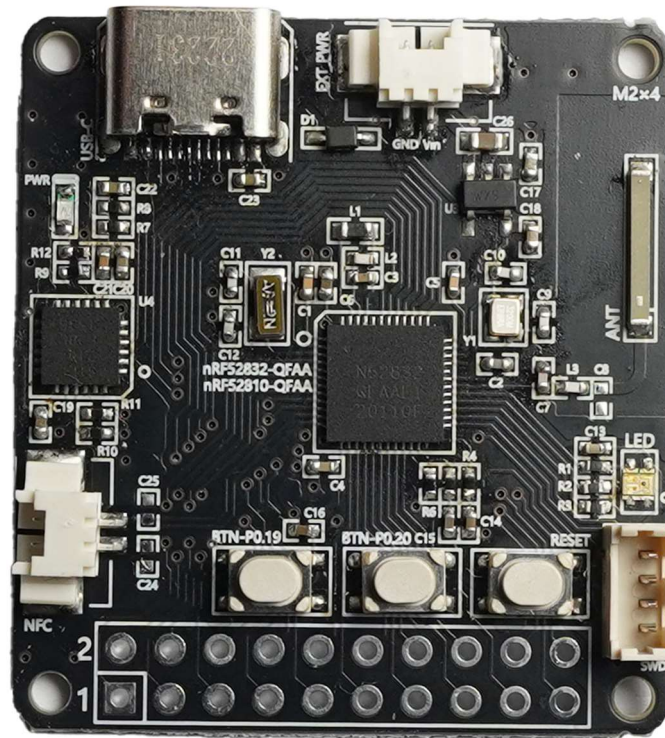


User Manual for

nRF52 Development Kit



V1.1.1

INDEX

1. 제품 개요 및 소개	1
1.1. 주요 특징 및 스펙	1
1.2. 핀 맵 및 각 부분별 명칭.....	4
1.3. 필수 툴 및 소프트웨어	5
1.4. 매뉴얼 업데이트 내역	6
2. 개발 환경 구축	7
2.1. 설치 순서 및 다운로드	7
2.2. nRF Connect 및 nRF Command Line Tool 설치	7
2.3. J-Link 혹은 OpenOCD 설치.....	13
2.4. Microsoft Visual Studio 설치 및 플러그인 설치	18
2.5. 샘플 프로젝트 : RGB LED 켜고 끄기.....	28
2.6. 샘플 프로젝트 : 버튼 및 LED 다루기.....	36
3. Roverdyn NCS Library	45
3.1. Roverdyn NCS 소개	45

1 제품 개요 및 소개

1.1. 주요 특징 및 스펙

본 제품은 블루투스 5 모뎀과 단정밀도 부동소수점 연산기(FPU)를 포함하는(nRF52832 버전만) Cortex-M4F 프로세서를 내장한 SoC(System on Chip)가 포함된 개발 키트로서 전용 SDK 와 무료 통합개발환경(IDE)를 제공하여 쉽고 빠르게 원하는 임베디드 시스템을 개발 할 수 있습니다. USB-C 커넥터를 내장하여 별다른 장비 없이 전원을 손쉽게 공급할 수 있으며 USB to UART Bridge 칩셋이 내장되어 있어 외부 디버거 없이 데스크탑 혹은 랩탑 환경하에서 빠르고 쉽게 디버깅 하거나 데이터를 주고 받을 수 있습니다. 2.54 핀 규격의 커넥터를 사용한 GPIO 핀과 MCU 의 GPIO 포트가 연결되어 있어 nRF52810 혹은 nRF52832 칩셋의 대부분의 핀을 자유로이 사용할 수 있습니다.

본 개발 키트는 별도의 디버거가 내장되어 있지 않으므로 J-Link 디버거 혹은 오픈 소스인 OpenOCD 를 지원하는 프로그래머 및 디버거를 사용하여 SWD 포트에 연결하여 소프트웨어를 디버깅하거나 프로그래밍 할 수 있습니다. SWD 포트는 Molex 4 핀 커넥터(Molex 53047-0410)로 제공되며필요에 따라 변환 어댑터를 사용하여 디버깅 할 수 있습니다.

■ Key Features

- 2 개의 MCU 타입 지원(nRF52810, nRF52832)
- USB-C 커넥터 채택(전원 공급 가능)
- 배터리 전원 공급 가능(단, USB-C 전원과 동시 사용 불가)
- USB to UART Bridge 내장(CP2104)
- Full-sized GPIO 커넥터 제공
- 2 개의 커스텀 버튼과 한 개의 리셋 버튼
- Molex Picoblade(53047-0410) SWD 인터페이스 제공
- RGB LED 내장
- External RTC 내장
- 고감도 칩 안테나 사용
- NFC 단자 제공

■ 적용 MCU

Part Number	RDEV-BR1-52810	RDEV-BR1-52832
MCU	nRF52810-QFAA	NRF52832-QFAA

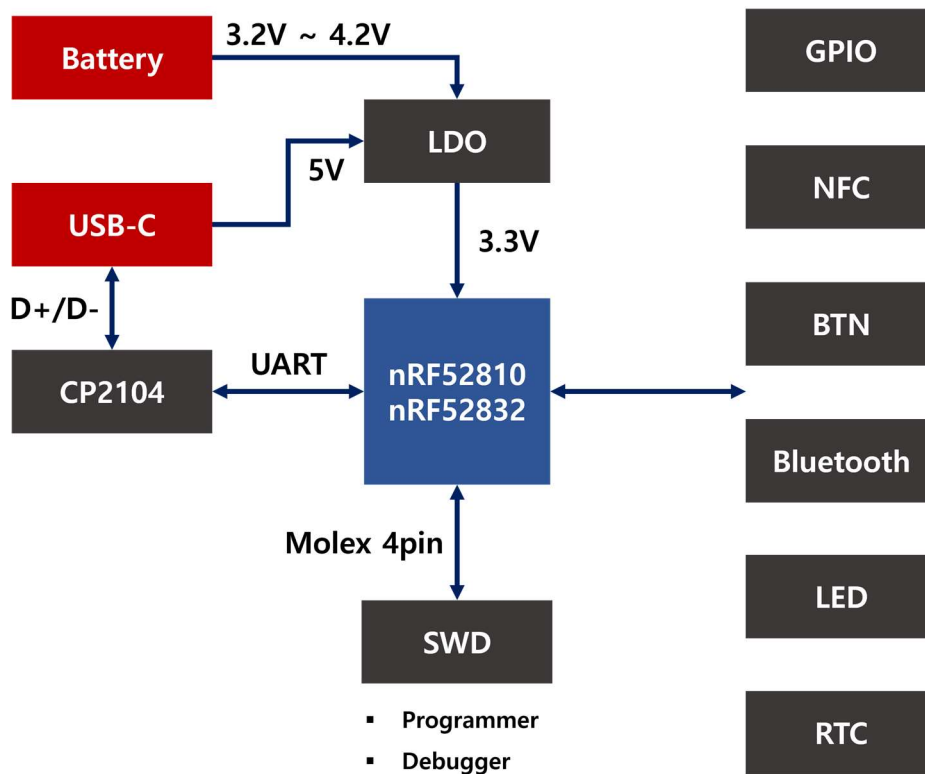
■ Specification

P/N	RDEV-BR1-52810	RDEV-BR1-52832
Dimensions	37mm(L) × 34mm(W) × 11.6mm(H)	
Weight	5.4g	
Op. Voltage	3.3VDC ~ 5VDC	
BLE Power	-20dBm ~ +4dBm	
Temperature	-20°C ~ 60°C	
CPU Freq.	64MHz	
RAM	24Kbytes	64Kbytes
FLASH	192Kbytes	512Kbytes
GPIO	32 채널	
I2C	1 채널	2 채널
UART	1 채널	
SPI	1 채널	2 채널
PWM	4 채널	12 채널
BLE	5.3(Mesh 지원 안함)	5.3(Mesh 지원)
FPU	없음	있음
ADC	8 채널	
TIMER	3 채널	5 채널
RTC	2 채널	3 채널

■ Bluetooth Specification

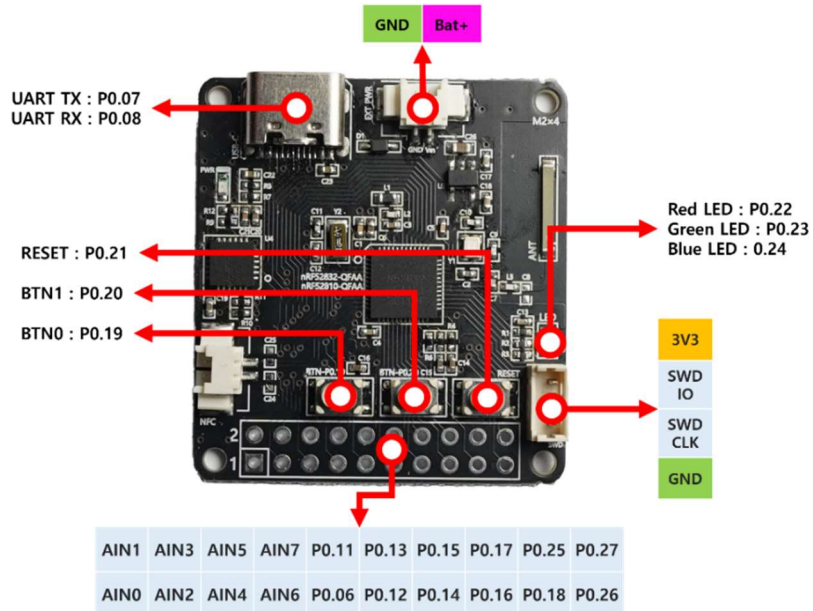
BluetoothLE Version	5.3
Frequency	2,400MHz ~ 2,483MHz
Bluetooth Type	BluetoothLE(1M, 2M)
UART Service UUID	6e400001-b5a3-f393-e0a9-e50e24dcca9e
Read Characteristic UUID	6e400003-b5a3-f393-e0a9-e50e24dcca9e
Write Characteristic UUID	6e400002-b5a3-f393-e0a9-e50e24dcca9e
Notify Descriptor	000002902-0000-1000-8000-00805f9b34fb
Power Consumption	Tx : 17.5 mW Rx : 12.2 mW

■ Block Diagram



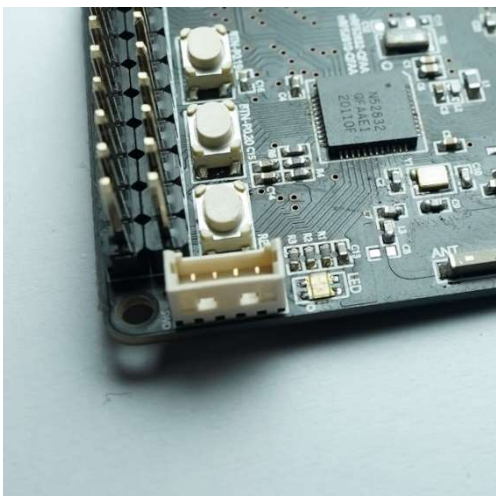
1.2. 핀 맵 및 각 부분의 이름

■ System Schematic & Pinout



■ Serial Wire Debug (SWD) Interface

SWD 포트는 Cortex-M SoC 의 플래시 메모리를 비롯하여 각종 기능을 디버깅 할 수 있는 단자입니다. NRF52 시리즈는 Cortex-M4 CPU 를 내장하고 있어 SWD 포트를 이용하여 Flash 메모리에 펌웨어를 다운로드 하거나 디버깅할 수 있습니다. 본 개발 보드는 4 핀 SWD 포트를 제공하며 Molex Picoblade(1.25mm 피치, P/N : 53047-0410) 커넥터를 이용하여 연결할 수 있습니다. 개발자는 J-Link 혹은 openOCD 디버거로 펌웨어를 다운로드하거나 디버깅할 수 있습니다.



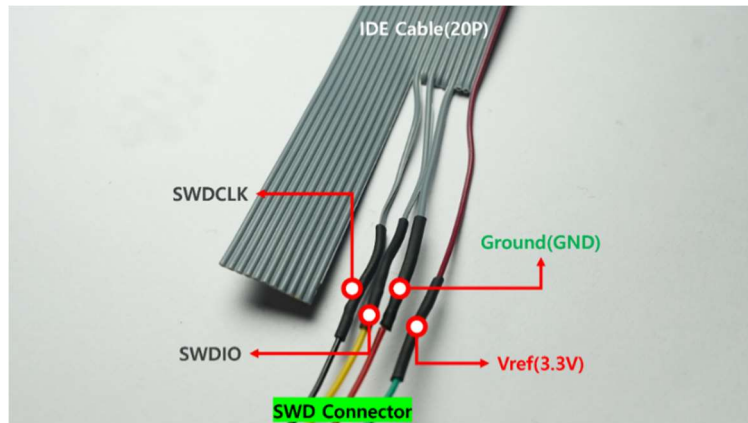
1.3. 필수 툴 및 소프트웨어

■ 개발에 필요한 필수 툴 및 소프트웨어 리스트

분류	이름	용도
하드웨어	J-Link	펌웨어 다운로드 및 디버깅
하드웨어	ST-Link V2	펌웨어 다운로드 및 디버깅(openOCD 환경 구축 필요)
하드웨어	Molex Picoblade Connector 4P	J-Link 혹은 ST-Link V2 와 개발 보드의 SWD 핀을 연결시켜 줌
소프트웨어	Microsoft Visual Studio Code	소스코드 작성, 프로젝트 관리 등 통합개발환경 역할 수행
소프트웨어	nRF Connect for Desktop	SDK 및 툴체인 설치, 펌웨어 다운로드 등
소프트웨어	nRF Command Line Tools	WEST 환경 제공(다운로드 및 디버깅)
소프트웨어	J-Link Software	J-link 드라이버 및 RTT Viewer 기능 제공
소프트웨어	openOCD	ST-Link V2 디버거를 사용하기 위한 소프트웨어 셋

■ SWD 핀 제작 및 변환 인터페이스 보드 (별매품)

- J-Link/ST-Link V2 의 20 핀 IDE 단자를 개발 보드의 4 핀 SWD 커넥터로 변환하는 인터페이스 보드는 별도로 구매할 수 있으며 IDE 케이블과 4 핀 Molex 커넥터는 구매 시 같이 제공됩니다.
 - ※ J-Link 및 ST-Link V2 는 포함되어 있지 않습니다
- 인터페이스 보드는 아래 사진과 같이 J-Link 혹은 ST-Link V2 와 연결하여 사용할 수 있습니다.
- 인터페이스 보드를 사용하지 않을 경우 IDE 케이블을 잘라서 4 핀 SWD 핀으로 개조할 수 있으며 아래 사진을 참조하시기 바랍니다.(구매 시 4 핀 Molex 커넥터는 제공됩니다)



1.4. 매뉴얼 수정 내역

순번	날짜	수정내용	비고
1	Feb. 1, 2024	매뉴얼 최초 배포	
2	Apr. 16, 2024	2.5, 2.6 및 챕터 3 생성	

2 개발 환경 구축

2.1. 설치 순서 및 다운로드

■ 설치 순서

1. Nordic Semiconductor 홈페이지에서 NRF SDK 및 NRF Command Line Tools 설치 파일을 다운로드 및 설치합니다.
2. J-link 혹은 openOCD 드라이버 및 소프트웨어를 다운로드 및 설치합니다.
3. Microsoft Visual Studio 를 설치합니다.
4. Microsoft Visual Studio 에서 컴파일 및 프로그래밍 환경에 관한 플러그인을 설치합니다.
5. 기타 환경 설정을 진행합니다.

■ 설치 순서

1. nRF Connect for Desktop 및 nRF Command Line Tools 설치
2. J-link 혹은 openOCD 설치
3. Microsoft Visual Studio Code 및 Plug-in 설치
4. 환경 설정 및 예제 컴파일

2.2. nRF Connect 및 nRF Command Line Tools 설치

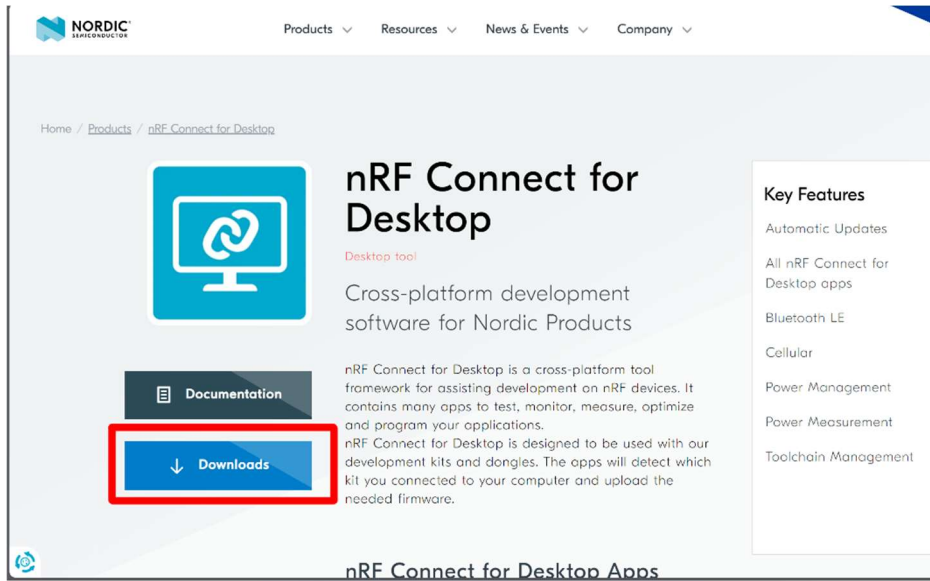
nRF Connect 는 nRF 시리즈 펌웨어를 개발하기 위해 Nordic Semiconductor 사에서 제공하는 종합 개발 툴입니다. nRF Connect 는 Nordic Semiconductor 사에서 설치파일을 다운로드 할 수 있습니다. nRF Command Line Tools 는 컴파일 된 펌웨어를 nRF 칩셋에 다운로드하거나 내부 플래시를 삭제하는 등의 작업을 수행할 수 있도록 환경을 구축해주는 툴입니다. 두 툴 모두 nRF52 시리즈 펌웨어를 개발하기 위해서 반드시 필요한 툴입니다.

