

# **TOPST VCP-G FreeRTOS 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](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 Windows Environment .....	4
2.1.1 Install WSL2 Ubuntu (Windows Environment Only).....	4
2.2 Linux Environment.....	5
3 Toolchain .....	6
3.1 Install and Set Up Toolchain .....	6
3.2 Verify Toolchain .....	8
4 Clone Source Code.....	9
4.1 Clone VCP-G Source Code .....	9
4.2 Source Code Structure .....	10
5 Build Guide.....	11
5.1 Execute Build Script .....	11
5.2 Makefiles and Build Systems .....	12
5.3 Initiate Build Process .....	12
6 Firmware Downloader .....	15
6.1 Prepare VCP-G Board .....	15
6.2 Connect Hardware to Host PC.....	15
6.3 Download Software on VCP-G Board .....	16
6.3.1 Execute FWDN in Windows Environment.....	16
6.3.2 Execute FWDN in Linux Environment .....	18
6.4 Verify Software on Board.....	20
6.5 Troubleshooting Common Issues.....	22
7 References .....	25
8 Revision History .....	26
Rev. 1.01: 2025-07-18.....	26
Rev. 1.00: 2025-02-28.....	26

## Figures

Figure 3.1 Download Toolchain .....	6
Figure 3.2 Extract Toolchain.....	6
Figure 3.3 Move Toolchain.....	7
Figure 3.4 Navigate to Toolchain Directory .....	8
Figure 3.5 Check Version of Installed GCC Compiler .....	8
Figure 4.1 Navigate to Desired Directory .....	9
Figure 4.2 Clone Repository.....	9
Figure 4.3 Navigate to Cloned Directory.....	10
Figure 4.4 Source Code Structure.....	10
Figure 5.1 End User License Agreement.....	11
Figure 5.2 Go to 'Proceed to confirm'.....	11
Figure 5.3 Accept Screen.....	12
Figure 5.4 Run make Command .....	13
Figure 5.5 Verify Build Output.....	13
Figure 5.6 Build Output File .....	14
Figure 6.1 Verify USB Connection .....	15
Figure 6.2 Set Board to Download Mode .....	16
Figure 6.3 Click fwdn_vcp.bat .....	16
Figure 6.4 Reset Board.....	18
Figure 6.5 Set Board to Download Mode .....	18
Figure 6.6 Execute Download Command .....	19
Figure 6.7 Reset Board .....	19
Figure 6.8 Install Minicom .....	20
Figure 6.9 Open Serial Connection .....	20
Figure 6.10 Open Serial Connection.....	21
Figure 6.11 Modify User Group Permissions.....	22
Figure 6.12 Launch minicom .....	23
Figure 6.13 Access Set up Screen.....	23
Figure 6.14 Modify Flow Control.....	24
Figure 6.15 Save and Exit.....	24

# 1 INTRODUCTION

This document provides guidelines for building the VCP-G SDK, including setting up the software development environment, using the toolchain, cloning the source code, building the SDK, and using the firmware downloader.

This document includes information on the following:

- Setting Host Environment
- Toolchain
- Clone Source Code
- 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 chapter describes how to set up Windows Subsystem for Linux (WSL) to use Linux on a Windows PC. You can install another version of Linux, but in this document, VCP-G SDK is described based on Ubuntu 22.04. If the OS of your host PC is Ubuntu, proceed to Chapter 2.3.

#### 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 LTS in Microsoft Store and download it.
  - If you need to download the Linux kernel update package, download the latest package [here](#).

## 2.2 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 to ensure 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
```

3. Install Common Development Tools

Install common development tools by entering the following command:

```
sudo apt install build-essential git
```

**Note:** This command installs both the **build-essential** package and **git**.

## 3 TOOLCHAIN

The VCP-G board uses the `gcc-linaro-7.2.1-2017.11-x86_64_arm-eabi` toolchain.

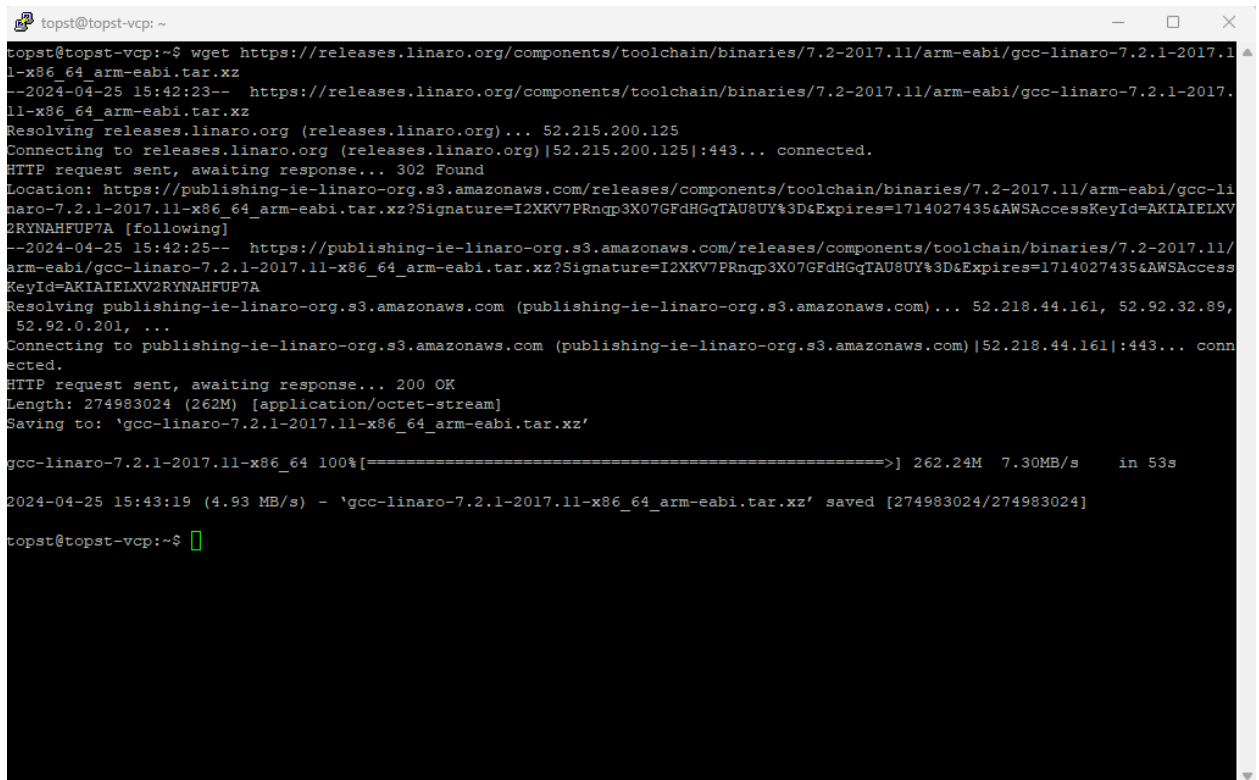
This toolchain is optimized for the Arm architecture and ensures compatibility with the TCC7045 chip on the VCP-G board.

### 3.1 Install and Set Up Toolchain

Follow these steps to download, extract, and set up the toolchain:

1. **Download the Toolchain:** Enter the `wget` command to download the toolchain from the Linaro website.

```
wget https://releases.linaro.org/components/toolchain/binaries/7.2-2017.11/arm-eabi/gcc-linaro-7.2.1-2017.11-x86_64_arm-eabi.tar.xz
```



```
topst@topst-vcp: ~
topst@topst-vcp:~$ wget https://releases.linaro.org/components/toolchain/binaries/7.2-2017.11/arm-eabi/gcc-linaro-7.2.1-2017.11-x86_64_arm-eabi.tar.xz
--2024-04-25 15:42:23-- https://releases.linaro.org/components/toolchain/binaries/7.2-2017.11/arm-eabi/gcc-linaro-7.2.1-2017.11-x86_64_arm-eabi.tar.xz
Resolving releases.linaro.org (releases.linaro.org)... 52.215.200.125
Connecting to releases.linaro.org (releases.linaro.org)|52.215.200.125|:443... connected.
HTTP request sent, awaiting response... 302 Found
Location: https://publishing-ie-linaro-org.s3.amazonaws.com/releases/components/toolchain/binaries/7.2-2017.11/arm-eabi/gcc-linaro-7.2.1-2017.11-x86_64_arm-eabi.tar.xz?Signature=I2XKV7PRnqp3X07GfDhGqTAU8UY%3D&Expires=1714027435&AWSAccessKeyId=AKIAIELXV2RYNAHFUP7A [following]
--2024-04-25 15:42:25-- https://publishing-ie-linaro-org.s3.amazonaws.com/releases/components/toolchain/binaries/7.2-2017.11/arm-eabi/gcc-linaro-7.2.1-2017.11-x86_64_arm-eabi.tar.xz?Signature=I2XKV7PRnqp3X07GfDhGqTAU8UY%3D&Expires=1714027435&AWSAccessKeyId=AKIAIELXV2RYNAHFUP7A
Resolving publishing-ie-linaro-org.s3.amazonaws.com (publishing-ie-linaro-org.s3.amazonaws.com)... 52.218.44.161, 52.92.32.89, 52.92.0.201, ...
Connecting to publishing-ie-linaro-org.s3.amazonaws.com (publishing-ie-linaro-org.s3.amazonaws.com)|52.218.44.161|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 274983024 (262M) [application/octet-stream]
Saving to: 'gcc-linaro-7.2.1-2017.11-x86_64_arm-eabi.tar.xz'

gcc-linaro-7.2.1-2017.11-x86_64 100%[=====>] 262.24M  7.30MB/s   in 53s

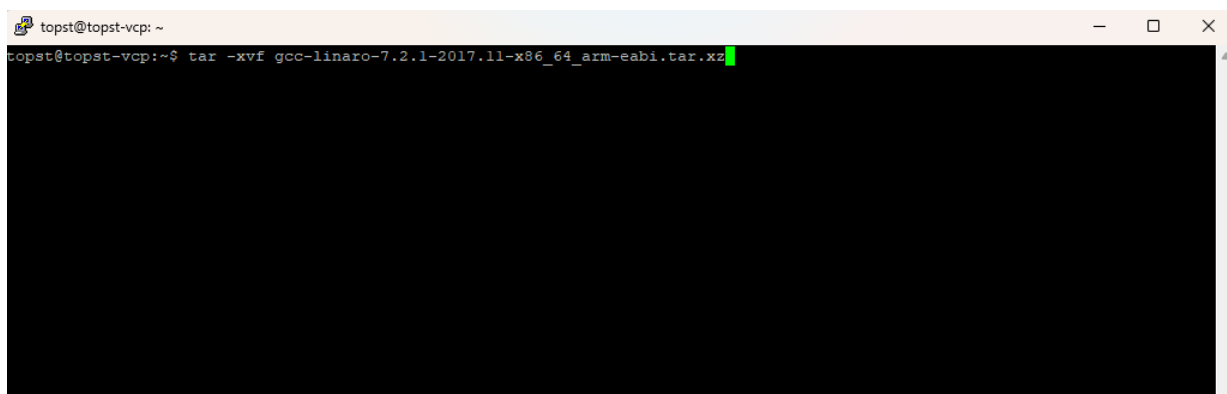
2024-04-25 15:43:19 (4.93 MB/s) - 'gcc-linaro-7.2.1-2017.11-x86_64_arm-eabi.tar.xz' saved [274983024/274983024]

topst@topst-vcp:~$
```

**Figure 3.1 Download Toolchain**

2. **Extract the Toolchain:** After the download is complete, extract the contents of the `.tar.xz` file.

