NXP Semiconductors

Document Number: AQSUG
Rev. P9.0.0_2.2.0-ga, 07/2019

Android™ Quick Start Guide

1 Overview

This document guides you through the processes of downloading and running this release package. It only explains how to download and run the default release image with default configuration. For details on using the release package, see the *Android™ User's Guide* (AUG) included in this release package.

2 Hardware Requirements

The hardware requirements for using this release package are as follows:

Supported system-on-chips (SoCs):

- i.MX 6Quad
- i.MX 6DualLite
- i.MX 6SoloX
- i.MX 6QuadPlus
- i.MX 7Dual
- i.MX 7ULP

Supported boards:

- SABRE-SD board and platform
- SABRE-AI board
- EVK board

Contents

3 Working with the i.MX 6Quad/6QuadPlus/6DualLite SABRE-SD Board and Platform

3.1 Board hardware

The figure below shows the different components of the SABRE-SD board.

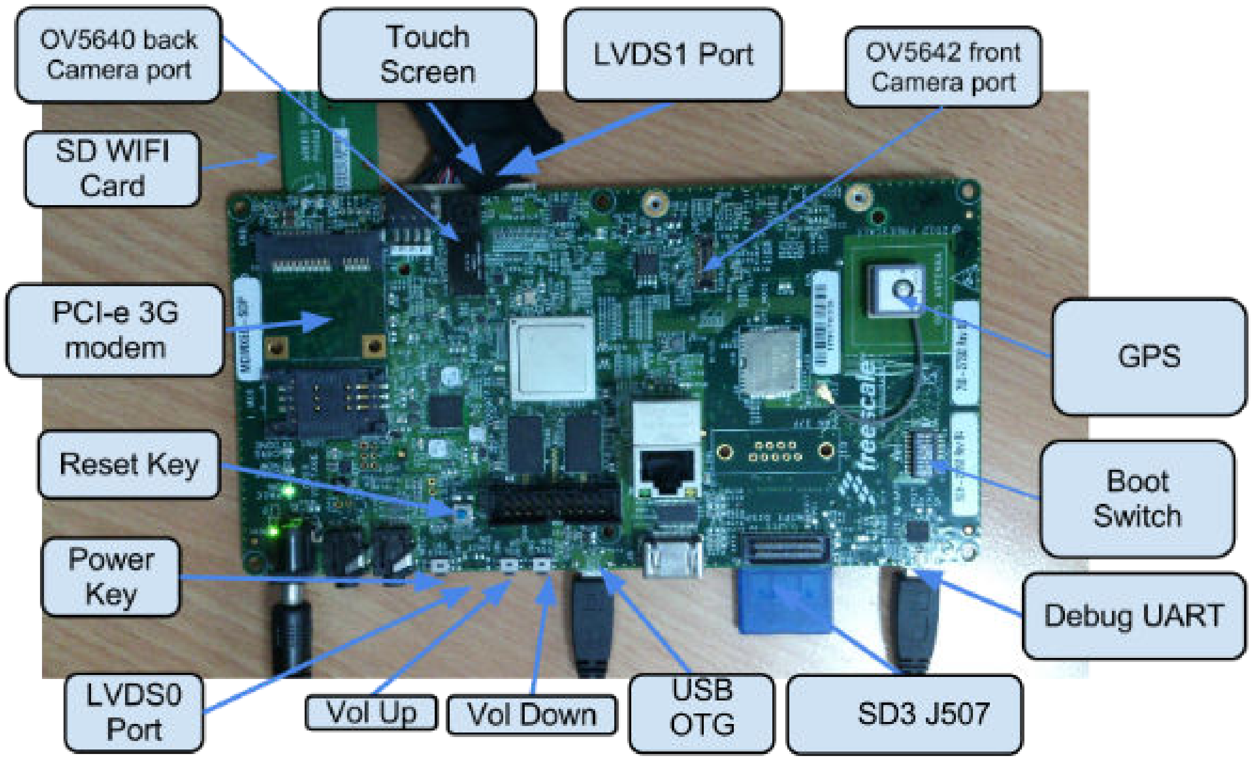

Figure 1. SABRE-SD Board

3.2 Board images

The table below describes the locations of the software images in android_p9.0.0_2.2.0-ga_image_6qsabresd.tar.gz on board partitions.

Table 1. Board images

Image name	Download target
/u-boot-imx6dl.imx	1 KB offset of eMMC boot0 partition or SD Card for i.MX 6DualLite SABRE-SD.
/u-boot-imx6q-ldo.imx	1 KB offset of eMMC boot0 partition or SD Card for 1.2 GHz i.MX 6Quad SABRE-SD.

Table continues on the next page...

Table 1. Board images (continued)

/u-boot-imx6q.imx	1 KB offset of eMMC boot0 partition or SD Card for 800 MHz or 1 GHz i.MX 6Quad SABRE-SD.
/u-boot-imx6qp-ldo.imx	1 KB offset of eMMC boot0 partition or SD Card for 1.2 GHz i.MX 6QuadPlus SABRE-SD.
/u-boot-imx6qp.imx	1 KB offset of eMMC boot0 partition or SD Card for 1 GHz i.MX 6QuadPlus SABRE-SD.
/u-boot-imx6dl-sabresd-uuu.imx	Bootloader used by UUU for i.MX 6DualLite SABRE-SD. It is not flashed to MMC.
/u-boot-imx6q-ldo-sabresd-uuu.imx	Bootloader used by UUU for 1.2 GHz i.MX 6Quad SABRE-SD. It is not flashed to MMC.
/u-boot-imx6q-sabresd-uuu.imx	Bootloader used by UUU for 800MHz/1GHz i.MX 6Quad SABRE-SD. It is not flashed to MMC.
/u-boot-imx6qp-ldo-sabresd-uuu.imx	Bootloader used by UUU for 1.2 GHz i.MX 6QuadPlus SABRE-SD. It is not flashed to MMC.
/u-boot-imx6qp-sabresd-uuu.imx	Bootloader used by UUU for 1 GHz i.MX 6QuadPlus SABRE-SD. It is not flashed to MMC.
/partition-table.img	Program to the first 17 KB, and then back up to the last 16.5 KB of eMMC user data partition or program to the first 0.5 KB wih MBR and the last 16.5 KB with GPT header and entries of the SD card. GPT table for 8 GB SD card/eMMC.
/partition-table-14GB.img	Program to the first 17 KB, and then back up to the last 16.5 KB of eMMC user data partition or program to the first 0.5 KB wih MBR and the last 16.5 KB with GPT header and entries of the SD card. GPT table for 16 GB SD card/eMMC.
/partition-table-28GB.img	Program to the first 17 KB, and then back up to the last 16.5 KB of eMMC user data partition or program to the first 0.5 KB wih MBR and the last 16.5 KB with GPT header and entries of the SD card. GPT table for 32 GB SD card/eMMC.
/dtbo-imx6dl.img	eMMC 1st partition for i.MX 6DualLite SABRE-SD.
/dtbo-imx6q-ldo.img	eMMC 1st partition for 1.2 GHz i.MX 6Quad SABRE-SD.
/dtbo-imx6q.img	eMMC 1st partition for 800 MHz or 1 GHz i.MX 6Quad SABRE-SD.
/dtbo-imx6qp-ldo.img	eMMC 1st partition for 1.2 GHz i.MX 6QuadPlus SABRE-SD.
/dtbo-imx6qp.img	eMMC 1st partition for 1 GHz i.MX 6QuadPlus SABRE-SD.
/boot.img	eMMC 2nd partition.
/recovery-imx6dl.img	eMMC 3rd partition for i.MX 6DualLite SABRE-SD.
/recovery-imx6dq-ldo.img	eMMC 3rd partition for 1.2 GHz i.MX 6Quad SABRE-SD.
/recovery-imx6q.img	eMMC 3rd partition for 800 MHz or 1 GHz i.MX 6Quad SABRE-SD.
/recovery-imx6qp-ldo.img	eMMC 3rd partition for 1.2 GHz i.MX 6QuadPlus SABRE-SD.
/recovery-imx6qp.img	eMMC 3rd partition for 1GHz i.MX 6QuadPlusSABRE-SD.
/system.img	eMMC 4th partition.
/vendor.img	eMMC 10th partition, holding platform binaries.
/vbmeta-imx6dl.img	eMMC 13th partition for i.MX 6DualLite SABRE-SD.
/vbmeta-imx6q-ldo.img	eMMC 13th partition for 1.2 GHz i.MX 6Quad SABRE-SD.
/vbmeta-imx6q.img	eMMC 13th partition for 800 MHz or 1 GHz i.MX 6Quad SABRE-SD.
/vbmeta-imx6qp-ldo.img	eMMC 13th partition for 1.2 GHz i.MX 6QuadPlus SABRE-SD.
/vbmeta-imx6qp.img	eMMC 13th partition for 1 GHz i.MX 6QuadPlus SABRE-SD.

3.3 Flashing board images

The board image files can be flashed into the target board by using Universal Update Utility (UUU). The UUU source code and binary file are released on github: uuu release page on github.

To achieve more flexibility, the following two script files are provided to invoke UUU to automatically flash all Android images.

- uuu_imx_android_flash.sh for Linux OS
- uuu_imx_android_flash.bat for Windows OS

For this release, these two scripts are validated on UUU 1.2.135 version. Download the corresponding version from github:

- For Linux OS, download the file named "uuu".
- For Windows OS, download the files named "uuu.exe".

Because the two files direclty invokes UUU, make sure that UUU is in a path contained by the system environment variable of "PATH".

Perform the following steps to download the board images:

1. Download the UUU binary file from github as described above.
   Place UUU in a directory contained by the system environment variable of "PATH". For example,
   - Linux OS users can put it in the directory of "/usr/sbin/", which is already contained by "PATH" with the system default configuration for all users. If UUU is put into a directory and the "PATH" environment is modified to include that directory, with "sudo" command to invoke UUU, the modified "PATH" may not take effect, since different users have different "PATH" values.
   - Windows OS users can put uuu.exe in any directory and then change the system variable of "PATH". "User varialbe" and "System variable" both need to be modified. After the modification, cmd/powershell needs to be restarted to make the modification take effect.
2. Make the board enter serial download mode.

   No dedicated boot dips are reserved for serial download mode on the SABRE-SD board. Therefore, a tricky method is used to enter serial download mode. Change the SABRE-SD SW6 (boot) to 00001100 (from 1-8 bit) to enter download mode.
3. Power on the board. Use the USB cable on the SABRE-SD OTG port to connect your PC with SABRE-SD.

