用户手册

[EVK-PH8800]

历史版本

Rev.	Note	Author
20160307	Initial	Baijy
20160315	1.增加 RS485-1 测试 2.修改 dtb 文件	Baijy
20160322	1. 增加 Wi-Fi 和蓝牙测试 2. 增加 PWRON RESETn 按键测试 3. 修改错误的命令	Rongdong
20160331	1. 增加 CAN 测试	Baijy
20160511	1. 更新命令返回结果到最新版本 2. U-boot 的 ETH 网口由 ETH0 改为 ETH1	Sandy
20160622	Rev01 版本	Sandy

目录

历史版本 2
目录 3
Release Note 5
1. 镜像版本 5
2. 功能列表 5
3. 已知问题 6
第 1 章 快速启动 7
1.1 烧写镜像到 SD 卡 7
1.2 从 SD 卡启动系统 8
1.3 从 SPI Flash 启动 9
第 2 章 功能测试 11
2.1 LED 测试 11
2.2 按键测试 11
2.2.1 KEY_MENU, KEY_BACK 11
2.2.2 PWRON_RESETn 12
2.3 RTC 测试 13
2.4 EEPROM 测试 13
2.5 EMMC 测试 14
2.6 ADC 测试 15
2.7 蜂鸣器测试 15
2.8 音频测试 15
2.9 HDMI/VGA 测试 16
2.10 LCD 测试 16
2.11 背光测试 16
2.12 触摸屏测试 16
2.13 串口测试 16
2.13.1 UART1 16
2.13.2 UART4 17
2.14 RS485 测试 18
2.14.1 RS485-2 和 RS485-3 18
2.14.2 RS485-1 和 RS485-2 18

2.15 CAN 测试 19
2.16 网络测试 19
2.17 USB 测试 19
2.17.1 Host 测试 19
2.17.2 OTG 测试 20
2.18 WIFI 测试 21
2.18.1 配置 WIFI 频段 21
2.18.2 连接 WIFI 22
2.19 Bluetooth 测试 24
2.19.1 复位蓝牙模块 24
2.19.2 初始化蓝牙模块 24
2.19.3 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 SPI Flash 25
3.3 Kernel 26
3.3.1 获取内核源码 26
3.3.2 编译并烧写镜像到 SD 卡 26

Release Note

1. 镜像版本

EVK-PH8800-Release-SDcard-EMMC-REV01.img

2. 功能列表

Feature List	EVK-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				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
SDCard	PH1800-P5-J3	MMC0	Null	Can access read write and boot
MicroSD_(TF)	PH1800-P5-J2	MMC0	Null	Can access read write and boot
External-RTC	PH8800-p8-u13	I2C1	DS3231SN	can read write and keep time off power
Integrated-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
Buzzer	PH1800-P14-PZ1	gpio	Null	System can control buzzer to beep or not
ADC	PH8800-P11-J1	ADC	Null	Can read the ad value from pin
Power-Button	PH1800-P14-S2	I2C0	TPS65218	Can get key value
ADC-Keys	PH1800-P14-S4/S5	ADC	Null	Can get key value
LCD	PH1800-P8-J9	RGB	Null	Can show picture on the screen
Backlight	PH1800-P8-J9	PWM	Null	System can control the LCD backlight
TouchScreen	PH1800-P8-J9	ADC-TSC	Null	System use touchscreen
VGA	PH1800-P9-U14	I2C1	CH7033	Can show picture on the screen
HDMI	PH1800-P9-U14	I2C1	CH7033	Can show picture on the screen
eMMC	PH8800-p8-u14	MMC1	MTFC4GACAAAM-4M IT	Can access read write