```
tar -xvf gcc-linaro-7.2.1-2017.11-x86_64_arm-eabi.tar.xz
```

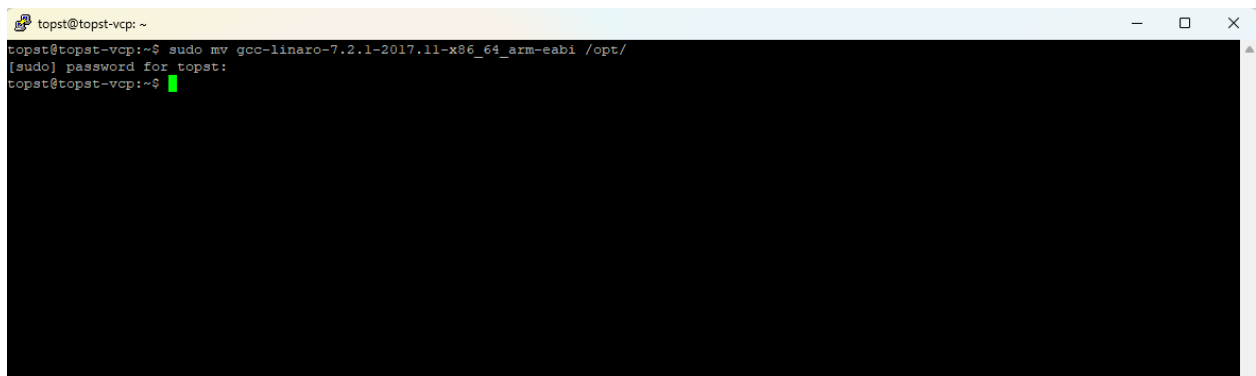


```
topst@topst-vcp: ~
topst@topst-vcp:~$ tar -xvf gcc-linaro-7.2.1-2017.11-x86_64_arm-eabi.tar.xz
```

**Figure 3.2 Extract Toolchain**

3. **Move the Toolchain to /opt:** The /opt directory is a standard location for optional software on Linux. Move the extracted toolchain to this directory.

```
sudo mv gcc-linaro-7.2.1-2017.11-x86_64_arm-eabi /opt/
```



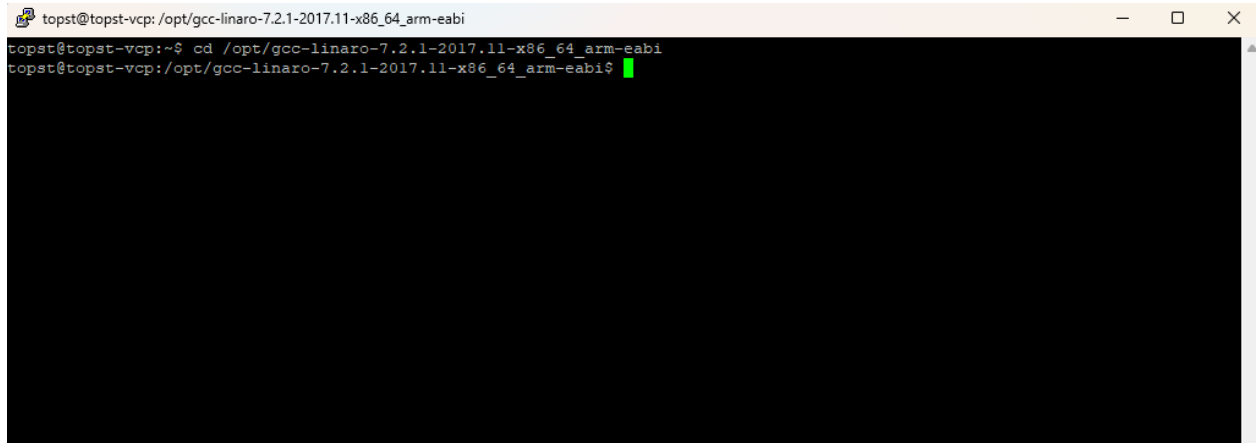
**Figure 3.3 Move Toolchain**

## 3.2 Verify Toolchain

To ensure that the toolchain is installed correctly:

1. Navigate to the Toolchain Directory.

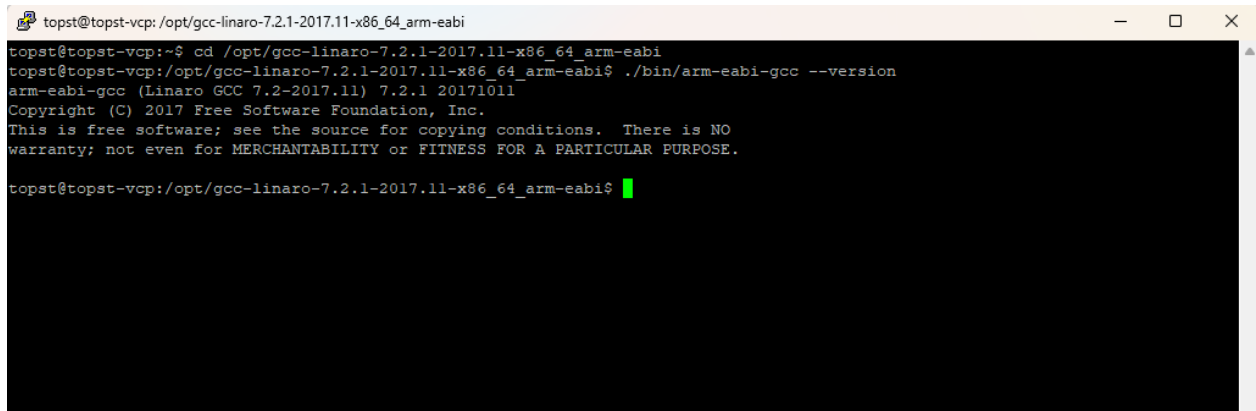
```
cd /opt/gcc-linaro-7.2.1-2017.11-x86_64_arm-eabi
```



**Figure 3.4 Navigate to Toolchain Directory**

2. Check the Version of the Installed GCC Compiler.

```
./bin/arm-eabi-gcc --version
```



**Figure 3.5 Check Version of Installed GCC Compiler**

After successfully installing the GCC Compiler, verify the installed GCC compiler version to ensure it matches **gcc-linaro-7.2.1-2017.11**.



## 4 CLONE SOURCE CODE

This chapter describes how to clone the source code using Git.

### 4.1 Clone VCP-G Source Code

To obtain the source code for the VCP-G, enter the `git clone` command. This command creates a copy of a remote repository on your local machine, allowing you to work with the code directly.

Follow these steps to clone the VCP-G source code:

1. **Open Terminal:** Launch the terminal application on your Ubuntu 22.04 system.
2. **Navigate to the Desired Directory:** Choose a suitable location to save the source code. For example, if you want to save the repository in the home directory, use the following command.

```
cd ~
```

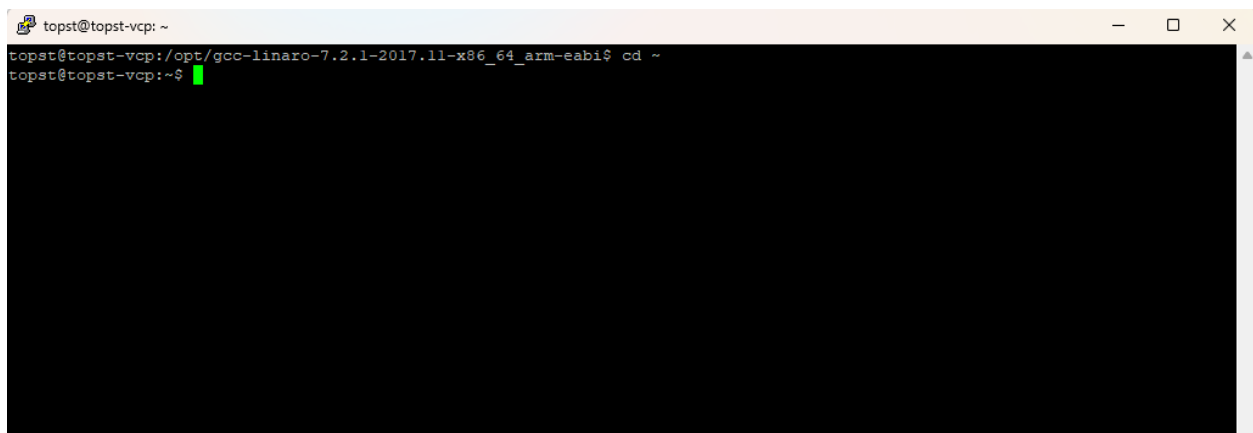


Figure 4.1 Navigate to Desired Directory

3. **Clone the Repository:** Use the following command to clone the VCP-G source code from the provided git address.

