

用户手册

[EVK-SM9300]

历史版本

Rev.	Note	Author
20161107	Initial	Sandy
20161228	Rev01	Sandy
20170407	Rev1.1: 1. Add Wifi & Bluetooth support 2. Update HDMI support	Sandy
20170725	Rev1.2.0r01 1. Update HDMI support 2. Add HDMI Audio support 3. Add PCIE networking card support 4. Change Camera module to EXP-C5DX00 See Detail in Release Note	Sandy
20170818	Rev1.2.1r01 1. Add Logo support 2. Add HDMI test methods See Detail in Release Note	Sandy

目录

历史版本	2
目录	3
Release Note	5
1. Image Version	5
2. 功能列表	5
3. Previous Version	7
第 1 章 快速启动	7
1.1 烧写镜像到 SD 卡	8
1.2 从 SD 卡启动系统	9
1.3 从 EMMC 启动系统	10
1.4 启动 LOGO	10
第 2 章 功能测试	12
2.1 LED 测试	12
2.2 RTC 测试	12
2.3 EEPROM 测试	13
2.3.1 EEPROM on SOM-SM9300	13
2.3.2 EEPROM on BB-ESDSM00	14
2.4 EMMC 测试	15
2.5 音频测试	15
2.6 LVDS 测试	16
2.7 背光测试	16
2.8 触摸屏测试	16
2.9 VGA 测试	16
2.10 HDMI 测试	17
2.10.1 HDMI 启动	17
2.10.2 HDMI 显示图像	17
2.10.3 HDMI 音频播放	17
2.11 串口测试	17
2.11.1 UART4 测试	17
2.12 RS485 测试	18
2.13 网络测试	18

2.14	PCIe 网卡测试	18
2.15	CAN 测试	19
2.16	USB 测试	20
2.16.1	Host 测试	20
2.16.2	OTG 测试	20
2.17	SATA 测试	21
2.18	Camera 测试	21
2.19	WIFI 测试	22
2.19.1	配置 WIFI	22
2.19.2	连接 WIFI	22
2.19.3	WIFI Host AP 功能	23
2.20	Bluetooth 测试	23
2.20.1	初始化蓝牙模块	23
2.20.2	测试蓝牙功能	24
第 3 章	系统编译	25
3.1	配置编译环境	25
3.2	编译 UBOOT	25
3.2.1	获取 uboot 源码	25
3.2.2	编译并烧写镜像到 SD 卡	25
3.2.3	编译并烧写镜像到 EMMC	25
3.3	Kernel	27
3.3.1	获取内核源码	27
3.3.2	编译并烧写镜像到 SD 卡	27
3.3.3	编译并烧写镜像到 EMMC 卡	27
3.4	File system	27
3.4.1	烧写 File system 到 SD 卡	27
3.4.2	烧写 File system 到 EMMC	27

Release Note

1. Image Version

EVK-SM9300-Image-SDcard-V1.2.1r01.img

EVK-SM9300-Image-EMMC-V1.2.1r01.img

2. 功能列表

EVK-SM9300	
Feature List	On-Chip Peripherals On-Board Peripherals Detail Functions(existing)
u-boot version	2015.04
kernel version	linux-3.14
Filesystem	filesystem
boot configuration	boot configuration NA boot from tf card & eMMC
CPU	imx6q
DDR3	DDR3 256M DDR3/1066 *4
eMMC	4G SD 4G bytes eMMC
EEPROM	I2C1 AT24C256W eeprom read and write
SATA	SATA connector 2*SATA
Integrated RTC	RTC NA
PMIC	
GPIO	GPIO connector
SDIO2	SDIO2 TF CARD
RMII	RMII AR8305 RJ45

usb_1	USB	USB2514(USB*4 hub)	
USB OTG	USB OTG	USB OTG	
CAN_1	CAN1	ADUM1201	CAN*2
CAN_2	CAN2		
RS485	UART0	ADM2483	
UART1	UART0	UART	
UART2	UART1	MAX3232/Bluetooth	DB9
UART4	UART5	MAX3232	
I2C0	I2C	I2C	
PCI-E	I2C4+PCI-E	PCI-E	connector
SIM	NA	NA	
Push button	button	button	
SPI	SPI1	connector	
MIPI	MIPI	connector	external camera
CSI	CSI	connector	external camera
WIFI	SDIO2	connector	
LVDS	LVDS	connector	
HDMI	HDMI	connector	
LCD	LCD	CH7026B	VGA connector
Touchscreen	SPI1	TSC2046	
Audio	I2S	WM8904	
LED	GPIO	LED	
J-TAG	J-TAG	NA	
Document			
ALL_recheck			
code-commit			
DBG_RS485			
DBG_CAN			

DBG_PCIE

DBG_audio

DBG_SPI-flash

3. Previous Version

This release is compatible for EVK-SM9300 REV02 (SOM-SM9300 REV02+BB-ESDSM00 REV02) only. EVK-SM9300 REV01, SOM-SM9300 REV01 are not supported.

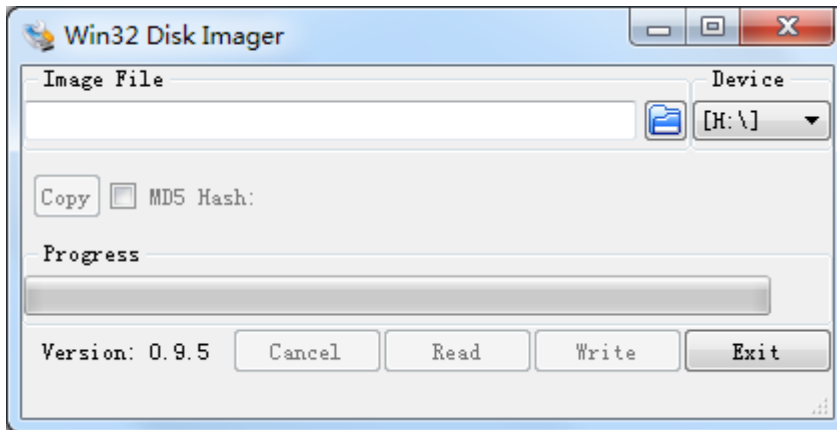
For the detail info, refer to following table.

