TOPST AI-G Linux SDK

Getting Started

Rev. 1.01 [G]
2025-07-18

TABLE OF CONTENTS

Contents

Figures

Tables

1 INTRODUCTION

This document provides guidelines for building the AI-G SDK, including setting up the host environment, building the SDK, and using the firmware downloader.

This document includes information on the following:

- Setting Host Environment
- Image Build Guide
- Firmware Downloader Guide

2 SETTING HOST ENVIRONMENT

This chapter provides instructions on how to set up the host PC environment, with separate guides for setting up on Windows and Ubuntu.

2.1 Windows Environment

This document describes how to set up Windows Subsystem for Linux (WSL) to use Linux on a Windows PC. The AI-G SDK is based on the Yocto Project, so the Linux version of AI-G SDK follows the Yocto project. You can install another version of Linux, but in this document, AI-G SDK is described based on Ubuntu 22.04. If the OS of your host PC is Ubuntu, proceed to Chapter 2.2.

2.1.1 Install WSL2 Ubuntu (Windows Environment Only)

1. Set Windows Features by clicking **Control Panel → Programs → Windows Features On/Off → Enable Virtual Machine Platform & Hyper-V**.

2. Execute Windows PowerShell with "**Run with administrator privileges**".

3. Enable the WSL2 System.

```
dism.exe /online /enable-feature /featurename:Microsoft-Windows-Subsystem-Linux /all /norestart
```

4. Enable the Virtual Machine feature.

```
dism.exe /online /enable-feature /featurename:VirtualMachinePlatform /all /norestart
```

5. Set WSL to the default version of 2 (WSL2).

```
wsl --set-default-version 2
```

6. Search for Ubuntu 22.04.3 LTS in Microsoft Store and download it.

   - If you need to download the Linux kernel update package, download the latest package here.

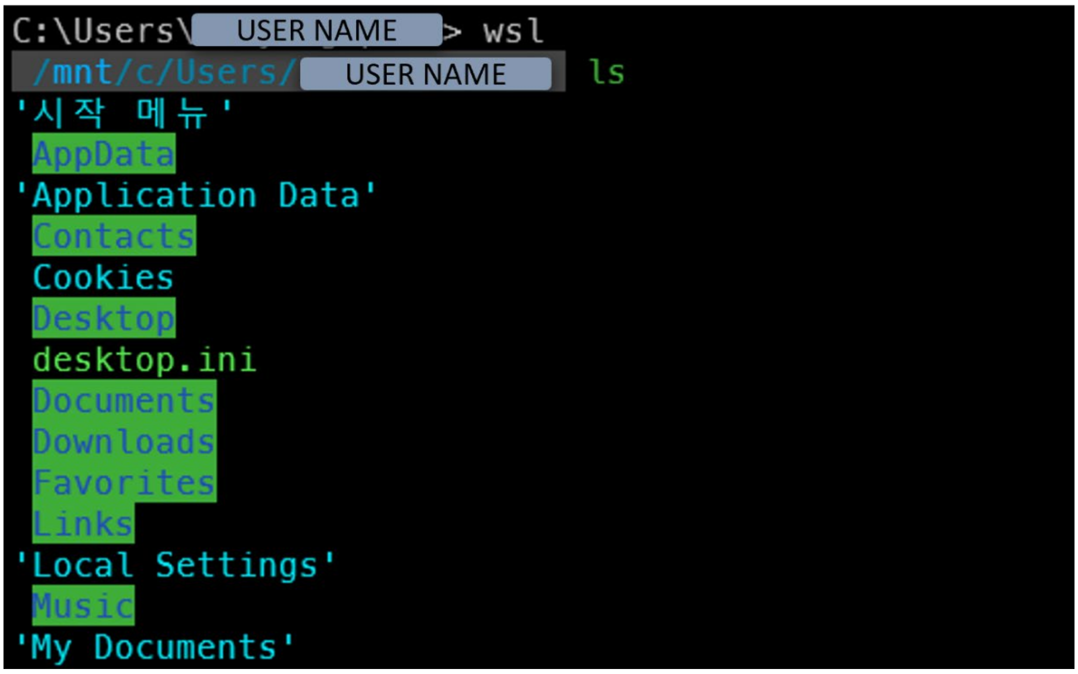
7. Choose any username during the Ubuntu installation.

2.1.2 Access Ubuntu Through WSL2

Open the Windows Command Prompt and enter the following command to access Ubuntu. When you access Ubuntu, it starts in the "/mnt/c/Users/[user name]" directory by default.

```
wsl  // access ubuntu
ls   // check contents in your directory
```

Refer to Figure 2.1 to check the result (results may vary depending on your system).



```
C:\Users\ USER NAME > wsl
/mnt/c/Users/ USER NAME  ls
'시작 메뉴'
 AppData
'Application Data'
 Contacts
 Cookies
 Desktop
 desktop.ini
 Documents
 Downloads
 Favorites
 Links
'Local Settings'
 Music
'My Documents'
```

Figure 2.1 WSL2 Screenshot

2.1.3 Install SSH and Samba

If SSH and Samba are already installed or you are not going to use them, you can skip this chapter.

After entering Ubuntu, you can use additional utilities such as SSH and Samba for a more convenient development environment. SSH and Samba allow you to execute commands on remote computers and copy files to other computers.

The following command requires the Host PC to be connected to the network. Check your network condition by using the following commands:

```
$ sudo apt-get update
$ sudo apt-get upgrade
$ ifconfig
```

Use the following command to install net-tools, SSH, and Samba.

```
$ sudo apt-get update
$ sudo apt install -y net-tools openssh-server samba
```

After installing SSH and Samba, you should set each program to your environment.

2.1.4 Install Utilities

Use the following commands to simultaneously install all required utilities. To use Yocto Project, the following utilities must be installed on the Host PC (personal computer or development server).

```
$ sudo apt-get install -y gawk wget git diffstat unzip texinfo gcc-multilib build-essential chrpath

$ sudo apt-get install -y socat cpio python3 python3-pip python3-pexpect xz-utils debianutils

$ sudo apt-get install -y iputils-ping python3-git python3-jinja2 libegl1-mesa libsdl1.2-dev pylint

$ sudo apt-get install -y xterm zstd ncftp curl git-lfs vim zip
```

2.1.5 Install Repo

You can download AI-G SDK by using ***Repo***.
To install ***Repo***, refer to the following website:

- https://source.android.com/source/downloading.html.

If ***Repo*** is already installed, you can use it without re-installing.
Before installing ***Repo***, make sure that Python version 3.6 or higher is installed.

Use the following command to install ***Repo***.

```
$ sudo apt-get install repo
```

If you see the error message '/usr/bin/env 'python' no such file or directory', use the following command to link **python** to **python3**.

```
$ sudo ln -sf /usr/bin/python3 /usr/bin/python
```

If there is a ***Repo*** error, use the following commands to download the latest version and place it in the /usr/bin/ folder.

```
$ mkdir -p ~/bin

$ curl http://commondatastorage.googleapis.com/git-repo-downloads/repo > ~/bin/repo

$ chmod a+x ~/bin/repo

$ sudo mv ~/bin/repo /usr/bin/repo
```

Proceed to Chapter 3.

2.2 Linux Environment

This chapter describes the setup process for Ubuntu as the host OS.

2.2.1 Setting Environment

Chapter 2.2.2 to Chapter 2.2.5 must be executed in the Ubuntu terminal. To open the terminal, use the shortcut **[Ctrl+Alt+T]**.

2.2.2 Install SSH and Samba

If SSH and Samba are already installed or you are not going to use them, you can skip this chapter.

After entering Ubuntu, you can use additional utilities such as SSH and Samba for a more convenient development environment. SSH and Samba allow you to execute commands on remote computers and copy files to other computers.

The following command requires the Host PC to be connected to the network. Check your network condition by using the following commands:

```
$ sudo apt-get update
$ sudo apt-get upgrade
$ ifconfig
```

Use the following command to install SSH and Samba:

```
$ sudo apt-get update
$ sudo apt install -y openssh-server samba net-tools
```

After installing SSH and Samba, you should set each program to your environment.

2.2.3 Install Utilities

Use the following commands to simultaneously install all required utilities. To use Yocto Project, the following utilities must be installed on Host PC (personal computer or development server).

```
$ sudo apt-get install -y gawk wget git diffstat unzip texinfo gcc-multilib build-essential chrpath

$ sudo apt-get install -y socat cpio python3 python3-pip python3-pexpect xz-utils debianutils

$ sudo apt-get install -y iputils-ping python3-git python3-jinja2 libegl1-mesa libsdl1.2-dev pylint

$ sudo apt-get install -y xterm zstd ncftp curl git-lfs vim zip
```

2.2.4 Install Repo

You can download the AI-G SDK by using ***Repo***.
To install ***Repo***, refer to the following website:

- https://source.android.com/source/downloading.html.

If ***Repo*** is already installed, you can use it without re-installing.
Before installing Repo, make sure Python version 3.6 or higher is properly set.

Use the following command to install Repo.

```
sudo apt-get install repo
```

If you see the error message '/usr/bin/env 'python' no such file or directory', execute the following command to link **python** to **python3**.

```
sudo ln -sf /usr/bin/python3 /usr/bin/python
```

If there is a ***Repo*** error, use the following commands to download the latest version and place it in the /usr/bin/ folder.

```
mkdir -p ~/bin && \
curl http://commondatastorage.googleapis.com/git-repo-downloads/repo &gt;~/bin/repo && \
chmod a+x ~/bin/repo && \
sudo mv ~/bin/repo /usr/bin/repo && \
```

2.2.5 Udev Rules for Telechips USB Device (Optional)

After you execute the following command, you no longer need to use **sudo** command to download ***FWDN*** in Linux (Host PC).

```
echo  "SUBSYSTEM==\"usb\",  ATTR{idVendor}==\"140e\",  MODE=\"0666\",  OWNER=\"${USER}\""  |  sudo  tee
/etc/udev/rules.d/99-topst.rules && \
sudo udevadm control --reload-rules && sudo udevadm trigger
```

3 BUILD GUIDE

This chapter provides guidance based on the Ubuntu OS installed on the host PC (regardless of whether it is WSL or a local Ubuntu installation). The image to be uploaded to the AI-G is built using the Yocto Project, and therefore the process must be carried out in the Ubuntu environment.

3.1 SDK Build Preparation

The AI-G SDK is based on Yocto Project 4.0 Kirkstone. Therefore, you must configure the Yocto Project environment on the Host PC to use the AI-G SDK. To download SDK, source-mirror, and tools, you must install utilities. To build the image smoothly, your PC must have at least 60 GB of available storage and a minimum of 16 GB of RAM.

