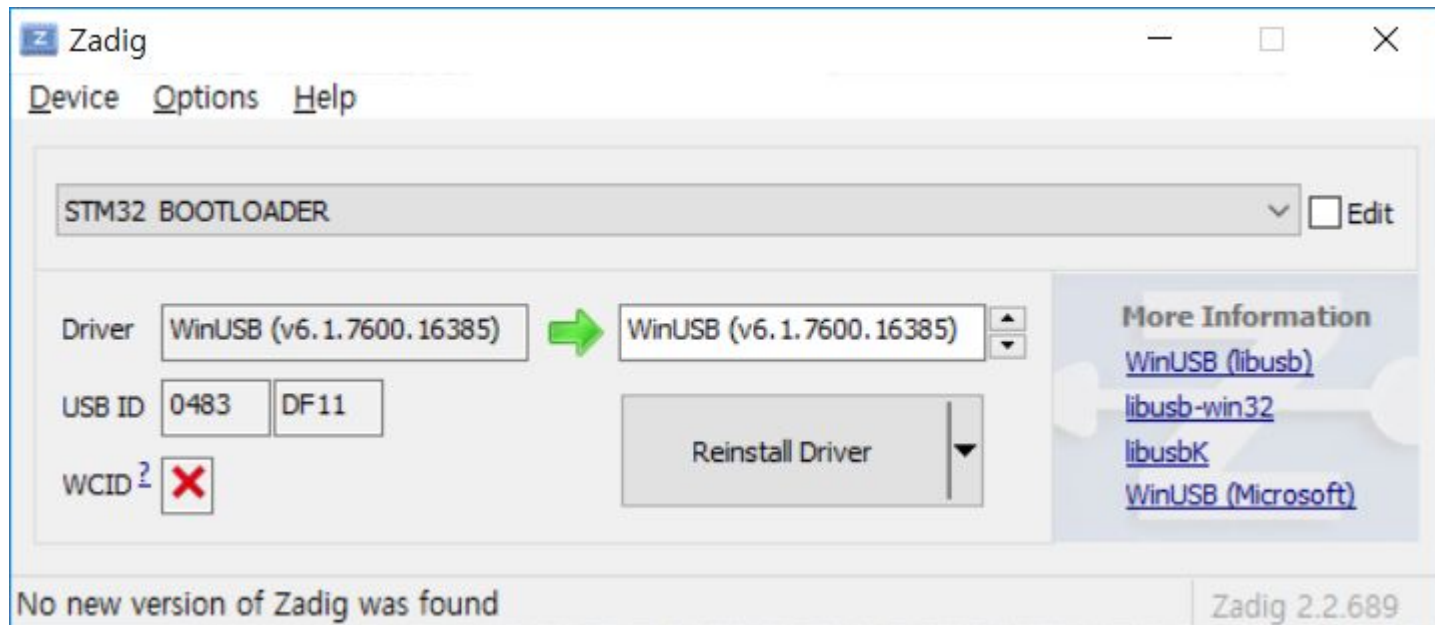
개발환경 설치

- DFU 드라이버 설치
 - DFU 유틸리티를 이용해서 다운로드 하기 위해서는 DFU 드라이버가 필요한데 ST에서 제공하는 드라이버가 아닌 WinUSB 드라이버를 설치
 - <http://zadig.akeo.ie/> 에서 zadig를 다운로드
 - Zadig실행 후 Options->List All Devices를 선택



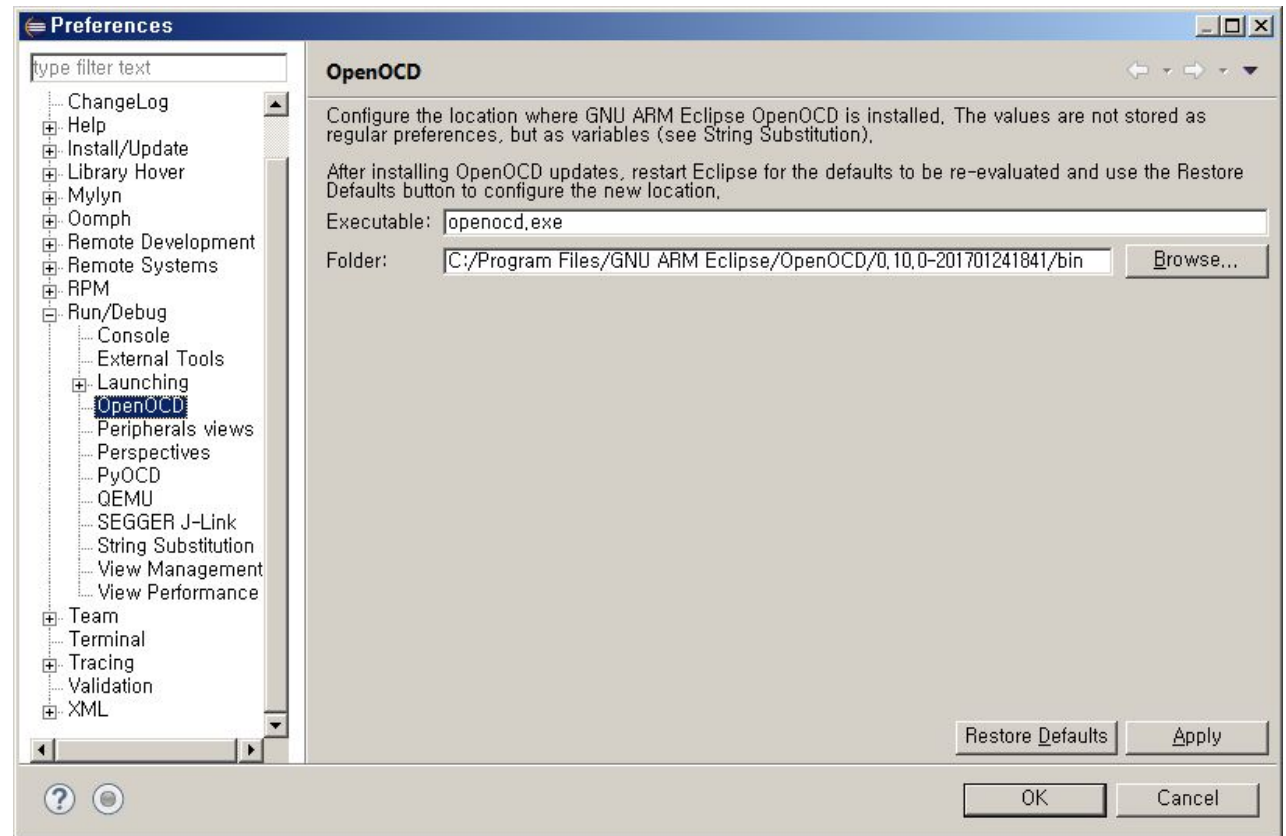
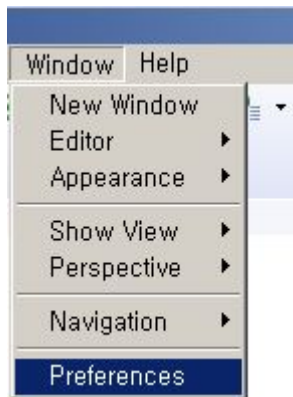
개발환경 설치

- DFU 드라이버 설치
 - STM32 BOOTLOADER를 선택하고 WinUSB 드라이버를 설치



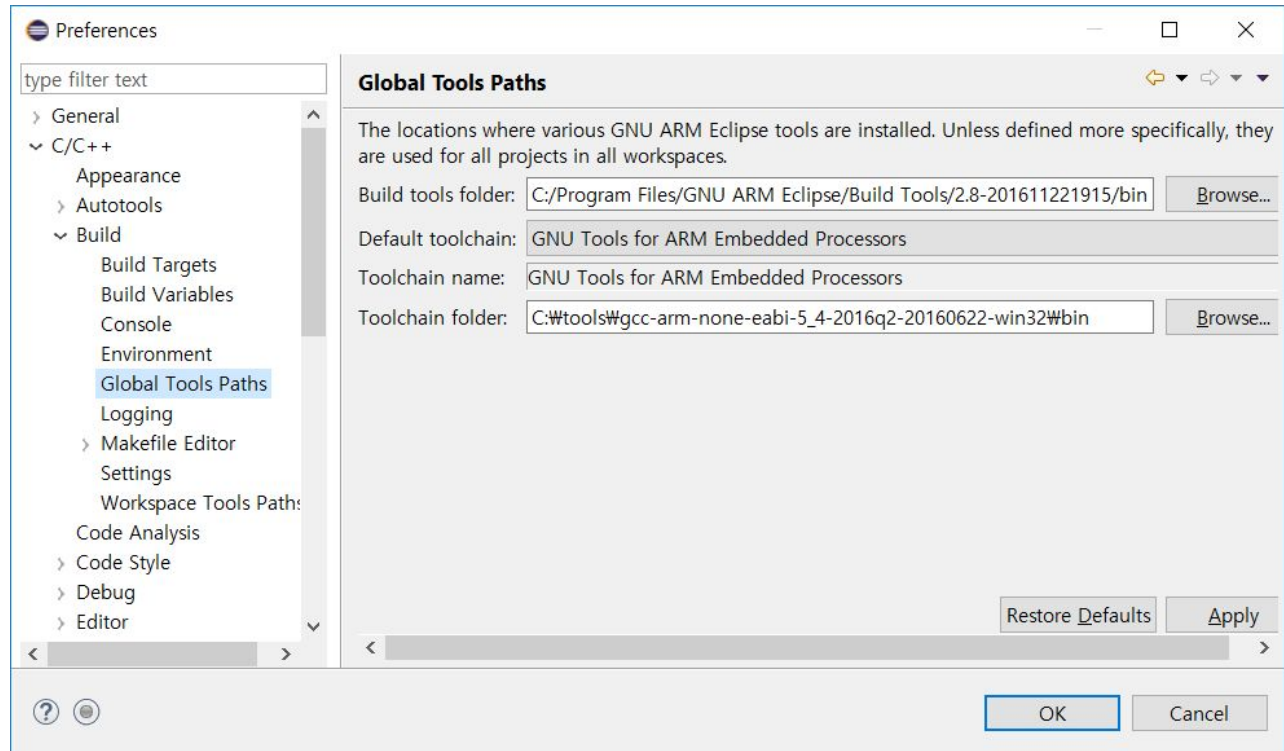
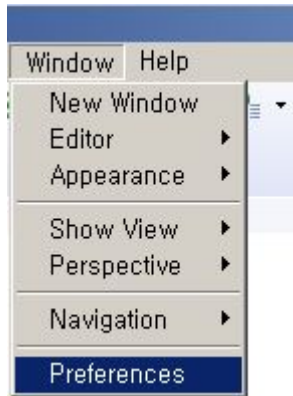
Eclipse 설정

