X3399 Linux 平台 用户手册



深圳市九鼎创展科技有限公司 www.9tripod.com



www.9tripod.com 优秀嵌入式技术服务商

版权声明

本手册版权归属深圳市九鼎创展科技有限公司所有,并保留一 切权力。非经九鼎创展同意(书面形式),任何单位及个人不得擅自摘 录本手册部分或全部,违者我们将追究其法律责任。

敬告:

在售开发板的手册会经常更新,请在 http://www.9tripod.com 网站下载最新手册,不再另行通知。



版本说明

版本号	日期	作者	描述
Rev.01	2017/2/18	lqm	原始版本



技术支持

如果您对文档有所疑问,您可以在办公时间(星期一至星期五上午 9:00~12:00;下午 1:30~6:00)通过拨打技术支持电话、E-mail、留言到 BBS 论坛(http://www.xboot.org)。

网址: www.9tripod.com
 联系电话: 销售 0755-33121205 (400-003-3436) 0755-61952310
 技术支持专线: 0755-61952306
 软件专线: 0755-61952309
 硬件专线: 0755-61952307
 E - mail: supports@9tripod.com

销售与服务网络

- 公司: 深圳市九鼎创展科技有限公司
- 地址: 深圳市宝安中心区兴业路宝安互联网产业基地 B 区 3003B 室
- 邮编: 518101
- 电话: 0755-33121205 (400-003-3436) 0755-61952310
- 网址: http://www.9tripod.com
- 论坛: http://www.xboot.org
- 淘宝: <u>http://armeasy.taobao.com</u>
- 阿里: <u>http://armeasy.1688.com</u>
- 速卖通: <u>www.aliexpress.com/store/2340163</u>

QQ 群:	
技术交流 QQ 群	QQ 群号
x210/i210一群	23831259
x210/i210 二群	211127570
x4412/ibox4412一群	16073601
x4412/ibox4412 二群	211128231
X4418/ibox4418论坛	199358213
x6818/ibox6818论坛	189920370
x3288/x3399 论坛	159144256



热烈欢迎广大同仁扫描右侧九鼎创展官方公众微信号,关注有礼,您将优先得知九鼎创 展最新动态!



日求

版权声	明		2
第1章	编译]	Linux+QT	7
1.1	安	关依赖包	7
1.2	编	译 uboot	7
1.3	编	译 android 内核	7
1.4	编	译文件系统	7
1.5	生	成 update-linux.img 文件系	⁵ 统8
第2章	烧写	inux QT 映像	9
2.1	W	'indows 下烧写映像文件	
	2.1.1	驱动安装	
	2.1.2	生成统一固件 update-li	1ux.img9
	2.1.3	烧录固件 update-linux.i	ng9
	2.1.4	多设备升级固件 update	-linux.img11
2.2	liı	nux 下烧写映像文件	
	2.2.1	生成固件 update-linux.i	mg12
	2.2.2	烧录固件 update-linux.i	mg
	2.2.3	使用 upgrade_tool 指令	烧写映像14
<u></u>	2.2.4	Rkflashkit	
第3章	QT 文	件糸统的搭建	
3.1	下 	载 buildroot	
3.2	凹	」置 buildroot	
3.3	编	译 buildroot	
3.4	洪]试 QT5 默认示例	
3.5		装 QT Creator	
3.6	编	译 QT Creator 默认示例	
第4草	qttest		
4.1	伊	「用 QT_demo 测试 LED 灯	
4.2	19	」用QI_demo 测试蜂鸣奋.	
4.3	19	「用QI_demo 调 T 育 尤	
4.4	化	「用QI_demo 测试按键	
4.5	伐	H QI_demo 测试音观	
4.0	伐	HQI_demo 测试触误用.	
4.7	仅	「用QI_demo 测试 TE上	
4.8	伐	出QI_demo 测试 IF 下	
4.9	13 日本	「田QI_demo 测试 U 盈 「田QT_demo 测述重白	
4.10) 位	「用QI_demo 测试 生加」	
4.11 笛 < 辛	" 空壮,	:历 Q1_ueillo 侧试大机 ubuntu 1/ 0/ 玄弦	
おり早	、 太 し in uv	100000014.04 示沉	
おり早	mux 讶	116/広川 久小 [7] 試 mn3	
62	11 古	后台运行程序	
优秀嵌入	式技术服	服务商 论坛:	www.xboot.org 电话: 0755-33133436





6.3	中止程序的运行	44
6.4	屏幕抓图	44
6.5	挂载 TF 卡	44
6.6	挂载 U 盘	45
6.7	保存系统时钟	45
6.8	掉电保存数据到 flash	46
6.9	设置开机自动运行程序	46
6.10) 查看开发板内存信息	46
第7章	linux 应用开发示例	
7.1	Hello World	
7.2	数学函数库调用	49
7.3	多线程编程示例	
7.4	多进程编程示例	51
7.5	makefile 编程示例	
7.6	mplayer 移植	
7.7	madplayer 移植	
7.8	屏幕抓图工具 gsnap 移植	
第8章	其他产品介绍	
8.1	核心板系列	
8.2	开发板系列	
8.3	卡片电脑系列	



第1章 编译 Linux+QT

安装依赖包 1.1

使用如下命令安装所需的软件包:

sudo apt-get update

sudo apt-get install git-core gnupg flex bison gperf libsdl1.2-dev libwxgtk3.0-dev build-essential zip curl zlib1g-dev gcc-multilib g++-multilib genromfs libc6-dev-i386 libncurses5-dev x11proto-core-dev libx11-dev ccache libgl1-mesa-dev libxml2-utils xsltproc unzip gperf lsb-core lib32z1-dev lib32ncurses5-dev lzop

编译 uboot 1.2

说明: OT 系统使用的 uboot 与内核与安卓完全相同,如果编译过 android 的 uboot 和内 核,则本小节步骤可以省略。

在 android 源码目录下执行如下命令编译 uboot,编译完成后映像文件 RK3399MiniLoaderAll_V1.05.bin(因版本不同,名称不一定相同)以及 uboot.img 会释放到 out/release 目录。

./mk.sh -u

1.3 编译 android 内核

说明:QT系统使用的 uboot 与内核与安卓完全相同,如果编译过 android 的 uboot 和内 核,则本小节步骤可以省略。

在 android 源码目录下执行如下命令编译 linux 内核,编译完成后映像文件 kernel.img、 resource.img 会释放到 out/release 目录。

./mk.sh -k

1.4 编译文件系统

在 android 源码目录下执行如下命令编译 linux 映像文件,编译完成后 linux 映像文件 linux-rootfs.img 会释放到 out/release 目录。

./mk.sh -b

在执行./mk.sh-b编译 buildroot 时,默认编译会少一些插件或库,会弹出一些错误信息。

NFIG AUTOCONFIG=/home/lqm/buildroot/output/build/buildroot-config/auto.comf 空加件改库, 会弾 f.h KCONFIG TRISTATE=/home/lqm/buildroot/output/build/buildroot-config/tristate.config BR2_CONFIG=/home/lqm/ external SKIP_LEGACY= /home/lqm/buildroot/output/build/buildroot-config/tristate.config BR2_CONFIG=/home/l ectory '/home/lqm/buildroot' t' on your build machine /dependencies.mk:28: recipe for taxate ild machine .mk:28: recipe for target 'core-dependencies' failed

该提示信息表明编译需要 git 包,执行如下指令安装:

sudo apt-get install git

Your Buildroot configuration uses pre-built tools for the x86 architecture, but your build machine uses the x86-64 architecture without the 32 bits compatibility
library.
If you're running a Debian/Ubuntu distribution, install the libc6-i386,
lib32stdc++6, and lib32z1 packages (or alternatively libc6:i386,
libstdc++6:i386, and zlib1g:i386).
For other distributions, refer to the documentation on how to install the 32 bits
compatibility (toraties.
support/dependencies/dependencies.mk:28: recipe for target 'core-dependencies' failed
make: *** [core-dependencies] Error 1
lqm@ubuntu:~/buildroot\$

优秀嵌入式技术服务商

论坛: www.xboot.org



该提示表明标准的 C++库没有安装,执行如下指令安装:

sudo apt-get install lib32stdc++-4.9-dev



该提示表明缺少 libz 库,执行如下指令安装:

sudo apt-get install lib32z1

再执行./mk.sh -b 指令即可正常编译了。编译完成后,最终打包好的文件系统 linux-rootfs.img 存放在 out/release 目录下。它包含了 QT5.6 的标准库,以及一些常用的 QT 示例。

1.5 生成 update-linux.img 文件系统

在 android 源码目录下执行如下命令生成单一 android 映像文件 update-linux.img: ./mk.sh -U

update-linux.img 为整个 QT 系统升级文件的单一映像,包括了 uboot,内核,文件系统等。注意,生成 update-linux.img 的先决条件是已经成功编译了 uboot,内核和 linux 文件系统,缺一不可。



第2章 烧写 linux QT 映像

2.1 Windows 下烧写映像文件

2.1.1 驱动安装

解压光盘 tools\x3399 烧写工具\windows 目录下的 DriverAssitant_v4.5.zip 文件,打开 "DriverInstall.exe",点击"驱动安装",提示安装驱动成功即可。

● 瑞芯微驱动助手	×
驱动安装	驱动卸载
▶ 瑞芯微驱动助手	DriverInstall
驱动安装	安装驱动成功.
	<u>确定</u>

注意事项:

1.目前支持的操作系统包括: XP,Win7_32,Win7_64,Win8_32,Win8_64。

2.XP 系统在驱动安装完后,若还提示"发现新设备", 安装驱动时选择"自动安装"。 3.若之前已经安装过老版本驱动,请先点击"驱动卸载"后再进行"驱动安装"。

2.1.2 生成统一固件 update-linux.img

我们已经在 mk 脚本中集成了生成统一固件的方法,在编译 linux 文件系统时(./mk.sh -b -U), 会自动生成 update-android.img,并释放到 out/release 目录。

2.1.3 烧录固件 update-linux.img

解压光盘 tools\x3399 烧写工具\windows 目录下的 AndroidTool_Release_v2.38.zip 文件, 得到 AndroidTool_Release_v2.38 文件夹,打开 AndroidTool.exe,选择"升级固件"选项卡, 点击"固件",在弹出窗口中选择已经生成的 update-linux.img 文件,如下图所示。

	深圳市九鼎创展科技有限公司	www.9tripod.com
		优秀嵌入式技术服务商
Android开发工具	Ę v2.38	
載鏡像 升级国	1件 高级功能	
固件	升级 切换 擦涂Flash	
固件版本:	6.0.01 Loader版本: 1.05 芯片信息: RK330C	
固件:	D:\DVD\x3399\DVD_X3399\Image\update=linux.img	
Demo		
	没有友现议备	

工具配置好后,插上开发板的 TypeC 线、串口线以及 5V DC 电源线,按下 RECOVERY(VOL+)键不放的同时,长按 POWER 键开机,在串口终端会有如下打印:



当 uboot 检测到 RECOVERY 键按下时,会有"rockusb key pressed"的打印提示,这时, 在烧录工具界面会提示发现一个 LOADER 设备,然后点击升级,即可开始升级过程(注: 如果提示发现一个 ADB 设备,点击切换按钮切换成 LOADER 设备即可)。



🔀 Android开发工具	Į v2.38	
下载镜像 升级固	1件 高级功能	
固件	升级 切换 擦涂Flash	
固件版本:	6.0.01 Loader版本: 1.05 芯片信息: RK330C	
固件:	D:\DVD\x3399\DVD_X3399\Image\update-linux.img	
🕅 Demo		
	友现一个LOADER设备	

点击升级,稍等片刻,烧写成功界面如下:

🔨 Android开发工!	Į v2.38	
下載遺像 升级团 日本	件 高級功能 升级 切換 擦涂Flash	测试设备开始 测试设备成功 校验芯片开始 校验芯片成功 装取FlashInfo开始
固件版本:	6.0.01 Loader版本: 1.05 芯片信息: RK330C	获取FlashInfo成功 准备IDB开始 准备IDB成功
固件:	D:\DVD\x3399\DVD_X3399\Image\update=linux.img	下载IDB开始 下载IDB开始
Demo Demo		重启设备成功 等待Loader开始 等待Loader开始 则试设备开始 则试设备开始 则试设备成功 下载团件开始 正在下载团件(100%) 正在校验固件(100%) 下载团件成功 重启设备开始 重启设备成功
	没有发现设备	

