

Qualcomm Robotics RB5 SDK Manager

User Guide

Rev. C

December 14, 2020

Revision history

Revision	Date	Description
A	July 30, 2020	Initial release
B	October 27, 2020	<ul style="list-style-type: none">• Add information on Windows 10 Professional and Windows 10 Enterprise operation• Update TROUBLESHOOTING
C	December 14, 2020	Optimize file construction

Thundercomm Confidential

Contents

1 OVERVIEW	4
2 PREREQUISITE	5
3 READ THIS FIRST	6
4 DOWNLOAD AND UNZIP SDK MANAGER	7
5 ON UBUNTU 18.04 HOST	8
6 ON OTHER UBUNTU HOST (UBUNTU 16.04, 20.04)	11
7 ON WINDOWS 10 (64-BIT) HOST	14
8 TROUBLESHOOTING.....	18

Thundercomm Confidential

1 Overview

SDK Manager provides a complete set of tools for generating and flashing the RB5 firmware, supporting systems of Ubuntu 16.04, Ubuntu 18.04, Ubuntu 20.04, Windows 10 Professional (64-bit) and Windows 10 Enterprise (64-bit).

Thundercomm Confidential

2 Prerequisite

- 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.

Thundercomm Confidential

3 Read This First

1. To register a Thundercomm Account, go to <http://www.thundercomm.com>.
2. Keep the internet connected during the image generation.
3. The full process lasts for at least 40 minutes, depending on Internet speed.
4. A working directory is needed to build with the write and read permission in SDK Manager. For Docker container user, create your target directory under */home/hostPC/*.
5. Docker Desktop is only supported in Windows 10 Professional (64-bit) and Windows 10 Enterprise (64-bit) OS.
6. Before flashing full build, generate image first.
7. USB 3.0 port and USB 3.0 cable are recommended for flashing images.
8. If a Linux host is used to flash the RB5 device (hereafter referred to as device), run the following command before connecting the device to the host.

```
$ sudo systemctl stop ModemManager
```

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

4 Download and Unzip SDK Manager

To download SDK Manager, go to:

https://www.thundercomm.com/app_en/product/1590131656070623#sdkmanager.

Unzip:

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

Thundercomm Confidential

5 On Ubuntu 18.04 Host

1. Preparation:

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.

2. Install qemu-user-static & openssh-server:

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

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

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

4. Launch SDK manager:

```
$sdkmanager
```

5. Run SDK Manager:


Step 1. Provide Thundercomm login credentials:

Credential Checking ...

Enter your Thundercomm username:

Enter your Thundercomm password:

Step 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*).

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

Step 3. Enter the number of available version for image repack, for example, 1:

Checking current version of release ...

Available versions:

1: QRB5165.x.x.x-xxxxxx

Step 4. Enter **1** when the message below appears on your screen, or enter **help** for more information:

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.

Step 5. 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/[working directory]*.

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

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

Step 2. Power on the device (Required voltage: 12 V).

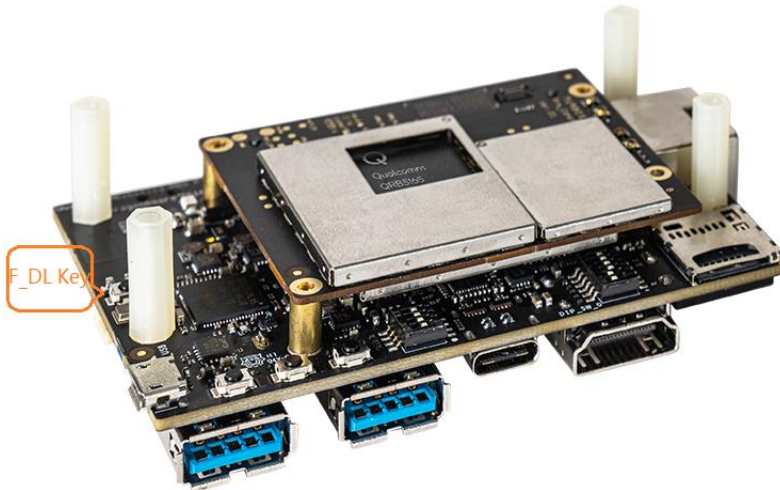


Figure 1. F_DL Key

Step 3. 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**).

Step 4. Release **F_DL key** after the board has been connected to your computer.