3.2 Yocto Project

The Yocto Project is an open-source project that focuses on embedded Linux development.
It uses a combination of Poky, which is a part of the Open Embedded project, and ***bitbake*** as the build system to make Linux images.
By using Yocto Project, you can simultaneously build the bootloader, kernel, and rootfs.

3.3 Task Process

Figure 3.1 shows the task process of the Yocto Project. You can download the source code from upstream repositories based on metadata and then build it. After the build is completed, package, image, and SDK are provided as results.

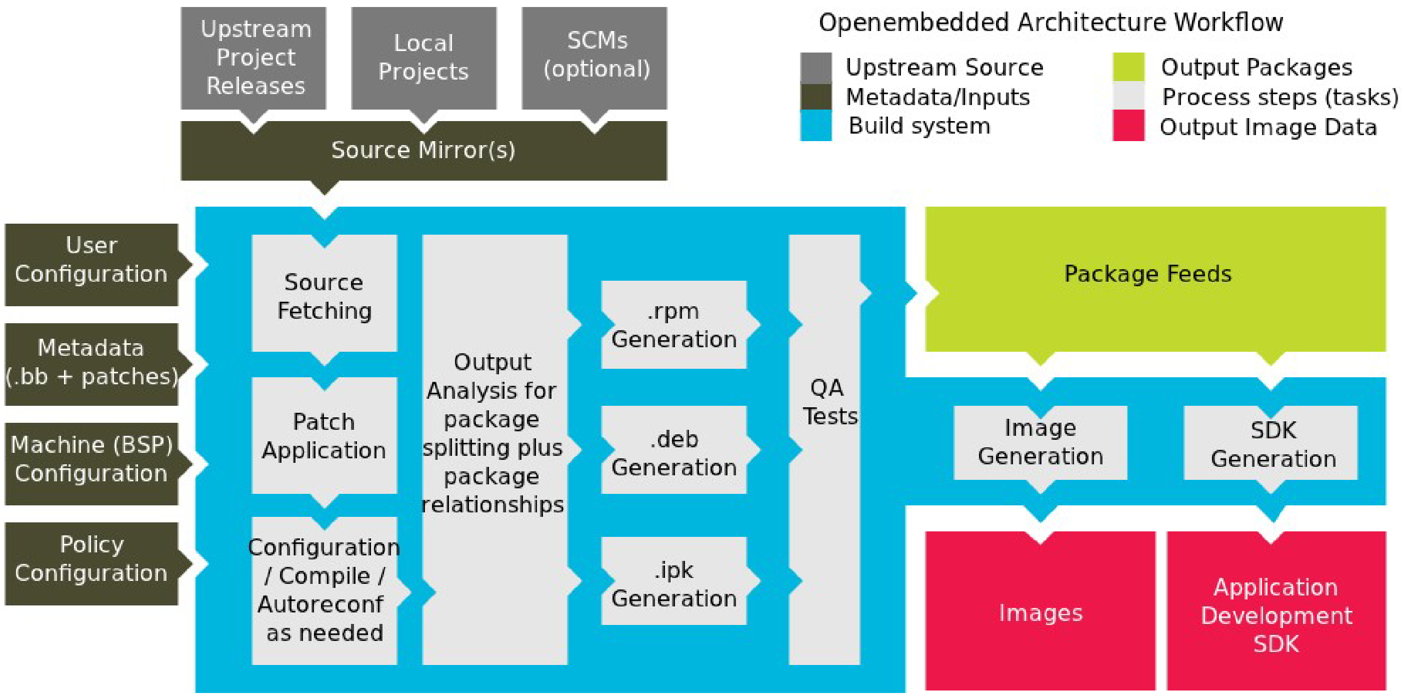


Figure 3.1 Yocto Project Task Process

3.4 Composition of AI-G SDK

After AI-G SDK is downloaded, the directory configuration is as follows.

Table 3.1 Composition of AI-G SDK

Item		Description
easy-setup.sh		Python script to automatically download and build the SDK
stitch-fai_ai.sh		Script for making AI-G fai images (minimal + Sample Application) **Note:** This is a symbolic link to "tools/ai-g/stitch-fai.sh".
stitch-fai_d3.sh		Script for making D3-G fai images (minimal + Sample Application) **Note:** This is a symbolic link to "tools/d3-g/stitch-fai.sh".
mktcimg		Tools related to the build process and ***FWDN***
tools		
poky	poky	Yocto Project 4.0 Kirkstone build system
	meta-openembedded	Layer that supports OE-Core
	meta-arm	Layer that supports Arm toolchain
	meta-telechips-ai-bsp	Layer that supports TOPST BSP
	meta-gplv2	Layer that supports packages that avoid GPLv3 license
	meta-topst	TOPST recipe

3.5 Ready to build

The following chapters describe how to configure the Yocto Project to build the AI-G image.

3.5.1 Set Email and Username in .gitconfig

To download the Yocto image for AI-G from the official TOPST git, configure your email and username.

1. Enter the following command to open the Git configuration file for your user account.

```
vi ~/.gitconfig
```

2. Enter the following information to configure your email and username.

```
[user]
    email = User email
    name = User name
```

3.5.2 Get AI-G SDK from Git

1. Create a new directory named **`topst-sdk`** and change the current directory to **`topst-sdk`**.

```
mkdir topst-sdk
cd topst-sdk
```

2. Execute the following command to initialize the repository.

```
 $ repo init -u https://github.com/topst-development/manifests.git -m linux_yp4.0_topst.xml
```

After executing the command, the following output is displayed.

```
 Downloading Repo source from https://gerrit.googlesource.com/git-repo

... A new version of repo (2.54) is available.
... New version is available at: /home/topst/topst-sdk/.repo/repo/repo
... The launcher is run from: /usr/bin/repo
!!! The launcher is not writable.  Please talk to your sysadmin or distro
!!! to get an update installed.


Your identity is: TopstDeveloper <topstdeveloper@gmail.com>
If you want to change this, please re-run 'repo init' with --config-name

repo has been initialized in /home/topst/topst-sdk
```

3. Execute the following command to synchronize the repository.

```
$ repo sync
```

After executing the command, the following output is displayed.

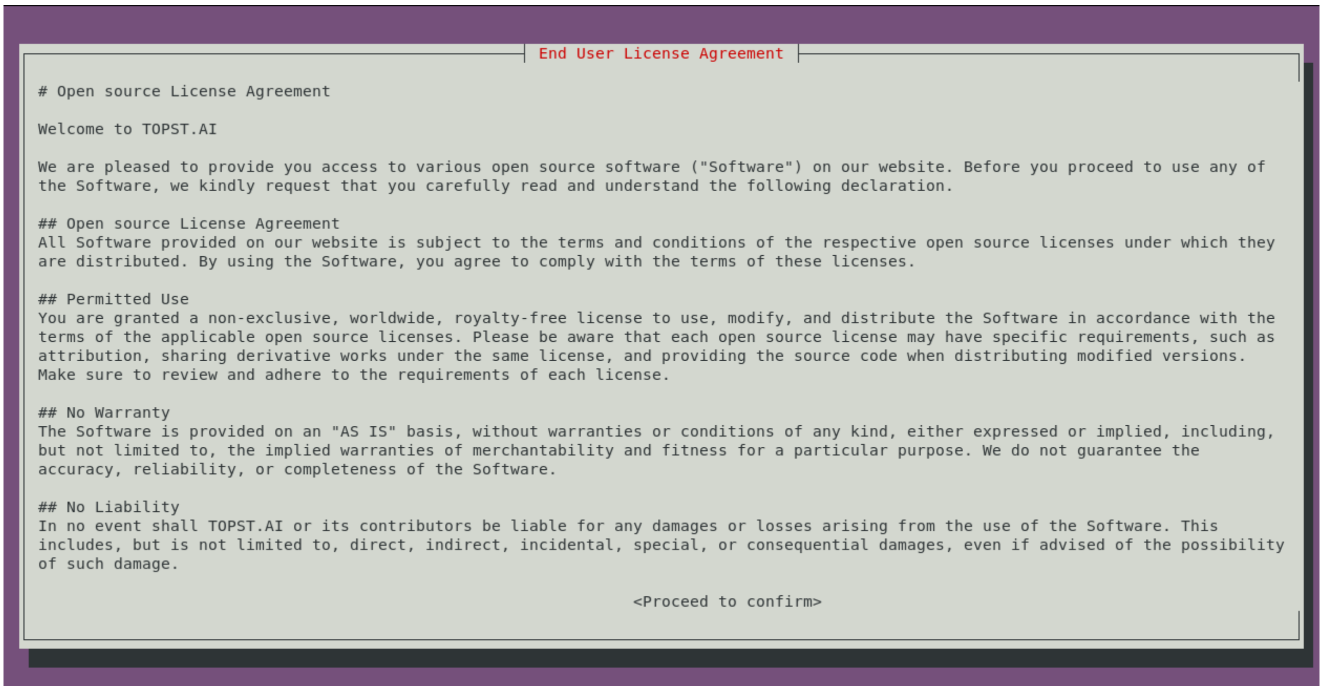
```
... A new version of repo (2.50) is available.
... New version is available at: /home/TOPST/topst-sdk/.repo/repo/repo
... The launcher is run from: /usr/bin/repo
!!! The launcher is not writable.  Please talk to your sysadmin or distro
!!! to get an update installed.

 Fetching: 100% (12/12), done in 33.103s
Checking out:  25% (3/12), done in 0.863s
Checking out:  75% (9/12), done in 0.415s
repo sync has finished successfully.
```

3.6 Execute Build Script

If you run ./**easy-setup_ai.sh** script, you can see the following screen.

Caution: If you re-run ./**easy-setup_ai.sh**, be careful as the built sources will be deleted if you select **yes**.

