<https://www.bilibili.com/video/BV1A84y1P7VJ/?spm_id_from=333.788.recommend_more_video.0&vd_source=af21ee908c04895a95a22f7a0c9e0013>

【正点原子】AI人工智能深度学习(RV1126/RK3568/RK3588)-第1期 准备篇 笔记：

**1.0内容介绍**

开发板使用RV1126本节讲解入门知识：

1.简单了解RV1126的SDK，在一键编译之后，生成了哪些目录和文件。

2.知道所需要的文件位置在哪里查找，在哪个目录中查找。

介绍了文件分区系统的情况，以及怎么拓展自己的分区。

3.可能希望系统开机后，自动跑AI应用。所以介绍了RV1126的开机自启动流程。实现开机自启动的功能

4.可能涉及使用以太网Etho进行推流的时候，能固定一个静态IP地址，所以讲解了怎么在RV1126中设置一个静态IP地址。

5.RKMedia接口的使用，如何使用RK实现视频的采集和处理。介绍了如何编译和测试。

**1.1瞄一瞄SDK编译之后的目录**

本节分为两部分：

1、SDK一键编译后Buildroot下生成的目录：

2、Buildroot/output/alientek\_rk1126目录。

**1、SDK一键编译后Buildroot下生成的目录：**

SDK编译后，默认会在output目录下生成两个目录：alientek\_rv1126和alientek\_rv1126\_recovery.

Buildroot/output/alientek\_rv1126

存放文件系统镜像以及交叉编译链（真正的**根文件系统镜像**存放目录）

Buildroot/output/alientek\_rv1126\_recovery

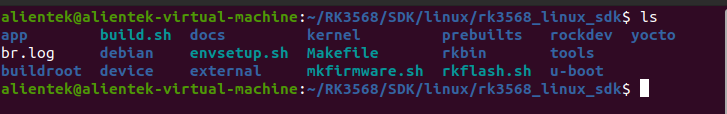
(非必须，负责系统升级等，kernal+dtb+ramdisk组成的根文件系统)

主要保存recovery里系统镜像。recovery里系统镜像主要是由内核（kernel）、设备树（dtb）、ramdisk组成到的根文件系统。recovery主要负责系统升级。

一般是不需要编译recovery的。只需要编译出alientek\_rv1126根文件系统镜像

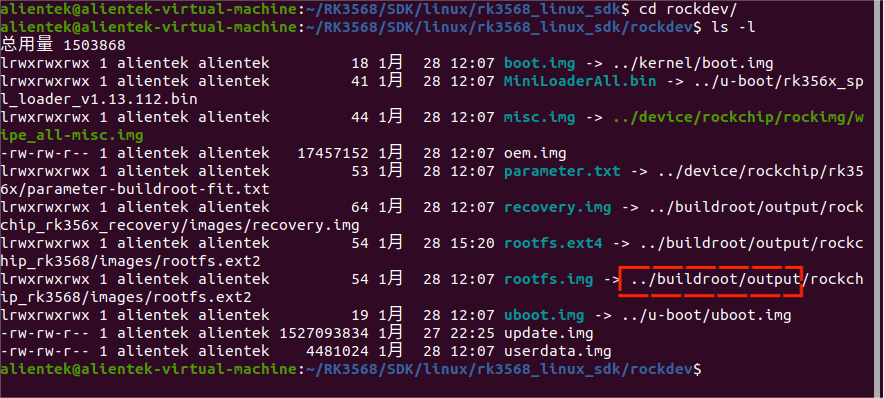
/\*

这就是我们自己的SDK目录，sdk一键编译之后，默认是将镜像打包到rockdev目录下，



**1.进入rockdev目录，**

2.使用ls -l命令查看镜像详情。



我们可以看到有一些文件其实是软链接文件，并且可以看到具体软链接到了哪里，

比如：

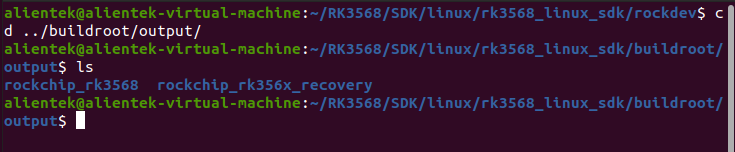
boot.img就是由kernel+dtb组成的，软链接到了../kernel/boot.img,也就是说，**内核镜像文件实际上是“../kernel/boot.img”文件**。

