

TOPST D3-G Linux SDK

Getting Started

**Rev. 1.01 [G]
2025-07-18**

※ 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 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 Windows Environment	4
2.1.1 Install WSL2 Ubuntu (Windows Environment Only).....	4
2.1.2 Accessing Ubuntu Through WSL2	5
2.1.3 Install SSH and Samba	5
2.1.4 Install Utilities.....	6
2.1.5 Install Repo	6
2.2 Linux Environment.....	7
2.2.1 Setting Environment.....	7
2.2.2 Install SSH and Samba	7
2.2.3 Install Utilities.....	7
2.2.4 Install Repo	7
2.2.5 Udev Rules for Telechips USB Device (Optional)	8
3 Build Guide.....	9
3.1 SDK Build Preparation	9
3.2 Yocto Project.....	9
3.3 Task Process	9
3.4 Composition of D3-G SDK.....	10
3.5 Ready to build	11
3.5.1 Set Email and Username in .gitconfig	11
3.5.2 Get D3-G SDK from Git	11
3.6 Execute Build Script	12
3.7 Make Firmware Downloader (FWDN) Image.....	15
4 Firmware Downloader	18
4.1 Firmware Download Sequence	18
4.2 USB Boot Mode (FWDN Mode).....	18
4.2.1 How to install VTC Driver	19
4.3 How to execute FWDN	19
4.3.1 Execute FWDN in Windows Environment	20
4.3.2 Execute FWDN in Linux Environment	21
4.4 Connect D3-G Board with Host PC.....	22
4.4.1 Connect D3-G Board with UART.....	22
5 Appendix.....	23
5.1 Download Ubuntu Image.....	23
5.2 Resize eMMC Storage (TOPST D3-G Ubuntu).....	23
6 References	24
7 Revision History	25
Rev. 1.01: 2025-07-18	25
Rev. 1.00: 2025-02-28	25

Figures

Figure 2.1 WSL2 Screenshot.....	5
Figure 3.1 Yocto Project Task Process	9
Figure 3.2 End User License Agreement.....	12
Figure 3.3 Go To 'Proceed to confirm'	12
Figure 3.4 Accept Screen.....	13
Figure 4.1 Connection Between Host PC and D3-G Board for FWDN	18
Figure 4.2 USB Connection in Windows Environment	19
Figure 4.3 USB Connection in Linux Environment.....	19
Figure 4.4 UART Connection with Host PC.....	22
Figure 4.5 Connected Screen (ID and Password are root).....	22

Tables

Table 3.1 Composition of D3-G SDK.....	10
--	----

1 INTRODUCTION

This document provides guidelines for building the D3-G SDK, including setting up the host environment, building the SDK, using the firmware downloader, and downloading Ubuntu.

This document includes information on the following:

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

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 chapter describes how to set up Windows Subsystem for Linux (WSL) to use Linux on a Windows PC. The D3-G SDK is based on the Yocto Project, so the Linux version of D3-G SDK follows the Yocto project. You can install another version of Linux, but in this document, D3-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
```

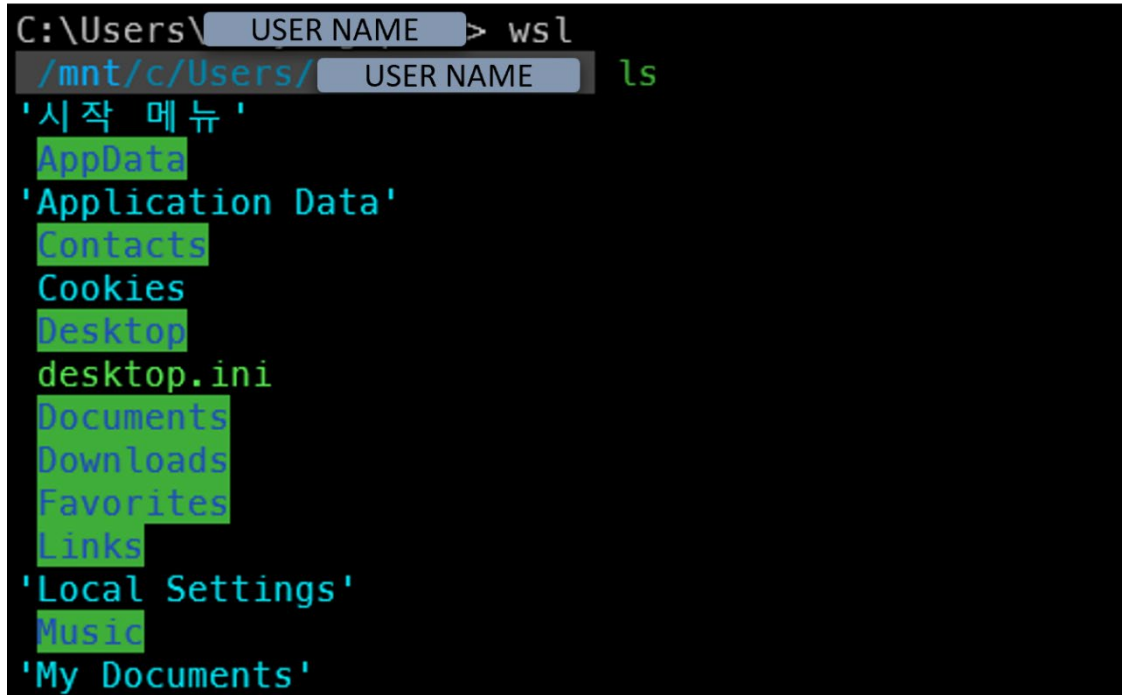
6. Search for Ubuntu 22.04.3 LTS in Microsoft Store and download it.
 - If you need to download the Linux kernel update package, download the latest package [here](#).
7. Choose any username during the Ubuntu installation.

2.1.2 Accessing 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 the D3-G SDK by using Android **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 ge
$ 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 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 libstdl1.2-dev pylint
$ sudo apt-get install -y xterm zstd ncftp curl git-lfs vim zip
```