- OpenOCD 패스 설정
 - Window->Preferences->Run/Debug->OpenOCD에 실행 파일과 폴더 설정



Eclipse 설정

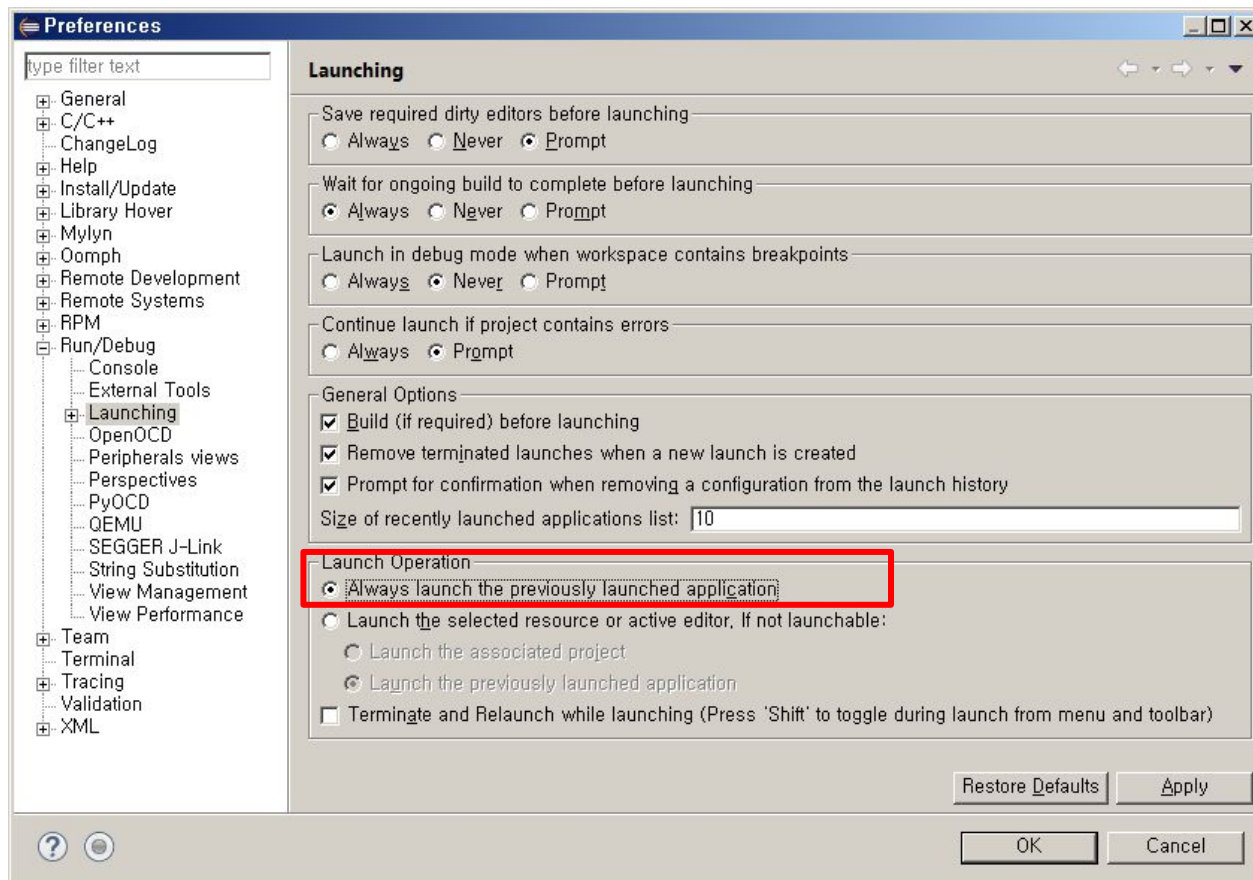
- Global Tools Paths 설정
 - Window->Preferences->C/C++->Global Tools Paths에 Build tools와 Toolchain 폴더를 지정한다.



Eclipse 설정

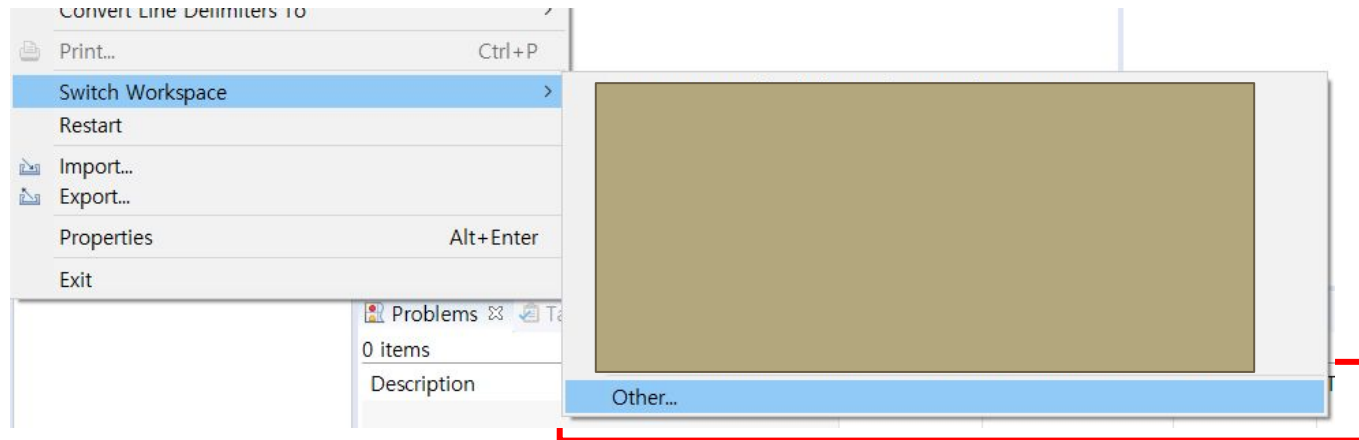
- Launching 설정

- 마지막 디버깅 실행 내용을 저장했다가 디버깅 실행시 저장된 디버깅 설정으로 실행됨



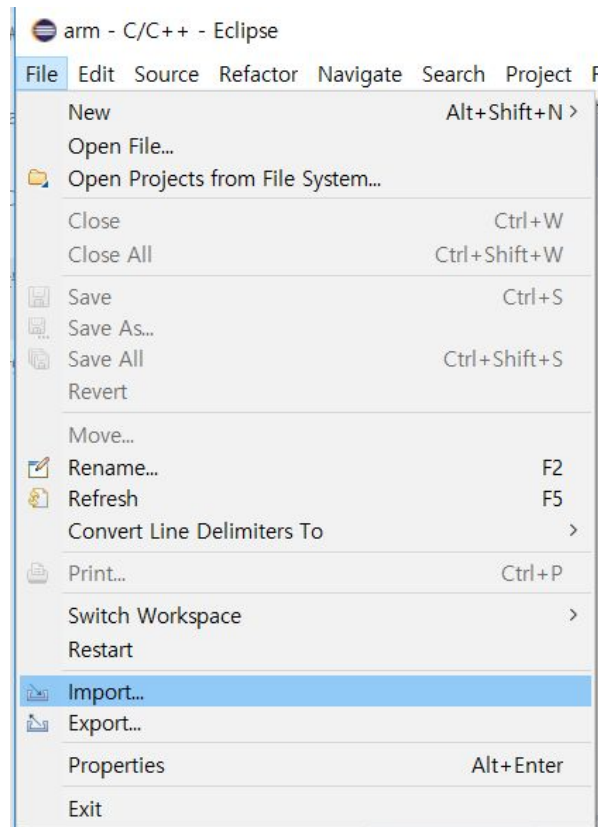
Workspace 변경

- 새로운 Workspace로 변경
 - 새로운 프로젝트를 사용하기 위해서 새로운 Workspace를 생성
 - File->Switch Workspace->Other를 선택하여 새로운 Workspace이름을 지정함



