QY-IMX6S Linux Function And Test Manual

Version NO.： 1.0
2017.4

QIYANG INTELLIGENT TECHNOLOGY CO., LTD

Copyright Reserved

Catalogue

Version Illustration:

Version	Hardware Platform	Description	Date	Revisor
1.0	QY-IMX6S-V1.2	Initial Published	2017-04-25	Hech

Preface

This Manual mainly introduce different interface function and testing method.

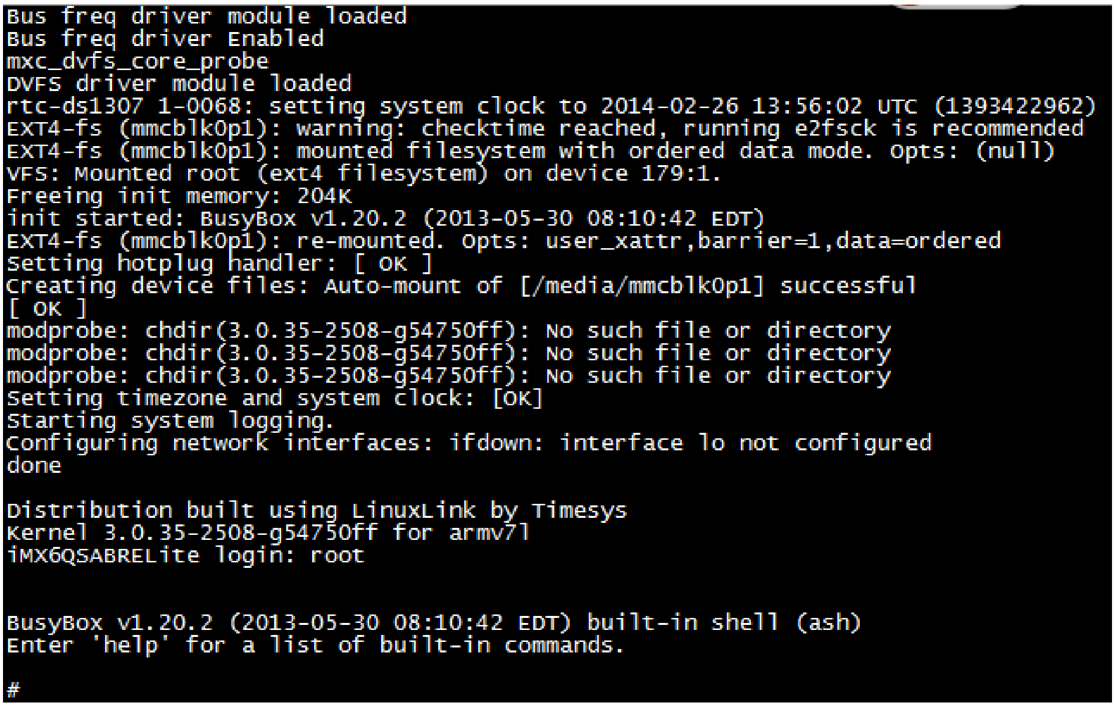
Please read carefully before using:

QY-IMX6S Hardware Manual .pdf & QY-IMX6S Linux User Manual. pdf

Ⅰ. Preparation

Before testing, please read ***QY-IMX6S Linux User Manual. pdf,*** and connect to the board according to this manual.

Power on mainboard, after system starts, then input root user to enter into file system of the board, as the following picture shown:

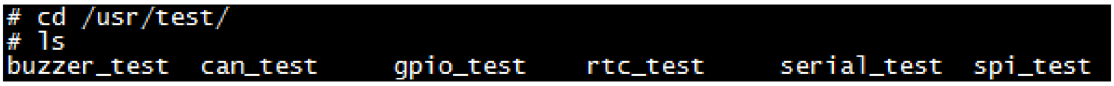


```
Bus freq driver module loaded
Bus freq driver Enabled
mxc_dvfs_core_probe
DVFS driver module loaded
rtc-ds1307 1-0068: setting system clock to 2014-02-26 13:56:02 UTC (1393422962)
EXT4-fs (mmcblk0p1): warning: checktime reached, running e2fsck is recommended
EXT4-fs (mmcblk0p1): mounted filesystem with ordered data mode. Opts: (null)
VFS: Mounted root (ext4 filesystem) on device 179:1.
Freeing init memory: 204K
init started: BusyBox v1.20.2 (2013-05-30 08:10:42 EDT)
EXT4-fs (mmcblk0p1): re-mounted. Opts: user_xattr,barrier=1,data=ordered
Setting hotplug handler: [ OK ]
Creating device files: Auto-mount of [/media/mmcblk0p1] successful
[ OK ]
modprobe: chdir(3.0.35-2508-g54750ff): No such file or directory
modprobe: chdir(3.0.35-2508-g54750ff): No such file or directory
modprobe: chdir(3.0.35-2508-g54750ff): No such file or directory
Setting timezone and system clock: [OK]
Starting system logging.
Configuring network interfaces: ifdown: interface lo not configured
done

Distribution built using LinuxLink by Timesys
Kernel 3.0.35-2508-g54750ff for armv7l
iMX6QSABRELite login: root


BusyBox v1.20.2 (2013-05-30 08:10:42 EDT) built-in shell (ash)
Enter 'help' for a list of built-in commands.

#
```

Mainboard test program in [/usr/test] directory, please switch to this directory, the following testing operations will be done in this directory.

```
cd /usr/test/
```

```
ls
```



```
# cd /usr/test/
# ls
buzzer_test   can_test      gpio_test     rtc_test      serial_test  spi_test
```

Ⅱ. Mainboard Test

2.1 Buzzer Test

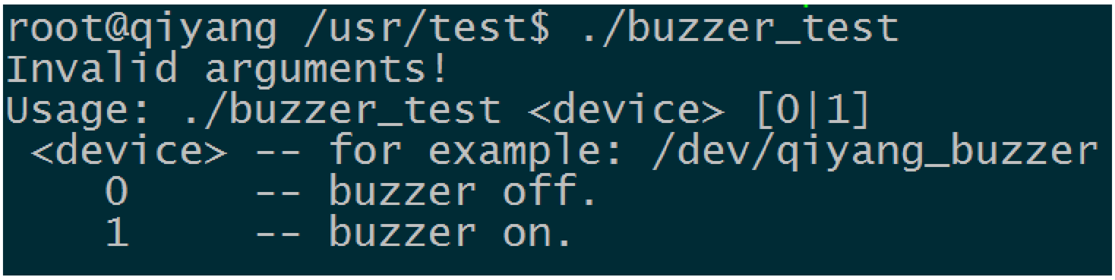
QY-IMX6S mainboard use [GPIO 6_31] to control the buzzer on the board. When set to low level, buzzer does not work; when set to high level, buzzer will work.

Test Principle:

This test is to realize buzzing.

Test Process and Result:

Run buzzer testing program [buzzer_test]



```
root@qiyang /usr/test$ ./buzzer_test
Invalid arguments!
Usage: ./buzzer_test <device> [0|1]
 <device> -- for example: /dev/qiyang_buzzer
    0     -- buzzer off.
    1     -- buzzer on.
```

Illustration: [buzzer_test <device>0] buzzer does not work, buzzer can be closed.

[buzzer_test <device> 1] buzzer does work.

1. Open buzzer, the mainboard is buzzing continuously, press [Ctrl+C] to exit the program.

```
/buzzer_test /dev/qiyang_buzzer 1
```

2. Close buzzer, press [ctrl+c] to exit the program.

```
/buzzer_test /dev/qiyang_buzzer 0
```

Device Node:

```
/dev/qiyang_buzzer
```

Test Code:

```
CD/Test Code/buzzer_test/buzzer_test.c
```

Driver Code:

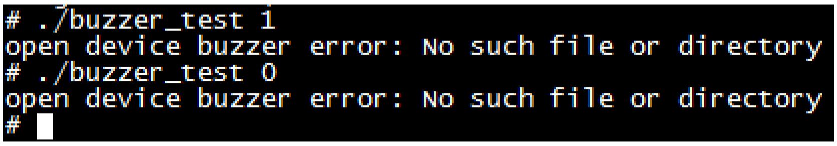
```
linux-3.0.101/drivers/misc/buzzer.c
```

The Kernel Options:

```
Device Drivers --->
        Misc devices --->
                <*>     BUZZER FOR QIYANG IMX6 BOARD
```

FAQ:

Debug UART prints information:



```
# ./buzzer_test 1
open device buzzer error: No such file or directory
# ./buzzer_test 0
open device buzzer error: No such file or directory
#
```

Please check the following items:

① ,Whether [/dev] directory has [qiyang_buzzer] device node or not.

② Kernel configuration select <*> BUZZER FOR QIYANG IMX6 BOARD

③ ,Device tree file enable [qiyang_buzzer]node

2.2 RTC Test

QY-IMX6S mainboard adopts I2C2 connect DS1338 Chip on the base board as external hardware clock. Please confirm that you put on the battery before testing RTC.

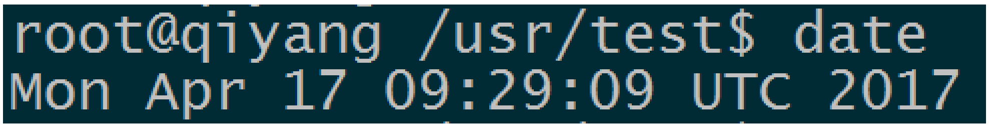
Test Principle:

Set system time through [date] system command, and then write system time into hardware clock through [hwclock] command. Through [rtc_test]program to read hardware timer and print it. After powering off, restart to check whether the clock is accurate.

Test Process & Result:

1.Execute [date] command on the board and check the current system clock.

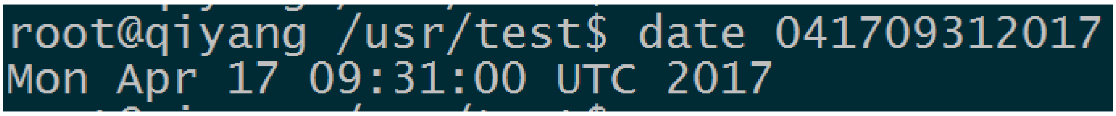
```
date
```



```
root@qiyang /usr/test$ date
Mon Apr 17 09:29:09 UTC 2017
```

2.Set system clock through [date] command, for example, to set on current PC display time.

```
date 022710412014      /*month day hour minute year*/
```

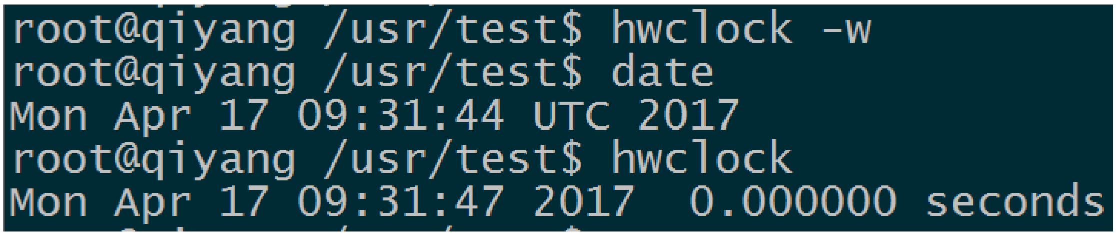


```
root@qiyang /usr/test$ date 041709312017
Mon Apr 17 09:31:00 UTC 2017
```

3.Use [hwclcok] command to write system time into hardware time chip.

```
[ hwclock –w]
```

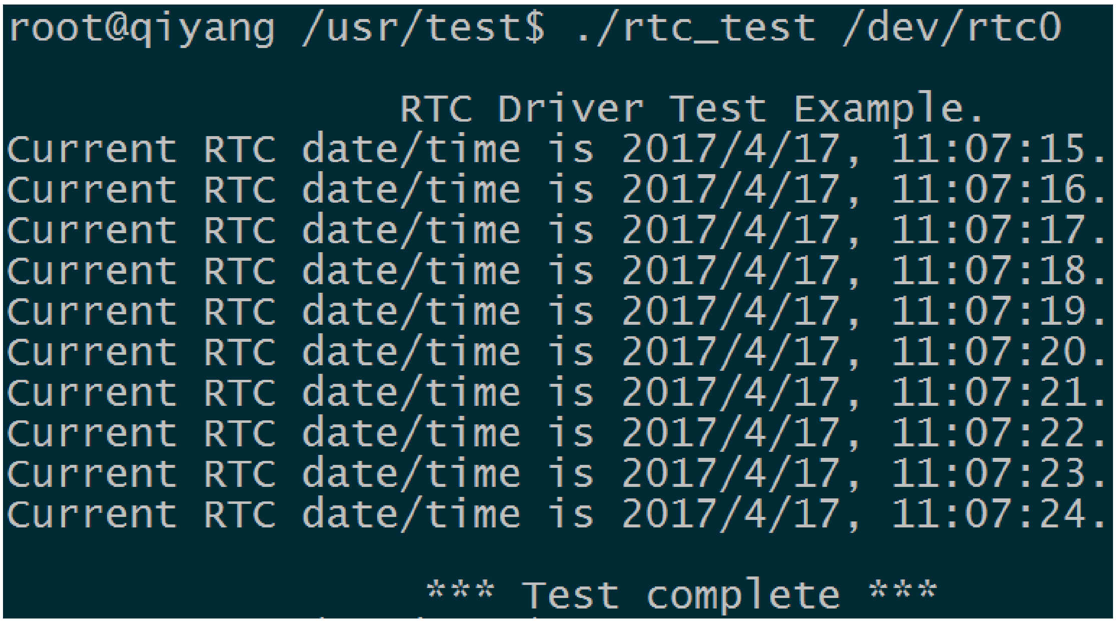
4.Check system and hardware clock by [date] command and [hwclock]command.



```
root@qiyang /usr/test$ hwclock -w
root@qiyang /usr/test$ date
Mon Apr 17 09:31:44 UTC 2017
root@qiyang /usr/test$ hwclock
Mon Apr 17 09:31:47 2017  0.000000 seconds
```

5.After setting successfully, execute [rtc_test] test program.

```
./rtc_test/dev/rtc0
```

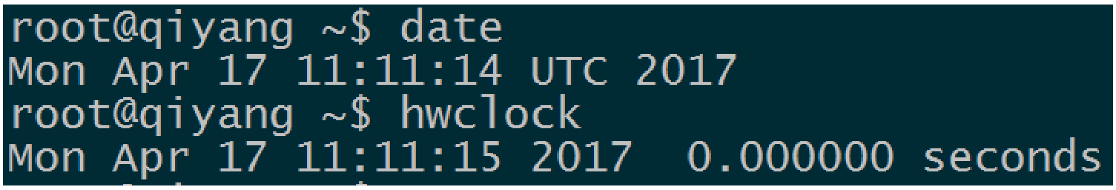


```
root@qiyang /usr/test$ ./rtc_test /dev/rtc0

                RTC Driver Test Example.
Current RTC date/time is 2017/4/17, 11:07:15.
Current RTC date/time is 2017/4/17, 11:07:16.
Current RTC date/time is 2017/4/17, 11:07:17.
Current RTC date/time is 2017/4/17, 11:07:18.
Current RTC date/time is 2017/4/17, 11:07:19.
Current RTC date/time is 2017/4/17, 11:07:20.
Current RTC date/time is 2017/4/17, 11:07:21.
Current RTC date/time is 2017/4/17, 11:07:22.
Current RTC date/time is 2017/4/17, 11:07:23.
Current RTC date/time is 2017/4/17, 11:07:24.

                 *** Test complete ***
```

After the program prints 10 of the RTC time, then exit the program. Or you can use [Ctrl+c] to exit the program early.

RTC works accurately, no losing seconds.

6.Powering off, then power on to check system and hardware clock by using [date] and [hwclock] command, to check time is saved or not and whether works normally.



```
root@qiyang ~$ date
Mon Apr 17 11:11:14 UTC 2017
root@qiyang ~$ hwclock
Mon Apr 17 11:11:15 2017  0.000000 seconds
```

7.After comparing with PC time, there is no error. If need to test long time work’s accuracy, you can separate power off and power on aging test for several days or weeks or months to test the time error.

Before published, our board has been tested for aging test for 1 month, and the time error does not exceed 2S.

As to our delivery goods, we will make it for aging test for more than 24 hours, and time error does not exceed 1S.

Device Node:

/dev/rtc

/dev/rtc0

Test Code：

```
CD/Source Code/Test Code/ rtc_test/rtc_test.c
```

Driver Code：

```
ds1338 Driver：linux-3.0.101/drivers/rtc/rtc-ds1307.c
```

```
i2c Driver：linux-3.0.101/drivers/i2c/busses/i2c-imx.c
```

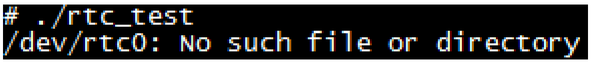
The Kernel Options：

```
    I2c2:
    Device Drivers --->
          I2C support    --->
                [*]      Enable compatibility bits for old user-space
                < *>      I2C device interface
                < >     I2C bus multiplexing support
                [*]      Autoselect pertinent helper modules
                          I2C Hardware Bus support    --->
                [ ]     I2C Core debugging messages
                [ ]     I2C Algorithm debugging messages
                [ ]     I2C Bus debugging messages
    rtc:
Device Drivers --->
          [*] Real Time Clock    --->
               [*]      Set system time from RTC on startup and resume
  (rtc0)    RTC used to set the system time
                [ ]     RTC debug support
                *** RTC interfaces ***
                [*]      /sys/class/rtc/rtcN (sysfs)
                [*]      /proc/driver/rtc (procfs for rtc0)
                [*]     /dev/rtcN (character devices)
                [ ]       RTC UIE emulation on dev interface
               <*>     Dallas/Maxim DS1307/37/38/39/40, ST M41T00, EPSON RX-8025
```

FAQ:

Phenomenon: Time can not be saved, time travel error is huge, can not check hardware clock

Debug UART print the following information:



```
# ./rtc_test
/dev/rtc0: No such file or directory
```

Please check the following items:

① Whether BT1 on the base board ,and supports electricity ?

