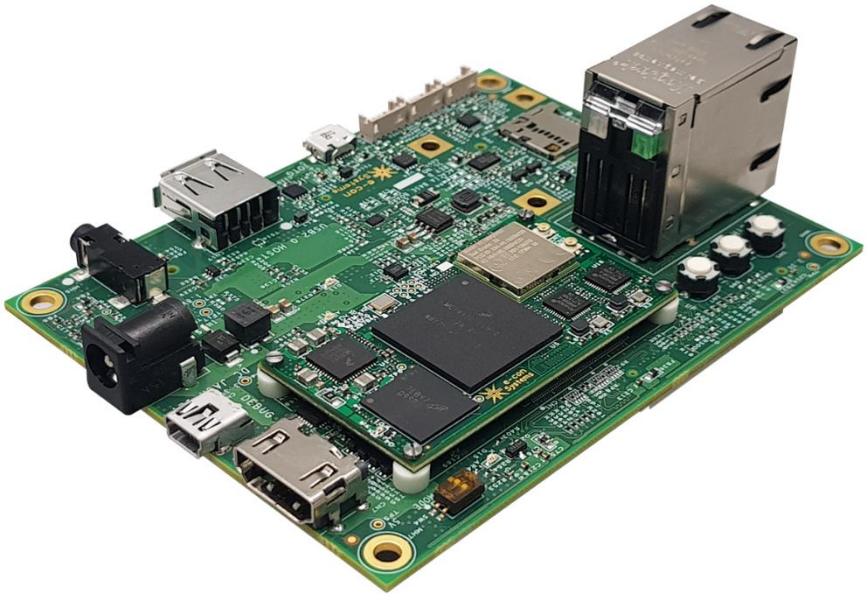
Acacia

Linux User Manual



Version 1.0

e-con Systems

5/7/2018

e-con Systems

Your Product Development Partner

Disclaimer

e-con Systems reserves the right to edit/modify this document without any prior intimation of whatsoever.

Contents

Introduction

This document explains you about Acacia board BSP in detail and information to customize BSP. This document can serve as the software user manual for developers interested in developing software on top of the eSOMiMX7 SOM with any target board. This user manual also explains software information about eSOMiMX7, it's boot process, peripheral details and how-to details for testing various interfaces or peripherals in the eSOMiMX7 Acacia development board.

Note: Refer to the *e-con_Acacia_Getting_Started_Manual.pdf*, to know some basic details about eSOMiMX7 SOM and Acacia kit.

The commands and output messages in this user manual are represented by different colors, the significance of the colors are listed in the following table.

Table 1: Notation of Colours

Color	Notation
Red	Commands running in Acacia
Green	Output messages in Acacia
Blue	Commands in Linux Development System
Orange	Output messages in Linux Development System

All U-Boot commands of Acacia are preceded by bootloader prompt as given below.

```
U-Boot >
```

All Linux commands of Acacia are preceded by shell prompt as given below.

```
root@esomimx7X-Ygb:~#
```

Where,

X is s (Solo SOM) or d (Dual SOM)

Ygb is 1 GB or 2 GB depending on the SOM type.

For example, prompt will be root@esomimx7d-1gb for Dual 1 GB configuration.

Note: For simplicity, the prompt is mentioned as root@esomimx7x throughout this document.

Block Diagram of Acacia

Acacia is the baseboard designed for evaluating the eSOMiMX7 SOM. The Acacia board is designed to exposes the interfaces supported by the iMX7 SoC. The list of important interfaces of Acacia are as follows:

- ADC
- Audio (Audio in and Audio Out)
- CAN
- HDMI (using Parallel Interface)
- RS485
- Micro SD
- Mini PCIe
- MIPI CSI
- MIPI DSI
- RS232 (with RTS and CTS)
- I2C
- USB OTG
- GigaBit Ethernet x 2 Nos
- GPIO Header
- Tamper Pins

Apart from the above interfaces, Acacia board also features an IMU including Accelerometer and Gyroscope device, SIM slot, coin cell for RTC, HDMI transceiver and USB-UART converter for the debug ports.

The following diagram shows the block diagram of the eSOMiMX7 baseboard.

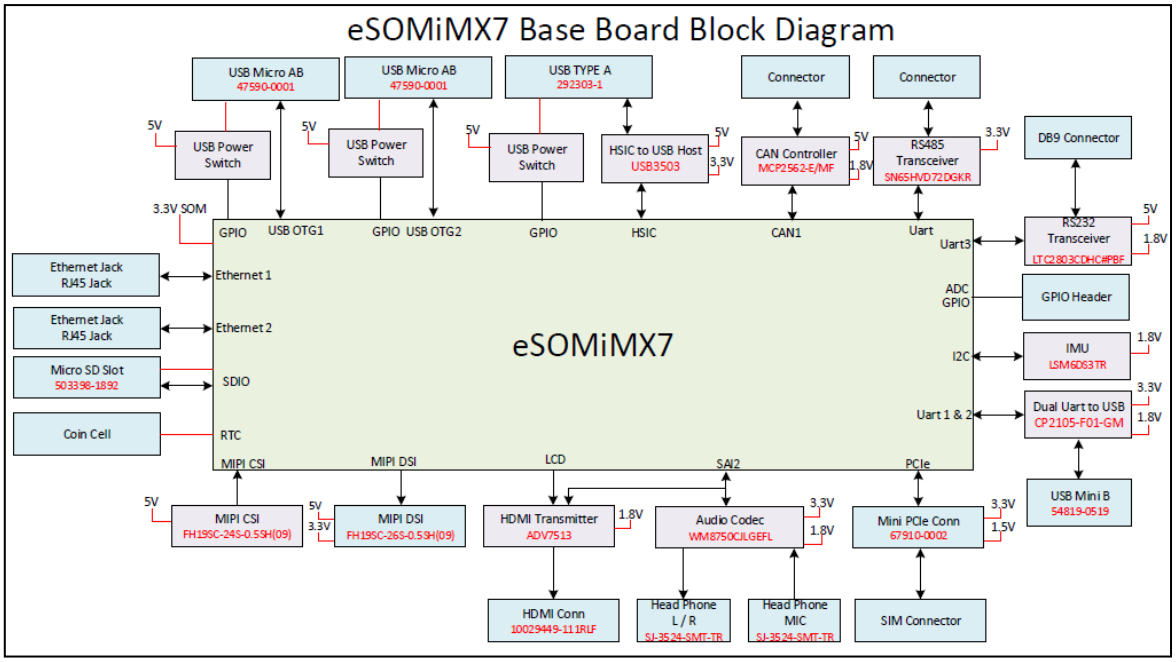


Figure 1: Block Diagram of eSOMiMX7 Baseboard

Boot Modes in eSOMiMX7 - Acacia

The boot device of eSOMiMX7 is selected based on the configuration of the boot mode switch (SW4) present in Acacia board. The following table lists the boot options with respect to boot mode switch.

Table 2: Boot Modes in eSOMiMX7

BOOT MODE SW4 1 and 2	Boot Type	Description
ON ON	Boot from Fuses	Boots based on the fused values
ON OFF	Serial download mode	USB download mode, iMX7 waits to boot from USB download
OFF ON	Internal boot	Boots from SD if card is present, otherwise waits for USB download
OFF OFF	Reserved	Reserved

Boot from Fuses Mode

By default, eSOMiMX7 is fused to boot from eMMC for Dual SOM and NAND for Solo SOM. So, when boot mode switch is configured to boot from fuses U-Boot can be loaded from eMMC in Dual SOM or NAND in Solo SOM.

Serial Download Mode

The serial downloader supports to download and program boot Image to the chip over the USB serial connection. MfgTools are used to program the boot binaries such as U-Boot, kernel, device tree blob (DTB) and root file system (rootfs) to eMMC or NAND.

Internal Boot Mode

This mode is similar to boot from fuses mode except that in this mode eFUSE settings are overridden using the GPIO boot select pins. You can use this mode to boot from SD card instead of eMMC.

Note: Please refer to the *Creating a Bootable SD Card using SD Image section* in *e-con_Acacia_Linux_Yocto_Prebuilt_Binaries_UserManual.pdf*, to know about creating the bootable SD card.

iMX7 SoC has an inbuilt BootROM which executes on power on reset (POR), this BootROM selects boot device based on switch settings and reads the Image Vector Table (IVT). The IVT contains configuration values to perform successful boot. BootROM copies bootloader to RAM. The bootloader does the remaining initializations and proceeds to boot the kernel image. Bootloader has driver support for most of the devices and provides a shell prompt with minimum commands. The main role of U-Boot is to initialize all supported devices, setup environmental variables and the bootloader will load the kernel and initial root file system image into the memory and then start the kernel, passing in the memory address of the image.

Before passing control to Linux kernel image, U-Boot verifies it by using the image header information. When control is handed over to kernel image, the compressed kernel image is decompressed and then kernel execution starts. The kernel will do mandatory initialization such as enabling MMU, probing of in-built drivers. Then the initial root file system image archive is unpacked by the kernel into a special instance of a tmpfs that becomes the initial root file system. The initial root file system will check for actual root file system and mount it. The location information of actual root file system is passed by boot argument, variable mmcroot in bootargs contains the root file system path.

The root file system consists of binaries, libraries, device files and configuration files in a structured manner. By default, the first user space process /sbin/init gets executed with information provided in /etc/inittab file. Based on the boot script files and configuration files, further loading of the module drivers and start of services takes place. At the final stage of booting, a shell prompt is launched in the debug console device.

The top and bottom view of the Acacia board are shown in the following figures.

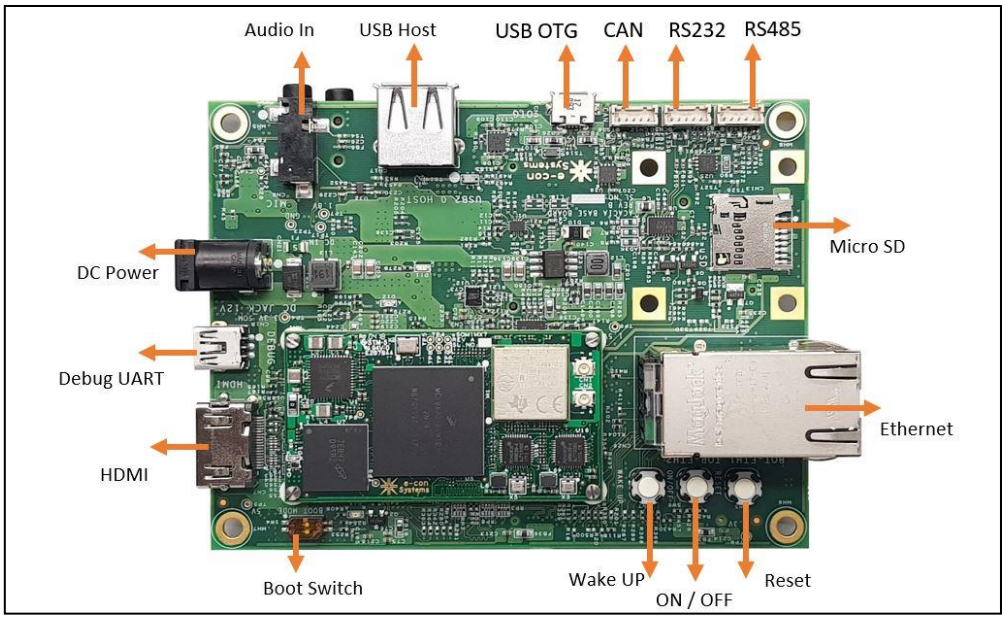


Figure 2: Acacia Board Top View

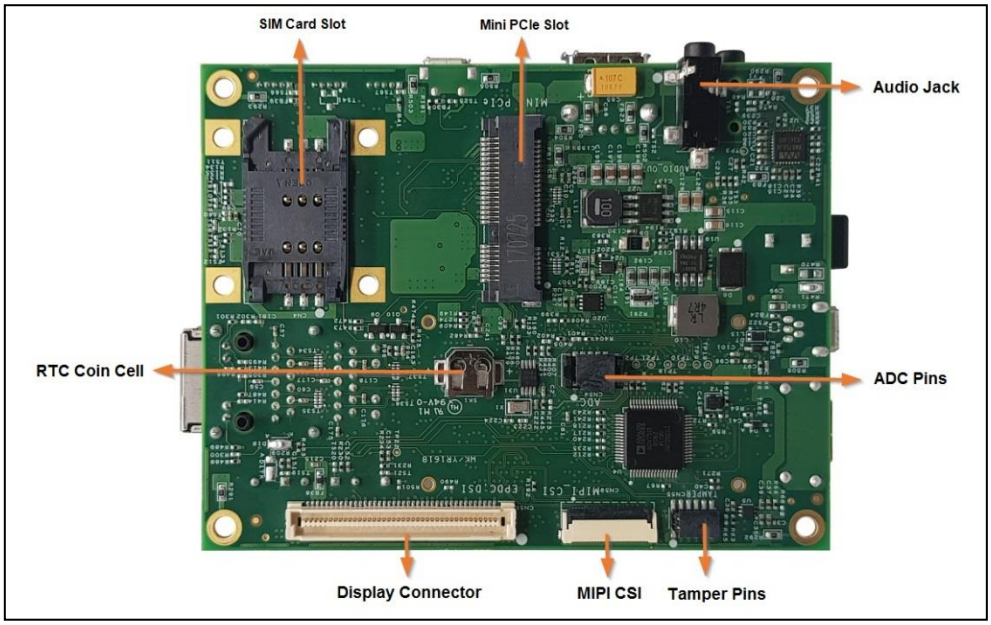


Figure 3: Acacia Board Bottom View

Serial Console Connection Details

The iMX7 has 7 UART interfaces, of which UART1 and UART3 are used as Debug UART for M4 and A7 core respectively. Both the UARTs are exposed as single USB composite device using a serial to USB converter chip in Acacia board. To receive

debug console message, you need to connect the USB Mini B to A cable between the Acacia board and the development system. The USB cable is supplied with Acacia development kit.

Performing the serial console connection involves the following steps:

1. Loading Serial Driver
2. Configuring Minicom
3. Receiving Console Message from Acacia

Loading Serial Driver

The driver for the USB console is inbuilt in the Linux systems. The driver module will be loaded automatically when the Acacia kit is connected to development system and powered **ON**.

The dmesg output log from the development system while connecting the USB debug port (CN18) of Acacia board appears as shown below.

```
$ sudo dmesg | tail
[8403779.561587] usb 3-1.1.3: New USB device found, idVendor=10c4, idProduct=ea70
[8403779.561592] usb 3-1.1.3: New USB device strings: Mfr=1, Product=2, SerialNumber=5
[8403779.561595] usb 3-1.1.3: Product: CP2105 Dual USB to UART Bridge Controller
[8403779.561597] usb 3-1.1.3: Manufacturer: Silicon Labs
[8403779.561599] usb 3-1.1.3: SerialNumber: 008A7A36
[8403779.562204] cp210x 3-1.1.3:1.0: cp210x converter detected
[8403779.562404] usb 3-1.1.3: cp210x converter now attached to ttyUSB0
[8403779.562708] cp210x 3-1.1.3:1.1: cp210x converter detected
[8403779.562819] usb 3-1.1.3: cp210x converter now attached to ttyUSB1
[8403779.575894] cp210x ttyUSB1: failed to set baud rate to 300
```

Run the following command to lists the driver modules loaded in the development system for USB to serial conversion support.

```
$ lsmod | grep cp210x
```

The output appears as shown below.

```
cp210x         24576  2
usbserial      49152  5 cp210x
```

Configuring Minicom

After confirming the driver load is successful, you need to use a serial terminal application to capture the debug console message. It is recommended to use minicom terminal application.

Run the following command to install minicom.

```
$ sudo apt-get install minicom
```

Run the following command to launch minicom.

```
$ sudo minicom -s
```

Note: -s will prevent minicom from exiting abruptly if it encounters error upon invocation.

Configuration menu appears in the minicom terminal as shown below:

```
             +-----[configuration]------+
            | Filenames and paths       |
            | File transfer protocols   |
            | Serial port setup         |
            | Modem and dialling         |
            | Screen and keyboard       |
            | Save setup as dfl         |
            | Save setup as..           |
            | Exit                      |
            | Exit from Minicom         |
            +---------------------
```

Navigate to **Serial port setup** option and press **Enter** key to view the serial port setup menu.

You need to configure the serial port settings according to Acacia's debug port configuration.

Note: The typical setup for the Acacia is shown below.

```
   +-----------------------------------------------------
------------------+
   | A -    Serial Device      : /dev/ttyUSB1           |
   | B - Lockfile Location     : /var/lock              |
   | C -   Callin Program      :                        |
   | D -  Callout Program      :                        |
   | E -    Bps/Par/Bits       : 115200 8N1             |
   | F - Hardware Flow Control : No                     |
   | G - Software Flow Control : No                     |
```

```
|                                                    |
|    Change which setting?                           |
+----------------------------------------------------+
```

Enter **A** to change the serial device. You can set it to **/dev/ttyUSB1**, so it accesses your A7 core's debug UART port though this can vary from PC to PC.

You can also change any other settings appropriate for your board such as disabling hardware and software flow control.

Connect the 12 V power cord supplied with Acacia kit and power **ON** the board.

Note: /dev/ttyUSB0 device node is for M4 debug terminal and **/dev/ttyUSB1** device node is for A7 debug terminal.

Receiving Console Message from Acacia

Power **ON** the Acacia board. The bootloader starts with debug print. The bootloader loads the kernel after a specified time interval.

Pressing any key within the boot delay displays the U-Boot prompt.

The U-Boot bootloader log for Solo and Dual SOM are shown below.

Solo SOM

```
U-Boot 2017.03v1.0_rc1 (Feb 09 2018 - 20:21:36 +0530)

CPU:   Freescale i.MX7S rev1.2 at 792MHz
CPU:   Extended Commercial temperature grade (-20C to
105C) at 34C
Reset cause: POR
Model: ESOMIMX7 SOLO BOARD
DRAM:  1 GiB
Board: ESOMIMX7 SOLO
eSOMiMX7 Release version: v1.0_rc1
PMIC: PFUZE3000 DEV_ID=0x30 REV_ID=0x11
NAND:  2048 MiB
MMC:   FSL_SDHC: 0
Display: TFT43AB (480x272)
Video: 480x272x24
Warning : bmp logo size is more than screen size
Unable to display logo...
In:    serial
Out:   serial
```

```
Err:   serial
Net:   No ethernet found.
Normal Boot
Hit any key to stop autoboot:  0
U-Boot >
```

Dual SOM

```
U-Boot 2017.03v1.0_rc1 (Feb 09 2018 - 20:21:36 +0530)

CPU:   Freescale i.MX7D rev1.2 at 792MHz
CPU:   Extended Commercial temperature grade (-20C to
105C) at 34C
Reset cause: POR
Model: ESOMIMX7 DUAL BOARD
DRAM:  1 GiB
Board: ESOMIMX7 DUAL
eSOMiMX7 Release version: v1.0_rc1
PMIC: PFUZE3000 DEV_ID=0x30 REV_ID=0x11
MMC: FSL_SDHC: 0, FSL_SDHC: 1
Display: TFT43AB (480x272)
Video: 480x272x24
Warning : bmp logo size is more than screen size
Unable to display logo...
In: serial
Out: serial
Err: serial
Net:   No ethernet found.
Normal Boot
Hit any key to stop autoboot:  0
U-Boot >
```

Run the following command to launch kernel from U-Boot prompt.

```
U-Boot > run bootcmd
```

Refer to the *Appendix*

section, for Solo SOM and Dual SOM complete boot log.