```
git clone https://github.com/topst-development/FreeRTOS-VCP.git topst-vcp
```

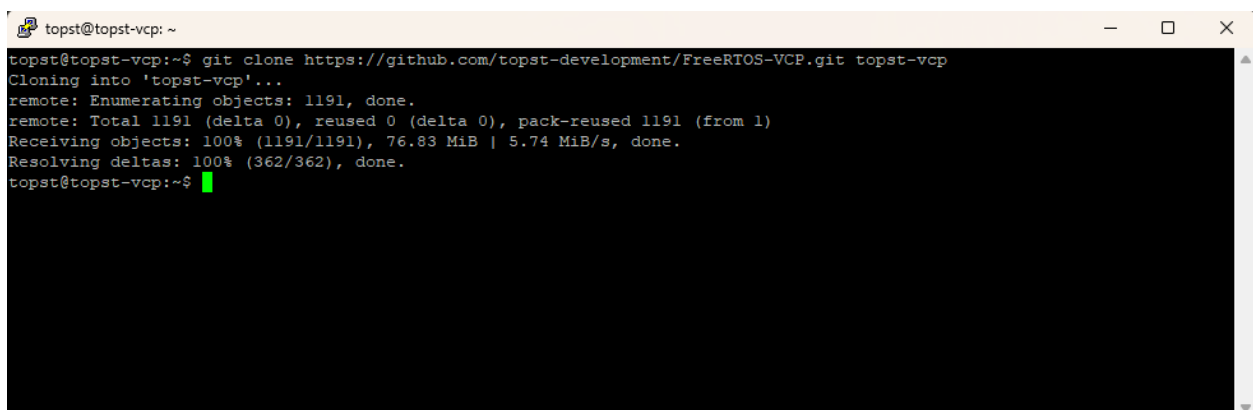
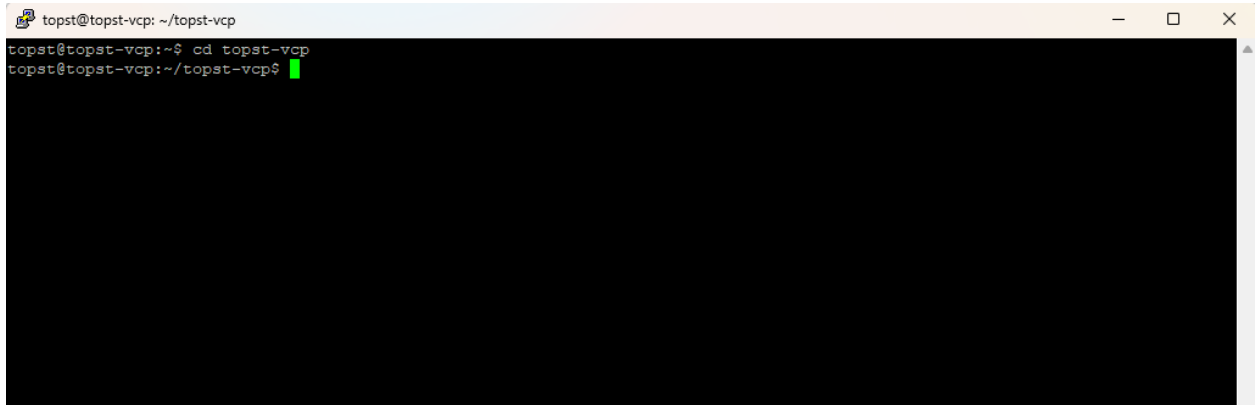


Figure 4.2 Clone Repository

4. **Navigate to the Cloned Directory:** After the cloning process is complete, use the following command to navigate to the directory containing the source code.

```
cd topst-vcp
```

A terminal window titled 'topst@topst-vcp: ~/topst-vcp' showing the command 'cd topst-vcp' being entered and executed. The prompt changes from '~\$' to '~/topst-vcp\$'.

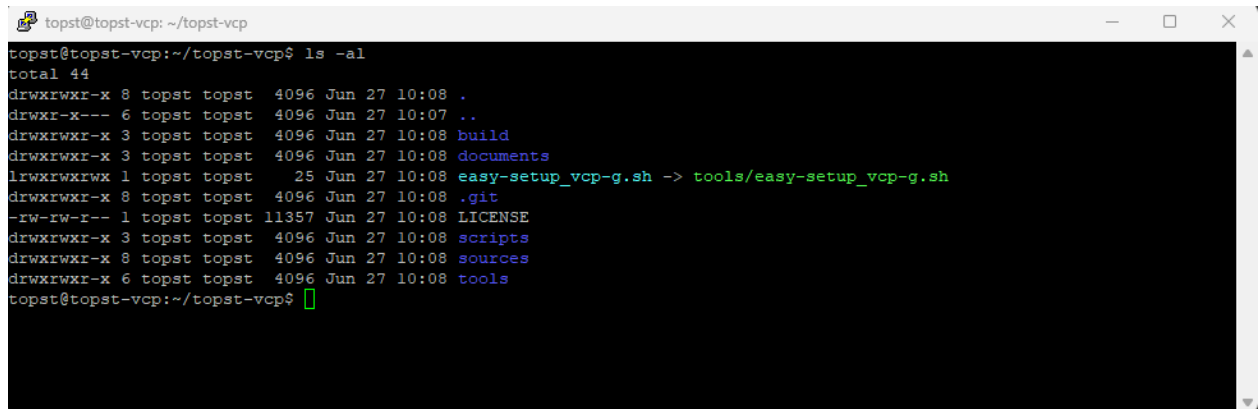
**Figure 4.3 Navigate to Cloned Directory**

The VCP-G source code is now available locally for building and development.

## 4.2 Source Code Structure

After cloning, enter the `ls` command to list the directory contents and review key files to understand the source code structure.

```
ls -al
```

A terminal window titled 'topst@topst-vcp: ~/topst-vcp' showing the output of the 'ls -al' command. The output lists the directory contents with permissions, owner, group, size, date, and filename. The files listed are: '.', '..', 'build', 'documents', 'easy-setup\_vcp-g.sh', '.git', 'LICENSE', 'scripts', 'sources', and 'tools'.

**Figure 4.4 Source Code Structure**

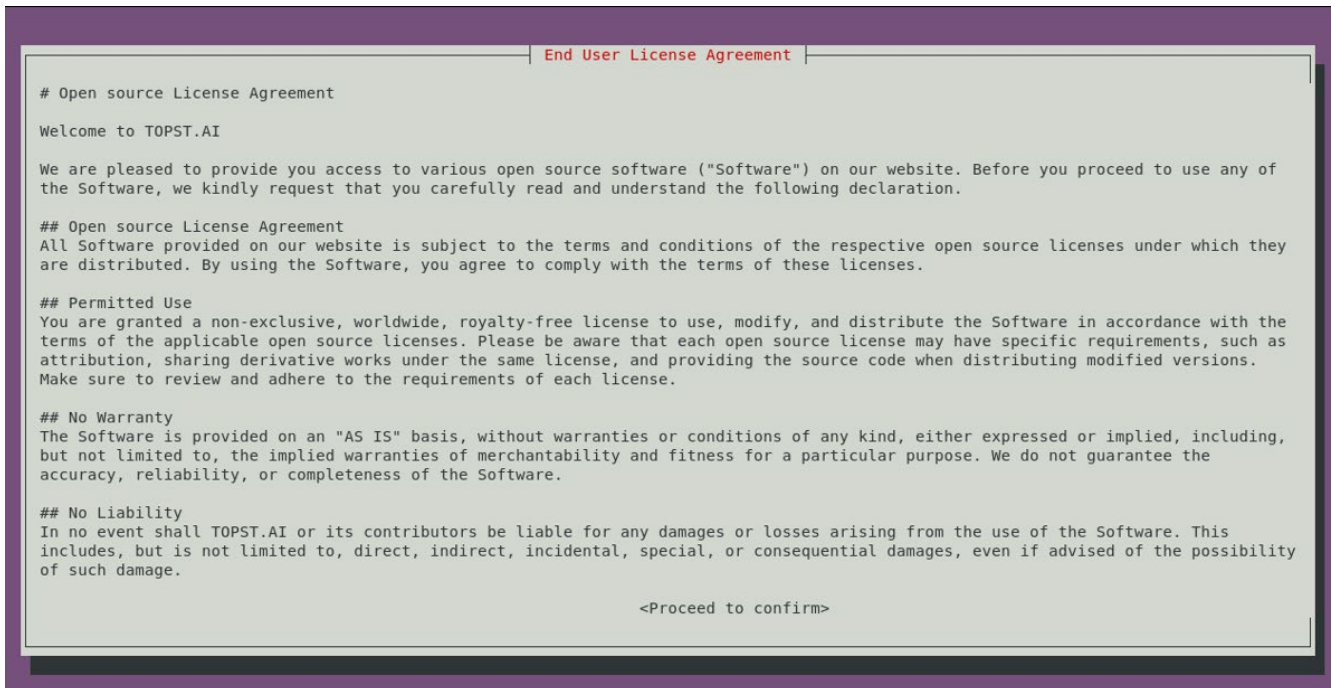
## 5 BUILD GUIDE

### 5.1 Execute Build Script

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

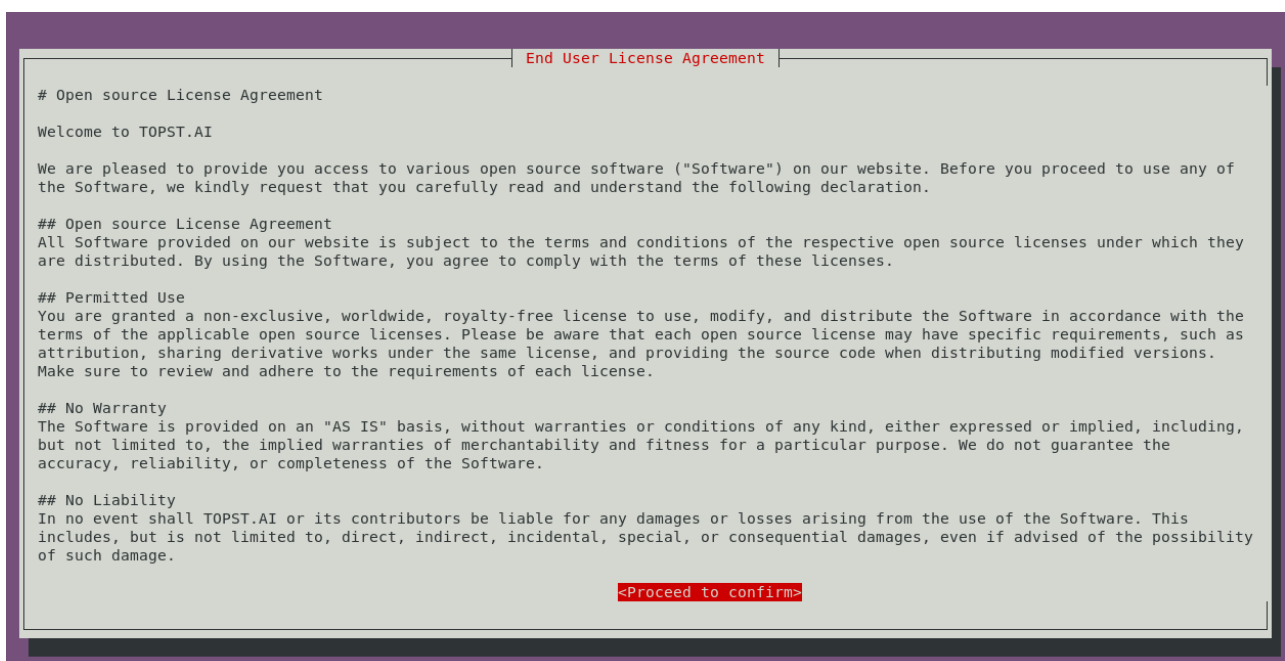
**Caution:** If you re-run `./easy-setup_vcp-g.sh`, be careful as the built sources will be deleted if you select **yes**.

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



**Figure 5.1 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]**.



**Figure 5.2 Go to 'Proceed to confirm'**

Then you can see the following screen.



**Figure 5.3 Accept Screen**

If you select Accept by pressing **[Enter]**, you can build the VCP-G SDK image by referring to Chapter 5.2 and Chapter 5.3.

## 5.2 Makefiles and Build Systems

A Makefile is a key component of many build systems. It contains rules and directives for the **make** utility to compile and link programs. By utilizing a Makefile, you can automate the build process, ensuring consistency and efficiency.

## 5.3 Initiate Build Process

To build the source code, follow these steps:

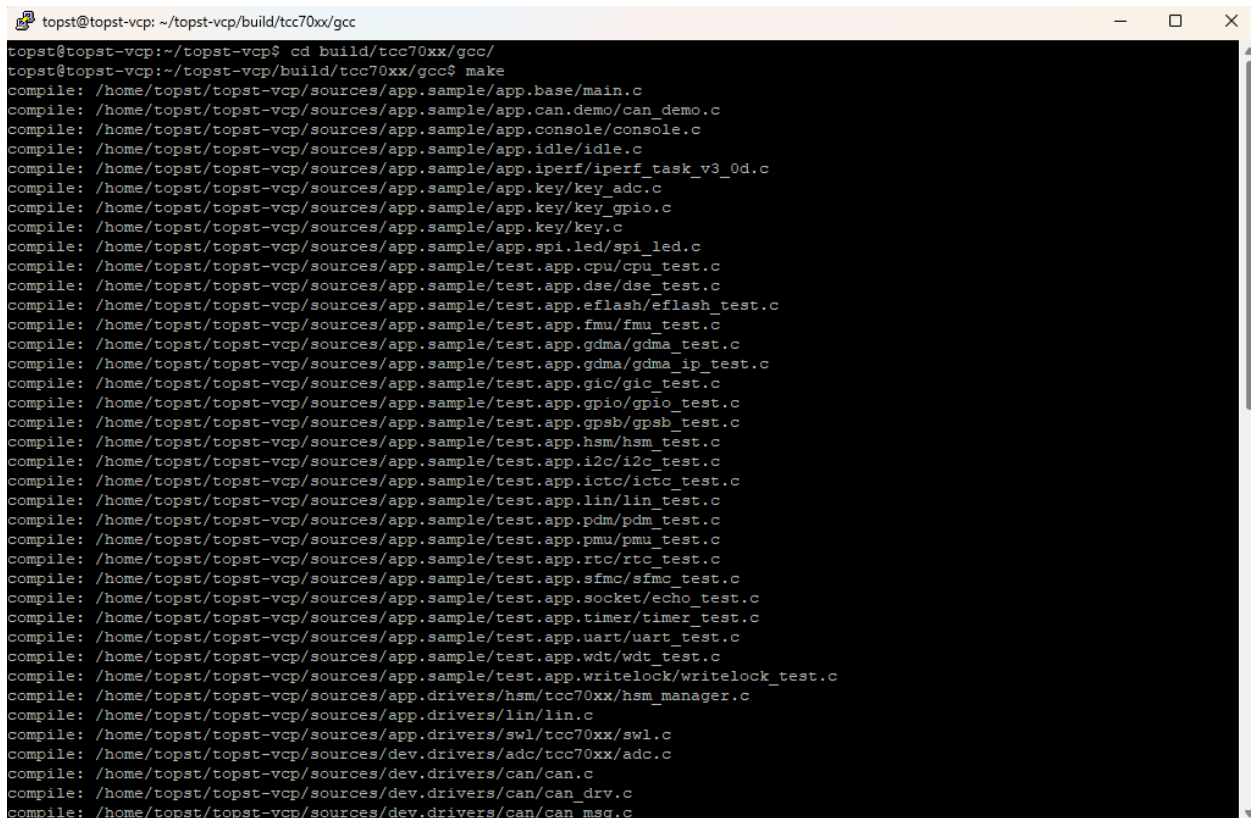
1. Navigate to the Build Directory.

