
F-Cue Linux
Board Support Package
User's Manual

Rev.1.00 2017/01/06

FUJITSU ELECTRONICS INC.

shaping tomorrow with you

1. The contents of this document may be changed without prior notice. Please verify with our sales department if you need any assistance.
2. The data and circuits given in this document are merely examples of using the semiconductor device and are not intended for installation in equipment that actually used. Further, our company does not bear any liabilities whatsoever regarding any third patent right violations or other rights violations arising out of the use of these data and circuits.
3. Transferring or copying the contents of this document without prior written permission from our company is prohibited.
4. The semiconductor devices of our company are intended for use in standard applications (office equipment such as computer, OA, etc., equipment related to industrial, communication, measurement, equipment for personal or home use, etc.). Customers considering use in special applications (aircrafts, space, nuclear power control, life sustaining medical applications, etc.) such as in equipment wherein any fault or wrong operation of these semiconductors can directly cause a threat to human life, or is likely to cause damage to human body, or wherein extremely high reliability is required are requested to consult our sales department before using.
5. Failures of semiconductor devices occur at certain probabilistic rates. The customers are requested to carry out safety designs such as equipment redundancy design, fire spreading countermeasure design, overcurrent prevention countermeasure design, wrong operation prevention design, etc., so that, even if our company's semiconductor device fails, as a result human injury accidents, fire accidents, and accidents causing social damage do not occur.
6. When the product described in this document comes under the goods or technology restricted under the Foreign Exchange and Foreign Trade Act of Japan, permission from the concerned authority is required at the time of exporting this product.

©2017 FUJITSU ELECTRONICS INC. Printed in Japan

Contents

1. Introduction.....	4
2. Development Environment.....	5
2.1. BSP.....	5
2.2. MSB7701 Board.....	5
2.2.1. Board Setting (DIP SW).....	5
3. Binary Building.....	6
3.1. Toolchain	6
3.2. Building kernel source code	7
3.2.1. Prepare kernel source code	7
3.2.1.1. Get the base kernel source code	7
3.2.1.2. Apply the patch to kernel source code.....	7
3.2.2. Buildroot details	7
3.2.2.1. Make the main kernel + modules.....	7
3.2.2.2. Strip and install the modules into the staging directory	8
3.2.2.3. Buildroot the dtbs.....	8
3.3. Building U-Boot source code	8
3.3.1. Prepare U-Boot source code.....	8
3.3.1.1. Get the base U-Boot source code.....	8
3.3.1.2. Apply the patch to U-Boot source code	8
3.3.2. Buildroot details	8
4. eMMC Preparation for Boot Device.....	9
4.1. Procedure for F-Cue Board.....	9
4.1.1. Preliminary preparations.....	9
4.1.2. Configuration of eMMC for boot device.....	9
4.1.2.1. Preparation booting micro SD card and boot Linux from micro SD card.....	9
4.1.2.2. Configuration of eMMC	9
5. micro SD card Preparation for Boot Device	10
5.1. Procedure for F-Cue Board.....	10
5.1.1. Preliminary preparations.....	10
5.1.2. Buildroot Linux kernel and U-boot	10
5.1.3. Making a micro SD card for Booting	10
5.2. Procedure for Linux PC.....	11
5.2.1. Partition setting	11
5.2.2. Format.....	11
5.2.3. Copy Image of the file system to the micro SD card.....	11
5.2.4. Copy Linux kernel and U-boot to the micro SD card.....	11
5.2.5. Remove the micro SD card.....	11

1. Introduction

The document is "User Manual" about F-Cue platform for Linux BSP.

This document describes the contents of the following required for Software developers to use the F-Cue Board, to the Software development.

This document describes how to set up the F-Cue Board, environment necessary to the software development, Build method of Kernel, Update method of Firmware to the Board.

2. Development Environment

2.1. BSP

- Boot loader
 - Linaro U-boot is used as boot loader of F-Cue Linux BSP. Boot loader copies operating system from storage to DRAM and transfers control to it.
- Linux kernel
 - F-Cue Linux BSP provides SoC/board dependent device drivers for F-Cue as patch file to mainline Linux kernel source code.
- Tool chain
 - GNU based ARM tool chain is required to build F-Cue Linux BSP.
 - Customers are advised to download the tool chain from Linaro's web site. Note that this BSP does not contain the tool chain.