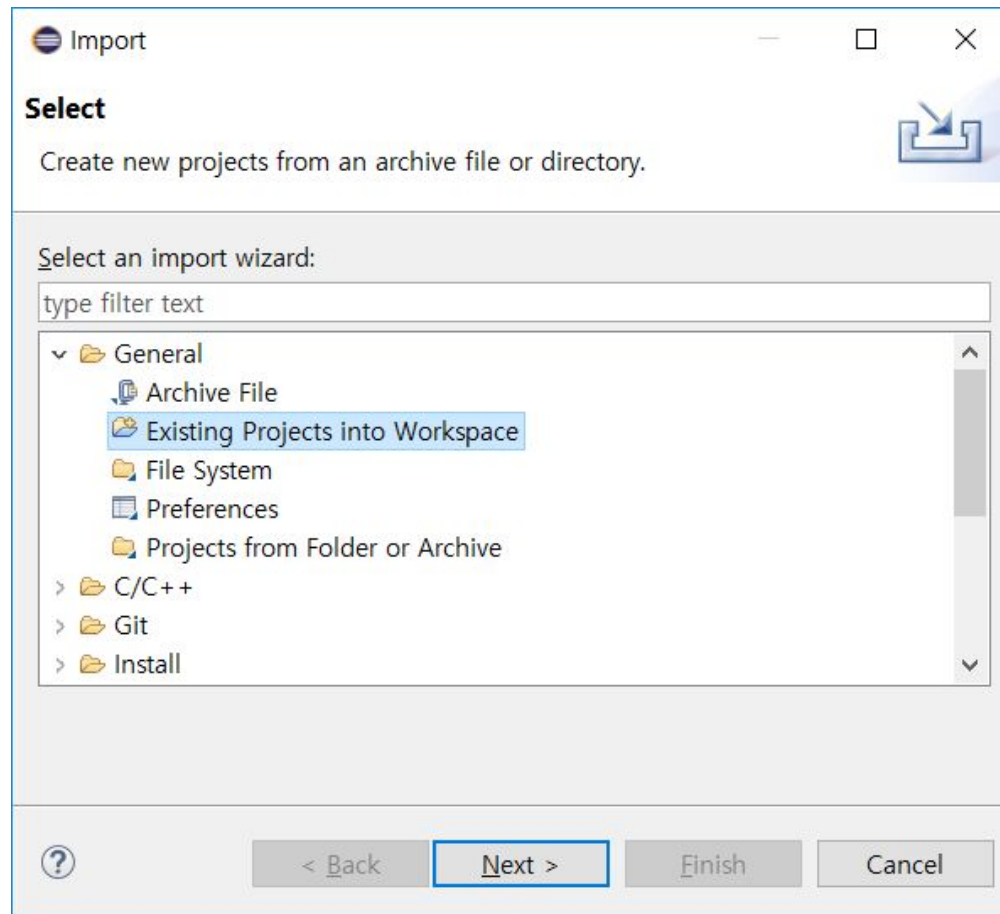
프로젝트 Import

- File->Import

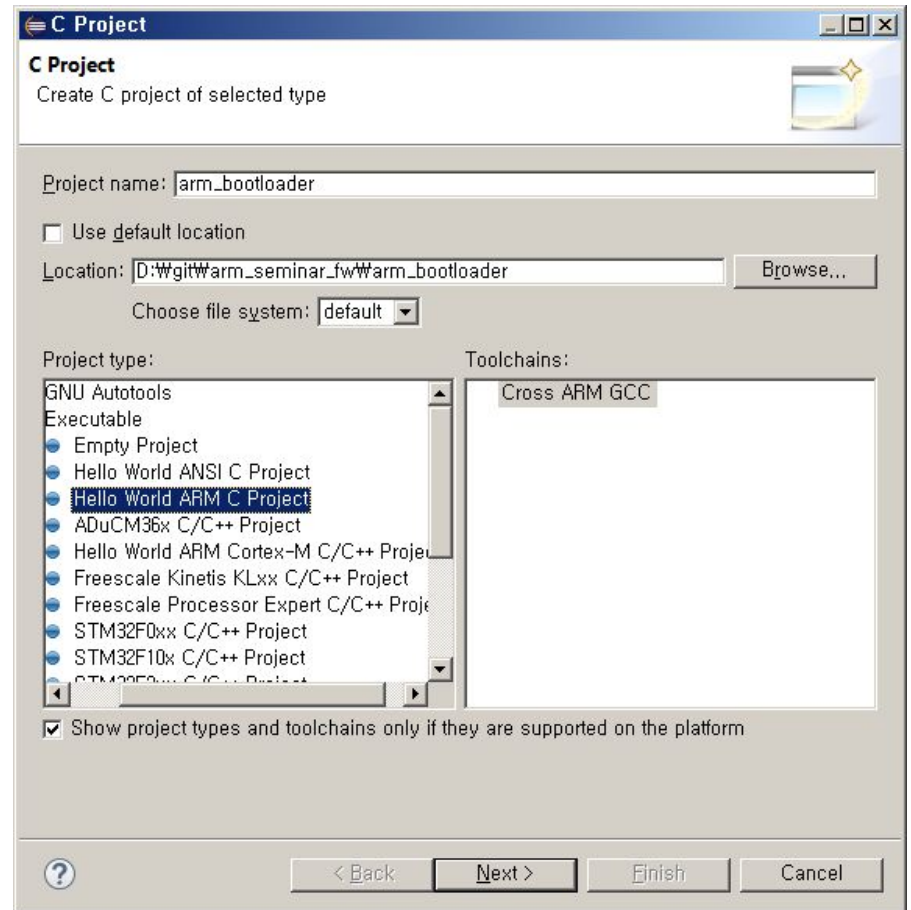
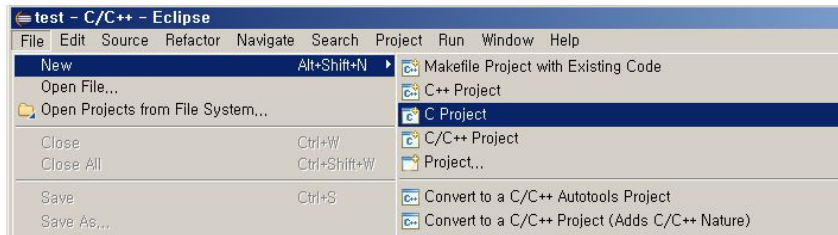