2.2.4 Install Repo

You can download the D3-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
```

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 D3-G is built using the Yocto Project, so the build process must be performed in the Ubuntu environment.

3.1 SDK Build Preparation

The D3-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 D3-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.

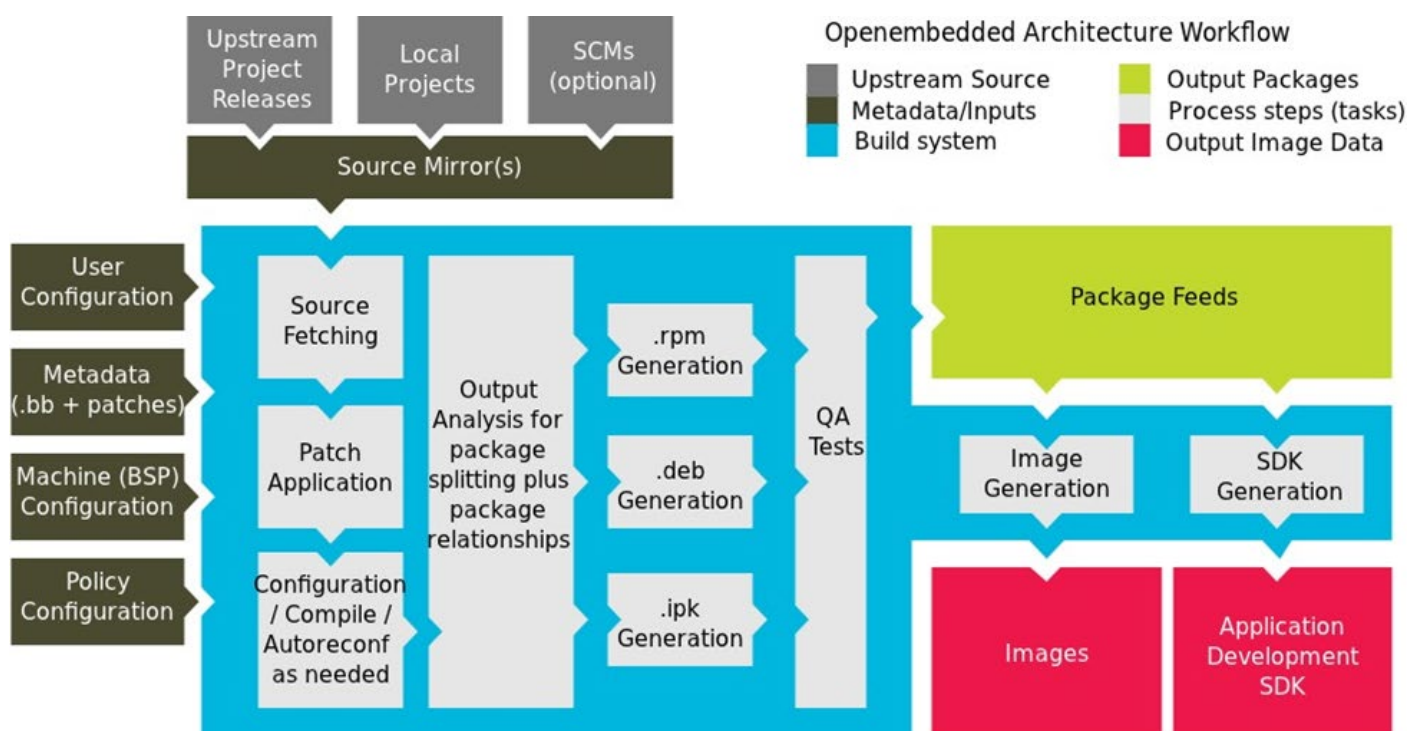


Figure 3.1 Yocto Project Task Process

3.4 Composition of D3-G SDK

After D3-G SDK is downloaded, the directory configuration is as follows.

Table 3.1 Composition of D3-G SDK

File		Description
easy-setup_ai.sh		Python script to automatically download and build the SDK Note: This is a symbolic link to “tools/ai-g/easy-setup.sh”.
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 <i>FWDN</i>
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-topst-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 D3-G image.

3.5.1 Set Email and Username in .gitconfig

To download the Yocto image for D3-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 set the username and email.

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

3.5.2 Get D3-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 (2.54) 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.

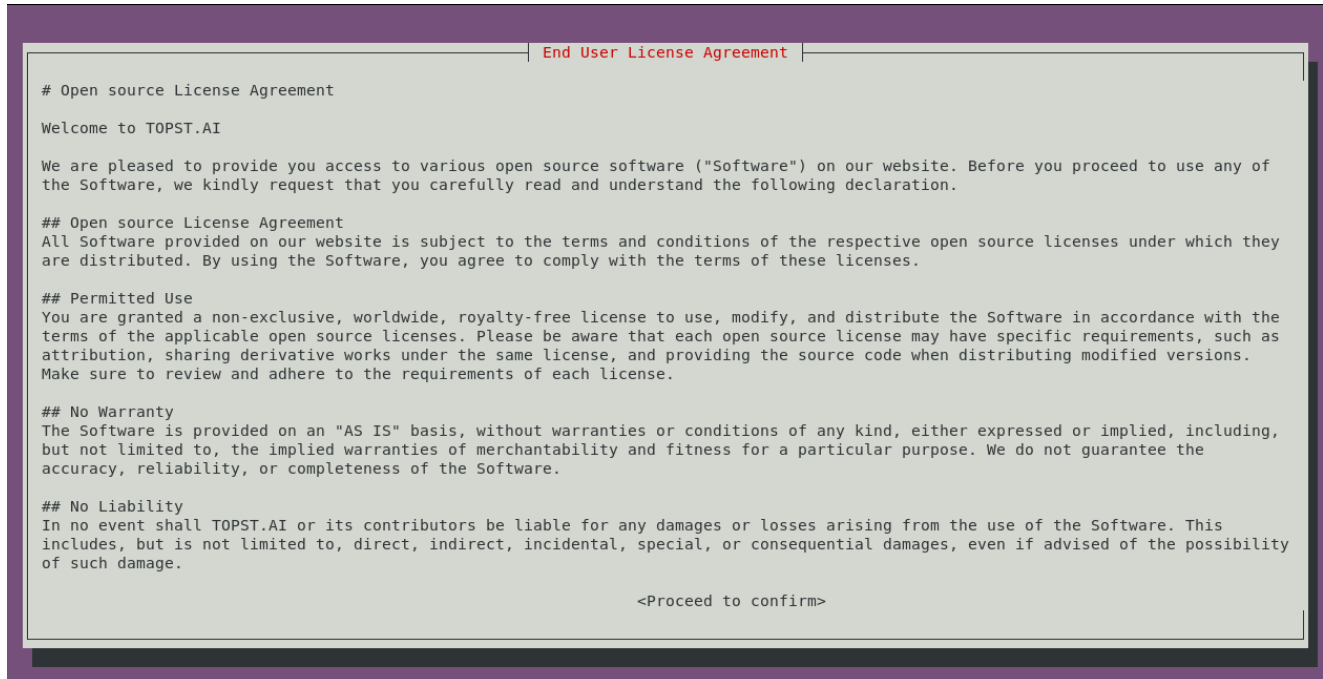
Fetching: 100% (12/12), done in 33.103s
Checking out: 25% (3/12), done in 0.863s
Checking out: 75% (9/12), done in 0.415s
repo sync has finished successfully.
```