```
cd build/tcc70xx/gcc/
```

2. Run the **make** command.

```
make
```

The **make** command reads the Makefile in the current directory and executes the build process.



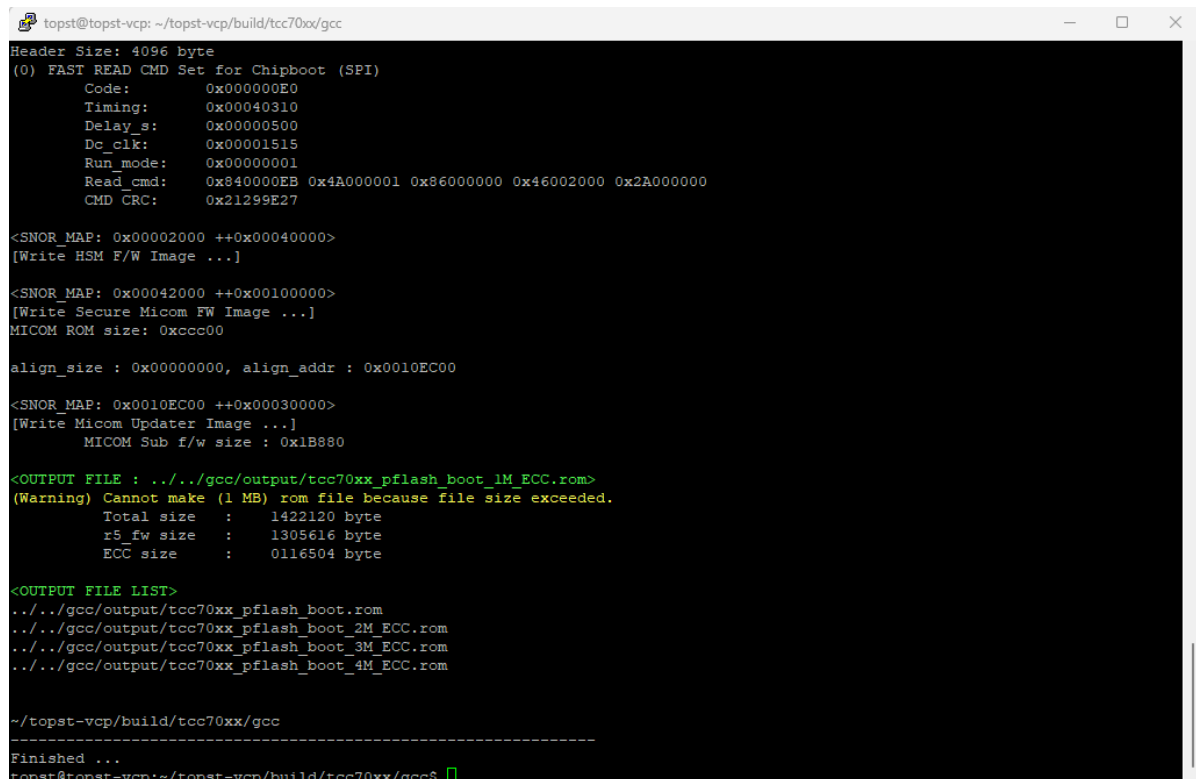
```

topst@topst-vcp: ~/topst-vcp/build/tcc70xx/gcc
topst@topst-vcp:~/topst-vcp/build/tcc70xx/gcc$ make
compile: /home/topst/topst-vcp/sources/app.sample/app.base/main.c
compile: /home/topst/topst-vcp/sources/app.sample/app.can.demo/can_demo.c
compile: /home/topst/topst-vcp/sources/app.sample/app.console/console.c
compile: /home/topst/topst-vcp/sources/app.sample/app.idle/idle.c
compile: /home/topst/topst-vcp/sources/app.sample/app.iperf/iperf_task_v3_0d.c
compile: /home/topst/topst-vcp/sources/app.sample/app.key/key_adc.c
compile: /home/topst/topst-vcp/sources/app.sample/app.key/key_gpio.c
compile: /home/topst/topst-vcp/sources/app.sample/app.key/key.c
compile: /home/topst/topst-vcp/sources/app.sample/app.spi.led/spi_led.c
compile: /home/topst/topst-vcp/sources/app.sample/test.app.cpu/cpu_test.c
compile: /home/topst/topst-vcp/sources/app.sample/test.app.dse/dse_test.c
compile: /home/topst/topst-vcp/sources/app.sample/test.app.eflash/eflash_test.c
compile: /home/topst/topst-vcp/sources/app.sample/test.app.fmu/fmu_test.c
compile: /home/topst/topst-vcp/sources/app.sample/test.app.gdma/gdma_test.c
compile: /home/topst/topst-vcp/sources/app.sample/test.app.gdma/gdma_ip_test.c
compile: /home/topst/topst-vcp/sources/app.sample/test.app.gic/gic_test.c
compile: /home/topst/topst-vcp/sources/app.sample/test.app.gpio/gpio_test.c
compile: /home/topst/topst-vcp/sources/app.sample/test.app.gpsb/gpsb_test.c
compile: /home/topst/topst-vcp/sources/app.sample/test.app.hsm/hsm_test.c
compile: /home/topst/topst-vcp/sources/app.sample/test.app.i2c/i2c_test.c
compile: /home/topst/topst-vcp/sources/app.sample/test.app.ictc/ictc_test.c
compile: /home/topst/topst-vcp/sources/app.sample/test.app.lin/lin_test.c
compile: /home/topst/topst-vcp/sources/app.sample/test.app.pdm/pdm_test.c
compile: /home/topst/topst-vcp/sources/app.sample/test.app.pmu/pmu_test.c
compile: /home/topst/topst-vcp/sources/app.sample/test.app.rtc/rtc_test.c
compile: /home/topst/topst-vcp/sources/app.sample/test.app.sfmc/sfmc_test.c
compile: /home/topst/topst-vcp/sources/app.sample/test.app.socket/echo_test.c
compile: /home/topst/topst-vcp/sources/app.sample/test.app.timer/timer_test.c
compile: /home/topst/topst-vcp/sources/app.sample/test.app.uart/uart_test.c
compile: /home/topst/topst-vcp/sources/app.sample/test.app.wdt/wdt_test.c
compile: /home/topst/topst-vcp/sources/app.sample/test.app.writelock/writelock_test.c
compile: /home/topst/topst-vcp/sources/app.drivers/hsm/tcc70xx/hsm_manager.c
compile: /home/topst/topst-vcp/sources/app.drivers/lin/lin.c
compile: /home/topst/topst-vcp/sources/app.drivers/swl/tcc70xx/swl.c
compile: /home/topst/topst-vcp/sources/dev.drivers/adc/tcc70xx/adc.c
compile: /home/topst/topst-vcp/sources/dev.drivers/can/can.c
compile: /home/topst/topst-vcp/sources/dev.drivers/can/can_drv.c
compile: /home/topst/topst-vcp/sources/dev.drivers/can/can_msg.c

```

Figure 5.4 Run make Command

3. **Verify the Build Output:** After the build process is complete, the following output files should be listed in the terminal.
  - output/tcc70xx\_pflash\_boot.rom
  - output/tcc70xx\_pflash\_boot\_2M\_ECC.rom
  - output/tcc70xx\_pflash\_boot\_3M\_ECC.rom
  - output/tcc70xx\_pflash\_boot\_4M\_ECC.rom



```

topst@topst-vcp: ~/topst-vcp/build/tcc70xx/gcc
Header Size: 4096 byte
(0) FAST READ CMD Set for Chipboot (SPI)
  Code:      0x000000E0
  Timing:    0x00040310
  Delay s:   0x00000500
  Dc_clk:    0x00001515
  Run_mode:  0x00000001
  Read_cmd:  0x840000EB 0x4A000001 0x86000000 0x46002000 0x2A000000
  CMD_CRC:   0x21299E27

<SNOR MAP: 0x00002000 ++0x00040000>
[Write HSM F/W Image ...]

