



QY-9G45EK Linux 测试手册

版本号 v1.1

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.....	4
1.3 PWM 测试.....	4
1.4 RTC 测试.....	5
1.5 串口测试.....	6
1.6 8x8 键盘测试.....	6
1.7 SPI 测试.....	7
1.8 I2C 测试.....	8
1.9 ebi 测试.....	8
1.10 USB 测试.....	9
1.11 音频测试.....	9
1.12 触摸屏测试.....	10
1.13 QT 测试.....	11

前言

欢迎使用杭州启扬智能科技有限公司产品 QY-9G45EK，本产品 Linux 部分包含 3 份手册：QY-9G45EK Linux 用户手册、QY-9G45EK 硬件说明手册以及 QY-9G45EK Linux 测试手册。硬件相关部分可以参考 QY-9G45EK 硬件说明手册，主板测试可以参考 QY-9G45EK Linux 测试手册。

使用之前请仔细阅读 QY-9G45EK Linux 用户手册以及 QY-9G45EK 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: 0571-87858822

技术支持 E-MAIL: 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 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-9G45EK 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：

```
U-Boot 1.3.4 Yxx_QY (Jul 20 2010 - 18:35:45)

+++++
DRAM: 128 MB
NAND: 128 MiB
In: serial
Out: serial
Err: serial
Net: QY macb0
MII_PHYSID1:0x181
macb0: Starting autonegotiation...
macb0: Autonegotiation timed out (status=0x7849)
macb0: link down (status: 0x7849)
Hit any key to stop autoboot: 0

NAND read: device 0 offset 0x400000, size 0x200000
2097152 bytes read: OK

NAND read: device 0 offset 0x2800000, size 0x53dff0
5496828 bytes read: OK
```

3、进入/usr/test 目录

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

4、进入 test 目录

```
#cd test  
  
#ls
```

在 test 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

```
#!/buzzer
```

测试结果:程序正常运行没有出现错误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: 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

在/usr/test 目录下运行 gpio_test 程序

```
#!/gpio_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, 并持续读 100此输入 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： gpio_test/gpio_test.c

驱动代码： linux-2.6.33/driver/char/at91_gpio.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 gpio0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 gpio0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 PWM 测试

CPU 芯片 AT91SAM9G45EK 内部包括 4 路独立 PWM 输出。在 QY-9G45EK 主板上, 包含了两路用户 PWM, 分别为 PWM1 和 PWM2, 在插件 J22 的 PIN9 和 PIN10。PWM3 被设计为板上 LED(D5)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 PWM3, 可以看到主板上的蓝色 LED会不断的闪烁。

测试代码： 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

1.4 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 072210502010 //设置当前时间为 2010-07-22 10:50:00
```

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

```
/usr/test # ./rtc_test
```

测试结果：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： 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。

1.5 串口测试

在 QY-9G45EK主板上，共有 5路串口，其中 J9被用来做为调试串口，其他四路可以当作 RS232 串口使用，其中 COM1和 COM2与 RS485接口复用。

串口测试程序位于/usr/test 目录下，测试如下：

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

```
# ./serial_test
```

测试结果：正确运行后可以用示波器测量每个串口，有波形出现。

测试代码：serial_test/serial_test.c

驱动代码：linux-2.6.30/driver/serial/atmel_serial.c

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

1.6 8x8 键盘测试

主板上配置了 8x8 键盘接口（J21），可复用为 IO 口，具体请参考硬件说明手册。

8x8 键盘测试程序位于/usr/test 目录下，测试如下：

接上 8x8 键盘

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

```
# ./keypad_test
```

正确运行后会出现 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串口上打印出了被按下哪个键

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

Keyboard key, the code is 62 Press down.
Wattting event ...

Wattting event ...

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

Wattting event ...

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

测试代码： keypad_test/keypad_test .c

驱动代码： /linux-2.6.30/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

1.7 SPI 测试

主板引出了 SPI 接口(J22)

在/usr/test 目录下运行 SPI 的测试程序 spi

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

```
# ./spi
```

测试结果:运行正确后，输出 0x55，用示波器可在 J22 接口测量到波形：

测试代码： spi_test/spi.c

驱动代码： linux.-2.6.30/drivers/spi/...

