## 1 . 系统编译环境的搭建

**双Android简介**

双android系统由三个系统构成，最底层的是SSOS系统，上层的是两个android系统，分别对应内域和外域。开机后进入SSOS系统，由SSOS系统创建容器，在分别启动内域和外域。

**系统需求**

本文档介绍的android编译环境搭建基于Ubuntu 12.04及Ubuntu 12.04以上版本。

**安装[Android](http://www.linuxidc.com/topicnews.aspx?tid=11" \t "_blank)6.0环境编译需要的相关包**

根据Ubuntu系统的版本不同，其安装Android环境编译需要的相关包也不尽相同。

如果Ubuntu系统为12.04，则在terminal中输入如下命令：

sudo apt-get install git gnupg flex bison gperf build-essential zip curl libc6-dev libncurses5-dev:i386 x11proto-core-dev libx11-dev:i386 libreadline6-dev:i386 libgl1-mesa-glx:i386 libgl1-mesa-dev g++-multilib mingw32 tofrodos python-markdown libxml2-utils xsltproc zlib1g-dev:i386

$ sudo ln -s /usr/lib/i386-linux-gnu/mesa/libGL.so.1 /usr/lib/i386-linux-gnu/libGL.so

通过上述的安装包可看出，在Ubuntu 12.04版本中安装的环境编译相关包包含一些32位的库，例如ia32-libs，但是在Ubuntu 12.04以上的版本中（例如Ubuntu 13.10）已经废弃了这些32位的库，所以我们需要其替代品来实现其相应的功能。如果Ubuntu系统为14.04,则在terminal中输入如下命令：

sudo apt-get install git-core gnupg flex bison gperf build-essential zip curl zlib1g-dev gcc-multilib g++-multilib libc6-dev-i386 lib32ncurses5-dev x11proto-core-dev libx11-dev lib32z-dev ccache libgl1-mesa-dev libxml2-utils xsltproc unzip

**安装openJDK**

在terminal中输入命令：

sudo apt-get install openjdk-7-jdk

sudo apt-get install openjdk-7-jre

默认安装在usr/lib/jvm下，安装成功了，可以通过java -version进行查看，如要设为默认的，可在terminal中输入命令：

sudo update-alternatives --install /usr/bin/java java /usr/lib/jvm/java-1.7.0-openjdk-amd64/bin/java 300  
sudo update-alternatives --install /usr/bin/javac javac /usr/lib/jvm/java-1.7.0-openjdk-amd64/bin/javac 300  
sudo update-alternatives --config java

## 2. 代码的下载

**下载 repo 工具:**

mkdir ~/bin

curl https://storage.googleapis.com/git-repo-downloads/repo > ~/bin/repo

chmod a+x ~/bin/repo

**修改REPO\_URL:**

vi ~/bin/repo

REPO\_URL = 'ssh://your\_gerrit\_account@cactus.insyber.com:29422/repo'

REPO\_REV = 'default'

**关于Gerrit帐号**

公司的 https://cactus.insyber.com 请找刘凤兰。

BSP内部的: http://bsp.insyber.com/ 请发邮件给李翔和凌伟。

**公钥添加**

生成公钥可用"ssh-keygen"命令, 若之前已经生成过，不必重新生成。

将用户根目录下cat ~/.ssh/id\_rsa.pub 内容全部复制到

gerrit （两个gerrit账户都需添加） --> Settings --> SSH Public Keys 密钥添加处

## REPO常用命令

查看帮助

repo help

查看命令详细帮助

repo help command

初始化

repo init -u URL[options]

同步

repo sync [project\_list](空代表所有工程)

查看修改

repo status [project\_list](空代表所有工程)

执行shell命令

repo forall [project\_list](空代表所有工程) -c “shell命令”

例如：repo forall -c git checkout castle\_internal

将commits push 到 gerrit上

repo upload [project\_list](空代表所有工程)

在执行repo upload前先确认本地配置是否正确

cactus.insyber.com 注册邮箱

登录https://cactus.insyber.com/#/settings/contact，单击Register New Email...按钮

邮箱格式：[yourname@cmos.com。然后登录内网邮箱确认一下。](mailto:yourname@cmos.com。然后登录内网邮箱确认一下。)

修改gitconfig

vi ~/.gitconfig

修改

[user]

[Email=yourname@cmos.com](mailto:Email=yourname@coms.com)

添加

[review "cactus.insyber.com"]

usename=your\_gerrit\_account

如果配置正确，但是使用repo upload 无法提交，试试：

git push cactus local\_branch\_name:refs/for/remote\_banch\_name 单个工程push 或者

repo forall [project\_list] -c git push cactus local\_branch\_name:refs/for/remote\_banch \_name

