

# TechDocs

**NCS build on WSL (25/10/20)**

---

일반적으로 개발환경을 윈도우즈에서 꾸미게 되는데, 멀티이미지를 빌드하게 될 경우 윈도우즈 시스템은 상당히 느린 환경을 제공하게 됩니다.

이를 개선하기 위해서는 우분투를 사용해야 하지만 현실적으로 윈도우 시스템을 버리기 어렵기 때문에,

마이크로소프트에서 제공하는 WSL(Windows Subsystem for Linux)이라는 경량 VM기반에서 Ubuntu를 실행하여 사용하는 방법을 안내합니다.

---

## 목차

### 빠른 설치 절차

#### Installation Guide

- Install WSL
- Create the required library link
- Run VSCode on WSL

#### Configuration for nRF Connect SDK

- Create nrfutil script

#### Configuration of VScode

- Configuration of Ozone

#### nRF Connect SDK Installation

- VSCode에 설치하기
  - 툴체인 설치
  - SDK (nRF Connect SDK ) 설치
- Toolchain Manager를 통한 설치 (NCS v2.xx 까지만 사용 가능)

#### Performance Test Result

- Result

#### Tips

- Command line tools 설치
- WSL 디스크 공간 관리 방법
- VIM
- Install UI packages
- WSL vhdx 파일 위치 확인하기
- WSL vhdx 위치 변경하기
- Windows 폴더 접근 방법

#### References



# 빠른 설치 절차

Windows DOS terminal or Powershell (여기서는 Ubuntu-24.04 버전으로 진행합니다)

```
wsl --install -d Ubuntu-24.04
wsl --set-default Ubuntu-24.04
wsl
```

- < For Powershell >

```
[Environment]::SetEnvironmentVariable(
    "Path",
    $env:Path + ";$env:USERPROFILE\nrfutil\bin",
    "User"
)

$src = "$env:USERPROFILE\nrfutil\bin\nrfutil-device\plugin-probe-worker.exe"
$dst = "$env:USERPROFILE\nrfutil\bin\plugin-probe-worker.exe"

if (Test-Path $src) {
    Copy-Item -Force $src -Destination $dst
    Write-Host "plugin-probe-worker.exe has been copied to the bin folder."
} else {
    Write-Host "Source file not found: $src"
}
```

- < For DOS terminal >

```
setx PATH "%PATH%;%USERPROFILE%\nrfutil\bin"
set SRC=%USERPROFILE%\nrfutil\bin\nrfutil-device\plugin-probe-worker.exe

set DST=%USERPROFILE%\nrfutil\bin\plugin-probe-worker.exe
if exist "%SRC%" (
    copy /Y "%SRC%" "%DST%"
    echo plugin-probe-worker.exe has been copied to the bin folder.
) else (
    echo Source file not found: %SRC%
)
```

WSL (Ubuntu terminal)



```
ln -s /lib/x86_64-linux-gnu/libunistring.so.5 /lib/libunistring.so.2
sudo apt-get update
sudo apt-get install wget ca-certificates
mkdir -p ~/.local/bin
echo '#!/bin/bash' > ~/.local/bin/nrfutil ; echo "nrfutil.exe \$@" >> ~/.local/bin/nrfutil
echo '#!/bin/bash' > ~/.local/bin/nrfutil-device ; echo "nrfutil-device.exe \$@" >> ~/.local/bin/nrfutil-device
chmod a+x ~/.local/bin/nrfutil
chmod a+x ~/.local/bin/nrfutil-device
```



```
mv ~/.vscode-server/extensions/nordic-semiconductor.nrf-connect-2025.9.798-linux-x64/platform/nrfutil/bin/nrfutil-device ~/.vscode-server/extensions/nordic-semiconductor.nrf-connect-2025.9.798-linux-x64/platform/nrfutil/bin/nrfutil-device.org
ln -s ~/.local/bin/nrfutil-device ~/.vscode-server/extensions/nordic-semiconductor.nrf-connect-2025.9.798-linux-x64/platform/nrfutil/bin/nrfutil-device
```

VSCode (WSL-Ubuntu terminal)

1. VSCode의 좌측 아래 녹색 아이콘을 클릭한 후 중앙 상단에 뜨는 리스트 중 **WSL connect** 을 선택
2. **Ctrl+Shift+P** 를 눌러 **settings.json** 입력 후 'Open Remote Settings (WSL)'를 선택 후 아래 내용과 같이 중괄호 내 2줄을 삽입



```
{
  "nrf-connect.ozonPath": "/mnt/c/Program Files/SEGGER/Ozone/Ozone.exe",
}
```

# Installation Guide

## Install WSL

1. Windows의 DOS Command terminal or Powershell 실행
2. `wsl --list --online` 을 입력하여 설치 가능한 배포판 목록을 확인
3. `wsl --install -d <Distro>` 명령을 통해서 설치하고자 하는 배포판을 설치
  - ex) `wsl --install -d Ubuntu-24.04`
4. 설치가 완료되면 PC 재부팅
5. 설치가 완료되면 Powershell 또는 DOS Terminal에서 `wsl` 을 입력하여 설치된 배포판을 구동

여러개의 배포판이 설치되어 있다면 `wsl -l -v` 명령을 통해서 설치된 배포판 목록을 확인하고 `wsl --set-version <distro name> 2` 명령을 통해서 `wsl` 명령 수행 시 기본으로 수행된 배포판을 지정할 수 있습니다.

## Create the required library link




```
sudo ln -s /lib/x86_64-linux-gnu/libunistring.so.5 /lib/libunistring.so.2
```

## Run VSCode on WSL

먼저 Windows에 먼저 VSCode가 설치되어 있어야 합니다.

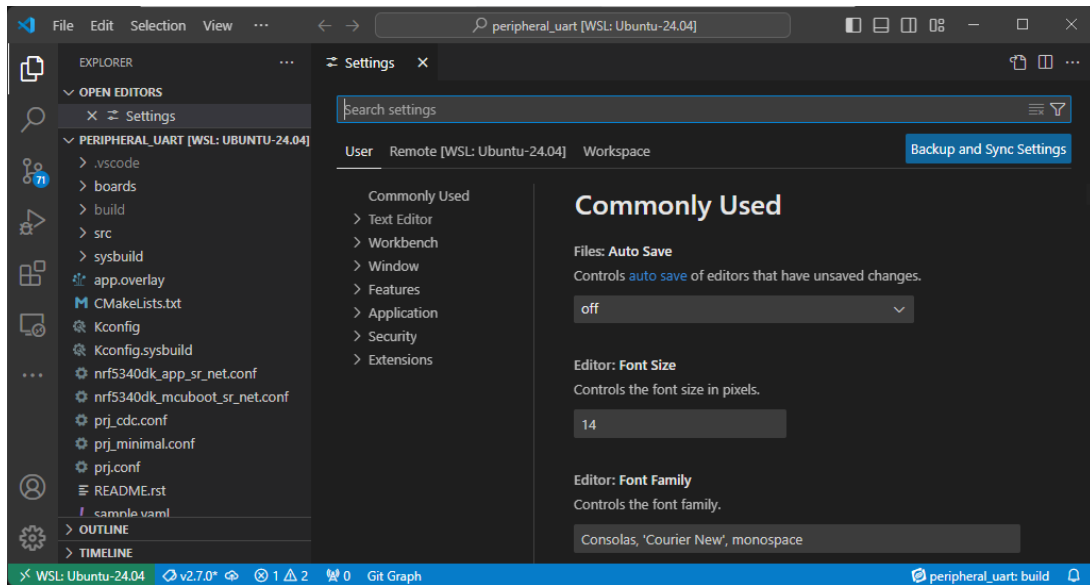
1. [Remote Development extension pack](#) 설치
2. 설치된 리눅스(우분투)를 업데이트



```
sudo apt-get update  
sudo apt-get install wget ca-certificates
```