Source Details

This section explains about the various sources of eSOMiMX7 in detail. The sources are as follows:

- Toolchain
- Bootloader
- Kernel and Device Tree
- Init RamFS
- Rootfs

The following table lists the source package and toolchain used in the eSOMiMX7 Acacia BSP.

Table 3: Source Details

Source	Package	Version
Toolchain	arm-none-linux-gnueabi-	GCC Version 6.2.0
Bootloader	U-Boot	2017.03
Kernel	Linux	4.9.11
Ramdisk Image	Busy Box	V1.24.1
Rootfs	Yocto	Morty L4.9.11-1.0.0

Toolchain

A toolchain is a set of distinct software development tools that are linked together by specific stages such as GCC, binutils and glibc. The toolchain used for embedded development is a cross toolchain, also known as cross compiler. The cross toolchain runs on a host system of a specific architecture but produce executable to run on a different architecture for example, ARM. This is called cross compilation and is the typical way of building embedded software.

The ARM toolchain is used to build the bootloader, kernel and root file system images.

For eSOMiMX7, GCC version 6.2.0 cross compiler is used for building all images. This toolchain package can be downloaded from https://releases.linaro.org/components/toolchain/binaries/6.2-2016.11/ link.

Bootloader

Bootloader is a piece of code that runs before any operating system is running. Bootloader is used to boot other operating systems. Usually, each operating system has a set of bootloaders specific for it. Bootloaders usually contain several ways to boot the OS kernel and also contain commands for debugging and/or modifying the

kernel environment. Since it is usually the first software to run after powerup or reset, it is highly processor and board specific.

For eSOMiMX7, U-Boot 2017.03 is used. On building the bootloader source, the image u-boot.bin is generated which is appended to the IVT configuration file to generate u-boot.imx image. U-boot is the primary loader which initializes clocks, UARTs, SD and eMMC or NAND memory, initializes all supporting devices, sets environmental variables and passes control to Linux kernel image. U-Boot has a set of builtin commands for booting the system, managing memory and updating an embedded system's firmware. Custom builtin commands can be created by modifying U-Boot source code.

Run the following command to get the complete list of U-Boot commands along with the brief description.

```
U-Boot > help
```

or

```
U-Boot > ?
```

U-Boot Environmental Variables

U-Boot uses environment variables to tailor its operation. The environment variables are used to configure the system. The environment variables configure settings such as display, default boot command, and so on. The U-Boot environment is kept on persistent (eMMC, NAND or SD card) storage and copied to RAM when U-Boot starts. The environment is protected by a CRC32 checksum.

The factory default environment variables and their values are stored in the U-Boot binary image itself.

Run the following U-Boot command to list the environment variables.

```
U-Boot > printenv
```

Run the following command to add an environment variable.

```
U-Boot > setenv <variable_name> <value>
```

The variables can also be edited by setenv command by specifying the new value.

Kernel and Device Tree

The kernel is a program that constitutes the central core of a computer operating system. It has complete control over everything that occurs in the system.

The kernel is the first part of the operating system to load into memory during booting, and it remains there for the entire duration of the computer session because its services are required continuously. Thus, it is important for it to be as

small as possible while still providing all the essential services needed by the other parts of the operating system and by the various application programs.

The kernel performs its tasks, such as executing processes and handling interrupts, in kernel space. Writing text in a text editor or running programs in a graphical user interface (GUI), is done in user space. This separation is made in order to prevent user data and kernel data from interfering with each other and thereby diminishing performance or causing the system to become unstable and crashing.

Device tree file supplies board specific information to the kernel.

Note: Refer to the *Device Tree*

section, to know more about the device tree concept.

The zImage is the kernel image and imx7s-acacia.dtb or imx7d-acacia.dtb is the DTB for esomimx7 Solo or Dual respectively. All these files are generated during the kernel build.

The zImage with the help of device tree file configures pins, initialize other peripherals such as display, touch, Wi-Fi, Bluetooth, Audio and so on. The zImage also mounts the root file system from the device specified in bootargs. After that, the kernel image takes control of the system.

Init RamFS

Init RamFS is the initial root file system, known as tiny image. The bootloader will pass the load address of init ramfs along with the kernel image through the boot command. Once kernel loaded it will unpack init ramfs image into a special instance of a tmpfs that becomes the initial root file system. The initial root file system mounts the actual root filesystem. The location information of actual root file system is passed by boot argument, variable mmcroot in bootargs contains the root filesystem path. The customized Busy Box V1.24.1 is used as initial root file system.

Rootfs

The rootfs is a mandatory component in booting a Linux system. Detecting and mounting rootfs is the final step of Linux boot sequence. Root filesystem is needed to access the devices present in board.

In Acacia, the rootfs can be stored in the following devices:

- eMMC or NAND
- SD Card
- USB Flash Disk

Storing rootfs on a storage device involves the following steps:

1. Creating partition in storage device
2. Formatting that partition in supported Linux file system format
3. Mounting formatted partition at a mount point
4. Copying created rootfs structure in that mounted location
5. Unmounting that partition of storage device

The details about location of rootfs are passed to kernel using the bootargs variable. Following is the boot argument related to root filesystem.

mmcroot=<devicename_where_rootfs_is_present>

For example, if rootfs is present in Partition 2 of SD then boot argument is mmcroot=/dev/mmcblk1p2.

eSOMiMX7 Acacia Interfaces and Peripherals

This section describes the eSOMiMX7 Acacia interfaces, peripherals and Acacia board specific information. NXP provides Linux reference manual for iMX7 SoC which provides all the information about iMX7 Linux BSP. You can download the document package from the https://www.nxp.com/webapp/Download?colCode=L4.9.11_1.0.0_LINUX_DOCS link and please refer to the *i.MX_Linux_Reference_Manual.pdf*. It is a generic document for all iMX series processor, so you cannot find some topics which are not applicable for iMX7 SoC.

RS-232 UART Interface

The iMX7 SoC provides 7 UART ports. You cannot use all the 7 UARTs in the design as it limits the option to use other interfaces. The following table provides information about the various UART port options.

Table 4: RS-232 UART Interface

UART Port	Used as
UART1	A7 Core debug UART. Can be used for other purpose if required.
UART2	Not Available for use. Pins used for different purpose.
UART3	M4 Core debug UART. Can be used for other purpose if required.
UART4	Not Available for use. Pins used for different purpose.
UART5	Terminated in CN50 of Acacia Board.
UART6	Bluetooth Module UART in SOM.
UART7	Terminated in CN51 as RS485 signals in Acacia, can be used as RS232 if required.

The UART5 port which is provided in Acacia baseboard connector (CN50) can be used with hardware flow control. All 4 signals such as RX, TX, RTS and CTS are terminated in connector CN50. The UART5 is exposed in /dev/ttymxc4 node in Acacia.

To receive and transmit the data through UART5 port of eSOMiMX7, follow these steps:

1. Run the following stty command to configure the baud rate of serial port.

```
root@esomimx7X:~# stty -F /dev/ttymxc4 115200
```

2. Run the following cat command to receive data in UART5 port of eSOMiMX7.

```
root@esomimx7X:~# cat /dev/ttymxc4
```

3. Press **CTRL+C** to exit from the cat command.

4. Run the following echo command to send data through UART5.

```
root@esomimx7X:~# echo "hello" > /dev/ttymxc4
```

RS-485 Interface

The UART7 Signals such as RX, TX and RTS signals are terminated in the connector CN51 of the Acacia board as RS-485 interface. The UART7 RTS signal (EPDC_DATA14) needs to be configured as GPIO to use it as Data Enable pin for RS-485 interface. You need to manually toggle the pin accordingly for RX and TX.

For RS-485 Transmit, run the following commands to drive the data enable signal high.

```
root@esomimx7X:~# echo 46 > /sys/class/gpio/export
root@esomimx7X:~# echo out >
/sys/class/gpio/gpio46/direction
root@esomimx7X:~# echo 1 > /sys/class/gpio/gpio46/value
```

Follow the transmit operation mentioned in the *RS-232 UART Interface*

section.

For RS-485 receive, run the following commands to drive the data enable signal low.

```
root@esomimx7X:~# echo 46 > /sys/class/gpio/export
root@esomimx7X:~# echo out >
/sys/class/gpio/gpio46/direction
root@esomimx7X:~# echo 0 > /sys/class/gpio/gpio46/value
```

Follow the receive operation mentioned in the *RS-232 UART Interface*

section.

Display Interface

Acacia development board supports MIPI DSI and HDMI (using parallel to HDMI converter) interfaces for display. The display interface can be selected using the **set_display** U-Boot command. By default, HDMI display is enabled with resolution set to 1280 x720 at 60 Hz.

The following table shows the commands to select HDMI and MIPI display.

Table 5: Display Interface

Display Interface	U-Boot Command
HDMI	U-Boot > set_display hdmi 1280x720@60
MIPI DSI	U-Boot > set_display mipi

HDMI Interface

HDMI interface support is not available in iMX7 SoC. The parallel display interface of iMX7 is converted to HDMI using a HDMI Transceiver. HDMI Transceiver driver is available in kernel image as builtin driver. The maximum supported resolution is 1920x1080 at 60 Hz. HDMI display support is available in both U-Boot and kernel. In U-Boot, boot logo is displayed in the HDMI display. The HDMI Transceiver driver source is shown below.

HDMI Transceiver driver source: **drivers/video/fbdev/mxc/adv7511/**.

MIPI Display Interface

Acacia is provided with the MIPI display port. Acacia is also provided with an add-on portrait MIPI display of resolution 480 x 800. Currently MIPI display support is available only in kernel. The display brightness of the MIPI display can be controlled using the PWM signal. A driver is included to control the PWM signal, this driver exposes sysfs entry through which you can control the brightness level.

Run the following command to control brightness.

```
root@esomimx7X:~# echo Y  >
/sys/class/backlight/backlight/brightness
```

Where, Y is in the range 0 to 8.

0 is the Lowest (Completely OFF) brightness level

8 is the Highest brightness level

Using Custom MIPI Display

You can connect any MIPI DSI compatible display to Acacia's MIPI display interface. You need to customize the MIPI display driver for the new display timing.

To add a new MIPI panel, follow these steps:

1. Modify lcd_panel value in imx7x-acacia.dts.
2. Add panel entry in mipi_dsi_lcd_db structure in mipi_dsi_samsung.c.
3. Add the MIPI panel driver.

For example, for the add-on MIPI DSI display panel OTM8019a module, follow these steps:

1. Set lcd_panel value as **TRULY-WVGA-MTF0397SWI.**
2. Add the following entry in mipi_dsi_lcd_db structure.

```
#ifdef CONFIG_FB_MXC_MTF0397SWI_PANEL
        {
        "TRULY-WVGA-MTF0397SWI",
```

```
      {mipid_otm8019a_get_lcd_videomode,
mipid_otm8019a_lcd_setup}
      },
#endif
```

3. Add the MIPI panel driver mxcfb_otm8019a_wvga.c with mipid_otm8019a_lcd_setup() function and mipid_otm8019a_get_lcd_videomode structure.

The mipid_otm8019a_get_lcd_videomode structure provides the display timing values for OTM8019A display panel.

Testing Display Interface

The display interface such as HDMI and MIPI can be tested using the test application provided in **/unit_tests/Display** folder.

Run the following command to test the display interface:

```
root@esomimx7X:~#
/unit_tests/Display/display_test_colour.o
```

This application will test the display by rendering different color patterns on the display.

Wi-Fi

The eSOMiMX7 SOM module has a TiWi BLE Wi-Fi and Bluetooth combo module. Wi-Fi module is interfaced with iMX7 through the MMC interface. Wi-Fi module offers 802.11 b/g/n high throughput and extended range along with Wi-Fi and Bluetooth coexistence in a power-optimized design. Two versions of Wi-Fi module are supported, one supporting 2.4 GHz and another version support for both 2.4 Ghz and 5 Ghz mode. Both versions support dual antenna for better throughput. The Wi-Fi module comes with pre-programmed MAC address. The Wi-Fi driver provided with the Linux kernel is used.

Wi-Fi driver source is available at drivers/net/wireless/ti/.

The rootfs packages which are required for Wi-Fi bring-up are as follows:

- net-tools
- wpa_supplicant
- ho
- stapd

Wi-Fi Commands

The important Wi-Fi commands which can be used in Acacia board is explained in this section. The Wi-Fi module driver gets loaded on boot-up.

Run the following command to assign IP address to the interface.

```
root@esomimx7X:~# ifconfig wlan0 <IP_address> up
```

For example,

```
root@esomimx7X:~# ifconfig wlan0 192.168.1.160 up
```

Run the following commands to check configuration of the Wi-Fi network interfaces.

```
root@esomimx7X:~# ifconfig wlan0
wlan0     Link encap:Ethernet  HWaddr c4:f3:12:56:27:cf
          BROADCAST MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0
frame:0
          TX packets:0 errors:0 dropped:0 overruns:0
carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)
```

To connect to access point (AP), follow these steps:

1. Run the following commands to create a configuration file that describes the AP properties.

```
root@esomimx7X:~# vi /etc/wpa_supplicant_econsys.conf
network={
                ssid="econsys"
                psk="12345abcde"
                scan_ssid=1
                key_mgmt=WPA-PSK
                pairwise=CCMP TKIP
                group=CCMP TKIP
}
```

2. Run the following command to connect to the AP using wpa_supplicant tool.

```
root@esomimx7X:~# wpa_supplicant -Dnl80211 -iwlan0 -c
/etc/wpa_supplicant_econsys.conf -B
```

On successful connection, you will get the following messages.

```
Successfully initialized wpa_supplicant
rfkill: Cannot open RFKILL control device
```

3. Run the following command to add default gateway.

```
root@esomimx7X:~# route add default gw <gateway_IP>
dev wlan0
```

For example,

```
root@esomimx7X:~# route add default gw 192.168.6.1 dev
wlan0
```

4. Run the following command to add gateway IP to the configuration file.

```
root@esomimx7X:~# vi /etc/resolv.conf
nameserver <gateway_IP>
nameserver 192.168.6.1
Search
```

Note: Refer to the *Wi-Fi Testing*

in the Appendix section, for details about the Wi-Fi hotspot and Wi-Fi direct mode.

Bluetooth

The Bluetooth module is present inside TiWi BLE Wi-Fi and Bluetooth combo module. The Bluetooth module supports both the classic and BLE v4.1 mode. Bluetooth module shares antenna with Wi-Fi. The Bluetooth subsystem is controlled through a HCI 4-wire UART interface. The UART6 port of iMX7 is connected to the Bluetooth module. The Bluetooth Enable (BT_EN) signal which is connected to the GPIO of iMX7 is used to enable the Bluetooth module. Classic profiles test with the Bluetooth module are FTP, HID, A2DP and Serial profile.

The rootfs packages which are required for Bluetooth bring-up are as follows:

- Bluez5
- obexftp

Follow the below procedure, to test Bluetooth FTP profile.

Bluetooth Commands

The Bluetooth commands are explained in this section.

1. Run the following command to kill the launched bluetoothd daemon.

```
root@esomimx7x:~# killall -9 bluetoothd
```

Note: It is recommended to kill the launched bluetoothd daemon.

2. Run the following hciattach commands to create hci0 interface.

```
root@esomimx7x:~# hciattach -t 10 -s 115200
/dev/ttymxc5 texas 3000000 flow
Found a Texas Instruments' chip!
Firmware file : /lib/firmware/ti-
connectivity/TIInit_11.8.32.bts
Loaded BTS script version 1
```

```
texas: changing baud rate to 3000000, flow control to
1
Device setup complete
```

3. Run the following command to bring up hci0 interface.

```
root@esomimx7x:~# hciconfig hci0 up
```

Note: Refer to the *Bluetooth Profile testing*

section *of Appendix* section, for more details about the Bluetooth profiles.

Camera Interface

eSOMiMX7 has a 2 Lane MIPI camera interface and its signals are terminated in connector CN14 of the Acacia board. The OV8865 Camera module is tested with Acacia's MIPI interface. Camera MIPI interface driver is available as loadable module driver in rootfs, on selecting camera module in U-Boot using setcam command, the camera driver will be loaded on boot-up. You can verify the camera module load in dmesg log as given below.

```
root@esomimx7X:~# dmesg | grep ov8865
camera ov8865_mipi is found
```

Note: This procedure is written with the assumption that the camera module is connected to the Acacia device during boot.

You can stream the camera output on HDMI or MIPI display using the ecamapp application. ecamapp is a sample console-based camera application which is available with Acacia's rootfs. The source of ecamapp is provided with the Acacia source Git.

Run the following command, to launch ecamapp.

```
root@esomimx7X:~# ecamapp
```

ecamapp is V4L2 based command line application to test the features of the camera sensor provided with Acacia. Refer to the *Acacia_Linux_Camera_OV8865_User_Guide.pdf*, for more details on features and other controls supported by the camera.

UVC Camera

eSOMiMX7 also support USB 2.0 cameras. You can stream the camera output on HDMI or MIPI display using gst-launch command. For example, below command streams a 720P video of a USB camera.

```
root@esomimx7X:~# export DISPLAY=:0.0

root@esomimx7X:~# export TERM=sun
```

```
root@esomimx7X:~# gst-launch-1.0 imxv4l2src
device=/dev/video1 ! "video/x-
raw,format=(string)YUY2,width=(int)1280,height=(int)720"
! videoconvert ! fpsdisplaysink text-overlay=false video-
sink=ximagesink sync=false -v
```

Note: The above command is an example command. Make sure you are using correct video node, supported video format and resolution.

To know details about video node, run the following command.

```
root@esomimx7X:~# v4l2-ctl --list-devices
```

To know details about supported format and resolution of video node, run the following command.

```
root@esomimx7X:~# v4l2-ctl -D -d /dev/video1 --list-
formats-ext
```

Ethernet Interface

Acacia kit provides 2 Gigabit Ethernet port for eSOMiMX7 Dual SOM and 1 Gigabit port for Solo SOM, Ethernet PHY is present in eSOMiMX7 SOM. On boot-up, both the ethernet interfaces are initialized and available as eth0 and eth1. In Acacia board, CN26 is the dual Ethernet connector, eth0 interface is top connector and eth1 interface is bottom connector.

Ethernet Commands

The Ethernet commands are explained in this section.

