用户手册

[SBC-PH8800]

历史版本

Rev.	Note	Author
20160902	Initial	Sandy

目录

Release Note

1. 镜像版本

SBC-PH8800_Shipment_Image_SDCard_REV01.img

SBC-PH8800_Shipment_Image_EMMC_Rev01.img

2. 功能列表

Feature List	SBC-PH8800			
	Schematic Page#	On-Chip Peripherals	On-Board Peripherals	Detail Functions(existing)
u-boot version	2015.09			Supports kernel boot
kernel version	4.1.6			Supports all below functionality
Filesystem	Debian			Default root file system used by debian
CPU	PH8800-U11	AM437X_ZDN		Null
DDRAM	PH8800-p7-u12/u7	DDR	MT41K256M16HA-125	Can access read write and run code
PMIC	PH8800-p3-u13	I2C0	TPS65218	Null
MicroSD_(TF)	SPH1800-P6-TF1	MMC0	Null	Can access read write and boot
External-RTC	SPH1800-P9-U55	I2C0	RX-8025TUB	can read write and keep time off power
Integrited-RTC	PH8800-u11	RTC	Null	can read write and keep time off power
LEDs	PH8800-p10-D3/D4	gpio	Null	System can control LED to light or not
Power-Button	PH1800-P14-S2	I2C0	TPS65218	Can get key value
LCD	SPH1800-P9-J9	RGB	Null	Can show picture on the screen
Backlight	SPH1800-P9-J9	PWM	Null	System can control the LCD backlight
TouchScreen	SPH1800-P9-J9	ADC-TSC	Null	System use touchscreen
eMMC	PH8800-p8-u14	MMC1	MTFC4GACAAAM-4M IT	Can access read write
EEPROM	PH8800-p8-u6	I2C0	CAT24C256W	Can access read write
SPI-FLASH	PH8800-p8-u3	QSPI	N25Q256A13EF840	1. Boot from SPI-Flash

CAN-1	SPH1800-p8-J61	CAN1	MC33901WEF	System can send and receive data between two board
CAN-2	SPH1800-p8-J61	CAN0	MC33901WEF	System can send and receive data between two board
UART-0	SPH1800-p7-CN4	UART0	NUll	System can send and receive data in loopback mode
UART-1	SPH1800-p7-J4	UART5	MAX3232CUE+	System can send and receive data in loopback mode
UART-2	SPH1800-p13-J58	UART3	Null	System can send and receive data in loopback mode
UART-4	SPH1800-p13-J58	UART1	MAX3232CUE+	System can send and receive data in loopback mode
RS485-2	SPH1800-p8-u5	SPI0	SC16IS752IPW	System can send and receive data between two board
RS485-3	SPH1800-p8-u5	SPI0	SC16IS752IPW	System can send and receive data between two board
USB-Host	SPH1800-p11-p3	USB1	Null	Can recognize U disk by USB host
CAMERA	SPH1800-p9j8	CSI&I2C1	Null	Could preview, take picture and record video
USB-OTG	SPH1800-p11-j13	USB0	Null	Can recognize U disk in host mode, and can work as usb ethernet in device mode
Ethernet-1	PH8800-P9-U9	RGMII1	KSZ9031RNXIA	Can ping the server
Ethernet-2	SPH1800-P12-J17	RGMII2	AR8035	Can ping the server
HDMI	SPH1800-P10-U34	I2C0	TDA19988BHN/C1,551	Can show picture on the screen
Audio	SPH1800-P10-U34	I2C0	TDA19988BHN/C1,551	can play wav

