

GM8136

SD CARD CONTROLLER

User Guide

Rev.: 1.0

Issue Date: August 2014



REVISION HISTORY

GM8136 SD Card User Guide

Date	Rev.	From	To
Aug. 2014	1.0	-	Original

Copyright © 2014 Grain Media, Inc.

All Rights Reserved.

Printed in Taiwan 2014

Grain Media and the Grain Media Logo are trademarks of Grain Media, Inc. in Taiwan and/or other countries. Other company, product and service names may be trademarks or service marks of others.

All information contained in this document is subject to change without notice. The products described in this document are NOT intended for use in implantation or other life support application where malfunction may result in injury or death to persons. The information contained in this document does not affect or change Grain Media's product specification or warranties. Nothing in this document shall operate as an express or implied license or indemnity under the intellectual property rights of Grain Media or third parties. All information contained in this document was obtained in specific environments, and is presented as an illustration. The results obtained in other operating environments may vary.

THE INFORMATION CONTAINED IN THIS DOCUMENT IS PROVIDED ON AN "AS IS" BASIS. In no event will Grain Media be liable for damages arising directly or indirectly from any use of the information contained in this document.

Grain Media, Inc.
5F, No. 5, Li-Hsin Road III, Hsinchu Science Park, Hsinchu City, Taiwan 300, R.O.C.

Grain Media's home page can be found at:
<http://www.grain-media.com>

TABLE OF CONTENTS

Chapter 1	Introduction.....	1
	1.1 Version of the Software	2
	1.2 Brief Description	2
	1.3 Terminology.....	2
	1.4 Development Environments	2
	1.5 Design Assumptions.....	3
Chapter 2	Setup and Configuration	7
	2.1 Kernel Driver Embedded	8
	2.2 Build Driver as Modules	9
	2.3 Using Software	10
Chapter 3	Test Case	11
	3.1 Test Driver 1: mmc_test	12
	3.2 Test Driver 2: mmc_block.....	13
	3.3 USB Auto-mount/Auto-umount.....	14
	3.4 Reference File	17

LIST OF TABLES

Table 1-1.	Host Development Environments.....	2
Table 1-2.	Target Development Environments.....	3
Table 1-3.	Modes for Data Transfers	3
Table 1-4.	Design Assumptions of Current Release	5

LIST OF FIGURES

Figure 2-1. Boot Process by Using Built-in Kernel Driver..... 8

Figure 2-2. Menu Configuration for Building Driver as Module 9

Figure 2-3. Mount SD Card in Platform 10

Figure 3-1. Insert and Execute mmc_test in Kernel 12

Figure 3-2. mmc_block Test 13

Chapter 1

Introduction

This chapter contains the following sections:

- 1.1 Version of the Software
- 1.2 Brief Description
- 1.3 Terminology
- 1.4 Development Environments
- 1.5 Design Assumptions

1.1 Version of the Software

Software release version: 1.0.0

1.2 Brief Description

FTSDC021 is a SD host controller that complies with the SD Host Controller Standard Specification, version 3.00. This document is for users to understand the Linux kernel 3.3 driver for FTSDC021. This driver provides the patch to drivers/host/mmc/sdhci.c and is mainly used for the vendor-defined register program.

Several new features in the standard specification support the SDXC card, such as the 1.8-V I/O signal voltage, and execute the tuning procedure in the SDR50 mode and SDR104 mode. In addition, because the AHB bus is used by FTSDC021, sdhci-ftsdc021.c will be used to register the driver to the platform bus so that the driver probe function can be called to complete the initialization job.

