User Manual

[EVK-PH8800]

Revision History

Rev.	Note	Author
20160307	Initial	Baijy
20160314	Translate	Sandy
20160315	Add RS485-1 Test. Modify the name of dtb files.	Baijy
20160323	Add WIFI and Bluetooth test Add PWRON RESETn Keypad test Modify some wrong instructions	Rongdong
20160331	Add CAN test	Baijy
20160511	Update the result of instructions to the newest version Modify the ETH interface of U-boot from ETH0 to ETH1	Sandy
20160622	Rev01 Release	Sandy

Catalog

Release Note

1. Images Version

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

2. Feature List

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-4MIT	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)	Can ping the server using 2.4Ghz Can search bluetooth device

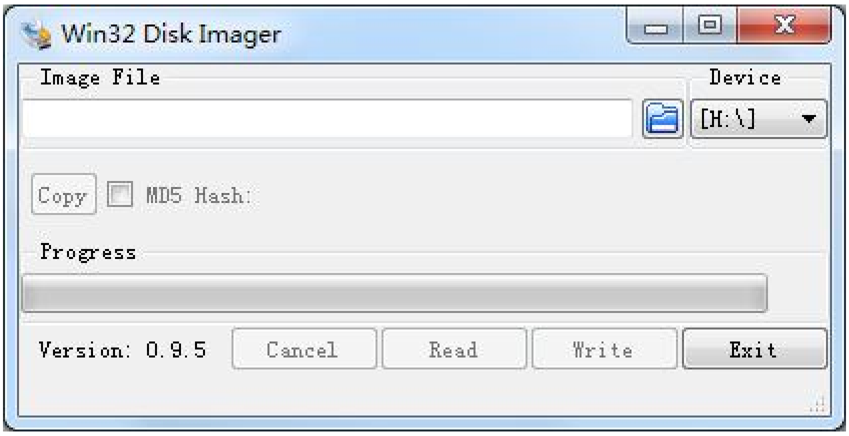
3. Known Issues

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

Chapter 1 Quick Start

1.1 Burn the System Images to the SD Card

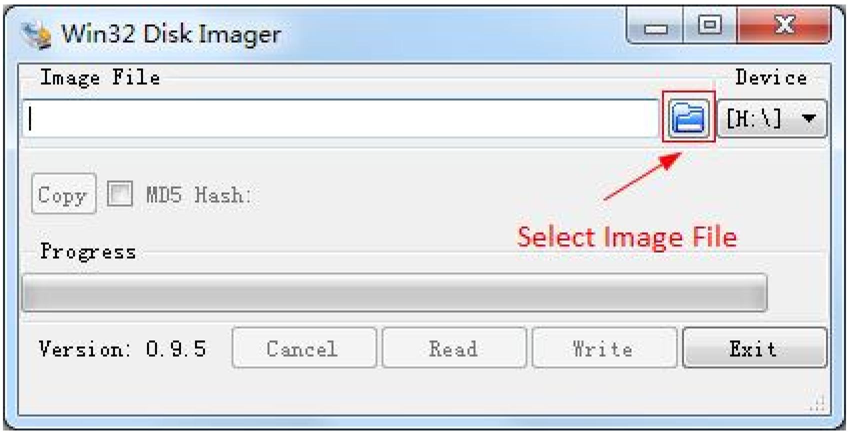
- Firstly, you should prepare a SD card, which is no less than 2GB.
- Then, download and install “Win32 Disk Imager” from https://sourceforge.net/projects/win32diskimager/.



Device: [H:\]

Version: 0.9.5

- Select system image: EVK-PH8800-Release-REV01\image\EVK-PH8800-Release-SDcard-EMMC-REV01.img



Device: [H:\]

Version: 0.9.5

- Click “Write” button to burn the images:

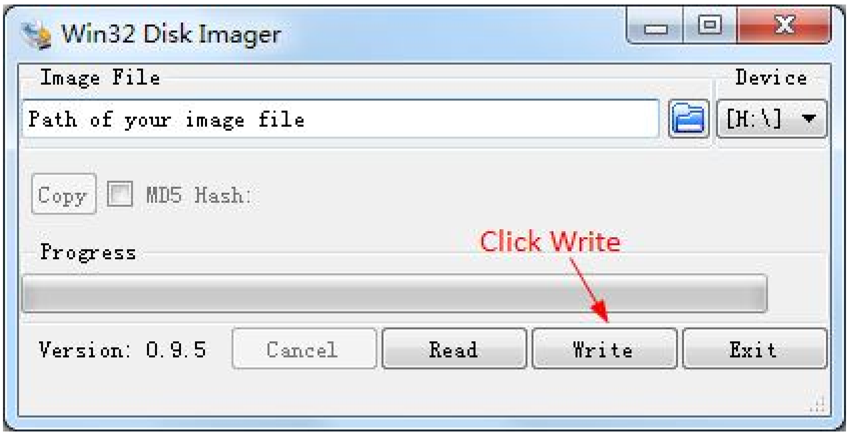


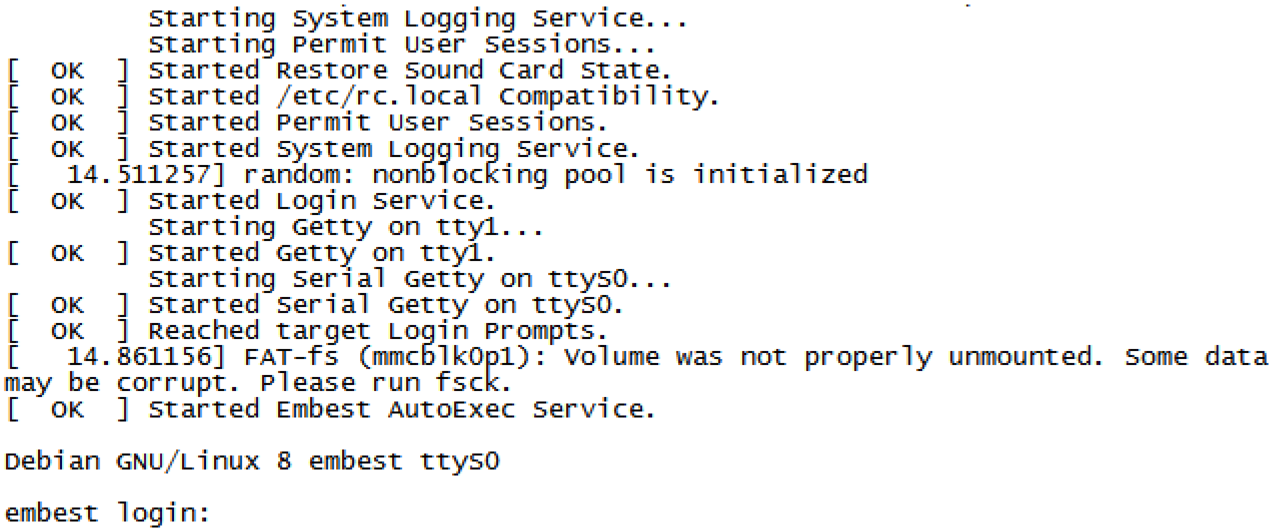
Image File: Path of your image file

Device: [H:\]

Version: 0.9.5

1.2 System Boot from SD Card

- Install the Serial Communication software (e.g. SecureCRT), select the corresponding port number, baudrate as 115200, data bits as 8, stop bits as 1, parity as none.
- Connect the DEBUG interface (J4) to the serial interface of PC with a DB9 crossed serial cable.
- Insert the SD card into the card slot (J3 or J2).
- Press S3 button, then powered the board with a 12V, 2A power. Release S3 after the power reset.
- Wait for the system boot up, then the serial output will show the following information:

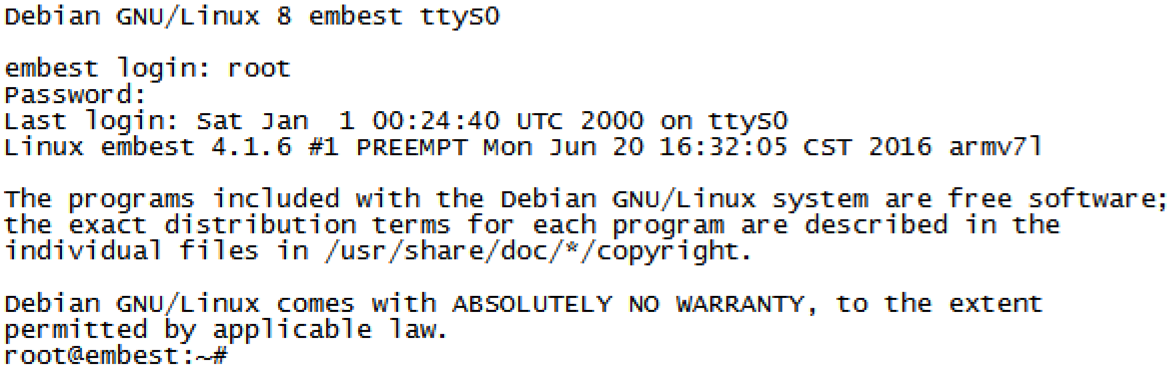


```
         Starting System Logging Service...
         Starting Permit User Sessions...
[  OK  ] Started Restore Sound Card State.
[  OK  ] Started /etc/rc.local Compatibility.
[  OK  ] Started Permit User Sessions.
[  OK  ] Started System Logging Service.
[   14.511257] random: nonblocking pool is 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:
```

Enter username and password as “root” to login;



```
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 System Boot from SPI Flash

Refer to 1.2, boot the system from SD Card, press “Enter” when the serial terminal prints the following info:

```
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#
```

(Press Enter now.)

Execute the following instructions on the serial terminal:

```
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
```

Enter following instruction to boot from SD Card first:

```
U-Boot# boot
```

Copy the EVK-PH8800-Release-SDcard-EMMC-REV01.img to a U-disk, then plug the U-disk to J15;

Execute the following instructions on the serial terminal:

```
root@embest:~# ls /dev/sd*
/dev/sda    /dev/sda1
root@embest:~# mount /dev/sda /mnt/
root@embest:~# dd    if=/mnt/EVK-PH8800-Release-SDcard-EMMC-REV01.img    of=/dev/mmcblk1
```

Note: Burn the EMMC takes a long time, please wait patiently.

Then power reset the board to boot from EMMC (**Don’t press S3 anymore**).

Chapter 2 Function test

First of all, please refer to Chapter 1.1 and boot up the system. Then test the functions according to the following guidance.

2.1 LED

User can control LED (D3, D4) indicators on SOM-PH8800 Board. After the system boot up, please execute the following instructions in serial terminal to implement the test; (D3 is attached to user_leds_d3, D4 to user_leds_d4)

Light out LED:

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

Light up 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 Button

2.2.1 KEY_MENU, KEY_BACK

Execute the following instructions on the serial terminal:

```
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)
```

Press the buttons:

```
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 ------------
```

Note: Press “CTRL+C” to exit the test.

2.2.2 PWRON_RESETn

Press PWR-RST button for more than 8s will reset the system;

Press PWR-RST button for less than 8s will work as normal button, like KEY_MENU and KEY_BACK.

Execute the following instructions on the serial terminal:

```
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)
```

Press the button:

```
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

Execute the following instructions on the serial terminal:

Check the current system time:

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

Set current time as 10:46, March 9, 2016

```
root@embest: # date 030910462016
Wed Mar  9 10:46:00 UTC 2016
```

Write system clock into RTC:

```
root@embest: # hwclock –w
```

Read RTC value:

```
root@embest: # hwclock
Wed 09 Mar 2016 10:46:23 AM UTC  -0.432561 seconds
```

The above information indicates that the hardware clock-RTC-has been set to March 9, 2016, so the system clock is saved in the hardware clock.

Reboot the system and check the current system time:

```
root@embest:~# date
Wed Mar  9 10:46:45 UTC 2016
```

2.4 EEPROM

Execute the following instructions on the serial terminal:

```
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
```

If write and read data are the same, the test passes.