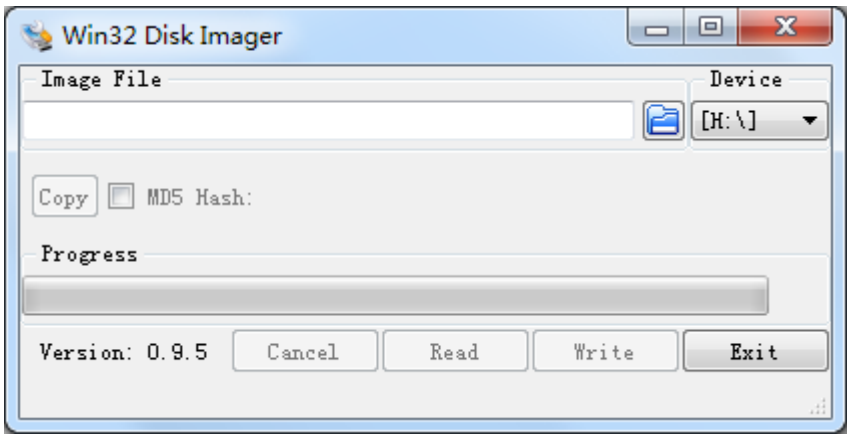
3. 已知问题

Known issue List	Detail
SPI-FLASH	Not Support: SPI-Flash access in kernel
Ethernet-1 & Ethernet -2	Bug: Board to board connect under high or low temperature environment could not working normally
LCD	Bug:4.3 inch Screen turn white for a while in boot
HDMI Audio	Not support Sony HDMI displayer

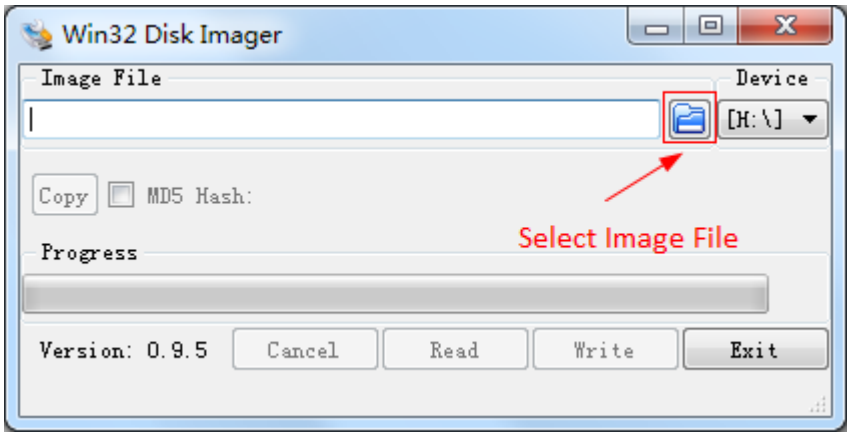
第1章 快速启动

1.1 烧写镜像到 SD 卡

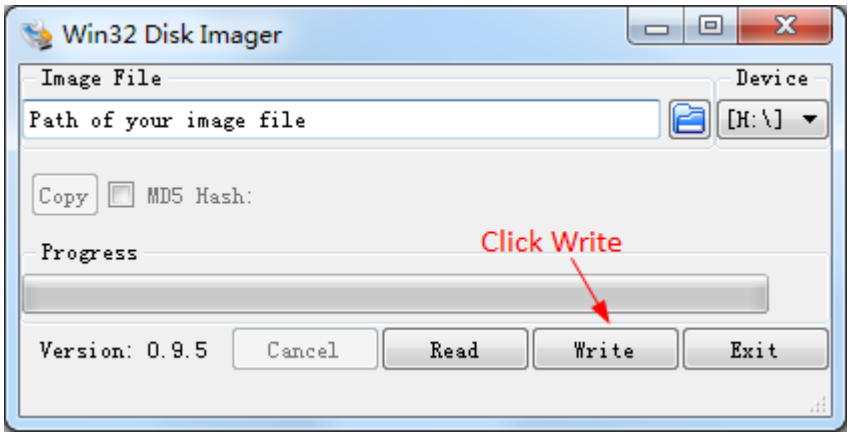
- 首先，你需要准备一张不小于 2G 的 SD 卡
- 然后，你需要从 https://sourceforge.net/projects/win32diskimager/ 下载并安装 Win32 Disk Imager



