



QY-9263K Linux 用户手册

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 Reserv

目 录

前言.....	1
一、准备工作.....	2
二、烧写 Linux 内核和文件系统.....	2
三、搭建嵌入式 Linux 开发环境.....	2
3.1 安装交叉编译器.....	2
3.2 NFS 网络文件系统.....	3
四、内核的编译.....	4
五、应用程序的开发.....	5
5.1 Hello World.....	5
5.2 交叉编译.....	6
5.3 运行应用程序.....	6
5.4 添加应用程序到文件系统.....	7
附录.....	8

前言

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

使用之前请仔细阅读 QY-9263K Linux 用户手册以及 QY-9263K 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硬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：杭州市西湖区西湖科技园西园一路 8 号 2 幢 5 层

邮编：310012

一、准备工作

装有 Linux 系统 (Ubuntu 或其它 Linux 发行版),本手册以 Ubuntu 11.04 操作为例

串口连接:通过提供的串口线将开发板的调试串口(J7)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(J16)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(J22)与 PC 机的 USB 连接

串口设置:打开终端通讯软件(迷你终端或 Windows 下的超级终端),选择所用到的串口并设置如下参数:波特率(115200)、数据位(8 位)、停止位(1 位)、校验位(无)、数据流控制(无)

找到核心板上的 J0 跳线,在下面的操作步骤中要用到它

二、烧写 Linux 内核和文件系统

如果开发板出厂默认的程序不是 Linux 系统,那么我们要先烧写 Linux 程序,具体烧写方法请参照:

《QY-9263K SAM-BA 烧写从 dataflash 启动的镜像 .pdf》

《QY-9263K SAM-BA 烧写从 nandflash 启动的镜像 .pdf》

三、搭建嵌入式 Linux 开发环境

3.1 安装交叉编译器

在 PC 机上编译目录板所用的 Linux 内核和其他应用程序时要用到交叉编译器,我们的主板光盘中做好了交叉编译工具链。

1、解压交叉编译工具链

```
#cd /opt
```

```
#tar -xvf arm-2007q1-10-arm-none-linux-gnueabi-i686-pc-linux-gnu.tar.bz2
```

这样在/opt 目录下就会有 arm-2007q1 文件夹,我们的工具链就有了

2、把交叉编译器的路径添加到环境变量里

```
#vi /etc/bash.bashrc
```

在文件后面添加以下路径:

```
PATH=/opt/arm-2007q1/bin:$PATH
```

3、使新的环境变量生效

```
#source /etc/profile
```

4、查看

在终端输入 `arm-none-linux-gnueabi-gcc -v` 命令来查看交叉编译是否安装正确,如图:

```
root@ubuntu:/# arm-none-linux-gnueabi-gcc -v
Using built-in specs.
Target: arm-none-linux-gnueabi
Configured with: /scratch/paul/arm/src/gcc-4.2/configure --build=i686-pc-linux-gnu --host=i686-pc-linux-gnu --target=arm-none-linux-gnueabi --enable-shared --enable-threads --disable-libmudflap --disable-libssp --disable-libgomp --disable-libstdcxx-pch --with-gnu-as --with-gnu-ld --prefix=/opt/codesourcery --enable-languages=c,c++ --enable-symvers=gnu --enable-__cxa_atexit --with-versuffix=CodeSourcery Sourcecory G++ Lite 2007q1-10 --with-pkgversion=CodeSourcery Sourcecory G++ Lite 2007q1-10 --with-bugurl=https://support.codesourcery.com/GNUToolchain/ --disable-nls --with-sysroot=/opt/codesourcery/arm-none-linux-gnueabi/libc --with-build-sysroot=/scratch/paul/arm/install/arm-none-linux-gnueabi/libc --enable-poison-system-directories --with-build-time-tools=/scratch/paul/arm/install/arm-none-linux-gnueabi/bin --with-build-time-tools=/scratch/paul/arm/install/arm-none-linux-gnueabi/bin
Thread model: posix
gcc version 4.2.0 20070413 (prerelease) (CodeSourcery Sourcecory G++ Lite 2007q1-10)
```

