

**IAC-IMX6-CM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

杭州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](#_Toc481744794)

[前言 4](#_Toc481744795)

[一、准备工作 5](#_Toc481744796)

[二、主板测试 6](#_Toc481744797)

[2.1、蜂鸣器测试 6](#_Toc481744798)

[2.2、RTC测试 8](#_Toc481744799)

[2.3、看门狗测试 12](#_Toc481744800)

[2.4、GPIO测试 13](#_Toc481744801)

[2.5、串口测试 17](#_Toc481744802)

[2.6、SPI测试 23](#_Toc481744803)

[2.7、CAN测试 25](#_Toc481744804)

[2.8、音频、视频测试 28](#_Toc481744805)

[2.9、网口测试 34](#_Toc481744806)

[2.10、USB测试 36](#_Toc481744807)

[2.11、SD卡测试 39](#_Toc481744808)

[2.12、SATA测试 41](#_Toc481744809)

[2.13、HDMI测试 43](#_Toc481744810)

[2.14、LVDS测试 44](#_Toc481744811)

[2.16、QT测试 45](#_Toc481744812)

[2.17、触摸屏测试 47](#_Toc481744813)

[2.18、CAMERA测试 49](#_Toc481744814)

[2.19、3G/4G测试 51](#_Toc481744815)

[2.20、wifi测试 53](#_Toc481744816)

[2.21、RS485测试 56](#_Toc481744817)

[三、测试小结 57](#_Toc481744818)

# 版本说明

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 版本 | 硬件平台 | 描述 | 日期 | 修订人 |
| 1.0 | 核心板： IAC-IMX6-CM-v2\_03 底板： IMX6\_MB\_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。**

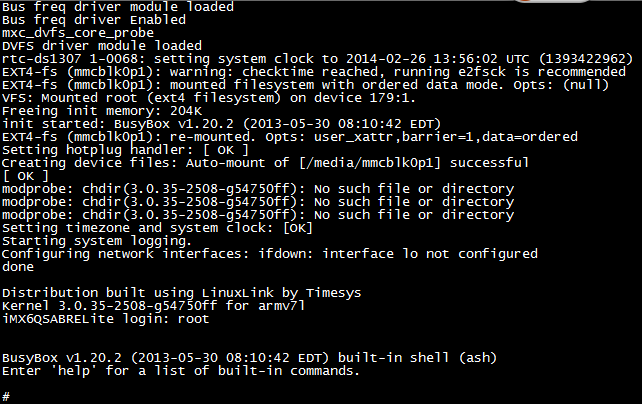
**使用之前请务必仔细阅读：**

**《IAC-IMX6-Kit 硬件说明书.pdf》和《IAC-IMX6-Kit Linux 用户手册.pdf》。**

# 一、准备工作

测试之前请阅读《IAC-IMX6-Kit 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：

 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



# 二、主板测试

## 2.1、蜂鸣器测试

IAC-IMX6-KIT主板使用GPIO 6\_31控制主板上的蜂鸣器。

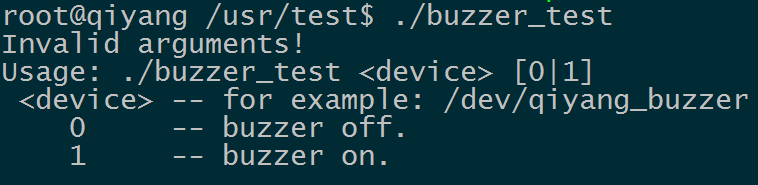
设置为低电平时，蜂鸣器不工作；设置为高电平时蜂鸣器鸣响。

**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



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：**

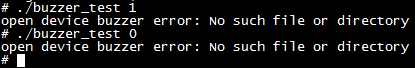
Device Drivers --->

Misc devices --->

<\*> BUZZER FOR QIYANG IMX6 BOARD

**可能遇到的问题：**

调试串口打印信息：



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

IAC-IMX6-KIT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：**

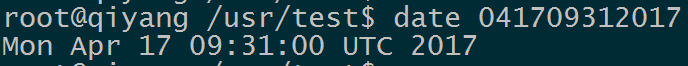
1、在板子上执行date命令，可查看到当前系统时钟。

date



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。

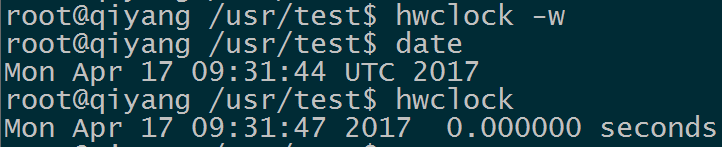
date 041709312017 /\*月日时分年\*/



3、用hwclock命令把系统时间写入硬件时钟芯片。

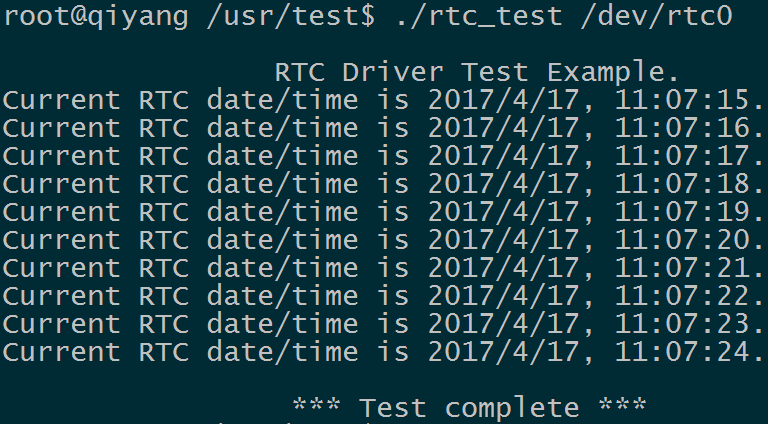
hwclock -w

4、分别用date和hwclock命令来查看系统和硬件时钟。



5、设置成功之后，执行rtc\_test测试程序。

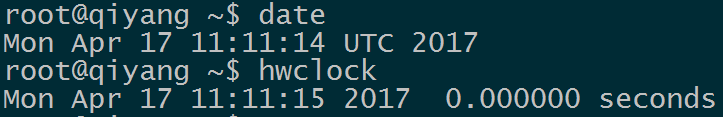
./rtc\_test /dev/rtc0



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，过一会再上电，再次查看系统和硬件时钟，看时间有没保存，走时是否精准。



