



## QY-9263S Linux 测试手册

版本号 v2.8

2011-11-20

杭州启扬智能有限公司版权所有

**QIYANG TECHNOLOGY Co., Ltd**

**Copyright Reserved**

## 目 录

前言.....	1
一、准备工作.....	2
1.1 配置.....	2
1.2 主板要求.....	2
1.3 启动主板.....	2
二、主板测试.....	3
1.1 蜂鸣器测试.....	3
1.2 GPIO 测试.....	3
1.3 定时器测试.....	5
1.4 PWM 测试.....	6
1.5 RTC 测试.....	6
1.6 串口测试.....	7
1.7 485 测试.....	8
1.8 8x8 键盘测试.....	9
1.9 USB 测试.....	10
1.10 音频测试.....	11
1.11 触摸屏测试.....	12
1.12 QT 测试.....	13
1.13 SPI 测试.....	14
1.14 PC104 测试.....	14
1.15 CAN 测试.....	15
附录.....	16

## 前言

欢迎使用杭州启扬智能科技有限公司产品 QY-9263S，本产品 Linux 部分包含 3 份手册：QY-9263S Linux 用户手册、QY-9263S 硬件说明手册以及 QY-9263S Linux 测试手册。硬件相关部分可以参考 QY-9263S 硬件说明手册，主板测试可以参考 QY-9263S Linux 测试手册。使用之前请仔细阅读 QY-9263S Linux 用户手册以及 QY-9263S 硬件说明手册！

## 公司简介

杭州启扬智能科技有限公司位于美丽的西子湖畔,是一家集研发、生产、销售为一体的高新技术产业。公司致力于成为嵌入式解决方案的专业提供商,为嵌入式应用领域客户提供软件开发工具和嵌入式系统完整解决方案。产品范围主要包括: Cirrus Logic EP93xx 系列 ARM9 主板、ATMEL AT91SAM926x 系列主板, FreeScale iMX 系列主板, TI Davinci 系列音/视频通用开发平台等等。可运行 Linux2.4/2.6、WinCE5.0/6.0 操作系统, 并可根据客户需求开发各种功能组合的嵌入式硬件系统。应用领域涉及: 工业控制、数据采集、信息通讯、医疗设备、视频监控、车载娱乐等等。

客户的需求是公司发展的动力, 公司将不断完善自身, 与客户互助互惠, 共同发展。

电话: 0571-87858811, 87858822

传真: 0571-89935912

技术支持 E-MAIL: [support@qiyangtech.cn](mailto:support@qiyangtech.cn)

网址: <http://www.qiyangtech.com>

地址: 杭州市西湖科技园西园一路 8号 2幢 5层

邮编: 310012

# 一、准备工作

## 1.1 配置

装有 Linux 系统 ( ubuntu 或其它 Linux 发行版 )

串口连接：通过提供的串口线将开发板的调试串口(J4)与 PC 机的串口连接

网络连接：通过网线将开发板的以太网接口(J10)与 PC 机的网络接口连接

USB 连接：通过 USB 连接线将开发板的 USB Device(J8)与 PC 机的 USB 连接

LCD 连接：通过提供的双排线将 LCD 屏连接到主板上

## 1.2 主板要求

以下的测试用例都在主板的 Linux 环境下运行，请确认主板已经正常启动 Linux 系统。  
相关文档请参考《QY-9263S Linux 用户手册》。

## 1.3 启动主板

1、打开 PC 机上的超级终端，设置 PC 上相应的串口，选择波特率为[115200]，停止位为[1]位，数据位为[8]位，无奇偶校验和数据流控制

2、接上电源，启动主板，在终端上可以看到系统启动的信息，如图：

```
RomBOOT
>Start AT91Bootstrap...

U-Boot 1.3.4 Yxx_QY (Aug 25 2010 - 15:05:54)

DRAM: 64 MB
NAND: 128 MiB
DataFlash:AT45DB161
Nb pages: 4096
Page Size: 528
Size= 2162688 bytes
Logical address: 0xC0000000
Area 0: C0000000 to C0003FFF (RO) Bootstrap
Area 1: C0004000 to C0007FFF Environment
Area 2: C0008000 to C0037FFF (RO) U-Boot
```