继续看目录：

rootfs.ext4和rootfs.img就是根文件系统镜像，对应的实际位置如图。

recovery.img就是recovery系统镜像。对应目录的位置如图。

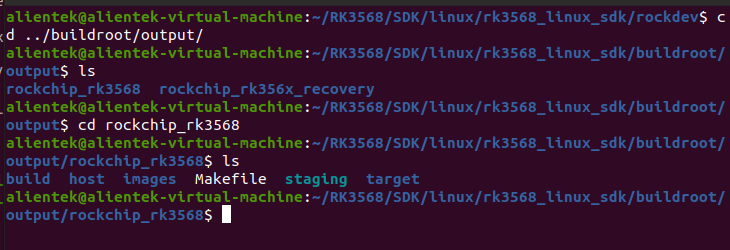
3.进入如图的rootfs.img 软链接的 ../buildroot/output/目录下：



可以看到：output目录下有两个目录（她是3个：alientek\_rv1126、alientek\_rv1126\_libs、alientek\_rv1126\_recovery）。其中rockchip\_rk3568 和rockchip\_rk356x\_recovery是SDK一键编译之后默认生成的目录.(她的alientek\_rv1126\_libs是前面编译BSP8的时候生成的)。

我们重点是查看rockchip\_rk3568目录。

4.进入rockchip\_rk3568：cd rockchip\_rk3568 ，ls

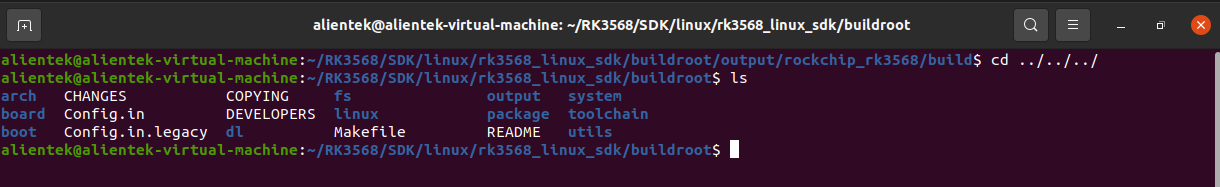


里面的内容如上图，我们依次分析这些目录：

**1、Build：build**指 编译，故用于存放buildroot编译过程中，**下载的包（如dl目录下的包）**，或者**SDK中的组件源码包(入external中的包)**（可以cd进入查看，有很多的包），

这些包的来源：

另开一个中端：进入如下路径：



再进入 dl目录：cd dl/

这个目录存放的就是buildroot编译过程中在线下载的包，由于是在线下载的，网络质量不佳可能下载得很慢，导致buildroot构建所需时间很长。所以建议直接将正点原子的dl压缩包提取到该路径。