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：



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\_5\_20使能看门狗，GPIO\_5\_18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：

J16上的GPIO1\_4、GPIO1\_7、GPIO1\_8、GPIO1\_9、GPIO7\_11、GPIO7\_12、GPIO7\_13、GPIO4\_5

J27上的GPIO4\_24、GPIO4\_26、GPIO4\_28、GPIO4\_30、GPIO5\_5、GPIO5\_7、GPIO5\_9、GPIO5\_11

**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！

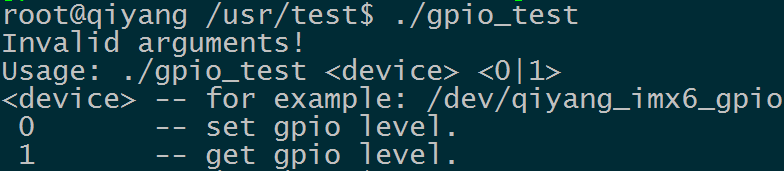
gpio\_test 1直接读取外接电平信号，用户可通过读取的电平值与连接的电平值做比较，确认gpio是否正常。

**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：



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、J16和J27的gpio无外接信号，执行

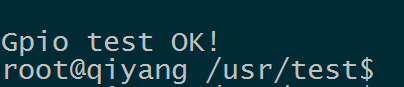
./gpio\_test /dev/qiyang\_imx6\_gpio 0



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：



再按回车，提示测试OK：



3、外部接3.3V到管脚（管脚内部弱上拉）

./gpio\_test /dev/qiyang\_imx6\_gpio 1



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IAC-IMX6主板上，共有3路串口，其中1路被用来作为调试串口, 其他2路可以当作RS232或RS485使用

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） | J7（收、发、GND对应1、2、3脚） | /dev/ttymxc0 |
| COM1 | J6（收、发、GND对应1  、2、5脚） | /dev/ttymxc3 |
| COM2 | J6（收、发、GND对应3、4、6脚） | /dev/ttymxc4 |

**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/ttymxcx" 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：**

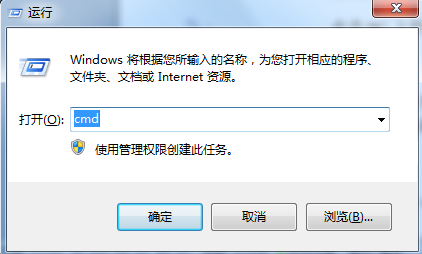
串口测试时PC机需要用到两个串口：

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。

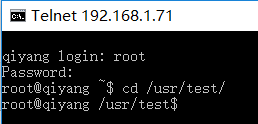
开发板默认IP地址为192.168.1.71点击PC上的开始按钮，选择“运行”，输入cmd，点击“确定”。

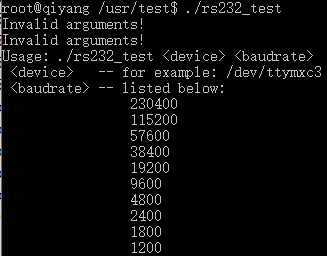


在运行的窗口里面输入telent 192.168.1.71



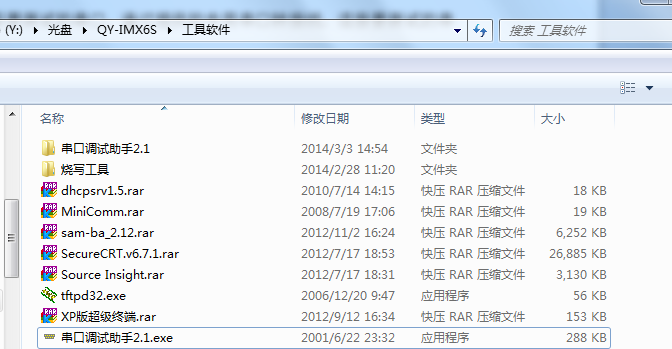
输入用户名root和密码root进入控制台，进入/usr/test测试目录



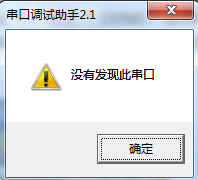


根据串口和硬件关系表，选择要测试的串口，通过提供的专用串口转接线，连接要测试的串口到PC机串口上。

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。

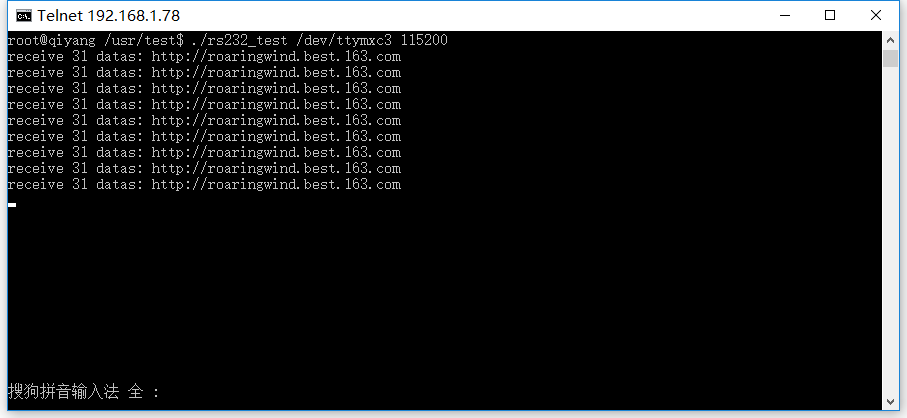
这里以COM1为例作介绍，其他串口测试方法一样。

./rs232\_test /dev/ttymxc3 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。