3. `code .` 을 터미널에서 입력하여 VSCode 실행

4. 윈도우에서 바로 VSCode를 실행할 경우 WSL용 Workspace를 선택하거나, 화면 좌측 아래의 초록색 아이콘을 클릭한 후 "Connect WSL"을 선택하여 Remote Server인 WSL에 연결



# Configuration for nRF Connect SDK

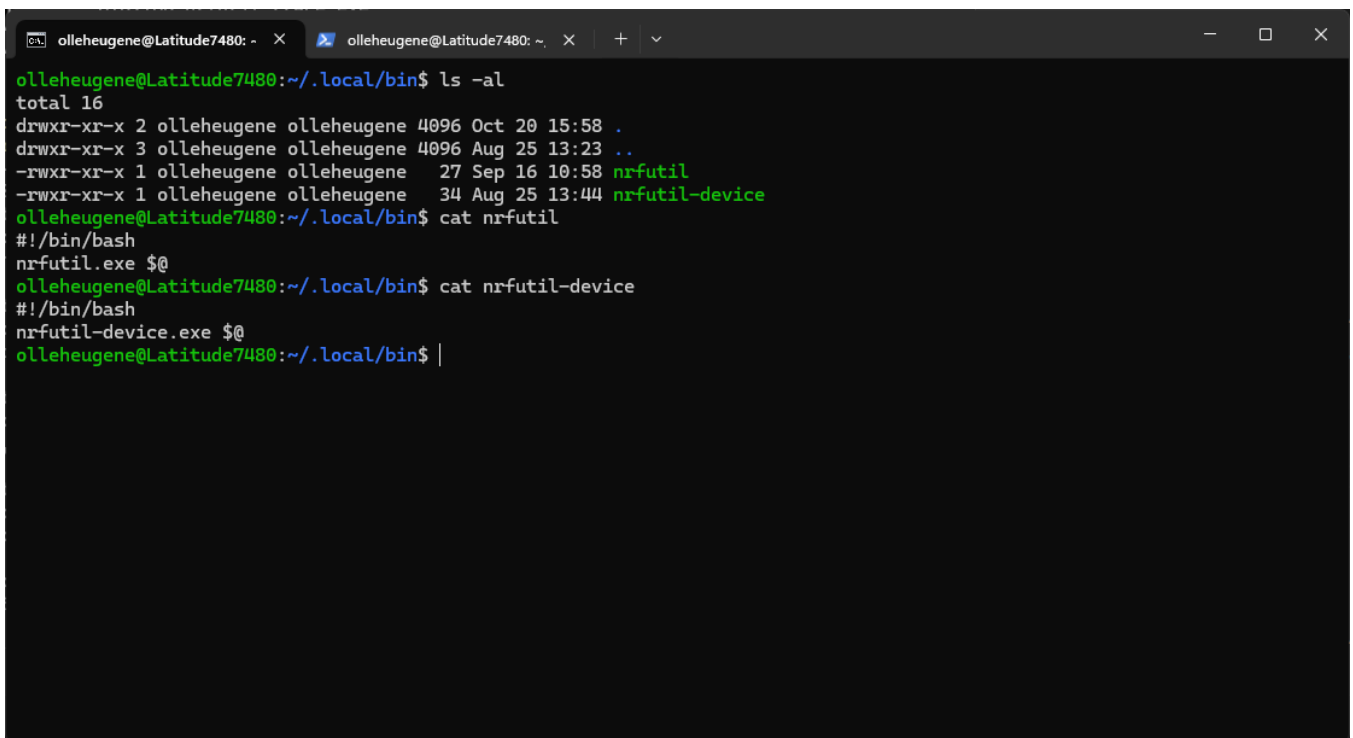
## Create nrfutil script

WSL에서 직접 COM Port, USB Port를 사용하는 것은 번거롭고 쉽지 않아 Windows에서 설치된 **nrfutil**을 직접 수행하여 사용하도록 설정 필요합니다.

이 과정은 VScode 와 west를 통해서 디바이스의 인식과 펌웨어 업데이트를 수행하기 위한 절차로서 번거롭지만 반드시 필요한 작업입니다.

### 1. 로컬에서 수행할 nrfutil 파일을 **~/local/bin/** 위치에 생성

```
mkdir -p ~/.local/bin
echo '#!/bin/bash' > ~/.local/bin/nrfutil ; echo "nrfutil.exe \"$@" >> ~/.local/bin/nrfutil
echo '#!/bin/bash' > ~/.local/bin/nrfutil-device ; echo "nrfutil-device.exe \"$@" >> ~/.local/bin/nrfutil-device
```



```
olleheugene@Latitude7480: ~$ ls -al
total 16
drwxr-xr-x 2 olleheugene olleheugene 4096 Oct 20 15:58 .
drwxr-xr-x 3 olleheugene olleheugene 4096 Aug 25 13:23 ..
-rwxr-xr-x 1 olleheugene olleheugene 27 Sep 16 10:58 nrfutil
-rwxr-xr-x 1 olleheugene olleheugene 34 Aug 25 13:44 nrfutil-device
olleheugene@Latitude7480: ~/.local/bin$ cat nrfutil
#!/bin/bash
nrfutil.exe $@
olleheugene@Latitude7480: ~/.local/bin$ cat nrfutil-device
#!/bin/bash
nrfutil-device.exe $@
olleheugene@Latitude7480: ~/.local/bin$ |
```