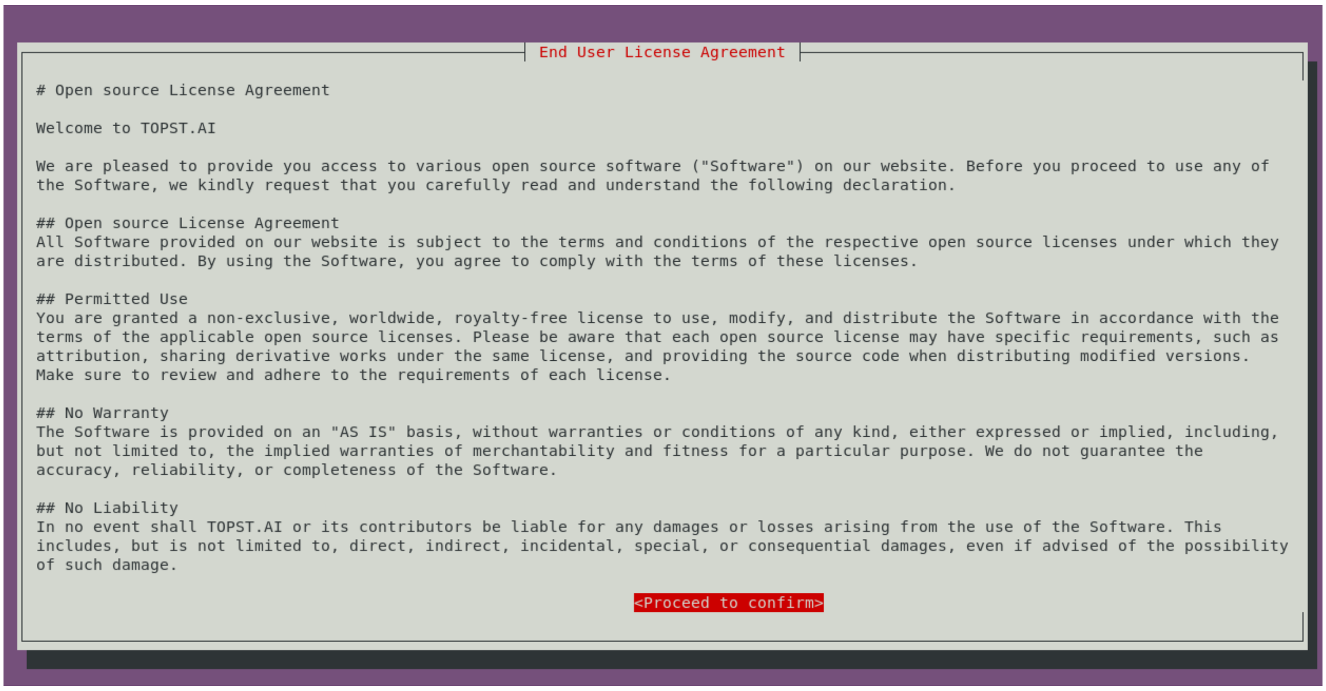
```
./easy-setup_ai.sh
```



```
End User License Agreement

# Open source License Agreement

Welcome to TOPST.AI

We are pleased to provide you access to various open source software ("Software") on our website. Before you proceed to use any of
the Software, we kindly request that you carefully read and understand the following declaration.

## Open source License Agreement
All Software provided on our website is subject to the terms and conditions of the respective open source licenses under which they
are distributed. By using the Software, you agree to comply with the terms of these licenses.

## Permitted Use
You are granted a non-exclusive, worldwide, royalty-free license to use, modify, and distribute the Software in accordance with the
terms of the applicable open source licenses. Please be aware that each open source license may have specific requirements, such as
attribution, sharing derivative works under the same license, and providing the source code when distributing modified versions.
Make sure to review and adhere to the requirements of each license.

## No Warranty
The Software is provided on an "AS IS" basis, without warranties or conditions of any kind, either expressed or implied, including,
but not limited to, the implied warranties of merchantability and fitness for a particular purpose. We do not guarantee the
accuracy, reliability, or completeness of the Software.

## No Liability
In no event shall TOPST.AI or its contributors be liable for any damages or losses arising from the use of the Software. This
includes, but is not limited to, direct, indirect, incidental, special, or consequential damages, even if advised of the possibility
of such damage.

                                                              <Proceed to confirm>
```

Figure 3.2 End User License Agreement

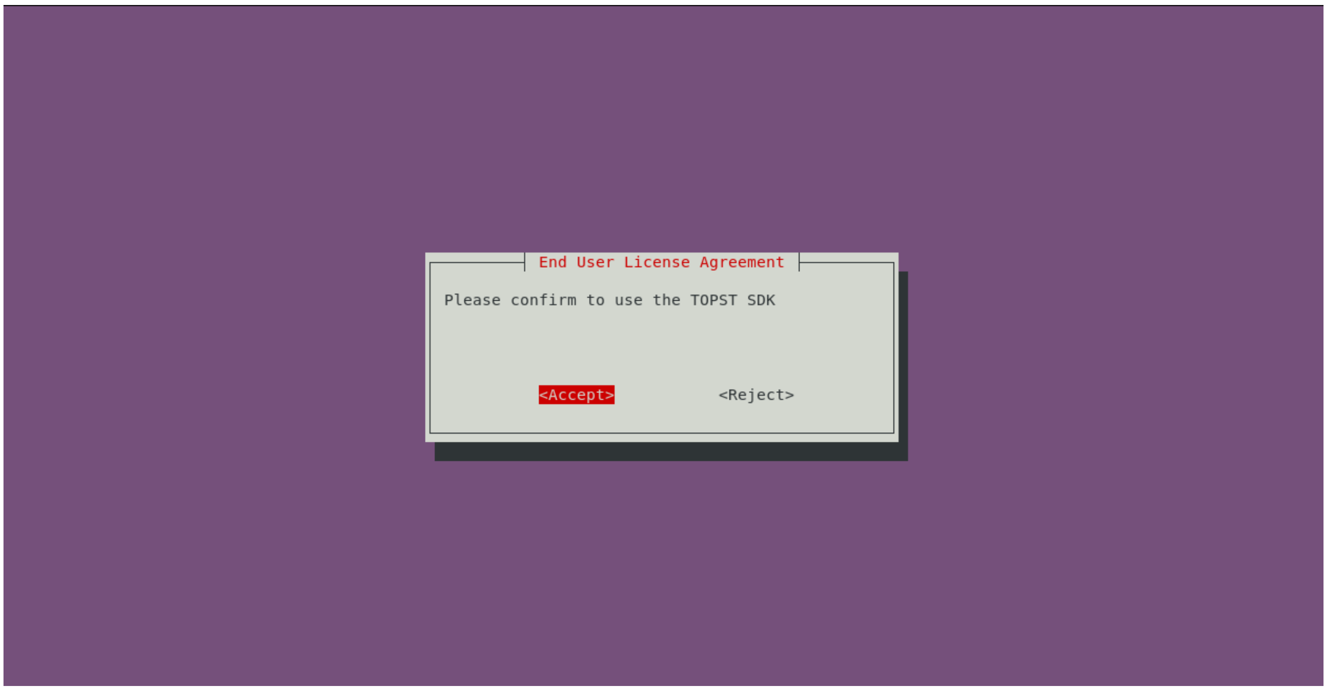
Scroll down to the bottom of the screen and read the following notice. After you read this notice, press the right arrow key and **[Enter]**.



```
End User License Agreement

# Open source License Agreement

Welcome to TOPST.AI

We are pleased to provide you access to various open source software ("Software") on our website. Before you proceed to use any of
the Software, we kindly request that you carefully read and understand the following declaration.

## Open source License Agreement
All Software provided on our website is subject to the terms and conditions of the respective open source licenses under which they
are distributed. By using the Software, you agree to comply with the terms of these licenses.

## Permitted Use
You are granted a non-exclusive, worldwide, royalty-free license to use, modify, and distribute the Software in accordance with the
terms of the applicable open source licenses. Please be aware that each open source license may have specific requirements, such as
attribution, sharing derivative works under the same license, and providing the source code when distributing modified versions.
Make sure to review and adhere to the requirements of each license.

## No Warranty
The Software is provided on an "AS IS" basis, without warranties or conditions of any kind, either expressed or implied, including,
but not limited to, the implied warranties of merchantability and fitness for a particular purpose. We do not guarantee the
accuracy, reliability, or completeness of the Software.

## No Liability
In no event shall TOPST.AI or its contributors be liable for any damages or losses arising from the use of the Software. This
includes, but is not limited to, direct, indirect, incidental, special, or consequential damages, even if advised of the possibility
of such damage.

                                                              <Proceed to confirm>
```

Figure 3.3 Go to 'Proceed to confirm'

Then you can see the following screen.



End User License Agreement

Please confirm to use the TOPST SDK

Figure 3.4 Accept Screen

If you select Accept by pressing **[Enter]**, you can build the AI-G SDK image by using the following command.

`topst-build.sh` is a shell script that sets up the core environment required to build images for D3-G and AI-G boards. Execute the following commands. By selecting option 1, you are now ready to install the main OS on the AI-G board.

```
$ source poky/meta-topst/topst-build.sh

Choose MACHINE
  1. ai-g-topst
  2. d3-g-topst-main
  3. d3-g-topst-sub
  4. d5-g-topst-main
  5. d5-g-topst-sub
select number(1-5) => 1
machine(ai-g-topst) selected.
You had no conf/local.conf file. This configuration file has therefore been
created for you from /home/topst/topst-sdk/poky/meta-topst/template/ai-g-topst/local.conf.sample
You may wish to edit it to, for example, select a different MACHINE (target
hardware). See conf/local.conf for more information as common configuration
options are commented.

You had no conf/bblayers.conf file. This configuration file has therefore been
created for you from /home/topst/topst-sdk/poky/meta-topst/template/ai-g-topst/bblayers.conf.sample
To add additional metadata layers into your configuration please add entries
to conf/bblayers.conf.

The Yocto Project has extensive documentation about OE including a reference
manual which can be found at:
    https://docs.yoctoproject.org

For more information about OpenEmbedded see the website:
    https://www.openembedded.org/

Yocto Project common targets are:
    core-image-minimal
    core-image-sato
    meta-toolchain
    adt-installer
    meta-ide-support
```

```


Telechips common targets are:
    telechips-topst-ai-image

    meta-toolchain-topst(Application Development Toolkit)


You can also run generated TOPST images on AI-G board

Other commonly useful commands are:
- 'devtool' and 'recipetool' handle common recipe tasks
- 'bitbake-layers' handles common layer tasks
- 'oe-pkgdata-util' handles common target package tasks
```

Now, you can build AI-G image by using the following command.

```
$ bitbake telechips-topst-ai-image
```

The build image is created in the following path:

- {TOPST_PATH}/build/ai-g-topst/tmp/deploy/images/ai-g-topst

3.7 Make Firmware Downloader (FWDN) Image

This option combines the binaries generated after building the SDK into a single image for the AI-G board.

