

---

# TOPST AI-G Linux SDK

## Getting Started

**Rev. 1.00**  
**2025-03-xx**

※ The information in this document is subject to change without notice and should not be construed as a commitment by Telechips, Inc.

Kindly visit [www.telechips.com](http://www.telechips.com) for more information.

© 2025 Telechips Inc. All rights reserved.

|  |  |  |
|--|--|--|
|  |  |  |
|--|--|--|

TABLE OF CONTENTS

Contents

TABLE OF CONTENTS ..... 2

1 Introduction ..... 3

2 Setting Host Environment ..... 4

2.1 Ubuntu Installation ..... 4

2.2 Install WSL2 Ubuntu (Windows Environment Only) ..... 4

2.3 Ubuntu Environment ..... 4

2.4 Setting Linux Environment..... 5

2.4.1 Set Locale ..... 5

2.4.2 Install SSH ..... 5

2.4.3 Install Utilities for Yocto Project (Optional) ..... 5

2.4.4 Install Repo (Optional)..... 6

2.4.5 Udev Rules for Telechips USB Device (Optional) ..... 6

3 Build Guide..... 7

3.1 TOPST AI-G SDK ..... 7

3.2 Yocto Project..... 7

3.3 Task Process ..... 7

3.4 Ready to build ..... 8

3.4.1 Set Email and Name in .gitconfig ..... 8

3.4.2 Get TOPST AI-G SDK from Git ..... 8

3.5 Composition of TOPST AI-G SDK..... 9

3.6 Execute Build Script ..... 10

3.6.1 Execute Build Script..... 10

3.7 Make Firmware Downloader (FWDN) Image..... 13

4 Firmware Downloader ..... 14

4.1 Firmware Download Sequence ..... 14

4.2 USB Boot Mode (FWDN Mode)..... 14

4.2.1 How to install VCP Driver ..... 15

4.2.2 Set Up Ethernet ..... 16

4.3 How to execute FWDN ..... 17

4.3.1 Execute FWDN in Windows Environment ..... 17

4.3.2 Execute FWDN in Linux Environment ..... 19

4.4 TOPST AI-G Connection with UART ..... 20

5 Appendix..... 22

5.1 Download Ubuntu Image..... 22

6 References ..... 23

7 Revision History ..... 24

Rev. 1.00: 2025-03-xx..... 24

Figures

Figure 3.1 Yocto Project Task Process ..... 7

Figure 4.1 Connection Between Host PC and TOPST AI-G for FWDN ..... 12

Figure 4.2 USB and Ethernet Socket for FWDN and Buttons to change Boot Mode ..... 13

Figure 4.3 Check the COM Port ..... 13

Figure 4.4 Setting Ethernet Device Properties for FWDN..... 14

Figure 4.5 Input Port Number for FWDN ..... 15

Figure 4.6 Finish Firmware Download ..... 16

Figure 4.7 UART Port ..... 16

Tables

Table 3.1 Composition of TOPST AI-G SDK..... 9

|  |  |  |
|--|--|--|
|  |  |  |
|--|--|--|

# 1 INTRODUCTION

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

The document includes the following:

- Setting Host Environment
- Build Guide
- Firmware Downloader Guide
- Ubuntu Download Guide

## 2 SETTING HOST ENVIRONMENT

### 2.1 Ubuntu Installation

TOPST AI-G Linux SDK is based on Yocto Project 4.0 Kirkstone, so the Linux version of TOPST AI-G SDK follows the Yocto project. You can install another version of Linux, but in this document, TOPST AI-G Linux SDK is described based on Ubuntu 22.04.

Linux distribution version:

- Ubuntu 22.04 (LTS)

### 2.2 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 WSL2.

```
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 LTS in Microsoft Store and download it.
7. Check Ubuntu 22.04 in WSL list.

```
wsl --list --online
```

8. Install Ubuntu 22.04.

```
wsl --install Ubuntu-22.04
```

9. Search for WSL in the Windows search box and execute it.

### 2.3 Ubuntu Environment

If you are using Ubuntu host, you can skip installing WSL2.

## 2.4 Setting Linux Environment

To set up a Linux environment on your host PC, follow these steps:

1. Execute WSL2 (Windows Environment Only)  
If you are using Windows, start WSL2 by executing one of the following commands in Windows PowerShell.

```
wsl
```

```
ubuntu
```

2. Update Package List  
Before installing any new software, update the list of available packages. This ensures you get the latest versions and dependencies. The following command fetches the latest list of available packages from the repositories.

```
sudo apt update && /  
sudo apt upgrade
```

### 2.4.1 Set Locale

After executing Ubuntu on WSL, you should set the locale to ensure proper language and regional settings. It is recommended to use `en_US.UTF-8`. Execute the following commands to use `en_US.UTF-8`.

```
sudo locale-gen en_US.UTF-8 && sudo update-locale LANG=en_US.UTF-8
```

After setting locale, you can check the locale type by using the following commands.

```
echo 'LANG=en_US.UTF-8' | sudo tee -a /etc/default/locale && \  
echo 'LC_ALL=en_US.UTF-8' | sudo tee -a /etc/default/locale
```

### 2.4.2 Install SSH

After you install a virtual machine, you can use additional utilities such as SSH for a more convenient development environment. You can execute commands on remote computers and copy files to other computers by using SSH.

If SSH is already installed or you are not going to use it, you can skip this chapter.

In Ubuntu, use the following command to install net-tools and SSH.

```
sudo apt-get install -y net-tools openssh-server
```

After downloading SSH, you should set each program to your environment.

### 2.4.3 Install Utilities for Yocto Project (Optional)

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

Execute the following command to install utilities.