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/ttymxc3

/dev/ttymxc4

**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/ttymxc3 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ttymxc3、ttymxc4设置节点是否存在

⑤、内核配置选项有没选上

## 2.6、SPI测试

**测试原理：**

IAC-IMX6主板上有一路spi，连接到nor flash（N25Q032）和触摸。这里测试nor flash，系统把nor flash分2个分区（字符设备：mtd0、mtd1，块设备mtdblock0、mtdblock1），大小分别是1MB和3MB，通过挂载方式，实现读写数据操作。

**测试步骤和结果：**

1、挂载分区0的块设备

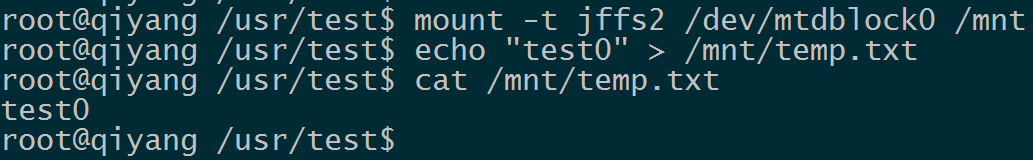
mount -t jffs2 /dev/mtdblock0 /mnt

2、写数据

echo "test0" > /mnt/temp.txt

3、读数据

cat /mnt/temp.txt



4、卸载

umount /mnt

5、分区1操作按照上面步骤

**设备节点：**

/dev/mtd0

/dev/mtd1

/dev/mtdblock0

/dev/mtdblock1

**驱动代码：**

linux-3.0.101/drivers/mtd/devices/m25p80.c

**内核对应选项:**

Device Drivers --->

SPI support --->

<\*> User mode SPI device driver support

Device Drivers --->

<\*> Memory Technology Device (MTD) support --->

Self-contained MTD device drivers --->

<\*> Support most SPI Flash chips (AT26DF, M25P, W25X, ...)

**可能遇到的问题：**

挂载失败，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

## 2.7、CAN测试

**测试原理：**

IAC-IMX6主板引出了两路CAN，文件系统中提供了测试CAN的方法，使用CAN工具进行测试。

can0的H、L分别在J4的1、2脚

can1的H、L分别在J4的3、4脚

**测试步骤和结果：**

以下以can0测试为例

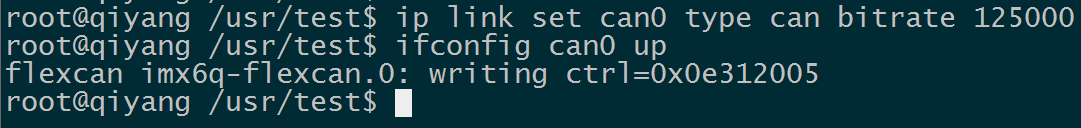
1、将两块IMX6的板子can0的H、L对应连接起来

2、上电后，2块板子配置can0

ip link set can0 type can bitrate 125000

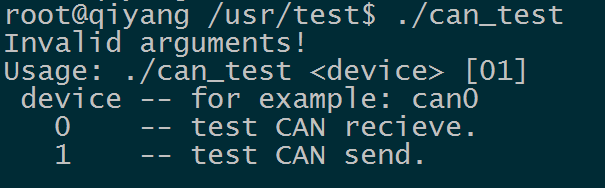
ifconfig can0 up

调试串口打印信息：



3、使用CAN测试程序进行测试：

./can\_test

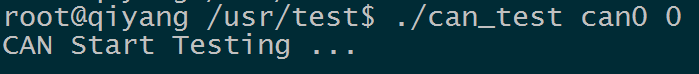


说明： ①、can\_test <device> 0 把CAN设置成接收数据 。

②、can\_test <device> 1把CAN设置成发送数据。

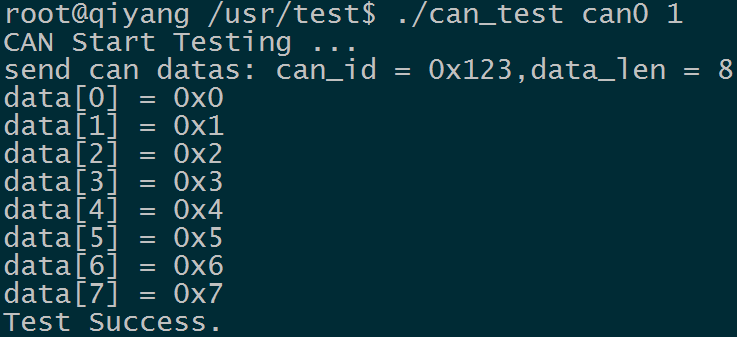
这里把第1块主板端的CAN当作接收端，在串口终端输入命令：

./can\_test can0 0



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



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

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：

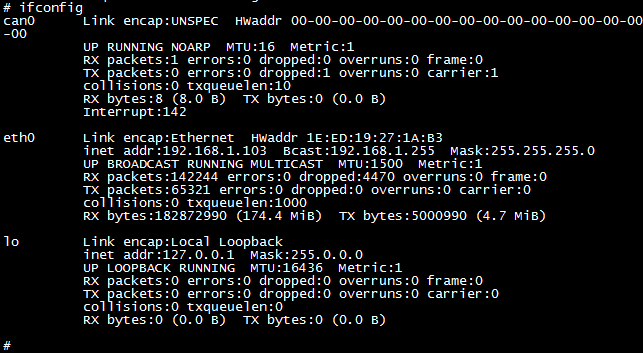
read can datas failed.