3.6 Execute Build Script

If you run `./easy-setup.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
```



The screenshot shows a terminal window with a purple border. At the top, a title bar reads "End User License Agreement". The terminal text is as follows:

```
# 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.

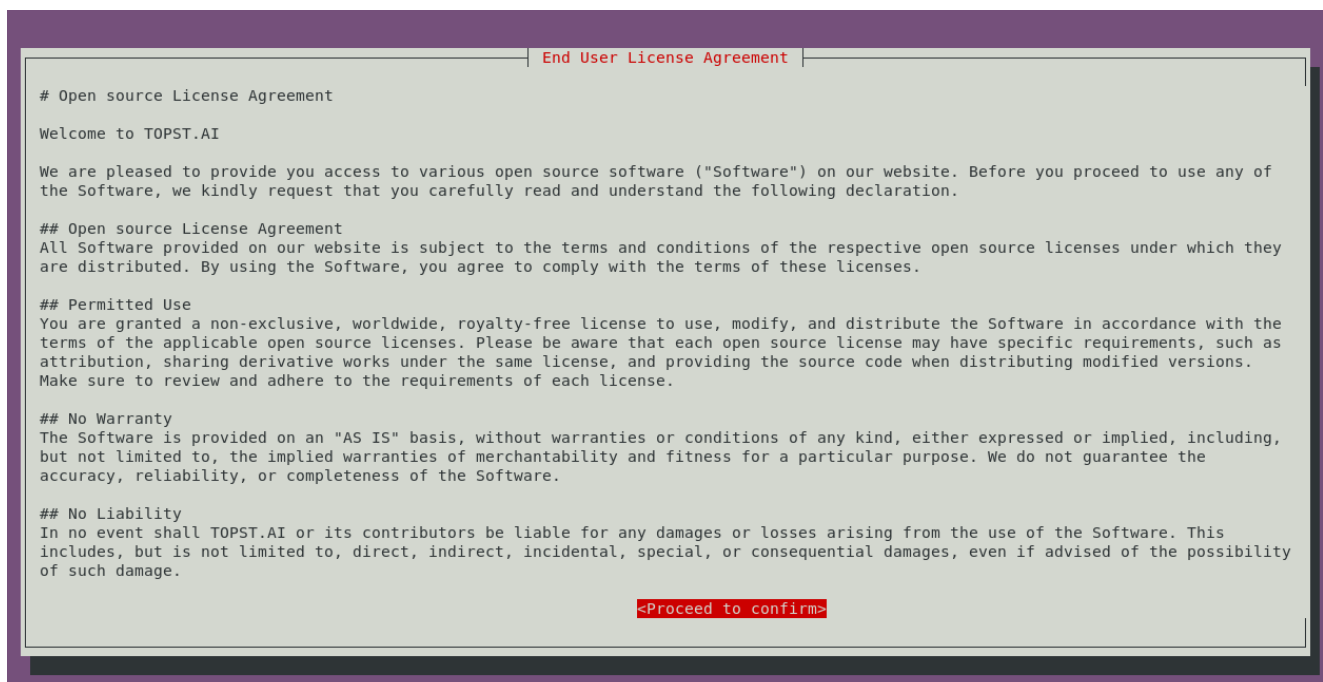
## No Warranty
The Software is provided on an "AS IS" basis, without warranties or conditions of any kind, either expressed or implied, including,
but not limited to, the implied warranties of merchantability and fitness for a particular purpose. We do not guarantee the
accuracy, reliability, or completeness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



This screenshot is identical to the previous one, but the prompt at the bottom, `<Proceed to confirm>`, is highlighted in red to indicate where the user should press the right arrow key and Enter.

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 the D3-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 2, you are now ready to install the main OS on the D3-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) => 2
machine(d3-g-topst-main) selected.
You had no conf/local.conf file. This configuration file has therefore been
created for you from /home/sooyong/topst_dev/poky/meta-topst/template/d3-g-topst-main/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/sooyong/topst_dev/poky/meta-topst/template/d3-g-topst-
main/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-minimal  
  telechips-topst-image-multimedia  
  telechips-topst-image  
  
  meta-toolchain-topst(Application Development Toolkit)
```

You can also run generated TOSPT images on D3-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 D3-G image by using the following command.

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

The build image is created in the following path:

- {TOPST_PATH}/build/d3-g-topst-main/tmp/deploy/images/d3-g-topst-main

3.7 Make Firmware Downloader (FWDN) Image

This option combines the binaries generated after building the SDK into a single image for the D3-G board.

The "output_d3g.fwdn.zip" including the output_d3g.fai build image and **FWDN** tool is created in the following path:

- ~/topst-sdk/

```
$ cd ~/topst-sdk
$ ./stitch-fai-d3.sh -f

Filesystem too small for a journal
[mktcimg] v1.2.1 - Nov 15 2021 19:33:18
location : bl3_ca72_a
location : 4096 sector(2097152 byte)
location : build-main/tmp/deploy/images/tcc8050-main/ca72_bl3.rom
location : boot
location : 122880 sector(62914560 byte)
location : build-main/tmp/deploy/images/tcc8050-main/tc-boot-tcc8050-main.img
location : system
location : 11534336 sector(5905580032 byte)
location : build-main/tmp/deploy/images/tcc8050-main/topst-tcc8050-main.ext4
location : dtb
location : 400 sector(204800 byte)
location : build-main/tmp/deploy/images/tcc8050-main/tcc8050-linux-topst-D3-G-pre-v0.1.dtb
location : env
location : 2048 sector(1048576 byte)
location : misc
location : 2048 sector(1048576 byte)
location : splash
location : 81920 sector(41943040 byte)
location : home
location : 3516416 sector(1800404992 byte)
location : /home/TOPST/topst-d3g/.stitch_XXRN1st/home-directory.ext4
location : data
location : 2048 sector(1048576 byte)
location : /home/TOPST/topst-d3g/.stitch_XXRN1st/user-data.ext4
path : build-main/tmp/deploy/images/tcc8050-main/ca72_bl3.rom
uuid : 372cbca7-25e2-480b-b8ec-2736ed1f02d0 , part-name : bl3_ca72_a
uuid : b12c0ebc-2802-4d04-92ae-a52bbc3082e5 , part-name : boot
uuid : bbd5770e-1f3d-4bc6-ac17-4abeecf14a1e , part-name : system
uuid : 541d9405-8d40-43cf-9255-74fc5c0fbef7 , part-name : dtb
uuid : d71a2a08-9ffe-4690-8b27-c6c371f5b4a8 , part-name : env
uuid : e9e9b9f3-27b9-4822-9ab5-bb09554dfde2 , part-name : misc
uuid : 49578c16-ea11-4f27-bc82-c1f9a60ce33b , part-name : splash
uuid : 2cf93985-4ae4-4ea0-bc67-6c991ac8eb7d , part-name : home
uuid : ccc4aaaa-92a1-4e02-b55d-215a0ad92d29 , part-name : data
crc32 of header : f33d4c30
crc32 of partition array : a65980dc
idx : 0 bl3_ca72_a
idx : 1 boot
idx : 2 system
idx : 3 dtb
idx : 7 home
idx : 8 data
crc32 of header : f33d4c30
crc32 of partition array : 61f17e08
Complete to make fai file