3、进入/usr/test 目录

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

```
#ls
```

#### 4、进入 test 目录

```
#cd test
```

```
#ls
```

```
看 /usr/test # ls  
到 audio_test keypad_test pio_test rs485_test serial_test tc_test  
t buzzer_test pc104 pwm_test rtc_test spi test.wav
```

est 目录下有预先编译好的测试程序。运行各个测试程序可以对相关模块进行测试

## 二、主板测试

### 1.1 蜂鸣器测试

在/usr/test 目录下运行蜂鸣器测试程序 `buzzer_test #cd /usr/test`

```
#!/buzzer_test
```

测试结果：程序正常运行的话就可以听到主板上的蜂鸣器发出短暂的蜂鸣声

测试源码：测试源码/buzzer\_test/buzzer\_test.c

驱动代码：linux-2.6.30/driver/char/at91\_buzzer.c

内核对应选项：

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

```
Character devices --->
```

```
AT91_BUZZER
```

可能遇到的问题：

```
open buzzer error! -1
```

请检查 /dev目录下是否有 buzzer 节点，如果没有则可用以下命令来创建：

```
#cd /dev
```

```
#mknod buzzer c 200 0
```

如果是在 PC 主机上创建则需要注意是进入到主板文件系统的 dev 目录下而不是 /dev

如果 /dev 目录下有 buzzer 节点且设备号也是准确的话请检查内核配置中是否添加进了 AT91\_BUZZER 选项。

### 1.2 GPIO 测试

QY-9263S主板设计了 16路独立 IQ 用户接口为 J5(请参考 QY-9263S硬件说明书 )



在/usr/test 目录下运行 pio\_test 程序

```
#!/pio_test
```

测试结果：测试程序把 PIO\_D0 ~ PIO\_D7设置为输入 IO, PIO\_D8 ~ PIO\_D15设置为输出 IO。输出 IO的状态为 0, 1, 0, 1, 0, 1, 0, 1, 并连续读 100次输入 IO的状态, 默认输入 IO状态都为 1 (用户可以通过外部设备改变 IO的状态, 再观察测试程序的输出结果)

测试代码：example/pio\_test/pio\_test.c

驱动代码：linux-2.6.30/driver/char/at91\_pio.c

内核对应选项：

Device Drivers --->

Character devices --->

AT91\_GPIO

可能遇到的问题：

open PIO device error! -1

请检查 /dev目录下是否有 pio0节点, 如果没有则可用以下命令来创建：

```
#cd /dev
```

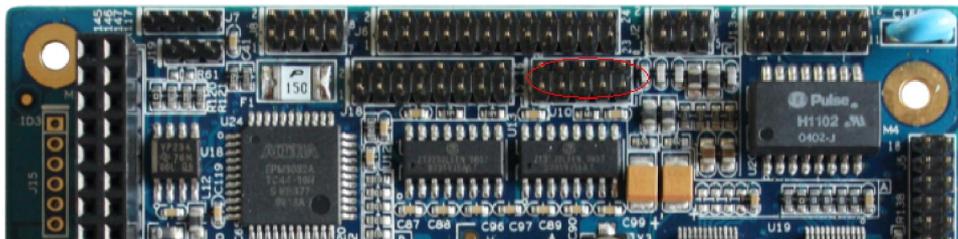
```
#mknod pio0 c 202 0
```

如果是在 PC 主机上创建则需要注意是进入到主板文件系统的 dev 目录下而不是 /dev

如果 /dev目录下有 pio0节点且设备号也是准确的话请检查内核配置中是否添加进了 AT91\_GPIO 选项。

## 1.3 定时器测试

AT91SAM9263 内部包括 3 个独立定时器。可作为外部脉冲计数和定时器使用。在 QY-9263S 主板上，我们设计了一路外部脉冲计数器（TC0）。用户接口为 J10 的 PIN2（TIOA0 具体请参考 QY-9263S 硬件说明手册）。