Run the following command to assign IP address to eth0 interface.

```
root@esomimx7X:~# ifconfig eth0 <IP_ADDRESS> up
```

For example,

```
root@esomimx7X:~# ifconfig eth0 192.168.6.160 up
```

Run the following command to add default gateway route.

```
root@esomimx7X:~# route add default gw <DEFAULT_GATEWAY>
eth0
```

For example,

```
root@esomimx7X:~# route add default gw 192.168.6.1 eth0
```

Run the following command to add DNS server.

```
root@esomimx7X:~# vi /etc/resolv.conf
```

```
search
nameserver 192.168.6.1
```

USB

The iMX7 processor has two high speed USB OTG controller. Among the 2 USB OTG ports, one is terminated as USB Host port and other port is terminated as USB OTG port in the eSOMiMX7 Acacia board.

USB Host

The Host port in Acacia supports the USB 2.0 high speed (HS) devices and it is backward compatible and supports low speed devices. USB mouse, keyboard, USB Mass storage can be connected to this USB Host port in connector (CN17).

USB OTG

The USB OTG port is terminated in connector (CN19). This port is used for USB Serial download port for MfgTools and other USB download tools for iMX. The USB client interface is also tested with mass storage gadget driver in Linux kernel.

USB OTG Client as Mass Storage

To use USB OTG as mass storage, you can use g_mass_storage driver module.

Run the following command to format and to insert g_mass_storage device.

```
root@esomimx7X:~# modprobe g_mass_storage
file=/dev/<Storage_Device_Node>
```

For example,

```
root@esomimx7X:~# modprobe g_mass_storage
file=/dev/mmcblk0p1
```

Note: The <Storage_Device_Node> is the device which is mounted to the host, when the eSOMiMX7 SOM module acts as client. It must not be the rootfs partition which is currently active.

RTC

The eSOMiMX7 Acacia board has a RTC chip (DS1338). The real time clock (RTC) is used to store the date and time and this helps to retain the time and date in case of power off. During ON state the time is maintained by two parties such as the clock within the processor (system time) and RTC (hardware clock). During OFF state the RTC chip is powered with a coin cell battery and thus the time is maintained in the RTC chip. The RTC chip is connected to iMX7 SoC through the I2C interface (I2C2). The RTC can be accessed using userspace commands.

RTC Commands

The RTC commands are explained in this section.

hwclock Command

To read date and time from RTC.

```
root@esomimx7X:~# hwclock
Wed Jul 10 22:37:19 2019  0.000000 seconds
```

date Command

To read date and time from system clock.

```
root@esomimx7X:~# date
Wed Jul 10 22:36:43 UTC 2019
```

date -s "YYYYMMDD hh:mm:ss" Command

To set system clock date and time.

```
root@esomimx7X:~# date -s "YYYYMMDD hh:mm:ss"
root@esomimx7X:~# date -s "20180328 15:38:00"
Wed Mar 28 15:38:00 UTC 2018
```

hwclock -w Command

To set RTC from system time.

```
root@esomimx7X:~# hwclock -w
```

hwclock -s Command

To update the system time from RTC.

```
root@esomimx7X:~# hwclock -s
```

Audio

The eSOMiMX7 Acacia board has WM8750 audio codec from Cirrus logic. The audio subsystem has an audio output port and a mic input port. The WM8750 configurations are set through the I2C3 port of iMX7. On boot the audio driver is loaded and you can use the audio through the standard commands. Acacia board does not have an inbuilt speaker you need to connect headphone to 3.5mm audio jack in CN5 of Acacia to listen to audio and also you need to connect MIC jack to CN6 connector of Acacia.

Audio Commands

The audio commands are explained in this section.

aplay Command

To play audio.

```
root@esomimx7X:~# aplay <filename>
```

For example,

```
root@esomimx7X:~# aplay /media_files/test_play.wav
```

arecord Command

To record audio.

```
root@esomimx7X:~#  arecord -f S32_LE <output_filename> -d
<duration_to_be_recorded_in_seconds> -r
<rate_of_recording>
```

For example,

```
root@esomimx7X:~# arecord -f S32_LE
/test_apps/test_record.wav -d 10 -r 44100
```

amixer sset Headphone <Volume> Command

To set audio volume.

```
root@esomimx7X: amixer sset Headphone <Volume>
```

For example,

```
root@esomimx7X: amixer sset Headphone 127
```

amixer sset "Mic Boost" <MicBoost> Command

To increase Mic sensitivity or Mic boost.

```
root@esomimx7X: amixer sset "Mic Boost" <MicBoost>
```

For example,

```
root@esomimx7X: amixer sset "Mic Boost" 127
```

amixer sset PCM <Volume> Command

To set volume of PCM mixer.

```
root@esomimx7X: amixer sset PCM <Volume>
```

For example,

```
root@esomimx7X: amixer sset PCM 255
```

amixer get Command

To verify the proper setting of volume.

```
root@esomimx7X: amixer get Headphone
root@esomimx7X: amixer get "Mic Boost"
root@esomimx7X: amixer get PCM
```

PCIe

Acacia baseboard has a mini PCI Express interface (CN3) located at the bottom of the board. The mPCIe slot with an USB3.0 hub (MPEXUSB3S22B from StarTech) and Gigabit Ethernet card (Realtek 8169 PCIe Ethernet card) are verified and by the e-con Systems. The PCIe slot also supports the USB port and SIM slot. The Huawei 4G modem (Huawei ME909U-521) to be interfaced are tested by the e-con Systems.

Note: PCIe support is available only in Dual SOM, iMX7 Solo do not support the PCIe interface.

LEDs

The eSOMIMX7 SOM has a debug LEDs connected to GPIO 204 and it is active high LED. This LED is turned ON at the bootloader start up, it is used to verify whether U-Boot is loaded, and device boot is successful.

You can also control this LED in kernel.

Run the following system commands, to control the LED operations in kernel.

To turn **ON** LED

```
root@esomimx7X:~# echo 204 > /sys/class/gpio/export
root@esomimx7X:~# echo out >
/sys/class/gpio/gpio204/direction
root@esomimx7X:~# echo "1" >
/sys/class/gpio/gpio204/value
```

To turn **OFF** LED

```
root@esomimx7X:~# echo "0" >
/sys/class/gpio/gpio204/value
```

CAN Interface

The flexible controller area network (FlexCAN) module is a communication controller implementing the CAN protocol according to the CAN 2.0B protocol specification. The FlexCAN module has four functional modes such as normal Mode (user or supervisor), freeze mode, listen-only mode and loop-back mode. The can-utils package is added in rootfs to support the various CAN operations to utilize CAN interface.

Note: These CAN commands are tested by connecting CAN interface of 2 Acacia boards (CN52) across each other.

CAN Commands

The CAN commands are explained in this section.

ip link

To bring up CAN port and set the speed for data transfer.

```
root@esomimx7x:~# ip link set can0 up type can bitrate
125000
```

candump

To read data received in CAN port.

```
root@esomimx7x:~# candump can0
```

cansend

To send a data frame through CAN with identifier 0x1A and 8 bytes of data.

```
root@esomimx7x:~# cansend can0 01a#11223344AABBCCDD
```

You will receive the can0 01A [8] 11 22 33 44 AA BB CC DD data in receiver side.

IMU

Acacia board features an IMU chip - LSM6DS3 from STMicroelectronics. It is a system-in-package featuring a high-performance 3-axis digital accelerometer and 3-axis digital gyroscope. It also features an integrated temperature sensor to get temperature values. IMU chip is connected to I2C3 of eSOMiMX7 module. IMU driver expose sysfs entries to read the values from the chip.

Run the following sample commands to read accelerometer, gyroscope and temperature value.

```
root@esomimx7x:~# /unit_tests/IMU/acc_read.sh
root@esomimx7x:~# /unit_tests/IMU/gyr_read.sh
root@esomimx7x:~# /unit_tests/MU/temperature_read.sh
```

The above commands will provide the RAW values from the sensor.

Device Tree

A device tree is a data structure for describing hardware and it provides platform specific information to the Linux kernel. The device tree source is available in the **arch/arm/boot/dts/** path in the Linux kernel.

The above directory contains the following files:

- DTS is device tree source file.
- DTSI is device tree source include file.
- DTB (output of compiled device tree source).

The device tree file for Acacia is splitted in to following types:

- Baseboard DTS
- SOM DTS
- SoC DTS

The main advantage of having splitted DTS is to reduce time for customization and better understanding of DTS files. To prepare DTS for a custom board, you must modify the baseboard DTS file alone there is no need to change SOM or SOC DTS files. Similarly, whenever you want to change the SOM from Solo to Dual or Dual to Solo then you need to change the baseboard DTS's include file accordingly.

The contents of Solo and Dual SoC are splitted, since the Dual SoC is superset of Solo and the Solo DTS files are included in the Dual related DTS file. The Dual related DTS only contains platform information specific to Dual (features not available in Solo).

The following table shows the DTS Type and its description for Dual and Solo SOM.

Table 6: DTS Type and its Description

SOM	DTS Type	DTS/DTSI File	Description
eSOMiMX7 Dual – Acacia	Baseboard DTS	arch/arm/boot/dts/imx7d-acacia.dts	Acacia board specific information for iMX7 Dual SOM. Adds SOM and SOC DTS as includes.
	SOM DTS	arch/arm/boot/dts/imx7d-esomimx7.dtsi	eSOMiMX7 Dual SOM specific information.
	SoC DTS	arch/arm/boot/dts/imx7d.dtsi	iMX7 Dual SoC specific information.
eSOMiMX7 Solo – Acacia	Baseboard DTS	arch/arm/boot/dts/imx7s-acacia.dts	Acacia board specific information for iMX7 Solo SOM. Adds SOM and SOC DTS as includes.

	SOM DTS	arch/arm/boot/dts/imx7s-esomimx7.dtsi	eSOMiMX7 Solo SOM specific information.
	SoC DTS	arch/arm/boot/dts/imx7s.dtsi	iMX7 SoC specific information.

The example of a platform data present in imx7s-esomimx7.dtsi (platform data which is used for Bluetooth), as Bluetooth module is present in SOM and as it is common to both Solo and Dual.

The following entry is included in imx7s-esomimx7.dtsi file.

```
/*Bluetooth*/
&uart6 {
    pinctrl-names = "default";
    pinctrl-0 = <&pinctrl_uart6>;
    assigned-clocks = <&clks IMX7D_UART6_ROOT_SRC>;
    assigned-clock-parents = <&clks
IMX7D_PLL_SYS_MAIN_240M_CLK>;
    fsl,uart-has-rtscts;
    resets = <&modem_reset>;
    status = "okay";
};
```

Modifying Device Tree for Custom Baseboard

All the SOM related device tree entries are present in imx7d-esomimx7.dtsi and imx7s-esomimx7.dtsi for Dual and Solo SOM respectively. For a custom baseboard, you must include these include files in your custom board device tree. You can take Acacia board's device tree as reference to create new device tree file, for example, mx7d-<CUSTOM_BOARD>.dts.

For example, assume that a new interface is added in i2c1. Then an entry must be added under i2c1 device node and the essential property related to i2c must be added.

The sample entry for i2c device is shown below.

```
&iomuxc {
      esomimx7 {
            pinctrl_lsm6ds3: imugrp-1 {
                  fsl,pins = <
                        MX7D_PAD_EPDC_BDR1__GPIO2_IO29
0x7d /* IMU_INT2 */
                        MX7D_PAD_EPDC_BDR0__GPIO2_IO28
0x7d /* IMU_INT1 */
```

```
                    >;
            };
      };
};
&i2c1 {
      clock-frequency = <100000>;
      pinctrl-names = "default";
      pinctrl-0 = <&pinctrl_i2c3>;
      status = "okay";

      imu:lsm6ds3@6a {
            pinctrl-names = "default";
            pinctrl-0 = <&pinctrl_lsm6ds3>;
            compatible = "st,lsm6ds3";
            reg = <0x6a>;
            st,drdy-int-pin = <1>;
      };
};
```

Partition Information

This section explains about the partition details of SD, eMMC and NAND boot device in this section.

The following section shows the partitions of boot device

SD Card Partitions

The following table shows the SD card partitioning details.

Table 7: SD card Partition

Partition Number	Partition Name	Size	Type	Content
1	-	3 MB	RAW	u-boot.imx + boot environment
2	boot	12 MB	FAT32	zImage, dtb and ramdisk image
3	rootfs	1.6 GB	EXT3	Yocto rootfs
4	data	Remaining	Unpartitioned*	None

*Can be partitioned manually

Note: In SD card, first 3MB is used to store bootloader this will also maintain compatibility between the eMMC and SD card partitions.

eMMC Partitions

According to JEDEC 4.4 standard, the eMMC hardware partitions are as follows:

- One main hardware partition (user data area [UDA]), [mmcblk0]
- Two boot partitions [mmcblk0boot0 and mmcblk0boot1]
- One replay protected memory block (RPMB) partition

Note: The first two types of hardware partitions are used, not the RPMB partition. The boot partitions have a simplified read access mechanism that facilitates the boot operation when powering ON.

The primary bootloader (U-Boot) is flashed in the first boot partition, as mmcblk0boot0. The second boot partition is not used. The user data area hardware partition is further partitioned for kernel and rootfs.

The following table shows the eMMC partitioning details.

Table 8: eMMC Partition

Partition Number	Partition Name	Size	Type	Content
1	mmcblk0boot0[#]	2 MB	RAW	u-boot.imx
2	mmcblk0boot1[#]	2 MB	RAW	Not Used
3	mmcblk0p1[*]	500 MB	FAT32	zImage, DTB and ramdisk image
4	mmcblk0p2	3 GB	EXT3	Yocto rootfs
X	Unpartitioned	Remaining	N/A	Can be used for data storage

[#]Hardware partition

*First 3MB is left without partitioning to store environment variables, so mmcblk0p1 starts at offset of 3MB.

NAND Partitions

The following table shows the NAND partitioning details.

Table 9: NAND Partition

S. No.	Partition Name	Size	Type	Partition Stores
1	mtd0	64 MB	RAW	Boot
2	Mtd1	16 MB	RAW	Kernel
3	Mtd2	16 MB	RAW	Device Tree
4	Mtd3	4 MB	RAW	Ramdisk image
5	Mtd4	1.9 GB	EXT3	Rootfs

Note:

- The 2 GB NAND is used at present.
- Rootfs partition size can vary based on NAND size used in future.

CPU Frequency

In eSOMiMX7, the processor can run up to maximum frequency of 1 GHz. The iMX7 Linux BSP supports CPU frequency scaling to switch CPU frequency based on the scaling mode. You can get or set CPU frequency using sysfs interface. The /sys/devices/system/cpu/cpu0/cpufreq node has several module parameters for setting up the frequency scaling.

Run the following command to get the maximum frequency of CPU.

```
root@esomimx7X:~# cat
/sys/devices/system/cpu/cpu0/cpufreq/cpuinfo_max_freq
996000
```

Run the following command to get current CPU frequency.

```
root@esomimx7X:~#  cat
/sys/devices/system/cpu/cpu0/cpufreq/cpuinfo_cur_freq
996000
```

Frequency Scaling

Use CPUfreq to reduce the power consumption and heat output on your system. CPUfreq is also referred to as the CPU speed scaling allows the clock speed of the processor to be adjusted on the fly. This enables the system to run at a reduced clock speed to save power. The rules for shifting frequencies, whether to a faster or slower clock speed, and when to shift frequencies, are defined by the CPUfreq governor.

The following table shows the description of the CPU Governor supported in Acacia BSP.

Table 10: Description of CPU Governor

CPU Governor	Description
Performance	Forces the CPU to use the highest clock frequency
Powersave	Forces the CPU to use the lowest clock frequency
Ondemand	A dynamic governor that allows the CPU to achieve maximum clock frequency when system load is high, and also minimum clock frequency when the system is idle.
Userspace	Allows userspace programs to set the frequency
Conservative	Like the Ondemand governor, adjusts the clock frequency according to usage but switches between frequencies more gradually.

Run the following command to change the scaling governor by setting CPU governor name.

```
root@esomimx7X:~# echo performance >
/sys/devices/system/cpu/cpu0/cpufreq/scaling_governor
```

Run the following command to set the CPU frequency with CPU governor set to userspace mode.

```
root@esomimx7X:~# echo
userspace>/sys/devices/system/cpu/cpu0/cpufreq/scaling_go
vernor

root@esomimx7X:~# echo 792000 >
/sys/devices/system/cpu/cpu0/cpufreq/scaling_setspeed
```

Updating Logo or Splash Screen

This section describes about uploading the logo and splash screen.

Changing Boot Logo

The eSOMiMX7 Acacia U-Boot supports 8-bit palletized, uncompressed bitmap (bmp) file for boot logo. The boot logo is processed during the u-boot build time and integrated with the u-boot binary. The maximum supported resolution of the boot logo image is 800 x 480.

Converting BMP and Embed to U-Boot

To convert the BMP and embed to U-Boot, follow these steps.

1. Open target BMP image in **GIMP** editor.
2. Go to **Image>Mode>Indexed**. The **Indexed Color Conversion** dialog box appears.

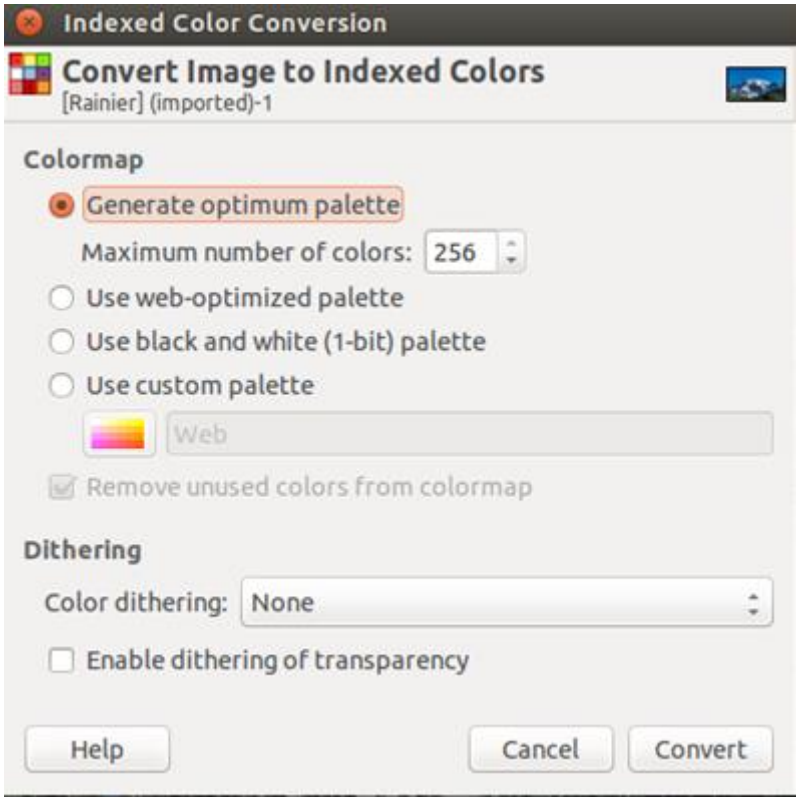


Figure 4: Converting the BMP and embedding to U-Boot

Note: Make sure the settings are as per the above screen.