1.3 Terminology

Item	Description
SD	Security Digital
MMC	Multimedia Card
SDIO	Security Digital Input/Output
SDR50	Single Data Rate (Up to 100MHz)
SDR140	Single Data Rate (Up to 208MHz)

1.4 Development Environments

Table 1-1. Host Development Environments

Item	Description
Host platform	PC with Intel Pentium II or above
Host RAM	256MB or above
Host OS	Any PC Linux for cross compiler
Host cross compiler	GCC 4.4.0 cross compiler by uclibc

Table 1-2. Target Development Environments

Item	Description
Target platform	GM8136 EVB
Target RAM	512MB or above
Target OS	Linux 3.3
Host cross compiler	GCC 4.4.0-cross compiler by uclibc

1.5 Design Assumptions

Users must define the transfer modes, as shown in Table 1-3, to change the way data transferred by host between the SD card and system memory.

Table 1-3. Modes for Data Transfers

Item	Description
Transfer_through_ADMA	Use the ADMA transfer mode
Transfer_through_SDMA	Use the SDMA transfer mode
Transfer_through_PIO	Use the PIO transfer mode

Since the information provided from SD may not always be correct, the additional timeout values will be required for the abnormal cards. Users can use the largest timeout value to avoid receiving the error message, "**mmcblk0: error -110 transferring data**". Please use `sdhci-ftsdc021.c` for `SDHCI_QUIRK_BROKEN_TIMEOUT_VAL`. `SDHCI_QUIRK_CAP_CLOCK_BASE_BROKEN` should be used if `FTSDC021` does not provide the base clock frequency information.

The following figures illustrate how to configure the SD/MMC driver for FTSDC021 in Linux 3.3 on the platform. Users should do it step by step and use the built-in kernel SD/MMC driver for FTSDC021.

```
.config - Linux/arm 3.3.0 Kernel Configuration

Linux/arm 3.3.0 Kernel Configuration
Arrow keys navigate the menu. <Enter> selects submenus --->.
Highlighted letters are hotkeys. Pressing <Y> includes, <N> excludes,
<M> modularizes features. Press <Esc><Esc> to exit, <?> for Help, </>
for Search. Legend: [*] built-in [ ] excluded <M> module < >

^ (-)
  CPU Power Management --->
  Floating point emulation --->
  Userspace binary formats --->
  Power management options --->
  [*] Networking support --->
  Device Drivers --->
  File systems --->
  Kernel hacking --->
  Security options --->
  *- Cryptographic API --->
v (+)

<Select>  < Exit >  < Help >
```

```
.config - Linux/arm 3.3.0 Kernel Configuration

Device Drivers
Arrow keys navigate the menu. <Enter> selects submenus --->.
Highlighted letters are hotkeys. Pressing <Y> includes, <N> excludes,
<M> modularizes features. Press <Esc><Esc> to exit, <?> for Help, </>
for Search. Legend: [*] built-in [ ] excluded <M> module < >

^ (-)
< > Multimedia support --->
  Graphics support --->
< > Sound card support --->
  [*] HID Devices --->
  [*] USB support --->
  MMC/SD/SDIO card support --->
< > Sony MemoryStick card support (EXPERIMENTAL) --->
  [ ] LED Support --->
  [ ] Accessibility support --->
  [ ] Real Time Clock --->
v (+)

<Select>  < Exit >  < Help >
```

In the MMC/SD/SDIO card support option, users should choose the following item. After setting the menu configuration in kernel, users should build the kernel image and the SD/MMC driver is built in the image.

```
.config - Linux/arm 3.3.0 Kernel Configuration

MMC/SD/SDIO card support
Arrow keys navigate the menu. <Enter> selects submenus --->.
Highlighted letters are hotkeys. Pressing <Y> includes, <N> excludes,
<M> modularizes features. Press <Esc><Esc> to exit, <?> for Help, </>
for Search. Legend: [*] built-in [ ] excluded <M> module < >

--- MMC/SD/SDIO card support
[ ] MMC debugging (NEW)
[ ] Assume MMC/SD cards are non-removable (DANGEROUS) (NEW)
[ ] MMC host clock gating (EXPERIMENTAL) (NEW)
*** MMC/SD/SDIO Card Drivers ***
<*> MMC block device driver (NEW)
(8)   Number of minors per block device (NEW)
[*]   Use bounce buffer for simple hosts (NEW)
< > SDIO UART/GPS class support (NEW)
< > MMC host test driver (NEW)
*** MMC/SD/SDIO Host Controller Drivers ***
<*> Secure Digital Host Controller Interface support
<*> SDHCI platform and OF driver helper
<*> FTSDC021 Secure Digital Host Controller Interface support
< > MMC/SD/SDIO over SPI (NEW)
< > Synopsys DesignWare Memory Card Interface (NEW)
< > VUB300 USB to SDIO/SD/MMC Host Controller support (NEW)
< > USB SD Host Controller (USHC) support (NEW)

<Select> < Exit > < Help >
```