<SNOR MAP: 0x00042000 ++0x00100000>
[Write Secure Micom FW Image ...]
MICOM ROM size: 0xc00000

align_size : 0x00000000, align_addr : 0x0010EC00

<SNOR MAP: 0x0010EC00 ++0x00030000>
[Write Micom Updater Image ...]
MICOM Sub f/w size : 0x1B880

<OUTPUT FILE : ../../gcc/output/tcc70xx_pflash_boot_1M_ECC.rom>
(Warning) Cannot make (1 MB) rom file because file size exceeded.
  Total size : 1422120 byte
  r5_fw size : 1305616 byte
  ECC size : 0116504 byte

<OUTPUT FILE LIST>
../../gcc/output/tcc70xx_pflash_boot.rom
../../gcc/output/tcc70xx_pflash_boot_2M_ECC.rom
../../gcc/output/tcc70xx_pflash_boot_3M_ECC.rom
../../gcc/output/tcc70xx_pflash_boot_4M_ECC.rom

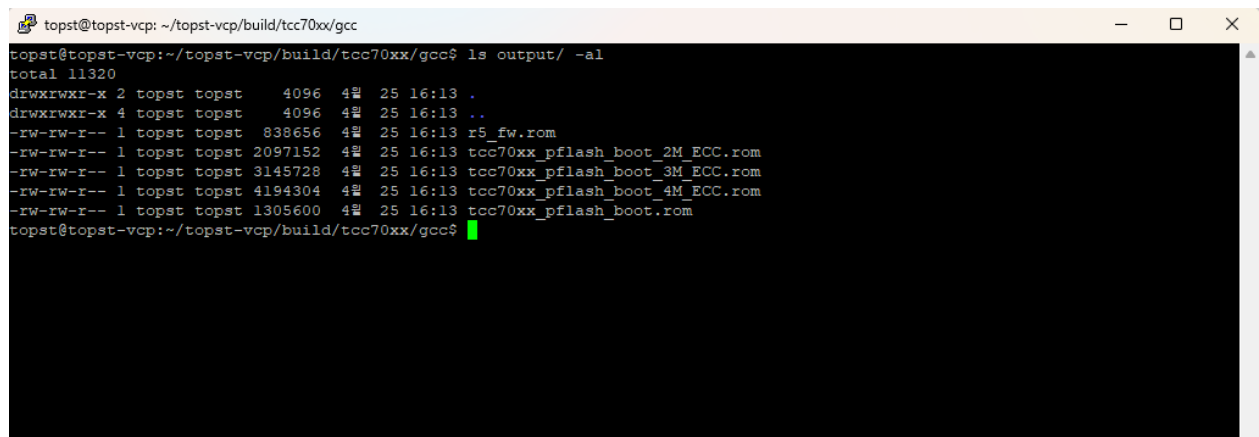
~/topst-vcp/build/tcc70xx/gcc
Finished ...
topst@topst-vcp:~/topst-vcp/build/tcc70xx/gcc$

```

Figure 5.5 Verify Build Output

To check the list of output files, use the following command:

```
ls output/ -al
```



```
topst@topst-vcv: ~/topst-vcv/build/tcc70xx/gcc$ ls output/ -al
total 11320
drwxrwxr-x 2 topst topst  4096 4월 25 16:13 .
drwxrwxr-x 4 topst topst  4096 4월 25 16:13 ..
-rw-rw-r-- 1 topst topst 838656 4월 25 16:13 r5_fw.rom
-rw-rw-r-- 1 topst topst 2097152 4월 25 16:13 tcc70xx_pflash_boot_2M_ECC.rom
-rw-rw-r-- 1 topst topst 3145728 4월 25 16:13 tcc70xx_pflash_boot_3M_ECC.rom
-rw-rw-r-- 1 topst topst 4194304 4월 25 16:13 tcc70xx_pflash_boot_4M_ECC.rom
-rw-rw-r-- 1 topst topst 1305600 4월 25 16:13 tcc70xx_pflash_boot.rom
topst@topst-vcv:~/topst-vcv/build/tcc70xx/gcc$
```

**Figure 5.6 Build Output File**

## 6 FIRMWARE DOWNLOADER

This chapter describes how to download Firmware Downloader (*FWDN*) to the VCP-G in a Linux-based development environment.

### 6.1 Prepare VCP-G Board

Before beginning the download process, ensure that the VCP-G board is in a stable position and free from any potential disturbances. Ensure that all switches and connectors are easily accessible and the 3.3V power cable is connected correctly.

### 6.2 Connect Hardware to Host PC

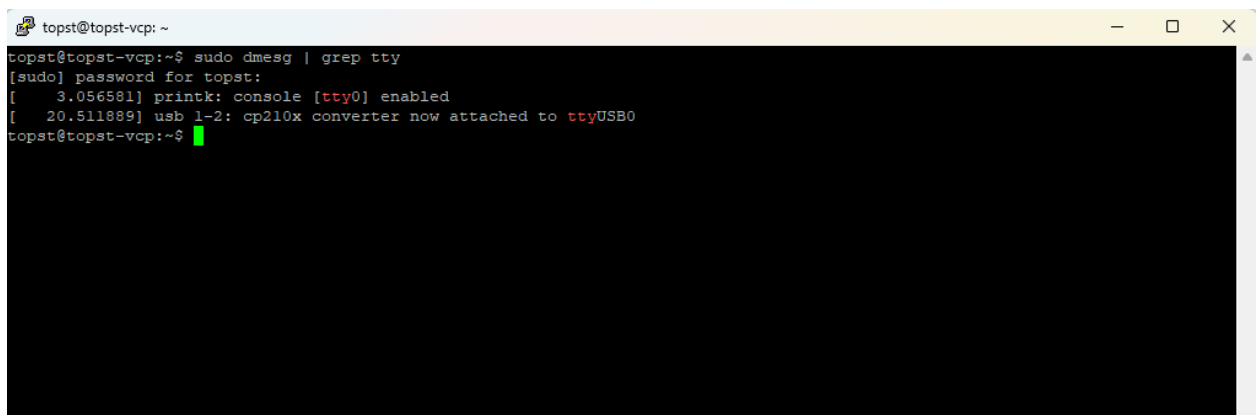
If the OS of your host PC is Ubuntu, proceed directly to step 3.

1. **Download usbipd-win:** usbipd-win project is required to use USB in WSL2.  
Download usbipd-win from <https://learn.microsoft.com/ko-kr/windows/wsl/connect-usb#attach-a-usb-device>.
2. **Run PowerShell and Attach the VCP-G (recognized as a COM port in Windows) to WSL2**  
Execute the following commands in Windows PowerShell (not Linux).

usbipd list
usbipd bind --busid <busid>
usbipd attach --wsl --busid <busid>

3. **Connect USB Type-C Cable:** Use a USB Type-C cable to connect the VCP-G board to the host PC.
4. **Verify USB Connection:** In WSL2, execute the following commands.

sudo apt-get install usbutils && lsusb
sudo dmesg   grep tty



```
topst@topst-vcp: ~  
topst@topst-vcp:~$ sudo dmesg | grep tty  
[sudo] password for topst:  
[  3.056581] printk: console [tty0] enabled  
[ 20.511889] usb 1-2: cp210x converter now attached to ttyUSB0  
topst@topst-vcp:~$
```

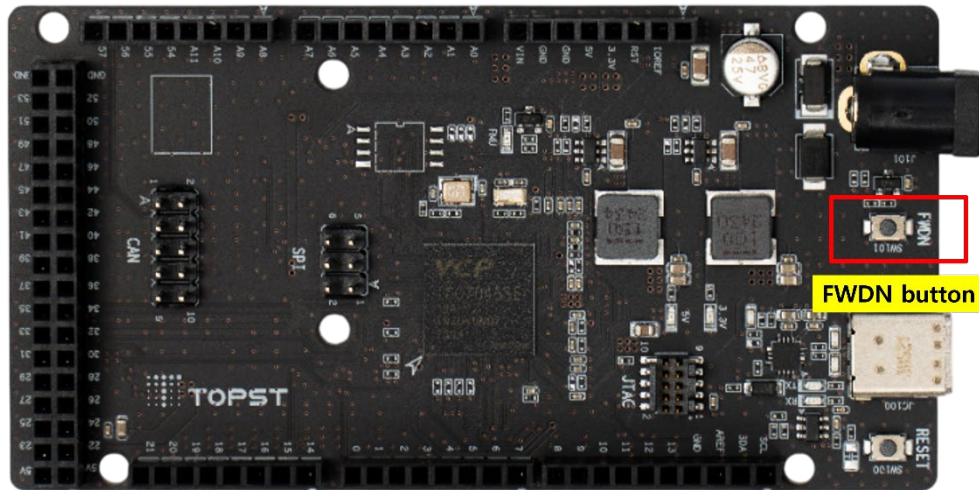
Figure 6.1 Verify USB Connection

If the output displayed in Figure 6.1 appears, the connection is successfully established.

## 6.3 Download Software on VCP-G Board

### 6.3.1 Execute FWDN in Windows Environment

1. **Set the Board to Download Mode:** Connect the power cable to the VCP-G board while pressing the FWDN button.



**Figure 6.2 Set Board to Download Mode**








2. Copy `tcc70xx_pflash_boot_2M_ECC.rom` to the `fwdn_vcp` folder.

```
cp ~/topst-vcp/build/tcc70xx/gcc/output/tcc70xx_pflash_boot_2M_ECC.rom ~/topst-vcp/tools/fwdn_vcp/
```

3. Copy `fwdn_vcp` folder to the C Drive.

```
cp -r ~/topst-vcp/tools/fwdn_vcp /mnt/c/
```

4. **Click `fwdn_vcp.bat`:** Use FWDN to download the built software to the 4 MB flash on the VCP-G.

 fwdn	2025-06-25 오후 1:48	파일	116KB
 fwdn	2025-06-25 오후 1:48	응용 프로그램	781KB
 fwdn_vcp	2025-06-25 오후 1:48	Windows 배치 파일	1KB
 tcc70xx_pflash_boot_2M_ECC.rom	2025-06-25 오후 1:48	ROM 파일	2,048KB
 vcp_fwdn.rom	2025-06-25 오후 1:48	ROM 파일	43KB
 vtcdrv v5.0.0.14	2025-06-25 오후 1:48	압축(ZIP) 파일	1,594KB
 VtcUsbPort.dll	2025-06-25 오후 1:48	응용 프로그램 확장	1,775KB

**Figure 6.3 Click `fwdn_vcp.bat`**

5. After clicking `fwdn_vcp.bat`, input the Com port num and press **[Enter]** to see the following log.