HW version		SW Version	Summary
Core-board	Base-board		
Rev 00	Rev 00	20160527	R&D Develop version
Rev 01	Rev 01	Rev 01	1 st release version
		Rev 02	Release Fix version
		Rev 03	Mass product version
	Rev01/ Rev02	Rev V1.1	Maintain version
Rev 02	Rev 02	V1.2.1r01	New HW/SW version

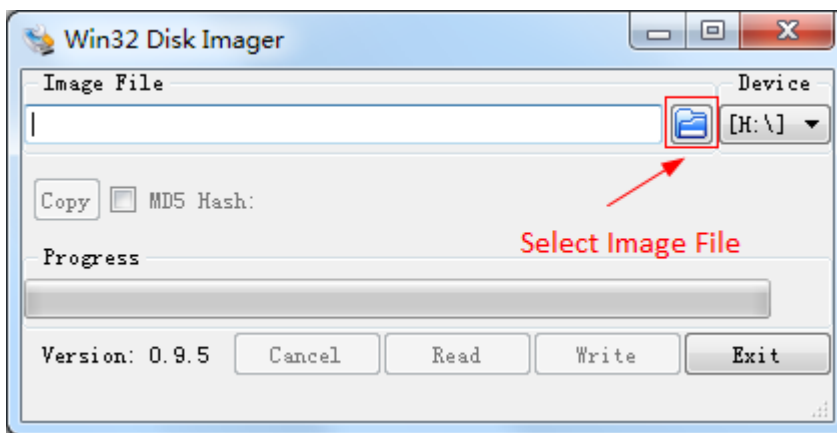
第1章 快速启动

1.1 烧写镜像到 SD 卡

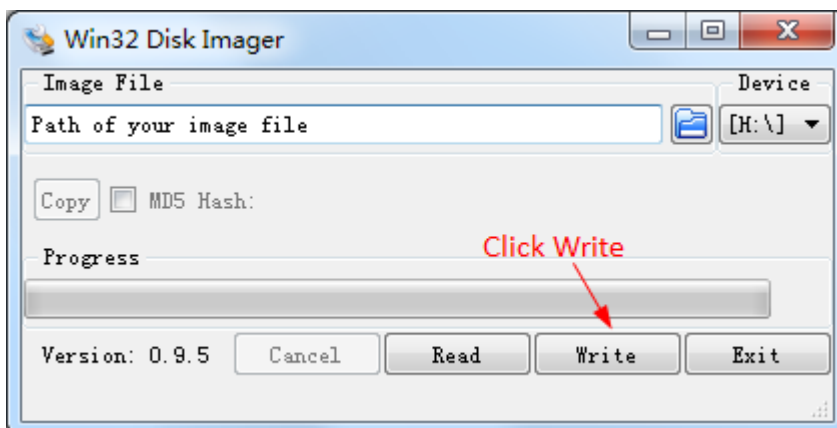
- 首先，你需要准备一张不小于 2G 的 SD 卡
- 然后，你需要从 <https://sourceforge.net/projects/win32diskimager/> 下载并安装 Win32 Disk Imager



- 选择需要烧写的镜像，EVK-SM9300-Image-SDcard-V1.2.1r01.img:



- 点击 Write 烧写镜像:



1.2 从 SD 卡启动系统

- 在 PC 上安装串口软件（例如 SecureCRT），选择正确的端口号，波特率 115200，8 位数据位，1 位停止位，无奇偶校验
- 用交叉串口线把板子上的 DEBUG 接口(J20)和 PC 的串口相连
- 把 SD 卡插入板上的插槽(J5)
- 参照下表设置拨码开关 J6:

BOOT CONFIG	
1	OFF
2	ON
3	OFF
4	ON

- 用 12V,2A 的电源，给板子供电(J1)，上电
- 系统启动完毕之后，串口显示如下

```
systemd[1]: Starting Journal Service...
systemd[1]: Started Journal Service.
systemd-journald[229]: Received request to flush runtime journal from PID 1
fec 2188000.ethernet eth0: Freescale FEC PHY driver [Generic PHY] (mii_bus:phy
_addr=2188000.ethernet:04, irq=-1)
IPv6: ADDRCONF(NETDEV_UP): eth0: link is not ready
flexcan 2090000.can can0: bit-timing not yet defined
flexcan 2094000.can can1: bit-timing not yet defined
IPv6: ADDRCONF(NETDEV_UP): usb0: link is not ready

Debian GNU/Linux 8 embest ttyMXC0

www.embest-tech.com

default username:password is [root:root]

embest login:
输入用户名和密码 root 登录:

Debian GNU/Linux 8 embest ttyMXC0

www.embest-tech.com

default username:password is [root:root]

embest login: root
Password:
Last login: Fri Aug 18 02:45:06 UTC 2017 on ttyMXC0
Linux embest 3.14.52 #1 SMP PREEMPT Mon Aug 14 15:53:58 CST 2017 armv7l

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
root@embest:~#
```

1.3 从 EMMC 启动系统

将 EVK-SM9300-Image-EMMC-V1.2.1r01.img 拷贝到 U 盘；

参考 1.2，先从 SD 卡启动系统，再将 U 盘插入 USB 接口（CON4）：

在串口终端输入：

```
root@embest:~# ls /dev/sd*
```

```
/dev/sda /dev/sda1
```

```
root@embest:~# mount /dev/sda1 /mnt/
```

```
root@embest:~# dd if=/mnt/EVK-SM9300-Image-EMMC-V1.2.1r01.img of=/dev/mmcblk0
```

注意：烧写时间较长，请耐心等待...