프로젝트 Import

- General -> Existing Projects into Wrokspace

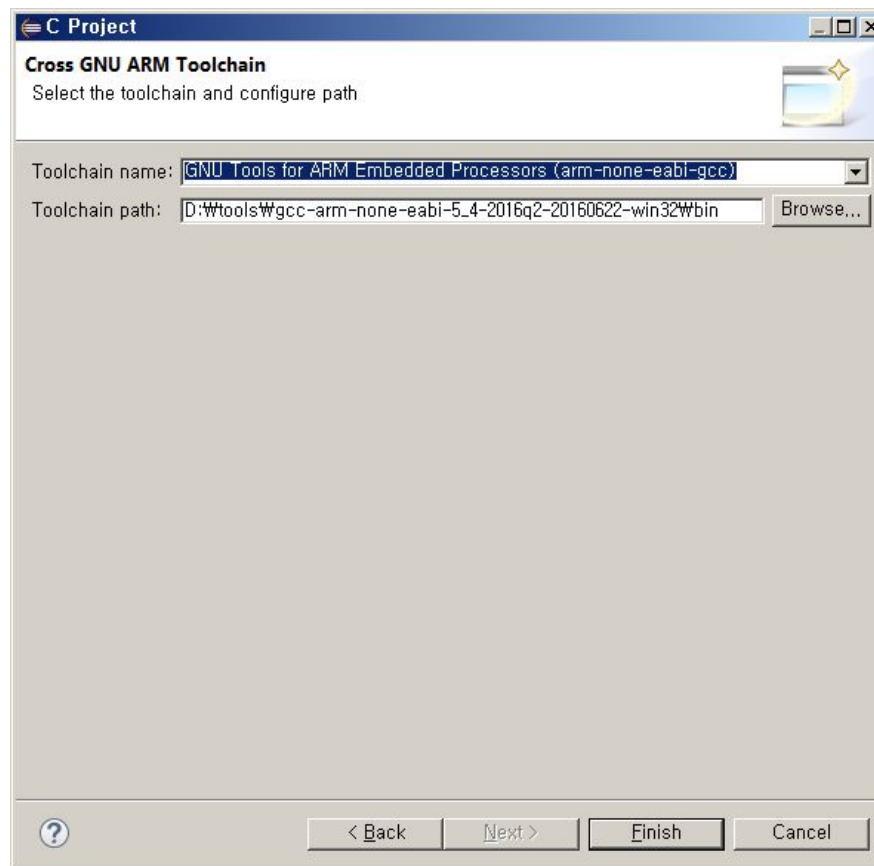


프로젝트 생성

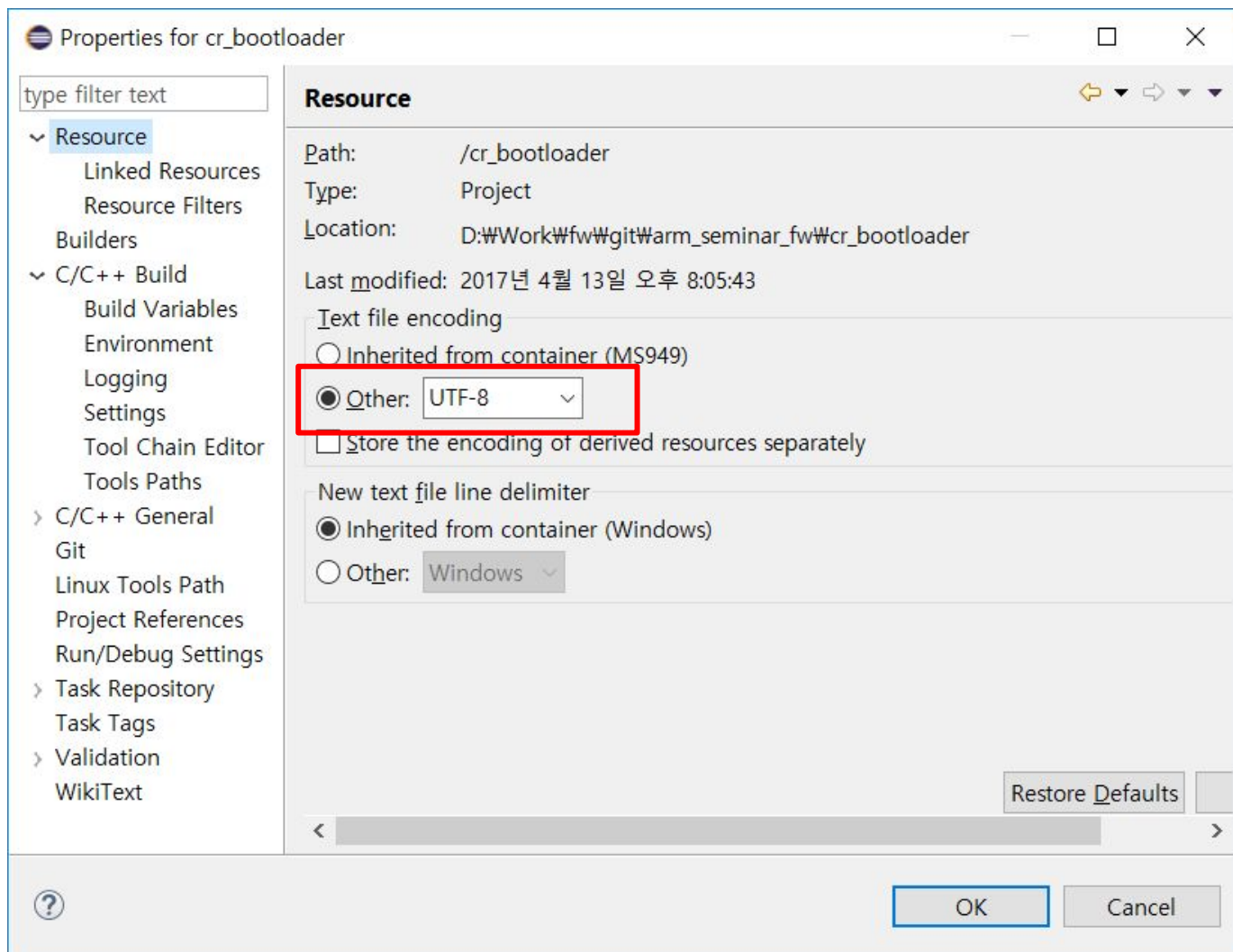


프로젝트 생성

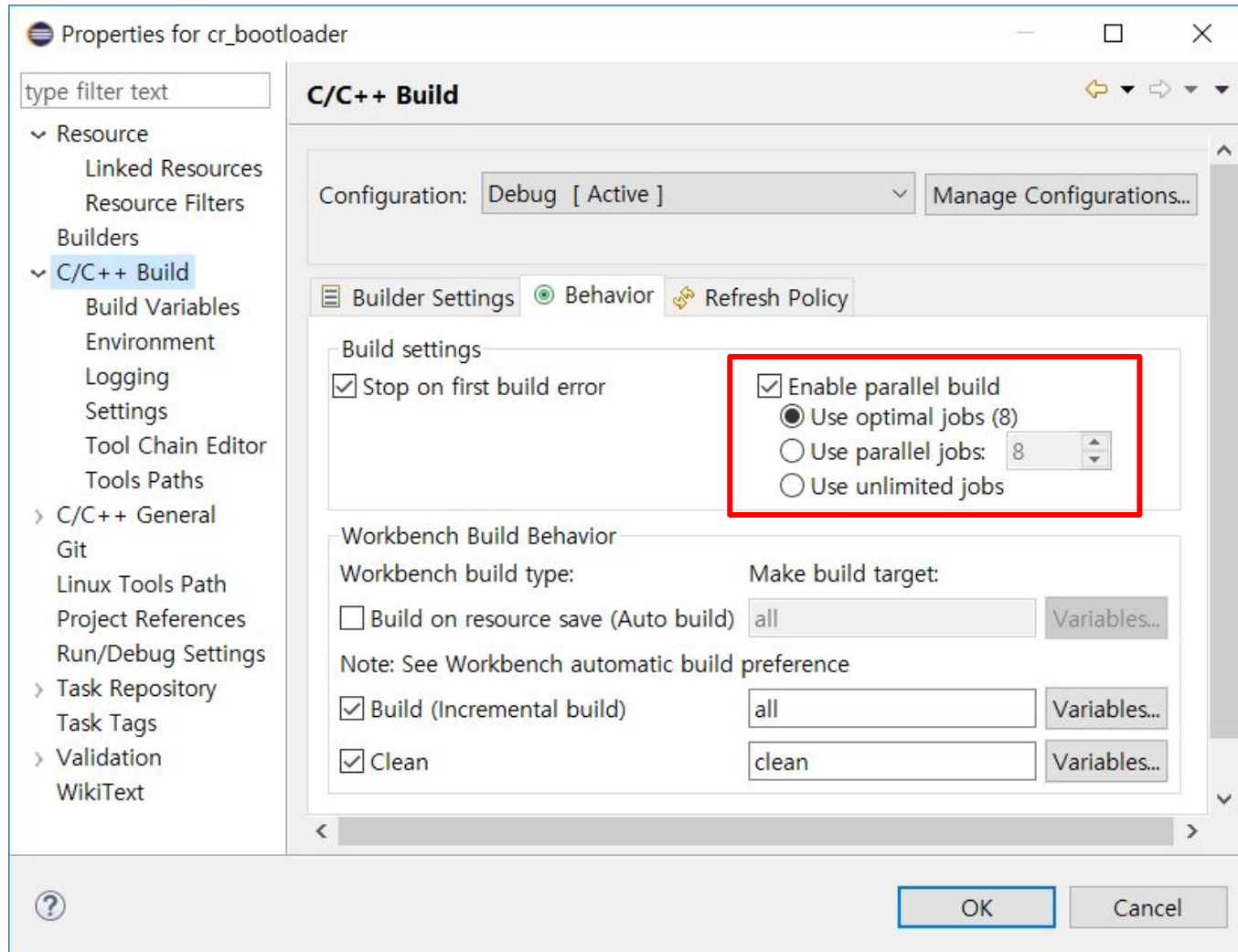
- Toolchain 패스 지정



프로젝트 설정

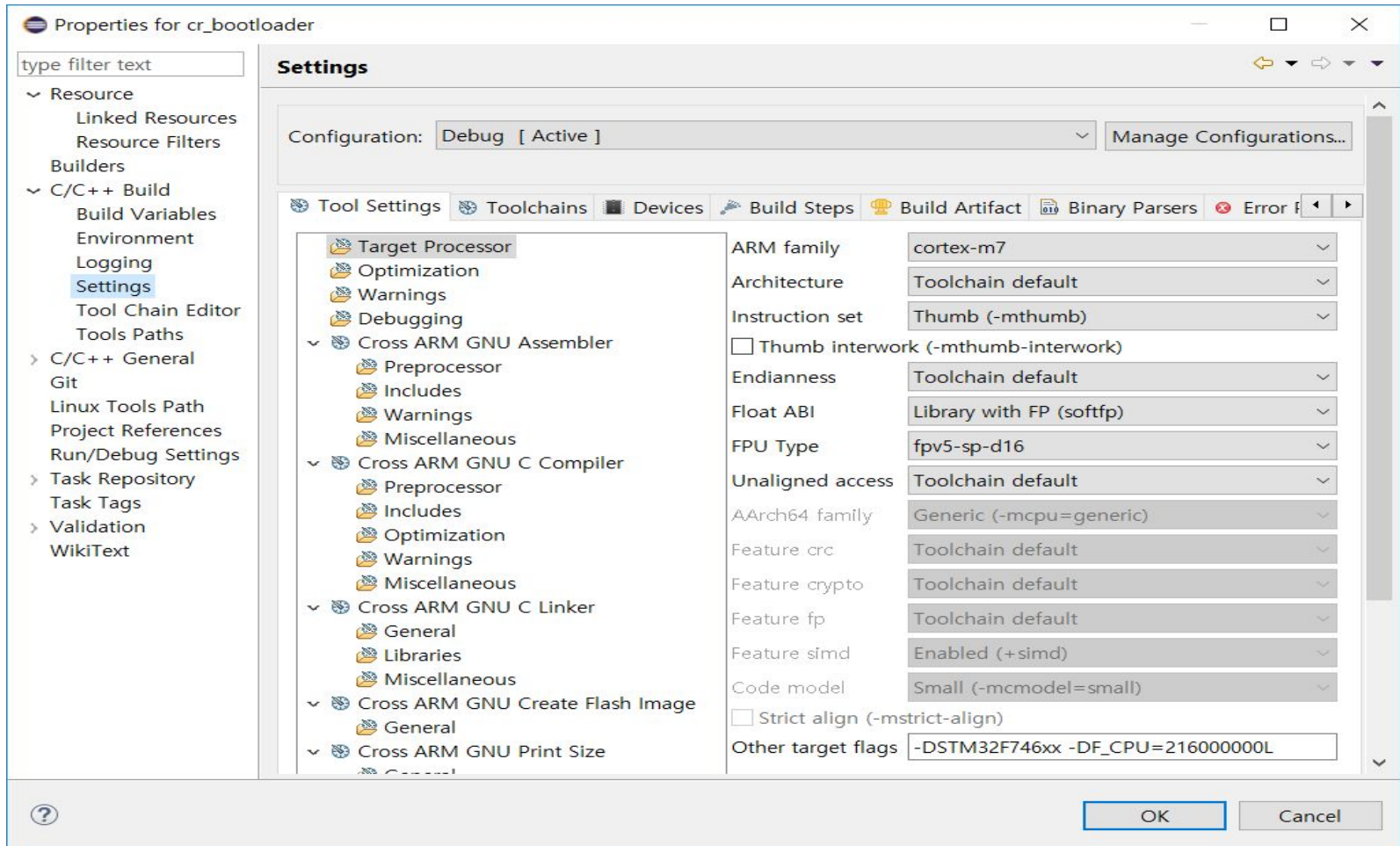


