I.MX6UL TF 卡启动 Linux 的实现

Date：2018/11/05　　公开资料

类别	内容
关键词	TF 卡，linux
摘要	EVB-P6UL 启动

深圳市盈鹏飞科技有限公司 www.embedall.com
电邮：info@embedall.com　　传真：0755-82523090
电话：0086-0755－82523090

修订历史：

版本	日期	原因
V1.00	2018-5-14	创建文档
V1.01	2018-11-05	创建文档

目录

前言

本文档作为 EVB-P6UL 评估板 开发板相关文档的一部分，主要用于指导用户熟悉 EVB-P6UL 评估板开发板如何使用 TF/SD 卡启动 Linux。

- 准备工具
  Linux 系统
  TF 卡一张（内存 4G 以上）
  TF 读卡器一个

- 开发环境
  宿主机：Ubuntu 16.04.3 LTS ，x86_64，3.13.0-128-generic
  嵌入式平台：EVB-P6UL 评估板（i.MX6UL）
  嵌入式系统：Linux 3.14.38

1 TF 卡启动 Linux 的原理

1.1 TF 卡简介

TF 卡又称 T-Flash 卡。全名：【TransFLash】又名【Micro SD】，由摩托罗拉与 SANDISK 共同研发，在 2004 年推出。是一种超小型卡(11*15*1MM)，约为 SD 卡的 1/4，可以算目前最小的储存卡了。MicroSD 卡是一种极细小的快闪存储器卡，其格式源自 SanDisk 创造，原本这种记忆卡称为 T-Flash，及后改称为 TransFlash；而重新命名为 microSD 的原因是因为被 SD 协会（SDA）采立。

其主要应用于移动电话，但因它的体积微小和储存容量的不断提升，现在已经使用于 GPS 设备、便携式音乐播放器和一些快闪存储器盘中。 它的体积为 1mm x 15mm x1mm ，差不多相等于手指甲的大小，是现时最细小的记忆卡。它也能通过 SD 转接卡来接驳于 SD 卡插槽中使用。

1.2 高速卡和低速卡

TF 卡又分为高速卡和低速卡，TF 高速卡和低速卡的区分方法：主要靠 TF 卡上的速度等级标志来识别，TF 卡符合的 SD 规范标准越高，速度分级等级越高，则读写速度越快。如图 1-1， 注意看他右上角的圆圈里的 4 说明这是普速卡，一般传文件速度在每秒 8MB 以下，反应比较慢，传文件更慢，如图 1-1，就是高速卡，注意看他的右边的圆圈的 10，说明是高速卡一般速度在 10MB 以上每秒。

图 1-1

图 1-1

1.3 TF 卡特性

卡容量：

1.标准容量卡（SDSC）：最大容量为 128MB~2GB（默认格式为 FAT16）

2.高容量卡（SDHC）：容量大小为 4~32GB 的卡（默认格式为 FAT32）

3.扩展容量卡（SDXC）：容量大小为 64GB~2TB 的卡（默认格式为 exFAT）

TF 卡一般作为外置扩展容量，那么不同用户用的卡肯定不一样，那为了区分不同的卡，SD3.0 协议中在初始化和识别卡的过程中会判断用户插入的卡是 SDSC/SDHC/SDXC 中的哪一种卡，比如在 R3 中的第 38Bit 的 CCS = 0b 时，表示插入的卡为 SDSC 卡，而 CCS = 1b 时，表示插入的卡为 SDHC 或者 SDXC 卡。

四线总线速率模式，如图 1-3 所示：

Bus Speed Mode*1	Max. Bus Speed [MB/s]	Max. Clock Frequency [MHz]	Signal Voltage [V]	Max. Power*2 [W]		
				SDSC*3	SDHC*4	SDXC*5
SDR104	104	208	1.8	-	2.88*6	2.88*6
SDR50	50	100	1.8	-	1.44	1.44
DDR50	50	50	1.8	-	1.44	1.44
SDR25	25	50	1.8	-	0.72	0.72
SDR12	12.5	25	1.8	-	0.36	0.36/0.54*7
High Speed	25	50	3.3	0.72	0.72	0.72
Default Speed	12.5	25	3.3	0.36	0.36	0.36/0.54*7

*1: The card supports a UHS-I mode shall support all lower UHS-I modes.
*2: Host may control power by the Power Limit function in CMD6 (Refer to Section 4.3.10.3).
*3: SDSC stands for SD Standard Capacity Memory Card and
*4: SDHC stands for SD High Capacity Memory Card.
*5: SDXC stands for SD Extended Capacity Memory Card.
*6: The actual maximum current may vary from the limit described in this table. It is limited by the Mechanical Addenda in the sections that define the thermal profile of the device and the connector profile.
*7: Host may select either maximum power by XPC in ACMD41 (Refer to Section 4.2.3.1). In SPI mode, XPC is not supported and the power shall be up to 0.36W (100mA at 3.6V on VDD1).

Table 3-6 : Bus Speed Modes of UHS-I Card

图 1-3 SD 卡速率模式

说明：

SDR 的意思是 Single Data Rate（单边数据采样，换句话说就是，要么上升沿采样，要么下降沿采样）；

DDR 的意思是 Double Data Rate（双边数据采样，换句话说，双边沿采样）。

1.4 TF 卡工作原理

1.4.1 EVB-P6UL TF 卡硬件设计

EVB-P6UL 的 TF 卡硬件设计兼容 SD 3.0 标准，即支持 SDXC 与 SDHC，原理图 1-3 所示。

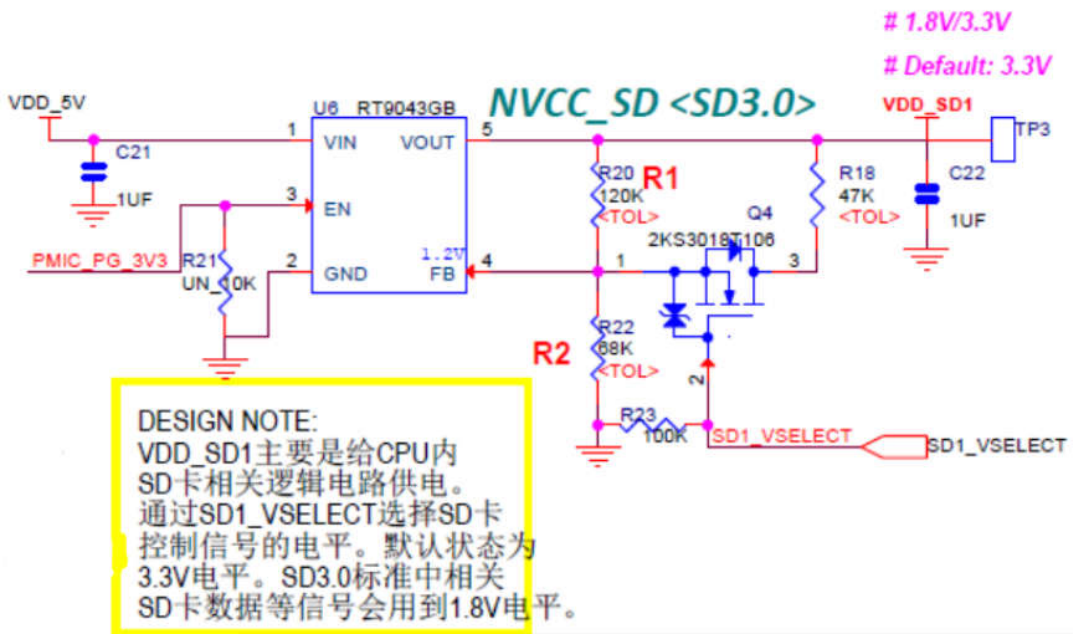


图 1-3 EVB-P6UL TF 卡原理设计

TF 卡的默认工作电压为 3.3V，通过设置 IO 脚“**SD1_VSELECT**”的电平（控制 2KS3018 是否导通，改变 FB 脚的电阻阻值），从而设置 RT9043GB 的反馈电压，最终调节输出电压（RT9043GBV_{out}）。RT9043GB 的输出电压与 FB 脚上电阻有关，相关的计算公式如图 1-4 所示。电阻已经选定，设置“**SD1_VSELECT**”为高电平即可输出 1.8V。

The output voltage divider R1 and R2 allows to adjust the output voltage for various application as shown in Figure 2.

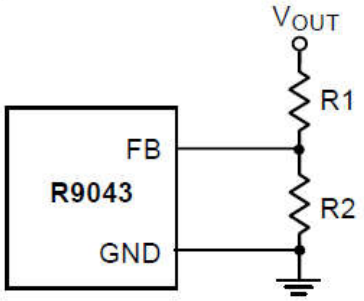


Figure 2. Output Voltage Setting

The output voltage is set according to the following equation:

$$V_{OUT} = V_{FB}\left(1+\frac{R1}{R2}\right)$$

Where V_{FB} is the feedback reference voltage (1.2V typical).

图 1-4RT9043GB 电压计算公式

2 tf 卡及 EMMC 启动

2.1 tf 卡及 EMMC 启动原理

I.MX6UL 根据 BOOT_MODE0 和 BOOT_MODE1 配置不同,分为如下四种启动方式.