- 选择需要烧写的镜像，SBC-PH8800_Shipment_Image_SDCard_Rev01.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 USB 转 TTL 模块把板子上的 DEBUG 接口(CN4)和 PC 相连
- 把 Micro SD 卡插入板上的插槽 TF1
- 按下按键 S3，用 5V,2A 的电源，给板子供电(J1)，上电复位后松开 S3
- 系统启动完毕之后，串口显示如下

```
[    7.409917] systemd[1]: Starting Journal Service...
[    7.426561] systemd[1]: Started Journal Service.
[    7.599897] systemd-udevd[163]: starting version 215
[    8.102171] systemd-journald[162]: Received request to flush runtime journal from PID 1
[    8.201122]  remoteproc0: failed to load am335x-pm-firmware.elf
[    8.237170]  remoteproc0: powering up wkup_m3
[    8.262756]  remoteproc0: Direct firmware load for am335x-pm-firmware.elf failed with error -2
[    8.344518]  remoteproc0: Falling back to user helper
[    9.573464]  remoteproc0: request_firmware failed: -11
[    9.580114]  remoteproc0: rproc_boot failed
[   10.134627] net eth0: initializing cpsw version 1.15 (0)
[   10.222955] net eth0: phy found : id is : 0x221622
[   10.754600] net eth1: initializing cpsw version 1.15 (0)
[   10.842988] net eth1: phy found : id is : 0x4dd072
[   11.409176] net can0: c_can_hw_raminit_wait_syscon: time out
[   11.491746] c_can_platform 481cc000.can can0: bit-timing not yet defined
[   11.553953] c_can_platform 481cc000.can can0: failed to open can device
[   11.616721] net can1: c_can_hw_raminit_wait_syscon: time out
[   11.710230] c_can_platform 481d0000.can can1: bit-timing not yet defined
[   11.745757] c_can_platform 481d0000.can can1: failed to open can device
[   12.276336] FAT-fs (mmcblk0p1): Volume was not properly unmounted. Some data may be corrupt. P
lease run fsck.

Debian GNU/Linux 8 embest ttyS0

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 ttyS0

www.embest-tech.com

default username:password is [root:root]

embest login: root
Password:
Linux embest 4.1.6 #1 PREEMPT Tue Sep 27 12:00:43 CST 2016 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 SPI Flash 启动

参考 1.2，先从 SD 卡启动，终端中打印如下信息时，按“回车键”进入 uboot:

U-Boot SPL 2015.07 (Sep 27 2016 - 11:42:48)

SPL: Please implement spl_start_uboot() for your board

SPL: Direct Linux boot not active!

reading u-boot.img

reading u-boot.img

U-Boot 2015.07 (Sep 27 2016 - 11:42:48 +0800)

I2C: ready

DRAM: 1 GiB

PMIC: TPS65218

MMC: OMAP SD/MMC: 0, OMAP SD/MMC: 1

reading uboot.env

** Unable to read "uboot.env" from mmc0:1 **

Using default environment

Net: <ethaddr> not set. Validating first E-fuse MAC

cpsw, usb_ether

Hit any key to stop autoboot: 0

U-Boot#

（按下 Enter 键）

在终端中执行以下命令:

U-Boot# run update_qspi_flash

switch to partitions #0, OK

mmc0 is current device

SD/MMC found on device

reading u-boot-spl.bin

56904 bytes read in 6 ms (9 MiB/s)

SF: Detected N25Q256 with page size 256 Bytes, erase size 4 KiB, total 32 MiB, mapped at 30000000

SF: 589824 bytes @ 0x0 Erased: OK

device 0 offset 0x0, size 0xde48

SF: 56904 bytes @ 0x0 Written: OK

reading u-boot.bin

288632 bytes read in 17 ms (16.2 MiB/s)

device 0 offset 0x20000, size 0x46778

SF: 288632 bytes @ 0x20000 Written: OK

U-Boot#

输入下列命令从 SD 卡启动系统：

U-Boot# boot

将 SBC-PH8800_Shipment_Image_EMMC_Rev01.img 拷贝到 U 盘，将 U 盘插入 USB 接口（P3）：

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

/dev/sda /dev/sda1

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

root@embest:~# dd if=/mnt/SBC-PH8800_Shipment_Image_EMMC_Rev01.img of=/dev/mmcblk1

注意：烧写时间较长，请耐心等待…

烧写结束后，上电复位并启动系统（不用按 S3）

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-PH8800 上的 LED（D3,D4）指示灯。在终端中执行以下命令来进行测试；（其中 D3 对应 user_leds_d3，D4 对应 user_leds_d4）

熄灭 LED:

```
root@embest:~# echo 0 > /sys/class/leds/user_leds_d3/brightness
root@embest:~# echo 0 > /sys/class/leds/user_leds_d4/brightness
```

点亮 LED:

```
root@embest:~# echo 1 > /sys/class/leds/user_leds_d3/brightness
root@embest:~# echo 1 > /sys/class/leds/user_leds_d4/brightness
```

2.2 RTC 测试

在串口终端输入：

查看当前时间：

```
root@embest:~# date
Sat Jan    1 00:02:07 UTC 2000
```

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

在串口终端输入以下命令：

root@embest:~# ./eeprom_test

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/mmcblk1

[929.393325] mmcblk1: p1 p2

0+1 records in

0+1 records out

17 bytes (17 B) copied, 0.135215 s, 0.1 kB/s

读 emmc 命令：

root@embest:~# dd if=/dev/mmcblk1 of=emmc_read bs=1K count=10

10+0 records in

10+0 records out

10240 bytes (10 kB) copied, 0.00446492 s, 2.3 MB/s

查看 emmc_read:

root@embest:~# cat emmc_read

emmc write test

测试成功；

2.5 ADC 测试

在串口终端输入以下命令，采样值返回：

root@embest:~# cat /sys/bus/platform/devices/TI-am335x-adc/iio\:device0/in_voltage4_raw

603

root@embest:~# cat /sys/bus/platform/devices/TI-am335x-adc/iio\:device0/in_voltage5_raw

599

root@embest:~# cat /sys/bus/platform/devices/TI-am335x-adc/iio\:device0/in_voltage6_raw

767

root@embest:~# cat /sys/bus/platform/devices/TI-am335x-adc/iio\:device0/in_voltage7_raw

847

2.6 HDMI 测试

打开 SD 卡中 uEnv.txt 文件，修改 fdtfile=embest-SOM_PH8800-BB_SPH1800-HDMI.dtb
用 HDMI 数据线相连接显示设备并重新启动系统；

2.7 HDMI AUDIO 测试

连接 HDMI 设备，执行以下命令播放默认音频文件

```
root@embest:~# aplay /boot/firmware/audio_sample.wav
Playing WAVE '/boot/firmware/audio_sample.wav' : Signed 16 bit Little Endian, Rate 22050 Hz, Stereo
```

2.8 LCD 测试

4.3 寸屏：
打开 SD 卡中 uEnv.txt 文件，修改 fdtfile= embest-SOM_PH8800-BB_SPH1800-4.3inch_LCD.dtb
连接显示屏到 J9，重新启动系统
7 寸屏：
打开 SD 卡中 uEnv.txt 文件，修改 fdtfile= embest-SOM_PH8800-BB_SPH1800-7inch_LCD.dtb
连接显示屏到 J9，重新启动系统

2.9 背光测试

背光的亮度设置范围为（1—8），1 表示亮度最低，8 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/brightness
```