```
sudo apt-get install -y gawk wget git-core diffstat unzip texinfo gcc-multilib build-essential chrpath  
socat cpio python3 python3-pip python3-pexpect xz-utils debianutils iputils-ping python3-git python3-  
jinja2 libstd1.2-dev pylint xterm vim zip zstd liblz4-tool
```

You can find these packages to install for Yocto Project as follows:

- <https://docs.yoctoproject.org/dev/migration-guides/migration-4.0.html>

## 2.4.4 Install Repo (Optional)

You can download TOPST AI-G SDK by using **Repo**.

If **Repo** is already installed, you do not need to reinstall it.

Before installing **Repo**, make sure Python version 3.6 or higher is properly set.

Execute 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 that allows the file "python" to point to **python3**.

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

If you see the error message **repo: no such file or directory**, execute the following command to download the latest version of **Repo** and put it directly into 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 && \  

```

## 2.4.5 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
```

You can confirm whether the udev rule is successfully applied by using the following command.

```
cat /etc/udev/rules.d/99-topst.rules
```

The following code is displayed if the rule is correctly added.

```
SUBSYSTEM=="usb", ATTR{idVendor}=="140e", MODE="0666", OWNER="TOPST"
```

## 3 BUILD GUIDE

### 3.1 TOPST AI-G SDK

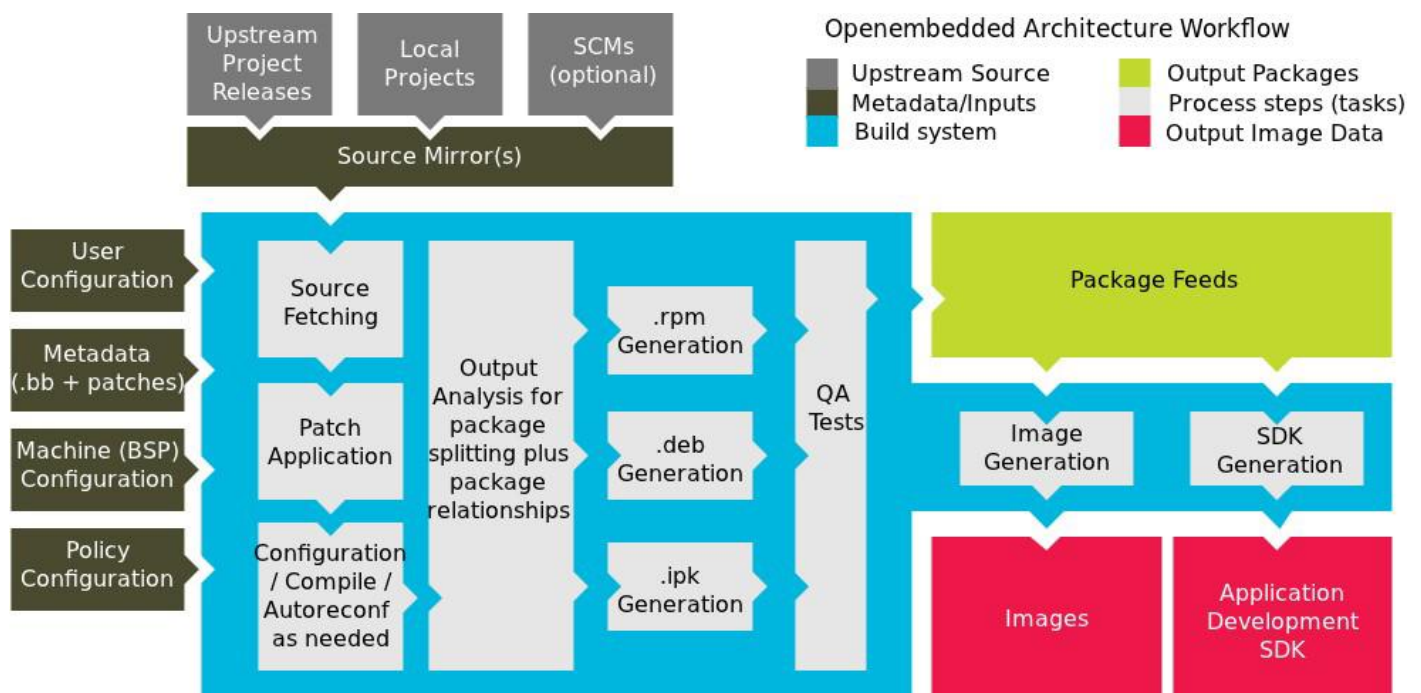
TOPST AI-G Linux SDK is based on Yocto Project 4.0 Kirkstone. Therefore, the Yocto Project environment must be set on the Host PC to use TOPST AI-G Linux SDK. To download SDK, source-mirror, and tools, you must install utilities.

### 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**

There is one way to download and build TOPST AI-G Linux SDK as follows:

- Manual operation

## 3.4 Ready to build

### 3.4.1 Set Email and Name in .gitconfig

To download TOPST AI-G from the git, configure your email and name.

1. Enter the following command.

```
vi ~/.gitconfig
```

2. Enter the following information.

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

### 3.4.2 Get TOPST AI-G SDK from Git

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

```
mkdir topst-aig && \
cd topst-aig
```

2. Execute the following command to initialize the repository.

```
repo init -u git@gitlab.com:topst-private-release/manifests.git -b topst-ai -m
linux_yp4.0_topst_ai_test.xml
```

After running the command, the following output is displayed.