EEPROM	PH8800-p8-u6	I2C0	CAT24C256W	Can access read write
SPI-FLASH	EC8800-p8-u3	QSPI	N25Q256A13EF840	1. Boot from SPI-Flash
SPI	PH1800-P14-J22	SPI1	Null	System can send and receive data in loopback mode
CAN-1	PH1800-P14-J22	CAN1	Null	System can send and receive data between two board
CAN-2	PH1800-P7-u9/u10	CAN0	TJA1040T	System can send and receive data between two board
UART-0	PH1800-P6-U4	UART0	MAX3232CUE+	System can send and receive data in loopback mode
UART-1	PH1800-P14-J21	UART5	Null	System can send and receive data in loopback mode
UART-2	PH1800-P14-J21	UART3	Null	System can send and receive data in loopback mode
UART-4	PH1800-P6-U4	UART1	MAX3232CUE+	System can send and receive data in loopback mode
RS485-1	PH1800-P7-U12	UART3	ADM2483	System can send and receive data between two board
RS485-2	PH1800-P6-U5	spi2	SC16IS762IPW	System can send and receive data between two board
RS485-3	PH1800-P6-U5	spi2	SC16IS762IPW	System can send and receive data between two board
USB-Host	PH1800-P10-U17	USB1	USB2514	Can recognize U disk by USB host
USB-OTG	PH1800-P10-J13	USB0	Null	Can recognize U disk in host mode, and can work as usb ethernet in device mode
Audio	PH1800-P12-U19	I2C1&Mcasp0	WM8904	can play and record wav
Ethernet-1	PH8800-P9-U9	RGMII1	KSZ9031RNXIA	Can ping the server
Ethernet-2	PH1800-P11-U18	RGMII2	AR8035	Can ping the server
WIFI & Bluetooth	PH1800-P13-J24/J25	UART1&MMC2 &MCAPS0&I2C1	EXP-WFB00(Jorjin WG7801-D0)	1. Can ping the server using 2.4Ghz 2. Can search bluetooth device

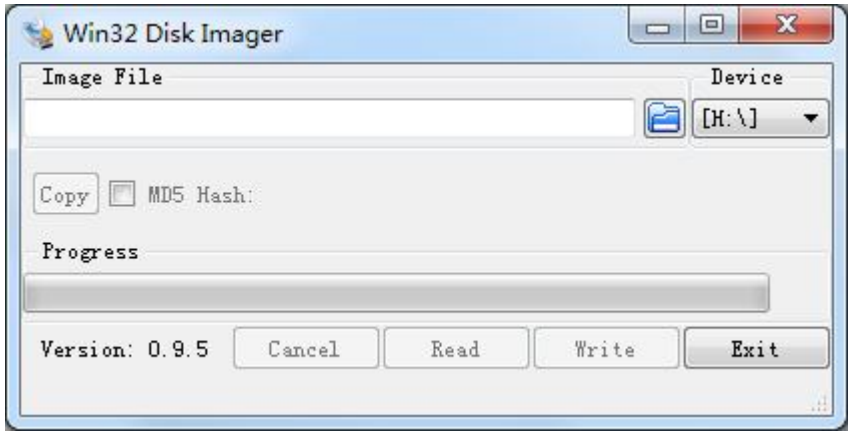
3. 已知问题

Known issue List	Detail
SPI-FLASH	Not Support: SPI-Flash access in kernel
CAMERA	Not Support: Could preview, take picture and record video
Ethernet-1 & Ethernet -2	Bug: Board to board connect could not working normally
LCD	Bug:4.3 inch Screen turn white for a while in boot

