

准备工作:

开始拷贝:

制作整包:

## 准备工作:

---

1. 使用USB转TTL串口线通过开发板debug串口连接开发板和电脑，并通过远程终端（如Xshell）登录开发板Ubuntu系统；
2. 使用网线连接开发板,并确定开发板和电脑在同一网段（linux中键入ifconfig可查看ip），若不在同一网段，设置到开发板同一网段IP地址：

```
ifconfig eth0 192.168.1.66 #设置IP地址，eth0为网卡名称
```

## 开始拷贝:

---

1. 在开发板的UBUNTU系统上面安装软件rsync:

```
sudo apt-get install rsync
```

若遇到Could not get lock /var/lib/dpkg/lock - open (11: Resource temporarily.....相关问题，百度删除相关文件即可（或重启）

- 2.在开发板的UBUNTU系统上面安装软件openssh的服务端openssh-server:

```
sudo apt-get install openssh-server
```

3. 修改开发板root登录权限:

```
sudo vi /etc/ssh/sshd_config
```

在sshd\_config中修改PermitRootLogin选项如下，用以确保root登录权限足够

```
# Authentication:
LoginGraceTime 120
#PermitRootLogin prohibit-password
PermitRootLogin yes
StrictModes yes
RSAAuthentication yes
```

#### 4.重启开发板或者重启ssh服务

```
/etc/init.d/ssh restart
```

#### 5.PC端安装openssh客户端openssh-client

```
sudo apt-get install openssh-client
```

#### 6.PC端安装rsync

```
sudo apt-get install rsync
```

#### 7.PC新建目录ubuntu\_make并进入：

```
mkdir ubuntu_make
cd ubuntu_make
```

#### 在PC端当前目录新建ubuntu目录：

```
mkdir ubuntu
```

#### 8.同步数据：

```
sudo rsync -avx root@<****开发板IP>:/ ubuntu/
```

#### 开发板IP通过ifconfig确认

#### 9.同步完成，PC进入ubuntu文件夹下面增加文件etc/onlyone.sh

```
cd ubuntu
sudo vi etc/onlyone.sh
```

#### onlyone.sh内容为：

```
#!/bin/sh

read line < /proc/cmdline

for arg in $line; do

    if [ "5" -le "$(expr length $arg)" ]; then

        if [ "root=" = "$(expr substr $arg 1 5)" ]; then

            {

                debug_arg=$(expr $arg : 'root=\(.\\*\)\')

                resize2fs $debug_arg

            }

        fi

    fi

done
```

10.PC端ubuntu文件夹下更改该文件权限:

```
sudo chmod 777 etc/onlyone.sh
```

11.返回上级目录 (make\_ubuntu) , 并拷贝make\_ubuntu.sh到当前文件夹:

```
cd ..

sudo cp */make_ubuntu.sh ./ #*为pc端原make_ubuntu.sh所在路径
```

运行make\_ubuntu.sh制作ubuntu固件:

```
sudo ./make_ubuntu.sh ubuntu
```

或制作ubuntu\_core:

```
sudo ./make_ubuntu.sh ubuntu_core
```

脚本运行完成, 在ubuntu\_make目录下生ubuntu-rootfs/ ubuntu-rootfs-core固件

```
rpdzkj@rpdzkj-dev:~/ubuntu_make$ ls
make_ubuntu.sh  ubuntu  ubuntu-rootfs
rpdzkj@rpdzkj-dev:~/ubuntu_make$
```

最后我们还需要将ubuntu-rootfs/ubuntu-rootfs-core文件重命名加上.img后缀

## 制作整包:

---

将上述ubuntu-rootfs.img/ubuntu-rootfs-core.img对应替换源码目录下ubuntu/目录中的ubuntu.img/ubuntu-core.img,返回上层目录,运行编译脚本:

```
./build.sh firmware
./build.sh updateimg
```

即可在rockdev/下生成update-\*.img