在 /usr/test 目录下运行 tc\_test 程序

```
#!/tc_test
```

测试结果：测试程序先通过 TC0 检测 10 次外部计数器，之后采用 TC2 进行 500ms 的定时器返回。

测试代码：测试源码 /tc\_test/tc\_test.c

驱动代码：linux-2.6.30/driver/char/at91\_tcb.c

内核对应选项：

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

```
Character devices --->
```

```
AT91_TC
```

可能遇到的问题：

```
open tc device error! -1
```

请检查 /dev 目录下是否有 tc0 tc1 tc2 节点，如果没有则可用以下命令来创建：

```
#cd /dev
```

```
#mknod tc0 c 201 0
```

```
#mknod tc1 c 201 1
```

```
#mknod tc2 c 201 2
```

如果是在 PC 主机上创建则需要注意是进入到主板文件系统的 dev 目录下而不是 /dev

如果 /dev 目录下有 pio0 节点且设备号也是准确的话请检查内核配置中是否添加进了 AT91\_TC 选项。

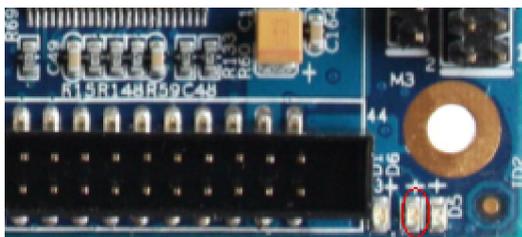
## 1.4 PWM 测试

AT91SAM9263 内部包括 4 路独立 PWM 输出。在 QY-9263S 主板上，包含了两路用户 PWM，分别为 PWM0 和 PWM1，在插件 J10 的 PIN6 和 PIN8。PWM2 被设计为板上 LED 的控制。

在/usr/test 目录下运行 pwm\_test 程序

```
# ./pwm_test
```

测试结果：程序测试 PWM2，可以看到主板上的蓝色 LED 会不断的闪烁。



测试代码：测试源码 /pwm\_test/pwm\_test.c

驱动代码：linux-2.6.30/driver/char/at91\_pwm.c

内核对应选项：

Device Drivers --->

Character devices --->

AT91\_PWM

## 1.5 RTC 测试

Linux2.6 内核已经包含对 RTC 的支持. 用户可以通过以下方式测试：

1、通过 date 命令设置当前时间

```
#date 091715482010 //设置当前时间为 2010-9-17 15:48:00
```

2、通过 hwclock 写入硬件 RTC

```
#hwclock -w
```

3、断电后重新启动系统, 用 date 命令可以查看当前时间

```
#date
```

4、如果要通过程序读 RTC 时间，可以参考 rtc\_test.c

运行/usr/test/rtc\_test

程序运行后，会连续读 10 秒钟的 RTC 时间并在串口打印。

测试代码：测试源码 /rtc\_test/rtc\_test.c

驱动代码：linux-2.6.30/driver/rtc/at91\_ds1338.c

内核对应选项：

Device Drivers --->

Real Time Clock --->

.....

可能遇到的问题：

运行 `hwclock -w` 命令的时候提示出错：

`hwclock: Could not access RTC: No such file or directory`

或者是运行 `rtc_test`的时候提示：

`/dev/rtc: No such file or directory`

请检查 `/dev`目录下是否有 `rtc`和 `rtc0`, `rtc`为 `rtc0`的软链接如果没有 `rtc`则用以下命令创建：

```
# cd /dev
```

```
# ln -s rtc0 rtc
```

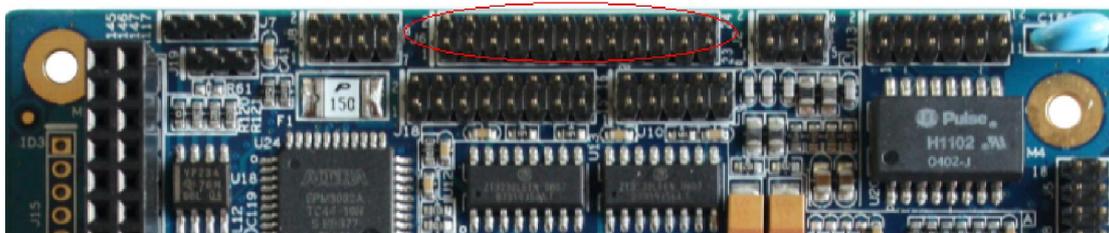
如果没有 `rtc0`则用以下命令创建：

```
# mknod rtc0 c 254 0
```

