



## QY-IMX6S嵌入式主板Linux

### 功能说明与测试手册

版本号：V 1.0  
2017年4月

浙江启扬智能有限公司版权所有  
QIYANG TECHNOLOGY Co., Ltd  
Copyright Reserved

---

## 目 录

|                          |    |
|--------------------------|----|
| 版本说明.....                | 3  |
| 前言.....                  | 4  |
| 一、准备工作.....              | 5  |
| 二、主板测试.....              | 6  |
| 2.1、蜂鸣器测试.....           | 6  |
| 2.2、RTC 测试.....          | 8  |
| 2.3、看门狗测试.....           | 11 |
| 2.4、GPIO 测试.....         | 12 |
| 2.5、串口测试.....            | 15 |
| 2.6、SPI 测试.....          | 21 |
| 2.7、CAN 测试.....          | 24 |
| 2.8、音频、视频测试.....         | 27 |
| 2.9、网口测试.....            | 30 |
| 2.10、USB 测试.....         | 32 |
| 2.11、SD 卡测试.....         | 34 |
| 2.12、SATA 测试.....        | 36 |
| 2.13、HDMI 测试.....        | 38 |
| 2.14、LCD 显示与 VGA 测试..... | 39 |
| 2.15、LVDS 测试.....        | 41 |
| 2.16、QT 测试.....          | 42 |
| 2.17、触摸屏测试.....          | 43 |
| 2.18、CAMERA 测试.....      | 45 |
| 2.19、3G/4G 测试.....       | 47 |
| 2.20、wifi 测试.....        | 49 |
| 2.21、RS485 测试.....       | 51 |
| 三、测试小结.....              | 52 |

## 版本说明

| 版本  | 硬件平台          | 描述         | 日期         | 修订人  |
|-----|---------------|------------|------------|------|
| 1.0 | QY-IMX6S-V1.2 | 初始版本, 首次发布 | 2017-04-25 | hech |

## 前言

本手册主要介绍各接口的功能以及测试方法。

使用之前请务必仔细阅读：

《QY-IMX6S 硬件说明书.pdf》和《QY-IMX6S Linux 用户手册.pdf》。

## 一、准备工作

测试之前请阅读《QY-IMX6S Linux 用户手册.pdf》，并按照该手册进行设置连接开发板。  
给主板上电，系统启动后，输入 root 用户进入到板子的文件系统中，如下图所示：

```

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.

#
    
```

主板测试程序位于/usr/test 目录下，请切换到该目录下，后续测试操作都在该目录下完成。

```

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

## 二、主板测试

### 2.1、蜂鸣器测试

QY-IMX6S 主板使用 GPIO 6\_31 控制主板上的蜂鸣器。

设置为低电平时，蜂鸣器不工作；设置为高电平时蜂鸣器鸣响。

#### 测试原理：

该项测试主要实现蜂鸣器鸣响。

#### 测试步骤和结果：

运行蜂鸣器测试程序 `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.
```

说明：`buzzer_test <device> 0` 蜂鸣器不工作，可以关闭蜂鸣器

`buzzer_test <device> 1` 蜂鸣器鸣响

1、打开蜂鸣器，听到主板上的蜂鸣器持续发出蜂鸣声。按 `ctrl+c` 键退出程序。

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

2、关闭蜂鸣器，按 `ctrl+c` 键退出程序。

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

#### 设备节点：

`/dev/qiyang_buzzer`

#### 测试源码：

光盘/测试源码/buzzer\_test/buzzer\_test.c

#### 驱动代码：

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

#### 内核对应选项：

Device Drivers --->

Misc devices --->

<\*> BUZZER FOR QIYANG IMX6 BOARD

#### 可能遇到的问题：

调试串口打印信息：

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

请按检查以下几项：

①、`/dev` 目录下是否有 `qiyang_buzzer` 设备节点

- 
- ②、内核配置中是否选中了<\*> BUZZER FOR QIYANG IMX6 BOARD 选项
  - ③、设备树文件是否使能了 qiyang\_buzzer 节点。

## 2.2、RTC 测试

QY-IMX6S 主板使用 I2C2 连接底板上的 DS1338 芯片作为外部硬件时钟,请在测试 RTC 之前,确保已经安上电池。

### 测试原理:

通过 date 系统命令设置系统时间,用 hwclock 命令把系统时间写入硬件时钟,通过 rtc\_test 测试程序读取硬件时钟并打印出来,断电后重启,查看时钟是否准确。

### 测试步骤和结果:

- 1、在板上执行 date 命令,可查看到当前系统时钟。

date

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

- 2、用 date 命令设置系统时钟,比如按当前 PC 的显示时间来设置。

date 041709312017 /\*月日時分年\*/

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

- 3、用 hwclock 命令把系统时间写入硬件时钟芯片。

hwclock -w

- 4、分别用 date 和 hwclock 命令来查看系统和硬件时钟。

```
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、设置成功之后,执行 rtc\_test 测试程序。

./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 ***
    
```

程序打印 10 条当前硬件时间后退出程序，提前退出程序请按 `ctrl+c` 查看是否精准走时，查看有无出现丢秒现象。

6、断电，过一段时间再上电，用 `date` 和 `hwclock` 命令查看系统和硬件时钟，检查时间有没有保存，走时是否精准。

```

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、通过和 PC 时间对比之后，发现时间无误差，若需测试长时间的走时精确度，可以分别断电和上电老化测试几天几周或者几个月来测试时钟误差。

本主板在发布之前都经过 1 个月的老化测试，时间误差不超过 2S  
对于发货产品均经过 24 小时以上老化测试，时间误差不超过 1S

**设备节点：**

`/dev/rtc`

`/dev/rtc0`

**测试源码：**

光盘/源码/测试源码/rtc\_test/rtc\_test.c。

**驱动代码：**

ds1338 驱动：linux-3.0.101/drivers/rtc/rtc-ds1307.c

i2c 驱动：linux-3.0.101/drivers/i2c/busses/i2c-imx.c

**内核对应选项：**

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
```

## 可能遇到的问题:

现象: 时间无法保存、走时偏差很大、无法查看硬件时钟。

调试串口打印信息:

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

请检查以下几项:

- ①、底板板上 BT1 是否安装了电池, 电池是否有电。
- ②、/dev 目录下是否有/dev/rtc0 和/dev/rtc1 节点。
- ③、内核配置项是否已经配置上。