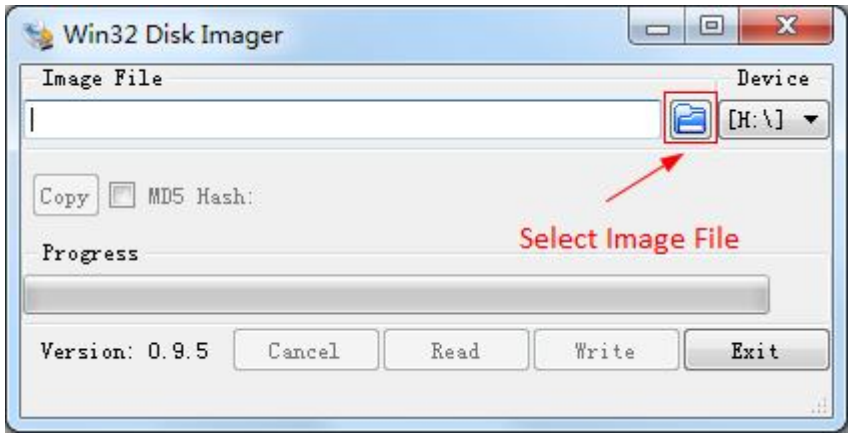
第1章 快速启动

1.1 烧写镜像到 SD 卡

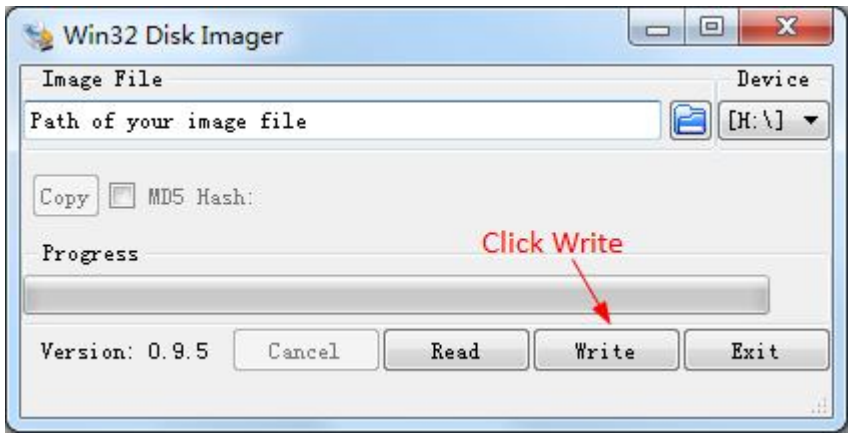
- 首先，你需要准备一张不小于 2G 的 SD 卡
- 然后，你需要从 https://sourceforge.net/projects/win32diskimager/ 下载并安装 Win32 Disk Imager



- 选择需要烧写的镜像，EVK-PH8800-Release-REV01\image\EVK-PH8800-Release-SDcard-EMMC-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
- 用交叉串口线把板子上的 DEBUG 接口(J4)和 PC 的串口相连
- 把 SD 卡插入板上的插槽(J3 或 J2)
- 按下按键 S3，用 12V,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

```
         Starting System Logging Service...
         Starting Permit User Sessions...
[  OK  ] Started Restore Sound Card State.
[  OK  ] Started /etc/rc.local Compatibility.
[  OK  ] Started Permit User Sessions.
[  OK  ] Started System Logging Service.
[   14.511257] random: nonblocking pool is initialized
[  OK  ] Started Login Service.
         Starting Getty on tty1...
[  OK  ] Started Getty on tty1.
         Starting Serial Getty on ttyS0...
[  OK  ] Started Serial Getty on ttyS0.
[  OK  ] Reached target Login Prompts.
[   14.861156] FAT-fs (mmcblk0p1): Volume was not properly unmounted. Some data
may be corrupt. Please run fsck.
[  OK  ] Started Embest AutoExec Service.

Debian GNU/Linux 8 embest ttyS0

embest login:
```

输入用户名和密码 root 登录；

```
Debian GNU/Linux 8 embest ttyS0

embest login: root
Password:
Last login: Sat Jan  1 00:24:40 UTC 2000 on ttyS0
Linux embest 4.1.6 #1 PREEMPT Mon Jun 20 16:32:05 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 (Jun 20 2016 - 16:13:34)

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 (Jun 20 2016 - 16:13:34 +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: 1

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 8 ms (6.8 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

288540 bytes read in 19 ms (14.5 MiB/s)

device 0 offset 0x20000, size 0x4671c

SF: 288540 bytes @ 0x20000 Written: OK

输入下列命令从 SD 卡启动系统：

U-Boot# boot

将 EVK-PH8800-Release-SDcard-EMMC-REV01.img 拷贝到 U 盘，将 U 盘插入 USB 接口（J15）：

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

/dev/sda /dev/sda1

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

root@embest:~# dd if=/mnt/EVK-PH8800-Release-SDcard-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 按键测试

2.2.1 KEY_MENU, KEY_BACK

在串口终端输入：

