Qualcomm Robotics SDK Manager

User Guide

Rev. E
Jan 12, 2023

Revision History

Revision	Date	Description
A	July 30, 2020	Initial release.
B	Oct 27, 2020	• Add information on Windows 10 Professional and Windows 10 Enterprise operation. • Update TROUBLESHOOTING.
C	Dec 14, 2020	Optimize file construction.
D	July 29, 2022	Add contents to make this user guide compatible with RB6.
E	Jan 12, 2023	• Add disk requirements in Chapter 2. • Update Step 5 in Chapter 5. • Update Step 4 in Chapter 6. • Update Step 9 - 4) in Chapter 7.

Table List

About This Document

- Illustrations in this documentation might look different from your product.
- Depending on the model, some optional accessories, features, and software programs might not be available on your device.
- Depending on the version of operating systems and programs, some user interface instructions might not be applicable to your device.
- Documentation content is subject to change without notice. Thundercomm makes constant improvements on the documentation of the products, including this guidebook.
- Function declarations, function names, type declarations, attributes, and code samples appear in a different format, for example, `cp armcc armcpp`.
- Code variables appear in angle brackets, for example, `<number>`.
- Button, tool, and key names appear in bold font, for example, click **Save** or press **Enter**.
- Commands to be entered appear in a different font; on the host computer use $ as shell prompt, while on the target device use # as shell prompt, for example,

```
$ adb devices
# logcat
```

- Part of the code that does not contain instructions appear in a different format, for example,

```
SUBSYSTEM=="usb", ATTR{idVendor}=="18d1", MODE="0777", GROUP="adm"
```

- Folders, path and files are formatted in italic, for example, *turbox_flash_flat.sh*.

Table of Contents

Chapter 1. Overview

SDK Manager provides a complete set of tools for generating and flashing the RBx firmware, supporting systems of Ubuntu 16.04, Ubuntu 18.04, Ubuntu 20.04, Windows 10 Professional (64-bit) and Windows 10 Enterprise (64-bit).

V3.0.0 Support Products List:

- Robotics RB5 Platform
- Robotics RB6 Non-pop Platform

Chapter 2. Requirements

- Required Operating System (OS): Ubuntu 18.04.
- Alternative: Run an Ubuntu 18.04 Docker on a host of Ubuntu 16.04, Ubuntu 18.04, Ubuntu 20.04, Windows 10 Professional (64-bit), or Windows 10 Enterprise (64-bit) system.
- Disk requirements:
  - It requires at least 0.8GB disk space to download a software version.
  - It requires at least 20GB disk space to download LU resources and generate *system.img* with current release.

Chapter 3. Read This First

- To register a Thundercomm Account, go to http://www.thundercomm.com.
- Keep the internet connected during the image generation.
- The full process lasts for at least 40 minutes, depending on Internet speed.
- A working directory is needed to be built with the write and read permission in SDK Manager. For Docker container user, create your target directory under */home/hostPC/*.
- Docker Desktop is only supported on Windows 10 Professional (64-bit) and Windows 10 Enterprise (64-bit) OS.
- Before flashing full build, generate image first.
- USB 3.0 port and USB 3.0 cable are recommended for flashing images.
- If a Linux host is used to flash the device, run the following command before connecting the device to the host.

```
$ sudo systemctl stop ModemManager
```

- Plug in a USB device before starting Option 2 (**EDL** programming sequence), if an Ubuntu 18.04 host is running the SDK Manager by the Ubuntu 18.04 Docker.

Chapter 4. SDK Manager Download and Unzip

You can download SDK manager from the following link:

https://thundercomm.s3.ap-northeast-1.amazonaws.com/uploads/web/common/TC-sdkmanager-3.0.0.zip

To unzip the SDK manager file, you can run the following command:

```
$ unzip TC-sdkmanager-x.x.x.zip
```

Chapter 5. On Ubuntu 18.04 Host

Prerequisite:

Required minimum package version: coreutils 8.28, fakechroot 2.19, fakeroot 1.22, kmod 24-1ubuntu3.2, libc6-arm64-cross 2.27, python 2.7.15, qemu-user-static 1:2.11+dfsg- 1ubuntu7.28, udev 237-3ubuntu10.42, unzip 6.0, wget 1.19.4.

Step 1. Install the dependency libraries to the host computer:

```
$ sudo apt-get install coreutils fakechroot fakeroot \
  kmod libc6-arm64-cross  python2.7 qemu-user-static wget udev openssh-server
```

Step 2. Unzip *TC-sdkmanager-x.x.x.zip* and navigate to *TC-sdkmanager-x.x.x* directory from a terminal window, and install or re-install SDK Manager:

```
$ sudo dpkg -i tc-sdkmanager-vx.x.x_amd64.deb
```

Step 3. Launch SDK Manager.

```
$ sdkmanager
```

Step 4. Run SDK Manager.