## 2.3、看门狗测试

### 测试原理：

硬件看门狗，GPIO\_3\_23 使能看门狗，GPIO\_3\_22 执行喂狗操作。

### 测试步骤和结果：

1、启用看门狗，并执行喂狗操作，可以看到系统不会重启。按 ctrl+c 键退出程序。

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

2、启用看门狗，但不执行喂狗操作，1.6 秒后系统会自动重启。

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

### 设备节点：

```
/dev/qy_watchdog
```

### 测试源码：

光盘/源码/测试源码/watchdog\_test/watchdog\_feed\_test.c。

光盘/源码/测试源码/watchdog\_test/watchdog\_notfeed\_test.c。

### 驱动代码：

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

### 内核对应选项：

```
Device Drivers --->
```

```
  [*] Misc devices --->
```

```
    [*] WATCH_DOG FOR QIYANG BOARD
```

### 可能遇到的问题：

①、执行./ watchdog\_notfeed\_test 系统不会自动重启，请检查看门狗芯片是否损坏并更换。

## 2.4、GPIO 测试

本测试针对以下 16 路 GPIO 引脚：

J5 上的 GPIO2\_0、GPIO2\_1、GPIO2\_2、GPIO2\_3、GPIO2\_4、GPIO2\_5、GPIO2\_6、GPIO2\_7

J31 上的 GPIO2\_16、GPIO2\_17、GPIO2\_18、GPIO2\_19、GPIO2\_20、GPIO2\_21、GPIO2\_22、GPIO2\_23

### 测试原理：

gpio\_test 0 测试 gpio 引脚无外部连接情况下，把所有引脚先设置成低电平，按回车后设置成高电平，通过外部测量 gpio 实际电平来确认 gpio 是否正常！

gpio\_test 1 直接读取外接电平信号，用户可通过读取的电平值与连接的电平值做比较，确认 gpio 是否正常。

### 测试步骤和结果：

- 1、运行 gpio 测试程序 gpio\_test

`./gpio_test`

出现如下打印信息：

```
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.
```

说明： gpio\_test <device> 0 设置 gpio 的高电平和低电平

gpio\_test <device> 1 获取 gpio 的电平

- 2、J5 和 J31 的 gpio 无外接信号，执行

`./gpio_test /dev/qiyang_imx6_gpio 0`

```

root@qiyang /usr/test$ ./gpio_test /dev/qiyang_imx6_gpio 0
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!
    
```

如上图所示，把每个 gpio 设置成低电平，用万用表测量对应的 gpio 的实际电平值来确定 gpio 是否正常，按回车之后把所有 gpio 设置成高电平，测量对应的 gpio 的实际电平值来确定 gpio 是否正常：

```

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!
    
```

再按回车，提示测试 OK：

```

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

### 3、外部接 3.3V 到管脚（管脚内部弱上拉）

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

有任何技术问题或需要帮助，请联系：[supports@qiyangtech.com](mailto:supports@qiyangtech.com)

第 13 页 共 53 页

购买产品，请联系销售：[sales@qiyangtech.com](mailto:sales@qiyangtech.com)

更多信息请访问：<http://www.qiyatech.com>

©2013 Qiyangtech 版权所有

```

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!
    
```

如上图所示，获取每一个 gpio 的电平状态，用户可以改变实际的接入 gpio 信号来确认 gpio 是否正常。

#### 设备节点:

/dev/qiyang\_imx6\_gpio

#### 测试源码:

光盘/测试源码/gpio\_test/gpio\_test.c

#### 驱动代码:

linux-3.0.101/drivers/misc/qy\_imx6\_gpio.c

#### 内核对应选项:

Device Drivers --->

Misc devices --->

<\*> IMX6 GPIO TEST FOR QIYANG BOARD

#### 可能遇到的问题:

调试串口显示:

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

请检查以下几项:

- ①、/dev 目录下是否有/dev/qiyang\_imx6\_gpio 节点
- ②、内核配置中是否选中了<\*> IMX6 GPIO TEST FOR QIYANG BOARD 选项

## 2.5、串口测试

在 QY-IMX6S 主板上，共有 5 路串口，其中 1 路被用来作为调试串口（J6），其他四路可以当作 RS232 使用，其中：

COM1(J31)和 COM1(J31)为 5 线 RS232，与 RS485 接口复用；

COM3（J2）和 COM4（J2）为 3 线 RS232。

本测试程序只针对 RS232 普通串口进行测试。

串口和硬件关系表：

| 串口        | 硬件位置                       | 设备节点        |
|-----------|----------------------------|-------------|
| DBG（调试串口） | J6（收、发、GND 对应 2、1、3 脚）     | /dev/ttymx0 |
| COM1      | J31（收、发、GND 对应 11、13、9 脚）  | /dev/ttymx1 |
| COM2      | J31（收、发、GND 对应 12、14、10 脚） | /dev/ttymx2 |
| COM3      | J2（收、发、GND 对应 1、3、5 脚）     | /dev/ttymx3 |
| COM4      | J2（收、发、GND 对应 2、4、6 脚）     | /dev/ttymx4 |

### 测试原理：

测试程序实现了一个串口每隔 1s 发送字符数据 `"/dev/ttymx $x$ " test string!`，其中  $x$  为实际测试的设备节点，同时通过多线程方式，阻塞读取串口数据并打印。

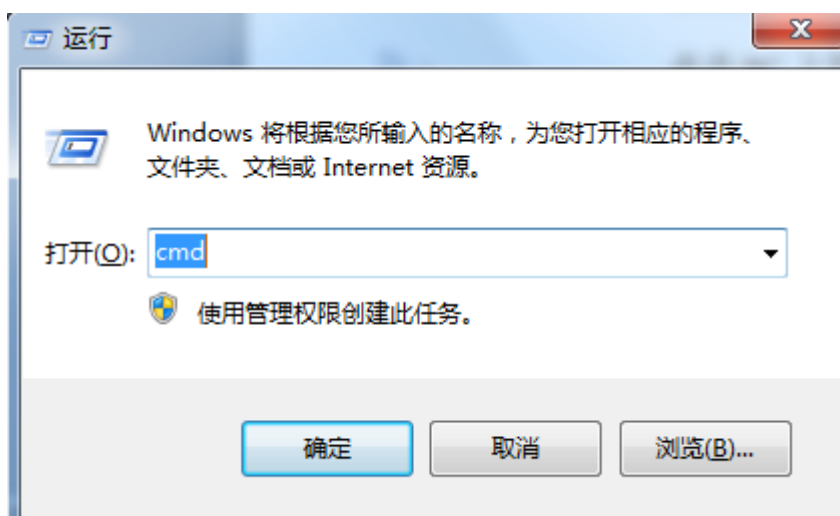
### 测试步骤和结果：

串口测试时 PC 机需要用到两个串口：

- ①、一个接调试串口，用于交互
- ②、一个接待测串口，用于测试串口收发数据

如果 PC 机上只有一个串口，串口连接待测串口，用网线连接开发板，通过 telnet 功能登录开发板系统来作为调试串口操作。

开发板默认 IP 地址为 192.168.1.71 点击 PC 上的开始按钮，选择“运行”，输入 `cmd`，点击“确定”。



在运行的窗口里面输入 `telnet 192.168.1.71`

有任何技术问题或需要帮助，请联系：[supports@qiyangtech.com](mailto:supports@qiyangtech.com)

购买产品，请联系销售：[sales@qiyangtech.com](mailto:sales@qiyangtech.com)

更多信息请访问：<http://www.qiytech.com>

©2013 Qiyangtech 版权所有

## 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
```

