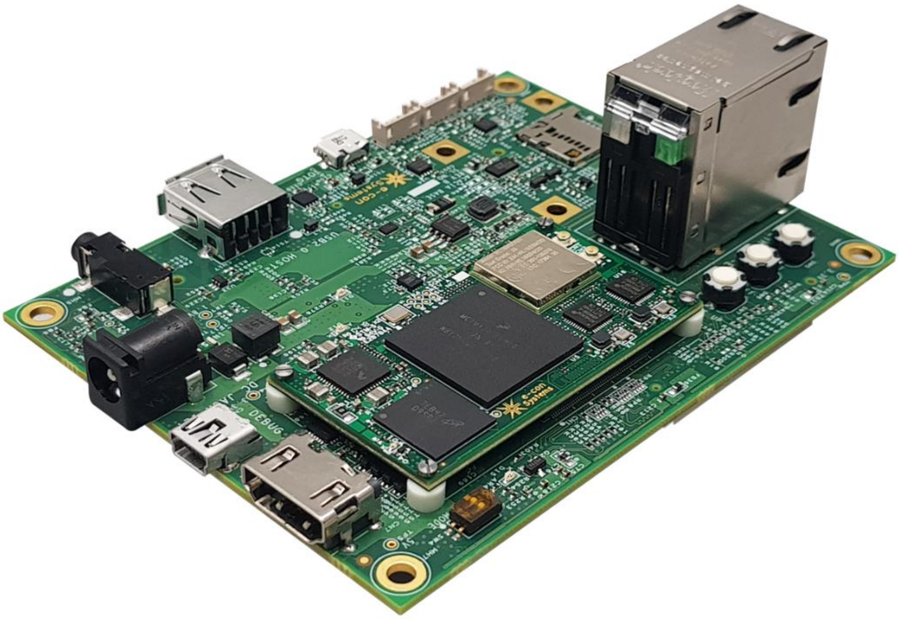
Acacia

Linux Yocto Prebuilt Binaries 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

The prebuilt binaries are available in the Developer Resources website, which you can flash to the SOM using the recommended tools. You can also update each binary for example, U-Boot, zImage, device tree blob (DTB) and rootfs individually.

Note: The prebuilt binaries provided with the package is only for Acacia board, it is recommended not to use these prebuilt binaries with your custom boards.

This document explains the procedure of writing the ready to use binaries available in the prebuilt binary package to SD, eMMC or NAND flash of the eSOMiMX7 module.

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 Colors

Color	Notation
Red	Commands running in Acacia kit
Blue	Commands in Linux development system
Green	Output messages in Linux development system

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

```
U-Boot  >
```

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

```
root@esomimx7x:~#
```

Where x is s (Solo SOM) or d (Dual SOM).

System Requirements

The system requirements are as follows:

- Minimum PC Requirements – 2.0 GHz CPU, 1GB RAM with USB 2.0 controller.
- OS Requirement – Ubuntu 14.04 or later.

Location of Prebuilt Binaries

You can download the ready to use binaries from e-con Systems Developer Resources website - http://developer.e-consystems.com. The ready to use binaries are available in the following path.

Acacia_YYYY_Linux_Prebuilt_Binaries_vX.XX

Where, YYYY is Solo for Solo SOM (or) Dual for Dual SOM

X.XX is BSP package release version

The package can be extracted using the following command.

```
$ tar -xf Acacia_YYYY_Linux_
Prebuilt_Binaries_vX.XX.tar.bz2
```

Contents of Prebuilt Binaries

The prebuilt binary package contains the bootloader, DTB, kernel and rootfs binaries. The binaries in the prebuilt binaries directory and its purpose are listed in the following table.