### 2. 리눅스 파일을 실행하기 위해서 권한 변경

```
chmod a+x ~/.local/bin/nrfutil
chmod a+x ~/.local/bin/nrfutil-device
```

### 3. VScode에서 장치 인식을 위한 파일 설정

#### 1. nrfutil-device를 지정하는 폴더 찾기

Welcome tab의 Generate Support Information (노란색 동그라미) 버튼을 눌러 jsonc 파일 생성 후 보여지는 내

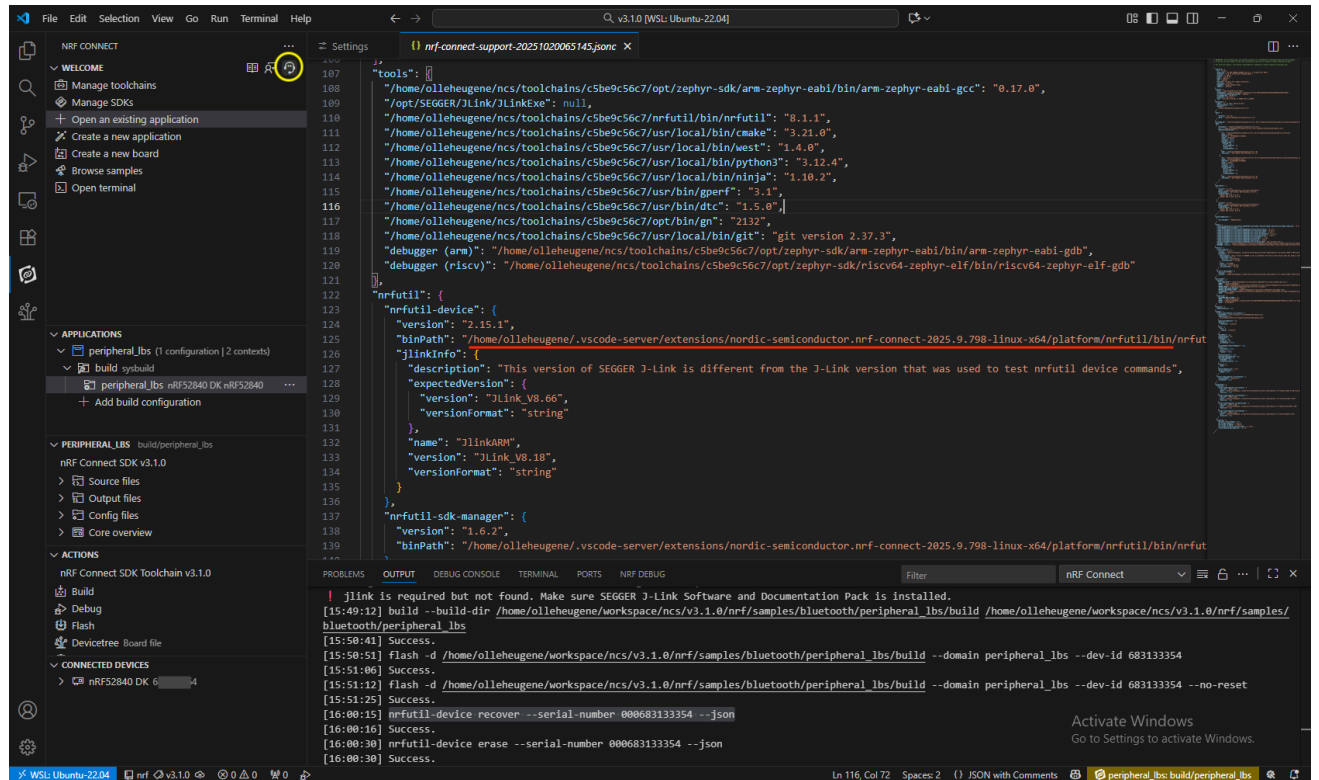


용에서 nrfutil-device의 binPath를 확인

#### Note

nrf connect extension의 버전에 따라 사용 폴더의 위치가 달라질 수 있어 버전이 업데이트 될 때마다 반복 설정 필요

/home/xxxx/.vscode\_server/extensions/nordic-semiconductor.nrf-connect-2025.9.798-linux-x64/platform/nrfutil/bin/



2. Terminal 창을 통해 해당 폴더로 이동
3. nrfutil-device 파일 삭제 또는 이름 변경 후 위에서 만들었던 ~/.local/bin/nrfutil-device 스크립트 파일을 링크



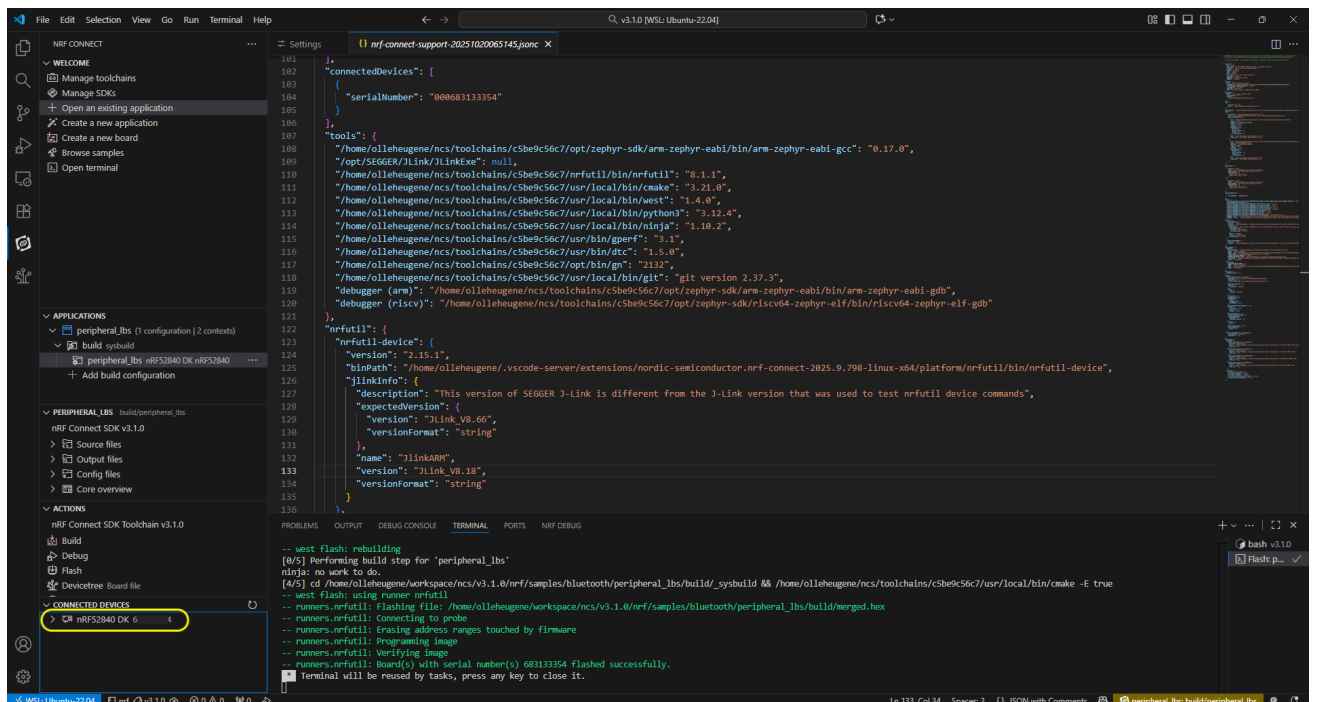
```
mv nrfutil-device nrfutil-device.org
```

```
ln -s ~/.local/bin/nrfutil
```

```
ln -s ~/.local/bin/nrfutil-device
```