如果是第一次烧写 EMMC，需要在 eMMC 用户分区使能 boot_config。（参考 3.2.3 相关内容）

后续烧写 EMMC 只需要执行上述命令即可。

烧写结束后掉电，参照下表设置拨码开关 J6，拔掉 SD 卡，上电启动。

BOOT CONFIG	
1	ON
2	OFF
3	OFF
4	ON

1.4 启动 LOGO

当连接到 LVDS 屏时，系统启动过程中，LVDS 屏会显示 Embest 的 LOGO 图片



A Premier Farnell Company

如连接到 HDMI 屏，需要在 HDMI 屏上显示此图片，则需要在 U-boot 执行过程中按下 Enter 键

```
U-Boot 2015.04 (Aug 14 2017 - 15:22:42)

CPU:   Freescale i.MX6Q rev1.5 at 792 MHz
CPU:   Temperature 47 C
Reset cause: POR
Board: SM9300
I2C:   ready
DRAM:  1 GiB
MMC:   FSL_SDHC: 0, FSL_SDHC: 1
Display: HDMI (1024x768)
In:    serial
Out:   serial
Err:   serial
PMIC:  PFUZE100 ID=0x10
'mmcdev' not defined, try to getenv "mmcdevs"
reading logo.bmp
There is boot LOGO ...
Net:   FEC [PRIME]
Error: FEC address not set.
```

```
Normal Boot
Hit any key to stop autoboot:  0
U-Boot >
```

然后执行下列命令:

```
U-Boot > setenv panel HDMI
```

```
U-Boot > saveenv
```

```
Saving Environment to MMC...
```

```
Writing to MMC(0)... done
```

然后重启系统, Logo 图片就会在 HDMI 屏上显示了。

如果要改回从 LVDS 屏启动, 则改成:

```
U-Boot > setenv panel SVGA-lvds
```

```
U-Boot > saveenv
```

```
Saving Environment to MMC...
```

```
Writing to MMC(0)... done
```

第2章 功能测试

首先, 请参考[第一章 1.1](#), 把系统启动起来. 然后跟随下面的指引测试各项功能.

2.1 LED 测试

用户能够控制 SOM-SM9300 上的 LED (D5) 指示灯. 在终端中执行以下命令来进行测试:

熄灭 LED:

```
root@embest:~# echo 1 > /sys/class/leds/system_leds_d5/brightness
```

点亮 LED:

```
root@embest:~# echo 0 > /sys/class/leds/system_leds_d5/brightness
```

2.2 RTC 测试

安装电池到板子, 在串口终端输入:

查看当前时间:

```
root@embest:~# date
```

```
Thu Jan 1 00:03:27 UTC 1970
```

设置时间 2016 年 3 月 9 日 10 时 46 分:

```
root@embest:~# date 030910462016
```

```
Wed Mar 9 10:46:00 UTC 2016
```

把系统时钟写入 RTC:

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

读取 RTC:

```
root@embest:~# hwclock
```

```
Wed 09 Mar 2016 10:46:23 AM UTC -0.432561 seconds
```

可以看到, 硬件时钟 RTC 被设置成 2016 年 3 月 9 日, 系统时钟被保存到硬件时钟里。

重启系统并查看时间:

```
root@embest:~# date
```

```
Wed Mar 9 10:46:45 UTC 2016
```

2.3 EEPROM 测试

2.3.1 EEPROM on SOM-SM9300

在串口终端输入以下命令：

```
root@embest:~# ./eeprom_test_50
```

```
data will write to EEPROM at 0x400
```

```
00 01 02 03 04 05 06 07 08 09 0a 0b 0c 0d 0e 0f
10 11 12 13 14 15 16 17 18 19 1a 1b 1c 1d 1e 1f
20 21 22 23 24 25 26 27 28 29 2a 2b 2c 2d 2e 2f
30 31 32 33 34 35 36 37 38 39 3a 3b 3c 3d 3e 3f
40 41 42 43 44 45 46 47 48 49 4a 4b 4c 4d 4e 4f
50 51 52 53 54 55 56 57 58 59 5a 5b 5c 5d 5e 5f
60 61 62 63 64 65 66 67 68 69 6a 6b 6c 6d 6e 6f
70 71 72 73 74 75 76 77 78 79 7a 7b 7c 7d 7e 7f
80 81 82 83 84 85 86 87 88 89 8a 8b 8c 8d 8e 8f
90 91 92 93 94 95 96 97 98 99 9a 9b 9c 9d 9e 9f
a0 a1 a2 a3 a4 a5 a6 a7 a8 a9 aa ab ac ad ae af
b0 b1 b2 b3 b4 b5 b6 b7 b8 b9 ba bb bc bd be bf
c0 c1 c2 c3 c4 c5 c6 c7 c8 c9 ca cb cc cd ce cf
d0 d1 d2 d3 d4 d5 d6 d7 d8 d9 da db dc dd de df
e0 e1 e2 e3 e4 e5 e6 e7 e8 e9 ea eb ec ed ee ef
f0 f1 f2 f3 f4 f5 f6 f7 f8 f9 fa fb fc fd fe ff
```

```
data read from EEPROM at 0x400
```

```
00 01 02 03 04 05 06 07 08 09 0a 0b 0c 0d 0e 0f
10 11 12 13 14 15 16 17 18 19 1a 1b 1c 1d 1e 1f
20 21 22 23 24 25 26 27 28 29 2a 2b 2c 2d 2e 2f
30 31 32 33 34 35 36 37 38 39 3a 3b 3c 3d 3e 3f
40 41 42 43 44 45 46 47 48 49 4a 4b 4c 4d 4e 4f
50 51 52 53 54 55 56 57 58 59 5a 5b 5c 5d 5e 5f
60 61 62 63 64 65 66 67 68 69 6a 6b 6c 6d 6e 6f
70 71 72 73 74 75 76 77 78 79 7a 7b 7c 7d 7e 7f
80 81 82 83 84 85 86 87 88 89 8a 8b 8c 8d 8e 8f
90 91 92 93 94 95 96 97 98 99 9a 9b 9c 9d 9e 9f
a0 a1 a2 a3 a4 a5 a6 a7 a8 a9 aa ab ac ad ae af
b0 b1 b2 b3 b4 b5 b6 b7 b8 b9 ba bb bc bd be bf
```

```
c0 c1 c2 c3 c4 c5 c6 c7 c8 c9 ca cb cc cd ce cf
d0 d1 d2 d3 d4 d5 d6 d7 d8 d9 da db dc dd de df
e0 e1 e2 e3 e4 e5 e6 e7 e8 e9 ea eb ec ed ee ef
f0 f1 f2 f3 f4 f5 f6 f7 f8 f9 fa fb fc fd fe ff
```