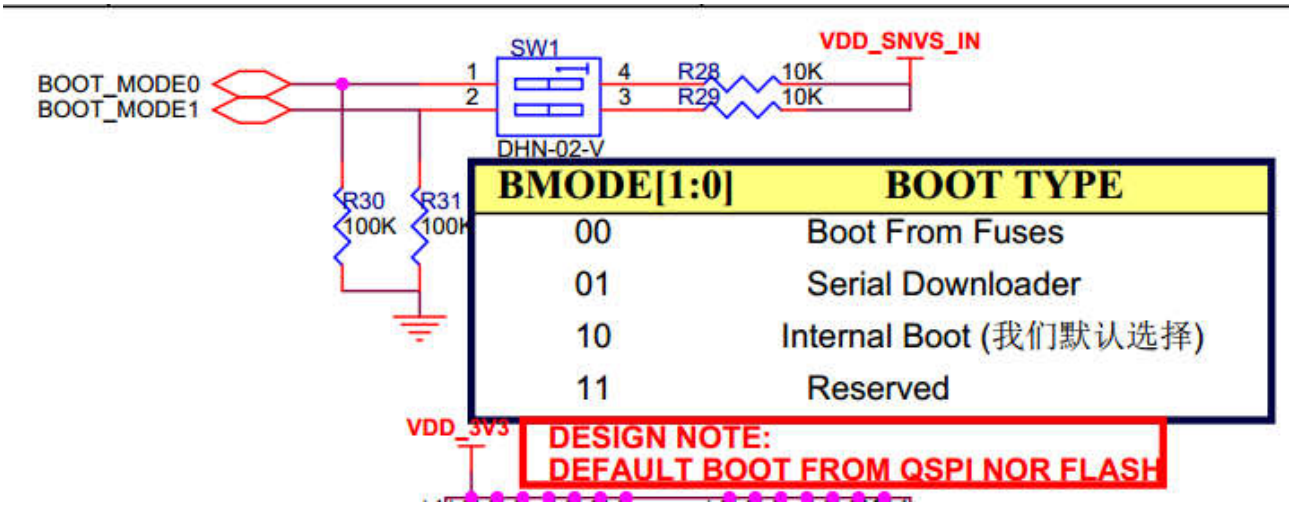


图 2-0 TF 卡启动 GPIO 配置

Fuses 启动模式（BOOT_MODE[1:0] = 00b)

当 BOOT_MODE[1:0]寄存器为 00b 是，选择为从 Fuses 模式下启动。该模式和内部启动模式很相似，只是有一点不同：在此模式下，GPIO 启动(internal boot 内部启动)重载引脚会被忽略。

内部启动模式（BOOT_MODE[1:0] = 0b10)

BOOT_MODE[1:0]寄存器的值为 0b10 时，选择为内部启动模式。在此模式下，处理器继续执行内部的启动代码。启动代码执行硬件初始化，从选择的启动设备中加载程序镜像，使用 HAB 执行镜像有效性检查，然后跳到程序镜像地址处。如果在内部启动中出项任何错误，启动代码就会跳到串行下载器。

内部启动模式一般只用于产品的开发阶段，因为此模式要占用大量的 GPIO 资源，而这些 GPIO 是 EIM 中关键的数据和地址控制线。在 Fuses 启动模式下，一旦这些 eFuses 被烧录，均不能被重擦修改，显而易见的是后者不利于开发中的摸索尝试（一旦出现错误，我们甚至要考虑更换 CPU）。在开发阶段，我们使用跳线来配置 efuses，然后我们调试测试，直至其稳定后，在最终的产品中，使用和跳线配置相对的 eFuses 值来烧录 fuse，最终可将这些 GPIO 上的跳线去除，而用于一般用途。

下载模式（BOOT_MODE[1:0] = 0b01)

该模式下,i.mx6ul 处于下载模式,可以通过 mfgtools 工具对 i.mx6ul 的存储媒质进行烧写.

EVB-P6UL 支持 4 种存储媒质的启动，分别是 SD/TF 卡启动、NAND 启动、eMMC 启动和 QSPI 启动(分为不同的四类产品,根据客户选择核心模块的不同)，通过对 BT_CFG1、BT_CFG2 和 BT_CFG4 三组

信号来配合设置来改变启动媒质，除了 BT_CFG4 之外，其他两组信号 BT_CFG1 和 BT_CFG2 直接影响启动模式选择（BT_CFG1+BT_CFG2+BT_CFG4 等于 24 位 LCD 信号），整个启动配置如下图 2-1、图 2-2 所示：

FUSE MAP <Default: QSPI BOOT>

	0/1	0/1	0/1	1	0	0	0	0
TYPE	BOOT_CFG1[7]	BOOT_CFG1[6]	BOOT_CFG1[5]	BOOT_CFG1[4]	BOOT_CFG1[3]	BOOT_CFG1[2]	BOOT_CFG1[1]	BOOT_CFG1[0]
QSPI	0	0	0	1	Reserved	DDRSMP: "000" - Default "001-111"		
WEIM	0	0	0	0	Memory Type: 0 - NOR Flash 1 - OneNAND	Reserved	Reserved	Reserved
Serial-ROM	0	0	1	1	Reserved	Reserved	Reserved	Reserved
SD/eSD	0	1	0	Fast Boot: 0 Regular 1 - Fast Boot	SD/SDIC Speed: 00 - Normal/SDR12 01 - High/SDR25 10 - SDR50 11 - SDR104		SD Power Cycle Enable: '0' - No power cycle '1' - Enabled via USDHC_RST_pad (uSDHC1 & 4 only)	SD Loopback Clock Source for SDR50 and SDR104 only: '0' - through SD pad '1' - direct
MMC/eMMC	0	1	1	Fast Boot: 0 - Regular 1 - Fast Boot	SD/MMC Speed: 0 - High/ 1- Normal	Ack Boot: Disable: 0 - Boot Ack Enabled 1 - Boot Ack Disabled	SD Power Cycle Enable: '0' - No power cycle '1' - Enabled via USDHC_RST_pad (uSDHC1 & 4 only)	SD Loopback Clock Source for SDR50 and SDR104 only: '0' - through SD pad '1' - direct
NAND	1	BT_TOGGLEMODE	Pages in Block: 00 - 64 01 - 128 10 - 256 11 - 256		Read Number Of Devices: 00 - 1 01 - 2 10 - 4 11 - Reserved		Num_Row_address_bytes: 00 - 3 01 - 4 10 - 5 11 - 5	

	0	0	0	0	1	0	0	0
TYPE	BOOT_CFG2[7]	BOOT_CFG2[6]	BOOT_CFG2[5]	BOOT_CFG2[4]	BOOT_CFG2[3]	BOOT_CFG2[2]	BOOT_CFG2[1]	BOOT_CFG2[0]
QSPI	Reserved	SPI/QSPI Flash Speed Mode: 00 - SDR mode 01 - DDR mode 10 - Reserved 11 - Reserved; 0 - select sampling at non-inverted clock - use clock delay 1 - select sampling at inverted clock 0 - Not Clock delay			0 - select sampling at non-inverted clock 1 - select sampling at inverted clock 0 - Use Clock Delay	Boot Frequencies (ARM/DDR): 0 - 528 / 400 MHz 1 - 396 / 200 MHz	Reserved	Reserved
WEIM	Muxing Scheme: 00 - A/D-0 01 - A-D1 10 - A-D2 11- Reserved			Overlook Page Size: 00 - 2KB 01 - 7KB 10 - 4KB 11 - Reserved	Reserved	Boot Frequencies (ARM/DDR): 0 - 528 / 400 MHz 1 - 396 / 200 MHz	Reserved	Reserved
Serial-ROM	Reserved	Reserved	Reserved	Reserved	Reserved	Boot Frequencies (ARM/DDR): 0 - 528 / 400 MHz 1 - 396 / 200 MHz	Reserved	Reserved
SD/eSD	SD Calibration Step: 00 - 1 10 - 2	Bus Width: 0 - 1 bit 1 - 4 bit		Port Select: 00 - uSDHC1 01 - uSDHC2 10 - Reserved 11 - Reserved		Boot Frequencies (ARM/DDR): 0 - 528 / 400 MHz 1 - 396 / 200 MHz	SDx VOLTAGE SELECTION: 0 - 3.3V 1 - 1.8V	Reserved
MMC/eMMC	Bus Width: 000 - 1bit 001 - 4bit 010 - 8bit 011 - 4bit DDR(MMC 4.4) 100 - 8bit DDR(MMC 4.4) Rest - reserved			Port Select: 00 - uSDHC1 01 - uSDHC2 10 - Reserved 11 - Reserved		Boot Frequencies (ARM/DDR): 0 - 528 / 400 MHz 1 - 396 / 200 MHz	SDx VOLTAGE SELECTION: 0 - 3.3V 1 - 1.8V	Reserved
NAND	Page Size, ECC#0 Threshold, Spare Area Size: 000 - 16 bit/4KB (default) 001 - 8 bit/4KB 010 - 8 bit/2KB 011 - 8 bit/4KB 100 - 8 bit/8KB 101 - 8 bit/16KB 110 - 8 bit/32KB 111 - 8 bit/64KB			BCH ECC#0 (bit): 00 - 2 01 - 4 10 - 8 11 - 8		Boot Frequencies (ARM/DDR): 0 - 528 / 400 MHz 1 - 396 / 200 MHz	Read_Time: 0 - 12ns 1 - 22ns (BCH Need)	Reserved

图 2-1 TF 卡启动 GPIO 配置

	0	0	0	0	0	0	0	0
TYPE	BOOT_CFG4[7]	BOOT_CFG4[6]	BOOT_CFG4[5]	BOOT_CFG4[4]	BOOT_CFG4[3]	BOOT_CFG4[2]	BOOT_CFG4[1]	BOOT_CFG4[0]
0x450	Infinite-Loop (Debug USE only) 0 - Disable 1- Enable	EEPROM Recovery Enable '0' - Disabled '1' - Enabled	CS select (SPI only): 00 - CS#0 (default) 01 - CS#1 10 - CS#2 11 - CS#3		SPI Addressing: 0 - 2-bytes (16-bit) 1 - 3-bytes (24-bit)		Port Select: 000 - eCSPI1 001 - eCSPI2 010 - eCSPI3 011 - eCSPI4 100 - Reserved 101 - Reserved 110 - Reserved 111 - Reserved	
0x460	L2_HW_INVALIDATE_DISABLE	Reserved	FORCE_COLD_BOOT (Ref Latched in SBMR)	BT_FUSE_SEL	DIR_BT_DIS	Reserved	SEC_CONFIG[1]	Reserved
0x460	Reserved (DDR3 configuration)							
0x460	JTAG_SMODE[1:0]	WDOG_ENABLE '0' - Disabled '1' - Enabled	SJC_DISABLE	Reserved	Reserved	Reserved	Reserved	Reserved
0x460	Reserved	Reserved	Reserved	TZASC_ENABLE	JTAG_HEO	KTE	Reserved	DLL_ENABLE 0 - Disable DLL for SD/eMMC 1 - Enable DLL for SD/Emmc
0x470	DLL Override: 0 - DLL Slave Mode for SDx/eMMC 1 - DLL Override Mode for SD/eMMC	Reserved	SD2_VOLTAGE SELECTION 0 - 3.3V 1 - 1.8V	Reserved	Disable SDMMC Manufacture mode 0 - Enable 1 - Disable	L1 I-Cache DISABLE	BT_MMU_DISABLE	Override Pad Settings (using PAD_SETTINGS value)
0x470	Reserved for unexpected requirements	eMMC 4.4 - RESET TO PRE-IDLE STATE	Override HYS bit for SD/MMC pads	USDHC_PAD_PULL_DOWN 0 - no action 1 - pull down	ENABLE_EMMC_22K_PULLUP 0 - 47K pullup 1 - 22K pullup	ADD_DS_SET_GPR1_16 0 - Set 1 - Don't set	USDHC_IOMUX_SION_BIT_ENABLE 0 - Disable 1 - Enable	USDHC_IOMUX SRE Enable 0 - Disable 1 - Enable
0x470	USDHC_CMD_DE_PRE_EN (SD/MMC debug)	LPB_BOOT (Core / DDR- Bus) '00' - LPB Disable '01' - 1 GPIO (default) '10' - Div by2 '11' - Div by 4		BT_LPB_POLARITY (GPIO polarity)	POWER_MNG_CFG (LDO's DCDC's) (Reserved - NOT USED)			
0x470	Override NAND Pad Set (using PAD_SETTINGS value)	MMC_DLL_DLY[6:0] Delay target for SD/eMMC DLL, it is applied to slave mode target delay or override mode target delay depends on DLL Override fuse bit value.						