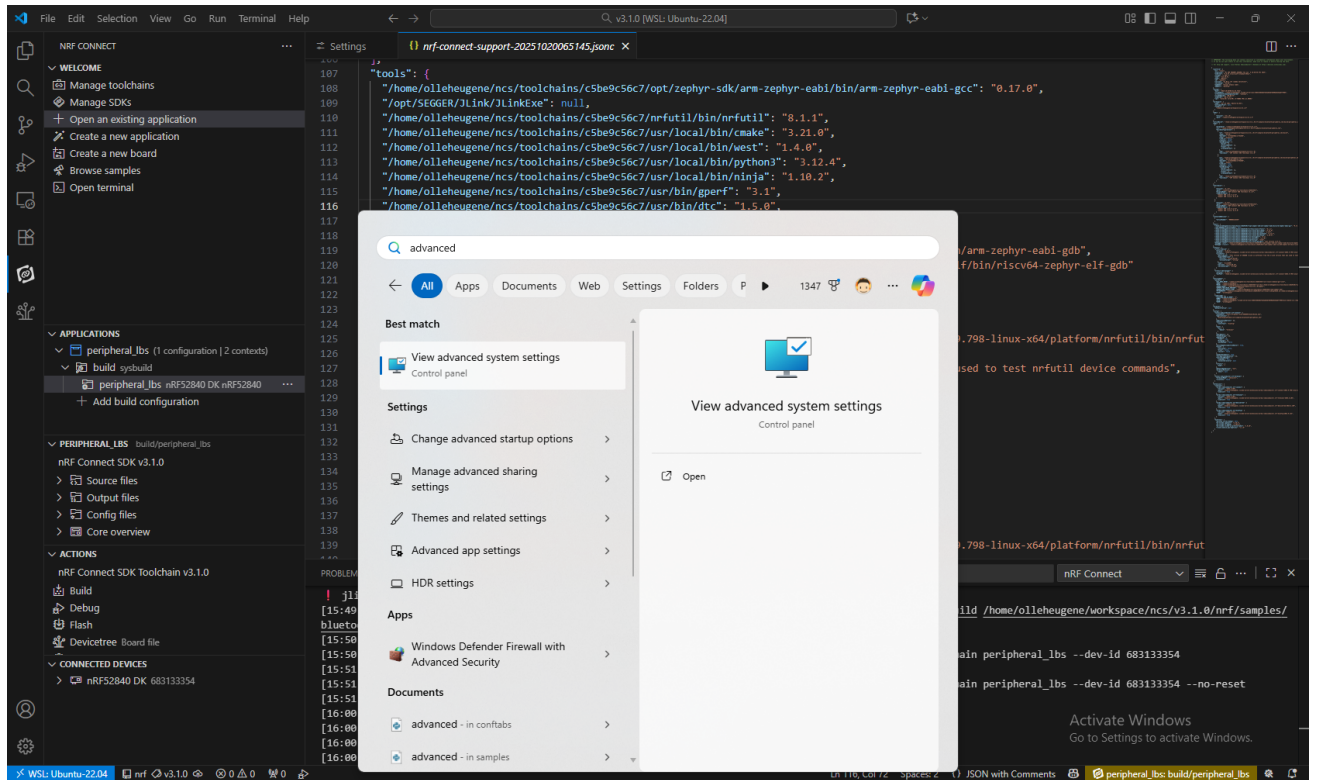
```
olleheugene@Latitude7480: ~$ cd /home/olleheugene/.vscode-server/extensions/nordic-semiconductor.nrf-connect-2025.9.798-linux-x64/platform/nrfutil/bin/
olleheugene@Latitude7480: ~/.vscode-server/extensions/nordic-semiconductor.nrf-connect-2025.9.798-linux-x64/platform/nrfutil/bin$ ls -al
total 33112
drwxr-xr-x 2 olleheugene olleheugene 4096 Oct 20 15:54 .
drwxr-xr-x 6 olleheugene olleheugene 4096 Oct 20 14:43 ..
-rwxr-xr-x 1 olleheugene olleheugene 15207224 Oct 20 14:43 nrfutil-device
-rwxr-xr-x 1 olleheugene olleheugene 18689784 Oct 20 14:43 nrfutil-sdk-manager
olleheugene@Latitude7480: ~/.vscode-server/extensions/nordic-semiconductor.nrf-connect-2025.9.798-linux-x64/platform/nrfutil/bin$ mv nrfutil-device nrfutil-device.org
olleheugene@Latitude7480: ~/.vscode-server/extensions/nordic-semiconductor.nrf-connect-2025.9.798-linux-x64/platform/nrfutil/bin$ ln -s ~/local/bin/nrfutil-device
olleheugene@Latitude7480: ~/.vscode-server/extensions/nordic-semiconductor.nrf-connect-2025.9.798-linux-x64/platform/nrfutil/bin$ ls -al
total 33112
drwxr-xr-x 2 olleheugene olleheugene 4096 Oct 20 15:55 .
drwxr-xr-x 6 olleheugene olleheugene 4096 Oct 20 14:43 ..
lrwxrwxrwx 1 olleheugene olleheugene 43 Oct 20 15:55 nrfutil-device -> /home/olleheugene/.local/bin/nrfutil-device
-rwxr-xr-x 1 olleheugene olleheugene 15207224 Oct 20 14:43 nrfutil-device.org
-rwxr-xr-x 1 olleheugene olleheugene 18689784 Oct 20 14:43 nrfutil-sdk-manager
olleheugene@Latitude7480: ~/.vscode-server/extensions/nordic-semiconductor.nrf-connect-2025.9.798-linux-x64/platform/nrfutil/bin$ |
```