2.3.2 EEPROM on BB-ESDSM00

```
root@embest:~# ./eeprom_test_57
```

```
data will write to EEPROM at 0x400
```

```
00 01 02 03 04 05 06 07 08 09 0a 0b 0c 0d 0e 0f
10 11 12 13 14 15 16 17 18 19 1a 1b 1c 1d 1e 1f
20 21 22 23 24 25 26 27 28 29 2a 2b 2c 2d 2e 2f
30 31 32 33 34 35 36 37 38 39 3a 3b 3c 3d 3e 3f
40 41 42 43 44 45 46 47 48 49 4a 4b 4c 4d 4e 4f
50 51 52 53 54 55 56 57 58 59 5a 5b 5c 5d 5e 5f
60 61 62 63 64 65 66 67 68 69 6a 6b 6c 6d 6e 6f
70 71 72 73 74 75 76 77 78 79 7a 7b 7c 7d 7e 7f
80 81 82 83 84 85 86 87 88 89 8a 8b 8c 8d 8e 8f
90 91 92 93 94 95 96 97 98 99 9a 9b 9c 9d 9e 9f
a0 a1 a2 a3 a4 a5 a6 a7 a8 a9 aa ab ac ad ae af
b0 b1 b2 b3 b4 b5 b6 b7 b8 b9 ba bb bc bd be bf
c0 c1 c2 c3 c4 c5 c6 c7 c8 c9 ca cb cc cd ce cf
d0 d1 d2 d3 d4 d5 d6 d7 d8 d9 da db dc dd de df
e0 e1 e2 e3 e4 e5 e6 e7 e8 e9 ea eb ec ed ee ef
f0 f1 f2 f3 f4 f5 f6 f7 f8 f9 fa fb fc fd fe ff
```

```
data read from EEPROM at 0x400
```

```
00 01 02 03 04 05 06 07 08 09 0a 0b 0c 0d 0e 0f
10 11 12 13 14 15 16 17 18 19 1a 1b 1c 1d 1e 1f
20 21 22 23 24 25 26 27 28 29 2a 2b 2c 2d 2e 2f
30 31 32 33 34 35 36 37 38 39 3a 3b 3c 3d 3e 3f
40 41 42 43 44 45 46 47 48 49 4a 4b 4c 4d 4e 4f
50 51 52 53 54 55 56 57 58 59 5a 5b 5c 5d 5e 5f
60 61 62 63 64 65 66 67 68 69 6a 6b 6c 6d 6e 6f
70 71 72 73 74 75 76 77 78 79 7a 7b 7c 7d 7e 7f
80 81 82 83 84 85 86 87 88 89 8a 8b 8c 8d 8e 8f
90 91 92 93 94 95 96 97 98 99 9a 9b 9c 9d 9e 9f
a0 a1 a2 a3 a4 a5 a6 a7 a8 a9 aa ab ac ad ae af
```

```
b0 b1 b2 b3 b4 b5 b6 b7 b8 b9 ba bb bc bd be bf
```

```
c0 c1 c2 c3 c4 c5 c6 c7 c8 c9 ca cb cc cd ce cf
```

```
d0 d1 d2 d3 d4 d5 d6 d7 d8 d9 da db dc dd de df
```

```
e0 e1 e2 e3 e4 e5 e6 e7 e8 e9 ea eb ec ed ee ef
```

```
f0 f1 f2 f3 f4 f5 f6 f7 f8 f9 fa fb fc fd fe ff
```

写数据与读到的数据相同，测试通过；

2.4 EMMC 测试

在串口终端执行：

```
root@embest:~# touch emmc_read emmc_write
```

编辑 emmc_write:

```
root@embest:~# vi emmc_write
```

例如写入 “emmc write test”

写 emmc 命令：

```
root@embest:~# dd if=emmc_write of=/dev/mmcblk0
```

```
0+1 records in
```

```
0+1 records out
```

```
17 bytes (17 B) mmcblk0: p1 p2
```

```
copied, 0.0305303 s, 0.6 kB/s
```

读 emmc 命令：

```
root@embest:~# dd if=/dev/mmcblk0 of=emmc_read bs=1K count=10
```

```
10+0 records in
```

```
10+0 records out
```

```
10240 bytes (10 kB) copied, 0.014218 s, 720 kB/s
```

查看 emmc_read:

```
root@embest:~# cat emmc_read
```

```
emmc write test
```

测试成功；

2.5 音频测试

板上带音频输入、输出接口，支持录放音，用户可使用如下命令进行测试：

录音测试，生成音频文件 k:

```
root@embest:~# arecord -t wav -c 1 -r 44100 -f S16_LE -v k
```