프로젝트 설정

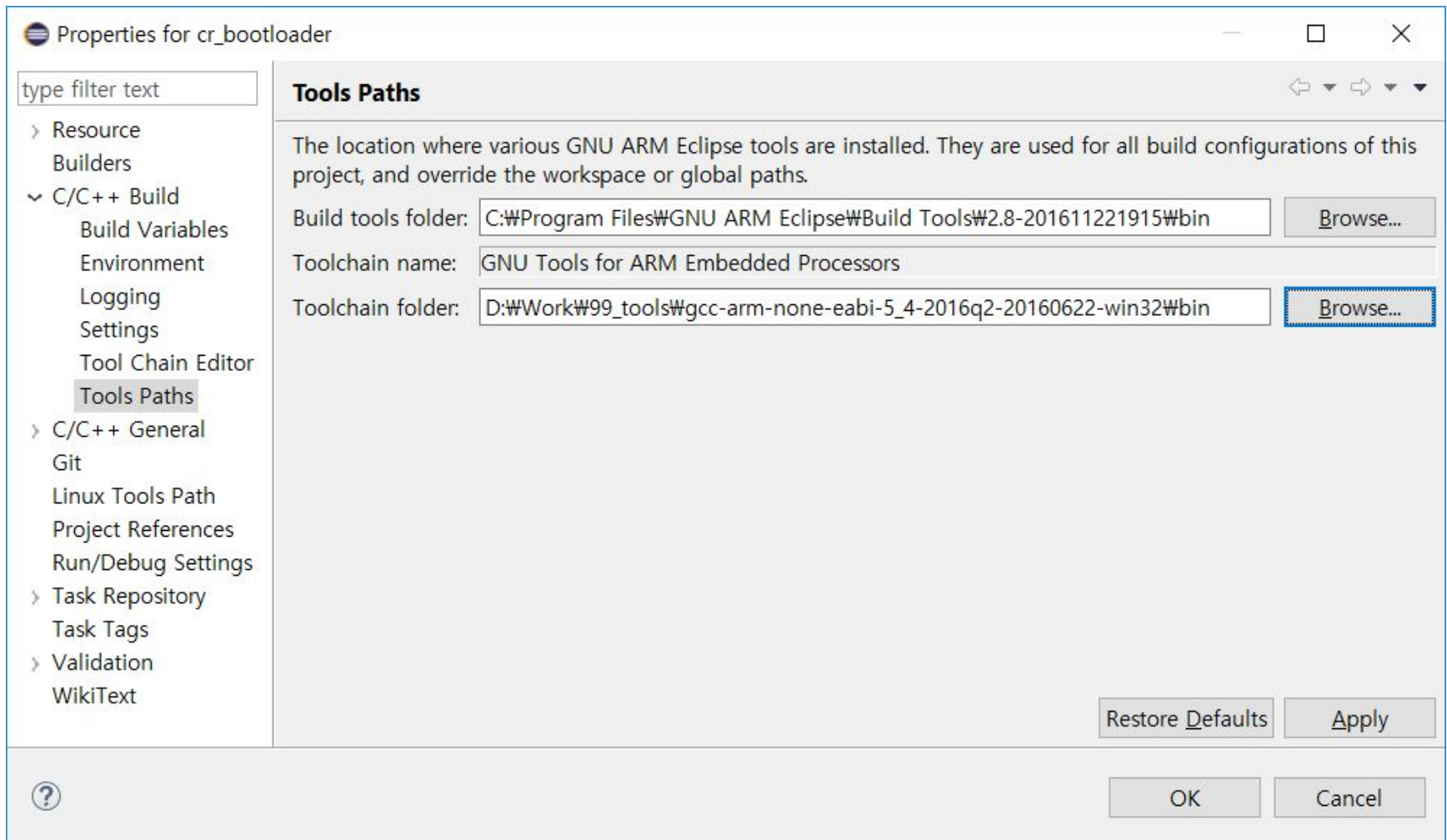


프로젝트 설정

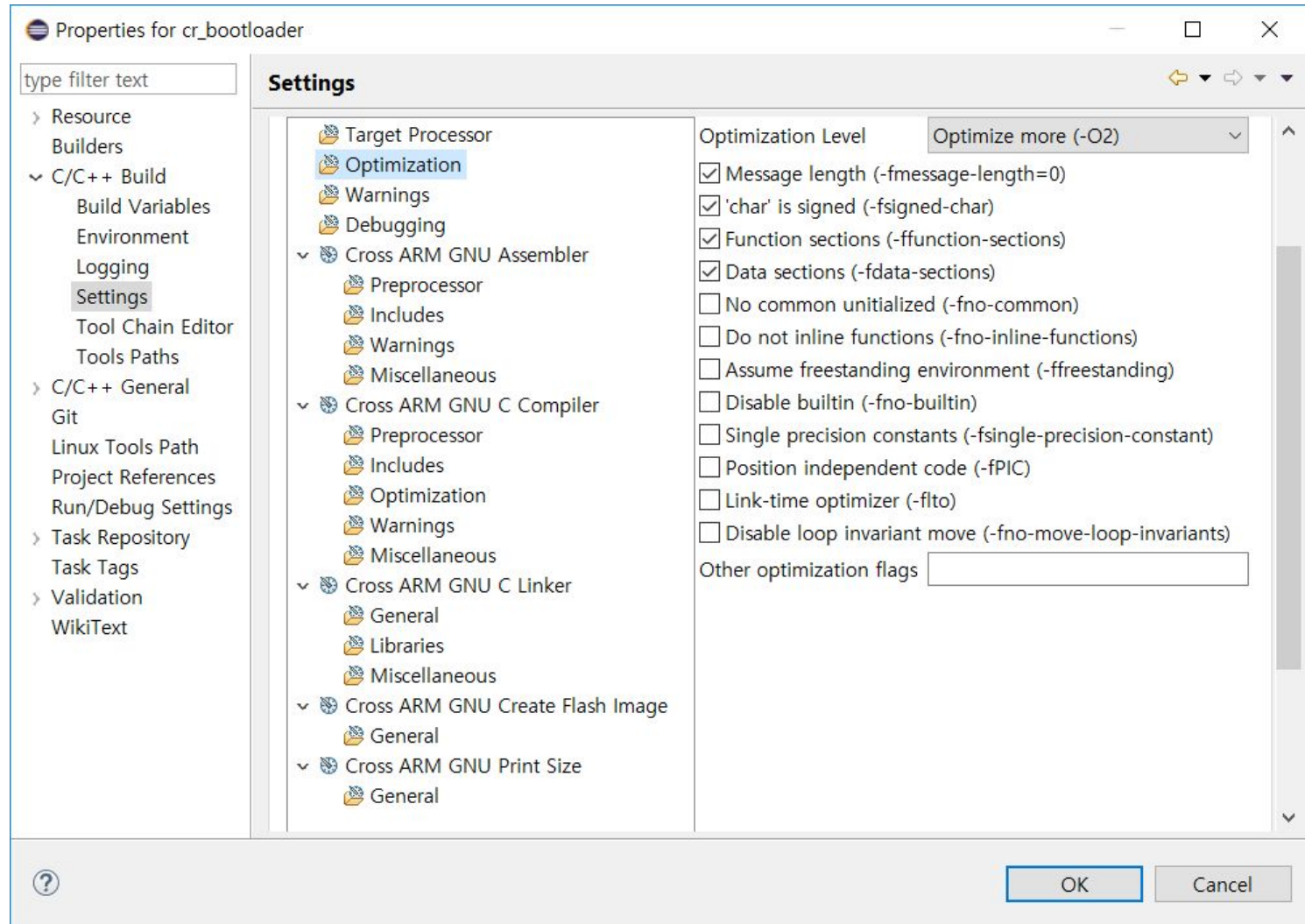
-DSTM32F746xx -DF_CPU=216000000L



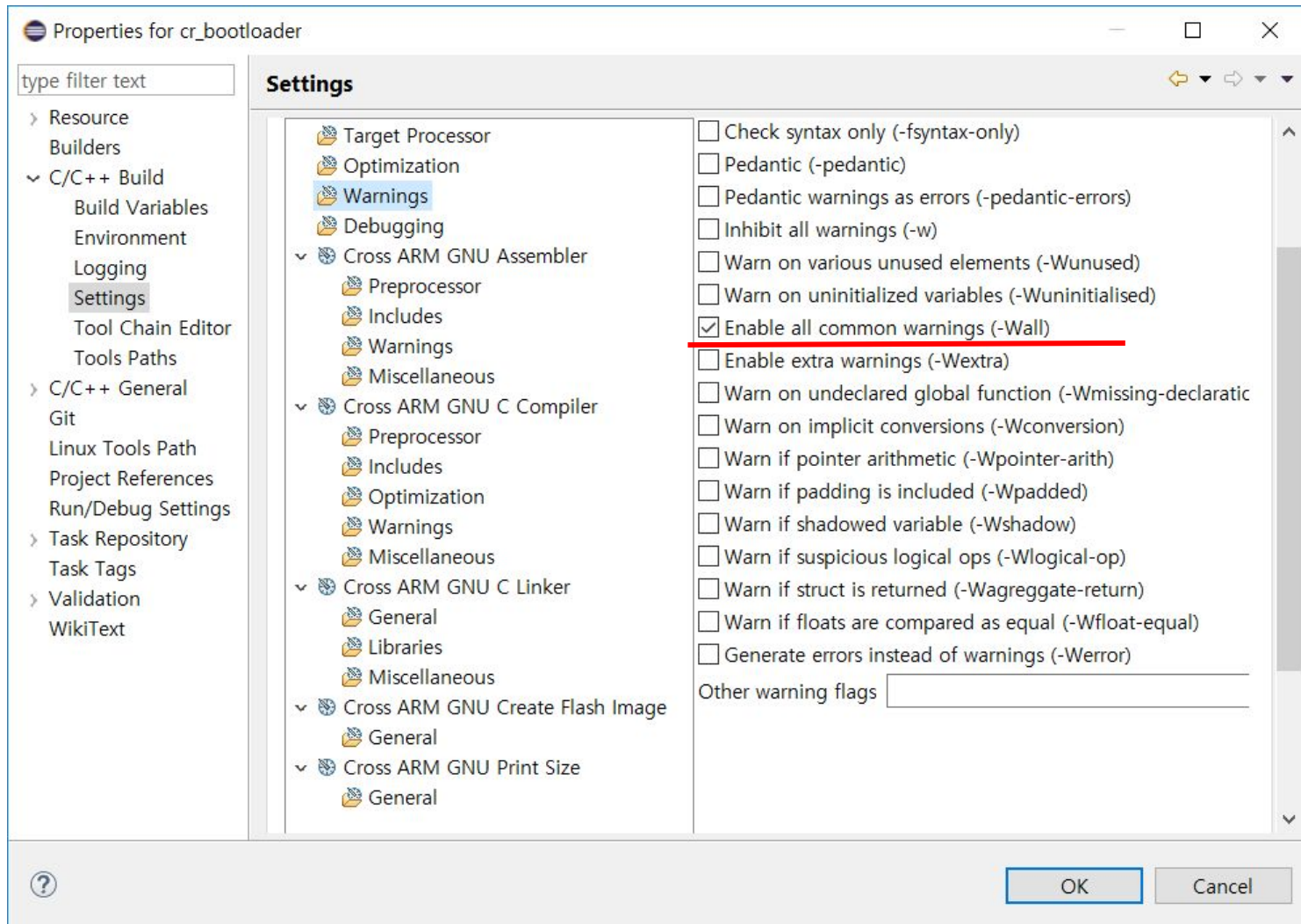
프로젝트 설정



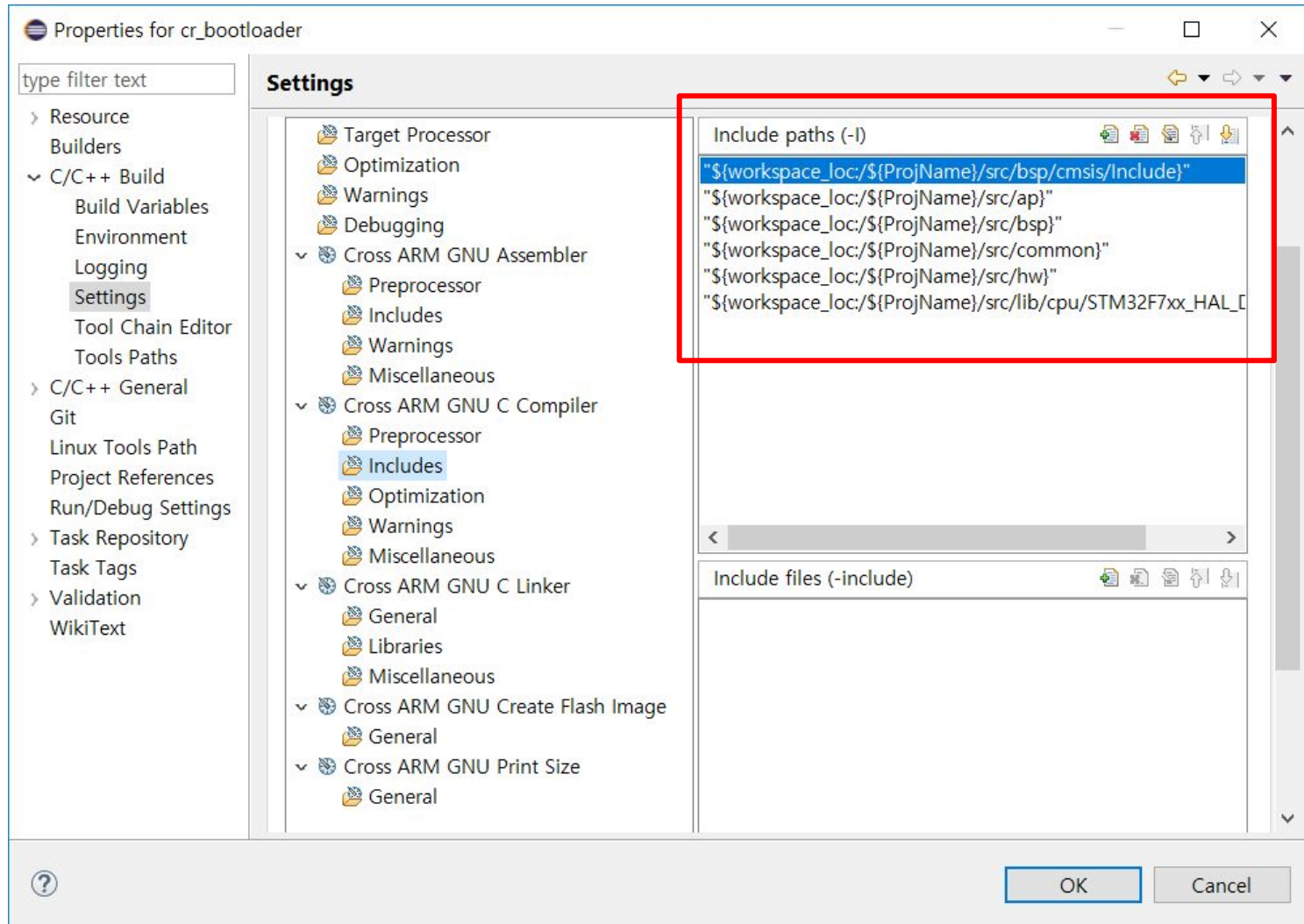
프로젝트 설정