图 2-2 TF 卡启动 GPIO 配置

而根据 EVB-P6UL 评估板原理图里面对 BOOT 的选择，只设置了几个信号来供用户进行配置选择，其他都已经被强制拉高或拉低，原理设计如图 2-3 所示。

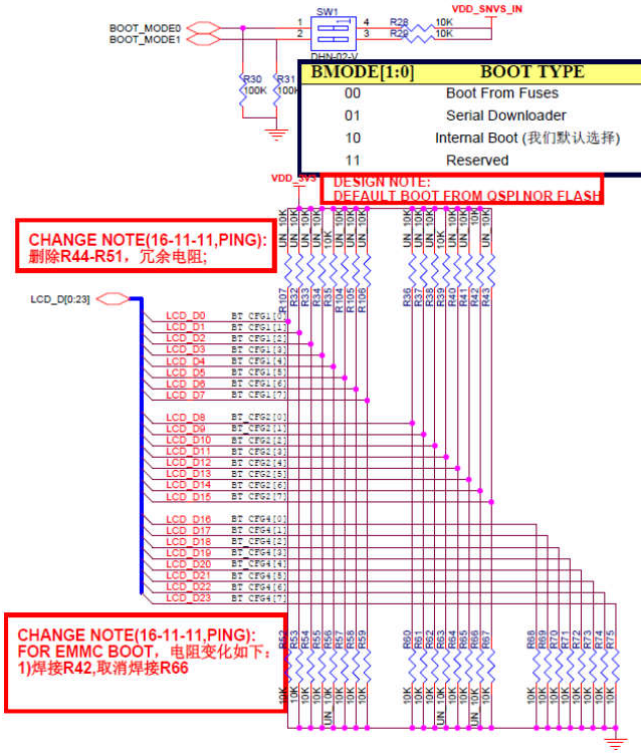


图 2-3 EVB-P6UL 启动 GPIO 配置

因 EVB-P6UL 采取了 internal boot mode,所以,这些配置的 gpio 起到了关键作用，直接决定了启动模式的选择，TF 卡关于信号的配置（启动电阻）说明参考如图 2-4、图 2-5 所示：

Fuse	SBMR1	定义	默认值	设置值	默认值定义
BOOT_CFG1[7:6]	bit7_6	启动设备接口	00	01	01:Boot from USDHC
BOOT_CFG1[5]	bit5	SD/MMC Sel	0	0	0:SD/esd/SDXC; 1:MMC/eMMC
BOOT_CFG1[4]	bit4	Fast Boot Suppo	0	0	0:Normal;1:Fast Boot

图 2-4

Fuse	SBMR1	定义	默认值	设置值	默认值定义
BOOT_CFG1[3:2]	bit3_2	SD 速度模式	00	00	0x:High/Normal;10:SDR50 ; 11:SDR104
BOOT_CFG1[1]	bit1	SD Power Cycle	0	0	0:No power cycle 1:Power cycle enabled vis SD_RST pad (on USDHC3/4)
BOOT_CFG1[0]	bit0	SD Loopback Clock SOURCE Sel (SDR50/104 only)	0	0	0:through SD pad 1:direct
BOOT_CFG2[7:6]	bit15_14	SD Calibration Step	00	00	'00'-1 TBD
BOOT_CFG2[5]	Bit13	Bus Width	0	1	0:1bit 1:4bit
BOOT_CFG2[4:3]	bit12_11	启动设备接口	00	00	00~11 : USDHC1~4
BOOT_CFG2[2]	bit10	BOOT Frequencies	0	0	0:500/400 MHz 1:250/200 MHz
BOOT_CFG2[1]	bit9	SD1 VOLTAGE SELECTION	0	0	0:3.3V 1:1.8V
BOOT_CFG2[0]	bit8	Override Pad Setting	0	0	0:Use the defaule valuse 1:Use PAD_SETTINGS values.

图 2-5

说明：目前只操作了普速卡。如若把 BOOT_DFG2[1]设为 1，GPIO_IO5 设为 1，VDD_SD1 输出 3.3V，板子可以正常启动。

缺陷:当操作高速卡时，即把 BOOT_DFG2[1]设为 1,GPIO1_IO5 为 0,VDD-SD1 输出为 1.8V，终端会打印信息提示提供电压不足，详见附录 2。

2.1 tf 卡启动原理及实现

我们刚刚交代了 tf 卡(接到端口 uSDHC1)和 emmc(接到端口 uSDHC2)启动的原理，为了更改为 TF 卡启动，我们必须去更改配置电阻，那么是否有更好的办法实现 TF 卡启动呢？经过我们仔细查看相关手册，我们发现，在 Fuses 启动(BOOT_MODE[1:0] = 00b)这个模式下，默认的启动就是从 uSDHC1 启动：

Table 8-15. USDHC Boot eFUSE Descriptions

Fuse	Config	Definition	GPIO	Shipped value	Settings
0x450[7:6]	OEM	Boot Device Selection	Yes	00	01 - Boot from USDHC Interface
0x450[5]	OEM	SD/MMC Selection	Yes	0	0 - SD/eSD/SDXC 1 - MMC/eMMC
0x450[4]	OEM	Fast Boot Support	Yes	0	0 - Normal Boot 1 - Fast Boot

Table continues on the next page...

Fuse	Config	Definition	GPIO	Shipped value	Settings
0x450[3:2]	OEM	SD/MMC Speed Mode, and eMMC Acknowledge Enabled Selection	Yes	00	MMC 0x – Normal Speed Mode 1x - High Speed Mode x0 - eMMC Fast boot acknowledge enable x1 - eMMC Fast boot acknowledge disable SD 00 – Normal/SDR12 01 – High/SDR25 10 - SDR50 11 - SDR104
0x450[1]	OEM	SD Power Cycle Enable/ eMMC Reset Enable	Yes	0	MMC 0 – No action 1 - eMMC reset enabled via SD_RST pad SD 0 - No power cycle 1 - Power cycle enabled via SD_RST pad

0x450[0]	OEM	SD Loopback Clock Source Sel(for SDR50 and SDR104 only)	Yes	0	0 - through SD pad 1 - direct
0x450[15:13]	OEM	SD MMC Bus Width Selection /SD Calibration Step	Yes	00	SD/eSD/SDXC (BOOT_CFG1[5]=0) Bus Width xx0 - 1-bit xx1 - 4-bit SD Calibration Step 00x - 1 delay cells 01x - 1 delay cells 10x - 2 delay cells 11x - 3 delay cells MMC/eMMC (BOOT_CFG1[5]=1) 000 - 1-bit 001 - 4-bit 010 - 8-bit

Table continues on the next page

Fuse	Config	Definition	GPIO	Shipped value	Settings
					101 - 4-bit DDR (MMC 4.4) 110 - 8-bit DDR (MMC 4.4) Else - reserved.
0x450[12:11]	OEM	USDHC Port Selection	Yes	00	00 - USDHC-1 01 - USDHC-2 1x - Reserved
0x450[9]	OEM	USDHC1 Voltage selection	Yes	0	0 - 3.3V 1 - 1.8V
0x460[31:30]	OEM	Power Cycle Selection	Yes	00	00 - 20ms 01 - 10ms 10 - 5ms 11 - 2.5ms
0X460[29]	OEM	Power Stable Cycle Selection	Yes	0	0 - 5ms 1 - 2.5ms
0X460[24]	OEM	SD/MMC DLL Enable Selection	Yes	0	0 - Disable DLL for SD/eMMC 1 - Enable DLL for SD/Emmc
0X470[7]	OEM	DLL Override Selection	Yes	0	0 – Not Override 1 - DLL Override Mode for SD/eMMC (Override by MMC_DLL_DLY, 0x470[19:16])

3 软件支持

要把板子跑起来，无非就是包括 U-boot、dtb、Kernel 和跟文件系统均写入正确，此次 TF 启动只修改 U-boot 和 dtb 的部分代码。

3.1 U-boot

u-boot 如果要支持 TF 卡启动，则启用 USDHC1，注释 USDHC2。修改 uboot-imx/include/configs/mx6ul_arm2.h，修改后如下图 3-1。

```
416 /*evb-p6ul usdhc1(MicroSD card) boot,2017/01/09*/
417 #define CONFIG_SYS_MMC_ENV_DEV          0   /* USDHC1*/
418 #define CONFIG_SYS_MMC_ENV_PART         0   /* boot area for uboot&dtb*/
419 #define CONFIG_MMCROOT                  "/dev/mmcblk0p2"  /* rootfs area*/
420
421 /*evb-p6ul usdhc2(emmc) boot,embedall,2017/01/09*/
422 /*#define CONFIG_SYS_MMC_ENV_DEV                1   USDHC2 */
423 /*#define CONFIG_SYS_MMC_ENV_PART               0   boot area for uboot&dtb */
424 /*#define CONFIG_MMCROOT                        "/dev/mmcblk1p2"   USDHC2 */
425
```

图 3-1

然后，修改 uboot-imx/board/freescale/mx6ul_14x14_ddr3_arm2/mx6ul_14x14_ddr3_arm2.c 将 USDHC1 设为高电平。如图 3-2 所示。

```
632 int board_mmc_init(bd_t *bis)
633 {
634         int i;
635
636         /*
637          * According to the board_mmc_init() the following map is done:
638          * (U-boot device node)    (Physical Port)
639          * mmc0                    USDHC1
640          * mmc1                    USDHC2
641          */
642         for (i = 0; i < CONFIG_SYS_FSL_USDHC_NUM; i++) {
643                 switch (i) {
644                 case 0:
645
646                         imx_iomux_v3_setup_multiple_pads(
647                                 usdhc1_pads, ARRAY_SIZE(usdhc1_pads));
648                         gpio_direction_input(USDHC1_CD_GPIO);
649
650                         usdhc_cfg[0].sdhc_clk = mxc_get_clock(MXC_ESDHC_CLK);
651                         /* 3.3V */
652                         gpio_direction_output(USDHC1_VSELECT, 1);
653                         gpio_direction_output(USDHC1_PWR_GPIO, 1);
654                         break;
```