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

--storage_size : 7818182656
--parttype : gpt
--area_name : "SD Data"
--outfile : /home/TOPST/topst-d3g/.stitch_XXRN1st/output.fai
--gptfile : /home/TOPST/topst-d3g/.stitch_XXRN1st/output.gpt
--fplist : /home/TOPST/topst-d3g/.stitch_XXRN1st/partition.single.list
--sector_size : 512
--sparse_fill : 0
```

```

=====
[+] Packaging FWDN binaries
  adding: output.fai (deflated 97%)
  adding: VtcUsbPort.dll (deflated 64%)
  adding: boot-firmware/ (stored 0%)
  adding: fwdn (deflated 63%)
  adding: fwdn.bat (deflated 40%)
  adding: fwdn.exe (deflated 57%)
  adding: fwdn.sh (deflated 44%)
  adding: output.gpt (deflated 98%)
  adding: output.gpt.back (deflated 97%)
  adding: output.gpt.prim (deflated 97%)
  adding: boot-firmware/bconf.dual.bin (deflated 94%)
  adding: boot-firmware/bconf.single.bin (deflated 93%)
  adding: boot-firmware/boot.dual.json (deflated 86%)
  adding: boot-firmware/boot.single.json (deflated 86%)
  adding: boot-firmware/ca53_b11.rom (deflated 51%)
  adding: boot-firmware/ca53_b12.rom (deflated 50%)
  adding: boot-firmware/ca72_b11.rom (deflated 51%)
  adding: boot-firmware/ca72_b12.rom (deflated 49%)
  adding: boot-firmware/dram_params.bin (deflated 78%)
  adding: boot-firmware/fwdn.json (deflated 41%)
  adding: boot-firmware/fwdn.rom (deflated 46%)
  adding: boot-firmware/hsm.bin (deflated 51%)
  adding: boot-firmware/hsm.cs.bin (deflated 13%)
  adding: boot-firmware/mcert.bin (deflated 96%)
  adding: boot-firmware/optee.rom (deflated 92%)

  adding: boot-firmware/scfw.rom (deflated 53%) Filesystem too small for a journal [mktcimg] v1.2.1 - Nov
15 2021 19:33:18 location : b13_ca72_a location : 4096 sector(2097152 byte) location : build/d3-g-topst-
main/tmp/deploy/images/d3-g-topst-main/ca72_b13.rom location : boot location : 122880 sector(62914560
byte) location : build/d3-g-topst-main/tmp/deploy/images/d3-g-topst-main/tc-boot-d3-g-topst-main.img
location : system location : 33554432 sector(17179869184 byte) location : build/d3-g-topst-
main/tmp/deploy/images/d3-g-topst-main/telechips-topst-image-d3-g-topst-main.ext4 location : dtb
location : 400 sector(204800 byte) location : build/d3-g-topst-main/tmp/deploy/images/d3-g-topst-
main/tcc8050-topst-d3-g.dtb path : build/d3-g-topst-main/tmp/deploy/images/d3-g-topst-main/ca72_b13.rom
uuid : 7eb23c82-ccc0-44ce-8237-3315fc34e3f5 , part-name : b13_ca72_a uuid : 1c76ef36-314d-4548-8207-
5ab1d1376ca2 , part-name : boot uuid : b32eb80f-e014-4f17-b140-77bf3e137ba0 , part-name : system uuid :
429d8444-87b0-4c1d-8b3f-278dec2616f3 , part-name : dtb crc32 of header : 2a7c0194 crc32 of partition
array : b181e432 idx : 0 b13_ca72_a idx : 1 boot idx : 2 system idx : 3 dtb crc32 of header : 2a7c0194
crc32 of partition array : 990446d3 Complete to make fai file