3. Click **Convert** to convert the image to indexed colors.
4. Go to **File>Export As**. The **Export Image as BMP** dialog box appears.

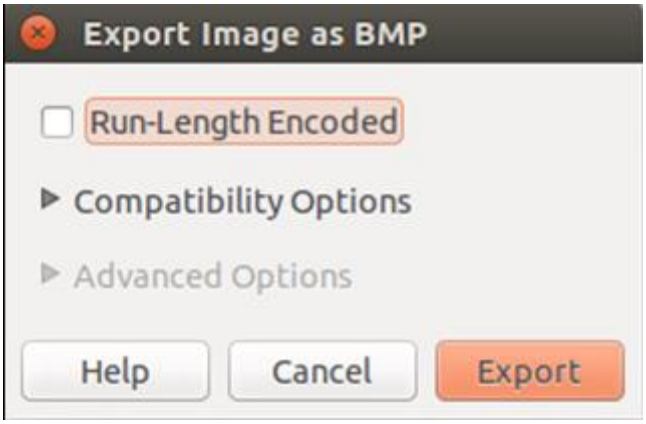


Figure 5: Export Image as BMP Dialog Box

5. Click **Export** to export the image.
6. Save the image file as **econ.bmp**.
7. Copy the image to **<YOCTO_DIR>/sources/esomimx7-uboot/tools/logos/** using cp command.
8. Change terminal to **<YOCTO_DIR>/sources/esomimx7-uboot directory** using **cd** command.
9. Commit the boot logo changes using git add and git commit commands.
10. Change terminal to build directory using cd command.
11. Build using bitbake esomimx7-image-gui.

    Once the build is successful, you can update the u-boot.imx in the target board.

    Reboot the device with updated U-Boot to see the updated logo.

    Note: To change the logo name to custom name instead of econ.bmp, you need to add support for vendor and board name in U-boot source. Also, you need to do necessary changes in Yocto build to add support for new vendor and board.

Changing Kernel Splash Screen

Kernel splash screen image supports the RGB888 and RGB444 BMP image. The maximum supported resolution is 1920x1080. The default splash screen image resolution is 1596x871.

To change the kernel splash screen, follow these steps:

1. Open a terminal in your Linux development PC.
2. Run the following command to install gdk-pixbuf-csource package.

```
$ sudo apt-get install libgtk2.0-dev
```

3. Run the following command to navigate to psplash git directory.

```
$ cd <BUILD_DIR>/tmp/work/cortexa7hf-neon-poky-linux-gnueabi/psplash/0.1+gitAUTOINC+88343ad23c-r15/git/
```

4. Copy your logo image as sample.bmp.

5. Run the following command to generate the splash screen image header file from the BMP file.

```
$ ./make-image-header.sh sample.bmp POKY
```

You will get sample.bmp-img.h in same directory.

6. Run the following command to copy the file as psplash-econ-img.h.

```
$ sudo cp sample.bmp-img.h ../psplash-econ-img.h
```

7. Navigate to build directory.
8. Run the following bitbake psplash command to customize your splash screen.

```
$ bitbake -c cleansstate esomimx7-image-gui
$ bitbake esomimx7-image-gui
```

After successful build, you will get the rootfs with splash screen image updated.

Flash the rootfs to your board to see the updated splash screen logo.

Bluetooth Profile testing

The Bluetooth profile testing involves the functions such as launching Bluetooth Daemons, Bluetooth pairing, Bluetooth file transfer and so on. The following sections describe each of the functions in detail.

Launching Bluetooth Daemons

Before using the profiles, some of the profiles requires daemons to be launched.

Run the following commands to launch the daemons.

```
root@esomimx7x:~# eval `dbus-launch --sh-syntax`
root@esomimx7x:~# DISPLAY=:0 /usr/libexec/bluetooth/obexd -na &
root@esomimx7x:~# /usr/libexec/bluetooth/bluetoothd -n &
root@esomimx7x:~# ofonod
```

Note: Make sure to follow the same sequence as shown above in launching the daemons.

The dbus-launch command is used to start a session bus instance of dbus-daemon. D-Bus is a message bus system, a simple way for applications to talk to one another. dbus-daemon is the D-Bus message bus daemon.

OBject EXchange (OBEX) is a communications protocol that facilitates the exchange of binary objects between devices. obexd is daemon server for OBEX.

bluetoothd is the Bluetooth daemon that is the stack. It provides the core operations such as search for devices, pair, and so on and Bluetooth profiles such as A2DP (music streaming), human interface devices (HID), and so on. Bluetooth daemon exposes a DBus API for the applications to talk to it.

ofonod is responsible for the mobile telephony part and bluetoothd uses it to place and answer phone calls. It is not, however, involved in call audio (the HFP audio) at all as this is usually routed, for efficiency, through PCM and completely bypasses the Bluetooth software stack.

Bluetooth Pairing

Pairing a device can be done using Bluetootctl application, you can launch bluetoothctl's interactive mode to do pairing. On launching the application, the list of devices which are already paired is displayed.

To initialize and pair the Bluetooth module, follow these steps:

1. Run the following commands to initialize Bluetooth module for scanning.

```
root@imx7dsabresd:~# bluetoothctl
[NEW] Controller 43:39:00:07:1F:AC BlueZ 5.37
[default]
[bluetooth]# power on
Changing power on succeeded
[bluetooth]# agent on
Agent registered
[bluetooth]# default-agent
Default agent request successful
[bluetooth]# pairable on
Changing pairable on succeeded
[bluetooth]# discoverable on
Changing discoverable on succeeded
[CHG] Controller 43:39:00:07:1F:AC Discoverable: yes
```

2. Run the following command to scan for available devices to find the MAC address of the device with which pairing must be done.

```
[bluetooth]# scan on
```

3. Run the following command to turn off scanning.

```
[bluetooth]# scan off
```

4. Run the following commands to pair and connect with the device.

```
[bluetooth]# pair <MAC ADDRESS>
[bluetooth]# connect <MAC ADDRESS>
```

5. Run the following command to quit the application after pairing.

```
[bluetooth]# quit
```

Bluetooth File Transfer

Once the pairing is successfully completed, it is time to connect the OBEX object push profile. To interact with OBEX, the obexctl tool is used.

To launch and use obexctl for file transfer, follow these steps:

1. Run the following commands to launch obexctl application.

```
root@imx7dsabresd:~# obexctl
[NEW] Client /org/bluez/obex
[obex]#
```

The obexctl application such as bluetoothctl, obexctl will also provide interactive mode to issue commands.

2. Run the following command to establish connection with the device to which file transfer is to be done.

```
[obex]# connect <MAC ADDRESS>
```

3. Run the following send command to send a file.

```
[obex]# send <file_name_with_absolute_path>
```

Note: Make sure to mention the absolute path of the file.

The received file will be available in the **/home/root/.cache/obexd** directory, there is no special commands for file receive.

4. Run the following command to quit the application.

```
[obex]# quit
```

Advanced Audio Distribution Profile (A2DP)

This is the Bluetooth stereo profile which defines how high quality stereo audio can be streamed from one device to another over a Bluetooth connection, for example, music streamed from a mobile phone to wireless headphones. Setting up the A2DP connection requires some configuration files which needs to be prepared and then followed by pulse audio initialization with pactl application.

Preparing Configuration Files

To prepare the configuration files, follow these steps:

1. Add the following lines in /etc/dbus-1/system.d/pulseaudio-system.conf file. Insert these lines before “</busconfig>”.

```
<policy user="pulse">
      <allow own="org.pulseaudio.Server"/>
      <allow send_destination="org.bluez"/>
      <allow
send_interface="org.freedesktop.DBus.ObjectManager"/>
</policy>
```

2. Add the following lines in /etc/dbus-1/system.d/bluetooth.conf file. Insert these lines before “</busconfig>”.

```
<policy user="pulse">
      <allow send_destination="org.bluez"/>
      <allow
send_interface="org.freedesktop.DBus.ObjectManager"/>
</policy>
```

3. Append the following lines in /etc/pulse/system.pa file at the end.

```
.ifexists module-bluetooth-policy.so
    load-module module-bluetooth-policy
.endif
.ifexists module-bluetooth-discover.so
    load-module module-bluetooth-discover
.endif
```

Creating HCI Interface

Refer to the *Bluetooth Commands* section in eSOMiMX7 *Acacia Interfaces and Peripherals* section, for details about creating the HCI interface.

Restarting Pulse Audio

Run the following commands to restart pulse audio.

```
root@esomimx7X:~# pulseaudio -k
root@esomimx7X:~# pulseaudio --start
```

Pairing and Playing Audio

Refer to the Bluetooth Pairing

section in Appendix, to pair to the Bluetooth Audio device.

And, follow these steps to play the audio:

1. Run the following commands to list the available audio cards using paplay through A2DP.

```
root@esomimx7X:~# pactl list cards
......
Card #2
Name: bluez_card.00_1A_7D_14_26_8B
Driver: module-bluez5-device.c
Owner Module: 24
Properties:
device.description = "Byte-DM5710BT"
device.string = "00:1A:7D:14:26:8B"
device.api = "bluez"
device.class = "sound"
device.bus = "bluetooth"
device.form_factor = "headset"
......
```

Note: The card name with bluez text is sound card for A2DP audio.

2. Run the following command to set the audio sink to the bluez card.

```
root@esomimx7X:~# pactl set-card-profile <card no>
a2dp_sink
```

For example, for the desired card number of 2.

```
root@esomimx7X:~# pactl set-card-profile 2 a2dp_sink
```

3. Run the following command to play the desired audio file by specifying the sink name of the desired card.

```
root@esomimx7X:~# paplay -p --
device=bluez_sink.XX_XX_XX_XX_XX_XX
/media_files/test_play.wav
```

For example,

```
root@esomimx7X:~# paplay -p --
device=bluez_sink.00_1A_7D_14_26_8B
/media_files/test_play.wav
```

Headset Profile (HSP)

The headset profiles are required for headset operations. The following sections describe each of the functions in detail.

Preparing configuration files

Refer to *Preparing Configuration Files*

section, to get details about the preparing configuration files.

Creating HCI Interface

Refer to the *Bluetooth Commands* section, to create HCI interface.

Pairing Devices

Refer to the *Bluetooth Pairing*

section, for pairing process.

Playing Audio File

Pulse audio tools can be used to play the audio files.

To play the audio files, follow these steps:

1. Run the following commands to list the available cards to check the card number of imx-audio-btlsr card.

```
root@esomimx7d-1gb:~# cat /proc/asound/cards
   0 [wm8750audio ]: wm8750-audio - wm8750-audio
                     wm8750-audio
```

```
   1 [imxaudiobtlsr ]: imx-audio-btlsr - imx-audio-
btlsr
                imx-audio-btlsr
```

2. Run the scotest application, to establish SCO connection.

```
root@esomimx7d-1gb:~# scotest -n <MAC_ADDRESS> &
```

3. Run the following command to play the desired audio file by specifying the sink name of the desired card.

```
root@esomimx7d-1gb:~# aplay -Dhw:1 <absolute path to
.wav file>
```

Recording audio

After establishing the SCO connection, run arecord with desired card id and format as S16_LE.

```
root@esomimx7d-1gb:~# arecord -Dhw:1 -f S16_LE
test_16le_8k_mono.wav
```

Bluetooth Low Energy (BLE) Commands

The BLE commands are explained in this section.

hcitool

To start a low energy scan using hcitool.

```
root@esomimx7X:~# hcitool lescan
LE Scan ...
68:C9:0B:06:A4:87 (unknown)
68:C9:0B:06:A4:87 (unknown)
68:C9:0B:06:A4:87 CC2650 SensorTag
```

gatttool

To connect to your BLE device.

```
root@esomimx7X:~# gatttool -b <BLE ADDRESS> -I
```

Where,

<BLE ADDRESS> is the address you obtained in the earlier steps, that is address of a BLE host device.

-I indicates you want to open up an interactive session.

If you want to connect the 68:C9:0B:06:A4:87 device, you can follow the procedure listed in the table.

connect

To establish connection with the device.

```
[68:C9:0B:06:A4:87][LE]> connect
```

```
Attempting to connect to 68:C9:0B:06:A4:87

Connection successful
```

primary

To get the primary UUIDs of the device connected.

```
[68:C9:0B:06:A4:87][LE]> primary

attr handle: 0x0001, end grp handle: 0x0007 uuid:
00001800-0000-1000-8000-00805f9b34fb

attr handle: 0x0008, end grp handle: 0x000b uuid:
00001801-0000-1000-8000-00805f9b34fb

attr handle: 0x000c, end grp handle: 0x001e uuid:
0000180a-0000-1000-8000-00805f9b34fb

.......
```

char-desc

To get all the available handles.

```
[68:C9:0B:06:A4:87][LE]> char-desc

handle: 0x0001, uuid: 00002800-0000-1000-8000-
00805f9b34fb

handle: 0x0002, uuid: 00002803-0000-1000-8000-
00805f9b34fb

handle: 0x0003, uuid: 00002a00-0000-1000-8000-
00805f9b34fb

.......
```

char-write-req

To write to a handle.

Syntax: char-write-req <handle> <data>

For example,

```
[68:C9:0B:06:A4:87][LE]> char-write-cmd 0x29 01
```

To read from a handle, use char-read-hnd command.

Syntax: char-read-hnd <handle>

```
[68:C9:0B:06:A4:87][LE]> char-read-hnd 0x25

Characteristic value/descriptor: 0a 26 00 00 00 00 00 00
00 00 b0 00 40 51 04 03 aa 00 f0
```

Wi-Fi Testing

Wi-Fi module present in eSOMiMX7 supports the Wi-Fi Direct and Wi-Fi SoftAP mode. The procedure to configure each mode is explained in the following sections.

Using Wi-Fi SoftAP Mode

To use Wi-Fi SoftAP mode, follow these steps:

1. Run the following command to create a **/etc/hostapd.conf** configuration file for hostapd to act upon.

```
root@esomimx7X:~# vi /etc/hostapd.conf
```

2. Add the following configurations in **hostapd.conf**.

```
interface=wlan0
driver=nl80211
channel=11
ssid=ESOM_IMX7
max_num_sta=5
hw_mode=g
preamble=1
dtim_period=2
beacon_int=100
logger_syslog=-1
logger_syslog_level=2
logger_stdout=-1
logger_stdout_level=2
#dump_file=/tmp/hostapd.dump
ctrl_interface=/var/run/hostapd
ctrl_interface_group=0
supported_rates=60 90 120 180 240 360 480 540
basic_rates=60 90 120 180 240
macaddr_acl=0
tx_queue_data3_aifs=7
tx_queue_data3_cwmin=15
tx_queue_data3_cwmax=1023
tx_queue_data3_burst=0
tx_queue_data2_aifs=3
tx_queue_data2_cwmin=15
tx_queue_data2_cwmax=63
tx_queue_data2_burst=0
tx_queue_data1_aifs=1
tx_queue_data1_cwmin=7
tx_queue_data1_cwmax=15
```

```
tx_queue_data1_burst=3.0
tx_queue_data0_aifs=1
tx_queue_data0_cwmin=3
tx_queue_data0_cwmax=7
tx_queue_data0_burst=1.5
wme_enabled=1
wme_ac_bk_cwmin=4
wme_ac_bk_cwmax=10
wme_ac_bk_aifs=7
wme_ac_bk_txop_limit=0
wme_ac_bk_acm=0
wme_ac_be_aifs=3
wme_ac_be_cwmin=4
wme_ac_be_cwmax=10
wme_ac_be_txop_limit=0
wme_ac_be_acm=0
wme_ac_vi_aifs=2
wme_ac_vi_cwmin=3
wme_ac_vi_cwmax=4
wme_ac_vi_txop_limit=94
wme_ac_vi_acm=0
wme_ac_vo_aifs=2
wme_ac_vo_cwmin=2
wme_ac_vo_cwmax=3
wme_ac_vo_txop_limit=47
wme_ac_vo_acm=0
auth_algs=3
ieee80211d=0
ieee80211n=1
uapsd_advertisement_enabled=1
wep_rekey_period=0
own_ip_addr=127.0.0.1
wpa_group_rekey=0
wpa_strict_rekey=0
wpa_gmk_rekey=0
wpa_ptk_rekey=0
eap_server=1
```

3. Run the following command to kill wpa_supplicant.

```
root@esomimx7X:~# killall - 9   wpa_supplicant
```

4. Run the hostapd application to make the module work in AP mode.

```
root@esomimx7X:~# hostapd /etc/hostapd.conf &
root@esomimx7X:~# ifconfig wlan0 192.168.5.1
```

5. Scan from other devices to find AP ESOM_IMX7 (as mentioned in hostapd.conf).
6. Connect other devices to this AP and assign static IP address in the range 192.168.5.XXX.

You can also run a DHCP server in device to assign dynamic IP to the stations.

Using Wi-Fi Direct Mode

To use Wi-Fi direct mode, follow these steps:

1. Run the following command to create a **/etc/p2p_supplicant.conf file** configuration file in both the boards.

```
root@esomimx7X:~# vi /etc/p2p_supplicant.conf
```

2. Add the following lines to **p2p_supplicant.conf** file.

```
ctrl_interface=/var/run/wpa_supplicant
update_config=0
device_name=IMX7-SOM
device_type=1-0050F204-1
config_methods=virtual_push_button physical_display
keypad
p2p_go_intent=0
country=US
driver_param=use_multi_chan_concurrent=1
use_p2p_group_interface=1
p2p_go_max_inactivity=60
p2p_go_ht40=1
disassoc_low_ack=1
```

The device_name can be modified in the configuration file as per the requirement.

3. Run the following commands to kill any instance of hostapd or wpa_supplicant application.

```
root@esomimx7X:~# killall -9 wpa_supplicant
root@esomimx7X:~# killall -9 hostapd
```

4. Run the wpa_supplicant command.

```
root@esomimx7X:~# wpa_supplicant -Dnl80211 -iwlan0 -c
/etc/p2p_supplicant.conf -B
```

5. Run the following commands to start wpa_cli tool in both the devices to interface with the wpa_supplicant, after launching wpa_supplicant.

```
root@esomimx7X:~# wpa_cli
wpa_cli v2.6
Copyright (c) 2004-2015, Jouni Malinen
<j@w1.fi> and contributors
This software may be distributed under the
terms of the BSD license.
See README for more details.
Selected interface 'wlan0'
Interactive mode
>
```

The wpa_cli tool provides an interactive command prompt which can be used to query current status, change configuration, trigger events and request interactive user input.