2.1.4 多设备升级固件 update-linux.img

该工具适合用户批量刷机,可以同时给多台设备烧录固件。

解压光盘 tools\x3399 烧写工具\windows 目录下的 FactoryTool-v1.42e.rar 文件,打开 FactoryTool.exe,点击"固件"选择 update-android.img,勾选"升级",点击"启动",如下 图所示:

步骤1点击固件,选择需要使用的 update-linux.img;

步骤2点击启动(选择升级按钮);

步骤 3 连接开发板 USB、DC 电源,按下 recovery 键(对应 VOL+键),对应 USB 口发 现设备,并实现自动升级;然后重复步骤 3 即可同时升级第二台、第三台设备,升级成功或 者失败的设备会在两边的列表中列出,移除成功或者失败的设备后可以继续连接需要升级的 设备。

>> 深圳 7	市九鼎创展	科技化	限公	司	www.9	tripod.com
					优秀嵌入。	式技术服务商
↓ 工厂工具 v1.33				R.Par 10	- 11/1-	
(固件) 启动	◎ 升级 ◎ 修复	Demo拷贝	. 😽 Ge	(选择 🔶 退出		
固件: D:\Tools\RKTools\windo	ws\AndroidTool_Release_v2.3\re	ockdev\update.img		固件版本:4.4.02		
步骤1 步骤2				Loader版本:0.02		
Demo				芯片信息:RK32		
ID 失败 4	设备列表	设备类型	ID	升级信息	ID	成功
	□					
	RootHub20					
	Port[1]	Hub	3			
	+++ Port[2]					
	++> Port[3]					
	Port[4]					
	Port[2]					
	RootHub20				_	
	Port[1]	Hub	1			
	Port[1]					
	Port[2]	Loader	6	▶ 上啷3	_	
	- Port[3]			JV JAK	_	
	Port[4]					
	Port[5]				_	
	Port[6]				_	
	Port[2]				_	
					1	
友情提示:						
					dt the	00000
1. 弗一次使用,称识USB端口万法	: 连接设备, 上具显示设备后记家	R绑定ID.标识所有	USB端口.		成功:	00000
2.插入设备升级,要等到工具开始	台升级后再接入下一台.				1. 51	00000
3. 升级过程中, 绿灯亮, 接入设备	,红灯亮,不要拔插设备.				失败:	00000
A 升级结束 成功以绿色背具思う	- 生断以红色皆具思示					00000
3. / 1 秋垣木, 10 タリム 林己目京亚/					总共:	00000
5. 所有成功设备会在右侧表格中	显示,所有失败设备会在左侧表	格中显示.				

2.2 linux 下烧写映像文件

2.2.1 生成固件 update-linux.img

我们已经在 mk 脚本中集成了生成统一固件的方法,在编译 android 文件系统时(./mk.sh -s),会自动生成 update-linux.img,并释放到 out/release 目录。

2.2.2 烧录固件 update-linux.img

工具路径: tools\x3399 烧写工具\linux\Linux_Upgrade_Tool_v1.24.zip

在升级之前将 update-linux.img 拷贝到 upgrade_tool 相同目录下,运行 upgrade_tool(需要 do)

sudo)

<u>work@ubuntu:~/Linux_Upgrade_Tool_v1.24</u> cp \$sourcetree/out/release/update-android.img . work@ubuntu:~/Linux_Upgrade_Tool_v1.24\$ sudo _./upgrade_tool

执行结果如下图,发现设备列表,输入要升级的 DevNo(设备号)选择设备:

```
      pubuntu: ~/3288/Linux_Upgrade_Tool_v1.2

        1 4)

      Download IDB Start

        ^Cwork@ubuntu:~/3288/Linux_Upgrade_Tool_v1.2$ sudo ./upgrade_tool
        List of rockusb connected
        DevNo=1 Vid=0x2207,Pid=0x320a,LocationID=128 Loader
        Found 1 rockusb,Select input DevNo,Rescan press <R>,Quit press <Q>:1
```

选择设备后弹出工具使用菜单如下图, 左侧是功能描述, 右侧是命令语法, 升级相关操作都在 upgrade command 列表下,忘记命令语法可以输入H进行查看,清屏输入CS,退出按Q。



www.9tripod.com

优秀嵌入式技术服务商

ubuntu: ~/3288/Linux_Upgrade_Tool_v1.2	t₊	■))	14:0
work@ubuntu:~/3288/Linux_Upgrade_Tool_v1.2\$ sudo ./upgrade_tool List of rockusb connected DevNo=1 Vid=0x2207,Pid=0x320a,LocationID=105 Loader Found 1 rockusb Select input DevNo Rescan press <r> Ouit press <</r>	0>:1		
i ound i rockubb, seccee enpae berno, keseun press sko, gate press s	Q- • -		
Tool Usage			
Help: H			
Quit: Q			
Version: V			
Clear Screen: CS			
Upgrade Command			
ChooseDevice: CD			
SwitchDevice: SD			
UpgradeFirmware: UF <firmware></firmware>			
UpgradeLoader: UL <loader></loader>			
DownloadImage: DI <-p -b -k -s -r -m image> [parameter	file]		
DownloadBoot: DB <loader></loader>			
EraseFlash: EF <loader firmware=""></loader>			
LowerFormat: LF			
Protessional Command			
ResetDevice: RD [subcode]			
ResetPipe: RP [pipe]			
ReadContpinio, RCI ReadSoctory RS (Read Soctory and International Intern	101		
WriteSector: WS <beginsec> <sectorlen> [-decode] [rt</sectorlen></beginsec>	ce]		
Readi RA: RI < ReginSec> <sectoriens [file]<="" td=""><td></td><td></td><td></td></sectoriens>			
WriteLRA: WL <beginsec> <file></file></beginsec>			
FraseBlock: FB <cs> <beninblock> <blokclen> [Force</blokclen></beninblock></cs>	1 _		
	-		

- CD 命令: 选择设备, 当执行的命令有包含设备重启操作时, 需重新选择设备, 当 改变操作设备时需重新选择
- SD 命令:msc 切换到 rockusb 升级模式。 当切换执行成功后, 需要重新选择设备
- UF 命令:升级完整 update.img 固件,当执行成功后需要重新选择设备
- UL 命令:升级 loader 功能,当执行成功后需要重新选择设备
- DI 命令:下载单独 image 镜像到指定扇区,例如升级 kernel.img 或者 system.img 都可以 直接使用此功能.例如下载 kernel.img: DI -k kernel.img parameter //如果之前通过 DI 下 载过 parameter,则再下载 kernel.img 时就可以不用指定最后的 parameter 参数
- DB 命令:下载 boot,在 maskrom 状态下,可以通过此功能,让 maskrom 设备进行 Rockusb 协议通讯
- EF 命令:擦除整个 nandflash
- LF 命令:低格保留块后面区域,只有在 loader 模式下使用 执行 uf update-linux.img 开始更新固件,下图为更新过程截图:

Rockusb>uf update-linux.img

```
Rockusb>uf update.img
Loading firmware...
Support Type:RK32 FW Ver:5.0.00 FW Time:2016-07-19 00:02:28
Loader ver:2.30 Loader Time:2016-05-27 11:28:11
Download Image Total(445737K),Current(12157K)
```



www.9tripod.com 优秀嵌入式技术服务商

Rockusb>uf update.img Loading firmware... Support Type:RK32

FW Ver:4.4.02 FW Time:2016-02-20 15:28:14 Loader ver:2.19 Loader Time:2016-02-19 18:04:38 Upgrade firmware ok. Rockusb>

备注: 也可通过配置 config.ini 文件配置升级映像文件,只需输入 UF 即可升级,请用 户自行尝试。

2.2.3 使用 upgrade tool 指令烧写映像

上一节我们介绍了通过 upgrade_tool 烧写统一固件 update-linux.img 的方法,熟悉三星 平台的开发者会发现,这种方法并不是很高效,真正操作起来,它远没有 fastboot 工具来的 迅速。其实, upgrade_tool 工具同样支持类似于 fastboot 的烧写方式。

为了烧写方便,可以在 mk 脚本中,默认在编译系统时,将烧写工具 upgrade_tool 拷贝 到 out/release 目录。

第一步:打开串口终端,并打开 minicom,用于适时监控串口调试信息:

第二步:按住 RECOVERY 键,连接 USB OTG 线和电源线,这时 uboot 打印信息将会 提示已经进入 USB 下载模式。如果接通电源后没来得及按住 RECOVERY 键,在按住 RECOVERY 键的同时,再按下复位键即可。

第二步:打开第二个串口终端,进入 out/release 目录;

第三步:在 out/release 目录下敲击如下指令,烧写相应的映像。

sudo upgrade_tool	di	-k kernel.img(烧写内核)
sudo upgrade_tool	di	-s system.img(烧写文件系统)
sudo upgrade_tool	di	resource resource.img(烧写资源文件)
sudo upgrade_tool	di	-r recovery.img(烧写急救文件)
sudo upgrade_tool	ul	RK3399MiniLoaderAll_V1.05.bin (烧写 bootloader)
sudo upgrade_tool	di	uboot uboot.img parameter.txt(烧写 uboot, 必须指定 parameter.txt)
sudo upgrade_tool	di	trust trust.img parameter.txt(烧写 trust, 必须指定 parameter.txt)
sudo upgrade_tool	uf	update-linux.img(烧写统一固件)

2.2.4 Rkflashkit

rkflashkit 有图形界面,后加了命令行支持,更是好用。

work@ubuntu:~/rktool\$ sudo apt-get install build-essential fakeroot

work@ubuntu:~/rktool\$ git clone https://github.com/linuxerwang/rkflashkit

work@ubuntu:~/rktool\$ cd rkflashkit

work@ubuntu:~/rktool\$./waf debian

work@ubuntu:~/rktool\$ sudo apt-get install python-gtk2

work@ubuntu:~/rktool\$ sudo dpkg -i rkflashkit_0.1.4_all.deb

注意: rkflashkit_0.1.4_all.deb 会因版本更新,版本数字可能会有所变化,如果执行失败, 执行 ls 命令查看下即可。

work@ubuntu:~/rktool/\$ sudo rkflashkit

如下是图形界面,在 Devices 下选择设备,选择要烧写的分区和对应的映像文件,点击 Flash image 即可。

优秀嵌入式技术服务商



www.9tripod.com

优秀嵌入式技术服务商

😣 🖻 🗊 RkFlashKit	
Devices	MAGIC: 0x5041524B
0x2207:0x320a 选择要烧写设备:	ATAG: 0x60000800 MACHINE: 3288 CHECK_MASK: 0x80
NAND Partitions	PWR_HLD: 0,0,A,0,1 #KERNEL_IMG: 0x62008000
uboot (0x00002000@0x00002000) ‡	#FDT_NAME: rk-kernel.dtb #RECOVER_KEY: 1,1,0,20,0
Image File to Flash 选择要烧写的分区	oot.hardware=rk30board androidboot.setinux=permissive androidb oot.hardware=rk30board androidboot.console=ttyFlQ0 init=/init initrd=0 x62000000,0x00800000 mtdparts=rk29xxnand:0x00002000@0x00002000
grade_To <mark>ol_v1.2/rockdev/Image/boot.img</mark> Choose	(uboot),0x00002000@0x00004000(misc),0x00008000@0x000060000(resou rce),0x00008000@0x0000e000(kernel),0x0001000@0x00016000(boot),0
Actions 选择要烧写的img文件	x00010000@0x00020000(ecche),0x0002000@0x00090000(kparic),0x0030 00040000@0x00050000(cache),0x00002000@0x00392000(kparic),0x0030 0000@0x00092000(svstem) 0x00008000@0x00392000(metadata) 0x0020
Flash image	0000@0x0039A000(userdata),0x00020000@0x0059A000(radical_update), -@0x005BA000(user)
Compare partition with image file	Partitions:
Backup Partition	uboot (0x00002000 @ 0x00002000) 4 MiB misc (0x00002000 @ 0x00004000) 4 MiB resource (0x00008000 @ 0x00006000) 16 MiB
Erase Partition	kernel (0x00008000 @ 0x0000E000) 16 MiB boot (0x00010000 @ 0x00016000) 32 MiB recovery (0x00010000 @ 0x00026000) 32 MiB
Reboot Device	backup (0x0001A000 @ 0x00036000) 52 MiB cache (0x00040000 @ 0x00050000) 128 MiB
Clear Log	system (0x00030000 @ 0x0092000) 1536 MiB metadata (0x00008000 @ 0x0092000) 16 MiB
	userdata (0x00200000 @ 0x0039A000) 1024 MiB radical_update (0x00020000 @ 0x0059A000) 64 MiB user (0x01765000 @ 0x005BA000) 11978 MiB Done!