若出现上图中所显示的交叉编译器的信息,则说明交叉编译器安装正确。

3.2 NFS 网络文件系统

NFS 是 Network FileSystem 的简称,可以让不同的主机通过网络访问远端的 NFS 服务器共享出来的文件。这样,将主机当作 NFS 服务器,我们就可以在开发板上通过网络访问主机的文件。在嵌入式中使用 NFS 使得应用程序的开发十分方便,具体的使用方法如下:

1、安装

```
$ sudo apt-get install nfs-kernel-server
```

2、创建共享目录,修改 /etc/exports 文件末尾

```
$ sudo vi /etc/exports
```

在文件末尾加入下面一行:

```
/ *(rw, sync, no_subtree_check, no_root_squash)
```

如图:

```
1 # /etc/exports: the access control list for filesystems which may be exported
  #
2 #       to NFS clients.  See exports(5).
3 #
4 # Example for NFSv2 and NFSv3:
5 # /srv/homes      hostname1(rw,sync,no_subtree_check) hostname2(ro,sync,no_
  subtree_check)
6 #
7 # Example for NFSv4:
8 # /srv/nfs4      gss/krb5i(rw,sync,fsid=0,crossmnt,no_subtree_check)
9 # /srv/nfs4/homes gss/krb5i(rw,sync,no_subtree_check)
10 #
11 / *(rw,sync,no_subtree_check,no_root_squash)
```

这样，就将主机的根目录当作了共享目录了

- 4、启动 NFS 服务器，启动 portmap 和 nfs-kernel-server 服务：

```
$ sudo service portmap start
```

```
$ sudo service nfs-kernel-server start
```

注：如果开发板不能挂载到宿主机可以试下以下方式重新启动 NFS 服务器

```
sudo exportfs -a
```

```
sudo /etc/init.d/portmap restart
```

```
sudo /etc/init.d/nfs-kernel-server restart
```

- 5、本地验证 NFS

```
$ sudo mount local:/ /mnt
```

```
$ ls /mnt
```

这时可以在/mnt 目录下查看到根目录下的内容，则说明 NFS 配置正确

四、内核的编译

光盘中有配置好的内核源码文件 linux-2.6.30-9263k-qiyang-v2.8.tar.gz 压缩包，将它拷到一个目录下，用户可以自行配置内核

- 1、解压内核源码压缩包

```
$ tar -xzf linux-2.6.30-9263k-qiyang-v2.8.tar.gz
```

解压后生成一个新目录 linux-2.6.30

- 2、配置内核

```
$ cd linux-2.6.30
```

```
$ make menuconfig
```

运行后会弹出一个内核配置菜单，用户可以对内核功能选项做必要的调整，配置好之后保存退出

3、编译和生成内核映像

编译，执行以下命令

```
$ make
```

生成内核映像

编译后，需要用到 mkimage 工具来生成内核映像，先查看/usr/bin 目录下是否有 mkimage 工具的文件

```
root@ubuntu:/# ls -l /usr/bin/mkimage
-rwxr-xr-x 1 root root 52240 2011-01-21 08:56 /usr/bin/mkimage
```

如果/usr/bin 目录下没有这个文件，您需要将光盘中的工具软件文件夹里的 mkimage 文件拷到/usr/bin 目录下，并执行 chmod 755 mkimage 修改权限，然后执行以下命令生成我们所需要的内核映像

```
$ make uImage
```

编译完了之后我们就可以在 arch/arm/boot/目录下找到重新编译好的内核映像 uImage。

五、应用程序的开发

5.1 Hello World

您可以在开发板上自由地开发应用程序，这里以 Hello World 为例。首先编写一个 Hello World 程序代码如下：

```
#include <stdio.h>

int main(void)
{
    printf("Hello World !\n");
    return 0;
}
```

保存在 hello.c 文件，我们用如下命令对它进行交叉编译

5.2 交叉编译

要使得程序能够运行在开发板上，需要使用之前安装好的交叉编译器对应用程序进行编译。我们可以使用如下命令进行编译：

```
$ arm-none-linux-gnueabi-gcc -o hello hello.c
```

```
$ file hello
```

```
hello: ELF 32-bit LSB executable, ARM, version 1 (SYSV), dynamically linked (uses shared libs), for GNU/Linux 2.6.14, not stripped
```