6. Run p2p_find command in wpa_cli prompt, to scan for available devices.

```
> p2p_find
P2P-DEVICE-FOUND 80:30:dc:15:b0:ed
p2p_dev_addr=80:30:dc:15:b0:ed pri_dev_type=0-
00000000-0 name='' config_methods=0x1108
dev_capab=0x25 gro
up_capab=0x0 new=0
```

7. Run p2p_stop_find command when another device is detected during scan.

```
> p2p_stop_find
> CTRL-EVENT-SCAN-STARTED
> OK
<3>P2P-FIND-STOPPED
```

8. Run the p2p_peers command to list the devices found when the scanning is completed.

```
> p2p_peers
> 80:30:dc:15:b0:ed
> OK
```

9. Run the command p2p_connect with the MAC address of the peer to connect as argument when the other device is listed as the available peer.

```
> p2p_connect 80:30:dc:15:b0:ed pbc freq=2412
> OK
```

```
<3>P2P-GO-NEG-SUCCESS role=client freq=2412 ht40=1
peer_dev=80:30:dc:15:b0:ed
peer_iface=80:30:dc:15:b0:ee wps_method=PBC

[  301.780717] IPv6: ADDRCONF(NETDEV_UP): p2p-wlan0-2:
link is not ready

<3>CTRL-EVENT-SCAN-RESULTS

<3>WPS-AP-AVAILABLE

[  302.212842] p2p-wlan0-2: authenticate with
80:30:dc:15:b0:ee

[  302.223415] p2p-wlan0-2: send auth to
80:30:dc:15:b0:ee (try 1/3)

[  302.255996] p2p-wlan0-2: authenticated

[  302.263409] p2p-wlan0-2: associate with
80:30:dc:15:b0:ee (try 1/3)

[  302.291161] p2p-wlan0-2: RX AssocResp from
80:30:dc:15:b0:ee (capab=0x411 status=0 aid=1)

[  302.311905] IPv6: ADDRCONF(NETDEV_CHANGE): p2p-
wlan0-2: link becomes ready

[  302.321413] p2p-wlan0-2: associated

<3>P2P-GROUP-FORMATION-SUCCESS

[  303.189084] p2p-wlan0-2: deauthenticating from
80:30:dc:15:b0:ee by local choice (Reason:
3=DEAUTH_LEAVING)

[  303.244512] p2p-wlan0-2: authenticate with
80:30:dc:15:b0:ee

[  303.253610] p2p-wlan0-2: send auth to
80:30:dc:15:b0:ee (try 1/3)

[  303.286016] p2p-wlan0-2: authenticated

[  303.293347] p2p-wlan0-2: associate with
80:30:dc:15:b0:ee (try 1/3)

[  303.413364] p2p-wlan0-2: associate with
80:30:dc:15:b0:ee (try 2/3)

[  303.441855] p2p-wlan0-2: RX AssocResp from
80:30:dc:15:b0:ee (capab=0x411 status=0 aid=1)

[  303.464848] p2p-wlan0-2: associated

<3>P2P-GROUP-STARTED p2p-wlan[  303.484324]

wlcore: Association completed.
```

10. Run the following q command to exit the wpa_cli tool once the connection is established.

```
> q
```

Once connection is established, a new interface will be created in both the devices.

```
root@esomimx7X:~# ifconfig
p2p-wlan0-2 Link encap:Ethernet  HWaddr
ec:24:b8:0f:46:3b
inet6 addr: fe80::ee24:b8ff:fe0f:463b/64 Scope:Link
UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
RX packets:16 errors:0 dropped:0 overruns:0 frame:0
TX packets:33 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:1000
RX bytes:3340 (3.2 KiB)  TX bytes:6758 (6.5 KiB)
```

11. Run the following command to assign IP address to the newly created interface and enable it.

```
root@esomimx7X:~# ifconfig p2p-wlan0-2 192.168.3.3
```

12. Run the ping command to test the connection.

```
root@esomimx7X:~# ping 192.168.3.4
PING 192.168.3.4 (192.168.3.4): 56 data bytes
64 bytes from 192.168.3.4: seq=0 ttl=64 time=24.391 ms
64 bytes from 192.168.3.4: seq=1 ttl=64 time=25.078 ms
64 bytes from 192.168.3.4: seq=2 ttl=64 time=24.821 ms
```

Note:

- You can use same procedure to connect to an Android device, make sure to assign IP with respect to Android device's subnet range.
- You can test file transfer from Android device using SuperBeam application. You need to use the weblink from SuperBeam in Acacia's browser to download.

Playing Sample Video

iMX7 does not feature a hardware video accelerator but you can use software video decoder to decode and stream the video files. You can use gstreamer pipelines to decode and stream.

Run the following sample gstreamer command to decode and stream a h.264 video.

```
gst-launch-1.0 filesrc location=$VIDEO  ! qtdemux !
h264parse ! avdec_h264 ! videoconvert ! fpsdisplaysink
video-sink=ximagesink text-overlay=false sync=false -v
```

where,

$VIDEO is the location of video file (Format is h.264).

Boot Log

```
U-Boot 2017.03v1.0_rc1+ (Mar 06 2018 - 13:51:17 +0530)

CPU: Freescale i.MX7S rev1.2 at 792MHz
CPU: Extended Commercial temperature grade (-20C to 105C)
at 49C
Reset cause: POR
Model: ESOMIMX7 DUAL BOARD
DRAM: 1 GiB
Board: ESOMIMX7 SOLO
eSOMiMX7 Release version: v1.0_rc1
No fsl,pins property in node hoggrp-lpsr-esomimx7s
PMIC: PFUZE3000 DEV_ID=0x30 REV_ID=0x11
NAND: 2048 MiB
MMC: FSL_SDHC: 0
*** Warning - bad CRC, using default environment

Display: TFT43AB (480x272)
Video: 480x272x24
Warning : bmp logo size is more than screen size
Unable to display logo...
In: serial
Out: serial
Err: serial
Net:
Error: ethernet@30be0000 address not set.
No ethernet found.
Normal Boot
Hit any key to stop autoboot: 0
watchdog enabled

NAND read: device 0 offset 0x4000000, size 0x800000
 8388608 bytes read: OK

NAND read: device 0 offset 0x5000000, size 0x100000
 1048576 bytes read: OK
Kernel image @ 0x80800000 [ 0x000000 - 0x6b2fe8 ]
## Flattened Device Tree blob at 83000000
```

```
 Booting using the fdt blob at 0x83000000
 Using Device Tree in place at 83000000, end 8300c66b


Starting kernel ...


Booting Linux on physical CPU 0x0
Linux version 4.9.11-1.0.0+g1cec328 (sombuild@sombuild-
PC) (gcc version 6.2.0 (GCC) ) #1 SMP PREEMPT Tue Mar 6
12:27:55 IST 2018
CPU: ARMv7 Processor [410fc075] revision 5 (ARMv7),
cr=10c53c7d
CPU: div instructions available: patching division code
CPU: PIPT / VIPT nonaliasing data cache, VIPT aliasing
instruction cache
OF: fdt:Machine model: ESOMIMX7 DUAL BOARD
Memory policy: Data cache writealloc
percpu: Embedded 14 pages/cpu @bf7ce000 s25740 r8192
d23412 u57344
Built 1 zonelists in Zone order, mobility grouping on.
Total pages: 260096
Kernel command line: console=ttymxc0,115200 ubi.mtd=3
root=ubi0:rootfs rootfstype=ubifs mtdparts=gpmi-
nand:64m(boot),16m(kernel),16m(dtb),-(ro)
PID hash table entries: 4096 (order: 2, 16384 bytes)
Dentry cache hash table entries: 131072 (order: 7, 524288
bytes)
Inode-cache hash table entries: 65536 (order: 6, 262144
bytes)
Memory: 1023996K/1048576K available (9216K kernel code,
546K rwdata, 3192K rodata, 1024K init, 464K bss, 24580K
reserved, 0K cma-reserved, 0K )
Virtual kernel memory layout:
 vector : 0xffff0000 - 0xffff1000 ( 4 kB)
 fixmap : 0xffc00000 - 0xfff00000 (3072 kB)
 vmalloc : 0xc0800000 - 0xff800000 (1008 MB)
 lowmem : 0x80000000 - 0xc0000000 (1024 MB)
 pkmap : 0x7fe00000 - 0x80000000 ( 2 MB)
 modules : 0x7f000000 - 0x7fe00000 ( 14 MB)
 .text : 0x80008000 - 0x80a00000 (10208 kB)
 .init : 0x80e00000 - 0x80f00000 (1024 kB)
 .data : 0x80f00000 - 0x80f88860 ( 547 kB)
 .bss : 0x80f8a000 - 0x80ffe060 ( 465 kB)
```

```
SLUB: HWalign=64, Order=0-3, MinObjects=0, CPUs=1,
Nodes=1

Preemptible hierarchical RCU implementation.

 Build-time adjustment of leaf fanout to 32.

 RCU restricting CPUs from NR_CPUS=4 to nr_cpu_ids=1.

RCU: Adjusting geometry for rcu_fanout_leaf=32,
nr_cpu_ids=1

NR_IRQS:16 16

arm_arch_timer: Architected cp15 timer(s) running at
8.00MHz (phys).

clocksource: arch_sys_counter: mask: 0xffffffffffffff
max_cycles: 0x1d854df40, max_idle_ns: 440795202120 ns

sched_clock: 56 bits at 8MHz, resolution 125ns, wraps
every 2199023255500ns

Switching to timer-based delay loop, resolution 125ns

Ignoring duplicate/late registration of
read_current_timer delay

clocksource: mxc_timer1: mask: 0xffffffff max_cycles:
0xffffffff, max_idle_ns: 637086815595 ns

Console: colour dummy device 80x30

Calibrating delay loop (skipped), value calculated using
timer frequency.. 16.00 BogoMIPS (lpj=80000)

pid_max: default: 32768 minimum: 301

Mount-cache hash table entries: 2048 (order: 1, 8192
bytes)

Mountpoint-cache hash table entries: 2048 (order: 1, 8192
bytes)

CPU: Testing write buffer coherency: ok

CPU0: update cpu_capacity 1024

CPU0: thread -1, cpu 0, socket 0, mpidr 80000000

Setting up static identity map for 0x80100000 -
0x80100058

Brought up 1 CPUs

SMP: Total of 1 processors activated (16.00 BogoMIPS).

CPU: All CPU(s) started in SVC mode.

devtmpfs: initialized

VFP support v0.3: implementor 41 architecture 2 part 30
variant 7 rev 5

clocksource: jiffies: mask: 0xffffffff max_cycles:
0xffffffff, max_idle_ns: 19112604462750000 ns

pinctrl core: initialized pinctrl subsystem

NET: Registered protocol family 16
```

```
DMA: preallocated 256 KiB pool for atomic coherent
allocations

cpuidle: using governor menu

hw-breakpoint: found 5 (+1 reserved) breakpoint and 4
watchpoint registers.

hw-breakpoint: maximum watchpoint size is 8 bytes.

imx7d-pinctrl 302c0000.iomuxc-lpsr: Invalid fsl,pins
property in node /soc/aips-bus@30000000/iomuxc-
lpsr@302c0000/imx7d-sdb/hoggrp-lpsr-esomims

imx7d-pinctrl 302c0000.iomuxc-lpsr: initialized IMX
pinctrl driver

imx7d-pinctrl 30330000.iomuxc: initialized IMX pinctrl
driver

mxs-dma 33000000.dma-apbh: initialized

SCSI subsystem initialized

usbcore: registered new interface driver usbfs

usbcore: registered new interface driver hub

usbcore: registered new device driver usb

30800000.aips-bus:usbphynop1 supply vcc not found, using
dummy regulator

i2c i2c-0: IMX I2C adapter registered

i2c i2c-0: can't use DMA, using PIO instead.

i2c i2c-1: IMX I2C adapter registered

i2c i2c-1: can't use DMA, using PIO instead.

i2c i2c-2: IMX I2C adapter registered

i2c i2c-2: can't use DMA, using PIO instead.

Linux video capture interface: v2.00

pps_core: LinuxPPS API ver. 1 registered

pps_core: Software ver. 5.3.6 - Copyright 2005-2007
Rodolfo Giometti <giometti@linux.it>

PTP clock support registered

MIPI CSI2 driver module loaded

imx rpmsg driver is registered.

Advanced Linux Sound Architecture Driver Initialized.

Bluetooth: Core ver 2.22

NET: Registered protocol family 31

Bluetooth: HCI device and connection manager initialized

Bluetooth: HCI socket layer initialized

Bluetooth: L2CAP socket layer initialized

Bluetooth: SCO socket layer initialized
```

```
clocksource: Switched to clocksource arch_sys_counter
VFS: Disk quotas dquot_6.6.0
VFS: Dquot-cache hash table entries: 1024 (order 0, 4096 bytes)
NET: Registered protocol family 2
TCP established hash table entries: 8192 (order: 3, 32768 bytes)
TCP bind hash table entries: 8192 (order: 4, 65536 bytes)
TCP: Hash tables configured (established 8192 bind 8192)
UDP hash table entries: 512 (order: 2, 16384 bytes)
UDP-Lite hash table entries: 512 (order: 2, 16384 bytes)
NET: Registered protocol family 1
RPC: Registered named UNIX socket transport module.
RPC: Registered udp transport module.
RPC: Registered tcp transport module.
RPC: Registered tcp NFSv4.1 backchannel transport module.
Bus freq driver module loaded
futex hash table entries: 256 (order: 2, 16384 bytes)
workingset: timestamp_bits=30 max_order=18 bucket_order=0
NFS: Registering the id_resolver key type
Key type id_resolver registered
Key type id_legacy registered
jffs2: version 2.2. (NAND) © 2001-2006 Red Hat, Inc.
fuse init (API version 7.26)
io scheduler noop registered
io scheduler deadline registered
io scheduler cfq registered (default)
backlight supply power not found, using dummy regulator
30730000.lcdif supply lcd not found, using dummy regulator
mxsfb 30730000.lcdif: failed to find mxc display driver mipi_dsi_samsung
Console: switching to colour frame buffer device 15x15
mxsfb 30730000.lcdif: mxs wait for pan flip timeout
mxsfb 30730000.lcdif: initialized
imx-sdma 30bd0000.sdma: no iram assigned, using external mem
imx-sdma 30bd0000.sdma: loaded firmware 4.2
pfuze100-regulator 0-0008: Full layer: 1, Metal layer: 1
```

```
pfuze100-regulator 0-0008: FAB: 0, FIN: 0

pfuze100-regulator 0-0008: pfuze3000 found.

30860000.serial: ttymxc0 at MMIO 0x30860000 (irq = 53,
base_baud = 1500000) is a IMX

console [ttymxc0] enabled

30a70000.serial: ttymxc4 at MMIO 0x30a70000 (irq = 59,
base_baud = 500000) is a IMX

30a80000.serial: ttymxc5 at MMIO 0x30a80000 (irq = 60,
base_baud = 5000000) is a IMX

30a90000.serial: ttymxc6 at MMIO 0x30a90000 (irq = 61,
base_baud = 1500000) is a IMX

imx sema4 driver is registered.

[drm] Initialized

[drm] Initialized vivante 1.0.0 20120216 on minor 0

brd: module loaded

loop: module loaded

fxos8700 1-001e: read chip ID 0x1 is not equal to 0xc7 or
0xc4

fxos8700: probe of 1-001e failed with error -22

fxas2100x 1-0020: read chip ID 0xfffffffa is not equal to
0xd1 for fxas21000 or 0xd6/0xd7 fxas21002!

fxas2100x: probe of 1-0020 failed with error -22

random: fast init done

nand: device found, Manufacturer ID: 0x2c, Chip ID: 0x48

nand: Micron MT29F16G08ABABAWP

nand: 2048 MiB, SLC, erase size: 512 KiB, page size:
4096, OOB size: 224

gpmi-nand 33002000.gpmi-nand: enable the asynchronous EDO
mode 4

Bad block table found at page 524160, version 0x01

Bad block table found at page 524032, version 0x01

4 cmdlinepart partitions found on MTD device gpmi-nand

Creating 4 MTD partitions on "gpmi-nand":

0x000000000000-0x000004000000 : "boot"

0x000004000000-0x000005000000 : "kernel"

0x000005000000-0x000006000000 : "dtb"

0x000006000000-0x000080000000 : "rootfs"

gpmi-nand 33002000.gpmi-nand: driver registered.

libphy: Fixed MDIO Bus: probed

CAN device driver interface
```

```
30a00000.can supply xceiver not found, using dummy
regulator

flexcan 30a00000.can: device registered
(reg_base=f5a00000, irq=55)

30be0000.ethernet supply phy not found, using dummy
regulator

pps pps0: new PPS source ptp0

fec 30be0000.ethernet (unnamed net_device)
(uninitialized): Invalid MAC address: 00:00:00:00:00:00

fec 30be0000.ethernet (unnamed net_device)
(uninitialized): Using random MAC address:
a2:41:ed:1b:81:1e

libphy: fec_enet_mii_bus: probed

fec 30be0000.ethernet eth0: registered PHC device 0

usbcore: registered new interface driver kaweth