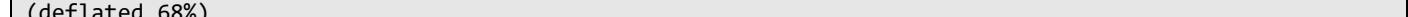









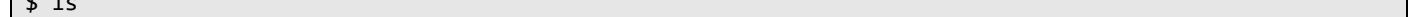
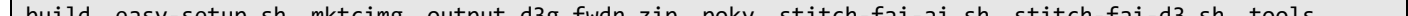




















































































































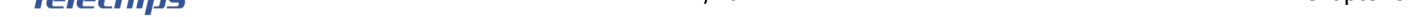




===== arguments info =====

--storage_size : 17818182656 --parttype : gpt --area_name : "SD Data" --outfile : /home/topst/topst-
sdk/.stitch_tOPE26E/output_d3g.fai --gptfile : /home/topst/topst-sdk/.stitch_tOPE26E/output_d3g.gpt --
fplist : /home/topst/topst-sdk/.stitch_tOPE26E/partition.single.list --sector_size : 512 --sparse_fill :
0

=====

[+] Packaging FWDN binaries adding: boot-firmware/ (stored 0%) adding: boot-firmware/boot.dual.json
(deflated 87%) adding: boot-firmware/prebuilt/ (stored 0%) adding: boot-
firmware/prebuilt/subcore_optee.rom (deflated 93%) adding: boot-firmware/prebuilt/mcert.bin (deflated
96%) adding: boot-firmware/prebuilt/fwdn.rom (deflated 49%) adding: boot-firmware/prebuilt/bconf.dual.bin
(deflated 95%) adding: boot-firmware/prebuilt/ca72_b11.rom (deflated 53%) adding: boot-
firmware/prebuilt/dram_params.bin (deflated 81%) adding: boot-firmware/prebuilt/hsm.cs.bin (deflated 13%)
adding: boot-firmware/prebuilt/ca72_b12.rom (deflated 54%) adding: boot-firmware/prebuilt/ca53_b11.rom
(deflated 53%) adding: boot-firmware/prebuilt/optee.rom (deflated 93%) adding: boot-
firmware/prebuilt/ca53_b12.rom (deflated 52%) adding: boot-firmware/prebuilt/hsm.bin (deflated 49%)
adding: boot-firmware/prebuilt/bconf.single.bin (deflated 93%) adding: boot-firmware/prebuilt/scfw.rom