放音测试，播放音频文件 k:

```
root@embest:~# aplay -t wav -c 2 -r 44100 -f S16_LE -v k
```

本次发布提供了 22_3, 32_3, 44_3 和 48_3 共 4 个音频文件，分别对应 22K, 32K, 44.1K 和 48K 几种采样频

率，可用下列命令测试：

```
root@embest:~# aplay audiofile/22_3
```

2.6 LVDS 测试

将 LVDS 屏幕接到底板的 J10 和 J13 脚上，系统启动后屏幕有输出，则表明 LVDS 功能正常。

注意:仅支持如下几种屏幕:

- 6bit 和 8bit 的 JEIDA 模式的屏幕
- 6bit 的 VESA 模式的屏幕（需要通过底板的选择引脚来配置）

2.7 背光测试

LVDS 的背光的亮度设置范围为（1—7），1 表示亮度最低，7 表示亮度最高，在串口终端下输入如下命令进行背光测试：

最暗：

```
root@embest:~# echo 1 > /sys/class/backlight/backlight.24/brightness
```

最亮：

```
root@embest:~# echo 7 > /sys/class/backlight/backlight.24/brightness
```

2.8 触摸屏测试

连接触摸屏模块到 J36

在串口终端输入以下命令：

```
root@embest:~# evtest /dev/input/event0
```

点击屏幕，串口终端出现打印信息：

```
Testing ... (interrupt to exit)
```

```
Event: time 1464572876.234689, type 3 (EV_ABS), code 0 (ABS_X), value 2149
```

```
Event: time 1464572876.234689, type 3 (EV_ABS), code 1 (ABS_Y), value 1949
```

```
Event: time 1464572876.234689, type 3 (EV_ABS), code 24 (ABS_PRESSURE), value 246
```

```
Event: time 1464572876.234689, ----- EV_SYN -----
```

```
Event: time 1464572876.254690, type 3 (EV_ABS), code 0 (ABS_X), value 1787
```

```
Event: time 1464572876.254690, type 3 (EV_ABS), code 1 (ABS_Y), value 2453
```

```
Event: time 1464572876.254690, type 3 (EV_ABS), code 24 (ABS_PRESSURE), value 248
```

注意：Ctrl+C 中断测试

2.9 VGA 测试

将 VGA 屏幕接到底板的 VGA 接口 CN2 上，系统启动后屏幕有输出，则表明 VGA 功能正常。

2.10 HDMI 测试

2.10.1 HDMI 启动

打开 uEnv.txt 文件，修改 fdtfile=embest_fsl_som_sm9300-HDMI.dtb

将 HDMI 屏幕接到底板 HDMI 接口 J9 上，系统启动后屏幕有输出，则表明 HDMI 功能正常。

2.10.2 HDMI 显示图像

不使用 HDMI.dtb 启动时，可用下列方式显示图像到 HDMI 设备

1. 显示摄像头图像

连接摄像头 EXP-C5DX00 到底板 J3 端口，执行下列程序，系统会将摄像头摄录的内容实时显示到 HDMI 设备：

```
root@embest:~# ./v4l2captrue_video0_fb2
```

2. 显示静态图片

执行下列命令，显示/root 目录下预存的 jpg 图片：

```
root@embest:~# ./hdm_i_show_jpg
```

2.10.3 HDMI 音频播放

切换为 embest_fsl_som_sm9300-HDMI_audio.dtb 并启动后，连接 HDMI 显示器及配套放音设备，播放音频文件

```
root@embest:~# aplay audiofile/22_3
```

2.11 串口测试

核心引出 3 个串口和底板连接，核心板上串口编号跟 CPU 串口编号一致。核心板串口编号和底板串口编号对应关系如下表：

核心板(CPU)	底板	功能
UART1	UART1	串口 Debug
UART4	UART3	RS232
UART2	UART0	RS485

UART1 为 Debug 功能，UART2 测试见 485 部分

2.11.1 UART4 测试

短接底板 J24 第 5，7 号 pin，做自发自收测试：

```
root@embest:~# ./uart_test -d /dev/ttyxc3 -b 115200
```

```
/dev/ttyxc3 SEND: 1234567890
```

```
/dev/ttyxc3 RECV 10 total
```

```
/dev/ttyxc3 RECV: 1234567890
```

以上输出表示测试成功

注意： Ctrl+C 中断测试

2.12 RS485 测试

使用导线将 A、B 两块板子的 J27 对接，接线次序如下表：

A 板 J27 PIN	B 板 J27 PIN
PIN-1	PIN-1
PIN-2	PIN-2
PIN-3	PIN-3

在 A 板上执行

```
root@embest:~# ./uart_test2 /dev/ttymx1 9600 0 100
```

在 B 板上执行

```
root@embest:~# ./uart_test2 /dev/ttymx1 9600 1
```

如果测试成功，在 B 板上会接收到如下信息。

```
****data length = 31 ****
```

```
41 54 31 32 33 34 35 36 37 38 39 58 59 5a 61 62 63 64 65 64 66 68 69 6a 6b 6c 6d 6e 0d 00 fb
```

```
***** receive data 135667 pkts...0 bytes.....
```

交换在 A,B 板上运行的指令，测试现象应该一样

2.13 网络测试

连接网线到 CON4，在串口终端中输入以下命令来设置 IP 地址：

```
root@embest:~# ifconfig eth0 192.168.52.64
```

网络测试：

```
root@embest:~# ping 192.168.52.1
```

2.14 PCIE 网卡测试

将 PCIE 网卡插入卡槽 PCIE_A X1(CON1)，然后启动系统，执行下列命令测试

```
root@embest:~# lspci
```

```
00:00.0 PCI bridge: Synopsys, Inc. Device abcd (rev 01)
```

```
01:00.0 Ethernet controller: Realtek Semiconductor Co., Ltd. RTL8111/8168/8411 PCI Express Gigabit Ethernet  
Controller (rev 09)
```