Table 2: Binaries Detail

Binary	Built	Purpose
u-boot.imx	u-boot.2017.03	Bootloader Binary
zImage	Linux 4.9.11	Kernel image
Solo SOM imx7s-acacia.dtb **Dual SOM** imx7d-acacia.dtb	Device tree source	Board specific information for Dual SOM
initramfs.cpio.gz.uboot	Busy Box V1.24.1	Initramfs Image
esomimx7.rootfs.tar.bz2	Yocto	Compressed root file system
Solo SOM esomimx7-X11-image-esomimx7s.sdcard.tar.bz2 **Dual SOM** esomimx7-X11-image-esomimx7d.sdcard.tar.bz2	Yocto	Compressed single bootable disk image (u-boot + zImage + dtb + rootfs)

Creating Bootable SD Card using SD Image

This section describes how to create the bootable SD card using SD card image.

Prerequisites

The prerequisites for creating the bootable SD card using SD card image are as follows:

- You need a Linux development system to create a bootable SD card.
- You need a SD card of minimum size 4 GB as rootfs and other partitions requires around 3 GB of size.

Description

Creating a bootable SD card is as simple as copying a file to SD card. The SD card need not be partitioned, if it is partitioned already then make sure you have unmounted the partition from the Linux development system before proceeding further.

The SD card image is present as compressed tar.bz2 file in the prebuilt binary.

Run the following commands to extract the tar.bz2 file and disk dump the extracted image to the SD card device node for different eSOMiMX6 variants.

Solo SOM

```
$ tar -xvf esomimx7-X11-image-esomimx7s.sdcard.tar.bz2
$ sudo dd if=esomimx7-X11-image-esomimx7s.sdcard
of=/dev/sdx bs=4M conv=notrunc
```

Dual SOM

```
$ tar -xvf esomimx7-X11-image-esomimx7d.sdcard.tar.bz2
$ sudo dd if=esomimxd-X11-image-esomimx7d.sdcard
of=/dev/sdx bs=4M conv=notrunc && sync
```

The sdx device node of the SD card is provided along with partition number, for example, /dev/sdb1 and /dev/sdb2.

Note: Make sure to enter the correct device node, which does not lead to corruption of OS partitions in your development system.

Once the dd command is completed, you can remove the SD card and insert it in the Acacia board in SD connector CN13. Make sure you have set boot switches to internal boot mode to boot the Acacia board from the SD card. The following table shows the boot mode pin details for internal boot.

Table 3: Boot Mode Pin Details for Internal Boot

SW4 PIN NUMBER	STATE
1	OFF
2	ON

Power **ON** the board after setting the above switch settings to boot from SD card.

Preparing a Bootable eMMC or NAND

Preparing a bootable eMMC or NAND involves the following steps:

1. Fusing - For writing iMX7 fuse registers to set the boot device to eMMC or NAND.
2. Flashing - For flashing the binaries to the corresponding boot device.

For both fusing and flashing, it is recommended to use the MfgTools utility available in the Developer Resources website.

The following table shows the details about the MfgTools packages and corresponding folder for fuse and flash tools.

Table 4: MfgTools Packages

SOM	Package	Tool for Fusing	Tool for Flashing
Solo	mfgtools_solo_vX.Y.tar.bz2	mfgtools_fuse_esomimx7_solo	mfgtools_flash_esomimx7_solo
Dual	mfgtools_dual_vX.Y.tar.bz2	mfgtools_fuse_esomimx7_dual	mfgtools_flash_esomimx7_dual

Refer to the *e-con_Acacia_Linux_MFGTools_Flash_Programming_UserManual.pdf* and *e-con_Acacia_Linux_MFGTools_Fuse_Programming_UserManual.pdf*, for details about how to use MfgTools for flashing and fusing.

Updating Individual Binaries in SD or eMMC

The procedures mentioned in previous section updates all the binaries, it is time consuming and it will also erase the user specific data present in SD or eMMC. Using the above procedures during development will impact your development time, so it is recommended to update the individual binaries which are modified.

Note: You can follow the steps in this section only when the SD or eMMC is used as boot device. Refer to the *Contents of Prebuilt Binaries* section and *Creating Bootable SD Card using SD Image* section, for information on configuring SD or eMMC as bootable.

The individual binaries of SD or eMMC can be updated using the ums command in U-Boot. The selected memory device is exposed as a USB mass storage device through a USB client port of Acacia, so that you can access the SD or eMMC as mass storage device in your development system.