The "output_nd.fwdn.zip" including the output_nd.fai build image and ***FWDN*** tool is created in the following path:

- ~/topst-sdk/

```
$ cd ~/topst-sdk/ && \
$ ./stitch-fai_ai.sh -f


[mktcimg]  v1.4.3  -  Apr  2  2024  14:17:19  ==========  Parsing  Partition  List==========  PART
1====================  name  :  bl3_ca53_a  size  :  4096  sector(2097152  byte)  location  :  build/ai-g-
topst/tmp/deploy/images/ai-g-topst/u-boot.rom PART 2==================== name : bl3_ca53_b size : 4096
sector(2097152   byte)   location   :   build/ai-g-topst/tmp/deploy/images/ai-g-topst/u-boot.rom   PART
3====================  name  :  boot  size  :  81920  sector(41943040  byte)  location  :  build/ai-g-
topst/tmp/deploy/images/ai-g-topst/tc-boot-ai-g-topst.img PART 4==================== name : system size :
8388608  sector(4294967296  byte)  location  :  build/ai-g-topst/tmp/deploy/images/ai-g-topst/telechips-
topst-ai-image-ai-g-topst.ext4  PART  5====================  name  :  dtb  size  :  400  sector(204800  byte)
location : build/ai-g-topst/tmp/deploy/images/ai-g-topst/tcc7500-lpd4x321.dtb PART 6====================
name  :  env  size  :  2048  sector(1048576  byte)  location  :  (NULL)  PART  7====================  name  :  misc
size : 2048 sector(1048576 byte) location : (NULL) uuid : e1345e29-c2d9-48c6-b6f6-e79f951075b6 , part-
name : bl3_ca53_a, first_lba : 34, last_lba : 4129 uuid : 8514440e-a994-4bfd-bf78-9cff3a2a3ede , part-
name : bl3_ca53_b, first_lba : 4130, last_lba : 8225 uuid : df1ebb96-4bc2-4b04-bd48-57e6d362e901 , part-
name : boot, first_lba : 8226, last_lba : 90145 uuid : 7961081c-8268-4c4c-a22b-f7241e8197fa , part-name :
system, first_lba : 90146, last_lba : 8478753 uuid : 8838e311-66da-46c5-9dd0-0d9c721e4452 , part-name :
dtb, first_lba : 8478754, last_lba : 8479153 uuid : 30d07c55-dd52-48c3-a6a2-e2ba30e9d6cf , part-name :
env, first_lba : 8479154, last_lba : 8481201 uuid : f4a2cada-fa39-4c0e-8fb9-88064fce5f30 , part-name :
misc, first_lba : 8481202, last_lba : 8483249 crc32 of header : 3c5c3dfa crc32 of partition array :
f120de17    DiskInfo.ullPartitionInfoStartOffset    :    208,    sizeof(tagDiskImageHeader)    =    96
DiskInfo.ullPartitionInfoStartOffset  :  17728,  ptbl.guid_entry[0].first_lba  =  34  idx  :  0  bl3_ca53_a
BunchHeader.ullTargetAddress   :   17408,   BunchHeader.ullLength   =   578048   idx   :   1   bl3_ca53_b
BunchHeader.ullTargetAddress   :   2114560,   BunchHeader.ullLength   =   578048   idx   :   2   boot
BunchHeader.ullTargetAddress   :   4211712,   BunchHeader.ullLength   =   17278976   idx   :   3   system
BunchHeader.ullTargetAddress   :   46154752,   BunchHeader.ullLength   =   337641472   idx   :   4   dtb
BunchHeader.ullTargetAddress : 4341122048, BunchHeader.ullLength = 55296 crc32 of header : 3c5c3dfa crc32
of partition array : 18c9eaf7 Complete to make fai file

===== arguments info =====

--storage_size  :  6368709120  --parttype  :  gpt  --area_name  :  "SD  Data"  --outfile  :  /home/topst/topst-
sdk/.stitch_AcJHHLv/output_nd.fai  --gptfile  :  /home/topst/topst-sdk/.stitch_AcJHHLv/output_nd.gpt  --
fplist : /home/topst/topst-sdk/.stitch_AcJHHLv/partition.list.ai --sector_size : 512 --sparse_fill : 0 -
-partition_offset : 34 LBA(17408) --last_part_align : 0

===========================

[+]  Packaging  FWDN  binaries  adding:  output_nd.fai  (deflated  74%)  adding:  boot-firmware/  (stored  0%)
adding:  boot-firmware/tcc750x_fwdn.json  (deflated  53%)  adding:  boot-firmware/prebuilt/  (stored  0%)
adding:   boot-firmware/prebuilt/mxic_mx25l25645g_QPI_100_DTR.bin   (deflated   86%)   adding:   boot-
firmware/prebuilt/mxic_mx25l6433f_QPI_100_STR.bin        (deflated        86%)        adding:        boot-
firmware/prebuilt/mcert.bin (deflated 96%) adding: boot-firmware/prebuilt/tcc750x_sic_snor.rom (deflated
71%)  adding:  boot-firmware/prebuilt/common_snor_config_READ_3B.bin  (deflated  90%)  adding:  boot-
firmware/prebuilt/fwdn.rom (deflated 51%) adding: boot-firmware/prebuilt/bconf.bin (deflated 97%) adding:
boot-firmware/prebuilt/dram_params.bin (deflated 82%) adding: boot-firmware/prebuilt/tcc750x_ap_snor.rom
(deflated   65%)   adding:   boot-firmware/prebuilt/ca53_bl1.rom   (deflated   49%)   adding:   boot-
firmware/prebuilt/optee.rom  (deflated  93%)  adding:  boot-firmware/prebuilt/ca53_bl2.rom  (deflated  52%)
adding: boot-firmware/prebuilt/hsm.bin (deflated 13%) adding: boot-firmware/prebuilt/sic-fw.rom (deflated
58%)  adding:  boot-firmware/tools/  (stored  0%)  adding:  boot-firmware/tools/mktcimg/  (stored  0%)  adding:
boot-firmware/tools/mktcimg/mktcimg          (deflated          60%)          adding:          boot-
firmware/tools/mktcimg/README_gpt_partition_table.txt      (deflated      34%)      adding:      boot-
firmware/tools/snor_image_maker/          (stored          0%)          adding:          boot-
firmware/tools/snor_image_maker/tcc750x_sic_snor.json      (deflated      80%)      adding:      boot-
firmware/tools/snor_image_maker/tcc750x_ap_snor.json      (deflated      85%)      adding:      boot-
firmware/tools/snor_image_maker/snor_image_maker.py (deflated 63%) adding: boot-firmware/tools/signtool/
(stored   0%)   adding:   boot-firmware/tools/signtool/tccsignv3.py   (deflated   75%)   adding:   boot-
firmware/tools/signtool/genkey.sh   (deflated   66%)   adding:   boot-firmware/tools/signtool/tccmcert.py
```

```
(deflated 73%) adding: boot-firmware/tools/signtool/keys/ (stored 0%) adding: boot-
firmware/tools/signtool/keys/rootkey/ (stored 0%) adding: boot-
firmware/tools/signtool/keys/rootkey/signkey.pem (deflated 27%) adding: boot-
firmware/tools/signtool/keys/rootkey/enckey.bin (stored 0%) adding: boot-
firmware/tools/signtool/keys/masterkey/ (stored 0%) adding: boot-
firmware/tools/signtool/keys/masterkey/signkey.pem (deflated 26%) adding: boot-
firmware/tools/signtool/keys/masterkey/enckey.bin (stored 0%) adding: boot-
firmware/tools/signtool/keys/imagekey/ (stored 0%) adding: boot-
firmware/tools/signtool/keys/imagekey/signkey.pem (deflated 27%) adding: boot-
firmware/tools/signtool/keys/imagekey/enckey.bin (stored 0%) adding: boot-
firmware/tools/signtool/keys/hsmkey/ (stored 0%) adding: boot-
firmware/tools/signtool/keys/hsmkey/signkey.pem (deflated 26%) adding: boot-
firmware/tools/signtool/keys/hsmkey/enckey.bin (stored 0%) adding: boot-firmware/tools/signtool/lib/
(stored 0%) adding: boot-firmware/tools/signtool/lib/tccscram.so (deflated 57%) adding: boot-
firmware/tools/signtool/README.md (deflated 20%) adding: boot-firmware/tools/signtool/tccsecfw.py
(deflated 77%) adding: boot-firmware/tools/signtool/ssshsm.py (deflated 72%) adding: boot-
firmware/tools/signtool/.gitignore (stored 0%) adding: boot-firmware/tools/signtool/tcckeys.py (deflated
60%) adding: boot-firmware/tools/signtool/tccscram.py (deflated 39%) adding: boot-
firmware/tools/image_merge/ (stored 0%) adding: boot-firmware/tools/image_merge/image_merge_v00.py
(deflated 61%) adding: boot-firmware/tools/snor_conf_maker/ (stored 0%) adding: boot-
firmware/tools/snor_conf_maker/mxic_mx25l6433f_QPI_100_STR.xml (deflated 65%) adding: boot-
firmware/tools/snor_conf_maker/snor_conf_maker.py (deflated 73%) adding: boot-
firmware/tools/snor_conf_maker/mxic_mx25l25645g_QPI_100_DTR.xml (deflated 65%) adding: boot-
firmware/tools/snor_conf_maker/common_snor_config_READ_3B.xml (deflated 64%) adding: boot-
firmware/tools/bconf_maker/ (stored 0%) adding: boot-firmware/tools/bconf_maker/tcc750x_bconf.xml
(deflated 56%) adding: boot-firmware/tools/bconf_maker/bconf_maker_v01.py (deflated 73%) adding: boot-
firmware/tools/dram_parameters/ (stored 0%) adding: boot-firmware/tools/dram_parameters/tcmktool.py
(deflated 59%) adding: boot-firmware/tools/dram_parameters/pms_table.py (deflated 80%) adding: boot-
firmware/tools/dram_parameters/dram_param.py (deflated 81%) adding: boot-
firmware/tools/dram_parameters/orbit_defines.py (deflated 70%) adding: boot-firmware/tcc750x_boot.json
(deflated 84%) adding: fwdn_nd (deflated 68%) adding: fwdn_nd.bat (deflated 57%) adding: fwdn_nd.exe
(deflated 68%) adding: fwdn_nd.sh (deflated 49%) adding: mktcimg (deflated 61%) adding: output_nd.gpt
(deflated 98%) adding: output_nd.gpt.back (deflated 98%) adding: output_nd.gpt.prim (deflated 98%) adding:
VtcUsbPort.dll (deflated 68%)
```

If you see the following log, it means the "output_nd.fwdn.zip" file is created.

```
$ ls
build  easy-setup.sh  mktcimg  output_nd.fwdn.zip  poky  stitch-fai-ai.sh  stitch-fai-d3.sh  tools
```

4 Firmware Downloader

This chapter describes how to download Firmware Downloader (*FWDN*) to the AI-G and log into the Linux console.
FWDN V8 is a PC tool for downloading firmware in Windows 10 or 11 64-bit and Linux environments. This chapter describes the case of downloading in Windows and Linux environments.