4. VSCode 를 재시작하면 CONNECTED DEVICE 탭에 현재 연결된 J-link 또는 DK의 Serial Number가 출력됩니다.

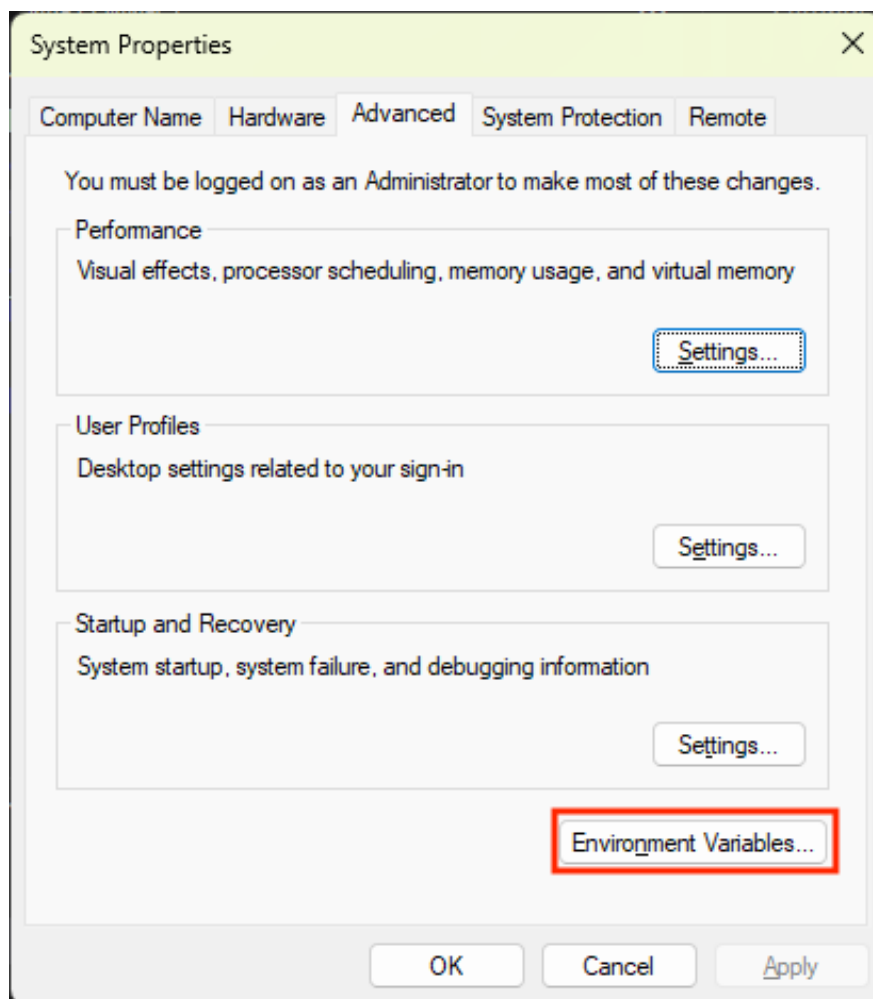


## 4. Windows Path 설정

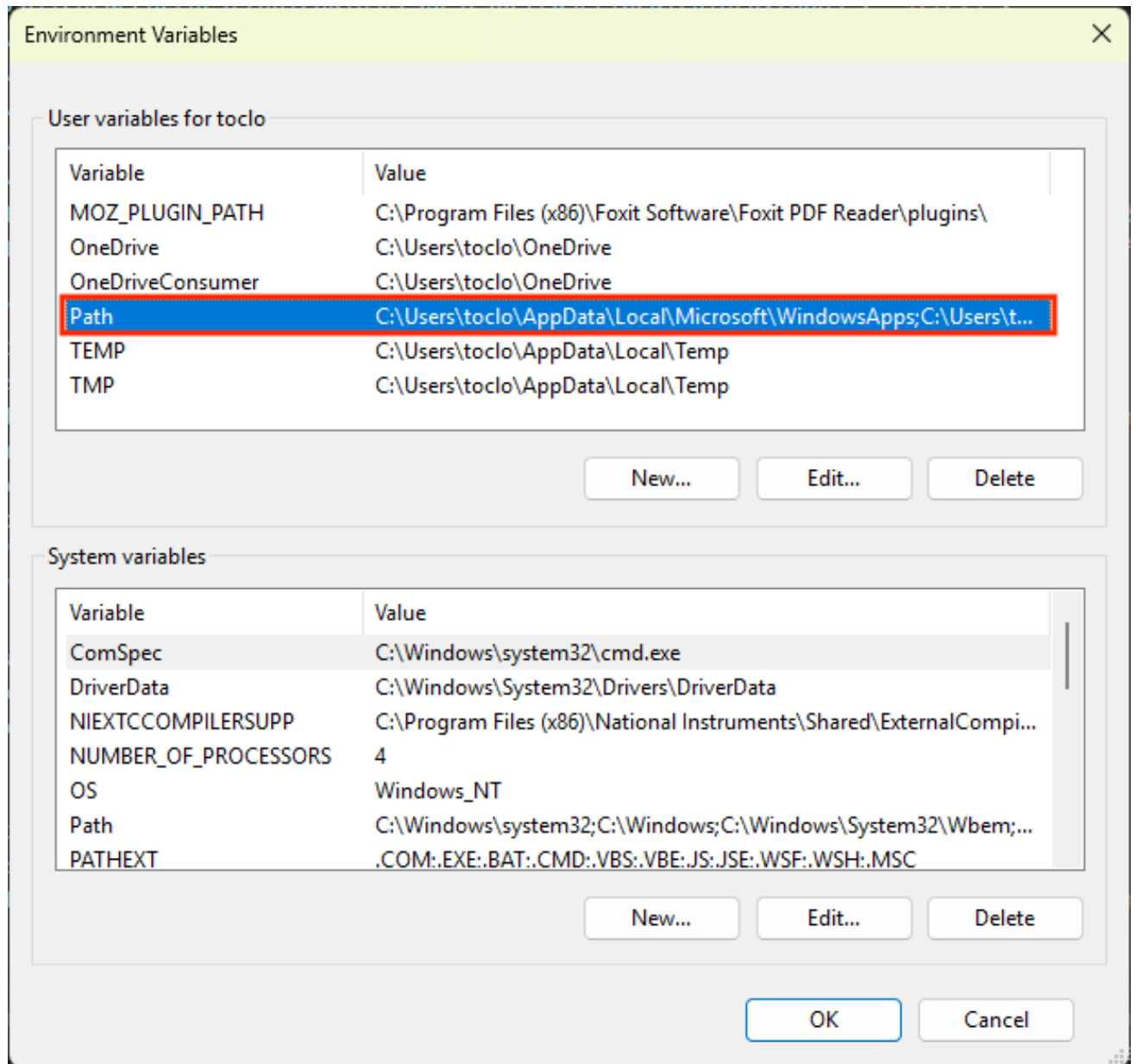
1. Windows 버튼을 누른 후 입력창에 View advanced system settings 선택