如果 `/dev`目录下有 `rtc`和 `rtc0`并正确则检查内核 `rtc`配置。

## 1.6 串口测试

在 QY-9263S主板上，共有 4路串口，主板上接口为 J6(具体请参考硬件说明书)，其中串口 0被用来做为调试串口，串口 1为 5线串口，串口 2和串口 3是三线串口并和 RS485复用。用户可以通过连接相关的接插件来决定采用 RS232还是 RS485



在 `/usr/test` 目录下运行 `serial_test` 程序

```
./serial_test
```

测试结果：除了调试串口外的其它串口每隔一秒发送一次数据（64 字节），将需要测试的串口分别接到 PC 机的串口，在 PC 接收到的数据如下：

```
0123456789 ; <=> ? @ ABCDEFGHIJKLMNOPQRSTUVWXYZ [\ ] ^ _ ` abcdefghijkl
0123456789 ; <=> ? @ ABCDEFGHIJKLMNOPQRSTUVWXYZ [\ ] ^ _ ` abcdefghijkl
0123456789 ; <=> ? @ ABCDEFGHIJKLMNOPQRSTUVWXYZ [\ ] ^ _ ` abcdefghijkl
0123456789 ; <=> ? @ ABCDEFGHIJKLMNOPQRSTUVWXYZ [\ ] ^ _ ` abcdefghijkl
0123456789 ; <=> ? @ ABCDEFGHIJKLMNOPQRSTUVWXYZ [\ ] ^ _ ` abcdefghijkl
0123456789 ; <=> ? @ ABCDEFGHIJKLMNOPQRSTUVWXYZ [\ ] ^ _ ` abcdefghijkl
```

测试代码：测试源码 /serial\_test/serial\_test.c

驱动代码：linux-2.6.30/driver/serial/atmel\_serial.c

内核对应选项：

Device Drivers --->

Character devices --->

Serial drivers --->

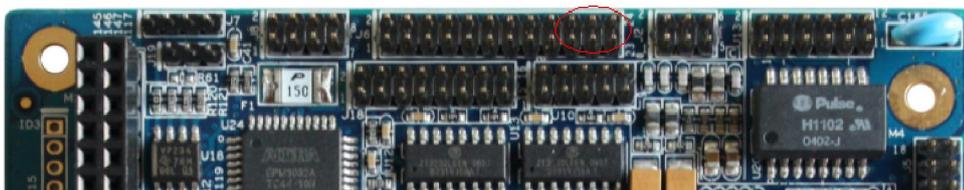
可能遇到问题的解决方法：

- 1 确认设备节点/dev/ ttyS1 (设备号 主-4 次-65)
- 2 确认串口是否接好，并且设置好 (波特率为 115200)
- 3 确认内核选项是否选上

## 1.7 485 测试

RS485的操作与 RS232相似，只是在发送和接收时需要通过操作 IO口来控制 485是发送还是接收。

测试 RS485需要 485设备，没有 485设备的情况下，可以将主板上的 2路 485直连 (用跳线帽将 J6的 19和 23脚短接、20和 24脚短接)，测试 RS485接口。



运行 /usr/test目录下的 rs485\_test程序，参数 0是测试 ttyS2发送 ttyS3接收，参数 1则是 ttyS3发送 ttyS2接收。

```
#!/rs485_test 0
```

测试结果：

在超级终端输入字符串回车后回通过另一路 485打印在控制台上。

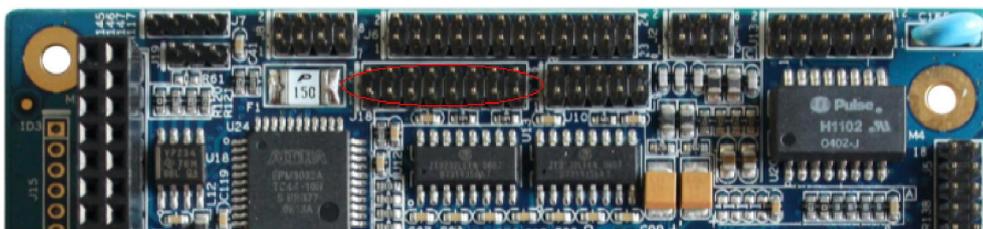
```
/usr/test # ./rs485_test 0
rs485 test ...
have a good time
have a good time
```

