



## IAC-IMX6-Kit Linux用户手册

版本号: V 1.0  
2014年03月

杭州启扬智能有限公司版权所有  
QIYANG TECHNOLOGY Co., Ltd  
Copyright Reserv

## 目 录

版本说明.....	错误！未定义书签。
前言.....	4
一、准备工作.....	5
二、烧写 Linux 系统镜像.....	5
三、功能说明与测试.....	5
四、搭建嵌入式 Linux 开发环境.....	5
4.1 安装交叉编译器.....	5
4.2 NFS 网络文件系统.....	7
五、测试源码编译.....	8
六、U-Boot 编译.....	9
七、内核的编译.....	10
八、应用程序的开发.....	11
1 Hello World.....	11
2 交叉编译.....	11
3 运行应用程序.....	12
4 添加应用程序到文件系统.....	13
九、结束语.....	14

## 版本更新记录

版本	硬件平台	描述	日期	修订人
1.0	IAC-IMX6-Kit	初始版本, 首次发布	2014-03-16	

## 前言

欢迎使用杭州启扬智能科技有限公司产品 IAC-IMX6-Kit, 本产品 Linux 部分包含 4 份手册:《IAC-IMX6-Kit Linux 用户手册.pdf》、《IAC-IMX6-Kit 硬件说明手册.pdf》以及《IAC-IMX6-Kit Linux 模块说明与测试手册.pdf》、《IAC-IMX6-Kit 系统镜像烧写手册.pdf》。硬件相关部分可以参考《IAC-IMX6-Kit 硬件说明手册.pdf》, 主板测试可以参考《IAC-IMX6-Kit Linux 模块说明与测试手册.pdf》, 镜像烧写参考《IAC-IMX6-Kit 系统镜像烧写手册.pdf》。

使用之前请仔细阅读《IAC-IMX6-Kit 硬件说明手册.pdf》!

## 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.com](mailto:support@qiyangtech.com)

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 3A 幢 5 层

邮编: 310012

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

第 4 页 共 15 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.qiyangtech.com>

©2012 Qiyangtech 版权所有

## 一、准备工作

- ◆ 一台安装 Windows XP 或 Windows 7 的 PC 机。
- ◆ PC 机安装了虚拟机，虚拟机中安装有 Linux 系统(Ubuntu 或其它 Linux 发行版)，若未安装，可参考提供的《虚拟机安装 ubuntu 指导手册》进行安装，本手册以 Windows XP 下虚拟机安装的 Ubuntu 12.04 操作为例。
- ◆ 串口连接：通过提供的串口线与 PC 机的串口连接，将串口线接开发板的 J7。
- ◆ 网络连接：通过网线将开发板的以太网接口(J8)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(J13)与 PC 机的 USB 连接
- ◆ 串口设置：打开 Windows 下终端通讯软件或光盘工具目录下提供的 putty.exe 软件，选择连接开发板所用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）

## 二、烧写 Linux 系统镜像

IAC-IMX6-Kit 有专门的烧写工具 Mfgtools，请根据需要进行烧写，具体烧写方法请参照：

《IAC-IMX6-Kit 系统镜像烧写手册.pdf》

## 三、功能说明与测试

板子文件系统中已经集成测试程序，启动板子后可在/usr/test 目录下找到相应测试程序，具体如何进行测试，请参照手册：

《IAC-IMX6-Kit Linux 模块说明与测试手册.pdf》。

## 四、搭建嵌入式 Linux 开发环境

### 4.1 安装交叉编译器

在 PC 机上 Ubuntu 12.04 环境下安装用于编译开发板 Linux 内核和其他应用程序所需的交叉编译器，我们的主板光盘中有做好的交叉编译工具链。

以下命令均非 `root` 用户下执行，若用户登录的是 `root` 用户，则在以下命令中去掉 `sudo`。将光盘编译器目录下的压缩包拷贝到 Ubuntu 12.04 普通目录下，再通过命令拷贝到 `/opt` 目录下，若不通过命令而直接拖拽拷贝，则可能因当前登录的用户没有写 `/opt` 目录的权限而导致失败。

将文件通过 `cp` 命令拷贝到 `/opt`，并解压，`/opt` 目录下就有了 `fsl-linaro-toolchain` 文件夹，这样我们就得到了交叉编译器。

```
test@ubt:~ $ ls
```

```
fsl-linaro-toolchain.tar.gz
```

```
test@ubt:~ $ sudo cp fsl-linaro-toolchain.tar.gz /opt
```

```
test@ubt:~ $ cd /opt
```

```
test@ubt:~/opt$ sudo tar xzvf fsl-linaro-toolchain.tar.gz
```