Connect the Acacia board to development system using USB micro-B to A cable in connector CN19 and power **ON** the device.

Follow the below procedures, to update the individual binaries such as U-Boot, zImage, device tree blob and rootfs in SD or eMMC.

Updating U-Boot in SD or eMMC

To update U-Boot in SD or eMMC, follow these steps:

1. Run the following commands in U-Boot prompt to expose eMMC or SD of Acacia as mass storage in the development system.

   For updating in SD:

   ```
   U-boot > ums 0 mmc 1
   ```

   For updating in eMMC:

   ```
   U-boot > ums 0 mmc 0
   ```

2. Run the following umount command to unmount the exposed partitions.

   ```
   $ sudo umount /dev/sdx*[1-9]
   ```

3. Run the following dd command to flash the U-Boot binary to the exposed storage device.

   ```
   $ sudo dd if=<PATH of UBOOT>/u-boot.imx of=/dev/sdx
   bs=1k seek=1
   ```

4. Reboot the Acacia board to boot the SOM with updated bootloader (U-Boot).

   The sdx device node of the SD card is provided along with partition number, for example, /dev/sdb and /dev/sdc.

   Note: Make sure to enter the correct device node, which does not lead to corruption of OS partitions in your development system.

Updating zImage in SD or eMMC

To update zImage in SD or eMMC, follow these steps:

1. Run the following commands in U-Boot prompt to expose eMMC or SD of Acacia as mass storage in development system.

   For updating in SD:

   ```
   U-boot > ums 0 mmc 1
   ```

   For updating in eMMC:

   ```
   U-boot > ums 0 mmc 0
   ```

   The partitions in SD or eMMC are mounted in your development system as boot and root.

2. Run the following command from Linux development machine to copy zImage to the exposed boot partition of the SD or eMMC.

   ```
   sudo cp <PATH to zImage>/zImage
   <MOUNT_POINT_OF_BOOT_PARTITION> && sync
   ```

3. Run the following umount command to unmount the exposed partitions.

   ```
   $ sudo umount /dev/sdx*[1-9]
   ```

4. Reboot the Acacia board to boot the SOM with zImage.

Updating DTB in SD or eMMC

To update DTB in SD or eMMC, follow these steps:

1. Run the following commands to expose eMMC or SD of Acacia as mass storage in development system.

   For updating in SD:

   ```
   U-boot > ums 0 mmc 1
   ```

   For updating in eMMC:

   ```
   U-boot > ums 0 mmc 0
   ```

Two partitions in SD or eMMC are mounted in your development system as boot and root.

2. Run the following command from Linux development machine to copy imx7x-aacia.dtb to the exposed boot partition of the SD or eMMC.

   ```
   sudo cp <PATH of device tree blob>/imx7x-aacia.dtb
   <MOUNT_POINT_OF_BOOT_PARTITION> && sync
   ```
3. Unmount the mounted SD card or eMMC from the Linux development machine.
4. Reboot the Acacia board to boot the SOM with the updated imx7X-aacia.dtb DTB.

   Note: In imx7**x**-acacia.dtb, **x** is s for Solo SOM or d for Dual SOM.

Updating Ramdisk Image in SD or eMMC

To update ramdisk image in SD or eMMC, follow these steps:

1. Run the following commands to expose eMMC or SD of Acacia as mass storage in development system.

   For updating in SD:

   ```
   U-boot > ums 0 mmc 1
   ```

   For updating in eMMC:

   ```
   U-boot > ums 0 mmc 0
   ```

   Two partitions in SD or eMMC are mounted in your development system as boot and root.

2. Run the following command from Linux development machine to copy the initramfs.cpio.gz.uboot to the exposed boot partition of the SD or eMMC.

   ```
   sudo cp <PATH of rootfs>/initramfs.cpio.gz.uboot
   <MOUNT_POINT_OF_BOOT_PARTITION> && sync
   ```
3. Unmount the mounted SD or eMMC from the Linux development machine.
4. Reboot the Acacia board to boot the SOM with the updated ramdisk image.