```
[main:27] FWDN VCP v0.1.1 - 2022.8.12 11:38:19
Com port num : 10
[FWDNWindowsUART::OpenPort:34] Complete open port(\\.\COM10)
[ProtocolCB::StartVCPFWDN:45] Complete to receive start res
[FWDN_VCP::LoadFwdnFW:144] Complete to send start msg
[ProtocolCB::CheckResPacket:172] Complete to receive ack for cmd(0xFFFF0000(RECEIVE_HSM_CMD))
[FWDN_VCP::WriteFile:284] Complete to send command(0xFFFF0000(RECEIVE_HSM_CMD))
[ProtocolCB::SendFile:151] Complete to send file
[ProtocolCB::CheckResPacket:172] Complete to receive ack for cmd(0xFFFF0000(RECEIVE_HSM_CMD))
```



```

[FWDN_VCP::WriteFile:295] Complete to send file - .\tcc70xx_pflash_boot_2M_ECC.rom
[FWDN_VCP::LoadFwdnFW:163] Complete to send hsm
[ProtocolCB::CheckResPacket:172] Complete to receive ack for cmd(0xFFFF0001(RECEIVE_FWDN_CMD))
[FWDN_VCP::WriteFile:284] Complete to send command(0xFFFF0001(RECEIVE_FWDN_CMD))
[ProtocolCB::SendFile:126] uiRemainSize = 43136
[ProtocolCB::SendFile:151] Complete to send file
[ProtocolCB::CheckResPacket:172] Complete to receive ack for cmd(0xFFFF0001(RECEIVE_FWDN_CMD))
[FWDN_VCP::WriteFile:295] Complete to send file - .\vcp_fwdn.rom
[FWDN_VCP::LoadFwdnFW:173] Complete to send fwdn
[FWDN_VCP::LoadFwdnFW:179] Complete to load FWDN F/W
RM=00000000
MT
MR0=0000a042
MR1=00020018
MR2=00000000
[ProtocolFW::CheckResPacket:260] Complete to receive ack for cmd(0xAAAA0016(VERSION_CMD))
[FWDN_VCP::GetDeviceVersion:77] FWDN Firmware Version(20230728)
[ProtocolFW::CheckResPacket:260] Complete to receive ack for cmd(0xAAAA0014(STORAGE_INFO_CMD))
[FWDN_VCP::InfoStorage:56]
#### SNOR Info ####
Manufacture ID: 0x9d
Device ID: 0x6015
Name: ISSI-IS25LP016D
Sector Size: 4 KiB (4096 Byte)
Total Capacity: 2 MiB (2097152 Byte)
4Byte Address Mode: Unsupported
#### EFLASH Info ####
DCYCRDCON 0x1e0002
DCYCWRCN 0x20100
Sector Size: 8 KiB
Page Size: 2 KiB

-----Storage init info-----
O : Init success
X : Init failed or not exist
SNOR : 0
eFlash : 0

[ProtocolFW::CheckResPacket:260] Complete to receive ack for cmd(0xAAAA0017(CHIP_INFO_CMD))
[FWDN_VCP::GetChipInfo:121] ---chip info---
Chip Number : 0x57045
Dual Bank : false
Expand Flash : true
ECC : true
[FWDN_VCP::PrintBankInfo:468] ---bank info---
bank - 0
eFlash offset : 0x0
eFlash size : 2097152 byte
SNOR offset : 0x0
SNOR size : 2097152 byte
[FWDN_VCP::PrintStorageOption:451] ---storage info---
eflash
offset : 0x0
size : 2097152 byte
[ProtocolFW::CheckResPacket:260] Complete to receive ack for cmd(0xAAAA0011(WRITE_CMD))
[FWDN_VCP::WriteFile:284] Complete to send command(0xAAAA0011(WRITE_CMD))
100% [||||||||||||||||||||||||||||||||] 2097152/2097152

```

6. **Reset the Board:** After the download process is completed, disconnect and reconnect the power cable.

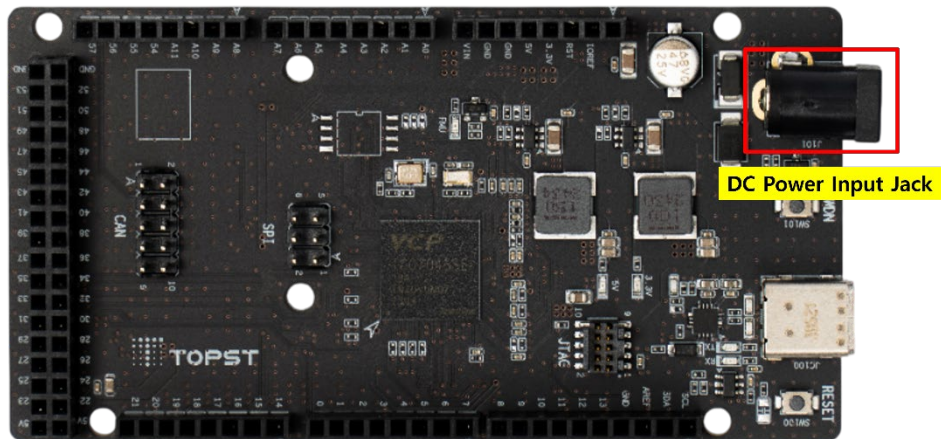


Figure 6.4 Reset Board

### 6.3.2 Execute FWDN in Linux Environment

1. **Set the Board to Download Mode:** Connect the power cable to the VCP-G board while pressing the FWDN button.

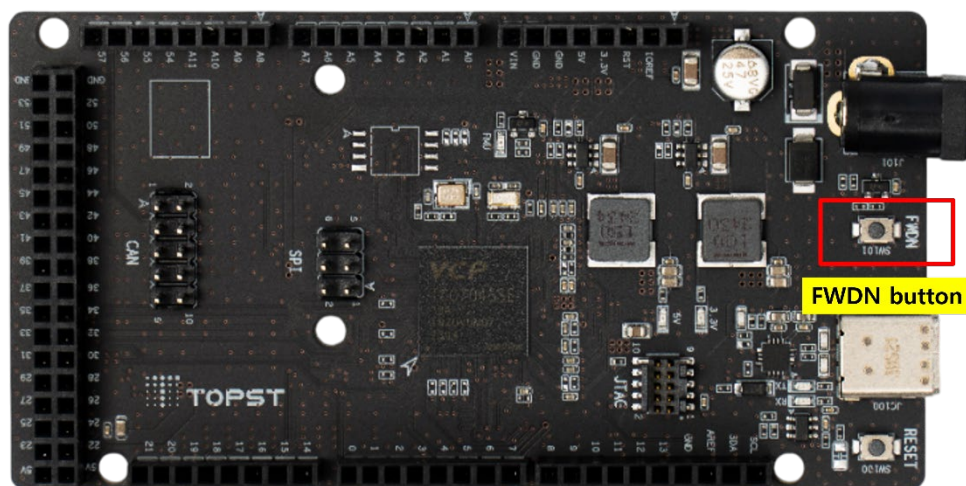


Figure 6.5 Set Board to Download Mode

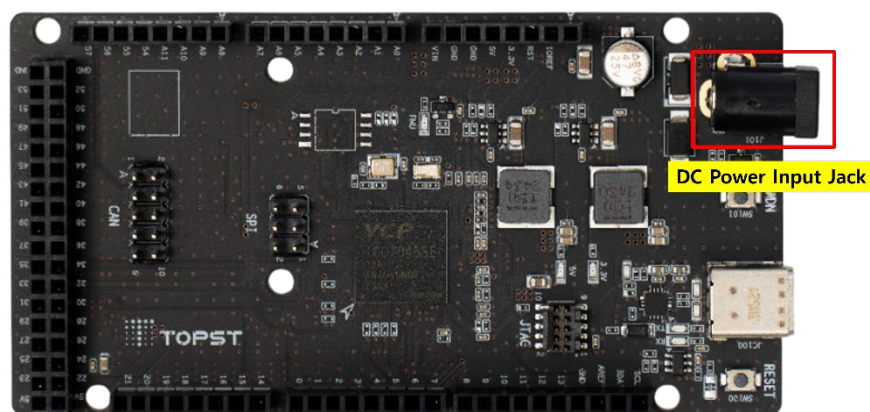
2. **Execute the Download Command:** Use *FWDN* to download the built software to the 4 MB flash on the VCP-G.

```
sudo ~/topst-vcp/tools/fwdn_vcp/fwdn --fwdn ~/topst-vcp/tools/fwdn_vcp/vcp_fwdn.rom -w
output/tcc70xx_pflash_boot_2M_ECC.rom
```

```
topst@topst-vcp: ~/topst-vcp/build/tcc70xx/gcc
topst@topst-vcp:~/topst-vcp/build/tcc70xx/gcc$ sudo ../../tools/fwdn_vcp/fwdn --fwdn ../../tools/fwdn_vcp/vcp_fwdn.rom
-w output/tcc70xx_pflash_boot 4M ECC.rom
[main:27] FWDN VCP v0.2.0 - 2022.8.12 11:59:48
[OpenPort:23] Complete open port(/dev/ttyUSB0)
Retry : 0[StartVCPFWDN:45] Complete to receive start res
[LoadFwdnFW:144] Complete to send start msg
[CheckResPacket:172] Complete to receive ack for cmd(0xFFFF0000(RECEIVE_HSM_CMD))
[WriteFile:284] Complete to send command(0xFFFF0000(RECEIVE_HSM_CMD))
84% [|||||] 110920/131072
```

**Figure 6.6 Execute Download Command**

3. **Reset the Board:** After the download process is completed, disconnect and reconnect the power cable.

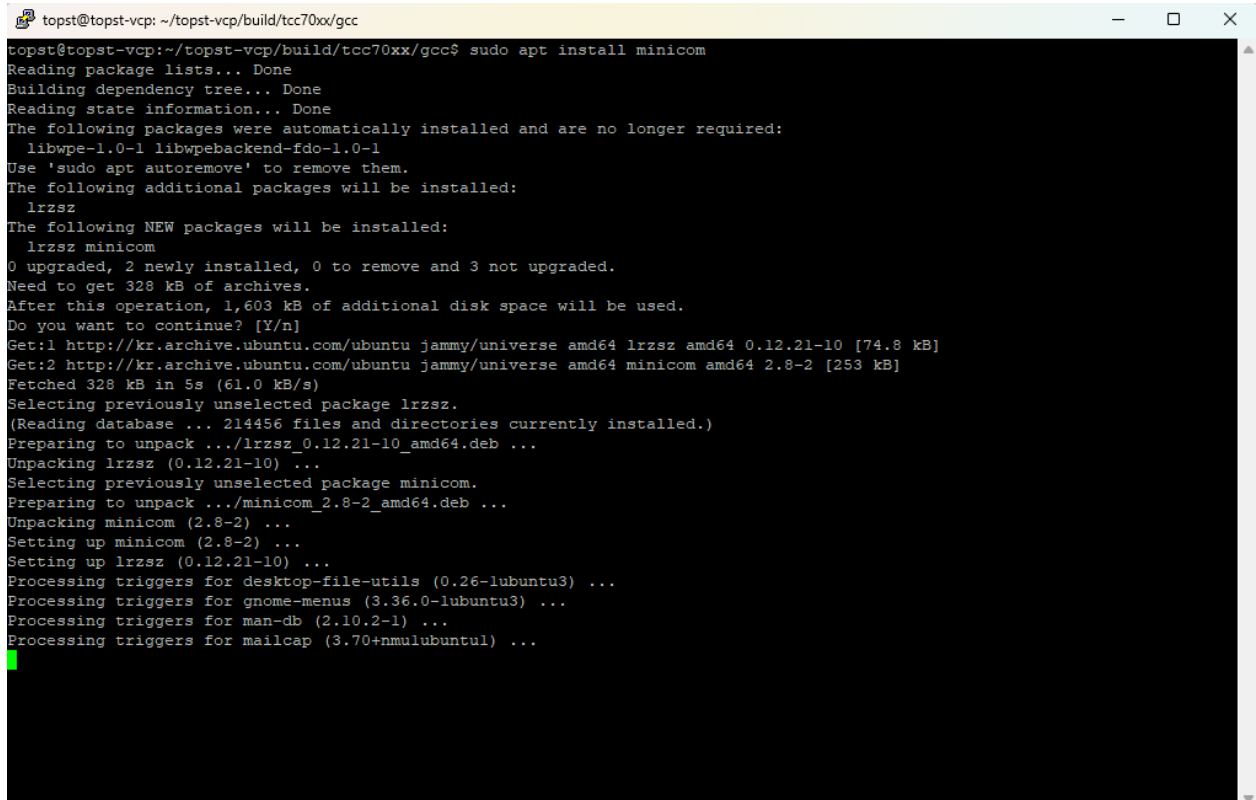
**Figure 6.7 Reset Board**

## 6.4 Verify Software on Board

After downloading the software to the board, follow these steps to verify that it is operating correctly.

1. Install minicom.

