

# Getting started with XMOS

*From setting up the environment to building and downloading with example code*

## Table of Contents

Table of Contents .....	1
List of Figures .....	4
1 우분투 설치하기 .....	6
1 - 1 Virtualbox에서 설치 .....	6
1 - 2 WSL에서 설치 .....	6
1 - 2 - 1 WSL 설치 .....	6
1 - 2 - 2 Ubuntu 설치 .....	7
2 환경 설정하기 .....	10
2 - 1 패키지 업데이트 .....	10
2 - 2 필수 패키지 설치 .....	10
2 - 3 cmake 3.25.1 설치 .....	10
3 git 설정하기 .....	11
3 - 1 SSH key 생성하기 .....	11
3 - 2 공개 키 github에 등록하기 .....	11
3 - 3 git 계정 설정 .....	12
3 - 4 github 연결 테스트 .....	12
4 xcore sdk 설치 .....	13
4 - 1 xcore sdk 다운로드 .....	13
4 - 2 xcore sdk 설치 .....	13
4 - 3 path 등록 .....	13

5 XTC Tools 설치 .....	14
5 - 1 XTC Tools 다운로드 .....	14
5 - 2 XTC Tools 복사 .....	14
5 - 3 XTC Tools 설치 .....	14
6 Windows에서 XTAG Driver 설치.....	15
6 - 1 Windows용 XTC Tools 다운로드 및 설치.....	15
6 - 2 usbipd 설치.....	15
6 - 3 usbipd를 Ubuntu에 설치.....	15
7 XK-VOICE-L71와 XTAG4와 PC간 연결 .....	16
8 usbipd 사용 방법.....	17
8 - 1 WSL에 연결할 수 있는 장치 확인.....	17
8 - 2 WSL에 장치 연결하기.....	17
8 - 3 WSL에서 장치 끊기.....	17
9 Ubuntu, Windows에서 XTAG 인식 및 연결.....	18
9 - 1 Ubuntu에서 XTAG 인식 및 연결 .....	18
9 - 2 Windows에서 XTAG 인식 및 연결.....	19
10 Target 다운로드 .....	21
10 - 1 빌드 된 firmware로 Target 다운로드.....	21
10 - 1 - 1 빌드 된 firmware 다운로드.....	21
10 - 1 - 2 Far-field Voice Local Command 다운로드.....	22
10 - 1 - 3 Low Power Far-field Voice Local Command 다운로드.....	23
10 - 1 - 4 Far-field Voice Assistant 다운로드 .....	24
10 - 2 예제 소스코드 빌드 후 Target 다운로드 .....	24
10 - 2 - 1 예제 소스코드 내려 받기 .....	24
10 - 2 - 2 Far-field Voice Local Command 빌드 및 다운로드.....	25

1 0 - 2 - 3 Low Power Far-field Voice Local Command 빌드 및 다운로드 .....	26
1 0 - 2 - 4 Far-field Voice Assistant 빌드 및 다운로드 .....	26
1 1 Sensory AI model 생성 .....	28
1 1 - 1 Sensory의 VoiceHub 접근 키 획득 .....	28
1 1 - 2 프로젝트 생성하기 .....	29
1 1 - 3 Wake Word .....	31
1 1 - 4 Simple Commands .....	32
1 2 Far-field Voice Local Command에 AI model 적용 .....	35
1 2 - 1 AI model 파일 복사 .....	35
1 2 - 2 nibble swap 파일 생성 .....	36
1 2 - 3 AI model c파일 수정 .....	36
1 2 - 4 UART baud rate 수정 .....	36
1 2 - 5 Command와 id의 일치화 .....	37
1 2 - 6 AI model의 command를 소스코드에 반영 .....	38
1 2 - 7 wav 수정하기 .....	39
1 2 - 8 cmake 수정 .....	40
1 3 Low Power Far-field Voice Local Command에 AI model 적용 .....	42
1 3 - 1 AI model 파일 복사 .....	42
1 3 - 2 nibble swap 파일 생성 .....	42
1 3 - 3 UART baud rate 수정 .....	43
1 3 - 4 AI model의 command를 소스코드에 반영 .....	43
1 3 - 5 AI model wake word를 소스코드에 반영 .....	44
Revision History .....	45

## List of Figures

figure 1-1 Install Ubuntu .....	7
figure 1-2 wait installing ubuntu.....	8
figure 1-3 set id and password.....	8
figure 1-4 ready the ubuntu.....	9
figure 3-1 ssh keys.....	11
figure 7-1 connection between XK-VOICE-L71 and XTAG4 and PC.....	16
figure 8-1 device list.....	17
figure 8-2 connect device .....	17
figure 8-3 disconnect device.....	17
figure 9-1 setup xmos devices.....	18
figure 9-2 check xmos devices.....	18
figure 9-3 available xmos devices.....	19
figure 9-4 Prompt for XTC Tools.....	19
figure 9-5 check connection the XTAG.....	20
figure 10-1 download pre-built example firmware.....	21
figure 10-2 pre-built example firmware files .....	22
figure 10-3 command words.....	23
figure 10-4 flash firmware .....	25
figure 11-1 Request Access to VoiceHub.....	28
figure 11-2 registration access to VoiceHub.....	29
figure 11-3 New Project .....	30
figure 11-4 select a project type .....	30
figure 11-5 Wake Word Project .....	31

figure 1 1 -6 add Wake Word.....	32
figure 1 1 -7 Command Set Project.....	33
figure 1 1 -8 add command set.....	34
figure 1 2 -1 List of files for only command set.....	35
figure 1 2 -2 copy the command files.....	35
figure 1 2 -3 generate nibble swap.....	36
figure 1 2 -4 command-pc62w-6.4.0-op10-dev-search.c 수정.....	36
figure 1 2 -5 modify the baud rate.....	37
figure 1 2 -6 define id for keywork and command.....	37
figure 1 2 -7 command set and id number .....	37
figure 1 2 -8 number of commands.....	38
figure 1 2 -9 asr_lut array variable.....	38
figure 1 2 -10 response wav.....	39
figure 1 2 -11 apply wav file name to source code.....	40
figure 1 2 -12 modify cmake file.....	41
figure 1 3 -1 wake word files and command files .....	42
figure 1 3 -2 copy the wake word and command files .....	42
figure 1 3 -3 generate nibble swap.....	43
figure 1 3 -4 modify the baud rate.....	43
figure 1 3 -5 number of commands.....	43
figure 1 3 -6 asr_lut array variable.....	44
figure 1 3 -7 modify wake word.....	44

## 1 우분투 설치하기

아래 두 가지 중 하나의 방법으로 Ubuntu(버전 20 이상)를 설치합니다.

### 1 - 1 Virtualbox에서 설치

Ubuntu 이미지를 다운로드 받고

[Download Ubuntu Desktop](#) | [Download | Ubuntu](#)

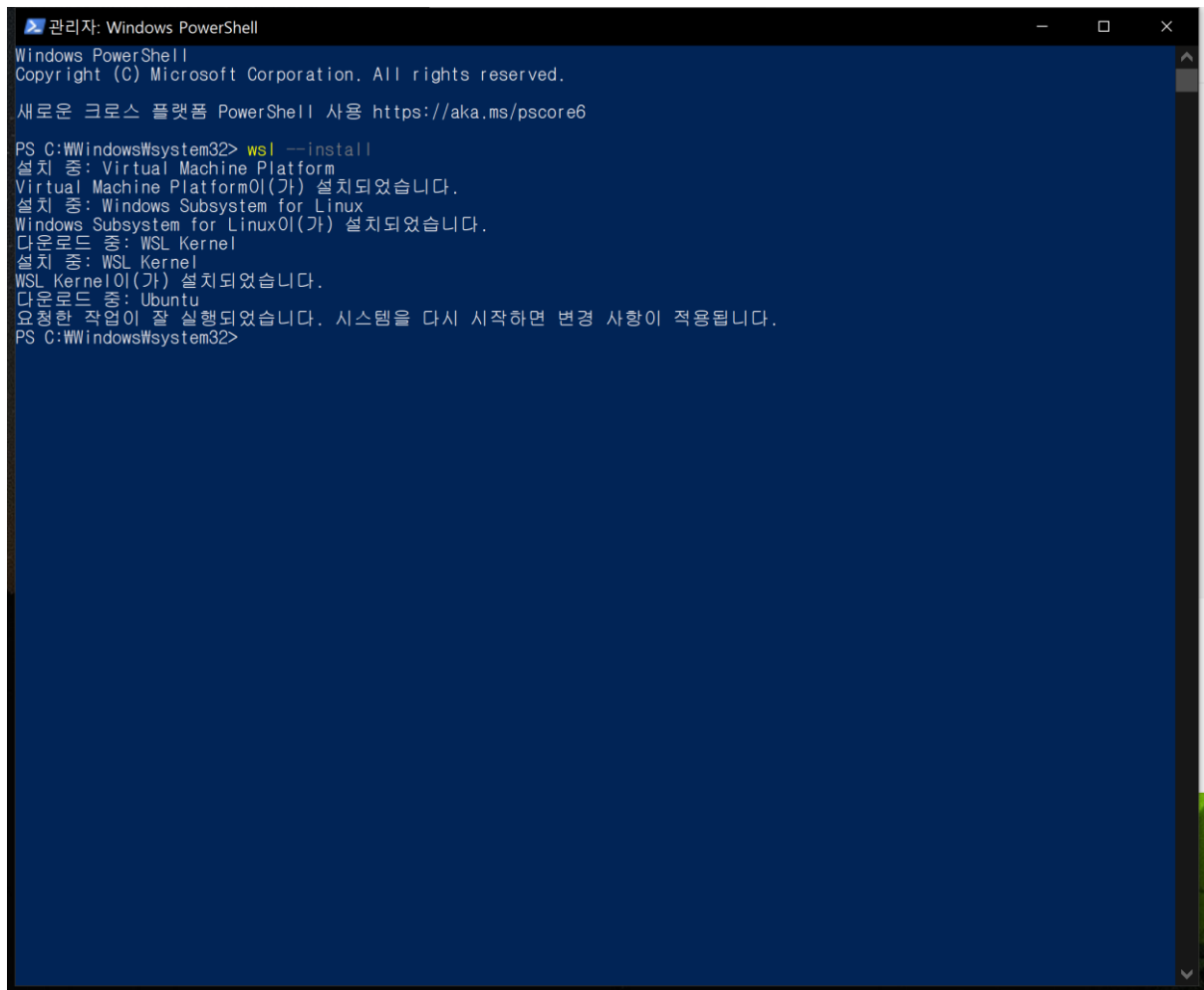
Virtualbox에서 Ubuntu를 설치합니다.

