Lablin 开发者手册

第 1 卷

MaxWit 开放实验室

http://maxwit.googlecode.com

2009 年 7 月 1 日

目　录

1 Host端发行版的选择及软件安装

1.1 关于Linux发行版

目前已测试通过的发行版有（包括 64 位版）：Debian5.0、Ubuntu 9.04、Ubuntu 8.10、Fedora Core 10，推荐使用 Debian5.0。若有人有兴趣测试并支持其他 Linux 发行版，欢迎把 patch 发给 MaxWit 项目维护者：

Conke Hu	conke@maxwit.com
Tiger Yu	tiger@maxwit.com
Fleya Hou	fleya@maxwit.com

1.2 安装软件包

必须安装的软件包：
gcc, g++, make, subversion, git-core, tftpd-hpa, tftp-hpa, nfs-kernel-server, libsdl 及其 dev 包, zlib 及其 dev 包, autoconf, automake, m4, libtool,
64 位系统上需要额外安装的软件包：
libc6-dev-i386
Debian 或 Ubuntu 系统上可通过如下命令安装有软件包：

```
# apt-get install gcc g++ make subversion git-core libncurses5-dev zlib1g-dev libsdl1.2-dev tftpd-hpa nfs-kernel-server autoconf automake m4 libtool
```

注：Ubuntu 用户还需执行以下操作

```
# dpkg-reconfigure dash
```

然后选择”no”，执行ls –l /bin/sh确认已指向/bin/bash

2 Host端设置

2.1 安装NFS Server

第一步，安装 NFS server 软件包

```
# apt-get install nfs-kernel-server
```

第二步，编辑/etc/exports 文件，添加下面两行：

```
/root/maxwit/rootfs         *(rw,sync,no_root_squash,no_subtree_check)
```

第三步，重启 NFS Server：

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

第四步，测试 NFS Server：

```
# mount -t nfs 192.168.0.111: /root/maxwit/rootfs    /mnt/
```

（假定本机 IP 为 192.168.0.111）

2.2 安装kermit

第一步，从源码安装 kermit(若前面已安装 kermit，则略过这一步)

```
make linux && make install
wget http://maxwit.googlecode.com/files/kermrc
cp –v kermrc ~/.kermrc
```

第二步，打开~/.kermrc，修改“set line”一行，确认你所用的串口设备，若用的是 USB-to-Serial 转接器，可以改成：“set line /dev/ttyUSB0”

2.3 安装TFTP Server

第一步，编译 tftp 软件（如果前面已经通过 apt 方式安装了 tftp，则跳过这一步）

```
# tar jxvf tftp-hpa-0.40.tar.bz2
# cd tftp-hpa-0.40
# ./configure –prefix=/usr
# make && make install
```

第二步，更改 tftpd 下载载目录

tftp 服务器的默认下载目录是/var/lib/tftpboot，我们要改为 ${HOME}/maxwit/images。打开/etc/inetd.conf，找到以“tftpd”开头的一行，将其中的/var/lib/tftpboot 改为 ${HOME}/maxwit/images：

```
tftpd dgram udp wait root /usr/sbin/in.tftpd /usr/sbin/in.tftpd –s ${HOME}/maxwit/images
```

第三步，/etc/init.d/tftpd-hpa restart

第四步，测试 tftp server

```
# cd /tmp

# echo hello > ~/images/test
# chmod 666 ~/images/test
# tftp 192.168.0.111（假定本机 IP 为 192.168.0.111）
> get test
> quit
# cat test
# rm test ~/images/test
```

3 使用Lablin源码

3.1 获取Lablin最新源码

在 MaxWit 开放实验室的开源项目主页（http://maxwit.googlecode.com）的“Source”页面上可以下载到全部源码。google 提供的默认下载方式是：

```
# cd
# svn checkout http://maxwit.googlecode.com/svn/trunk/ maxwit-read-only
```