## 2. Environment Variables... 버튼 선택



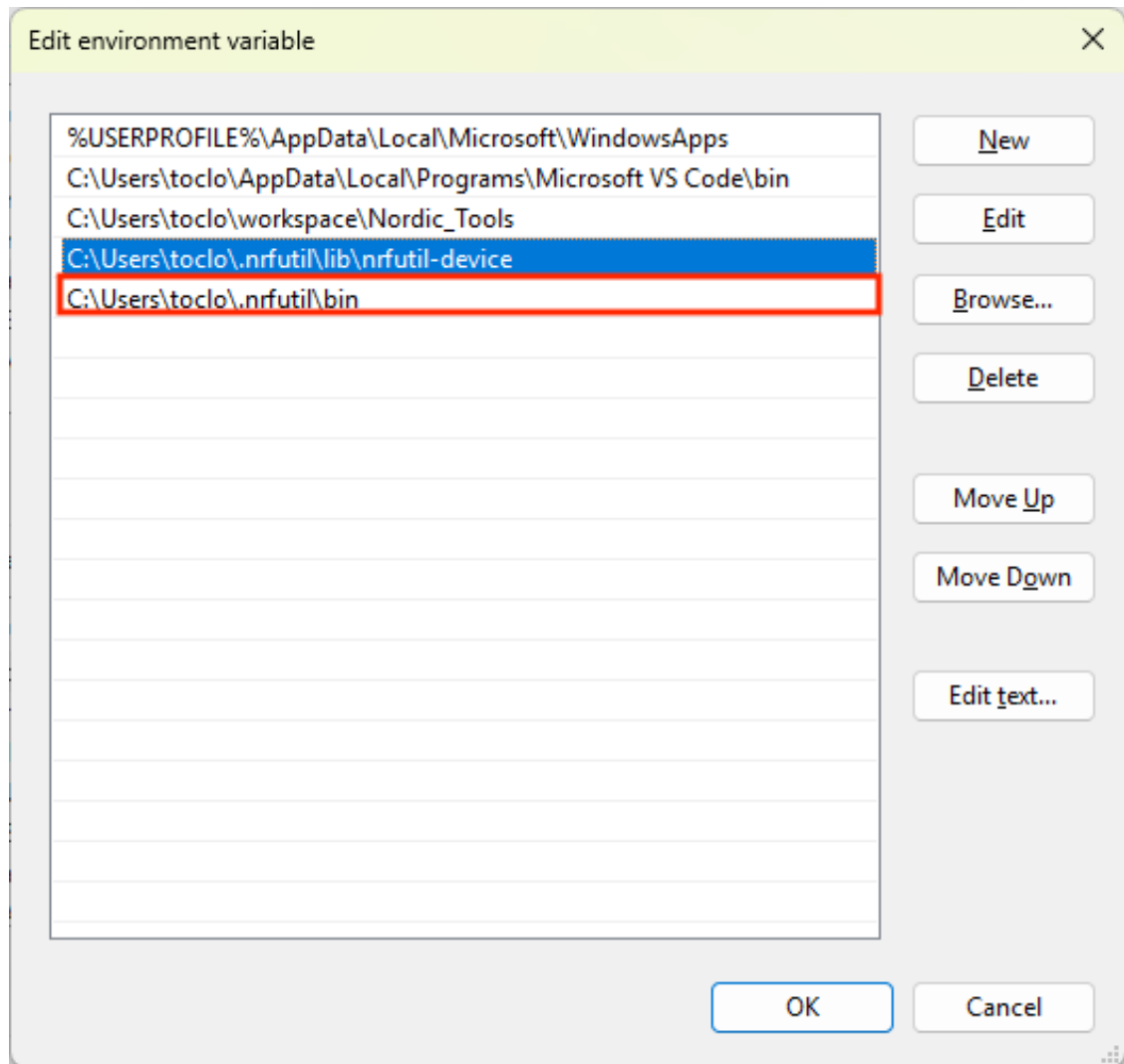
## 3. Path 항목을 더블 클릭 또는 Edit 버튼을 누름



4. nrfutil의 bin폴더의 파일들을 사용자 계정 어디에서든 이용 가능하도록 path 에 추가

**Note**

nrfutil이 설치되는 위치는 주로 "C:\Users\사용자계정\nrfutil"에 설치되면 실행파일은 "C:\Users\사용자계정\nrfutil\bin"에 설치



5. plugin-probe-worker.exe 복사

C:\Users\사용자계정\nrfutil\lib\nrfutil-device\plugin-probe-worker.exe 를 C:\Users\사용자계정\nrfutil\bin으로 복사

# Configuration of VScode

## Debug 설정

Debug 설정은 nrfutil script를 설정하는 것과 같은 방법으로 Toolchain의 파일들을 생성하고 링크해야 합니다.

이와 같은 설정이 번거롭고 복잡할 경우 Ozone을 아래와 같은 방법으로 설정 및 대체 툴로 사용가능합니다.

## Configuration of Ozone

### Ozone Path

Set the path to use the **Debug with Ozone** for J-Link installation.

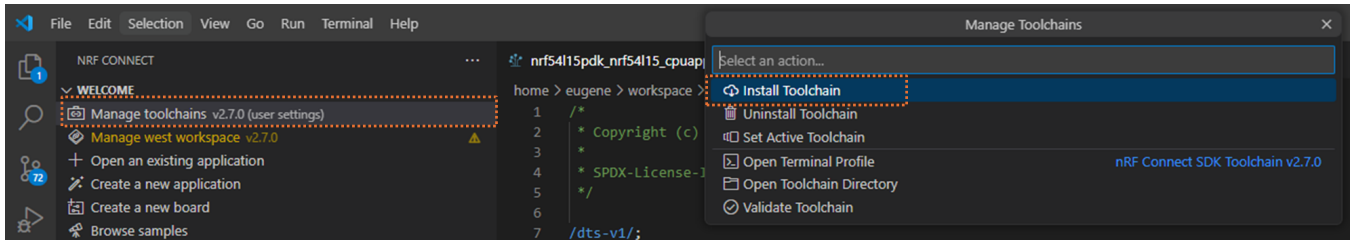
```
/mnt/c/Program Files/SEGGER/Ozone/Ozone.exe
```

# nRF Connect SDK Installation

## VSCode에 설치하기

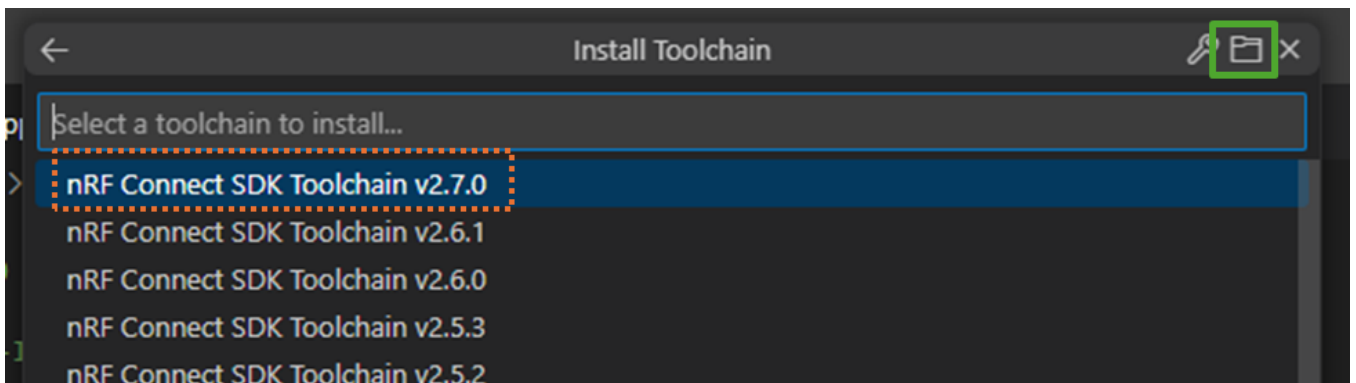
### 툴체인 설치

1. VSCode를 실행하여 설치된 nRF Connect Extension을 선택합니다 (또는 `ctrl` + `alt` + `n` 을 눌러 선택)
2. "Manage toolchains" 선택

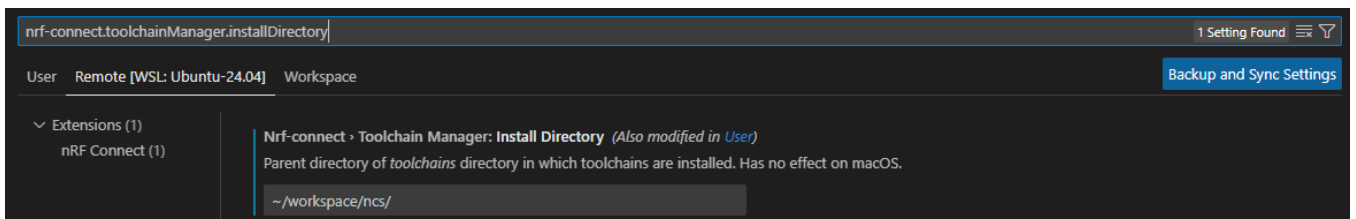


3. 화면 중간 상단의 "Install Toolchain"를 선택 (이미 설치되어 있을 경우 "Set Active Toolchain"을 통해서 변경)