```
root@embest:~# evtest /dev/input/event0
Input driver version is 1.0.1
Input device ID: bus 0x10 vendor 0x1 product 0x1 version 0x100
Input device name: "adc_keypad"
Supported events:
  Event type 0 (EV_SYN)
  Event type 1 (EV_KEY)
    Event code 139 (KEY_MENU)
    Event code 158 (KEY_BACK)
Key repeat handling:
  Repeat type 20 (EV_REP)
    Repeat code 0 (REP_DELAY)
      Value      -1
    Repeat code 1 (REP_PERIOD)
      Value      -1
```

Properties:

Testing ... (interrupt to exit)

按下按键：

Event: time 946685117.143847, type 1 (EV_KEY), code 158 (KEY_BACK), value 1

Event: time 946685117.143847, ---------- EV_SYN ----------

[320.052799] input input0: key 158 up

Event: time 946685117.227621, type 1 (EV_KEY), code 158 (KEY_BACK), value 0

Event: time 946685117.227621, ---------- EV_SYN ----------

Event: time 946685119.813824, type 1 (EV_KEY), code 139 (KEY_MENU), value 1

Event: time 946685119.813824, ---------- EV_SYN ----------

[322.772800] input input0: key 139 up

Event: time 946685119.947630, type 1 (EV_KEY), code 139 (KEY_MENU), value 0

Event: time 946685119.947630, ---------- EV_SYN ----------

测试成功按 Ctrl+C 退出；

2.2.2 PWRON_RESETn

长按 8s 以上，系统复位；8s 以下作为普通按键使用，方法如 KEY_MENU ， KEY_BACK；

在串口终端输入：

root@embest:~# evtest /dev/input/event2

Input driver version is 1.0.1

Input device ID: bus 0x18 vendor 0x0 product 0x0 version 0x0

Input device name: "tps65218_pwrbutton"

Supported events:

Event type 0 (EV_SYN)

Event type 1 (EV_KEY)

Event code 116 (KEY_POWER)

Properties:

Testing ... (interrupt to exit)

按下按键：

Event: time 946685191.953554, type 1 (EV_KEY), code 116 (KEY_POWER), value 1

Event: time 946685191.953554, ---------- EV_SYN ----------

Event: time 946685192.114087, type 1 (EV_KEY), code 116 (KEY_POWER), value 0

Event: time 946685192.114087, ---------- EV_SYN ----------

2.3 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

2.4 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；

2.5 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；

2.6 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

391

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

529

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

3989

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

3996

2.7 蜂鸣器测试

打开蜂鸣器：

root@embest:~# echo 1 > /sys/class/misc/buzzer_ctl/state

关闭蜂鸣器：

root@embest:~# echo 0 > /sys/class/misc/buzzer_ctl/state

2.8 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

2.9 HDMI/VGA 测试

打开 SD 卡中 uEnv.txt 文件，修改 fdtfile=embest-SOM_PH8800-BB_EPH1800-HDMI-VGA.dtb

连接 HDMI 或 VGA 重新启动系统

2.10 LCD 测试

4.3 寸屏：

打开 SD 卡中 uEnv.txt 文件，修改 fdtfile= embest-SOM_PH8800-BB_EPH1800-4.3inch_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_EPH1800-7inch_LCD.dtb

连接显示屏到 J9，重新启动系统

2.11 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
```