② [/dev/rtc0] and [/dev/rtc1] node in [/dev]directory ?

③.The kernel configuration has been configurated ?

2.3. Watchdog Test

Test Principle:

Hardware watchdog, [GPIO_3_23] enables watchdog, [GPIO_3_22] executes ‘feeding dog’ operations.

Test Process & Result:

1.Boot Watchdog, and execute ‘feeding dog’ operations. You can see the system will not reboot. Press [Ctrl+C] to exit the program.

```
./watchdog_feed_test /dev/qy_watchdog
```

2.Boot Watchdog, but do not execute ‘feeding dog’ operations, the system reboots after1.6 seconds.

```
./watchdog_notfeed_test /dev/qy_watchdog
```

Device Node:

```
/dev/qy_watchdog
```

Test Code:

```
CD/Source Code /Test Code/watchdog_test/watchdog_feed_test.c
```

```
CD/Source Code /Test Code/watchdog_test/watchdog_notfeed_test.c
```

Driver Code:

```
linux-3.0.101/drivers/misc/watchdog.c
```

The kernel Options:

Device Drivers --->

```
[*] Misc devices --->
    [*]  WATCH_DOG FOR QIYANG BOARD
```

FAQ:

1. The system did not reboot, after executing [./watchdog_notfeed_test]. Please check the watchdog’s chipset if it is damaged.

2.4 GPIO Test

This test focus on the following 16-ch GPIO pin definition:

J5[GPIO2_0,GPIO2_1, ,GPIO2-2,GPIO2-3,GPIO2-4,GPIO2_5,GPIO2_6,GPIO2_7].

J31[GPIO2_16,GPIO2_17,GPIO2_18,GPIO2_19,GPIO2_20,GPIO2_21,GPIO2_22,GPIO2_23].

Test Principle:

[Gpio_test 0] test the situation when gpio does not have external connection, set all pin to low level or high level, through external measure gpio actual level to confirm gpio is normal or not.

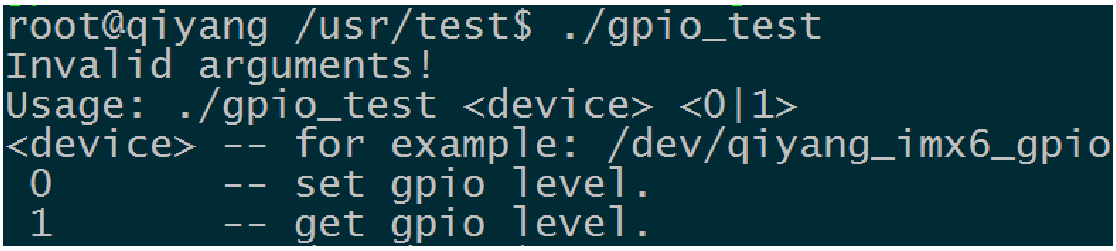
[Gpio_test 1] will read external level signal directly, user can compare read level data and connected level data to confirm gpio is normal.

Test Process & Result:

Run gpio test program [gpio_test]

```
./gpio_test
```

It hints the below information:



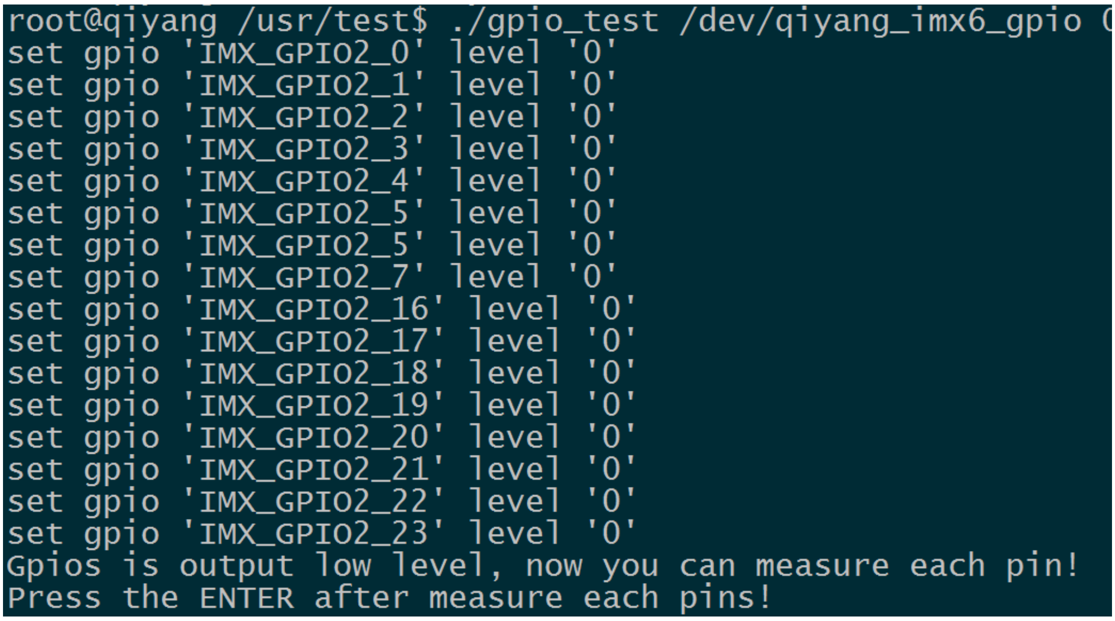
```
root@qiyang /usr/test$ ./gpio_test
Invalid arguments!
Usage: ./gpio_test <device> <0|1>
<device> -- for example: /dev/qiyang_imx6_gpio
 0        -- set gpio level.
 1        -- get gpio level.
```

Illustration: [gpio_test <device> 0] set gpio high level and low level

[gpio_test <device> 1] obtain gpio level

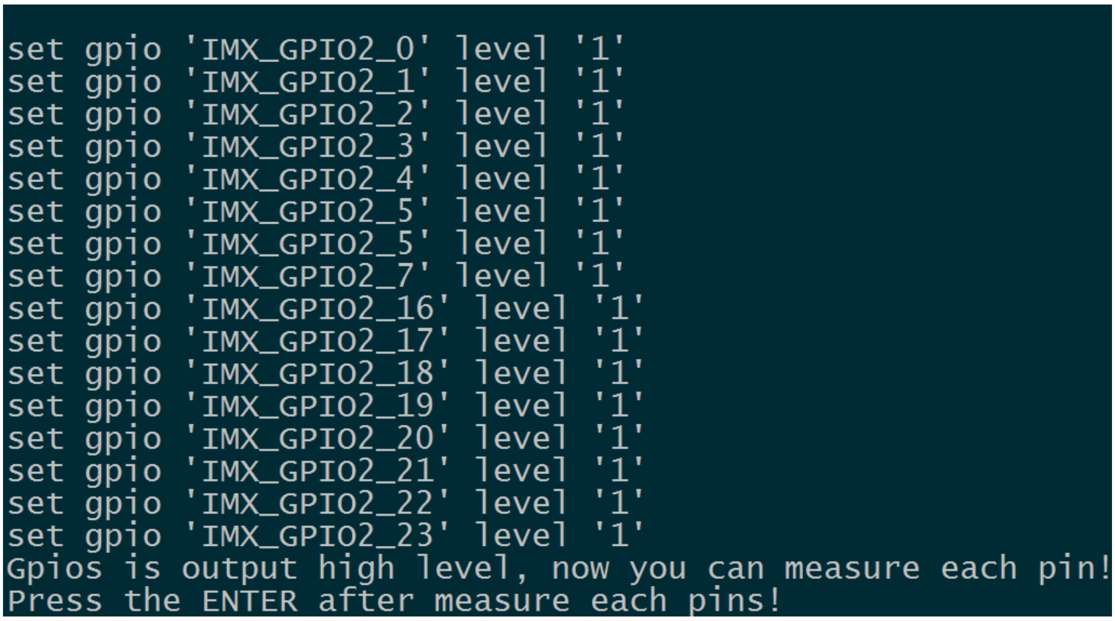
2. J5 and J31 GPIO is without external signal, execute

   ./gpio_test /dev/qiyang_imx6_gpio 0



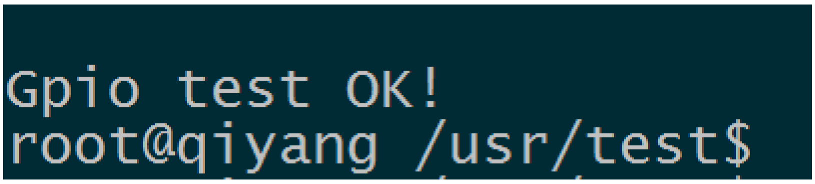
```
root@qiyang /usr/test$ ./gpio_test /dev/qiyang_imx6_gpio
set gpio 'IMX_GPIO2_0' level '0'
set gpio 'IMX_GPIO2_1' level '0'
set gpio 'IMX_GPIO2_2' level '0'
set gpio 'IMX_GPIO2_3' level '0'
set gpio 'IMX_GPIO2_4' level '0'
set gpio 'IMX_GPIO2_5' level '0'
set gpio 'IMX_GPIO2_5' level '0'
set gpio 'IMX_GPIO2_7' level '0'
set gpio 'IMX_GPIO2_16' level '0'
set gpio 'IMX_GPIO2_17' level '0'
set gpio 'IMX_GPIO2_18' level '0'
set gpio 'IMX_GPIO2_19' level '0'
set gpio 'IMX_GPIO2_20' level '0'
set gpio 'IMX_GPIO2_21' level '0'
set gpio 'IMX_GPIO2_22' level '0'
set gpio 'IMX_GPIO2_23' level '0'
Gpios is output low level, now you can measure each pin!
Press the ENTER after measure each pins!
```

As the above picture shown, set each [GPIO] to low level. Use multimeter to measure corresponding GPIO’s actual level value to confirm GPIO is normal or not. Then press [Enter] to set all GPIO to high level, to measure corresponding GPIO’s actual level value to confirm GPIO is normal or not.



```
set gpio 'IMX_GPIO2_0' level '1'
set gpio 'IMX_GPIO2_1' level '1'
set gpio 'IMX_GPIO2_2' level '1'
set gpio 'IMX_GPIO2_3' level '1'
set gpio 'IMX_GPIO2_4' level '1'
set gpio 'IMX_GPIO2_5' level '1'
set gpio 'IMX_GPIO2_5' level '1'
set gpio 'IMX_GPIO2_7' level '1'
set gpio 'IMX_GPIO2_16' level '1'
set gpio 'IMX_GPIO2_17' level '1'
set gpio 'IMX_GPIO2_18' level '1'
set gpio 'IMX_GPIO2_19' level '1'
set gpio 'IMX_GPIO2_20' level '1'
set gpio 'IMX_GPIO2_21' level '1'
set gpio 'IMX_GPIO2_22' level '1'
set gpio 'IMX_GPIO2_23' level '1'
Gpios is output high level, now you can measure each pin!
Press the ENTER after measure each pins!
```

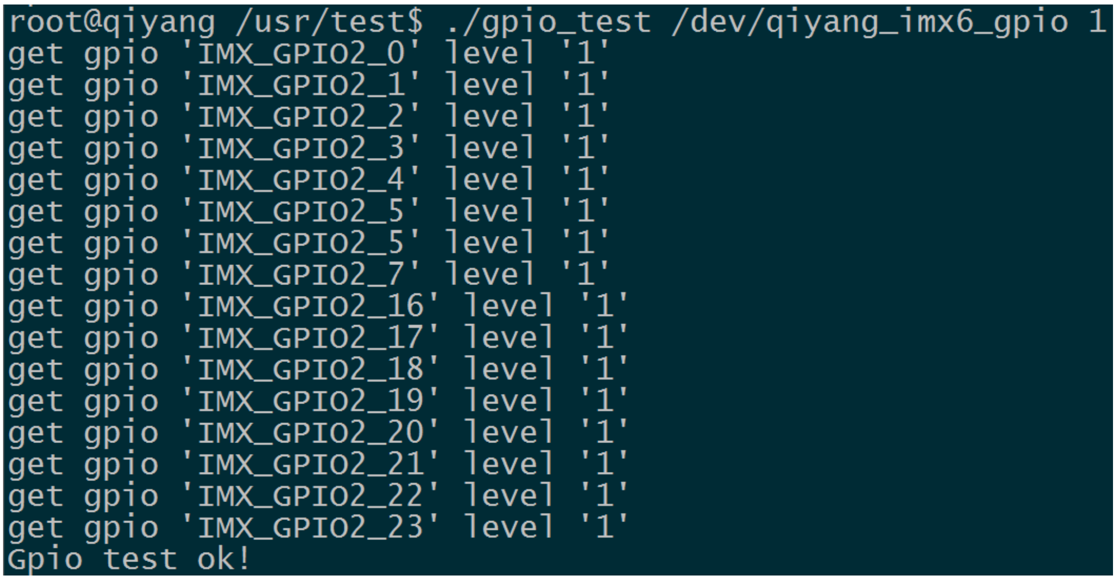
Then press [Enter], it hints test [OK].



```
Gpio test OK!
root@qiyang /usr/test$
```

3.External connect 3.3V to the pin (Internal Pin is weak pull-up)

```
./gpio_test /dev/qiyang_imx6_gpio 1
```



```
root@qiyang /usr/test$ ./gpio_test /dev/qiyang_imx6_gpio 1
get gpio 'IMX_GPIO2_0' level '1'
get gpio 'IMX_GPIO2_1' level '1'
get gpio 'IMX_GPIO2_2' level '1'
get gpio 'IMX_GPIO2_3' level '1'
get gpio 'IMX_GPIO2_4' level '1'
get gpio 'IMX_GPIO2_5' level '1'
get gpio 'IMX_GPIO2_5' level '1'
get gpio 'IMX_GPIO2_7' level '1'
get gpio 'IMX_GPIO2_16' level '1'
get gpio 'IMX_GPIO2_17' level '1'
get gpio 'IMX_GPIO2_18' level '1'
get gpio 'IMX_GPIO2_19' level '1'
get gpio 'IMX_GPIO2_20' level '1'
get gpio 'IMX_GPIO2_21' level '1'
get gpio 'IMX_GPIO2_22' level '1'
get gpio 'IMX_GPIO2_23' level '1'
Gpio test ok!
```

As above picture shown, obtaining each gpio level status, users can change the actual connected gpio signal to confirm gpio is normal or not.

Device Node:

/dev/qiyang_imx6_gpio

Test Code:

CD/Test Code/gpio_test/gpio_test.c

Driver Code:

linux-3.0.101/drivers/misc/qy_imx6_gpio.c

The Kernel Option:

```
Device Drivers --->
          Misc devices --->
                  <*> IMX6 GPIO TEST FOR QIYANG BOARD
```

FAQ:

Debug UART shows:

QY-IMX6S-v1.x Gpio Start Testing...

Please check the following items:

①Whether there is [/dev/qiyang_imx6_gpio node] in [/dev directory]

②Whether the kernel configuration select [<*> IMX6 GPIO TEST FOR QIYANG BOARD].

2.5. Serial Port Test

There are 5-ch serial ports: J6 is as the debug UART. The other 4-ch could be used as the RS232 serial port.

COM1(J31) and COM2(J31) are 5 wire serial ports and multiplexes with RS485 interface.

COM3(J2) and COM4(J2) is 3 wire serial ports.

This test program only aims at testing of common RS232.

The relations between serial port and hardware:

Serial Port	Hardware Location	Device Node
DBG（Debug UART）	J6（Rx、Tx、GND TO Pin 2、Pin1、Pin3）	/dev/ttymxc0
COM1	J31（Rx、Tx、GND TO Pin11、Pin13、Pin9）	/dev/ttymxc1
COM2	J31（Rx、Tx、GND TO Pin12、Pin14、Pin10）	/dev/ttymxc2
COM3	J2（Rx、Tx、GND TO Pin1、Pin3、Pin5）	/dev/ttymxc3

COM4	J2（Rx、Tx、GND TO Pin2、Pin4、Pin6）	/dev/ttymxc4

Test Principle:

Test program realizes that 1 serial port send character data "/dev/ttymxcx" test string!" every other 1s, X is actual testing device node, through multithreading way to block reading serial ports data and then print.

Test Process & Result:

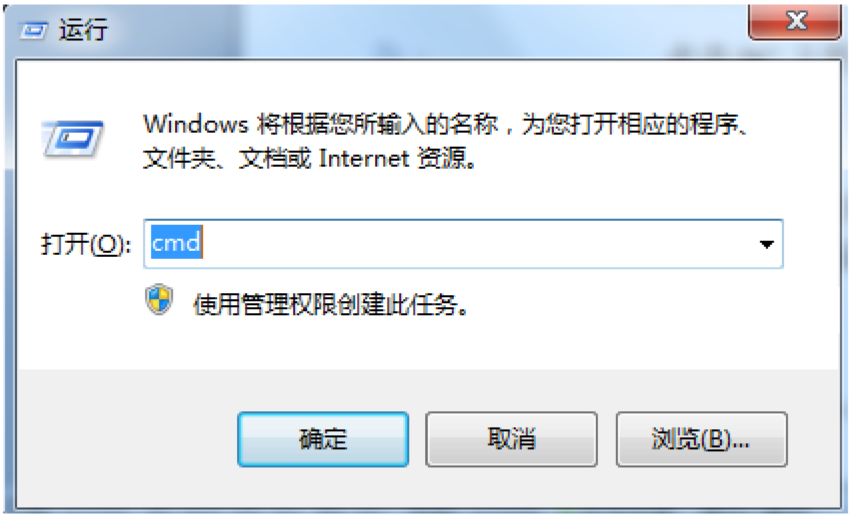
When do serial ports testing, PC needs 2 serial ports

② .One connect to debugging port for interaction

③ .One connect to under test ports to receive and transmit data

If there is only 1 serial port, port connect to under test ports , connect development board by network cable. Through [telnet] function to log in development system as debugging ports operation.