4.1 Firmware Download Sequence

The downloading sequence of ***FWDN*** is as follows:

1. Set the boot mode to USB boot mode (FWDN mode).
2. Open Windows prompt or Linux console.
3. Connect ***FWDN V8*** to the board.
4. Download the fai file.

4.2 USB Boot Mode (FWDN Mode)

You can transfer the built image to AI-G board by using ***FWDN***. AI-G supports ***FWDN*** by using Ethernet and UART.

Note: The USB Type-C FWDN port is used for firmware downloader (***FWDN***).

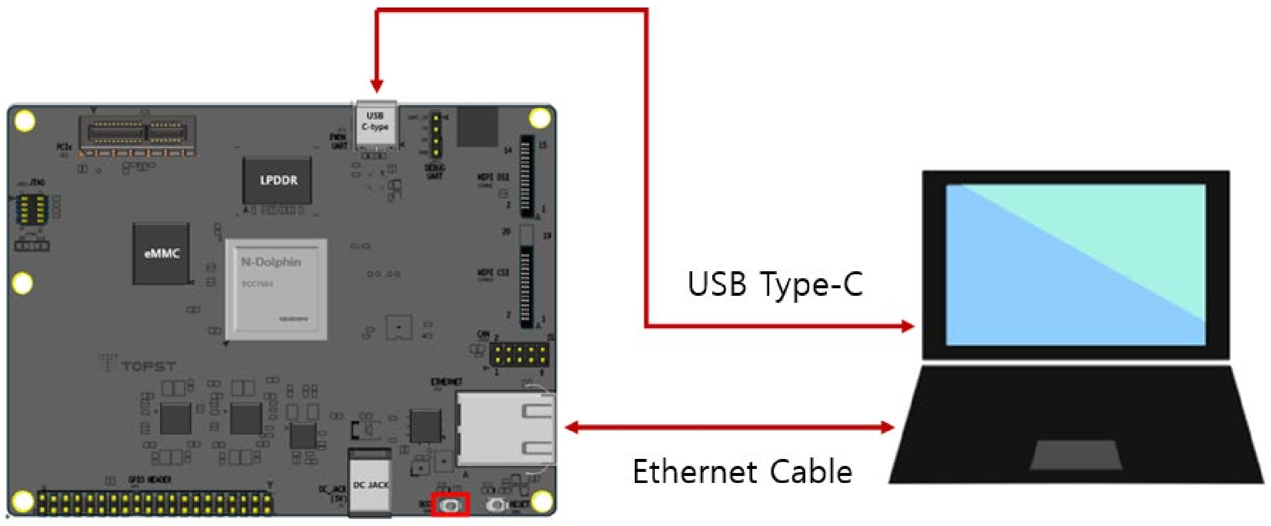


Figure 4.1 Connection Between Host PC and AI-G Board for FWDN

To use ***FWDN V8***, connect the AI-G board to the Host PC as follows:

1. Check that VTC driver is installed on the Host PC. If the VTC driver is not installed, install it as shown in Chapter 4.2.1.
2. Prepare one USB Type-C cable and one Ethernet Cable.
3. To enter USB Boot mode, connect the USB Type-C cable to the USB Type-C FWDN Port on the AI-G board and the Host PC.
4. Connect the Ethernet Cable (RJ45) to the Ethernet port on the AI-G board and the Host PC.
5. Connect the power cable to the AI-G board while pressing the FWDN button.

4.2.1 How to install VCP Driver

Install the Vendor Telechips Certification (VTC) driver (found on VCP driver) on the Host PC by running as administrator. After you successfully connect the board to the Host PC in USB Boot mode (FWDN mode) as shown above, the Telechips VTC USB driver is set as shown in Figure 4.2.

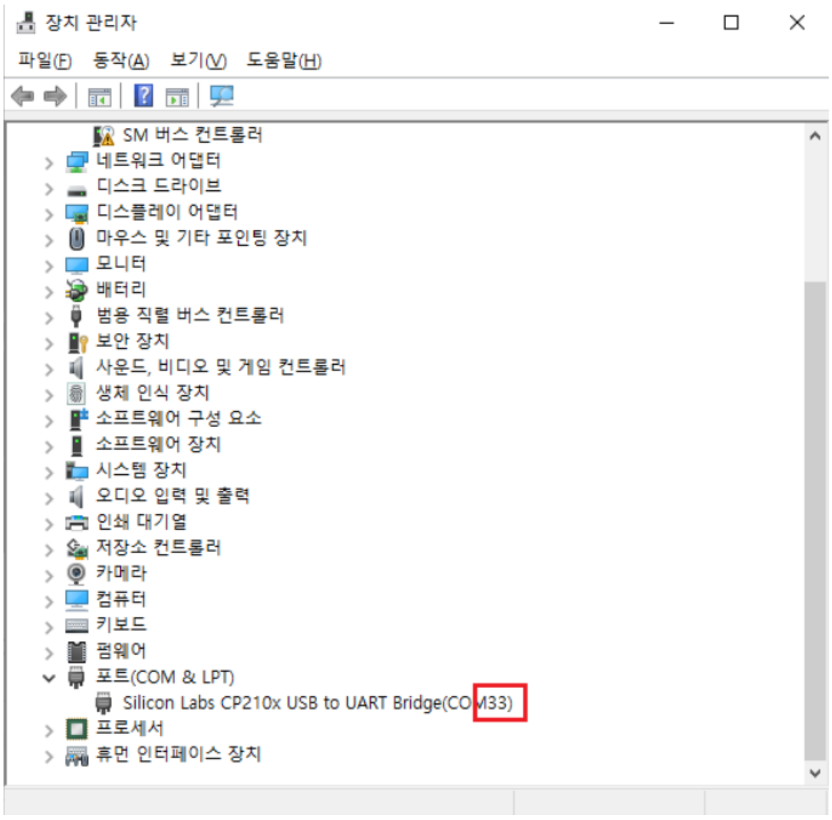
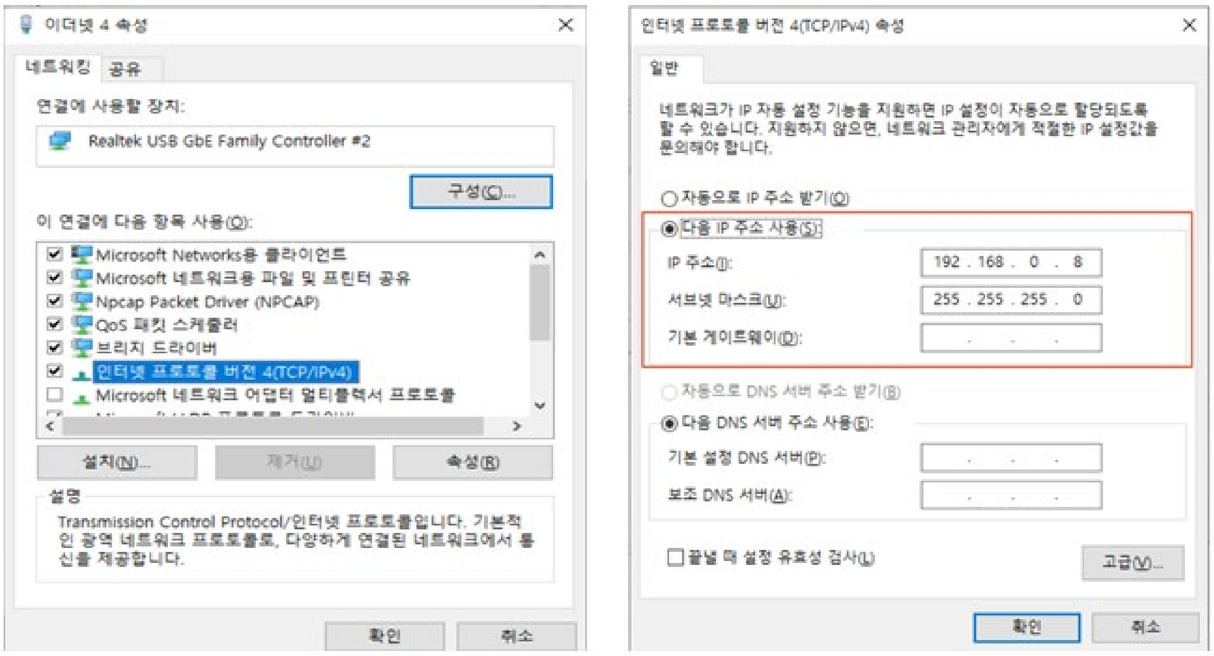


Figure 4.2 Check COM Port

4.2.2 Set Up Ethernet

Host PC Network Configuration

- Control Panel → Network and Internet → Network and Sharing Center → Change adapter settings → Right-click on Ethernet → Select Properties



이더넷 4 속성

네트워킹 | 공유

연결에 사용할 장치:

Realtek USB GbE Family Controller #2

구성(C)...

이 연결에 다음 항목 사용(O):

- ☑ Microsoft Networks용 클라이언트
- ☑ Microsoft 네트워크용 파일 및 프린터 공유
- ☑ Npcap Packet Driver (NPCAP)
- ☑ QoS 패킷 스케줄러
- ☑ 브리지 드라이버
- ☑ 인터넷 프로토콜 버전 4(TCP/IPv4)
- ☐ Microsoft 네트워크 어댑터 멀티플렉서 프로토콜

설치(N)... 제거(U) 속성(R)

설명

Transmission Control Protocol/인터넷 프로토콜입니다. 기본적인 광역 네트워크 프로토콜로, 다양하게 연결된 네트워크에서 통신을 제공합니다.

확인 취소

인터넷 프로토콜 버전 4(TCP/IPv4) 속성

일반

네트워크가 IP 자동 설정 기능을 지원하면 IP 설정이 자동으로 할당되도록 할 수 있습니다. 지원하지 않으면, 네트워크 관리자에게 적절한 IP 설정값을 문의해야 합니다.

○ 자동으로 IP 주소 받기(O)

● 다음 IP 주소 사용(S):

IP 주소(I):	192 . 168 . 0 . 8
서브넷 마스크(U):	255 . 255 . 255 . 0
기본 게이트웨이(D):	. . .