注：Lablin 源码即 maxwit-read-only 目录必须放到${HOME} 目录或其子目录下 。

3.2 Lablin 源码目录介绍

```
|-- build                 // Lablin building Menu
|-- build-all -> build    //  编译 build 中的所有选项
|-- core                  //  公共环境变量和函数   host 端设置
|-- g-bios                // BootLoader
|-- toolchain             // cross toolchain (目前支持平台 ARM  和 MIPS)
|-- application           //  应用程序及库的编译
|-- host                  //  主机端环境的搭建
|-- kernel                //  编译 kernel
|-- document              // Lablin  使用文档
`-- readme.txt
```

Lablin Building Menu(如下)：

```
[Lablin Building Menu] (configured for s3c2440)
      1). Build Host Environment
      2). Build GNU Toolchain
      3). Build Bootloader (g-bios)
      4). Build Linux Kernel (linux-2.6.29.4)
      5). Build Basic System (busybox or coreutils)
      6). Build Lib/App (vim,perl)
      7). Build Lib/App (alsa,libmad,mpg123,madplay)
      8). Build Lib/App (MPlayer)
      9). Build Lib/App (jpeg,gif,tiff,png,fbv)
     10). Build Lib/App (SDL,DirectFB)
     11). Build Lib/App (usbutils,tslib)
     12). Build 3D Game (doom)
```

13). Testing on QEMU
14). Create File System Images (yaffs2,jffs2,cramfs,etc.)
x). Exit

Your choice[1-14]?

以下是各选项的详解:

1). Build Host Environment	设置 host 环境，这是最先要执行的
2). Build GNU Toolchain	编译 cross toolchain，支持 arm，mips（龙芯）等平台，
3). Build Bootloader (g-bios)	编译开发板的 bootloader，可根据自己开、发平台来选择。目前 g-bios 支持三星和 ATMEL 系列的开发板。
4). Build Linux Kernel	默认编译 realview 的 kernel
5). Build Basic System	编译一个基本的系统
6). Build Lib/App (vim,perl)	编译应用程序 vim 和 per
7).Build Lib/App (mpg123,madplay)	编译 mpg123，madplay 播放器及相关的依赖库
8). Build Lib/App (MPlayer)	编译 MPlayer 视频播放器
9). Build Lib/App (jpeg,gif,tiff,png,fbv)	编译看图软件 fbv 及相关的依赖库
10). Build Lib/App (SDL,DirectFB)	编译 2/3D 图形库
11). Build Lib/App (usbutils,tslib)	编译 usb 及 touchscreen 应用开发库
12). Build 3D Game (doom)	编译 3D 游戏 Doom
13). Testing on QEMU	用 qemu 测试编译好的系统（注：编译生成的文件系统在 $HOME/maxwit/rootfs 目录下）
14). Create File System Images (yaffs2,jffs2,cramfs,etc.)	创建 roofsimage，即将$HOME/maxwit/rootfs 目录下做成各种文件类型的 image，并存放到$HOME/maxwit/images 目录中

3.3 Lablin生成目录介绍

Lablin 生成文件目录是当前用户的$HOME 目录下，即${HOME}/maxwit

```
|-- build   //源码包编译的地方
|-- images //存放各种 image（kernel image、bootload image、rootfs image）
|-- rootfs   //根文件系统
|-- sysroot   //cross toolchain
`-- utils       //编译过程中所需的工具
```

3.4 安装Toolchain

第一步，打开~/maxwit-read-only/core/bmw_base，编辑“export TARGET_SOC=...”这一行，更改你的目标 SOC，若不确定，就用默认值☺

第二步，cd ~/maxwit-read-only，执行：
root@fleya-laptop:~/dev-maxwit# ./build

[Lablin Building Menu] (configured for s3c2440)

1). Build Host Environment
2). Build GNU Toolchain
3). Build Bootloader (g-bios)
4). Build Linux Kernel (linux-2.6.29.4)
5). Build Basic System (busybox or coreutils)
6). Build Lib/App (vim,perl)
7). Build Lib/App (alsa,libmad,mpg123,madplay)
8). Build Lib/App (MPlayer)
9). Build Lib/App (jpeg,gif,tiff,png,fbv)
10). Build Lib/App (SDL,DirectFB)
11). Build Lib/App (usbutils,tslib)
12). Build 3D Game (doom)
13). Testing on QEMU
14). Create File System Images (yaffs2,jffs2,cramfs,etc.)
x). Exit