图 3-2

3.2 内核

EVB-P6UL 采用 3.14.38 版本的 Linux 内核，驱动描述使用 dts。因存储媒质及应用接口不同，对应的 dtb 也不同，EVB-P6UL 评估板主要有 5 路 UART（双网络）和 8 路 UART（单网络）两种，另外有使能 CAMERA 接口产品等，为了便于管理这些 dtb 文件，我们采用关键字区别法来区别。例如："CSI"、"UART"、"RES"、"CAP"来判定应该烧写那个 dtb，命名规则举例说明：

- evbp6ul-m256f512-res-8uarts.dtb：

  m256，即 memory 256MB；f512，即 Nand Flash 512MB；res，电阻式触摸屏；8uarts，8 路 UART（单网络）。

- evbp6ul-m256s64-res-5uarts.dtb：

  m256，即 memory 256MB；s64 表示 QSPI Nor Flash 为 64M；res 表示电容式触摸屏；5uarts 表示 5 路 UART（双网络）。

- evbp6ul-m256e2g-cap-5uarts.dtb：

  m256，即 memory 256MB；e2g 表示 emmc Flash 为 2G；cap 表示电容式触摸屏；5uarts 表示 5 路 UART（双网络）。

说明：选择正确的 dtb 才能使板子正常工作，具体可看/fsl-release-bsp/Makefile,然后选择对应的 dtb 编译。

确认 dts 中 usdhc1（EVB-P6UL 硬件设计 usdhc1 为 TF 卡）的驱动描述，如下图 3-3、图 3-4、图 3-5 所示。

图 3-3

代码清单如下：

```
&usdhc1 {
pinctrl-names = "default", "state_100mhz", "state_200mhz";
pinctrl-0 = <&pinctrl_usdhc1>;
pinctrl-1 = <&pinctrl_usdhc1_100mhz>;
pinctrl-2 = <&pinctrl_usdhc1_200mhz>;
cd-gpios = <&gpio1 19 0>;
keep-power-in-suspend;
```

```
enable-sdio-wakeup;
vmmc-supply = <&reg_sd1_vmmc>;
status = "okay";
};
```

```
reg_sd1_vmmc: regulator@1 {
        compatible = "regulator-fixed";
        regulator-name = "VSD_3V3";
        regulator-min-microvolt = <3300000>;
        regulator-max-microvolt = <3300000>;
        gpio = <&gpio1 5 GPIO_ACTIVE_HIGH>;
        enable-active-high;
};
```

图 3-4

代码清单如下：

```
reg_sd1_vmmc: regulator@1 {
        compatible = "regulator-fixed";
        regulator-name = "VSD_3V3";
        regulator-min-microvolt = <3300000>;
        regulator-max-microvolt = <3300000>;
        gpio = <&gpio1 5 GPIO_ACTIVE_HIGH>;
        enable-active-high;
};
```

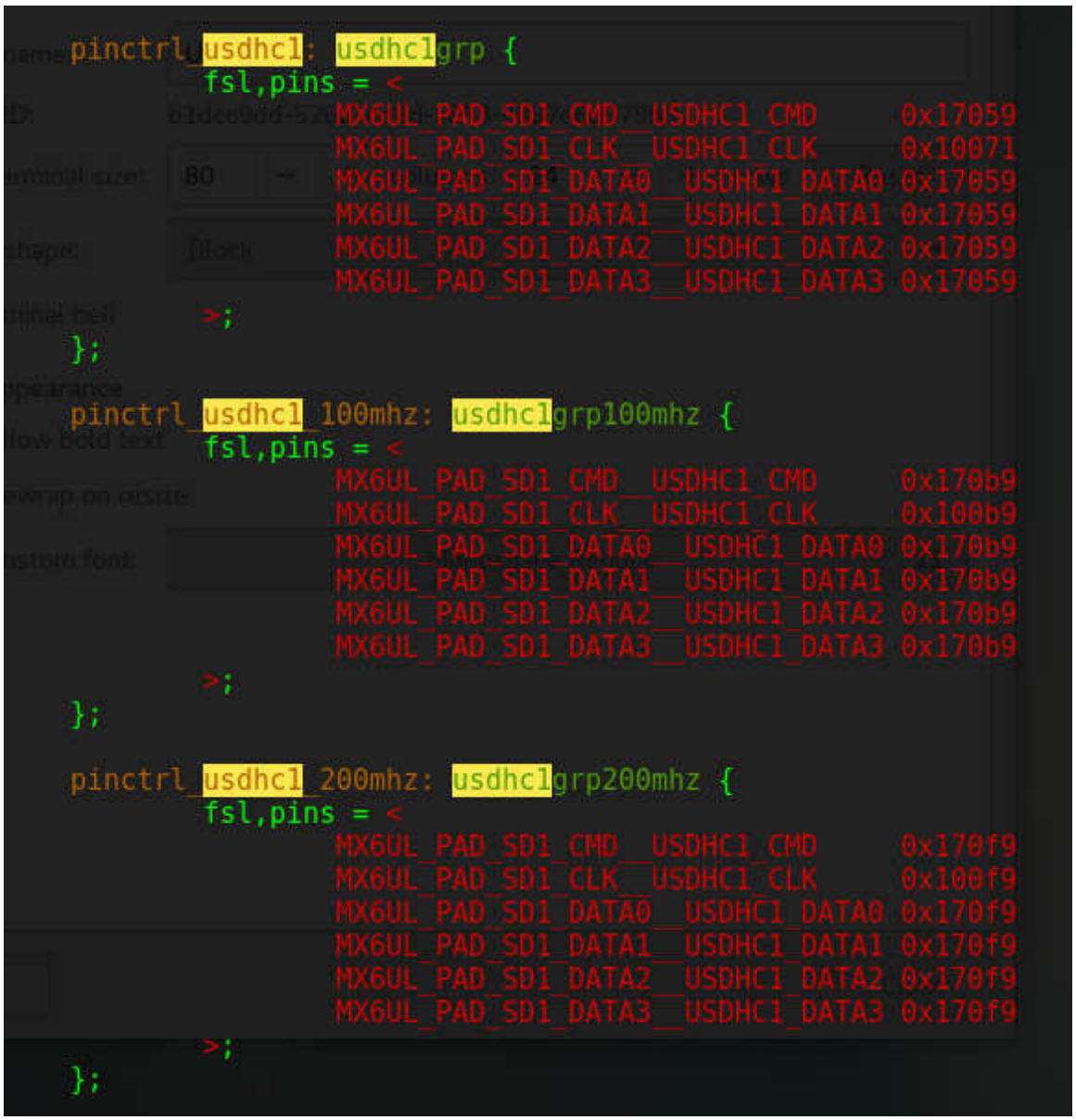


图 3-5

代码清单如下：

```
pinctrl_usdhc1: usdhc1grp {
    fsl,pins = <
        MX6UL_PAD_SD1_CMD__USDHC1_CMD       0x17059
        MX6UL_PAD_SD1_CLK__USDHC1_CLK       0x10071
        MX6UL_PAD_SD1_DATA0__USDHC1_DATA0 0x17059
        MX6UL_PAD_SD1_DATA1__USDHC1_DATA1 0x17059
        MX6UL_PAD_SD1_DATA2__USDHC1_DATA2 0x17059
        MX6UL_PAD_SD1_DATA3__USDHC1_DATA3 0x17059
    >;
};
```

```
pinctrl_usdhc1_100mhz: usdhc1grp100mhz {
        fsl,pins = <
                MX6UL_PAD_SD1_CMD__USDHC1_CMD       0x170b9
                MX6UL_PAD_SD1_CLK__USDHC1_CLK       0x100b9
                MX6UL_PAD_SD1_DATA0__USDHC1_DATA0 0x170b9
                MX6UL_PAD_SD1_DATA1__USDHC1_DATA1 0x170b9
                MX6UL_PAD_SD1_DATA2__USDHC1_DATA2 0x170b9
                MX6UL_PAD_SD1_DATA3__USDHC1_DATA3 0x170b9
        >;
};

pinctrl_usdhc1_200mhz: usdhc1grp200mhz {
        fsl,pins = <
                MX6UL_PAD_SD1_CMD__USDHC1_CMD       0x170f9
                MX6UL_PAD_SD1_CLK__USDHC1_CLK       0x100f9
                MX6UL_PAD_SD1_DATA0__USDHC1_DATA0 0x170f9
                MX6UL_PAD_SD1_DATA1__USDHC1_DATA1 0x170f9
                MX6UL_PAD_SD1_DATA2__USDHC1_DATA2 0x170f9
                MX6UL_PAD_SD1_DATA3__USDHC1_DATA3 0x170f9
        >;
};
```

4 TF 卡启动 Linux

EVB-P6UL 可选用 MfgTools 或 Linux 主机（PC，x86 平台）烧录 TF 卡。Linux 内核镜像包括 u-boot、zImage、DTB 和根文件系统。u-boot 裸写到 TF 卡，zImage 与 DTB 存储到 u-boot 之后的 FAT 分区，根文件系统存储到内核分区之后的 ext3/ext4 分区。

以下两节，将分别介绍 Linux 主机和 MFGTools 制作 TF。

4.1Linux 主机制作 TF 卡

Linux 主机制作 TF 卡，使用脚本完成。步骤如下：

1、将 p6ultfboot.tar.bz2 复制到 Linux 主机，并解压，指令如下

```
$tar -jxf p6ultfboot.tar.bz2
```

解压后得到图 4-1 所示的文件：

图 4-1

其中，**mksdcard.sh** 为制作 TF 卡的脚本文件；**rootfs-p6ul-mmc.tar** 为根文件系统，该文件根据需要替换，文件名为格式 **rootfs*.tar**；sdboot 目录下的 **u-boot.imx、zImage、evbp6ul.dtb** 分别为 u-boot、内核镜像和 dtb 文件。

2、使用 root 用户（权限）执行制作脚本 createSdcard.sh

```
$ sudosh createSdcard.sh
```

提示选择设备号时，根据实际情况输入“#”所在列的数值。若同时有多个 TF 卡，设备号与设备对应（name 对应的列），制作 TF 卡时不建议同时插入多张卡。如图 4-2 所示。