Development board defaulted IP address is 192.168.1.71. Click start button on PC, select [RUN], then type [cmd] and click confirm.



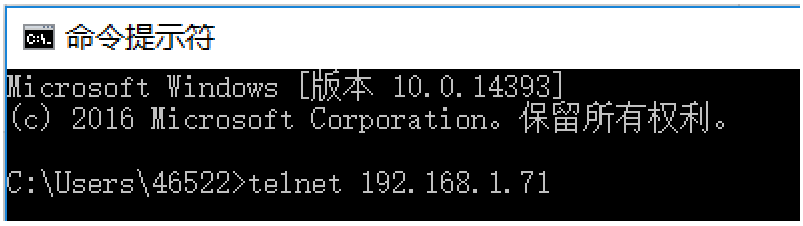
运行

Windows 将根据您所输入的名称，为您打开相应的程序、文件夹、文档或 Internet 资源。

打开(O): cmd

使用管理权限创建此任务。

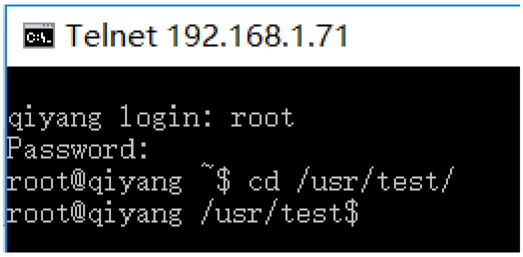
Input [telent 192.168.1.71] in running Windows.



命令提示符

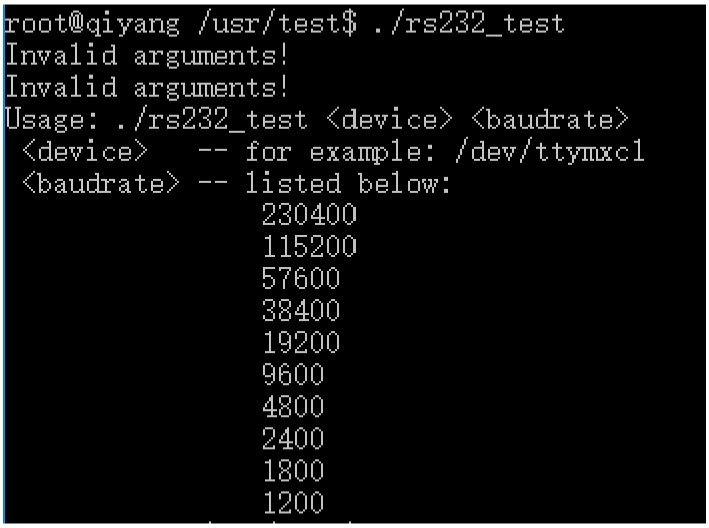
```
Microsoft Windows [版本 10.0.14393]
(c) 2016 Microsoft Corporation。保留所有权利。

C:\Users\46522>telnet 192.168.1.71
```

Input username [root] and password , users can access console, enter [/usr/test] test directory.



Telnet 192.168.1.71

```
qiyang login: root
Password:
root@qiyang ~$ cd /usr/test/
root@qiyang /usr/test$
```

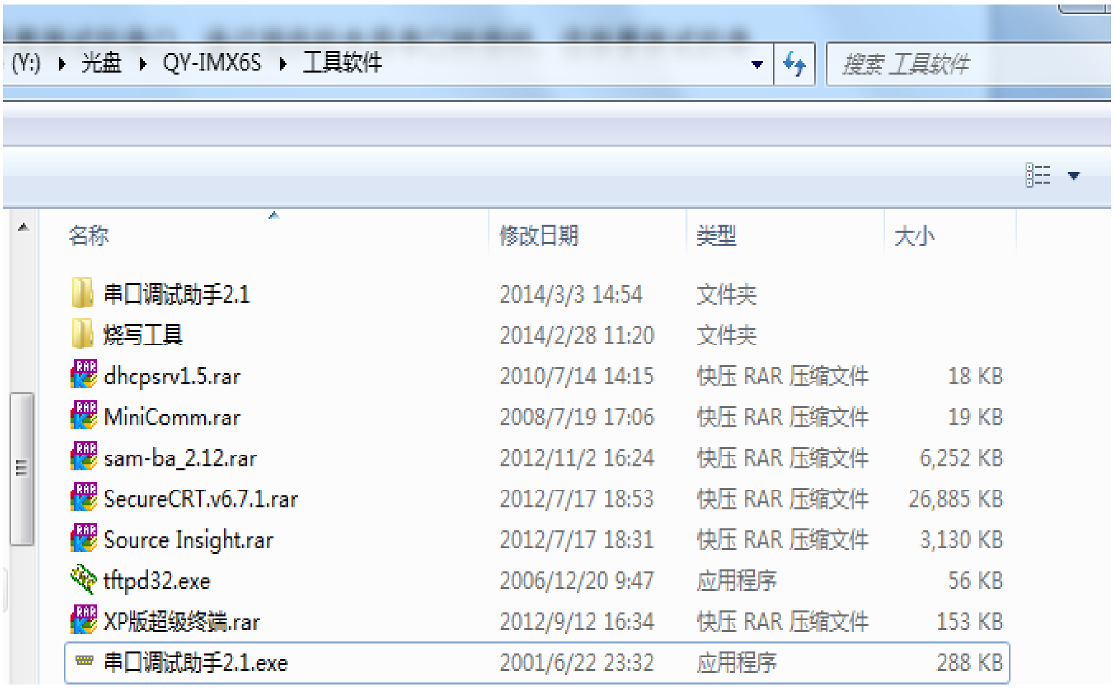


```
root@qiyang /usr/test$ ./rs232_test
Invalid arguments!
Invalid arguments!
Usage: ./rs232_test <device> <baudrate>
 <device>   -- for example: /dev/ttymxc1
 <baudrate> -- listed below:
                230400
                115200
                57600
                38400
                19200
                9600
                4800
                2400
                1800
                1200
```

According to the relation tablet between serial port and hardware, select the serial port which need to be tested.

Connect the serial port to UART on PC through special UART cable.

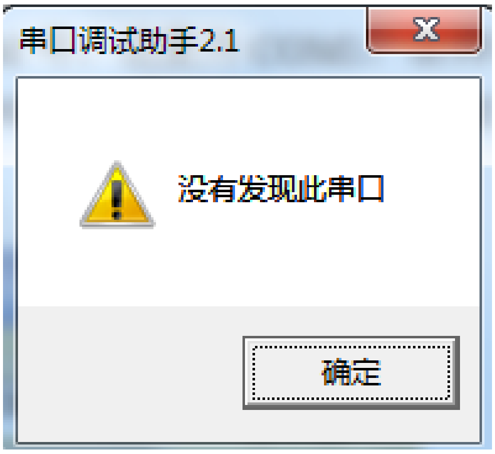
Open the CD/serial debugging tool in PC.



(V:) ▸ 光盘 ▸ QY-IMX6S ▸ 工具软件

名称	修改日期	类型	大小
串口调试助手2.1	2014/3/3 14:54	文件夹	
烧写工具	2014/2/28 11:20	文件夹	
dhcpsrv1.5.rar	2010/7/14 14:15	快压 RAR 压缩文件	18 KB
MiniComm.rar	2008/7/19 17:06	快压 RAR 压缩文件	19 KB
sam-ba_2.12.rar	2012/11/2 16:24	快压 RAR 压缩文件	6,252 KB
SecureCRT.v6.7.1.rar	2012/7/17 18:53	快压 RAR 压缩文件	26,885 KB
Source Insight.rar	2012/7/17 18:31	快压 RAR 压缩文件	3,130 KB
tftpd32.exe	2006/12/20 9:47	应用程序	56 KB
XP版超级终端.rar	2012/9/12 16:34	快压 RAR 压缩文件	153 KB
串口调试助手2.1.exe	2001/6/22 23:32	应用程序	288 KB

If it hints: No such serial port

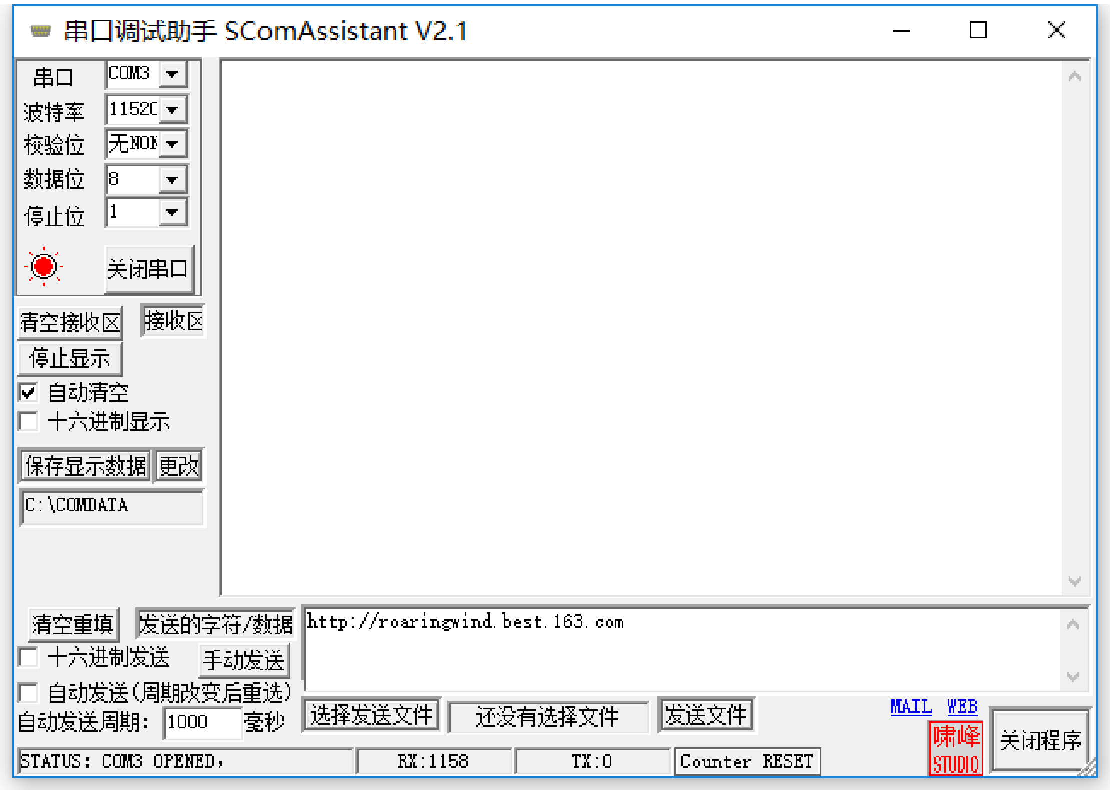


串口调试助手2.1

没有发现此串口

It means the COM on PC be occupied by terminal, close the occupied terminal, then use serial debugging tool.

Set UART attribute, serial ports correspond to COM number on PC, here it is COM3, Baud Rate is [115200], Data Bit [8-bit], Stop Bit [1], Parity Bit [NONE].



串口调试助手 SComAssistant V2.1

串口 COM3
波特率 1152C
校验位 无NOI
数据位 8
停止位 1

☑ 自动清空
□ 十六进制显示

C:\COMDATA

http://roaringwind.best.163.com

□ 十六进制发送
□ 自动发送(周期改变后重选)
自动发送周期: 1000 毫秒

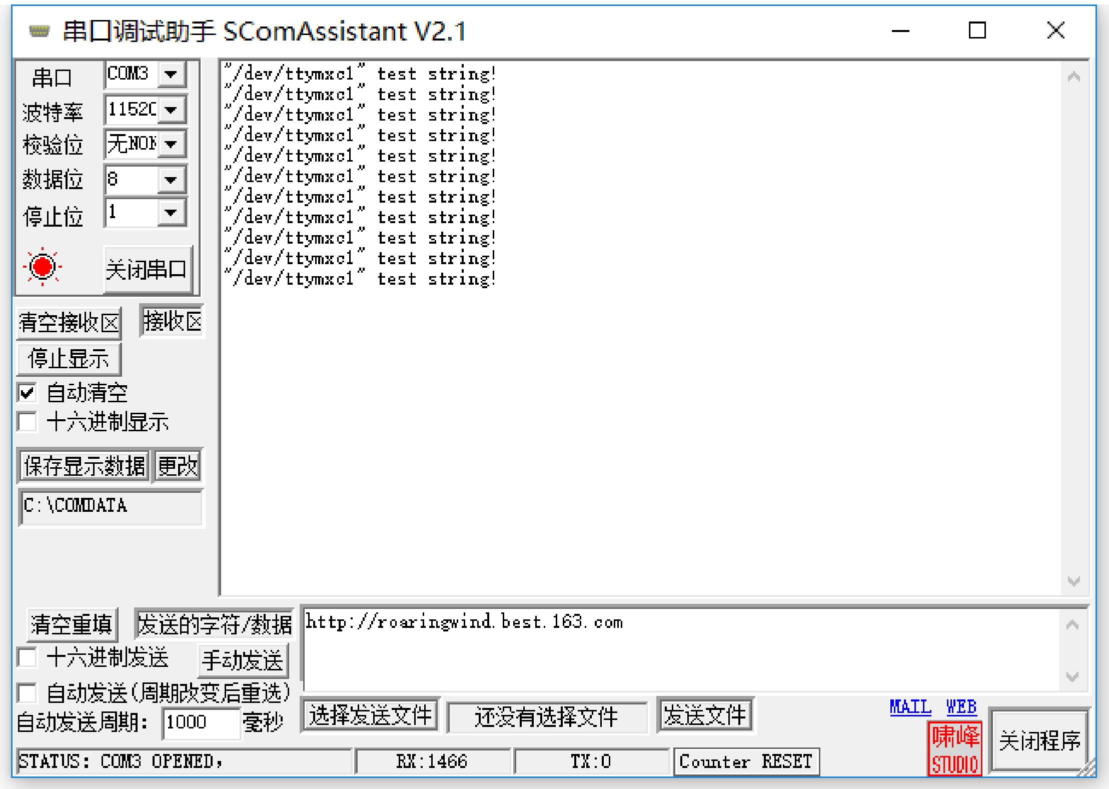
STATUS: COM3 OPENED, RX:1158 TX:0

After connected and set the UART, start to test.

Testing COM1,COM2,COM3,COM4, here ,we take example on COM1. The other UARTs testing method are same.

```
./rs232_test /dev/ttymxc1 115200
```

After running, the serial debugging assistant shows the received date.



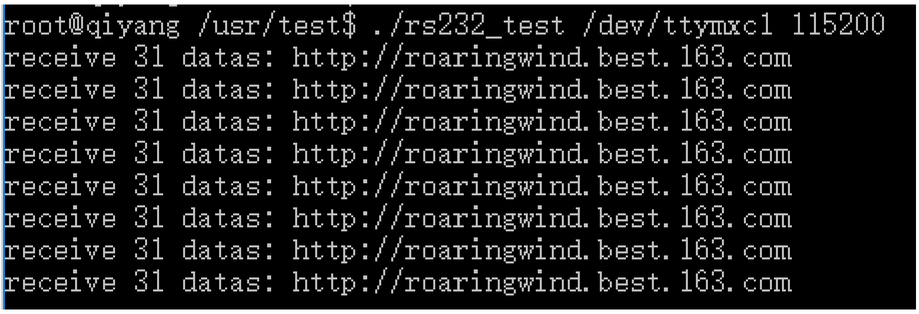
串口 COM3
波特率 11520
校验位 无NON
数据位 8
停止位 1
☑ 自动清空
□ 十六进制显示
C:\COMDATA
□ 十六进制发送
□ 自动发送(周期改变后重选)
自动发送周期: 1000 毫秒

```
"/dev/ttymxc1" test string!
"/dev/ttymxc1" test string!
"/dev/ttymxc1" test string!
"/dev/ttymxc1" test string!
"/dev/ttymxc1" test string!
"/dev/ttymxc1" test string!
"/dev/ttymxc1" test string!
"/dev/ttymxc1" test string!
"/dev/ttymxc1" test string!
"/dev/ttymxc1" test string!
"/dev/ttymxc1" test string!
```

http://roaringwind.best.163.com

STATUS: COM3 OPENED, RX:1466 TX:0

Click [Send Manually] on serial debugging assistant, [telnet] shows the received date.



```
root@qiyang /usr/test$ ./rs232_test /dev/ttymxc1 115200
receive 31 datas: http://roaringwind.best.163.com
receive 31 datas: http://roaringwind.best.163.com
receive 31 datas: http://roaringwind.best.163.com
receive 31 datas: http://roaringwind.best.163.com
receive 31 datas: http://roaringwind.best.163.com
receive 31 datas: http://roaringwind.best.163.com
receive 31 datas: http://roaringwind.best.163.com
receive 31 datas: http://roaringwind.best.163.com
```

Telnet and serial debugging assistant both can receive data and no errors. It means the UART function is normal.

After finishing test on COM1, use [Ctrl+C] to exit the program, continue to test other UARTs.

Device Node:

/dev/ttymxc1

/dev/ttymxc2

/dev/ttymxc3

/dev/ttymxc4

Test Code:

/CD/Test Code/serial _test/rs232_test.c

Driver Code:

linux-3.0.101/drivers/tty/serial/imx. c

The kernel configurations:

```
Device Drivers --->
            Character devices    --->
                    Serial drivers    --->
                            [*] IMX serial port support
```

FAQ:

The debug UART prints following information：

open serial device /dev/ttymxc1 error!

If ports communication is abnormal or can not communicate, please check the following items:

① . Connecting wire is normal or not ?

②. PC port connect to ports debugging software configuration is right or not ?

② . Serial ports hardware and testing program node is corresponded or not ?