- 把交叉编译器的路径添加到环境变量中。

```
test@ubt:~/opt $ sudo vi /etc/bash.bashrc
```

在打开的文件最后一行添加以下命令：

```
PATH=/opt/ fsl-linaro-toolchain/bin:$PATH
```

- 使新的环境变量生效。

```
test@ubt:~/opt $ source /etc/ bash.bashrc
```

- 查看。

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

```

Using built-in specs.
COLLECT_GCC=arm-none-linux-gnueabi-g++
COLLECT_LTO_WRAPPER=/home/wujj/bin/fsl-linaro-toolchain/bin/./libexec/gcc/arm-fsl-linux-gnueabi/4.6.2/lto-wrapper
Target: arm-fsl-linux-gnueabi
Configured with: /work/build/.build/src/gcc-linaro-4.6-2011.06-0/configure --build=i686-build_pc-linux-gnu --host=i686-build_pc-linux-gnu --target=arm-fsl-linux-gnueabi --prefix=/work/fsl-linaro-toolchain-2.13 --with-sysroot=/work/fsl-linaro-toolchain-2.13/arm-fsl-linux-gnueabi/multi-libs --enable-languages=c,c++ --with-pkgversion='Freescale MAD -- Linaro 2011.07 -- Built at 2011/08/10 09:20' --enable__cxa_atexit --disable-libmudflap --disable-libgomp --disable-libssp --with-gmp=/work/build/.build/arm-fsl-linux-gnueabi/build/static --with-mpfr=/work/build/.build/arm-fsl-linux-gnueabi/build/static --with-mpc=/work/build/.build/arm-fsl-linux-gnueabi/build/static --with-ppc=/work/build/.build/arm-fsl-linux-gnueabi/build/static --with-cloog=/work/build/.build/arm-fsl-linux-gnueabi/build/static --with-libelf=/work/build/.build/arm-fsl-linux-gnueabi/build/static --with-host-libstdcxx='-static-libgcc -wl,-Bstatic,-lstdc++,-Bdynamic -lm -L/work/build/.build/arm-fsl-linux-gnueabi/build/static/lib -lpwl' --enable-threads=posix --enable-target-optspace --enable-plugin --enable-multilib --with-local-prefix=/work/fsl-linaro-toolchain-2.13/arm-fsl-linux-gnueabi/multi-libs --disable-nls --enable-c99 --enable-long-long --with-system-zlib
Thread model: posix
gcc version 4.6.2 20110630 (prerelease) (Freescale MAD -- Linaro 2011.07 -- Built at 2011/08/10 09:20)
    
```

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。

## 4.2 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_
6 #   subtree_check)
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

## 五、测试源码编译

在光盘/源码/测试源码目录下，提供了各测试部分源码，可根据需要进行修改编译。所有以下操作均需在系统已安装交叉编译器并加入到环境变量中前提下执行，若未安装交叉编译器，则需参照 4.1 安装交叉编译器。



选择要进行编译的模块源码，此处以 `buzzer` 为例，将目录：光盘/源码/测试源码/`buzzer_test` 及共用头文件目录：光盘/源码/测试源码/`include`，拷贝到 `ubuntu 12.04` 目录下，此处以 `test` 用户主目录为例，打开 `linux` 终端，进入到 `buzzer_test` 目录，执行 `make` 命令，在该目录下将得到编译后的可执行文件 `buzzer_test`：

```
test@ubt:~ $ ls
buzzer_test  include
test@ubt:~ $ cd buzzer_test
test@ubt:~/buzzer_test $ make
```

## 六、U-Boot 编译

光盘提供了已经编译好的引导镜像及源码，请尽量使用光盘自带镜像作为引导程序，若需对该引导程序进行重新编译，请参照如下步骤执行，执行如下命令前，请务必已经正确安装交叉编译器，并添加到环境变量中，若未安装，请参考本手册 3.1 小结进行安装：

源码目录：

光盘/源码/u-boot/ `u-boot-2009.08.tar.gz`

命令：

拷贝源码到 `linux` 环境下，打开命令终端，将路径切换到源码所在位置。此处以 `ubuntu 12.04` 环境为例，若使用的是普通账户，则执行如下命令时需添加 `sudo`，若是 `root` 账户，则不需添加。

- 解压源码，得到目录 `u-boot-2009.08`，切换到这个目录下。

```
test@ubt:~ $ ls
u-boot-2009.08.tar.gz
test@ubt:~ $ sudo tar xzvf u-boot-2009.08.tar.gz
test@ubt:~ $ cd u-boot-2009.08
test@ubt:~/u-boot-2009.08 $ make distclean
test@ubt:~/u-boot-2009.08 $ make mx6q_sabresd_config
test@ubt:~/u-boot-2009.08 $ make
```