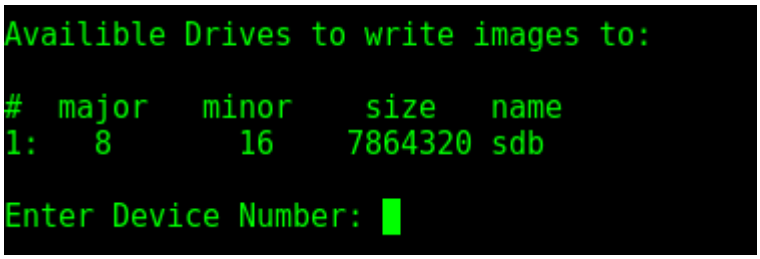


图 4-2

提示重分区是，输入“y”。如图 4-3 所示。

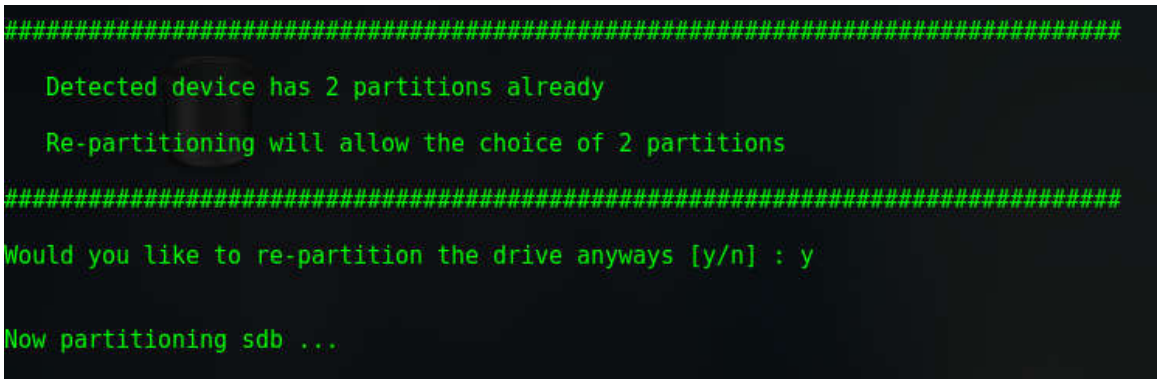


图 4-3

当制作完成，显示如图 4-4 所示信息。

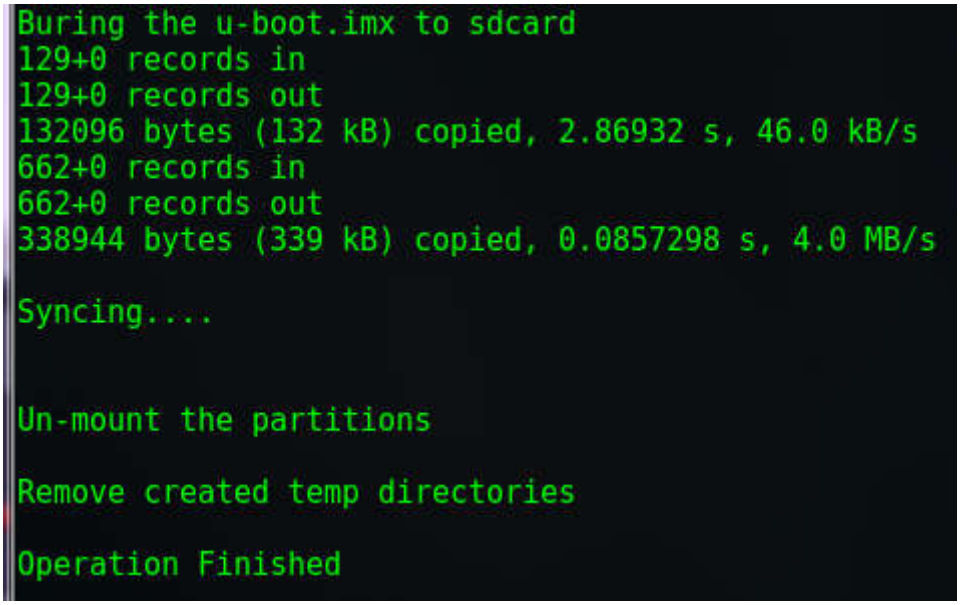


图 4-4

说明：除了图 4-4 所提示的信息，还应查看脚本执行的所有信息是否有错误提示。

4.2 MfgTools 烧录 TF 卡

使用 MfgTools 制作 TF 启动卡的步骤如下：

MfgTools 的烧录脚本分为 QT 与无 QT 两种，分别如下：

mfgtool2-linux-mx6ul-console-sd （无 QT）

mfgtool2-linux-mx6ul-qt4-sd （QT4.8）

1、拨码开关 SW5 切换至下载模式（Serial Downloader），连接电源、debug console、USB device 接口（CN11）。上电，运行烧录脚本（例如，mfgtool2-linux-mx6ul-console-sd），当 USB device 连接成功后插入 TF 卡，USB device 与 PC 连接成功，MfgTools 出现“**HID-compliant device**”，如图 4-5 所示。

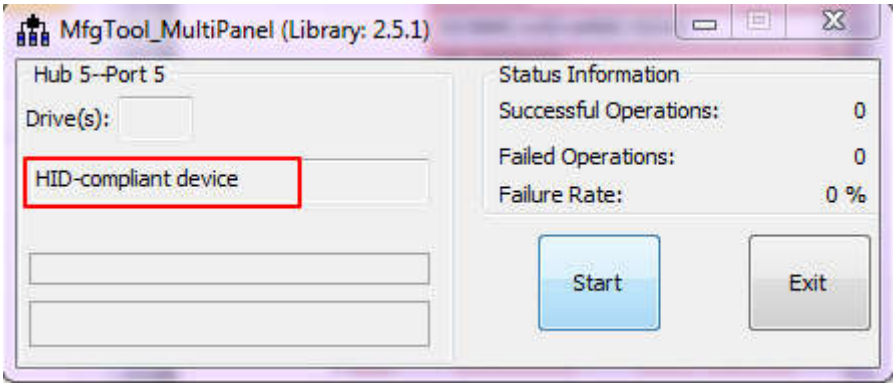


图 4-5

说明：由于启动设备优先级的原因，TF 卡需在 MfgTools 连接成功后安装，否则出现 MfgTools 不连接的现象，如图 4-6 所示。若不了解接线以及拨码开关的使用，请参考产品开发光盘中《EVB-P6UL linux 系统烧录手册》。

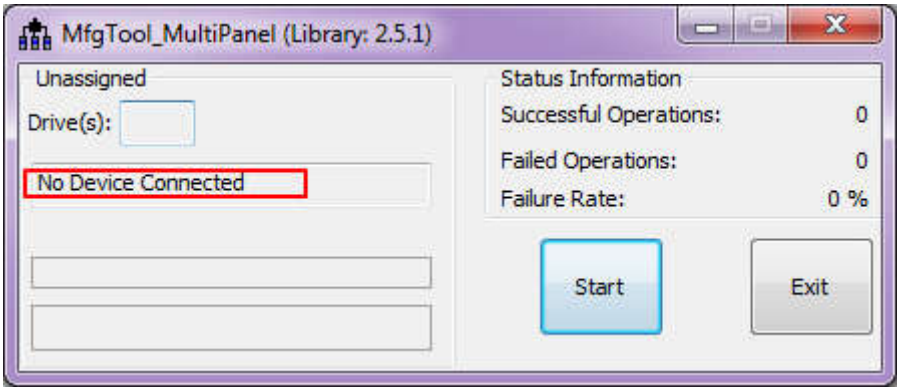


图 4-6

2、点击 MfgTools 的“start”按钮开始烧录，此时 debug console（用超级终端、putty 此类软件查看）打印 Linux 系统启动、烧录等信息。当弹出如图 4-7 的格式化对话窗时，cancel 或关闭。

图 4-7

3、烧录完成，弹出图 4-8 窗口。

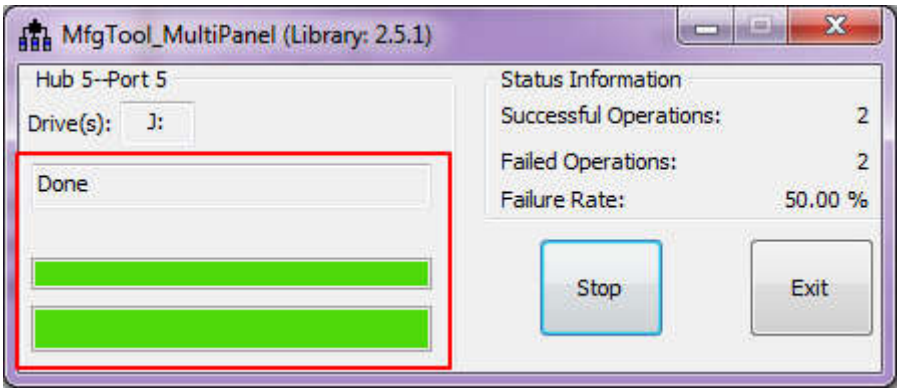


图 4-8

4、断电，拨码开关切换至“00”（Boot From Fuses），从 TF 卡引导 Linux。

说明：关于 i.MX6UL 从 TF 卡引导的介绍，参考附录 1。

5 免责声明

本手册所陈述的产品文本及相关软件版权均属深圳市盈鹏飞科技有限公司所有，其产权受国家法律保护，未经本公司授权，其它公司、单位、代理商及个人不得非法使用和拷贝。

您若需要我公司产品及相关信息，请及时与我们联系，我们将热情接待。

深圳盈安信科技有限公司将会不断地完善本手册的相关技术内容，请客户适时从公司网站下载最新版本，不再另行通知。

附录 1

i.MX6UL 有 4 种不同的启动模式，其中一种为保留模式，剩余三种可用。Serial Downloader 模式用于 UART 或 USB OTG 下载，Internal Boot 模式为正常启动模式。在 EVB-P6UL 上，通过拨码开关（2-bit）切换模式，拨码开关的配置如表 1A-1 所示。