   NOTE

   - There are two USB micro ports in SABRE-SD: USB-to-UART, USB OTG.
   - The USB-to-UART port is known as debug UART, and the USB OTG port is known as USB in the hardware image above.
   - The debug UART port can be used to watch the logs of the hardware boot processing.
4. Decompress release_package/android_p9.0.0_2.2.0-ga_image_6qsabresd.tar.gz, which contains the image files and uuu_imx_android_flash tool.
5. Execute the uuu_imx_android_flash tool to flash images.

   uuu_imx_android_flash tool can be executed with options to get help information and specify the images to be flashed. For i.MX 6DualLite SABRE-SD, i.MX 6QuadPlus SABRE-SD, and i.MX 6Quad SABRE-SD, related options are described in the following table.

Table 2. Options for the uuu_imx_android_flash tool

Option	Description
-h	Displays the help information for this tool.

Table continues on the next page...

Table 2. Options for the uuu_imx_android_flash tool (continued)

Option	Description
-f soc_name	Specifes the SoC information. For i.MX 6DualLite SABRE-SD, it is "imx6dl". For i.MX 6QuadPlus SABRE-SD, it is "imx6qp". For i.MX 6Quad SABRE-SD, it is "imx6q". This option is **mandatory**.
-c card_size	Specifies which partition table image file to flash. For i.MX 6DualLite SABRE-SD, i.MX 6QuadPlus SABRE-SD, and i.MX 6Quad SABRE-SD, it can be followed with "14" or "28". If this option is not used, default "partition-table.img" is flashed.
-d dev	Specifies some images with "dev" in its name. For i.MX 6QuadPlus SABRE-SD and i.MX 6Quad SABRE-SD, it can be "ldo". If this option is not used, default dtbo, recovery, and vbmeta images are flashed.
-e	Erases the user data after images are flashed.
-D directory	Specifies the directory where the images to be flashed. If this option is not used, images in the current working directory is flashed.
-t target_dev	Specifies the target device. For i.MX 6DualLite SABRE-SD, i.MX 6QuadPlus SABRE-SD, and i.MX 6Quad SABRE-SD, it can be "emmc" and "sd". If this option is not used, images are flashed to eMMC.
-p board	Specifies the board information. For i.MX 6DualLite SABRE-SD, i.MX 6QuadPlus SABRE-SD, and i.MX 6Quad SABRE-SD, it should be "sabresd". If "-d ldo" is not used, this option is **mandatory**.
-daemon	Run UUU in daemon mode. This option is used to flash multiple boards of the same type.
-i	If the script is executed with this option, no images is flashed. The script just loads U-Boot to RAM and executes to fastboot mode. This option is used for development.

- On the Linux OS, open the shell terminal. For example, you can execute a command as follows:

```
> sudo ./uuu_imx_android_flash.sh -f imx6qp -p sabresd -e
```

- On the Windows OS, run `cmd/powershell` in administrator mode, the corresponding command is as follows:

```
> .\uuu_imx_android_flash.bat -f imx6qp -p sabresd -e
```

With the command above executed, the default images are flashed into eMMC for 1 GHz version i.MX 6QuadPlus SABRE-SD.

NOTE

- To flash the SD card, execute the tool with "-t sd". To flash eMMC, it does not need to use the -t option.
- If your SD card is 8 GB or the onboard eMMC is used, it does not need to use the -c option.
- If your SD card is 16 GB, execute the tool with "-c 14".
- If your SD card is 32 GB, execute the tool with "-c 28".
- To download images for 1 GHz i.MX 6QuadPlus SABRE-SD boards, it does not need to use the -d option.
- To download images for 800 MHz/1 GHz i.MX 6Quad SABRESD boards, it does not need to use the -d option.
- To download images for 1.2 GHz i.MX 6QuadPlus SABRESD boards, execute the tool with "-d ldo".
- To download images for 1.2 GHz i.MX 6Quad SABRESD boards, execute the tool with "-d ldo".

- If "-d ldo" is used, it does not need to use "-p sabresd", because only i.MX 6QuadPlus SABRESD and i.MX 6Quad SABRESD have 1.2 GHz version, while i.MX 6QuadPlus SABRE-AI and i.MX 6Quad SABRE-AI does not.
- uuu_imx_android_flash.bat generates the temporary file under the current working directory. Make sure you have the write permission under the current working directory.
- If uuu_imx_android_flash.bat is used to flash images on a remote server through samba, you need to map the remote resource to the local environment first. Take the following command as an example:

```
> net use z: \\10.193.108.179\daily_images
```

"z" in the command represents an available drive letter, which can be other available drive letter.

6. Wait the uuu_imx_android_flash execution to be completed. If there is no error, the command window prompts that images are already flashed.
7. Power down the board.
8. Change the boot device to eMMC or SD card in one of the following methods:
    - Change "Boot Switch(SW6)" to 11100110 (from 1-8 bit) to switch the board back to eMMC 4-bit boot mode.
    - Change "Boot Switch(SW6)" to 11010110 (from 1-8 bit) to switch the board back to eMMC 8-bit boot mode.
    - Change "Boot Switch(SW6)" to 01000010 (from 1-8 bit) to switch the board back to SD Card boot mode.

3.4 Booting

When the Android system image is programmed and the Boot Switch(es) is configured, the system is ready to be powered on.

There are three kinds of display configurations supported in this release between LVDS display pannels and HDMI output.

- To enable the LVDS1 display, see Section Booting with single display: LVDS display.
- To enable single HDMI display, see Section Booting with single display: HDMI display.
- To enable LVDS1 and HDMI output dual display feature, see Section Booting with dual displays: LVDS and HDMI displays.

NOTE

There are two LVDS ports in SABRE SD hardware: LVDS0 and LVDS1. LVDS1 is the primary display in this release. The LVDS1 port is a nearby miniPCIe interface (see the SABRE-SD board image above).

3.4.1 Booting with single display: LVDS display

In the U-Boot prompt, set the U-Boot environment variables as shown below:

```
U-Boot > setenv bootargs console=ttymxc0,115200 init=/init
video=mxcfb0:dev=ldb,fbpix=RGB32,bpp=32 video=mxcfb1:off video=mxcfb2:off video=mxcfb3:off
vmalloc=128M androidboot.console=ttymxc0 consoleblank=0  androidboot.hardware=freescale
cma=320M galcore.contiguousSize=33554432 loop.max_part=7
U-Boot > saveenv
```

With the settings above, the Android platform does not start the shell console. To disable selinux, append "androidboot.selinux=permissive" to the U-Boot's bootargs. Boot environment variables are as follows:

```
U-Boot > setenv append_bootargs androidboot.selinux=permissive
U-Boot > saveenv
```

3.4.2 Booting with single display: HDMI display

In the U-Boot prompt, set the U-Boot environment variables as shown below:

```
U-Boot > setenv bootcmd boota mmc2
U-Boot > setenv bootargs console=ttymxc0,115200 androidboot.console=ttymxc0 consoleblank=0
vmalloc=128M init=/init video=mxcfb0:dev=hdmi,1920x1080M@60,bpp=32 video=mxcfb1:off
video=mxcfb2:off video=mxcfb3:off androidboot.hardware=freescale cma=448M
galcore.contiguousSize=33554432 loop.max_part=7
U-Boot > saveenv
```

With the settings above, the Android platform does not start the shell console. To disable selinux, append "androidboot.selinux=permissive" to the U-Boot's bootargs. Boot environment variables are as follows:

```
U-Boot > setenv append_bootargs androidboot.selinux=permissive
U-Boot > saveenv
```

3.4.3 Booting with dual displays: LVDS and HDMI displays

In the U-Boot prompt, set the U-Boot environment variables as shown below:

```
U-Boot > setenv bootargs console=ttymxc0,115200 androidboot.console=ttymxc0 consoleblank=0
vmalloc=128M init=/init video=mxcfb0:dev=ldb,fbpix=RGB32,bpp=32 video=mxcfb1:dev=hdmi,
1920x1080M@60,bpp=32 video=mxcfb2:off video=mxcfb3:off androidboot.hardware=freescale
cma=320M galcore.contiguousSize=33554432 loop.max_part=7
U-Boot > saveenv
```

With the settings above, the Android platform does not start the shell console. To disable selinux, append "androidboot.selinux=permissive" to the U-Boot's bootargs. Boot environment variables are as follows:

```
U-Boot > setenv append_bootargs androidboot.selinux=permissive
U-Boot > saveenv
```

3.5 Board reboot

After completing download and setup, reboot the board and wait for the Android platform to boot up.

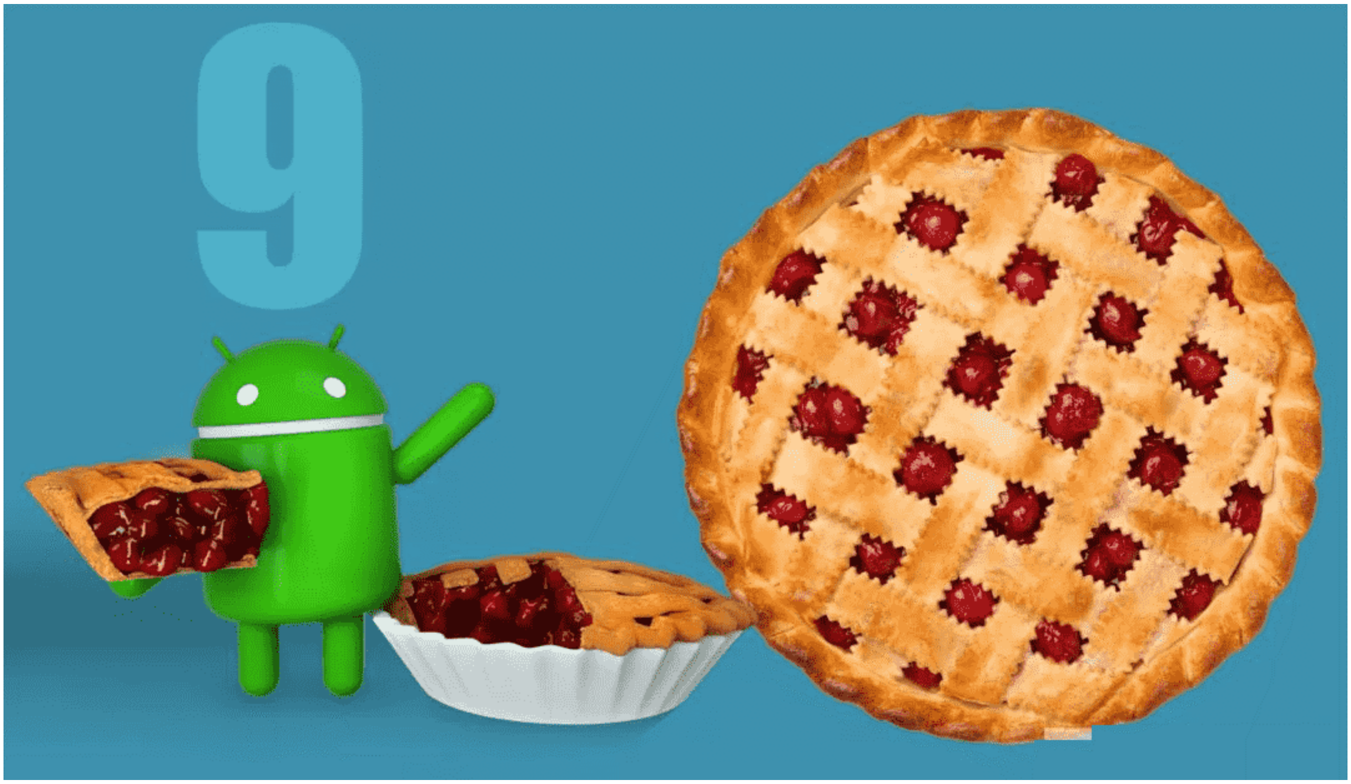

Figure 2. Android Pie image

4 Working with the i.MX 6QuadPlus/6Quad/6DualLite SABRE-AI Platform

4.1 Board hardware

The following figure shows the different components of the i.MX 6Quad/6DualLite SABRE-AI board.