1) Provide Thudercomm login credentials:

```
Credential Checking ...
Enter your Thundercomm username:
Enter your Thundercomm password:
```

2) In case you want to change the path of installation, provide a working directory when a target directory is required (for example: */home/user*). Then, enter the absolute target directory for SDK Manager to overwrite existing files (Default directory: */home/user*).

```
Enter absolute target directory for Image resources (overwrites existing files,
default: /home/user/):
```

➲ **NOTE:** Docker users shall provide a working directory as */home/hostPC/[workingdirectory]*.

3) Enter the number selection product, for example, `1`.

```
Choose a product
Enter 1 to use RB5, 2 to use RB6:
```

➲ **NOTE:** If the product only supports single platform, the SDK Manager will automatically skip Step 4 - 4) and directly go to Step 4 - 5).

4) Enter the number of available platform for Robotics RBx device, for example, 1

```
Choose a platform for Robotics RBx device
Enter 1 to use LU platform, 2 to use LE platform:
```

5) Enter the number of available version for image repack, for example, `1`:

```
Checking current versions of release ...
Available versions:
1: QRB5165.x.x.x-xxxxxx
… …
Select one number of available version ( 1 | 2 | 3 ...) to continue with:
```

6) Enter `1` when the message below appears on your screen:

```
---------------------------------------------------------------------------------
SDK has been successfully set up and is ready to be used
Type 'help' for commands
---------------------------------------------------------------------------------
>
```

➲ **NOTE:**

- This step lasts for at least 40 minutes.
- Enter `help` for more information:

```
> help
commands:
help = Show usage help for LU platform
1 = Download LU resources and generate system.img with current release
2 = Flash full build (require system.img generation first)
q = exit sdk manager
```

7) The system images are successfully generated in the working directory with the following messages displayed.

```
--------------------------------------------------------------------------------
Move sparse images to full_build ...done
You may proceed to flash full_build to your device
--------------------------------------------------------------------------------
```

➲ **NOTE:** For Docker users, the system image is generated in */home/hostPC/[workingdirectory]*.

Step 5.Disconnect the device from the computer, then follow the operation steps below to flash full build:

1) Power off the device (unplug power cable and USB cable).

2) Press the F_DL key.

3) Power on the device (Required voltage: 12 V).

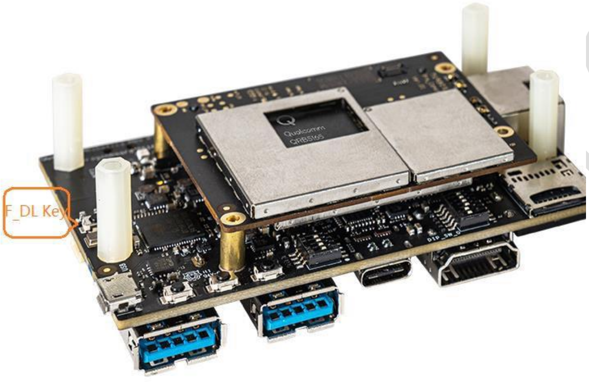


Figure 5-1.RB5 F_DL Key

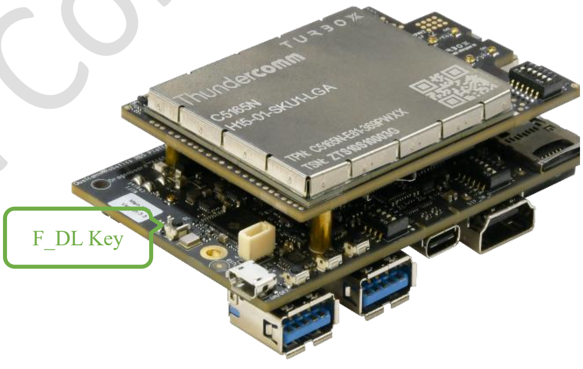


Figure 5-2.RB6 F_DL Key

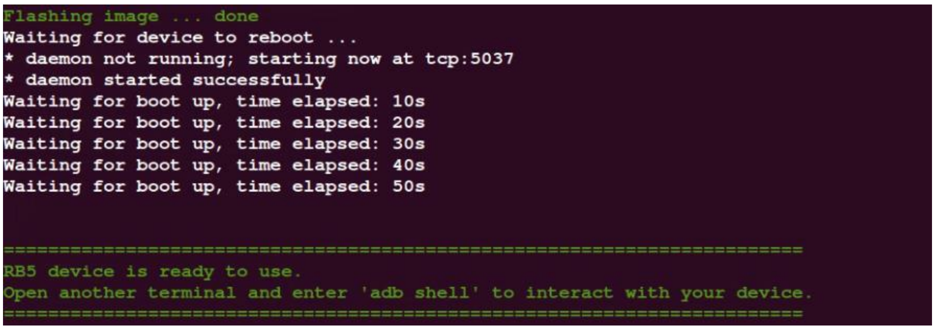
4) Keep pressing **F_DL key** while connecting the board to your computer with a Type-C USB (This step will switch the device to **EDL** mode).