○ 자동으로 DNS 서버 주소 받기(B)

● 다음 DNS 서버 주소 사용(E):

기본 설정 DNS 서버(P):	. . .
보조 DNS 서버(A):	. . .

☐ 끝낼 때 설정 유효성 검사(L)

고급(V)...

확인 취소

Figure 4.3 Setting Ethernet Device Properties for FWDN

4.3 How to execute FWDN

Before executing ***FWDN***, transfer the image and tools created in the Ubuntu (WSL2) to the Windows Environment.
If you are using Ubuntu host, you can skip the following steps and go to Chapter 4.3.2.

1. Unzip "output_nd.fwdn.zip".

```
$ cd ~/topst-sdk/ && \
$ mkdir images && \
$ mv ./output_nd.fwdn.zip ./images && \
$ cd images && \
$ unzip output_nd.fwdn.zip && \
```

2. Copy "images" folder to Windows C drive.

```
$ cd .. && \
$ cp -r ./images /mnt/c/
```

Refer to Chapter 4.3.1 or Chapter 4.3.2 depending on the OS of your Host PC for the following steps.

4.3.1 Execute FWDN in Windows Environment

Before running ***FWDN***, WMIC must be installed to identify the port number connected to the FWDN port of the AI-G board.

1. Open **Settings** from the Start menu.

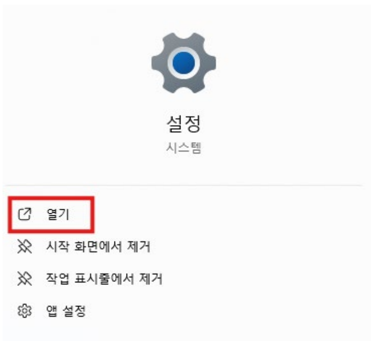


Figure 4.4 Open Settings from Start Menu

2. Click on the **System** menu.

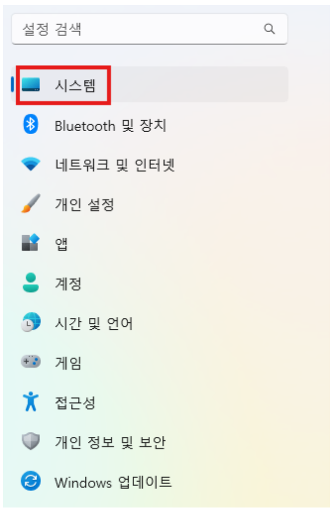


Figure 4.5 Go to System Menu

3. Click the **Selective Features** menu.

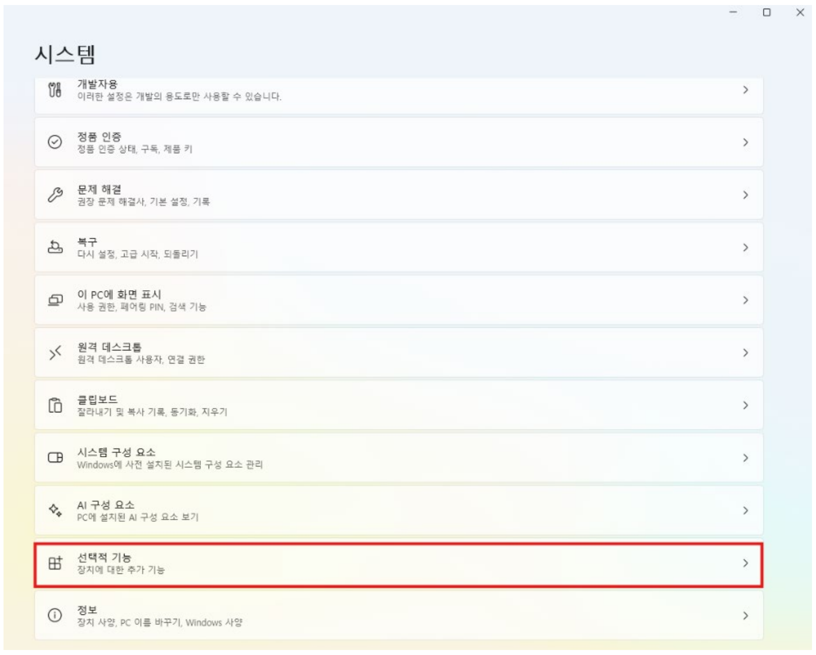

시스템
개발자용
이러한 설정은 개발의 용도로만 사용할 수 있습니다.
정품 인증
정품 인증 상태, 구독, 제품 키
문제 해결
권장 문제 해결사, 기본 설정, 기록
복구
다시 설정, 고급 시작, 되돌리기
이 PC에 화면 표시
사용 권한, 페어링 PIN, 검색 기능
원격 데스크톱
원격 데스크톱 사용자, 연결 권한
클립보드
잘라내기 및 복사 기록, 동기화, 지우기
시스템 구성 요소
Windows에 사전 설치된 시스템 구성 요소 관리
AI 구성 요소
PC에 설치된 AI 구성 요소 보기
선택적 기능
장치에 대한 추가 기능
정보
장치 사양, PC 이름 바꾸기, Windows 사양