输入用户名 root 和密码 root 进入控制台，进入/usr/test 测试目录

## Telnet 192.168.1.71

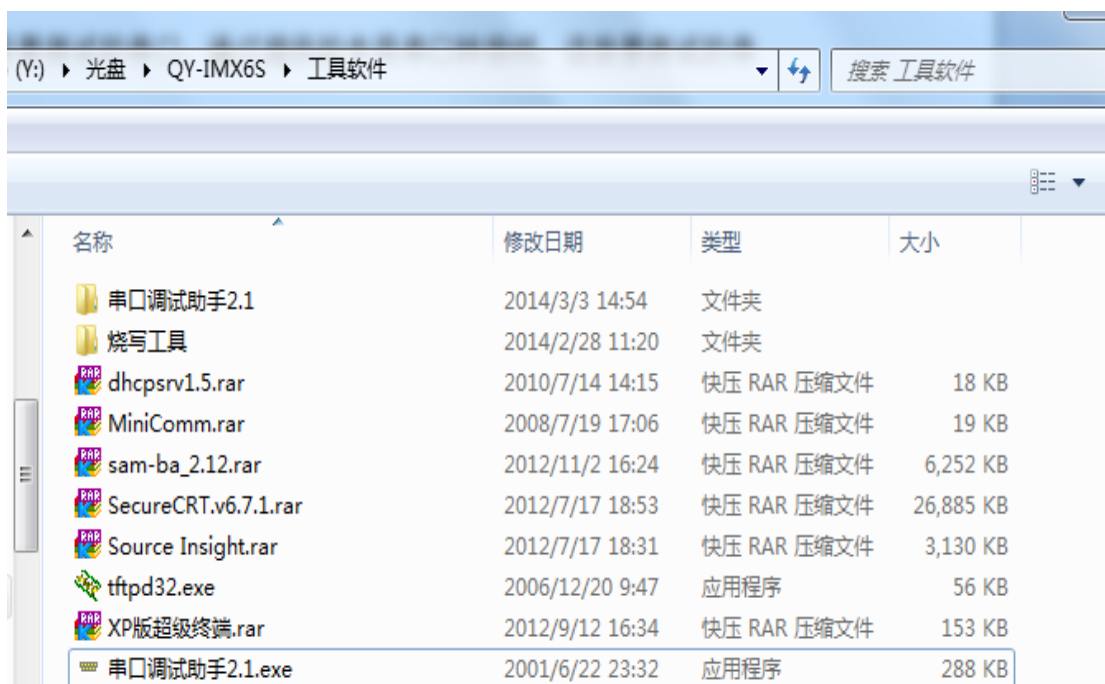
```
qiyang login: root
Password:
root@qiyang ~$ cd /usr/test/
root@qiyang /usr/test$

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

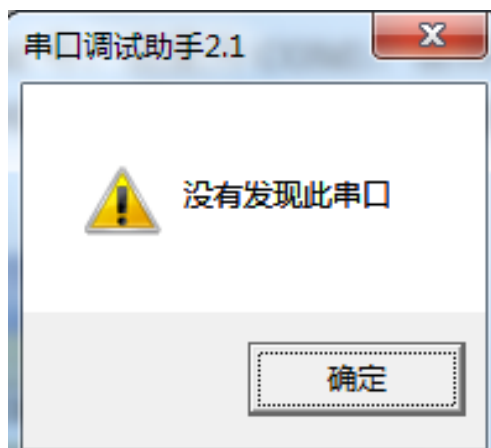
根据串口和硬件关系表，选择要测试的串口，通过提供的专用串口转接线，连接要测试的串口到 PC 机串口上。

PC 机打开光盘中的串口调试工具。



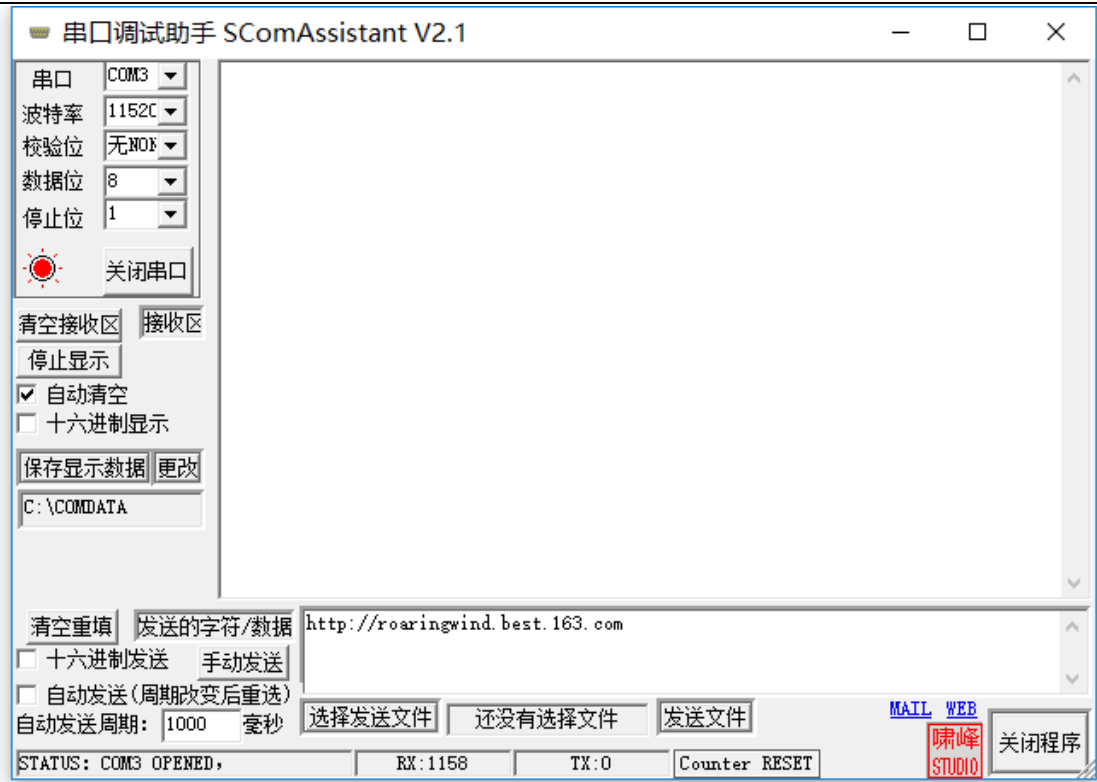


如果打开之后提示如下：



表示默认的 PC 机的 COM 口被终端占用了，关闭被占用的终端，然后再串口调试工具。

设置串口属性，串口对应 PC 机的 COM 号，这里为 COM3，波特率为 115200，数据位为 8 位，停止位为 1，奇偶校验为 NONE。



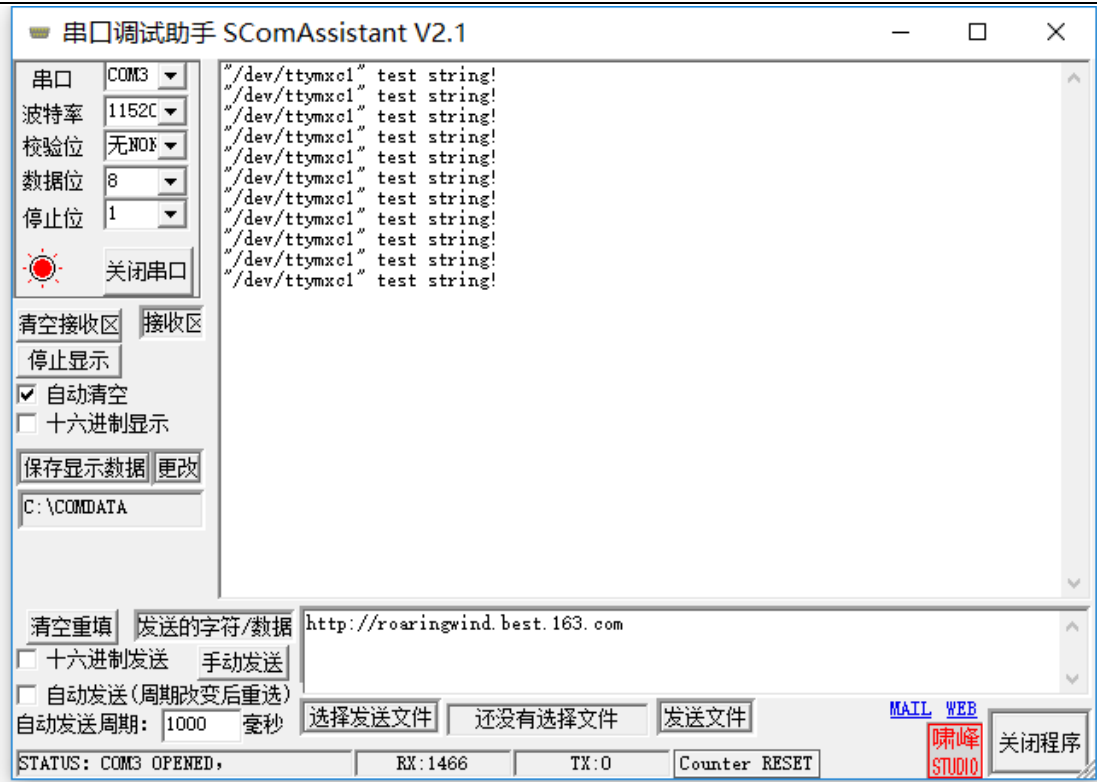
串口连接好并设置好之后就可以开始测试了。

分别测试 COM1、COM2、COM3、COM4 这 4 个串口。

这里以 COM1 为例作介绍，其他串口测试方法一样。