■ nRF Connect for Desktop 다운로드 및 설치

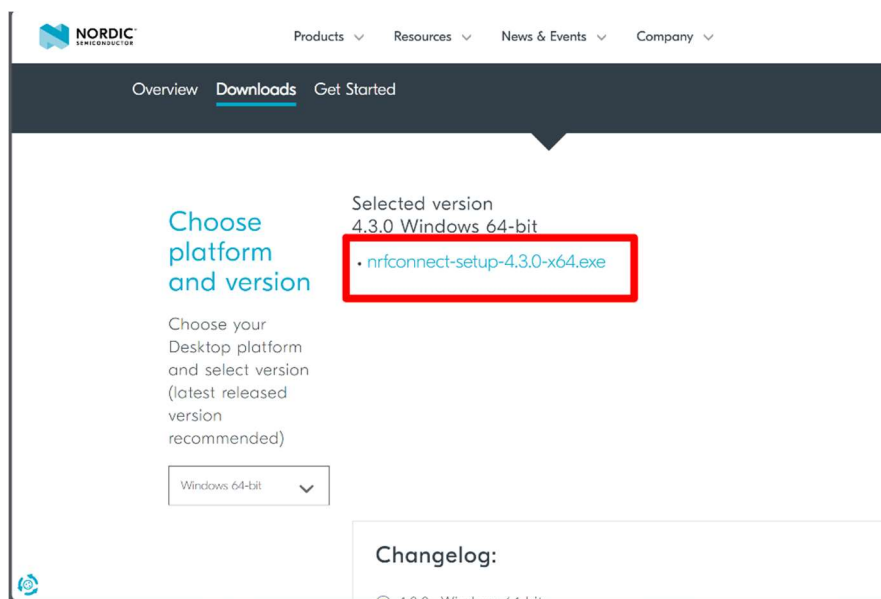
nRF Connector for Desktop 은 아래 다운로드 링크를 클릭하여 다운로드 할 수 있습니다. Nordic Semiconductor 사의 라이선스 정책에 의해 재배포는 금지되어 있어 직접 해당 홈페이지에서 다운로드 하여야 합니다.

[다운로드\(클릭\)](#)

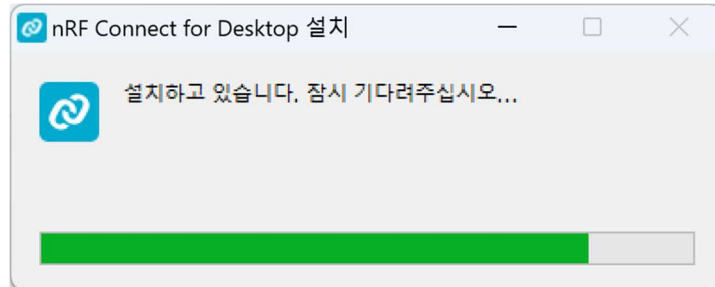
위 링크에 접속하면 다운로드 페이지가 나타나며 왼쪽 아래의 "Download" 버튼을 눌러 실제 다운로드 페이지로 이동할 수 있습니다.



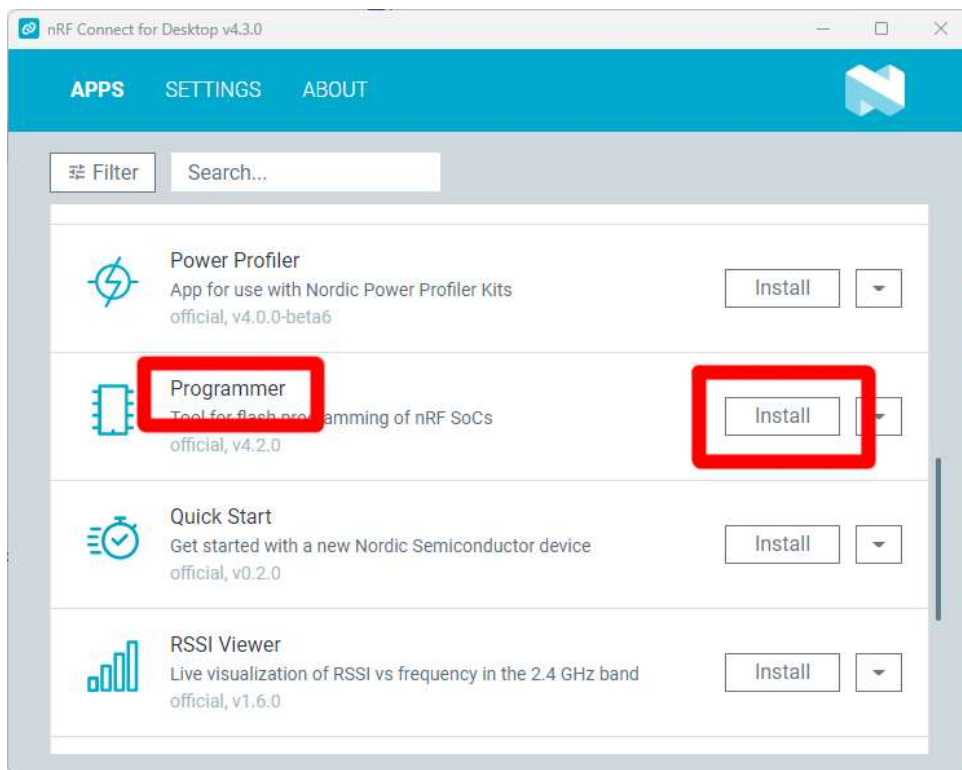
다운로드 페이지에 접속하면 최신 버전의 nRF Connect for Desktop 을 다운로드 할 수 있습니다. 다음 스크린샷을 참조하시기 바랍니다.



nRF Connect 설치 파일을 다운로드 한 후 설치 파일을 [마우스 오른쪽 클릭] - [관리자 권한으로 실행] 메뉴를 눌러 관리자 권한으로 설치할 수 있습니다. 설치는 다음과 같이 자동으로 진행됩니다.



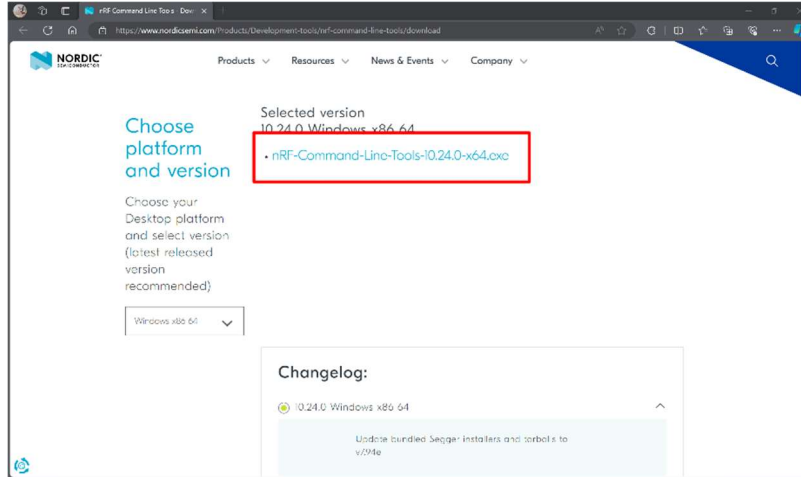
설치 이후 다음과 같은 포털 페이지가 나타나며 “Programmer” 항목을 찾은 후 “Install” 버튼을 눌러 프로그래머를 설치합니다. 만약 BLE 단말을 인증하기 위해 무선 Rx 및 Tx 항목을 측정해야 할 경우 “Direct Test Mode” 항목을 찾은 후 설치하면 됩니다.



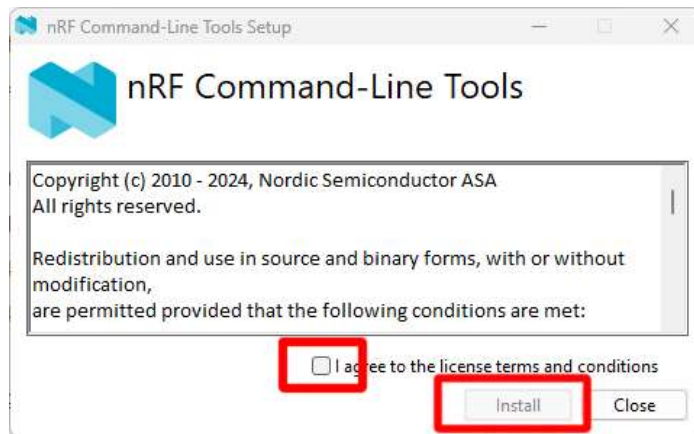
■ nRF Command Line Tools 다운로드 및 설치

nRF Command Line Tools 은 다음 링크에서 다운로드 할 수 있습니다.

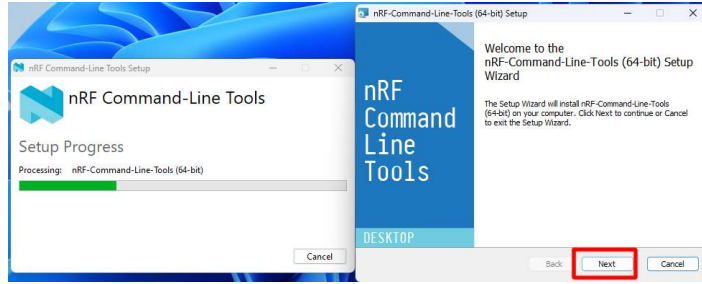
[다운로드\(클릭\)](#)



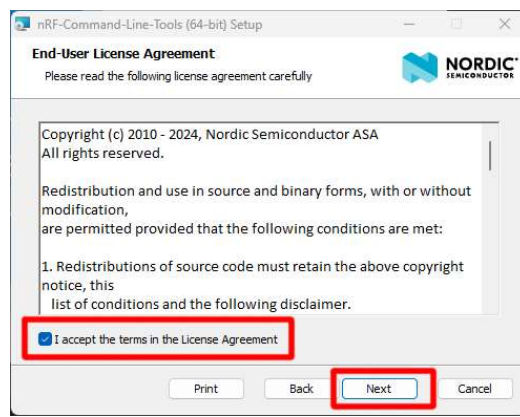
다운로드 페이지 접속 후 위 스크린샷을 참조해서 설치 파일을 다운로드 할 수 있습니다. 아래 순서를 참조하여 설치하면 됩니다.



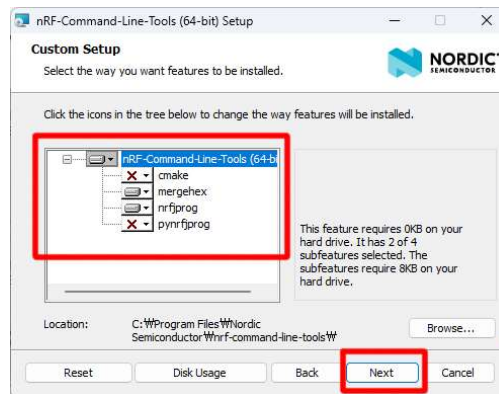
1. 위 그림을 참조하여 라이선스 조항에 동의한 후 [Install] 버튼을 눌러 다음 설치 단계를 진행합니다.



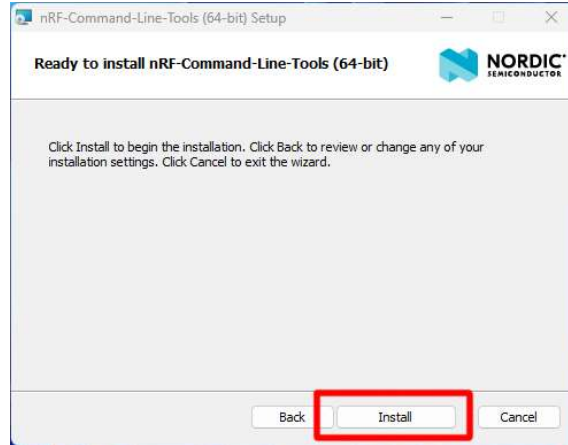
2. 설치가 진행되며 추가 설치 항목이 나타나며 [Next] 버튼을 눌러 진행합니다.



3. 엔드 유저 라이선스에 대한 내용을 동의하고 [Next] 버튼을 눌러 설치를 진행합니다.



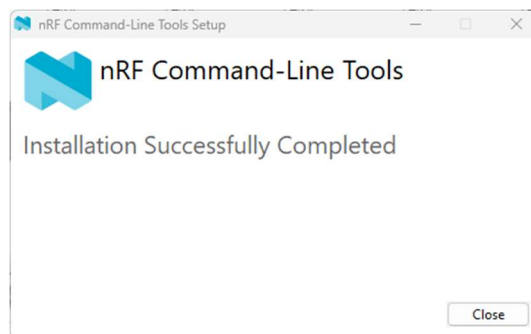
4. 설치 패키지에 포함된 항목 중 "mergehex" 및 "nrfjprog" 항목만 선택하고 [Next] 버튼을 눌러 다음 설치를 진행합니다. mergehex 는 hex 파일을 합쳐주는 툴이며 nrfjprog 는 컴파일된 펌웨어를 다운로드 할 수 있도록 해주는 툴입니다.



5. [Install] 버튼을 눌러 설치를 진행합니다.



6. 설치가 완료되면 “J-link” 소프트웨어를 설치하는 설치 창이 나타납니다. 최신 버전 설치를 위해서 해당 항목은 [Cancel] 버튼을 눌러 설치를 취소합니다.



7. 설치가 완료되면 위와 같은 창이 나타나며 [Cancel] 버튼을 눌러 종료합니다.

2.3. J-Link 혹은 openOCD 설치

J-Link 와 openOCD 는 Windows 혹은 Linux 환경에서 컴파일한 펌웨어를 다운로드하거나 프로그래밍 할 수 있도록 구성하는 드라이버를 설치하여야 사용할 수 있습니다. 보통 J-Link 를 사용하면 편리하게 펌웨어의 다운로드 및 디버깅을 할 수 있으나 ST-Link-V2 등을 사용하면 좀 더 저렴한 가격으로 동일한 동작을 수행할 수 있습니다. J-Link 및 openOCD 를 사용하기 위해서는 각각 별도의 소프트웨어를 설치하여야 사용할 수 있습니다.

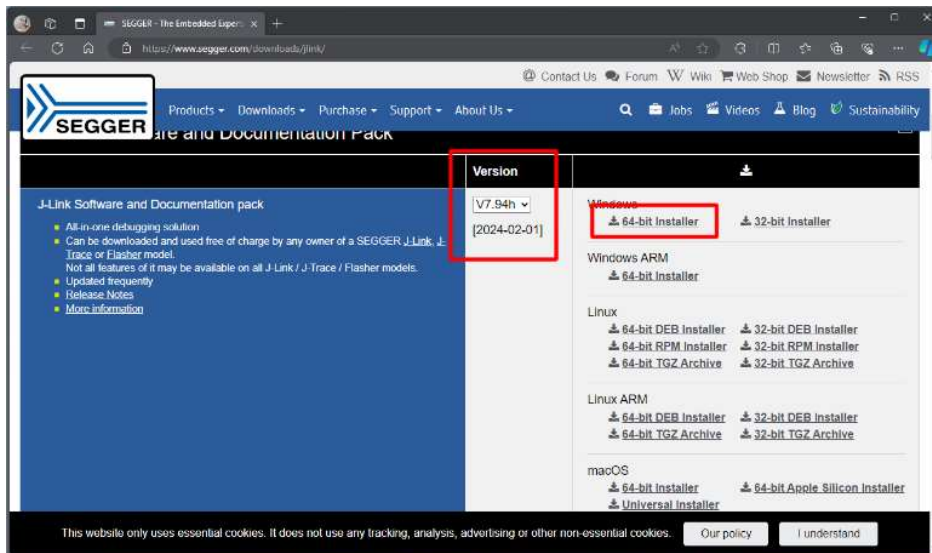
J-Link 및 openOCD 는 아래 링크에서 다운로드 할 수 있습니다.