表 1A-1

BOOT_MODE[1:0]	Boot Type
00	Boot From Fuses
01	Serial Downloader
10	Internal Boot（normal boot）
11	Reserved

i.MX6UL 的启动流程如图 A-1 所示：

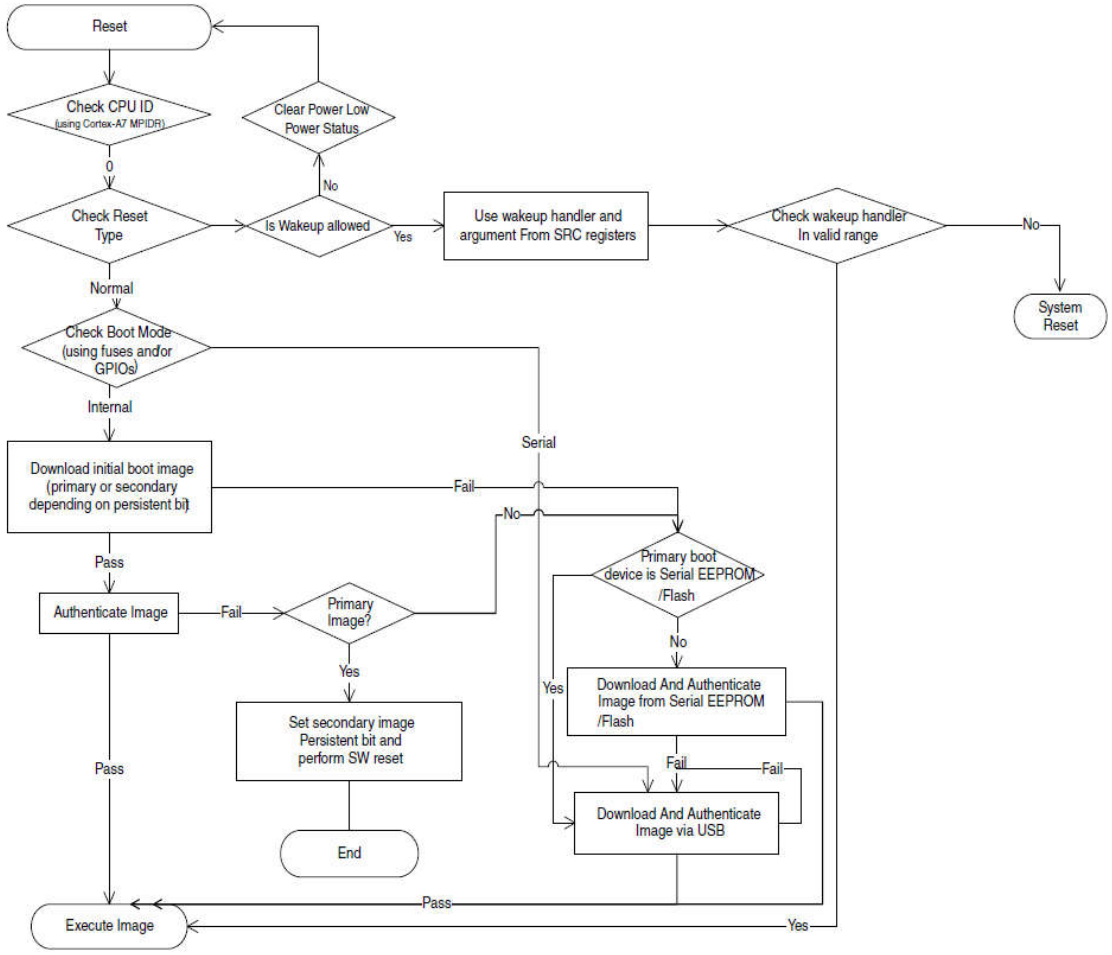


图 1A-1：i.MX6UL 启动流程

Boot From Fuses 模式覆盖 GPIO 启动配置，启动 ROM code 仅使用 eFUSE。当 Internal Boot

和 recover boot 失效，SDMMC_MFG_DISABLE fuse 位不设置，以及设置 EEPROM Recovery fuse 位，启动模式将在 Serial Downloader 模式进入 SD/MMC manufacture 模式。在 manufacture 模式中，使用 SD/MMC 1-bit 总线而不管 fuse 设置。

工作于 manufacture 模式，SD 或 MMC 卡将扫描 uSDHC1 接口。如果检测到设备（SD 卡）且卡内有有效的镜像，系统将向将会被加载运行。SD/MMC manufacture 模式启动流程如图 1A-2 所示。

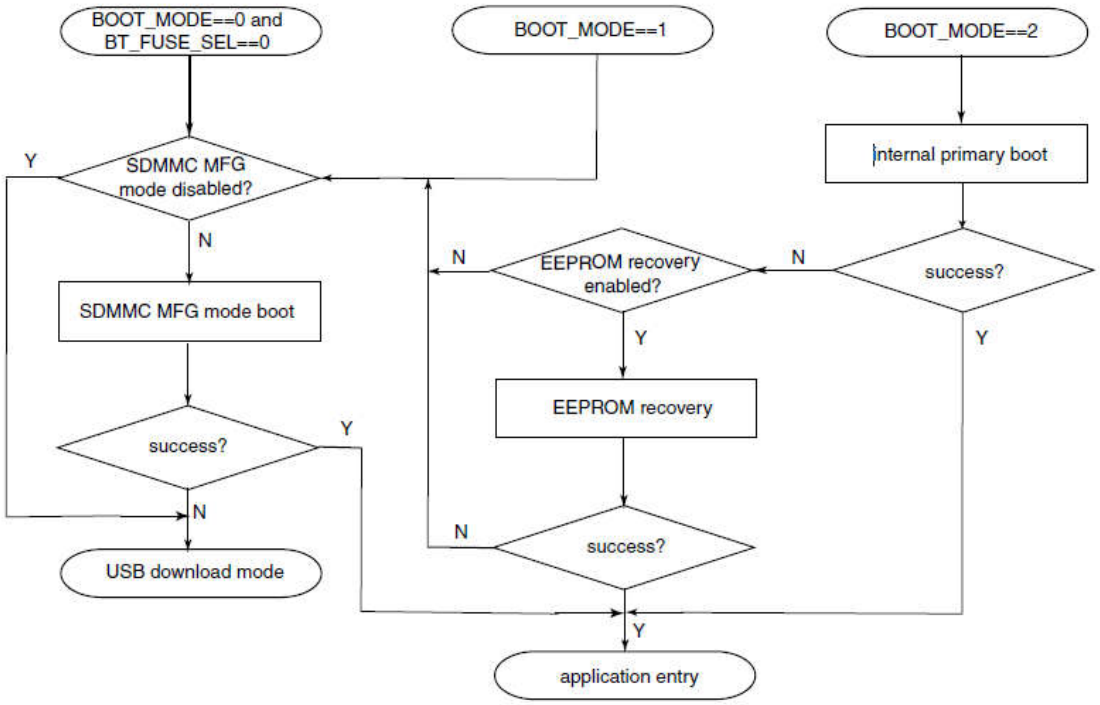
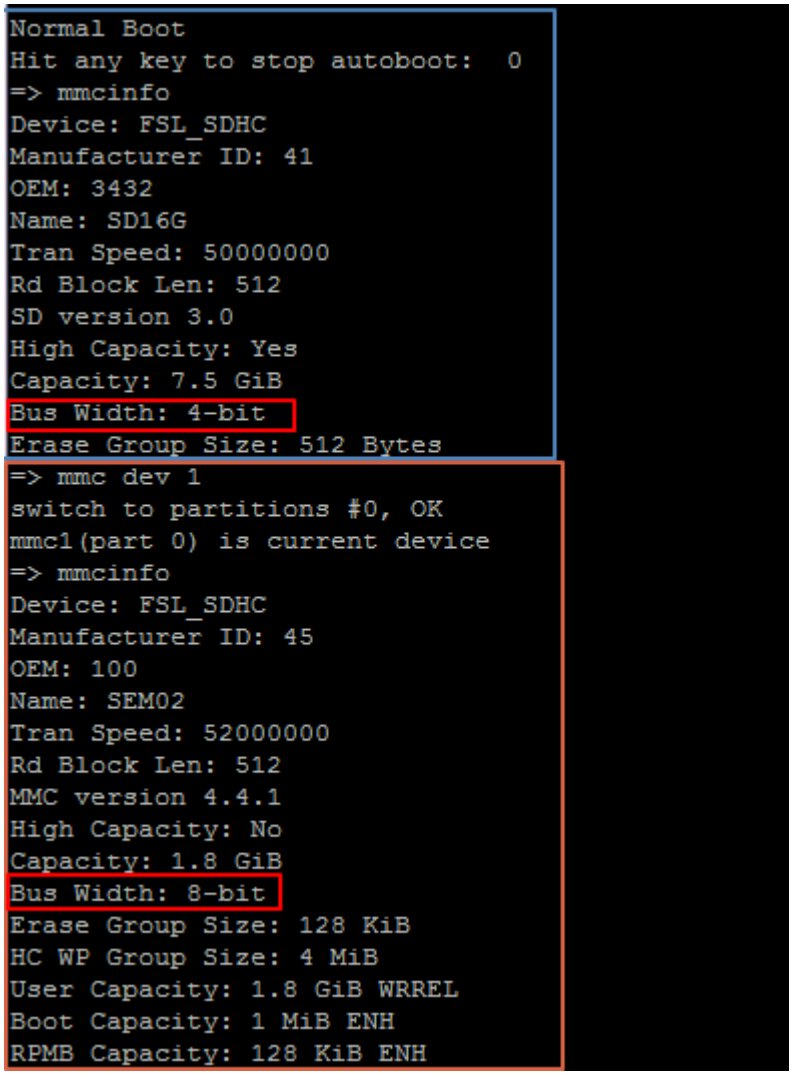