```
Downloading Repo source from https://gerrit.googlesource.com/git-repo
... A new version of repo (2.50) is available.
... New version is available at: /home/TOPST/topst-aig/.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: TOPST <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-aig
```

3. Execute the following command to synchronize the repository.

```
repo sync
```

After running the command, the following output is displayed.

```
... A new version of repo (2.50) is available.
... New version is available at: /home/TOPST/topst-aig/.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% (9/9), done in 13.638s
Checking out: 100% (9/9), done in 0.510s
repo sync has finished successfully.
```



## 3.5 Composition of TOPST AI-G SDK

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

**Table 3.1 Composition of TOPST AI-G SDK**

| Item          |                       |           | Description   |
|---------------|-----------------------|-----------|---|
| easy-setup.sh |                       |           | Python script to automatically download and build the SDK   |
| stitch-fai.sh |                       |           | Script for making fai images (minimal + Sample Application)   |
| boot-firmware |                       |           | Tools related to the build process and <b><i>FWDN</i></b>   |
| fwdn v8       |                       |           |   |
| mktcimg       |                       |           |   |
| poky          | poky                  |           |   |
|               | meta-openembedded     |           | Support OE-Core Layer   |
|               | meta-arm              |           | Support ARM toolchain Layer   |
|               | meta-telechips-ai-bsp |           | Support TOPST BSP Layer   |
|               | meta-gplv2            |           | Support packages that avoid GPLv3 license   |
|               | meta-telechips        | meta-core | Recipes that require modification from Open-Source Software (OSS) used by Telechips TOPST AI-G SDK or recipes that are not in Yocto Project 4.0 |
|               | meta-topst-ai         |           | TOPST AI-G recipe   |

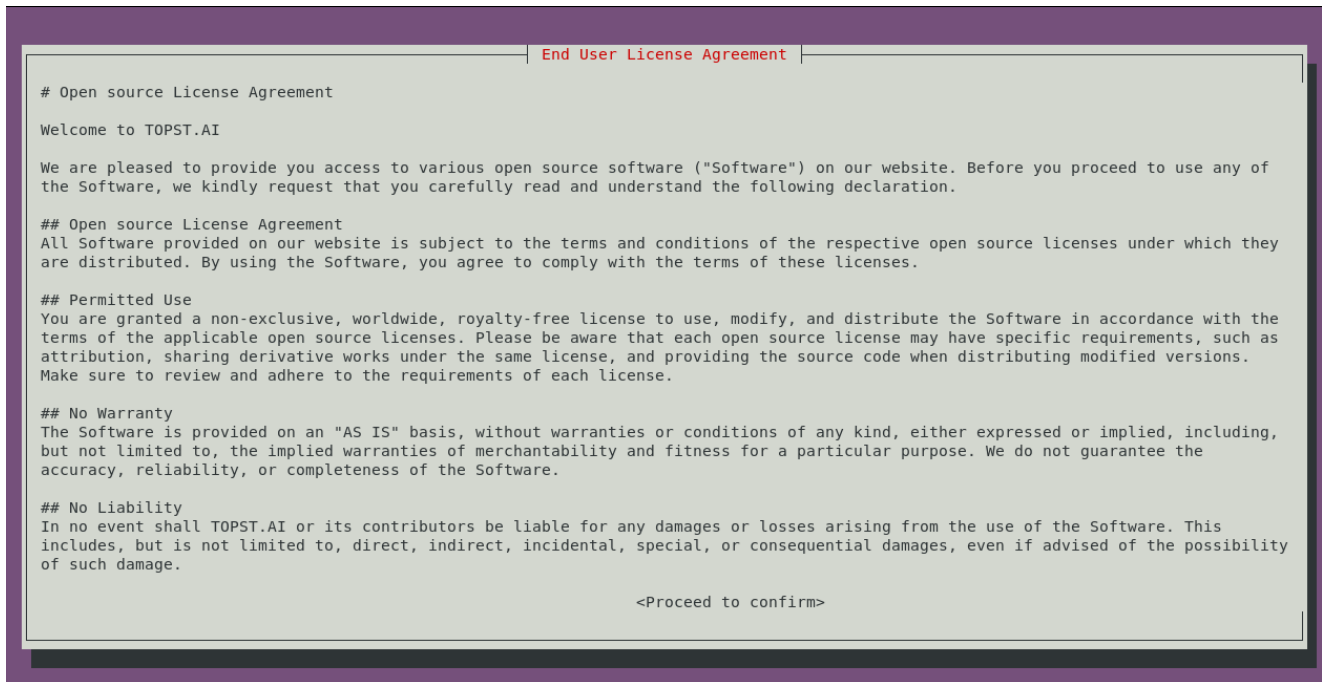
## 3.6 Execute Build Script

### 3.6.1 Execute Build Script

If you run `./easy-setup.sh` script, you can see the following screen.

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

```
./easy-setup.sh
```



```
# 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.