2.2. MSB7701 Board

2.2.1. Board Setting (DIP SW)

DIP SW	Content Memo	Memo
Boot ROM Select SW	For Flash Memory of Boot ROM	BRSEL

This DIP SW select the Flash Memory that U-Boot is placed and monitor setting.
Loader load U-Boot from the selected Flash Memory to DDR.

4	3	2	1	Monitor	U-Boot Place position
1 (OFF)	1 (OFF)	0 (ON)	0 (ON)	HDMI	eMMC
1 (OFF)	0 (ON)	0 (ON)	0 (ON)	HDMI	micro SD Card
0 (ON)	1 (OFF)	0 (ON)	0 (ON)	MIPI-DSI	eMMC
0 (ON)	0 (ON)	0 (ON)	0 (ON)	MIPI-DSI	micro SD Card

(*) ON/OFF status is DIP SW value of F-Cue Board.

3. Binary Building

Refer to the following procedures if you restructure Linux without using Prebuild Linux image.

3.1. Toolchain

We used the following arm eabi le hardfloat toolchain

http://releases.linaro.org/archive/14.11/components/toolchain/binaries/arm-linux-gnueabi/gcc-linaro-4.9-2014.11-x86_64_arm-linux-gnueabi.tar.xz

For reference, installation of the above is simple

```
# cd /opt
# tar xvf /path/to/gcc-linaro-4.9-2014.11-x86_64_arm-linux-gnueabi.tar.xz
```

/opt is an arbitrary directory.

Then edit your user `~/.bash_profile` to append this path/bin to your \$PATH

```
PATH=$PATH:/opt/gcc-linaro-4.9-2014.11-x86_64_arm-linux-gnueabi/bin
export CROSS_COMPILE=arm-linux-gnueabi-
export ARCH=arm
```

If you open a new terminal session, you should be able to use just the cross compiling prefix **arm-linux-gnueabi-**

Eg)

```
$ arm-linux-gnueabi-gcc
```

3.2. Building kernel source code

3.2.1. Prepare kernel source code

3.2.1.1. Get the base kernel source code

This step is to get the Linux kernel source code from the linux git.

```
$ git clone git://git.kernel.org/pub/scm/linux/kernel/git/torvalds/linux.git
```

Change path of "Linux kernel source code"

```
$ cd linux  
$ git checkout v3.10
```

3.2.1.2. Apply the patch to kernel source code

This step is to apply the patch for F-Cue to kernel source code

```
$ cd linux  
$ patch -p1 < msb7701-v3.10-delta_YYYY.MM-N.patch
```

Source for the Device Tree can be found in two files:

- arch/arm/boot/dts/mb86s71.dtsi – MB86S71 SoC definitions
- arch/arm/boot/dts/msb7701.dts – F-Cue Board specific definitions (includes the above)

3.2.2. Buildroot details

3.2.2.1. Make the main kernel + modules

This step is to build the kernel + modules source code.

```
$ make msb7701_defconfig  
$ make -j8
```

The main kernel is in arch/arm/boot/Image (uncompressed).

3.2.2.2. Strip and install the modules into the staging directory

```
$ mkdir ./staging
$ rm -rf ./staging/*;
$ make INSTALL_MOD_STRIP=1 modules_install INSTALL_MOD_PATH=./staging
```

The directory structure in ./staging (./staging/lib/modules...) should be copied to the rootfs / to provide the modules.