③ . Whether [ttymxc1,ttymxc2,ttymxc3,ttymxc4] node be set correctly in [/dev] directory ?

⑤. The kernel configuration is selected or not ?

2.6.SPI Test

Mainboard extracts SPI interface (J5), total 3-ch chip selects, can be externally connected to 3-ch SPI device.

3-channel SPI all use SPI common driver [spidev], driver realize the basic operation of device's data read and write. When using, write application program according to external device's time sequence.

Test Principle:

Because of no external SPI peripheral connection, test program only writes data testing to SPI interface. Test program will send a group data every other 1 second. If short connect the spi receiving pin and transmitting pin, interrupt in debugging ports, it will print this group data. It also can be measured on the wave through oscilloscope. If relating to spi half duplex write operation and full duplex write operation, you can refer to this source code.

Relation Chart:

Chipsets are corresponding to the device node as below:

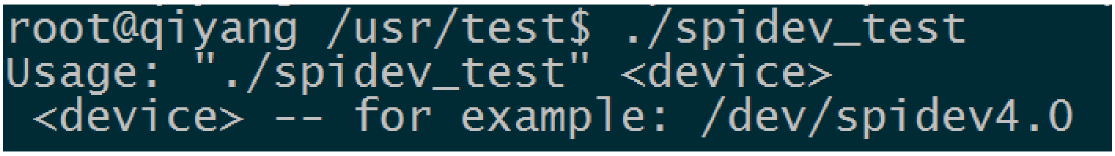
Chipset	Device Node	Hardware Location
NCS0	/dev/spidev4.0	Pin 6 on J5
NCS1	/dev/spidev4.1	Pin 7 on J5
NCS2	/dev/spidev4.2	Pin 8 on J5

Chip selects corresponded to device node

Test Process & Result:

1.Run test program [spidev_test]

```
./spidev_test
```



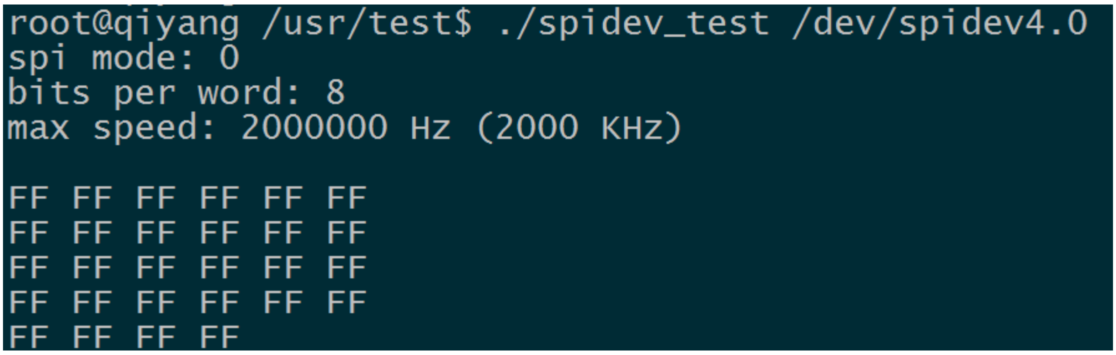
```
root@qiyang /usr/test$ ./spidev_test
Usage: "./spidev_test" <device>
 <device> -- for example: /dev/spidev4.0
```

Illustration: [spidev_test /dev/spidev4.0] tests NCS0 chip select channel

[spidev_test /dev/spidev4.1] tests NCS1 chip select channel [spidev_test /dev/spidev4.2] tests NCS2 chip select channel

2.Here, we take example of NCS0 chip select as an example to introduce, other chips select's test method is same.

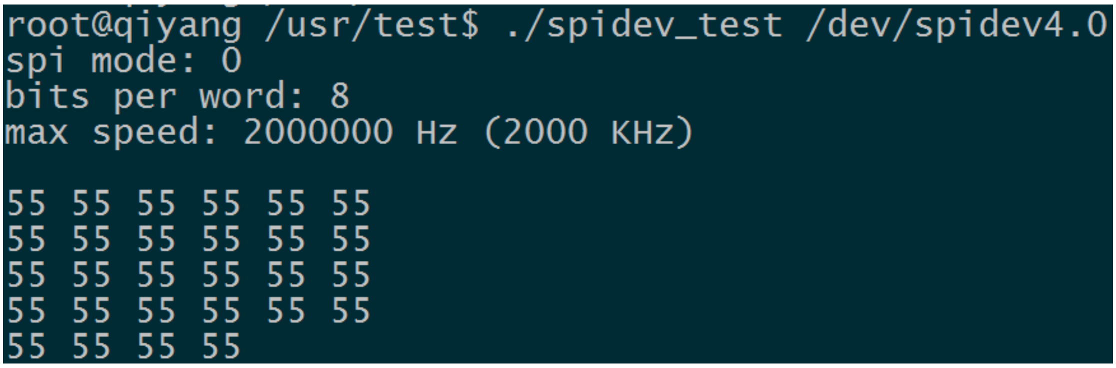
/spidev_test /dev/spidev4.0



```
root@qiyang /usr/test$ ./spidev_test /dev/spidev4.0
spi mode: 0
bits per word: 8
max speed: 2000000 Hz (2000 KHz)

FF FF FF FF FF FF
FF FF FF FF FF FF
FF FF FF FF FF FF
FF FF FF FF FF FF
FF FF FF FF
```

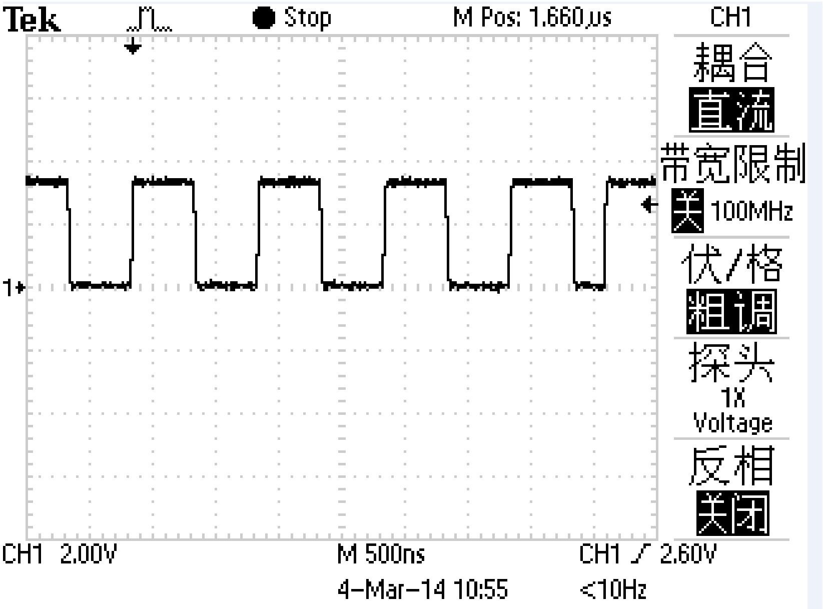
3. Short connect [SPI5_MOSI (Pin4 on J5)] and [SPI5_MISO (Pin3 on J5)]

./spidev_test/dev/spidev4.0



```
root@qiyang /usr/test$ ./spidev_test /dev/spidev4.0
spi mode: 0
bits per word: 8
max speed: 2000000 Hz (2000 KHz)

55 55 55 55 55 55
55 55 55 55 55 55
55 55 55 55 55 55
55 55 55 55 55 55
55 55 55 55
```

Use oscilloscope to check wave on [SPI5_MOSI].



Device Node:

SPI5_NCS0: /dev/spidev4.0

SPI5_NCS1: /dev/spidev4.1

SPI5_NCS2: /dev/spidev4.2

Test Code:

CD/Test Source Code/ spi _test/spidev_test.c

Driver Code:

linux-3.0.101/drivers/spi/spidev.c

The kernel Options:

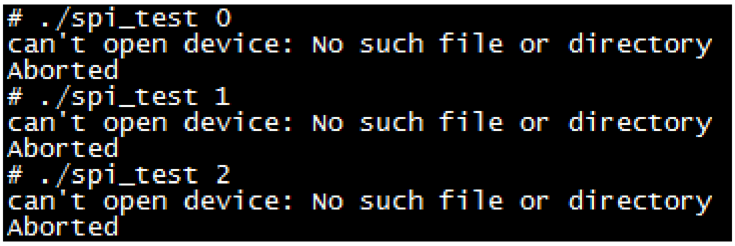
Device Drivers --->

SPI support --->

<*> User mode SPI device driver support

FAQ:

Debug UART prints information:



```
# ./spi_test 0
can't open device: No such file or directory
Aborted
# ./spi_test 1
can't open device: No such file or directory
Aborted
# ./spi_test 2
can't open device: No such file or directory
Aborted
```

Please check the following 2 items：

① Whether [/dev] directory has current channel’s device node?

② Whether kernel configuration is selected ?

Spi can not be transmitted normally, please check the following 2 items:

① Whether spi’s maximum transmitting speed rate fits the actual requirements ?

② Whether application program fits opposite device time sequence requirements ?

2.7. CAN Test

QY-IMX6S mainboard boots 2-ch CAN, 1-ch is CAN driver output, 1-ch is TTL output, test program is for CAN driver outputting CAN signal, it means testing CAN0.

Test Principle:

The file system provides the method to test CAN, use CAN tool to test.

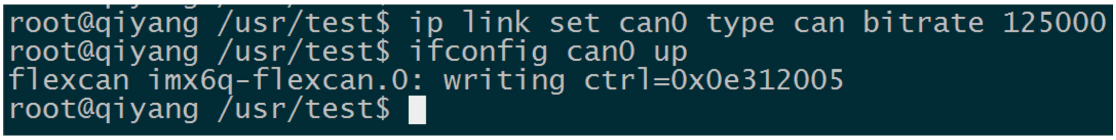
Test Process & Result

1.Need to use 2-ch CAN to test, connect the current CAN to another CAN interface. You could open two IMX6 boards, then connect the two UART terminal to PC . When entering into the system, then configure and open CAN.

```
ip link set can0 type can bitrate 125000
```

```
ifconfig can0 up
```

The Debug UART print the following information:

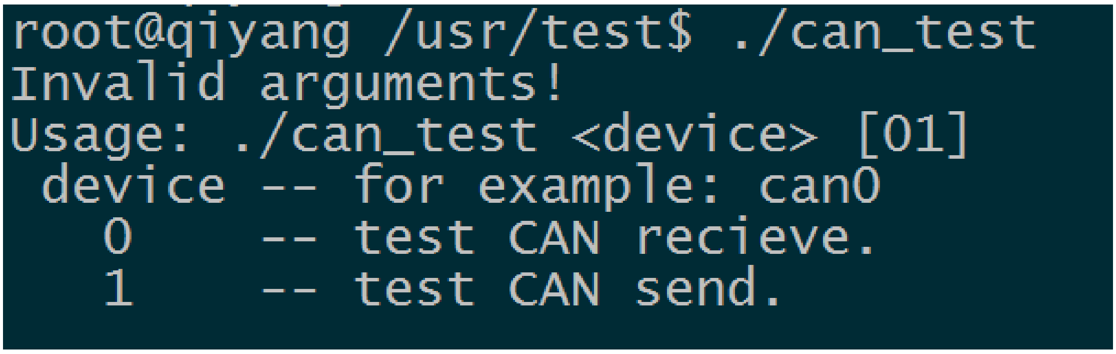


```
root@qiyang /usr/test$ ip link set can0 type can bitrate 125000
root@qiyang /usr/test$ ifconfig can0 up
flexcan imx6q-flexcan.0: writing ctrl=0x0e312005
root@qiyang /usr/test$ 
```

2.Connect J2(PIN16) on board to J2(PIN16) on another board. J2(PIN14) on board to connect another J2(PIN14). So one CAN as transmitting terminal ,and the another CAN as the receiving terminal, then exchange to test.

3.Use [CAN] test program to test:

```
# ./can_test
```



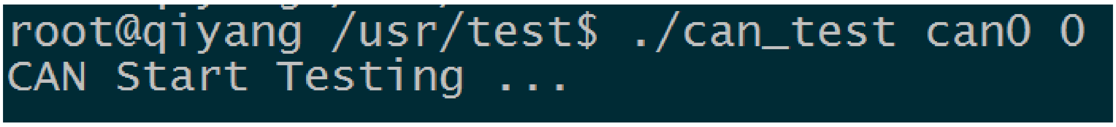
```
root@qiyang /usr/test$ ./can_test
Invalid arguments!
Usage: ./can_test <device> [01]
 device -- for example: can0
   0    -- test CAN recieve.
   1    -- test CAN send.
```

Illustration: ①[can_test <device> 0] set CAN as the receiving data.

②[can_test <device>] set CAN as the sending data.

4.Here, take the CAN on the Board 1 as the receiving terminal, input command on UART terminal.

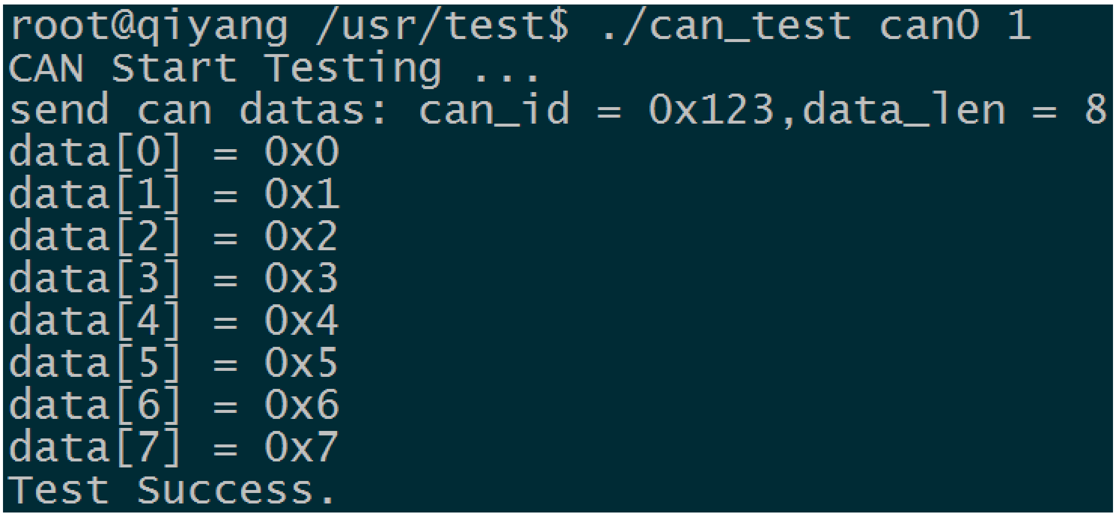
```
./can_test can0 0
```



```
root@qiyang /usr/test$ ./can_test can0 0
CAN Start Testing ...
```

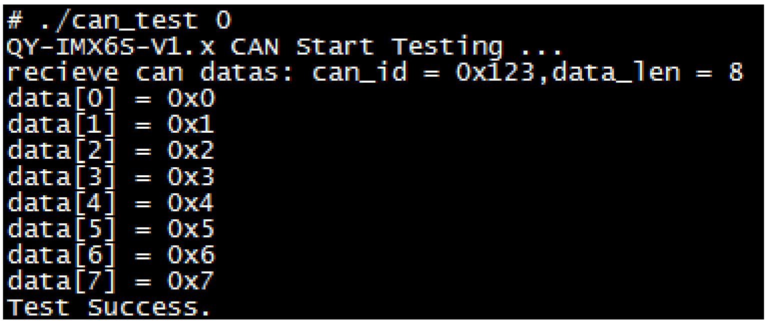
5. Here, take the CAN on the Board 2 as the transmitting terminal, input command on UART terminal.

```
./can_test can0 1
```



```
root@qiyang /usr/test$ ./can_test can0 1
CAN Start Testing ...
send can datas: can_id = 0x123,data_len = 8
data[0] = 0x0
data[1] = 0x1
data[2] = 0x2
data[3] = 0x3
data[4] = 0x4
data[5] = 0x5
data[6] = 0x6
data[7] = 0x7
Test Success.
```

The debug UART terminal on the Board 1 received the CAN data from the Board 2.



```
# ./can_test 0
QY-IMX6S-V1.x CAN Start Testing ...
recieve can datas: can_id = 0x123,data_len = 8
data[0] = 0x0
data[1] = 0x1
data[2] = 0x2
data[3] = 0x3
data[4] = 0x4
data[5] = 0x5
data[6] = 0x6
data[7] = 0x7
Test Success.
```

6.Then exchange to test, take the CAN on the Board 2 as the receiving terminal, take the CAN on the Board 1as the transmitting terminal. The test method is same.