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



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

IAC-IMX6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：**

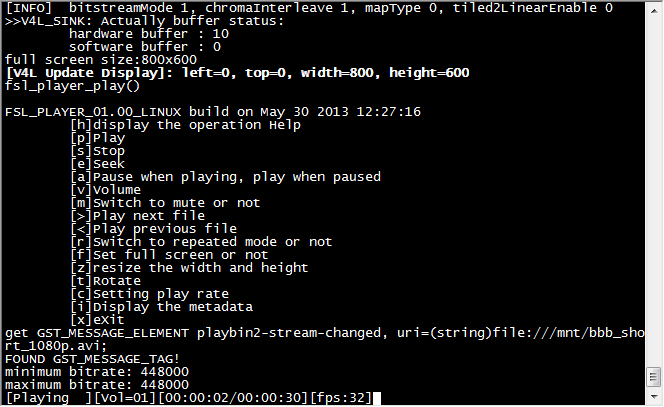
可通过gplay命令，播放音视频。

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J18接口。

**测试步骤和结果：**

1、在开发板目录/usr/ test下，执行如下命令进行播放：

gplay bbb\_short\_1080p.avi

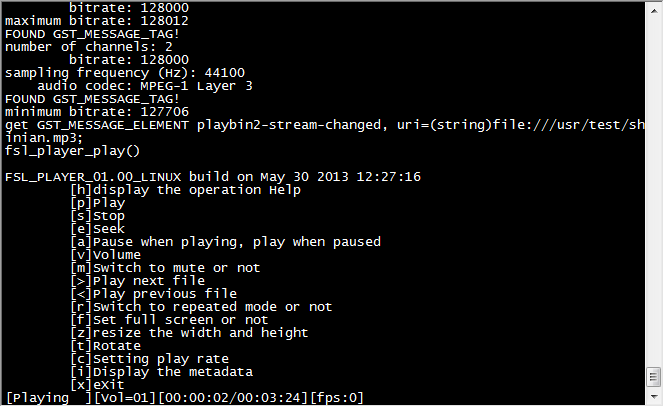


执行之后，在显示屏上就能看到1080P高清视频demo的播放了。

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



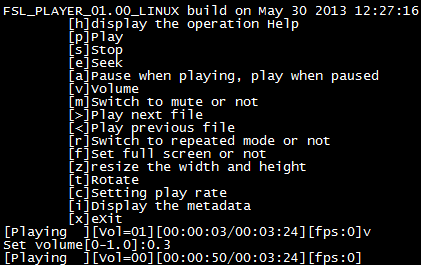
在音频输出接口上就能听到mp3音乐了。

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：



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J17，然后在终端中输入以下命令即可录音：

arecord -d 10 -D plughw:0 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[2.13](#_2.13、HDMI测试)和[2.14](#_2.14、LVDS测试)小节。

②、默认情况下请使用大于当前影片分辨率屏，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J8

1、用一根网线连接板子J8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 （板子上电已经默认设置为这个）

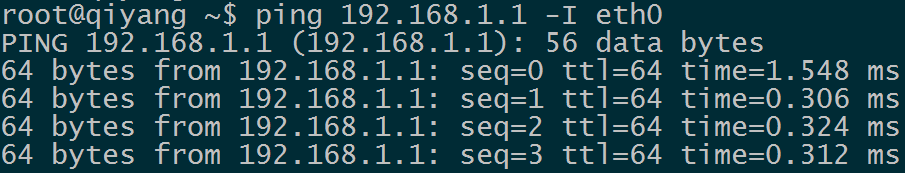
echo nameserver 114.114.114.114 > /etc/resolv.conf

route add default gw 192.168.1.1 dev eth0

3、测试内网，输入：

ping 192.168.1.1 -I eth0

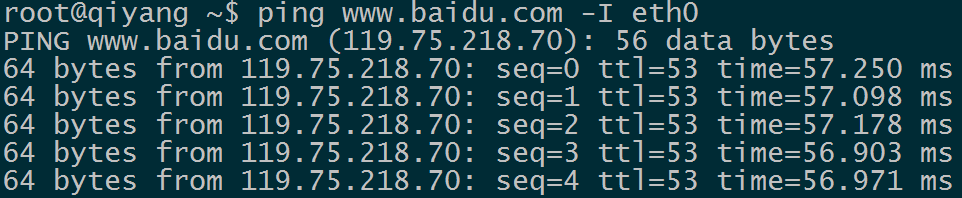
如果打印下面信息表明正确，输入ctrl+c退出



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



**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IAC-IMX6主板上，共有5路usb口：

①、1路（J13）作为device口, 用于usb方式下载固件程序；

②、1路J14外接wifi模块；

③、1路集成到miniPCIe接口上（U19）；

④、2路（J15）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。

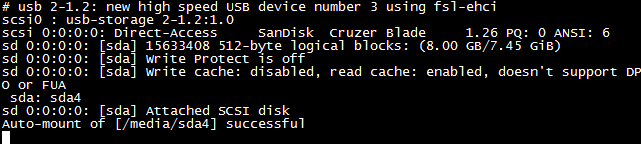
识别后在/dev目录下生成该设备节点/dev/sda及分区节点/dev/sda1（若有多个分区，则数字部分依次增加）。