(deflated 57%) adding: boot-firmware/prebuilt/tcc8050_snor.cs.rom (deflated 93%) adding: boot-
firmware/boot.single.json (deflated 87%) adding: boot-firmware/fwdn.json (deflated 50%) adding: fwdn
(deflated 69%) adding: fwdn.bat (deflated 40%) adding: fwdn.exe (deflated 62%) adding: fwdn.sh (deflated
40%) adding: output_d3g.fai (deflated 73%) adding: output_d3g.gpt (deflated 99%) adding:
output_d3g.gpt.back (deflated 98%) adding: output_d3g.gpt.prim (deflated 98%) adding: VtcUsbPort.dll

```

4 FIRMWARE DOWNLOADER

This chapter describes how to download Firmware Downloader (**FWDN**) to the D3-G and log in to 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)

FWDN writes a ROM image to the D3-G board through USB communication on the Host PC.

The D3-G board has one Boot Mode button and supports two types of boot modes:

- USB Boot Mode (FWDN Mode) : Used to write a ROM image by using the **FWDN** program on your Host PC
- eMMC Boot Mode : Used to boot the D3-G board by using a ROM image that is stored in an eMMC device

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

To use **FWDN**, connect the D3-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.
3. To enter USB Boot mode, connect the power cable to the D3-G board while pressing the FWDN switch.
4. Connect the USB Type-C cable to the USB Type-C FWDN Port on the D3-G board and the Host PC.

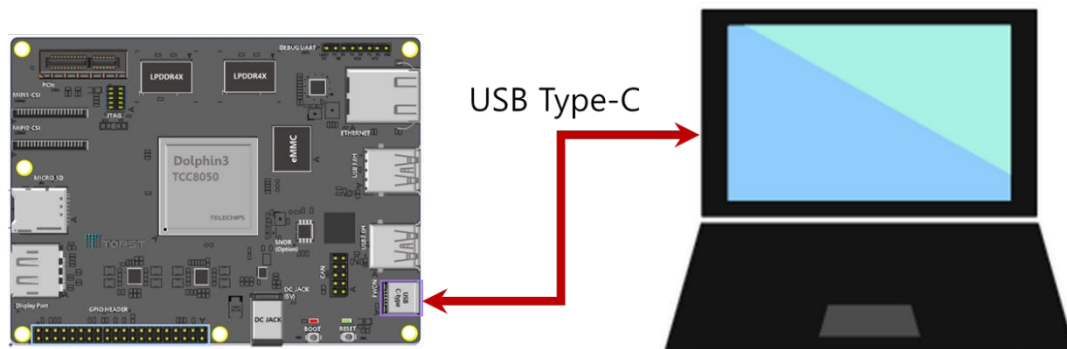


Figure 4.1 Connection Between Host PC and D3-G Board for FWDN

4.2.1 How to install VTC 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 and Figure 4.3.

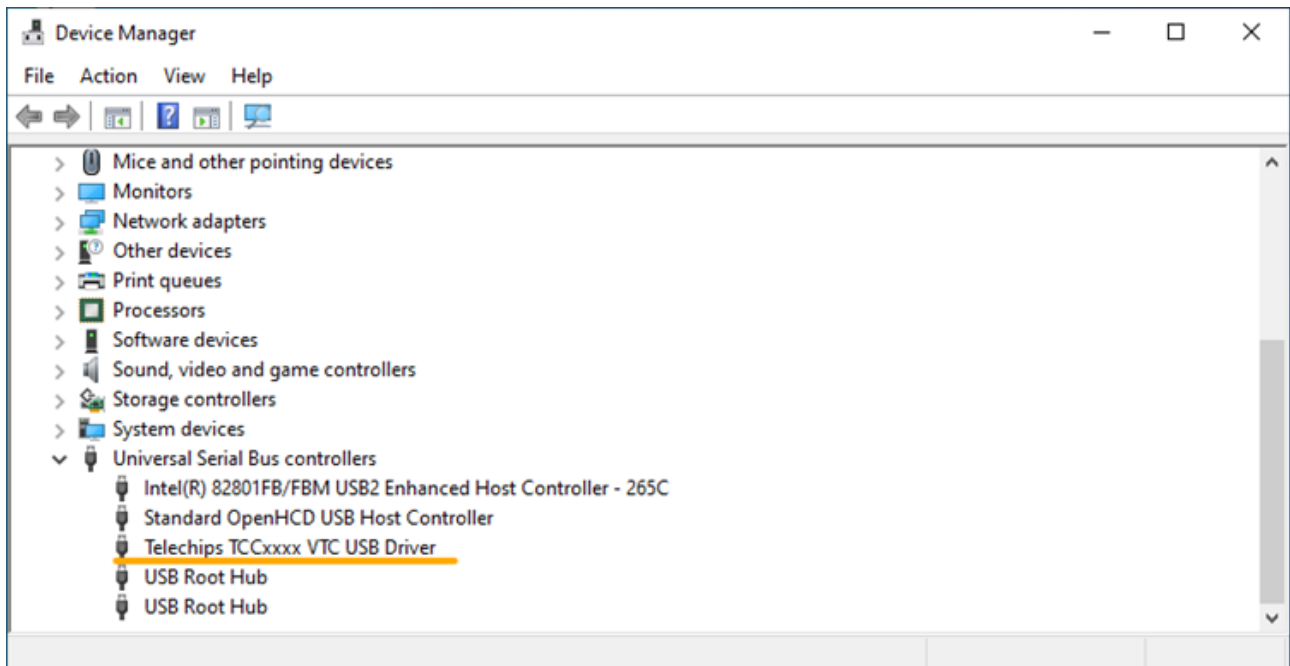


Figure 4.2 USB Connection in Windows Environment