2.12 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。

2.13 串口测试

开发板上有 4 个串口，其中 UART0(J4)为 debug 接口；UART2 为 RS485 功能

2.13.1 UART1

短接 J21 第 3，5 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

2.13.2 UART4

短接 J5 第 2，3 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

2.14 RS485 测试

2.14.1 RS485-2 和 RS485-3

分别短接 J6 的 8，5 号引脚；7，4 号引脚：

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；

2.14.2 RS485-1 和 RS485-2

短接 J7 的 4 号引脚和 J6 的 7 号引脚；短接 J7 的 5 号引脚和 J6 的 8 号引脚：

串口终端输入如下命令（在后台运行）：

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

接着输入：

root@embest:~# ./uart_test -d /dev/ttyS3 -b 9600 -s "c"

/dev/ttyS3 SEND: c

/dev/ttySC0 RECV 1 total

/dev/ttySC0 RECV: c

/dev/ttySC0 SEND: a

/dev/ttyS3 RECV 1 total

/dev/ttyS3 RECV: a

ttySC0,ttyS3 分别发送数据，并能接收数据；

2.15 CAN 测试

测试方法如下：

连接 J7 的 7,8 引脚，在串口终端输入：

```
root@embest:~# ip link set can0 type can bitrate 50000 loopback on
root@embest:~# ip link set can0 up
[ 1050.007965] c_can_platform 481cc000.can can0: setting BTR=1c1d BRPE=0000
```

执行以下命令在后台接受数据包:

```
root@embest:~# candump can0 &
```

执行以下命令在发送数据包:

```
root@embest:~# cansend can0 123#11223344556677
root@embest:~#  can0  123  [7]  11 22 33 44 55 66 77
  can0  123  [7]  11 22 33 44 55 66 77
```

关闭设备：

```
root@embest:~# ip link set can0 down
read: Network is down
[ 1130.014498] c_can_platform 481cc000.can can0: setting BTR=1c1d BRPE=0000
[1]+  Exit 1                  candump can0
```