### 1 - 2 WSL에서 설치

#### 1 - 2 - 1 WSL 설치

Windows Powershell을 관리자 권한으로 실행하고 `wsl --install`으로 WSL을 설치합니다.

figure 1-1 Install Ubuntu



```
관리자: Windows PowerShell
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

새로운 크로스 플랫폼 PowerShell 사용 https://aka.ms/pscore6

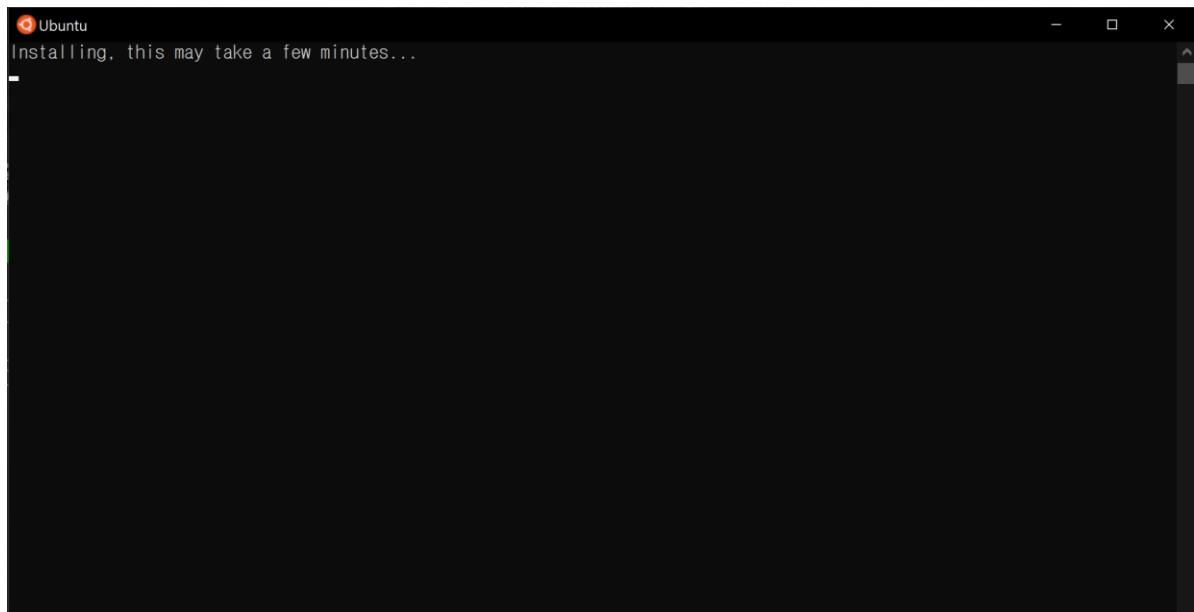
PS C:\Windows\System32> wsl --install
설치 중: Virtual Machine Platform
Virtual Machine Platform이(가) 설치되었습니다.
설치 중: Windows Subsystem for Linux
Windows Subsystem for Linux이(가) 설치되었습니다.
다운로드 중: WSL Kernel
설치 중: WSL Kernel
WSL Kernel이(가) 설치되었습니다.
다운로드 중: Ubuntu
요청한 작업이 잘 실행되었습니다. 시스템을 다시 시작하면 변경 사항이 적용됩니다.
PS C:\Windows\System32>
```

Windows를 다시 시작합니다.

## 1-2-2 Ubuntu 설치

Windows를 다시 시작한 후 자동으로 Ubuntu를 설치합니다.

**figure 1-2 wait installing ubuntu**



얼마 뒤에 username과 password를 설정하면 WSL Ubuntu 설치는 끝납니다.

**figure 1-3 set id and password**

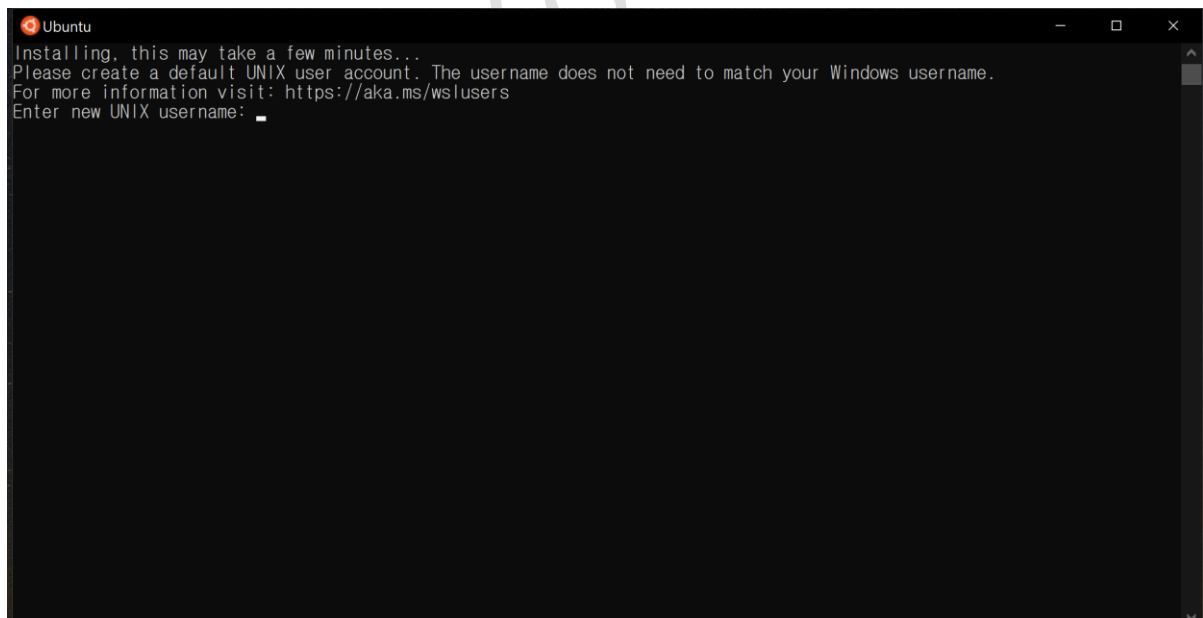




figure 1 -4 ready the ubuntu

```
mason@LAPTOP-026PQERJ: ~
Retype new password:
passwd: password updated successfully
Installation successful!
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

Welcome to Ubuntu 20.04 LTS (GNU/Linux 5.10.16.3-microsoft-standard-WSL2 x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

System information as of Mon Dec 19 22:48:02 KST 2022

System load: 0.03          Processes:            8
Usage of /:  0.4% of 250.98GB Users logged in:      0
Memory usage: 0%          IPv4 address for eth0: 172.22.168.160
Swap usage:  0%

0 updates can be installed immediately.
0 of these updates are security updates.

The list of available updates is more than a week old.
To check for new updates run: sudo apt update

This message is shown once once a day. To disable it please create the
/home/mason/.hushlogin file.
mason@LAPTOP-026PQERJ:~$
```

## 2 환경 설정하기

### 2 - 1 패키지 업데이트

Ubuntu 설치 후 패키지 업데이트를 진행합니다.

```
$ sudo apt update && sudo apt upgrade
```

### 2 - 2 필수 패키지 설치

필수 패키지를 설치합니다.

```
$ sudo apt install make g++ libssl-dev pkg-config libusb-1.0-0-dev libncurses5 wget git libfuse2
```

### 2 - 3 cmake 3.25.1 설치

Ubuntu에서 cmake-3.25.1.tar.gz를 다운로드 받고

```
$ wget https://cmake.org/files/v3.25/cmake-3.25.1.tar.gz
```

cmake-3.25.1을 설치합니다.

```
$ tar -zxvf cmake-3.25.1.tar.gz
```

```
~/cmake-3.25.1$ ./bootstrap
```

```
~/cmake-3.25.1$ make
```

```
~/cmake-3.25.1$ sudo make install
```

cmake-3.25.1이 제대로 설치되었는지 확인합니다.

```
$ cmake --version
```

## 3 git 설정하기

### 3 - 1 SSH key 생성하기

ssh key를 생성합니다.

```
$ ssh-keygen -t ed25519 -C "your_email@example.com"
```

다음 문구에서는 입력하지 않고 enter를 눌러 넘어갑니다.

Enter a file in which to save the key (/home/YOU/.ssh/ALGORITHM):

Enter passphrase (empty for no passphrase):

Enter same passphrase again:

### 3 - 2 공개 키 github에 등록하기

github에서 Setting → SSH and GPG keys 항목으로 들어가

SSH keys에서 Ubuntu에서 생성한 공개 키를 추가합니다.

공개 키는 Ubuntu ~/.ssh/id\_ed25519.pub에 있고

```
$ cat ~/.ssh/id_ed25519.pub
```

위 명령으로 쉘에서 보이는 전체 내용을 그대로 복사하고 github의 SSH keys에 붙여 넣어 추가합니다.


figure 3-1 ssh keys

## SSH keys

[New SSH key](#)

This is a list of SSH keys associated with your account. Remove any keys that you do not recognize.

### Authentication Keys

**WorkPC\_WSL\_Ubuntu-22.04**

SSH

Last used within the last week — Read/write

Delete

### 3 - 3 git 계정 설정

Ubuntu에서 git 정보에 github의 계정을 등록합니다.

```
$ git config --global user.name "user-name"
```

```
$ git config --global useremail "user-email"
```

\*user-name과 user-email을 github의 정보와 동일하게 입력해주세요.

### 3 - 4 github 연결 테스트

아래 명령어로 연결 테스트를 진행합니다.

```
$ ssh -T git@github.com
```

아래 메시지가 나오면 정상적으로 연결된 것입니다.