该工具也支持命令行,使用 help 命令查看使用方法

work@ubuntu:~/rktool/rkflashkit\$ rkflashkit --help

Usage: <cmd> [args] [<cmd> [args]...]

part	part List partition		
flash @-	<partition> <image file=""/></partition>	Flash partition with image file	
cmp @<	PARTITION> <image file=""/>	Compare partition with image file	
backup	@ <partition><image file=""/></partition>	Backup partition to image file	
erase (@ <partition></partition>	Erase partition	
reboot	Re	boot device	

For example, flash device with boot.img and kernel.img, then reboot:

sudo rkflashkit flash @boot boot.img @kernel.img kernel.img reboot work@ubuntu:~/rktool/rkflashkit\$



第3章 QT 文件系统的搭建

前面章节的介绍,都是九鼎创展工程师已经移植的文件系统包,如果换一个平台,一切 从零开始,我们如何构建 linux QT 文件系统呢?本章节将会带您一步步搭建 linux 文件系统。

3.1 下载 buildroot

在 buildroot 官网下载最新的 buildroot 包,下载地址如下:

http://buildroot.uclibc.org/download.html

本文使用的是 buildroot 2016.02, 用户下载最新版本与本文介绍版本配置可能略有不同, 根据下文配置适当修改即可。

3.2 配置 buildroot

将下载的 buildroot 包拷贝到 ubuntu 系统用户目录并解压,得到 buildroot 目录,通过命 令终端进入 buildroot 目录,执行 make menuconfig,进入配置界面:

😑 lqm - TeamViewer - 免费许可证(仅非商业用途	b)		
lqm@lqm: /home/DISK1/x3399_marsh	mallow/buildroot		🚽 😤 7°C 👜 🧮 🔃 📣 19:20 🕸
lqm@lqm: /home/DISK1/x339	/9_marshmallow/buildroot	× lqm@lqm: /home/DISK1/x3399_marshmallow	×
/home/DISK1/x3399_mars	hmallow/buildroot/.config - Buildroot 2016.08.1-gf	F57e812 Configuration	
Arrow keys navigat selectes a feature selected [] feat	Buildroot 2016-08. e the menu. <enter> selects submenus> (or en t, while ≺N> will exclude a feature. Press <esc>∈ ure is excluded</esc></enter>	l-gf576812 Configuration bty submenus), Highlighted letters are hoti Esc> to exit, <7> for Help, for Search. Lege	keys. Pressing <y> end: [*] feature is</y>
	Target options > Build options > Toolchain > System configuration >		
P	Kernel> Target packages>		
2	Bootloaders> Bootloaders> Host utilities> Legacy config options>		
±			
	<pre><select> < Exit > <</select></pre>	Help > < Save > < Load >	

进入 Target options 菜单,



😁 lqm - TeamViewer - 免费许可证(仅非商业用途)	
lqm@lqm: /home/DISK1/x3399_marshmallow/buildroot	🔜 👘 🖄 👘 🛄 🤐 👘 19:22 🔅
Iqm@lqm: /home/DISK1/x3399_marshmallow/buildroot	× lqm@lqm: /home/DISK1/x3399_marshmallow
/home/DISK1/x3399_marshmallow/buildroot/.config - Buildroot 2016	
> larget options	Target options
Arrow keys navigate the menu. <enter> selects submenus></enter>	• (or empty submenus). Highlighted letters are hotkeys. Pressing <y></y>
selectes a feature, while <n> will exclude a feature. Press</n>	s <esc><esc> to exit, <? > for Help, for Search. Legend: [*] feature is</esc></esc>
Target Architecture	(ELF)>
<pre><select> < Exit</select></pre>	:> < Help > < Save > < Load >

在 Target Architecture 中选择 ARM(little endian),在 Target Binary Format 中选择 ELF, 退回上一级,进入 Toolchain 目录,按下图配置:

😑 lqm - Tea	mViewer - 免费许可证(仅非商业用途)		- • ×
lqm@lqm	: /home/DISK1/x3399_marshmallow/buildroot		🜒 19:23 🔅
0	lqm@lqm:/home/DISK1/X3399_marshmallow/buildroot	X lqm@lqm:/home/DISK1/x3399_marshmallow	×
	> Toolchain		
	Arrow keys navigate the menu. <enter> selects submenus> (or</enter>	empty submenus). Highlighted letters are hotkeys. Pressing <y></y>	
	<pre>selectes a feature, while <n> will exclude a feature. Press <est selected [] feature is excluded</est </n></pre>	<pre>c><esc> to exit, <?> for Help, for Search. Legend: [*] feature is</esc></pre>	
	Toolchain type (External Toolchain (Linaro Aarchó4 Toolchain criaint (Toolcha)	toolchain)> 2016.02)>	
	[*] Copy gdb server to the Tai	rget	
· >_ •	() Target Optimizations		
	<pre>() Target linker options [] Register toolchain within</pre>	Eclipse Buildroot plug-in	
	L		
	<select> < Exit ></select>	< Help > < Save > < Load >	

退回上一级,进入 System configuration 目录,作如下配置:





退回上一级,进入 Target packages 目录,再进入 Graphic libraries and applications (graphic/text)目录,选择 Qt5,如下图所示:

🔁 lqm - Tea	mViewer - 免费许可证(仅非商业用途)		- • • ×
tqm@tqm	lqm@lqm: /home/DISK1/x3399_marshmallow/buildroot	× lgm@lgm:/home/DISK1/x3399 marshmallow	×
0	/home/DISK1/x3399_marshmallow/buildroot/.config - Build	root 2016.08.1-gf57e812 Configuration	
	Arrow keys navigate the menu. <enter> selects a feature, while <<< while <<< while <</enter>		
	[] directTb exa [] tbdump (Fram [] tbgrab [] tb-test-app [] tbterm [] tbv [] treerdp	mples ebuffer Capture Tool)	
2	[] imagemagick *** linux-fu [] LiTE (toolbo [] mesa3d	sion needs a Linux kernel to be built *** x engine)	
	[] ocrad [] psplash [] SDL [* odl2		
	[*] DirectFB v *** X11 vi [*] sdl2_gfx [*] sdl2_image	ideo driver deo driver needs X.org ***	
	[*] sdl2_ttf *** Other GU [] Ot [*] Ot5>	Is ***	
	*** 07 lühra [] qextserialpo [] qison [] quazip [] qwt [] rekut	ries and helper libraries *** rt	
	[] weston [] X.org X. Wind [] ×keyboard-co	ow System nfig	
	<select></select>	< Exit > < Help > < Save > < Load >	

注意不要选 Qt, 它对应 QT4.8 版本。进入 Qt5 菜单, 按下图配置:



🔁 lqm - T	TeamViewer - 免费许可证(仅非商业用途)			
lqm@lq	m:/home/DISK1/x3399_marshmallow/buildroot	■ × ×	📩 7°C 🚵 🗮 🔃 🤅	👀) 19:25 🔅
	lqm@lqm: /home/DISK1/x3399_marshmallow/buildroot × lqm@	lqm: /home/DISK1/x3399_marshmallow		×
Q	<pre>/home/DISK1/x3399_marshmallow/buildroot/.config - Buildroot 2016.08.1-gf57e812 C > Target packages > Graphic libraries and applications (graphic/text) > 015</pre>	onfiguration		
	Arrow keys navigate the menu. <enter> selects submenus> (or empty subme selectes a feature, while «N> will exclude a feature. Press <esc> to e selected [] feature is excluded qtSbase [*] Approve free license [() config file [*] Compile and install examples (wit [] MySQL Plugin [] PostgreSQL Plugin [] PostgreSQL Plugin [] PostgreSQL Plugin</esc></enter>	nus). Highlighted letters are h xit, for Help, for Search. Lu h code)	otkeys. Pressing <y> egend: [*] feature is</y>	-
	<pre>splite 3 support (pt Splite)*- gut module [*] widgets module [*] widgets module *** opend.support needs an Ope [*] Itnuxfb support [*] directfb support *** X.org XEB backend available *** opt 5 backend available *** opt 5 backend available f(linuxfb) Default graphical platform [*] off support [*] default graphical platform [*] off support [*] PBG support [*] PBG support [*] PBG support [*] Bnable fIU support [*] enable fIU support [*] enable fIU support [*] offsingstion [*] offsensors [*] offsensors [*] offserialbus *** offserialbus</pre>	> nGL-capable backend *** if X.org is enabled *** DpenGL and EGL are enabled *** /or bluez(5)_utils *** < Save > < Load >		
6				
انتصب				
🔁 lqm - T	TeamViewer - 免费许可证(仅非商业用途)			
lqm@lqı	m: /home/DISK1/x3399_marshmallow/buildroot		- 😤 7°C 🚵 🗎 🔱 4	()) 19:25 🏨
0	lqm@lqm:/home/DISK1/x3399_marshmallow/buildroot × lqm@	Iqm: /home/DISK1/x3399_marshmallow		×
Y	/home/DISK1/x3399_marshmallow/buildroot/.config - Buildroot 2016.08.1-gf57e812 C	onfiguration		
	Qt5			1
	Arrow keys navigate the menu. <enter> selects submenus> (or empty subme selectes a feature, while <a> will exclude a feature. Press <esc><esc><to e<br="">selected [] feature is excluded</to></esc></esc></enter>	nus). Highlighted letters are h xit, for Help, for Search. Lu	otkeys. Pressing <y> egend: [*] feature is</y>	

<pre> Solite 3 support (Qt SQLite)> gut module [*] wingets multe "** OpenCL support needs an OpenCL-capable backend **** "** OpenCL support needs an OpenCL-capable backend **** "*** Qts backend available if X.org is enabled **** "*** gifs backend available if OpenCL and EGL are enabled **** "time gifs backend available if OpenCL and EGL are enabled **** "time gifs backend available if OpenCL and EGL are enabled **** "time gifs backend available if OpenCL and EGL are enabled **** "time gifs backend available if OpenCL and EGL are enabled **** "time gifs backend available if OpenCL and EGL are enabled **** "time support "openCL support "openC suppo</pre>
eSelects < Exit > < Help > < Save > < Load >

退回 menuconfig 的开始界面,进入 Filesystem images 菜单,作如下配置:





到此, buildroot 配置完成。默认配置保存在 buildroot 根目录的.config 中,我们可以备份该配置文件,以防后续配置出错。执行如下指令备份配置文件:

cp .config x3399_defconfig

3.3 编译 buildroot

配置完成后,执行 make 指令即可编译 buildroot 了。编译 buildroot 会会依赖一些第三方 插件和库,在第一章的编译文件系统小节中,已经给出了需要安装的包,在编译之前需要提 前安装,否则会报错。编译完成后,文件系统映像 rootfs.ext2 会生成到 output/images 目录。

3.4 测试 QT5 默认示例

将 uboot,内核,文件系统烧写进开发板,进入 linux 文件系统后,可以进入 /usr/lib/qt/examples 目录测试 QT 示例。

进入 gui/analogclock 目录,执行 analogclock 文件,指令如下:

./analogclock &

这时,在开发板上可以看到有一个时钟图案被绘制出来(会覆盖默认的 QTTEST 程序),如下图所示:



www.9tripod.com

优秀嵌入式技术服务商



进入 gui/rasterwindow 目录,执行 rasterwindow 文件,指令如下:

./ rasterwindow &

这时,在开发板上可以看到一个标注有 QWindow 的图案被绘制出来,如下图所示:

	Tfcard Test	
QWindow		
Accord Ter	Tfcard Test	instea
-	Tfcard Test	

进入 qpa/windows 目录,执行 windows 文件,指令如下: ./windows &

优秀嵌入式技术服务商

论坛: www.xboot.org



这时,在开发板上可以看到有三幅图案被绘制出来,如下图所示:



进入 sql/drilldown 目录,执行如下指令:

./drilldown &

这时,在开发板上可以看到有四个 QT 画面被绘制出来,如下图所示:

Ot Quek	®	
Ot Ot Ouick		
Qt Creator Qt Project		

进入 sql/books 目录,执行如下指令:

./books &

这时,在开发板上可以看到有一个对话框被绘制出来,如下图所示:



www.9tripod.com 优秀嵌入式技术服务商

Books					
Title	hor Na	Genre	Year	Ratinț	ıils
Details				<u>></u>	
Title:	Found	ation			ment just hole lot nteresting.
Author:	lsaac A	Asimov		•	ut for If with a
Genre:	Scienc	e Fictio	n	•	C evaluation
Year:	1951			▲ ▼	e less. Create more.
Rating:	3			<u>^</u>	

进入 sql/masterdetail 目录,执行如下指令:

./masterdetail &

这时,在开发板上可以看到有一个对话框绘制出来,如下图所示:

<u>File</u> <u>H</u>elp

Artist	Details
<all></all>	With Or
Album	cross-platform development just got a whole lot more interesting.
title artist year 🕯	Find out for yourself with a free Oft evaluation
S Ane Brun 2003	
A Ane Brun 2005	Code less. Create more.
-	

默认 buildroot 编译出了很多示例,这里不带一一列举,有兴趣的读者可以自行尝试。

3.5 安装 QT Creator

通常我们使用 QT Creator 创建基于 QT 的工程。在 QT 官网下载最新的 QT 安装包,下载地址如下:

<u>http://download.qt.io/official_releases/qt/</u> 打开链接页面如下:

优秀嵌入式技术服务商

论坛: www.xboot.org

http://download.qt.io/official_releases/qt/ ② 敬游游戏 論 ALL游戏 節 海宝特素 ≧ Google □ HA0123网址 Links - □ 大喜花下乾姑 □ 海宝购物 □ 网址大全 □ 软件下載 即 账房 10 350网店器 Qt Home Bug Tracker Qt Home Bug Tracker ▲ Parent Directory - ■ 5.5// 01 kd 2015.00:13	文 不 AK 子 M 务平台・后 Code Review Pla
http://download.qt.io/official_releases/qt/ ● 敬游游戏 ▲ ALL游戏 ③ 淘宝特索 ③ Google HA0123网址 Links 大喜花下载站 淘宝网物 网址大金 软件下载 ④ 账房 ⑤ 350网店器 Qt Home Bug Tracker Qt Home Bug Tracker ▲ Parent Directory -	务平台-后… Code Review Pia
② 微游游戏 肇 ALL游戏 题 海宝特卖 S Google → HA0123网址 Links - 大雪花下號站 海宝购物 → 网址大全 → 软件下號 题 账房 ■ 350网店服 Qt Downloads Qt Home Bug Tracker Name Last modified Size ↑ Parent Directory - ● 5.5/ 01.11/2015.00.13	资平台-后 Code Review Pla
Qt Home Bug Tracker Name Last modified Size • Parent Directory - • 55/ 01 bit 2015 09:13	Code Review Pla
Name Last modified Size ↑ Parent Directory - ■ 5.5/ 01.1xl 2015.09.13	
Name Last modified Size ↑ Parent Directory - ← 55/ 01 bit 2015 00:13	
	Metadata
⇒ 5.5/ 01 Jul 2015 00:13	
3 .57 01-301-2013 09.13	
■ 5.4/ 02-Jun-2015 07:53 -	
■ 5.3/ 16-Sep-2014 08:45 -	
■ 5.2/ 24-Feb-2014 13:07 -	
■ 5.1/ 06-May-2014 12:44 -	
■ 5.0/ 03-Jul-2013 11:57 -	
■ 4.8/ 25-May-2015 17:55 -	

点击 5.5,下载安装包,得到名为 qt-opensource-linux-x64-5.5.0-2.run 的文件,将它拷贝 到 ubuntu 的用户目录,使用如下指令安装:

./ qt-opensource-linux-x64-5.5.0-2.run

安装完成后,QT Creator 也就安装完成了。

3.6 编译 QT Creator 默认示例

点击 ubuntu 图标,输入 qt,将会查找含有 qt 的文件,同时,安装好的 QT Creator 也会 被列出来,如下图所示:



📵 Ubunt	tu 64 位 - ˈ	VMware V	Vorkstation								×
文件(E)	编辑(E)	查看(V)	虚拟机(<u>M</u>)	选项卡(工)	帮助(出)	-	P P		2 🗖		
🗗 Ubur	ntu 64 位	×									
800	•							🌧 25°	े 🛅 📬	⊲)) 4:11 AM	чţ
0	10	at			Particular I					Filter resul	ts ⊧
		Applic	ations		-	Refer TO					
		Арриса				And a local division of the					_
		0									
(9)		Y	1								
		Ot Cr	eator								
· -		(Comm	nunity)								
100		Files &	Folders	See fewer	results 🔻						
14			_							and the second s	
FA		- 20									
G				Cold Barry						C++	
E		qtt	est		qttest	qttest		Qt5.5.0		wgBeep.cpp	
-			Contraction							_	
< 1										(1)	
12		c+	+			No. State		BINGST.		17	
		wgKe	у.срр		qttest	qt5.sh		qttest.pro	q	ttest.pro.user	
		-								and a	
-239A		- 11	11			tor.oz		1411		1111	
								0			
要將輸入這	自向到该虚	拟机, 请 将	鼠标指针移入	其中或按 Ct	rl+G。			in (0 🚱 🗖 🖬		

点击 QT 图标, QT Creator 将会运行,如下图所示:



回 Ubuntu	i 64 位 - VMv	vare Workstation			- • •
文件(E)	编辑(E) 查	看(<u>V)</u> 虚拟机(<u>M</u>) 选项卡(<u>T</u>) 帮助(ш 📕 - 🖶 🖓 💭 💭	🗖 🚍 🗮 🔯 🛅	
🕞 Ubunt	tu 64 位 ×				
× _ 1	Termina	al File Edit View Search Te	erminal Help	🌧 25°C 🛗 🐧 🗤	4:14 AM 🔱
	<u>F</u> ile <u>E</u> o	dit <u>B</u> uild <u>D</u> ebug <u>A</u> nalyze	<u>T</u> ools <u>W</u> indow <u>H</u> elp		
	Q	Projects	Qt 5.5.0 GCC 64bit ‡	Search in Examples	
	welcome				
	Edit	Examples	Enginio Todos	The second	(III
	×.	Tutorials	✓ Do awesome coding	• • • •	\cup
	Design		✓ Make alpha release		
		New to Qt?	Blog about it		
	Debug	Learn how to develop your own applications and explore	New todo		
	Projects	Qt Creator.	Enginio QML Examples	Planets Example	
		Get Started Now	Tags: qml enginio todos	Tags: canvas3d planets	
	Analyze	L Qt Account			
• 0t.	?	Qt Cloud Services		C Calender Exemple	
	Help	Doline Community		1 Wednesday" January 2014 > processry 2014 See Non Tae Wed Thue 81 See	
		Blogs		400701 1 5 6 7 8 9 20 11 2 12 13 14 15 36 17 18	
		🕜 User Guide		5 15 30 17 12 12 30 34 15 4 38 27 38 29 30 11 1	
		Ŭ		<u><u></u><u></u><u></u><u></u><u></u><u></u><u></u><u></u><u></u><u></u><u></u><u></u><u></u><u></u><u></u><u></u><u></u><u></u><u></u></u>	
			QML Video Shader Effect	Qt Quick Controls - Calend	
			Tags: video android qml effects multimedia shader	Tags: quick calendar controls	
5	\nearrow				5
要協論)守	向到这虚拟机	P* Type to locate (Ctr 诸格留标指针移入其中或按 Ctrl+G	I Issues 2 Searc 3 Ap	opu 4 Comp 5 QML/	
SCITIAN (VE)	1.11. A CERT AND A CERT AND A	HEIGHNINGTERING CHEIGHNING CHEIG			

默认 QT 的配置是针对 X86 架构的,这时编译出来的示例只能在 PC 机上运行。我们打 开 Image Composition Example 示例,它是一个图片叠加显示的示例,找到该示例,单击即 可。打开后的界面如下:



www.9tripod.com

优秀嵌入式技术服务商

	<u>F</u> ile <u>E</u> dit	<u>B</u> uild	Debug Analyze Tools Window Help		
		cube i	magecomposition	_	_
	Welcome	Build &	Run Editor Code Style Dependencies		
	Edit	Add Kil Manage	Vesktop Qt 5.5.0 CCC 64bit KitS Build Run		
	26	> 1	Build Settings		
· >_	Design	۲ 🕈	Edit build configuration: Debug 🛟 🛛 Add 💌 Remove Rename		
	Debug	(General		
	Projects		Shadow build: 🥑		
	Analyze		Build directory: widgets/painting/build-imagecomposition-Desktop_Qt_5_5_0_GCC_64bit-Debug	Browse	
· Qt	(?) Help	I	Build Steps		
=			qmake: qmake imagecomposition.pro -r -spec linux-g++ CONFIG+=debug	Details 👻	
			Make: make in /home/lqm/Qt5.5.0/Examples/Qt-5.5/widgets/painting/build-imagecomp	Details 👻	
		ĺ	Add Build Step 💌		
		(Clean Steps		
			Make: make clean in /home/lqm/Qt5.5.0/Examples/Qt-5.5/widgets/painting/build-imagecompo	Details 👻	
1		(Add Clean Step 💌		
	image ition	I	Build Environment		
			Use System Environment	Details 🔻	
\sim	Debug				
	>				

点击左下脚绿色的三脚箭头,开始编译工程。在 Compile Output 栏会显示编译的整个过程。编译完成后,提示如下:

ڬ — 🗖	25°C	計 14 40) 4:49 AM 役
File Edit Build Debug Analyze	Tools Window Help	
Projects © 72 @ E+	🗠 😢 🔶 📾 🖻 imagecomposition/main.cpp 🕫 🛪 «Select Symbol» 🗧	‡ # Line: 1, Col: 1 🛛 🕀
Course Course	9 September 2015 All The Gr Company Ltd. Conter: http://www.cti.au/licensing/ Conter: http://www.cti.au/licensing/ This fit is part of the wangles of the C Taulkit. * 07.06021 (CONCER-005) * Works with this fils under the terms of the ED License as follows: * Medistributions and use in surce and bisary forms, with ar without * Medistributions of surce code must rains the above copyright * ext: * Addition of surce code must rains the above copyright * Addition is history form must raproduce the Above Copyright * Addition is history form must raproduce the Above Copyright * Addition is history form must raproduce the Above Copyright * Statistic that and the Company Ltd. The Conce Copyright * Statistic the sum of the C company Ltd or the many at its * Statistic the sum of the C company Ltd or the many at its * catribution may be used to ombrase ar provide stice of the Copyright * catribution may be used to ombrase ar provide stice of the Statistic to the Copyright stice of	
Cost Descuences Cost Descuences Cost Descuences Cost Descuences Cost Descuences Cost Descuences Cost Descuences Cost Descuences Cost Descuences Cost Descuences Cost Descuences Cost Descuences Cost Descuences Cost Descuences Cost Descuences Cost Descuences Cost Descuences Cost Descuences Cost Descuences Cost Descuences Cost Descuences Cost Descuences Cost Descuences Cost Descuences Cost Descuences Cost Descuences Cost Descuences Cost Descuences Cost Descuences Cost Descuences Cost Descuences Cost Descuences Cost Descuences Cost Descuences Cost Descuences Cost Descuences Cost Descuences Cost Descuences Cost Descuences Cost Descuences Cost Descuences Cost Descuences Cost Descuences Cost Descuences Cost Descuences Cost Descuences Cost Descuences Cost Descuences Cost Descuences Cost Descuences Cost Descuences Cost Descuences Cost Descuences Cost Descuences Cost Descuences Cost Descuences Cost Descuences Cost Descuences Cost Descuences Cost Descuences Cost Descuences Cost Descuences Cost Descuences Cost Descuences Cost Descuences Cost Descuences Cost Descuences Cost Descuences Cost Descuences Cost Descuences Cost Descuences Cost Descuences Cost Descuences Cost Descuences Cost Descuences Cost Descuences Cost Descuences Cost Descuences Cost Descuences Cost Descuences Cost Descuences Cost Descuences Cost Descuences Cost Descuences Cost Descuences Cost Descuences Cost Descuences Cost Descuences Cost Descuences Cost Descuences Cost Descuences Cost Descuences Cost Descuences Cost Descuences Cost Descuences Cost Descuences Cost Descuences Cost Descuences Cost Descuences Cost Descuences Cost Descuences Cost Descuences	Construction Construction 0000000 000000000000000000000000000000000000	Clude/OfMidgets -I/ Magoser.cpg Ilde/OfMidgets -I/ .cpg Percensition -I/home/ Mailtan/ //Off.S.S./S.S/gcc_64/