```
./rs232_test /dev/ttymxcl 115200
```

运行之后调试串口助手显示收到的数据。



在调试串口助手上点击手动发送，telnet 界面上显示收到的数据。

```

root@qiyang /usr/test$ ./rs232_test /dev/ttymxcl 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 和串口调试助手上都能收到数据并且数据显示无误，表示串口功能正常。

COM1 测试完毕后，Ctrl+C 退出程序，继续其他串口测试。

### 设备节点:

- /dev/ttymxcl
- /dev/ttymxcl2
- /dev/ttymxcl3
- /dev/ttymxcl4

### 测试源码:

/光盘/测试源码/serial\_test/rs232\_test.c

### 驱动代码:

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

### 内核对应选项:

Device Drivers --->

Character devices --->

Serial drivers --->

[\*] IMX serial port support

### 可能遇到的问题:

调试串口打印信息:

open serial device /dev/ttymx1 error!

如果串口通讯异常或者无法通讯, 请检查以下几项:

- ①、连接线是否正常
- ②、PC 端接串口调试软件配置是否正确
- ③、串口硬件和测试程序节点上是否对应
- ④、/dev 目录下 ttymx1、ttymx2、ttymx3、ttymx4 设置节点是否存在
- ⑤、内核配置选项有没选上

## 2.6、SPI 测试

主板引出了 SPI 接口(J5)，共有三个片选信号，可外接三路 SPI 设备。

三路 SPI 均使用了 SPI 通用驱动 `spidev`，驱动中实现了对设备数据读写的基本操作，使用时，只需要根据外接设备的时序来编写应用程序即可。

### 测试原理：

由于未外接 SPI 外设，测试程序只对 SPI 接口进行简单的写数据测试，测试程序将每隔 1s 发送一组数据。若短接 spi 的接收和发送脚，在调试串口中断会打印出该组数据，也可通过示波器测量到该波形。如需涉及到 spi 的半双工写操作与全双工读操作，可参考该源码。

关系表：

片选与设备节点对应如下：

| 片选   | 设备节点           | 硬件位置     |
|------|----------------|----------|
| NCS0 | /dev/spidev4.0 | J5 的 6 脚 |
| NCS1 | /dev/spidev4.1 | J5 的 7 脚 |
| NCS2 | /dev/spidev4.2 | J5 的 8 脚 |

### 测试步骤和结果：

1、运行测试程序 `spidev_test`

```
./spidev_test
```

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

说明：  
`spidev_test /dev/spidev4.0` 测试 NCS0 片选通道  
`spidev_test /dev/spidev4.1` 测试 NCS1 片选通道  
`spidev_test /dev/spidev4.2` 测试 NCS2 片选通道

2、这里以 NCS0 片选为例进行介绍，其他片选测试方法一样。

```
./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、将 SPI5\_MOSI (J5 的 4 引脚) 和 SPI5\_MISO (J5 的 3 引脚) 引脚短接。