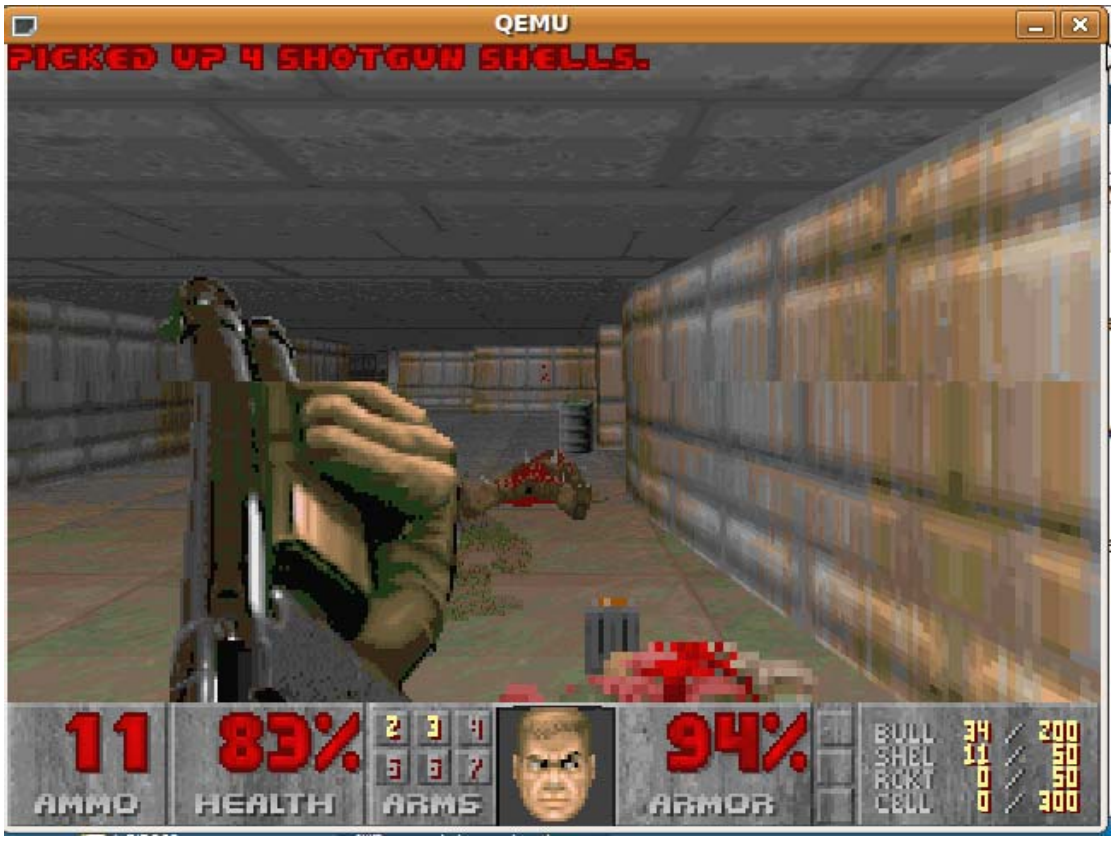
Your choice[1-14]?
先选择 1，再选择“2”，回车。这个过程比较漫长，不过在推荐的系统上一定能过，因为已测过不知多少次了☺

3.5 编译Lablin基本系统

cd ~/maxwit-read-only
./build

选择 5 Build Linux System Core
至少要编译“Basic System”，这样一个基本的嵌入式 Linux 系统就能跑起来，但想要加入更丰富的软件，还要继续编译“Libraries”、“Applications”和“Game”等其他模块
（第一次执行时，脚本会自动下载所需的第三方源码包。）
再依次选择 4、13，看到小企鹅和 console 了吗？尽情玩吧，目前 Lablin 里已经有不少有趣的小东东了。当然，最有趣也更重要的是，我们一起参与开发，一起来完善她！☺
如果你选择了 12，就可以在自己的开发板上玩游戏了。当然如果手头上没有开发板的话也

可在模拟器中玩。选择 13 一样也可以玩 3D 游戏!



