

# DC100 Linux SDK Quick Start

---

Release Version : V1.0.0

Release Date : 2023 - 12 - 14

Security Level : ☐Top-Secret ☐Secret ☐Internal ☒Public

## Overview

This document presents the basic usage of the SDK to help users get started with the DC100 Linux SDK quickly and easily.

## Product Version

Chipset	Kernel Version
RV1126	Linux 4.19

## Intended Audience

This guide document is intended for:

- Technical support engineer
- Software development engineer

## Revision History

Version	Author	Date	Revision History
V1.0.0	LEE	2023-12-14	Initial version

## Contents

### DC100 Linux SDK Quick Start

1. Set up Development Environment .....	5
2. SDK Configuration Framework Introduction .....	5
2.1 SDK Project Directory Introduction .....	5
2.2 DC100 Modules Directory Introduction .....	6
2.3 DC100 Development Documents .....	6
2.3.1 Documents Index.....	6
2.4 DC100 Development Tools.....	7
2.4.1 Window Tools.....	7
2.4.2 Linux Tools .....	7
2.5 SDK Configuration Framework.....	7
3. SDK Building Introduction.....	8
3.1 SDK Download Address .....	8
3.2 Cross-compile Tool Introduction.....	8
3.3 To Select Board Configuration .....	8
3.3.1 The Directory of SDK Board Config (device/rockchip/rv1126_rv1109).....	8
3.4 View Building Commands .....	9
3.5 Full Automatic Building.....	9
4. Update Firmware .....	10
4.1 Firmware upgrade using SD card.....	10
4.2 Firmware upgrade using Network.....	10
5. DC100 Function Introduction and Precaution.....	11
5.1 Get Device IP Address by DC100 example Tool .....	11
5.1.1 Get Device IP Address .....	11
5.1.2 Set as static IP Address .....	11
5.2 How to Debug With DC100 via Network .....	11
5.2.1 Debug With SSH .....	11

5.2.2 Debug With SCP ..... 12

# 1. Set up Development Environment

---

Recommended OS: Ubuntu 18.04 LTS

To set up the build environment, install the software package with the command below.

```
sudo apt-get install repo device-tree-compiler \
git-core u-boot-tools mtools \
parted libudev-dev libusb-1.0-0-dev \
python-linaro-image-tools linaro-image-tools \
autoconf autotools-dev libsigsegv2 m4 \
intltool libdrm-dev curl sed make binutils \
build-essential gcc g++ bash patch gzip gawk \
bzip2 perl tar cpio python unzip rsync \
file bc wget libncurses5 libqt4-dev libglib2.0-dev \
libgtk2.0-dev libglade2-dev cvs git \
mercurial openssh-client subversion asciidoc w3m \
dblatex graphviz python-matplotlib \
libc6:i386 libssl-dev expect fakeroot cmake flex \
bison liblz4-tool libtool keychain /
lib32gcc-7-dev g++-7 libstdc++-7-dev
```

## 2. SDK Configuration Framework Introduction

---

### 2.1 SDK Project Directory Introduction

The SDK directory contains kernel, buildroot, app, u-boot, device, docs, external, and other directories. Each directory or subdirectory corresponds to a git project, and commits must be performed in that directory.

- buildroot : Customized root file system.
- App : Store applications.
- External : Related libraries, including video.
- Kernel : kernel code.
- Device/rockchip : Stores some scripts and prepared files for building and packaging firmware of each chip.
- Docs : Stores development guides.
- Prebuilts : Stores cross-compilation toolchain.
- Rkbin : Stores firmware and tools.
- Rockdev : Stores building output firmware.

- Tools : Stores some commonly used tools.
- U-boot : U-Boot code.