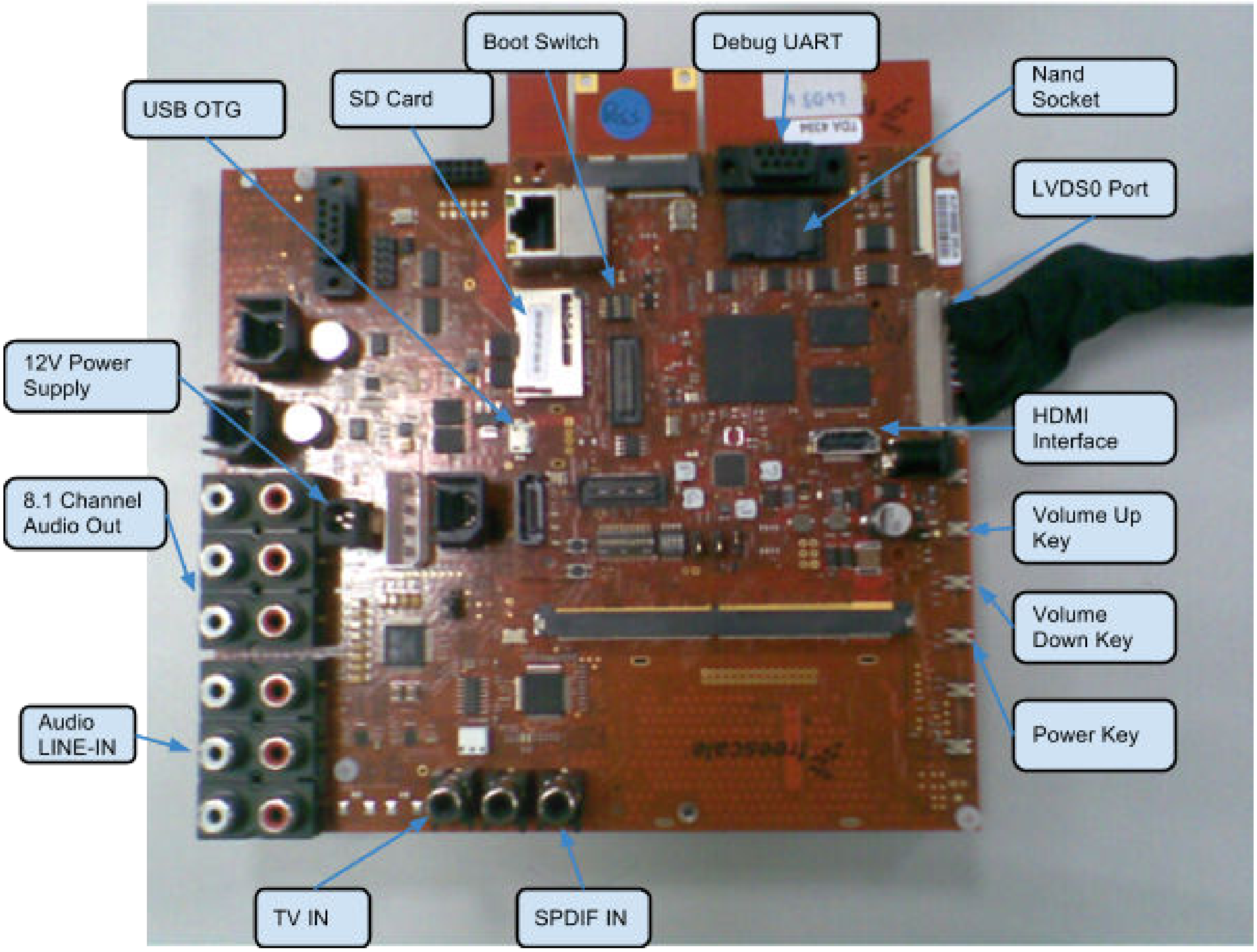

Figure 3. i.MX 6Quad/6DualLite SABRE-AI board

The following figure shows the different components of the i.MX 6QuadPlus SABRE-AI board.

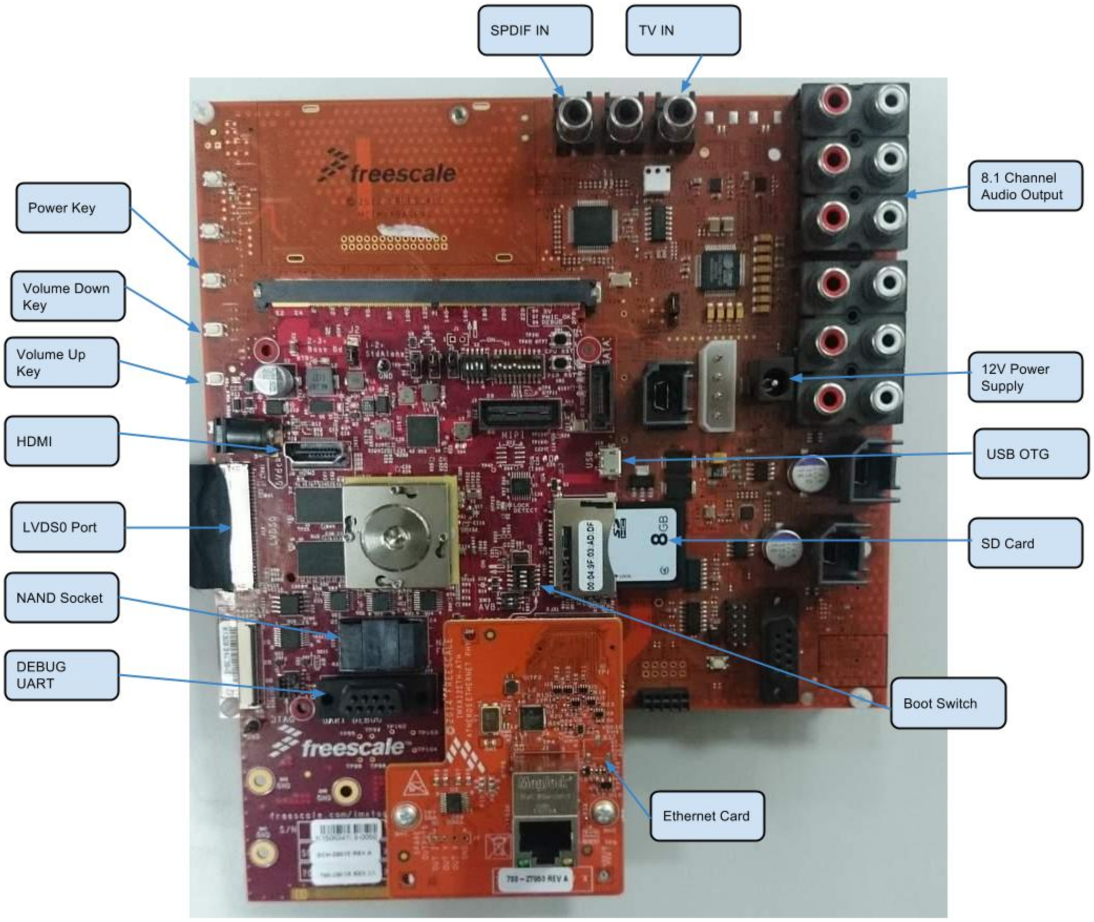

Figure 4. i.MX 6QuadPlus SABRE-AI Board

4.2 Board images

The table below describes the location in the board partitions of the software images in android_p9.0.0_2.2.0-ga_image_6qsabreauto.tar.gz on board partitions.

Table 3. Board images

Image name	Download target
/u-boot-imx6dl.imx	1 KB offset of SD card for i.MX 6DualLite SABRE-AI.
/u-boot-imx6q.imx	1 KB offset of SD card for i.MX 6Quad SABRE-AI.
/u-boot-imx6qp.imx	1 KB offset of SD card for i.MX 6QuadPlus SABRE-AI.
/u-boot-imx6dl-sabreauto-UUU.imx	Bootloader used by UUU for i.MX 6DualLite SABRE-AI. It is not flashed to MMC.
/u-boot-imx6q-sabreauto-UUU.imx	Bootloader used by UUU for i.MX 6Quad SABRE-AI. It is not flashed to MMC.
/u-boot-imx6qp-sabreauto-UUU.imx	Bootloader used by UUU for i.MX 6QuadPlus SABRE-AI. It is not flashed to MMC.
/partition-table.img	Program to the first 0.5 KB wih MBR and the last 16.5 KB with GPT header and entries of the SD card. GPT table for 8 GB SD card.

Table continues on the next page...

Table 3. Board images (continued)

/partition-table-14GB.img	Program to the first 0.5 KB wih MBR and the last 16.5 KB with GPT header and entries of the SD card. GPT table for 16 GB SD card.
/partition-table-28GB.img	Program to the first 0.5 KB wih MBR and the last 16.5 KB with GPT header and entries of the SD card. GPT table for 32 GB SD card.
/dtbo-imx6dl.img	SD 1st partition for i.MX 6DualLite SABRE-AI.
/dtbo-imx6q.img	SD 1st partition for i.MX 6Quad SABRE-AI.
/dtbo-imx6qp.img	SD 1st partition for i.MX 6QuadPlus SABRE-AI.
/boot.img	SD 2nd partition.
/recovery-imx6dl.img	SD 3rd partition for i.MX 6DualLite SABRE-AI.
/recovery-imx6q.img	SD 3rd partition for i.MX 6Quad SABRE-AI.
/recovery-imx6qp.img	SD 3rd partition for i.MX 6QuadPlus SABRE-AI.
/system.img	SD 4th partition.
/vendor.img	SD 10th partition, holding platform binaries.
/vbmeta-imx6dl.img	SD 13th partition for i.MX 6DualLite SABRE-AI.
/vbmeta-imx6q.img	SD 13th partition for i.MX 6Quad SABRE-AI.
/vbmeta-imx6qp.img	SD 13th partition for i.MX 6QuadPlus SABRE-AI.