**ssos系统下载编译**

**添加host**

sudo vi /etc/hosts

192.168.160.124 cactus.insyber.com

192.168.12.149 bsp.insyber.com

1，安装编译需要工具：

sudo apt-get install autoconf automake libtool libc++-dev

2，下载源码

ssos (For MTK):

repo init -u ssh://您的BSPGerrit帐号@bsp.insyber.com:29418/manifest -b castle\_devel --repo-url=ssh://您的BSPGerrit帐号@bsp.insyber.com:29418/repo --repo-branch=devel

repo sync -c

3，编译

1.、source build/envsetup.sh

2、lunch mtk\_mt6735\_debug

3、make -j8

**内外域代码下载、编译、烧写**

建议内外域代码分别建目录

双安卓 For MTK(外域) :：

repo init -u ssh://您的cactus帐号@cactus.insyber.com:29422/manifests -b castle\_external

repo sync -c

双安卓 For MTK(内域):

repo init -u ssh://您的cactus帐号@cactus.insyber.com:29422/manifests -b castle\_internal

repo sync -c

编译(内外域)：

source build/envsetup.sh

./buildmodem\_L08A.sh

lunch full\_aus6735\_66c\_c\_m-userdebug

make -j4

编译完成后，在Android工程目录下的 out/target/product/aus6735\_66c\_c\_m/ 目录里可以找到编译生成的所有文件。

单独编译某个模块（内外域）：

假如修改了系统应用Calculator

1. 先加载环境：

source build/envsetup.sh

./buildmodem\_L08A.sh

lunch full\_aus6735\_66c\_c\_m-userdebug

1. 编译代码

(1) make + 模块名(Android.mk 文件中的定义LOCAL\_PACKAGE\_NAME := Calculator)

make Calculator

(2) mm: 编译当前目录下所有模块

cd packages/apps/Calculator/ && mm

(3) mmm: 编译指定目录下所有模块

mmm packages/apps/Calculator/

3. 编译结果保存在 out/target/product/aus6735\_66c\_c\_m/system/app/Calculator/Calculator.apk

编译其他的模块可以在 out/target/product/aus6735\_66c\_c\_m/system 目录下对应的文件夹找到生成的target

## 3. 代码烧写

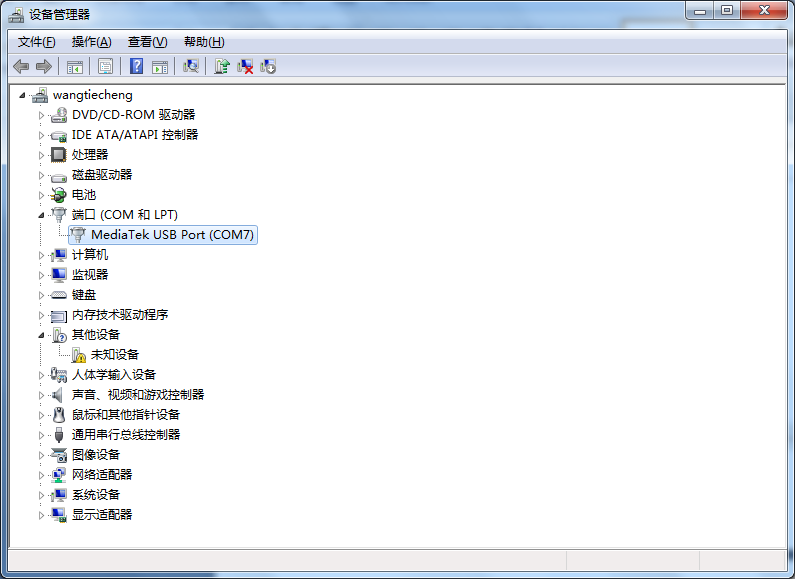
**烧写工具使用步骤：**

1) MTK烧写工具驱动安装

MTK烧写工具及驱动请在<https://exmail.qq.com/cgi-bin/ftnExs_download?k=04346232358991cdc467b5a91136030616035153000004040d195006030619000b52061f0e500600140c5556055755055a035b07533633345440091c4457463404&t=exs_ftn_download&code=94b26644&s=email>上下载，其中驱动程序位于该压缩包中的mtk\tool\SP\_Drivers\_EXE\_v2.0\CDC目录下，当手机通过USB数据线连接电脑时，提示安装驱动文件则选中该目录即可。