## 2.2 DC100 Modules Directory Introduction

Modules directory path	Modules introduction
external/linux-rga	Raster Graphic Acceleration (RGA2)
external/recovery	Recovery and rockchip upgrade code
external/rkwifibt	WIFI and BT
external/rk_pcba_test	PCBA test code
external/mpp	Encoding and decoding
external/dcmedia	DC100 Multimedia interface
external/camera_engine_dcaiq	Image processing algorithm module
external/rknpu	NPU driver
external/common_algorithm	Video general algorithm library
external/rknn-toolkit	Development kit for model transformation, reasoning and performance evaluation
external/tof_seneor	tof_sensor library
external/ thermal_camera	thermal_camera library
app/dc100_test	DC100 example code

## 2.3 DC100 Development Documents

### 2.3.1 Documents Index

```

├─ docs
│   ├── DC100 Linux SDK Quick Start.pdf
│   ├── DC100 ISP Development Guide.pdf
│   ├── DC100 DCMedia Development Guide.pdf
│   ├── DC100 TOF Sensor Guide.pdf
│   └─ DC100 Thermal camera Guide.pdf
└─ external
    └─ rknpu
        └─ doc
            ├── Rockchip_User_Guide_RKNN_API_V1.3.3_CN.pdf
            └─ Rockchip_User_Guide_RKNN_API_V1.3.3_EN.pdf

```

## 2.4 DC100 Development Tools

### 2.4.1 Window Tools

Path : SDK/tools/windows/

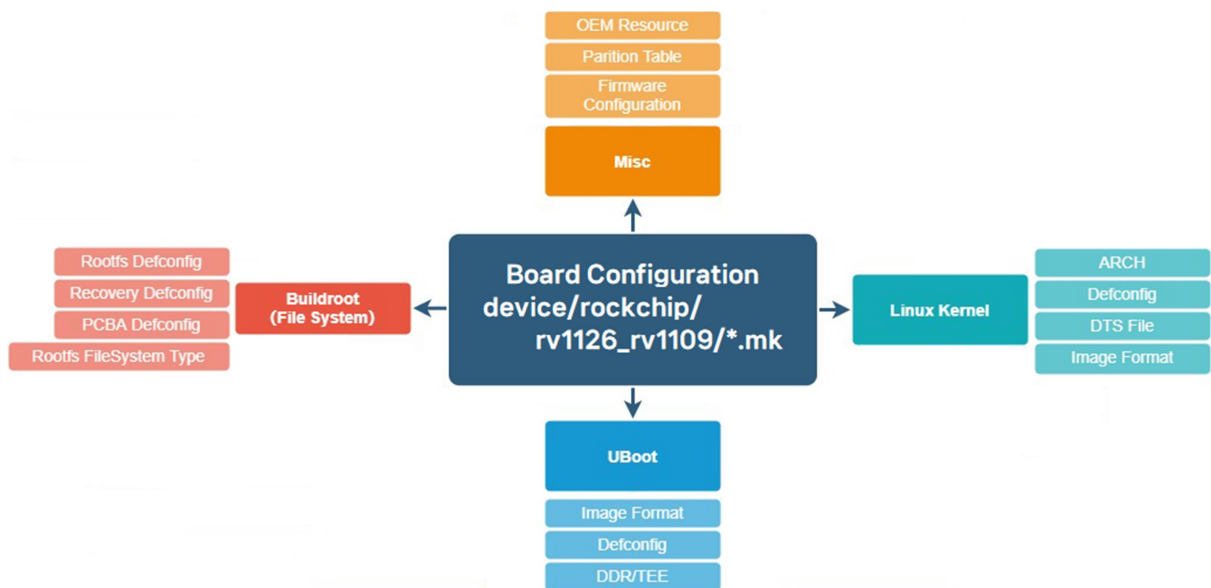
Tool's Name	Tool's Description
SDDiskTool	Sd card firmware create tool
DC100_example_tool	DC100 example run tool.

### 2.4.2 Linux Tools

Path : SDK/tools/linux/

Tool's Name	Tool's Description
Linux_Pack_Firmware	Make update.img tool.

## 2.5 SDK Configuration Framework



### 3. SDK Building Introduction

---

#### 3.1 SDK Download Address

```
repo init -u https://github.com/OpenAiCamera-DC100/manifests.git -b main -m dc100_linux_v1.0.0.xml  
  
repo sync && repo forall -c git lfs pull
```

#### 3.2 Cross-compile Tool Introduction

There are two cross-compile tools in the SDK, as follow:

Directory	Introduction
prebuilts/gcc/linux-x86/arm/gcc-arm-8.3-2019.03-x86_64-arm-linux-gnueabihf	used to build rootfs and app
prebuilts/gcc/linux-x86/arm/gcc-linaro-6.3.1-2017.05-x86_64_armlinux-gnueabihf	used to build U-Boot and Linux kernel