2.16 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
```

测试 eth1 时，将网线接到 J16，上述命令中 eth0 改成 eth1 即可。

2.17 USB 测试

2.17.1 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；

2.17.2 OTG 测试

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. 从设备

连接 J13 到 PC 端，打开设备管理器，识别到如下设备：

2.18 WIFI 测试

2.18.1配置 WIFI 频段

开机进入系统后，默认为 2.4G 频段，如果要使用 5G 频段，请先配置 WIFI，提供两种配置方法，配置方法如下：

1. 进入目录 /usr/sbin/wlconf，输入命令./configure-device.sh

root@embest:~# cd /usr/sbin/wlconf/

root@embest:/usr/sbin/wlconf# ./configure-device.sh

根据提示 输入： y 1837 y 2 2,就可以了

Please provide the following information.

Are you using a TI module? [y/n] : **y**

What is the chip flavor? [1801/1805/1807/1831/1835/1837 or 0 for unknown] : **1837**

Should Japanese standards be applied? [y/n] : **y**

How many 2.4GHz antennas are fitted? [1/2] : **2**

How many 5GHz antennas are fitted? [0/1/2] : **2**

[1461.083174] wlcore: down

The device has been successfully configured.

TI Module: y

Chip Flavor: 1837

Number of 2.4GHz Antennas Fitted: 2

Number of 5GHz Antennas Fitted: 2

Diversity Support: y

SISO40 Support: y

Japanese Standards Applied: y

Class 2 Permissive Change (C2PC) Applied: n

root@embest:/usr/sbin/wlconf# [1461.954230] wlcore: wl18xx HW: 183x or 180x, PG 2.2 (ROM 0x11)

[1462.005515] wlcore: loaded

[1462.008412] wlcore: driver version: R8.6_SP1

[1462.362905] wlcore: PHY firmware version: Rev 8.2.0.0.233

[1462.595072] wlcore: firmware booted (Rev 8.9.0.1.55)

2. 进入目录 /usr/sbin/wlconf，输入命令：

root@embest:~# cd /usr/sbin/wlconf

root@embest:/usr/sbin/wlconf# ./wlconf -o /lib/firmware/ti-connectivity/wl18xx-conf.bin -I /usr/sbin/wlconf/official_inis/WG7833-B0A_INI_rev1.ini

两个方法任选其一就可以，配置之后就可以使用 5GWIFI 了，同时这个配置也是兼容 2.4G 的。使用 5G WIFI 只需要在第一次使用前配置一下，再次使用无需配置。

2.18.2连接 WIFI

在串口终端输入：

root@embest:~# cd /usr/share/wl18xx/

root@embest:/usr/share/wl18xx# ./sta_start.sh

root@embest:/usr/share/wl18xx# Successfully initialized wpa_supplicant

[94.422934] cfg80211: Calling CRDA for country: US

Could not read interface p2p-dev-wlan0 flags: No such device

[94.599340] cfg80211: Regulatory domain changed to country: US

[94.605627] cfg80211: DFS Master region: FCC

[94.610029] cfg80211: (start_freq - end_freq @ bandwidth), (max_antenna_gain, max_eirp), (dfs_cac_time)

[94.621813] cfg80211: (2402000 KHz - 2472000 KHz @ 40000 KHz), (N/A, 3000 mBm), (N/A)

[94.631326] cfg80211: (5170000 KHz - 5250000 KHz @ 80000 KHz, 160000 KHz AUTO), (N/A, 1700 mBm), (N/A)

[94.642261] cfg80211: (5250000 KHz - 5330000 KHz @ 80000 KHz, 160000 KHz AUTO), (N/A, 2300 mBm), (0 s)

[94.654119] cfg80211: (5490000 KHz - 5730000 KHz @ 160000 KHz), (N/A, 2300 mBm), (0 s)

[94.662666] cfg80211: (5735000 KHz - 5835000 KHz @ 80000 KHz), (N/A, 3000 mBm), (N/A)

[94.672235] cfg80211: (57240000 KHz - 63720000 KHz @ 2160000 KHz), (N/A, 4000 mBm), (N/A)

p2p-dev-wlan0: CTRL-EVENT-REGDOM-CHANGE init=USER type=COUNTRY alpha2=US

root@embest:/usr/share/wl18xx# ./sta_connect-ex.sh embest-test WPA-PSK 12345678

其中 embest-test 是 SSID， 12345678 是 Wi-Fi 密码

netid=0

==========

OK

OK

OK

OK

root@embest:/usr/share/wl18xx# wlan0: SME: Trying to authenticate with b0:48:7a[1017.520349] wlan0: authenticate with b0:48:7a:4b:0b:2a

:4b:0b:2a (SSID='embest-test' freq=2437 MHz)

[1017.531999] wlan0: send auth to b0:48:7a:4b:0b:2a (try 1/3)

[1017.571449] wlan0: authenticated

wlan0: Trying to associate with b0:48:7a:4b:0b:2a (SSID='embest-test' freq=2437 MHz)

[1017.583246] wlan0: associate with b0:48:7a:4b:0b:2a (try 1/3)