"Install Toolchain"을 선택하면 아래와 같이 설치가능한 툴체인의 리스트가 보임



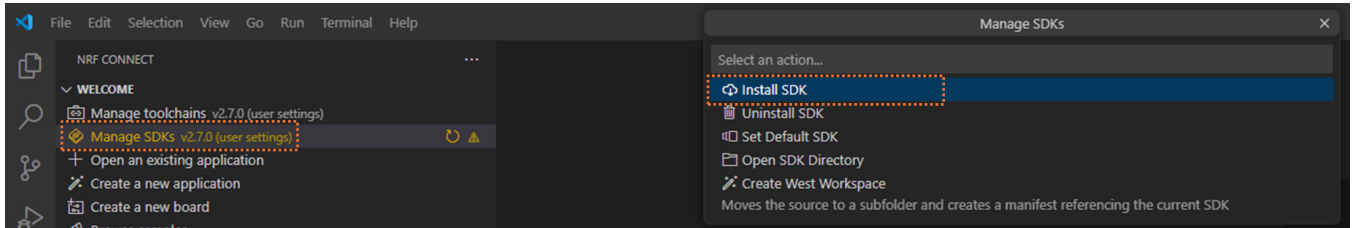
툴체인이 설치될 경로를 변경하려면 위 그림의 우측 상단의 폴더 아이콘을 클릭한 후 중간 탭인 "Remote [WSL:Ubuntu-24.04]"을 선택하여 원하는 폴더 위치를 입력



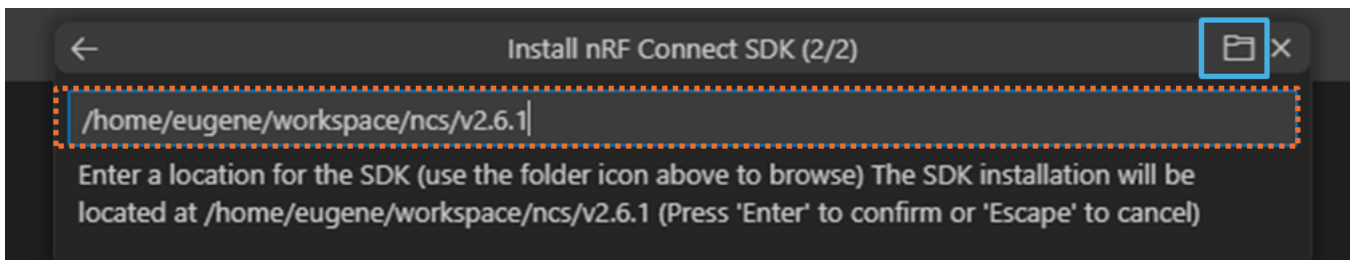
## SDK (nRF Connect SDK ) 설치

Workspace 또는 폴더가 현재 프로젝트에 선택되어 있으면 "Manage SDKs" 메뉴가 보이지 않으므로, 반드시 File > Close Workspace 또는 File > Close Folder를 선택한 후 수행

1. VSCode를 실행하여 설치된 nRF Connect Extension을 선택합니다 (또는 `ctrl` + `alt` + `n` 을 눌러 선택)
2. "Manage SDKs" 메뉴 및 "Install SDK"를 선택



3. 설치하고자하는 위치를 아래 그림처럼 폴더 아이콘을 클릭하거나 직접 입력

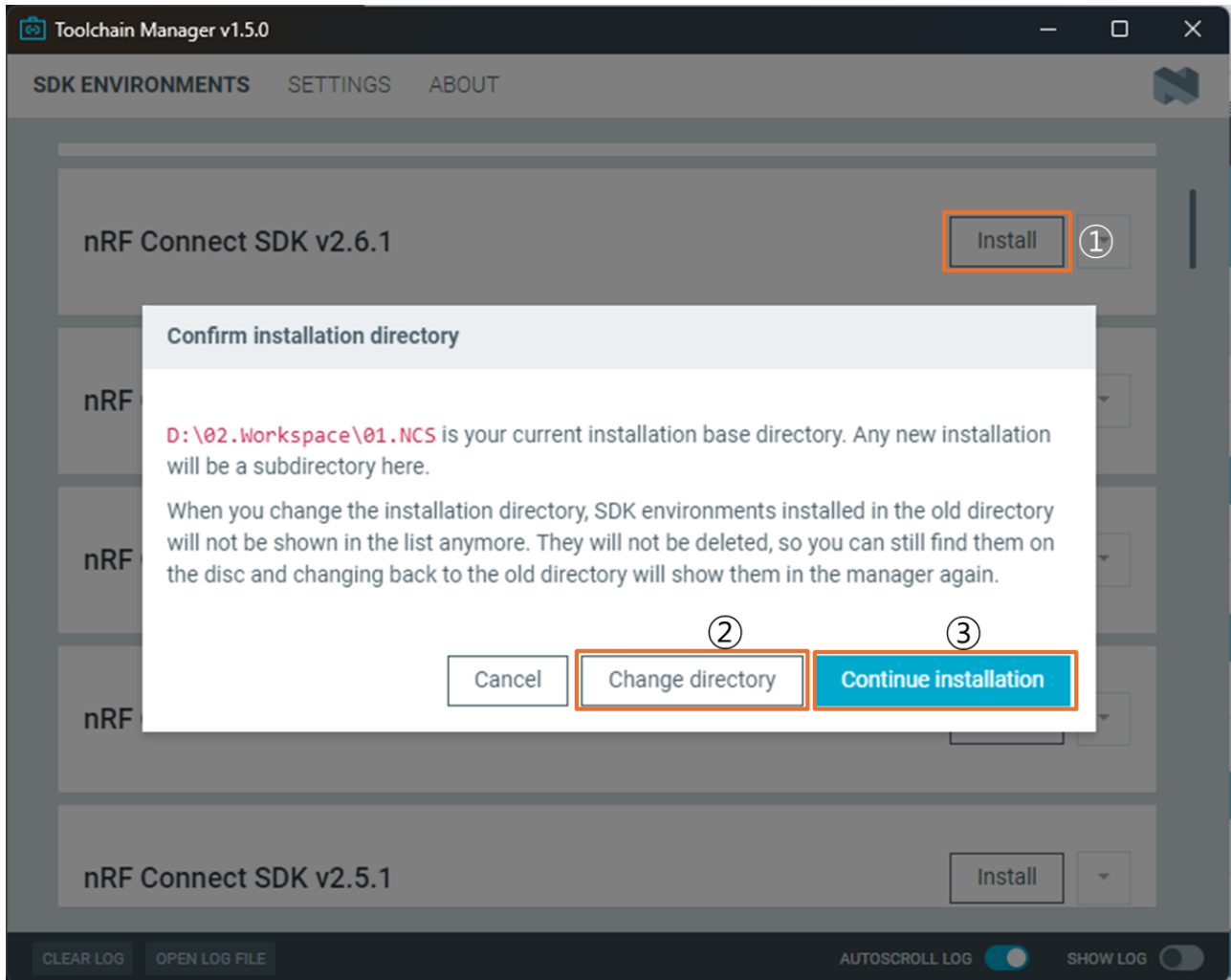


## Toolchain Manager를 통한 설치 (NCS v2.xx 까지만 사용 가능)

1. nRF Connect for desktop의 Toolchain Manager를 실행
2. **Install** (①) 버튼을 누른 후 **Continue Installation** (③)을 눌러 SDK를 설치