5) Release **F_DL key** after the board has been connected to your computer.

6) Start flashing process from the SDK manager (command 2).

7) SDK Manager shall detect the device and start the flashing process automatically.

8) After the flashing process been finished, the board will reboot automatically (this step may take some time).



```
Flashing image ... done
Waiting for device to reboot ...
* daemon not running; starting now at tcp:5037
* daemon started successfully
Waiting for boot up, time elapsed: 10s
Waiting for boot up, time elapsed: 20s
Waiting for boot up, time elapsed: 30s
Waiting for boot up, time elapsed: 40s
Waiting for boot up, time elapsed: 50s

======================================================================
RB5 device is ready to use.
Open another terminal and enter 'adb shell' to interact with your device.
======================================================================
```

Figure 5-3.Flash Process

9) When your device has successfully boot up, enter the command below in a new terminal window of the host computer:

```
$ adb wait-for-device shell
```

Chapter 6. On Other Ubuntu Hosts (Ubuntu 16.04, 20.04)

Step 1. Install qemu-user-static, openssh-server and udev to the host computer.

```
$ sudo apt-get install qemu-user-static openssh-server udev -y
```

Step 2. To install Docker, refer to: https://docs.docker.com/engine/install/ubuntu/.

Step 3. Generate Ubuntu 18.04 docker image:

Unzip *TC-sdkmanager-x.x.x.zip* and navigate to *TC-sdkmanager-x.x.x* directory from a new terminal window, then execute the following commands:

```
# Ubuntu terminal #
$ sudo docker build -t ubuntu:18.04-sdkmanager .
------------------------------------------------------------------------------
```

➲ **NOTE:**

- Make sure to include the space and full stop at the end of the command: . .
- Generated Docker image name: ubuntu:18.04-sdkmanager.

Step 4. Create Docker container:

```
$ sudo docker run -v /home/${USER}:/home/hostPC/ --privileged -v /dev/:/dev -v
/run/udev:/run/udev -d --name sdkmanager_container -p 36000:22
ubuntu:18.04-sdkmanager
------------------------------------------------------------------------------
Host PC's /home/${USER} is mounted on /home/hostPC in Docker container
sdkmanager_container: container name
```

➲ **NOTE**: With the above command, a Docker container name can be generated after `sdkmanager_container:`.

Step 5. Launch SDK Manager in Docker container.

```
$ sudo docker exec -it sdkmanager_container sdkmanager
```

Step 6. Run SDK Manager. Refer to Step 4 of Chapter 5. On Ubuntu 18.04 Host.

Step 7. Disconnect the device from the computer, then follow the operation steps below to flash full build:

Refer to Step 5 of Chapter 5. On Ubuntu 18.04 Host.

Step 8. When your device has successfully boot up, enter the command below in a new terminal window of the host computer.

```
$ adb wait-for-device shell
```

Chapter 7. On Windows 10 (64-bit) Host

Step 1. To download Docker Desktop, go to:

https://hub.docker.com/editions/community/docker-ce-desktop-windows/

Step 2. Open **Dashboard** from Docker notification menu to launch Docker Desktop.

Step 3. Open Windows PowerShell and enter **docker images** to verify docker installation. If PowerShell console instructs **error**, either the installation or Docker Desktop operation fails.

Step 4. Generate Ubuntu 18.04 docker image.

1) Unzip *TC-sdkmanager-x.x.x.zip* and navigate to *TC/sdkmanager/x.x.x* directory from a Windows PowerShell.

2) Execute the following commands:

```
# Windows PowerShell #
$ docker build -t ubuntu:18.04-sdkmanager .
------------------------------------------------------------------------------
```

➲ **NOTE**:

- Make sure to include the 'space' and 'period' at the end of the command.
- Generated docker image name: `ubuntu:18.04-sdkmanager`.

Step 5. Create docker container.

```
$ docker run -it -d --name sdkmanager_container ubuntu:18.04-sdkmanager
```

➲ **NOTE:** A docker container name can be generated after `sdkmanager_container` with the above command.

