

0.前导工作

创建工具目录：

在上位机根目录下建立工作目录/opt

执行命令:\$sudo mkdir /opt/linux

拷贝光盘文件到用户家目录下，假设在用户家目录下的 myc-dir 文件夹

1.交叉编译工具链

(1) 将 myc-dir/03-Tools 目录下的 gcc-linaro-arm-linux-gnueabi-4.7-2013.04-20130415_linux.tar.bz2文件解压到/opt/linux 目录下

执行命令：\$cd myc-dir

\$sudo tar -xvzf 03-Tools/Cross_compiler/gcc-linaro-arm-linux-gnueabi-4.7-2013.04-20130415_linux.tar.bz2 /opt/linux

(2) 将交叉编译工具链配置到系统中

执行命令：\$sudo nano /etc/bash.bashrc

在打开的文件末尾添加以下内容：

#gcc 交叉编译器

export PATH=\$PATH:/opt/linux/gcc-linaro-arm-linux-gnueabi-4.7-2013.04-20130415_linux/bin/

(3) 测试交叉编译工具链是否添加成功

在终端输入：\$arm-linux-后按 TAB 键盘，能自动补全为 arm-linux-gnueabi-说明配置成功

2.Bootloader

(1) 说明：Bootloader 包含 Bootstrap（一级）和 Uboot（二级），Bootstrap 在上电后对底板进行简单的硬件电路的初始化并引导 Uboot，无需关心

Uboot 进一步初始化硬件电路，并引导 Linux kernel，开机画面需要在 Uboot 源码部分修改。

(2) 交叉编译 Uboot

执行命令：\$cd myc-dir/04-Linux_Source/U-boot

\$tar -xvzf u-boot-at91.tar.bz2

\$make clean

\$make ARCH=arm

(3) 修改 Uboot

修改 Uboot，交叉编译后，将生成的 U-boot.bin 文件拷贝到自动化烧录的文件夹里面替换原有的 U-boot.bin 文件即可

3.Linux Kernel

按需修改内核文件，交叉编译后替换自动化烧录文件夹里对应的 uImage 文件

4.File System

无需修改

5.qt4 (1) 上位机配置 qt 交叉编译工具

执行命令：\$cd myc-dir/04-Linux_Source/Qt_Arm/

\$sudo tar -xvzf qt-embedded-linux-opensource-src-4.5.3.tar.gz
/opt/linux

(2)将 qt 相关环境变量配置到系统中

执行命令：\$sudo nano /etc/bash.bashrc

在打开的文件末尾添加以下内容：

#qt 交叉编译工具链环境变量

```
export PATH=/opt/linux/qt4.5.3-arm/bin:$PATH
export
```

```
LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/opt/linux/qt4.5.3-arm/lib
```

6.挂载 NFS

(1) 安装 nfs 服务

执行命令：\$sudo apt-get install nfs-kernel-server nfs-common

(2)配置权限

执行命令：\$sudo nano /etc/exports

在文件末尾追加以下内容：

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

(3) 重启 nfs 服务

执行命令：\$sudo service nfs-kernel-server restart

(4) 测试 nfs

假设上位机 IP 为192.168.1.100

执行命令：\$sudo mount -o nolock 192.168.1.100:/home /mnt

在上位机/mnt 目录下显示内容与/home 目录内容相同，则 nfs 服务配置

正确

7.调试工具 GDB+GDBSERVER

(1) 下载 gdb 源代码，解压后进入源码根目录

(2) 配置编译选项

执行命令：\$sudo ./configure --target=arm-linux

```
$sudo make
```

```
$sudo make install
```

(3)编译通过后，进入 gdbserver 目录

执行命令：\$./configure --target=arm-linux --host=arm-linux

```
$make CC=/opt/linux/gcc-linaro-arm-linux-gnueabi-hf-4.7-
```

```
2013.04-20130415_linux/bin/arm-linux-gnueabi-hf-gcc
```

(4) 目录下生成的 gdbserver 文件即为目标文件，gdb 远程调试的客户端，将其放在 /home 目录下

执行命令：\$cp gdbserver ~/

8.配置下位机

(1) 测试网络

执行命令：#ping baidu.com

命令能正常运行则忽略 DNS 配置，否则，配置 DNS

(2) 配置 DNS

执行命令：#vi /etc/resolv.conf

添加以下内容：

```
nameserver 192.168.1.1
```

其中，192.168.1.1为 DNS 地址

(3) 挂载上位机 NFS 到开发板，假设上位机 IP 为192.168.1.100

执行命令：#mount -o nolock 192.168.1.100:/home /mnt

命令无异常输出即挂载成功

执行命令：#ls /mnt

能看到上位机/home 目录下所有内容，并可以读写，其中，中文字符乱

码显示

(4) 使用 gdbserver 调试下位机程序

配置调试工具

执行命令：`#cp /mnt/gdbserver /usr/bin`

(5) 测试调试工具

将上位机交叉编译后得到的目标文件拷贝到下位机任意目录，并进入该目录,假设待调试程序为 test

下位机操作如下：

执行命令：`#gdbserver 192.168.1.100:2345 test`

其中，192.168.1.100为上位机 IP，2345为端口

上位机操作如下：

执行命令：`#arm-linux-gdb test`

进入 gdb 调试界面后，

执行命令：`(gdb)target remote 192.168.1.55:2345`

其中，192.168.1.55为下位机 IP，2345为与上位机保持一致的端口