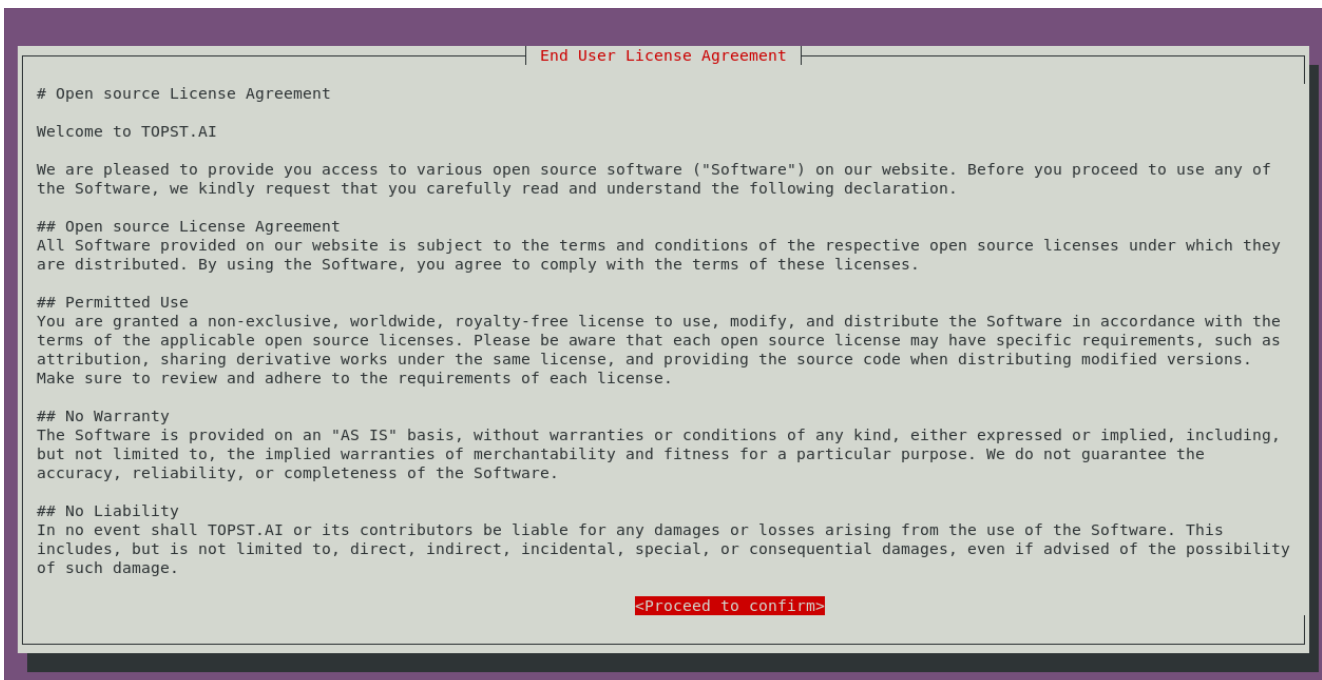
## No Warranty
The Software is provided on an "AS IS" basis, without warranties or conditions of any kind, either expressed or implied, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. We do not guarantee the accuracy, reliability, or completeness 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 this notice. After you read this notice, press the right arrow key and **[Enter]**.



```
# 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.



**Figure 3.4 Accept Screen**

If you select **Accept** by pressing **[Enter]**, you can build following the command.

[+] Please execute the following command to initiate the build process

```
source poky/meta-topst-ai/topst-ai-build.sh
bitbake telechips-topst-image
```

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

Choose MACHINE

1. ai-g-topst-main

select number(1-1) => 1

machine(ai-g-topst-main) selected.

You had no conf/local.conf file. This configuration file has therefore been created for you from /home/sunam/test/poky/meta-topst-ai/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/sunam/test/poky/meta-topst-ai/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-image
```

```
meta-toolchain-topst(Application Development Toolkit)
```

You can also run generated TOSPT images on TOPST 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

```
bitbake telechips-topst-image
```

**Note:** The build takes about 4 hours to complete.

## 3.7 Make Firmware Downloader (FWDN) Image

This option combines binaries into an image for the TOPST AI-G platform.

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

■ ~/topst-aig/

```
cd ~/topst-aig/ && \
./stitch-fai.sh -f
```

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

```
$ bitbake telechips-topst-image -f -c make_fai
Loading cache: 100%
|#####| Time: 0:00:00
Loaded 2933 entries from dependency cache.
Parsing recipes: 100%
|#####| Time: 0:00:04
Parsing of 1847 .bb files complete (1846 cached, 1 parsed). 2934 targets, 426 skipped, 0 masked, 0 errors.
WARNING: No recipes in default available for:
  /home/topst/topst-aig/poky/meta-topst-ai/recipes-bsp/drivers/tcc-vpu_1.0.0.bbappend
  /home/topst/topst-aig/poky/meta-topst-ai/recipes-bsp/u-boot/u-boot-tcc.bbappend

Build Configuration:
BB_VERSION      = "2.0.0"
BUILD_SYS       = "x86_64-linux"
NATIVELSBSTRING = "universal"
TARGET_SYS      = "aarch64-telechips-linux"
MACHINE         = "ai-g-topst-main"
DISTRO          = "poky-telechips-systemd"
DISTRO_VERSION  = "4.0.17"
TUNE_FEATURES   = "aarch64 armv8a crc cortexa53"
TARGET_FPU      = ""
LINUX_VERSION   = "5.10.205"
KBUILD_DEFCONFIG = "tcc750x_defconfig"
IMAGE_FEATURES  = " debug-tweaks  ssh-server-openssh"
GCCVERSION      = "arm-11.2"
GLIBCVERSION    = "2.35"
TOPST_FEATURES  = " network"
meta
meta-poky       = "HEAD:6d1a878bbf24c66f7186b270f823fcd82e35383"
meta-gplv2      = "HEAD:d2f8b5cdb285b72a4ed93450f6703ca27aa42e8a"
meta-arm-toolchain = "HEAD:d7b7b6fb6c7c5545e718e44f38853d1718ce5446"
meta-oe         = "HEAD:8bb16533532b6abc2eded7d9961ab2a108fd7a5b"
meta-core       = "HEAD:52a6f7e6fe2a64c0e8daf497cb5a9e371c2a5b09"
meta-topst-ai-bsp = "HEAD:ce8c4aec1ab821064c73fb61a2106116500ab508"
meta-topst-ai    = "HEAD:df9d2246862bed26b1295a8ad7c44b01c517d64b"

NOTE: Tainting hash to force rebuild of task /home/topst/topst-aig/poky/meta-topst-ai/recipes-telechips-topst/images/telechips-topst-image.bb, do_make_fai
WARNING: /home/topst/topst-aig/poky/meta-topst-ai/recipes-telechips-topst/images/telechips-topst-image.bb:do_make_fai is tainted from a forced run
Initialising tasks: 100%
|#####| Time: 0:00:01
Sstate summary: Wanted 4 Local 0 Mirrors 0 Missed 4 Current 71 (0% match, 94% complete)
NOTE: Executing Tasks
NOTE: Tasks Summary: Attempted 328 tasks of which 309 didn't need to be rerun and all succeeded.
NOTE: Writing buildhistory
NOTE: Writing buildhistory took: 3 seconds

Summary: There were 44 WARNING messages.
```