Test Code:

CD/Test Code /can _test/can_test.c

Driver Code:

linux-3.0.101/drivers/net/can/flexcan.c

The Kernel Options:

```
[*] Networking support    --->
    <*>    CAN bus subsystem support    --->
        <*>    Raw CAN Protocol (raw access with CAN-ID filtering)
        <*>    Broadcast Manager CAN Protocol (with content filtering)
               CAN Device Drivers    --->
                   <*> Virtual Local CAN Interface (vcan)
                   <*> Platform CAN drivers with Netlink support
                   [*]    CAN bit-timing calculation
                   <*> Support for Freescale FLEXCAN based chips
```

FAQ:

Debug UART prints the following information:

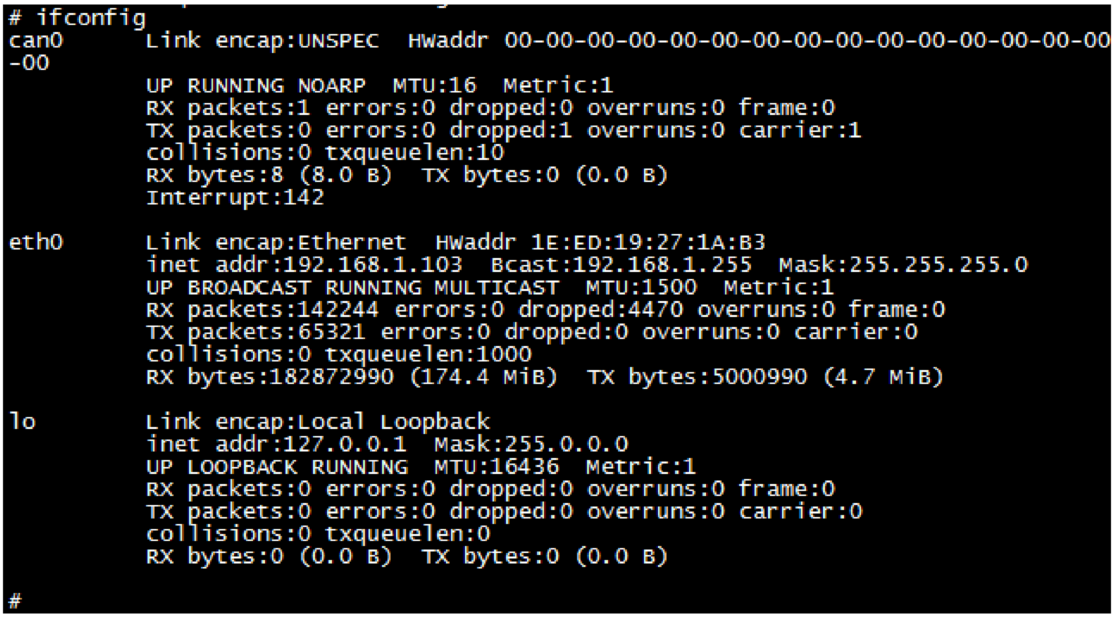
```
read can datas failed.
```

Or

```
send can datas failed.
```

Please check the following items:

1.Use [ifconfig] to check whether the current [can0] is up.



```
# ifconfig
can0      Link encap:UNSPEC  HWaddr 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00
-00
          UP RUNNING NOARP  MTU:16  Metric:1
          RX packets:1 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:1 overruns:0 carrier:1
          collisions:0 txqueuelen:10
          RX bytes:8 (8.0 B)  TX bytes:0 (0.0 B)
          Interrupt:142

eth0      Link encap:Ethernet  HWaddr 1E:ED:19:27:1A:B3
          inet addr:192.168.1.103  Bcast:192.168.1.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:142244 errors:0 dropped:4470 overruns:0 frame:0
          TX packets:65321 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:182872990 (174.4 MiB)  TX bytes:5000990 (4.7 MiB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

#
```

2. If using [ifconfig can0 up] is failed, please check whether the kernel configuration enables [can0].

3.If transmitting is successfully, but the opposite end do not receive. Please check whether the two ends [bitrate] is right.

4.Please check whether the 2*CAN connectivity is right.

2.8. Audio &Video Test

QY-I.MX6S mainboard supports video display function by software decoding, file system provides [gplay] tool to support audio and video play.

Test Principle:

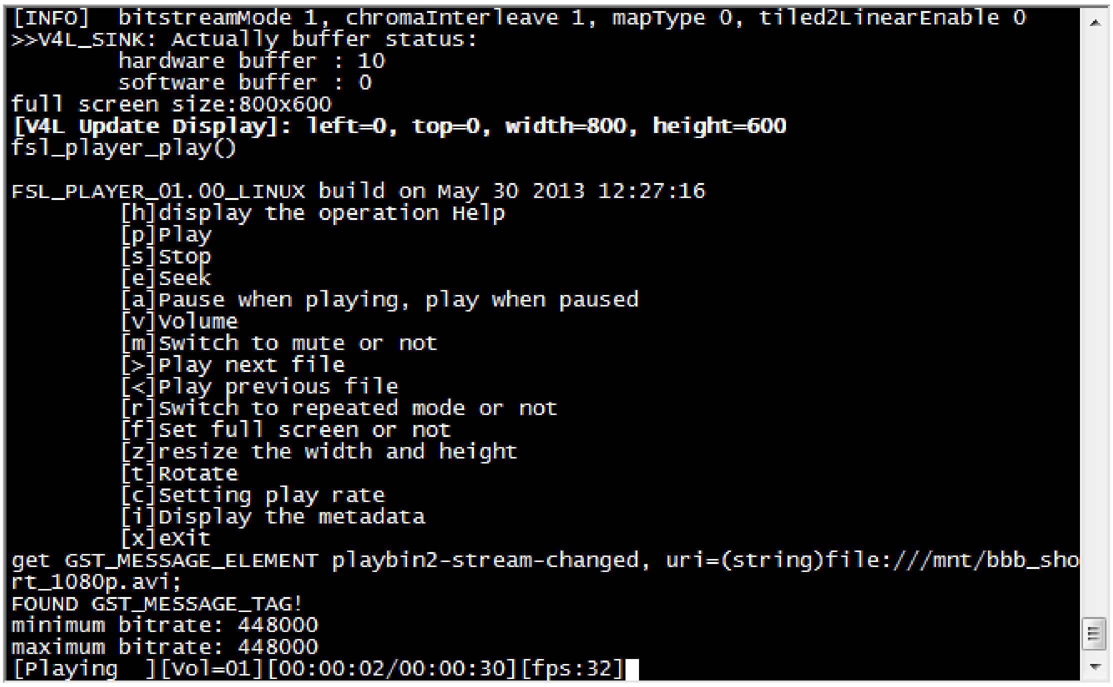
Play audio & video through [gplay] command.

Please ensure the LCD and VGA be connected well, and the earphone or amplifier be connected to J33 well.

Test Process & Test Result:

1、Execute below command to play:

```
gplay bbb_short_1080p.avi
```



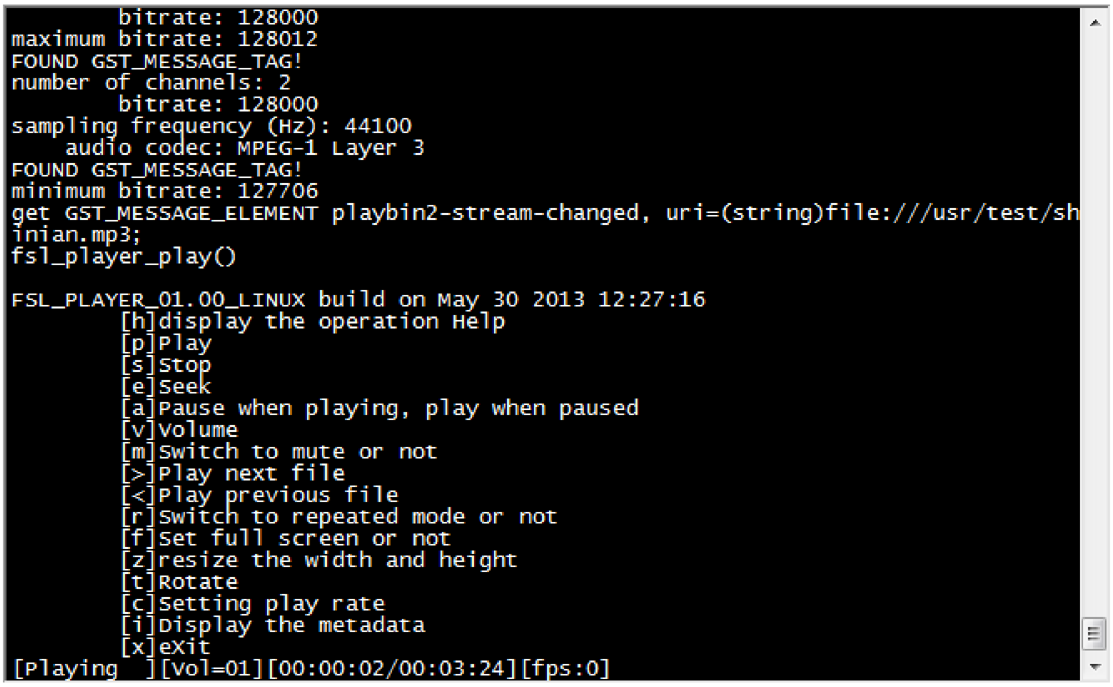
```
[INFO]  bitstreamMode 1, chromaInterleave 1, mapType 0, tiled2LinearEnable 0
>>V4L_SINK: Actually buffer status:
        hardware buffer : 10
        software buffer : 0
full screen size:800x600
[V4L Update Display]: left=0, top=0, width=800, height=600
fsl_player_play()

FSL_PLAYER_01.00_LINUX build on May 30 2013 12:27:16
        [h]display the operation Help
        [p]Play
        [s]Stop
        [e]Seek
        [a]Pause when playing, play when paused
        [v]Volume
        [m]Switch to mute or not
        [>]Play next file
        [<]Play previous file
        [r]Switch to repeated mode or not
        [f]Set full screen or not
        [z]resize the width and height
        [t]Rotate
        [c]Setting play rate
        [i]Display the metadata
        [x]exit
get GST_MESSAGE_ELEMENT playbin2-stream-changed, uri=(string)file:///mnt/bbb_sho
rt_1080p.avi;
FOUND GST_MESSAGE_TAG!
minimum bitrate: 448000
maximum bitrate: 448000
[Playing  ][Vol=01][00:00:02/00:00:30][fps:32]
```

After executing, the demo with 1080P resolution will be shown on LCD or VGA

Video supports [avi、mp4、flv、3gp、mov、ts、vob、mpg、dat] video format.

2、There is audio test file [shinian.mp3] in current directory [/usr/test], you can play this audio file directly to test.

gplay shinian.mp3



```
          bitrate: 128000
maximum bitrate: 128012
FOUND GST_MESSAGE_TAG!
number of channels: 2
          bitrate: 128000
sampling frequency (Hz): 44100
     audio codec: MPEG-1 Layer 3
FOUND GST_MESSAGE_TAG!
minimum bitrate: 127706
get GST_MESSAGE_ELEMENT playbin2-stream-changed, uri=(string)file:///usr/test/sh
inian.mp3;
fsl_player_play()

FSL_PLAYER_01.00_LINUX build on May 30 2013 12:27:16
        [h]display the operation Help
        [p]Play
        [s]Stop
        [e]Seek
        [a]Pause when playing, play when paused
        [v]Volume
        [m]Switch to mute or not
        [>]Play next file
        [<]Play previous file
        [r]Switch to repeated mode or not
        [f]Set full screen or not
        [z]resize the width and height
        [t]Rotate
        [c]Setting play rate
        [i]Display the metadata
        [x]eXit
[Playing  ][vol=01][00:00:02/00:03:24][fps:0]
```

You can hear music from audio output interface.

[gplay] supports [mp2、mp3、m4a、aac、wav、ogg、amr] audio format.

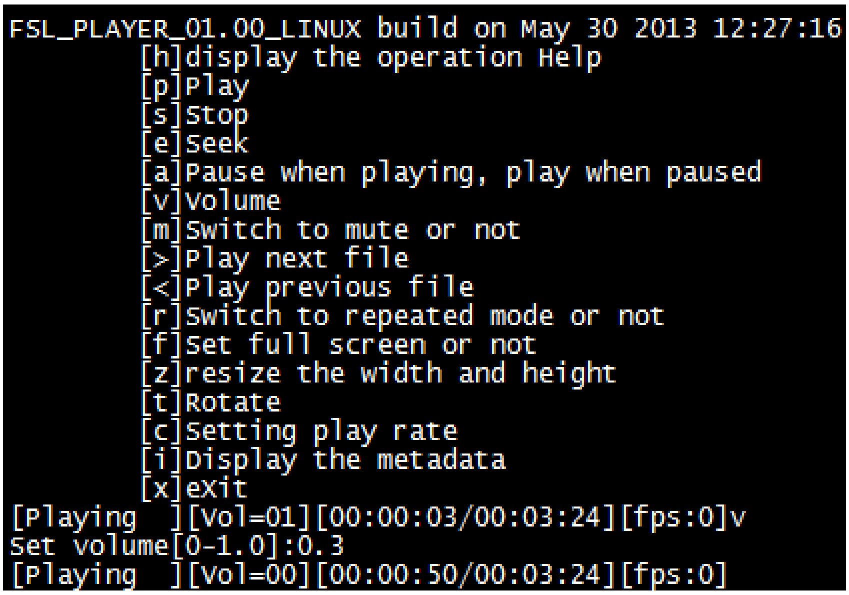
[gplay] has generated powerful play control function, you can control through debug UART:

Button	Function

p	Play
s	Stop
e	Seek
a	Pause when playing, play when paused
v	Volume
m	Switch to mute or not
>	Play next file
<	Play previous file
r	Switch to repeated mode or not
f	Set full screen or not
z	resize the width and height
t	Rotate
c	Setting play rate
i	Display the metadata

x	eXit

Here, we take example by adjusting the volume:



```
FSL_PLAYER_01.00_LINUX build on May 30 2013 12:27:16
        [h]display the operation Help
        [p]Play
        [s]Stop
        [e]Seek
        [a]Pause when playing, play when paused
        [v]Volume
        [m]Switch to mute or not
        [>]Play next file
        [<]Play previous file
        [r]Switch to repeated mode or not
        [f]Set full screen or not
        [z]resize the width and height
        [t]Rotate
        [c]Setting play rate
        [i]Display the metadata
        [x]eXit
[Playing  ][Vol=01][00:00:03/00:03:24][fps:0]v
Set volume[0-1.0]:0.3
[Playing  ][Vol=00][00:00:50/00:03:24][fps:0]
```

Execute [gplay bbb_short_1080p.avi] or [gplay shinian.mp3] command.

Input [v], it hints [Set volume[0-1.0]], range [0~1.0], here, we input [0.3], you can hear the sound is light.

Record Test:

Use microphone to connect J25, then input the following command in terminal:

```
arecord -d 10 -D plughw:1 test.wav
```

It generates [test.wav] in terminal, then use the below command to play the previous record.

```
gplay test.wav
```

Test Code:

```
/Test Code/video bbb_short_1080p.avi
```

Tool：gplay

FAQ:

① If there is issues on display part, please refer to the LCD Charter to check.

② In default, please use the LCD which the resolution is greater than the current image, or the abnormal circumstance will be appeared.

2.9. LAN Test

Test Principle:

Set Board's network, use [ping] to check whether the network is connected well.

Test Process & Test Result:

Network eth0 is J13

1. Connect J13 to Router (Switcher) by a network cable, use another

network cable to connect computer and Router (Switcher), ensure it could access the network.

2.Set board network

Configure automatically, input:

```
udhcpc -i eth0
```

Configure manually, input:

```
ifconfig eth0 192.168.1.71
```

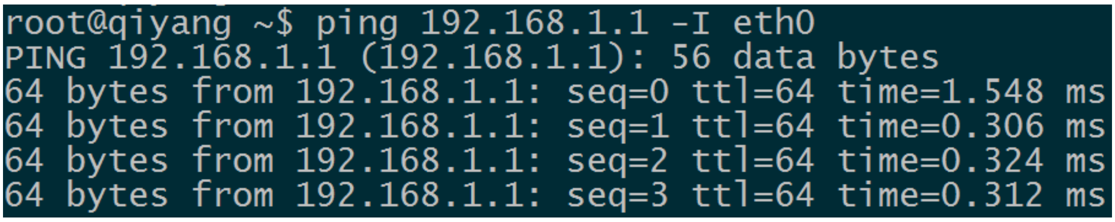
（Default setting）

```
echo nameserver 114.114.114.114 > /etc/resolv.conf
route add default gw 192.168.1.1 dev eth0
```

3. Test internal network, input;

```
ping 192.168.1.1 -I eth0
```

If it prints the below information correctly, input [Ctrl+C] to exit.

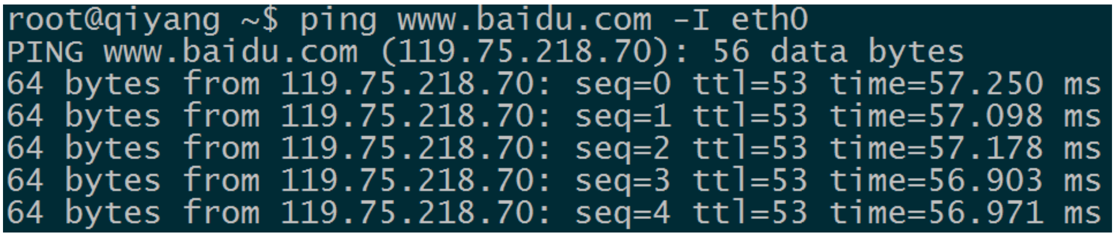


```
root@qiyang ~$ ping 192.168.1.1 -I eth0
PING 192.168.1.1 (192.168.1.1): 56 data bytes
64 bytes from 192.168.1.1: seq=0 ttl=64 time=1.548 ms
64 bytes from 192.168.1.1: seq=1 ttl=64 time=0.306 ms
64 bytes from 192.168.1.1: seq=2 ttl=64 time=0.324 ms
64 bytes from 192.168.1.1: seq=3 ttl=64 time=0.312 ms
```

4、Test Internet ,input :

```
ping www.baidu.com -I eth0
```

If it prints the below information correctly, input [Ctrl+C] to exit.



```
root@qiyang ~$ ping www.baidu.com -I eth0
PING www.baidu.com (119.75.218.70): 56 data bytes
64 bytes from 119.75.218.70: seq=0 ttl=53 time=57.250 ms
64 bytes from 119.75.218.70: seq=1 ttl=53 time=57.098 ms
64 bytes from 119.75.218.70: seq=2 ttl=53 time=57.178 ms
64 bytes from 119.75.218.70: seq=3 ttl=53 time=56.903 ms
64 bytes from 119.75.218.70: seq=4 ttl=53 time=56.971 ms
```

FAQ:

If there is issue, please check the below items:

① Check whether Network LAN is working.

② Check whether the router is working.

2.10 USB Test

Supports 3 formats:fat32,exFAT,NTFS

There are 5-ch USB on QY-I.MX6S mainboard:

① .1-ch (J10) is as device , used to download the firmware program.