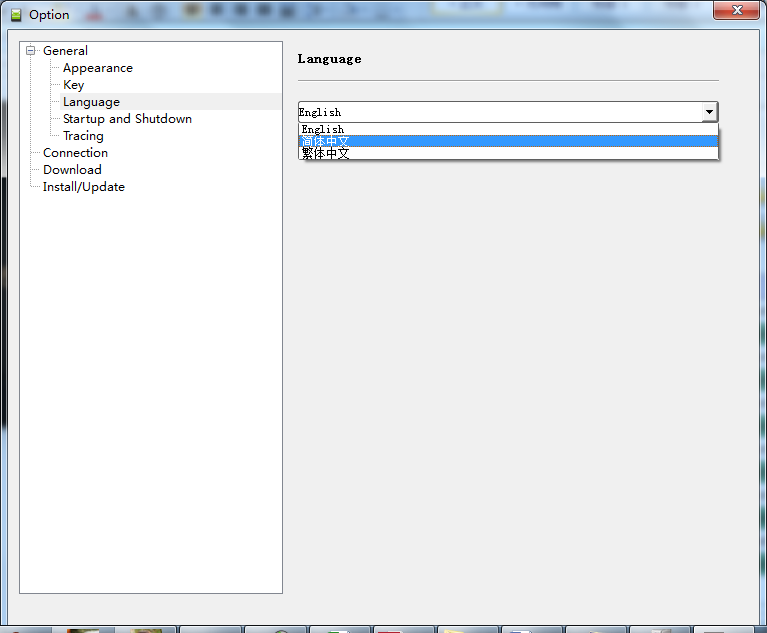
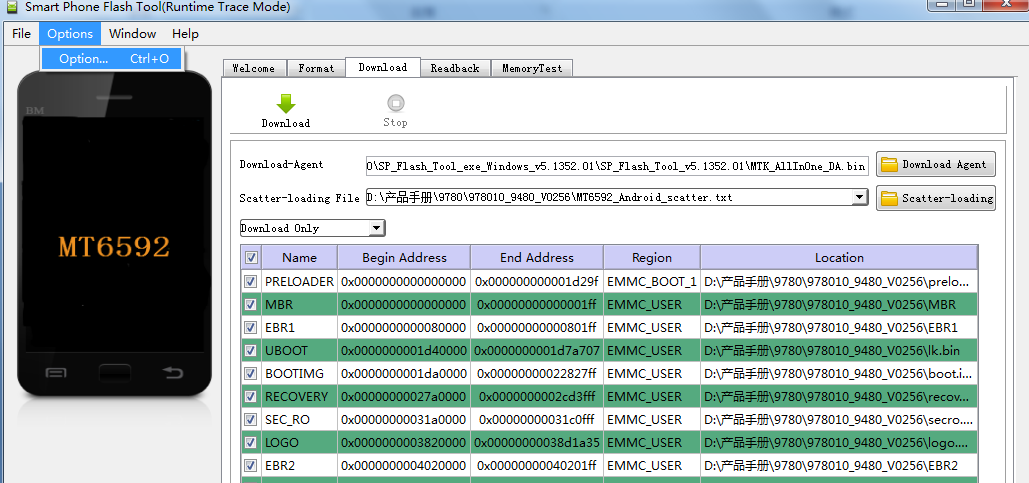
注：工具请在非虚拟机环境下使用。

检查设备是否正确安装。如出现下图所示情况（识别MediaTek USB Port），则说明驱动安装成功。

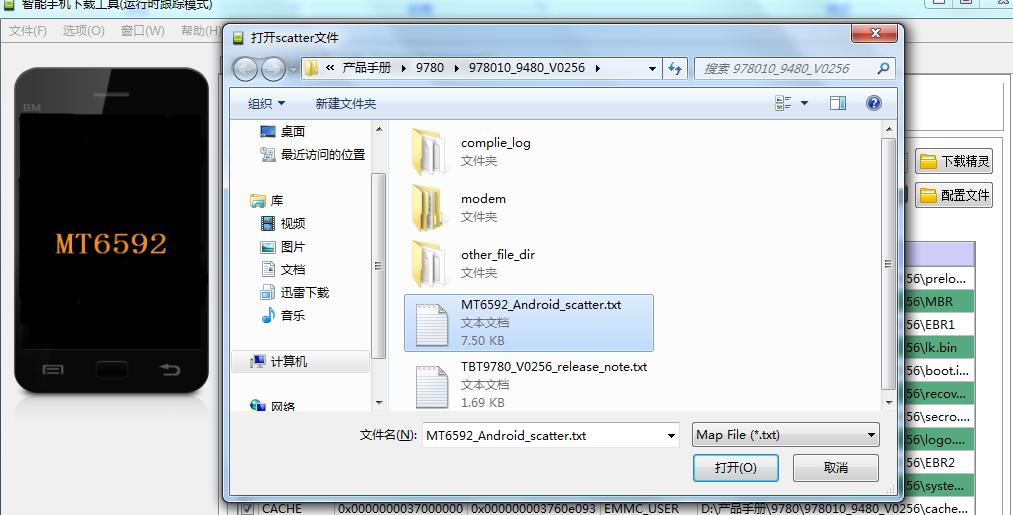


2) 烧写工具配置

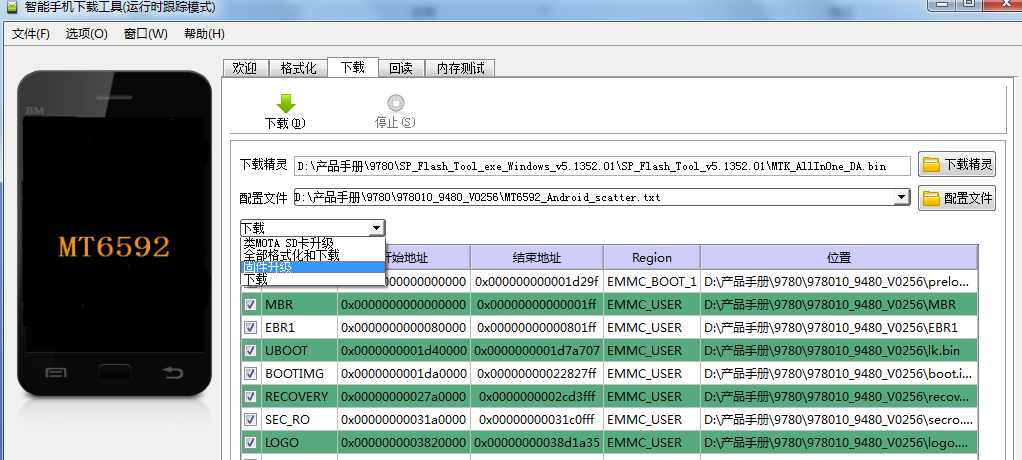
打开工具，在开始升级前，可选择显示语言为中文，其工具所在目录为mtk\tool\MTK\_Flash\_Tool，打开工具后，选择菜单栏上的Options选项：