## 4 FIRMWARE DOWNLOADER

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

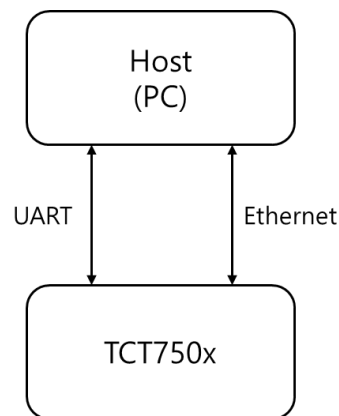
### 4.1 Firmware Download Sequence

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 board.
4. Download fai file.

### 4.2 USB Boot Mode (FWDN Mode)

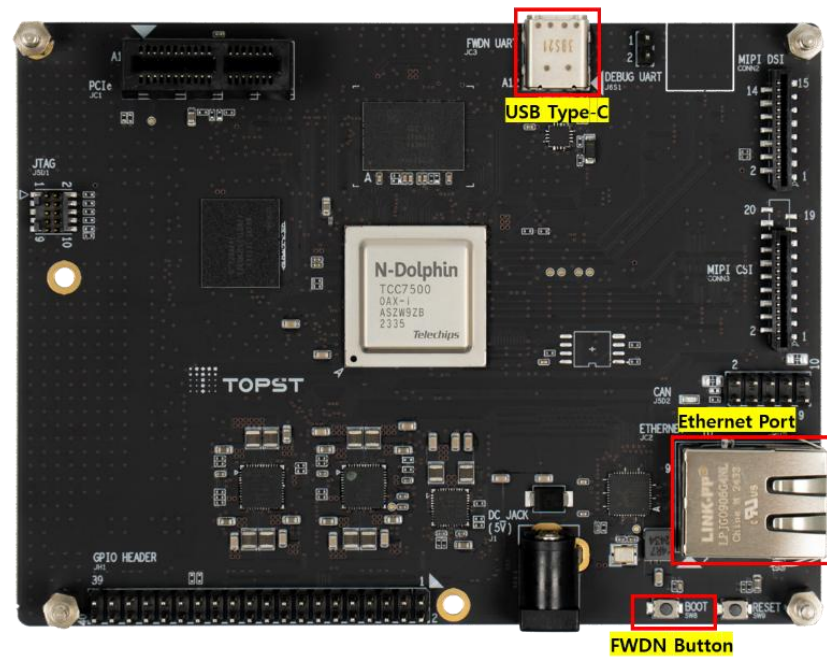
You can transfer the built image to TCT750x by using **FWDN**. TCT750x provides **FWDN** by using Ethernet and UART.



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

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

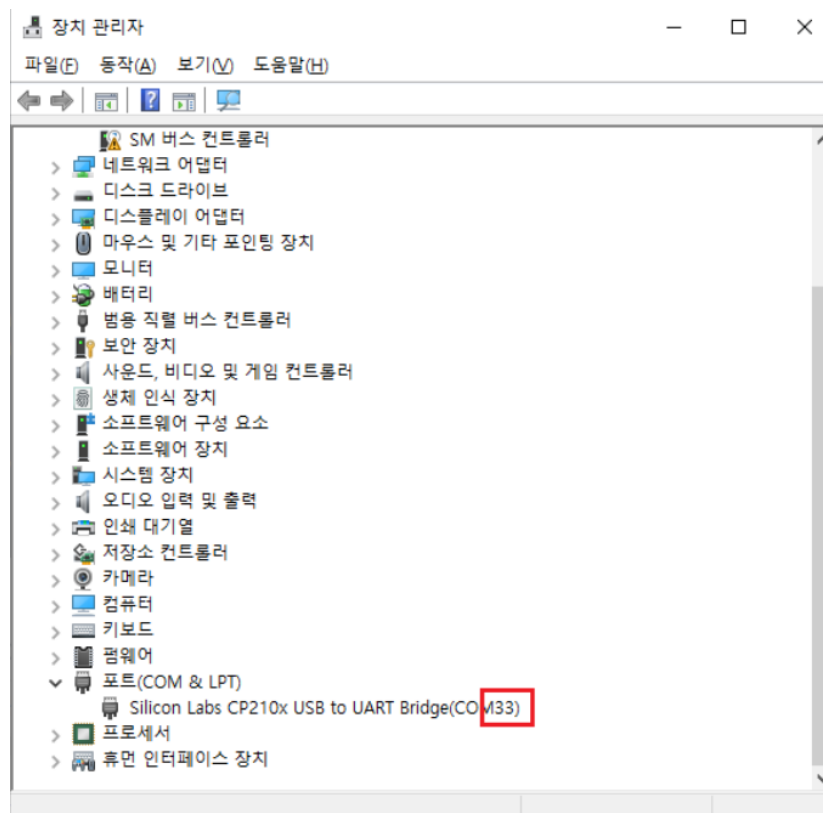
1. Check that VTC driver is installed on the Host PC. If the VTC driver is not installed, install it as shown in Chapter 4.2.1.
2. Prepare one USB Type-C cable and one Ethernet Cable.
3. To enter USB Boot mode, connect the USB Type-C cable to the USB Type-C FWDN Port on the TOPST AI-G and the Host PC.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