Hi [user.name]! You've successfully authenticated, but GitHub does not provide shell access.

## 4 xcore sdk 설치

### 4-1 xcore sdk 다운로드

Ubuntu에서 다음 명령어로 xcore sdk를 다운로드합니다.

```
$ git clone --recurse-submodules git@github.com:xmos/xcore_iot.git
```

### 4-2 xcore sdk 설치

```
$ cd xcore_iot
```

위 명령으로 sdk폴더로 들어간 다음, 아래 명령으로 sdk를 설치해줍니다.

```
$ cmake -B build_host
```

```
$ cd build_host
```

```
$ sudo make install
```

설치가 완료되면 /opt/xmos/bin에 sdk가 설치됩니다.

### 4-3 path 등록

.bashrc에 설치된 xcore sdk를 path에 등록합니다.

```
$ sudo vi ~/.bashrc
```

```
PATH=$PATH:/opt/xmos/bin/
```

위 PATH줄을 입력하고 저장합니다.

## 5 XTC Tools 설치

### 5 - 1 XTC Tools 다운로드

[Software tools | XMOS](#)에서 Linux 64-bit를 눌러 Linux용 XTC Tools을 다운로드합니다.

### 5 - 2 XTC Tools 복사

다운로드한 XTC Tools을 Ubuntu로 복사합니다.

```
$ cp /mnt/c/Users/user/Downloads/Tools-15---Linux-64_15_2_1.tgz ~
```

### 5 - 3 XTC Tools 설치

XTC Tools 압축파일을 풀고 set environment를 실행하면 됩니다.

```
$ tar -xf Tools-15---Linux-64_15.1.4.tgz -C ~
```

```
$ cd XMOS/XTC/15.2.1
```

```
~/XMOS/XTC/15.2.1$ source SetEnv
```

\*Shell을 새로 열 때마다 source SetEnv를 실행해주어야 합니다.

## 6 Windows에서 XTAG Driver 설치

Windows의 WSL을 이용해 Ubuntu를 설치하여 사용하고 있다면 다음 순서를 진행해주세요.

### 6 - 1 Windows용 XTC Tools 다운로드 및 설치

[Software tools | XMOS](#)에서 'Microsoft Windows \*'를 눌러 윈도우용 XTC Tools를 다운로드한 후 설치합니다.

이 과정에서 XTAG Driver를 설치됩니다.

### 6 - 2 usbipd 설치

[Releases · dorssel/usbipd-win \(github.com\)](#)에서 usbipd-win\_3.1.0.msi를 다운로드한 뒤 설치합니다.

설치 뒤 Windows를 재부팅합니다.

### 6 - 3 usbipd를 Ubuntu에 설치

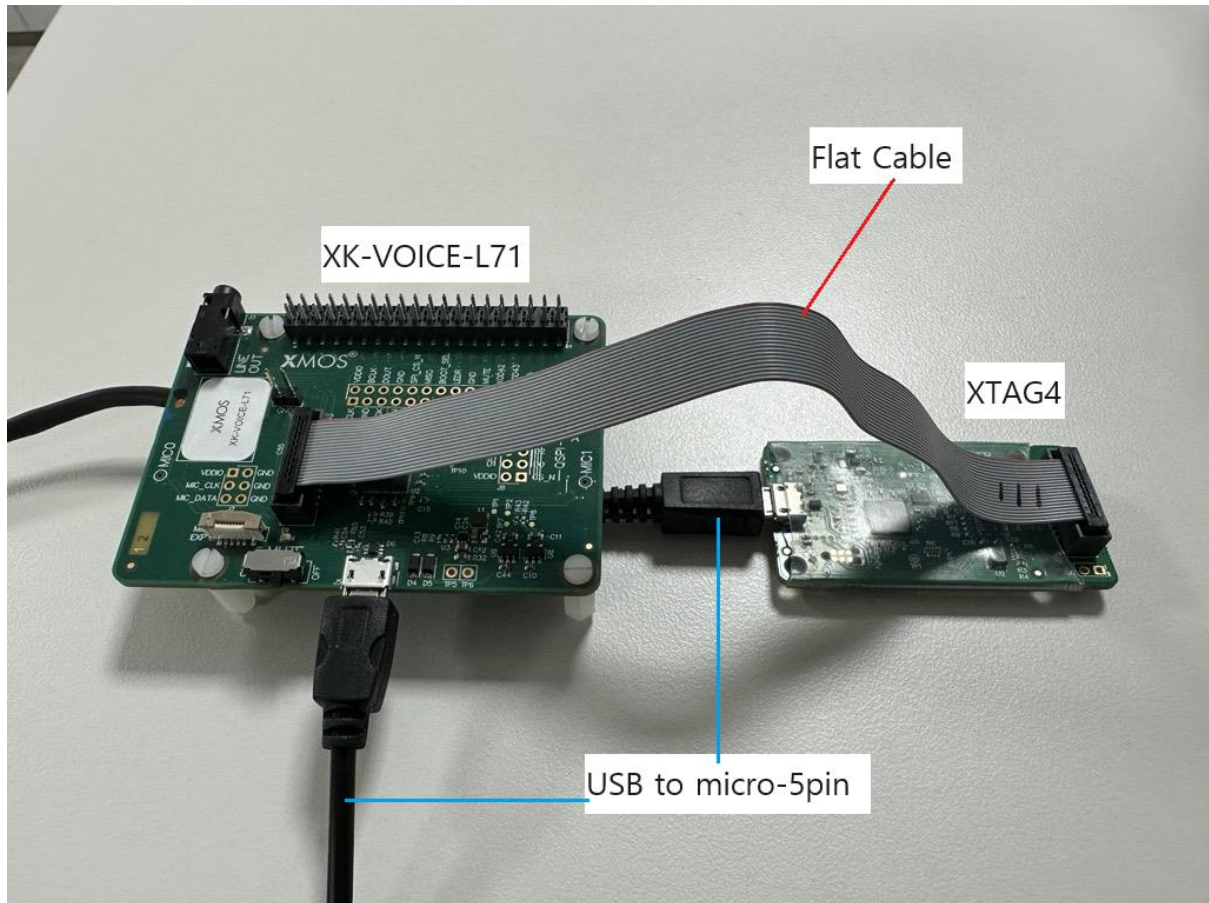
재부팅 뒤 Ubuntu를 실행한 뒤 다음 명령으로 usbipd를 Ubuntu에 설치합니다.

```
$ sudo apt install linux-tools-generic hwddata
```

```
$ sudo update-alternatives --install /usr/local/bin/usbip usbip /usr/lib/linux-tools/*-generic/usbip 20
```

## 7 XK-VOICE-L71와 XTAG4와 PC간 연결

figure 7 -1 connection between XK-VOICE-L71 and XTAG4 and PC



그림과 같이 XK-VOICE-L71과 XTAG4와 PC를 연결합니다.

XK-VOICE-L71과 XTAG4에 연결된 micro5pin Cable을 각각 PC에 연결합니다.

특히 Windows에 XTC Tool이 설치되어 있지 않으면 XTAG4 장치를 제대로 인식할 수 없습니다.



## 8 usbipd 사용 방법

Windows Powershell을 관리자 권한으로 실행합니다. usbipd는 XTAG4를 WSL에 연결하는 방법입니다.

### 8-1 WSL에 연결할 수 있는 장치 확인

> usbipd wsl list

figure 8-1 device list

```
PS C:\Windows\system32> usbipd wsl list
BUSID  VID:PID  DEVICE                                STATE
2-7    27c6:6a94 Goodix MOC Fingerprint               Not attached
2-10   8087:0026 인텔(R) 무선 Bluetooth(R)           Not attached
4-1    0bda:8153 Realtek USB GbE Family Controller     Not attached
7-1    20b1:f7d5 XMOS XTAG-4                           Not attached
```

### 8-2 WSL에 장치 연결하기

> usbipd wsl attach -b 7-1

figure 8-2 connect device

```
PS C:\Windows\system32> usbipd wsl attach -b 7-1
PS C:\Windows\system32> usbipd wsl list
BUSID  VID:PID  DEVICE                                STATE
2-7    27c6:6a94 Goodix MOC Fingerprint               Not attached
2-10   8087:0026 인텔(R) 무선 Bluetooth(R)           Not attached
4-1    0bda:8153 Realtek USB GbE Family Controller     Not attached
7-1    20b1:f7d5 XMOS XTAG-4                           Attached - WSL
```

\*BUSID는 연결하는 포트마다 달라집니다.

### 8-3 WSL에서 장치 끊기

> usbipd wsl detach -b 7-1

figure 8-3 disconnect device

```
PS C:\Windows\system32> usbipd wsl detach -b 7-1
PS C:\Windows\system32> usbipd wsl list
BUSID  VID:PID  DEVICE                                STATE
2-7    27c6:6a94 Goodix MOC Fingerprint               Not attached
2-10   8087:0026 인텔(R) 무선 Bluetooth(R)           Not attached
4-1    0bda:8153 Realtek USB GbE Family Controller     Not attached
7-1    20b1:f7d5 XMOS XTAG-4                           Not attached
```

## 9 Ubuntu, Windows에서 XTAG 인식 및 연결

### 9-1 Ubuntu에서 XTAG 인식 및 연결

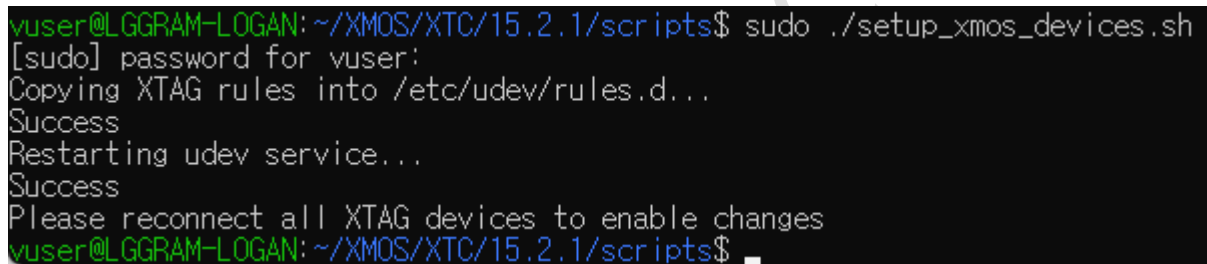
Ubuntu에서 ~/XMOS/XTC/15.2.1/scripts로 이동한 뒤

```
$ cd ~/XMOS/XTC/15.2.1/scripts
```