[J-Link 다운로드](#)

[openOCD 다운로드](#)

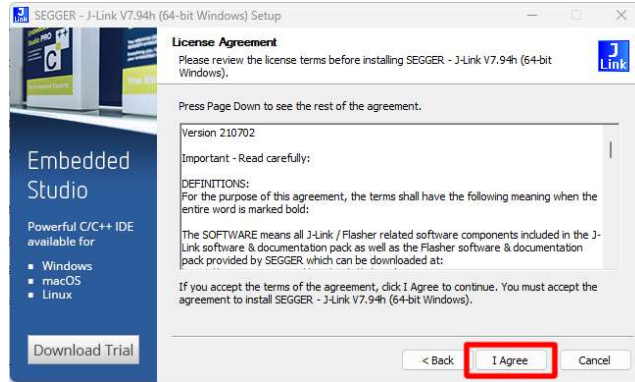
(openOCD 는 버전이 달라질 수 있습니다)

■ J-Link 다운로드 및 설치

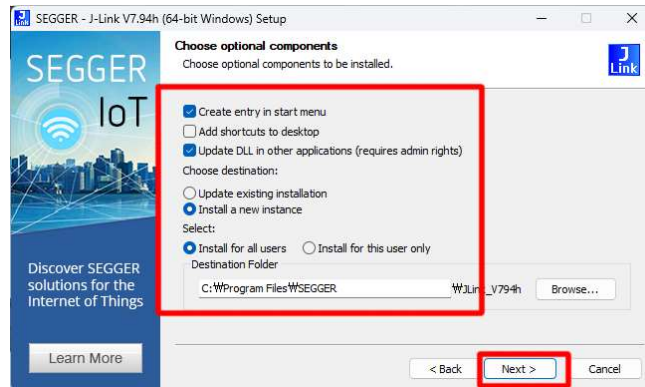


1. SEGGER J-Link 소프트웨어 다운로드 페이지로 이동하여 패키지를 다운로드 합니다. 다운로드 할 버전을 확인하고 사용하는 운영체제에 대해 맞는 설치 패키지를 다운로드합니다. 위 이미지는 Windows 11 을 기준으로 설치할 때의 예시입니다.

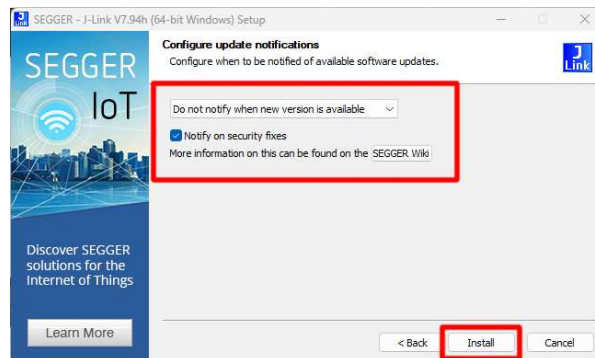
본 사용자 매뉴얼의 저작권에 대한 모든 권리는 로버다인(Roverdyn)에 있으며 로버다인의 허가 없이 복제, 기록, 재배포 및 번역할 수 없습니다. All rights to copyright in this User's Manual are in Roverdyn and may not be reproduced, recorded, redistributed or translated without Roverdyn's permission.



- 다운로드 완료된 설치 파일을 “관리자 권한으로 실행”하여 설치를 시작하여 [Next] 버튼을 누른 후 [I Agree] 버튼을 눌러 라이선스 조항에 동의합니다.

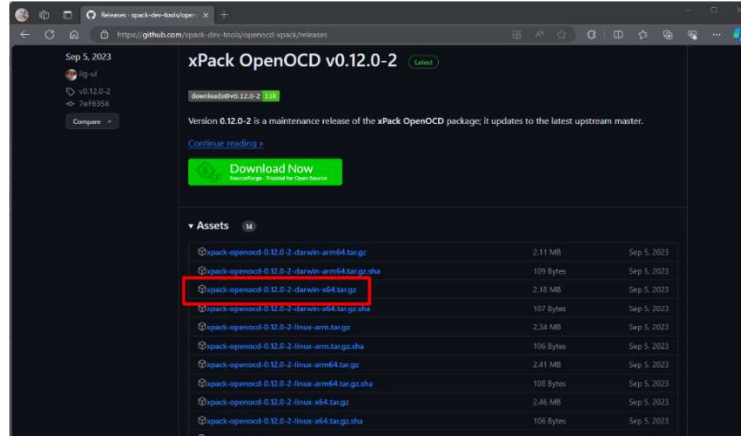


- 옵션 설치 항목을 확인합니다. 특별한 요구 사항이 없는 경우 위와 같이 설정하고 [Next] 버튼을 눌러 설치를 진행합니다.

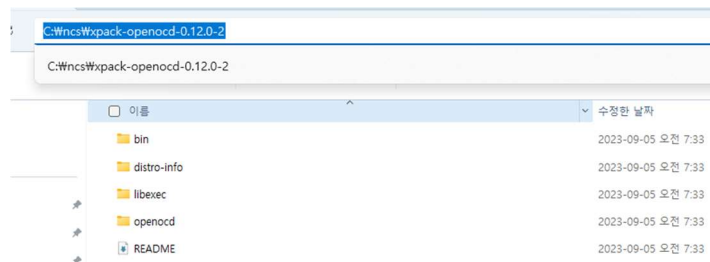


- 새 버전에 대한 알림 여부를 설정 한 후 [Install] 버튼을 눌러 설치를 시작합니다.

■ openOCD 다운로드 및 설치



1. openOCD 는 오픈소스 프로젝트로서 Github 에서 소스 코드 및 Release 버전을 다운로드 할 수 있습니다. 위의 예제 이미지에서는 Windows 11 을 기준으로 표기하였습니다.



2. 다운로드 한 압축 파일을 해제한 후 원하는 위치에 압축 해제한 폴더를 위치시킵니다. 예를들어 위 이미지에서는 C 드라이브의 NCS 폴더 내부에 압축 해제된 파일을 이동시켰습니다.



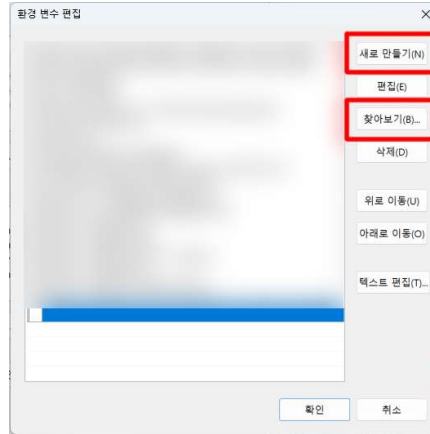
3. [시작]-[실행] 창을 연 후 "sysdm.cpl"을 입력하여 시스템 속성 창을 열어줍니다.



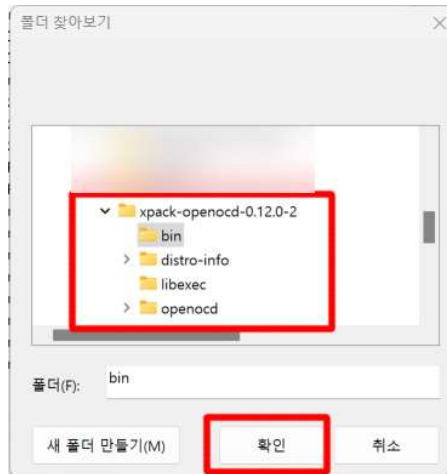
4. 시스템 속성 창이 나타나면 상단 탭 중 [고급] 탭을 클릭하여 들어간 후 아래쪽의 [환경 변수] 버튼을 눌러 환경변수 편집창을 열어줍니다.



5. 환경 변수 설정 창이 나타나면 아래 항목 중 [시스템 변수]-[Path] 항목을 선택한 후 아래쪽의 [편집] 버튼을 눌러 편집창을 열어줍니다.



6. 편집창이 나타나면 우측의 [새로 만들기] 버튼을 누른 후 [찾아보기] 버튼을 눌러 경로 설정창을 열어줍니다.



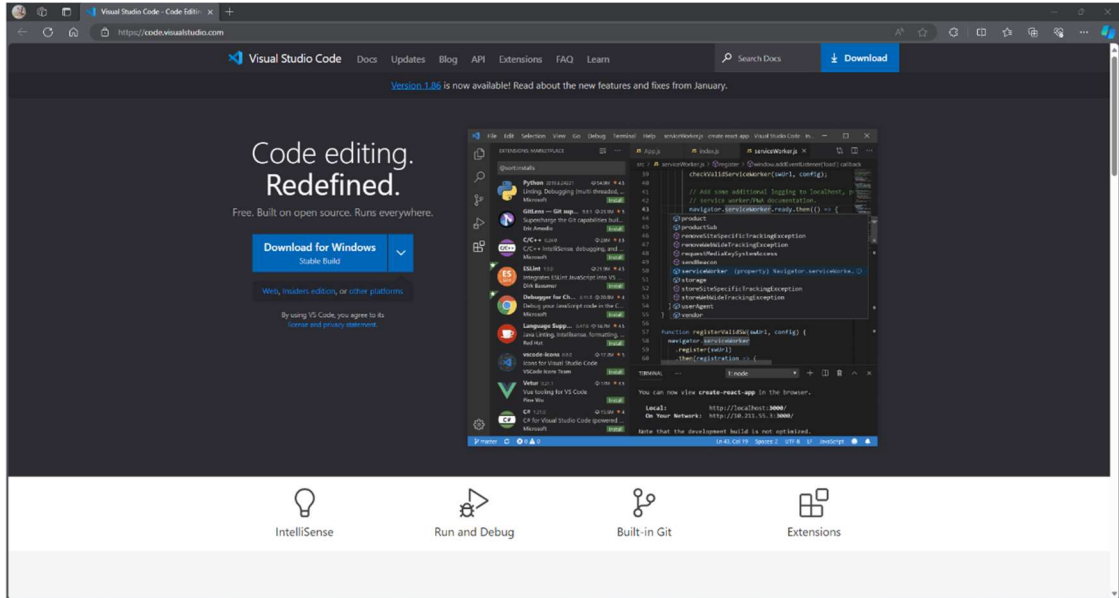
7. 2 번 항목에서 얻은 openOCD 압축 해제 폴더 경로를 찾아서 그 중 "bin" 폴더를 선택한 후 "확인" 버튼을 눌러 경로 입력을 완료합니다.



8. 입력된 경로를 확인한 후 [확인] 버튼을 눌러 경로 설정을 마무리합니다.

2.4. Microsoft Visual Studio 설치 및 플러그인 설치

2.4.1. Visual Studio Code(VSCoDe) 개요



Visual Studio Code 는 Microsoft 사에서 출시한 텍스트 에디터로서 IntelliSense 를 포함하여 개발자의 편의성을 중시하고 다양한 개발 환경을 지원하여 크로스 플랫폼 환경에서의 개발 환경까지 구성할 수 있으며 강력한 기능 확장을 지원하는 것이 강점입니다. VSCoDe 는 기본적으로 단순한 텍스트 에디터 기능만 제공하지만 확장 기능을 이용하여 필요한 컴파일러, 디버거 등을 설치하여 통합 개발 환경(IDE)로 구성하는 것이 가능합니다.

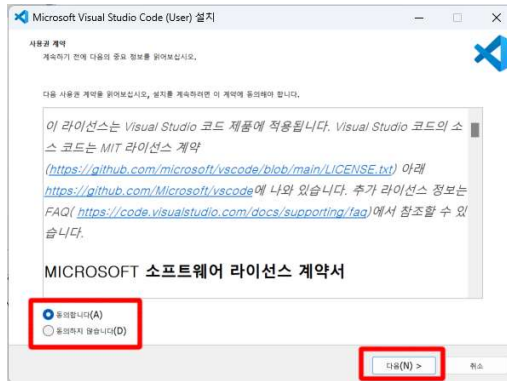
Visual Studio Code 는 마이크로소프트사의 공식 홈페이지에서 다운로드 할 수 있습니다. 아래 링크를 클릭하여 다운로드 페이지로 이동할 수 있습니다.

[다운로드\(클릭\)](#)

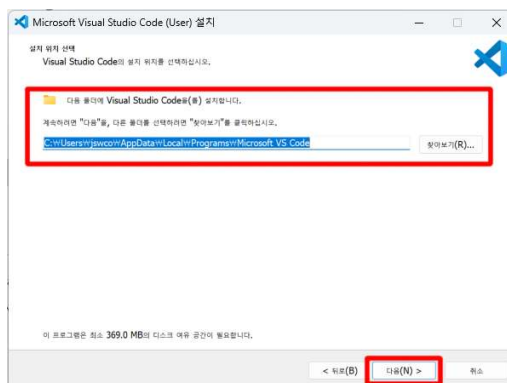
Visual Studio Code 를 다운로드 하기 전 사용하는 운영체제를 확인한 후 다운로드 하여야 합니다. Mac 및 Linux 모두 설치 패키지를 지원하며 기본적으로 64-bit 버전을 제공합니다. 또한 다음 권장 요구 사항을 만족하는지 확인하여야 합니다.

CPU : 1.6GHz 이상, 1GB RAM 이상, Windows 10 이상, Ubuntu 16.04 이상, macOS 10.13 이상

2.4.2. Microsoft Visual Studio 설치



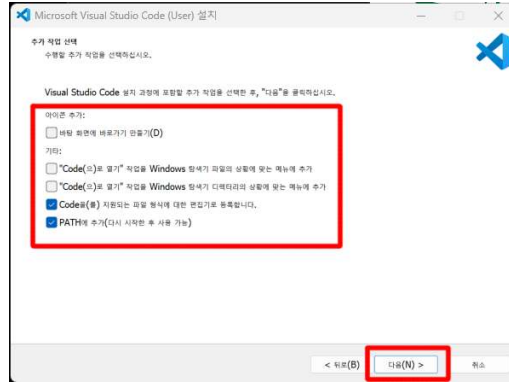
1. 다운로드 한 설치 파일을 마우스 오른쪽 클릭 한 후 "관리자 권한으로 실행"을 눌러 설치파일을 실행합니다. 그리고 사용권 계약에 대해 "동의합니다"를 선택한 후 [다음] 항목을 눌러 설치를 진행합니다.



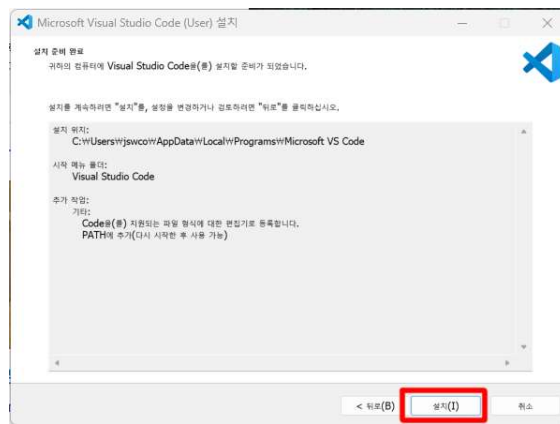
2. 프로그램의 설치 위치를 지정합니다. 특별한 사유가 없으면 위와 같이 기본 경로를 지정한 후 [다음] 버튼을 눌러 설치를 진행합니다.



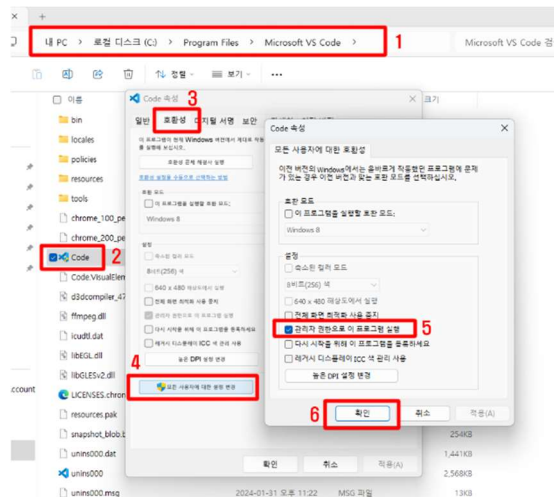
3. 시작 메뉴 폴더에 바로가기 위치 여부를 설정한 후 [다음] 버튼을 눌러 설치를 진행합니다.



4. 바로가기 및 기타 설정을 확인한 후 [다음] 버튼을 눌러 설치를 진행합니다.