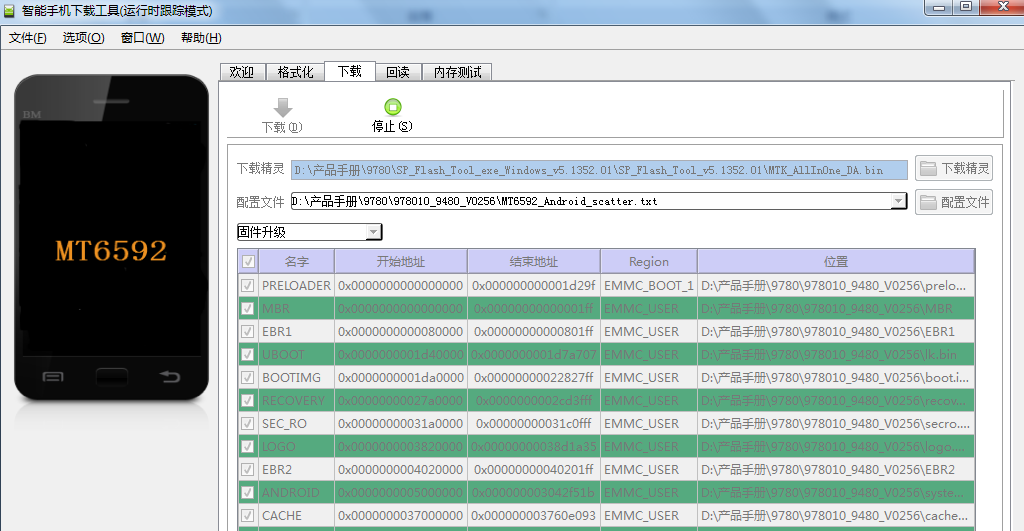
3) 点击右侧配置文件按钮，选择升级软件包(包需解压)注意看文件名，必要时可以选择两次