Table 1-4. Design Assumptions of Current Release

Item	Description
Related hardware	FTSDC021
Interrupt parameters	<ul style="list-style-type: none"> • IRQ 15 (SDC#0) • IRQ 17 (SDC#1) • Level triggered • Active high
Physical address range	<ul style="list-style-type: none"> • 0x92800000 ~ 0x92800FFF (SDC#0) • 0x93800000 ~ 0x93800FFF (SDC#1)

Chapter 2

Setup and Configuration

This chapter contains the following sections:

- 2.1 Kernel Driver Embedded
- 2.2 Build Driver as Modules
- 2.3 Using Software

2.1 Kernel Driver Embedded

This is the default option of the platform and users are suggested using this option. Figure 2-1 is the booting output message of the built-in driver.

```
usbcore: registered new interface driver usb-storage
USB Mass Storage support registered.
mousedev: PS/2 mouse device common for all mice
ftiic010 ftiic010.0: irq 18, mapped at 9f86a000
sdhci: Secure Digital Host Controller Interface driver
sdhci: Copyright(c) Pierre Ossman
mmc0: SDHCI controller on ftsdc021.0 [ftsdc021.0] using ADMA
sdhci-pltfm: SDHCI platform and OF driver helper
usbcore: registered new interface driver usbhid
usbhid: USB HID core driver
TCP cubic registered
NET: Registered protocol family 17
Warning: unable to open an initial console.
Freeing init memory: 2000K

BusyBox v1.19.4 (2013-03-11 11:10:36 CST) built-in shell (ash)
Enter 'help' for a list of built-in commands.

/ # mmc0: new SDHC card at address 7043
mmcblk0: mmc0:7043 SR04G 3.69 GiB
mmcblk0: p1

/ #
```

Figure 2-1. Boot Process by Using Built-in Kernel Driver

2.2 Build Driver as Modules

Grain Media does not suggest using this option. It is more complicated and waste time. Figure 2-2 shows the menu configuration for building the FTSDC021 driver as a module.

```
.config - Linux/arm 3.3.0 Kernel Configuration

MMC/SD/SDIO card support
Arrow keys navigate the menu.  <Enter> selects submenus --->.
Highlighted letters are hotkeys.  Pressing <Y> includes, <N> excludes,
<M> modularizes features.  Press <Esc><Esc> to exit, <?> for Help, </>
for Search.  Legend: [*] built-in  [ ] excluded <M> module  < >

--- MMC/SD/SDIO card support
[ ] MMC debugging
[ ] Assume MMC/SD cards are non-removable (DANGEROUS)
[ ] MMC host clock gating (EXPERIMENTAL)
*** MMC/SD/SDIO Card Drivers ***
<*> MMC block device driver
(8)   Number of minors per block device
[*]   Use bounce buffer for simple hosts
< > SDIO UART/GPS class support
< > MMC host test driver
*** MMC/SD/SDIO Host Controller Drivers ***
<*> Secure Digital Host Controller Interface support
<*> SDHCI platform and OF driver helper
<M> FTSDC021 Secure Digital Host Controller Interface support
< > MMC/SD/SDIO over SPI
< > Synopsys DesignWare Memory Card Interface
< > VUB300 USB to SDIO/SD/MMC Host Controller support
< > USB SD Host Controller (USHC) support

<Select>  < Exit >  < Help >
```