② .1-ch J12 to connect wifi module

③ 1-ch has been generated to the miniPCIE interface (J29);

④ 2-ch (J11) are as Host, used to test the host interace.

Test Principle:

USB Host supports hot plug, system will recognize and prints USB Flash Disk after inserting the USB Flash Disk.

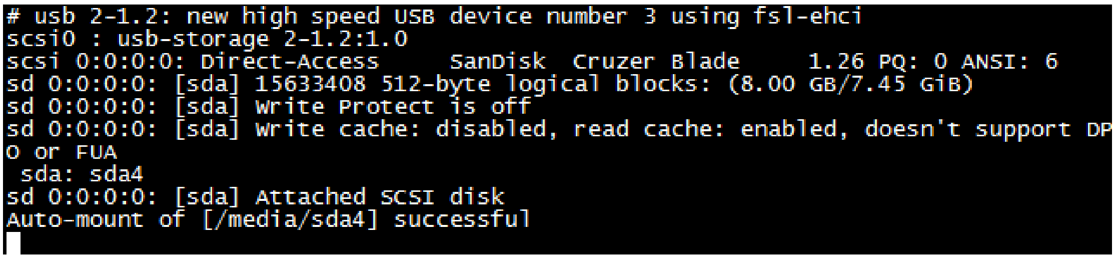
After recognition, it generates device node [/dev/sda] and partition node [/dev/sda1] in [/dev] directory. (If there are several partitions, then the number will be increased.)

Finally, the system will mount all of the partitions to the [/media/]directory, we can judge whether the interface is normal by writing and reading the relative files in this directory.

Test Process & Test Result:

Test on the USB Flash Dish with one partition

Insert USB Flash Disk to J11, debug UART prints the following information:

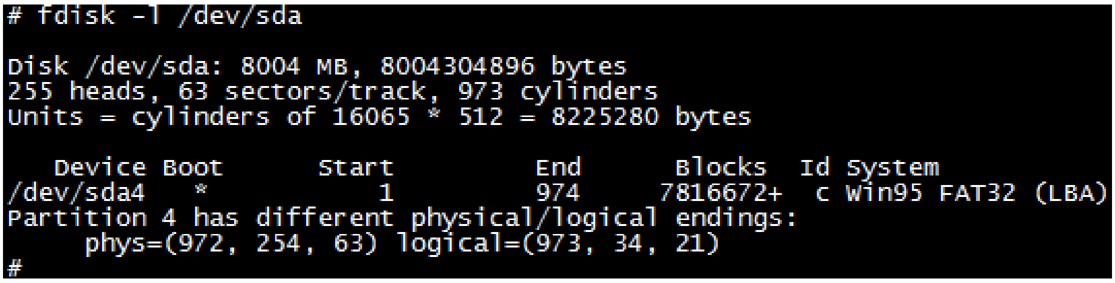


```
# usb 2-1.2: new high speed USB device number 3 using fsl-ehci
scsi0 : usb-storage 2-1.2:1.0
scsi 0:0:0:0: Direct-Access     SanDisk  Cruzer Blade     1.26 PQ: 0 ANSI: 6
sd 0:0:0:0: [sda] 15633408 512-byte logical blocks: (8.00 GB/7.45 GiB)
sd 0:0:0:0: [sda] Write Protect is off
sd 0:0:0:0: [sda] Write cache: disabled, read cache: enabled, doesn't support DP
O or FUA
 sda: sda4
sd 0:0:0:0: [sda] Attached SCSI disk
Auto-mount of [/media/sda4] successful
```

As shown, the contents in USB Flash Disk has been recognized, the device node [sda], child node [sda4].

1. Check USB Flash Disk through [fdisk] command:

```
fdisk -l /dev/sda
```

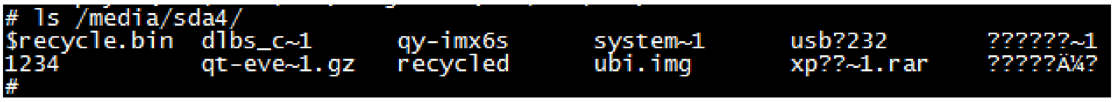


```
# fdisk -l /dev/sda

Disk /dev/sda: 8004 MB, 8004304896 bytes
255 heads, 63 sectors/track, 973 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes

   Device Boot      Start         End      Blocks  Id System
/dev/sda4   *           1         974     7816672+  c Win95 FAT32 (LBA)
Partition 4 has different physical/logical endings:
     phys=(972, 254, 63) logical=(973, 34, 21)
#
```

Here ,the USB Flash Disk has been mounted into [/media/sda4] directory.

1、 Check contents in USB Flash Disk.

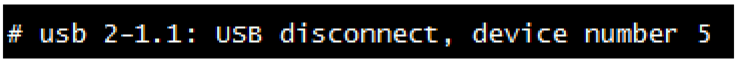
```
ls /media/sda4
```



```
# ls /media/sda4/
$recycle.bin  dlbs_c~1     qy-imx6s     system~1     usb?232      ??????~1
1234          qt-eve~1.gz  recycled     ubi.img      xp??~1.rar   ?????Ä¼?
#
```

2、 Test USB Flash Disk writing and reading through creating ,copying ,deleting files.

3、 Use the same method to test 2-ch USB Host, when finishing , pull out the USB Flash Disk, it prints the information as below:



```
# usb 2-1.1: USB disconnect, device number 5
```

Device Node:

USB Flash Disk:[/dev/sda]

First Partition in USB Flash Disk:[/dev/sda1]

If there are several partitions, the partition [n] corresponds to the [/dev/sdan].

Test Code:

Test Command:[fdisk]

Test Code:

linux-3.0.101 /drivers/usb/host/ehci-hcd.c

The Kernel Options:

```
    Device Drivers --->
            USB support    --->
                  <*>      EHCI    HCD (USB 2.0) support
                  [*]      Support for Freescale controller
                  [*]        Support for DR host port on Freescale controller
                  [*]      Root Hub Transaction Translators
                  <*>      USB OTG pin detect support
```

FAQ:

Inserting USB Flash Disk, there is no any printed information or the recognized error, the USB Flash Disk may be damaged. You can format it on PC at first ,then try again. Or you can change another USB Flash Disk to test.

2.11.SD Card Test

Supports 3 formats: fat32,exFAT,NTFS

QY-IMX6S provide 1-ch SD Card interface (J14) for users to use.

Test Principle:

On board SD card interface support hot plug, after inserting SD card, system will recognize SD card and print the relative information of SD Card.

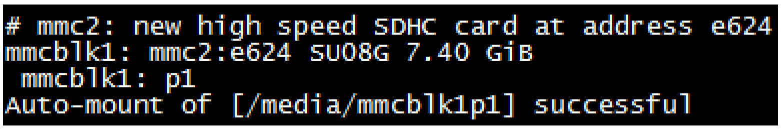
Generate device node and partition node in [/dev] directory. Then system will automatically mount all partition to [/media/] directory. Through read and write the corresponding directory files, you can judge whether interface is normal or not.

Test Process & Result:

The following test process is taken on the SD card which is with only one partition. If with several partitions, please use the same test method.

Insert SD card and generate device node [/dev/mmcblk0]. Partition [n] corresponds to partition device node [/dev/mmcblk0pn]

1.Here, insert a Kingston 8G SD card, print information as follows:

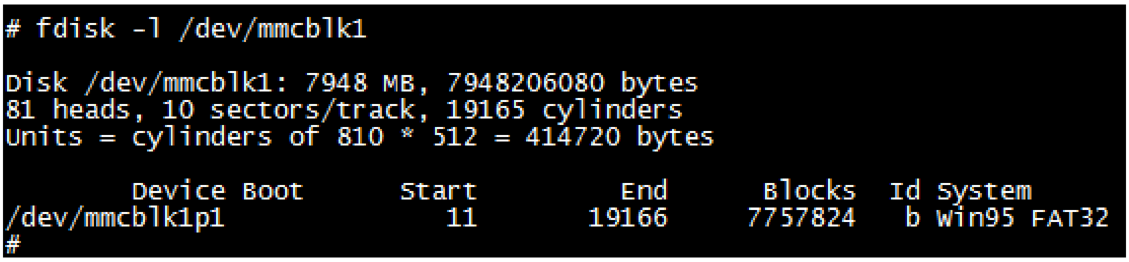


```
# mmc2: new high speed SDHC card at address e624
mmcblk1: mmc2:e624 SU08G 7.40 GiB
 mmcblk1: p1
Auto-mount of [/media/mmcblk1p1] successful
```

As above picture shows, it will show some SD card basic information, here the device node is [mmcblk1], partition is [pl].

2.You can also use [fdisk] command to check SD information.

[fdisk -1 /dev/mmcblk1]

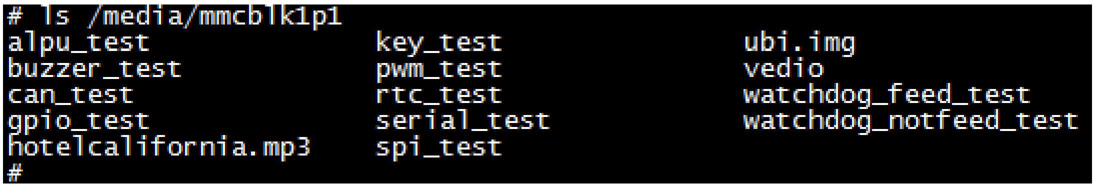


```
# fdisk -l /dev/mmcblk1

Disk /dev/mmcblk1: 7948 MB, 7948206080 bytes
81 heads, 10 sectors/track, 19165 cylinders
Units = cylinders of 810 * 512 = 414720 bytes

        Device Boot      Start         End      Blocks  Id System
/dev/mmcblk1p1              11       19166     7757824   b Win95 FAT32
#
```

Here automatically mount SD Card to [/media/mmcblk0p] directory.

3.Check SD card information

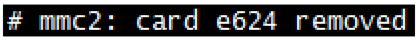
ls/media/mmcblk1p1



```
# ls /media/mmcblk1p1
alpu_test               key_test                ubi.img
buzzer_test             pwm_test                vedio
can_test                rtc_test                watchdog_feed_test
gpio_test               serial_test             watchdog_notfeed_test
hotelcalifornia.mp3     spi_test
#
```

4. Can test SD card read and write through creating, copying, deleting files.

5. Pull out SD card, prints the information as follows:



```
# mmc2: card e624 removed
```

Test Code:

```
Test Tool: fdisk
```

Driver Code:

```
linux-3.0.101/drivers/mmc/host/sdhci-esdhc-imx..c
```

The Kernel Options:

```
Device Drivers --->
    <*> MMC/SD/SDIO card support    --->
        [*]   Assume MMC/SD cards are non-removable(DANGEROUS)
        <*>         MMC block device driver
        (8)   Number of minors per block device
        [*]   Use bounce buffer for simple hosts
        <*> SDHCI support on the platform specific bus
        [*]   SDHCI platform support for the Freescale eSDHC i.MX contr
```

FAQ:

① After plugging in SD Card, without any print information, or can recognize but shows read and write error, it may because of SD Card damage. You can try again after formatting on PC, and also you can change a SD Card to try testing again.

② After plugging in SD Card, it can be recognized but hint write protection, please confirm whether the SD Card hardware write protection has been dialed to [lock] setting or not.

2.12. SATA Test

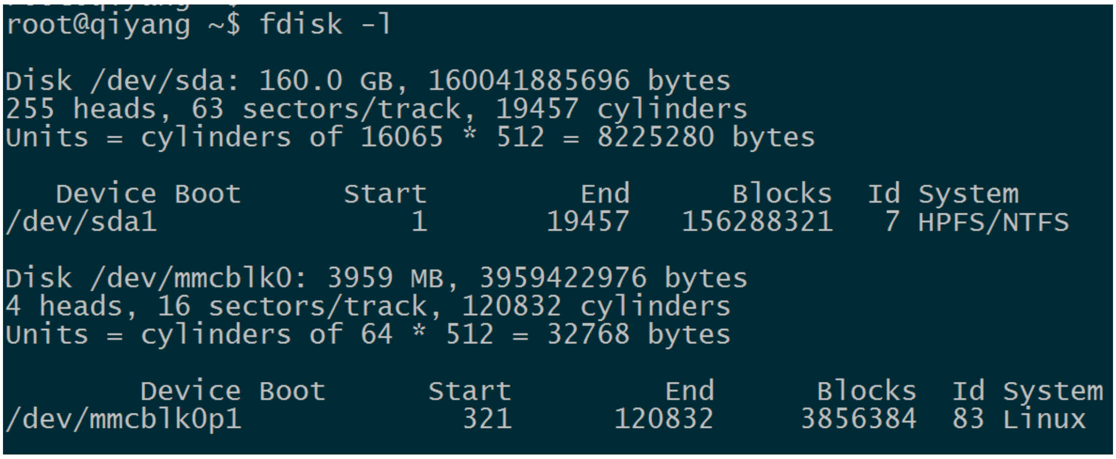
Test Process & Result:

1. Connect SATA (J27) to hardware disk, and provide power

connector on J28 12V and 5V

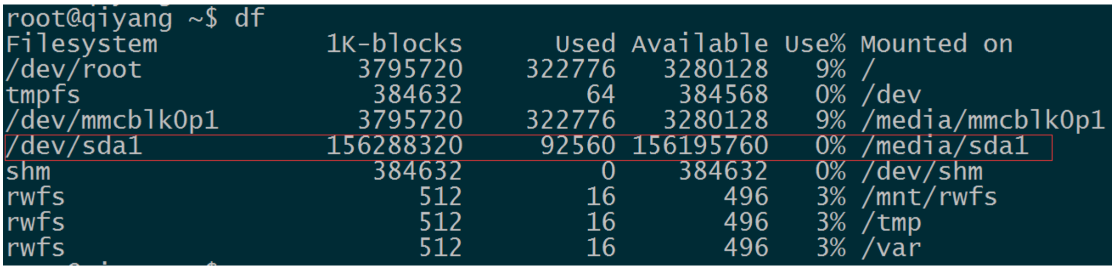
2. Power on , use command to check hardware disk's content:

fdisk-l



```
root@qiyang ~$ fdisk -l

Disk /dev/sda: 160.0 GB, 160041885696 bytes
255 heads, 63 sectors/track, 19457 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes

   Device Boot      Start         End      Blocks  Id System
/dev/sda1               1       19457   156288321   7 HPFS/NTFS

Disk /dev/mmcblk0: 3959 MB, 3959422976 bytes
4 heads, 16 sectors/track, 120832 cylinders
Units = cylinders of 64 * 512 = 32768 bytes

        Device Boot      Start         End      Blocks  Id System
/dev/mmcblk0p1             321      120832     3856384  83 Linux
```

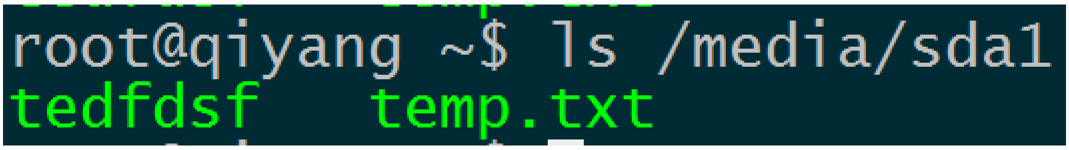
df



```
root@qiyang ~$ df
Filesystem           1K-blocks      Used Available Use% Mounted on
/dev/root              3795720    322776   3280128   9% /
tmpfs                   384632        64    384568   0% /dev
/dev/mmcblk0p1         3795720    322776   3280128   9% /media/mmcblk0p1
/dev/sda1            156288320     92560 156195760   0% /media/sda1
shm                     384632         0    384632   0% /dev/shm
rwfs                       512        16       496   3% /mnt/rwfs
rwfs                       512        16       496   3% /tmp
rwfs                       512        16       496   3% /var
```

Here, we mount the hardware disk to [/media/sda1]directory, user can check the hardware's content directly.

ls /media/sda1



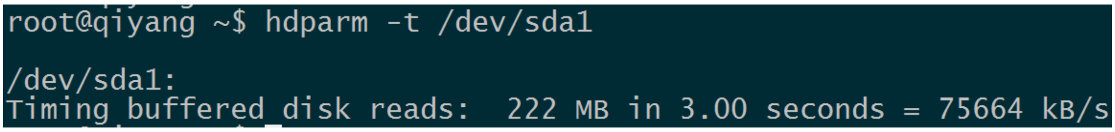
```
root@qiyang ~$ ls /media/sda1
tedfdsf     temp.txt
```

User can test SD card's read an write through creating ,coping, deleting files.

3. Hardware Disk Read and Write Test

```
hdparm -t /dev/sda1
```

If it can test the hardware disk’s write and read speed, it means the SATA works normally.



```
root@qiyang ~$ hdparm -t /dev/sda1

/dev/sda1:
Timing buffered disk reads:  222 MB in 3.00 seconds = 75664 kB/s
```

2.13. HDMI Test

J15 is HDMI interface on board, it supports two kinds of resolutions: 1920x1080 and 1280x720.

Powering on, input in the u-boot command line.

If using 1920x1080 resolution, power on, then input the following command in u-boot command line.

```
set bootargs_mmc ‘setenv bootargs ${bootargs} root=${mmcroot} rootwait rw video=mxcfb0:dev=hdmi,1920x1080M@60,if=RGB24 video=mxcfb1:off video=mxcfb2:off video=mxcfb3:off video=mxcfb4:off'
```

If using 1280 x 720 resolution, power on , then input in the u-boot command line.

```
set bootargs_mmc ‘setenv bootargs ${bootargs} root=${mmcroot} rootwait rw video=mxcfb0:dev=hdmi,1280X720M@60,if=RGB24 video=mxcfb1:off video=mxcfb2:off video=mxcfb3:off video=mxcfb4:off'
```

Save and boot system.

```
saveenv;boot
```

Test Principle:

Set frame buffer in uboot and start HDMI driver, set resolution.