아래 3단계 과정으로 인식시켜주면 됩니다.

```
~/XMOS/XTC/15.2.1/scripts$ sudo ./setup_xmos_devices.sh
```

**figure 9-1 setup xmos devices**

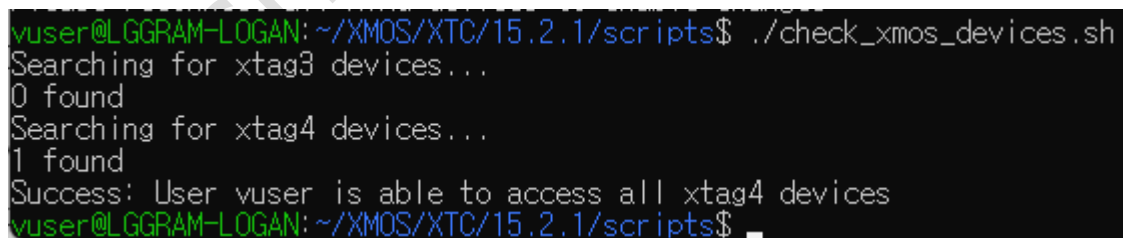


```
vuser@LGGRAM-LOGAN:~/XMOS/XTC/15.2.1/scripts$ sudo ./setup_xmos_devices.sh
[sudo] password for vuser:
Copying XTAG rules into /etc/udev/rules.d...
Success
Restarting udev service...
Success
Please reconnect all XTAG devices to enable changes
vuser@LGGRAM-LOGAN:~/XMOS/XTC/15.2.1/scripts$
```

연결된 Device의 연결을 해제한 뒤 다시 연결해 줍니다.

```
~/XMOS/XTC/15.2.1/scripts$ ./check_xmos_devices.sh
```

**figure 9-2 check xmos devices**



```
vuser@LGGRAM-LOGAN:~/XMOS/XTC/15.2.1/scripts$ ./check_xmos_devices.sh
Searching for xtag3 devices...
0 found
Searching for xtag4 devices...
1 found
Success: User vuser is able to access all xtag4 devices
vuser@LGGRAM-LOGAN:~/XMOS/XTC/15.2.1/scripts$
```

```
~/XMOS/XTC/15.2.1/scripts$ xrun -l
```

figure 9-3 available xmos devices

```
vuser@LGGRAM-LOGAN: ~/XNOS/XTC/15.2.1/scripts$ xrun -l  
Available XNOS Devices  
-----  
ID      Name      Adapter ID      Devices  
--      -  
0       XNOS XTAG-4  DM7BULJM       P[0]  
vuser@LGGRAM-LOGAN: ~/XNOS/XTC/15.2.1/scripts$
```

xrun -l 명령 실행 후 연결된 XTAG가 연결이 끊어지는 경우가 있는데, 다시 연결해주면 됩니다. 그러면 정상적으로 리스트를 확인하실 수 있습니다. (xflash -l 명령도 동일하게 작동됩니다.)

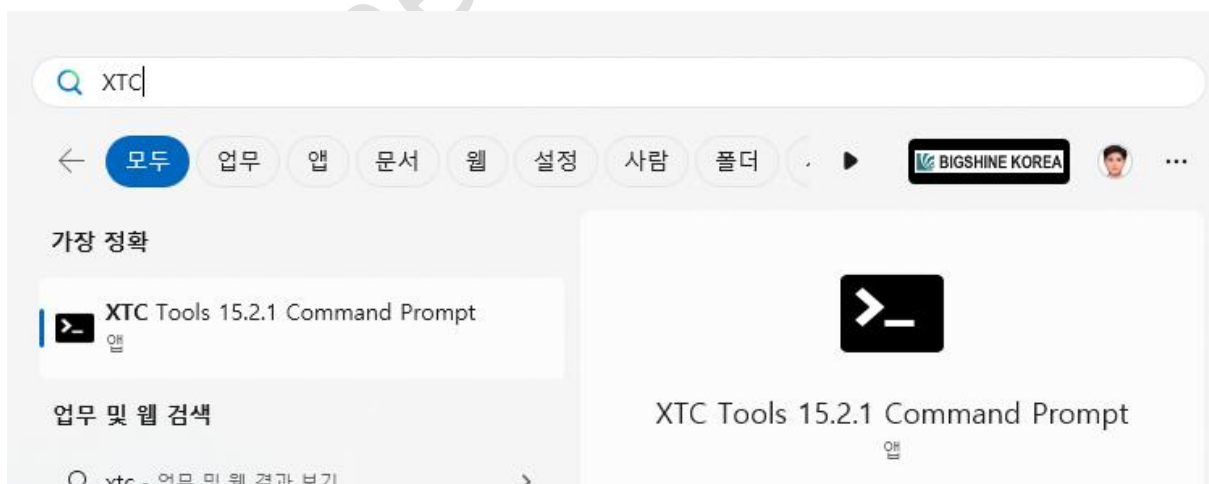
우측의 Devices 상태가 P[n]으로 표시되어야 firmware를 다운로드할 수 있는 상태입니다.

다시 연결할 때 7장의 usbipd명령으로 re-attach합니다.

#### 9-2 Windows에서 XTAG 인식 및 연결

Windows 상에서도 XTAG를 연결할 수 있습니다. Windows에서 XTAG를 연결할 경우 7장의 usbipd 명령을 사용하지 않아도 되는 장점이 있습니다. Windows용 XTC Tool이 설치되어 있어야 합니다.

figure 9-4 Prompt for XTC Tools



윈도우의 검색창에서 XTC Tools 15.2.1 Command Prompt를 검색하고 실행합니다.

figure 9 -5 check connection the XTAG

```
C:\Users\bsk>xflash -l

Available XMOS Devices
-----

ID      Name                Adapter ID          Devices
--      -
0       XMOS XTAG-4         DM7BULJM           P[0]

C:\Users\bsk>
```

'xflash -l' 명령으로 XTAG의 연결을 확인합니다.

우측의 Devices 상태가 P[n]으로 표시되어야 firmware를 다운로드할 수 있는 상태입니다.

## 1 0 Target 다운로드

XMOS 홈페이지에서 이미 빌드한 firmware를 다운로드하는 방법과

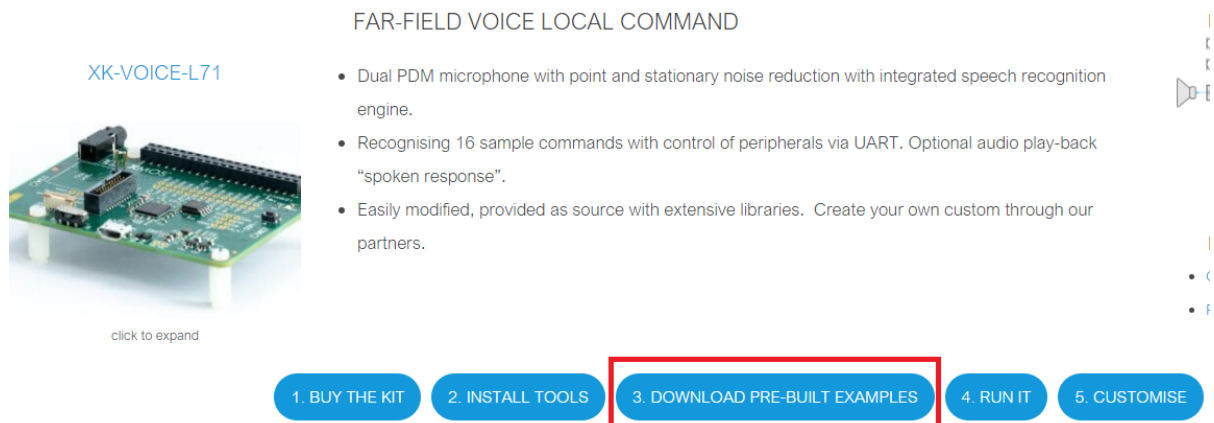
예제 소스코드를 빌드한 후 다운로드하는 방법이 있습니다.

1 0 - 1 빌드 된 firmware로 Target 다운로드

1 0 - 1 - 1 빌드 된 firmware 다운로드

[XK-VOICE-L71 페이지](#)에서 '3. DOWNLOAD PRE-BUILT EXAMPLES'를 눌러 빌드 된 firmware를 다운로드합니다.

**figure 1 0 -1 download pre-built example firmware**



The screenshot displays the product page for the XK-VOICE-L71. On the left is an image of the green circuit board with the text 'XK-VOICE-L71' above it and 'click to expand' below it. To the right of the image, under the heading 'FAR-FIELD VOICE LOCAL COMMAND', are three bullet points describing the product's features. At the bottom of the page is a horizontal navigation bar with five blue buttons: '1. BUY THE KIT', '2. INSTALL TOOLS', '3. DOWNLOAD PRE-BUILT EXAMPLES', '4. RUN IT', and '5. CUSTOMISE'. The third button, '3. DOWNLOAD PRE-BUILT EXAMPLES', is highlighted with a red rectangular border.

XK-VOICE-L71


click to expand










FAR-FIELD VOICE LOCAL COMMAND

- Dual PDM microphone with point and stationary noise reduction with integrated speech recognition engine.
- Recognising 16 sample commands with control of peripherals via UART. Optional audio play-back "spoken response".
- Easily modified, provided as source with extensive libraries. Create your own custom through our partners.