Figure 2-2. Menu Configuration for Building Driver as Module

2.3 Using Software

After booting Linux, the FTSDC021 driver is also brought up. Users can insert the SD card. Then, the card detecting message will be shown on the terminal. Next, users can use “mdev -s” to generate the device node which is /dev/mmcblk0p1, as shown in Figure 2-3. Users can then use “mount -t vfat /dev/mmcblk0p1 /mnt/sd” to mount the SD card.

```
sdhci: Copyright(c) Pierre Ossman
mmc0: SDHCI controller on ftsc021.0 [ftsc021.0] using ADMA
sdhci-pltfm: SDHCI platform and OF driver helper
usbcore: registered new interface driver usbhid
usbhid: USB HID core driver
TCP cubic registered
NET: Registered protocol family 17
Warning: unable to open an initial console.
Freeing init memory: 2000K
mmc0: new SDHC card at address 7043
mmcblk0: mmc0:7043 SR04G 3.69 GiB
  mmcblk0: p1

BusyBox v1.19.4 (2013-03-11 11:10:36 CST) built-in shell (ash)
Enter 'help' for a list of built-in commands.

/ # mdev -s
/ # mkdir /mnt/sd
/ # mount -t vfat /dev/mmcblk0p1 /mnt/sd/
/ # df
Filesystem            1K-blocks      Used Available Use% Mounted on
/dev/mmcblk0p1         3861392         4    3861388   0% /mnt/sd
/ #
```

Figure 2-3. Mount SD Card in Platform

Chapter 3

Test Case

This chapter contains the following sections:

- 3.1 Test Driver 1: mmc_test
- 3.2 Test Driver 2: mmc_block
- 3.3 USB Auto-mount/Auto-umount
- 3.4 Reference File

3.1 Test Driver 1: mmc_test

mmc_test is the built-in test program in kernel 3.3. This driver provides basic raw-data read/write tests for the memory cards. Users should insert the mmc_test driver into the kernel after the steps described in Section 2.3 are completed. The procedure is shown in Figure 3-1. It includes 18 test cases in mmc_test.

Command: **echo "#" > /sys/devices/platform/ftsd010/mmc_host\:mmc0/mmc0\:b368/test**

```
/sys/class/mmc_host/mmc0/mmc0:0002/test
/sys/class/mmc_host/mmc0/mmc0:0002/type
/sys/class/mmc_host/mmc0/mmc0:0002/uevent
/ $ echo 18 > /sys/class/mmc_host/mmc0/mmc0\:0002/test
mmc_test_store
mmc0: Starting tests of card mmc0:0002...
mmc0: Test case 18. Correct xfer_size at read (midway failure)...

mmc0: Timeout waiting for hardware interrupt.
Timeout in DMA read/write
ret :0
mmc0: Result: OK
mmc0: Tests completed.
/ $
/ $ echo 1mmc_test_storemmc_host/mmc0/mmc0\:0002/test
> /sys/class/mmcmmc0: Starting tests of card mmc0:0002...
_hmmc0: Test case 1. Basic write (no data verification)...
ost/mmc0/mmc0\:0002/test
ret :0
mmc0: Result: OK
mmc0: Tests completed.
/ $ echo 2 > /sys/class/mmc_host/mmc0/mmc0\mmc_test_store
:0002/test
mmc0: Starting tests of card mmc0:0002...
mmc0: Test case 2. Basic read (no data verification)...
ret :0
mmc0: Result: OK
```

Figure 3-1. Insert and Execute mmc_test in Kernel

3.2 Test Driver 2: mmc_block

mmc_block.ko should already be inserted or built in the kernel; otherwise, the mount command will fail. Users can use the “mount” and “ls” commands to perform the block test.