Step 6. Launch SDK Manager.

```
$ docker exec -it sdkmanager_container sdkmanager
```

Step 7. Run SDK Manager. Refer to Step 4 of Chapter 5. On Ubuntu 18.04 Host.

Step 8. Copy the full build from Docker container to a Windows Host computer.

```
$ docker cp  sdkmanager_container:[target_directory]/[name_of_selected_re-
lease]/full_build [destionation path on Windows host PC]
------------------------------------------------------------------------------
Example: docker cp sdkmanager_container:/home/hostPC/demo_0803/QRB5165.x.x.x-
xxxxxx/full_build D:\
------------------------------------------------------------------------------
```

Step 9. Flash the device.

1) For more information about the MULTIDL_TOOL operation, go to:

https://www.thundercomm.com/product/qualcomm-robotics-rb6-development-kit/#documents

2) Download *MULTIDL_TOOL_v1.0.14.zip* file and install **MULTIDL_TOOL**.

Refer to MULTIDL_TOOL_USER_GUIDE.pdf.

3) Follow the steps below to check if your device is in Emergency Download (EDL) mode:

Option 1: Enter adb reboot edl.

Option 2: Press **F_DL key** to power on.

➲ **NOTE:**

- Check the Device Manager to ensure that the device has been detected as Qualcomm HS-USB QLoader 9008 (COMx), or you might need to download and install the correct USB drivers.
- To download and install the correct USB drivers, go to:

  https://docs.thundercomm.com/turbox_doc/documents/products/qualcomm-robotics-rb6-development-kit

4) Flash the full build with **MULTIDL_TOOL**:

a) Launch **MULTIDL_TOOL**.

b) Click **Options** -- > **Configuration** to set the full-build path.

c) Enter `123456` at the password prompt.

d) Choose **Flat Build**, set Flash Type as ufs.

e) Select your local full-build path and programmer file (*prog_firehose_ddr.elf*).

f) Click **Load XML** to load xml files. When xml files prompt, select all XML files and all Patch files in the *full_build\ufs* folder. Keep other settings as default; then click **OK** to return to the home page.

g) Disconnect the device from the computer and power it off.

h) Press the F_DL key.

i) Power on the device (required voltage: 12 V).

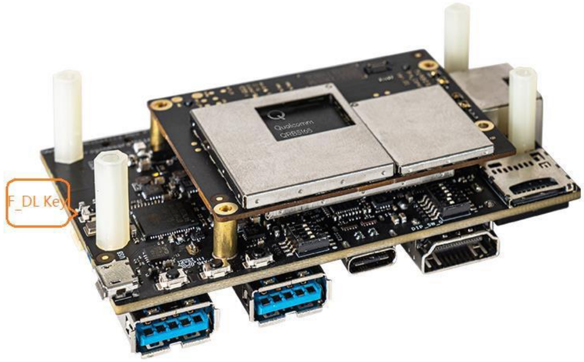


Figure 7-1.RB5 F_DL Key

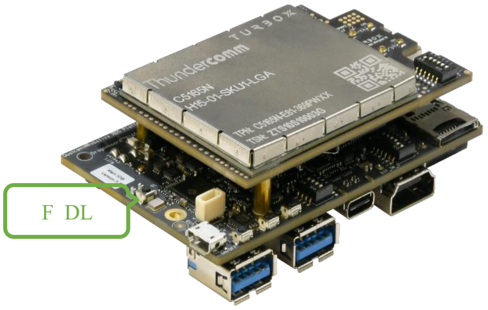


Figure 7-2.RB6 F_DL Key

j) Keep pressing **F_DL key** while connecting the board to your computer with a Type-C USB.

➲ **NOTE:** This step will switch the device to EDL mode.

k) Release **F_DL key** after the board has been connected to your computer.

l) Start the flashing process by clicking **START** button corresponding to your device port in **MULTIDL_TOOL** window.

m) After the flashing process is finished, the board will reboot automatically. This step may take some time.

Chapter 8. Troubleshoot

Refer to Table 8-1 for the solutions to problems that have definite symptoms.