4.3 Flashing board images

The board image files can be flashed into the target board by using Universal Update Utility (UUU). The UUU source code and binary file are released on github: uuu release page on github.

To achieve more flexibility, the following two script files are provided to invoke UUU to automatically flash all Android images.

- uuu_imx_android_flash.sh for Linux OS
- uuu_imx_android_flash.bat for Windows OS

For this release, these two scripts are validated on UUU 1.2.135 version. Download the corresponding version from github:

- For Linux OS, download the file named "uuu".
- For Windows OS, download the files named "uuu.exe".

Because the two files direclty invokes UUU, make sure that UUU is in a path contained by the system environment variable of "PATH".

Perform the following steps to download the board images:

1. Download the UUU binary file from github as described above. Install UUU into a directory contained by the system environment variable of "PATH".
2. Make the board enter serial download mode.

   Change the SABRE-AI S3 (boot mode) to 0101 (from 1-4 bit) to enter download mode.

   Change S2,S1(boot configuration) to 0010,0100100000 to boot from the SD card.
3. Power on the board. Use the USB cable on the SABRE-AUTO OTG port to connect your PC with SABRE-AI.

NOTE

The USB micro port on SABRE-AI is J10.

4. Decompress release_package/android_p9.0.0_2.2.0-ga_image_6qsabreauto.tar.gz, which contains the image files and uuu_imx_android_flash tool.
5. Execute the uuu_imx_android_flash tool to flash images.

   uuu_imx_android_flash tool can be executed with options to get help information and specify the images to be flashed. For i.MX 6DualLite SABRE-AI, i.MX 6QuadPlus SABRE-AI, and i.MX 6Quad SABRE-AI, related options are described in the following table.

Table 4. Options for the uuu_imx_android_flash tool

Option	Description
-h	Displays the help information for this tool.
-f soc_name	Specifes the SoC information. For i.MX 6DualLite SABRE-AI, it is "imx6dl". For i.MX 6QuadPlus SABRE-AI, it is "imx6qp". For i.MX 6Quad SABRE-AI, it is "imx6q". This option is **mandatory**.
-c card_size	Specifies which partition table image file to flash. For i.MX 6DualLite SABRE-AI, i.MX 6QuadPlus SABRE-AI, and i.MX 6Quad SABRE-AI, it can be followed with "14" or "28". If this option is not used, default "partition-table.img" is flashed.
-e	Erases the user data after images are flashed.
-D directory	Specifies the directory where the images to be flashed. If this option is not used, images in the current working directory is flashed.
-p board	Specifies the board information. For SABRE-AI, it should be "sabreauto". This option is **mandatory** for SABRE-AI, since the same SoC may be used on SABRE-SD.
-daemon	Run UUU in daemon mode. This option is used to flash multiple boards of the same type.
-i	If the script is executed with this option, no images is flashed. The script just loads U-Boot to RAM and executes to fastboot mode. This option is used for development.

- On the Linux OS, open the shell terminal. For example, you can execute a command as follows:

```
> sudo ./uuu_imx_android_flash.sh -f imx6qp -p sabreauto -e
```

- On the Windows OS, run `cmd/powershell` in administrator mode, the corresponding command is as follows:

```
> .\uuu_imx_android_flash.bat -f imx6qp -p sabreauto -e
```

With the command above executed, the default images are flashed into the SD card for i.MX 6QuadPlus SABRE-AI.

NOTE

- SABRE-AI only supports SD card as the target device, so it does not need to specify the target device when executing the uuu_imx_android_flash tool.
- If your SD card is 8 GB, it does not need to use the -c option.
- If your SD card is 16 GB, execute the tool with "-c 14".
- If your SD card is 32 GB, execute the tool with "-c 28".
- uuu_imx_android_flash.bat generates the temporary file under the current working directory. Make sure you have the write permission under the current working directory.
- If uuu_imx_android_flash.bat is used to flash images on a remote server through samba, you need to map the remote resource to the local environment first. Take the following command as an example:

```
> net use z: \\10.193.108.179\daily_images
```

"z" in the command represents an available drive letter, which can be other available drive letter.

6. Wait the uuu_imx_android_flash execution to be completed. If there is no error, the command window prompts that images are already flashed.
7. Power down the board.
8. Change the boot device to SD card.

   Change "Boot Switch (S3)" to 0010 (from 1-4 bit) to switch the board back to SD boot mode.

4.4 Booting

When the Android system image is programmed and the Boot Switch(es) is configured, the system is ready to be powered on.

There are two hardware displays supported in this release: one LVDS display panel and one HDMI output.

- To enable the LVDS0 display, see Section Booting with single display: LVDS display.
- To enable the HDMI display, see Section Booting with single display: HDMI display.
- To enable LVDS0 and HDMI output dual display feature, see Section Booting with dual displays: LVDS and HDMI displays.

NOTE

There are two LVDS ports in SabreAuto hardware, LVDS0 and LVDS1. LVDS0 is taken as the primary display in this release. The LVDS0 port is in the CPU board. The LVDS1 is in the base board.

4.4.1 Booting with single display: LVDS display

To boot from SD, set the U-Boot environment variables as shown below:

```
U-Boot > setenv bootargs console=ttymxc3,115200 init=/init
video=mxcfb0:dev=ldb,fbpix=RGB32,bpp=32 video=mxcfb1:off video=mxcfb2:off video=mxcfb3:off
vmalloc=128M androidboot.console=ttymxc3 consoleblank=0 androidboot.hardware=freescale
cma=512M galcore.contiguousSize=67108864 loop.max_part=7
U-Boot > saveenv
```

With the settings above, the Android platform does not start the shell console. To disable selinux, append "androidboot.selinux=permissive" to the U-Boot's bootargs. Boot environment variables are as follows:

```
U-Boot > setenv append_bootargs androidboot.selinux=permissive
U-Boot > saveenv
```

4.4.2 Booting with single display: HDMI display

To boot from SD, set the U-Boot environment variables as shown below:

```
U-Boot > setenv bootargs console=ttymxc3,115200 androidboot.console=ttymxc3 consoleblank=0
vmalloc=128M init=/init video=mxcfb0:dev=hdmi,1920x1080M@60,bpp=32 video=mxcfb1:off
video=mxcfb2:off video=mxcfb3:off androidboot.hardware=freescale cma=512M
galcore.contiguousSize=67108864 loop.max_part=7
U-Boot > saveenv
```

With the settings above, the Android platform does not start the shell console. To disable selinux, append "androidboot.selinux=permissive" to the U-Boot's bootargs. Boot environment variables are as follows:

```
U-Boot > setenv append_bootargs androidboot.selinux=permissive
U-Boot > saveenv
```

4.4.3 Booting with dual displays: LVDS and HDMI displays

To boot from SD, set the U-Boot environment variables as shown below:

```
U-Boot > setenv bootargs console=ttymxc3,115200 androidboot.console=ttymxc3 consoleblank=0
vmalloc=128M init=/init video=mxcfb0:dev=ldb,bpp=32 video=mxcfb1:dev=hdmi,
1920x1080M@60,bpp=32 video=mxcfb2:off video=mxcfb3:off androidboot.hardware=freescale
cma=512M galcore.contiguousSize=67108864 loop.max_part=7
U-Boot > saveenv
```

With the settings above, the Android platform does not start the shell console. To disable selinux, append "androidboot.selinux=permissive" to the U-Boot's bootargs. Boot environment variables are as follows:

```
U-Boot > setenv append_bootargs androidboot.selinux=permissive
U-Boot > saveenv
```

4.5 Board reboot

After you have completed download and setup, reboot the board and wait for the Android platform to boot up.

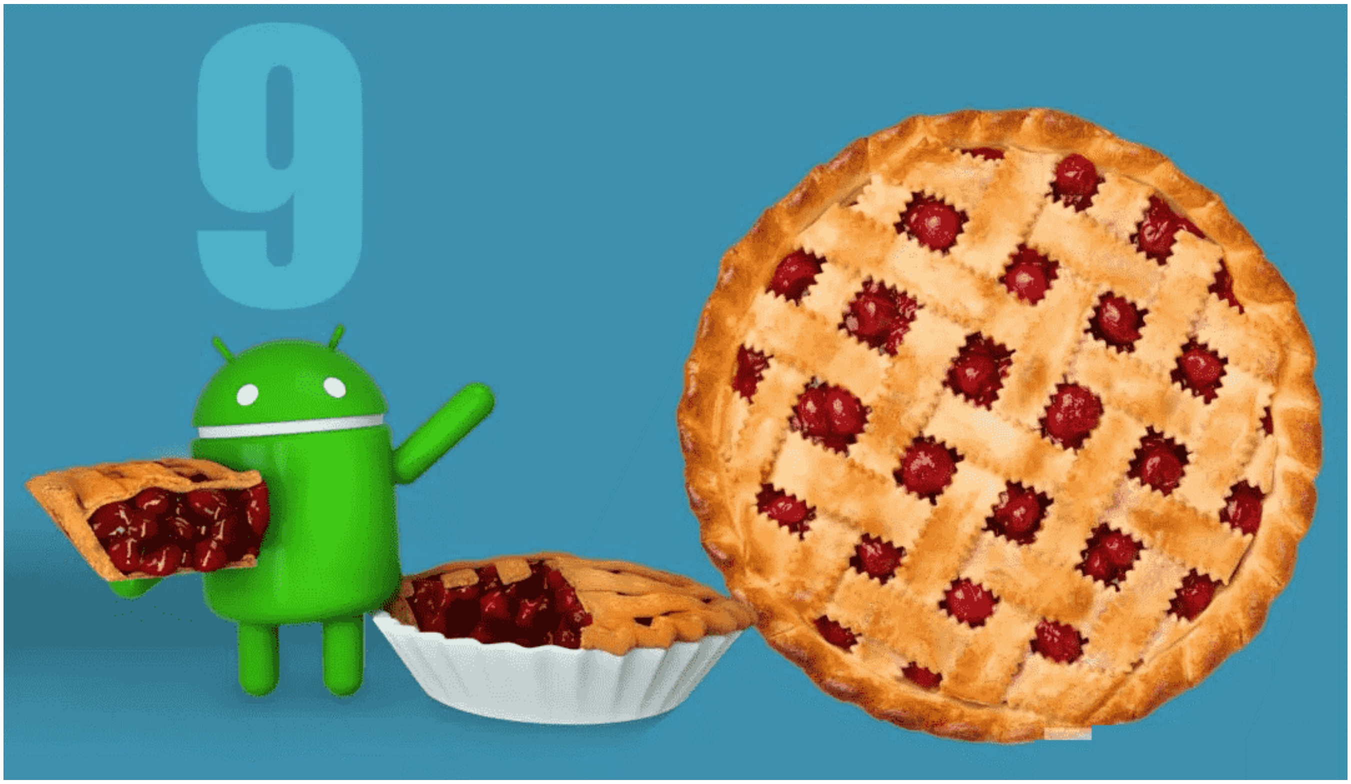

Figure 5. Android Pie image

5 Working with the i.MX 6SoloX SABRE-SD Board

5.1 Board hardware

The figure below shows the different components of the i.MX 6SoloX SABRE-SD board.