最亮：

```
root@embest:~# echo 8 > /sys/class/backlight/backlight/brightness
```

2.10 触摸屏测试

连接显示屏到 J9,在串口终端输入以下命令执行触摸屏校准程序：

```
root@embest:~# ts_calibrate
```

按照屏幕上提示，点击 “+” 图标 5 次完成校准。

2.11 串口测试

开发板上有 4 个串口，其中 UART0(CN4)为 debug 接口

2.11.1 UART1

短接 J4 第 2，3 号接口:

```
root@embest:~# ./uart_test -d /dev/ttyS5 -b 115200
/dev/ttyS5 SEND: 1234567890
/dev/ttyS5 RECV 1 total
/dev/ttyS5 RECV: 1
/dev/ttyS5 RECV 1 total
/dev/ttyS5 RECV: 2
/dev/ttyS5 RECV 1 total
/dev/ttyS5 RECV: 3
/dev/ttyS5 RECV 1 total
/dev/ttyS5 RECV: 4
/dev/ttyS5 RECV 1 total
/dev/ttyS5 RECV: 5
/dev/ttyS5 RECV 1 total
/dev/ttyS5 RECV: 6
/dev/ttyS5 RECV 1 total
/dev/ttyS5 RECV: 7
/dev/ttyS5 RECV 1 total
/dev/ttyS5 RECV: 8
/dev/ttyS5 RECV 1 total
/dev/ttyS5 RECV: 9
/dev/ttyS5 RECV 1 total
/dev/ttyS5 RECV: 0
```

注意：Ctrl+C 中断串口测试

2.11.2 UART2

短接 J58 第 16，17 号接口：

```
root@embest:~# ./uart_test -d /dev/ttyS3 -b 9600
/dev/ttyS3 SEND: 1234567890
/dev/ttyS3 RECV 1 total
/dev/ttyS3 RECV: 1
/dev/ttyS3 RECV 1 total
/dev/ttyS3 RECV: 2
```

/dev/ttyS3 RECV 1 total
/dev/ttyS3 RECV: 3
/dev/ttyS3 RECV 1 total
/dev/ttyS3 RECV: 4
/dev/ttyS3 RECV 1 total
/dev/ttyS3 RECV: 5
/dev/ttyS3 RECV 1 total
/dev/ttyS3 RECV: 6
/dev/ttyS3 RECV 1 total
/dev/ttyS3 RECV: 7
/dev/ttyS3 RECV 1 total
/dev/ttyS3 RECV: 8
/dev/ttyS3 RECV 1 total
/dev/ttyS3 RECV: 9
/dev/ttyS3 RECV 1 total
/dev/ttyS3 RECV: 0

注意：Ctrl+C 中断串口测试

2.11.3 UART4

短接 J58 第 14，15 号接口：

root@embest:~# ./uart_test -d /dev/ttyS1 -b 9600
/dev/ttyS1 SEND: 1234567890
/dev/ttyS1 RECV 1 total
/dev/ttyS1 RECV: 1
/dev/ttyS1 RECV 1 total
/dev/ttyS1 RECV: 2
/dev/ttyS1 RECV 1 total
/dev/ttyS1 RECV: 3
/dev/ttyS1 RECV 1 total
/dev/ttyS1 RECV: 4
/dev/ttyS1 RECV 1 total
/dev/ttyS1 RECV: 5
/dev/ttyS1 RECV 1 total
/dev/ttyS1 RECV: 6
/dev/ttyS1 RECV 1 total
/dev/ttyS1 RECV: 7
/dev/ttyS1 RECV 1 total
/dev/ttyS1 RECV: 8

/dev/ttyS1 RECV 1 total

/dev/ttyS1 RECV: 9

/dev/ttyS1 RECV 1 total

/dev/ttyS1 RECV: 0

注意：Ctrl+C 中断串口测试

2.12 RS485 测试

2.12.1 RS485-2 和 RS485-3

分别短接 J62 的 7，9 号引脚；8，10 号引脚((即 RS485-A3 TO RS485-A2, RS485-B3 to RS485-B2)：

串口终端输入如下命令（在后台运行）：

root@embest:~# ./uart_test -d /dev/ttySC1 -b 9600 -s "a" &

接着输入：

root@embest:~# ./uart_test -d /dev/ttySC0 -b 9600 -s "c"