Step 5. Start flashing process from the SDK manager (command 2).

Step 6. SDK manager shall detect the device and start the flashing process automatically.

Step 7. 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 2. Wait for the flash process to be finished

7. With your device successfully boots up, enter the command below in a new terminal window of the host computer:

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

6 On Other Ubuntu Host (Ubuntu 16.04, 20.04)

1. Install qemu-user-static & openssh-server & udev to the host computer.

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

2. Refer to *Install using the repository* to install Docker:

<https://docs.docker.com/engine/install/ubuntu/>

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 'period' at the end of the command.
- Generated docker image name: **ubuntu:18.04-sdkmanager**.

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: Run the above command to generate a docker container name after **sdkmanager_container**.

5. Launch SDK Manager in Docker container:

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

6. Run SDK Manager:

Step 1. Provide Thundercomm login credentials:

Credential Checking ...

Enter your Thundercomm username:

Enter your Thundercomm password:

Step 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*).

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

Step 3. Enter the number of available version for image repack, for example, 1:

Checking current version of release ...

Available versions:

1: QRB5165.x.x.x-xxxxxx

Step 4. Enter **1** when the message below appears on your screen, or enter **help** for more information:

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.

Step 5. 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/[working directory]*.

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

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

Step 2. Power on the device (Required voltage: 12 V).

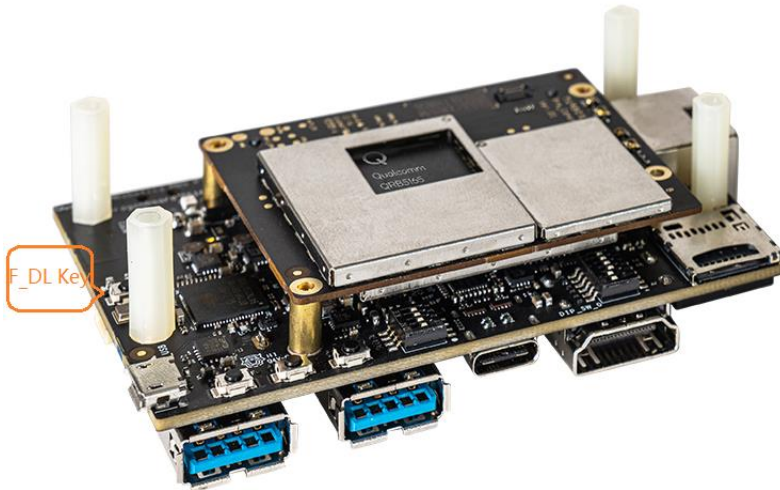


Figure 3. F_DL Key

Step 3. 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**).

Step 4. Release **F_DL key** after the board has been connected to your computer.

Step 5. Start flashing process from the SDK manager (command 2).

Step 6. SDK manager shall detect the device and start the flashing process automatically.

Step 7. 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 4. Wait for the flash process to be finished

8. With your device successfully boots up, enter the command below in a new terminal window of the host computer:

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

7 On Windows 10 (64-bit) Host

1. To download Docker Desktop, go to:
<https://hub.docker.com/editions/community/docker-ce-desktop-windows/>
2. Open **Dashboard** from docker notification menu to launch Docker Desktop.
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.
4. Generate Ubuntu 18.04 docker image:
Unzip *TC-sdkmanager-x.x.x.zip* and navigate to *TC/sdkmanager/x.x.x* directory from a Windows PowerShell, then 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**.

5. Create docker container.

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

 **NOTE:** Run the above command to generate a docker container name after **sdkmanager_container**.

6. Launch SDK Manager.

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

7. Run SDK Manager.

Step 1. Provide Thundercomm login credentials:

Credential Checking ...

Enter your Thundercomm username:

Enter your Thundercomm password:

Step 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*).

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

Step 3. Enter the available version number for image repack, for example, 1:

Checking current version of release ...

Available versions:

1: QRB5165.x.x.x-xxxxxx

Step 4. Enter 1 when the message below appears on your screen, or enter help for more information:

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.

Step 5. 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/[working directory]*.

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

\$ docker cp sdkmanager_container:[target_directory]/[name_of_selected_release]/full_build [destination path on Windows host PC]

Example: docker cp sdkmanager_container:/home/hostPC/demo_0803/QRB5165.x.x.x-xxxxxx/full_build D:

9. Flash the Device

Step 1. For more information about the MULTIDL_TOOL operation, go to https://www.thundercomm.com/app_en/product/1590131656070623#doc.