#### 3.3 To Select Board Configuration

##### 3.3.1 The Directory of SDK Board Config (device/rockchip/rv1126\_rv1109)

Board Configuration	Product Use	Storage Medium
BoardConfig.mk	DC100 default	eMMC
BoardConfig-cmos-dc100.mk	DC100 default	eMMC

Command of select board configure :

Select DC100 default board configuration

```
./build.sh lunch  
  
processing board option: lunch  
processing option: lunch  
  
You're building on Linux  
Lunch menu...pick a combo:
```



```
0. default BoardConfig.mk
1. BoardConfig-cmos-dc100.mk
Which would you like? [0]:
Lunching for Default BoardConfig.mk boards...
switching to board: /home/cmos/RV1126_RV1109_SDK/device/rockchip/rv1126_rv1109/BoardConfig.mk
```

### 3.4 View Building Commands

Execute the following command in the root directory: `./build.sh -h | help`

```
./build.sh help
Usage: build.sh [OPTIONS]
Available options:
BoardConfig*.mk    -switch to specified board config
lunch              -list current SDK boards and switch to specified board config
uboot              -build uboot
spl                -build spl
loader             -build loader
kernel             -build kernel
modules            -build kernel modules
toolchain          -build toolchain
rootfs             -build default rootfs, currently build buildroot as default
buildroot          -build buildroot rootfs
recovery           -build recovery
all                -build uboot, kernel, rootfs, recovery image
cleanall           -clean uboot, kernel, rootfs, recovery
firmware           -pack all the image we need to boot up system
updateimg          -pack update image
otapackage         -pack ab update otapackage image
save               -save images, patches, commands used to debug
allsave            -build all & firmware & updateimg & save

Default option is 'allsave'.
```

### 3.5 Full Automatic Building

If you run the following command in the SDK directory, all build processes will be completed automatically.

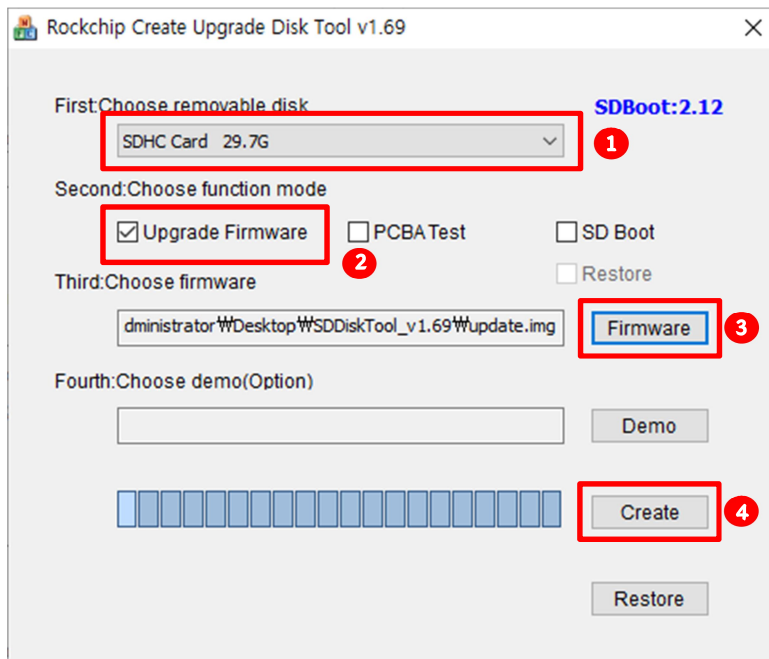
```
./build.sh all          # Only build with u-Boot, kernel, Rootfs and Recovery
                        # then use ./mkfirmware.sh to package firmware.

./build.sh allsave      # build with ./build.sh all, and then do these as follow:
                        # 1. package firmware to directory "rockdev"
```

```
# 2. package update.img  
# NOTE: ./build.sh allsave is the same as ./build.sh
```

## 4. Update Firmware

### 4.1 Firmware upgrade using SD card



- ① Select SD card.
- ② Check Upgrade Firmware.
- ③ Select Firmware. (SDK/rockdev/update.img)
- ④ Click the Create button.
- ⑤ Insert the Sd card and connect the power.