编译完成后,编译出来的映像会自动运行,一个叠加的蝴蝶图像界面显示出来了,界面 如下:



www.9tripod.com

优秀嵌入式技术服务商



下面我们将该示例编译到开发板上运行。使用 QT Creator 打开上面的示例工程,如下 图所示:



选择 Projects 一栏,可以看到最顶端的框图 1,显示 Desktop Qt 5.5.0 GCC 64bit,表明

优秀嵌入式技术服务商



它通过 64 位的 GCC 编译, 框图 2 表明在 debug 模式下编译, 相对 release 模式, debug 模式 下含有大量调试信息, 编译出来的映像会比较大。通常发布映像时, 我们选择 release 模式。 框图 3 指定了编译的路径。框图 4 指定了 qmake 和交叉编译工具。框图 5 为调试按键, 第 一个用于选择编译模式为 debug 或 release, 第二个为运行按钮, 第三个为单步调试按钮, 第 四个为编译按钮。框图 6 为一些输出信息, 如 Compile Output, 会给出整个编译的信息。 在框图 1 中, 点击 Tools->Options, 如下图:

<u>File Edit Build Debug Analyze Tools Window Help</u>	_
S Options	_
Filter Build & Run	
Environment General Kits Qt Versions Compilers Debuggers CMake	
Text Editor Ame qmake Location	Add
FakeVim ▼ Auto-detected Qt 5.4.1 GCC 64bit /home/jason/Qt5.4.1/5.4/gcc_64/bin/qmake	Remove
	Clean Up
Qt Quick	
🕓 Build & Run 👘	
Sebugger	
Analyzer	
Version Control	
Android	
BlackBerry	
awx QNX	
Devices	
Apply Cancel	<u><u>o</u>k</u>
P- Type to locate (Ctr 1 Issues 2 Search Results 3 Application 4 Compile Ou 5 QML/JS Con	¢ ^

在左测对话框中选择 Build & Run,在后边选择 Qt Versions,默认 Manual 为空, Auto-detected 选择默认的 qmake,它将运行在 X86 平台的 linux 系统上,因此我们要手动添 加在 ARM 平台上运行的 qmake。

在 buildroot 编译文件系统时,我们选中 QT5 后,将会在 buildroot 的 output/host/usr/bin 目录生成支持 ARM 平台的 qmake,点击 Add,指向该路径的 qmake:



点击 Open, 即在 Manual 中添加了支持 ARM 平台的 qmake。再选择 Compilers 一栏, 如下图所示:

优秀嵌入式技术服务商

论坛: www.xboot.org



<u>File Edit Build Debug Analyze Tools Window H</u> elp	
😣 🗈 Options	
Filter Build & Run	
Environment General Kits Qt Versions Compilers Debuggers CMake	
Text Editor Name Type	Add 🔻
FakeVim	Clone
Image: Second Content (Salphin) Geographic (Salphin) Geographin) Geographic (Salphin) Geographic (Salphin) Geographic	Remove
() C++	
Qt Quick	
🕓 Build & Run	
💭 Debugger	
Analyzer	
Version Control	
Android	
BlackBerry	
Parker QNX	
Devices	
	OK
Apply <u>C</u> ancer	
P- Type to locate (Ctr] 1 Issues 2 Search Results 3 Application 4 Compile Ou 5 QML/JS Con	• •

默认 Manual 为空, Auto-detected 为支持 X86 32 位和 64 位的 GCC, 这将直接导致编译 出来只能在 PC 机上运行。点击 Add, 选择 GCC, 如下图所示:

<u>F</u> ile	Ed	lit <u>B</u> uild <u>D</u> ebug <u>A</u> r	nalyze <u>T</u> ools <u>W</u> indow <u>H</u> elp	
		imagecomposition		
Qt		Build & Run Edi	itor Code Style Dependencies	
Welcom	ie	X 🗆 Options		
Edit		Filter	Build & Run	
Design		Environment	General Kits Qt Versions Compilers Debuggers CMake	
		Text Editor	Name Type	Add 🔻
Debug		🚮 FakeVim	▼ Auto-detected	Clone
		Help	GCC (x86 64bit in /usr/bin) GCC GCC (x86 32bit in /usr/bin) GCC	Remove
Project	s	{} c++	▼ Manual GCC-ARM-LINUX GCC	
Analyz	e	Qt Quick		
2		🚯 Build & Run		
Help		🔎 Debugger		
		Analyzer	Name: GCC-APM-LINUX	
		Version Control	Compiler path:	
		💩 Android		
			Platform codegen flags:	
			Platform linker flags:	
			ABI: arm-lin: () arm () - linux () - generic () - elf () - 32bit ()	
		Devices		
		· · · · · · · · · · · · · · · · · · ·	Apply Cancel	<u>O</u> K

在 Name 中重命名,以区别 ARM 和 PC 平台,如我们命名为 GCC-ARM-LINUX,它将 会直接显示在前面 QT 工程界面的框图 1 中,到时我们编译 QT 工程时,能够一目了然,编 译出来的到底是 PC 平台还是 ARM 平台。在 Compiler path 中指定交叉编译工具,在 buildroot 中默认已经自动下载并安装了交叉编译工具,我们指定到如下路径即可:

buildroot/output/host/opt/ext-toolchain/bin/arm-linux-gnueabihf-g++

设置完成后,点击OK,完成设置。

```
优秀嵌入式技术服务商
```



再回到 QT 的工程界面,我们发现框图 1 中的配置仍然针对 PC 机,点击框图 1 中的 Manage Kits,如下图所示:

Environment	General	Kits	Ot Versions	Compilers	Debuggers	CMake		
Text Editor	Name		•		55			Add
FakeVim	▼ Auto-	detec	ted					Close
Help	De Manu	e sktop Jal	Qt 5.4.1 GCC	64bit (defa	ılt)			Demous
°} c++								Remove
Ot Quick								Make Defaul
Build & Run								
Debugger								
Analvzer								
Version Control								
Android								
BlackBerry								
Devices								
							Apply <u>C</u> a	ncel OK

注意,这时默认 Manual 仍然为空,点击 Add,会弹出一个有很多选项的对话框,我们 按如下方式配置:

Compile:选择之前添加的 GCC, Qt Version:选择之前添加的 Qt5.5.1;

Filter		Build & Run		
Environment	ĥ	General Kits Q	t Versions Compilers Debuggers CMake	
Text Editor		Name		Add
FakeVim		▼ Auto-detected	j t 5 4 1 CCC 54bit (default)	Clone
Help		▼ Manual		Remove
() c++		Qt 5.5.1 (S	ystem)	Make Default
📣 Qt Quick				
🕕 Build & Run	8	Device type:	Desktop	
🐊 Debugger		Device:	Local PC (default for Desktop)	Manage
Analyzer		Sysroot:		Browse
Version Control		Compiler:	GCC	Manage
👨 Android		Environment:	No Changes to apply	Change
BlackBerry		Debugger:	System GDB at /usr/bin/gdb	Manage
anx QNX	U	Qt version:	Qt 5.5.1 (System)	Manage
Devices		Qt mkspec:]
2	Θl			

这里的 Name,我们可以点击 Qt Versions,选择 Manual 中的 qmake,下面就会有名称出来,将它拷贝过来即可,如下图所示:

优秀嵌入式技术服务商	论坛: www.xboot.org	电话:	0755-33133436



10										
Environment	General	Kits	Qt Versions	Compilers	Debuggers	CMake				
Text Editor	Name		qr	nake Locatio	n					Add
FakeVim	▼ Auto- Qt ▼ Manu	-detect t 5.5.0 C ual	ed GCC 64bit /h	ome/lqm/Qt	5.5.0/5.5/gcc	_64/bin/q	make			Remove
} C++	Qt	t 5.4.1 (System) /h	ome/lqm/sai	mba/buildroo	t/output/	'host/usr/b	in/qmake		Clean U
Qt Quick										
Qt Quick										
Qt Quick Build & Run Debugger	È									
Qt Quick Build & Run Debugger Analyzer	Ξ									
Qt Quick Build & Run Debugger Analyzer Version Control	Version	ı name:	Qt %{Qt:	Version} (Sys	tem)					
Qt Quick Build & Run Debugger Analyzer Version Control Android	Version qmake l	ı name: locatior	Qt %{Qt:\ n: /home/lqr	Version} (Sys n/samba/bui	tem) ildroot/outpu	t/host/us	r/bin/qmał	ce	Browse	
Qt Quick Build & Run Debugger Analyzer Version Control Android BlackBerry	Version qmake l	name: locatior	Qt %{Qt: h: /home/lqr	Version} (Sys n/samba/bui	tem) ildroot/outpu	t/host/us	r/bin/qmał	.ce	Browse	
Qt Quick Build & Run Debugger Analyzer Version Control Android BlackBerry × QNX	Version qmake l Qt versi	i name: locatior ion 5.4.	Qt %{Qt:* n: /home/lqr 1 for Embed	Version} (Sys n/samba/bui ded Linux	tem) ildroot/outpu	t/host/us	r/bin/qmał	.ce	Browse Details •	

在框图一中,点击设置编译环境的下拉箭头,如下图所示:

File	Edit	<u>B</u> uild	Debug Analyze Tools Window Help	
	ir	nageco	omposition	
QL		Build &	Run Editor Code Style Dependencies	
Welcom	e	Add Ki	it V OF 5.4.1 (System)	
Edit		Manage	e Kits Build Run Change Kit Desktop Qt 5.4.1 GCC 64bit	
26		>	Build Settings Remove Kit Qt 5.5.1 (System)	ĥ
Design			Edit build configuration: Debug 🛟 Add 🔹 Remove Rename	
Debug			General	
Project	s		Shadow build: 🥑	
Analyz			Build directory: es/Qt-5.4/widgets/painting/build-imagecomposition-Qt_5_5_1_System_2-Debug Browse	
(?) Help			A build for a different project exists in /home/jason/Qt5.4.1/Examples/Qt-5.4/widgets/ painting/build-imagecomposition-Qt_5_5_1_System_2-Debug, which will be overwritten.	
imageit	ion		Build Steps	
Debug			qmake: qmake image composition.pro -r -spec devices/linux-buildroot-g++ CONFIG+=det Detail	S 🔻
			Make: make in /home/jason/Qt5.4.1/Examples/Qt-5.4/widgets/painting/build-imagecom Detail	S ▼
			Add Build Step v	
			Clean Steps	
		₽• T)	ype to locate (Ctr 1 Issues 2 Search Results 3 Application 4 Compile Ou 5 QML/JS Con 🖨	

选择 Change Kit->Qt5.5.1(System),更改后的界面如下:



到此,配置完毕,点击框图 5 中的编译按钮,在 Compile Output 中可以看到编译信息如下:



这时,在框图2中指定的目录中已经生成了能够在ARM平台运行的映像了,如下图所

jason@ubuntu:~/Qt5.4.1/Examples/Qt-5.4/widgets/painting/build-imagecomposition-Q t_5_5_1_System_2-Debug\$ file imagecomposition imagecomposition: ELF 32-bit LSB executable, ARM, EABI5 version 1 (SYSV), dynam ically linked (uses shared libs), for GNU/Linux 3.4.5, BuildID[sha1]=a7014b11ea0 9398999d069c052262f03bef95ef2, not stripped jason@ubuntu:~/Qt5.4.1/Examples/Qt-5.4/widgets/painting/build-imagecomposition-Q t_5_5_1_System_2-Debug\$

将该文件拷贝到 x3399 开发板上运行,可以看到美丽的蝴蝶图案显示出来了。



示:



第4章 qttest 测试程序

4.1 使用 QT_demo 测试 LED 灯



点击上面的四盏灯可以看到开发板 led 的灯的亮和灭。

4.2 使用 QT_demo 测试蜂鸣器

进入 QT5.6.1 系统后,默认会运行我们自主编写的测试 demo,测试界面如下:



按住 Beep 键时,蜂鸣器鸣叫,松开时,蜂鸣器停止鸣叫。

论坛: www.xboot.org



www.9tripod.com 优秀嵌入式技术服务商

4.3 使用 QT_demo 调节背光

测试界面如下:



滑动圆形滑轮,可对开发板背光进行亮暗调节。

4.4 使用 QT_demo 测试按键

测试界面如下:

	KeyBoard Test	
brightness 255	Press any key!	Ac online: Battery Vol: Battery Capacity:
		_{pattery} vol: . Herry Capacity:

按下开发板任一独立按键,图中界面即会显示相应键值,同时,按下时提示[keydown], 抬起时提示[keyup]。

4.5 使用 QT_demo 测试音频

将喇叭或耳机接到开发板的对应接口,点击下图中的 Play Sound 按钮,会播放测试歌

优秀嵌入式技术服务商

论坛: www.xboot.org



www.9tripod.com 优秀嵌入式技术服务商

曲:

			Audio Test		
	AC Online:				
Press	Battery Vol:		יlay Sound		
	Battery Capacity:	1			
	Battery capacity:				
	Battery Vol:				

4.6 使用 QT_demo 测试触摸屏

进入如下界面:

	·	TouchScreen Test	
[keydown	Play Sound		
			Network Test
Pre-trave	AN APPA		Retwork Test

单击黄色矩形框,界面会进入全屏模式,这时我们可以任意书写来测试触摸屏了,测试示例图片如下:



4.7 使用 QT_demo 测试网络

将网线连接开发板的有线以太网接口,点击界面中的 Network Test 按钮,如果网络已经 连通,则会添加 DNS,如果没有连通,则会提示相应错误,如下图所示:



连接网线后测试时,会有如下提示:



4.8 使用 QT_demo 测试 TF 卡

	Tfcard Test		
Network Test	Tfcard Test	Udisk Test	
Hetroit Ies	Tfcard Test	udik Test	

将 SD 卡插入开发板的 SD 卡槽,点击 Tfcard Test,界面上会列出 SD 卡中的内容。



4.9 使用 QT_demo 测试 U 盘

将 U 盘接到开发板的任何一个 USB HOST 接口,点击 Udisk Test,数据框中会列出 U 盘中的数据,如果找不到,则会提示无法 mount U 盘,如图:



4.10 使用 QT_demo 测试重启

点击 Reboot 按钮,开发板将重启。





4.11 使用 QT_demo 测试关机

点击 Poweroff 按钮,开发板将会关机。

	Poweroff Test	
Reboot	Poweroff	
Hadren		



www.9tripod.com 优秀嵌入式技术服务商

第5章 安装 ubuntu14.04 系统

目前我们仅移植了 android6.0 以及 dabian 系统,很快 ubuntu 系统就会与大家见面,敬 请期待。



第6章 linux 底层开发示例

6.1 播放 mp3

注: linux 系统缺省登录账户为 root, 密码: 123456

将存放有 mp3 文件的 TF 卡插到开发板的任意卡槽,使用如下命令挂载 TF 卡:

cd / mkdir sdcard mount /dev/mmcblk1p1 /sdcard cd sdcard

使用如下命令播放:

./mplayer *.avi

./mplayer *.mp3

连接串口后,可以通过 PC 键盘的 0 或 9 调节音量。也可以使用 madplayer 播放音乐。

6.2 在后台运行程序

在上一节中给出了播放音乐的示例,但是这时候 mplayer 已经占据了终端控制台,在音 乐播放完之前,我们无法再使用终端控制台了。又比如我们开发一款产品时,就需要在启动 文件系统后运行一个应用程序,如果运行了一个程序,终端控制台就被占用了,那将极大的 限制我们的功能。为止,我们可以将程序放在后台运行。使用方法很简单,我们只需在执行 的指令后面添加一个"&"即可。如播放音乐时使用如下命令:

./mplayer *.mp3 &

6.3 中止程序的运行

中止程序的运行有多种方式,最直接的方式就是直接按 ctrl+c。如前面我们正在播放一段音频文件,我们可以按 ctrl+c 退出程序。但是如果程序在后台运行,那么我们按 ctrl+c 就不管用了。这时我们可以使用 kill 命令。

kill+**PID**

kill+文件名

6.4 屏幕抓图

本文档中的各个图片,都是采用 gsnap 这个工具进行抓图的。进入 QT 图形界面后,我 们能在 LCD 上看到丰富多彩的人机交互界面。通过 gsnap 可以抓取到图形界面精彩的瞬间。 在控制台终端输入如下命令:

gsnap test_pic.jpg /dev/fb0

这时在当前目录将会保存 test_pic.jpg 图像文件。详细的 gsnap 移植步骤在后面会有详细描述。

6.5 挂载 TF 卡

进入 QT 图形界面后,在命令终端会有控制台出现,这时可以通过控制台查看文件系统的内容。将 TF 卡插到开发板的任意卡槽,串口终端会有如下提示:

IN TE LL	× _	-	1) 00	47 -	
(冗秀歌)	正 入	、 投 フ	不用反	务商	

论坛: www.xboot.org



这时在文件系统的/dev 目录将会自动生成一个名叫 mmcblk1p1 的块设备文件。它就是 对应的 TF 卡的设备文件,使用如下命令挂载 TF 卡到/sdcard 目录:

mkdir /sdcard

mount /dev/mmcblk1p1 /sdcard

查看/scard 目录下的内容,即是我们 TF 卡中的内容,如下图所示:

[root8x3399:-]# root8x3399:-]# d /sdcard/ root8x399:-]# d /sdcard/ root8x399:-j# cd /sdcard/ root8x399:/sdcard # 1s LOST. DIR libjpeg.so. root8x3399:/sdcard #	math 62 mkfile 62.0.0 process	threa	d		9	1
<pre>mmcbik0ort0 mmcbik0pl1 mmcbik0ort0 mmcbik0pl3 mmcbik0ort1 mmcbik0pl3 mmcbik0ort1 mmcbik0pl3 mmcbik0ort1 mmcbik0pl1 mmcbik0ort0 mmcbik0pl1 mmcbik0pl1 mmcbik0pl3 mmcbik0pl3 mmcbik0pl3 mmcbik0pl3 mmcbik0pl3 mmcbik0pl3 mmcbik0pl3 mmcbik0pl3 mmcbik0pl3 mmcbik0pl3 mmcbik0pl3 mmcbik0pl3 mmcbik0pl3 mmcbik0pl3 mmcbik0pl3 mmcbik0pl3 mmcbik0pl3 mmcbik0pl3 mmcbik0pl3 mmcbik0pl3 mmcbik0pl3 mmcbik0pl3 mmcbik0pl3 mmcbik0pl3 mmcbik0pl3 mmcbik0pl3 mmcbik0pl3 mmcbik0pl3 mmcbik0pl3 mmcbik0pl3 mmcbik0pl3 mmcbik0pl3 mmcbik0pl3 mmcbik0pl3 mmcbik0pl3 mmcbik0pl3 mmcbik0pl3 mmcbik0pl3 mmcbik0pl3 mmcbik0pl3 mmcbik0pl3 mmcbik0pl3 mmcbik0pl3 mmcbik0pl3 mmcbik0pl3 mmcbik0pl3 mmcbik0pl3 mmcbik0pl3 mmcbik0pl3 mmcbik0pl3 mmcbik0pl3 mmcbik0pl3 mmcbik0pl3 mmcbik0pl3 mmcbik0pl3 mmcbik0pl3 mmcbik0pl3 mmcbik0pl3 mmcbik0pl3 mmcbik0pl3 mmcbik0pl3 mmcbik0pl3 mmcbik0pl3 mmcbik0pl3 mmcbik0pl3 mmcbik0pl3 mmcbik0pl3 mmcbik0pl3 mmcbik0pl3 mmcbik0pl3 mmcbik0pl3 mmcbik0pl3 mmcbik0pl3 mmcbik0pl3 mmcbik0pl3 mmcbik0pl3 mmcbik0pl3 mmcbik0pl3 mmcbik0pl3 mmcbik0pl3 mmcbik0pl3 mmcbik0pl3 mmcbik0pl3 mmcbik0pl3 mmcbik0pl3 mmcbik0pl3 mmcbik0pl3 mmcbik0pl3 mmcbik0pl3 mmcbik0pl3 mmcbik0pl3 mmcbik0pl3 mmcbik0pl3 mmcbik0pl3 mmcbik0pl3 mmcbik0pl3 mmcbik0pl3 mmcbik0pl3 mmcbik0pl3 mmcbik0pl3 mmcbik0pl3 mmcbik0pl3 mmcbik0pl3 mmcbik0pl3 mmcbik0pl3 mmcbik0pl3 mmcbik0pl3 mmcbik0pl3 mmcbik0pl3 mmcbik0pl3 mmcbik0pl3 mmcbik0pl3 mmcbik0pl3 mmcbik0pl3 mmcbik0pl3 mmcbik0pl3 mmcbik0pl3 mmcbik0pl3 mmcbik0pl3 mmcbik0pl3 mmcbik0pl3 mmcbik0pl3 mmcbik0pl3 mmcbik0pl3 mmcbik0pl3 mmcbik0pl3 mmcbik0pl3 mmcbik0pl3 mmcbik0pl3 mmcbik0pl3 mmcbik0pl3 mmcbik0pl3 mmcbik0pl3 mmcbik0pl3 mmcbik0pl3 mmcbik0pl3 mmcbik0pl3 mmcbik0pl3 mmcbik0pl3 mmcbik0pl3 mmcbik0pl3 mmcbik0pl3 mmcbik0pl3 mmcbik0pl3 mmcbik0pl3 mmcbik0pl3 mmcbik0pl3 mmcbik0pl3 mmcbik0pl3 mmcbik0pl3 mmcbik0pl3 mmcbik0pl3 mmcbik0pl3 mmcbik0pl3 mmcbik0pl3 mmcbik0pl3 mmcbik0pl3 mmcbik0pl3 mmcbik0pl3 mmcbik0pl3 mmcbik0pl3 mmcbik0pl3 mmcbik0pl3 mmcbik0pl3 mmcbik0pl3 mmcbik0pl3 mmcbik0pl3 mmcbik0pl3 mmcbik0pl3 mmcbik0pl3 mmcbik0pl3 mmcbik0pl3</pre>	<pre>mmcDikOp3 mmcDikOp5 mmcDikOp5 mmcDikOp2 mmcDikOp3 mmcDikOp5 mmcDikOp5 mmcDikOp5 blk1p1 /sdcard/ blk1p1 /sdcard/ p1): volume was</pre>	mmcD1k0p8 mmcD1k0p9 mmcD1k0p9 mmcD1k0p7 mmcD1k0p8 mmcD1k0p8 mmcD1k0ppmb mmcD1k0ppmb mmcD1k1		5	英	-
[root@x3399:~]# [root@x3399:~]# mkdir /sdcard [root@x3399:~]# mount /dev/mm mma8452 daemon mmcblkOp10						