内核对应选项：

Device Drivers --->

SPI support --->

Atmel SPI Controller

User mode SPI device driver support

1.8 I2C 测试

主板引出了 I2C 接口 (J20)

在/usr/test 目录下运行 I2C 的测试程序 i2c

```
# ./i2c
```

运行没有错误，用示波器测量对应的应的引脚有波形输出

测试代码： i2c_test/i2c.c

驱动代码： linux-2.6.30/drivers/i2c/...

内核对应选项：

Device Drivers --->

I2C support --->

I2C device interface

Autoselect pertinent helper modules

I2C Hardware Bus support --->

GPIO-based bitbanging I2C

1.9 ebi 测试

主板引出了 ebi 接口 (J19)

在/usr/test 目录下运行 ebi 的测试程序 ebi

```
# ./ebi_test
```

测试结果：正确运行后，结果如下：

```
/usr/test # ./ebi_test
AT91 EBI test ...
ebi_open 0
ebi_base0 = c9000000
ebi_open 1
ebi_base1 = da000000
ebi_write addr =c9000000,data = 5a5a
ebi_write addr =c9000001,data = 5a5a
ebi_write addr =c9000003,data = 5a5a
ebi_write addr =c9000006,data = 5a5a
```

测试代码： ebi_test/ebi_test.c

驱动代码： linux.-2.6.30/drivers/char/at91_ebi.c

内核对应选项：

Device Drivers --->

Character devices --->

AT91_EBI

1.10 USB 测试

当插入 U 盘后会出现如下一些提示：

```
/ # usb 1-2: new high speed USB device using atmel-ehci and address 2
usb 1-2: configuration #1 chosen from 1 choice
scsi0 : SCSI emulation for USB Mass Storage devices
scsi 0:0:0:0: Direct-Access   General   USB Flash Disk   1100 PQ: 0 ANSI: 0 CCS
sd 0:0:0:0: [sda] 15794176 512-byte hardware sectors: (8.08 GB/7.53 GiB)
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] Assuming drive cache: write through
   sda: sda1
sd 0:0:0:0: [sda] Attached SCSI removable disk
/ #
```

然后可以用以下命令来查看 U 盘的设备节点

```
# fdisk -l
```

```
/ # fdisk -l
Disk /dev/sda: 8086 MB, 8086618112 bytes
26 heads, 16 sectors/track, 37966 cylinders
Units = cylinders of 416 * 512 = 212992 bytes

   Device Boot      Start         End      Blocks   Id  System
/dev/sda1 *           1         37967     7897048    b   Win95 FAT32
/ #
```

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/flash
```

1.11 音频测试

开发板提供了两路音频输入 line-in(J4)和 mic(J5)，还提供了一路音频输出 headphone-out (J3)，这里只测试音频输出，将耳机接到 headphone-out 输出端(J3)，运行/usr/test 目录下的 audio_test，audio_test 会播放通目录下的 test.wav 波形文件

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

```
# ./audio_test
```

测试结果：正确运行的话会听到一段音乐

测试代码：audio_test/audio_test .c

驱动代码：/linux-2.6.30/sound/..

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

1.12 触摸屏测试

接好 LCD 以及触摸屏运行/usr/test/tslib/bin 目录下的校准程序 ts_calibrate 后就可以进行测试了，具体操作如下：

```
# cd /usr/test/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：运行无错的话就会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：见 tslib-1.3源码目录下的 tests文件夹。

驱动代码：linux-2.6.30/drivers/input/touchscreen/atmel_tsadcc.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/ts0 节点是否存在
- 2、确定以下环境变量是否存在：

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'

如果没有则可用 `export XXX=xxxxx` 来进行导出。

- 3、请确认内核配置选项里是否配置了触摸屏

1.13 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
```

- 1、运行 hello 窗口程序

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

测试结果：可以看到 hello 的窗口

- 2、运行 canvas 测序

```
# cd canvas
```

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

```
# ./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：

鼠标可以通过导出以下环境变量来支持：

```
export QWS_MOUSE_PROTO="USB:/dev/input/mouse1"
```

x 为设备 ID，具体是多少可以查看/dev/input 目录

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

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.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