```
sudo apt install minicom
```

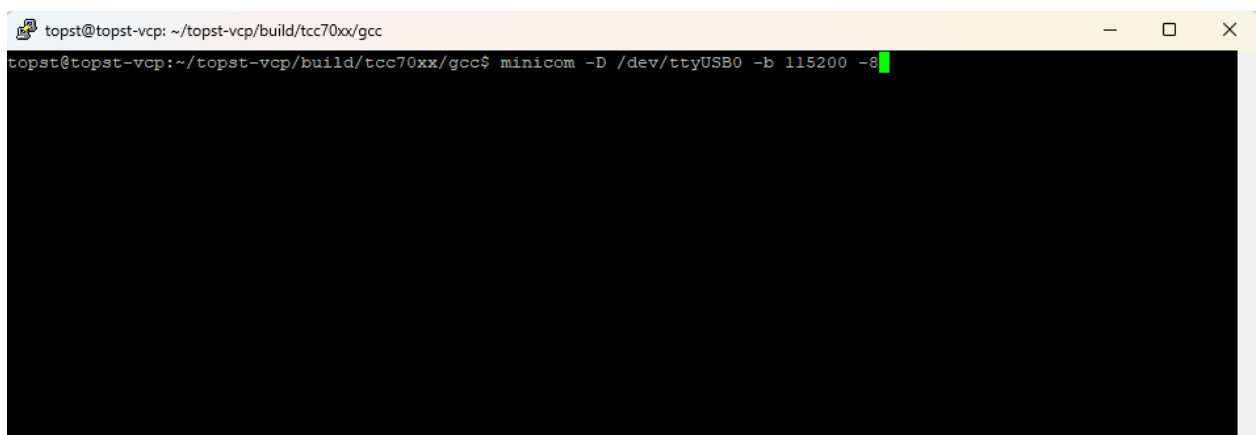
A terminal window titled 'topst@topst-vcp: ~/topst-vcp/build/tcc70xx/gcc' showing the command 'sudo apt install minicom' and its output. The output indicates that 'lrzsz' and 'minicom' will be installed, along with their dependencies. It shows the progress of downloading and unpacking these packages, and finally, setting them up. The terminal output is as follows:

```
topst@topst-vcp: ~/topst-vcp/build/tcc70xx/gcc$ sudo apt install minicom
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following packages were automatically installed and are no longer required:
  libwpe-1.0-1 libwpebackend-fdo-1.0-1
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
  lrzsz
The following NEW packages will be installed:
  lrzsz minicom
0 upgraded, 2 newly installed, 0 to remove and 3 not upgraded.
Need to get 328 kB of archives.
After this operation, 1,603 kB of additional disk space will be used.
Do you want to continue? [Y/n]
Get:1 http://kr.archive.ubuntu.com/ubuntu jammy/universe amd64 lrzsz amd64 0.12.21-10 [74.8 kB]
Get:2 http://kr.archive.ubuntu.com/ubuntu jammy/universe amd64 minicom amd64 2.8-2 [253 kB]
Fetched 328 kB in 5s (61.0 kB/s)
Selecting previously unselected package lrzsz.
(Reading database ... 214456 files and directories currently installed.)
Preparing to unpack .../lrzsz_0.12.21-10_amd64.deb ...
Unpacking lrzsz (0.12.21-10) ...
Selecting previously unselected package minicom.
Preparing to unpack .../minicom_2.8-2_amd64.deb ...
Unpacking minicom (2.8-2) ...
Setting up minicom (2.8-2) ...
Setting up lrzsz (0.12.21-10) ...
Processing triggers for desktop-file-utils (0.26-1ubuntu3) ...
Processing triggers for gnome-menus (3.36.0-1ubuntu3) ...
Processing triggers for man-db (2.10.2-1) ...
Processing triggers for mailcap (3.70+nmulubuntu1) ...
```

Figure 6.8 Install Minicom

2. **Open Serial Connection:** Use the following command to initiate a serial connection.

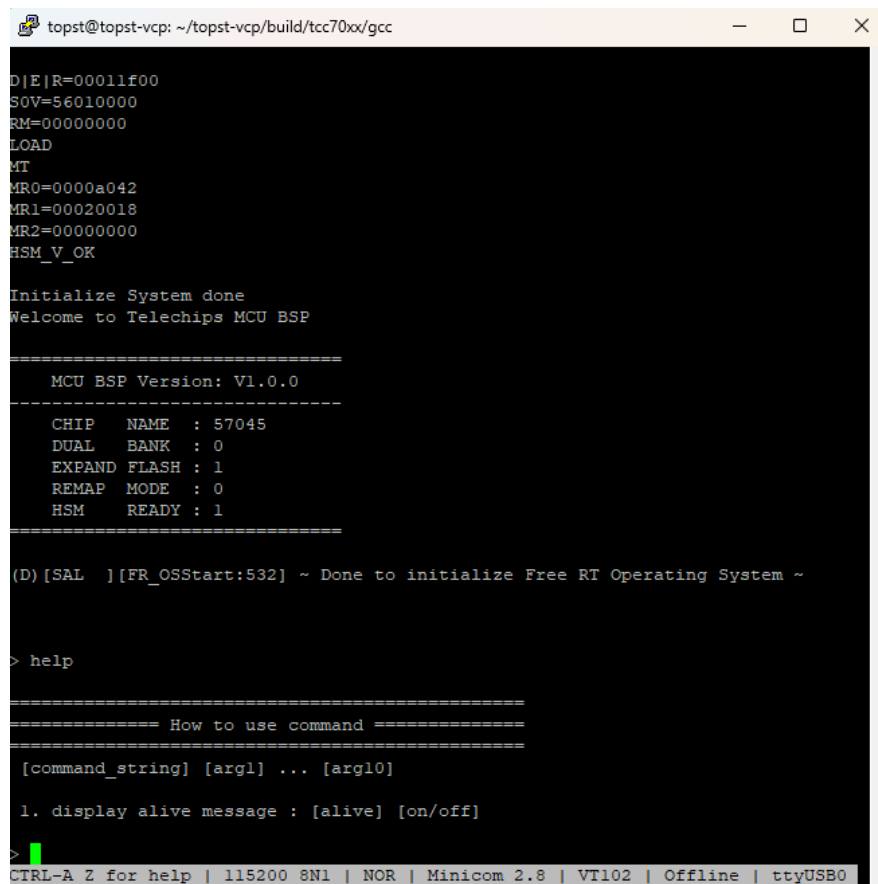
```
minicom -D /dev/ttyUSB0 -b 115200 -8
```

A terminal window titled 'topst@topst-vcp: ~/topst-vcp/build/tcc70xx/gcc' showing the command 'minicom -D /dev/ttyUSB0 -b 115200 -8' being executed. The command is entered at the prompt, and a green cursor is visible at the end of the line. The terminal output is as follows:

```
topst@topst-vcp: ~/topst-vcp/build/tcc70xx/gcc$ minicom -D /dev/ttyUSB0 -b 115200 -8
```

Figure 6.9 Open Serial Connection

After completing steps 1 and 2, the following output appears on the terminal. This indicates that the SDK has been successfully loaded and is running properly on the VCP-G board.



```
topst@topst-vcp: ~/topst-vcp/build/tcc70xx/gcc

D|E|R=00011f00
SOV=56010000
RM=00000000
LOAD
MT
MR0=0000a042
MR1=00020018
MR2=00000000
HSM_V_OK

Initialize System done
Welcome to Telechips MCU BSP

=====
MCU BSP Version: V1.0.0
=====

CHIP  NAME  : 57045
DUAL  BANK  : 0
EXPAND FLASH : 1
REMAP  MODE  : 0
HSM    READY : 1
=====

(D)[SAL ][FR_OSStart:532] ~ Done to initialize Free RT Operating System ~

> help

===== How to use command =====
[command_string] [arg1] ... [arg10]

1. display alive message : [alive] [on/off]

>
```

CTRL-A Z for help | 115200 8N1 | NOR | Minicom 2.8 | VT102 | Offline | ttyUSB0

**Figure 6.10 Open Serial Connection**

## 6.5 Troubleshooting Common Issues

This chapter provides solutions to common issues encountered while working with the VCP-G.

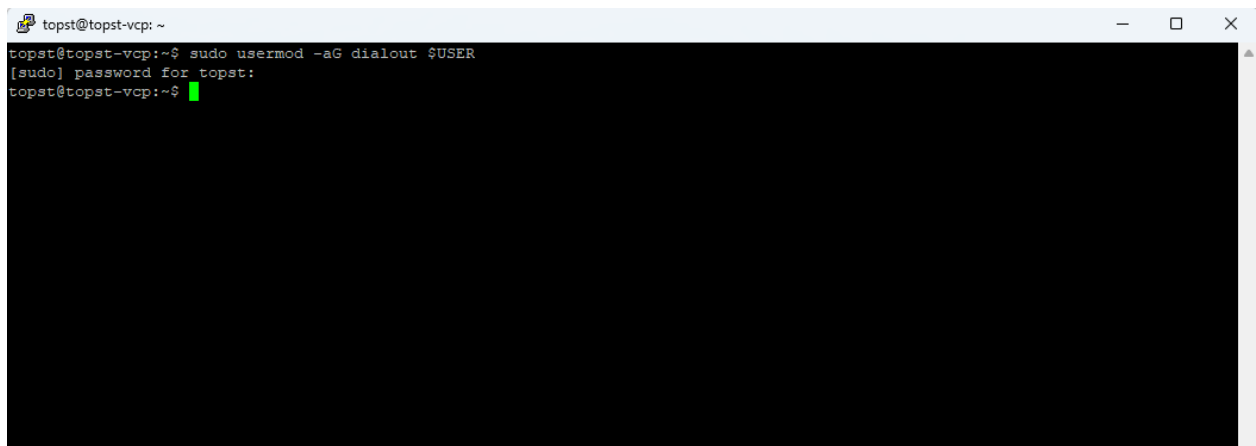
**Issue** : The *FWDN* reports a lack of permission to access the ttyUSB0 device.

**Solution** : This issue occurs when your user account (**\$USER**) does not have the necessary permissions to access serial devices. To resolve this, add the user account to the dialout group.

1. **Modify User Group Permissions:** Execute the following command.