6.6 挂载 U 盘

进入 QT 图形界面后,在命令终端会有控制台出现,这时可以通过控制台查看文件系统的内容。插入 U 盘后,串口终端会有如下提示:

[1000ex333551/	
[root@x3399:/]	
[root#x3399:/	
1000TOX22001/	
11000ex33355.7	
[rootex3399:/	
[root@x3399:/	# [713.766576] usb 1-1.2: new high-speed USB device number 5 using ehci-platform
713,868425	ush 1-1.2: New USB device found. Idvendor=Obda. ideroduct=0150
713,868538	usb 1-1.2: New US8 device strings: Mfr=1. Product=2. SerialNumber=3
713.868566	ush 1-1.2: Product: USR2.0-CBW
713,868590	usb 1-1.2: Manufacture: Generic
713 868612	ush 1-1 2: SecialNumber: 20120926571200000
713 874051	ush-storane 1-1 2:1 0: USB Wass Storane device detected
713.875889	sesi host?: usb-storane 1-1.2:1.0
714 878722	sesi 2:0:0:0: Direct-Arcess Generic- Nulti-Card 1 00 DD: 0 AUST: 4
714 881440	sd 2:0:0:0: Attached scsi generic sd0 type 0
715.572461	sd 2:0:0:0: [sda] 15433728 512-byte logical blocks: (7:90 GB/7:36 GB)
	sd 2:0:0:0: Isdal write protect is off
715 574380	ad 2101010: Edda write cache: disabled read cache: enabled doesn't support DBO or FUA
715 580850	ada terestat
1 713.300039	sual sual
[/15.5890/4]	so 2:0:0:0: [Soa] Attached ScS1 Penovable disk
prootex3399:/	
[root@x3399:/]	
[

这时在文件系统的/dev 目录将会自动生成一个名叫 sda4 的块设备文件。它就是对应的 U 盘设备文件,使用如下命令挂载 U 盘到/udisk 目录:

mkdir /udisk

mount /dev/sda4 /udisk

查看/udisk 目录下的内容,即是我们 U 盘中的内容,如下图所示:

6.7 保存系统时钟

Linux 可以使用 date 指令更改时间日期。例如:

date -s 201607211433 #设置为 2016 年 7 月 21 日 14:33 分

hwclock -w #把刚设置的时间存入 RTC 寄存器

hwclock -s #恢复 linux 系统时钟为 RTC 寄存器值, 一般将该指令放在 rcS 中开机自动执行。

论坛: www.xboot.org



6.8 掉电保存数据到 flash

由于本系统采用了 ext4 文件系统,因此可以很方便的保存数据,确保掉电后数据不丢 失。如我们从 U 盘中拷备一首歌曲到/root 目录:

cp /udisk/muyangqu.mp3 /

重启开发板,我们发现在 root 目录仍然存在刚才拷备的这首歌曲,说明掉电后数据并 没有丢失。

6.9 设置开机自动运行程序

借助启动脚本可以设置各种程序开机后自动运行,这点很类似于 WINDOWS 的 Autobat 自动批处理文件。启动脚本位于/etc/init.d/rcS 中,我们可以将自己想要开机运行的程序或是 开机执行的指令放在 rcS 里面。比如我们想制作一个简单的开机音乐,我们就完全可以在 rcS 中添加如下语句:

./mplayer start.mp3 &

这时,开机后就会播放名叫 start.mp3 的音乐了。注意 start.mp3 需要在当前执行指令所 在目录。

6.10 查看开发板内存信息

X3399 开发板默认配置 2GB LPDDR3 SDRAM,在 uboot 启动时,打印信息上会给出 RAM 大小信息:



在进入文件系统后,可以通过 cat 命令查询 Linux 系统分配到的 SDRAM 大小。执行如下命令:

cat /proc/meminfo

优秀嵌入式技术服务商

论坛: www.xboot.org





第7章 linux 应用开发示例

本手册给出的所有应用程序全部在九鼎创展 x3399 开发板上运行,这里仅给出了一些 比较基础,常用的应用程序,旨在为用户打开 Linux 世界奇妙的大门,用户定能举一反三, 编写出属于自己的更加丰富完美的程序。

声明:以下所有应用程序全部为九鼎创展科技有限公司原创作品,所有内容全经我们严格测试,建议用户按照下面步骤动手编译一遍,以增强自己的理解,不推荐直接使用我们提供好的文件。另外,敬请商业人士勿侵犯版权。

7.1 Hello World

第一步: 生成可执行文件

在 x3399_marshmallow 目录新建 app-ex 目录,在 app-ex 目录新建 hello 目录,然后在 hello 目录下新建 hello.c 和 makefile 两个文件:

vim hello.c

这是一个最基础的应用程序,如果我们声明了交叉编译工具(export PATH=/home/lixu/projects/3399/x3399_marshmallow/buildroot/output/host/opt/ext-toolchain/bin: \$PATH)(这里我们用到 buildroot 中的交叉编译工具),可以直接敲入命令进行编译(由于开发板环境缺少 c 库,编译时需要加上-static 静态编译):

```
arm-linux-gcc-4.4.3 -o hello hello.c -static
```

说明:由于开发板文件系统缺少标准函数库,编译时加上-static 选项;使用的编译器为 aarch64-linux-gnu-gcc。

编译完成后,在当前目录会生成 hello 可执行文件,我们可以使用 file 命令查询执行文件是否为 ARM 体系文件:

oot®x3399:/mmt]# file hello llo: ELF 64-bit L58 executable, ARM aarch64, version 1 (SYSV), statically linked, for GNU/Linux 3.7.0, BuildID[sha1]=2bf4107645d9d63369d9afb8228311c651b224ba, not strippe

第二步:将可执行文件下载到开发板运行 比较常用的方式有以下四种:

- 1) 通过串口和 sz/rz 工具
- 2) 复制到存储媒介,如 SD 卡,U 盘等
- 3)通过 NFS 挂载文件系统,这时不用将可执行文件拷备到开发板了,推荐调试使用这 种方式!
- 4) 通过 ftp 传输

这里介绍第二种方法,以TF卡为例,其他方法请读者自行尝试。

将生成的 hello 文件拷备到 TF 卡, 再将 TF 卡插入开发板的 TF 卡接口, 将 TF 卡 mount 到/mnt 目录:

cd /

优秀嵌入式技术服务商

论坛: www.xboot.org



mount /dev/mmcblk1p1 /mnt

进入 mnt 目录,可以看到刚才拷贝的 hello 文件了:

root@x3399:/mnt]# .jpg gsnap **hello** root@x3399:/mnt]# root@x3399:/mnt]# root@x3399:/mnt]#

运行 hello:

./hello

打印信息如下:

[rootex3399:/mnt]# chmod 777 hello [rootex3399:/mnt]# [rootex3399:/mnt]# [rootex3399:/mnt]# [rootex3399:/mnt]# [rootex3399:/mnt]# ./hello hello,9tripod[rootex3399:/mnt]#

表明,程序已经成功运行。

前面编译文件我们需要通过手敲命令执行,我们可以通过编写 makefile 来代替手敲的动

作。

在 hello 目录下新建 makefile 文件: vim makefile

hello:

aarch64-linux-gnu-gcc -o hello hello.c -static

clean:

rm-f hello *.o

直接在 hello 目录下敲 make 就可以生成 hello 文件。运行的效果和前面的完全相同。

7.2 数学函数库调用

建立程序编译路径:

mkdir math

cd math

vim math.c

编辑如下内容:

#include <stdio.h>

#include <stdlib.h>

#include <math.h>

int main(void)

{

double a=9.0; printf("sqrt(%f)=%f\n",a,sqrt(a));

return 0;

}

编辑 makefile 文件,内容如下:

```
OBJS=math.o
```

LDFLAGS = -lm -static

CC = yourtool chain path/bin/aarch64-linux-gnu-gcc

```
优秀嵌入式技术服务商
```

论坛: www.xboot.org



www.9tripod.com 优秀嵌入式技术服务商

math:\${OBJS}

\${CC} -o \$@ \$^ \$(LDFLAGS)

clean:

{

rm -f math *.o

执行 make,将生成的可执行文件 math 下载到开发板上运行,如下图所示:

7.3 多线程编程示例

建立程序编译路径:

```
mkdir thread
cd thread
vim thread.c
     编辑如下内容:
#include<stddef.h>
#include<stdio.h>
#include<unistd.h>
#include"pthread.h"
void function1(void);
void function2(void);
int func_flag=0;
pthread_mutex_t mutex;
main()
     pthread_t reader;
     pthread_mutex_init(&mutex,NULL);
     pthread_create(&reader,NULL,(void*)&function1,NULL);
     function2();
void function2(void)
     while(1)
     {
          pthread_mutex_lock(&mutex);
          if(func_flag==0)
           {
                printf("excute function2.\n");
                func_flag=1;
```

优秀嵌入式技术服务商

论坛: www.xboot.org



编辑 makefile 文件, 内容如下:

OBJS=thread.o

LDFLAGS = -lpthread -static

CC = yourtoolchainpath/bin/aarch64-linux-gnu-gcc

thread:\${OBJS}

 ${CC} -0$

clean:

rm -f thread *.o

执行 make,将生成的可执行文件 thread 下载到开发板上运行,如下图所示:

The set of	
[root@x3399:/mnt]# ./thread	
excute function2.	
excute function1	
excute function:	
excute function2.	
excute function1.	
excute function2.	
evente function1	
excuce function.	
excute function2.	
excute function1.	
excute function?	
excute function2.	

7.4 多进程编程示例

在 Linux 下通用调用 fork 函数创建新的进程。调用 fork 时,系统将产生一个与当前进程相同的进程。它与原有的进程具有相同的数据,连接关系和在程序同一处执行时的连续性。通常将原有的进程叫父进程,新创建的进程叫子进程。

fork 调用将分两次返回,从父子进程返回。进程创建语法如下:

#include <unistd.h>

pid_t pid;

pid = fork();

```
优秀嵌入式技术服务商
```

```
论坛: www.xboot.org
```



如果 pid 返回 0,表示说明从子进程返回,否则从父进程返回,此时返回的是进程的 ID 号。我们可以通过 getpid()函数来获得进程的 ID 号。 首先建立程序编译目录: mkdir process cd process vim process.c 编辑如下内容: #include<stdio.h> #include<unistd.h> #include<sys/types.h> main() { pid_t pid; pid=fork(); if (pid<0) { printf("fork is error!\n"); return 1; } else if (pid == 0){ while (1) { printf("the child process is running now.pid=%d\n",getpid()); sleep(1);//linux 延时函数,延时1秒 } } else { while (1) { printf("the perent process is running now.pid=%d\n",getpid()); sleep(1); } } return 0; } 编辑 makefile 文件,内容如下: OBJS=process.o LDFLAGS = -lpthread -static

CC = yourtoolchainpath/bin/aarch64-linux-gnu-gcc

```
优秀嵌入式技术服务商
```

```
论坛: www.xboot.org
```



process:\${OBJS}

\${CC} -o \$@ \$^ \$(LDFLAGS)

clean:

rm -f process *.o

执行 make 编译,将生成的可执行文件下载到开发板运行,仔细观察串口监控信息:

the perent process is running now.pid=336
the child process is running now.pid=337
the perent process is running now.pid=336
the child process is running now.pid=337
the perent process is running now pid=336
the child process is running now.pid=337
the perent process is running now.pid=336
the child process is running now.pid=337
the perent process is running now.pid=336
the child process is running now.pid=337
the perent process is running now.pid=336
the child process is running now.pid=337
the perent process is running now.pid=336
the child process is running now.pid=337
the perent process is running now.pid=336
the child process is running now.pid=337

7.5 makefile 编程示例

在上面的很多测试程序实例中,我们都编写了一些简单的 makefile 文件。下面我们介 绍 makefile 的基本的语法。

makefile 就好比批处理文件,里面写了一系列集合,当运行 make 编译时,便会按 makefile 提供的命令及顺序完成编译。

这里我们给出三个文件: main.c, func.c, func.h。主程序在 main.c 中, 在 main.c 中程 序会调用 func.c 中的函数, func.c 中的函数又会用到 func.h 中定义的变量。 main.c 文件内容如下:

```
#include "func.h"
```

extern int fd;