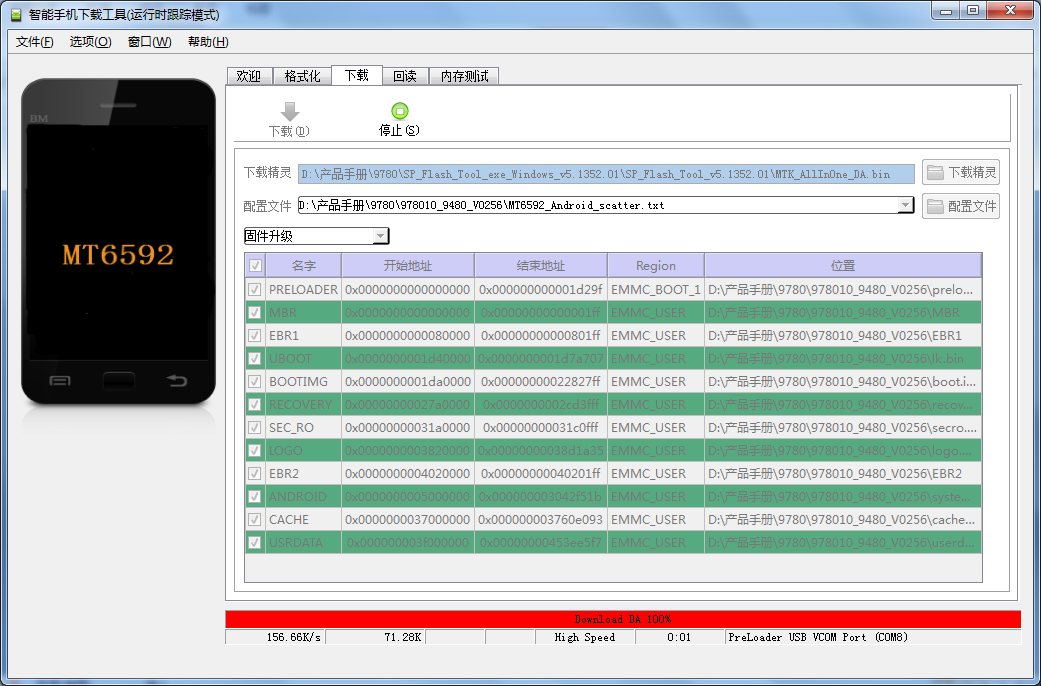
4) 在下拉选项中选择“固件升级”或“下载”，其他选项不要选



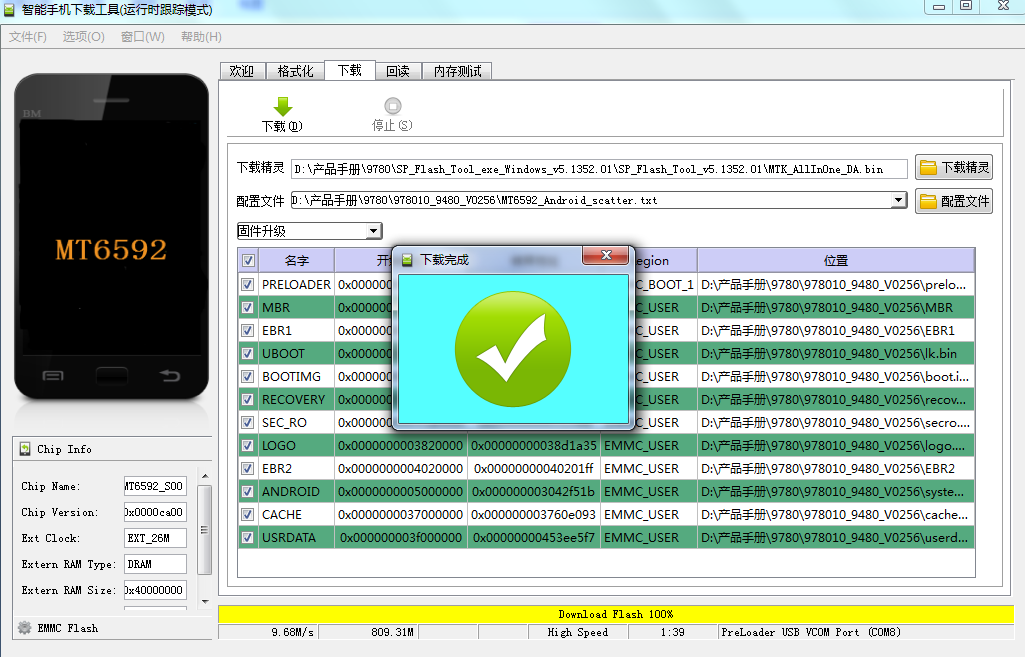
5) 点击下载，手机，关机状态，插上数据线，（必要时可以按音量键，个别手机不用电池也可）