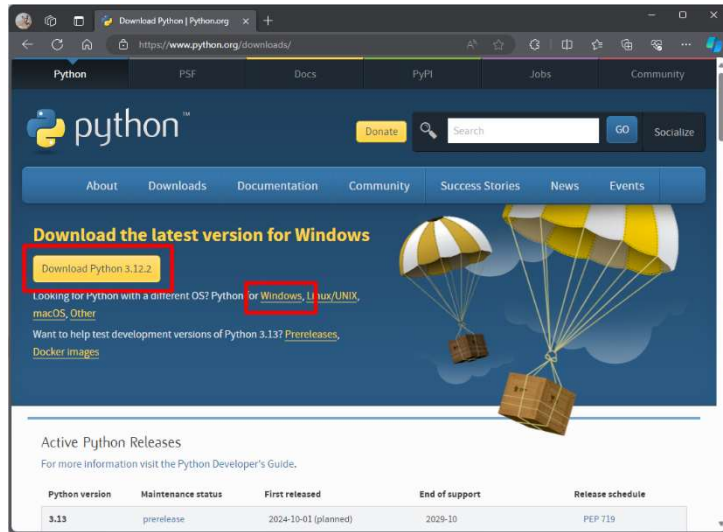


5. 설치 내용을 확인한 후 [설치] 버튼을 눌러 설치를 진행합니다.



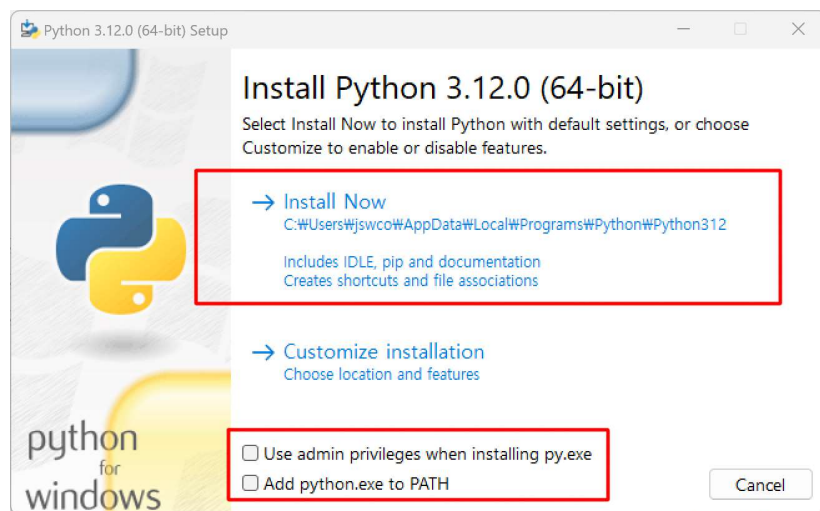
6. "C:\Program Files\Microsoft VS Code"로 이동하여 "Code.exe" 항목을 찾은 후 [마우스 오른쪽 클릭]-[속성]-[호환성]-[모든 사용자에게 대한 설정 변경]-[관리자 권한으로 이 프로그램 실행] - [확인]을 순서대로 눌러 관리자 권한으로 설정합니다.

2.4.3. Python 및 West 설치

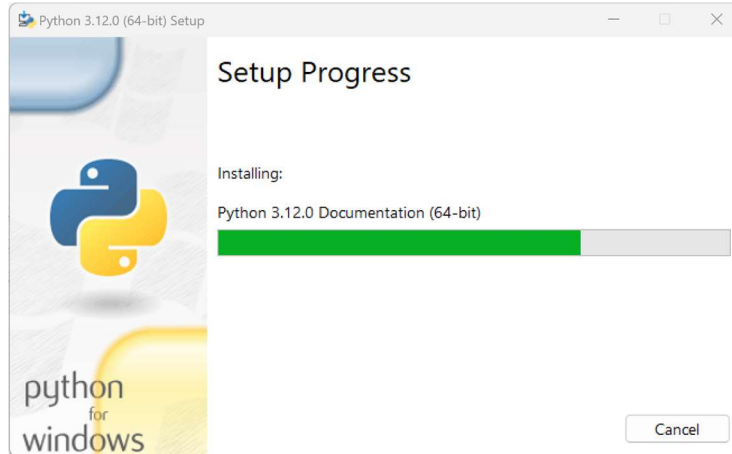


1. 파이썬 공식 홈페이지에서 설치 프로그램을 다운로드 합니다. 다운로드 링크는 아래에 있습니다. 다운로드 시 자신의 운영체제에 맞는 설치 패키지를 선택하여야 합니다.

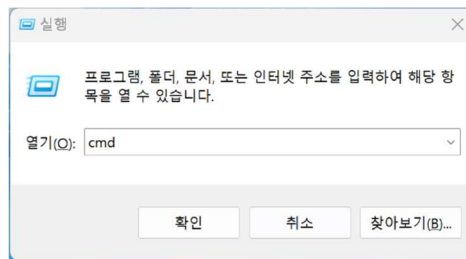
[Download Python | Python.org](https://www.python.org/downloads/)



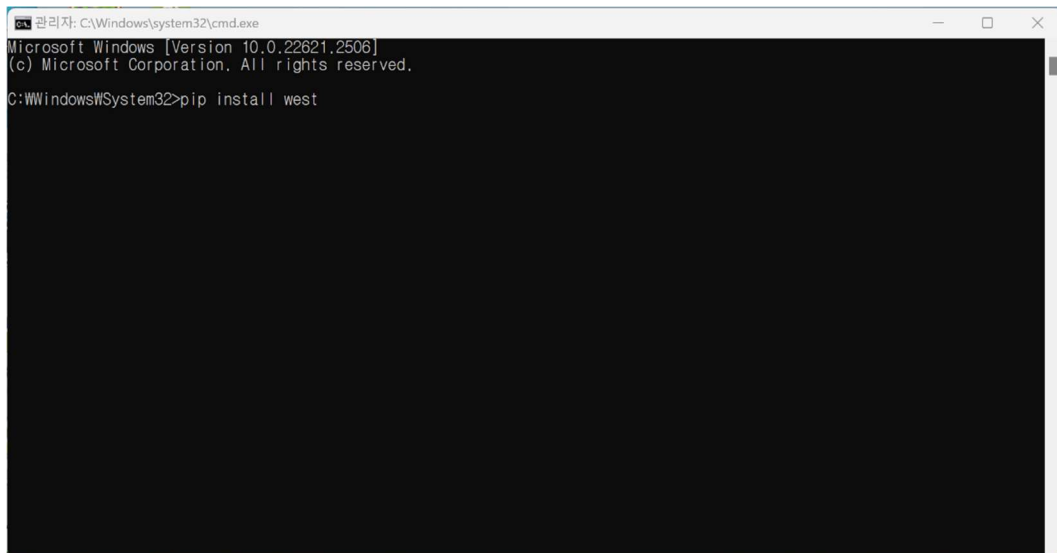
2. 다운로드 받은 설치 프로그램을 마우스 오른쪽 클릭 하여 [관리자 권한으로 실행] 으로 실행시킵니다. 설치 후에 별도로 설정하지 않도록 설치 시 아래쪽의 두 항목을 선택한 후 [Install Now] 버튼을 눌러 설치를 진행합니다.



3. Python 의 설치 과정은 자동으로 진행됩니다.



4. West 설치를 위하여 [시작]-[실행] 혹은 [Windows Key] + R 키를 눌러 "실행" 창을 불러온 후 "cmd" 항목을 입력한 후 [Ctrl] + Enter 키를 눌러 관리자 권한으로 명령 툴을 실행합니다.



5. 명령 창이 나타나면 다음 명령어를 입력하여 Python 의 West 패키지를 설치합니다.

```
pip install west
```



```

관리자: C:\Windows\system32\cmd.exe - pip install west
Microsoft Windows [Version 10.0.22621.2506]
(c) Microsoft Corporation. All rights reserved.

C:\Windows\System32>pip install west
Collecting west
  Obtaining dependency information for west from https://files.pythonhosted.org/packages/7a/6c/fe925971cb8890a944f49bbf4796126874a79854cc0d6783fb356f3fff31/west-1.2.0-py3-none-any.whl.metadata
  Downloading west-1.2.0-py3-none-any.whl.metadata (5.2 kB)
Collecting colorama (from west)
  Downloading colorama-0.4.6-py2.py3-none-any.whl (25 kB)
Collecting PyYAML>=5.1 (from west)
  Obtaining dependency information for PyYAML>=5.1 from https://files.pythonhosted.org/packages/2b/9f/fbade56564ad486809c27b222d0f7e6a89c01f6b4fe208402e90d4443a99/PyYAML-6.0.1-cp312-cp312-win_amd64.whl.metadata
  Downloading PyYAML-6.0.1-cp312-cp312-win_amd64.whl.metadata (2.1 kB)
Collecting pykwalify (from west)
  Downloading pykwalify-1.8.0-py2.py3-none-any.whl (24 kB)
Collecting setuptools (from west)
  Obtaining dependency information for setuptools from https://files.pythonhosted.org/packages/bb/26/7945080113158354380a12ce26873dd6c1ebd88d47f5bc24e2c5bb38c16a/setuptools-68.2.2-py3-none-any.whl.metadata
  Downloading setuptools-68.2.2-py3-none-any.whl.metadata (6.3 kB)
Collecting packaging (from west)
  Obtaining dependency information for packaging from https://files.pythonhosted.org/packages/ec/1a/610693ac4ee14fcdf2d9bf3c493370e4f2ef7ae2e19217d7a237ff42367d/packaging-23.2-py3-none-any.whl.metadata
  Downloading packaging-23.2-py3-none-any.whl.metadata (3.2 kB)
Collecting docopt>=0.6.2 (from pykwalify->west)
  Downloading docopt-0.6.2.tar.gz (25 kB)
  Installing build dependencies ...
  
```

6. west 설치 명령을 실행시키면 자동으로 의존성 패키지와 필수 패키지를 설치하게 됩니다.

```

관리자: C:\Windows\system32\cmd.exe
Microsoft Windows [Version 10.0.22631.3085]
(c) Microsoft Corporation. All rights reserved.

C:\Windows\System32>west -V
West version: v1.2.0

C:\Windows\System32>
  
```

7. 설치가 완료된 후 다음 명령어를 입력하여 정상 설치 여부를 확인할 수 있습니다. west 가 정상적으로 설치되어 있지 않거나 path 환경 변수가 정상적으로 설정되어 있지 않다면 “west 은(는) 내부 또는 외부 명령, 실행할 수 있는 프로그램, 또는 배치 파일이 아닙니다.” 라는 문구가 출력되며 실행되지 않습니다.

```
west -V
```

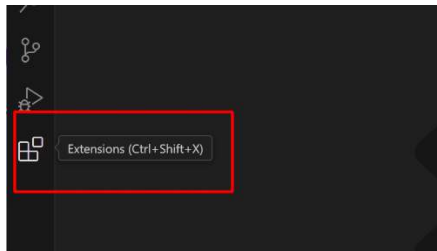
2.4.4. VS Code Plug-in 설치

nRF52 시리즈의 펌웨어를 개발하고 다운로드 하기 위해서는 Visual Studio Code 에 몇 가지 패키지를 설치하여야 합니다. VSCode 는 기본적으로 단순한 텍스트 에디터이므로 작성한 코드를 컴파일하고 다운로드 할 확장 기능(Extension)이 필요합니다.

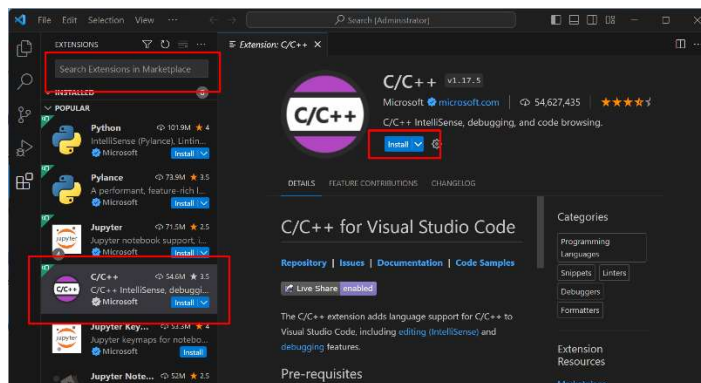
대표적으로 다음과 같은 확장 기능을 설치하여야 합니다.

- C/C++ Extension(필수)
- CMake Tools(필수)
- C/C++ Extension Pack(필수)
- nRF Connect for VS Code Extension(필수)
- Git Extension(선택)

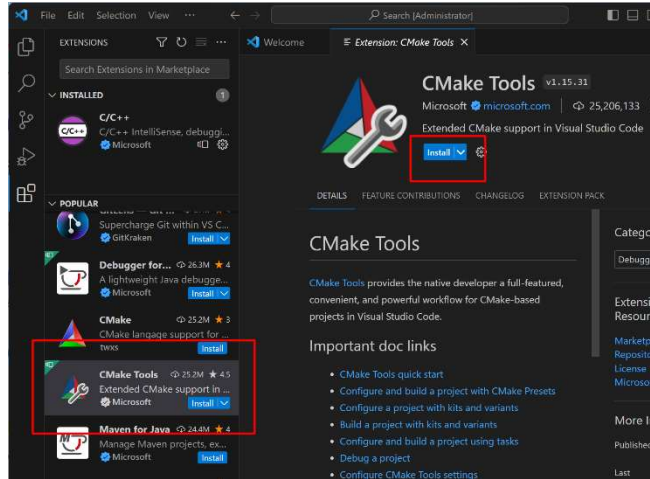
확장 기능을 설치할 때 반드시 VSCode 를 관리자 권한으로 실행하여야 합니다.



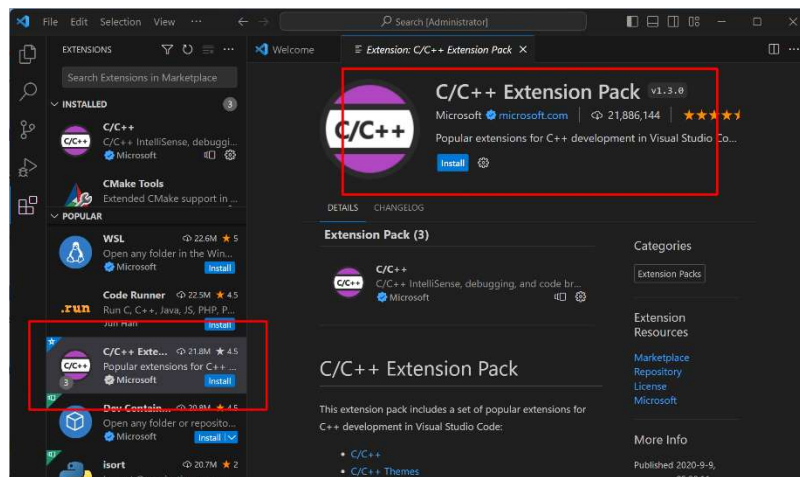
1. VSCode 를 실행한 후 좌측의 Extensions 항목을 눌러 확장 기능 설치 패널을 열어줍니다.



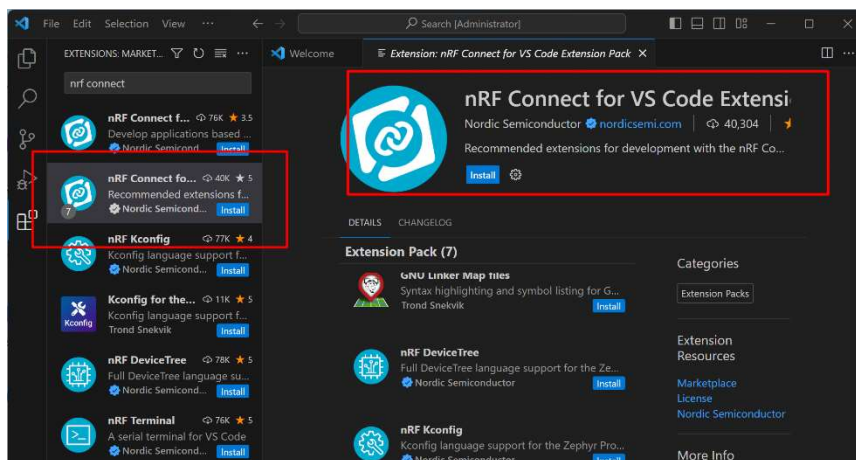
2. 확장 패널 상단의 검색 창에 "C/C++"을 입력하여 검색한 후 C/C++ 항목이 나타나면 선택 후 나타나는 상세 창에 있는 [Install] 버튼을 눌러 설치를 진행합니다.



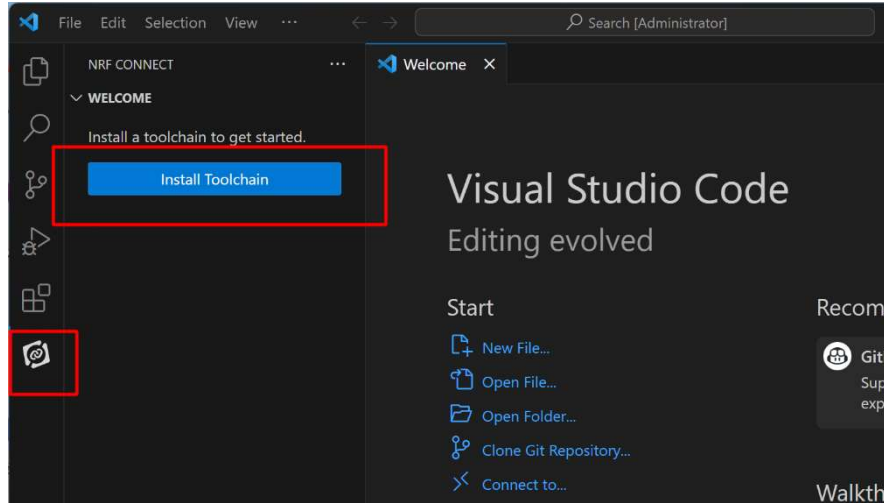
3. 위와 같은 방법으로 “CMake Tools”를 검색하여 설치를 진행합니다.