Booting development board, video output is HDMI.

Test Process & Result:

Connect HDMI cable to J15 on board.

Show output information on HDMI displayer, users can modify the resolution and output format to adapt different screens.

2.14. LCD Display and VGA Test

QY-IMX6S mainboard provide 1-ch LCD TFT display interface(J21), 1-ch VGA interface(J24),2-ch LVDS interface (J16 and J17). Current kernel supports resolution LCD [480x272,640x480,800x480,800x600], VGA [1280x1024,1024x768].

4.3 Inch LCD Touch Panel supports 480x272,model No.:

QY-AT043TN24, after powering on, input the following command in u-boot.

```
set bootargs_mmc 'setenv bootargs ${bootargs} root=/dev/mmcblk0p1 rootwait rw
video=mxcfb0:dev=lcd,QY-LCD-480x272,if=RGB24'
```

5.6 Inch LCD Touch Panel, resolution is 640x480,model no.:QY-AT056TN53, after powering on, input the following command in u-boot .

```
set bootargs_mmc 'setenv bootargs ${bootargs} root=/dev/mmcblk0p1 rootwait rw
video=mxcfb0:dev=lcd,QY-LCD-640x480,if=RGB24'
```

7 Inch LCD Touch Panel, resolution is 800x480, model no.: QY-AT070TN83, after powering on, input the following command in u-boot.

```
set bootargs_mmc 'setenv bootargs ${bootargs} root=/dev/mmcblk0p1 rootwait rw
video=mxcfb0:dev=lcd,QY-LCD-800X480,if=RGB24'
```

8 Inch LCD Touch Panel, resolution:800 x 600 ,model no.: QY-AT080TN52, after power on , input the following command in u-boot.

```
set bootargs_mmc 'setenv bootargs ${bootargs} root=/dev/mmcblk0p1 rootwait rw
video=mxcfb0:dev=lcd,QY-LCD-800X600,if=RGB24'
```

VGA Resolution:1024 x768, after powering on, input the following command in u-boot.

```
set bootargs_mmc 'setenv bootargs ${bootargs} root=/dev/mmcblk0p1 rootwait rw
video=mxcfb0:dev=lcd,QY-VGA-1024X768,if=RGB24'
```

VGA resolution: 1280 x1024, after powering on, input the following command in u-boot.

```
set bootargs_mmc 'setenv bootargs ${bootargs} root=/dev/mmcblk0p1 rootwait rw
video=mxcfb0:dev=lcd,1280x1024M@60,if=RGB24'
```

save and boot system

```
saveenv;boot
```

Test Principle:

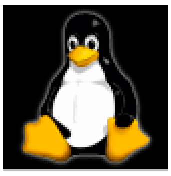
System boots, LCD or VGA will show the penguin picture which kernel owns.

Test Process & Result:

When testing LCD, connect LCD to J21 onboard by cable; When testing VGA, connect LCD to J24 onboard by cable.

Please make sure that the configured kernel resolution and current connected LCD or VGA required resolution are same.

Power on to mainboard, after system boots, LCD or VGA will show the following picture on the left up corner:



Solo core shows1 logo, dual core shows 2 logos, quad core shows 4 logos.

To check whether the logo picture is distortion or jitter, to confirm whether display normal or not. You can also run qt program to test LCD and VGA according to this chapter 2.9.

Test Code:

None

Driver Code:

linux-3.0.101/drivers/video/mxc/mxc_lcdif.c

linux-3.0.101/drivers/video/mxc/mxcfb_hx8369_wvga.c

The Kernel Options:

```
        Device Drivers   --->
Graphics support   --->
<*> MXC Framebuffer support
<*>     MXC EDID support
                    <*>     Synchronous Panel Framebuffer
                    <*>     MXC LDB
                    <*>     MXC MIPI_DSI
                    <*>     TRULY WVGA Panel
                    <*>     E-Ink Panel Framebuffer
```

FAQ:

① LCD Display has no image, please confirm whether select the items according to above kernel option.

② .Display image position match LCD monitor’s size or not, please confirm whether current kernel resolution is the same as current using LCD monitor data.

2.15. LVDS Test

2-CH LVDS interface (J16, J17). QY-HJ070NA , 7 inch LCD resolution: 1024x600.

After powering on, input the following command in u-boot.

```
setenv bootargs_mmc 'setenv bootargs ${bootargs} root=/dev/mmcblk0p1 rootwait rw
video=mxcfb0:dev=ldb,LDB-WSVGA,if=RGB666 ldb=dul0'
```

```
saveenv;boot
```

Load to kernel.

Test Principle:

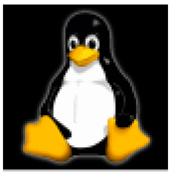
Set frame buffer in uboot and boot HDMI driver, set resolution. Boot development board, video output is HDMI. System boots, it shows penguin picture which kernel owns.

Test Process & Result:

Connect LVDS LCD cable to J16 onboard, touch panel cable to J19 onboard, backlight powering cable to J18 onboard.

Please make sure that the configured kernel resolution and current connected LCD or VGA required resolution are same.

Power on mainboard, after system boots, LCD or VGA will display the following picture on the left up corner:



Solo core shows1 logo, dual core shows 2 logos, quad core shows 4 logos.

To check whether the logo picture is distortion or jitter, to confirm whether display normal or not. You can also run qt program to test LCD and VGA according to this chapter 2.9.

After executing, you will see the demo with 1080P video on LCD or VGA.

2.16. QT Test

QY-IMX6S mainboard standard configuration file system has [4.8.4 qt] library, this program will take you to QT world.

System supports usb mouser and touch panel operation.

Test Principle:

Execute [Imx6_qt_test] program, on the LCD monitor , it shows QT Image. Through touch panel or mouse, Users can move cursor to interact with QT

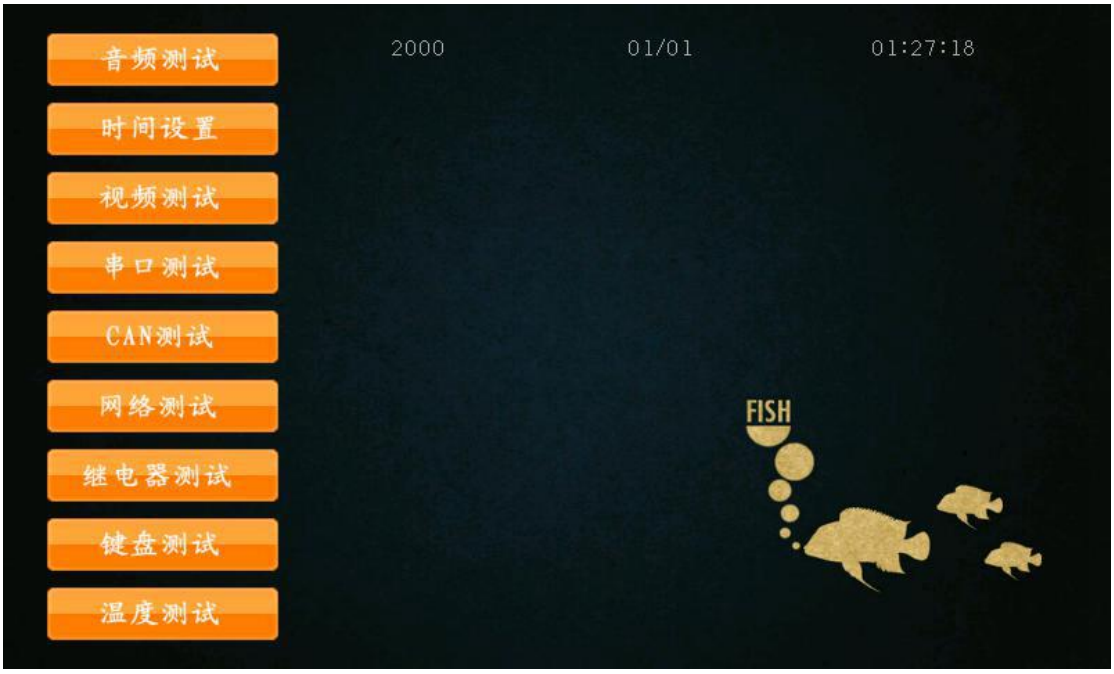
Test Process & Result:

1. Connect USB mouser before powering on, the new touch panel needs to be calibrate. The calibration operation ,please refer to Charter 2.17.

2. Run QT program.

```
./Imx6_qt_test -qws
```

After running, system will load and show QT program image, as shown:



Operate QT image through touch panel or USB mouse

Device Tree file:

None

Driver Code:

None

The Kernel Options:

None

FAQ:

Please refer to touch panel and LCD display questions.

2.17. Touch Panel Test

Touch panel supports resistive touch panel and capacitive touch panel, resistive touch panel pairs with LCD screen. The capacitive touch panel pairs with LVDS screen. The resolution must be same as the LCD resolution in uboot.

If using resistive touch panel, please refer to Charter 2.14 to connect. If using capacitive touch panel, please refer above Charter 2.15

Test Principle:

Use [Tslib] touch panel test tool to calibrate touch panel,

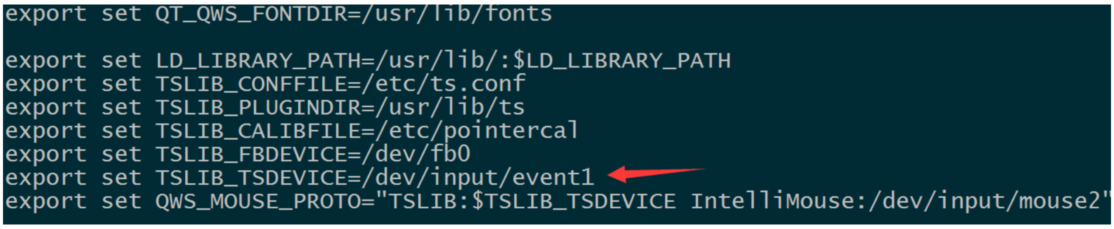
after calibrating, use test tool to drag and draw line. You can see the cursor is moving around the current touch point, and it moves after the touch point position.

Test Process & Result:

1.The default environmental variables match capacitive touch panel. If using resistive touch panel, please modify:

[Vi/etc/OtEnv]

Change the[event2] to [event1]



```
export set QT_QWS_FONTDIR=/usr/lib/fonts

export set LD_LIBRARY_PATH=/usr/lib/:$LD_LIBRARY_PATH
export set TSLIB_CONFFILE=/etc/ts.conf
export set TSLIB_PLUGINDIR=/usr/lib/ts
export set TSLIB_CALIBFILE=/etc/pointercal
export set TSLIB_FBDEVICE=/dev/fb0
export set TSLIB_TSDEVICE=/dev/input/event1
export set QWS_MOUSE_PROTO="TSLIB:$TSLIB_TSDEVICE IntelliMouse:/dev/input/mouse2"
```

Environmental variables take effect:

```
source /etc/QtEnv
```

2.Execute touch panel's calibration program [ts_calibrate]:

```
ts_calibrate
```

It shows cross picture on the left up corner, the serial terminal will prints the coordinate point base on current cross picture. Click the cross picture, this cross picture will skip to another corner. It is finished after four corners and central point be calibrated.

3. After calibration, execute [ts_test] to test touch panel's precision.

Can click drag button or line button to do testing, you will see the mouse or line will move along with the touch place.

Test Code:

Test Tool: `ts_calibrate, ts_test`

Corresponding tool Source Code: `CD/Test Source Code/tslib`

Driver Code:

```
linux-3.0.101/drivers/input/touchscreen/ads7846.c
```

The Kernel Options:

```
Device Drivers --->
    Input device support    --->
            [*]    Touchscreens    --->
```

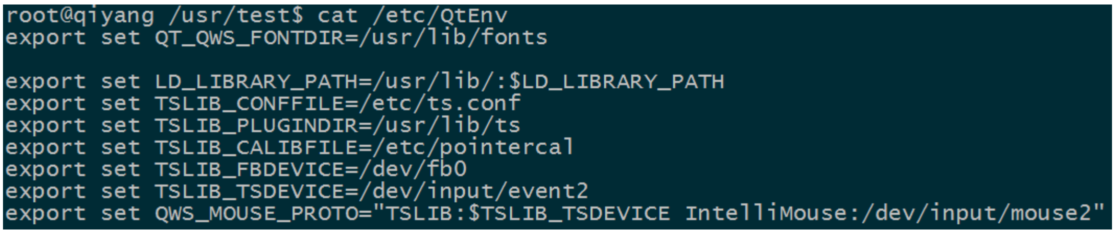
<*> ADS7846/TSC2046/AD7873 and AD(S)7843 based touchscreens

FAQ:

①.If LCD shows the image unnormal, please confirm whether LCD resolution configuration match the current display.

②.After clicking touch panel, cursor display position has a little error and jitter with clicking position, this situation is normal.

③ .If other questions, please check [tslib] environmental variables ,comparing the following environment variables is the same or not.



```
root@qiyang /usr/test$ cat /etc/QtEnv
export set QT_QWS_FONTDIR=/usr/lib/fonts

export set LD_LIBRARY_PATH=/usr/lib/:$LD_LIBRARY_PATH
export set TSLIB_CONFFILE=/etc/ts.conf
export set TSLIB_PLUGINDIR=/usr/lib/ts
export set TSLIB_CALIBFILE=/etc/pointercal
export set TSLIB_FBDEVICE=/dev/fb0
export set TSLIB_TSDEVICE=/dev/input/event2
export set QWS_MOUSE_PROTO="TSLIB:$TSLIB_TSDEVICE IntelliMouse:/dev/input/mouse2"
```

2.18. Camera Test

Test Principle:

The board supports OV5640 camera, run command, test camera.

The board has camera interface J22. We use the OV5640 camera.

Connect OV5640 to J22.

Test Process & Result:

1. J22 connect OV5640 camera
2. Display, refer to Chart 2.13,2.14,2.15
3. Input command

```
gst-launch -v mfw_v4lsrc ! mfw_v4lsink
```

It shows captured image from camera on LCD.

Driver Code:

```
linux-3.0.101\drivers\media\video\mxc\capture\mxc_v4l2_capture.c
```

The Kernel Options:

```
<*> Device Drivers --->
    <*> Multimedia support --->
        <*> Vedio caputure adapters --->
            <*> MXC Video For Linux Camera --->
                < > CSI camera support
                <*> OmniVision ov5640 camera support
                -*- camera clock
                <*> Select Overlay Rounting (Queue ipu device for overlay library)--->
                <*> Pre-processor Encoder library
                <*> IPU CSI Encoder library
```

FAQ:

Debug UART prints information:

```
# gst-launch -v mfw_v4lsrc ! mfw_isink
MFW_GST_V4LSRC_PLUGIN 3.0.5 build on May 30 2013 12:26:32.
MAX resolution 1024x768
MFW_GST_ISINK_PLUGIERROR: v4l2 capture: slave not found!
N 3.0.5 build on May 30 2013 12:27:13.
Setting pipeline to PAUSED ...
ERROR: Pipeline doesn't want to pause.
Setting pipeline to NULL ...
Freeing pipeline ...
[--->FINALIZE isink
#
```

Please check the following items.:

1. Whether cable connection is correct.
2. Whether kernel configuration option is selected.
3. Whether the kernel download is correct.

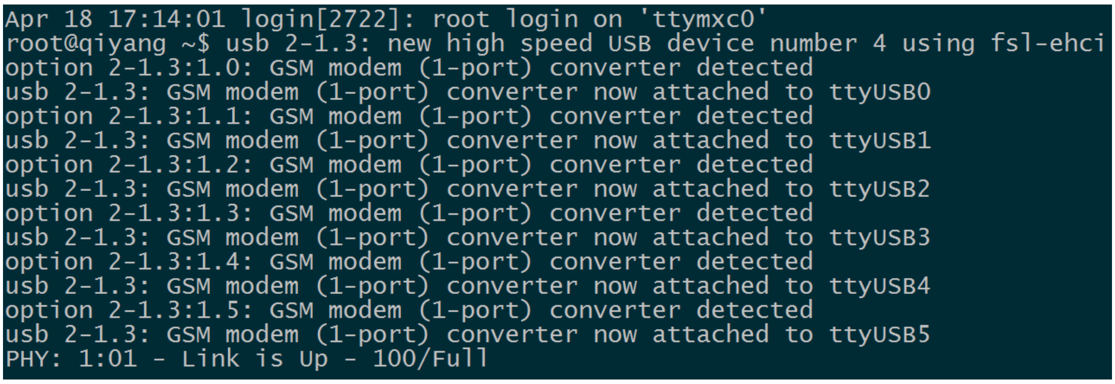
2.19、3G/4G Test

QY-I.MX6S Development board brings J29 minipcie interface, it can connect 3G module.

J30 can connect to other 3G cards. It supports module, SIM7100CE, SIM7600CE.

It supports China mobile 3G,Unicom 3G: UC15

Inserting SIM card on SIM socket, powering on , you can get the printed information.