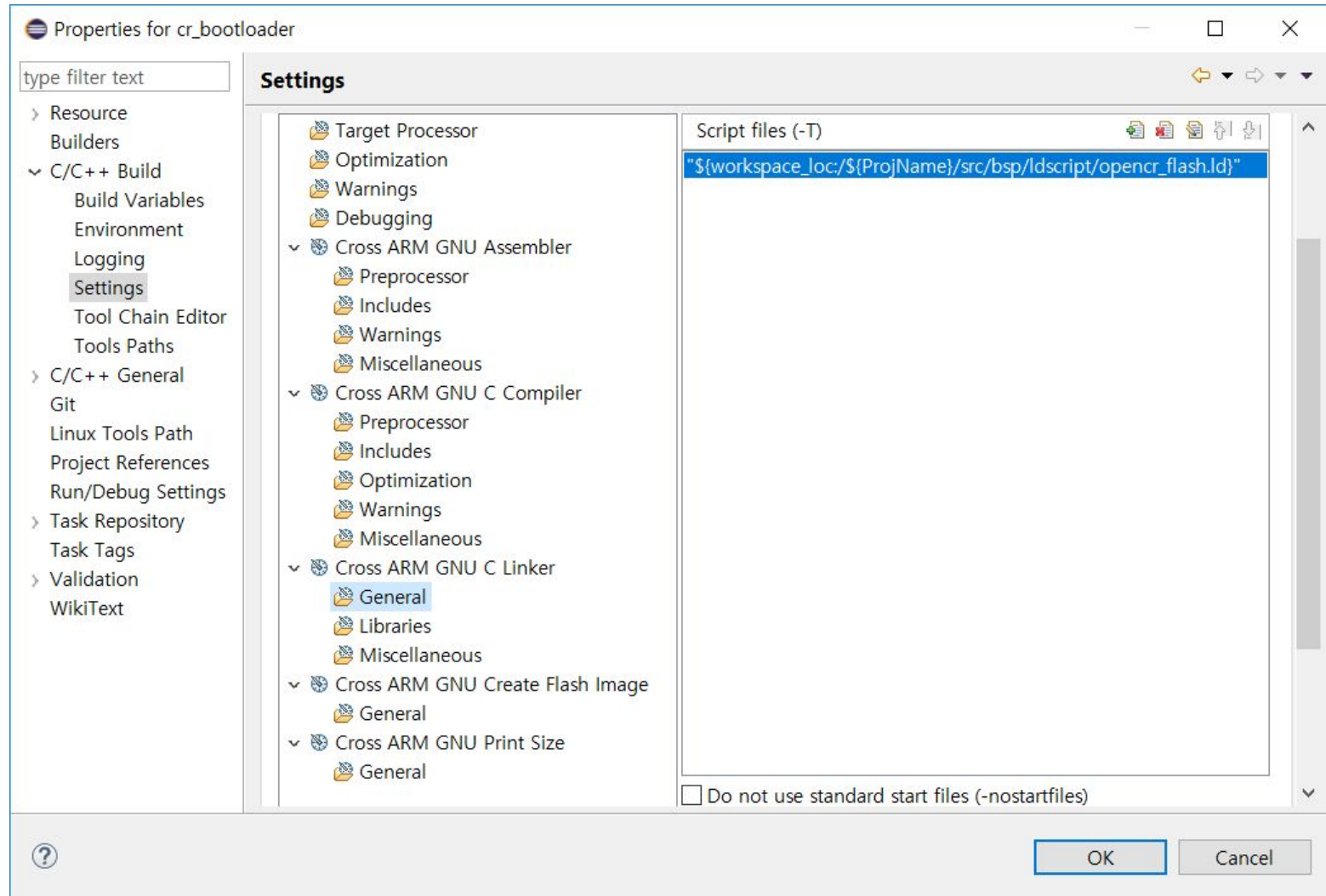
프로젝트 설정



프로젝트 설정



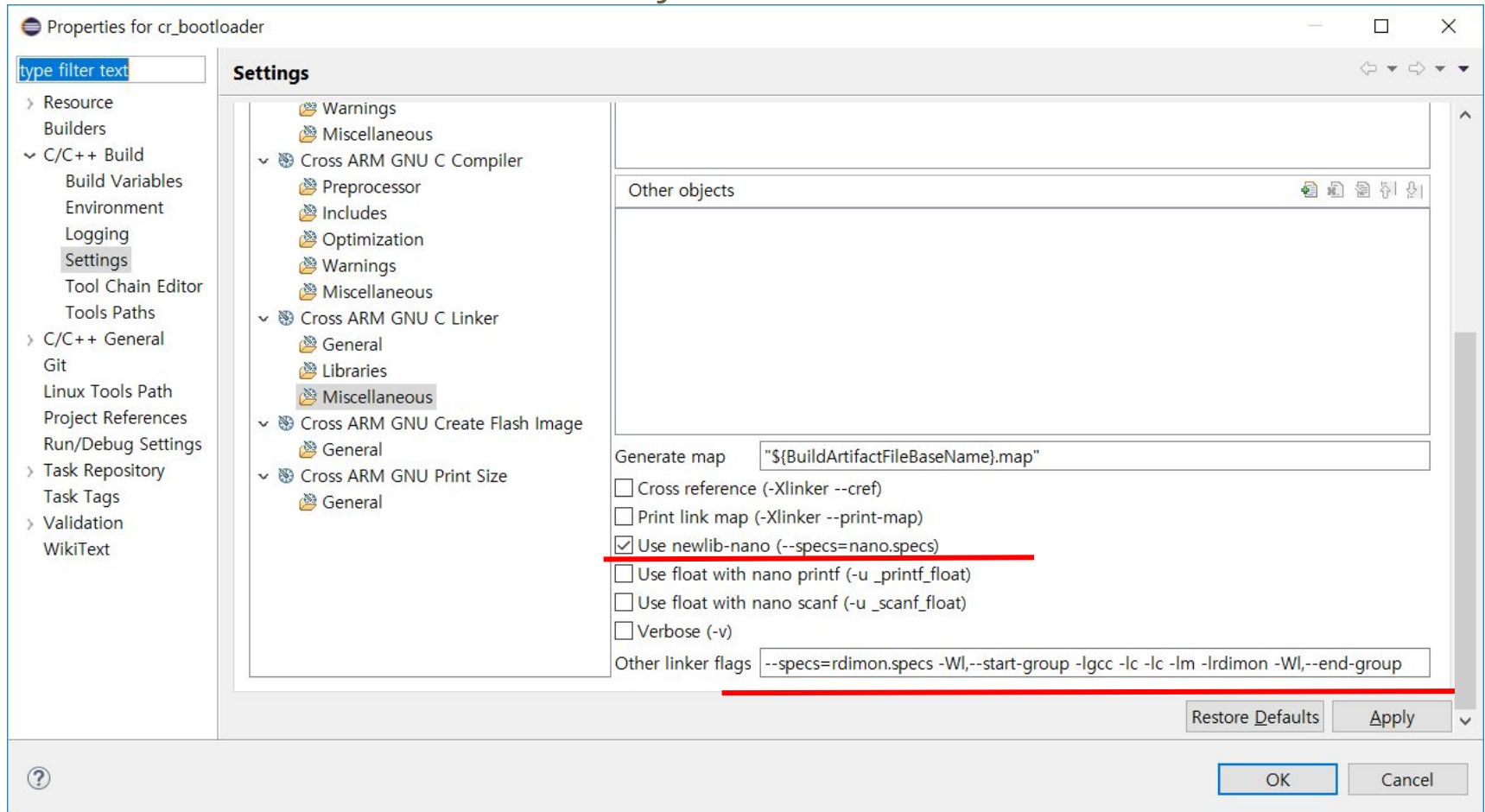
프로젝트 설정



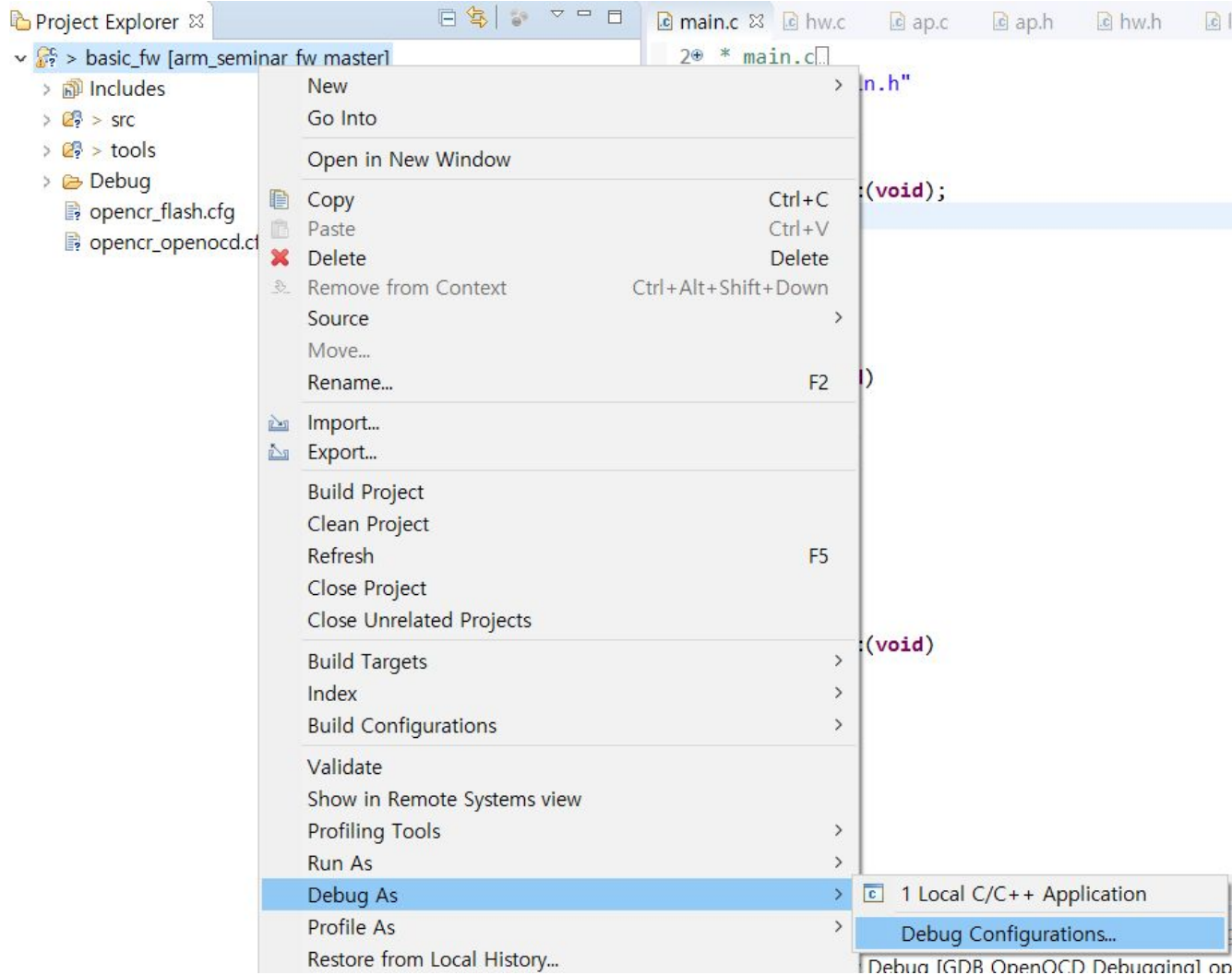
프로젝트 설정

--specs=rdimon.specs -Wl,--start-group -lgcc -lc -lc -lm -lrdimon -Wl,--end-group

> 위의 옵션을 사용하지 않는 경우 syscalls.c 를 추가한다.