```
sudo usermod -aG dialout $USER
```

2. **Log Out and Log Back In:** Log out of the current session and log back in to apply the changes. After this, try accessing the ttyUSB0 device again.



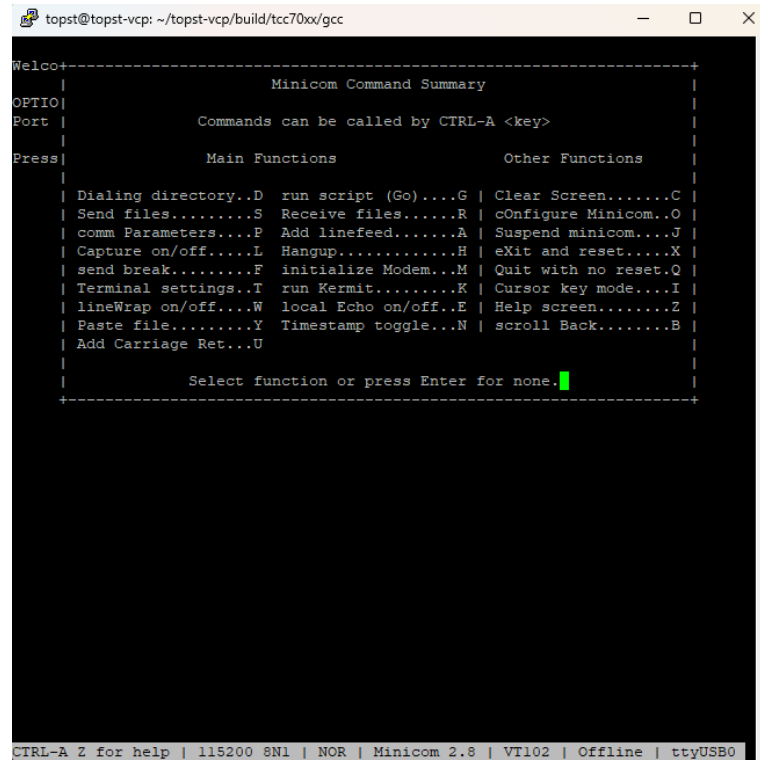
**Figure 6.11 Modify User Group Permissions**

**Issue** : When using minicom, there is no proper communication or communication with the VCP-G is unstable.

**Solution :** This issue may occur if minicom’s default flow control setting is set to **hardware**. The hardware flow control must be set to **No** for proper operation.

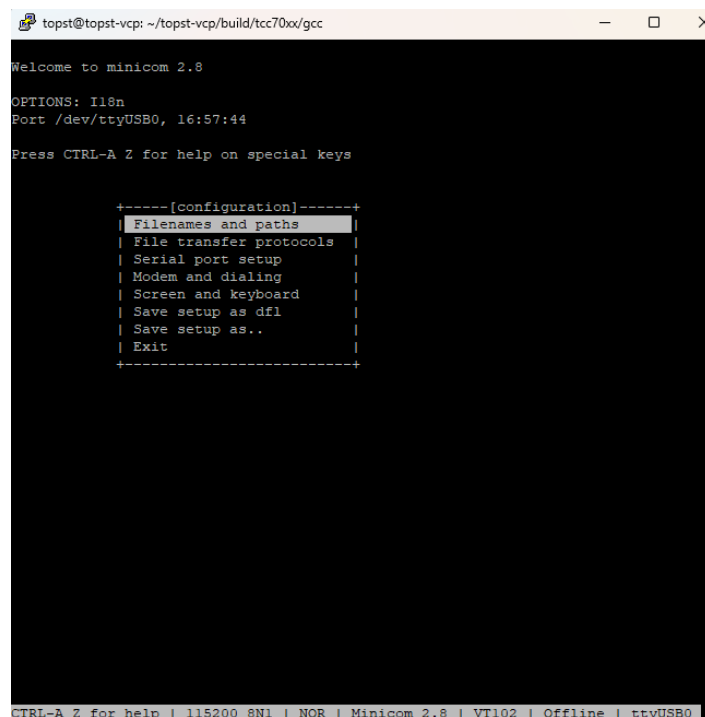
1. **Start minicom:** Use the following command.

```
minicom -D /dev/ttyUSB0 -b 115200 -8
```



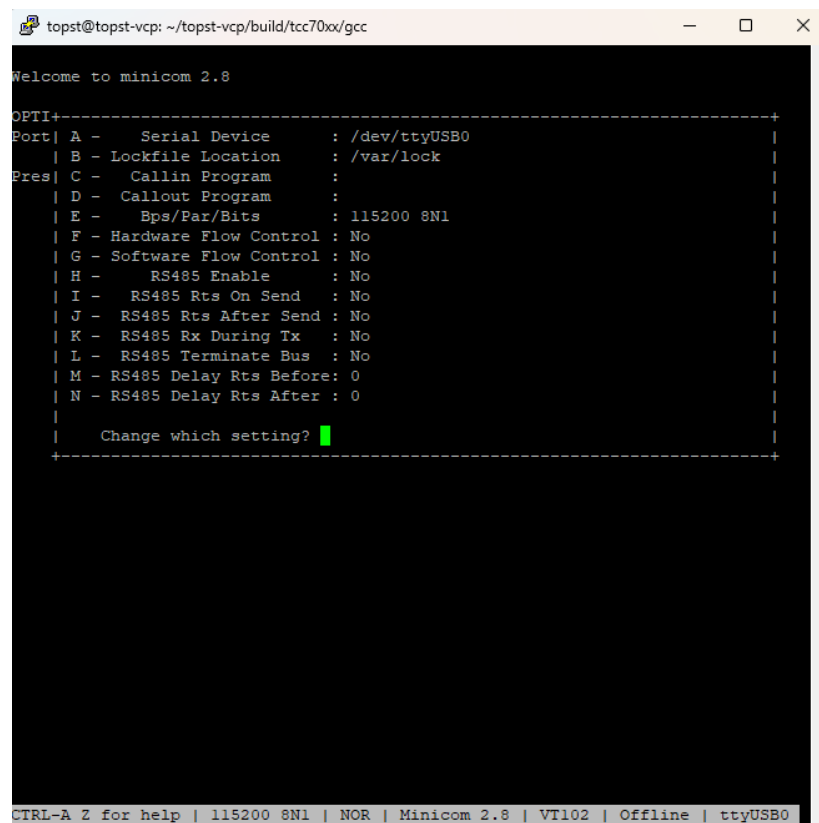
### Figure 6.12 Launch minicom

2. **Access the Setup Screen:** While in minicom, press **[Ctrl+A]** then press **[o]** to open the setup menu.



**Figure 6.13 Access Set up Screen**

3. **Navigate to Serial Port Setup:** Select **Serial port setup** from the options.
4. **Modify Flow Control:** Inside the serial port setup, press **[F]** to set the hardware flow control to **No**.



```

topst@topst-vcp: ~/topst-vcp/build/tcc70xx/gcc
Welcome to minicom 2.8

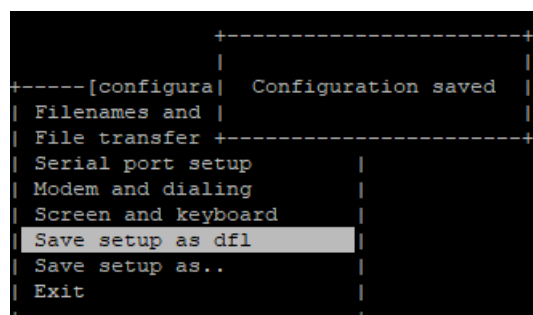
OFTI+-----+
Port| A -   Serial Device       : /dev/ttyUSB0   |
    | B - Lockfile Location    : /var/lock      |
Pres| C -   Callin Program      :               |
    | D - Callout Program      :               |
    | E -   Bps/Par/Bits       : 115200 8N1     |
    | F - Hardware Flow Control : No            |
    | G - Software Flow Control: No            |
    | H -   RS485 Enable       : No            |
    | I -   RS485 Rts On Send  : No            |
    | J -   RS485 Rts After Send: No           |
    | K -   RS485 Rx During Tx : No            |
    | L -   RS485 Terminate Bus: No           |
    | M - RS485 Delay Rts Before: 0            |
    | N - RS485 Delay Rts After: 0            |
    |                                     |
    |   Change which setting? [F]             |
+-----+

CTRL-A Z for help | 115200 8N1 | NOR | Minicom 2.8 | VT102 | Offline | ttyUSB0

```

Figure 6.14 Modify Flow Control

5. **Exit and Save:** Exit the setup and save the configuration. minicom should now communicate properly with the VCP-G board.



```

+-----+
+-----[configura| Configuration saved |
| Filenames and |
| File transfer +-----+
| Serial port setup
| Modem and dialing
| Screen and keyboard
| Save setup as dfl
| Save setup as..
| Exit
+-----+

```

Figure 6.15 Save and Exit

**Note:** If you are using a different serial communication tool other than minicom, ensure its flow control setting is also set to **No** for proper operation.



## 7 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.

## 8 REVISION HISTORY

### Rev. 1.01: 2025-07-18

- Updated
  - Chapter 1: Description
  - Chapter 2: Description
  - Chapter 2.2: Changed chapter title from "Setting Linux Environment" to "Linux Environment"
  - Chapter 4.1: Description
  - Chapter 6.1: Changed chapter title from "Prepare TOPST VCP-G" to "Prepare VCP-G Board"
  - Chapter 6.2: Description
  - Chapter 6.3: Changed chapter title from "Download Software on TOPST VCP-G Board" to "Download Software on VCP-G Board"
  - Chapter 6.3.1: Description
  - Chapter 6.3.2:
    - Description
    - Figure 6.5
    - Figure 6.7
- Added
  - Chapter 2.1: Windows Environment
  - Chapter 5.1: 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."