```
./spidev_test /dev/spidev4.0
```

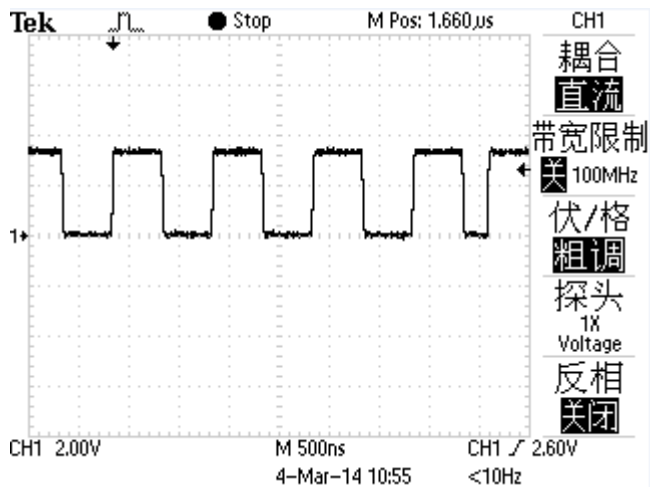
```

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

```

用示波器查看 SPI5\_MOSI 的波形:



### 设备节点:

SPI5\_NCS0: /dev/spidev4.0

SPI5\_NCS1: /dev/spidev4.1

SPI5\_NCS2: /dev/spidev4.2

### 测试源码:

光盘/测试源码/spi\_test/spidev\_test.c

### 驱动代码:

linux-3.0.101/drivers/spi/spidev.c

### 内核对应选项:

```

Device Drivers --->
    SPI support --->
        <*> User mode SPI device driver support

```

### 可能遇到的问题:

调试串口打印信息:

```
# ./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
```

请检查以下 2 项：

- ①、/dev 目录下是否存在当前通道的设备节点
- ②、内核配置选项是否选上

spi 不能正常发送，请检查以下 2 项：

- ①、spi 最大传输速率是否符合实际要求
- ②、应用层程序是否符合对端设备的时序要求

## 2.7、CAN 测试

QY-IMX6S 主板引出了两路 CAN，一路是 can 驱动输出，一路是 ttl 输出，测试程序针对 can 驱动输出的 can 信号，即 can0 进行测试。

### 测试原理：

文件系统中提供了测试 CAN 的方法，使用 CAN 工具进行测试。

### 测试步骤和结果：

1、需使用两路 CAN 方可进行测试，可将当前 CAN 连接到其他 CAN 接口。可以打开两块 IMX6 的板子，连接两个串口终端，进入系统后先配置和打开 CAN。

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

```
ifconfig can0 up
```

调试串口打印信息：

```
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、将一块板子 J2 的 16 脚与另一块板子 J2 的 16 脚相连，J2 的 14 脚与 J2 的 14 脚相连。其中一个 CAN 作为发送端，另一个作为接收端，之后再对调测试。

3、使用 CAN 测试程序进行测试：

```
./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.
```

说明： ①、can\_test <device> 0 把 CAN 设置成接收数据。

②、can\_test <device> 1 把 CAN 设置成发送数据。

4、这里把第 1 块主板端的 CAN 当作接收端，在串口终端输入命令：

```
./can_test can0 0
```

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

5、把第 2 块主板端的 CAN 当作发送端，在串口终端输入命令：

```
./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.
    
```

此时在第 1 块主板的调试串口终端收到第 2 块主板发送的 CAN 数据

6、然后对调测试，把第 2 块主板端的 CAN 当作接收端，把第 1 块主板端的 CAN 当作发送端，测试方法一样。

#### 测试代码：

光盘/测试源码/can\_test/can\_test.c

#### 驱动代码：

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

#### 内核对应选项：

```

[*] 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
    
```

#### 可能遇到的问题：

调试串口打印信息：

read can datas failed.

或者

send can datas failed.

请检查以下几项：

- ①、使用 ifconfig 查看当前 can0 口是否 up

```
# ifconfig
can0      Link encap:UNSPEC  Hwaddr 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00
-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、若使用 `ifconfig can0 up` 失败，请检查内核配置选项是否使能 `can0` 功能。
- ③、发送成功，但对端未接收到，请检查这两个端口设置的 `bitrate` 是否一致。
- ④、检查 2 个 CAN 口的硬件连接是否正确。

## 2.8、音频、视频测试

QY-IMX6S 主板通过软件解码支持视频播放功能，文件系统提供了 `gplay` 工具来支持音视频播放。

### 测试原理：

可通过 `gplay` 命令，播放音视频。

播放视频之前请外接好 LCD 或 VGA。播放音频之前请外接好耳塞或者音响到 J33 接口。

### 测试步骤和结果：

1、执行如下命令进行播放：

```
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]
```

执行之后，在显示接口 LCD 或者 VGA 上就能看到 1080P 高清视频 demo 的播放。

视频播放支持 avi、mp4、flv、3gp、mov、ts、vob、mpg、dat 等视频格式。

2、在当前 `/usr/test` 目录下已经存放了音频测试文件 `shinian.mp3`，可以直接播放该音频文件进行测试。

```
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]
    
```

在音频输出接口上就能听到 mp3 音乐了。

gplay 支持 mp2、mp3、m4a、aac、wav、ogg、amr 等音频格式。

在播放过程中 gplay 还集成了强大的播放控制功能,可以通过调试串口输入来控制:

| 按键 | 功能                                   |
|----|--------------------------------------|
| 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                 |

下面以调节音量来举例说明：

```
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]
```

执行 `gplay bbb_short_1080p.avi` 或者 `gplay shinian.mp3` 命令后。

输入“v”，提示“Set volume[0-1.0]:”，输入范围为 0~1.0，这里输入“0.3”，音频输出接口上听到声音变轻。

### 录音测试：

使用麦克风接好录音接口 J25，然后在终端中输入以下命令即可录音：

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

在终端中可以看出生成一个 `test.wav`，然后使用下面命令可以把刚才录音播放出来

```
gplay test.wav
```

### 测试代码：

/测试源码/video bbb\_short\_1080p.avi

播放工具：gplay

### 可能遇到的问题：

- ①、若显示部分有问题，请参照 LCD 显示章节排查 LCD 问题。
- ②、默认情况下请使用大于当前影片分辨率的 LCD，否则可能出现播放不正常情况。

## 2.9、网口测试

### 测试原理:

设置板子网络，用 ping 工具检查网络是否连通

### 测试步骤和结果:

网络 eth0 接口为 J13。

1、用一根网线连接板子 J13 网口和路由器（交换机亦可），用另一根网线连接电脑和路由器（交换机亦可），确保网络能上网。

2、设置板子网络

自动配置，输入：

```
udhcpc -i eth0
```

手动配置，输入：

```
ifconfig eth0 192.168.1.71 （板上电已经默认设置为这个）
```

```
echo nameserver 114.114.114.114 > /etc/resolv.conf
```

```
route add default gw 192.168.1.1 dev eth0
```

3、测试内网，输入：

```
ping 192.168.1.1 -I eth0
```

如果打印下面信息表明正确，输入 ctrl+c 退出

```
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、测试外网，输入：

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

如果打印下面信息表明正确，输入 ctrl+c 退出

```
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
```