**Figure 4.2 USB and Ethernet Socket for FWDN and Buttons to change Boot Mode**

### 4.2.1 How to install VCP Driver

Install the UART Bridge VCP driver (found on [VCP driver](#)) on the Host PC. UART Bridge VCP driver is set up as shown in Figure 4.3.



**Figure 4.3 Check COM Port**

## 4.2.2 Set Up Ethernet

### Host PC Network Configuration

- Control Panel → Network and Internet → Network Connectivity → Set Ethernet device properties for FWDN

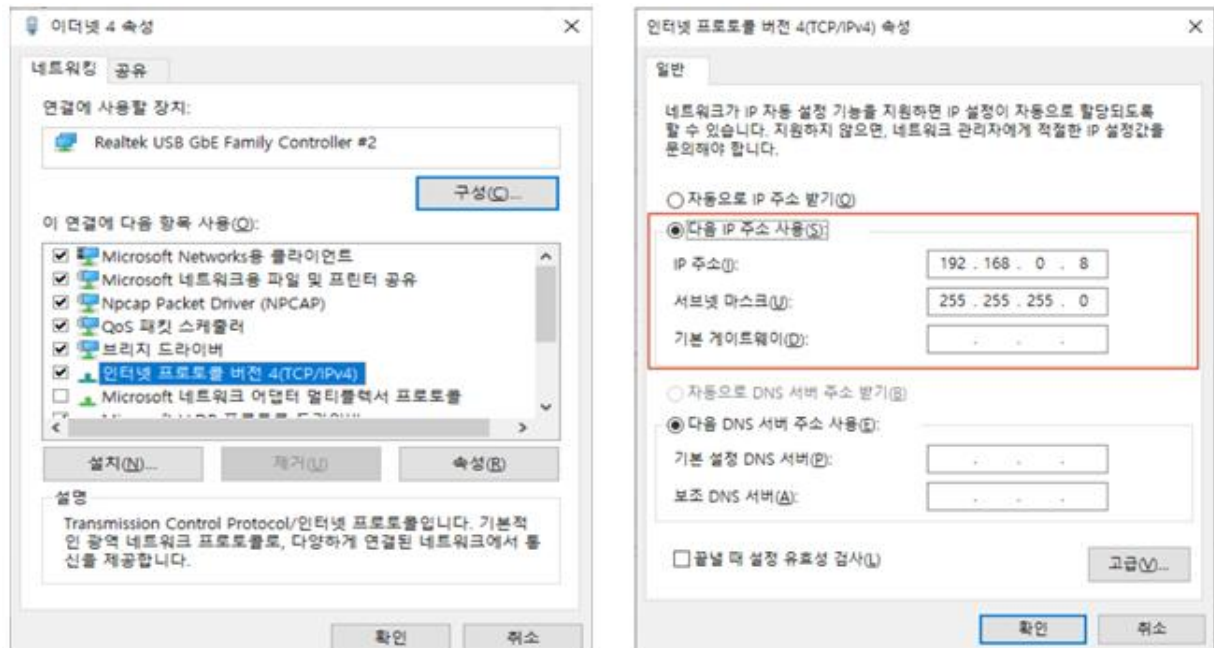


Figure 4.4 Setting Ethernet Device Properties for FWDN



## 4.3 How to execute FWDN

Before executing **FWDN**, you must first transfer the image and **FWDN** tool you created after the build from 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.fwdn.zip".

```
cd ~/topst-aig/ && \
mkdir images && \
mv ./output.fwdn.zip ./images && \
cd images && \
unzip output.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

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

```
cd C:\images
```

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

```
.\fwdn.bat

C:\images>fwdn.exe --fwdn boot-firmware\fwdn.json
[main:30] FWDN V8 v1.4.6 - 2021.12.13 13:42:37
[FWDN_V8::LoadFWDNRom:403] Start to load FWDN rom
[FWDN_V8::LoadMCERT:592] C:\images\boot-firmware\mcert.bin
[FWDN_V8::LoadHSM:609] C:\images\boot-firmware\hsm.cs.bin
[FWDN_V8::SendFWDNHeader:634] C:\images\boot-firmware\fwdn.rom - Header
[FWDN_V8::SendFWDNBody_V8:537] C:\images\boot-firmware\fwdn.rom - Body
[FWDN_V8::LoadFWDNRom:414] Complete to load FWDN rom
[FWDN_V8::GetFWDNRomVersion:1526] fwdn.rom version : 21.9.29
[FWDN_V8::GetFileAndWriteCommand:748] C:\images\boot-firmware\dram_params.bin
[FWDN_V8::PrintDeviceInfo:1183] -----Device info-----
[FWDN_V8::PrintDeviceInfo:1184]