/dev/ttySC0 SEND: c

/dev/ttySC1 RECV 1 total

/dev/ttySC1 RECV: c

/dev/ttySC1 SEND: a

/dev/ttySC0 RECV 1 total

/dev/ttySC0 RECV: a

ttySC0,ttySC1 分别发送数据，并能接收数据；

2.13 CAN 测试

SBC-PH8800 上有两个 CAN，可以用自身的 CAN0 和 CAN1 进行测试。连接 J62 的 1,3 引脚，2，4 引脚测试方法如下：

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

[116.797032] c_can_platform 481cc000.can can0: setting BTR=1c1d BRPE=0000

root@embest:~# ip link set can1 up

[116.860898] c_can_platform 481d0000.can can1: setting BTR=1c1d BRPE=0000

2. 收发数据

can1 接收，can0 往 can1 发数据

root@embest:~# candump can1&

root@embest:~# cansend can0 123#01020304050607

root@embest:~# can1 123 [7] 01 02 03 04 05 06 07

用 ps 和 kill 命令关闭 candump，换成 can0 接收，can1 往 can0 发数据

root@embest:~# candump can0&

root@embest:~# cansend can1 123#11121314151617

root@embest:~# can0 123 [7] 11 12 13 14 15 16 17

3. 测试完毕关闭设备

root@embest:~# ip link set can0 down

read: Network is down

root@embest:~# [409.786888] c_can_platform 481cc000.can can0: setting BTR=1c1d BRPE=0000

root@embest:~# ip link set can1 down

[415.503272] c_can_platform 481d0000.can can1: setting BTR=1c1d BRPE=0000

[2]+ Exit 1 candump can0

用户可以根据以上命令进行相互收发测试，还可以设置不同的波特率进行通信，在设置不同波特率之前必须先关闭设备，可设置的波特率有：

25KBPS（250000）

50KBPS（50000）

125KBPS（125000）

500KBPS（500000）

650KBPS（650000）

1MKBPS（1000000 ）

以上的波特率均能正常通信，还有其它波特率可以设置，用户可以自己尝试，看能否通信。另外也可以外接其他板的 can 接口测试。

2.14 网络测试

连接网线到 J17，在串口终端中输入以下命令来设置 IP 地址：

root@embest:~# ifconfig eth0 192.168.2.64

网络测试：

root@embest:~# ping 192.168.2.1

eth1 测试时，断开 J17 网线，连接外接网卡，执行同样的命令（eth0 改成 eth1）。

2.15 USB 测试

2.15.1 Host 测试

将 U 盘插入 USB host 接口（J15）,串口显示磁盘信息:

```
[  937.902749] usb 1-1.2: new high-speed USB device number 4 using xhci-hcd
[  938.023750] usb 1-1.2: New USB device found, idVendor=058f, idProduct=6366
[  938.030999] usb 1-1.2: New USB device strings: Mfr=1, Product=2, SerialNumber=3
[  938.039779] usb 1-1.2: Product: Flash Card Reader/Writer
[  938.046076] usb 1-1.2: Manufacturer: Generic
[  938.050558] usb 1-1.2: SerialNumber: 058F63666438
[  938.059201] usb-storage 1-1.2:1.0: USB Mass Storage device detected
[  938.069433] scsi host3: usb-storage 1-1.2:1.0
[  939.073423] scsi 3:0:0:0: Direct-Access     Multiple Card  Reader     1.00 PQ: 0 ANSI: 0
[  939.551759] sd 3:0:0:0: [sda] 15515648 512-byte logical blocks: (7.94 GB/7.39 GiB)
[  939.560184] sd 3:0:0:0: [sda] Write Protect is off
[  939.568026] sd 3:0:0:0: [sda] No Caching mode page found
[  939.575739] sd 3:0:0:0: [sda] Assuming drive cache: write through
[  939.589938]  sda: sda1
[  939.600578] sd 3:0:0:0: [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；

2.15.2 OTG 测试

2.15.2.1 1 主设备

通过转接线连接 U 盘到 J13:

```
[  880.127626] xhci-hcd xhci-hcd.0.auto: xHCI Host Controller
[  880.134829] xhci-hcd xhci-hcd.0.auto: new USB bus registered, assigned bus number 3
[  880.148726] xhci-hcd xhci-hcd.0.auto: hcc params 0x0238f06d hci version 0x100 quirks 0x00010010
[  880.159328] xhci-hcd xhci-hcd.0.auto: irq 194, io mem 0x48390000
[  880.167206] usb usb3: New USB device found, idVendor=1d6b, idProduct=0002
[  880.175323] usb usb3: New USB device strings: Mfr=3, Product=2, SerialNumber=1
[  880.183769] usb usb3: Product: xHCI Host Controller
[  880.188905] usb usb3: Manufacturer: Linux 4.1.6+ xhci-hcd
[  880.195618] usb usb3: SerialNumber: xhci-hcd.0.auto
[  880.207218] hub 3-0:1.0: USB hub found
```

```
[    880.218080] hub 3-0:1.0: 1 port detected
[    880.222687] xhci-hcd xhci-hcd.0.auto: xHCI Host Controller
[    880.233442] xhci-hcd xhci-hcd.0.auto: new USB bus registered, assigned bus number 4
[    880.241707] usb usb4: We don't know the algorithms for LPM for this host, disabling LPM.
[    880.252038] usb usb4: New USB device found, idVendor=1d6b, idProduct=0003
[    880.260133] usb usb4: New USB device strings: Mfr=3, Product=2, SerialNumber=1
[    880.268622] usb usb4: Product: xHCI Host Controller
[    880.274473] usb usb4: Manufacturer: Linux 4.1.6+ xhci-hcd
[    880.280171] usb usb4: SerialNumber: xhci-hcd.0.auto
[    880.292998] hub 4-0:1.0: USB hub found
[    880.299620] hub 4-0:1.0: 1 port detected
[    880.532745] usb 3-1: new high-speed USB device number 2 using xhci-hcd
[    880.673750] usb 3-1: New USB device found, idVendor=058f, idProduct=6366
[    880.680830] usb 3-1: New USB device strings: Mfr=1, Product=2, SerialNumber=3
[    880.689456] usb 3-1: Product: Flash Card Reader/Writer
[    880.695612] usb 3-1: Manufacturer: Generic
[    880.699948] usb 3-1: SerialNumber: 058F63666438
[    880.713047] usb-storage 3-1:1.0: USB Mass Storage device detected
[    880.724837] scsi host2: usb-storage 3-1:1.0
[    881.733406] scsi 2:0:0:0: Direct-Access     Multiple Card    Reader           1.00 PQ: 0 ANSI: 0
[    882.211615] sd 2:0:0:0: [sda] 15515648 512-byte logical blocks: (7.94 GB/7.39 GiB)
[    882.220103] sd 2:0:0:0: [sda] Write Protect is off
[    882.227790] sd 2:0:0:0: [sda] No Caching mode page found
[    882.235398] sd 2:0:0:0: [sda] Assuming drive cache: write through
[    882.249459]  sda: sda1
[    882.260011] sd 2:0:0:0: [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；

2.15.2.2 2. 从设备

连接 J13 到 PC 端，打开设备管理器，识别到如下设备：

2.16 Camera 测试

2.16.1摄像模式

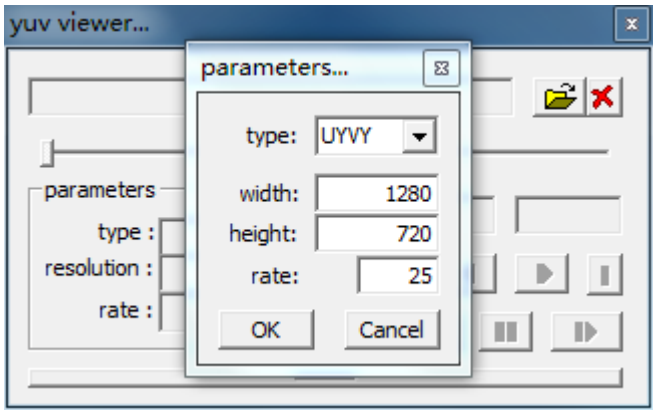
root@embest:~# ./mxc_v4l2_capture –iw1280 -ih 720 -ow 1280 -oh 720 -c 25 -f UYVY /boot/firmware/test.yuv