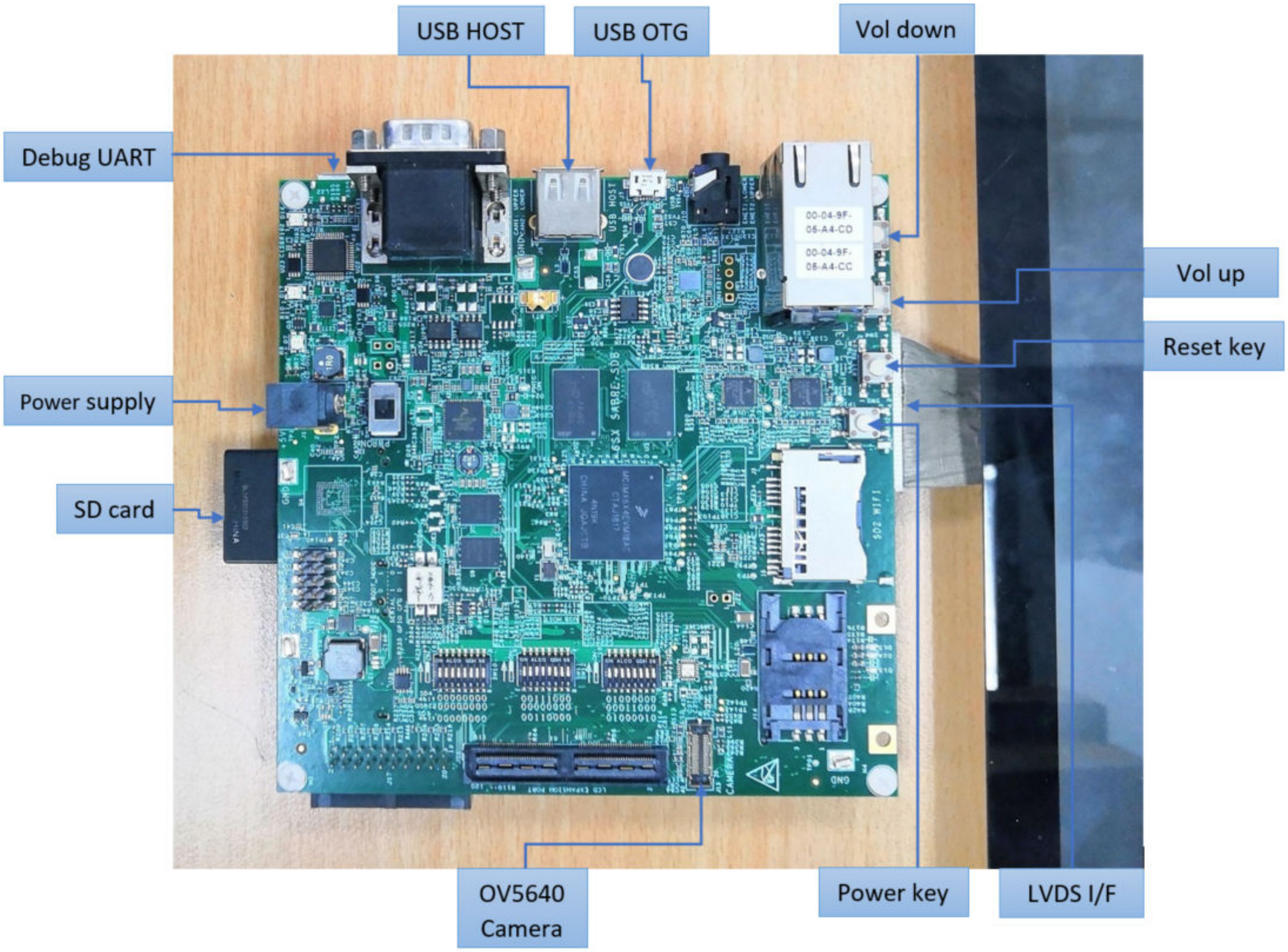

Figure 6. i.MX 6SoloX SABRE-SD board

5.2 Board images

The table below describes the location in the board partitions of the software images in android_p9.0.0_2.2.0-ga_image_6sxsabresd.tar.gz on board partitions.

Table 5. Board images

Image name	Download target

Table continues on the next page...

Table 5. Board images (continued)

/u-boot-imx6sx.imx	1 KB offset of SD card.
/u-boot-imx6sx-sabresd-uuu.imx	Bootloader used by UUU for i.MX 6SoloX SABRE-SD. It is not flashed to the SD card.
/partition-table.img	Program to the first 0.5 KB wih MBR and the last 16.5 KB with GPT header and entries of the SD card. GPT table for 8 GB SD card.
/partition-table-14GB.img	Program to the first 0.5 KB wih MBR and the last 16.5 KB with GPT header and entries of the SD card. GPT table for 16 GB SD card.
/partition-table-28GB.img	Program to the first 0.5 KB wih MBR and the last 16.5 KB with GPT header and entries of the SD card. GPT table for 32 GB SD card.
/dtbo-imx6sx.img	SD 1st partition.
/boot.img	SD 2nd partition.
/recovery-imx6sx.img	SD 3rd partition.
/system.img	SD 4th partition.
/vendor.img	SD 10th partition, holding platform binaries.
/vbmeta-imx6sx.img	SD 13th partition.

5.3 Flashing board images

The board image files can be flashed into the target board by using Universal Update Utility (UUU). The UUU source code and binary file are released on github: uuu release page on github.

To achieve more flexibility, the following two script files are provided to invoke UUU to automatically flash all Android images.

- uuu_imx_android_flash.sh for Linux OS
- uuu_imx_android_flash.bat for Windows OS

For this release, these two scripts are validated on UUU 1.2.135 version. Download the corresponding version from github:

- For Linux OS, download the file named "uuu".
- For Windows OS, download the files named "uuu.exe".

Because the two files direclty invokes UUU, make sure that UUU is in a path contained by the system environment variable of "PATH".

Perform the following steps to download the board images:

1. Download the UUU binary file from github as described above. Install UUU into a directory contained by the system environment variable of "PATH".
2. Make the board enter serial download mode.

   Change the SoloX SABRE-SD board's S1 (boot mode) to 10 (from 1-2 bit) to enter download mode.
3. Power on the board. Use the USB cable on the SABRE-SD OTG port to connect your PC with SABRE-SD.

NOTE

- There are two USB micro ports in SABRE-SD: USB-to-UART, USB OTG.
- The USB-to-UART port is known as debug UART, and the USB OTG port is known as USB in the hardware image above.
- The debug UART port can be used to watch the logs of the hardware boot processing.

4. Decompress release_package/android_p9.0.0_2.2.0-ga_image_6sxsabresd.tar.gz, which contains the image files and uuu_imx_android_flash tool.
5. Execute the uuu_imx_android_flash tool to flash images.

   uuu_imx_android_flash tool can be executed with options to get help information and specify the images to be flashed. For i.MX 6SoloX SABRE-SD, related options are described in the following table.

Table 6. Options for the uuu_imx_android_flash tool

Option	Description
-h	Displays the help information for this tool.
-f soc_name	Specifes the SoC information. For i.MX 6SoloX SABRE-SD, it is "imx6sx". This option is **mandatory**.
-c card_size	Specifies which partition table image file to flash. For i.MX 6SoloX SABRE-SD, it can be followed with "14" or "28". If this option is not used, default "partition-table.img" is flashed.
-e	Erases the user data after images are flashed.
-D directory	Specifies the directory where the images to be flashed. If this option is not used, images in the current working directory is flashed.
-daemon	Run UUU in daemon mode. This option is used to flash multiple boards of the same type.
-i	If the script is executed with this option, no images is flashed. The script just loads U-Boot to RAM and executes to fastboot mode. This option is used for development.

- On the Linux OS, open the shell terminal. For example, you can execute a command as follows:

```
> sudo ./uuu_imx_android_flash.sh -f imx6sx -p sabresd -e
```

- On the Windows OS, run `cmd/powershell` in administrator mode, the corresponding command is as follows:

```
> .\uuu_imx_android_flash.bat -f imx6sx -p sabresd -e
```

  With the command above executed, the default images are flashed into the SD card for i.MX 6SoloX SABRE-SD.

NOTE

- i.MX 6SoloX SABRE-SD only supports SD card as the target device, so it does not need to specify the target device when executing the uuu_imx_android_flash tool.
- If your SD card is 8 GB or the onboard eMMC is used, it does not need to use the -c option.
- If your SD card is 16 GB, execute the tool with "-c 14".
- If your SD card is 32 GB, execute the tool with "-c 28".
- uuu_imx_android_flash.bat generates the temporary file under the current working directory. Make sure you have the write permission under the current working directory.
- If uuu_imx_android_flash.bat is used to flash images on a remote server through samba, you need to map the remote resource to the local environment first. Take the following command as an example:

```
> net use z: \\10.193.108.179\daily_images
```

  "z" in the command represents an available drive letter, which can be other available drive letter.

6. Wait for the uuu_imx_android_flash execution to be completed. If there is no error, the command window prompts that images are already flashed.
7. Power down the board.
8. Change the boot device to SD card:
    - Change S1 (boot mode) to 01 (from 1-2 bit).
    - Change "Boot Switch (SW10,11,12)" to 00000000 (from 1-8 bit), 00111000 (from 1-8 bit), 01000000 (from 1-8 bit) to switch the board back to SD4 boot mode.

5.4 Booting with single display: LVDS display

When the Android system image is programmed and the Boot Switch(es) is configured, the system is ready to be powered on.

In the U-Boot prompt, set the U-Boot environment variables as shown below:

```
U-Boot > setenv bootargs console=ttymxc0,115200 init=/init androidboot.console=ttymxc0
consoleblank=0 androidboot.hardware=freescale vmalloc=128M cma=320M
galcore.contiguousSize=33554432 loop.max_part=7
U-Boot > saveenv
```

With the settings above, the Android platform does not start the shell console. To disable selinux, append "androidboot.selinux=permissive" to the U-Boot's bootargs. Boot environment variables are as follows:

```
U-Boot > setenv append_bootargs androidboot.selinux=permissive
U-Boot > saveenv
```

5.5 Board reboot

After you have completed download and setup, reboot the board and wait for the Android platform to boot up.