----- Detail of Storages -----
#### eMMC Info ####
Manufacture ID: 0x15
OEM: 0x100
Name: 8GTF4
User Capacity: 7.3 GiB (7818182656 Byte)
Boot Capacity: 4 MiB (4194304 Byte)
RPMB Capacity: 512 KiB (524288 Byte)
Speed Mode: HS200
#### SNOR Info ####
Manufacture ID: 0xc2
Device ID: 0x2016
Name: MXIC-MX25L3233F
Sector Size: 4 KiB (4096 Byte)
Total Capacity: 4 MiB (4194304 Byte)
4Byte Address Mode: Unsupported

----- Summary of Storages -----
eMMC : 0
SNOR : 0
UFS : X
```

```

- 0 : Init success
- X : Init failed or not exist

----- Summary of DRAM Init -----
DRAM Init : Success (Result 0x0 )
DRAM Size : 4096MB

[FWDN_V8::PrintDeviceInfo:1185] -----
[main:142] Complete FWDN
[FWDNLogger::PrintCurTime:111] 24/04/25-09:57:47

C:\images>fwdn.exe --storage emmc --low-format
[main:30] FWDN V8 v1.4.6 - 2021.12.13 13:42:37
[FWDN_V8::GetFWDNRomVersion:1526] fwdn.rom version : 21.9.29
[FWDN_V8::LowformatCommand:1352] Start low-format
[FWDN_V8::LowformatCommand:1353] low-format can take a long time
[FWDN_V8::LowformatCommand:1382] Complete low-format
[main:142] Complete FWDN
[FWDNLogger::PrintCurTime:111] 24/04/25-09:57:50

C:\images>fwdn.exe -w boot-firmware\boot.single.json
[main:30] FWDN V8 v1.4.6 - 2021.12.13 13:42:37
[FWDN_V8::GetFWDNRomVersion:1526] fwdn.rom version : 21.9.29
[main:117] Start write command
[FWDN_V8::GetFileAndWriteCommand:748] C:\images\boot-firmware\bconf.single.bin
[FWDN_V8::GetFileAndWriteCommand:748] C:\images\boot-firmware\bconf.single.bin
[FWDN_V8::GetFileAndWriteCommand:748] C:\images\boot-firmware\mcert.bin
[FWDN_V8::GetFileAndWriteCommand:748] C:\images\boot-firmware\mcert.bin
[FWDN_V8::GetFileAndWriteCommand:748] C:\images\boot-firmware\dram_params.bin
[FWDN_V8::GetFileAndWriteCommand:748] C:\images\boot-firmware\dram_params.bin
[FWDN_V8::GetFileAndWriteCommand:748] C:\images\boot-firmware\hsm.cs.bin
[FWDN_V8::GetFileAndWriteCommand:748] C:\images\boot-firmware\hsm.cs.bin
[FWDN_V8::GetFileAndWriteCommand:748] C:\images\boot-firmware\scfw.rom
[FWDN_V8::GetFileAndWriteCommand:748] C:\images\boot-firmware\scfw.rom
[FWDN_V8::GetFileAndWriteCommand:748] C:\images\boot-firmware\optee.rom
[FWDN_V8::GetFileAndWriteCommand:748] C:\images\boot-firmware\optee.rom
[FWDN_V8::GetFileAndWriteCommand:748] C:\images\boot-firmware\ca72_b11.rom
[FWDN_V8::GetFileAndWriteCommand:748] C:\images\boot-firmware\ca72_b11.rom
[FWDN_V8::GetFileAndWriteCommand:748] C:\images\boot-firmware\ca53_b11.rom
[FWDN_V8::GetFileAndWriteCommand:748] C:\images\boot-firmware\ca53_b11.rom
[FWDN_V8::GetFileAndWriteCommand:748] C:\images\boot-firmware\ca72_b12.rom
[FWDN_V8::GetFileAndWriteCommand:748] C:\images\boot-firmware\ca72_b12.rom
[FWDN_V8::GetFileAndWriteCommand:748] C:\images\boot-firmware\ca53_b12.rom
[FWDN_V8::GetFileAndWriteCommand:748] C:\images\boot-firmware\ca53_b12.rom
[main:125] Complete write command
[main:142] Complete FWDN
[FWDNLogger::PrintCurTime:111] 24/04/25-09:57:53
100% [||||||||||||||||||||||||||||||||] 859264/859264
C:\images>fwdn.exe -w "output.fai" --storage emmc --area user
[main:30] FWDN V8 v1.4.6 - 2021.12.13 13:42:37
[FWDN_V8::GetFWDNRomVersion:1526] fwdn.rom version : 21.9.29
[main:117] Start write command
[FWDN_V8::GetFileAndWriteCommand:748] output.fai
[main:125] Complete write command
[main:142] Complete FWDN
[FWDNLogger::PrintCurTime:111] 24/04/25-10:05:21
100% [||||||||||||||||||||||||||||||||] 7238688960/7238688960
** When writing FAI files without low-format, there may be garbage values in partition where data is not
written.

```

After **FWDN** is completed, remove the USB Type-C cable from the FWDN port and remove the power cable.

### 4.3.2 Execute FWDN in Linux Environment

In a Linux environment, you can download TOPST AI-G image by entering the following command.

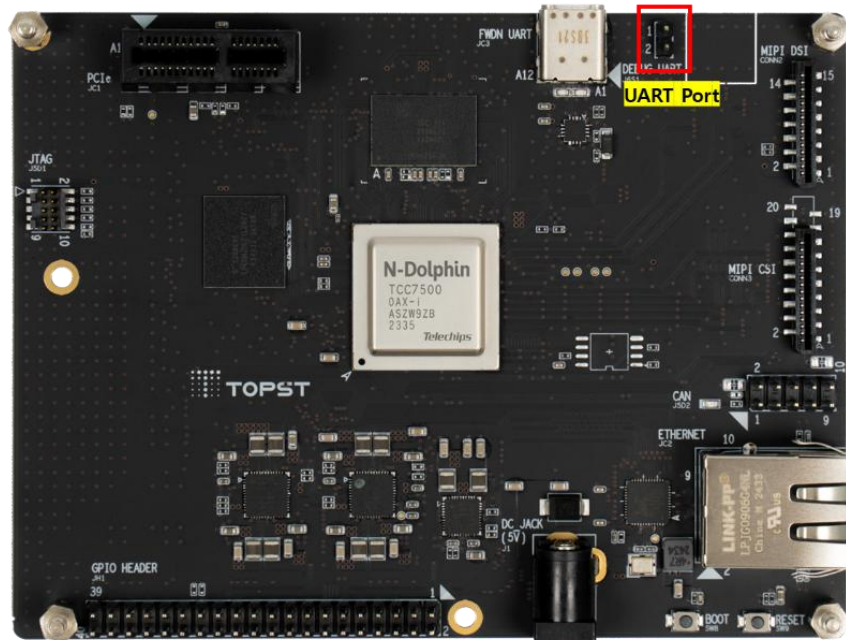
```
./fwdn.sh
```

After **FWDN** is completed, remove the USB Type-C cable from the FWDN port and remove the power cable.

## 4.4 TOPST AI-G Connection 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 (CP210x Universal Windows Driver) and PL2303\_prolific driver in the Windows environment. (CP210x Universal Windows Driver: [Download link](#), PL2303\_prolific Driver: [Download link](#))
2. Install a terminal emulator such as Tera Term or PuTTY.
3. Connect the Host PC and the UART Pin on the TOPST AI-G. Use a USB to TTL Cable.



**Figure 4.7 UART Port**

4. Connect the black cable to GND pin.
5. Connect the white cable (RXD) to TX pin of A72 UART pins and the green cable (TXD) to RX pin of UART pins.
6. Run the terminal emulator application.
7. Open Device Manager on your PC and check the port number that is being used for UART.
8. Enter the Verified Port number in Device Manager into **Serial line** field in the terminal emulator. Set **Speed** (bsp) to 115200 and **Flow control** to None.
9. Connect the power cable. Then, the TOPST AI-G boots in the default eMMC boot mode.