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。

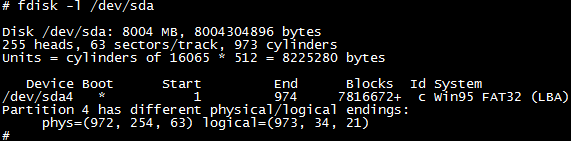
1、将正常使用的U盘插入J15口，调试串口打印如下信息：



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、用fdisk命令来查看SD的信息

fdisk -l /dev/sda



在这里已经自动将U盘挂载到/media/sda4目录下，可以直接查看U盘里的内容。

3、查看U盘里的内容。

ls /media/sda4



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：



**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

IAC-IMX6主板提供了1路SD卡接口（J20）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

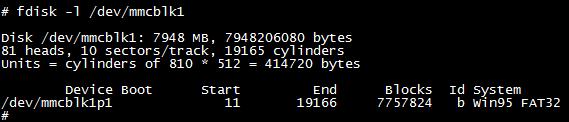
在这里插入一张金斯顿8G的SD卡，打印信息如下：



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。

也可以用fdisk命令来查看SD的信息

fdisk -l /dev/mmcblk1



这里已经自动将SD卡挂载到/media/mmcblk0p1目录下，可以直接查看SD卡里的内容。

ls /media/mmcblk1p1



可以通过创建、拷贝、删除文件来测试SD卡的读写。

拔出SD卡，打印信息如下：



**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:**

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。

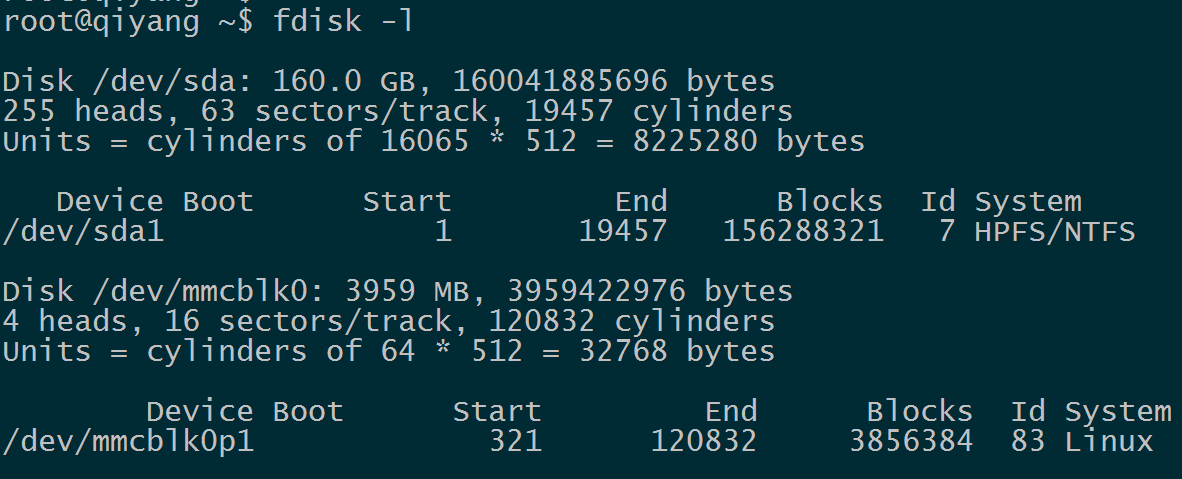
## 2.12、SATA测试

**测试步骤和结果：**

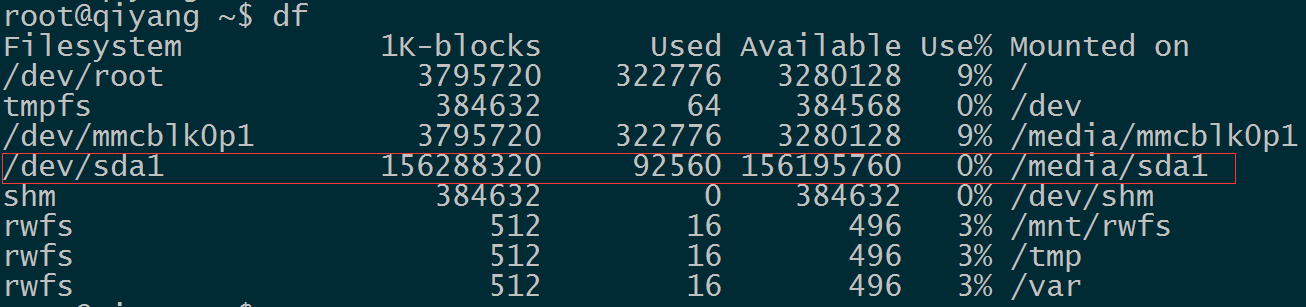
1、将主板上的SATA接口J28与硬盘相连，并给硬盘供电接口J30供电12V和5V。

2、板子上电，用命令来查看硬盘的信息

fdisk -l

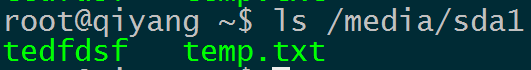


df



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

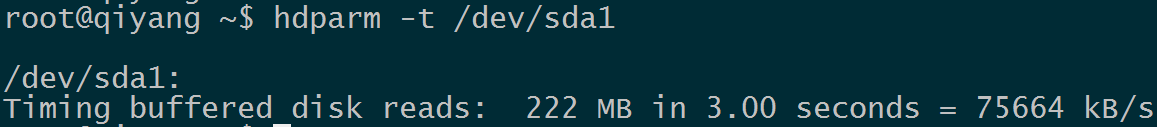


可以通过创建、拷贝、删除文件来测试SD卡的读写。

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



## 2.13、HDMI测试

主板J23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

setenv bootargs\_mmc 'setenv bootargs ${bootargs} root=${mmcroot} 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

setenv bootargs\_mmc 'setenv bootargs ${bootargs} root=${mmcroot} 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J23。

在HDMI显示器上显示输出信息，可以修改分别率、输出格式适应不同的屏幕。

## 2.14、LVDS测试

2路LVDS接口（J9和J10）。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 ${bootargs} root=${mmcroot} rootwait rw video=mxcfb0:dev=ldb,LDB-WSVGA,if=RGB666 ldb=dul0 fec\_mac=DC:07:C1:01:ff:f7'

saveenv

boot

载入内核

800x600分辨率：

上电后在u-boot命令行中输入

setenv bootargs\_mmc 'setenv bootargs ${bootargs} root=${mmcroot} rootwait rw video=mxcfb0:dev=ldb,800x600M@60,if=RGB666 ldb=sin0 fec\_mac=DC:07:C1:01:ff:f7'

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J9，触摸信号线连接到板子J12，背光电源线连接到板子J11。