1. BUY THE KIT 2. INSTALL TOOLS 3. DOWNLOAD PRE-BUILT EXAMPLES 4. RUN IT 5. CUSTOMISE

**figure 1 0 -2 pre-built example firmware files**

 XCORE-VOICE-Solution---Software---Examples\_2\_0\_0.zip

-  build\_metadata.json
-  example\_ffd.xe
-  example\_ffd\_data\_partition.bin
-  example\_ffva\_int\_fixed\_delay.xe
-  example\_ffva\_int\_fixed\_delay\_data\_partition.bin
-  example\_ffva\_ua\_adec\_altarch.xe
-  example\_ffva\_ua\_adec\_altarch\_data\_partition.bin
-  example\_low\_power\_ffd.xe
-  example\_low\_power\_ffd\_data\_partition.bin

#### 1 0 - 1 - 2 Far-field Voice Local Command 다운로드

Far-field Voice Local Command는 wake word와 command를 사용하여 동작합니다.

sleep mode없이 일반적인 동작 중에 wake word와 command를 받아 처리하는 firmware입니다.

wake word와 command는 다음과 같습니다.

figure 1 0 -3 command words

### **Wakewords**

- Hello XMOS

### **Dictionary Commands**

- Switch on the TV
- Switch off the TV
- Channel up
- Channel down
- Volume up
- Volume down
- Switch on the lights
- Switch off the lights
- Brightness up
- Brightness down
- Switch on the fan
- Switch off the fan
- Speed up the fan
- Slow down the fan
- Set higher temperature
- Set lower temperature

Far-field Voice Local Command는 아래 명령으로 target에 다운로드할 수 있습니다.

```
xflash --factory example_ffd.xe --data example_ffd_data_partition.bin --boot-partition-size 0x100000
```

1 0 - 1 - 3 Low Power Far-field Voice Local Command 다운로드

Low Power Far-field Voice Local Command는 Far-field Voice Local Command에서 Low Power를 추

가했습니다. wakeword를 들으면 깨어나고 얼마 후에 다시 sleep mode로 들어가 전력소모를 줄일 수 있는 firmware입니다.

wakeword와 command는 Far-field Voice Local Command와 동일합니다.

아래 명령으로 Low Power Far-field Voice Local Command를 target에 다운로드할 수 있습니다.

```
xflash --factory example_low_power_ffd.xe --data example_low_power_ffd_data_partition.bin --boot-partition-size 0x100000
```

#### 1 0 - 1 - 4 Far-field Voice Assistant 다운로드

Far-field Voice Assistant는 DSP영역, Acoustic Echo Canceller, Interference Canceller, Noise Suppressor, Automatic Gain Control 기능을 시험해 볼 수 있는 firmware입니다.

아래 명령으로 Far-field Voice Assistant를 target에 다운로드할 수 있습니다.

```
xflash --factory example_ffva_ua_adec_altarch.xe --data example_ffva_ua_adec_altarch_data_partition.bin --boot-partition-size 0x100000
```

ua는 USB Accessory로 PC에 연결해서 확인할 수 있습니다.

int는 전 처리된 소리를 I2S, I2C 통신으로 뒤 단에 보내는 firmware이며, 아래 명령으로 target에 다운로드를 할 수 있습니다.

```
xflash --factory example_ffva_int_fixed_delay.xe --data example_ffva_int_fixed_delay_data_partition.bin --boot-partition-size 0x100000
```

#### 1 0 - 2 예제 소스코드 빌드 후 Target 다운로드

##### 1 0 - 2 - 1 예제 소스코드 내려 받기

아래 명령으로 XMOS의 예제 소스코드를 내려 받습니다.

예제 소스코드는 github에서 확인하실 수 있습니다. ([링크](#))



```
$ git clone --recurse-submodules git@github.com:xmos/sln_voice.git
```

만약 이미 내려 받은 소스코드가 있다면 아래 명령으로 소스코드를 갱신할 수 있습니다.

```
$ git pull
```

```
$ git submodule update --init --recursive
```

## 1 0 - 2 - 2 Far-field Voice Local Command 빌드 및 다운로드

sln\_voice폴더 아래에서 아래 명령으로 빌드할 수 있습니다.

```
~/sln_voice$ cmake -B build -DCMAKE_TOOLCHAIN_FILE=xmos_cmake_toolchain/xs3a.cmake
```

```
~/sln_voice$ cd build
```

```
~/sln_voice/build$ make example_ffd
```

아래 명령으로 Target에 다운로드할 수 있습니다.

```
~/sln_voice/build$ make flash_app_example_ffd
```

**figure 1 0 -4 flash firmware**

```
user@LOGGAM-LOGAN: ~/sln_voice/build$ make flash_app_example_ffd
[ 0%] Create filesystem
Filesystem image example_ffd_fat.fs successfully populated with contents of directory /home/user/sln_voice/examples/ffd/filesystem_support/english_usa
[ 0%] Built target make_fs_example_ffd
[ 0%] Built target lib_tflite_micro_patch
[ 6%] Built target inferencing_lib_nn
[ 26%] Built target inferencing_tflite_micro
[ 26%] Built target framework_core_clock_control
[ 30%] Built target lib_l2_cache
[ 30%] Built target lib_otpinfno
[ 30%] Built target lib_random
[ 30%] Built target lib_trycatch
[ 63%] Built target lib_xcore_math
[ 63%] Built target lib_l2c
[ 63%] Built target lib_l2s
[ 63%] Built target lib_mlc_array
[ 63%] Built target lib_spi
[ 63%] Built target lib_uart
[ 66%] Built target lib_qspi_fast_read
[ 66%] Built target fwk_voice_module_lib_vmr_inference
[ 66%] Built target fwk_voice_module_lib_vmr_features
[ 66%] Built target fwk_voice_module_lib_ns
[ 70%] Built target fwk_voice_module_lib_aec
[ 70%] Built target fwk_voice_module_lib_lc
[ 70%] Built target fwk_voice_module_lib_agc
[ 83%] Built target tile1_example_ffd
[ 96%] Built target tile0_example_ffd
[100%] Merge tile 0,1 of tile1_example_ffd.xe into tile0_example_ffd.xe to create example_ffd.xe
[100%] Built target example_ffd
[100%] Copy Sensory NET file
[100%] Built target example_ffd_model.bin
[100%] Flash application and data partition
Site 0 has finished successfully.
[100%] Built target flash_app_example_ffd
user@LOGGAM-LOGAN: ~/sln_voice/build$
```

아래 명령으로 XTAG4를 통해 디버깅을 할 수 있습니다.

```
~/sln_voice/build$ make debug_example_ffd
```

아래 명령으로 빌드 이미지를 제거할 수 있습니다.

```
~/sln_voice/build$ make clean
```

#### 1 0 - 2 - 3 Low Power Far-field Voice Local Command 빌드 및 다운로드

sln\_voice폴더 아래에서 아래 명령으로 빌드할 수 있습니다.

```
~/sln_voice$ cmake -B build -DCMAKE_TOOLCHAIN_FILE=xmos_cmake_toolchain/xs3a.cmake
```

```
~/sln_voice$ cd build
```

```
~/sln_voice/build$ make example_low_power_ffd
```

아래 명령으로 Target에 다운로드할 수 있습니다.

```
~/sln_voice/build$ make flash_app_example_low_power_ffd
```

아래 명령으로 XTAG4를 통해 디버깅을 할 수 있습니다.

```
~/sln_voice/build$ make debug_example_low_power_ffd
```

#### 1 0 - 2 - 4 Far-field Voice Assistant 빌드 및 다운로드

sln\_voice폴더 아래에서 아래 명령으로 빌드할 수 있습니다.

```
~/sln_voice$ cmake -B build -DCMAKE_TOOLCHAIN_FILE=xmos_cmake_toolchain/xs3a.cmake
```

```
~/sln_voice$ cd build
```

```
~/sln_voice/build$ make example_ffva_ua_adec_altarch
```

아래 명령으로 Target에 다운로드할 수 있습니다.

```
~/sln_voice/build$ make flash_app_example_ffva_ua_adec_altarch
```

아래 명령으로 XTAG4를 통해 디버깅을 할 수 있습니다.

```
~/sln_voice/build$ make debug_example_ffva_ua_adec_altarch
```

ua는 USB Accessory입니다. PC에서 녹음할 수 있으며 PC의 소리를 XMOS에 연결된 스피커로 송출할 수 있습니다.

I2S, I2C통신으로 시험하고 싶으시다면 example\_ffva\_int\_fixed\_delay으로 다운로드하시면 됩니다.

## 1 1 Sensory AI model 생성

언어 인식 AI model을 만들기 위해 미국 회사, Sensory라는 3<sup>rd</sup> party를 이용합니다.

Sensory의 VoiceHub는 웹페이지 기반으로 언어 인식 AI model을 생성할 수 있습니다.

Sensory의 솔루션은 기본적으로 유료입니다. 라이선스를 구입하여 사용해야 합니다.

하지만 VoiceHub의 사용은 무료로 제공되고 있습니다. 언제든지 유료로 전환될 수 있습니다.

### 1 1 - 1 Sensory의 VoiceHub 접근 키 획득

Sensory의 홈페이지에서 VoiceHub를 사용하기 위한 VoiceHub의 접근 키를 얻어야 합니다.