4 运行Lablin（基于实际硬件平台）

4.1 编译Bootloader

这里，我们使用 g-bios 作为 Lablin 的 bootloader，当然，你也可以使用其他的 bootloader，但我们认为 g-bios 更强、更方便，可以提高整个工作效率。

BTW，我们以 ATMEL AT91SAM9263 为示例硬件平台，你也可以使用 S3C24XX 或其他平台。另外，为了简化，目前的这个文档只介绍 NFS root 方式启动，当然，Native 方式（直接从 Flash 启动）也支持得很好，你可以自己试试。

第一步，配置

cd ~/maxwit-read-only/g-bios

./configure

Platform 选择 AT91SAM9263

第二步，编译

make

会生成 g-bios-th.bin 和 g-bios-bh.bin 并已自动 copy 到~/maxwit/images 下，以备后继下载和烧录。(g-bios-th.bin 和 g-bios-bh.bin 分别对应 g-bios 上半部分和下半部分。)

4.2 编译Linux Kernel

第一步，解压 Linux Kernel
tar jxf linux-2.6.28.tar.bz2
cd linux-2.6.28

第二步，配置 Kernel
make ARCH=arm at91sam9263ek_defconfig
make ARCH=arm menuconfig
然后编译以下几个选项：
(1) “Kernel Features”→ **选上 EABI 选项，并去掉 OABI!**
(2) “Network file system” → 选上“NFS client”和“Root file system on NFS”
保存、退出

第三步，编译
make ARCH=arm CROSS_COMPILE=arm-maxwit-linux-gnueabi- –j4
cp –v arch/arm/boot/zImage ~/maxwit/images

4.3 烧录images

第一步，接上 USB 线和网线。

第二步，先将跳线拨到 off 位置，然后上电！
lsusb
你将会看到 “Atmel ….”的字样，否则，先断电，然后重做第一步和第二步。

第三步，安装开发板驱动。
首先删除所有 usbserial 模块：
lsmod | grep ^usbserial
rmmod xxx
rmmod usbserial
然后安装开发板驱动：
modprobe usbserial vendor=0x03eb product=0x6124
查看 USB 设备是否已创建：
ls /dev/ttyUSB*

第四步，安装并运行 SAM-BA
unzip sam-ba_cdc_2.7.linux_01.zip
cd sam-ba_cdc_2.7.linux_01
./sam-ba_cdc_2.7.linux_01
选择 AT91SAM9263-EK, 然后点击 “Connect”

第五步，烧写 g-bios 的上半部。

将跳线拨到 on 位置，然后执行下列操作：
“Enable Dataflash” -> Execute
“Erase All” -> Execute
“Send Boot File” -> Execute -> Select g-bios-th.bin. OK

第六步，烧写 g-bios 下半部。
可参考 g-bios 开发者手册烧录 g-bios-bh.bin
再按下开发板 Reset 键，然后选择 1（从 Flash 中启动），回车，进入 g-bios 命令行。

4.4 启动Lablin

如前所述，但为了方便起见，我们先用 TFTP + NFS 方式启动 Linux。
在 g-bios 命令行下，输入：
boot –t zImage –n
【说明】

- -t [filename]：用 tftp 方式下载指定的 kernel image
- -n [nfs_server:/nfs/path/]： 用 NFS 方式 mout rootfs。也可以加上参数，如：“-n 192.168.0.111:/path/to/nfs”。
- boot 程序具有记录功能，即，能记住用户输入的参数，换句话说，再次输入 boot 时不再需要输入参数了，除非你想重设参数☺

现在看到“maxwit login: ”的登陆提示符了吗？
用 root 用户登陆，输入 root 回车，进入系统

Lablin 里已经有不少有趣的小东东了☺ Enjoy yourself now!

这个文档可能有疏漏的地方，但实际的操作步骤我们已测过多遍，所以一定能跑起来，若有任何疑问，可以在 ChinaUnix 的 MaxWit 版块上发贴讨论，或直接给我们发 mail，不怕路远的朋友还可以直接来上海的 MaxWit 开放实验室 face-to-face 地交流！