Command: **mount -t vfat /dev/mmcblk0 /mnt/sd**

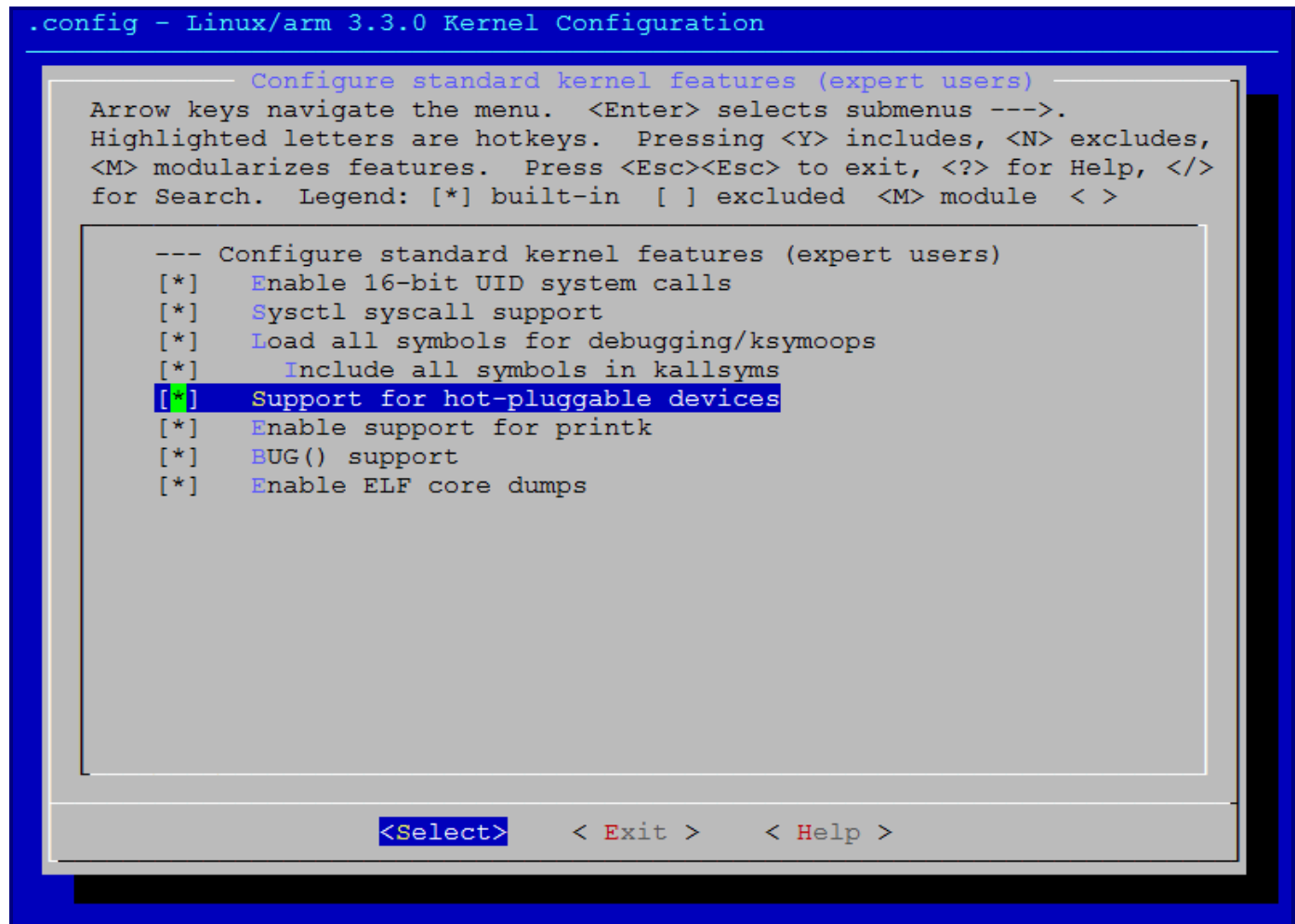
```
/ $  
/ $ ls -al /dev/mmc*  
brw-rw----    1 root    root      179,   0 Jan  1 00:00 /dev/mmcblk0  
/ $  
/ $ mount -t vfat /dev/mmcblk0 /mnt/sd  
/ $ cd /mnt/sd  
/mnt/sd $ ls  
/mnt/sd $ █
```