Updating rootfs in SD or eMMC

To update rootfs in SD or eMMC, follow these steps:

1. Run the following commands to expose eMMC or SD of Acacia as mass storage in development system.

   For updating in SD:

   ```
   U-boot > ums 0 mmc 1
   ```

For updating in eMMC:

```
U-boot > ums 0 mmc 0
```

Two partitions in SD or eMMC are mounted in your development system as boot and root.

2. Run the following command from Linux development machine to extract and copy the esomimx7.rootfs.tar.bz2 to the exposed root partition of the SD or eMMC.

```
sudo tar -xvjf <PATH of
rootfs>/esomimx7.rootfs.tar.bz2 -C
<MOUNT_POINT_OF_ROOTFS> && sync
```

3. Unmount the mounted SD or eMMC from the Linux development machine.
4. Reboot the Acacia board to boot the SOM with updated rootfs.

Updating Individual Binaries in NAND

Updating the binaries in NAND can be done only from the debug terminal of Acacia, ums option is not available for NAND device. Since this procedure can be done only in the Acacia device, it is mandatory to boot the device (kernel) from SD to perform the below procedure. Copy all the prebuilt binaries to an USB Flash disk and insert it to Acacia device before updating the individual binaries in NAND.

Follow the below procedures, to update the individual binaries in NAND.

Configuring U-Boot

The environment configurations are set for NAND boot since the U-Boot for Solo is built for NAND as boot device, so you need to modify the environment variables to boot kernel from SD card.

Run the following commands to modify the environment variables.

```
U-boot > setenv fdt_file imx7s-acacia.dtb
U-boot > setenv bootfile zImage
U-boot > setenv image zImage
U-boot > setenv fdt_addr 0x83000000
U-boot > setenv boot_fdt try
U-boot > setenv script boot.scr
U-boot > setenv loadaddr 0x80800000
U-boot > setenv mmcpart 1
U-boot > setenv mmcdev 0
U-boot > setenv mmcroot "/dev/mmcblk2p2 rootwait rw"
U-boot > setenv mmcargs "U-boot > setenv bootargs
console=${console},${baudrate} root=${mmcroot}
mtdparts=gpmi-nand:64m(boot),16m(kernel),16m(dtb),-
(rootfs)"
U-boot > setenv loadfdt "fatload mmc ${mmcdev}:${mmcpart}
${fdt_addr} ${fdt_file}"
U-boot > setenv loadbootscript "fatload mmc
${mmcdev}:${mmcpart} ${loadaddr} ${script};"
U-boot > setenv bootscript "echo Running bootscript from
mmc ...; source"
U-boot > setenv loadimage "fatload mmc
${mmcdev}:${mmcpart} ${loadaddr} ${image}"
U-boot > setenv mmcboot "echo Booting from mmc ...; run
mmcargs; if test ${boot_fdt} = yes || test ${boot_fdt} =
try; then if run loadfdt; then bootz ${loadaddr} -
```

```
${fdt_addr}; else if test ${boot_fdt} = try; then bootz;
else echo WARN: Cannot load the DT; fi; fi; else bootz;
fi;"
U-boot > boot
```

After executing the above commands, the Acacia device will boot the kernel from SD card.

Updating U-Boot in NAND

To update U-Boot in NAND, follow these steps:

1. Boot device from SD and boot to kernel.
2. Run the following command to erase the U-Boot partition mtd0.

```
root@esomimx7x:~# flash_erase /dev/mtd0 0 0
```

3. Run the following command to write the U-Boot to NAND.

```
root@esomimx7x:~# kobs-ng init -x -v --
chip_0_device_path=/dev/mtd0 /media/sdx1/u-boot.imx
```

The U-Boot is updated in NAND after executing the above nandwrite command.