기본 설치 위치인 "C:\ncs"가 아닌 위치에 설치할 경우 **Change directory** (②) 버튼을 눌러 위치를 변경





# Performance Test Result

- Test Sample : smp\_svr
- Build option : `west build -b nrf5340dk_nrf5340_cpuapp --pristin -- -DEXTRA_CONF_FILE=overlay-bt.conf`
- Build environment : 32GB RAM / SSD(PM9A1 NVMe) / i7-10850H@2.7GHz / Windows 11 Enterprise (64bits) / WSL Ubuntu-24.04

## Result

OS	Elapsed time
Windows	<b>241~249</b> seconds
Linux	<b>39~61</b> seconds

# Tips

## Command line tools 설치

wget [https://nsscprodmedia.blob.core.windows.net/prod/software-and-other-downloads/desktop-software/nrf-command-line-tools/sw/versions-10-x-x/10-24-2/nrf-command-line-tools\\_10.24.2\\_amd64.deb](https://nsscprodmedia.blob.core.windows.net/prod/software-and-other-downloads/desktop-software/nrf-command-line-tools/sw/versions-10-x-x/10-24-2/nrf-command-line-tools_10.24.2_amd64.deb)

```
sudo dpkg -i ./nrf-command-line-tools_10.24.2_amd64.deb
```

```
sudo apt install /opt/nrf-command-line-tools/share/JLink_Linux_V794e_x86_64.deb --fix-broken
```

아래와 같이 설치 에러가 발생할 경우 **sudo apt install libusb-1.0.0** 으로 관련 패키지 설치필요

```
dpkg: dependency problems prevent configuration of nrf-command-line-tools:
nrf-command-line-tools depends on libusb-1.0-0 (>= 2:1.0.16); however:
Package libusb-1.0-0 is not installed.
```

## WSL 디스크 공간 관리 방법

WSL은 디스크 공간을 필요에 따라 자동으로 늘려주는 하지만 줄여주지는 않음.

따라서, 실제 물리 공간이 부족하게 되는 경우가 발생하는데 이를 해결하기 위해서 아래와 같은 방법을 사용

물론, 필요하지 않은 파일들은 직접 WSL 안에서 삭제하는 작업이 선행되어야 함

powershell

```
wsl --shutdown
diskpart
```

diskpart shell

```
select vdisk file="ext4.vhdx 경로"
attach vdisk readonly
compact vdisk
detach vdisk
```

ex)

```
DISKPART> select vdisk file="d:\00.wsl\ext4.vhdx"
DiskPart successfully selected the virtual disk file.
DISKPART> attach vdisk readonly
100 percent completed
DiskPart successfully attached the virtual disk file.
DISKPART> compact vdisk
100 percent completed
DiskPart successfully compacted the virtual disk file.
DISKPART> detach vdisk
DiskPart successfully detached the virtual disk file.
DISKPART> exit_
```

## VIM

NCS 빌드를 위한 별도의 환경변수 실행으로 VIM이 동작되지 않은 경우 관련 패키지를 사용할 수 있도록 환경 변수로 등록 필요

```
export LD_PRELOAD=/usr/lib/x86_64-linux-gnu/libexpat.so
```

## Install UI packages

WSL에 설치된 Ubuntu는 기본적으로 GUI가 없는 버전으로 설치되기 때문에, Command line으로 사용하는 것이 불편할 경우 일부 필요한 GUI 프로그램을 설치하여 운용할 수 있습니다. (nRF Connect for Desktop의 Toolchain manager를 통해서 쉽게 터미널 환경을 위한 환경 설정 스크립트 파일을 생성할 수 있습니다)

- nRF Connect for Desktop

```
cd ~
```

```
wget https://nsscprodmedia.blob.core.windows.net/prod/software-and-other-downloads/desktop-software/nrf-connect-for-desktop/5-0-0/nrfconnect-5.0.0-x86\_64.appimage
```

```
chmod a+x ~/nrfconnect-5.0.0-x86_64.appimage
~/nrfconnect-5.0.0-x86_64.appimage
```

- Google Chrome

```
cd ~
```

```
wget https://dl.google.com/linux/direct/google-chrome-stable\_current\_amd64.deb
```

```
sudo apt install --fix-missing ./google-chrome-stable_current_amd64.deb
google-chrome
```

## WSL vhdx 파일 위치 확인하기

Powershell command



```
(Get-ChildItem -Path HKCU:\Software\Microsoft\Windows\CurrentVersion\Lxss | Where-Object { $_.GetValue("DistributionName") -eq '<distribution-name>' }).GetValue("BasePath") + "\ext4.vhdx"
```

ex)



```
PS C:\Users\euuy> (Get-ChildItem -Path HKCU:\Software\Microsoft\Windows\CurrentVersion\Lxss | Where-Object {  
$_GetValue("DistributionName") -eq 'Ubuntu-24.04' }).GetValue("BasePath") + "\ext4.vhdx"  
\\?\D:\00.WSL\ext4.vhdx
```

## WSL vhdx 위치 변경하기

저장장치를 분할하여 사용하거나 별도의 위치에 WSL을 위치시켜 사용하고자 할 경우 가상환경 파일의 위치를 변경하는 방법에 대해서 설명합니다.

### 1. TAR 파일로 내보내기



```
wsl --export <Distro name> <Target location>  
ex) wsl --export Ubuntu-24.04 D:\ubuntu-24-04.tar
```

### 2. 기존 distro 삭제



```
wsl --unregister <distro>  
ex) wsl --unregister Ubuntu-24.04
```

### 3. TAR 파일 가져오기



```
wsl --import <distro> <install location> <exported file path>  
ex) wsl --import Ubuntu-24.04 D:\wsl\ubuntu-24-04\ D:\ubuntu-24-04.tar
```

### 4. 기본 distro 설정하기



```
wsl --set-default Ubuntu-24.04
```

### 5. 내보내기/가져오기 수행 시 root 계정으로 변경되는 것을 다시 개인 계정으로 변경하도록 설정


#### 1. Log-in WSL

#### 2. open wsl.conf




```
vi /etc/wsl.conf
```

### 3. Add default user



```
[user]  
default=<user>
```

### 4. Shutdown WSL on Windows terminal and restart



```
wsl --shutdown  
wsl
```

## Windows 폴더 접근 방법

WSL(Linux)에서 Windows의 드라이브를 접근하고자 할 경우 `/mnt/<drive>`를 사용하면 됩니다.

예를들어 WSL에서 C 드라이브 이동 시



```
cd /mnt/c
```

WSL에서 C 드라이브 파일을 WSL로 이동 시



```
cp /mnt/c/test.c ./test.c
```

## References

- [Linux용 Windows 하위 시스템이란? | Microsoft Learn](#)
- [WSL 설치 | Microsoft Learn](#)
- [WSL으로 Linux GUI 앱 실행 | Microsoft Learn](#)
- [WSL에서 VS Code 사용 시작 | Microsoft Learn](#)
- [WSL 디스크 공간을 관리하는 방법 | Microsoft Learn](#)

### Manual Install

[Installing manually \(nordicsemi.com\)](#)