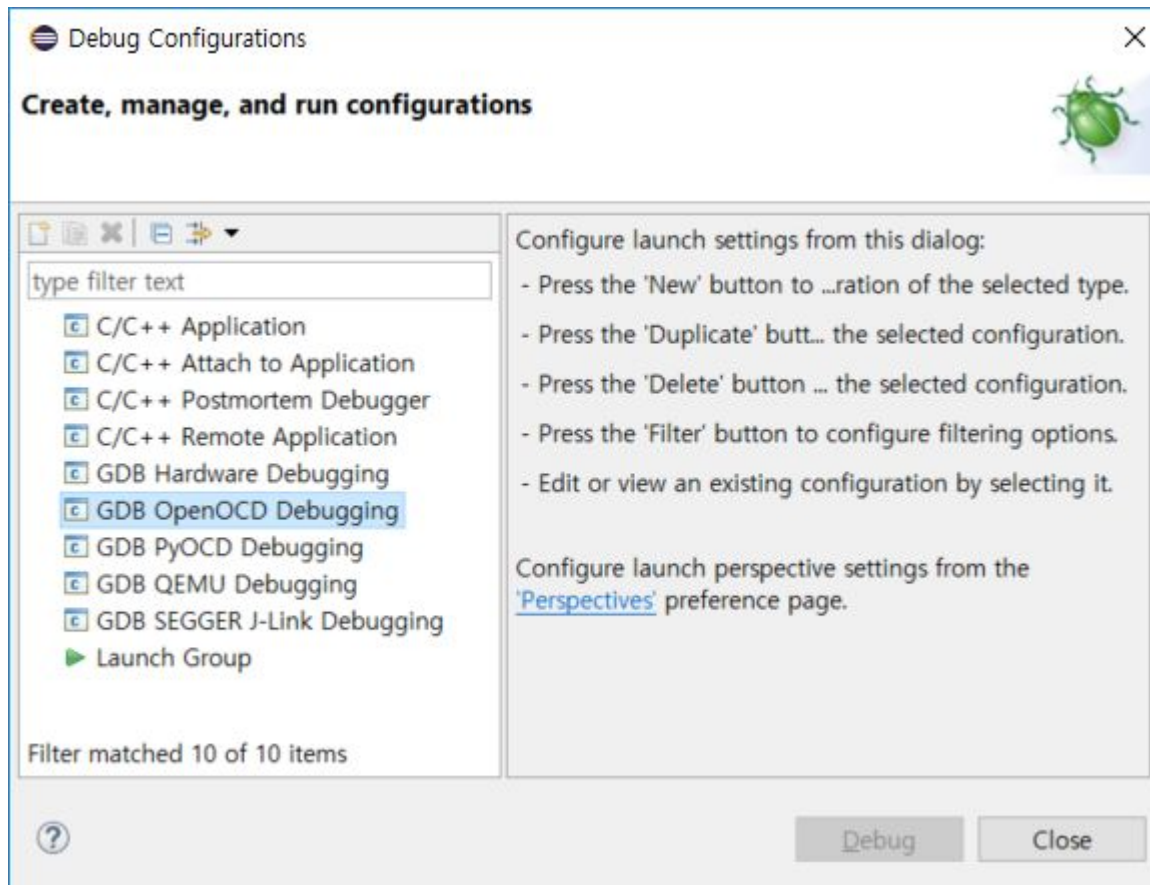


디버그 설정 - OpenOCD



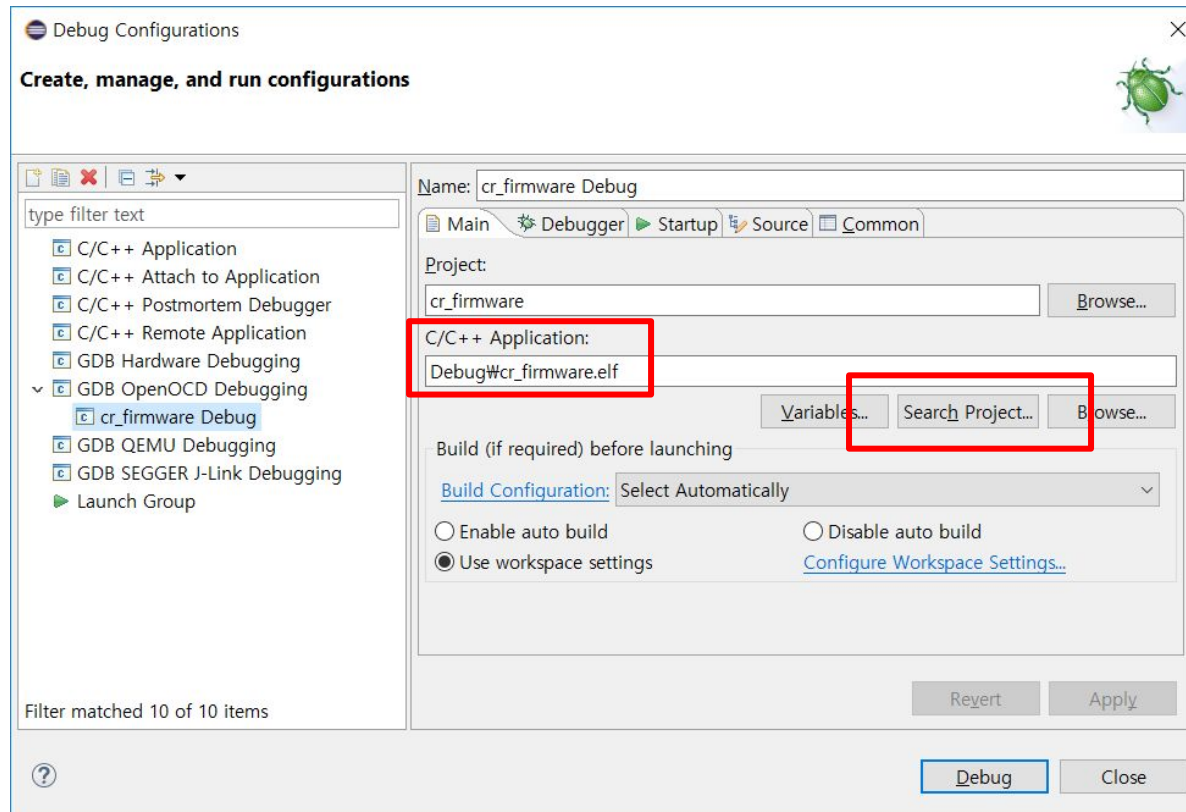
디버깅 설정 - OpenOCD

- GDB OpenOCD Debugging 에서 더블 클릭



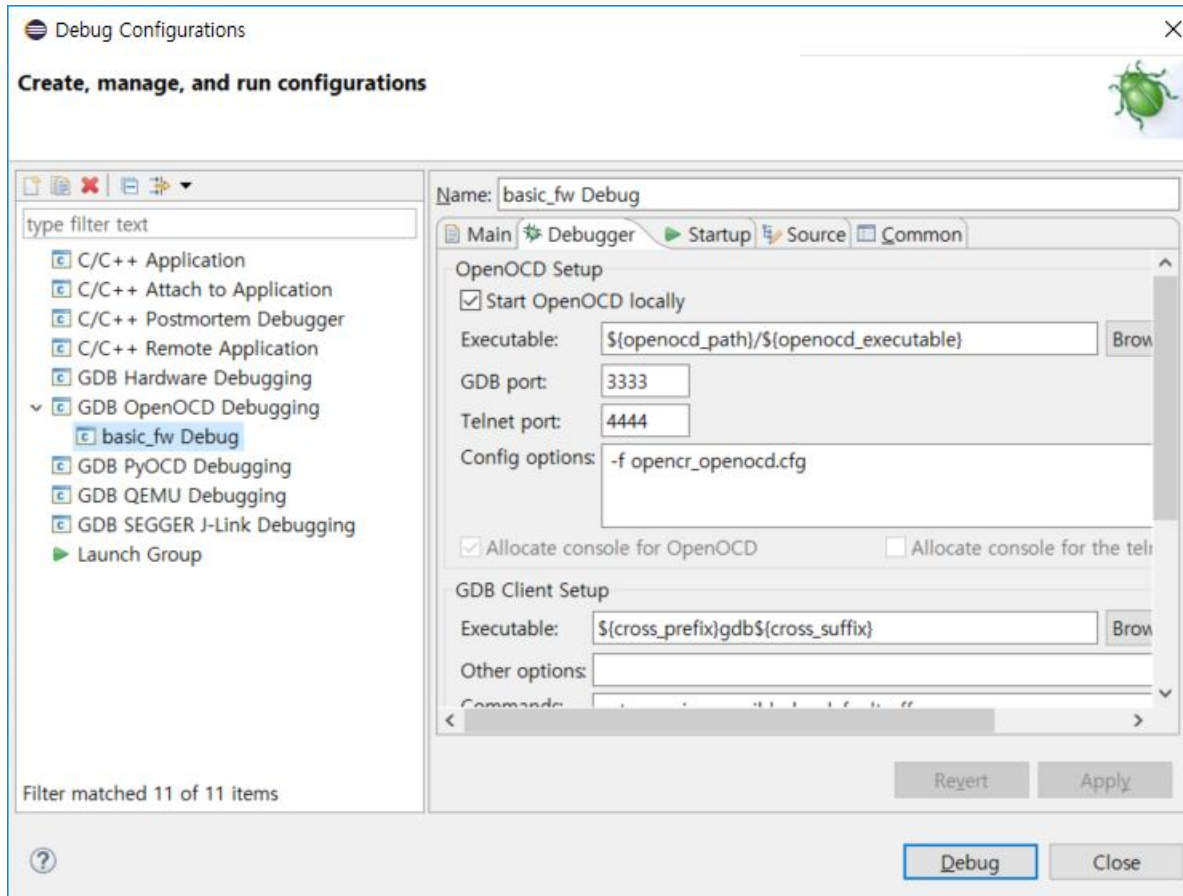
디버깅 설정 - OpenOCD

- Main->C/C++ Application이 빈 공백인 경우 Search Project로 elf파일을 선택함



디버깅 설정 - OpenOCD

-f opencr_openocd.cfg



디버깅 설정 - OpenOCD

- opencr_openocd.cfg의 내용을 옵션에 직접 입력 가능

```
interface hla
hla_layout stlink
hla_device_desc "ST-LINK/V2"
hla_vid_pid 0x0483 0x3748

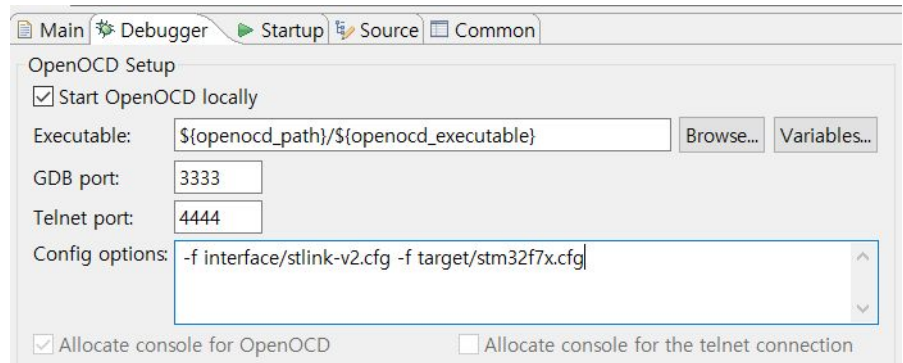
transport select hla_swd

# increase working area to 256KB
set WORKAREASIZE 0x40000

adapter_nsrst_delay 100

source [find target/stm32f7x.cfg]

reset_config none
```



-f interface/stlink-v2.cfg -f target/stm32f7x.cfg

디버깅 설정 - OpenOCD

- OpenOCD 추가 명령어 입력
 - Config options에 -c 를 이용하여 추가하고자 하는 명령어를 입력



The image shows a screenshot of the 'OpenOCD Setup' dialog box. It contains several fields and checkboxes. The 'Config options' field is highlighted with a red rectangle, showing the command: `-f interface/stlink-v2.cfg -f target/stm32f1x.cfg -c "init"`. Other fields include 'Executable' with a path, 'GDB port' set to 3333, and 'Telnet port' set to 4444. Checkboxes for 'Start OpenOCD locally', 'Allocate console for OpenOCD', and 'Allocate console for the telnet connection' are also visible.