### 4.2 Firmware upgrade using Network

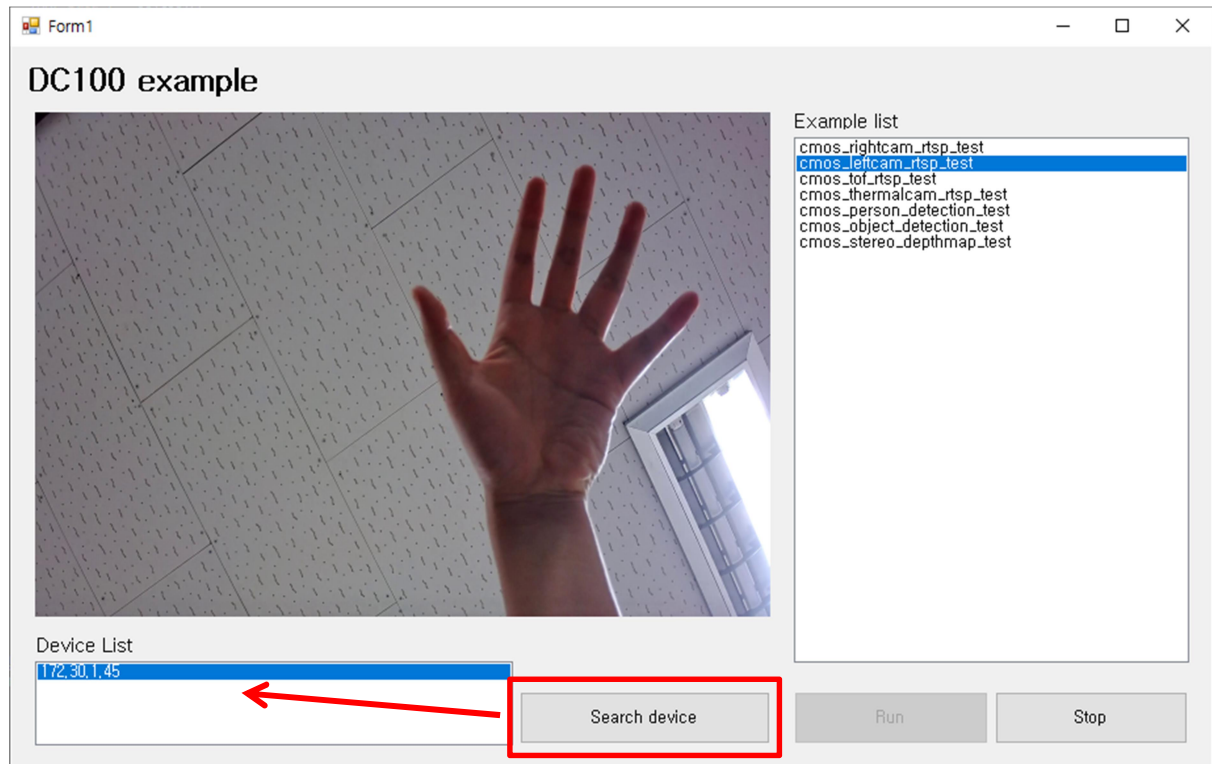
Install apache2 on your Linux PC using the following command and reboot.

```
apt-get install apache2  
/etc/init.d/apache2 restart  
cp SDK/rockdev/update.img /var/www/html/
```

Upgrade using the following command in the DC100 command window. (SSH or  
updateEngine --image\_url=http://DC100 IP address/update.img --misc=update --  
savepath=/userdata/update.img --reboot &

## 5. DC100 Function Introduction and Precaution

### 5.1 Get Device IP Address by DC100 example Tool



#### 5.1.1 Get Device IP Address

After running the DC100 example tool, when you click the search device button, the IP address of the connected device is created in the device list.

#### 5.1.2 Set as static IP Address

(To be updated)

## 5.2 How to Debug With DC100 via Network

### 5.2.1 Debug With SSH

After installing SSH Tool (putty, telnet ...), enter IP and connect.

account : root

password : cmos

### 5.2.2 Debug With SCP

**Upload test-file from PC to /tmp folder on DC100 board**

spc test-file root@DC100 IP Adress:/tmp/

root@ DC100 IP Adress's password:

**default passwd : cmos**