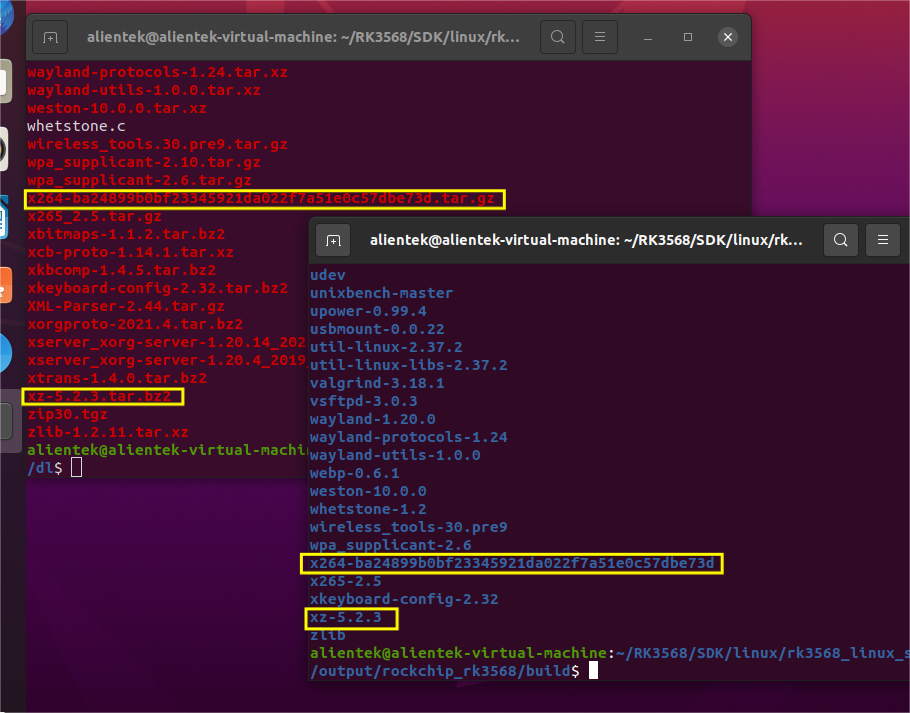
我们对比：

蓝色：

~/RK3568/SDK/linux/rk3568\_linux\_sdk/buildroot/output/rockchip\_rk3568/build 路径下的文件 和

红色：

~/RK3568/SDK/linux/rk3568\_linux\_sdk/buildroot/dl 路径下的文件得：



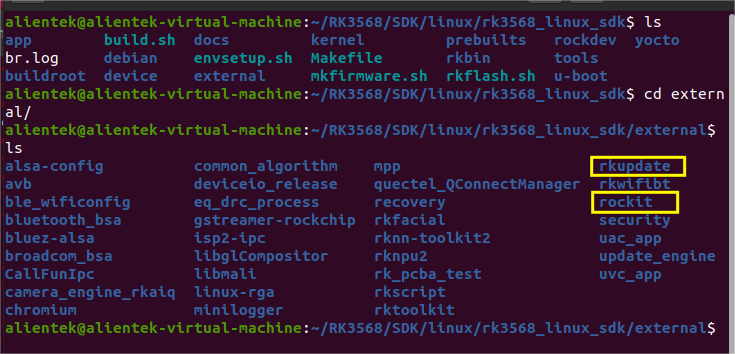
**1.蓝色路径下的包，实质上是从红色路径下解压过去的。**

再来看：

~/RK3568/SDK/linux/rk3568\_linux\_sdk/external 路径：

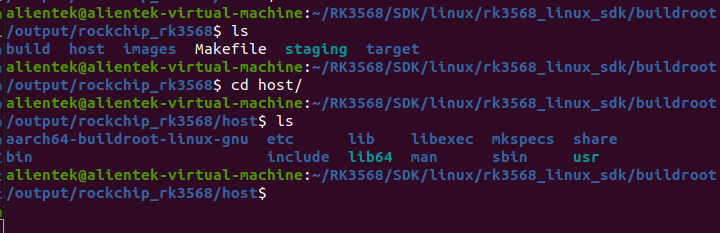
里面有：rkmedia、rkupdate、rockit、rockface等等，他们在蓝色路径下可以找到对应的，但红色路径下没有这几个目录。

2.因此，**蓝色路径build下的这几个目录，实质上是从external拷贝的**。



因此，**buil目录**，故用于存放buildroot编译过程中，**下载的包（如dl目录下的包）**，或者**SDK中的组件源码包(入external中的包)，**当它有了这些包之后才便于编译。

**2、Host：**编译后，交叉编译工具的目录，包括交叉编译工具，以及库文件和头文件等等。



1.查看bin目录：

cd bin：进入bin目录，可以看到里面是编译后的可执行文件。我们可使用该目录下编译出来的的交叉编译工具来编译我们的应用程序，如：

我们可以在makefile文件中指定交叉编译工具的绝对路径:(下是她的，我的没找到...后续再看，课程时间：11.04)

（我觉得：

路径应修改为：

**/home/alientek/RK3568/SDK/linux/rk3568\_linux\_sdk/buildroot/output/rockchip\_rk3568/host/bin**

编译工具可能是(找了个相似的，猜的)：**aarch64-linux-g++）**

IMG_256

我们可以通过如上的指定，来编译我们的应用程序。

2.进入aarch64-buildroot-linux-gnu目录：

cd aarch64-buildroot-linux-gnu

进入sysroot：

cd sysroot/

进入 usr:

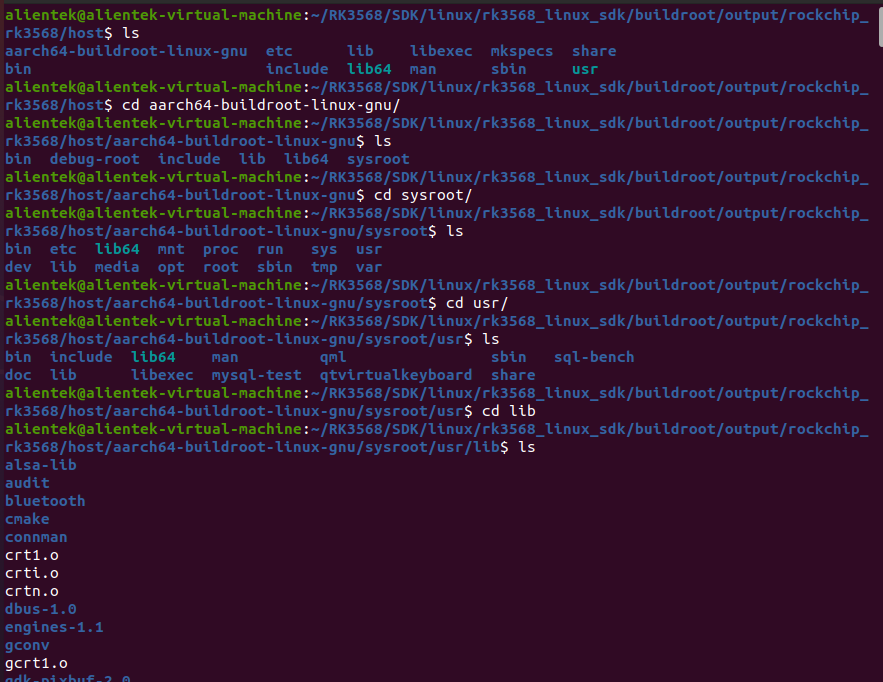
cd usr/

进入lib：

cd lib

ls查看详情：

实际操作如下：



路径：

/home/alientek/RK3568/SDK/linux/rk3568\_linux\_sdk/buildroot/output/rockchip\_rk3568/host/aarch64-buildroot-linux-gnu/sysroot/usr/lib

这个目录下，就是库文件，所有文件系统编译后得到的库文件都在这个目录下可以找到。例如：libzbar.so.0就是开源的二维码算法库，我们可以通过调用该库实现二维码识别，

头文件就在：（在上面的路径下输入：cd .. 、 cd include/）

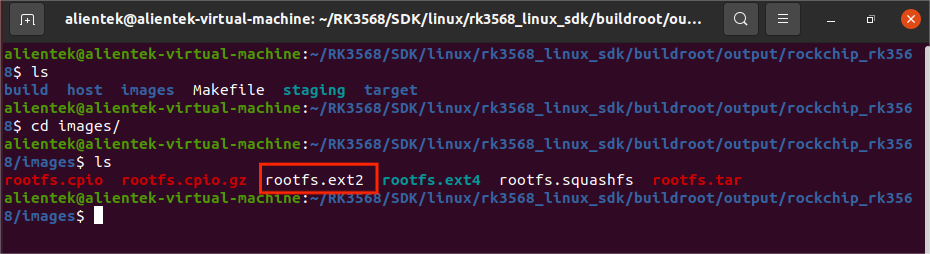
/home/alientek/RK3568/SDK/linux/rk3568\_linux\_sdk/buildroot/output/rockchip\_rk3568/host/aarch64-buildroot-linux-gnu/sysroot/usr/include