图 1A-2

附录 2

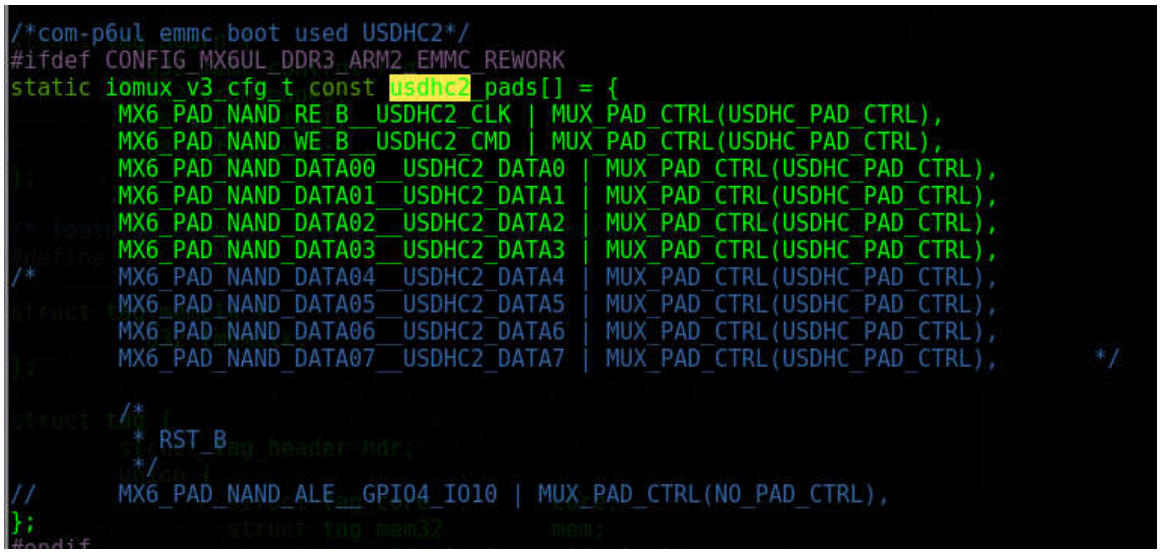
读取 eMMC 位宽（bus width）

在 u-boot 中，输入 mmcinfo 即可读取 eMMC 的位宽，如下图所示：



图中信息，SD 卡（图片蓝色矩形框标注）的位宽为 4-bit，eMMC（图片橙色矩形框标注）的位宽为 8-bit。以上信息，修改 u-boot 源码与 GPIO 配置电阻后所得到，修改部分如下：

修改 u-boot 源码 board/freescale/mx6ul_14x14_ddr3_arm2/mx6ul_14x14_ddr3_arm2.c 中的文件，如下图所示：



代码清单如下：

```
/*com-p6ul emmc boot used USDHC2*/
#ifdef CONFIG_MX6UL_DDR3_ARM2_EMMC_REWORK
static iomux_v3_cfg_t const usdhc2_pads[] = {
        MX6_PAD_NAND_RE_B__USDHC2_CLK | MUX_PAD_CTRL(USDHC_PAD_CTRL),
        MX6_PAD_NAND_WE_B__USDHC2_CMD | MUX_PAD_CTRL(USDHC_PAD_CTRL),
        MX6_PAD_NAND_DATA00__USDHC2_DATA0 | MUX_PAD_CTRL(USDHC_PAD_CTRL),
        MX6_PAD_NAND_DATA01__USDHC2_DATA1 | MUX_PAD_CTRL(USDHC_PAD_CTRL),
        MX6_PAD_NAND_DATA02__USDHC2_DATA2 | MUX_PAD_CTRL(USDHC_PAD_CTRL),
        MX6_PAD_NAND_DATA03__USDHC2_DATA3 | MUX_PAD_CTRL(USDHC_PAD_CTRL),
/*      MX6_PAD_NAND_DATA04__USDHC2_DATA4 | MUX_PAD_CTRL(USDHC_PAD_CTRL),
        MX6_PAD_NAND_DATA05__USDHC2_DATA5 | MUX_PAD_CTRL(USDHC_PAD_CTRL),
        MX6_PAD_NAND_DATA06__USDHC2_DATA6 | MUX_PAD_CTRL(USDHC_PAD_CTRL),
        MX6_PAD_NAND_DATA07__USDHC2_DATA7 | MUX_PAD_CTRL(USDHC_PAD_CTRL),      */

        /*
         * RST_B
         */
//      MX6_PAD_NAND_ALE__GPIO4_IO10 | MUX_PAD_CTRL(NO_PAD_CTRL),
};
#endif
```

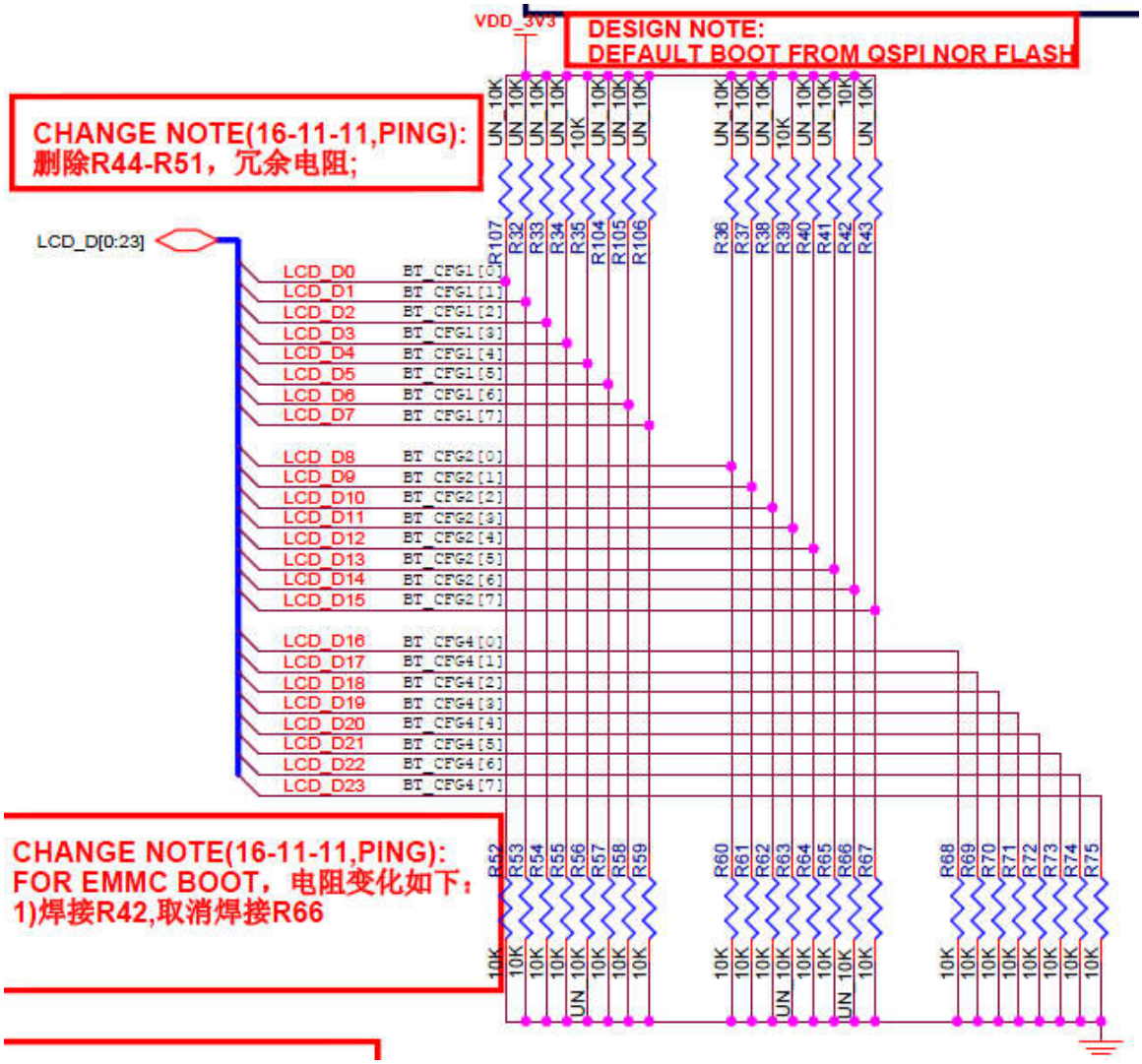
{USDHC2_BASE_ADDR, 0, **8**},改为{USDHC2_BASE_ADDR, 0, **4**},

更改 GPIO 配置电阻，使得 eMMC 的位宽为 4-bit。

Table 5-6. MMC/eMMC Boot Fusemap

Addr	7	6	5	4	3	2	1	0
0x450[7:0] (BOOT_CFG1)	0	1	1	Fast Boot: 0 - Regular 1 - Fast Boot	SD/MMC Speed 0 - High 1- Normal	Fast Boot Acknowledge Disable: 0 - Boot Ack Enabled 1 - Boot Ack Disabled	SD Power Cycle Enable 0 - No power cycle 1 - Enabled via USDHC_RST pad	SD Loopback Clock Source Sel (for SDR50 and SDR104 only) 0 - through SD pad 1 - direct
0x450[15:8] (BOOT_CFG2)	Bus Width: 000 - 1-bit 001 - 4-bit 010 - 8-bit 101 - 4-bit DDR (MMC 4.4) 110 - 8-bit DDR (MMC 4.4) Else - Reserved			Port Select: 00 - eSDHC1 01 - eSDHC2 10 - Reserved 11 - Reserved		Boot Frequencies (ARM/DDR) 0 - 500 / 400 MHz 1 - 250 / 200 MHz	SD VOLTAGE SELECTION 0 - 3.3V 1 - 1.8V	Reserved

即，COM-P6UL 中 **BOOT_CFG2[7]** 、**BOOT_CFG2[6]**、**BOOT_CFG2[5]**分别设为 0、0、1（0 表示低电平，1 表示高电平）。如下图所示：



但是，修改过后的，在 u-boot 中并未显示 eMMC 为 4-bit。

此问题留待后续解决

附录 3

TF 卡启动流程

符合 SD3.0 标准的设备，上电时从 TF 卡控制器获取总线电源（SD bus power）。i.MX6UL 启动时获取数据位宽的 fuse 配置、软复位 eSHDC，设置大约 400KHz 频率，具体流程如图 1-5、图 1-6、图 1-7 所示。