### 可能遇到的问题:

如果报错, 请检查以下几项:

- ① 网线是否良好
- ② 检查路由器是否能上网

## 2.10、USB 测试

支持 3 种格式：fat32、exFAT、NTFS

在 QY-IMX6S 主板上，共有 5 路 usb 口：

- ①、1 路（J10）作为 device 口，用于 usb 方式下载固件程序；
- ②、1 路 J12 外接 wifi 模块；
- ③、1 路集成到 miniPCIe 接口上（J29）；
- ④、2 路（J11）作为 host 口使用，本测试针对 host 口进行测试。

### 测试原理：

开发板 usb host 口支持热插拔，将 U 盘插入后系统会自动识别并打印出 U 盘相关信息。

识别后在/dev 目录下生成该设备节点/dev/sda 及分区节点/dev/sda1（若有多个分区，则数字部分依次增加）。

最终系统会自动将所有分区挂载到/media/目录下，通过读写对应目录下的文件来判断该接口是否正常。

### 测试步骤和结果：

测试以只有一个分区的 U 盘为例。

- 1、将正常使用的 U 盘插入 J11 口，调试串口打印如下信息：

```
# 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 D
0 or FUA
sda: sda4
sd 0:0:0:0: [sda] Attached SCSI disk
Auto-mount of [/media/sda4] successful
```

如上图所示，显示 U 盘的一些基本信息，U 盘识别的设备节点为 sda，子节点为 sda4。

- 2、通过 fdisk 命令来查看 U 盘信息：

```
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)
#
```

在这里已经自动将 U 盘挂载到/media/sda4 目录下。

- 3、查看 U 盘里的内容。

```
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   ??????AX?
```

- 4、可以通过创建、拷贝、删除文件来测试 U 盘的读写。
- 5、用相同的方法来测试 2 个 host usb 口，测试完毕后拔出 U 盘，打印信息如下：



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

### 设备节点:

U 盘: /dev/sda

U 盘第一分区: /dev/sda1

若有多个分区, 依次类推, 第 n 个分区对应/dev/sdan。

### 测试源码:

测试命令: fdisk

### 驱动代码:

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

### 内核对应选项:

```
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
```

### 可能遇到的问题:

插入 U 盘后, 无任何打印信息或者能识别但出现读写错误, 可能 U 盘损坏, 可以在 PC 机上格式化之后再试一下, 也可以换一个 U 盘测试一下。

## 2.11、SD 卡测试

支持 3 种格式：fat32、exFAT、NTFS

QY-IMX6S 主板提供了 1 路 SD 卡接口（J14）可供用户使用。

### 测试原理：

板载 SD 卡接口支持热插拔，将 SD 卡插入后，系统会识别该 SD 卡，并打印出 SD 卡相关信息。

在/dev 目录下生成该设备节点及分区节点，之后系统会自动将所有分区挂载到/media/ 目录下，通过读写对应目录下文件，可判断该接口是否正常。

### 测试步骤和结果：

以下测试步骤以只有一个分区的 SD 卡为例，若有多个分区，则测试方法类似。插入 SD 卡产生的设备节点为/dev/mmcblk0，分区 n 对应的分区设备节点为/dev/mmcblk0pn

1、在这里插入一张金斯顿 8G 的 SD 卡，打印信息如下：

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

如上图所示，显示 SD 的卡一些基本信息，这里的设备节点为 mmcblk1，分区为 p1。

2、用 fdisk 命令来查看 SD 的信息

```
fdisk -l /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
#
```

这里已经自动将 SD 卡挂载到/media/mmcblk0p1 目录下。

3、查看 SD 卡里的内容。

```
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、可以通过创建、拷贝、删除文件来测试 SD 卡的读写。

5、拔出 SD 卡，打印信息如下：

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

### 测试源码：

测试工具：fdisk

### 驱动代码：

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

### 内核对应选项：

有任何技术问题或需要帮助，请联系：[supports@qiyangtech.com](mailto:supports@qiyangtech.com)

第 34 页 共 53 页

购买产品，请联系销售：[sales@qiyangtech.com](mailto:sales@qiyangtech.com)

更多信息请访问：<http://www.qiytech.com>

©2013 Qiyangtech 版权所有

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

### 可能遇到的问题:

- ①、插入 SD 卡后，无任何打印信息或者能识别但出现读写错误，可能 SD 卡损坏，可以用读卡器外接在 PC 机上格式化之后再试一下，也可以换一个 SD 卡测试一下。
- ②、插入 SD 卡后能识别但提示写保护，请确认 SD 卡上的硬件写保护是不是拨到了 lock 设置。

## 2.12、SATA 测试

### 测试步骤和结果:

- 1、将主板上的 SATA 接口 J27 与硬盘相连，并给硬盘供电接口 J28 供电 12V 和 5V。
- 2、板子上电，用命令来查看硬盘的信息

**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/mmcbk0: 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/mmcbk0p1          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/mmcbk0p1         3795720    322776  3280128   9% /media/mmcbk0p1
/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
    
```

这里已经自动将硬盘挂载到/media/sda1 目录下，可以直接查看硬盘里的内容。

**ls /media/sda1**

```

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

可以通过创建、拷贝、删除文件来测试 SD 卡的读写。

- 3、硬盘读写测试:

**hdparm -t /dev/sda1**

如果能测出硬盘的读写速度，则 SATA 接口工作正常

```

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

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



## 2.13、HDMI 测试

主板 J15 接口为 HDMI 接口，支持 2 种分辨率：1920x1080 和 1280X720

上电后，在 u-boot 命令行中输入

如果使用 1920x1080 分辨率，上电后在 u-boot 命令行中输入

```
set bootargs_mmc 'setenv bootargs ${bootargs} root=/dev/mmcblk0p1 rootwait
video=mxcfb0:dev=hdmi,1920x1080M@60,if=RGB24 video=mxcfb1:off video=mxcfb2:off
video=mxcfb3:off video=mxcfb4:off'
```

如果使用 1280X720 分辨率，上电后在 u-boot 命令行中输入

```
set bootargs_mmc 'setenv bootargs ${bootargs} root=/dev/mmcblk0p1 rootwait
video=mxcfb0:dev=hdmi,1280X720M@60,if=RGB24 video=mxcfb1:off video=mxcfb2:off
video=mxcfb3:off video=mxcfb4:off'
```