OpenOCD Setup

☒ Start OpenOCD locally

Executable:

GDB port:

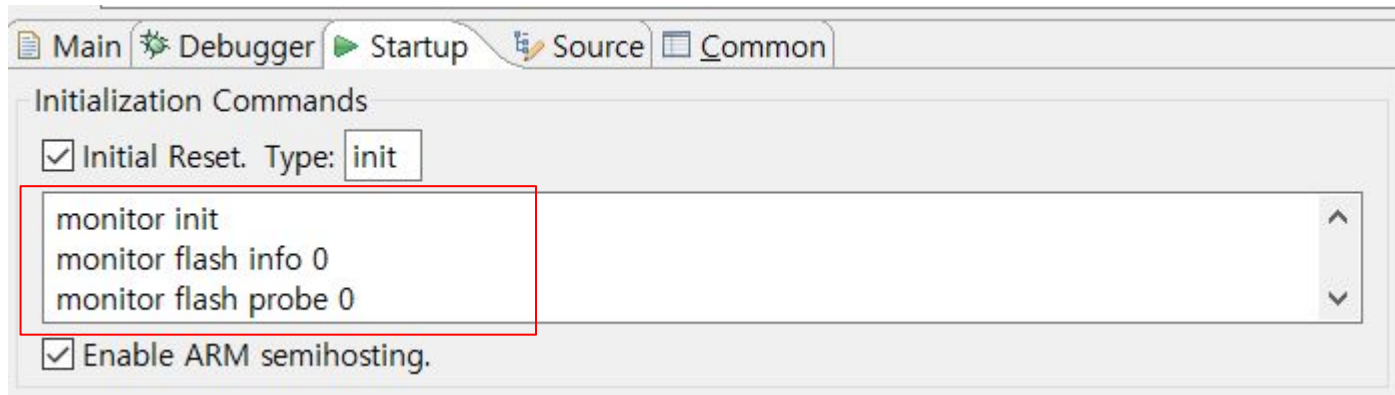
Telnet port:

Config options:

☒ Allocate console for OpenOCD ☐ Allocate console for the telnet connection

디버깅 설정 - OpenOCD

- OpenOCD 추가 명령어 입력
 - Startup의 초기화 명령어에 monitor 명령 이후에 필요한 OpenOCD 명령어를 입력

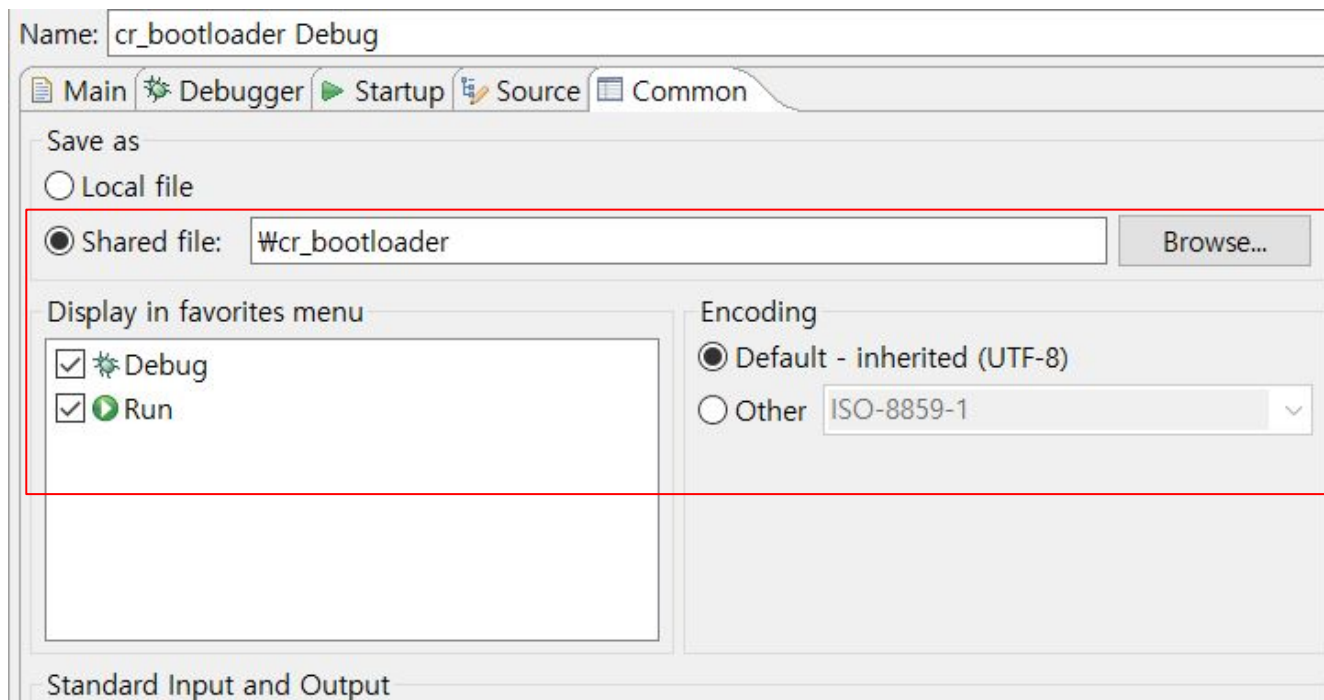


- Flash의 메모리 보호를 해제하는 명령의 예제

```
monitor init
monitor flash info 0
monitor flash probe 0
monitor flash protect 0 0 127 off
monitor flash info 0
```

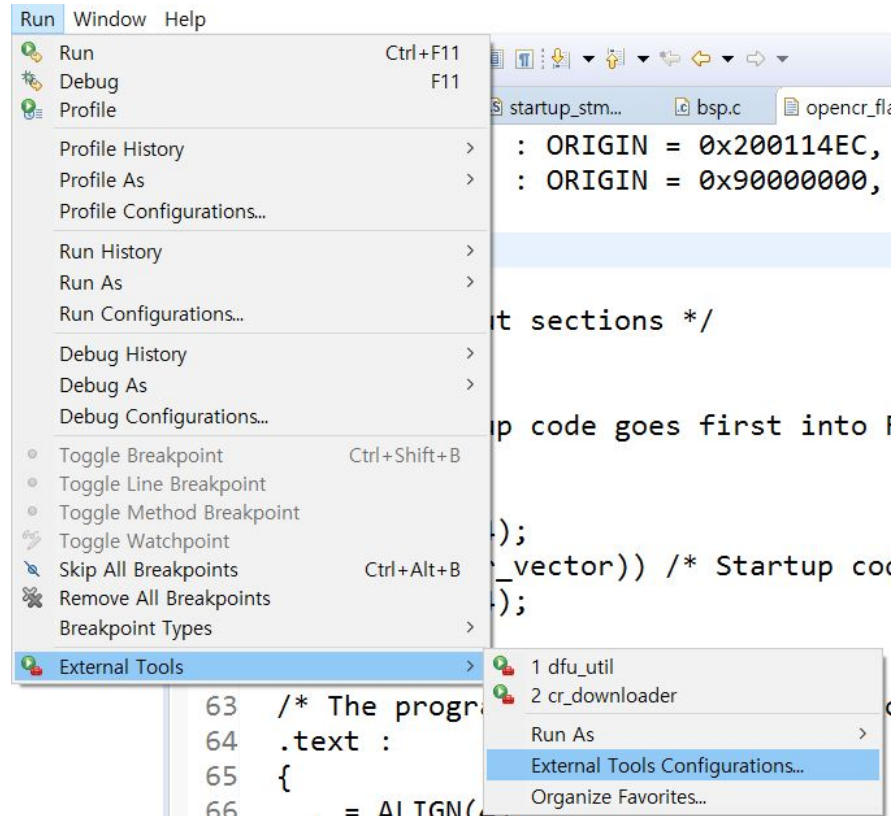
디버깅 설정 - OpenOCD

- OpenOCD 설정 저장
 - 디버깅 설정은 기본적으로 Local 즉 Workspace에 저장되어 Workspace를 변경하면 재설정해야 함으로 Common탭에서 Shared file로 하면 프로젝트 Import시 디버깅 설정도 같이 로드 된다.



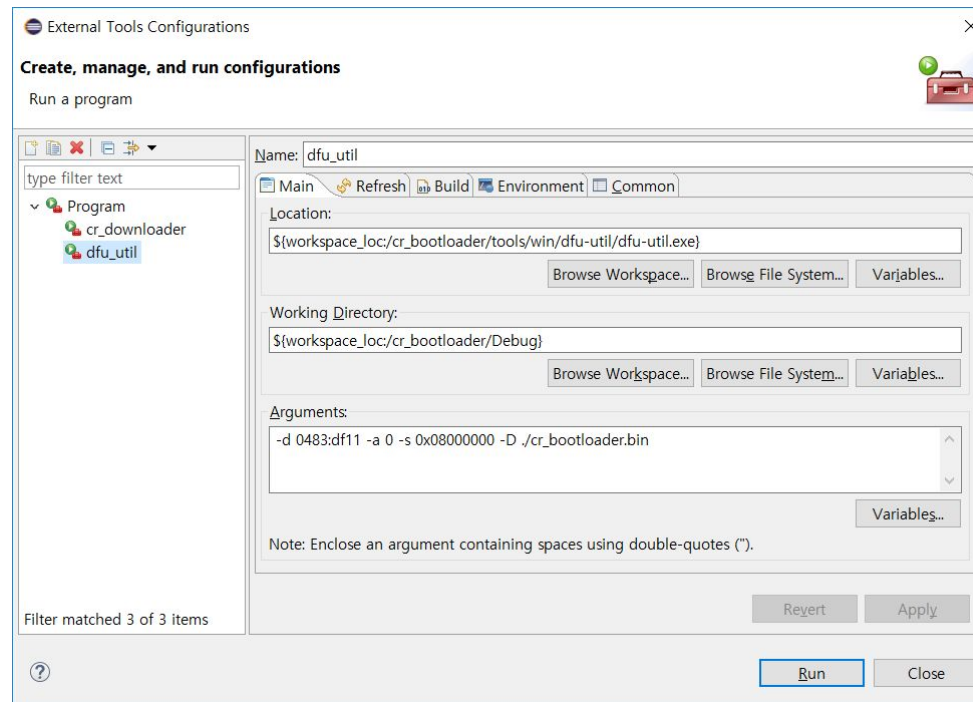
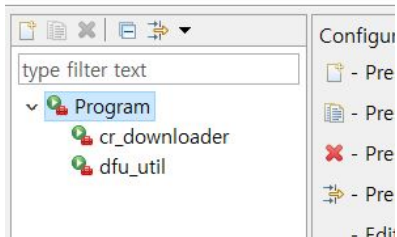
External Tools

- 외부 프로그램을 단축키처럼 등록해서 사용 가능
- Run->External Tools->External Tools Configurations 선택



External Tools

- Program 더블 클릭하여 신규 생성



이름

명령어 위치

작업 디렉토리

명령행 옵션