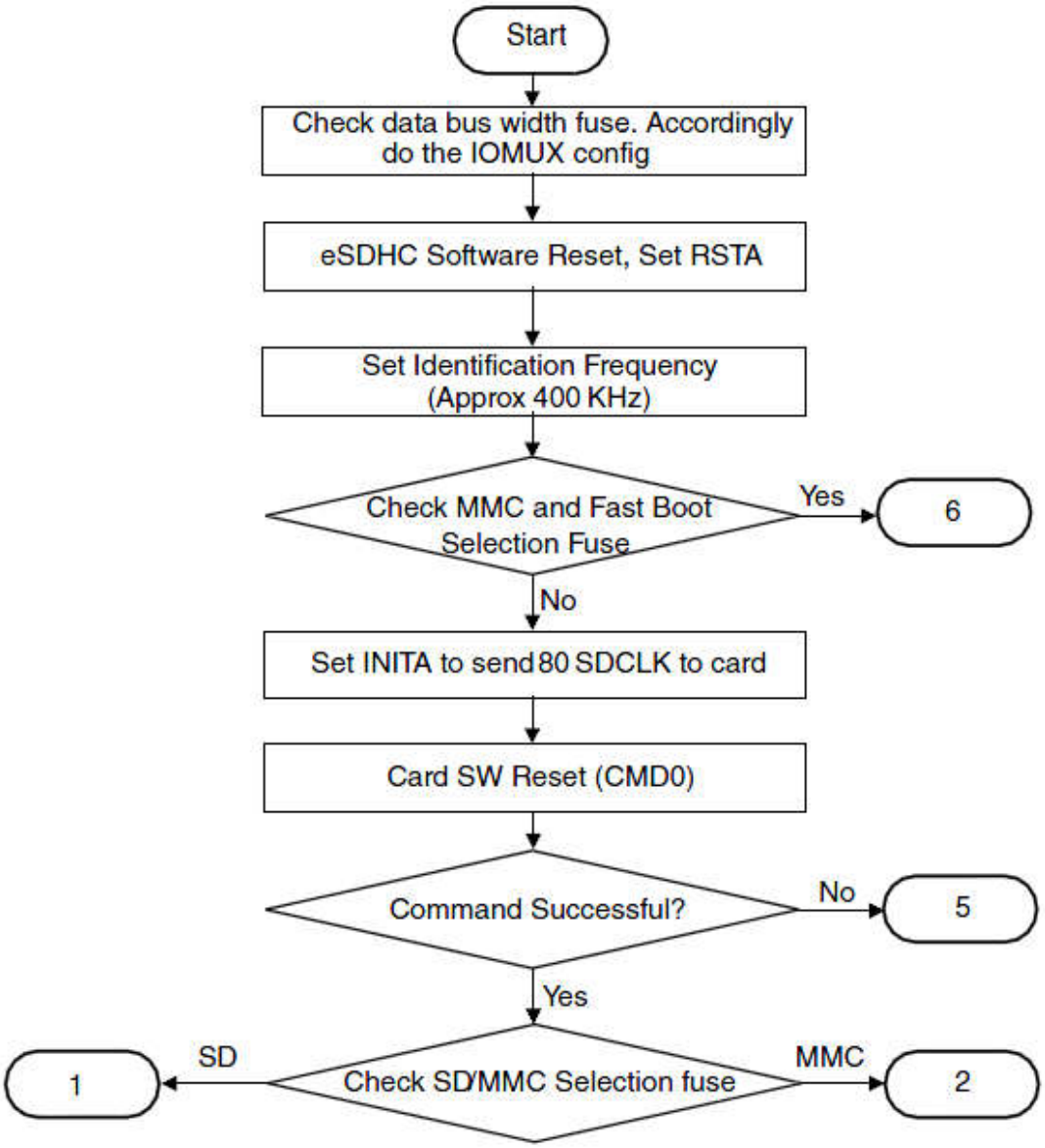


图 1-5 i.MX6UL MMC 与 eMMC 启动

说明：当启动设备为 SD 时，跳转到“1”，见图 1-6。

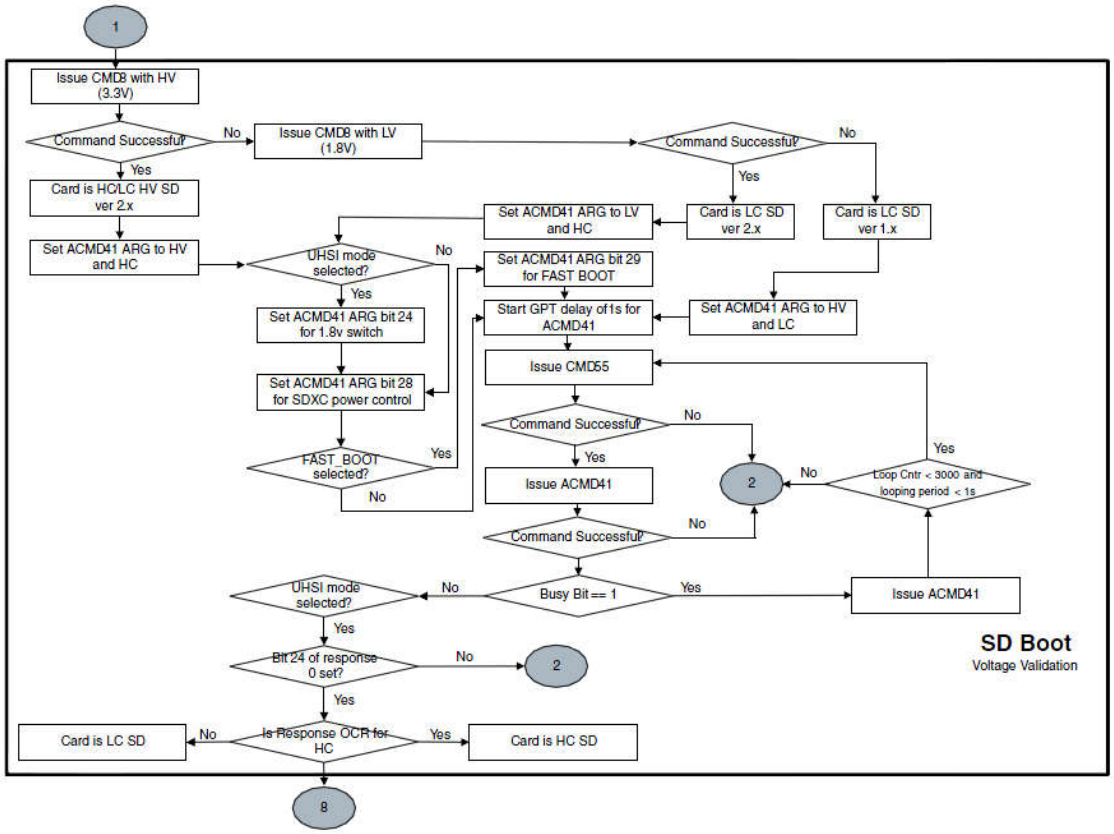


图 1-6 SD 卡电压确认

说明：当 OCR（Operation Conditions Register）得到主机控制的响应时，跳转到流程 8，见图 1-7。

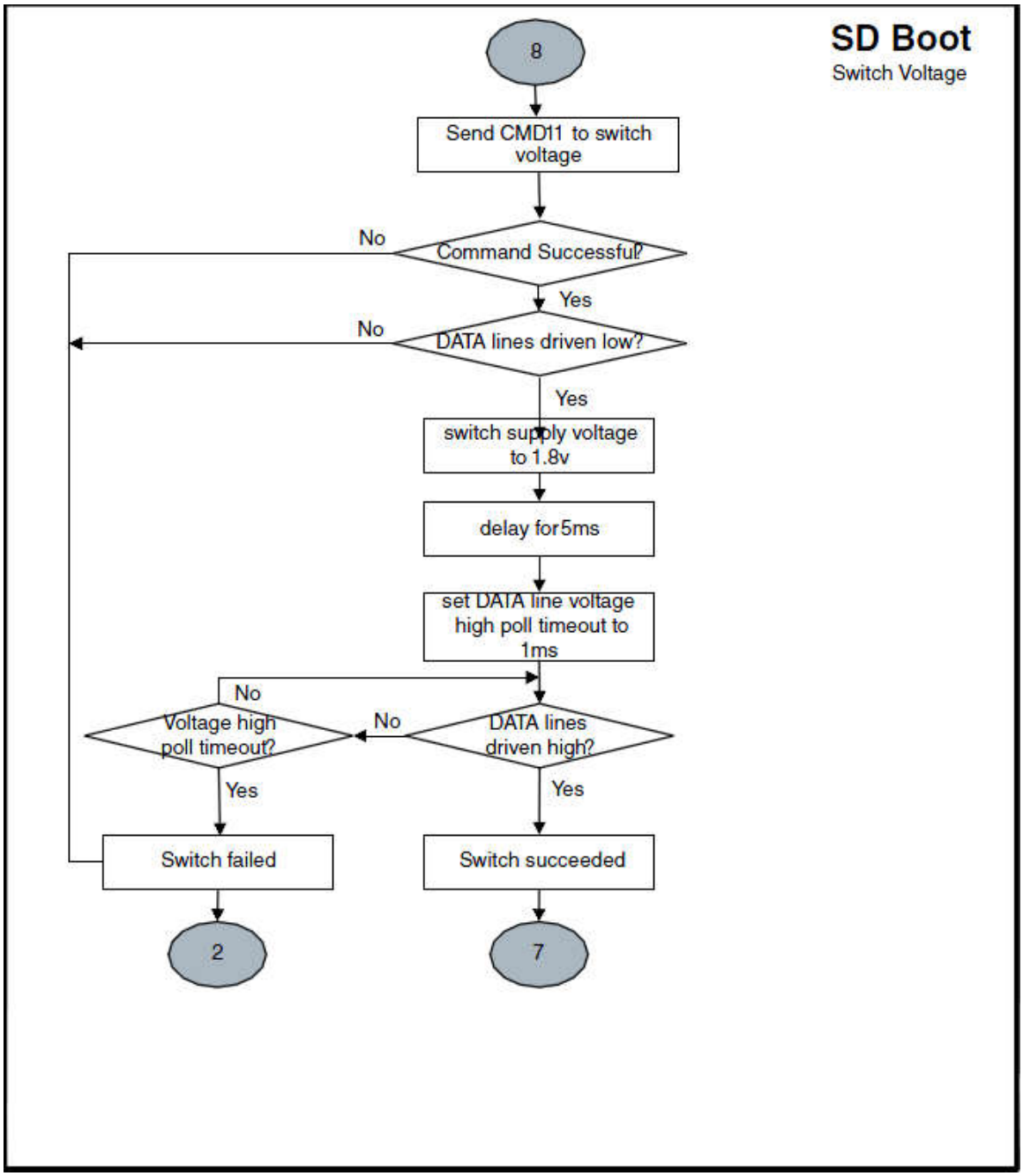


图 1-7 SD 卡电平切换

所有 SD 卡初始化电平均为 3.3V，使用命令，切换 SDHC 与 SDXC 卡至 1.8V 工作。初始化在上电或插入卡时，主机根据电平选择 SPI 总线或者 1-bit SD 总线。此后，主机可能发送指令切换至 4-bit SD 总线接口，如果 SD 卡支持该模式。对于各种类型的卡，可选或强制工作于 4-bit SD 总线模式。当检测 SD 支持 4-bit 总线，主机再发送指令，尝试切换至更高速率。