连接网线到 PCIE 网卡，用下列命令查看是否有 eth1 网卡

```
root@embest:~# ifconfig
```

Note:如果 CON4 连接网线，将其断开

设置 IP 并进行网络测试：

```
root@embest:~# ifconfig eth1 192.168.52.66
```

```
root@embest:~# ping 192.168.52.1
```

2.15 CAN 测试

将 CAN0 和 CAN1 连接，即将 J25 与 J26 对接。

注意：要使用 CAN 的功能，需要把 R285,R288 焊接 100R 或 120R 的电阻。

测试步骤：

1. 打开 CAN0 CAN1

```
root@embest:~# ip link set can0 type can bitrate 50000 triple-sampling on
```

```
root@embest:~# ip link set can1 type can bitrate 50000 triple-sampling on
```

```
root@embest:~# ip link set can0 up
```

```
flexcan 2090000.can can0: writing ctrl=0x27292085
```

```
root@embest:~# ip link set can1 up
```

```
flexcan 2094000.can can1: writing ctrl=0x27292085
```

2. 发送与接收：

CAN1 接收，CAN0 往 CAN1 发数据，显示如下表示测试收发成功。

```
root@embest:~# candump can1&
```

```
[4] 589
```

```
root@embest:~# cansend can0 123#01020304050607
```

```
can1 123 [7] 01 02 03 04 05 06 07
```

用查看命令可以看到 CAN0 Tx 增加了 1 个 packet，7 个 bytes。CAN1 RX 增加了 1 个 packet，7 个 bytes

```
root@embest:~# ip -d -s link show can0
```

```
2: can0: <NOARP,UP,LOWER_UP,ECHO> mtu 576 qdisc pfifo_fast state UNKNOWN mode DEFAULT group default qlen 10
```

```
link/can promiscuity 0
```

```
can <TRIPLE-SAMPLING> state ERROR-ACTIVE (berr-counter tx 0 rx 0) restart-ms 0
```

```
bitrate 50000 sample-point 0.866
```

```
tq 1333 prop-seg 6 phase-seg1 6 phase-seg2 2 sjw 1
```

```
flexcan: tseg1 4..16 tseg2 2..8 sjw 1..4 brp 1..256 brp-inc 1
```

```
clock 30000000
```

```
re-started bus-errors arbit-lost error-warn error-pass bus-off
```

```
0 0 0 0 0 0
```

```
RX: bytes packets errors dropped overrun mcast
```

```
0 0 0 0 0 0
```

```
TX: bytes packets errors dropped carrier collsns
```

```
7 1 0 0 0 0
```

注意：

两个 can 端口必须设置成相同的波特率。

3. 关闭 can0 can1

```
root@embest:~# ip link set can0 down
```

```
root@embest:~# ip link set can1 down
```

2.16 USB 测试

2.16.1 Host 测试

底板的 CON4 有两个 USB host 接口, 其中靠近 PCB 板的一个与核心板的 USB host 连接, 将 U 盘插入这个 USB Host 口, 串口显示磁盘信息:

```
usb 1-1: new high-speed USB device number 2 using ci_hdrc
usb-storage 1-1:1.0: USB Mass Storage device detected
scsi1 : usb-storage 1-1:1.0
scsi 1:0:0:0: Direct-Access    Generic  Flash Disk    8.07 PQ: 0 ANSI: 4
sd 1:0:0:0: [sda] 7823360 512-byte logical blocks: (4.00 GB/3.73 GiB)
sd 1:0:0:0: [sda] Write Protect is off
sd 1:0:0:0: [sda] Write cache: disabled, read cache: enabled, doesn't support DPO or FUA
sda: sda1
sd 1:0:0:0: [sda] Attached SCSI removable disk
```

串口终端输入如下命令:

```
root@embest:~# ls /dev/sd*
/dev/sda  /dev/sda1
/dev 下存在设备节点;
```

2.16.2 OTG 测试

1 主设备

通过转接线连接 U 盘到底板的 Mini USB 口 J14:

```
root@embest:~# ci_hdrc ci_hdrc.0: timeout waiting for 00000800 in 12
ci_hdrc ci_hdrc.0: EHCI Host Controller
ci_hdrc ci_hdrc.0: new USB bus registered, assigned bus number 2
ci_hdrc ci_hdrc.0: USB 2.0 started, EHCI 1.00
hub 2-0:1.0: USB hub found
hub 2-0:1.0: 1 port detected
usb 2-1: new high-speed USB device number 2 using ci_hdrc
usb-storage 2-1:1.0: USB Mass Storage device detected
scsi5 : usb-storage 2-1:1.0
scsi 5:0:0:0: Direct-Access    Generic  Flash Disk    8.07 PQ: 0 ANSI: 4
sd 5:0:0:0: [sda] 7823360 512-byte logical blocks: (4.00 GB/3.73 GiB)
sd 5:0:0:0: [sda] Write Protect is off
sd 5:0:0:0: [sda] Write cache: disabled, read cache: enabled, doesn't support DPO or FUA
sda: sda1
```

```
sd 5:0:0:0: [sda] Attached SCSI removable disk
```

串口终端输入如下命令:

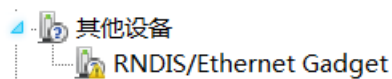
```
root@embest:~# ls /dev/sd*
```

```
/dev/sda /dev/sda1
```

/dev 下存在设备节点;

2. 从设备

连接 J14 到 PC 端, 打开设备管理器, 识别到如下设备:



2.17 SATA 测试

启动系统前, 将 J11 连接到 SATA 接口, 给硬盘供电。

启动系统后, 运行下面的命令查看 500G 硬盘连接。

```
root@embest:~# fdisk -l
```

```
Disk /dev/sda: 465.8 GiB, 500107862016 bytes, 976773168 sectors
```

```
Units: sectors of 1 * 512 = 512 bytes
```

```
Sector size (logical/physical): 512 bytes / 512 bytes
```

```
I/O size (minimum/optimal): 512 bytes / 512 bytes
```

```
Disklabel type: dos
```

```
Disk identifier: 0xd274d274
```