这个目录放的就是配置buildroot组建的头文件。前面提到的zbar开源二维码算法库的头文件就在这里：zbar.h（**我们没有该库**）

**3、Images：**

回到~/RK3568/SDK/linux/rk3568\_linux\_sdk/buildroot/output/rockchip\_rk356路径。

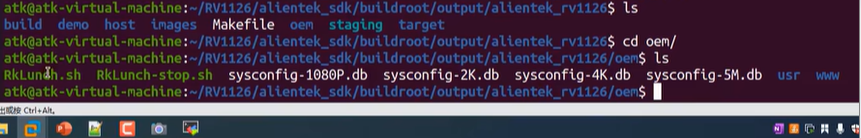
images就是镜像：它就是存放文件系统镜像的，就是红框中的内容。



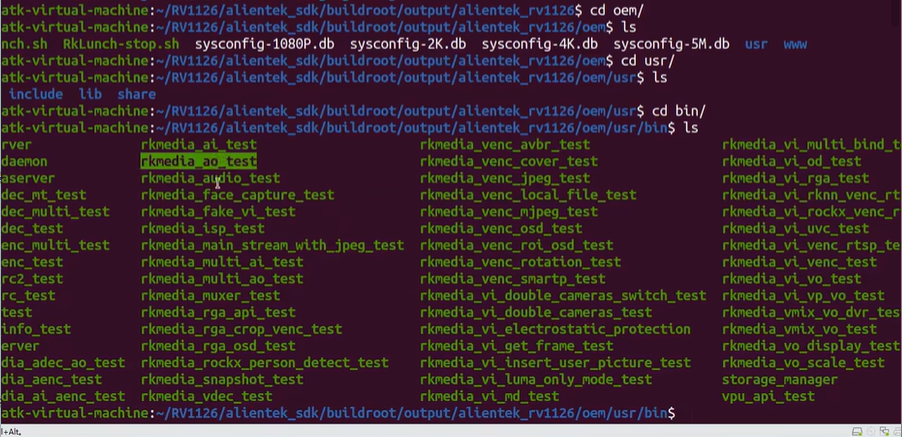
**4.demo：**存放例程(我们没有demo目录)

**5.oem:oem**目录下包含出厂的一些库，还有可执行文件，以及脚本和配置文件等：（我们没有）

如她所说：第一和第二是出场系统摄像头脚本，执行第一个脚本文件就启动摄像头应用，执行第二个文件就关闭了摄像头应用。



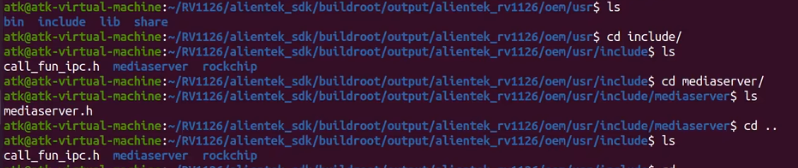
进入 **./usr/bin目录**，里面存放了一些可执行文件：



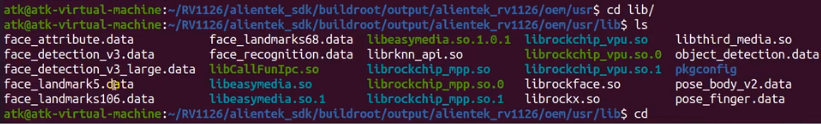
如rkmedia\_ai.test、rkmedia\_ao.test、rkmedia\_audio.test就是rkmedia的可执行文件，他们就是编译rkmedia例程得到的可执行文件。我们可以运行这些文件来测试rkmedia。

里面还有mipi相关的测试文件。

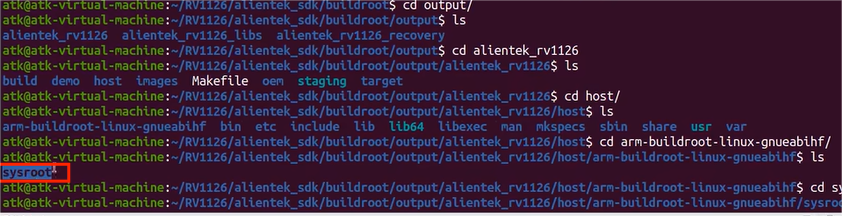
进入 ./usr/include目录：里面存放的就是一些和测试相关的头文件。



进入 ./use/lib目录：lib里面存放的就是库文件，就是**buil的SDK中的组件要用到的模型文件，**是rk已经封装好了的。



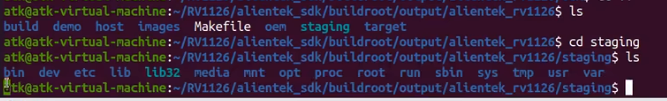
**6.staging** ： 该目录是指向host中的sysroot的 一个符号链接：