保存并启动系统

```
saveenv;boot
```

### 测试原理：

在 uboot 中设置帧缓存设备和启用 HDMI 驱动，设置分辨率。启动开发板，视频输出为 HDMI 输出

### 测试步骤和结果：

HDMI 数据线连接到板子 J15。

在 HDMI 显示器上显示输出信息，可以修改分辨率、输出格式适应不同的屏幕。

## 2.14、LCD 显示与 VGA 测试

QY-IMX6S 主板提供了 1 路 LCD TFT 显示接口 (J21)、1 路 VGA 接口 (J24)、2 路 LVDS 接口 (J16 和 J17)。当前内核支持的分辨率分别为 LCD 支持 480x272、640x480、800x480、800x600, VGA 支持 1280\*1024、1024x768。

4.3 寸 LCD 触摸屏, 分辨率为 480x272 (型号: QY-AT043TN24), 上电后在 u-boot 命令行中输入:

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

5.6 寸 LCD 触摸屏, 分辨率为 640x480 (型号: QY-AT056TN53), 上电后在 u-boot 命令行中输入:

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

7 寸 LCD 触摸屏, 分辨率为 800x480 (型号: QY-AT070TN83), 上电后在 u-boot 命令行中输入:

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

8 寸 LCD 触摸屏, 分辨率为 800x600 (型号: QY-AT080TN52), 上电后在 u-boot 命令行中输入:

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

VGA 分辨率 1024x768, 上电后在 u-boot 命令行中输入:

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

VGA 分辨率 1280x1024, 上电后在 u-boot 命令行中输入:

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

保存并启动系统

```
saveenv;boot
```

### 测试原理:

系统启动后, LCD 显示屏或 VGA 将显示内核自带的企鹅图像。

### 测试步骤和结果:

测试 LCD 屏时，数据线连接到板子 J21；测试 VGA 屏时，数据线连接到 J24。

请确保配置的内核分辨率和当前连接的 LCD 或者 VGA 要求的分辨率一致。

开发板上电，系统启动之后，LCD 或者 VGA 显示器的左上角显示如下图片：



单核显示 1 个 logo，双核 2 个，4 核 4 个。

可以查看该 logo 图片颜色有没失真抖动，来确认显示是否正常，也可以按照本章 2.9 小节介绍的运行 qt 程序来测试 LCD 或者 VGA 显示。

### 测试代码：

无。

### 驱动代码：

linux-3.0.101/drivers/video/mxc/mxc\_lcdif.c

linux-3.0.101/drivers/video/mxc/mxcfb\_hx8369\_wvga.c

### 内核对应选项：

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

### 可能遇到的问题：

- ①、LCD 显示器没有图像，请确定是否按照如上内核选项选中相关项。
- ②、显示图像位置大小等与 LCD 显示器不匹配，请确认当前内核分辨率与当前使用的 LCD 显示器大小是否一致。



## 2.15、LVDS 测试

2 路 LVDS 接口 (J16 和 J17)。支持的型号 QY-HJ070NA 的 7 寸屏分辨率 1024x600  
上电后在 u-boot 命令行中输入

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

```
saveenv;boot
```

载入内核

### 测试原理:

在 uboot 中设置帧缓存设备和启用 HDMI 驱动, 设置分辨率。启动开发板, 视频输出为 HDMI 输出, 系统启动后, 显示内核自带的企鹅图像。

### 测试步骤和结果:

LVDS 屏的显示信号线连接到板子 J16, 触摸信号线连接到板子 J19, 背光电源线连接到板子 J18。

请确保配置的内核分辨率和当前连接的屏要求的分辨率一致。

开发板上电, 系统启动之后, 屏显示器的左上角显示如下图片:



单核显示 1 个 logo, 双核 2 个, 4 核 4 个。

可以查看该 logo 图片颜色有没失真抖动, 来确认显示是否正常, 也可以按照本章 2.9 小节介绍的运行 qt 程序来测试屏显示

## 2.16、QT 测试

QY-IMX6S 主板标配文件系统带有 4.8.4 的 QT 库，以及本司的 demo 程序，该程序将带你走进 QT 的世界。

系统默认同时支持 usb 接口鼠标与触摸操作。

### 测试原理：

执行 Imx6\_qt\_test 程序，将在 LCD 屏幕上显示 QT 界面，通过触摸屏或者鼠标可移动光标与 QT 进行交互。

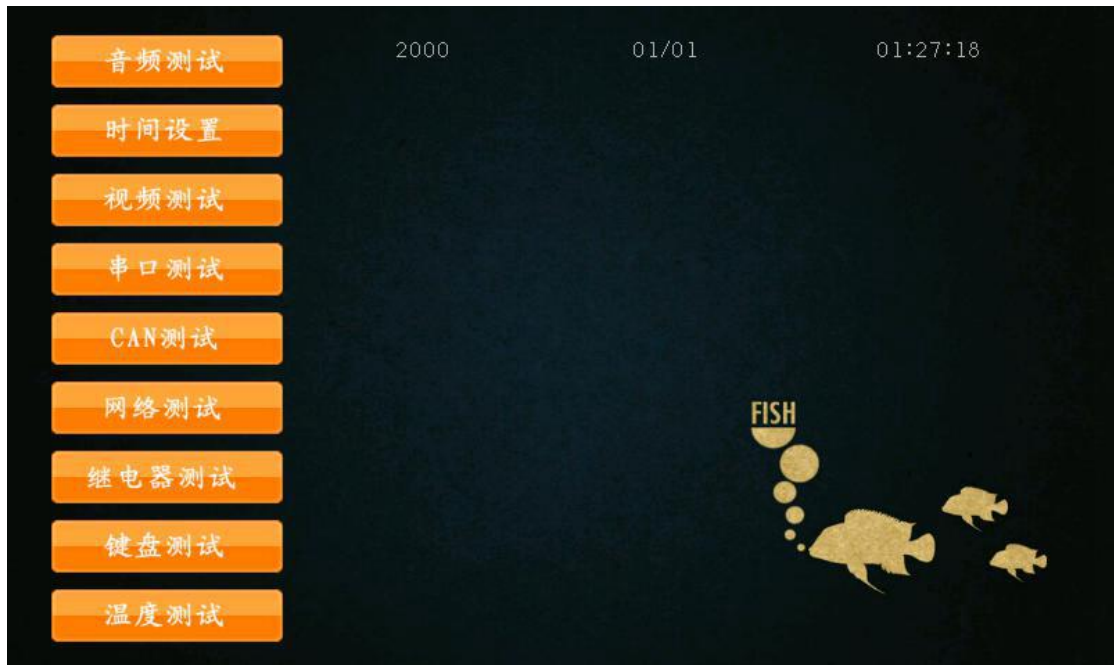
### 测试步骤和结果：

1、上电前接上 usb 接口鼠标，新的触摸屏需要校准一次，校准请参考 [2.17 节](#)。

2、运行 QT 程序

```
./Imx6_qt_test -qws
```

运行之后系统开始加载显示 QT 程序界面，如下图：



可以通过触摸屏或 USB 鼠标进行 QT 界面操作。

### 设备树文件：

无。

### 驱动代码：

无。

### 内核对应选项：

无。

### 可能遇到的问题：

请参考触摸屏及 LCD 显示部分问题解决方法。

## 2.17、触摸屏测试

触摸屏支持电阻触摸屏和电容触摸屏，默认电阻触摸屏和 LCD 显示屏配套，而电容触摸屏和 LVDS 显示屏配套，屏幕分辨率必须与 uboot 配置的 LCD 分辨率一致。