Device	Boot	Start	End	Sectors	Size	Id	Type
/dev/sda1	*	63	83891429	83891367	40G	c	W95 FAT32 (LBA)
/dev/sda2		83891430	976768064	892876635	425.8G	f	W95 Ext'd (LBA)
/dev/sda5		83891493	209744639	125853147	60G	7	HPFS/NTFS/exFAT
/dev/sda6		209744703	402701354	192956652	92G	7	HPFS/NTFS/exFAT
/dev/sda7		402701418	595658069	192956652	92G	7	HPFS/NTFS/exFAT
/dev/sda8		595658133	786526334	190868202	91G	7	HPFS/NTFS/exFAT
/dev/sda9		786526398	881882176	95355779	45.5G	7	HPFS/NTFS/exFAT
/dev/sda10		881883136	976766975	94883840	45.3G	7	HPFS/NTFS/exFAT

2.18 Camera 测试

连接摄像头模块到 J3, 将 LVDS 屏幕接到底板的 J10 和 J13 脚上, 然后参照下列文档执行测试:

EXP-C5DX00 freescale 用户手册: EXP-C5DX00-Freescale-UM-V1.0-CN.docx

2.19 WIFI 测试

WIFI 和 Bluetooth 模块仅在从 EMMC 启动时可用。

注意:

1. SD 卡需拔出
2. EMMC 的 uEnv.txt 中, fdt_file=embest_fsl_som_sm9300_wifi_bt.dtb (发布镜像中已处理)
3. 需要用 CON4 联网并下载安装 udhcpd, udhcpc, expect 程序

```
apt-get install udhcpd
```

```
apt-get install expect
```

```
apt-get install udhcpc
```

2.19.1 配置 WIFI

开机进入系统后, 在串口终端输入:

```
root@embest:~# cd wl_code
```

```
root@embest:~/wl_code# ./wl18xx_init.sh
```

```
0x00490B00
```

```
0x00490D00
```

第一次启动系统时, 执行 ./wl18xx_init.sh 后需要重启, 后续可直接继续下一步操作。

2.19.2 连接 WIFI

在串口终端输入:

```
root@embest:~# cd wl_code
```

```
root@embest:~/wl_code# ./sta_start.sh
```

```
root@embest:~/wl_code# ./sta_connect-ex.sh embest WPA-PSK 12345678
```

其中 embest 是 SSID, 12345678 是 Wi-Fi 密码, 加密方式为 WPA-PSK, 当系统打印如下信息时, 表明连接成功:

```
wlan0: authenticated
```

```
wlan0: Trying to associate with bc:d1:77:09:56:ea (SSID='embest' freq=2412 MHz)
```

```
wlan0: associate with bc:d1:77:09:56:ea (try 1/3)
```

```
wlan0: RX AssocResp from bc:d1:77:09:56:ea (capab=0x431 status=0 aid=6)
```

```
wlan0: associated
```

```
IPv6: ADDRCONF(NETDEV_CHANGE): wlan0: link becomes ready
```

```
wlan0: Associated with bc:d1:77:09:56:ea
```

```
wlan0: Limiting TX power to 20 (20 - 0) dBm as advertised by bc:d1:77:09:56:ea
```

```
wlan0: WPA: Key negotiation completed with bc:d1:77:09:56:ea [PTKw/core: Association completed.
```

```
=CCMP GTK=CCMP]
```

```
wlan0: CTRL-EVENT-CONNECTED - Connection to bc:d1:77:09:56:ea completed [id=0 id_str=]
```

```
Sending discover...
```

```
Sending select for 192.168.2.82...
```

```
Lease of 192.168.2.82 obtained, lease time 691200
```

用 ping 命令测试 wifi 连接

```
root@embest:~/wl_code# ping www.baidu.com
```

```
PING www.a.shifen.com (103.235.46.39) 56(84) bytes of data.
```

```
64 bytes from 103.235.46.39: icmp_seq=1 ttl=50 time=122 ms
```

执行下列命令关闭 WIFI

```
root@embest:~/wl_code# ./sta_stop.sh
```

2.19.3 WIFI Host AP 功能

连接网线到 CON4，然后执行 ap_start.sh 命令打开 WIFI Host AP 功能，ap_stop.sh 关闭 WIFI Host AP 功能

```
root@embest:~/wl_code# ./ap_start.sh
```

```
Configuration file: hostapd.conf
```

```
HT (IEEE 802.11n) with WPA/WPA2 requires CCMP/GCMP to be enabled, disabling HT capabilities
```

```
WPS: WPA/TKIP configuration without WPA2/CCMP forced WPS to be disabled
```

```
rftkill: Cannot open RFTKILL control device
```

```
wlcore: down
```

```
wlan0: interface state UNINITIALIZED->COUNTRY_UPDATE
```

```
Using interface wlan0 with hwaddr 54:4a:16:18:4b:49 and ssid "embest-wifi"
```

```
wlan0: interface state COUNTRY_UPDATE->ENABLED
```

```
wlan0: AP-ENABLED
```

注意：

WIFI 热点的默认 SSID 为 embest-wifi，密码为 12345678。如需修改默认 SSID 和密码，可修改 /root/wl_code/hostapd.conf 的对应参数，打开 WIFI Host AP 之后，即可用手机或其他无线设备连接这个热点。

2.20 Bluetooth 测试

2.20.1 初始化蓝牙模块

```
root@embest:~/wl_code# ./bt_start.sh
```

如果初始化成功，串口将打印如下信息：

```
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
```

```
spawn bluetoothctl
```