测试代码：测试源码 /rs485\_test/ rs485\_test.c

其他同串口测试部分

## 1.8 8x8 键盘测试

主板上配置了 8x8 键盘接口(可复用为 IO 口)J18 具体请参考硬件说明手册，接上键盘。



在/usr/test 目录下运行 keypad\_test 程序

```
# ./keypad_test
```

无错运行后会出现 Watting event ... 然后按下 8x8 键盘进行输入

测试结果：按下键盘后可以看到超级终端上打印出了被按下哪个键

```
/usr/test $ ./keypad_test
Watting event ...

Keyboard key, the code is : 62      Press down.
Watting event ...

Watting event ...

Keyboard key, the code is : 62      Press up.
Watting event ...

Watting event ...

Keyboard key, the code is : 9      Press down.
Watting event ...
```

测试代码：测试源码 /audio\_test/keypad\_test.c

驱动代码：/linux-2.630/drivers/char/at91\_keypad.c

内核对应选项：

Device Drivers --->

Character devices --->

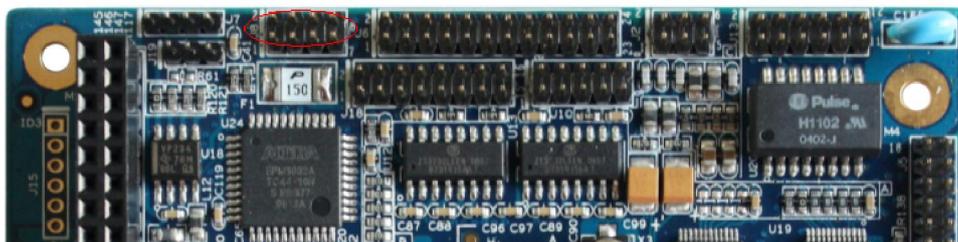
AT91\_KEYPAD

可能遇到问题的解决方法：

- 1 确认设备节点/dev/input/event0 (设备号 主-13 次-64)
- 2 确认键盘是否接好
- 3 确认内核配置是否选上了 Keypad

## 1.9 USB 测试

主板上 有 2 组 USB HOST ，其接口为 J8。



当插入 U 盘后会出现如下一些提示：

```
/usr/test $ usb 1-1: new full speed USB device using at91_ohci and address 4
usb 1-1: configuration #1 chosen from 1 choice
scsi0 : SCSI emulation for USB Mass Storage devices
scsi 0:0:0:0: Direct-Access    Netac    OnlyDisk    PMAP PQ: 0 ANSI: 0 CCS
sd 0:0:0:0: [sda] 3913728 512-byte hardware sectors (2004 MB)
sd 0:0:0:0: [sda] Write Protect is off
sd 0:0:0:0: [sda] Assuming drive cache: write through
sd 0:0:0:0: [sda] 3913728 512-byte hardware sectors (2004 MB)
sd 0:0:0:0: [sda] Write Protect is off
sd 0:0:0:0: [sda] Assuming drive cache: write through
sda: unknown partition table
sd 0:0:0:0: [sda] Attached SCSI removable disk
```

然后用以下命令来查看 U 盘的设备节点

```
# fdisk -l
```

```
/usr/test $
/usr/test $ fdisk -l

Disk /dev/sda: 1999 MB, 1999568384 bytes
32 heads, 63 sectors/track, 1937 cylinders
Units = cylinders of 2016 * 512 = 1032192 bytes

   Device Boot      Start         End      Blocks   Id  System
/dev/sda1          1         1937    1952464+   b   Win95 FAT32
/usr/test $
```