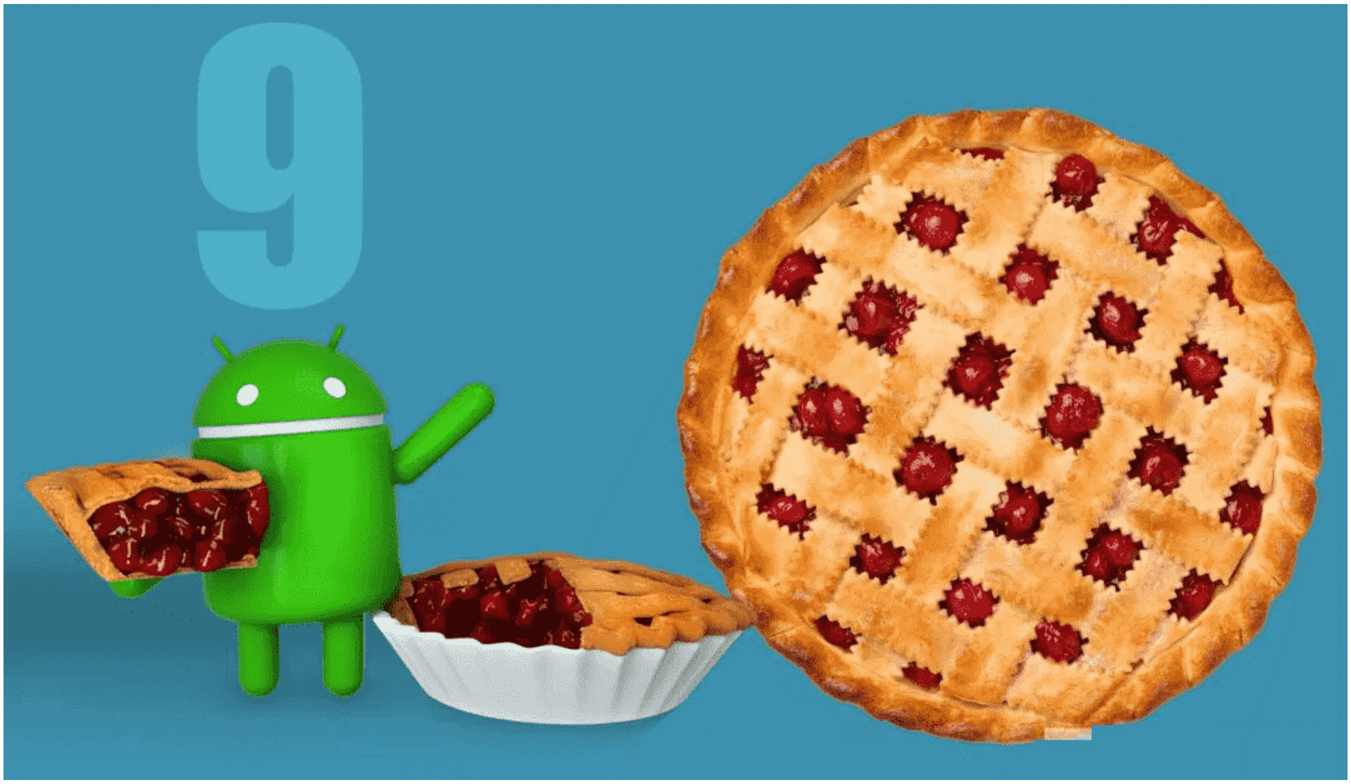

Figure 7. Android Pie image

6 Working with the i.MX 7Dual SABRE-SD Board

6.1 Board hardware

The figure below shows the different components of the i.MX 7Dual SABRE-SD board.

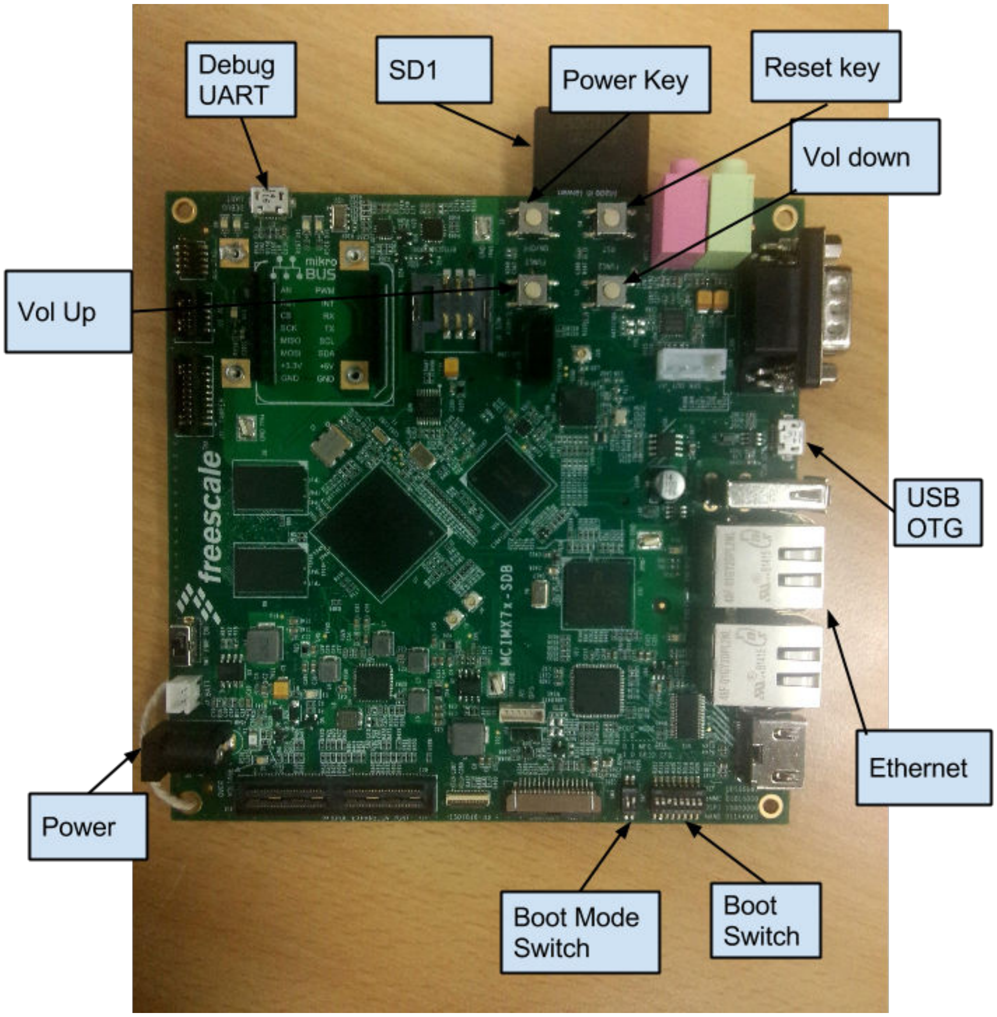

Figure 8. i.MX 7Dual SABRE-SD board

6.2 Board images

The table below describes the location in the board partitions of the software images in android_p9.0.0_2.2.0-ga_image_7dsabresd.tar.gz on board partitions.

Table 7. Board images

Image name	Download target
/u-boot-imx7ulp.imx	1 KB offset of SD card.
/u-boot-imx7d-sabresd-uuu.imx	Bootloader used by UUU for i.MX 7ULP SABRE-SD. It is not flashed to the SD card.
/partition-table.img	Program to the first 0.5 KB wih MBR and the last 16.5 KB with GPT header and entries of the SD card. GPT table for 8 GB SD card.
/partition-table-14GB.img	Program to the first 0.5 KB wih MBR and the last 16.5 KB with GPT header and entries of the SD card. GPT table for 16 GB SD card.
/partition-table-28GB.img	Program to the first 0.5 KB wih MBR and the last 16.5 KB with GPT header and entries of the SD card. GPT table for 32 GB SD card.

Table continues on the next page...

Table 7. Board images (continued)

/dtbo-imx7d.img	SD 1st partition.
/boot-imx7d.img	SD 2nd partition.
/recovery-imx7d.img	SD 3rd partition.
/system.img	SD 4th partition.
/vendor.img	SD 10th partition, holding platform binaries.
vbmeta-imx7d.img	SD 13th partition

6.3 Flashing board images

The board image files can be flashed into the target board by using Universal Update Utility (UUU). The UUU source code and binary file are released on github: uuu release page on github.

To achieve more flexibility, the following two script files are provided to invoke UUU to automatically flash all Android images.

- uuu_imx_android_flash.sh for Linux OS
- uuu_imx_android_flash.bat for Windows OS

For this release, these two scripts are validated on UUU 1.2.135 version. Download the corresponding version from github:

- For Linux OS, download the file named "uuu".
- For Windows OS, download the files named "uuu.exe".

Because the two files direclty invokes UUU, make sure that UUU is in a path contained by the system environment variable of "PATH".

Perform the following steps to download the board images:

1. Download the UUU binary file from github as described above. Install UUU into a directory contained by the system environment variable of "PATH".
2. Make the board enter serial download mode.

   Change the board's sw3 (boot mode) to 01 (from 1-2 bit) to enter download mode.

3. Power on the board. Use the USB cable on the board OTG port to connect your PC with the board.

   NOTE

   - There are two USB micro ports on the i.MX 7Dual SDB board: USB-to-UART, USB OTG.
   - The USB-to-UART port is known as debug UART, and the USB OTG port is known as USB in the hardware image above.
   - The debug UART port can be used to watch the logs of the hardware boot processing.
   - The SD card should be plugged in after the board is powered on.

4. Decompress release_package/android_p9.0.0_2.2.0-ga_image_7dsabresd.tar.gz, which contains the image files and uuu_imx_android_flash tool.
5. Execute the uuu_imx_android_flash tool to flash images.

uuu_imx_android_flash tool can be executed with options to get help information and specify the images to be flashed. For i.MX 7Dual SABRE-SD, related options are described in the following table.

Table 8. Options for the uuu_imx_android_flash tool

Option	Description
-h	Displays the help information for this tool.
-f soc_name	Specifes the SoC information. For i.MX 7Dual SABRE-SD, it is "imx7d". This option is **mandatory**.
-c card_size	Specifies which partition table image file to flash. For i.MX 7Dual SABRE-SD, it can be followed with "14" or "28". If this option is not used, default "partition-table.img" is flashed.
-e	Erases the user data after images are flashed.
-D directory	Specifies the directory where the images to be flashed. If this option is not used, images in the current working directory is flashed.
-daemon	Run UUU in daemon mode. This option is used to flash multiple boards of the same type.
-i	If the script is executed with this option, no images is flashed. The script just loads U-Boot to RAM and executes to fastboot mode. This option is used for development.