{

```
int main(int argc,char **argv)
    fd = open(DEVICE_NAME,0);//打开设备
    if(fd == -1)
    {
         printf("open device %s error n", DEVICE NAME);
         return 0;
    }
    else
    {
         printf("open device %s ok! \n", DEVICE NAME);
    }
    while(1)
    {
         glint_led();
    }
    close(fd);
```

```
优秀嵌入式技术服务商
```



www.9tripod.com 优秀嵌入式技术服务商

return 0;

}

```
该文件会调用 glint_led ()函数,这个函数在 func.c 中。func.c 的内容如下: #include "func.h"
```

void glint_led(void)

{

```
ioctl(fd,LED_ON);
sleep(1);
ioctl(fd,LED_OFF);
sleep(1);
```

}

这里仅仅是一个读取按键的函数,供 main 函数调用。该函数需要用到了一些变量,另 外还需要一些头文件支持,这些都存放在 func.h 中,其内容如下:

#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <sys/ioctl.h>

int fd;

很明显,这是基于 LED 测试程序,人为的分成的三个文件。我们的目的不在于分离代码,而在于学习 makefile 的编写方法。

当不使用 makefile 时,我们使用如下指令编译:

aarch64-linux-gnu-gcc-gcc -o mkfile main.c func.c

编译完成后,将会生成可执行文件 mkfile。将它下载到开发板上运行,和前面的按键测 试完全相同。现在我们尝试编写第一个属于自己的 makefile:

mkfile:main.o func.o

aarch64-linux-gnu-gcc -o mkfilemain.o func.o

main.o:main.c

aarch64-linux-gnu-gcc -c main.c -o main.o

func.o:func.c func.h

aarch64-linux-gnu-gcc -c func.c -o func.o

clean:

rm -f mkfile *.o

执行 make 后,编译器会依次编译 main.c 和 func.c 文件,生成 main.o 和 func.o 文件, 最后将这两个.o 文件打包到可执行文件 mkfile 中。这时将 mkfile 文件下载到开发板运行, 效果和前面的是一样的。我们可以执行 make clean 指令清除生成的.o 文件和可执行文件。

```
优秀嵌入式技术服务商
```



www.9tripod.com

优秀嵌入式技术服务商

makefile 具有很强大的推理功能,我们完全可以简化上面的代码。优化后的代码如下:

OBJS=main.o func.o

CC= aarch64-linux-gnu-gcc

mkfile:\${OBJS}

 ${CC} -0$

main.o: func.o:func.h

clean:

rm -f mkfile *.o

可见,这次比上面的完整版要简化多了。前面通过变量 OBJS 定义了要编译的源文件, 变量 CC 给出了交叉编译工具。\$@ 表示目标文件的全称,即 mkfile, \$^表示所有被依赖的 文件,并以空格分开,即 main.o func.o。后面的 clean 为清除指令,执行 make clean 后会执 行 clean 后面的指令。需要注意的是,rm 指令后面千万不要使用\$@符号来表征我们要删除 的目标文件,因为这时候\$@已经不再表示 mkfile 了,而表示 clean。同样,使用 make 指令 编译,一样能够生成我们需要的目录文件 mkfile。

上面的 makefile 使用了变量以及预定义变量。第一句即定义了变量 OBJS,将它赋值为 main.o func.o,第二句定义了变量 CC,将它赋值为一个交叉编译工具定义。引用变量时,通过\${*}表示,这里*表示前面定义的变量。

上面使用了\$@和\$^两个预定义变量, GNU make 主要有以下七种预定义变量:

\$*	不包含扩展名的目标文件名称
\$+	所有的依赖文件,以空格分开,以出现的先后为序,可能包含重复的依赖文件
\$<	第一个依赖文件的名称
\$?	所有的依赖文件,以空格分开,这些依赖文件的修改日期比目标的创建日期晚
\$@	目标的完整名称
\$	所有的依赖文件,以空格分开,不包含重复的依赖文件
\$%	如果目标是归档成员,则该变量表示目标的归档成员名称。例如,如果目标名
	称为 mytarget.so(image.o),则\$@ 为 mytarget.so,而\$% 为 image.o。

对比以上几种编译方式,我们不然发现,其实最简的还是第一种,因为它就一句话就 搞定了。那么在 makefile 中,我们是否也可以精简到只有一句话呢?答案是肯定的。我们继 续利用 makefile 强大的推理功能进行简化,得到如下 makefile 代码:

OBJS=main.o func.o

CC= aarch64-linux-gnu-gcc

mkfile:\${OBJS}

\${CC} -0 \$@ \$^

clean:

rm -f mkfile *.o

这次,makefile 真正编译的代码,就只有上面红色部分一条指令了。和前面比较,不难 发现,单独对 main.c 和 func.c 两个文件编译的指令已经去掉了。前面我们提到,makefile 具有强大的推理功能,我们在生成目标文件 mkfile 时,makefile 会推理出它需要 main.c 和

优秀嵌入式技术服务商

论坛: www.xboot.org



func.c 两个文件,因此它首先就会去编译这两个文件,最后再执行目标文件的生成。因此我 们完全可以将它们省去。

这里只是 makefile 的一点基础,读者可以借助于其他书籍对 makefile 作更深一层的了解。

7.6 mplayer 移植

有两种移植 mplayer,第一种方法就是直接下载 mplayer 源码包,手动配置编译环境, 再通过交叉编译环境编译。该方法比较传统,我们有更方便的方法,直接在 buildroot 中选 中 mplayer,再 make 即可, buildroot 会自动下载 mplayer 源码包,自动配置编译环境并编译。

在 buildroot 的 menuconfig 配置选项中,进入 Target packages-> Audio and video applications 选项,选中 mplayer,如下图所示:



保存退出,在 buildroot 下直接 make,生成文件系统后更新到开发板,mplayer 就打包 进文件系统了。

7.7 madplayer 移植

madplayer 的移植方法和 mplayer 的移植方法完全雷同,直接在 buildroot 中选中 madplayer 的选项,再 make 即可。

7.8 屏幕抓图工具 gsnap 移植

屏幕抓图的方法有多种,有不少爱好者自己动手写抓图小程序。这里我们使用 JPEG 库 来处理。具体用到了 jpegsrc.v6b.tar.gz 和 gsnap.tar.gz 两个源码包(这 2 个压缩包请读者自己 到网络上面下载)。

一: 安装 libjpeg

解压 jpeg 库源码包,进入根目录:

```
tar zxf jpegsrc.v6b.tar.gz
```

cd jpeg-6b

```
二: 配置编译环境:
```

./configure

--prefix=/home/lixu/projects/3399/x3399_marshmallow/buildroot/output/host/opt/ext-toolchain

```
优秀嵌入式技术服务商
```

论坛: www.xboot.org



--exec-prefix=/home/lixu/projects/3399/x3399_marshmallow/buildroot/output/host/opt/ext-toolch ain --target=arm-linux --enable-shared --enable-static

CC=/home/lixu/projects/3399/x3399_marshmallow/buildroot/output/host/opt/ext-toolchain/bin/aa rch64-linux-gnu-gcc

AR=/home/lixu/projects/3399/x3399_marshmallow/buildroot/output/host/opt/ext-toolchain/bin/aa rch64-linux-gnu-ar rc

AR2=/home/lixu/projects/3399/x3399_marshmallow/buildroot/output/host/opt/ext-toolchain/bin/a arch64-linux-gnu-ranlib

三: 配置好之后, make 编译, 如果提示缺少 libtool 文件:

iixu@lixu:-/project5/3399/x3399_marshmallow/gsnap/jpeg-6b\$ make /libtool --mode-compile /home/lixu/projects/3399/x3399_marshmallow/buildroot/output/host/opt/ext-toolchain/bin/aarch64-linux-gnu-gcc -O2 -I. -c ./jcaplmin.c nake: ./libtool : 命令未找到

下载 libtool-2.2.6a.tar.gz,解压编译获得 libtool 工具。

ixu@lixu:~/projects/3399/x3399_marshmallow/gsnap/jpeg-6b\$ cp /usr/share/libtool-2.2.6/libltdl/config/config.guess ./ ixu@lixu:~/projects/3399/x3399_marshmallow/gsnap/jpeg-6b\$ cp /usr/share/libtool-2.2.6/libltdl/config/config.sub ./

再 make clean, 重复之前的第二步: 配置编译环境。

四:在 x3399_marshmallow/buildroot/output/host/opt/ext-toolchain下建立 man/man1 目录:

 $cd \quad x3399_marshmallow/buildroot/output/host/opt/ext-toolchain$

mkdir -p man/man1

五:编译,安装

make

make install

这时,在 x3399_marshmallow/buildroot/output/host/opt/ext-toolchain/man/man1 目录下将 会生成以下文件:

cjpeg.1 djpeg.1 jpegtran.1 rdjpgcom.1 wrjpgcom.1

在 x3399_marshmallow/buildroot/output/host/opt/ext-toolchain/lib 目录下生成以下文件:

libjpeg.a libjpeg.la libjpeg.so libjpeg.so.62 libjpeg.so.62.0.0

六: 解压 gsnap

tar zxf gsnap.tar.gz

cd gsnap

七: 修改 makefile

all:

/home/lixu/projects/3399/x3399_marshmallow/buildroot/output/host/opt/ext-toolchain/bin/aa rch64-linux-gnu-gcc -g gsnap.c -ljpeg -o gsnap -static

clean:

rm -f gsnap

八:编译,得到可执行文件 gsnap

make

九:将 jpeg 库文件复制到文件系统的 lib 目录,注意保持文件的链接属性

cp -a libjpeg.s* "文件系统路径"/lib

十:将可执行文件 gsnap 复制到文件系统的 sbin 目录

cp gsnap "文件系统路径"/sbin

```
优秀嵌入式技术服务商
```

论坛: www.xboot.org



十一:重新制作文件系统,下载到开发板上,使用如下命令即可截获图形界面:

gsnap 1.jpg /dev/fb0

同样可以将图片保持为 bmp,png 等其他格式。



第8章 其他产品介绍

8.1 核心板系列

处理器型号	核心板型号	备注
S3C6410	X6410CV1	200PIN 插针接口
S5PV210	X210CV3	180PIN 邮票孔接口
	X210CV4	144PIN 邮票孔接口
	G210CV1	200PIN 金手指接口
	I210CV2	200PIN 插针接口
Exynos4412	X4412CV3	180PIN 邮票孔接口
S5P4418	X4418CV3.3	180PIN 邮票孔接口
	I4418CV2	200PIN 板对板连接器
S5P6818	X6818CV3.3	180PIN 邮票孔接口
	I6818CV2	200PIN 板对板连接器
RK3288	X3288CV3	180PIN 邮票孔接口
	I3288CV1	220PIN 邮票孔接口
RK3399	X3399CV2.1	200PIN 邮票孔接口

8.2 开发板系列

处理器型号	开发板型号	备注
S3C6410	x6410开发板	x6410cv1 评估板
S5PV210	x210开发板	x210cv3 评估板
	g210开发板	g210cv1 评估板
	i210开发板	i210cv2 评估板
Exynos4412	x4412 开发板	x4412cv3 评估板
S5P4418	x4418 开发板	x4418cv3 评估板
S5P6818	x6818 开发板	x6818cv3 评估板
	i6818 开发板	i6818cv2 评估板
RK3288	x3288 开发板	x3288cv3 评估板
	i3288 开发板	i3288cv1 评估板
RK3399	x3399 开发板	x3399cv2.1 评估板

8.3 卡片电脑系列

处理器型号	卡片电脑型号	备注
Exynos4412	ibox4412 卡片电脑	
S5P4418	ibox4418 卡片电脑	
S5P6818	ibox6818 卡片电脑	
RK3399	ibox3399卡片电脑	

说明:产品详细规格,以及更多其他产品请关注九鼎创展官方网站和论坛。

论坛: www.xboot.org