pegasus: v0.9.3 (2013/04/25), Pegasus/Pegasus II USB
Ethernet driver

usbcore: registered new interface driver pegasus

usbcore: registered new interface driver rtl8150

usbcore: registered new interface driver r8152

usbcore: registered new interface driver asix

usbcore: registered new interface driver ax88179_178a

usbcore: registered new interface driver cdc_ether

usbcore: registered new interface driver net1080

usbcore: registered new interface driver cdc_subset

usbcore: registered new interface driver zaurus

usbcore: registered new interface driver cdc_ncm

ehci_hcd: USB 2.0 'Enhanced' Host Controller (EHCI)
Driver

ehci-mxc: Freescale On-Chip EHCI Host driver

usbcore: registered new interface driver usb-storage

usbcore: registered new interface driver usb_ehset_test

30b10200.usbmisc supply vbus-wakeup not found, using
dummy regulator

30b30200.usbmisc supply vbus-wakeup not found, using
dummy regulator

30b30000.usb supply vbus not found, using dummy regulator

ci_hdrc ci_hdrc.1: EHCI Host Controller

ci_hdrc ci_hdrc.1: new USB bus registered, assigned bus
number 1

ci_hdrc ci_hdrc.1: USB 2.0 started, EHCI 1.00
```

```
hub 1-0:1.0: USB hub found
hub 1-0:1.0: 1 port detected
mousedev: PS/2 mouse device common for all mice
input: 30370000.snvs:snvs-powerkey as
/devices/soc0/soc/30000000.aips-
bus/30370000.snvs/30370000.snvs:snvs-
powerkey/input/input0
ft6236 1-0038: failed to read from controller: -6
read mpl3115 chip id 0xffffffa
mpl3115 1-0060: read chip ID 0x1 is not equal to 0xc4!
mpl3115: probe of 1-0060 failed with error -22
rtc-ds1307 1-0068: SET TIME!
rtc-ds1307 1-0068: rtc core: registered ds1338 as rtc0
rtc-ds1307 1-0068: 56 bytes nvram
snvs_rtc 30370000.snvs:snvs-rtc-lp: rtc core: registered
30370000.snvs:snvs- as rtc1
i2c /dev entries driver
IR NEC protocol handler initialized
IR RC5(x/sz) protocol handler initialized
IR RC6 protocol handler initialized
IR JVC protocol handler initialized
IR Sony protocol handler initialized
IR SANYO protocol handler initialized
IR Sharp protocol handler initialized
IR MCE Keyboard/mouse protocol handler initialized
IR XMP protocol handler initialized
usbcore: registered new interface driver uvcvideo
USB Video Class driver (1.1.1)
imx2-wdt 30280000.wdog: timeout 60 sec (nowayout=0)
Bluetooth: HCI UART driver ver 2.3
Bluetooth: HCI UART protocol H4 registered
Bluetooth: HCI UART protocol BCSP registered
Bluetooth: HCI UART protocol ATH3K registered
usbcore: registered new interface driver bcm203x
usbcore: registered new interface driver btusb
usbcore: registered new interface driver ath3k
sdhci: Secure Digital Host Controller Interface driver
sdhci: Copyright(c) Pierre Ossman
sdhci-pltfm: SDHCI platform and OF driver helper
```

```
sdhci-esdhc-imx 30b40000.usdhc: Got CD GPIO

mmc2: SDHCI controller on 30b40000.usdhc [30b40000.usdhc]
using ADMA

sdhci-esdhc-imx 30b50000.usdhc: could not get ultra high
speed state, work on normal mode

usb 1-1: new high-speed USB device number 2 using ci_hdrc

mmc1: SDHCI controller on 30b50000.usdhc [30b50000.usdhc]
using ADMA

usbcore: registered new interface driver usbhid

usbhid: USB HID core driver

sdhci-esdhc-imx 30b50000.usdhc: card claims to support
voltages below defined range

input: ST LSM6DS3 Accelerometer Sensor as
/devices/soc0/soc/30800000.aips-bus/30a40000.i2c/i2c-2/2-
006a/input/input1

input: ST LSM6DS3 Gyroscope Sensor as
/devices/soc0/soc/30800000.aips-bus/30a40000.i2c/i2c-2/2-
006a/input/input2

input: ST LSM6DS3 Significant Motion Sensor as
/devices/soc0/soc/30800000.aips-bus/30a40000.i2c/i2c-2/2-
006a/input/input3

input: ST LSM6DS3 Step Counter Sensor as
/devices/soc0/soc/30800000.aips-bus/30a40000.i2c/i2c-2/2-
006a/input/input4

input: ST LSM6DS3 Step Detector Sensor as
/devices/soc0/soc/30800000.aips-bus/30a40000.i2c/i2c-2/2-
006a/input/input5

input: ST LSM6DS3 Tilt Sensor as
/devices/soc0/soc/30800000.aips-bus/30a40000.i2c/i2c-2/2-
006a/input/input6

mmc1: new high speed SDIO card at address 0001

lsm6ds3 2-006a: lsm6ds3: probed

imx_wm8750_probe

imx_wm8750_probe snd_soc_register_card

imx-wm8750 sound: wm8750-hifi <-> 308b0000.sai mapping ok

imx-wm8750 sound: ASoC: no source widget found for Main
MIC

imx-wm8750 sound: ASoC: Failed to add route Main MIC ->
direct -> RINPUT1

imx_wm8750_late_probe codec_dai name is wm8750-hifi

hub 1-1:1.0: USB hub found

hub 1-1:1.0: 2 ports detected

imx_wm8750_set_bias_level SND_SOC_BIAS_STANDBY
```

```
NET: Registered protocol family 26
NET: Registered protocol family 10
sit: IPv6, IPv4 and MPLS over IPv4 tunneling driver
NET: Registered protocol family 17
can: controller area network core (rev 20120528 abi 9)
NET: Registered protocol family 29
can: raw protocol (rev 20120528)
can: broadcast manager protocol (rev 20161123 t)
can: netlink gateway (rev 20130117) max_hops=1
Bluetooth: RFCOMM TTY layer initialized
Bluetooth: RFCOMM socket layer initialized
Bluetooth: RFCOMM ver 1.11
Bluetooth: BNEP (Ethernet Emulation) ver 1.3
Bluetooth: BNEP filters: protocol multicast
Bluetooth: BNEP socket layer initialized
Bluetooth: HIDP (Human Interface Emulation) ver 1.2
Bluetooth: HIDP socket layer initialized
8021q: 802.1Q VLAN Support v1.8
Key type dns_resolver registered
cpu cpu0: dev_pm_opp_get_opp_count: OPP table not found
(-19)
cpu cpu0: failed to init OPP table: -19
ubi0: attaching mtd3
random: crng init done
ubi0: scanning is finished
ubi0: attached mtd3 (name "rootfs", size 1952 MiB)
ubi0: PEB size: 524288 bytes (512 KiB), LEB size: 516096
bytes
ubi0: min./max. I/O unit sizes: 4096/4096, sub-page size
4096
ubi0: VID header offset: 4096 (aligned 4096), data
offset: 8192
ubi0: good PEBs: 3900, bad PEBs: 4, corrupted PEBs: 0
ubi0: user volume: 1, internal volumes: 1, max. volumes
count: 128
ubi0: max/mean erase counter: 1/0, WL threshold: 4096,
image sequence number: 1716550765
ubi0: available PEBs: 0, total reserved PEBs: 3900, PEBs
reserved for bad PEB handling: 76
ubi0: background thread "ubi_bgt0d" started, PID 116
```

```
input: gpio-keys as /devices/soc0/gpio-keys/input/input8

rtc-ds1307 1-0068: setting system clock to 2000-01-01
00:00:09 UTC (946684809)

wlreg_on: disabling

usb_otg1_vbus: disabling

usb_otg2_vbus: disabling

VDD_SD1: disabling

mipi_dsi_pwr_on: disabling

VLDO2: disabling

ALSA device list:

 #0: wm8750-audio

UBIFS (ubi0:0): UBIFS: mounted UBI device 0, volume 0,
name "rootfs", R/O mode

UBIFS (ubi0:0): LEB size: 516096 bytes (504 KiB),
min./max. I/O unit sizes: 4096 bytes/4096 bytes

UBIFS (ubi0:0): FS size: 1965809664 bytes (1874 MiB, 3809
LEBs), journal size 33546240 bytes (31 MiB, 65 LEBs)

UBIFS (ubi0:0): reserved for root: 4952683 bytes (4836
KiB)

UBIFS (ubi0:0): media format: w4/r0 (latest is w4/r0),
UUID 50824624-3C3E-4217-BAB4-4AC101C9D9BB, small LPT
model

VFS: Mounted root (ubifs filesystem) readonly on device
0:13.

devtmpfs: mounted

Freeing unused kernel memory: 1024K (80e00000 - 80f00000)

systemd[1]: System time before build time, advancing
clock.

systemd[1]: systemd 230 running in system mode. (-PAM -
AUDIT -SELINUX +IMA -APPARMOR +SMACK +SYSVINIT +UTMP -
LIBCRYPTSETUP -GCRYPT -GNUTLS +AC)

systemd[1]: Detected architecture arm.


Welcome to NXP i.MX Release Distro 4.9.11-1.0.0 (morty)!


systemd[1]: Set hostname to <esomimx7s-1gb>.

systemd[1]: Initializing machine ID from random
generator.

systemd[1]: Installed transient /etc/machine-id file.

systemd[1]: connman.service: Cannot add dependency job,
ignoring: Unit connman.service is masked.
```

```
systemd[1]: sysinit.target: Found ordering cycle on
sysinit.target/start

systemd[1]: sysinit.target: Found dependency on
alignment.service/start

systemd[1]: sysinit.target: Found dependency on
basic.target/start

systemd[1]: sysinit.target: Found dependency on
sockets.target/start

[ SKIP ] Ordering cycle found, skipping alignment.service

[ OK ] Listening on udev Kernel Socket.

[ OK ] Started Dispatch Password Requests to Console
Directory Watch.

[ OK ] Reached target Swap.

[ OK ] Listening on Journal Socket.

[ OK ] Listening on Syslog Socket.

[ OK ] Created slice System Slice.

 Mounting Debug File System...

 Starting Load Kernel Modules...

[ OK ] Listening on Journal Socket (/dev/log).

[ OK ] Created slice system-getty.slice.

 Mounting Temporary Directory...

 Starting Setup Virtual Console...

 Starting Journal Service...

[ OK ] Created slice system-serial\x2dgetty.slice.

[ OK ] Created slice User and Session Slice.

[ OK ] Started Forward Password Requests to Wall
Directory Watch.

[ OK ] Reached target Paths.

[ OK ] Reached target Slices.

 Starting File System Check on Root Device...

[ OK ] Listening on udev Control Socket.

[ OK ] Listening on /dev/initctl Compatibility Named
Pipe.

[ OK ] Reached target Remote File Systems.

[ OK ] Mounted Debug File System.

[ OK ] Mounted Temporary Directory.

[ OK ] Started Journal Service.

[FAILED] Failed to start Load Kernel Modules.

See 'systemctl status systemd-modules-load.service' for
details.
```

```
[ OK ] Started Setup Virtual Console.
[ OK ] Started File System Check on Root Device.
 Starting Remount Root and Kernel File Systems...
 Starting Apply Kernel Variables...
 Mounting FUSE Control File System...
UBIFS (ubi0:0): background thread "ubifs_bgt0_0" started,
PID 146
 Mounting NFSD configuration filesystem...
[ OK ] Mounted FUSE Control File System.
[ OK ] Started Remount Root and Kernel File Systems.
[ OK ] Started Apply Kernel Variables.
[FAILED] Failed to mount NFSD configuration filesystem.
See 'systemctl status proc-fs-nfsd.mount' for details.
 Starting Rebuild Hardware Database...
 Starting Flush Journal to Persistent Storage...
 Starting Create System Users...
systemd-journald[136]: Received request to flush runtime
journal from PID 1
[ OK ] Started Create System Users.
[ OK ] Started Flush Journal to Persistent Storage.
 Starting Create Static Device Nodes in /dev...
[ OK ] Started Create Static Device Nodes in /dev.
[ OK ] Reached target Local File Systems (Pre).
 Mounting /var/volatile...
 Starting udev Kernel Device Manager...
[ OK ] Mounted /var/volatile.
 Starting Load/Save Random Seed...
[ OK ] Reached target Local File Systems.
 Starting Rebuild Dynamic Linker Cache...
 Starting Create Volatile Files and Directories...
 Starting Commit a transient machine-id on disk...
 Starting Rebuild Journal Catalog...
[ OK ] Started Load/Save Random Seed.
[ OK ] Started Commit a transient machine-id on disk.
[ OK ] Started udev Kernel Device Manager.
[ OK ] Started Create Volatile Files and Directories.
 Starting Network Time Synchronization...
 Starting Update UTMP about System Boot/Shutdown...
```

```
[ OK ] Started Rebuild Journal Catalog.
[ OK ] Started Network Time Synchronization.
[ OK ] Started Update UTMP about System Boot/Shutdown.
[ OK ] Reached target System Time Synchronized.
[ OK ] Started Rebuild Dynamic Linker Cache.
[ OK ] Started Rebuild Hardware Database.
 Starting Update is Completed...
 Starting udev Coldplug all Devices...
[ OK ] Started Update is Completed.
[ OK ] Created slice system-systemd\x2dbacklight.slice.
 Starting Load/Save Screen Backlight...htness of
backlight:backlight...
[ OK ] Started udev Coldplug all Devices.
 Starting Start Psplash Boot Screen...
[ OK ] Started Load/Save Screen Backlight Brightness of
backlight:backlight.
[ OK ] Reached target System Initialization.
[ OK ] Listening on D-Bus System Message Bus Socket.
[ OK ] Listening on RPCbind Server Activation Socket.
[ OK ] Listening on dropbear.socket.
[ OK ] Listening on Avahi mDNS/DNS-SD Stack Activation
Socket.
[ OK ] Reached target Sockets.
[ OK ] Reached target Basic System.
[ OK ] Started D-Bus System Message Bus.
mxc_mipi-csi 30750000.mipi-csi: mipi csi v4l2 device
registered
CSI: Registered sensor subdevice: mxc_mipi-csi.0
mxc_mipi-csi 30750000.mipi-csi: lanes: 2, hs_settle: 13,
clk_settle: 2, wclk: 1, freq: 240000000
ov8865_mipi: loading out-of-tree module taints kernel.
1-0024 supply DOVDD not found, using dummy regulator
1-0024 supply DVDD not found, using dummy regulator
ov8865_write_reg:write reg error:reg=fffd,val=80
camera ov8865 init failed
ov8865_mipi: probe of 1-0024 failed with error -1



 power enable gpio is 139
```

```
 Starting Install MySQL Community Server Database...
[ OK ] Started Kernel Logging Service.
[ OK ] Started Job spooling tools.
[ OK ] Started Xserver startup with a display manager.
 Starting Avahi mDNS/DNS-SD Stack...
 Starting /etc/rc.local Compatibility...
[ OK ] Started Periodic Command Scheduler.
 Starting Network Time Service (one-shot ntpdate mode)...
 Starting Save/Restore Sound Card State...
[ OK ] Started System Logging Service.
[ OK ] Started Daily Cleanup of Temporary Directories.
[ OK ] Reached target Timers.
 Starting Telephony service...
 Starting Login Service...
 Starting Console System Startup Logging...
[ OK ] Started Start Psplash Boot Screen.
[ OK ] Started /etc/rc.local Compatibility.
[ OK ] Started Network Time Service (one-shot ntpdate
mode).
[ OK ] Found device /dev/ttymxc0.
[ OK ] Started Console System Startup Logging.
[ OK ] Started Avahi mDNS/DNS-SD Stack.
[ OK ] Started Telephony service.
[ OK ] Reached target Sound Card.
[ OK ] Started Getty on tty1.
[ OK ] Started Serial Getty on ttymxc0.
[ OK ] Reached target Login Prompts.
 Starting Terminate Psplash Boot Screen...
[ OK ] Started Updates psplash to basic.
[ OK ] Started Terminate Psplash Boot Screen.
[ OK ] Started Login Service.
wlcore: wl18xx HW: 183x or 180x, PG 2.2 (ROM 0x11)
[ OK ] Started Save/Rwlcore: loaded
estore Sound Card State.
mxsfb 30730000.lcdif: mxs wait for pan flip timeout
```

```
NXP i.MX Release Distro 4.9.11-1.0.0 esomimx7s-1gb
ttymxc0


esomimx7s-1gb login:
```