- On the Linux OS, open the shell terminal. For example, you can execute a command as follows:

```
> sudo ./uuu_imx_android_flash.sh -f imx7d -e
```

- On the Windows OS, run `cmd/powershell` in administrator mode, the corresponding command is as follows:

```
> .\uuu_imx_android_flash.bat -f imx7d -e
```

With the command above executed, the default images are flashed into the SD card for i.MX 7Dual SABRE-SD.

NOTE

- i.MX 7Dual SABRE-SD only supports SD card as the target device, so it does not need to specify the target device when executing the uuu_imx_android_flash tool.
- If your SD card is 8 GB, it does not need to use the -c option.
- If your SD card is 16 GB, execute the tool with "-c 14".
- If your SD card is 32 GB, execute the tool with "-c 28".
- uuu_imx_android_flash.bat generates the temporary file under the current working directory. Make sure you have the write permission under the current working directory.
- If uuu_imx_android_flash.bat is used to flash images on a remote server through samba, you need to map the remote resource to the local environment first. Take the following command as an example:

```
> net use z: \\10.193.108.179\daily_images
```

"z" in the command represents an available drive letter, which can be other available drive letter.

6. Wait for the uuu_imx_android_flash execution to be completed. If there is no error, the command window prompts that images are already flashed.
7. Power down the board.
8. Change the boot device to SD card:

- Change SW3 (boot mode) to 10 (from 1-2 bit).
- Change SW2 to switch the board back to SD4 boot mode 00100000 (SD).

6.4 Booting with single display: LVDS display or HDMI display

In the U-Boot prompt, set the U-Boot environment variables as shown below:

```
U-Boot > setenv bootargs console=ttymxc0,115200 init=/init androidboot.console=ttymxc0
consoleblank=0 androidboot.hardware=freescale cma=320M loop.max_part=7
U-Boot > saveenv
```

With the settings above, the Android platform does not start the shell console. To disable selinux, append "androidboot.selinux=permissive" to the U-Boot's bootargs. Boot environment variables are as follows:

```
U-Boot > setenv append_bootargs androidboot.selinux=permissive
U-Boot > saveenv
```

6.5 Board reboot

After you have completed download and setup, reboot the board and wait for the Android platform to boot up.

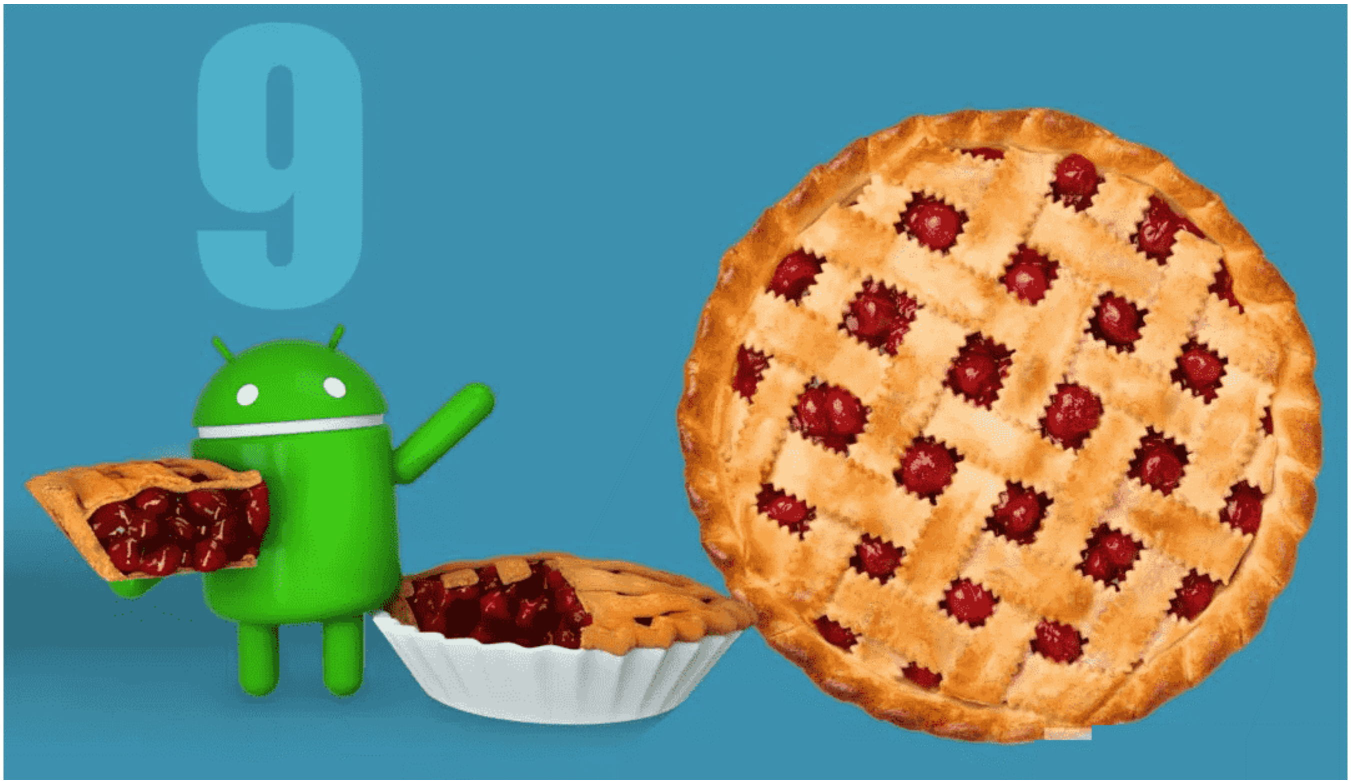

Figure 9. Android Pie image

7 Working with the i.MX 7ULP EVKB-SD Board

7.1 Board hardware

The figure below shows the different components of the i.MX 7ULP EVKB REV. A SD board.

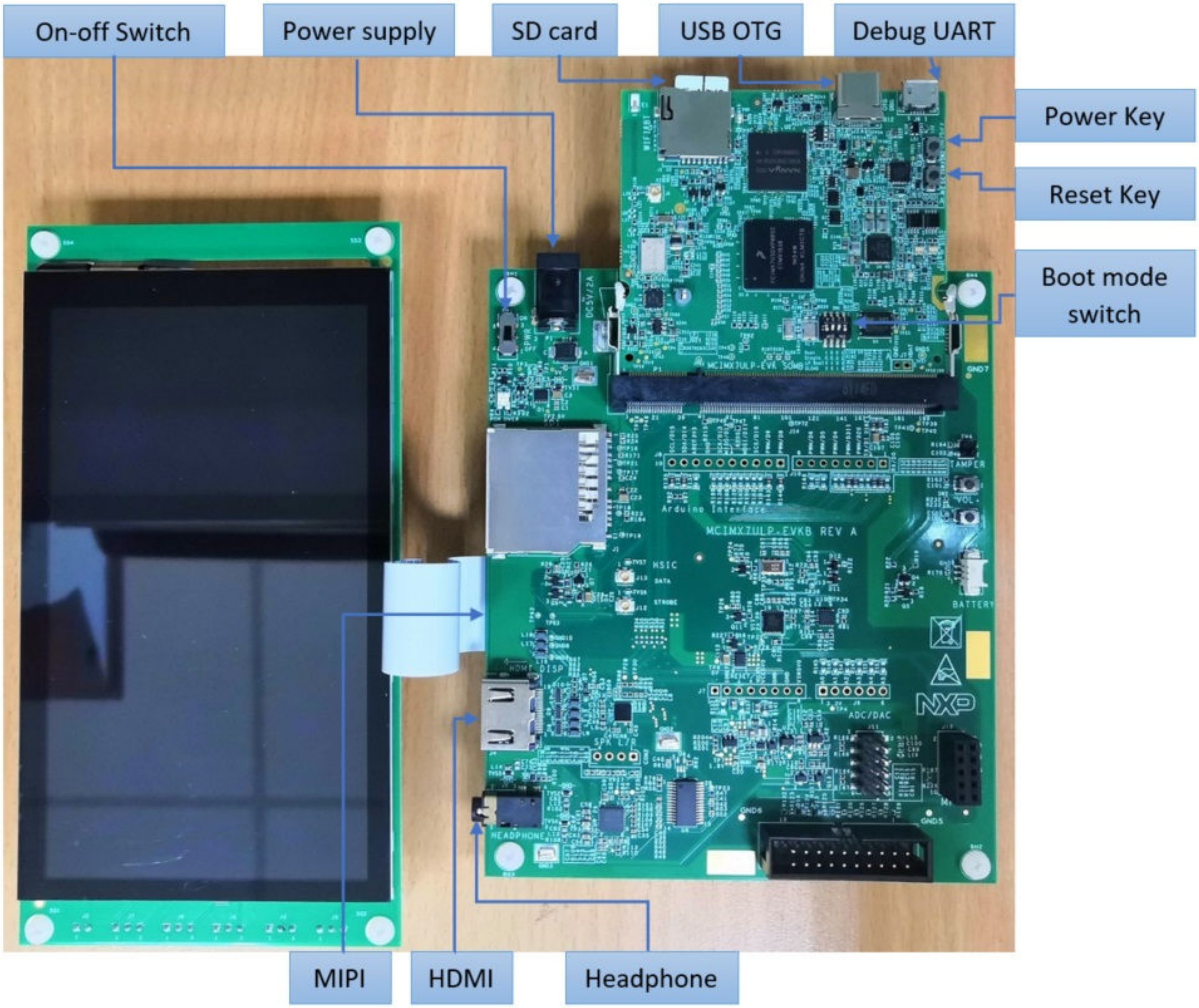

Figure 10. i.MX 7ULP EVKB Rev. A SD board

NOTE

The i.MX 7ULP EVKB Rev. A board needs the following hardware rework to support the MIPI display:

- Remove the resistors: R41 (0 ohm), R43-R46 (0 ohm), R51 (0 ohm).
- Populate the resistors: R30-R35 (0 ohm).
- After reworking, the HDMI display is not supported any more.

7.2 Board images

The table below describes the location in the board partitions of the software images in android_p9.0.0_2.2.0-ga_image_7ulpevk.tar.gz on board partitions.

Table 9. Board images

Image name	Download target
/u-boot-imx7ulp.imx	1 KB offset of SD card.
/u-boot-imx7ulp-evk-uuu.imx	Bootloader used by UUU for i.MX 7ULP EVKB-SD. It is not flashed to the SD card.
/imx7ulp_m4_demo.img	0 KB offset of NOR flash.
/partition-table.img	Program to the first 0.5 KB wih MBR and the last 16.5 KB with GPT header and entries of the SD card. GPT table for 8 GB SD card.
/partition-table-14GB.img	Program to the first 0.5 KB wih MBR and the last 16.5 KB with GPT header and entries of the SD card. GPT table for 16 GB SD card.
/partition-table-28GB.img	Program to the first 0.5 KB wih MBR and the last 16.5 KB with GPT header and entries of the SD card. GPT table for 32 GB SD card.
/dtbo-imx7ulp.img	SD 1st partition.
/dtbo-imx7ulp-mipi.img	SD 1st partition.
/boot.img	SD 2nd partition.
/recovery-imx7ulp.img	SD 3rd partition.
/system.img	SD 4th partition.
/vendor.img	SD 10th partition, holding platform binaries.
vbmeta-imx7ulp.img	SD 13th partition, holding verification data.
vbmeta-imx7ulp-mipi.img	SD 13th partition, holding verification data.