```
test@ubuntu:~/build$ lsusb
Bus 001 Device 002: ID 140e:b201 Telechips, Inc.
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 002 Device 001: ID 1d6b:0001 Linux Foundation 1.1 root hub
```

Figure 4.3 USB Connection in Linux Environment

Important: Use VTC driver V5.0.0.14 or higher. To check the version, check the Device Manager in the Windows environment.

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_d3g.fwdn.zip".

```
$ cd ~/topst-sdk
$ mkdir images
$ mv ./output_d3g.fwdn.zip ./images
$ cd images
$ unzip output_d3g.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
```

2. Enter the `.\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. Refer to Chapter 4.4 to start communicating with the host PC.

4.3.2 Execute FWDN in Linux Environment

Download D3-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 D3-G Board with Host PC

This chapter explains how to connect the host PC to the D3-G board through UART for firmware download and serial communication.

4.4.1 Connect D3-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 UART Pin on the D3-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 (bps)** to **115200** and **Flow control** to **None**.
9. Connect the power cable. Then, the D3-G board boots in the default eMMC boot mode.

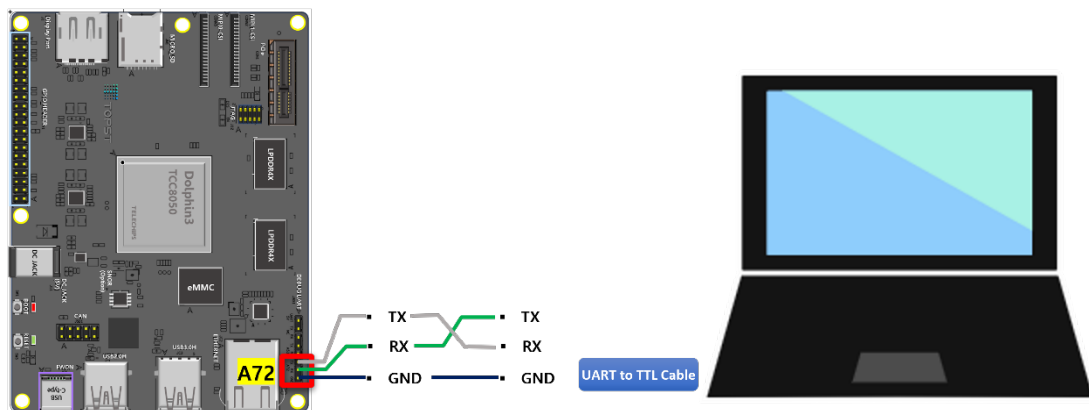


Figure 4.4 UART Connection with Host PC

Figure 4.5 shows a successful login.

Both the username and password for login are set to **root**.