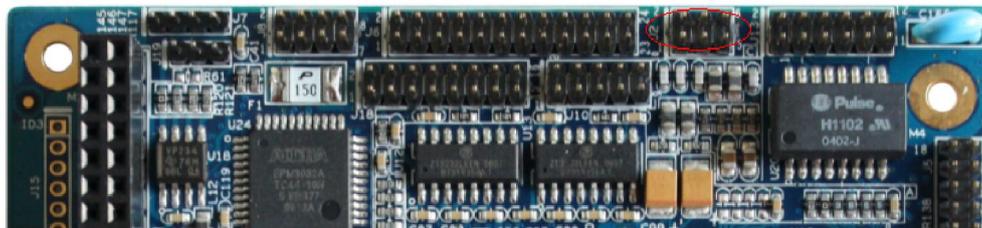
这里看到的是 sda1，所以可以通过以下命令来挂载 U 盘。

目前版本中会自动挂在到/mnt/usb 目录下，如果没有可以通过以下手动方式挂载

```
# mount /dev/sda1 /mnt
```

## 1.10 音频测试

主板上 2 路音频，一路为输出另一路为输入，内部为模拟直通。其在主板上的接口为 J2



运行/usr/test 目录下的 audio\_test，audio\_test 会播放目录下的 test.wav 波形文件

```
# ./audio_test
```

测试结果：正确运行后会听到一段音乐

测试代码：测试源码 /audio\_test/audio\_test.c

驱动代码：/linux-2.6.30/sound/am/at91-ac97.c

内核对应选项：

Device Drivers --->

Sound card support--->

设备节点：

dsp 14 3

mixer 14 0

```
crw-rw---- 1 root root 5, 1 Aug 19 14:05 console
lrwxrwxrwx 1 root root 11 Aug 19 14:05 core ->
crw-rw---- 1 root root 14, 3 Aug 19 14:05 dsp
crw-rw---- 1 root root 29, 0 Aug 19 14:05 fb0
crw-rw---- 1 root root 1, 1 Aug 19 14:05 mem
crw-rw---- 1 root root 14, 0 Aug 19 14:05 mixer
crw-rw---- 1 root root 90, 0 Aug 19 14:05 mtd0
```

可能遇到问题的解决方法：

- 1、确认设备节点是否正确
- 2、确认音量是否调好
- 3、确认内核是否配置支持音频
- 4、确认波形文件是否有问题

## 1.11 触摸屏测试

先对触摸屏进行校准，运行/usr/test/tslib/bin 目录下的校准程序 ts\_calibrate

```
# cd /usr/ tslib/bin
```

```
# ./ ts_calibrate
```

运行上面的校准程序后就可以在 LCD 屏上看到一个十字图标点中它 ,然后又会出现一个同样点中它依次点中 5 个十字图标就完成触摸屏的校准了 ,接下去可以运行测试测序了。

```
# ./ts_test
```

测试结果：看到调试串口输出当前的坐标点，LCD 屏上的十字图标会跟着走。

测试代码：源码 /tslib-1.3.tar.gz

驱动代码：linux-2.6.30/drivers/input/touchscreen/ ads7846.c

内核对应选项：

Device Drivers --->

Input device support --->

Touchscreens --->

ADS7846/TSC2046 and ADS7843 based touchscreens

可能遇到问题的解决方法：

- 1、确定 /dev/input/event1节点是否存在
- 2、确定 /etc/init.d/rcS文件中是否存在以下环境变量：

```
export QTDIR=/
```

```
export QWS_MOUSE_PROTO="TPanel:/dev/input/event1"
```

```
TSLIB_CALIBFILE='/etc/pointercal'  
TSLIB_CONFFILE='/usr/tslib/etc/ts.conf'  
TSLIB_CONSOLEDEVICE='none'  
TSLIB_FBDEVICE='/dev/fb0'  
TSLIB_PLUGINDIR='/usr/tslib/share/ts/plugins'  
TSLIB_TSDEVICE='/dev/input/event1'  
TSLIB_TSEVENTYPE=''  
T_ROOT='/usr/tslib'
```

如果没有则将以上内容添加到 /etc/init.d/rcS 文件中

触摸屏在该系统中对应的事件是 event1 (/dev/input/event1)