root@embest:~# sync

摄像头会录制一段分辨率 1280*720，帧率 25 的视频。

用 Ctrl+C 结束录制。此时在 sd 卡下会生成 test.yuv 文件。

连接 SD 卡到电脑，用 Pyuv.exe 打开。参数设置如下：



注意：Tool 目录下提供了 Pyuv.exe.

目前摄像头支持的最大分辨率是 720P(1280*720).

2.16.2拍照模式

root@embest:~# ./capture_jpeg_to_display 1.jpg

摄像头拍摄 640*480 格式的图片，并将图片全部图像显示到 LCD 屏

第3章 系统编译

3.1 配置编译环境

将 release 文件夹的所有内容拷贝到 Linux 环境下的$HOME 目录下（可能需要先解压 rar 文件），编译工具 gcc-linaro-4.9-2015.05-x86_64_arm-linux-gnueabihf 在$HOME/S5_tool 目录下，用如下命令解压：

$xz -d gcc-linaro-4.9-2015.05-x86_64_arm-linux-gnueabihf.tar.xz

$tar –xvf gcc-linaro-4.9-2015.05-x86_64_arm-linux-gnueabihf.tar

导入环境变量：

$export

CROSS_COMPILE=$HOME/tool/gcc-linaro-4.9-2015.05-x86_64_arm-linux-gnueabihf/bin/arm-linux-gnueabihf-

$export ARCH=arm

3.2 编译 UBOOT

3.2.1 获取 uboot 源码

Uboot 源码在$HOME/S4_Sourcecode/目录下，解压 u-boot*.tar.gz :

$ cd $HOME/S4_Sourcecode/

$ tar -zxvf u-boot*.tar.gz

3.2.2 编译并烧写镜像到 SD 卡

$ cd $HOME/S4_Sourcecode/u-boot

$ make distclean

$make som_ph8800_defconfig

$make

编译完成后在$HOME/S4_Sourcecode/u-boot 目录下生成 MLO, u-boot.img，将两个文件拷贝到 SD 卡中；

3.2.3 编译并烧写镜像 SPI Flash

$ cd $HOME/S4_Sourcecode/u-boot

$ make distclean

$make som_ph8800_qspiboot_defconfig

$make

编译完成后在$HOME/S4_Sourcecode/u-boot 目录下生成 u-boot.bin，$HOME/S4_Sourcecode/u-boot*/spl 目录下生成 u-boot-spl.bin,将两个文件拷贝到 SD 卡中;

从 SD 卡启动，在 uboot 阶段执行：

U-Boot# run update_qspi_flash

等待执行结束，这两个文件就烧写到 SPI flash 中。

（参考 1.3 从 SPI Flash 启动系统）

3.3 Kernel

3.3.1 获取内核源码

内核源码存在$HOME/S4_Sourcecode/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/ release/S4_Sourcecode/linux*

$ make distclean

$ make embest_ti_8800_defconfig

$ make

编译完成后在

- 目录$HOME/release/S4_Sourcecode/linux*/arch/arm/boot 下生成 zImage 文件。
- 目录 $HOME/ release/S4_Sourcecode/linux*/arch/arm/boot/dts 中生成下列 3 个文件：

1. embest-SOM_PH8700-BB_SPH1800-4.3inch_LCD.dtb
2. embest-SOM_PH8700-BB_SPH1800-7inch_LCD.dtb
3. embest-SOM_PH8700_BB_SPH1800-HDMI.dtb

dtb 文件分别对应 4.3 寸屏，7 寸屏，（配置方法参考 LCD 测试， HDMI 测试）

将文件拷贝到 SD 卡中。