Updating zImage in NAND

To update zImage in NAND, follow these steps:

1. Boot device from SD and boot to kernel.
2. Run the following command to erase the kernel partition mtd1.

```
root@esomimx7x:~# flash_erase /dev/mtd1 0 0
```

3. Run the following command to write the kernel (zImage) to NAND.

```
root@esomimx7x:~# nandwrite -p /dev/mtd1 -p
/media/sdx1/zImage
```

The kernel image is updated in NAND after executing the above command.

Updating DTB in NAND

To update DTB in NAND, follow these steps:

1. Boot device from SD and boot to kernel.
2. Run the following command to erase the DTB partition mtd2.

```
root@esomimx7x:~# flash_erase /dev/mtd2 0 0
```

3. Run the following command to write the DTB to NAND.

```
root@esomimx7x:~# nandwrite -p /dev/mtd2 -p
/media/sdx1//imx7x-acacia.dtb
```

The DTB is updated in NAND after executing the above command.

Updating Ramdisk Image in NAND

To update ramdisk image in NAND, follow these steps:

1. Boot device from SD and boot to kernel.
2. Run the following command to erase the initrd partition mtd3.

```
root@esomimx7x:~# flash_erase /dev/mtd3 0 0
```

3. Run the following command to write the ramdisk image to NAND.

```
root@esomimx7x:~# nandwrite -p /dev/mtd3 -p
/media/sdx1//initramfs.cpio.gz.uboot
```

The ramdisk image is updated in NAND after executing the above command.

Updating rootfs in NAND

To update rootfs in NAND, follow these steps:

1. Boot device from SD and boot to kernel.
2. Run the following command to format rootfs partition mtd4.

```
root@esomimx7x:~# ubiformat /dev/mtd4
```

3. Run the following ubiattach command to attach mtd3 device.

```
root@esomimx7x:~# ubiattach /dev/ubi_ctrl -m 4
```

4. Run the following command to create UBI volume for the attached device.

```
root@esomimx7x:~# ubimkvol /dev/ubi0 -Nrootfs -m
```

5. Run the following commands to create a mount point for mtd4.

```
root@esomimx7x:~# mkdir -p /mnt/mtd4
root@esomimx7x:~#  mount -t ubifs ubi0:rootfs
/mnt/mtd4
```

6. Run the following command to extract the rootfs to the mounted partition mtd4.

```
root@esomimx7x:~#  tar -jxf
/media/sda1/esomimx7.rootfs.tar.bz2 -C /mnt/mtd4
```

7. Run the following command to unmount the partition.

```
root@esomimx7x:~# umount /mnt/mtd4
```

The rootfs is updated in NAND.

Troubleshooting

In this section, you can view the list of commonly occurring issues and their troubleshooting steps.

I have followed the step in *Creating a Bootable SD Card using SD Image* section, but when I insert the SD card and power ON the board, board is not booting.

You can follow the below steps to solve this issue:

1. Insert the SD card to the development PC.
2. Check whether the boot and root partitions are present, if not please refer to the *Creating Bootable SD Card using SD Image* section to dump SD image again (SD card image may not be written properly).

   If the partitions are present, then please refer to the *Creating Bootable SD Card using SD Image* section to verify the switch settings in SW4 of Acacia is configured properly.

I want to update selective binaries in eMMC or SD, I have given ums command from U-Boot, but nothing is showing up in PC.

1. Check whether the correct mmc device detail is entered, for eMMC ums 0 mmc 0 and for SD card ums 0 mmc 1.
2. Check the micro B cable connected to connector CN19 (OTG1) of Acacia device with development PC.

FAQ

1. **What is the minimum SD card size used for DD image?**

   It is recommended to use minimum 4 GB micro SD card.

2. **Can the NAND file system support ums command? or I need to expose NAND as mass storage using ums command?**

   No, NAND file system do not support ums command.

What’s Next?

You can refer to following documents to get information about building BSP and eSOMiMX7 Acacia BSP.fds.

- *e-con Acacia Linux Build Guide*
- *e-con Acacia Linux User Manual*

Glossary

DTB: Device Tree Blob

eMMC: Embedded MultiMedia Controller

OTG: On-The-Go

RAM: Random Access Memory

SOM: System on Module

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