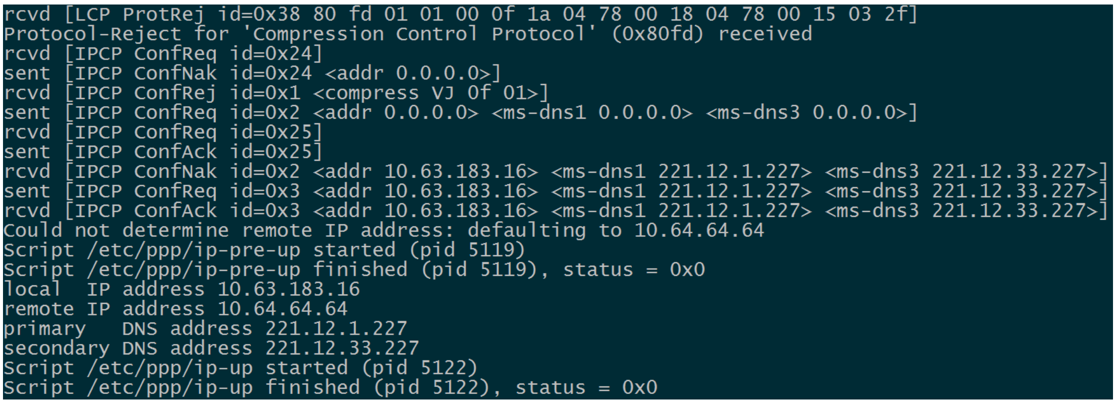


```
Apr 18 17:14:01 login[2722]: root login on 'ttymxc0'
root@qiyang ~$ usb 2-1.3: new high speed USB device number 4 using fsl-ehci
option 2-1.3:1.0: GSM modem (1-port) converter detected
usb 2-1.3: GSM modem (1-port) converter now attached to ttyUSB0
option 2-1.3:1.1: GSM modem (1-port) converter detected
usb 2-1.3: GSM modem (1-port) converter now attached to ttyUSB1
option 2-1.3:1.2: GSM modem (1-port) converter detected
usb 2-1.3: GSM modem (1-port) converter now attached to ttyUSB2
option 2-1.3:1.3: GSM modem (1-port) converter detected
usb 2-1.3: GSM modem (1-port) converter now attached to ttyUSB3
option 2-1.3:1.4: GSM modem (1-port) converter detected
usb 2-1.3: GSM modem (1-port) converter now attached to ttyUSB4
option 2-1.3:1.5: GSM modem (1-port) converter detected
usb 2-1.3: GSM modem (1-port) converter now attached to ttyUSB5
PHY: 1:01 - Link is Up - 100/Full
```

Execute on terminal:

```
pppd call 3g4gnet &
```

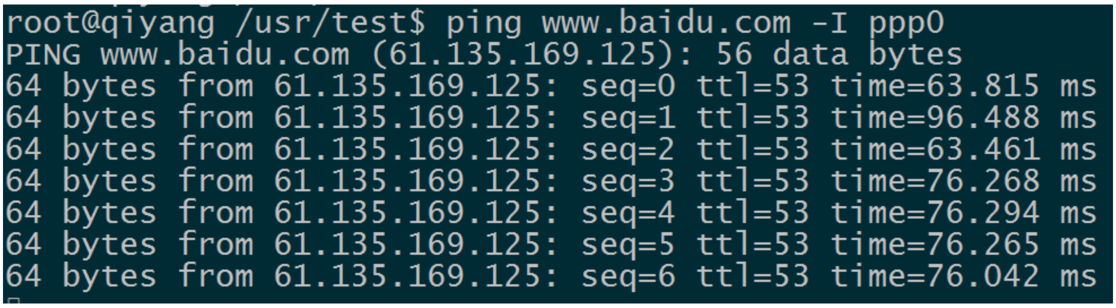
It prints following information , it means to connect Internet successfully.



```
rcvd [LCP ProtRej id=0x38 80 fd 01 01 00 0f 1a 04 78 00 18 04 78 00 15 03 2f]
Protocol-Reject for 'Compression Control Protocol' (0x80fd) received
rcvd [IPCP ConfReq id=0x24]
sent [IPCP ConfNak id=0x24 <addr 0.0.0.0>]
rcvd [IPCP ConfRej id=0x1 <compress VJ 0f 01>]
sent [IPCP ConfReq id=0x2 <addr 0.0.0.0> <ms-dns1 0.0.0.0> <ms-dns3 0.0.0.0>]
rcvd [IPCP ConfReq id=0x25]
sent [IPCP ConfAck id=0x25]
rcvd [IPCP ConfNak id=0x2 <addr 10.63.183.16> <ms-dns1 221.12.1.227> <ms-dns3 221.12.33.227>]
sent [IPCP ConfReq id=0x3 <addr 10.63.183.16> <ms-dns1 221.12.1.227> <ms-dns3 221.12.33.227>]
rcvd [IPCP ConfAck id=0x3 <addr 10.63.183.16> <ms-dns1 221.12.1.227> <ms-dns3 221.12.33.227>]
Could not determine remote IP address: defaulting to 10.64.64.64
Script /etc/ppp/ip-pre-up started (pid 5119)
Script /etc/ppp/ip-pre-up finished (pid 5119), status = 0x0
local  IP address 10.63.183.16
remote IP address 10.64.64.64
primary   DNS address 221.12.1.227
secondary DNS address 221.12.33.227
Script /etc/ppp/ip-up started (pid 5122)
Script /etc/ppp/ip-up finished (pid 5122), status = 0x0
```

Visit Internet, press [Ctrl+C]to exist:

```
ping www.baidu.com -I ppp0
```



```
root@qiyang /usr/test$ ping www.baidu.com -I ppp0
PING www.baidu.com (61.135.169.125): 56 data bytes
64 bytes from 61.135.169.125: seq=0 ttl=53 time=63.815 ms
64 bytes from 61.135.169.125: seq=1 ttl=53 time=96.488 ms
64 bytes from 61.135.169.125: seq=2 ttl=53 time=63.461 ms
64 bytes from 61.135.169.125: seq=3 ttl=53 time=76.268 ms
64 bytes from 61.135.169.125: seq=4 ttl=53 time=76.294 ms
64 bytes from 61.135.169.125: seq=5 ttl=53 time=76.265 ms
64 bytes from 61.135.169.125: seq=6 ttl=53 time=76.042 ms
```

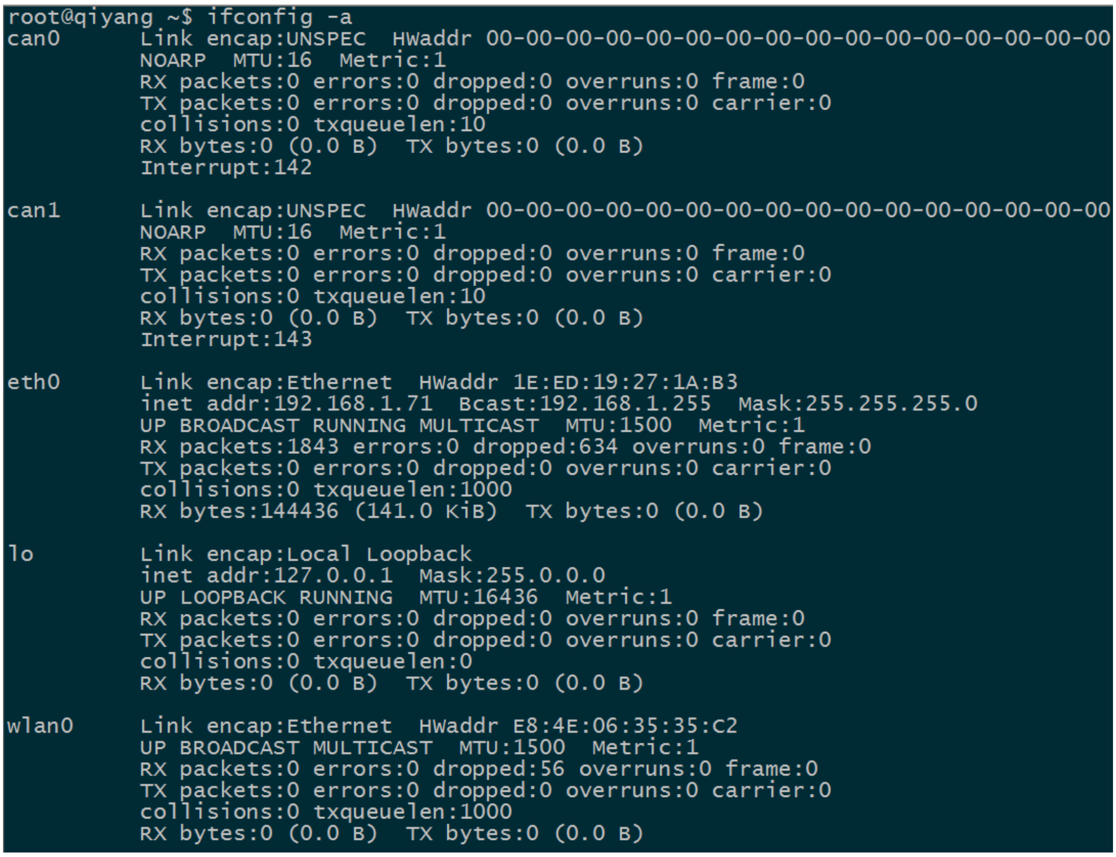
2.20. Wifi Test

QY-I.MX6S brings J12 or J11. It can be connected to WIFI. The module supports RTL8188CUS.

Input RTL8188CUS on J12 or J11. It outputs on terminal as follows:

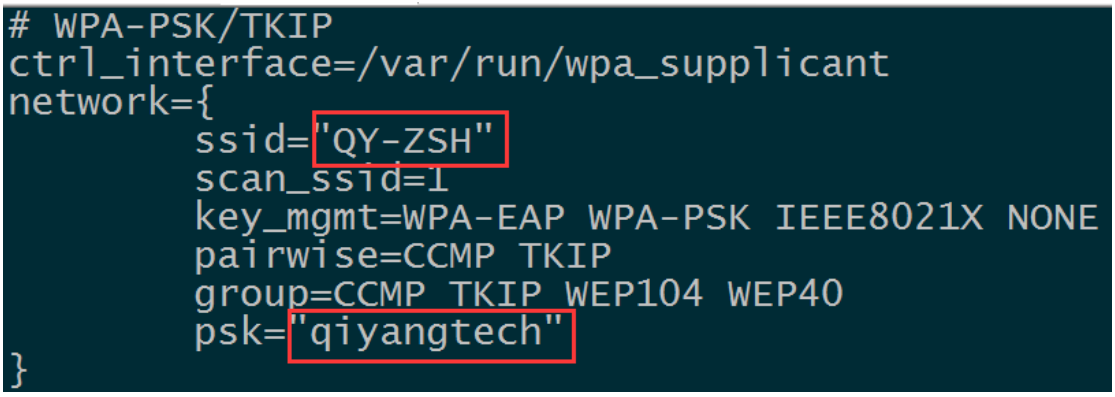
```
ifconfig -a
```

It prints [wlan0]



```
root@qiyang ~$ ifconfig -a
can0      Link encap:UNSPEC  HWaddr 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00
          NOARP  MTU:16  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:10
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)
          Interrupt:142

can1      Link encap:UNSPEC  HWaddr 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00
          NOARP  MTU:16  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:10
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)
          Interrupt:143

eth0      Link encap:Ethernet  HWaddr 1E:ED:19:27:1A:B3
          inet addr:192.168.1.71  Bcast:192.168.1.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:1843 errors:0 dropped:634 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:144436 (141.0 KiB)  TX bytes:0 (0.0 B)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

wlan0     Link encap:Ethernet  HWaddr E8:4E:06:35:35:C2
          UP BROADCAST MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:56 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)
```

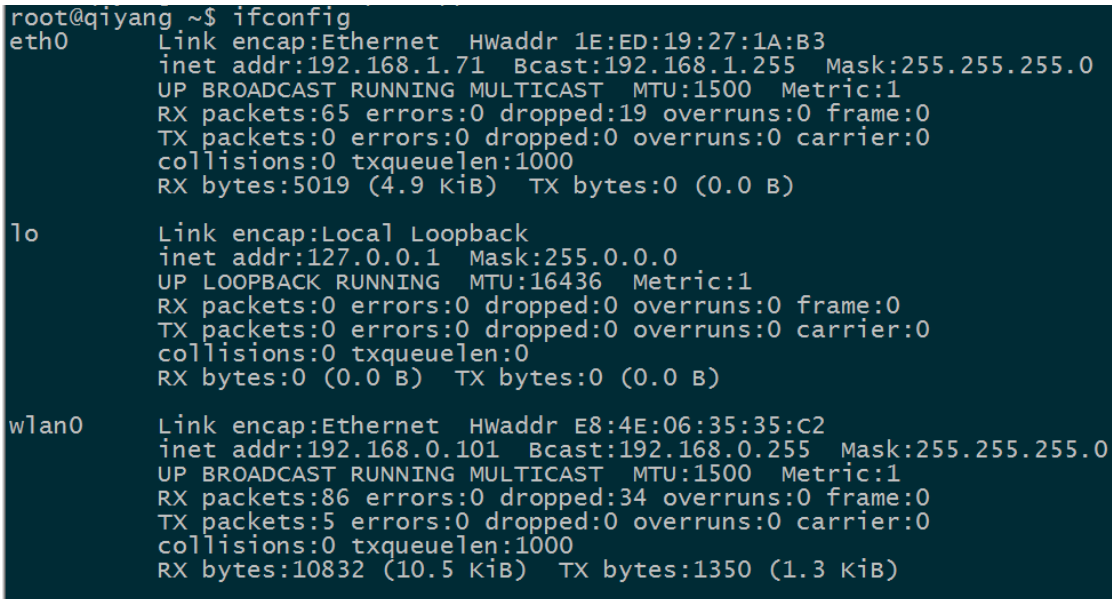
Modify configuration file [/etc/wpa_supplicant.conf], modify the below [ssid] and [psk] to the corresponding user name and password.



```
# WPA-PSK/TKIP
ctrl_interface=/var/run/wpa_supplicant
network={
        ssid="QY-ZSH"
        scan_ssid=1
        key_mgmt=WPA-EAP WPA-PSK IEEE8021X NONE
        pairwise=CCMP TKIP
        group=CCMP TKIP WEP104 WEP40
        psk="qiyangtech"
}
```

Input [sync], power on again, wait for some seconds, input:

```
ifconfig
```

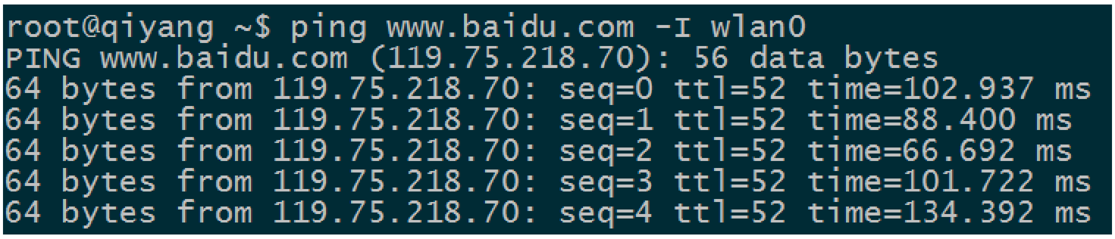


```
root@qiyang ~$ ifconfig
eth0      Link encap:Ethernet  HWaddr 1E:ED:19:27:1A:B3
          inet addr:192.168.1.71  Bcast:192.168.1.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:65 errors:0 dropped:19 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:5019 (4.9 KiB)  TX bytes:0 (0.0 B)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

wlan0     Link encap:Ethernet  HWaddr E8:4E:06:35:35:C2
          inet addr:192.168.0.101  Bcast:192.168.0.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:86 errors:0 dropped:34 overruns:0 frame:0
          TX packets:5 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:10832 (10.5 KiB)  TX bytes:1350 (1.3 KiB)
```

If it shows [wlan0], it means the module has been mounted successfully.

Test Internet.

ping www.baidu.com -I wlan0



```
root@qiyang ~$ ping www.baidu.com -I wlan0
PING www.baidu.com (119.75.218.70): 56 data bytes
64 bytes from 119.75.218.70: seq=0 ttl=52 time=102.937 ms
64 bytes from 119.75.218.70: seq=1 ttl=52 time=88.400 ms
64 bytes from 119.75.218.70: seq=2 ttl=52 time=66.692 ms
64 bytes from 119.75.218.70: seq=3 ttl=52 time=101.722 ms
64 bytes from 119.75.218.70: seq=4 ttl=52 time=134.392 ms
```

The above information means the module works normally.

2.21. RS485 Test

On QY-I.MX6S development board, it has reserved 2-ch RS485, multiplex with RS232.

If using RS485, it needs to remove RS232 chipset on hardware, and solder RS485 circuit.

RS485 and Hardware Relation Chart:

UART	Hardware Location	Device Node
COM1	J31（A、B TO Pin11, Pin13）	/dev/ttymxc1
COM2	J31（A、B TO Pin12, Pin14）	/dev/ttymxc2

Test Principle:

Test program can achieve RS485 transmitting and receiving signal, need to use two boards. Take RS485 as receiving end, another RS485 as transmitting end. Check the date whether it is correct.

Test Process & Result：

Here, we take example of testing RS485 on COM1:

1.Use 2 boards, Board 1 and Board 2. connect PIN A, PINB on J31 by Dupont Line separately.

2. Power on, when in Uboot, press [Enter], input the following

command:

```
setenv bootargs_mmc 'setenv bootargs ${bootargs} root=${mmcroot} rootwait rw rs485=1,2'
```

```
saveenv;boot
```

3.System boots, Board 1 as RS485 receiving end, input command :

```
/usr/test/rs485_test /dev/ttymxc1 115200 0
```

4.Board 2 as RS485 transmitting end, input command:

```
/usr/test/rs485_test /dev/ttymxc1 115200 1
```

5.Check Board 1 printed information, user can see the printed information every 1 second.

```
receive 28 datas: "/dev/ttymxc2" test string!
```

6.Exchange the above Step3, Step 4 command, to see the Board 2 transmitting and receiving date.

Ⅲ. Test Summary

QY-I.MX6S development board function tests are finished. If you meet any issues in test process, you can use the test code to check.

Zhejiang Qiyang Intelligent Technology Co., Ltd

Tel: 86 -571-87858811 / 87858822

Fax: 86-571-89935912

Technology Support: 86-571-89935913

E-MAIL: supports@qiyangtech.com

Website: http://www.qiytech.com

Address: 3rd floor, Building A, WSCG Building, NO.6
Xiyuan 8th Road, Sandun Town, Xihu District, Hangzhou
City, Zhejiang ,PRC .310030

Post Code: 310030