Figure 3-2. mmc_block Test

3.3 USB Auto-mount/Auto-umount

If users want to use auto-mount/auto-umount for the SD card, the following steps should be executed:

1. Open the hot-pluggable function in the kernel menu configuration. The path is General setup -> Configure standard kernel features -> Support for hot-pluggable devices. Please refer to the image below.



2. Add the commands in "arm-linux-3.3/target/rootfs-cpio/etc/init.d/rc.sysinit". Please refer to the image below.

```
1 #!/bin/sh
2
3 PATH=/bin:/sbin:/usr/bin:/usr/sbin
4 export PATH
5
6 # mount root filesystem in read-write mode
7 /bin/busybox echo "Mounting root fs rw ..."
8 /bin/busybox mount -n -o remount,rw /
9
10 if [ ! -e /proc/mounts ]; then
11     /bin/busybox mount -t sysfs sysfs /sys
12     /bin/busybox mount -t proc proc /proc
13 fi
14
15 echo "-----Starting mdev....."
16 /bin/echo /sbin/mdev > /proc/sys/kernel/hotplug
17
18 /bin/busybox mdev -s
19 /bin/busybox --install -s
20
21 # mount all other filesystems
22 /bin/busybox echo "Mounting other filesystems ..."
23 /bin/busybox mount -a
```