Figure 4.6 Click Selective Features Menu

4. Click "**View Features"** button next to **Add Optional Features**.

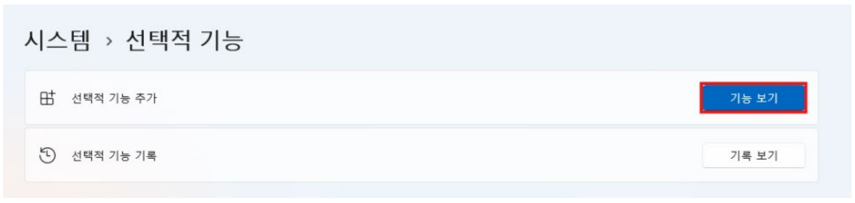


Figure 4.7 Click View Features Button Next to Add Optional Features

5. Type "WMIC" in the search box.

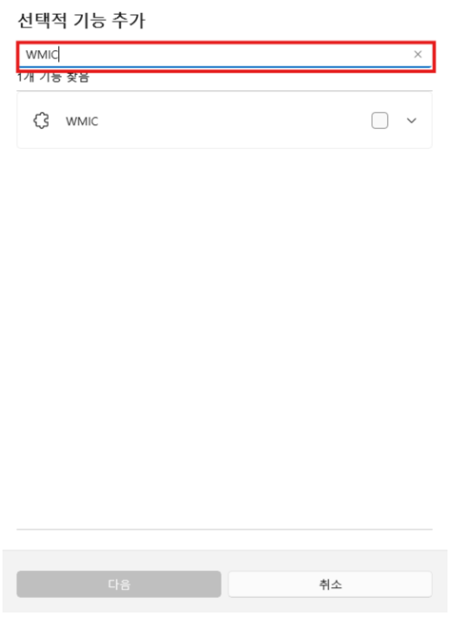


Figure 4.8 Type WMIC in Search Box

6. Select the WMIC item and click the "**Next**" button to complete the installation.

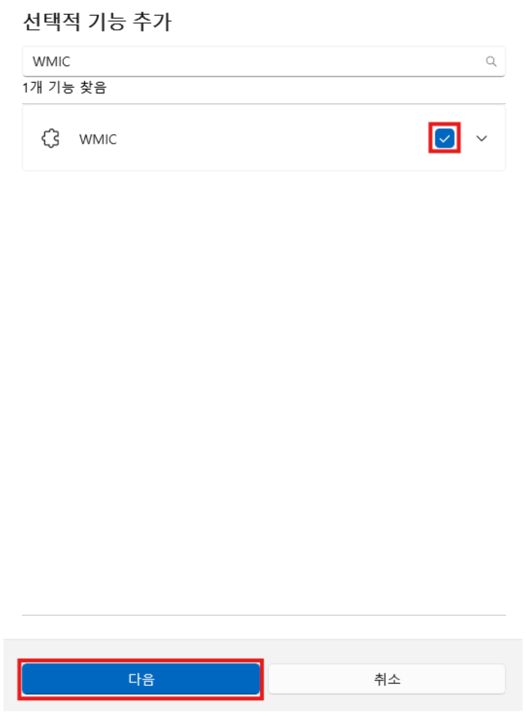


Figure 4.9 Select WMIC Item and Click Next Button to complete Installation

7. Execute Windows PowerShell and go to "C:\images".

```
$ cd C:\images
```

8. Enter the `.\fwdn_nd.bat` command to start the firmware download.
   The "fwdn_nd.bat" is an executable file that automatically downloads firmware by using ***FWDN V8***.

```
$ ./fwdn_nd.bat

TOPST AI-G FWDN Batch File

Find Port.
Silicon Labs CP210x USB to UART Bridge(COM44)

Input USB Port Number : 44


Step 1. Connect between Host PC and TOPST AI-G
[FWDNLogger::PrintCurTime:100] 25/06/12-16:11:36
[main:24] FWDN N-Dolphin v1.0.1 - 2024.9.24 16:18:11
[main:92] Start FWDN
[FWDN_ND::ImageSend:442] Complete to send MCERT image
[FWDN_ND::ImageSend:442] Complete to send HSM image
[FWDN_ND::ImageSend:442] Complete to send FWDN_BL1 image
[FWDN_ND::ImageSend:442] Complete to send FWDN_DRAM_PARAMETER image
[FWDN_ND::ImageSend:442] Complete to send FWDN_BL2 image
[FWDN_ND::ImageSend:442] Complete to send FWDN_BL3 image
[FWDN_ND::InfoDevice:123] ############## Device Info ##############
[PrintIP:527] ip      : 192.168.0.100
[PrintIP:527] mask    : 255.255.255.0
[PrintIP:527] gateway : 0.0.0.0
[FWDN_ND::InfoDevice:132] port : 8080
[FWDN_ND::InfoDevice:135]
mmc0_user: 7818182656 byte block_size: 512 byte
mmc0_boot0: 4194304 byte block_size: 512 byte
mmc0_boot1: 4194304 byte block_size: 512 byte
snor1: 0 byte
```

```
snor2: 0 byte
fw_version: tcc750x_v0.0.111
fw_build_id: g90e2b777
fw_build_date: 2024-09-24-17:11:42+0900

----- Firmware Information -----
VERSION    : tcc750x_v0.0.111
BUILD ID   : g90e2b777
BUILD DATE : 2024-09-24-17:11:42+0900

----- Detail of Storages -----
#### eMMC Info ####
Manufacture ID: 0x15
OEM: 0x100
Name: 0x8GTF4
User Capacity: 7.3 GiB (7818182656 Byte)
Boot Capacity: 4 MiB (4194304 Byte)
RPMB Capacity: 512 KiB (524288 Byte)
Speed Mode: HS200
#### SNOR Info ####

----- Summary of Storages -----
eMMC : O
SNOR : X
- O : Init success
- X : Init failed or not exist

----- Summary of DRAM Init -----
DRAM Init : Success
DRAM Size : TBD MB

----- PMU Status -----
PMU_USER   : 0x0
PMU_SIC_USER   : 0x80
PMU_CONFIG : 0x0
* BM : PMU_CONFIG[1:0]
-  0 : USB BOOT
-  1 : SNOR BOOT
-  2 : eMMC BOOT
-  3 : eMMC with SIC BOOT

--------------------------------

[FWDN_ND::LoadFwdnFW:676] TCP Ready OK
[main:156] Complete FWDN, Total Time = 18424 ms
[main:162] **Notification : For more detailed FWDN Log, use the -g or --debug option.

Step 2. Low-format
[FWDNLogger::PrintCurTime:100] 25/06/12-16:11:55
[main:24] FWDN N-Dolphin v1.0.1 - 2024.9.24 16:18:11
[main:92] Start FWDN
[FWDN_ND::InfoDevice:123] ############## Device Info ##############
[PrintIP:527] ip      : 192.168.0.100
[PrintIP:527] mask    : 255.255.255.0
[PrintIP:527] gateway : 0.0.0.0
[FWDN_ND::InfoDevice:132] port : 8080
[FWDN_ND::InfoDevice:135]
mmc0_user: 7818182656 byte block_size: 512 byte
mmc0_boot0: 4194304 byte block_size: 512 byte
mmc0_boot1: 4194304 byte block_size: 512 byte
snor1: 0 byte
snor2: 0 byte
fw_version: tcc750x_v0.0.111
fw_build_id: g90e2b777
fw_build_date: 2024-09-24-17:11:42+0900

----- Firmware Information -----
VERSION    : tcc750x_v0.0.111
BUILD ID   : g90e2b777
BUILD DATE : 2024-09-24-17:11:42+0900

----- Detail of Storages -----
#### eMMC Info ####
Manufacture ID: 0x15
OEM: 0x100
Name: 0x8GTF4
User Capacity: 7.3 GiB (7818182656 Byte)
Boot Capacity: 4 MiB (4194304 Byte)
RPMB Capacity: 512 KiB (524288 Byte)
Speed Mode: HS200
#### SNOR Info ####

----- Summary of Storages -----
eMMC : O
SNOR : X
- O : Init success
- X : Init failed or not exist

----- Summary of DRAM Init -----
DRAM Init : Success
DRAM Size : TBD MB

----- PMU Status -----
PMU_USER   : 0x0
PMU_SIC_USER   : 0x80
PMU_CONFIG : 0x0
* BM : PMU_CONFIG[1:0]
-  0 : USB BOOT
-  1 : SNOR BOOT
-  2 : eMMC BOOT
-  3 : eMMC with SIC BOOT

--------------------------------

[ParseStorageSize:558] 1
[ParseStorageSize:583] 7818182656,7818182656
[FWDN_ND::LoadFwdnFW:676] TCP Ready OK
[FWDN_ND::LowformatCommand:1495] Lowformat Start(eMMC)
[main:156] Complete FWDN, Total Time = 1755 ms
[main:162] **Notification : For more detailed FWDN Log, use the -g or --debug option.

Step 3. Download boot-firmware
[FWDNLogger::PrintCurTime:100] 25/06/12-16:11:57
[main:24] FWDN N-Dolphin v1.0.1 - 2024.9.24 16:18:11
[main:92] Start FWDN
[FWDN_ND::InfoDevice:123] ############## Device Info ##############
[PrintIP:527] ip      : 192.168.0.100
[PrintIP:527] mask    : 255.255.255.0
[PrintIP:527] gateway : 0.0.0.0
[FWDN_ND::InfoDevice:132] port : 8080
[FWDN_ND::InfoDevice:135]
mmc0_user: 7818182656 byte block_size: 512 byte
mmc0_boot0: 4194304 byte block_size: 512 byte
mmc0_boot1: 4194304 byte block_size: 512 byte
snor1: 0 byte
snor2: 0 byte
fw_version: tcc750x_v0.0.111
fw_build_id: g90e2b777
fw_build_date: 2024-09-24-17:11:42+0900

----- Firmware Information -----
VERSION    : tcc750x_v0.0.111
BUILD ID   : g90e2b777
BUILD DATE : 2024-09-24-17:11:42+0900

----- Detail of Storages -----
#### eMMC Info ####
Manufacture ID: 0x15
```

```
OEM: 0x100
Name: 0x8GTF4
User Capacity: 7.3 GiB (7818182656 Byte)
Boot Capacity: 4 MiB (4194304 Byte)
RPMB Capacity: 512 KiB (524288 Byte)
Speed Mode: HS200
#### SNOR Info ####

----- Summary of Storages -----
eMMC : O
SNOR : X
- O : Init success
- X : Init failed or not exist

----- Summary of DRAM Init -----
DRAM Init : Success
DRAM Size : TBD MB

----- PMU Status -----
PMU_USER   : 0x0
PMU_SIC_USER   : 0x80
PMU_CONFIG : 0x0
* BM : PMU_CONFIG[1:0]
-  0 : USB BOOT
-  1 : SNOR BOOT
-  2 : eMMC BOOT
-  3 : eMMC with SIC BOOT

--------------------------------


[FWDN_ND::LoadFwdnFW:676] TCP Ready OK
[main:121] Start Write
[FWDN_ND::GetFileAndWriteCommand:1388] C:\images\boot-firmware\.\prebuilt\bconf.bin
[FWDN_ND::GetFileAndWriteCommand:1388] C:\images\boot-firmware\.\prebuilt\bconf.bin
[FWDN_ND::GetFileAndWriteCommand:1388] C:\images\boot-firmware\.\prebuilt\mcert.bin
[FWDN_ND::GetFileAndWriteCommand:1388] C:\images\boot-firmware\.\prebuilt\mcert.bin
[FWDN_ND::GetFileAndWriteCommand:1388] C:\images\boot-firmware\.\prebuilt\dram_params.bin
[FWDN_ND::GetFileAndWriteCommand:1388] C:\images\boot-firmware\.\prebuilt\dram_params.bin
[FWDN_ND::GetFileAndWriteCommand:1388] C:\images\boot-firmware\.\prebuilt\hsm.bin
[FWDN_ND::GetFileAndWriteCommand:1388] C:\images\boot-firmware\.\prebuilt\hsm.bin
[FWDN_ND::GetFileAndWriteCommand:1388] C:\images\boot-firmware\.\prebuilt\ca53_bl1.rom
[FWDN_ND::GetFileAndWriteCommand:1388] C:\images\boot-firmware\.\prebuilt\ca53_bl1.rom
[FWDN_ND::GetFileAndWriteCommand:1388] C:\images\boot-firmware\.\prebuilt\ca53_bl2.rom
[FWDN_ND::GetFileAndWriteCommand:1388] C:\images\boot-firmware\.\prebuilt\ca53_bl2.rom
[FWDN_ND::GetFileAndWriteCommand:1388] C:\images\boot-firmware\.\prebuilt\optee.rom
[FWDN_ND::GetFileAndWriteCommand:1388] C:\images\boot-firmware\.\prebuilt\optee.rom
[main:156] Complete FWDN, Total Time = 3307 ms
[main:162] **Notification : For more detailed FWDN Log, use the -g or --debug option.
100%[||||||||||||||||||||||||||||||] 328064/328064
Step 4. Download FAI file (system image)
[FWDNLogger::PrintCurTime:100] 25/06/12-16:12:00
[main:24] FWDN N-Dolphin v1.0.1 - 2024.9.24 16:18:11
[main:92] Start FWDN
[FWDN_ND::InfoDevice:123] ############## Device Info ##############
[PrintIP:527] ip      : 192.168.0.100
[PrintIP:527] mask    : 255.255.255.0
[PrintIP:527] gateway : 0.0.0.0
[FWDN_ND::InfoDevice:132] port : 8080
[FWDN_ND::InfoDevice:135]
mmc0_user: 7818182656 byte block_size: 512 byte
mmc0_boot0: 4194304 byte block_size: 512 byte
mmc0_boot1: 4194304 byte block_size: 512 byte
snor1: 0 byte
snor2: 0 byte
fw_version: tcc750x_v0.0.111
fw_build_id: g90e2b777
fw_build_date: 2024-09-24-17:11:42+0900

----- Firmware Information -----
```

```
VERSION    : tcc750x_v0.0.111
BUILD ID   : g90e2b777
BUILD DATE : 2024-09-24-17:11:42+0900

----- Detail of Storages -----
#### eMMC Info ####
Manufacture ID: 0x15
OEM: 0x100
Name: 0x8GTF4
User Capacity: 7.3 GiB (7818182656 Byte)
Boot Capacity: 4 MiB (4194304 Byte)
RPMB Capacity: 512 KiB (524288 Byte)
Speed Mode: HS200
#### SNOR Info ####

----- Summary of Storages -----
eMMC : O
SNOR : X
- O : Init success
- X : Init failed or not exist

----- Summary of DRAM Init -----
DRAM Init : Success
DRAM Size : TBD MB

----- PMU Status -----
PMU_USER   : 0x0
PMU_SIC_USER   : 0x80
PMU_CONFIG : 0x0
* BM : PMU_CONFIG[1:0]
-  0 : USB BOOT
-  1 : SNOR BOOT
-  2 : eMMC BOOT
-  3 : eMMC with SIC BOOT

--------------------------------

[ParseStorageSize:558] 1
[ParseStorageSize:583] 7818182656,7818182656
[FWDN_ND::LoadFwdnFW:676] TCP Ready OK
[main:121] Start Write
[FWDN_ND::GetFileAndWriteCommand:1388] output_nd.fai
[main:156] Complete FWDN, Total Time = 71367 ms
[main:162] **Notification : For more detailed FWDN Log, use the -g or --debug option.
100%[||||||||||||||||||||||||||||] 356166304/356166304
TOPST AI-G FWDN is complete!
```

After ***FWDN*** is completed, remove the USB Type-C cable from the FWDN port and remove the power cable.
Refer to Chapter 4.4 to start communicating with the host PC.