sysroot内容是：

IMG_256

而staging内容如下：



可看到两个一模一样。

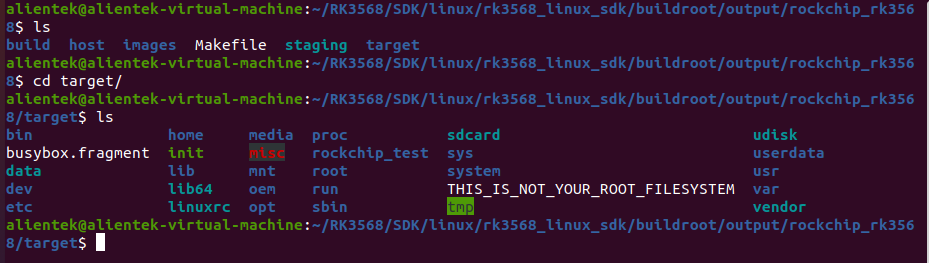
**7.target：**target指目标。目标就是我们最终要获取到的。Target目录是用来制作loop文件系统（制作/挂载loop文件系统）的，这个目录存放的就是linux系统的基本的目录结构。

如：bin目录包含了一些可执行文件。

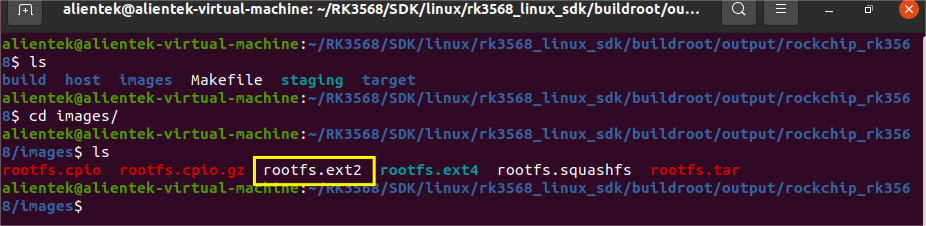
Lib下有一些库

...

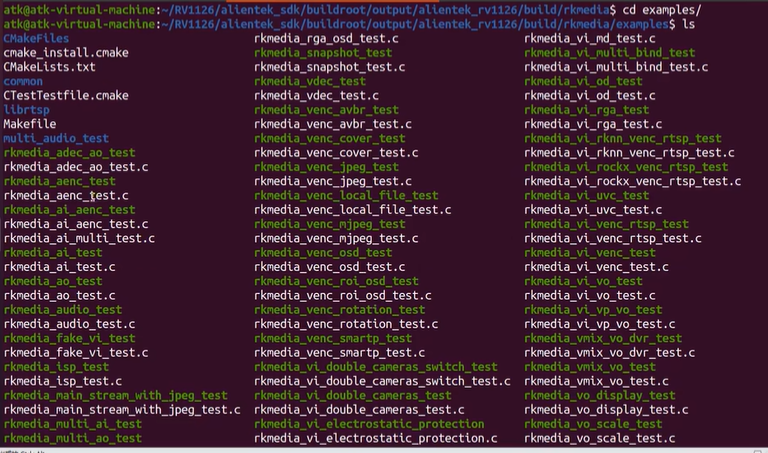
**我们烧录的根文件系统其实就是长target目录这个样子：**



那么output目录就介绍完了，我们要烧录到开发板中的镜像，就是rootfs.ext2



后续如果要编译一些组件或者软件包，就可以在build目录下 来操作。



比如修改好文件之后，直接在该目录下编译，然后把编译生成的可执行文件放到开发板中去测试。

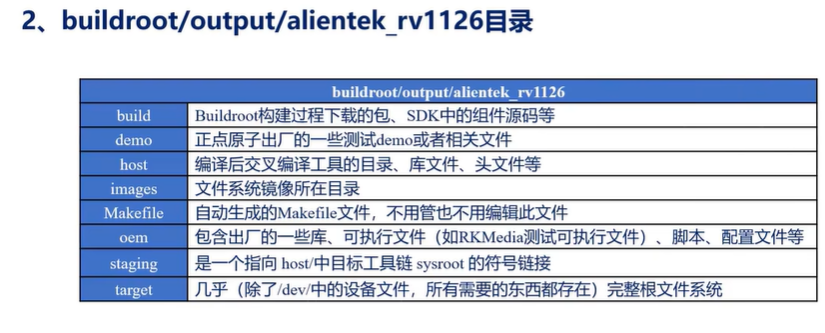
\*/

回到ppt:

2.buildroot/uotput/alientek\_rv1126目录

真正的根文件系统镜像存放目录





初步了解Buildroot文件系统，可参考：

正点原子工程师翻译的《04【正点原子】Buildroot用户手册中文版（正点原子翻译）》