6) 出现红色滚动条时说明连接成功



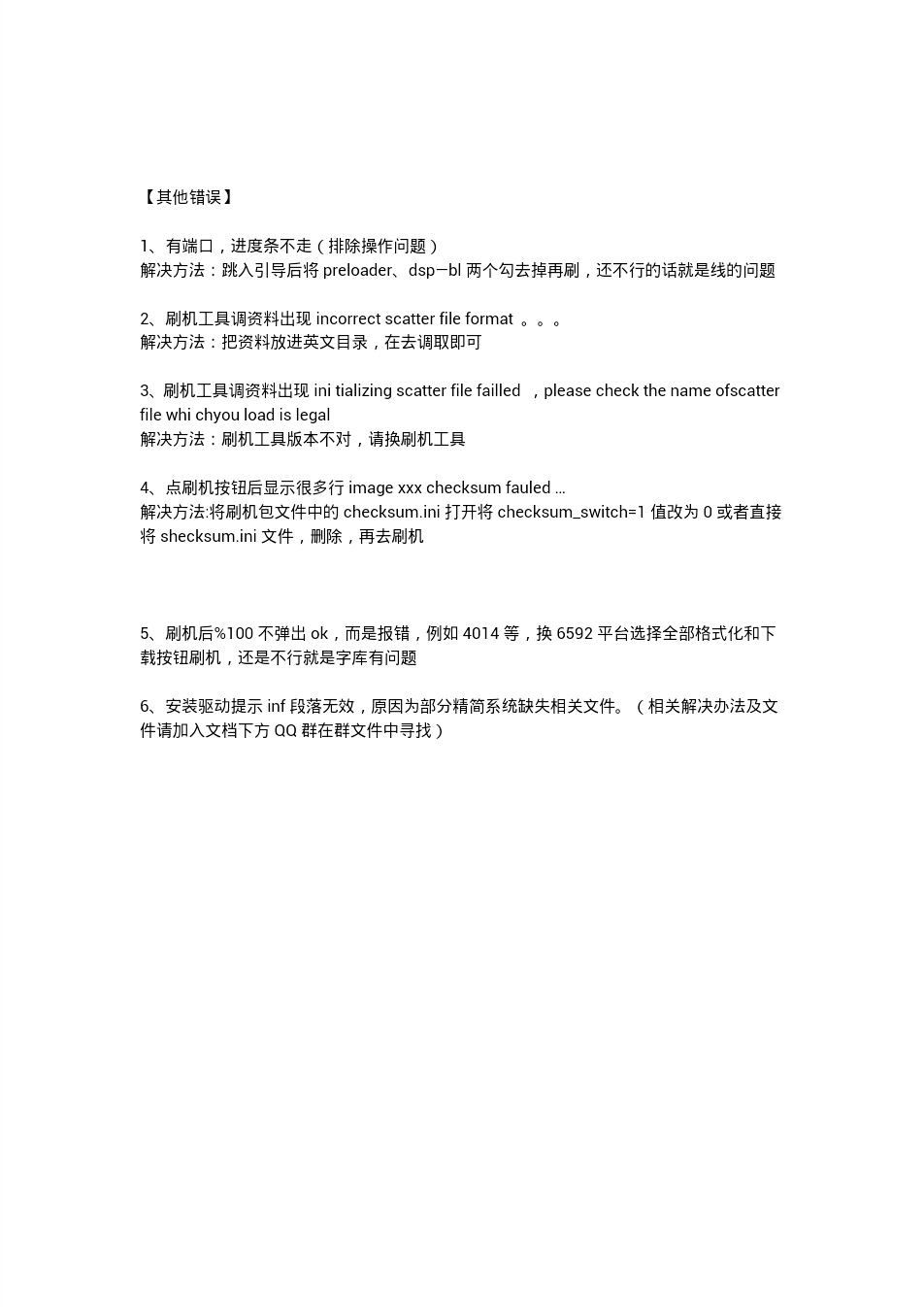
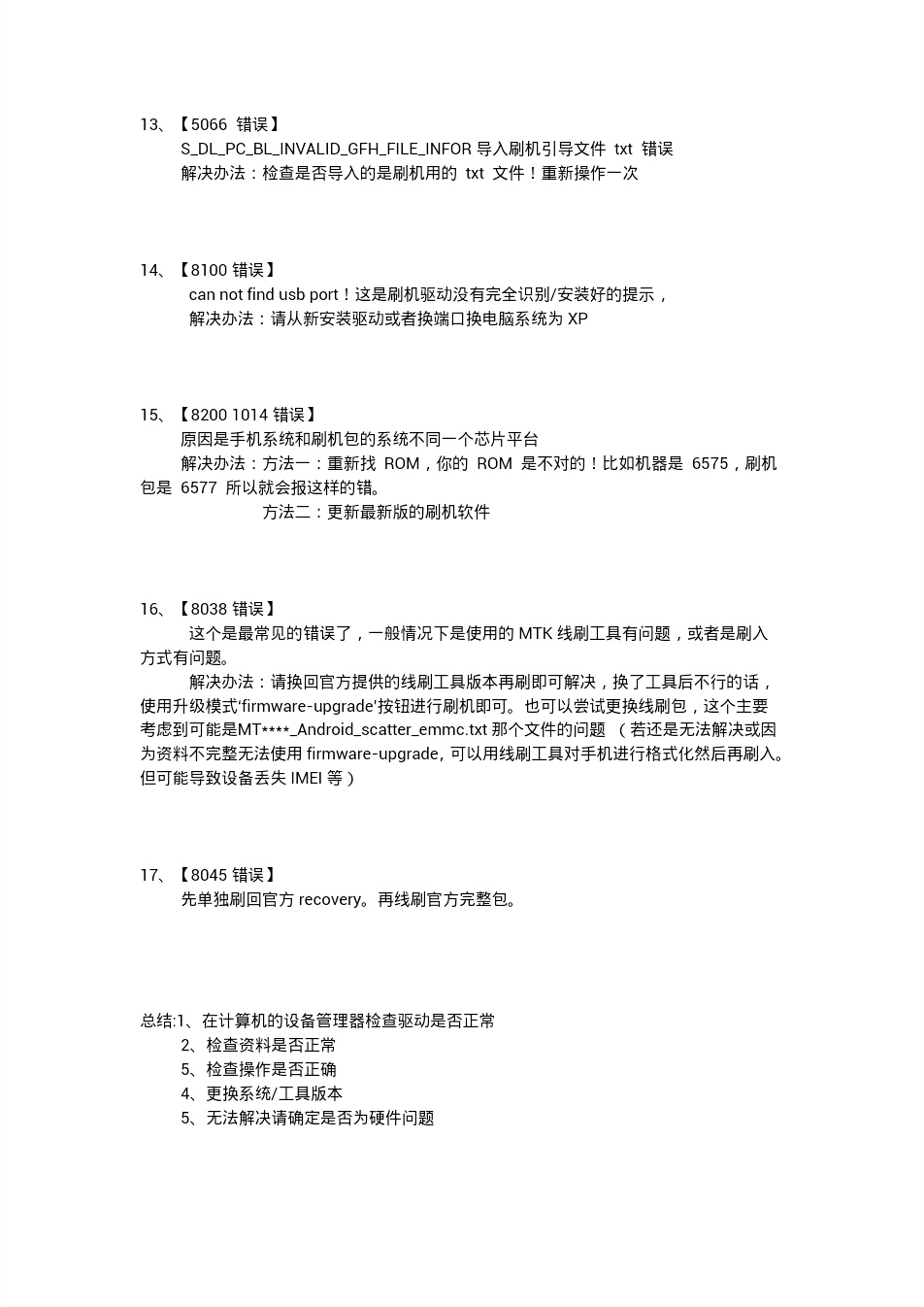