```

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 4.5 Connected Screen (ID and Password are topst)**

## 5 APPENDIX

### 5.1 Download Ubuntu Image

1. Download the AI-G\_Ubuntu folder. (다운로드 링크)
2. Run "fwdn\_ubuntu.batch" file.
3. After **FWDN** is completed, remove the USB Type-C cable from the FWDN port and remove the power cable.

## 6 REFERENCES

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

## 7 REVISION HISTORY

### Rev. 1.00: 2025-03-xx

- Official version release



## DISCLAIMER

This material is being made available solely for your internal use with its products and service offerings of Telechips, Inc ("Telechips"). and/or licensors and shall not be used for any other purposes. This material may not be altered, edited, or modified in any way without Telechips' prior written approval. Unauthorized use or disclosure of this material or the information contained herein is strictly prohibited, and you agree to indemnify Telechips and licensors for any damages or losses suffered by Telechips and/or licensors for any unauthorized uses or disclosures of this material, in whole or part. Further, Telechips, Inc. reserves the right to revise this material and to make changes to its content, at any time, without obligation to notify any person or entity of such revisions or changes.

THIS MATERIAL IS BEING PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, WHETHER EXPRESSED, IMPLIED, STATUTORY OR OTHERWISE. TO THE MAXIMUM EXTENT PERMITTED BY LAW, TELECHIPS AND/OR LICENSORS SPECIFICALLY DISCLAIM ALL WARRANTIES OF TITLE, MERCHANTABILITY, NON-INFRINGEMENT, FITNESS FOR A PARTICULAR PURPOSE, SATISFACTORY QUALITY, COMPLETENESS OR ACCURACY, AND ALL WARRANTIES ARISING OUT OF TRADE USAGE OR OUT OF A COURSE OF DEALING OR COURSE OF PERFORMANCE. MOREOVER, NEITHER TELECHIPS, INC. NOR LICENSORS, SHALL BE LIABLE TO YOU OR ANY THIRD PARTY FOR ANY EXPENSES, LOSSES, USE, OR ACTIONS HOWSOEVER INCURRED OR UNDERTAKEN BY YOU IN RELIANCE ON THIS MATERIAL.

THIS MATERIAL IS DESIGNED FOR GENERAL PURPOSE, AND ACCORDINGLY YOU ARE RESPONSIBLE FOR ALL OR ANY OF INTELLECTUAL PROPERTY LICENSES REQUIRED FOR ACTUAL APPLICATION. TELECHIPS, INC. DOES NOT PROVIDE ANY INDEMNIFICATION FOR ANY INTELLECTUAL PROPERTIES OWNED BY THIRD PARTY.

## COPYRIGHT STATEMENT

Copyright in this material provided by Telechips, Inc. is owned by Telechips unless otherwise noted. For reproduction or use of Telechips' copyright material, prior written consent should be obtained from Telechips. That prior written consent, if given, will be subject to conditions that Telechips' name should be included and interest in the material should be acknowledged when the material is reproduced or quoted, either in whole or in part. You must not copy, adapt, publish, distribute, or commercialize any contents contained in the material in any manner without the written permission of Telechips. Trademarks used in Telechips' copyright material are the property of Telechips.

**For customers who use Google technology:**

"Copyright © 2013 Google Inc. All rights reserved."