공식 홈페이지에서 상단 메뉴에서 'Resources' → 'Sensory VoiceHub'로 들어가 '중간에 있는 'Request Access to VoiceHub'를 통해 VoiceHub 접근 키를 요청하세요. ([링크](#))

**figure 1 1 -1 Request Access to VoiceHub**

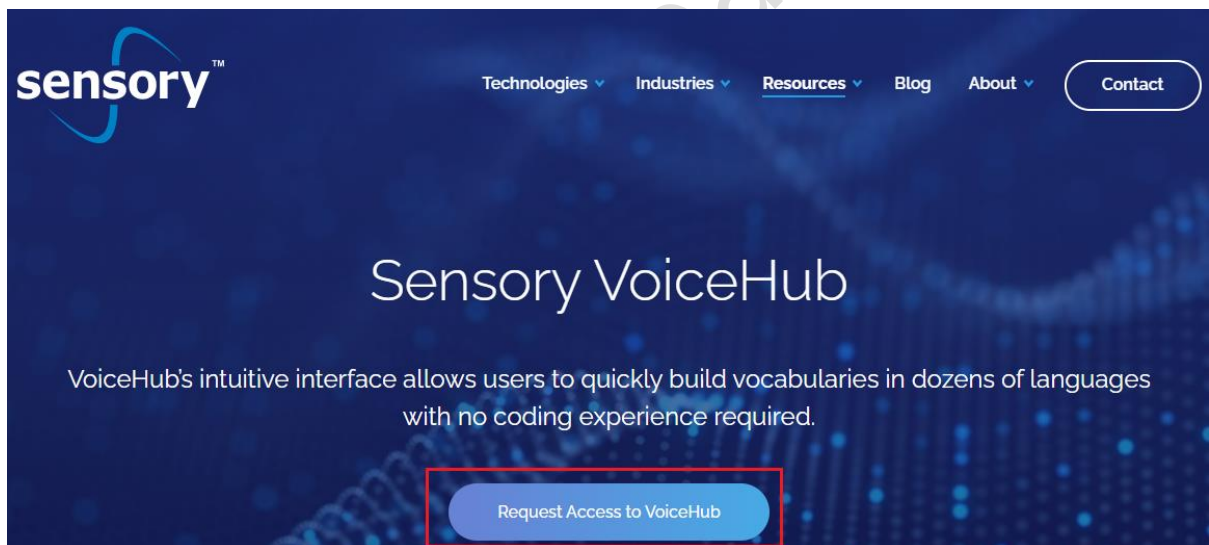


figure 1 1 -2 registration access to VoiceHub

# VoiceHub

You're one step closer to accessing the VoiceHub Portal! Please complete the form below and a member of our team will be in touch shortly.

First name *	Last name *		
<input type="text"/>	<input type="text"/>		
Company Email *	Company name *		
<input type="text" value="Please enter a company email address."/>	<input type="text"/>		
Project Industry *	Country *	State/Region *	
<input type="text" value="Please Select"/>	<input type="text"/>	<input type="text"/>	
VoiceHub Project Type	Format TrulyHandsfree model for a specific DSP or MCU? *		
<input type="text" value="Please Select"/>	<input type="text" value="Please Select"/>		
Project Description *			
<input type="text" value="Please enter a detailed description"/>			

protected by reCAPTCHA  
Privacy - Terms

Submit

각 입력란을 알맞게 기입하고 제출하면 얼마 후 입력한 Company Email 주소로 접속 키를 받게됩니다.

## 1 1 -2 프로젝트 생성하기

'New Project'를 누르면 3가지 유형의 프로젝트를 선택할 수 있습니다.

figure 1 1 -3 New Project

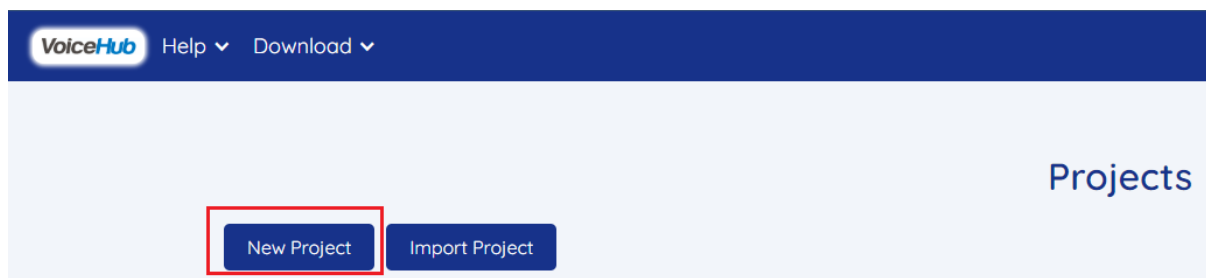
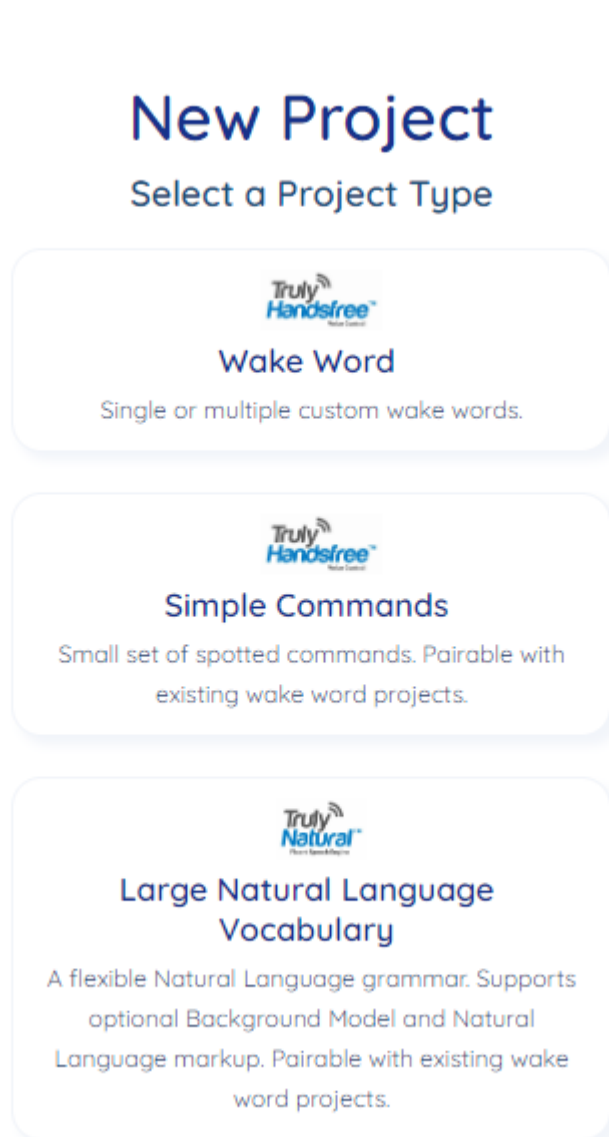


figure 1 1 -4 select a project type



각각의 프로젝트의 용도는 다음과 같습니다.

Wake Word: 슬립 모드에서 깨우는 키워드 생성 프로젝트

Simple Commands: 명령어셋 또는 Wake Word와 명령어셋 AI model 생성 프로젝트

Large Natural Language Vocabulary: 자연어와 유사하게 음성인식하는 AI model 생성 프로젝트

1 1 - 3 Wake Word

Device의 Sleep Mode를 깨우는 Wake Word를 설정하는 곳입니다.

figure 1 1 -5 Wake Word Project

The screenshot shows the 'Wake Word Project' configuration page. At the top, there are buttons for 'Learn More', 'Share', 'Duplicate', and 'Delete'. Below these are input fields for 'Project Name' (containing 'Wake Word') and 'Description' (containing 'Describe task'). The configuration section includes five numbered items: 1. SDK Version (6.21.0), 2. Language (English (USA)), 3. Initial Size (925 KB), 4. Output Format (THF/TNL SDK: snsr file), and 5. Wake Word (a text input field with a '+' button). Below the configuration section are three tabs: '6 Build', '7 Test', and '8 Download'. The 'Build' tab is active, showing a 'Build Status' section with the text 'Not Built', a timer '00:00:00', and two buttons: 'Build' and 'Clean & Rebuild'.

1. Sensory의 Target SDK로 가능한 한 최신버전으로 선택합니다.

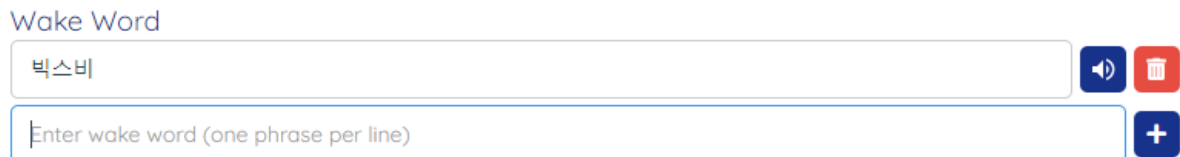
2. Wake Word의 언어를 설정합니다.

3. 초기 음향 모델의 크기입니다. 어휘, 언어 개수 등으로 최종 음향 모델의 크기는 커질 수 있습니다.

4. Target Device를 선택합니다. DSP: XMOS xcore.ai (requires tsilce ...)를 선택하시면 됩니다.

5. wake word로 인식할 단어를 추가합니다. wake word를 입력한 후 우측 '+'를 누르면 추가됩니다.

**figure 1 1 -6 add Wake Word**



6. Wake Word 음향 모델을 빌드합니다.

7. 빌드한 음향 모델을 웹 브라우저에서 테스트할 수 있습니다.

8. 스마트폰에서 테스트하거나 실제 XMOS에 다운로드할 음향 모델을 다운로드할 수 있습니다.

'Download Model'→음향 모델 파일 다운로드, 'Download Application'→스마트폰에 다운로드

#### 1 1 - 4 Simple Commands

Device에 명령을 내릴 단어를 설정하는 곳입니다.

sleep mode가 아니라면 Simple Commands의 있는 word를 wake word로 설정할 수 있습니다.



figure 1 1 -7 Command Set Project

Command Set Project

Learn More Share Duplicate Delete

Project Name  
Command Set

Description  
Describe task

1 SDK Version 2 Language  
6.21.0 English (USA)

3 Wake Word 4 Initial Size  
<None> 925 KB

5 Output Format  
THF/TNL SDK: snsr file

6 Command Set  
Enter command phrase (one phrase per line)

7 Build 8 Test 9 Download

Build Status







Not Built 00:00:00

Build Clean & Rebuild

1. Sensory의 Target SDK로 가능한 한 최신버전으로 선택합니다.
2. Command의 언어를 선택합니다.
3. 선택사항으로, Command를 빌드할 때 같이 빌드할 Wake Word를 선택합니다.
4. 초기 음향 모델의 크기입니다. 어휘, 언어 개수 등으로 최종 음향 모델의 크기는 커질 수 있습니다.
5. Target Device를 선택합니다. DSP: XMOS xcore.ai (requires tsilce ...)를 선택하시면 됩니다.
6. Command로 인식할 단어를 추가합니다. command word를 입력한 후 우측 '+'를 누르면 추가됩니다.

figure 1 1 -8 add command set

Command Set

불켜줘	 
불꺼줘	 
문열어	 
문닫아	 
<input type="text" value="Enter command phrase (one phrase per line)"/> 	

7. Command Set 음향 모델을 빌드합니다. (Wake Word 선택 시 Wake Word도 같이 빌드)
8. 빌드한 음향 모델을 웹 브라우저에서 테스트할 수 있습니다. (Wake Word 선택 시 Wake Word를 Command와 동시에 테스트 가능)
9. 스마트폰에서 테스트하거나 실제 XMOS에 다운로드할 음향 모델을 다운로드할 수 있습니다.  
'Download Model'→음향 모델 파일 다운로드, 'Download Application'→스마트폰에 다운로드

## 1 2 Far-field Voice Local Command에 AI model 적용

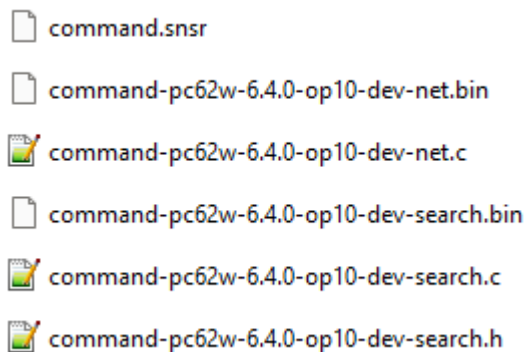
Sensory의 VoiceHub에서 AI model을 만들었다면, 이 AI model을 XCORE에 이식하는 방법입니다.