Step 2. Download *MULTIDL_TOOL_v1.0.14.zip* file and install the MULTIDL_TOOL (Refer to *MULTIDL_TOOL_USER_GUIDE.pdf*).

Step 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 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://www.thundercomm.com/app_en/product/1590131656070623#doc.

Step 4. Flash the full build with the MULTIDL_TOOL:

- 1) Launch MULTIDL_TOOL.
- 2) Click **Options >> Configuration** to set full-build path.
- 3) Enter **123456** at the password prompt.
- 4) Choose **Flat Build**, set Flash Type as **ufs**.
- 5) Select your local full-build path and programmer file (*prog_firehose_dds.elf*).
- 6) Click **Load XML** to load xml files. When xml files prompted, select all the XML files and all the Patch files in the *full_build\ufs* folder. Keep other settings as default. Then click **OK** to go back to main page.
- 7) Disconnect the device from the computer and power it off.
- 8) Power on the device (Required voltage: 12 V).

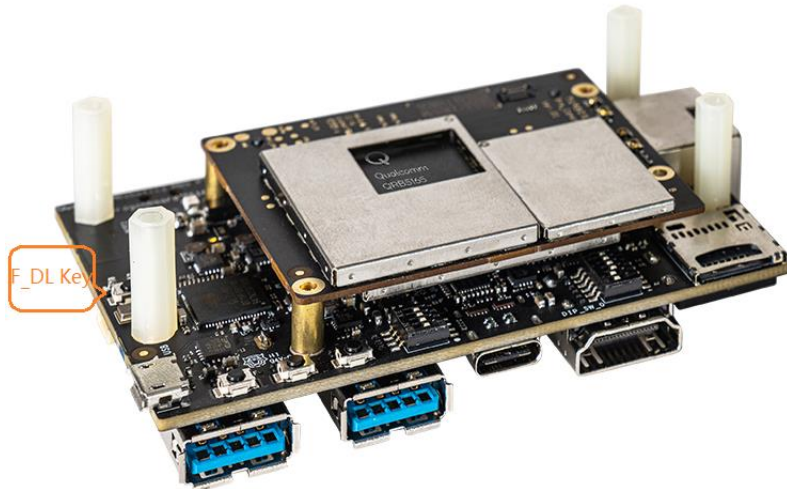


Figure 5. F_DL Key

- 9) 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).
- 10) Release **F_DL key** after the board has been connected to your computer.
- 11) Start the flashing process by clicking **START** button corresponding to your device port on MULTIDL_TOOL window.
- 12) After the flashing process been finished, the board will reboot automatically (this step may take some time).

8 Troubleshooting

Use the troubleshooting information to find solutions to problems that have definite symptoms.

Problem	Solution
<p>Internet Timeout Issue:</p> <p>Internet timeout issue may happen during the image generation process, such as Unable to fetch.</p>	<p>Try to run Command 1 again.</p>
<p>APT Source Issue</p>	<p>If the download fails, check the internet connection and the source list.</p>
<p>Device Boot Up Issue:</p> <p>SDK Manager cannot detect the device after the reboot.</p>	<ol style="list-style-type: none">1. If Ubuntu 18.04 is used on Docker, check whether adb kill-server is entered on the host PC before flashing image.2. Reboot your device manually, open a terminal, then enter adb shell.3. Check if any debian packages are modified.
<p>Process Issue</p> <p>The flashing process of Ubuntu system does not function well.</p>	<p>Copy the full folder to a computer with Windows system, then flash the image using Thundercomm MULTIDL_TOOL.</p> <p>For further information, refer to: MULTIDL_TOOL_USER_GUIDE_v2.pdf.</p>
<p>SDK Manager Flash Issue</p>	<p>Enter the following command on host machine before restart the flash.</p> <pre>\$ sudo systemctl stop ModemManager</pre>

Reference:

1. *Quick Start Guide*: <https://developer.qualcomm.com/qualcomm-robotics-rb5-kit/quick-start-guide>
2. *Hardware Reference Guide*: <https://developer.qualcomm.com/qualcomm-robotics-rb5-kit/hardware-reference-guide>
3. *Software Reference Manual*: <https://developer.qualcomm.com/qualcomm-robotics-rb5-kit/software-reference-manual>

Thundercomm Confidential