执行最后的 make 命令后，终端会打印出一大堆信息，最后的信息如图所示：

```
arm-none-linux-gnueabi-objcopy -O srec u-boot u-boot.srec  
arm-none-linux-gnueabi-objcopy --gap-fill=0xff -O binary u-boot  
u-boot.bin
```

说明 u-boot 编译成功，在源码当前目录下就会生成 u-boot.bin 文件。

## 七、内核的编译

光盘中有配置好的内核源码文件 qy-imx6-linux-3.0.35.tar.gz

压缩包，将它拷到一个 Ubuntu12.04 目录下，用户可以自行配置内核。编译内核之前，请确保当前的交叉编译器已正确安装到系统中。

1. 解压源码，得到源码目录。

```
test@ubt:~ $ ls
```

```
qy-imx6-linux-3.0.35.tar.gz
```

```
test@ubt:~ $ sudo tar -xzvf qy-imx6-linux-3.0.35.tar.gz
```

解压后生成一个新目录 linux-3.0.35

2. 配置内核。

```
test@ubt:~ $ cd linux-3.0.35
```

```
test@ubt:~/linux-3.0.35 $ 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. 编译内核。

```
test@ubt:~/linux-3.0.35 $ 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。

## 八、应用程序的开发

### 1 Hello World

您可以在 PC 机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。

### 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。将以上文件放到 Ubuntu12.04 目录中，我们可以使用如下命令进行编译：

```
test@ubt:~ /$ ls
hello.c
```

```
test@ubt:~ /$ arm-none-linux-gnueabi-gcc -o hello hello.c
```

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

### 3 运行应用程序

这样我们就得到了可以在开发板上执行的程序了，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 : 192.168.1.104

目标板 IP: 192.168.1.103

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 192.168.1.98
```

```
# ping 192.168.1.104
PING 192.168.1.104 (192.168.1.104): 56 data bytes
64 bytes from 192.168.1.104: seq=0 ttl=64 time=7.023 ms
64 bytes from 192.168.1.104: seq=1 ttl=64 time=0.223 ms
64 bytes from 192.168.1.104: seq=2 ttl=64 time=0.298 ms
64 bytes from 192.168.1.104: seq=3 ttl=64 time=0.261 ms
64 bytes from 192.168.1.104: seq=4 ttl=64 time=0.286 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 192.168.1.104:/ /mnt
```

```
# cd /mnt
```

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。

## 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根文件系统。

- ◆ 从光盘中拷贝根文件系统源码到 Ubuntu12.04，并解压。

先新建一个文件夹 rootfs，解压出来的文件就放在这个文件夹里。

```
test@ubt:~ /$ mkdir rootfs
```

把光盘中的文件系统压缩文件 rootfs-mx6q-qy.tar.bz2 拷贝到该文件目录下。

```
test@ubt:~ /$ cd rootfs
```

```
test@ubt:~ / rootfs $ ls
```

```
rootfs-bicycle.tar.bz2
```

```
test@ubt:~ / rootfs $ sudo tar -jxvf rootfs-mx6q-qy.tar.bz2
```

- ◆ 对解压出来的 rootfs 目录进行操作

```
test@ubt:~/rootfs $ ls
```

```
bin      etc      lib      media    opt      root     sbin     tmp      var  
dev      home    linuxrc  mnt     proc     run      sys      usr
```

可以在所列的目录中加入您的应用程序。

- ◆ 压缩该目录下的文件，压缩之前先删除原先的 rootfs-mx6q-qy.tar.bz2

```
test@ubt:~/rootfs $ rm rootfs-mx6q-qy.tar.bz2
```

```
test@ubt:~/rootfs $ tar -jcvf rootfs-mx6q-qy.tar.bz2 -R *
```

压缩完后，在当前目录下就有一个新生成的 tar -jcvf rootfs-mx6q-qy.tar.bz2 文件。

## 九、结束语

以上内容可能不够翔实，有任何技术问题或建议，欢迎联系我们：[supports@qiyangtech.com](mailto:supports@qiyangtech.com)，也可登录我司论坛进行交流：<http://www.qiytech.com/bbs/>，关于更多产品的信息，欢迎联系销售 [sales@qiyangtech.com](mailto:sales@qiyangtech.com)，或登录 <http://www.qiytech.com/index.html>。

## 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](mailto: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 3A 幢 5 层

邮编：310012