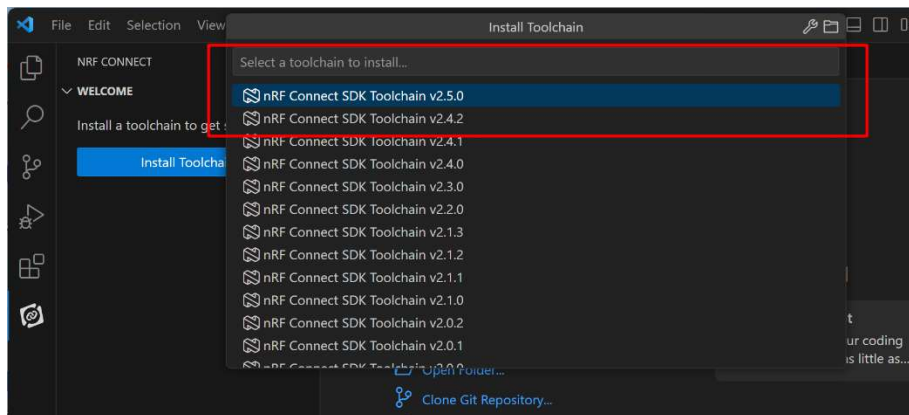
4. 위와 같은 방법으로 “C/C++ Extension Pack”을 설치합니다.



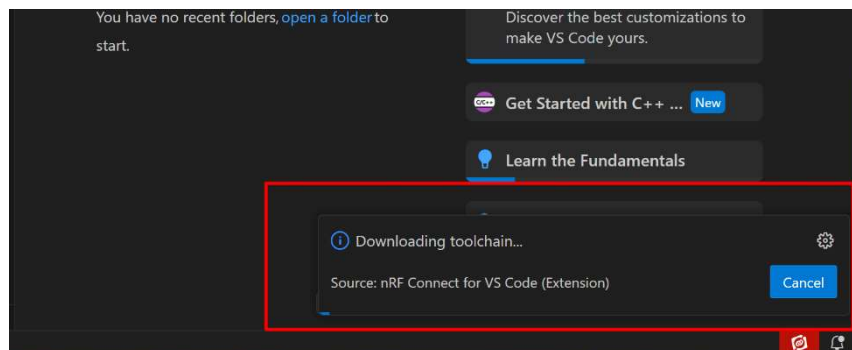
5. 위와 같은 방법으로 “nRF Connect for VS Code Extension”을 설치합니다.



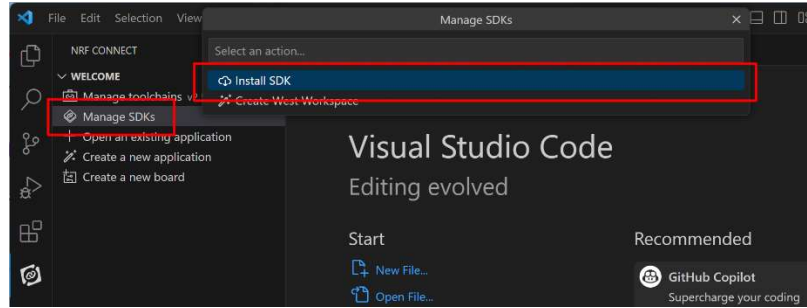
6. "nRF Connect for VS Code Extension"을 설치한 후 VS Code 를 종료한 후 다시 실행시킵니다. 그리고 왼쪽의 NRF CONNECT 패널이 나타나는 것을 확인한 후 해당 패널을 눌러 [Install Toolchain] 버튼을 눌러 툴체인을 설치합니다.



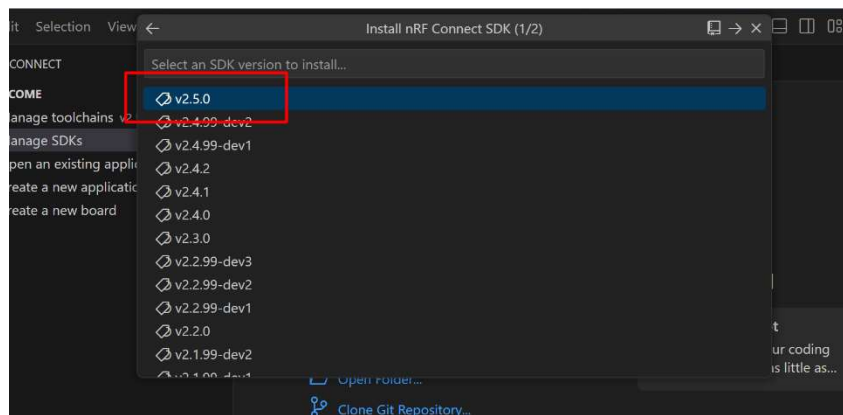
7. 툴 체인 설치 창이 나타나면 상단의 최신 버전을 눌러 설치를 진행합니다.



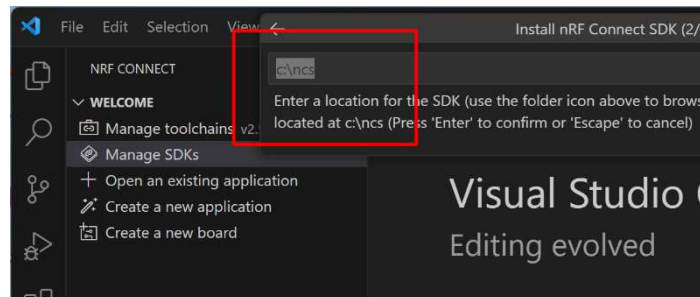
8. Toolchain 을 선택하면 자동으로 설치가 진행되며 우측 하단에 설치 진행률이 나타납니다.



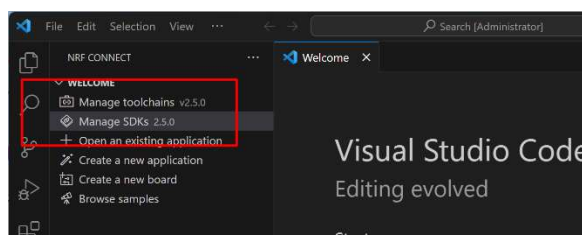
9. 툴체인 설치가 완료된 후 [Manage SDK] 항목이 나타납니다. 해당 메뉴를 누른 후 액션 창이 나타나면 [Install SDK]를 눌러 SDK 설치를 진행합니다.



10. SDK 버전을 선택하는 창이 나타납니다. 최신 버전을 선택하여 설치를 진행합니다.



11. 마지막으로 SDK 를 설치할 폴더를 설정합니다. 기본 설정대로 진행하는 것을 권장합니다.

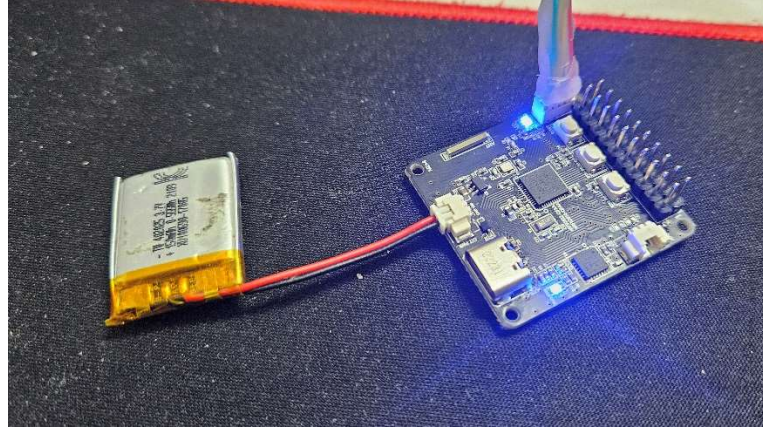


12. 설치가 완료된 후 툴체인 버전과 SDK 버전이 정상적으로 나타나는지 확인합니다.

본 사용자 매뉴얼의 저작권에 대한 모든 권리는 로버다인(Roverdyn)에 있으며 로버다인의 허가 없이 복제, 기록, 재배포 및 번역할 수 없습니다.
All rights to copyright in this User's Manual are in Roverdyn and may not be reproduced, recorded, redistributed or translated without Roverdyn's permission.

2.5. 샘플 프로젝트 : RGB LED 켜고 끄기

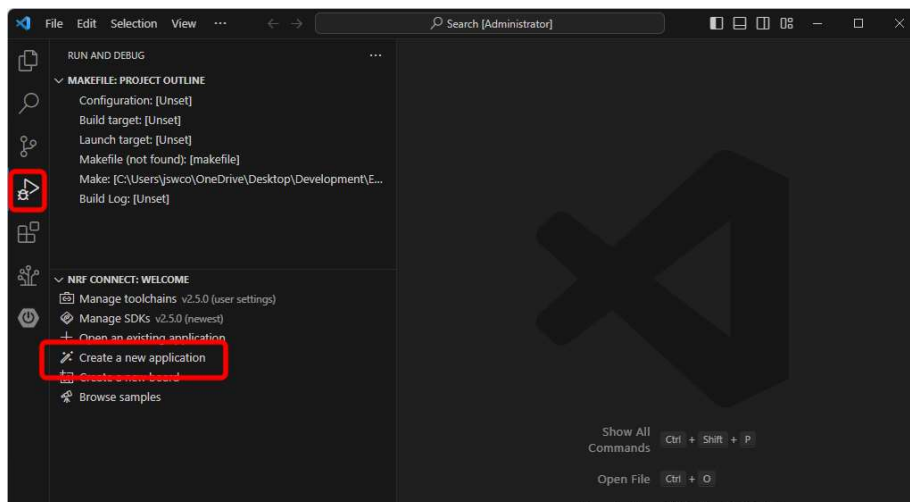
■ 개요



이 예제는 개발 보드에 내장되어 있는 3 색 LED 의 색상이 변화하도록 동작하는 펌웨어입니다. 애플리케이션을 실행시키면 LED 의 색상이 1 초 간격으로 빨간색(Red) - 초록색(Green) - 파란색(Blue) 순서대로 켜지며 무한히 반복됩니다.

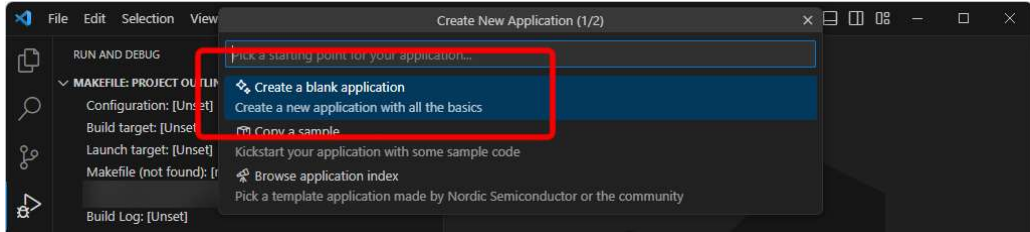
■ 프로젝트 생성

1. 프로젝트 생성을 위해 Visual Studio Code 를 실행시킵니다. 이 때, 개발 환경이 구축되어 있지 않다면 위의 [2.4] 항목을 참조하여 기본 개발 환경을 구축하여야 합니다.



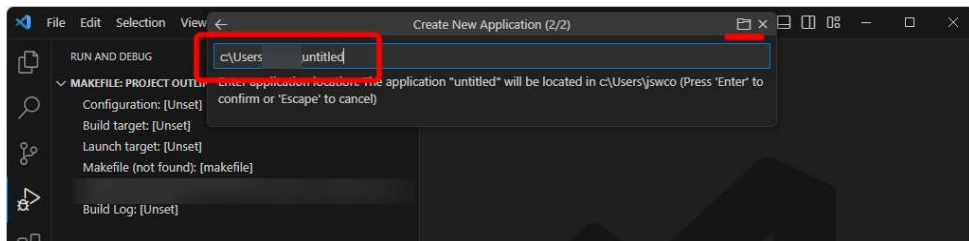
개발 환경이 구축되었다면 왼쪽의 [Run and Debug] 아이콘을 누른 후 [NRF CONNECT: WELCOME] – [Create a new application] 항목을 눌러 새로운 프로젝트를 생성합니다.

- 상단의 “Create New Application” 패널이 나타나는지 확인합니다.

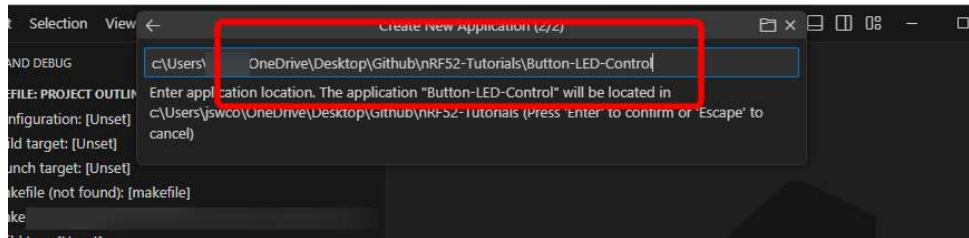


해당 패널에서 [Create a blank application] 메뉴를 선택하여 빈 애플리케이션을 생성합니다.

- 프로젝트 폴더가 위치할 경로를 작성합니다. 프로젝트 폴더의 경로명은 가능한 한글 없이 영문만 사용하는 것을 권장합니다.

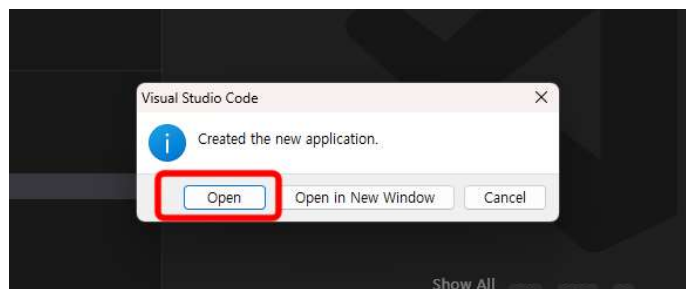


프로젝트 경로는 직접 입력이 가능하지만, Windows 탐색기를 이용하여 설정할 수 있습니다. 우측 상단의 폴더 모양의 아이콘을 클릭하면 탐색기를 이용하여 경로 설정을 할 수 있습니다.



탐색기를 이용하여 경로를 설정할 때에는 폴더는 해당 탐색기에서 별도로 생성해주어야 합니다.

- 프로젝트가 생성되면 [Open] 버튼을 눌러 프로젝트 페이지를 열어줍니다.