Dual SOM Complete Boot Log

```
U-Boot 2017.03v1.0_rc1+ (Mar 05 2018 - 22:50:30 +0530)

CPU: Freescale i.MX7D rev1.2 996 MHz (running at 792 MHz)
CPU: Commercial temperature grade (0C to 95C) at 43C
Reset cause: POR
Model: ESOMIMX7 DUAL BOARD
DRAM: 1 GiB
Board: ESOMIMX7 DUAL
eSOMiMX7 Release version: v1.0_rc1
No fsl,pins property in node hoggrp-lpsr-esomimx7s
Invalid fsl,pins property in node hoggrp-esomimx7d
PMIC: PFUZE3000 DEV_ID=0x30 REV_ID=0x11
MMC: FSL_SDHC: 0, FSL_SDHC: 1
Display: TFT43AB (480x272)
Video: 480x272x24
Warning : bmp logo size is more than screen size
Unable to display logo...
In: serial
Out: serial
Err: serial
Net: eth0: ethernet@30be0000 [PRIME], eth1:
ethernet@30bf0000
Normal Boot
Hit any key to stop autoboot: 0
watchdog enabled
switch to partitions #0, OK
mmc1 is current device
switch to partitions #0, OK
mmc1 is current device
reading boot.scr
** Unable to read file boot.scr **
reading zImage
```

```
7293672 bytes read in 324 ms (21.5 MiB/s)
Booting from mmc ...
reading imx7d-acacia.dtb
48177 bytes read in 20 ms (2.3 MiB/s)
Kernel image @ 0x80800000 [ 0x000000 - 0x6f4ae8 ]
## Flattened Device Tree blob at 83000000
 Booting using the fdt blob at 0x83000000
 Using Device Tree in place at 83000000, end 8300ec30

Starting kernel ...

Booting Linux on physical CPU 0x0
Linux version 4.9.11-1.0.0+g1cec328 (sombuild@sombuild-
PC) (gcc version 6.2.0 (GCC) ) #1 SMP PREEMPT Mon Mar 5
21:23:08 IST 2018
CPU: ARMv7 Processor [410fc075] revision 5 (ARMv7),
cr=10c53c7d
CPU: div instructions available: patching division code
CPU: PIPT / VIPT nonaliasing data cache, VIPT aliasing
instruction cache
OF: fdt:Machine model: ESOMIMX7 DUAL BOARD
Reserved memory: created CMA memory pool at 0xac000000,
size 320 MiB
OF: reserved mem: initialized node linux,cma, compatible
id shared-dma-pool
Memory policy: Data cache writealloc
percpu: Embedded 14 pages/cpu @ab71a000 s25868 r8192
d23284 u57344
Built 1 zonelists in Zone order, mobility grouping on.
Total pages: 260096
Kernel command line: console=ttymxc0,115200
root=/dev/mmcblk2p2 rootwait rw
PID hash table entries: 4096 (order: 2, 16384 bytes)
Dentry cache hash table entries: 131072 (order: 7, 524288
bytes)
Inode-cache hash table entries: 65536 (order: 6, 262144
bytes)
Memory: 695584K/1048576K available (9216K kernel code,
548K rwdata, 3352K rodata, 1024K init, 465K bss, 25312K
reserved, 327680K cma-reserved,)
Virtual kernel memory layout:
 vector : 0xffff0000 - 0xffff1000 ( 4 kB)
```

```
 fixmap : 0xffc00000 - 0xfff00000 (3072 kB)
 vmalloc : 0xc0800000 - 0xff800000 (1008 MB)
 lowmem : 0x80000000 - 0xc0000000 (1024 MB)
 pkmap : 0x7fe00000 - 0x80000000 ( 2 MB)
 modules : 0x7f000000 - 0x7fe00000 ( 14 MB)
 .text : 0x80008000 - 0x80a00000 (10208 kB)
 .init : 0x80e00000 - 0x80f00000 (1024 kB)
 .data : 0x80f00000 - 0x80f89260 ( 549 kB)
 .bss : 0x80f8b000 - 0x80fff620 ( 466 kB)
SLUB: HWalign=64, Order=0-3, MinObjects=0, CPUs=2,
Nodes=1
Preemptible hierarchical RCU implementation.
 Build-time adjustment of leaf fanout to 32.
 RCU restricting CPUs from NR_CPUS=4 to nr_cpu_ids=2.
RCU: Adjusting geometry for rcu_fanout_leaf=32,
nr_cpu_ids=2
NR_IRQS:16 16
arm_arch_timer: Architected cp15 timer(s) running at
8.00MHz (phys).
clocksource: arch_sys_counter: mask: 0xffffffffffffff
max_cycles: 0x1d854df40, max_idle_ns: 440795202120 ns
sched_clock: 56 bits at 8MHz, resolution 125ns, wraps
every 2199023255500ns
Switching to timer-based delay loop, resolution 125ns
Ignoring duplicate/late registration of
read_current_timer delay
clocksource: mxc_timer1: mask: 0xffffffff max_cycles:
0xffffffff, max_idle_ns: 637086815595 ns
Console: colour dummy device 80x30
Calibrating delay loop (skipped), value calculated using
timer frequency.. 16.00 BogoMIPS (lpj=80000)
pid_max: default: 32768 minimum: 301
Mount-cache hash table entries: 2048 (order: 1, 8192
bytes)
Mountpoint-cache hash table entries: 2048 (order: 1, 8192
bytes)
CPU: Testing write buffer coherency: ok
CPU0: update cpu_capacity 1024
CPU0: thread -1, cpu 0, socket 0, mpidr 80000000
Setting up static identity map for 0x80100000 -
0x80100058
```

```
CPU1: update cpu_capacity 1024

CPU1: thread -1, cpu 1, socket 0, mpidr 80000001

Brought up 2 CPUs

SMP: Total of 2 processors activated (32.00 BogoMIPS).

CPU: All CPU(s) started in SVC mode.

devtmpfs: initialized

VFP support v0.3: implementor 41 architecture 2 part 30
variant 7 rev 5

clocksource: jiffies: mask: 0xffffffff max_cycles:
0xffffffff, max_idle_ns: 19112604462750000 ns

pinctrl core: initialized pinctrl subsystem

NET: Registered protocol family 16

DMA: preallocated 256 KiB pool for atomic coherent
allocations

cpuidle: using governor menu

DDR type is LPDDR3!

hw-breakpoint: found 5 (+1 reserved) breakpoint and 4
watchpoint registers.

hw-breakpoint: maximum watchpoint size is 8 bytes.

imx7d-pinctrl 302c0000.iomuxc-lpsr: Invalid fsl,pins
property in node /soc/aips-bus@30000000/iomuxc-
lpsr@302c0000/imx7d-sdb/hoggrp-lpsr-esomims

imx7d-pinctrl 302c0000.iomuxc-lpsr: Invalid fsl,pins
property in node /soc/aips-bus@30000000/iomuxc-
lpsr@302c0000/imx7d-sdb/hoggrp-lpsr-acaciad

imx7d-pinctrl 302c0000.iomuxc-lpsr: initialized IMX
pinctrl driver

imx7d-pinctrl 30330000.iomuxc: Invalid fsl,pins property
in node /soc/aips-bus@30000000/iomuxc@30330000/imx7d-
sdb/hoggrp-esomimx7d

imx7d-pinctrl 30330000.iomuxc: initialized IMX pinctrl
driver

MU is ready for cross core communication!

mxs-dma 33000000.dma-apbh: initialized

vgaarb: loaded

SCSI subsystem initialized

usbcore: registered new interface driver usbfs

usbcore: registered new interface driver hub

usbcore: registered new device driver usb

30800000.aips-bus:usbphynop1 supply vcc not found, using
dummy regulator
```

```
30800000.aips-bus:usbphynop2 supply vcc not found, using
dummy regulator
i2c i2c-0: IMX I2C adapter registered
i2c i2c-0: can't use DMA, using PIO instead.
i2c i2c-1: IMX I2C adapter registered
i2c i2c-1: can't use DMA, using PIO instead.
i2c i2c-2: IMX I2C adapter registered
i2c i2c-2: can't use DMA, using PIO instead.
Linux video capture interface: v2.00
pps_core: LinuxPPS API ver. 1 registered
pps_core: Software ver. 5.3.6 - Copyright 2005-2007
Rodolfo Giometti <giometti@linux.it>
PTP clock support registered
MIPI CSI2 driver module loaded
imx rpmsg driver is registered.
Advanced Linux Sound Architecture Driver Initialized.
Bluetooth: Core ver 2.22
NET: Registered protocol family 31
Bluetooth: HCI device and connection manager initialized
Bluetooth: HCI socket layer initialized
Bluetooth: L2CAP socket layer initialized
Bluetooth: SCO socket layer initialized
clocksource: Switched to clocksource arch_sys_counter
VFS: Disk quotas dquot_6.6.0
VFS: Dquot-cache hash table entries: 1024 (order 0, 4096
bytes)
NET: Registered protocol family 2
TCP established hash table entries: 8192 (order: 3, 32768
bytes)
TCP bind hash table entries: 8192 (order: 4, 65536 bytes)
TCP: Hash tables configured (established 8192 bind 8192)
UDP hash table entries: 512 (order: 2, 16384 bytes)
UDP-Lite hash table entries: 512 (order: 2, 16384 bytes)
NET: Registered protocol family 1
RPC: Registered named UNIX socket transport module.
RPC: Registered udp transport module.
RPC: Registered tcp transport module.
RPC: Registered tcp NFSv4.1 backchannel transport module.
Bus freq driver module loaded
```

```
futex hash table entries: 512 (order: 3, 32768 bytes)
workingset: timestamp_bits=30 max_order=18 bucket_order=0
NFS: Registering the id_resolver key type
Key type id_resolver registered
Key type id_legacy registered
jffs2: version 2.2. (NAND) © 2001-2006 Red Hat, Inc.
fuse init (API version 7.26)
io scheduler noop registered
io scheduler deadline registered
io scheduler cfq registered (default)
OF: PCI: host bridge /soc/pcie@0x33800000 ranges:
OF: PCI: No bus range found for /soc/pcie@0x33800000,
using [bus 00-ff]
OF: PCI: IO 0x4ff80000..0x4ff8ffff -> 0x00000000
OF: PCI: MEM 0x40000000..0x4feffff -> 0x40000000
imx6q-pcie 33800000.pcie: phy link never came up
imx6q-pcie 33800000.pcie: Link never came up
imx6q-pcie 33800000.pcie: failed to initialize host
imx6q-pcie: probe of 33800000.pcie failed with error -110
backlight supply power not found, using dummy regulator
ADV7511 probing phase
random: fast init done
adv7511 1-0039: ADV7511: read edid fail
30730000.lcdif supply lcd not found, using dummy
regulator
mxsfb 30730000.lcdif: failed to find mxc display driver
Console: switching to colour frame buffer device 160x45
mxsfb 30730000.lcdif: initialized
imx-sdma 30bd0000.sdma: no iram assigned, using external
mem
imx-sdma 30bd0000.sdma: loaded firmware 4.2
pfuze100-regulator 0-0008: Full layer: 1, Metal layer: 1
pfuze100-regulator 0-0008: FAB: 0, FIN: 0
pfuze100-regulator 0-0008: pfuze3000 found.
30860000.serial: ttymxc0 at MMIO 0x30860000 (irq = 52,
base_baud = 1500000) is a IMX
console [ttymxc0] enabled
30a70000.serial: ttymxc4 at MMIO 0x30a70000 (irq = 58,
base_baud = 500000) is a IMX
```

```
30a80000.serial: ttymxc5 at MMIO 0x30a80000 (irq = 59,
base_baud = 5000000) is a IMX

30a90000.serial: ttymxc6 at MMIO 0x30a90000 (irq = 60,
base_baud = 1500000) is a IMX

imx sema4 driver is registered.

[drm] Initialized

[drm] Initialized vivante 1.0.0 20120216 on minor 0

brd: module loaded

loop: module loaded

fxos8700 1-001e: read chip ID 0x1 is not equal to 0xc7 or
0xc4

fxos8700: probe of 1-001e failed with error -22

fxas2100x 1-0020: read chip ID 0xfffffffa is not equal to
0xd1 for fxas21000 or 0xd6/0xd7 fxas21002!

fxas2100x: probe of 1-0020 failed with error -22

libphy: Fixed MDIO Bus: probed

CAN device driver interface

30a00000.can supply xceiver not found, using dummy
regulator

flexcan 30a00000.can: device registered
(reg_base=f5a00000, irq=54)

30be0000.ethernet supply phy not found, using dummy
regulator

pps pps0: new PPS source ptp0

libphy: fec_enet_mii_bus: probed

fec 30be0000.ethernet eth0: registered PHC device 0

30bf0000.ethernet supply phy not found, using dummy
regulator

pps pps1: new PPS source ptp1

libphy: fec_enet_mii_bus: probed

fec 30bf0000.ethernet eth1: registered PHC device 1

usbcore: registered new interface driver kaweth

pegasus: v0.9.3 (2013/04/25), Pegasus/Pegasus II USB
Ethernet driver

usbcore: registered new interface driver pegasus

usbcore: registered new interface driver rtl8150

usbcore: registered new interface driver r8152

usbcore: registered new interface driver asix

usbcore: registered new interface driver ax88179_178a

usbcore: registered new interface driver cdc_ether

usbcore: registered new interface driver net1080
```

```
usbcore: registered new interface driver cdc_subset
usbcore: registered new interface driver zaurus
usbcore: registered new interface driver cdc_ncm
usbcore: registered new interface driver cdc_mbim
ehci_hcd: USB 2.0 'Enhanced' Host Controller (EHCI)
Driver
ehci-pci: EHCI PCI platform driver
ehci-mxc: Freescale On-Chip EHCI Host driver
usbcore: registered new interface driver cdc_acm
cdc_acm: USB Abstract Control Model driver for USB modems
and ISDN adapters
usbcore: registered new interface driver cdc_wdm
usbcore: registered new interface driver usb-storage
usbcore: registered new interface driver usbserial
usbcore: registered new interface driver
usbserial_generic
usbserial: USB Serial support registered for generic
usbcore: registered new interface driver option
usbserial: USB Serial support registered for GSM modem
(1-port)
usbcore: registered new interface driver usb_ehset_test
30b10200.usbmisc supply vbus-wakeup not found, using
dummy regulator
30b30200.usbmisc supply vbus-wakeup not found, using
dummy regulator
30b20200.usbmisc supply vbus-wakeup not found, using
dummy regulator
30b30000.usb supply vbus not found, using dummy regulator
ci_hdrc ci_hdrc.1: EHCI Host Controller
ci_hdrc ci_hdrc.1: new USB bus registered, assigned bus
number 1
ci_hdrc ci_hdrc.1: USB 2.0 started, EHCI 1.00
hub 1-0:1.0: USB hub found
hub 1-0:1.0: 1 port detected
mousedev: PS/2 mouse device common for all mice
input: 30370000.snvs:snvs-powerkey as
/devices/soc0/soc/30000000.aips-
bus/30370000.snvs/30370000.snvs:snvs-
powerkey/input/input0
ft6236 1-0038: failed to read from controller: -6
read mpl3115 chip id 0xfffffffa
```



```
mpl3115 1-0060: read chip ID 0x1 is not equal to 0xc4!
mpl3115: probe of 1-0060 failed with error -22
rtc-ds1307 1-0068: rtc core: registered ds1338 as rtc0
rtc-ds1307 1-0068: 56 bytes nvram
snvs_rtc 30370000.snvs:snvs-rtc-lp: rtc core: registered
30370000.snvs:snvs- as rtc1
i2c /dev entries driver
IR NEC protocol handler initialized
IR RC5(x/sz) protocol handler initialized
IR RC6 protocol handler initialized
IR JVC protocol handler initialized
IR Sony protocol handler initialized
IR SANYO protocol handler initialized
IR Sharp protocol handler initialized
IR MCE Keyboard/mouse protocol handler initialized
IR XMP protocol handler initialized
usbcore: registered new interface driver uvcvideo
USB Video Class driver (1.1.1)
imx2-wdt 30280000.wdog: timeout 60 sec (nowayout=0)
Bluetooth: HCI UART driver ver 2.3
Bluetooth: HCI UART protocol H4 registered
Bluetooth: HCI UART protocol BCSP registered
Bluetooth: HCI UART protocol LL registered
Bluetooth: HCI UART protocol ATH3K registered
usbcore: registered new interface driver bcm203x
usbcore: registered new interface driver btusb
usbcore: registered new interface driver ath3k
sdhci: Secure Digital Host Controller Interface driver
sdhci: Copyright(c) Pierre Ossman
sdhci-pltfm: SDHCI platform and OF driver helper
sdhci-esdhc-imx 30b40000.usdhc: Got CD GPIO
mmc2: SDHCI controller on 30b40000.usdhc [30b40000.usdhc]
using ADMA
sdhci-esdhc-imx 30b50000.usdhc: could not get ultra high
speed state, work on normal mode
usb 1-1: new high-speed USB device number 2 using ci_hdrc
mmc2: host does not support reading read-only switch,
assuming write-enable
```

```
mmc1: SDHCI controller on 30b50000.usdhc [30b50000.usdhc]
using ADMA
mmc2: new high speed SDHC card at address aaaa
mmcblk2: mmc2:aaaa SS08G 7.40 GiB
 mmcblk2: p1 p2
mmc0: SDHCI controller on 30b60000.usdhc [30b60000.usdhc]
using ADMA
caam 30900000.caam: ERA source: CCBVID.
caam 30900000.caam: Entropy delay = 3200
caam 30900000.caam: Instantiated RNG4 SH0
hub 1-1:1.0: USB hub found
hub 1-1:1.0: 2 ports detected
caam 30900000.caam: Instantiated RNG4 SH1
caam 30900000.caam: device ID = 0x0a16030000000000 (Era
8)
caam 30900000.caam: job rings = 3, qi = 0
mmc0: new HS400 MMC card at address 0001
mmcblk0: mmc0:0001 W62704 3.56 GiB
mmcblk0boot0: mmc0:0001 W62704 partition 1 2.00 MiB
mmcblk0boot1: mmc0:0001 W62704 partition 2 2.00 MiB
mmcblk0rpmb: mmc0:0001 W62704 partition 3 512 KiB
 mmcblk0: p1 p2
caam algorithms registered in /proc/crypto
caam_jr 30901000.jr0: registering rng-caam
caam 30900000.caam: caam pkc algorithms registered in
/proc/crypto
platform caam_sm: blkkey_ex: 8 keystore units available
platform caam_sm: 64-bit clear key:
platform caam_sm: [0000] 00 01 02 03 04 0f 06 07
platform caam_sm: 64-bit black key:
platform caam_sm: [0000] 59 c9 d6 bb 50 eb 41 8e
platform caam_sm: [0008] c4 40 47 90 04 27 b6 60
platform caam_sm: 128-bit clear key:
platform caam_sm: [0000] 00 01 02 03 04 0f 06 07
platform caam_sm: [0008] 08 09 0a 0b 0c 0d 0e 0f
platform caam_sm: 128-bit black key:
platform caam_sm: [0000] 01 36 a0 18 6e 95 66 e7
platform caam_sm: [0008] 76 ef 79 32 38 a3 72 df
platform caam_sm: 192-bit clear key:
```

```
platform caam_sm: [0000] 00 01 02 03 04 0f 06 07
platform caam_sm: [0008] 08 09 0a 0b 0c 0d 0e 0f
platform caam_sm: [0016] 10 11 12 13 14 15 16 17
platform caam_sm: 192-bit black key:
platform caam_sm: [0000] be 92 40 57 8a ba fb 35
platform caam_sm: [0008] 0d f6 e6 5e 3e 37 85 2c
platform caam_sm: [0016] 69 a6 58 fb 77 eb 0c 0f
platform caam_sm: [0024] 4d c2 33 30 d3 57 be f5
platform caam_sm: 256-bit clear key:
platform caam_sm: [0000] 00 01 02 03 04 0f 06 07
platform caam_sm: [0008] 08 09 0a 0b 0c 0d 0e 0f
platform caam_sm: [0016] 10 11 12 13 14 15 16 17
platform caam_sm: [0024] 18 19 1a 1b 1c 1d 1e 1f
platform caam_sm: 256-bit black key:
platform caam_sm: [0000] 6d 80 57 c0 09 41 ef 3c
platform caam_sm: [0008] be ae 26 c3 a4 cb 08 0b
platform caam_sm: [0016] d1 f1 71 91 b5 fb 55 bc
platform caam_sm: [0024] dc 7b 8c 26 c7 12 61 74
usb 1-1.2: new high-speed USB device number 3 using
ci_hdrc
platform caam_sm: 64-bit unwritten blob:
platform caam_sm: [0000] 00 00 00 00 00 00 00 00
platform caam_sm: [0008] 00 00 00 00 00 00 00 00
platform caam_sm: [0016] 00 00 00 00 00 00 00 00
platform caam_sm: [0024] 00 00 00 00 00 00 00 00
platform caam_sm: [0032] 00 00 00 00 00 00 00 00
platform caam_sm: [0040] 00 00 00 00 00 00 00 00
platform caam_sm: [0048] 00 00 00 00 00 00 00 00
platform caam_sm: [0056] 00 00 00 00 00 00 00 00
platform caam_sm: [0064] 00 00 00 00 00 00 00 00