Far-field Voice Local Command는 wake word를 사용하지 않고, 대신 Command Set 중에서 1개 이상의 keyword를 설정해 wake word처럼 사용할 수 있습니다.

### 1 2 - 1 AI model 파일 복사

VoiceHub에서 받은 command set 압축파일을 풀면 6개의 파일이 있습니다.

**figure 1 2 -1 List of files for only command set**



- command.snsr
- command-pc62w-6.4.0-op10-dev-net.bin
- command-pc62w-6.4.0-op10-dev-net.c
- command-pc62w-6.4.0-op10-dev-search.bin
- command-pc62w-6.4.0-op10-dev-search.c
- command-pc62w-6.4.0-op10-dev-search.h

이 6개의 파일을 WSL로 복사합니다.

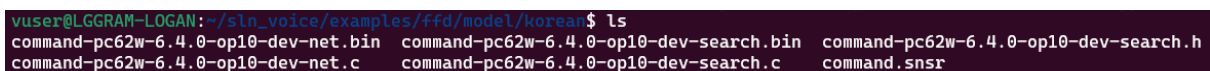
먼저 6개 파일을 저장할 폴더부터 만듭니다.

```
~/sln_voice/examples/ffd/model$ mkdir korean
```

그리고 korean 폴더에 command파일들을 복사합니다.

```
~/sln_voice/examples/ffd/model/korean$ cp -R /mnt/{AI model 파일 위치}
```

**figure 1 2 -2 copy the command files**



```
vuser@LGGRAM-LOGAN:~/sln_voice/examples/ffd/model/korean$ ls
command-pc62w-6.4.0-op10-dev-net.bin  command-pc62w-6.4.0-op10-dev-search.bin  command-pc62w-6.4.0-op10-dev-search.h
command-pc62w-6.4.0-op10-dev-net.c   command-pc62w-6.4.0-op10-dev-search.c   command.snsr
```

## 1 2 - 2 nibble swap 파일 생성

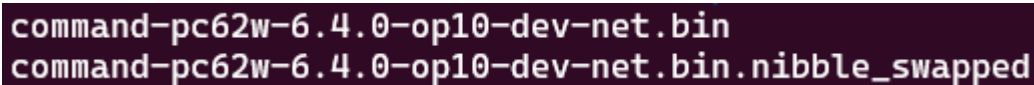
AI model 파일 중 command-pc62w-6.4.0-op10-dev-net.bin의 nibble swap 파일을 생성해야 합니다.

다음 명령으로 nibble swap 파일을 생성할 수 있습니다.

```
~/sln_voice/examples/ffd/model/korean$ nibble_swap command-pc62w-6.4.0-op10-dev-net.bin  
command-pc62w-6.4.0-op10-dev-net.bin.nibble_swapped
```

nibble swap 파일을 구분하기 위해 '.nibble\_swapped'을 붙였습니다.

**figure 1 2 -3 generate nibble swap**



```
command-pc62w-6.4.0-op10-dev-net.bin  
command-pc62w-6.4.0-op10-dev-net.bin.nibble_swapped
```

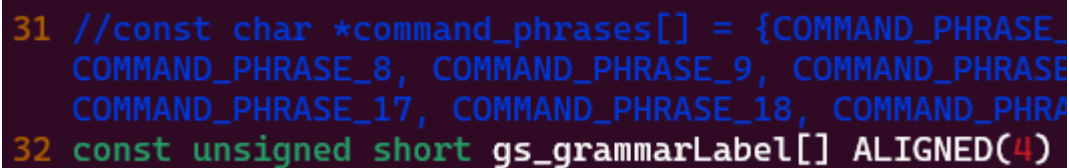
## 1 2 - 3 AI model c파일 수정

```
~/sln_voice/examples/ffd/model/Korean$vi command-pc62w-6.4.0-op10-dev-search.c
```

31줄의 command\_phrases는 주석 처리하고

32줄의 gs\_command\_grammarLabel은 gs\_grammarLabel로 수정합니다.

**figure 1 2 -4 command-pc62w-6.4.0-op10-dev-search.c 수정**



```
31 //const char *command_phrases[] = {COMMAND_PHRASE_  
    COMMAND_PHRASE_8, COMMAND_PHRASE_9, COMMAND_PHRASE_  
    COMMAND_PHRASE_17, COMMAND_PHRASE_18, COMMAND_PHRASE_  
32 const unsigned short gs_grammarLabel[] ALIGNED(4)
```

## 1 2 - 4 UART baud rate 수정

app\_conf.h에서 UART 통신 baud rate를 수정할 수 있습니다. 기본은 9600이고 원하는 값으로 수정합니다.

```
~/sln_voice/examples/ffd/src$ vi app_conf.h
```

86줄 appconfUART\_BAUD\_RATE 9600 → 115200

figure 1 2 -5 modify the baud rate

```
85 #ifndef appconfUART_BAUD_RATE
86 #define appconfUART_BAUD_RATE 115200
87 #endif
```

1 2 - 5 Command와 id의 일치화

소스코드에서 Sensory AI model에 맞춰 keyword와 command의 id를 수정합니다.

25, 26줄을 수정합니다.

~/sl\_n\_voice/examples/ffd/src/intent\_engine\$ vi intent\_engine.c

figure 1 2 -6 define id for keyword and command

```
#define IS_KEYWORD(id) (id == 18)
#define IS_COMMAND(id) (id > 0 && id < 18)
```

VoiceHub에서 Command Set을 입력할 때 최상단은 id가 1이고, 아래로 갈수록 큰 숫자의 id를 가집니다.

figure 1 2 -7 command set and id number

Command Set		
모드변경	id=1	 
멈춰	2	 
시작	3	 
시간올러	4	 
시간내려	5	 
분올러	6	 
분내려	7	 
초올러	8	 
초내려	9	 
일올러	10	 
일내려	11	 
사무실불켜	12	 
사무실불꺼	13	 
내방불켜	14	 
내방불꺼	15	 
선종기켜	16	 
선종기꺼	17	 
박사인	18	 

figure 12-7에서는 wake word라고 할 수 있는 keyword를 가장 높은 id에 배정되었고 나머지는 모

두 command입니다. 이 점을 참고하여 keyword id와 command id를 수정합니다.

1 2 - 6 AI model의 command를 소스코드에 반영

~/sln\_voice/examples/ffd/src/intent\_engine\$ vi intent\_engine\_io.c

23줄에 ASR\_NUMBER\_OF\_COMMANDS를 Command Set의 총 개수로 수정합니다.

figure 1 2 -8 number of commands

```
#define ASR_NUMBER_OF_COMMANDS (18)
```

32줄에 있는 asr\_lut는 VoiceHub에서 작성한 command set과 동일한 순서로 수정합니다.

figure 1 2 -9 asr\_lut array variable

```
{1, 1, "Switch Mode"},  
{2, 2, "Pause the Clock"},  
{3, 3, "Continue the Clock"},  
{4, 4, "Hour up"},  
{5, 5, "Hour down"},  
{6, 6, "Miniute up"},  
{7, 7, "Miniute down"},  
{8, 8, "Second up"},  
{9, 9, "Second down"},  
{10, 10, "Day up"},  
{11, 11, "Day down"},  
{12, 12, "Switch on the office light"},  
{13, 13, "Switch off the office light"},  
{14, 14, "Switch on the myroom light"},  
{15, 15, "Switch off the myroom light"},  
{16, 16, "Switch on the fan"},  
{17, 17, "Switch off the fan"},  
{18, 18, "Bigko"},
```

첫 번째 값은 command의 id값입니다.

두 번째 값은 command를 받으면 소리로 응답하는 wav의 id값입니다.

세 번째 값은 command를 받으면 debug에서 출력할 문장입니다.

asr\_lut에는 command set의 개수만큼 추가하면 됩니다.

## 1 2 - 7 wav 수정하기

먼저 아래 경로에 한국어 전용 폴더를 만듭니다.

```
~/sln_voice/examples/ffd/filesystem_support$mkdir korean
```

이 폴더에 원하는 소리의 wav파일을 복사합니다.

**figure 1 2 -10 response wav**

```
vuser@LGGRAM-LOGAN:~/sln_voice/examples/ffd/filesystem_support/korean$ ls  
11.wav 12.wav 20.wav 21.wav 22.wav 23.wav 24.wav 25.wav 26.wav 27.wav 28.wav 29.wav 30.wav 31.wav 32.wav 33.wav 34.wav 50.wav 52.wav
```

wav의 성능은 sampling rate는 16bit PCM, 16000Hz입니다.

XK-VOICE-L71의 Flash Memory는 2MB로 wav를 많이 넣을 수 없으니 참고 바랍니다.