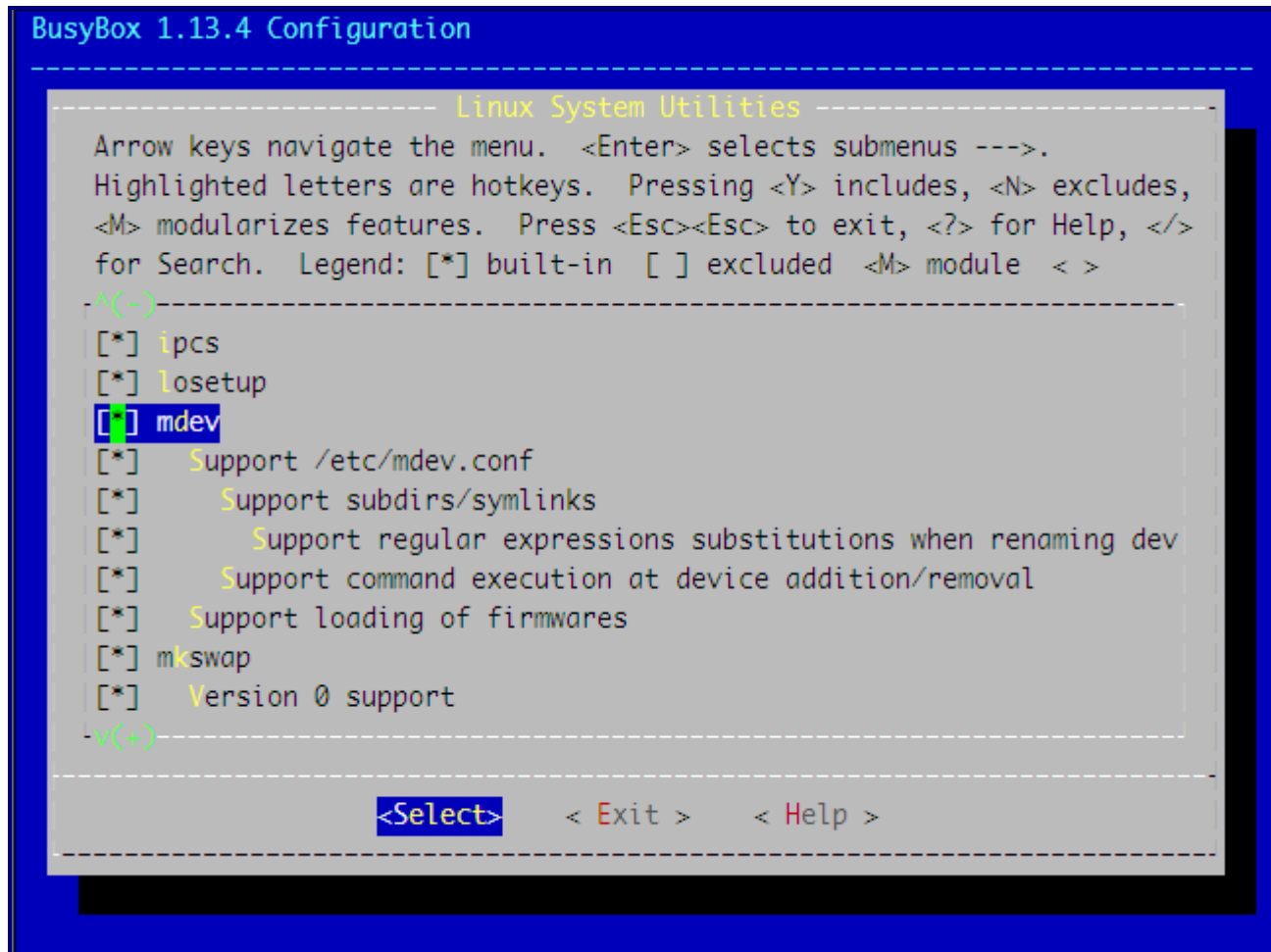
3. Create the "arm-linux-3.3/target/rootfs-cpio/etc/mdev.conf" file with the following content (Line 1 is used for USB and Line 2 is used for SD card.)

```
1 sd.* 0:0 660 */sbin/automount.sh $MDEV
2 mmc.* 0:0 660 */sbin/automount.sh $MDEV
```

4. Create the "arm-linux-2.6/target/rootfs-cpio/sbin/automount.sh" file with the following content (For example)

```
1 MOUNT=/bin/mount
2 UMOUNT=/bin/umount
3 MKDIR=/bin/mkdir
4 RMDIR=/bin/rmdir
5 case $1 in
6     "mmcblk0")
7         domount $1
8         ;;
9     "mmcblk0p1")
10        domount $1
11        ;;
12    "sda")
13        domount $1
14        ;;
15    "sdb")
16        domount $1
17        ;;
18    "sdc")
19        domount $1
20        ;;
21    "sdd")
22        domount $1
23        ;;
24    *)
25        exit 1
26        ;;
27 esac
28
29 function domount{
30     M=`mount | grep $1`
31     if [ "$M" = "" ]; then
32         if [ ! -d /mnt/$1 ]; then
33             $MKDIR /mnt/$1
34         fi
35         $MOUNT -t vfat /dev/$1 /mnt/$1
36     else
37         $UMOUNT -lf /mnt/$1
38         $RMDIR /mnt/$1
39     fi
40 }
```


5. Under "arm-linux-2.6/user/busybox-1.13.4", use "make menuconfig" to enable the mdev related functions. The item is in "Linux System Utilities". Please refer to the image below.



6. Use "make" to generate busybox and copy busybox to the path, "arm-linux-2.6/target/rootfs-cpio/bin"
7. Build the Linux image and users can get the auto-mount/auto-unmount feature for the SD card

3.4 Reference File

FTSDC021 Data Sheet