platform caam_sm: [0072] 00 00 00 00 00 00 00 00
platform caam_sm: [0080] 00 00 00 00 00 00 00 00
platform caam_sm: [0088] 00 00 00 00 00 00 00 00
platform caam_sm: 128-bit unwritten blob:
platform caam_sm: [0000] 00 00 00 00 00 00 00 00
platform caam_sm: [0008] 00 00 00 00 00 00 00 00
platform caam_sm: [0016] 00 00 00 00 00 00 00 00
platform caam_sm: [0024] 00 00 00 00 00 00 00 00
```

```
platform caam_sm: [0032] 00 00 00 00 00 00 00 00
platform caam_sm: [0040] 00 00 00 00 00 00 00 00
platform caam_sm: [0048] 00 00 00 00 00 00 00 00
platform caam_sm: [0056] 00 00 00 00 00 00 00 00
platform caam_sm: [0064] 00 00 00 00 00 00 00 00
platform caam_sm: [0072] 00 00 00 00 00 00 00 00
platform caam_sm: [0080] 00 00 00 00 00 00 00 00
platform caam_sm: [0088] 00 00 00 00 00 00 00 00
platform caam_sm: 196-bit unwritten blob:
platform caam_sm: [0000] 00 00 00 00 00 00 00 00
platform caam_sm: [0008] 00 00 00 00 00 00 00 00
platform caam_sm: [0016] 00 00 00 00 00 00 00 00
platform caam_sm: [0024] 00 00 00 00 00 00 00 00
platform caam_sm: [0032] 00 00 00 00 00 00 00 00
platform caam_sm: [0040] 00 00 00 00 00 00 00 00
platform caam_sm: [0048] 00 00 00 00 00 00 00 00
platform caam_sm: [0056] 00 00 00 00 00 00 00 00
platform caam_sm: [0064] 00 00 00 00 00 00 00 00
platform caam_sm: [0072] 00 00 00 00 00 00 00 00
platform caam_sm: [0080] 00 00 00 00 00 00 00 00
platform caam_sm: [0088] 00 00 00 00 00 00 00 00
platform caam_sm: 256-bit unwritten blob:
platform caam_sm: [0000] 00 00 00 00 00 00 00 00
platform caam_sm: [0008] 00 00 00 00 00 00 00 00
platform caam_sm: [0016] 00 00 00 00 00 00 00 00
platform caam_sm: [0024] 00 00 00 00 00 00 00 00
platform caam_sm: [0032] 00 00 00 00 00 00 00 00
usb-storage 1-1.2:1.0: USB Mass Storage device detected
platform caam_sm: [0040] 00 00 00 00 00 00 00 00
platform caam_sm: [0048] 00 00 00 00 00 00 00 00
scsi host0: usb-storage 1-1.2:1.0
platform caam_sm: [0056] 00 00 00 00 00 00 00 00
platform caam_sm: [0064] 00 00 00 00 00 00 00 00
platform caam_sm: [0072] 00 00 00 00 00 00 00 00
platform caam_sm: [0080] 00 00 00 00 00 00 00 00
platform caam_sm: [0088] 00 00 00 00 00 00 00 00
platform caam_sm: 64-bit black key in blob:
platform caam_sm: [0000] f4 bd f7 72 2a c0 40 f5
```

```
platform caam_sm: [0008] a1 dc 16 3c 89 2d fd fb
platform caam_sm: [0016] 16 ca 4b 69 72 3a 06 9b
platform caam_sm: [0024] eb df 99 76 24 56 e5 16
platform caam_sm: [0032] 89 27 36 e0 0b 9d 44 fb
platform caam_sm: [0040] 5a bf 69 42 7e 62 c7 71
platform caam_sm: [0048] be 5a 80 15 30 0a f0 c6
platform caam_sm: [0056] 00 00 00 00 00 00 00 00
platform caam_sm: [0064] 00 00 00 00 00 00 00 00
platform caam_sm: [0072] 00 00 00 00 00 00 00 00
platform caam_sm: [0080] 00 00 00 00 00 00 00 00
platform caam_sm: [0088] 00 00 00 00 00 00 00 00
platform caam_sm: 128-bit black key in blob:
platform caam_sm: [0000] 41 50 2f 59 7b 24 53 d5
platform caam_sm: [0008] 5c 01 b9 70 c9 5a eb ac
platform caam_sm: [0016] 60 2e 0e 33 5a 66 44 2a
platform caam_sm: [0024] 25 d5 3a 82 0f 93 4b 9f
platform caam_sm: [0032] cf 8a 8b 74 ac 64 0c ed
platform caam_sm: [0040] 8a 14 5d 57 02 fa 9a 31
platform caam_sm: [0048] c1 8c cc b1 61 b7 1d 32
platform caam_sm: [0056] b2 59 82 45 33 4a 92 6c
platform caam_sm: [0064] 00 00 00 00 00 00 00 00
platform caam_sm: [0072] 00 00 00 00 00 00 00 00
platform caam_sm: [0080] 00 00 00 00 00 00 00 00
platform caam_sm: [0088] 00 00 00 00 00 00 00 00
platform caam_sm: 192-bit black key in blob:
platform caam_sm: [0000] 98 a9 fc 1b c5 06 4a 1a
platform caam_sm: [0008] da ad 7d 27 6d 49 64 60
platform caam_sm: [0016] 38 c0 25 3d b0 d7 02 08
platform caam_sm: [0024] 2a 25 75 2c bd d1 9b 06
platform caam_sm: [0032] 55 b1 13 e2 9e 26 01 87
platform caam_sm: [0040] f1 b4 93 2d 45 b2 21 7d
platform caam_sm: [0048] 4c af e6 7a 10 ca 89 33
platform caam_sm: [0056] c4 ee 47 e2 79 21 b3 e8
platform caam_sm: [0064] 89 ef ee 5f 76 30 38 aa
platform caam_sm: [0072] 00 00 00 00 00 00 00 00
platform caam_sm: [0080] 00 00 00 00 00 00 00 00
platform caam_sm: [0088] 00 00 00 00 00 00 00 00
platform caam_sm: 256-bit black key in blob:
```

```
platform caam_sm: [0000] 4d 5e e3 14 57 e6 07 4b
platform caam_sm: [0008] 7c 12 99 b1 e2 1c 0b 7b
platform caam_sm: [0016] 01 88 6c 55 5e 11 c4 c8
platform caam_sm: [0024] 1c 64 7f d2 aa 22 71 3e
platform caam_sm: [0032] 0a 49 fd ee 83 c8 02 74
platform caam_sm: [0040] 6a 44 a3 c4 a3 4c c5 11
platform caam_sm: [0048] 2a 5f 64 a4 8c 65 93 62
platform caam_sm: [0056] fd 86 62 f5 31 69 a4 32
platform caam_sm: [0064] 0e e7 17 2b b7 da b6 f2
platform caam_sm: [0072] ac 9f 62 1a e9 47 19 b0
platform caam_sm: [0080] 00 00 00 00 00 00 00 00
platform caam_sm: [0088] 00 00 00 00 00 00 00 00
platform caam_sm: restored 64-bit black key:
platform caam_sm: [0000] 74 7f 0d 63 64 4e 26 ef
platform caam_sm: [0008] 91 9f 07 71 3c 4f b6 06
platform caam_sm: restored 128-bit black key:
platform caam_sm: [0000] 01 36 a0 18 6e 95 66 e7
platform caam_sm: [0008] 76 ef 79 32 38 a3 72 df
platform caam_sm: restored 192-bit black key:
platform caam_sm: [0000] be 92 40 57 8a ba fb 35
platform caam_sm: [0008] 0d f6 e6 5e 3e 37 85 2c
platform caam_sm: [0016] 2a 3e 67 de b7 42 40 44
platform caam_sm: [0024] 02 3d 2f c5 12 b4 61 db
platform caam_sm: restored 256-bit black key:
platform caam_sm: [0000] 6d 80 57 c0 09 41 ef 3c
platform caam_sm: [0008] be ae 26 c3 a4 cb 08 0b
platform caam_sm: [0016] d1 f1 71 91 b5 fb 55 bc
platform caam_sm: [0024] dc 7b 8c 26 c7 12 61 74
snvs-secvio 30370000.caam-snvs: can't get snvs clock
snvs-secvio 30370000.caam-snvs: violation handlers armed
- non-secure state
usbcore: registered new interface driver usbhid
usbhid: USB HID core driver
input: ST LSM6DS3 Accelerometer Sensor as
/devices/soc0/soc/30800000.aips-bus/30a40000.i2c/i2c-2/2-
006a/input/input1
input: ST LSM6DS3 Gyroscope Sensor as
/devices/soc0/soc/30800000.aips-bus/30a40000.i2c/i2c-2/2-
006a/input/input2
```

```
input: ST LSM6DS3 Significant Motion Sensor as
/devices/soc0/soc/30800000.aips-bus/30a40000.i2c/i2c-2/2-
006a/input/input3

input: ST LSM6DS3 Step Counter Sensor as
/devices/soc0/soc/30800000.aips-bus/30a40000.i2c/i2c-2/2-
006a/input/input4

input: ST LSM6DS3 Step Detector Sensor as
/devices/soc0/soc/30800000.aips-bus/30a40000.i2c/i2c-2/2-
006a/input/input5

input: ST LSM6DS3 Tilt Sensor as
/devices/soc0/soc/30800000.aips-bus/30a40000.i2c/i2c-2/2-
006a/input/input6

lsm6ds3 2-006a: lsm6ds3: probed

imx_wm8750_probe

imx_wm8750_probe snd_soc_register_card

imx-wm8750 sound: wm8750-hifi <-> 308b0000.sai mapping ok

imx-wm8750 sound: ASoC: no source widget found for Main
MIC

imx-wm8750 sound: ASoC: Failed to add route Main MIC ->
direct -> RINPUT1

imx_wm8750_late_probe codec_dai name is wm8750-hifi

imx_wm8750_set_bias_level SND_SOC_BIAS_STANDBY

NET: Registered protocol family 26

NET: Registered protocol family 10

sit: IPv6, IPv4 and MPLS over IPv4 tunneling driver

NET: Registered protocol family 17

can: controller area network core (rev 20120528 abi 9)

NET: Registered protocol family 29

can: raw protocol (rev 20120528)

can: broadcast manager protocol (rev 20161123 t)

can: netlink gateway (rev 20130117) max_hops=1

Bluetooth: RFCOMM TTY layer initialized

Bluetooth: RFCOMM socket layer initialized

Bluetooth: RFCOMM ver 1.11

Bluetooth: BNEP (Ethernet Emulation) ver 1.3

Bluetooth: BNEP filters: protocol multicast

Bluetooth: BNEP socket layer initialized

Bluetooth: HIDP (Human Interface Emulation) ver 1.2

Bluetooth: HIDP socket layer initialized

8021q: 802.1Q VLAN Support v1.8

Key type dns_resolver registered
```

```
cpu cpu0: dev_pm_opp_get_opp_count: OPP table not found
(-19)

imx_thermal 30000000.aips-bus:tempmon: Commercial CPU
temperature grade - max:95C critical:90C passive:85C

input: gpio-keys as /devices/soc0/gpio-keys/input/input8

rtc-ds1307 1-0068: setting system clock to 2000-01-01
00:26:02 UTC (946686362)

wlreg_on: disabling

wlan-en-regulator: disabling

usb_otg1_vbus: disabling

usb_otg2_vbus: disabling

mipi_dsi_pwr_on: disabling

ALSA device list:

 #0: wm8750-audio

EXT4-fs (mmcblk2p2): couldn't mount as ext3 due to
feature incompatibilities

EXT4-fs (mmcblk2p2): recovery complete

EXT4-fs (mmcblk2p2): mounted filesystem with ordered data
mode. Opts: (null)

VFS: Mounted root (ext4 filesystem) on device 179:26.

devtmpfs: mounted

Freeing unused kernel memory: 1024K (80e00000 - 80f00000)

systemd[1]: System time before build time, advancing
clock.

systemd[1]: systemd 230 running in system mode. (-PAM -
AUDIT -SELINUX +IMA -APPARMOR +SMACK +SYSVINIT +UTMP -
LIBCRYPTSETUP -GCRYPT -GNUTLS +AC)

systemd[1]: Detected architecture arm.


Welcome to NXP i.MX Release Distro 4.9.11-1.0.0 (morty)!


systemd[1]: Set hostname to <esomimx7d-1gb>.

scsi 0:0:0:0: Direct-Access JetFlash Transcend 8GB 1100
PQ: 0 ANSI: 4

sd 0:0:0:0: [sda] 15826944 512-byte logical blocks: (8.10
GB/7.55 GiB)

sd 0:0:0:0: [sda] Write Protect is off

sd 0:0:0:0: [sda] No Caching mode page found

sd 0:0:0:0: [sda] Assuming drive cache: write through

 sda: sda1

sd 0:0:0:0: [sda] Attached SCSI removable disk
```

```
systemd[1]: connman.service: Cannot add dependency job,
ignoring: Unit connman.service is masked.

systemd[1]: basic.target: Found ordering cycle on
basic.target/start

systemd[1]: basic.target: Found dependency on
sockets.target/start

systemd[1]: basic.target: Found dependency on
rpcbind.socket/start

systemd[1]: basic.target: Found dependency on
sysinit.target/start

systemd[1]: basic.target: Found dependency on
alignment.service/start

systemd[1]: basic.target: Found dependency on
basic.target/start

[ SKIP ] Ordering cycle found, skipping Sockets

[ SKIP ] Ordering cycle found, skipping alignment.service

[ OK ] Listening on udev Kernel Socket.

[ OK ] Created slice System Slice.

[ OK ] Listening on udev Control Socket.

[ OK ] Listening on Syslog Socket.

[ OK ] Started Dispatch Password Requests to Console
Directory Watch.

[ OK ] Listening on /dev/initctl Compatibility Named
Pipe.

[random: crng init done

 OK ] Listening on Journal Socket (/dev/log).

[ OK ] Created slice User and Session Slice.

[ OK ] Reached target Remote File Systems.

[ OK ] Started Forward Password Requests to Wall
Directory Watch.

[ OK ] Created slice system-getty.slice.

[ OK ] Created slice system-serial\x2dgetty.slice.

[ OK ] Reached target Paths.

[ OK ] Listening on Journal Socket.

 Starting Setup Virtual Console...

 Starting Remount Root and Kernel File Systems...

EXT4-fs (mmcblk2p2): re-mounted. Opts: (null)

 Starting Journal Service...

 Mounting Debug File System...

 Starting Load Kernel Modules...

[ OK ] Reached target Slices.
```

```
[ OK ] Reached target Swap.
 Mounting Temporary Directory...
[ OK ] Mounted Debug File System.
[ OK ] Mounted Temporary Directory.
[ OK ] Started Journal Service.
[ OK ] Started Setup Virtual Console.
[ OK ] Started Remount Root and Kernel File Systems.
[FAILED] Failed to start Load Kernel Modules.
See 'systemctl status systemd-modules-load.service' for
details.
 Starting Apply Kernel Variables...
 Mounting NFSD configuration filesystem...
 Mounting FUSE Control File System...
 Starting udev Coldplug all Devices...
 Starting Create Static Device Nodes in /dev...
 Starting Flush Journal to Persistent Storage...
[ OK ] Mounted FUSE Control File System.
[ OK ] Started Apply Kernel Variables.
[FAILED] Failed to mount NFSD configuration filesystem.
See 'systemctl status proc-fs-nfsd.mount' for details.
[ OK ] Started Create Static Device Nodes in /dev.
systemd-journald[220]: Received request to flush runtime
journal from PID 1
[ OK ] Reached target Local File Systems (Pre).
 Mounting /var/volatile...
 Starting udev Kernel Device Manager...
[ OK ] Mounted /var/volatile.
[ OK ] Started Flush Journal to Persistent Storage.
[ OK ] Started udev Kernel Device Manager.
 Starting Load/Save Random Seed...
[ OK ] Reached target Local File Systems.
 Starting Create Volatile Files and Directories...
[ OK ] Started udev Coldplug all Devices.
[ OK ] Started Load/Save Random Seed.
[ OK ] Started Create Volatile Files and Directories.
 Starting Network Time Synchronization...
 Starting Update UTMP about System Boot/Shutdown...
 Starting Start Psplash Boot Screen...
```

```
[ OK ] Created slice system-systemd\x2dbacklight.slice.
 Starting Load/Save Screen Backlight...htness of
backlight:backlight...
[ OK ] Started Start Psplash Boot Screen.
[ OK ] Started Load/Save Screen Backlight Brightness of
backlight:backlight.
[ OK ] Started Update UTMP about System Boot/Shutdown.
[ OK ] Found device /dev/ttymxc0.
mxc_mipi-csi 30750000.mipi-csi: mipi csi v4l2 device
registered
CSI: Registered sensor subdevice: mxc_mipi-csi.0
ov8865_mipi: loading out-of-tree module taints kernel.
1-0024 supply DOVDD not found, using dummy regulator
1-0024 supply DVDD not found, using dummy regulator
mxc_mipi-csi 30750000.mipi-csi: lanes: 2, hs_settle: 13,
clk_settle: 2, wclk: 1, freq: 240000000
[ OK ] Started Network Time Synchronization.
[ OK ] Reached target System Time Synchronized.
ov8865_write_reg:write reg error:reg=fffd,val=80
camera ov8865 init failed
[ OK ] Reached target System Initialization.
ov8865_mipi: probe of 1-0024 failed with error -1
[ OK ] Reached target Basic System.
[ OK ] Started Xserver startup with a display manager.
 Starting Console System Startup Logging...
 Starting Terminate Psplash Boot Screen...
[ OK ] Started System Logging Service.
[ OK ] Started Daily Cleanup of Temporary Directories.
[ OK ] Reached target Timers.
[ OK ] Started Updates psplash to basic.
[ OK ] Started Kernel Logging Service.
[ OK ] Listening on D-Bus System Message Bus Socket.
 Starting Telephony service...
 Starting Login Service...
[ OK ] Listening on dropbear.socket.
[ OK ] Started Periodic Command Scheduler.
 Starting Save/Restore Sound Card State...
[ OK ] Started Job spooling tools.
```

```
[ OK ] Listening on Avahi mDNS/DNS-SD Stack Activation
Socket.
 Starting Avahi mDNS/DNS-SD Stack...
 Starting Network Time Service (one-shot ntpdate mode)...
[ OK ] Started D-Bus System Message Bus.
[ OK ] Started Avahi mDNS/DNS-SD Stack.
[ OK ] Started Telephony service.
 Starting Install MySQL Community Server Database...
 Starting /etc/rc.local Compatibility...
[ OK ] Listening on RPCbind Server Activation Socket.FAT-
fs (mmcblk2p1): Volume was not properly unmounted. Some
data may be corrupt. Please.


[ OK ] Started Console System Startup Logging.
[ OK ] Started Terminate Psplash Boot Screen.
[ OK ] Started Network Time Service (one-shot ntpdate
mode).
[ OK ] Started /etc/rc.local Compatibility.
EXT4-fs (mmcblk0p2): mounting ext3 file system using the
ext4 subsystem
FAT-fs (mmcblk0p1): Volume was not properly unmounted.
Some data may be corrupt. Please run fsck.
[ OK ] Started Install MySQL Community Server Database.
EXT4-fs (mmcblk0p2): recovery complete
EXT4-fs (mmcblk0p2): mounted filesystem with ordered data
mode. Opts: (null)
[ OK ] Started Save/Restore Sound Card State.
FAT-fs (sda1): Volume was not properly unmounted. Some
data may be corrupt. Please run fsck.
[ OK ] Reached target Sound Card.
[ OK ] Started Serial Getty on ttymxc0.
[ OK ] Started Getty on tty1.
[ OK ] Reached target Login Prompts.
[ OK ] Started Login Service.
[ OK ] Reached target Multi-User System.
[ OK ] Reached target Graphical Interface.
 Starting Update UTMP about System Runlevel Changes...
[ OK ] Started Update UTMP about System Runlevel Changes.


NXP i.MX Release Distro 4.9.11-1.0.0 esomimx7d-1gb
ttymxc0
```

```
esomimx7d-1gb login:
```

What's Next?

You can refer to the *Acacia Linux Build Guide* document, to get information about building the BSP.

Glossary

AADP: Advanced Audio Distribution Profile

AP: Access Point

BLE: Bluetooth Low Energy

DTB: Device Tree Blob

HID: Human Interface Devices

eMMC: Embedded MultiMedia Controller

OBEX: OBject EXchange

OTG: On-The-Go

RAM: Random Access Memory

Rootfs: Root file System

SOM: System on Module

IVT: Image Vector Table

Support

Contact Us

If you need any support on Acacia product, please contact us using the Live Chat option available on our website - https://www.e-consystems.com/

Creating a Ticket

If you need to create a ticket for any type of issue, please visit the ticketing page on our website - https://www.e-consystems.com/create-ticket.asp

RMA

To know about our Return Material Authorization (RMA) policy, please visit the RMA Policy page on our website - https://www.e-consystems.com/RMA-Policy.asp

General Product Warranty Terms

To know about our General Product Warranty Terms, please visit the General Warranty Terms page on our website - https://www.e-consystems.com/warranty.asp

Revision History

Rev	Date	Description	Author
1.0	07 May 2018	Initial Draft	SOM-Team