如果使用电阻触摸屏，按照上面的[第 2.14 节](#)连线，如果使用电容触摸屏，请按照上面的[第 2.15 节](#)连线。

### 测试原理：

通过 tslib 触摸屏测试工具，进行触摸屏校准，校准后使用测试工具进行拖拽或画线，可看到光标将显示在当前触摸点的附近，并随触摸位置的移动而移动。

### 测试步骤和结果：

- 1、默认的环境变量匹配电容触摸屏，如果使用电阻触摸屏，请修改一次：

```
vi /etc/QtEnv
```

把里面的 event2 改成 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"
```

环境变量立即生效：

```
source /etc/QtEnv
```

- 2、执行触摸校准程序 ts\_calibrate:

```
ts_calibrate
```

显示屏将在左上角显示十字图，串口终端将打印当前十字图的坐标点，点击该十字图后，该十字图将跳转到另一个角，直至四角和中心点完成触摸校准后结束。

- 3、校准完成后，执行 ts\_test 来测试触摸精度:

```
ts_test
```

可以点击拖拽按钮或者画线按钮进行测试，将看到鼠标或者线将沿着触摸点轨迹移动。

### 测试代码：

测试工具：ts\_calibrate, ts\_test

对应工具源码：光盘/测试源码/tslib

### 驱动代码：

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

### 内核对应选项：

```
Device Drivers --->
```

```
Input device support --->
```

```
[*] Touchscreens --->
```

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

### 可能遇到的问题：

- ①、若屏显示图像有异常，确定当前的屏分辨率配置是否与当前使用的屏相匹配。
- ②、点击触摸屏后，光标显示位置与点击位置有微小误差及抖动，该情况属于正常现象。

- ③、若有其他问题，请查看/etc/QtEnv 文件 tslib 部分环境变量，对比如下环境变量是否相同：

```
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 测试

### 测试原理:

板子支持 OV5640 摄像头，运行命令，测试摄像头是否正常。

主板上留出 camera 接口 J22，笔者测试的是 OV5640 摄像头，把 OV5640 与 J22 连接。

### 测试步骤和结果:

- 1、J22 接上 OV5640 摄像头
- 2、显示需要显示的屏，屏的操作参考 [2.13 节](#)、[2.14 节](#)、[2.15 节](#)
- 3、输入命令

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

可以在屏上显示 camera 的图形。

### 驱动代码:

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

### 内核对应选项:

```
<*> 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
```

### 可能遇到问题:

调试串口打印信息:

```
# 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

#

请检查以下几项：

- ①、接线是否正确
- ②、内核配置选项是否已经选上
- ③、内核是否下载正确

## 2.19、3G/4G 测试

QY-IMX6S 开发板引出 J29 miniPCI 接口，可以接 3G 模块及其可以接 3G 卡的 J30

支持全网通模块：SIM7100CE、SIM7600CE

支持移动 3G、联通 3G 模块：UC15

抽屉式卡槽，插上模块和模块支持的手机卡，上电后可以看到打印的信息。

```
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 fs1-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
```

在终端执行：

```
pppd call 3g4gnet &
```

可以打印出以下信息，说明联网成功。

```
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
```

访问外网，按 ctrl+c 退出：

```
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 测试

QY-IMX6S 主板引出 J12 或 J11 是接 WIF 接口的，模块支持 RTL8188CUS。在 J12 或 J11 端口插上 RTL8188CUS，在终端输出

`ifconfig -a`

会打印出 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)
    
```

修改配置文件/etc/wpa\_supplicant.conf，把下面的 ssid 和 psk 改成对应的用户名和密码

```

# 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"
}
    
```

输入 `sync`，板子重新上电，稍等几秒，输入

## 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)
```

如果显示有 `wlan0`，说明模块已经加载成功

测试外网

## 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
```

显示以上信息说明模块已经正常工作。

## 2.21、RS485 测试

在 QY-IMX6S 主板上，预留有 2 路 RS485，跟 RS232 复用。如果需要使用 RS485，需要修改硬件去掉复用的 RS232，并焊接 RS485 电路。

RS485 和硬件关系表：

| 串口   | 硬件位置                 | 设备节点        |
|------|----------------------|-------------|
| COM1 | J31 (A、B 对应 11、13 脚) | /dev/ttymx1 |
| COM2 | J31 (A、B 对应 12、14 脚) | /dev/ttymx2 |

### 测试原理：

测试程序可实现 RS485 发或收，用 2 块板子，把 RS485 对接，一块做 RS485 的收，另一块做 RS485 的发，观察数据是否正常。

### 测试步骤和结果：

下面以测试 COM1 的 RS485 为例。

- 1、用 2 块板子，以下分别称为 1 板和 2 板，把 J31 的 A、B 脚用杜邦线对应连接。
- 2、第一次上电启动到 uboot 时，按回车，输入下面

```
set bootargs_base 'setenv bootargs console=ttymx0,115200 imx6_feature=rs485:2'
saveenv;boot
```

- 3、系统启动后，1 板作为 RS485 的收，输入

```
/usr/test/rs485_test /dev/ttymx1 115200 0
```

- 4、2 板作为 RS485 的发，输入

```
/usr/test/rs485_test /dev/ttymx1 115200 1
```

- 5、观察 1 板的打印信息，可以看到每隔 1 秒打印下面信息：

```
receive 28 datas: "/dev/ttymx2" test string!
```

- 6、上面的第 3 和第 4 步骤的命令对调收发，观察 2 板是否有接收数据

### 三、测试小结

QY-IMX6S 主板的功能测试到此完毕，对于测试过程中出现的问题，可根据提供的测试源码进行排查。

## 浙江启扬智能科技有限公司

电话：0571-87858811 / 87858822

传真：0571-89935912

技术支持：0571-87858811 转 805

E-MAIL: [supports@qiyangtech.com](mailto:supports@qiyangtech.com)

网址： <http://www.qiytech.com>

地址：杭州市西湖科技园西园一路 8 号 3A 幢 5 层

邮编：310012