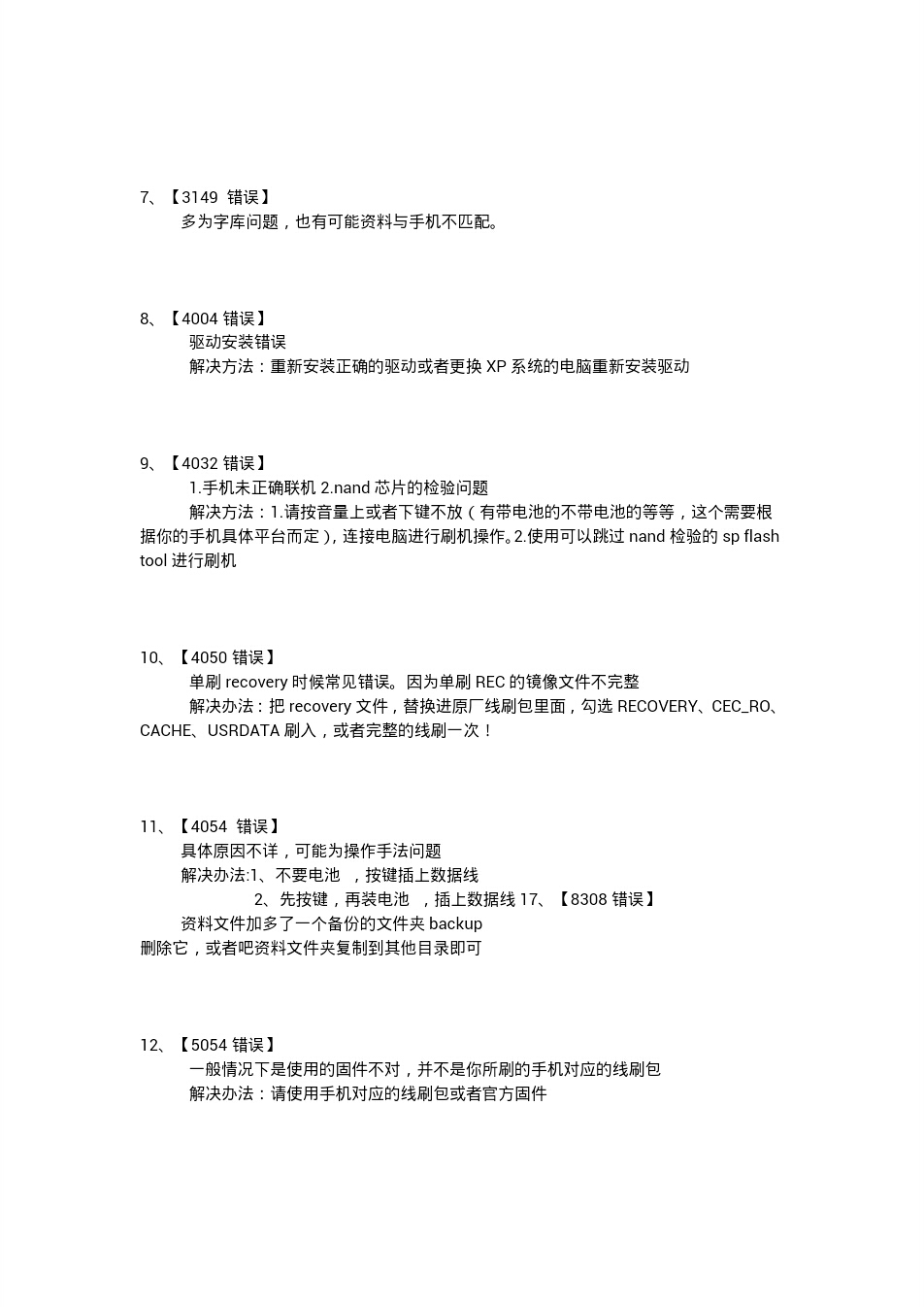
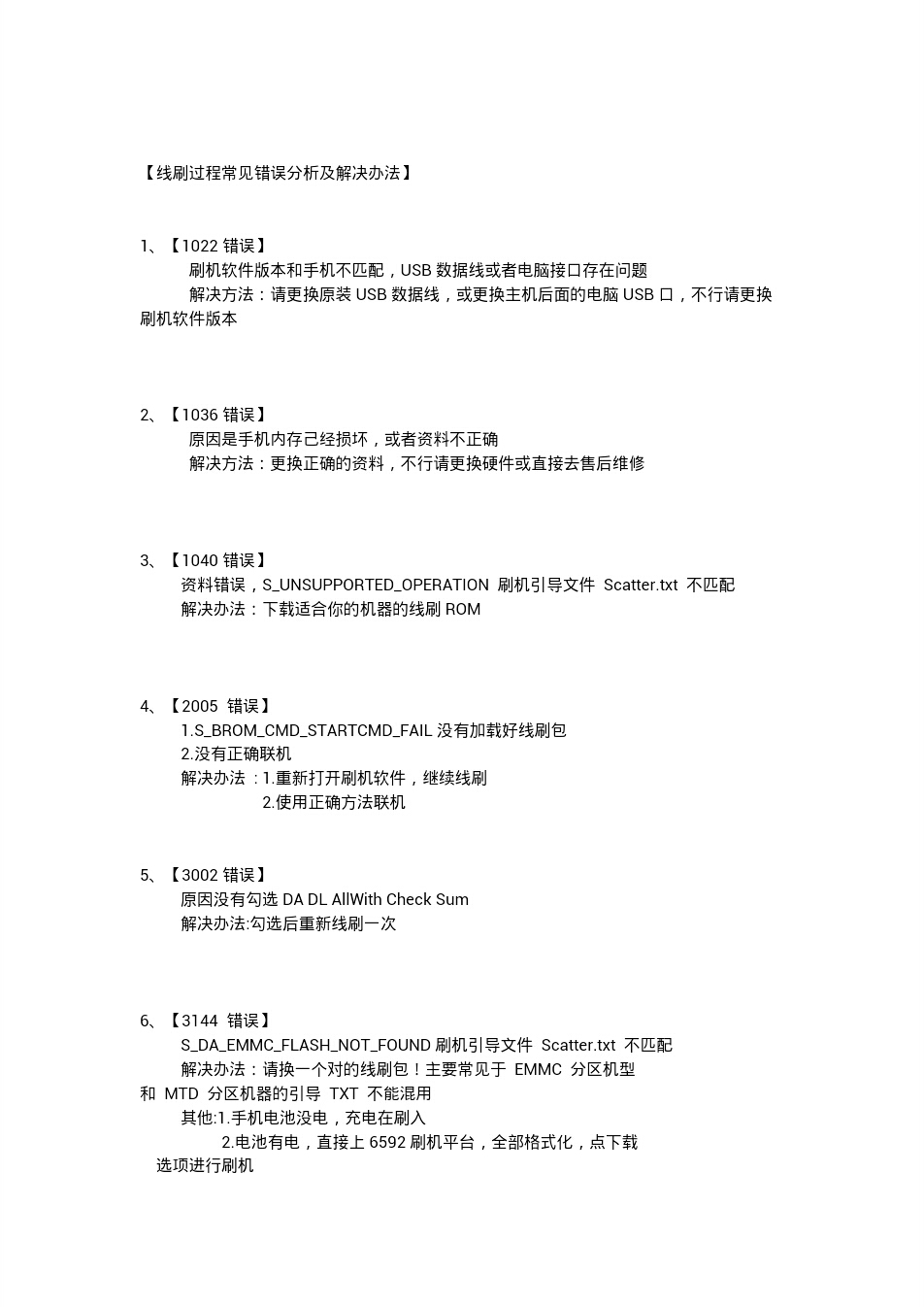
7) 当出现黄色滚动条开始走，并且达到100时即升级完成，升级过程中耐心等待，中途不要断电。