2.20.2 测试蓝牙功能

扫描蓝牙设备:

```
root@embest:~/wl_code# hcitool scan
```

```
Scanning ...
```

```
28:A0:2B:2A:32:7A      iPhone
```

可执行下列命令连接手机并匹配:

```
root@embest:~/wl_code# rfcomm connect 0 28:A0:2B:2A:32:7A
```

Note: 28:A0:2B:2A:32:7A 需改成扫描到的蓝牙设备的地址，目前只支持连接 iPhone。

第3章 系统编译

3.1 配置编译环境

将发布文件夹的 02Linux 的全部内容拷贝到 Linux 环境下的 \$HOME 目录下(解压 rar 文件), 编译工具 fsl-linaro-toolchain-master.tar.gz 在 \$HOME/04LinuxTools 目录下, 用如下命令解压:

```
$tar -xvzf fsl-linaro-toolchain-master.tar.gz
```

导入环境变量:

```
$export CROSS_COMPILE=$HOME/04LinuxTools/fsl-linaro-toolchain-master/bin/arm-fsl-linux-gnueabi-
```

```
$export ARCH=arm
```

3.2 编译 UBOOT

3.2.1 获取 uboot 源码

Uboot 源码在 \$HOME/01LinuxSourceCode 目录下, 解压 u-boot*.tar.gz :

```
$ cd $HOME/01LinuxSourceCode
```

```
$ tar -xvzf u-boot*.tar.gz
```

3.2.2 编译并烧写镜像到 SD 卡

```
$ cd $HOME/01LinuxSourceCode/u-boot
```

```
$ make distclean
```

```
$make embest_fsl_sm9300_sdc_card_defconfig
```

```
$make
```

编译完成后在 \$HOME/01LinuxSourceCode/u-boot 目录下生成 u-boot.imx, 将 u-boot.imx 导入到 SD 卡中: (sdx 用 Linux 系统下 SD 卡的实际编号代替)

```
$ sudo dd if=u-boot.imx of=/dev/sdx bs=512 seek=2 conv=fsync
```

3.2.3 编译并烧写镜像到 EMMC

```
$ cd $HOME/01LinuxSourceCode/u-boot
```

```
$ make distclean
```

```
$make embest_fsl_sm9300_emmc1_defconfig
```

```
$make
```

编译完成后在 \$HOME/01LinuxSourceCode/u-boot 目录下生成 u-boot.imx, 将 u-boot.imx 导入到 SD 卡中:

参考 1.2, 先从 SD 卡启动系统

启动系统后, 在串口终端中执行下面的命令操作 EMMC 实现烧写过程:

1. 对 EMMC 分区

```
$ sudo dd if=/dev/zero of=/dev/mmcblk0 bs=1K count=1
```

```
$ echo -e " o\n\n\np\n1\n20480\n+64M\na\n\t\nc\n\n\np\n2\n151552\n\nw\n\n " | fdisk /dev/mmcblk0
```

```
$ sudo mkfs.vfat /dev/mmcblk0p1
```

```
$ sudo mkfs.ext4 /dev/mmcblk0p2
```

```
$ sudo fdisk /dev/mmcblk0 -l
```

2. 将刚刚拷贝的 u-boot.imx(for emmc)烧写到 emmc 中

```
$ sudo dd if=u-boot.imx of=/dev/mmcblk0 bs=512 seek=2 conv=fsync
```

3. 在 eMMC 用户分区使能 boot_config:

```
$ sudo echo 56 > /sys/block/mmcblk0/device/boot_config
```

4. 检查是否使能成功:

```
$ sudo cat /sys/block/mmcblk0/device/boot_info
```

若使能成功, 会打印以下信息:

```
boot_partition:0x78;
```

```
BOOT_ACK:1 - Boot acknowledge sent during boot operation
```

```
BOOT_PARTITION-ENABLE: 7 - User area enabled for boot
```

这样 U-boot 就被烧写到 EMMC 中了。

3.3 Kernel

3.3.1 获取内核源码

内核源码存在\$HOME/01LinuxSourceCode 目录下,解压 linux*.tar.gz

```
$ tar -zxvf linux*.tar.gz
```

3.3.2 编译并烧写镜像到 SD 卡

```
$ cd $HOME/01LinuxSourceCode/linux
```

```
$ make distclean
```

```
$ make embest_fsl_sm9300_defconfig
```

```
$ make
```

编译完成后在

- \$HOME/01LinuxSourceCode/linux/arch/arm/boot 目录下生成 zImage
- \$HOME/01LinuxSourceCode/linux/arch/arm/boot/dts 目录下生成
 - embest_fsl_som_sm9300.dtb
 - embest_fsl_som_sm9300-HDMI.dtb
 - embest_fsl_som_sm9300_wifi_bt.dtb

将文件拷贝到 SD 卡中。

3.3.3 编译并烧写镜像到 EMMC 卡

编译过程同上

将 zImage 和 dtb 文件拷贝到 U 盘, 连接 U 盘到 EVK-SM9300

先从 EMMC 启动, 挂载 U 盘到文件系统, 再用 cp 命令拷贝 U 盘文件到/boot/firmware 目录中。

3.4 File system

3.4.1 烧写 File system 到 SD 卡

用 dd 命令烧写 file system 的 img 文件到 SD 卡的第二分区, 如:

```
dd if=debian-8.x-armhf-2016-09-29.img of=/dev/sda2
```

3.4.2 烧写 File system 到 EMMC

从 SD 卡启动系统后, 将装有 file system 的 img 文件的 U 盘连接到开发板, 然后用 dd 命令烧写 file system 的 img 文件到 EMMC 的第二分区, 如:

```
root@embest:~# dd if=/mnt/debian-8.x-armhf-2016-09-29.img of=/dev/mmcblk0p2
```

单击下面可查看定价，库存，交付和生命周期等信息

[>>Avnet manufacturing service\(英蓓特\)](#)