[1017.721188] wlan0: RX AssocResp from b0:48:7a:4b:0b:2a (capab=0x431 status=0 aid=2)

[1017.735614] wlan0: associated

wlan0: Associated with b0:48:7a:4b:0b:2a[1017.739377] cfg80211: Calling CRDA for country: US

[1017.764361] cfg80211: Regulatory domain changed to country: US

[1017.770526] cfg80211: DFS Master region: FCC

[1017.775904] cfg80211: (start_freq - end_freq @ bandwidth), (max_antenna_gain, max_eirp), (dfs_cac_time)

[1017.786369] cfg80211: (2402000 KHz - 2472000 KHz @ 40000 KHz), (N/A, 3000 mBm), (N/A)

[1017.795875] cfg80211: (5170000 KHz - 5250000 KHz @ 80000 KHz, 160000 KHz AUTO), (N/A, 1700 mBm), (N/A)

[1017.807298] cfg80211: (5250000 KHz - 5330000 KHz @ 80000 KHz, 160000 KHz AUTO), (N/A, 2300 mBm), (0 s)

[1017.818171] cfg80211: (5490000 KHz - 5730000 KHz @ 160000 KHz), (N/A, 2300 mBm), (0 s)

[1017.827331] cfg80211: (5735000 KHz - 5835000 KHz @ 80000 KHz), (N/A, 3000 mBm), (N/A)

[1017.836317] cfg80211: (57240000 KHz - 63720000 KHz @ 2160000 KHz), (N/A, 4000 mBm), (N/A)

p2p-dev-wlan0: CTRL-EVENT-REGDOM-CHANGE init=COUNTRY_IE type=COUNTRY alpha2=US

wlan0: WPA: Key negotiation completed with b0:48:7a:4b:0b:2a [PTK=CCMP GTK=TKIP]

wlan0: CTRL-EV[1017.906052] wlcore: Association completed.

ENT-CONNECTED - Connection to b0:48:7a:4b:0b:2a completed [id=3 id_str=]

用 ping 命令测试 wifi 连接

root@embest:/usr/share/wl18xx# 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

2.19 Bluetooth 测试

2.19.1复位蓝牙模块

root@embest:~# echo 0 > /sys/class/leds/PH1800\:bt_en/brightness

root@embest:~# echo 1 > /sys/class/leds/PH1800\:bt_en/brightness

2.19.2初始化蓝牙模块

root@embest:~# hciattach /dev/ttyS5 texas 115200

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/TIInit_11.8.32.bts

Loaded BTS script version 1

texas: changing baud rate to 3000000, flow control to 1

Device setup complete

2.19.3测试蓝牙功能

root@embest:~# hciconfig hci0 up

root@embest:~# hcitool scan

超级终端窗口显示信息如下：

Scanning ...

00:12:FE:B7:75:A0 Lenovo-TD80t

第3章 系统编译

3.1 配置编译环境

将 EVK-PH8800-Release-REV01 文件夹拷贝到 Linux 环境下的$HOME 目录下，编译工具 gcc-linaro-4.9-2015.05-x86_64_arm-linux-gnueabihf 在$HOME/EVK-PH8800-Release-REV01/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/EVK-PH8800-Release-REV01/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/EVK-PH8800-Release-REV01/sourcecode/目录下，解压 u-boot*.tar.gz :

$ cd $HOME/EVK-PH8800-Release-REV01/sourcecode/

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

3.2.2 编译并烧写镜像到 SD 卡

$ cd $HOME/EVK-PH8800-Release-REV01/sourcecode/u-boot*

$ make distclean

$make som_ph8800_defconfig

$make

编译完成后在$HOME/EVK-PH8800-Release-REV01/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/EVK-PH8800-Release-REV01/sourcecode/u-boot*

$ make distclean

$make som_ph8800_qspiboot_defconfig

$make

编译完成后在$HOME/EVK-PH8800-Release-REV01/sourcecode/u-boot*目录下生成 u-boot.bin，

$HOME/EVK-PH8800-Release-REV01/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/EVK-PH8800-Release-REV01/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/EVK-PH8800-Release-REV01/sourcecode/linux*

$ make distclean

$ make embest_ti_8800_defconfig

$ make

编译完成后在$HOME/EVK-PH8800-Release-REV01/sourcecode/linux*/arch/arm/boot 目录下生成 zImage

在$HOME/EVK-PH8800-Release-REV01/sourcecode/linux*/arch/arm/boot/dts 目录下生成

embest-SOM_PH8800-BB_EPH1800-4.3inch_LCD.dtb

embest-SOM_PH8800-BB_EPH1800-7inch_LCD.dtb

embest-SOM_PH8800-BB_EPH1800-HDMI-VGA.dtb

dtb 文件分别对应 4.3 寸屏，7 寸屏，HDMI 显示（配置方法参考 HDMI/VGA 测试，LCD 测试;

将文件拷贝到 SD 卡中。