Table 8-1. Troubleshooting information

Problem	Solution
Internet Timeout Issue: Internet timeout issue may happen during the image generation process, such as **Unable to fetch**.	Try to run **Command 1** again.
APT Source Issue	If the download fails, check the internet connection and the source list.
Device Boot Up Issue: SDK Manager cannot detect the device after reboot.	• If Ubuntu 18.04 is used on Docker, check whether **adb kill-server** is entered on the host PC before flashing image. • Reboot your device manually, open a terminal, then enter **adb shell**. • Check if any debian packages are modified.
Process Issue: The flashing process of Ubuntu system does not function well.	Copy the full folder to a computer with Windows system, then flash the image using **Thundercomm MULTIDL_TOOL**. For further information, refer to: MULTIDL_TOOL_USER_GUIDE_v2.pdf.
SDK Manager Flash Issue	Enter the following command on host machine before restarting the flash: `$ sudo systemctl stop ModemManager`

Chapter 9. Reference

Table 9-1. Troubleshooting information

Robotics RB5 Platform	
Quick Start Guide	https://developer.qualcomm.com/qualcomm-robotics-rb5-kit/quick-start-guide
Hardware Reference Guide	https://developer.qualcomm.com/qualcomm-robotics-rb5-kit/hardware-reference-guide
Software Reference Manual	https://developer.qualcomm.com/qualcomm-robotics-rb5-kit/software-reference-manual
Robotics RB6 Non-Pop Platform	
Quick Start Guide	https://developer.qualcomm.com/qualcomm-robotics-rb6-kit/quick-start-guide
Hardware Reference Guide	https://developer.qualcomm.com/qualcomm-robotics-rb6-kit/hardware-reference-guide
Software Reference Manual	https://developer.qualcomm.com/qualcomm-robotics-rb6-kit/software-reference-manual

Appendix 1. Notices

Thundercomm may have patents or pending patent programs covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries to service@thundercomm.com.

THUNDERCOMM PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some jurisdictions do not allow disclaimer of express or implied warranties in certain transactions; therefore, this statement may not apply to you.

Changes are made periodically to the information herein; these changes will be incorporated in new editions of the publication. To provide better service, Thundercomm reserves the right to improve and/or modify the products and software programs described in the manuals, and the content of the manual, at any time without additional notice.

The software interface and function and hardware configuration described in the manuals included with your development board or system on module might not match exactly the actual configuration of that you have purchased. For the configuration of the product, refer to the related contract (if any) or product packing list, or consult the distributor for the product sales. Thundercomm may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

The products described in this document are not intended for use in implantation or other life support applications where malfunction may result in injury or death to persons. The information contained in this document does not affect or change Thundercomm product specifications or warranties. Nothing in this document shall operate as an express or implied license or indemnity under the intellectual property rights of Thundercomm or third parties. All information contained in this document was obtained in specific environments and is presented as an illustration. The result obtained in other operating environments may vary.

The information of this document should not be as any invitation for offer or any advice to the visitors. Please consult the professional comments from the sales consultant prior to do any actions of investment or purchase.

Thundercomm may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Any references in this publication to non-Thundercomm Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this Thundercomm product, and use of those Web sites is at your own risk. Thundercomm shall not be responsible for the content of the third party.

Any performance data contained herein was determined in a controlled environment. Therefore, the result obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

This document is copyrighted by Thundercomm and the property right of the date mentioned in this document, including but not limited trademarks, patents, copyrights, trade name etc. are not covered by any open-source license. Thundercomm may update this document at any time without notice.

Anyone doesn’t have the right to amend, reprint, republication, reproduce, transmit, distribute or any other way to use this document in business or public purpose without the prior written consent by Thundercomm.

E-mail messages sent to Thundercomm via the Internet are not guaranteed to be completely secure.Thundercomm shall not be liable for any loss incurred by the surfer when transmitting any information over the Internet or for any loss incurred by Thundercomm when sending any information over the Internet at your request.

Thundercomm has all rights under other relevant exemptions provided by laws and regulations, and Thundercomm's failure to claim or delay in claiming such rights shall not be deemed to be a waiver of such rights by Thundercomm.

Thundercomm reserves the right of final interpretation of this document.

Appendix 2. Trademarks

Thundercomm, Thundercomm TurboX, TURBOX, Thundersoft turbox are trademarks of Thundercomm Corporation or its associate companies in China and/or other countries. Intel, Intel SpeedStep, Optane, and Thunderbolt are trademarks of Intel Corporation or its subsidiaries in the U.S. and/or other countries. Microsoft, Windows, Direct3D, BitLocker, and Cortana are trademarks of the Microsoft group of companies. Mini DisplayPort (mDP), DisplayPort, and VESA are trademarks of the Video Electronics Standards Association. The terms HDMI and HDMI High-Definition Multimedia Interface are trademarks or registered trademarks of HDMI Licensing LLC in the United States and other countries. Wi-Fi, Wi-Fi Alliance, WiGig, and Miracast are registered trademarks of Wi-Fi Alliance. USB-C is a registered trademark of USB Implementers Forum. All other trademarks are the property of their respective owners.