그리고 추가한 wav파일에 대해 소스코드에 적용해야 합니다.

```
~/sln_voice/examples/ffd/src/intent_handler/audio_response$ vi audio_response.c
```

22줄의 audio\_files\_en에 wav파일명을 수정하면 됩니다.

figure 1 2 -11 apply wav file name to source code

```
"50.wav" // sleep //0
"20.wav" // 모드 변경 //1
"21.wav" // 멈춰
"22.wav" // 시작
"23.wav" // 시간 올려
"24.wav" // 시간 내려 //5
"25.wav" // 분 올려
"26.wav" // 분 내려
"27.wav" // 초 올려
"28.wav" // 초 내려
"29.wav" // 일 올려 //10
"30.wav" // 일 내려
"31.wav" // 사무실 불 켜
"32.wav" // 사무실 불 꺼
"33.wav" // 내방 불 켜
"34.wav" // 내방 불 꺼 //15
"11.wav" // 선풍기 켜
"12.wav" // 선풍기 꺼
"52.wav" // 빅코 //18
```

첫 번째는 id가 0이며, 이 wav는 command 시간이 끝나고 대기로 진입할 때 나오는 소리입니다.

두 번째부터는 id가 1부터 시작하고, command에 따라서 지정된 wav가 재생됩니다. 이 순서는 intent\_engine\_io.c의 asr\_lut에서 어떤 wav를 재생할 것인지 지정한 wav의 id입니다.

#### 1 2 -8 cmake 수정

빌드할 때 복사한 AI model을 찾을 수 있도록 경로를 추가해 줍니다.

```
~/sln_voice/examples/ffd$ vi ffd.cmake
```

```
set(MODEL_LANGUAGE "korean")
```

```
set(SENSORY_COMMAND_SEARCH_HEADER_FILE
```

```
"${FFD_SRC_ROOT}/model/${MODEL_LANGUAGE}/command-pc62w-6.4.0-op10-dev-search.h")
```

```
set(SENSORY_COMMAND_SEARCH_SOURCE_FILE
```

```
"${FFD_SRC_ROOT}/model/${MODEL_LANGUAGE}/command-pc62w-6.4.0-op10-dev-search.c")
```



```
set(SENSORY_COMMAND_NET_FILE "${FFD_SRC_ROOT}/model/${MODEL_LANGUAGE}/command-  
pc62w-6.4.0-op10-dev-net.bin.nibble_swapped")
```

**figure 1 2 -12 modify cmake file**

```
#set(MODEL_LANGUAGE "english_usa")  
#set(SENSORY_COMMAND_SEARCH_HEADER_FILE "${FFD_SRC_ROOT}/model/${MODEL_LANGUAGE}/command-pc62w-6.4.0-op10-prod-search.h")  
#set(SENSORY_COMMAND_SEARCH_SOURCE_FILE "${FFD_SRC_ROOT}/model/${MODEL_LANGUAGE}/command-pc62w-6.4.0-op10-prod-search.c")  
#set(SENSORY_COMMAND_NET_FILE "${FFD_SRC_ROOT}/model/${MODEL_LANGUAGE}/command-pc62w-6.4.0-op10-prod-net.bin.nibble_swapped")  
set(MODEL_LANGUAGE "korean")  
set(SENSORY_COMMAND_SEARCH_HEADER_FILE "${FFD_SRC_ROOT}/model/${MODEL_LANGUAGE}/command-pc62w-6.4.0-op10-dev-search.h")  
set(SENSORY_COMMAND_SEARCH_SOURCE_FILE "${FFD_SRC_ROOT}/model/${MODEL_LANGUAGE}/command-pc62w-6.4.0-op10-dev-search.c")  
set(SENSORY_COMMAND_NET_FILE "${FFD_SRC_ROOT}/model/${MODEL_LANGUAGE}/command-pc62w-6.4.0-op10-dev-net.bin.nibble_swapped")
```

## 1 3 Low Power Far-field Voice Local Command에 AI model 적용

절전 모드 기능이 들어간 Far-field Voice Local Command입니다. 절전 모드에서 깨우기 위해서 wake word를 사용하며 VoiceHub에서 빌드한 wake word와 command set을 모두 이식해야 합니다.

### 1 3 - 1 AI model 파일 복사

VoiceHub에서 받은 wake word와 command set의 압축파일을 풀면 총 12개의 파일이 있습니다.

figure 1 3 -1 wake word files and command files

wakeword.snsr	command.snsr
wakeword-pc60w-7.0.0-op10-dev-net.bin	command-pc62w-6.4.0-op10-dev-net.bin
wakeword-pc60w-7.0.0-op10-dev-net.c	command-pc62w-6.4.0-op10-dev-net.c
wakeword-pc60w-7.0.0-op10-dev-search.bin	command-pc62w-6.4.0-op10-dev-search.bin
wakeword-pc60w-7.0.0-op10-dev-search.c	command-pc62w-6.4.0-op10-dev-search.c
wakeword-pc60w-7.0.0-op10-dev-search.h	command-pc62w-6.4.0-op10-dev-search.h

이 12개 파일을 WSL로 복사합니다.

먼저 이 12개를 아래 경로에 저장합니다.

```
~/sln_voice/examples/low_power_ffd/model$
```

figure 1 3 -2 copy the wake word and command files

```
vuser@LGGRAM-LOGAN: ~/sln_voice/examples/low_power_ffd/model$ ls
command-pc62w-6.4.0-op10-dev-net.bin      command-pc62w-6.4.0-op10-dev-search.c  wakeword-pc60w-7.0.0-op10-dev-net.bin  wakeword-pc60w-7.0.0-op10-dev-search.h
command-pc62w-6.4.0-op10-dev-net.bin.nibble_swapped  command-pc62w-6.4.0-op10-dev-search.h  wakeword-pc60w-7.0.0-op10-dev-net.c    wakeword.snsr
command-pc62w-6.4.0-op10-dev-net.c        command.snsr                            wakeword-pc60w-7.0.0-op10-dev-search.bin
command-pc62w-6.4.0-op10-dev-search.bin  wakeword-pc60w-7.0.0-op10-dev-search.c
```

### 1 3 - 2 nibble swap 파일 생성

AI model 파일 중 command-pc62w-6.4.0-op10-dev-net.bin의 nibble swap 파일을 생성해야 합니다.

다음 명령으로 nibble swap 파일을 생성할 수 있습니다.

```
~/sln_voice/examples/low_power_ffd/model$ nibble_swap command-pc62w-6.4.0-op10-dev-net.bin
```

command-pc62w-6.4.0-op10-dev-net.bin.nibble\_swapped

nibble swap 파일을 구분하기 위해 '.nibble\_swapped'을 붙였습니다.

**figure 1 3 -3 generate nibble swap**

```
command-pc62w-6.4.0-op10-dev-net.bin
command-pc62w-6.4.0-op10-dev-net.bin.nibble_swapped
```

### 1 3 - 3 UART baud rate 수정

app\_conf.h에서 UART 통신 baud rate를 수정할 수 있습니다. 기본은 9600이고 원하는 값으로 수정합니다.

~/sln\_voice/examples/low\_power\_ffd/src\$ vi app\_conf.h

83줄 appconfUART\_BAUD\_RATE 9600 → 115200

**figure 1 3 -4 modify the baud rate**

```
82 #ifndef appconfUART_BAUD_RATE
83 #define appconfUART_BAUD_RATE 9600
84 #endif
```

### 1 3 - 4 AI model의 command를 소스코드에 반영

~/sln\_voice/examples/low\_power\_ffd/src/intent\_engine\$ vi intent\_engine\_io.c

26줄에 ASR\_NUMBER\_OF\_COMMANDS를 Command Set의 총 개수로 수정합니다.

**figure 1 3 -5 number of commands**

```
25 // look up table to converting ASR IDs to wav file IDs or strings
26 #define ASR_NUMBER_OF_COMMANDS (16)
```

34줄에 있는 asr\_lut는 VoiceHub에서 작성한 command set과 동일한 순서로 수정합니다.

figure 1 3 -6 asr\_lut array variable

```
{1, "Switch on the TV"},  
{2, "Switch off the TV"},  
{3, "Channel up"},  
{4, "Channel down"},  
{5, "Volume up"},  
{6, "Volume down"},  
{7, "Switch on the lights"},  
{8, "Switch off the lights"},  
{9, "Brightness up"},  
{10, "Brightness down"},  
{11, "Switch on the fan"},  
{12, "switch off the fan"},  
{13, "Speed up the fan"},  
{14, "Slow down the fan"},  
{15, "Set highter temperature"},  
{16, "Set lower temperature"}
```

첫 번째 값은 command의 id입니다.

두 번째 값은 debug에서 출력할 문장입니다.

asr\_lut에는 command set의 개수만큼 추가하면 됩니다.

1 3 - 5 AI model wake word를 소스코드에 반영

wake word는 1개만 사용할 수 있습니다.

VoiceHub에서 wake word를 생성할 때 1개만 등록합니다.

~/sln\_voice/examples/low\_power\_ffd/src/wakeword\$ vi wakeword.c

43줄에 변경한 단어로 변경합니다. WAKEWORD\_PHRASE의 문장은 debug에서 출력합니다.

figure 1 3 -7 modify wake word

```
43 #define WAKEWORD_PHRASE    "Bigshine"  
44 #define WAKEWORD_ID        0x1  
45 #define IS_WAKEWORD(id)    ((id) == (WAKEWORD_ID))
```

## Revision History

Owner	Description	Date	version
Logan Lee	1. 초기 문서 작성	Sep 6, 2023	1.0
Logan Lee	1. 제목 변경 2. 사진 캡션 및 그림 목록 추가 3. 필수 패키지 'libfuse2' 추가 4. 7, 8, 9장이 8, 9, 10장으로 변경 5. '7장 XK-VOICE-L71와 XTAG4와 PC간 연결' 추가 6. '9-2 Windows에서 XTAG 인식 및 연결' 추가 7. 10장 제목 수정 및 내용 추가 8. '11장 Sensory AI model 생성' 추가 9. '12장 Far-field Voice Local Command에 AI model 적용' 추가 10. '13장 Low Power Far-field Voice Local Command에 AI model 적용' 추가	Dec 01, 2023	2.0