```
U-Boot 2020.01-g101f36a6-dirty (May 13 2024 - 02:52:13 +0000)
CPU: ARM Cortex-A72 Processor, Rev: r0p3
Model: Telechips TCC8050 TOPST D3 PRE
tcc_sc_mmc: 0
Loading Environment from MMC... *** Warning - bad CRC, using default environment

In: serial@16600000
Out: serial@16600000
Err: serial@16600000
write error to device: 0000000033f3fa70 register: 0x2d!
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 0x20080000 size 12825 KiB
Kernel command line: root=/dev/ram quiet fsck.repair=yes
## Flattened Device Tree blob at 23000000
Booting using the fdt blob at 0x23000000
Loading Kernel Image
Using Device Tree in place at 0000000023000000, end 0000000023021fff
OPTEE may not be stored in storage.

Starting kernel ...

[ 0.049289] [ERROR][PL011] no SC node
[ 0.052959] [ERROR][PL011] no SC firmware handle
[ 1.140859] telechips-pcie 11000000.pcie: timeout waiting for link_up
Mounting devtmpfs
Mounting procfs
Mounting sysfs
[ 1.358924] EXT4-fs (mmcblk0p3): Mount option "noload" incompatible with ext2
[ 1.428239] systemd[1]: Failed to find module 'autofs4'

Telechips Baseline (Poky/meta-telechips/meta-core) 3.1.15 telechips-tcc8050-main ttyAMA0
telechips-tcc8050-main login: root
Password:
root@telechips-tcc8050-main:~#
```

Figure 4.5 Connected Screen (ID and Password are root)

5 APPENDIX

Ubuntu OS is also provided. By following this chapter, you can download the Ubuntu image, upload it to the D3-G board, and expand the allocated eMMC storage capacity.

5.1 Download Ubuntu Image

1. Download the D3-G_Ubuntu folder.
 - https://drive.google.com/file/d/1oc2qwaXUt6-QDME3s5WXKVHzAg4xqVyc/view?usp=drive_link
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.

5.2 Resize eMMC Storage (TOPST D3-G Ubuntu)

When you log in after booting the board, you must resize the eMMC storage first. Follow these steps to resize eMMC storage.

1. To resize partitions by modifying size and layout on the disk, use the following command.

```
$ parted
```

2. Extend GUID Partition Table (GPT) partition.

```
$ rescue
```

```
$ Fix
```

```
$ 0
```

```
$ 100%
```

3. Use the **p** (print) command to check that the partition type is ext4.

```
$ p
```

4. Resize partition 4.

```
$ resizepart 4
```

```
$ Yes
```

```
$ 100%
```

5. Reboot the board.
6. Resize the ext4 filesystem on partition 4.

```
$ resize2fs /dev/mmcblk0p4
```

7. Check the changed partition size by using the following command.

```
$ df -h
```

You can confirm that the available space is 27 GB after resizing.

6 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.

7 REVISION HISTORY

Rev. 1.01: 2025-07-18

- Updated
 - Chapter 2: Description
 - 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 D3-G SDK" to "SDK Build Preparation"
 - Description
 - Chapter 3.3: 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 D3-G SDK from Git" to "Get D3-G SDK from Git"
 - Description
 - Chapter 3.6: Description
 - Chapter 3.7: Description
 - Chapter 4: Description
 - Chapter 4.2:
 - Description
 - Figure 4.1
 - Chapter 4.2.1: Description
 - Chapter 4.3: Description
 - Chapter 4.3.1: Description
 - Chapter 4.3.2: Description
 - Chapter 4.4.1:
 - Changed chapter title from "TOPST D3-G Connection with UART" to "Connect D3-G Board with UART"
 - Figure 4.4
 - Chapter 5: Description
 - Chapter 5.1: Changed D3-G Ubuntu folder link
- Added
 - Chapter 2.1: Windows Environment
 - Chapter 2.2.1: Setting Environment
 - Chapter 4.4: Connect D3-G Board with Host PC
- Deleted
 - Chapter 1.3: Ubuntu Environment
 - Chapter 2.4.1: Set Locale
 - Chapter 3.6.1: Execute Build Script

Rev. 1.00: 2025-02-28

- 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."