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

## 2.15、QT测试

IAC-IMX6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。

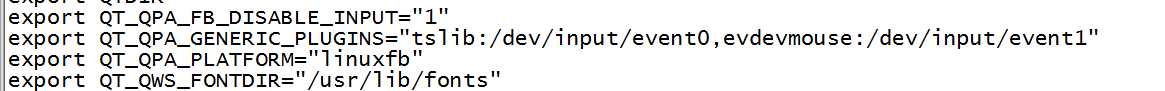
**测试原理：**

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：**

系统默认同时支持鼠标与触摸操作，具体的环境变量都在/etc/profile中设置

vi /etc/profile.d/QtEnv.sh

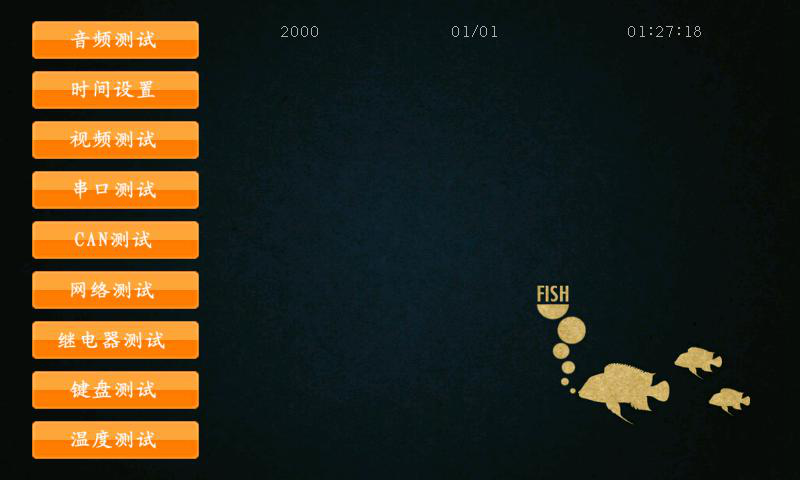


若使用触摸操作，在运行QT程序之前请先使用ts\_calibrate对触摸屏进行校准，校准准确之后再执行Imx6\_qt\_test程序。

cd /usr/test/

./Imx6\_qt\_test

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。

## 2.16、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。

如果使用电容触摸屏，请按照上面的[第2.14节](#_2.14、LCD显示与VGA测试_*)连线。

**测试原理：**

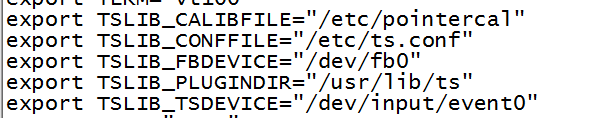
通过tslib触摸屏测试工具，进行触摸屏校准，校准后使用测试工具进行拖拽或画线，可看到光标将显示在当前触摸点的附近，并随触摸位置的移动而移动。

**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/profile.d/QtEnv.sh

把里面的event2改成event1



环境变量立即生效：

source /etc/profile.d/QtEnv.sh

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纽进行测试，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 --->

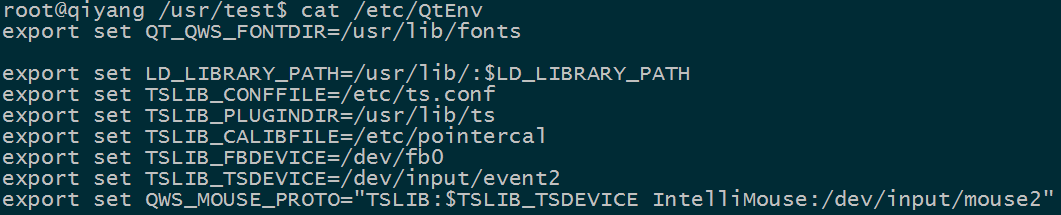
<\*> ADS7846/TSC2046/AD7873 and AD(S)7843 based touchscreens

**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：



## 2.17、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J25，笔者测试的是OV5640摄像头，把OV5640与J25连接。

**测试步骤和结果：**

1、J25接上OV5640摄像头

2、显示需要显示的屏，屏的操作参考[2.13节](#_2.13、HDMI测试)、[2.14节](#_2.14、LCD显示与VGA测试_*)

3、输入命令

/unit\_tests/mxc\_v4l2\_overlay.out -iw 1024 -ih 600 -ow 1024 -oh 600 -di /dev/video1