■ 예제 코드

```

/*
MIT License

Copyright (c) 2024. Seonguk Jeong, Roverdyn Inc.

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.
*/
#include <zephyr/kernel.h>

/* LED pin no. definition */
#define LED_RED      22
#define LED_GREEN    23
#define LED_BLUE     24
int main(void)
{
    // Debug message for start
    printk("Application start.\n");

    // Set gpio direction for LEDs & turn off
    NRF_GPIO->DIRSET = (1 << LED_RED) | (1 << LED_GREEN) | (1 << LED_BLUE);
    NRF_GPIO->OUTSET = (1 << LED_RED) | (1 << LED_GREEN) | (1 << LED_BLUE);

    // Main loop
    while(true){
        // Turn off all LEDs & Turn on Red
        NRF_GPIO->OUTSET = (1 << LED_RED) | (1 << LED_GREEN) | (1 << LED_BLUE);
        NRF_GPIO->OUTCLR = (1 << LED_RED);
        printk("Red LED on\n");
        k_msleep(1000);

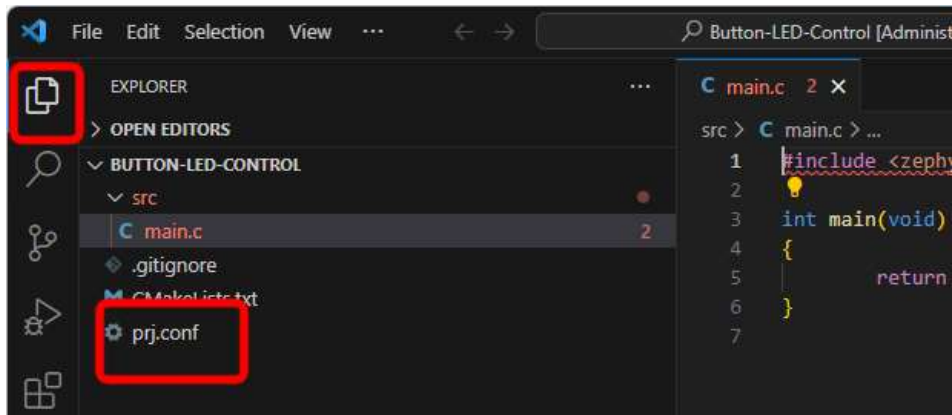
        // Turn off all LEDs & Turn on Green
        NRF_GPIO->OUTSET = (1 << LED_RED) | (1 << LED_GREEN) | (1 << LED_BLUE);
        NRF_GPIO->OUTCLR = (1 << LED_GREEN);
        printk("Green LED on\n");
        k_msleep(1000);

        // Turn off all LEDs & Turn on Blue
        NRF_GPIO->OUTSET = (1 << LED_RED) | (1 << LED_GREEN) | (1 << LED_BLUE);
        NRF_GPIO->OUTCLR = (1 << LED_BLUE);
        printk("Blue LED on\n");
        k_msleep(1000);
    }
    return 0;
}

```

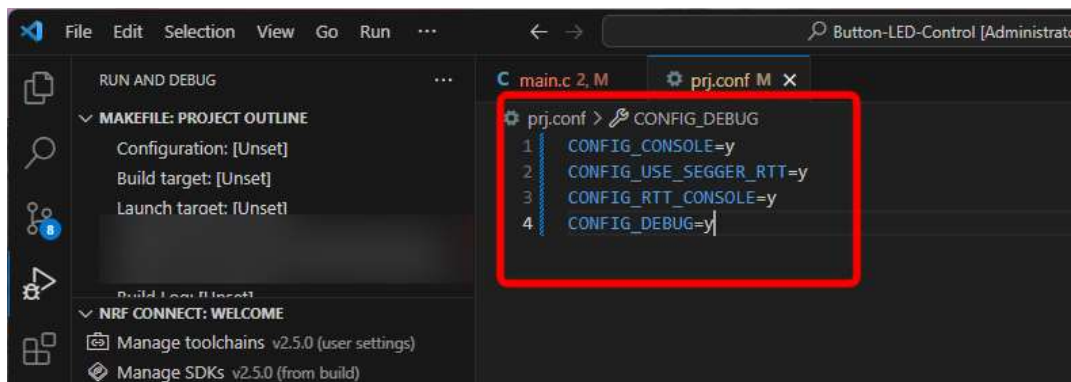

■ 프로젝트 설정

1. 애플리케이션을 실행하기 위해서 프로젝트 설정 파일(prj.conf)을 열어줍니다.



프로젝트 설정 파일은 왼쪽 상단의 [Explorer] 아이콘을 눌러 찾을 수 있습니다. prj.conf 항목을 더블클릭하여 열어줍니다.

2. 다음과 같이 설정 항목을 작성합니다.

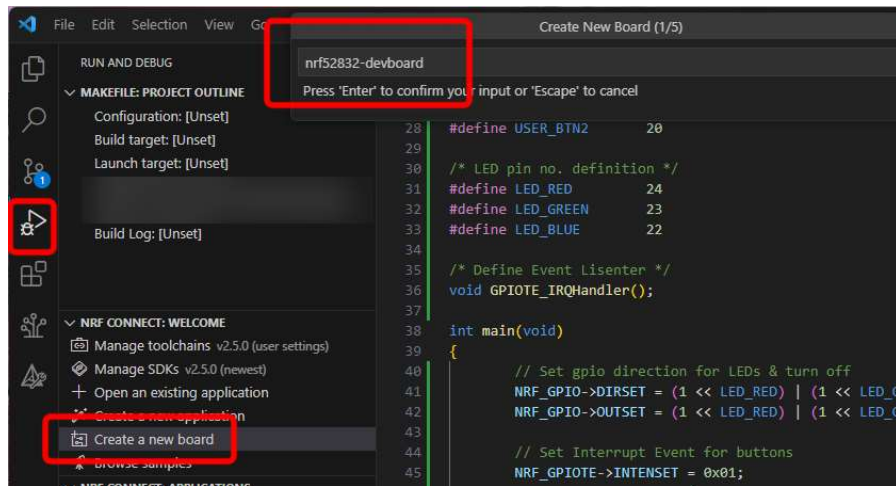


```
CONFIG_DEBUG=y
CONFIG_CONSOLE=y
CONFIG_USE_SEGGER_RTT=y
CONFIG_RTT_CONSOLE=y
```

■ 컴파일 및 다운로드

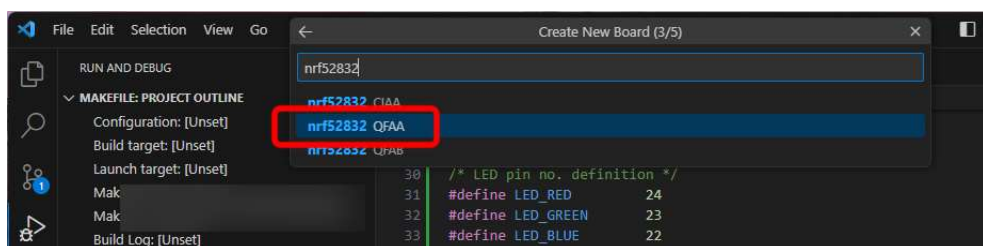
작성한 애플리케이션 코드는 컴파일 과정을 통하여 개발 보드로 다운로드 할 수 있는 포맷(*.elf 혹은 *.hex) 으로 변환됩니다. ARM Cortex-M 의 바이너리 파일(hex)은 Intel Hex 포맷으로 구성되어 있습니다. 다음 절차를 따라 작성한 코드를 컴파일 및 다운로드 할 수 있습니다.

1. 좌측 아이콘 중 [Run and Debug] 항목을 누른 다음 [NRF CONNECT: WELCOME] – [Create a new board] 메뉴를 눌러줍니다.



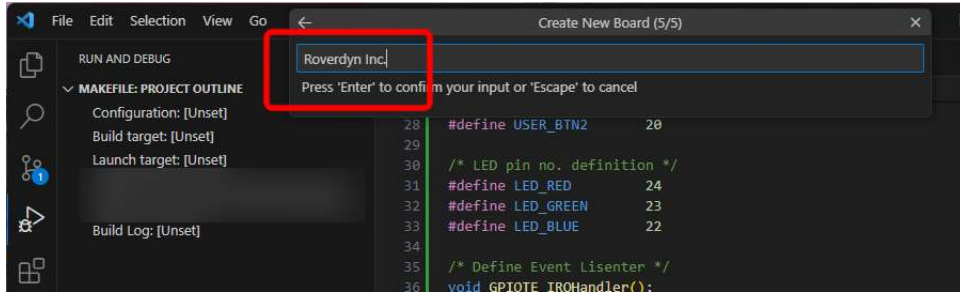
새 보드 생성 창이 나타나면 개발 보드의 이름을 설정합니다. 개발보드의 이름을 설정한 후 [Enter] 키를 눌러 다음 항목으로 진행합니다. 다음 항목은 컴퓨터가 읽을 수 있는 형태의 이름을 지정하는데, 별도로 설정하지 말고 다시 [Enter]를 눌러 진행합니다.

2. 가지고 있는 개발 보드의 MCU 타입을 선택합니다.



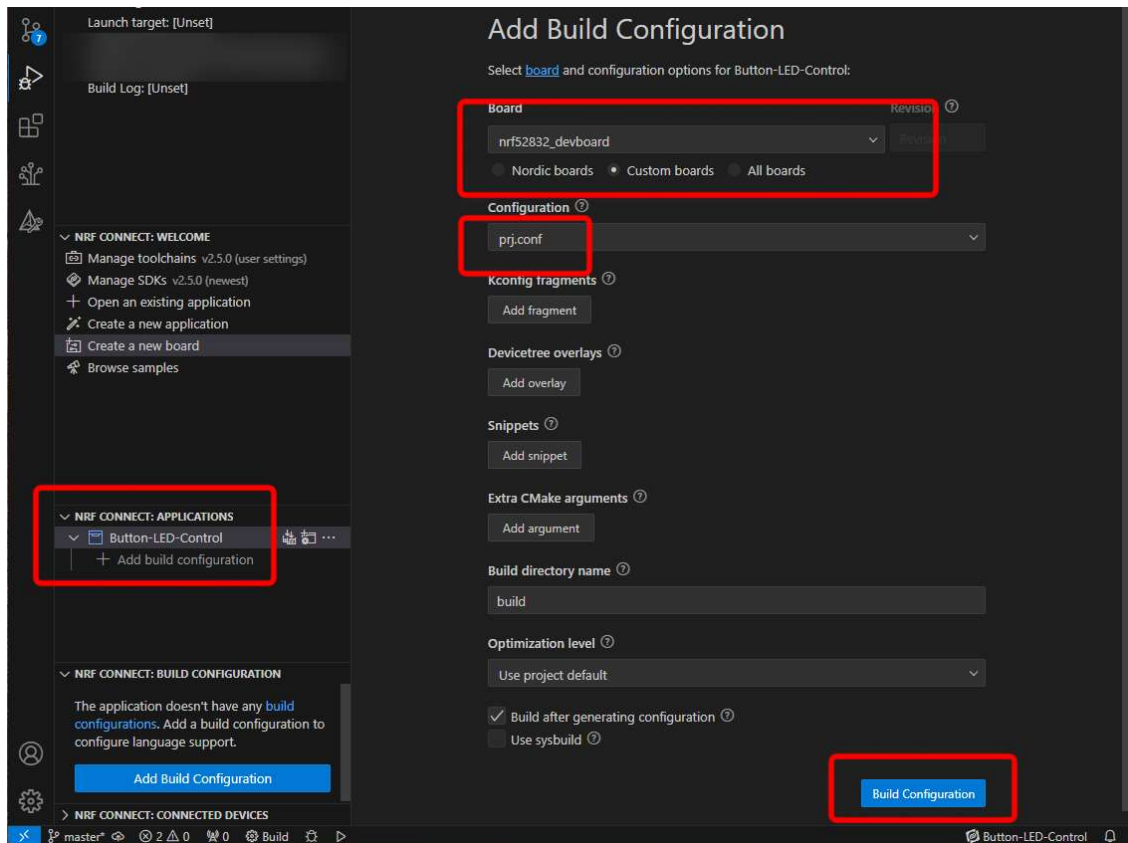
Roverdyn 의 개발 보드는 Nrf52810-QFAA 및 Nrf52832-QFAA 가 내장되어 있으므로 검색하여 선택합니다.

3. 개발 보드의 제조사명을 작성합니다. 자유롭게 설정할 수 있습니다.



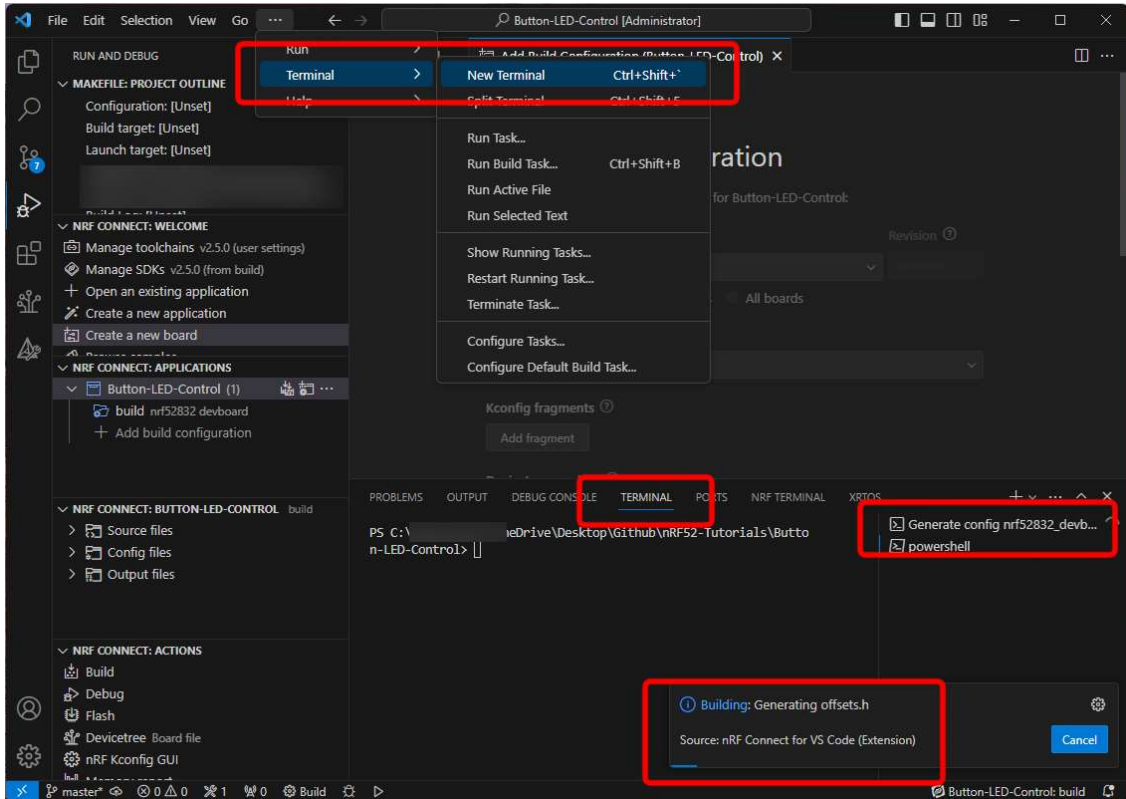
이 절차는 생략 될 수 있으며 이 경우 **[Enter]** 키를 눌러 다음으로 진행합니다.

- 다음으로 빌드 환경을 생성합니다. 빌드 환경 생성전에 반드시 1~3 번 항목을 마무리하여 빌드할 대상 보드를 생성해주어야 합니다. [NRF CONNECT: APPLICATION] 패널을 확장한 후 [Add build configuration] 메뉴를 눌러 설정 창을 열어줍니다.



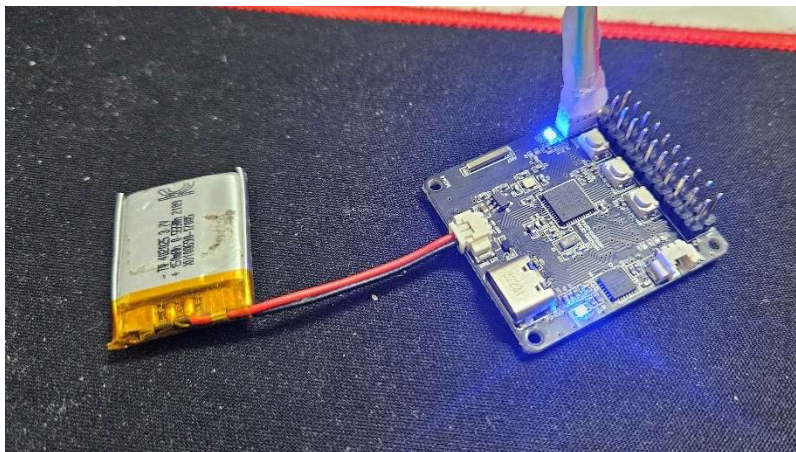
설정 창에서 [Custom board] 버튼을 눌러준 후 1~3 번 항목에서 생성한 보드를 선택합니다. 그리고 [Configuration] 항목에서 "prj.conf"를 선택한 다음 우측 하단의 [Build Configuration] 버튼을 눌러 빌드를 시작합니다.

- 빌드가 시작되면 우측 하단에 빌드 진행 패널이 나타납니다. 자세한 진행 사항을 보려면 상단 메뉴 중 [Terminal]-[Terminal]-[New Terminal] 항목을 눌러줍니다.



새 터미널 창이 생성되면 우측 하단 항목 중 [Generate config *****] 항목을 눌러 상세한 빌드 사항을 알 수 있습니다.

- 빌드가 완료되면 J-Link 디버거를 컴퓨터와 연결한 후 개발 보드에 전원을 인가하고 디버거와 개발 보드를 SWD 인터페이스를 이용하여 연결합니다.



2.6. 샘플 프로젝트 : 버튼 및 LED 다루기

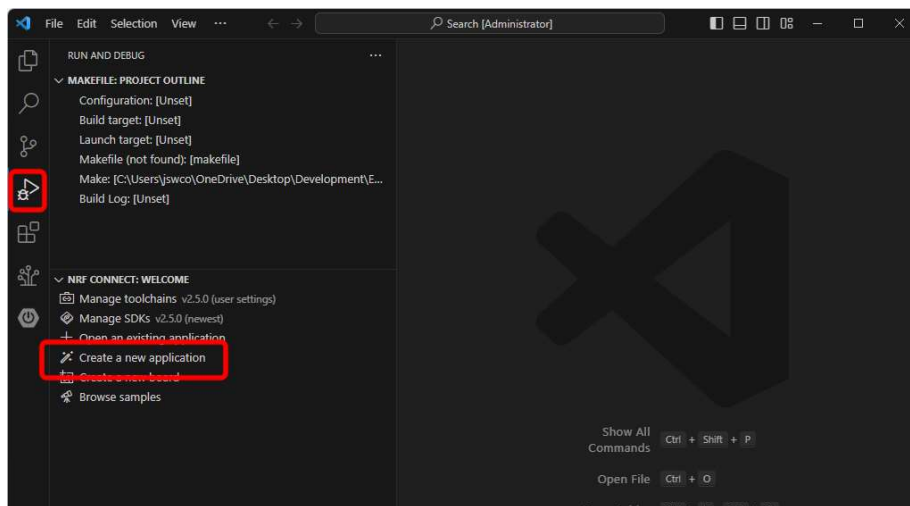
■ 개요



본 개발 키트는 2 개의 사용자 버튼과 1 개의 리셋 버튼을 포함하고 있습니다. 이 예제는 두 개의 사용자 버튼을 누르면 LED 의 색상이 변화하도록 동작하는 펌웨어입니다. 버튼 2 를 누르면 LED 의 색상이 Red, Green, Blue 순서대로 켜지며 버튼 1 을 누르면 LED 의 색상이 Blue, Green, Red 순서대로 켜집니다.