编译后可以看到在当前目录下生成了用于 ARM 平台的可执行的二进制文件 hello，

5.3 运行应用程序

这样我们就得到了可以在主板上执行的程序了，下面介绍如何在主板运行应用程序，这里介绍了两种方法：挂载 NFS 服务器和添加到文件系统

我们可以使用之前配置好的 NFS 服务器，在开发板上挂载主机的 NFS 服务器后，就可以在开发板上操作主机上的文件，如复制文件、运行程序等。这种方式十分方便于调试，具体实现方法如下：

- 1、确保通过网线将开发板与 PC 机连接好，并在 PC 机上开启了 NFS 服务
- 2、设置开发板的 IP 与 PC 机的 IP 在同一网段，如：

```
PC 机 IP : 192.168.1.241
```

```
目标板 IP : 192.168.1.110
```

```
Network Mask: 255.255.255.0
```

```
Broadcast IP: 192.168.1.255
```

- 3、测试网络连接

在开发板上 ping 主机，在超级终端运行如下命令：

```
# ping 192.168.1.241
```

```
/ # ping 192.168.1.241
PING 192.168.1.241 (192.168.1.241): 56 data bytes
84 bytes from 192.168.1.241: icmp_seq=0 ttl=64 time=0.6 ms
84 bytes from 192.168.1.241: icmp_seq=1 ttl=64 time=0.5 ms
84 bytes from 192.168.1.241: icmp_seq=2 ttl=64 time=0.4 ms
84 bytes from 192.168.1.241: icmp_seq=3 ttl=64 time=0.4 ms
84 bytes from 192.168.1.241: icmp_seq=4 ttl=64 time=0.4 ms
84 bytes from 192.168.1.241: icmp_seq=5 ttl=64 time=0.4 ms
84 bytes from 192.168.1.241: icmp_seq=6 ttl=64 time=0.4 ms
84 bytes from 192.168.1.241: icmp_seq=7 ttl=64 time=0.4 ms
```


在主机上以同样的方式 ping 开发板，如果主机和开发板可以互相 ping 通，则说明网络连接正常。

4、挂载主机 NFS 服务器

在超级终端输入以下命令：

```
#mount -o nolock 192.168.1.241:/ /mnt  
  
#cd /mnt  
  
#ls
```

挂载正确后，可以在开发板的/mnt 目录下看到主机的根目录，然后可以进入到程序所在的目录运行应用程序。通过这种方式，可以方便调试程序，直到程序调试正确后，再将应用程序添加到文件系统中，烧写到 Flash。这样，避免了不断的烧写程序到 Flash。

5.4 添加应用程序到文件系统

在将应用程序调试好后，可以将它直接添加到文件系统，这样您就可以把它和文件系统一起烧到 flash 上了。下面介绍如何将应用程序添加到 jffs2 文件系统。

解开压缩的 jffs2 文件系统源码 jffs2.tar.gz

```
$ tar -xzvf jffs2.tar.gz
```

对解压出来的 jffs2 目录进行操作

```
$ cd jffs2
```

// 可以加入您的应用程序到这个映像文件目录下

生成 jffs2 文件系统映像

生成 jffs2 文件系统映像需要 mkfs.jffs2 工具，先将光盘中工具软件里的 mkfs.jffs2 文件拷到/usr/bin 目录下。

```
$ mkfs.jffs2 -e 0x20000 -l -s 0x800 -p -n -d jffs2 / -o jffs2.jffs2
```

这样，就将应用程序添加到 jffs2 文件系统中，并在当前目录下生成了新的 jffs2 文件系统映像 jffs2.jffs2。

附录

Release Notes

版本号	时间	修订内容
v2.8	2011-11-20	升级 ramdisk 文件系统 升级 jffs2 文件
v2.6	2010-9-20	升级到 linux-2.6.30 u-boot-1.3.4
v2.5	2010-04-15	同步 uboot 配置参数
v2.4	2009-09-21	同步
v2.0	2009-08-05	增加 jffs2 文件系统
v1.2	2008-12-10	烧写内核部分修改
v1.0	2008-06-28	首次发布
v0.9	2008-05-14	初稿

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