2.5 EMMC

Execute the following instructions on the serial terminal:

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

Modify emmc_write value:

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

E.g. Write “emmc write test” into the system

Write emmc instructions:

```
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
```

Read emmc instructions:

```
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
```

Check emmc_read value:

```
root@embest:~# cat emmc_read
emmc write test
```

Test passes;

2.6 ADC

Execute the following instructions on the serial terminal to get the sampling values returned:

```
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 Buzzer

Open the buzzer:

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

Shut off the buzzer:

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

2.8 Audio

The board has input/output interfaces which support audio recording and playing. Users can test the audio function with the following instructions:

Audio recording test will generate audio file K:

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

Audio playing test will play the audio file K:

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

2.9 HDMI/VGA

Open the uEnv.txt file from SD card, modify fdtfile=embest-SOM_PH8800-BB_EPH1800-HDMI-VGA.dtb

Connect HDMI/VGA cable, then reboot the system

2.10 LCD

4.3” LCD:

Open the uEnv.txt file from SD card, modify fdtfile= embest-SOM_PH8800-BB_EPH1800-4.3inch_LCD.dtb

Connect the screen module to J9, then reboot the system.

7” LCD:

Open the uEnv.txt file from SD card, modify fdtfile= embest-SOM_PH8800-BB_EPH1800-7inch_LCD.dtb

Connect the screen module to J9, then reboot the system.

2.11 Backlight

The backlight brightness has a range from 1 to 8, in which 8 means highest brightness, 1 means lowest.

Execute the following instructions on the serial terminal to implement the backlight test:

The darkest:

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

The brightest:

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

2.12 Touchscreen

Connect the screen module to J9, execute the following instructions on the serial terminal to implement the touch screen calibration program:

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

Following the notes on LCD, click the “+” icon for five times to complete the calibration.

2.13 Serial

The board has 4 serial interfaces, while the UART0 (J4) is the debug interface, UART2 is used as RS485. Execute the following instructions on the serial terminal to test UART 1and UART4:

2.13.1 UART1

Short Pin 3 and 5 in J21:

```
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
```

Note: Press “CTRL+C” to exit the serial test.

2.13.2 UART4

Short Pin 2 and 3 in J5:

```
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
```

Note: Press “CTRL+C” to exit the serial test.

2.14 RS485

2.14.1 RS485-2 and RS485-3

Short connect Pin 5 and 8, Pin 4 and 7 in J6:

Execute the following instructions on the serial terminal (in the background):

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

Then enter the following:

```
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 will send data separately, receive data correctly;

2.14.2 RS485-1 and RS485-2

Short connect Pin 4 in J7 with Pin 7 in J6, Pin 5 in J7 with Pin 8 in J6

Execute the following instructions on the serial terminal (in the background):

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

Then enter the following:

```
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 will send data separately, receive data correctly;

2.15 CAN

Test method as below:

Connect Pin 7 and 8 in J7, then execute the following instructions on the serial terminal:

```
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
```

Execute the following instructions to receive data packet in the background:

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

Execute the following instructions to send data packet:

```
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
```

Shut off the device:

```
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 Network

Execute the following instructions on the serial terminal:

Configure the IP address:

```
root@embest:~# ifconfig eth0 192.168.2.64
```

Testing network interface:

```
root@embest:~# ping 192.168.2.1
```

2.17 USB

2.17.1 USB Host

Insert the U disk to the USB Host interface (J15), serial terminal will display the disk information:

```
[  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
```

Execute the following instructions on the serial terminal:

```
root@embest:~# ls /dev/sd*
/dev/sda      /dev/sda1
```

Storage nodes locate under /dev;

2.17.2 USB OTG

1. Master Device

Connect U disk to J13 with an OTG cable:

```
[   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.
```

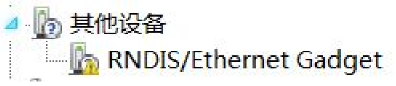
Execute the following instructions on the serial terminal:

```
root@embest:~# ls /dev/sd*
/dev/sda      /dev/sda1
```

Storage nodes locate under /dev;

2. Slave Device

Connect J13 to PC, open the device manager, and check if the following device is recognized:



其他设备
RNDIS/Ethernet Gadget

2.18 WIFI

2.18.1 Configure WIFI Antennas

After the first boot, the wifi is working at 2.4GHz frequency in default, if you need to use the 5GHz frequency, please configure the WIFI module at first. Here we provide two methods:

1. Enter the path /usr/sbin/wlconf, enter command ./ configure-device.sh

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

Then enter “y   1837   y   2   2” according to the prompt:

```
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. Enter path /usr/sbin/wlconf, enter the command:

```
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
```

You just need to choose one method, then you can use 5G WIFI. This configuration support 2.4G, too. The operation only need to be executed before the first use of WIFI. You don't need to execute again when you open WIFI or boot system again.

2.18.2 Connect WIFI

To connect WIFI, execute the following instructions on the serial terminal:

```
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
```

Note: In above instructions, “embest-test” is the SSID of the WIFI, “12345678” is the password.

Then the serial terminal will show:

```
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=]
```

Test the wifi connection with ping command:

```
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 Test

2.19.1 Reset Bluetooth Module

```
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 Initialize the Bluetooth Module

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

If the initialization success, serial terminal will print the following information:

```
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 Bluetooth Scan Test

```
root@embest:~# hciconfig hci0 up
root@embest:~# hcitool scan
```

serial terminal will print the following information:

```
Scanning ...
        00:12:FE:B7:75:A0       Lenovo-TD80t
```

Chapter 3 System Compilation

3.1 Building Development Environment

Copy the release folder to Linux's $HOME directory, while the compilation tool gcc-linaro-4.9-2015.05-x86_64_arm-linux-gnueabihf under path $HOME/EVK-PH8800-Release-REV01/tool. Use the following instructions to extract it:

```
$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
```

Import the environment variable:

```
$export
CROSS_COMPILE=$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 Compiling U-Boot

3.2.1 Get the U-Boot Source Code

U-boot source code locates under path $HOME/EVK-PH8800-Release-REV01/sourcecode/, extract the u-boot*.tar.gz:

```
$ cd $HOME/EVK-PH8800-Release-REV01/sourcecode/
$ tar -zxvf u-boot*.tar.gz
```

3.2.2 Compile and Burn the Images to SD Card

```
$ cd $HOME/EVK-PH8800-Release-REV01/sourcecode/u-boot*
$ make distclean
$make som_ph8800_defconfig
$make
```

When the compilation finished, it will generate a MLO and u-boot.img under path $HOME/EVK-PH8800-Release-REV01/sourcecode/u-boot, copy the two files to SD card;

3.2.3 Compile and Burn the Images to SPI Flash

```
$ cd $HOME/EVK-PH8800-Release-REV01/sourcecode/u-boot*
$ make distclean
$make som_ph8800_qspiboot_defconfig
$make
```

When the compilation finished, it will generate:

1. **u-boot.bin** under path $HOME/EVK-PH8800-Release-REV01/sourcecode/u-boot*
2. **u-boot-spl.bin** under path $HOME/EVK-PH8800-Release-REV01/sourcecode/u-boot*/spl

Copy the two files to SD card;

Boot from SD card, execute the following instructions in U-Boot phase:

```
U-Boot# run update_qspi_flash
```

Wait for the execute finished, the two files are burn into SPI flash.

(Refer to 1.3 System Boot from SPI Flash)

3.3 Compiling Kernel

3.3.1 Get Kernel Source Code

The source code of the kernel locate under $HOME/EVK-PH8800-Release-REV01/sourcecode/, extract the linux*.tar.gz

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

3.3.2 Compile and Burn the Images to SD Card

```
$ cd $HOME/EVK-PH8800-Release-REV01/sourcecode/linux*
$ make distclean
$ make embest_ti_8800_defconfig
$ make
```

When the compilation finished, it will generate

- zImage under $HOME/EVK-PH8800-Release-REV01/sourcecode/linux*/arch/arm/boot;
- the following 3 files under $HOME/EVK-PH8800-Release-REV01/sourcecode/linux*/arch/arm/boot/dts

1. embest-SOM_PH8800-BB_EPH1800-4.3inch_LCD.dtb
2. embest-SOM_PH8800-BB_EPH1800-7inch_LCD.dtb
3. embest-SOM_PH8800-BB_EPH1800-HDMI-VGA.dtb

The dtb files are corresponding for 4.3” LCD, 7” LCD and HDMI display. (Refer to 2.9 HDMI/VGA and 2.10 LCD)

Copy the files to SD Card.