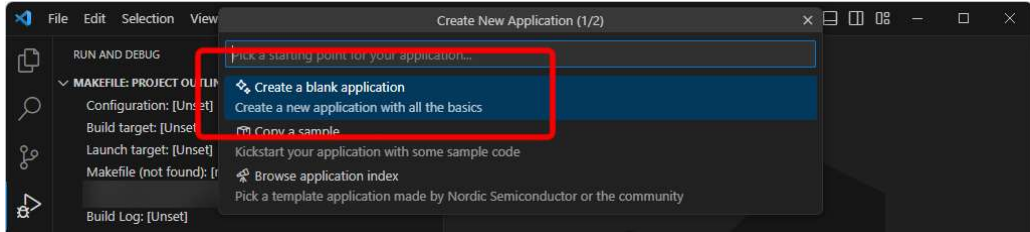
■ 프로젝트 생성

5. 프로젝트 생성을 위해 Visual Studio Code 를 실행시킵니다. 이 때, 개발 환경이 구축되어 있지 않다면 위의 [2.4] 항목을 참조하여 기본 개발 환경을 구축하여야 합니다.



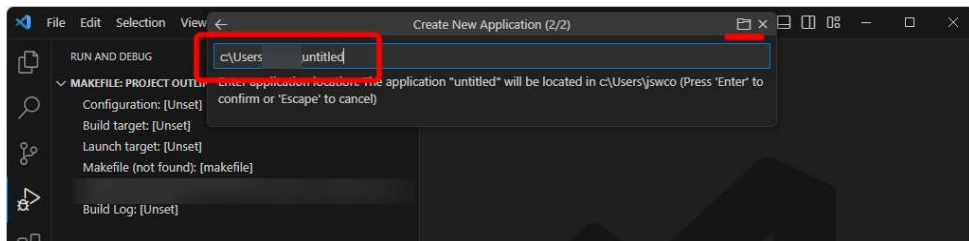
개발 환경이 구축되었다면 왼쪽의 [Run and Debug] 아이콘을 누른 후 [NRF CONNECT: WELCOME] – [Create a new application] 항목을 눌러 새로운 프로젝트를 생성합니다.

6. 상단의 “Create New Application” 패널이 나타나는지 확인합니다.

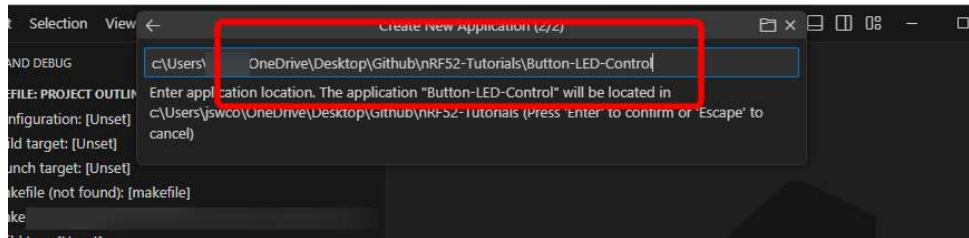


해당 패널에서 [Create a blank application] 메뉴를 선택하여 빈 애플리케이션을 생성합니다.

7. 프로젝트 폴더가 위치할 경로를 작성합니다. 프로젝트 폴더의 경로명은 가능한 한글 없이 영문만 사용하는 것을 권장합니다.

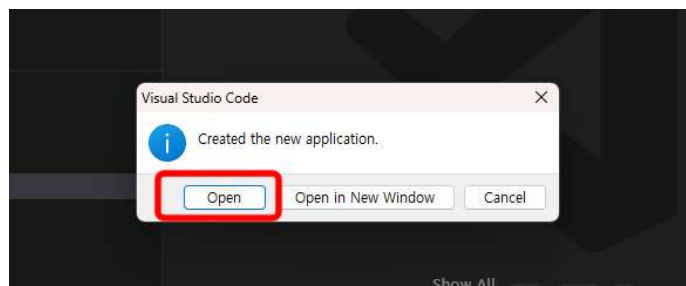


프로젝트 경로는 직접 입력이 가능하지만, Windows 탐색기를 이용하여 설정할 수 있습니다. 우측 상단의 폴더 모양의 아이콘을 클릭하면 탐색기를 이용하여 경로 설정을 할 수 있습니다.



탐색기를 이용하여 경로를 설정할 때에는 폴더는 해당 탐색기에서 별도로 생성해주어야 합니다.

8. 프로젝트가 생성되면 [Open] 버튼을 눌러 프로젝트 페이지를 열어줍니다.



■ 예제 코드

```

/*
MIT License

Copyright (c) 2024. Seonguk Jeong, Roverdyn Inc.

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.
*/
#include <zephyr/kernel.h>

/* Button pin no. definition */
#define USER_BTN1      19
#define USER_BTN2      20

/* LED pin no. definition */
#define LED_RED        22
#define LED_GREEN      23
#define LED_BLUE       24

/* Define Event Listener */
void GPIOTE_IRQHandler();

/* Global Variables */
static uint8_t led_order = 0;

int main(void)
{
    // Debug message for start
    printk("Application start.\n");

    // Set gpio direction for LEDs & turn off
    NRF_GPIO->DIRSET = (1 << LED_RED) | (1 << LED_GREEN) | (1 << LED_BLUE);
    NRF_GPIO->OUTSET = (1 << LED_RED) | (1 << LED_GREEN) | (1 << LED_BLUE);

    // Set Button GPIO as Input
    NRF_GPIO->PIN_CNF[USER_BTN1] = 0x0C;
    NRF_GPIO->PIN_CNF[USER_BTN2] = 0x0C;

    // Set Interrupt Event for buttons
    NRF_GPIOTE->INTENSET = 0x01; // Enable EVENT_IN[0]
    NRF_GPIOTE->INTENSET = (0x01 << 4); // Enable EVENT_IN[4]

    // Set GPIOTE for user button 1
    NRF_GPIOTE->CONFIG[0] = 0x01; // Set Event mode for EVENT_IN[0]
    NRF_GPIOTE->CONFIG[0] |= (USER_BTN1 << 8); // Set pin no. for EVENT_IN[0]
}

```



```

NRF_GPIOTE->CONFIG[0] |= (0x02 << 16);           // Set HiToLo polarity for
EVENT_IN[0]

// Set GPIOTE for user button 2
NRF_GPIOTE->CONFIG[4] = 0x01;                     // Set Event mode for EVENT_IN[4]
NRF_GPIOTE->CONFIG[4] |= (USER_BTN2 << 8);        // Set pin no. for EVENT_IN[4]
NRF_GPIOTE->CONFIG[4] |= (0x02 << 16);           // Set HiToLo polarity for
EVENT_IN[4]

// Enable IRQHandler for interrupt
IRQ_DIRECT_CONNECT(GPIOTE_IRQn, 0, GPIOTE_IRQHandler, 0);
irq_enable(GPIOTE_IRQn);

// Main loop
while(true){
    k_usleep(1);
}
return 0;
}

void GPIOTE_IRQHandler(){
// Interrupt event for button 1
if(NRF_GPIOTE->EVENTS_IN[0] == 1){
    // Button 1 : Shift color to left
    NRF_GPIOTE->EVENTS_IN[0] = 0;
    if(led_order == 0){
        led_order = 2;
    }else{
        led_order--;
    }

// Turn off all LEDs
NRF_GPIO->OUTSET = (1 << LED_RED) | (1 << LED_GREEN) | (1 << LED_BLUE);
switch(led_order){
    case 0:
        printk("Button 1 pressed : LED Red\n");
        NRF_GPIO->OUTCLR = (1 << LED_RED);
        break;
    case 1:
        printk("Button 1 pressed : LED Green\n");
        NRF_GPIO->OUTCLR = (1 << LED_GREEN);
        break;
    case 2:
        printk("Button 1 pressed : LED Blue\n");
        NRF_GPIO->OUTCLR = (1 << LED_BLUE);
        break;
}
}

// Interrupt event for button 2
if(NRF_GPIOTE->EVENTS_IN[4] == 1){
    // Button 2 : Shift color to right
    NRF_GPIOTE->EVENTS_IN[4] = 0;
    if(led_order == 2){
        led_order = 0;
    }else{
        led_order++;
    }

// Turn off all LEDs
NRF_GPIO->OUTSET = (1 << LED_RED) | (1 << LED_GREEN) | (1 << LED_BLUE);
switch(led_order){

```

```

    case 0:
        printk("Button 2 pressed : LED Red\n");
        NRF_GPIO->OUTCLR = (1 << LED_RED);

        break;
    case 1:
        printk("Button 2 pressed : LED Green\n");
        NRF_GPIO->OUTCLR = (1 << LED_GREEN);

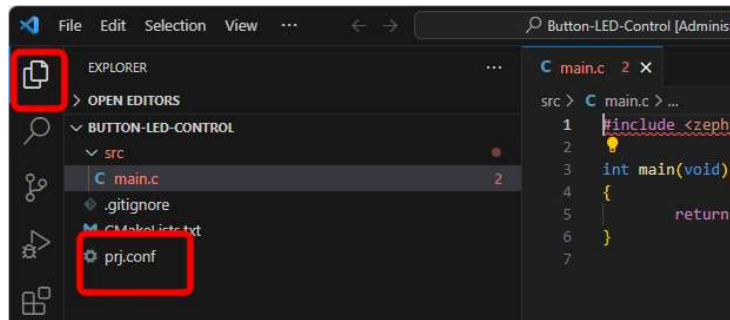
        break;
    case 2:
        printk("Button 2 pressed : LED Blue\n");
        NRF_GPIO->OUTCLR = (1 << LED_BLUE);

        break;
    }
}
}
}

```

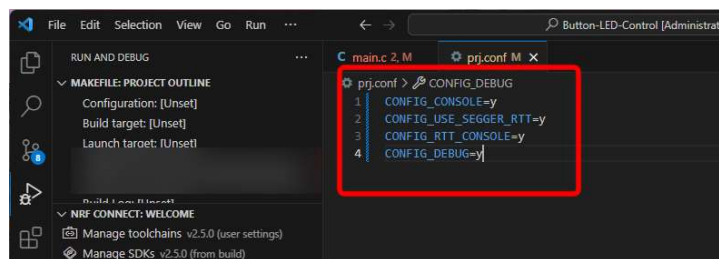
■ 프로젝트 설정

1. 애플리케이션을 실행하기 위해서 프로젝트 설정 파일(prj.conf)을 열어줍니다.



프로젝트 설정 파일은 왼쪽 상단의 [Explorer] 아이콘을 눌러 찾을 수 있습니다. prj.conf 항목을 더블클릭하여 열어줍니다.

2. 다음과 같이 설정 항목을 작성합니다.



```

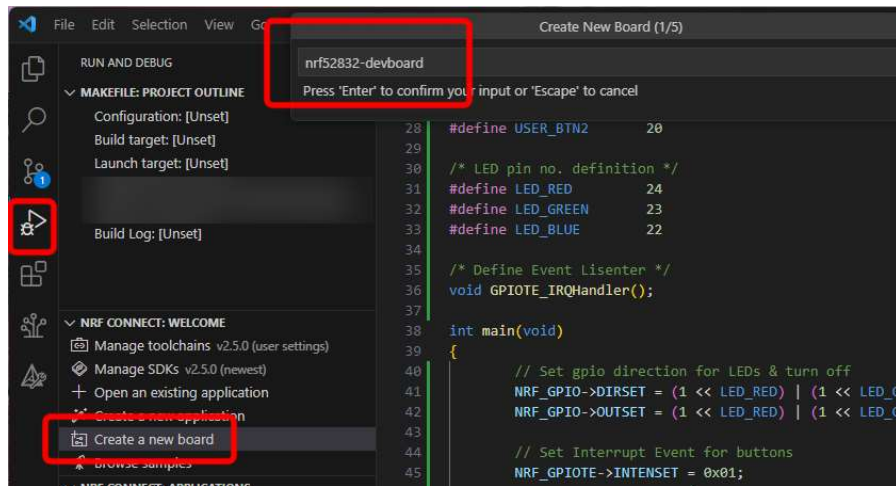
CONFIG_DEBUG=y
CONFIG_CONSOLE=y
CONFIG_USE_SEGGER_RTT=y
CONFIG_RTT_CONSOLE=y

```

■ 컴파일 및 다운로드

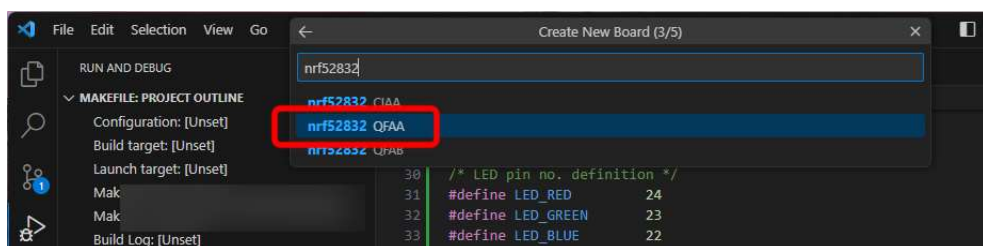
작성한 애플리케이션 코드는 컴파일 과정을 통하여 개발 보드로 다운로드 할 수 있는 포맷(*.elf 혹은 *.hex) 으로 변환됩니다. ARM Cortex-M 의 바이너리 파일(hex)은 Intel Hex 포맷으로 구성되어 있습니다. 다음 절차를 따라 작성한 코드를 컴파일 및 다운로드 할 수 있습니다.

9. 좌측 아이콘 중 [Run and Debug] 항목을 누른 다음 [NRF CONNECT: WELCOME] – [Create a new board] 메뉴를 눌러줍니다.



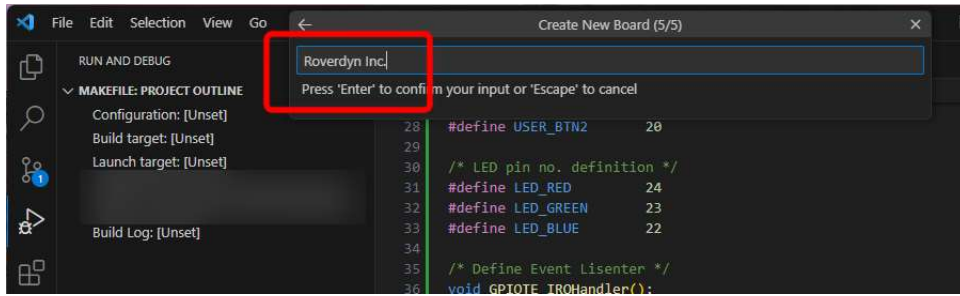
새 보드 생성 창이 나타나면 개발 보드의 이름을 설정합니다. 개발보드의 이름을 설정한 후 [Enter] 키를 눌러 다음 항목으로 진행합니다. 다음 항목은 컴퓨터가 읽을 수 있는 형태의 이름을 지정하는데, 별도로 설정하지 말고 다시 [Enter]를 눌러 진행합니다.

10. 가지고 있는 개발 보드의 MCU 타입을 선택합니다.