3.2.2.3. Buildroot the dtbs

```
$ make dtbs
```

The dtbs is in arch/arm/boot/dts/msb7701.dtb

3.3. Building U-Boot source code

3.3.1. Prepare U-Boot source code

3.3.1.1. Get the base U-Boot source code

This step is to download the U-Boot source code from Linaro git.

```
$ git clone git://git.linaro.org/boot/u-boot-linaro-stable.git
$ cd u-boot-linaro-stable
$ git checkout 2013.02.2
```

3.3.1.2. Apply the patch to U-Boot source code

This step is to apply the patch for F-Cue to U-Boot source code

```
$ patch -p1 < msb7701-bsp-u-boot-linaro-stable-2013.02.2_YYYY.MM-N.patch
```

3.3.2. Buildroot details

This step is to build the u-boot source code.

```
$ make msb7701_config
$ make -j8
```

4. eMMC Preparation for Boot Device

In this manual, it describes about the configuration procedure of the eMMC which is used when booting Linux BSP from eMMC. In fact, there is no way to write something to eMMC directly, which means there are no eMMC writers or tools for on board eMMC of the F-Cue Board. Therefore, Linux booted from micro SD card should be used to configure the eMMC.

4.1. Procedure for F-Cue Board

4.1.1. Preliminary preparations

Environment F-Cue Board, booting micro SD card

Image of the file system Use the prebuild Linux image.

4.1.2. Configuration of eMMC for boot device

To configure the eMMC, it needs following processes.

1. Prepare booting micro SD card and boot Linux from micro SD card
2. Configure eMMC, which create partition, format the partition and copy or make necessary information from micro SD card to eMMC.

4.1.2.1. Preparation booting micro SD card and boot Linux from micro SD card

Please refer to “5 micro SD card Preparation for Boot Device”.

There are no special things for eMMC writing micro SD card.

Then, please boot Linux from micro SD card.

4.1.2.2. Configuration of eMMC

First, Copy “linaro-trusty-alip-20141024-684_msb7701_YYYYMMDD.tar.xz” to any flash memory.

Copy or create necessary folders and files to eMMC by executing the script from Linux command prompt.

```
$ setup_emmc.sh linaro-trusty-alip-20141024-684_msb7701_YYYYMMDD.tar.xz
```

The process has been completed.

5. micro SD card Preparation for Boot Device

In this chapter, it describes about the configuration procedure of the micro SD card which is used when booting Linux BSP from micro SD card.

5.1. Procedure for F-Cue Board

5.1.1. Preliminary preparations

Environment F-Cue Board, booting eMMC

Image of the file system Use the prebuild Linux image.

5.1.2. Buildroot Linux kernel and U-boot

Refer to "3. Binary Building" if you restructure Linux without using Prebuild Linux image.

5.1.3. Making a micro SD card for Booting

First, Copy "linaro-trusty-alip-20141024-684_msb7701_YYYYMMDD.tar.xz" to any flash memory.

Copy or create necessary folders and files to the micro SD card by executing the script from Linux command prompt.

```
$ setup_sd.sh linaro-trusty-alip-20141024-684_msb7701_YYYYMMDD.tar.xz
```

The process has been completed.

5.2. Procedure for Linux PC

In case F-Cue Board does not boot, it describes about the configuration procedure of micro SD card by Linux PC.

“sdb” means the device name which is mounted micro SD card.

“/media/rootfs” means the mountpoint of micro SD card.

“\$” indicates the command prompt of Linux OS.

5.2.1. Partition setting

First, delete all the existing partitions.

```
$ sudo fdisk -u /dev/sdb
$ d
```

Next, make one new partition.

```
$ n
$ p
$ 1
$ 2048
$ (Enter)
$ w
```

5.2.2. Format

Format the new partition (sdb1).

```
$ sudo mkfs.ext4 /dev/sdb1 -L rootfs
```

A re-mount on the micro SD card is recommended, after the format was completed.

5.2.3. Copy Image of the file system to the micro SD card

Extract “prebuild Linux image” to the rootfs on the micro SD card.

```
$ sudo tar xf linaro-trusty-alip-20141024-684_msb7701_YYYYMMDD.tar.xz -C /media/rootfs
```

5.2.4. Copy Linux kernel and U-boot to the micro SD card

Refer to the following procedures if you restructure Linux without using prebuild Linux image.

Copy your kernel and others to the rootfs on the SD card.

If your kernel has any .ko files, Copy your .ko file to the rootfs on the SD card.

5.2.5. Remove the micro SD card

Detach micro SD card after unmounting.

The process has been completed.

Revision History

Date	Version	Description
2017/01/06	1.00	First Edition