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

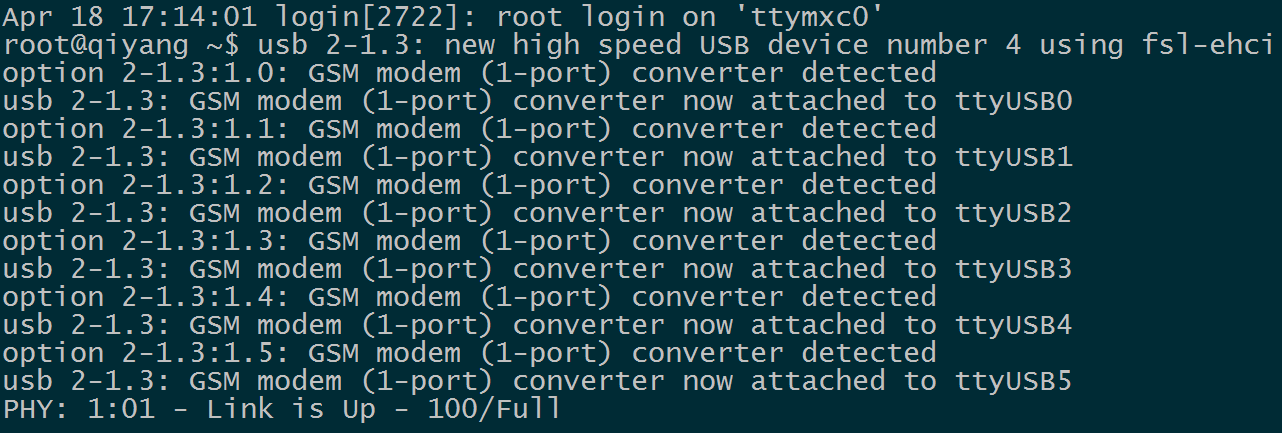
## 2.18、3G/4G测试

IAC-IMX6-Kit开发板引出U30 miniPCI接口，可以接3G模块及其可以接3G卡的u20

支持全网通模块：SIM7100CE、SIM7600CE

支持移动3G、联通3G模块：UC15

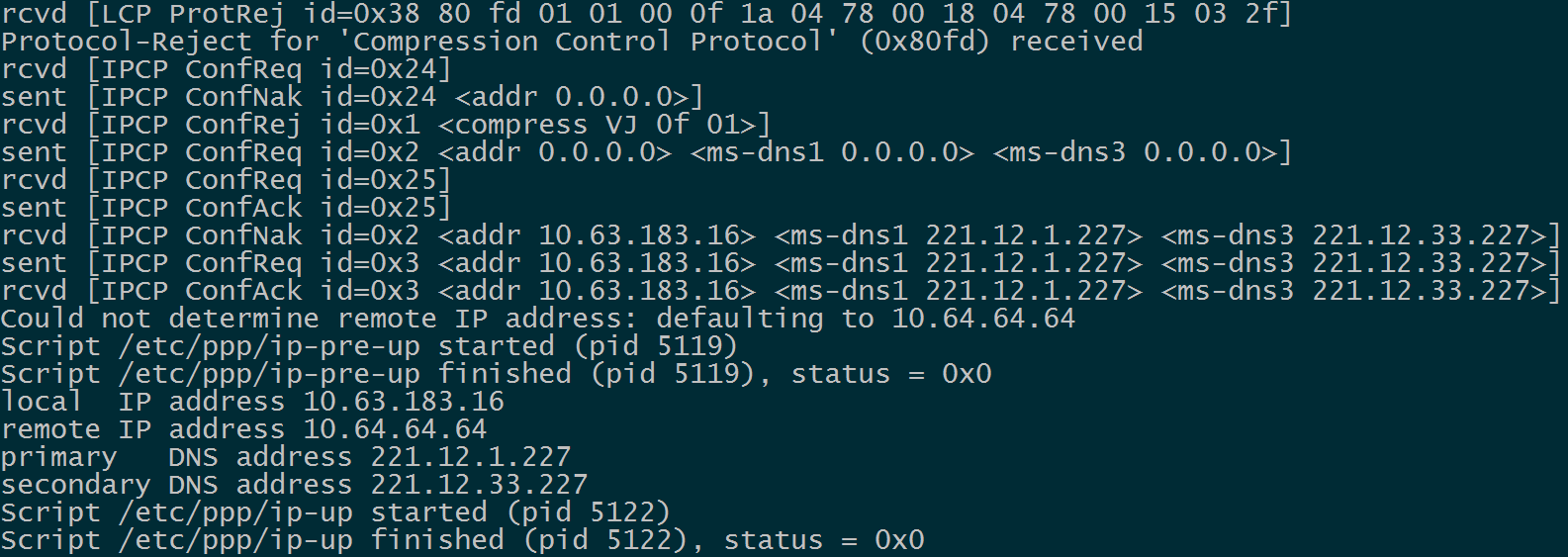
抽屉式卡槽，插上模块和模块支持的手机卡，上电，可以看到打印的信息。



在终端执行：

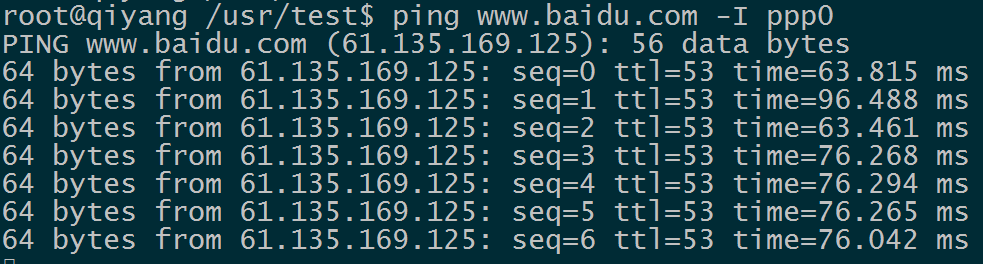
pppd call 3g4gnet &

可以打印出以下信息，说明联网成功。



访问外网，按ctrl+c退出：

ping www.baidu.com -I ppp0

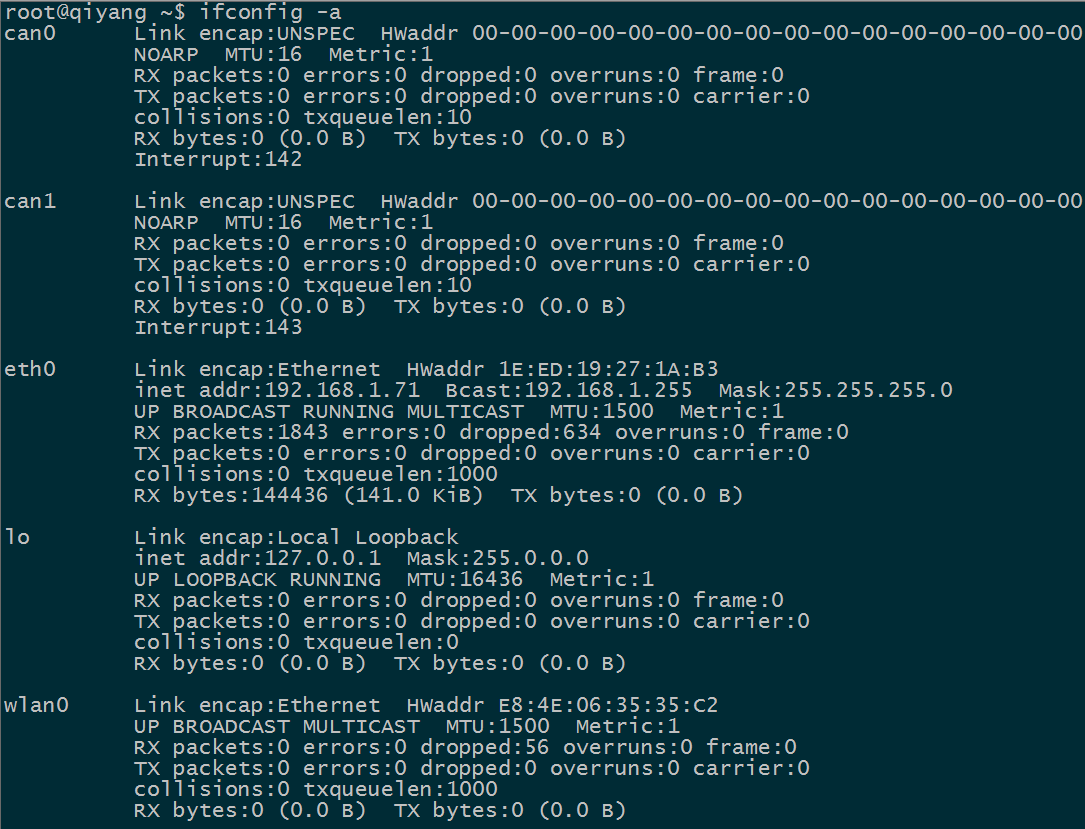


## 2.19、wifi测试

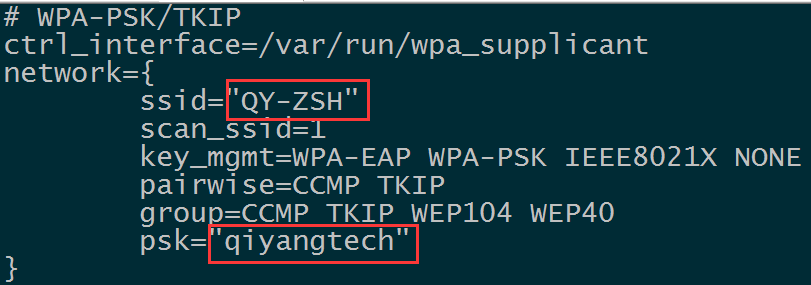
IAC-IMX6 板子引出J14或J15 是接WIF接口的，模块支持RTL8188CUS。在J14或J15端口插上RTL8188CUS，在终端输出

ifconfig -a

会打印出wlan0

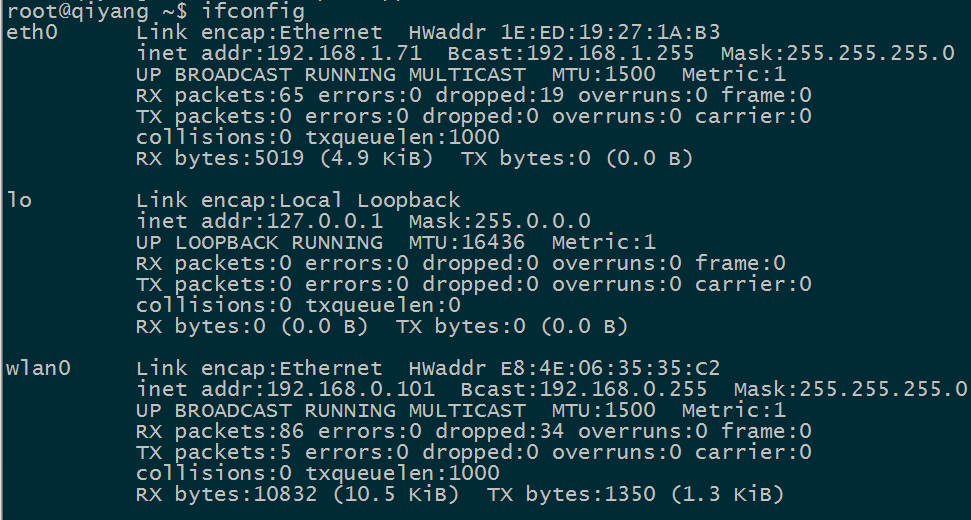


修改配置文件/etc/wpa\_supplicant.conf，把下面的ssid和psk改成对应的用户名和密码



输入sync，板子重新上电，稍等几秒，输入

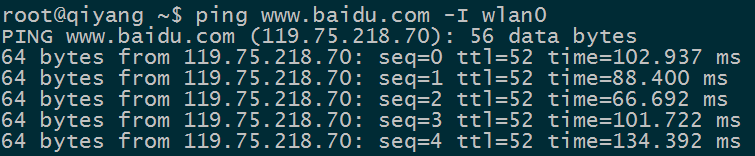
ifconfig



如果显示有wlan0，说明模块已经加载成功

关闭以太网：ifconfig eth0 down，然后测试外网

ping www.baidu.com -I wlan0



显示以上信息说明模块已经正常工作。

## 2.20、RS485测试

在IAC-IMX6板子上，预留有2路RS485，跟RS232复用。RS485和RS232电路可以同时存在。

RS485和硬件关系表：

|  |  |  |
| --- | --- | --- |
| 串口 | 硬件位置 | 设备节点 |
| COM1 | J6（A、B对应7、8脚） | /dev/ttymxc3 |
| COM2 | J6（A、B对应9、10脚） | /dev/ttymxc4 |

**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J6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

setenv bootargs\_mmc 'setenv bootargs ${bootargs} root=${mmcroot} rootwait rw rs485=3,4'

saveenv;boot

3、系统启动后，1板作为RS485的收，输入

/usr/test/rs485\_test /dev/ttymxc3 115200 0

4、2板作为RS485的发，输入

/usr/test/rs485\_test /dev/ttymxc3 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/ttymxc2" 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

IAC-IMX6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。

**杭州启扬智能科技有限公司**

**电话：0571-87858811 / 87858822**

**传真：0571-89935912**

**支持：0571-89935913**

**E-MAIL：supports@qiyangtech.com**

**网址：** [**http://www.qiytech.com**](http://www.qiyangtech.com)

**地址：杭州市西湖科技园西园一路8号3A幢5层**

**邮编：310012**