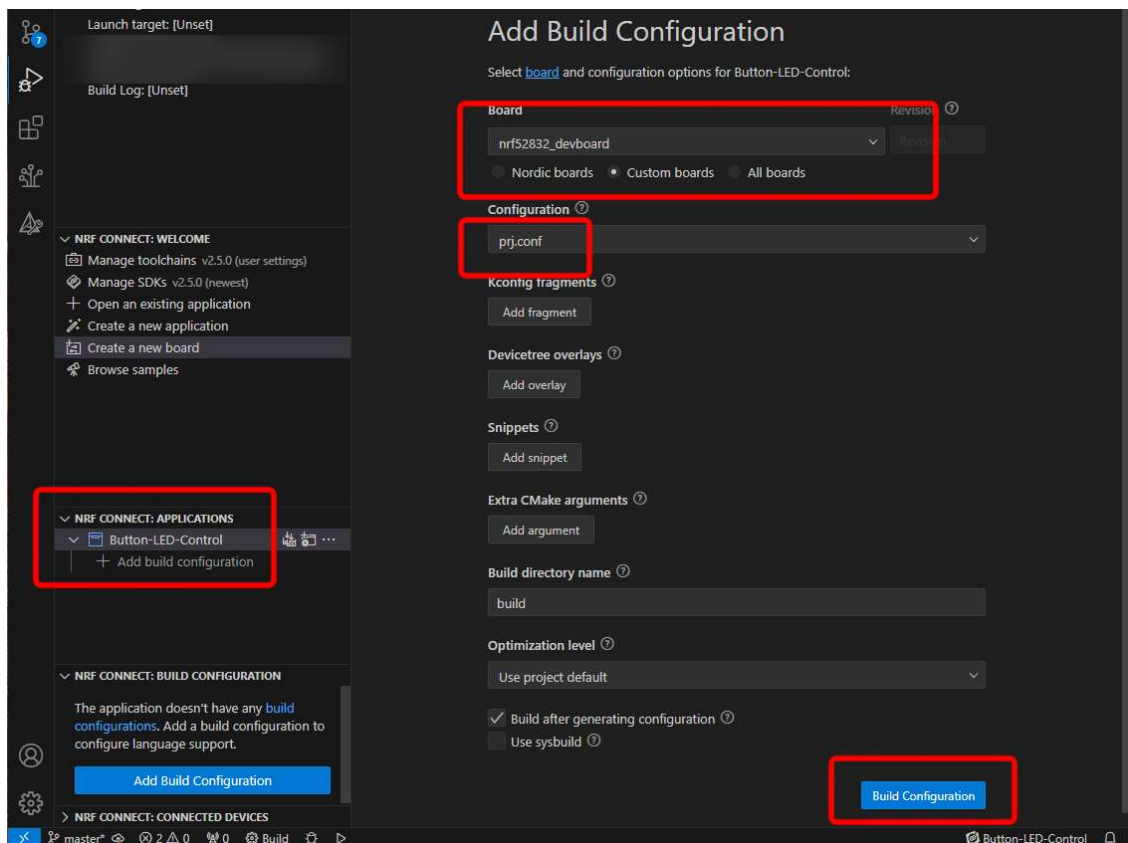
Roverdyn 의 개발 보드는 Nrf52810-QFAA 및 Nrf52832-QFAA 가 내장되어 있으므로 검색하여 선택합니다.

11. 개발 보드의 제조사명을 작성합니다. 자유롭게 설정할 수 있습니다.



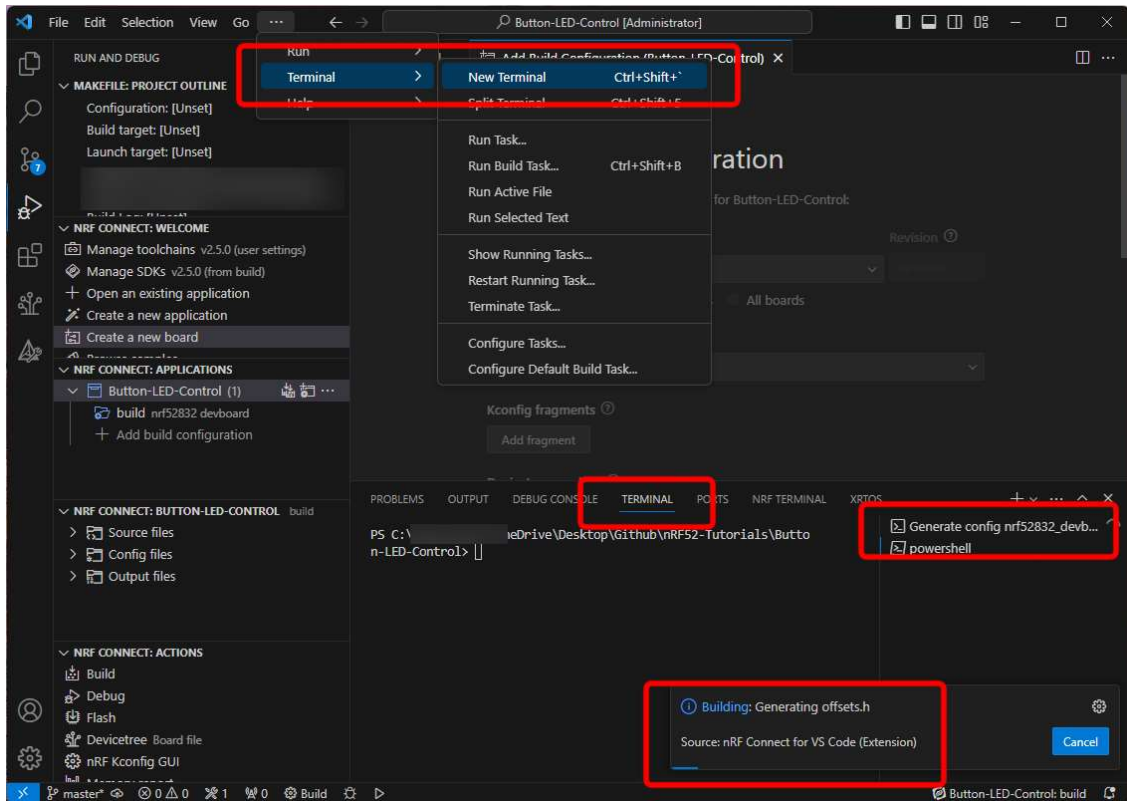
이 절차는 생략 될 수 있으며 이 경우 **[Enter]** 키를 눌러 다음으로 진행합니다.

12. 다음으로 빌드 환경을 생성합니다. 빌드 환경 생성전에 반드시 1~3 번 항목을 마무리하여 빌드할 대상 보드를 생성해주어야 합니다. [NRF CONNECT: APPLICATION] 패널을 확장한 후 [Add build configuration] 메뉴를 눌러 설정 창을 열어줍니다.



설정 창에서 [Custom board] 버튼을 눌러준 후 1~3 번 항목에서 생성한 보드를 선택합니다. 그리고 [Configuration] 항목에서 "prj.conf"를 선택한 다음 우측 하단의 [Build Configuration] 버튼을 눌러 빌드를 시작합니다.

13. 빌드가 시작되면 우측 하단에 빌드 진행 패널이 나타납니다. 자세한 진행 사항을 보려면 상단 메뉴 중 [Terminal]-[Terminal]-[New Terminal] 항목을 눌러줍니다.

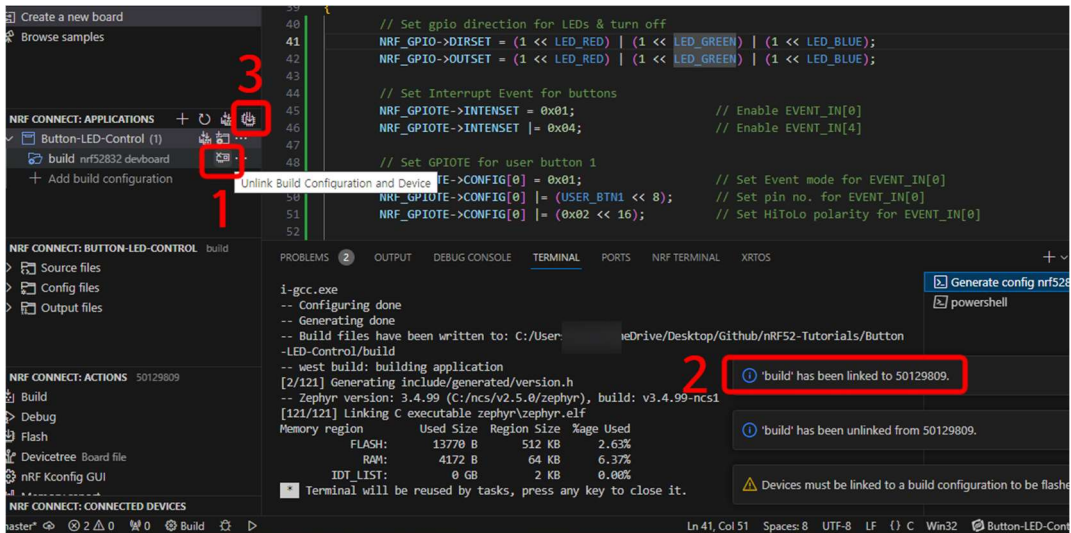


새 터미널 창이 생성되면 우측 하단 항목 중 [Generate config *****] 항목을 눌러 상세한 빌드 사항을 알 수 있습니다.

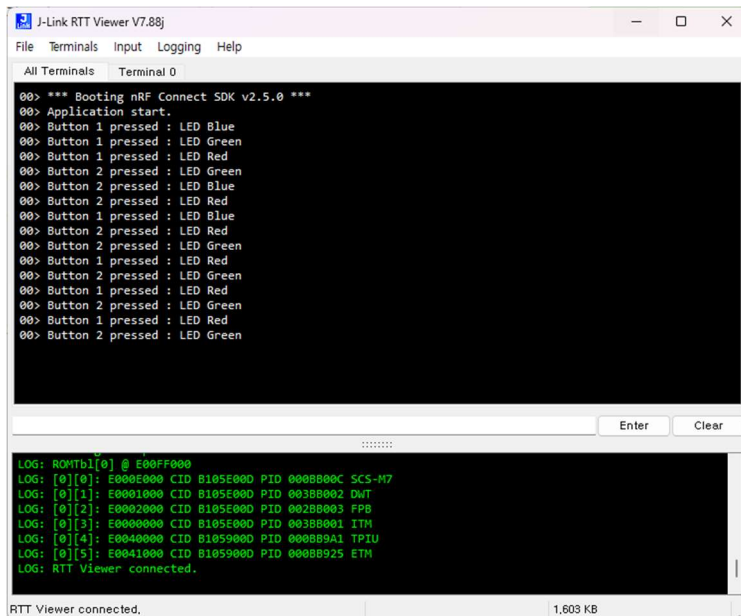
14. 빌드가 완료되면 J-Link 디버거를 컴퓨터와 연결한 후 개발 보드에 전원을 인가하고 디버거와 개발 보드를 SWD 인터페이스를 이용하여 연결합니다.



15. 컴파일이 완료된 애플리케이션을 다운로드 하기 위해 [NRF CONNECT: APPLICATIONS] 패널에서
 - 1) Build configuration 의 우측에 위치한 J-link 연결 아이콘을 눌러서 디버거를 연결한 다음
 - 2) 컴파일이 완료되었는지 확인하고 3) Flash 아이콘을 눌러 애플리케이션을 개발 보드로 다운로드합니다.

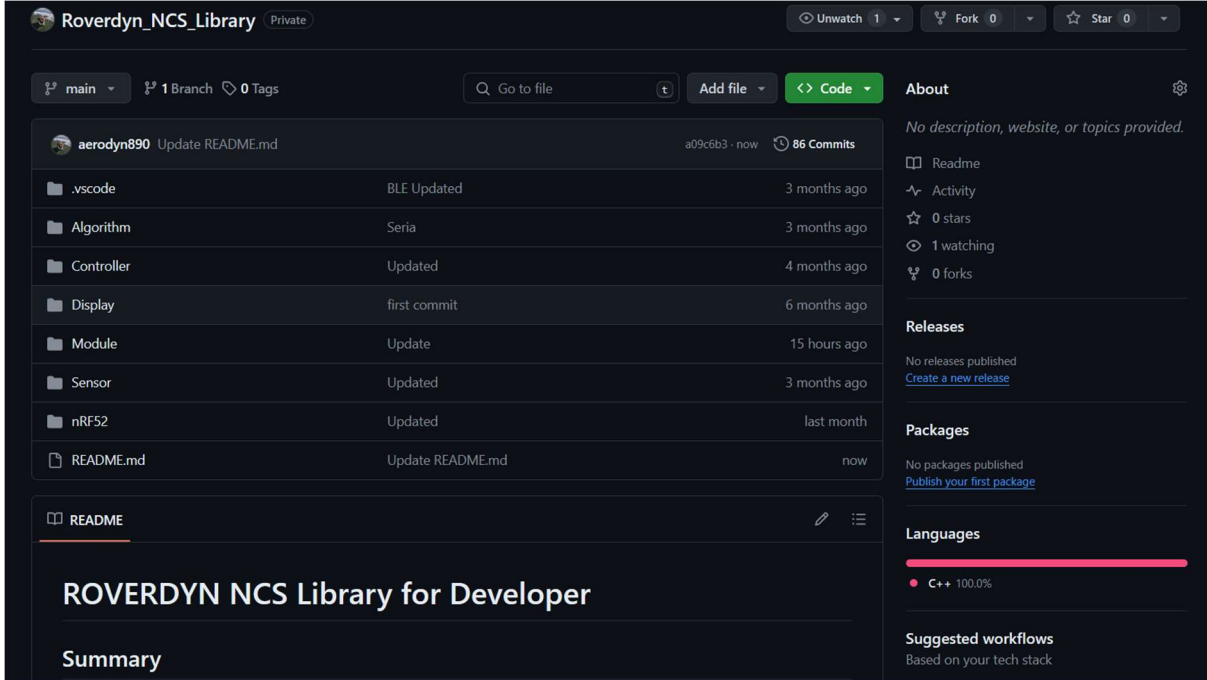


16. 다운로드가 완료된 후 리셋 버튼을 눌러 애플리케이션을 실행시킬 수 있습니다.



[J-Link RTT Viewer] 프로그램을 실행시키거나 개발 보드의 Button 1, Button 2 를 눌러 LED 의 변화를 관찰할 수 있습니다.

3 Roverdyn NCS Library



3.1. Roverdyn NCS Library 소개

■ 개요

Roverdyn NCS Library 는 Nrf52 Family MCU 를 위해 개발된 C/C++ 기반의 라이브러리입니다. Zephyr Project 를 기반으로 하며 Nordic Semiconductor 사의 SDK 를 보다 사용하기 쉽게 구성하고 정의되었으며 가능한 단순하고 적은 수의 매개 변수를 입력 받아서 동작하도록 디자인되었습니다. 사용자는 단순히 라이브러리 헤더를 Include 하여 클래스를 정의한 후 바로 활용할 수 있습니다.

■ 다운로드 및 사용

Roverdyn NCS Library 는 Github 및 로버다인의 홈페이지에서 제공하고 있습니다. 라이브러리의 사용 및 수정, 재배포는 가능하나 원 저자를 꼭 밝혀야 합니다.

- Github Repository : https://github.com/aerodyn890/Roverdyn_NCS_Library.git
- Roverdyn Web : https://roverdyn.com/?page_id=27598

■ 라이선스

Roverdyn NCS Library 는 자유로운 개발을 위해 MIT 라이선스를 사용합니다. 이 라이브러리에 적용되는 MIT 라이선스의 원문은 다음과 같습니다.

<p>English</p>	<p>MIT License</p> <p>Copyright (c) 2024. Seonguk Jeong, Roverdyn Inc.</p> <p>Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:</p> <p>The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.</p> <p>THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.</p>
<p>한국어</p>	<p>MIT 라이선스</p> <p>Copyright (c) 2024. 로버다인 정성욱.</p> <p>이에 따라 사본을 얻는 모든 사람에게 무료로 허가가 부여됩니다. 본 소프트웨어 및 관련 문서 파일("소프트웨어")의 취급 권리를 제한 없이 포함하는 소프트웨어에서 사용, 복사, 수정, 병합, 게시, 배포, 하위 라이선스 및/또는 판매 소프트웨어의 사본 및 소프트웨어가 있는 사람을 허용합니다. 다음 조건에 따라 그렇게 하도록 제공됩니다:</p> <p>상기 저작권 고지 및 본 허가 고지는 본 소프트웨어의 모든 사본 또는 상당 부분에 포함되어야 합니다.</p> <p>본 소프트웨어는 상품성, 특정 목적에 대한 적합성 및 비침해를 포함하지만 이에 제한되지 않는 명시적 또는 묵시적 보증 없이 "있는 그대로" 제공됩니다. 어떠한 경우에도 저작자 또는 저작권자는 본 소프트웨어 또는 본 소프트웨어의 사용 또는 기타 거래로 인해 발생한 계약, 불법 행위 또는 기타 모든 청구, 손해 또는 기타 책임에 대해 책임을 지지 않습니다.</p>

본 사용자 매뉴얼의 저작권에 대한 모든 권리는 로버다인(Roverdyn)에 있으며 로버다인의 허가 없이 복제, 기록, 재배포 및 번역할 수 없습니다.
All rights to copyright in this User's Manual are in Roverdyn and may not be reproduced, recorded, redistributed or translated without Roverdyn's permission.

Technical omnipotence

Roverdyn Inc.

- General Development Board
- Navigation & Guidance Development Board
- Flight Control Computer Development Board
- ARS & AHRS with GNSS
- Android & Web applications

Web : <https://www.roverdyn.com>

Tel : +82-070-7808-9803

E-mail : support@roverdyn.com

(14348) 72, Iljik-ro, Gwangmyeong-si, Gyeonggi-do, Republic of Korea