7.3 Flashing board images

The board image files can be flashed into the target board by using Universal Update Utility (UUU). The UUU source code and binary file are released on github: uuu release page on github.

To achieve more flexibility, the following two script files are provided to invoke UUU to automatically flash all Android images.

- uuu_imx_android_flash.sh for Linux OS
- uuu_imx_android_flash.bat for Windows OS

For this release, these two scripts are validated on UUU 1.2.135 version. Download the corresponding version from github:

- For Linux OS, download the file named "uuu".
- For Windows OS, download the files named "uuu.exe".

Because the two files direclty invokes UUU, make sure that UUU is in a path contained by the system environment variable of "PATH".

Perform the following steps to download the board images:

1. Download the UUU binary file from github as described above. Install UUU into a directory contained by the system environment variable of "PATH".
2. Make the board enter serial download mode.

   Change the board's sw1 (boot mode) to 01xx (from 1-4 bit) to enter download mode.

3. Power on the board. Use the USB cable on the board OTG port to connect your PC with the board.

NOTE

- There are two USB micro ports on the i.MX 7ULP EVKB Rev. A board: USB-to-UART, USB OTG.
- The USB-to-UART port is known as debug UART, and the USB OTG port is known as USB in the hardware image above.
- The debug UART port can be used to watch the logs of the hardware boot processing.
- There are two UART consoles on the i.MX 7ULP EVKB Rev. A board. One is for Cortex-M4 core and the other is for Cortex-A7 core.
- To check Android boot information, choose the Cortex-A7 console.
- The SD card should be plugged in after the board is powered on.

4. Decompress release_package/android_p9.0.0_2.2.0-ga_image_7ulpevk.tar.gz, which contains the image files and uuu_imx_android_flash tool.
5. Execute the uuu_imx_android_flash tool to flash images.

uuu_imx_android_flash tool can be executed with options to get help information and specify the images to be flashed. For i.MX 7ULP EVKB-SD, related options are described in the following table.

Table 10. Options for the uuu_imx_android_flash tool

Option	Description
-h	Displays the help information for this tool.
-f soc_name	Specifes the SoC information. For i.MX 7ULP EVKB-SD, it is "imx7ulp". This option is **mandatory**.
-c card_size	Specifies which partition table image file to flash. For i.MX 7ULP EVKB-SD, it can be followed with "14" or "28". If this option is not used, default "partition-table.img" is flashed.
-m	Flashes the Cortex-M4 image. If this option is not used, the Cortex-M4 image is not flashed.
-d dev	Specifies some images with "dev" in their names. For i.MX 7ULP EVKB-SD, it can be "mipi". If this option is not used, the default dtbo, recovery, and vbmeta images are flashed.
-e	Erases the user data after images are flashed.
-D directory	Specifies the directory where the images to be flashed. If this option is not used, images in the current working directory is flashed.
-daemon	Run UUU in daemon mode. This option is used to flash multiple boards of the same type.
-i	If the script is executed with this option, no images is flashed. The script just loads U-Boot to RAM and executes to fastboot mode. This option is used for development.

- On the Linux OS, open the shell terminal. For example, you can execute a command as follows:

```
> sudo ./uuu_imx_android_flash.sh -f imx7ulp -m -e
```

- On the Windows OS, run `cmd/powershell` in administrator mode, the corresponding command is as follows:

```
> .\uuu_imx_android_flash.bat -f imx7ulp -m -e
```

With the command above executed, the default images are flashed into the SD card for i.MX 7ULP EVKB-SD. The Cortex-M4 image is flashed to NOR flash.

NOTE

- i.MX 7ULP EVKB-SD only supports SD card as the target device, so it does not need to specify the target device when executing the uuu_imx_android_flash tool.
- If your SD card is 8 GB or the onboard eMMC is used, it does not need to use the -c option.
- If your SD card is 16 GB, execute the tool with "-c 14".
- If your SD card is 32 GB, execute the tool with "-c 28".
- uuu_imx_android_flash.bat generates the temporary file under the current working directory. Make sure you have the write permission under the current working directory.
- If uuu_imx_android_flash.bat is used to flash images on a remote server through samba, you need to map the remote resource to the local environment first. Take the following command as an example:

  ```
  > net use z: \\10.193.108.179\daily_images
  ```

  "z" in the command represents an available drive letter, which can be other available drive letter.

6. Wait for the uuu_imx_android_flash execution to be completed. If there is no error, the command window prompts that images are already flashed.
7. Power down the board.
8. Change the boot device to SD card:

   Change SW1 (boot mode) to 1001 (from 1-4 bit) to boot from the SD card.

7.4 Booting with single display: HDMI or MIPI display

In the U-Boot prompt, set the U-Boot environment variables as shown below:

```
U-Boot > setenv bootargs console=ttyLP0,115200 init=/init androidboot.console=ttyLP0
consoleblank=0 androidboot.hardware=freescale vmalloc=128M cma=320M loop.max_part=7
U-Boot > saveenv
```

With the settings above, the Android platform does not start the shell console. To disable selinux, append "androidboot.selinux=permissive" to the U-Boot's bootargs. Boot environment variables are as follows:

```
U-Boot > setenv append_bootargs androidboot.selinux=permissive
U-Boot > saveenv
```

7.5 Board reboot

After you have completed download and setup, reboot the board and wait for the Android platform to boot up.

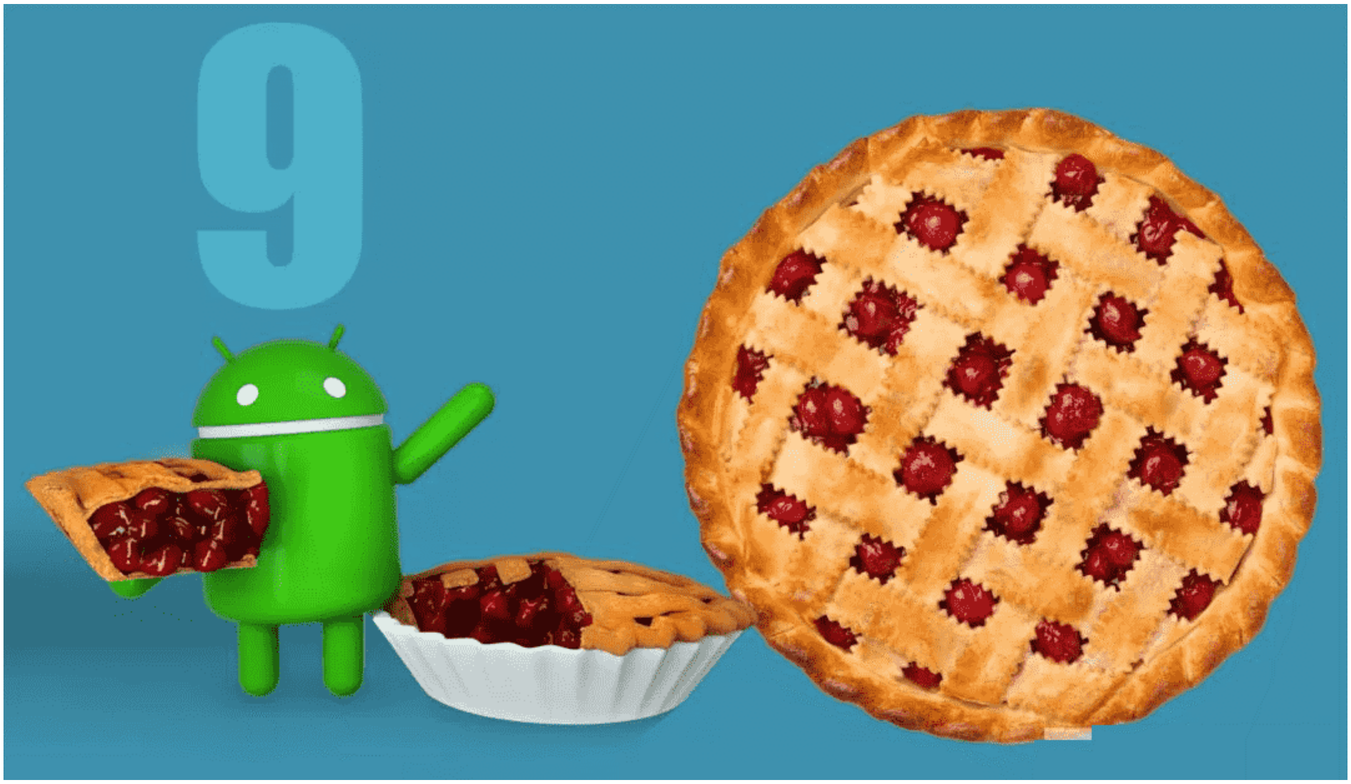

Figure 11. Android Pie image

8 Revision History

Table 11. Revision history

Revision number	Date	Substantive changes
O8.0.0_1.0.0	02/2018	Initial release
O8.0.0_1.0.0	10/2018	Updated the Graphic - HW 3D acceleration feature for the i.MX 7Dual to N/A.
P9.0.0_2.2.0-ga	07/2019	i.MX 6 and i.MX 7 GA release.

How to Reach Us:

Home Page:
nxp.com

Web Support:
nxp.com/support

Information in this document is provided solely to enable system and software implementers to use NXP products. There are no express or implied copyright licenses granted hereunder to design or fabricate any integrated circuits based on the information in this document. NXP reserves the right to make changes without further notice to any products herein.

NXP makes no warranty, representation, or guarantee regarding the suitability of its products for any particular purpose, nor does NXP assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. “Typical” parameters that may be provided in NXP data sheets and/or specifications can and do vary in different applications, and actual performance may vary over time. All operating parameters, including “typicals,” must be validated for each customer application by customer’s technical experts. NXP does not convey any license under its patent rights nor the rights of others. NXP sells products pursuant to standard terms and conditions of sale, which can be found at the following address: nxp.com/SalesTermsandConditions.

While NXP has implemented advanced security features, all products may be subject to unidentified vulnerabilities. Customers are responsible for the design and operation of their applications and products to reduce the effect of these vulnerabilities on customer's applications and products, and NXP accepts no liability for any vulnerability that is discovered. Customers should implement appropriate design and operating safeguards to minimize the risks associated with their applications and products.

Document Number AQSUG
Revision P9.0.0_2.2.0-ga, 07/2019