8) 拔数据线，重新插拔下电池，开机。首次开机需要更新系统，耐心等待3到5分钟，中途不要动手机，以防断电。

9) 手机开机，测试没问题时，关机重启，再正常使用。

**常见错误及解决办法：**



## 4. 设备连接

telnet 192.168.100.100

用户: root(无密码)

## 系统的调试

Push 文件及登录Android系统

1，本地机器建立ftp服务器

sudo apt-get install vsftpd

2，登陆到ssos系统

telnet 192.168.100.100 (用户名root,无密码)

3，使用ssos自带的工具ftpget将本地文件下载到ssos系统中

ftpget -u username -p password 192.168.100.101 filename（主目录下）

4，mount Android文件系统(启动lotusd之前，如果已经启动了，请重启手机)

mkdir android1

mount -o remount rw /

mount /dev/byname/system\_ins1 android1 (在/dev/byname/目录下xxx1对应外域， xxx2对应内域)

cp filename android1目录下，修改权限

umount android1

5，启动lotusd，另起一个终端登录到ssos，输入：pstree –p | grep init

可以看到打印内容类似于：

lotusd(1580)-+-lxc-start(1581)---init(1586)-+-adbd...

`-lxc-start(1582)---init(1587)-+-adbd...

运行attachns 1586 /system/bin/sh 可以进入Android的shell

如果无法进入Android shell，重启手机，重新执行步骤5

双Android手动切换

下载ssos最新的代码，编译，将 out/recovery/bin/nstool拷贝到主目录下，运行ftpget下载到ssos系统中，修改权限chmod a+x nstool

nstool -h 查看使用说明

nstool active pid 切换相应的Android到前台

例如：nstool active 1586

nstool active 1587

单独起内域或外域(/usr/etc/lotus)：

firefly-android.conf 对应外域

lite-android.conf 对应内域

登录到ssos后，mount /usr 为读写权限

mount -o remount rw /usr

cd /usr/etc/lotus

将conf文件移动到/data备份

启动内域：mv firefly-android.conf /data

启动外域：mv lite-android.conf /data

执行lotusd命令

使用fastboot 烧写image：

使用最新的双系统版本（如果使用老版本，可以先烧写最新ssos系统编译出的lk.bin，先将lk.bin下载到ssos，输入命令：dd if=lk.bin of=/dev/byname/lk）

登录到ssos系统：

toolbox reboot bootloader

fastboot 烧写：

fastboot flash <partition name> <image name>

分区和image对应关系：

system\_ins1(2) out/target/product/aus6735\_66c\_c\_m/system.img

userdata\_ins1(2) out/target/product/aus6735\_66c\_c\_m/userdata.img

root\_ins1(2) out/target/product/aus6735\_66c\_c\_m/root.img

如何生成root.img, 在bootimage编译完成之后可使用下面命令 ：

sudo ./build/tools/mkramfs.sh out/target/product/aus6735\_66c\_c\_m/

Fastboot使用：

out/host/linux-x86/bin/fastboot flash system\_ins1 out/target/product/aus6735\_66c\_c\_m/system.img

由于Android5.x及以上版本默认开启了dm-verity机制，所以system分区无法remount，需要先disable dm-verify, 具体方法如下:

确保当前版本问userdebug版本

adb disable-verity

adb reboot

adb root

adb remount /system