- 3、请确认内核配置选项里是否配置了触摸屏

## 1.12 QT 测试

在做 **QT 测试前请进行触摸屏校准**，然后可以运行 /usr/qt/example 目录下的测试测序了。

```
# cd /usr/qt/example
```

运行 hello 窗口程序

```
# ./hello -qws
```

测试结果：可以在 LCD 屏上看到 hello 的窗口

运行 canvas 测试程序

```
#cd canvas
```

```
# ./ canvas -qws
```

测试结果：可以看到带很多图片的一个窗口，图片可以通过触摸屏来拖动。

使用鼠标键盘：

- 1、鼠标可以通过导出以下环境变量来支持：

```
export QWS_MOUSE_PROTO="USB:/dev/input/mousex"
```

x 为设备 ID，具体是多少可以查看 /dev/input 目录

- 2、键盘可以通过导出以下环境变量来支持：

```
export QWS_KEYBOARD="USB:/dev/input/eventx"
```

x 为设备 ID，具体是多少可以查看 /dev/input 目录

- 3、触摸屏可以通过导出以下环境变量来支持：

```
QWS_MOUSE_PROTO="TPanel:/dev/input/event1"
```

## 1.13 SPI 测试

QY-9263S 开发板引出了 SPI1 供用户使用，J10 接口的 1、3、5、7 引脚分别对应 SPI1 的 MISO、MOSI、SPCK、nCS0，另外还有 3 个片选引脚和 J18 的 1、3、5 引脚复用(参考原理图)。

运行/usr/test 目录下的 spi 程序

```
./spi x (x=0, 1, 2, 3)选择 SPI1 的片选 0、1、2 或 3
```

测试结果：

该测试程序是每 1 秒往外发送数据 0x55，所以运行程序后用示波器测量 MOSI 引脚(J10 的 pin1) 能够看到波形，同时 SPCK 引脚(J10 的 pin5) 有时钟波形，相应的片选引脚上有一个低电平信号

测试代码：example/spi\_test/spi.c

驱动代码：/linux-2.6.30/drivers/spi/

内核选项：

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

```
  SPI support --->
```

```
    Atmel SPI Controller
```

```
      User mode SPI device driver support
```

## 1.14 PC104 测试

运行 /usr/test 目录下的 pc104 程序

```
#!/pc104
```

测试结果：每秒从 pc104 的 0 地址读出数据，用示波器测量 B13 引脚有一个低电平信号

```
/usr/test # ./pc104
pcx_open 2
pcx_base2 = c4a00000
pcx_read addr =c4a00000, data = 552..
pcarg.pcx_data=0x552
pcx_read addr =c4a00000, data = ffff..
pcarg.pcx_data=0xffff
pcx_read addr =c4a00000, data = c618..
pcarg.pcx_data=0xc618
```

测试代码：example/pc104/pc104.c

内核选项：

Device Drivers --->

Character devices --->

AT91\_PC104

可能出现的错误：

打开设备文件失败,这时候查看一下文件系统中 dev目录下有没有 at91\_pc104这个设备文件,如果没有,用命令创建：

```
mknod at91_pc104 c 222 2
```

如果创建了还不行,看看内核中有没有添加相应的选项

## 1.15 CAN 测试

QY-9263S 开发板提供了一路 CAN 总线接口,在软件上将 CAN 封装成 socket 接口,具体的使用方法请查看《QY-9263S Linux sockecan 使用手册》

## 附录

### 附A : Release Notes

版本号	时间	修订内容
v2.8	2011-11-20	更新内容组织形式 添加 CAN 总线的测试
v2.6	2010-9-17	添加 spi 测试程序 添加 pc104 测试程序 升级到 linux-2.6.30 u-boot-1.3.4
v2.4	2009-09-21	修正 rs485 测试中引脚连接描述错误 修正 USB 描述不正确
v2.0	2009-08-20	内容组织形式更改 添加测试内容： 8x8 键盘测试 rs485 测试 USB 测试 触摸屏测试 QT 测试
v0.9	2008-05-14	首版

## 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：support@qiyangtech.com

网址：<http://www.qiyangtech.com>

地址：杭州市西湖科技园西园一路 8号 2幢 5层

邮编：310012