4.3.2 Execute FWDN in Linux Environment

Download AI-G image by entering the following command.

```
$ ./fwdn.sh
```

After executing ***FWDN***, remove the USB Type-C cable from the FWDN port and remove the power cable.
Refer to Chapter 4.4 to start communicating with the host PC.

4.4 Connect AI-G Board with Host PC

This chapter describes how to connect the host PC to the AI-G board through UART for firmware download and serial connection.

4.4.1 Connect AI-G Board with UART

Perform the following steps and verify that the firmware download is successfully completed by using the UART connection.

1. Install the serial port driver (for example, CP210x Universal Windows Driver) and PL2303_prolific driver in the Windows environment.
2. Install a terminal emulator such as Tera Term or PuTTY.
3. Connect the Host PC and the UART Pin on the AI-G board. Use a USB-to-TTL cable.
4. Connect the black cable to the GND pin.
5. Connect the white cable (RXD) to TX pin of the UART pins and the green cable (TXD) to RX pin of the UART pins.
6. Run the terminal emulator application.
7. Open Device Manager on your PC and check the port number assigned to the UART device.
8. In the terminal emulator, enter the verified port number into the **Serial line** field. Set **Speed (bsp)** to **115200** and **Flow control** to **None.**
9. Connect the power cable. Then, the AI-G board boots in the default eMMC boot mode.

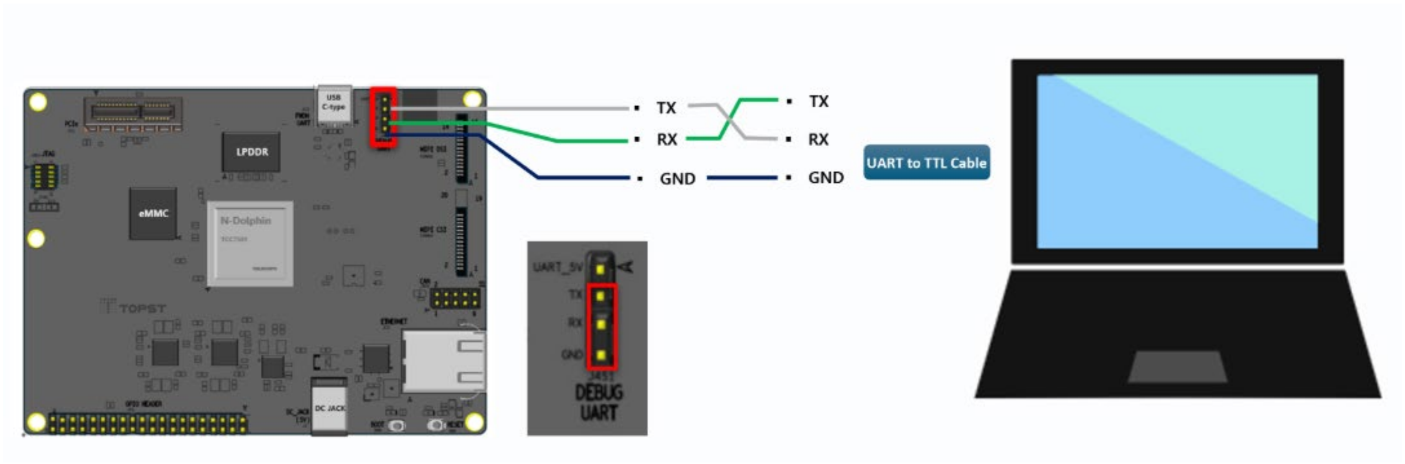
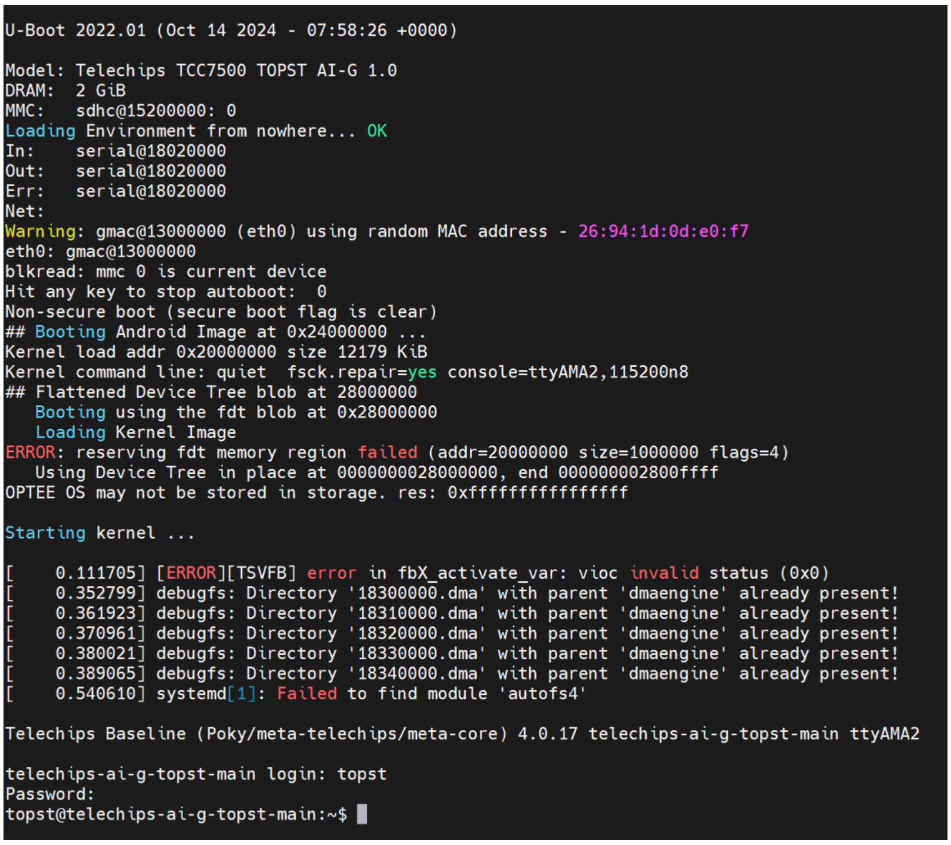


Figure 4.11 UART Connection with Host PC

Figure 4.12 shows a successful login.
Both the username and password for login are set to **topst**.



```
U-Boot 2022.01 (Oct 14 2024 - 07:58:26 +0000)

Model: Telechips TCC7500 TOPST AI-G 1.0
DRAM:  2 GiB
MMC:   sdhc@15200000: 0
Loading Environment from nowhere... OK
In:    serial@18020000
Out:   serial@18020000
Err:   serial@18020000
Net:
Warning: gmac@13000000 (eth0) using random MAC address - 26:94:1d:0d:e0:f7
eth0: gmac@13000000
blkread: mmc 0 is current device
Hit any key to stop autoboot:  0
Non-secure boot (secure boot flag is clear)
## Booting Android Image at 0x24000000 ...
Kernel load addr 0x20000000 size 12179 KiB
Kernel command line: quiet  fsck.repair=yes console=ttyAMA2,115200n8
## Flattened Device Tree blob at 28000000
   Booting using the fdt blob at 0x28000000
   Loading Kernel Image
ERROR: reserving fdt memory region failed (addr=20000000 size=1000000 flags=4)
   Using Device Tree in place at 0000000028000000, end 000000002800ffff
OPTEE OS may not be stored in storage. res: 0xffffffffffffffff

Starting kernel ...

[    0.111705] [ERROR][TSVFB] error in fbX_activate_var: vioc invalid status (0x0)
[    0.352799] debugfs: Directory '18300000.dma' with parent 'dmaengine' already present!
[    0.361923] debugfs: Directory '18310000.dma' with parent 'dmaengine' already present!
[    0.370961] debugfs: Directory '18320000.dma' with parent 'dmaengine' already present!
[    0.380021] debugfs: Directory '18330000.dma' with parent 'dmaengine' already present!
[    0.389065] debugfs: Directory '18340000.dma' with parent 'dmaengine' already present!
[    0.540610] systemd[1]: Failed to find module 'autofs4'

Telechips Baseline (Poky/meta-telechips/meta-core) 4.0.17 telechips-ai-g-topst-main ttyAMA2

telechips-ai-g-topst-main login: topst
Password:
topst@telechips-ai-g-topst-main:~$
```

Figure 4.12 Connect Screen (ID and Password are topst)

5 REFERENCES

[1] Contact TOPST for more details: topst@topst.ai

Note: Reference documents can be provided whenever available, depending on the terms of a contract. If the reference documents are unavailable, the contents directly related to your development can be guided.

6 REVISION HISTORY

Rev. 1.01: 2025-07-18

- Updated
  - Chapter 2: Description
  - Chapter 2.2: Changed chapter title from "Setting Linux Environment" to "Linux Environment"
  - Chapter 2.2.2:
    - Changed chapter title from "Install SSH" to "Install SSH and Samba"
    - Description
  - Chapter 2.2.3:
    - Changed chapter title from "Install Utilities for Yocto Project (Optional)" to "Install Utilities"
    - Description
  - Chapter 2.2.4: Changed chapter title from "Install Repo (Optional)" to "Install Repo"
  - Chapter 2.2.5: Description
  - Chapter 3: Description
  - Chapter 3.1:
    - Changed chapter title from "TOPST AI-G SDK" to "SDK Build Preparation"
    - Description
  - Chapter 3.4: Table 3.1
  - Chapter 3.5: Description
  - Chapter 3.5.1:
    - Changed chapter title from "Set Email and Name in .gitconfig" to "Set Email and Username in .gitconfig"
    - Description
  - Chapter 3.5.2:
    - Changed chapter title from "Get TOPST AI-G SDK from Git" to "Get AI-G SDK from Git"
    - Description
  - Chapter 3.6: Description
  - Chapter 3.7: Description
  - Chapter 4: Description
  - Chapter 4.2:
    - Added **Note**
    - Figure 4.1
  - Chapter 4.2.1: Description
  - Chapter 4.2.2: Description
  - Chapter 4.3: Description
  - Chapter 4.3.1: Description
  - Chapter 4.3.2: Description
  - Chapter 4.4:
    - Changed chapter title from "TOPST AI-G Connection with UART" to "Connect AI-G Board with UART"
    - Figure 4.11

- Added
  - Chapter 2.1: Windows Environment
  - Chapter 2.2.1: Setting Environment
  - Chapter 4.4: Connect AI-G Board with Host PC

- Deleted
  - Chapter 2.4.1: Set Locale
  - Chapter 3.6.1: Execute Build Script
  - Chapter 5: Appendix

Rev. 1.00: 2025-02-28

- Official version release