

君正®

## M200 Android4.4 开发指南

---

Date: Jan. 2015



北京君正集成电路股份有限公司  
Ingenic Semiconductor Co., Ltd.

君正®

**M200 Android4.4开发指南**

Copyright © Ingenic Semiconductor Co. Ltd 2015. All rights reserved.

**Release history**

| <b>Date</b> | <b>Revision</b> | <b>Change</b> |
|-------------|-----------------|---------------|
| Jan. 2015   | 1.0             | First release |

**Disclaimer**

This documentation is provided for use with Ingenic products. No license to Ingenic property rights is granted. Ingenic assumes no liability, provides no warranty either expressed or implied relating to the usage, or intellectual property right infringement except as provided for by Ingenic Terms and Conditions of Sale.

Ingenic products are not designed for and should not be used in any medical or life sustaining or supporting equipment.

All information in this document should be treated as preliminary. Ingenic may make changes to this document without notice. Anyone relying on this documentation should contact Ingenic for the current documentation and errata.

北京君正集成电路股份有限公司

地址：北京市海淀区西北旺东路 10 号院东区 14 号楼 君正大厦

电话:(86-10)56345000

传真:(86-10)56345001

<http://www.ingenic.com/>

## 目录

|         |   |    |
|---------|---|----|
| 1       | 概述 .....  | 1  |
| 2       | 搭建开发环境.....   | 3  |
| 2.1     | 准备 Ubuntu 12.04_64 位系统的基础环境 .....                   | 3  |
| 2.2     | 安装 JDK .....  | 3  |
| 2.3     | 交叉编译工具链.....  | 5  |
| 2.4     | 下载源码.....   | 5  |
| 3       | 板级配置 .....  | 7  |
| 3.1     | 参考设备 .....  | 7  |
| 3.2     | 配置 Bootloader.....                                  | 7  |
| 3.3     | 板级 kernel 配置 .....                                  | 7  |
| 3.3.1   | 内核 config 文件简介 .....                                | 7  |
| 3.3.2   | 板级文件 .....  | 7  |
| 3.4     | Android system 的板级配置 .....                          | 8  |
| 3.4.1   | 板级配置文件简介.....                                       | 8  |
| 4       | 编译 .....  | 11 |
| 4.1     | 代码目录说明 .....  | 11 |
| 4.2     | 编译 Android BSP .....                                | 11 |
| 4.3     | 编译 u-boot.....                                      | 12 |
| 4.4     | 单独编译 Kernel .....                                   | 12 |
| 4.5     | 单独编译 system .....                                   | 12 |
| 4.6     | 准备烧录文件 .....  | 12 |
| 5       | 辅助工具使用.....   | 13 |
| 5.1     | 串口调试.....   | 13 |
| 5.2     | ADB 的简介及使用方法.....                                   | 13 |
| 5.2.1   | Linux 主机 ADB 环境配置方法.....                            | 13 |
| 5.2.2   | Windows 下 ADB 环境配置方法 .....                          | 14 |
| 5.3     | USB 烧录工具 .....                                      | 15 |
| 5.3.1   | USBCloner 烧录工具使用说明.....                             | 15 |
| 5.3.1.1 | Windows 下首次安装驱动.....                                | 15 |
| 5.3.1.2 | 使用过君正 USBBurnTool 烧录工具（要替换成君正 USBCloner 烧录工具） ..... | 19 |
| 5.3.2   | 配置并烧录 .....   | 23 |
| 5.3.3   | Linux 下使用 USBCloner 烧录工具.....                       | 32 |
| 6       | 个性化定制 .....   | 33 |
| 6.1     | 开机动画.....   | 33 |
| 6.1.1   | 替换已经制作好的动画 .....                                    | 33 |
| 6.1.2   | 制作 bootanimation.zip .....                          | 33 |

---

|     |                    |    |
|-----|--------------------|----|
| 7   | 关注君正.....          | 35 |
| 7.1 | 新浪微博号：北京君正官微 ..... | 35 |
| 7.2 | 微信号：北京君正.....      | 35 |
| 7.3 | 北京君正官网 .....       | 36 |

# 1 概述

本文用于指导开发者开展基于君正 M200 开发板 Dorado 的 Android4.4 系统的基础开发。该文档主要介绍了以下几个方面的内容(阅读此文档时请注意红色字体提示):

- 1) 如何基于 64 位 PC 机搭建基础开发和编译环境;
- 2) 下载 Android 4.4 源码 git 仓库;
- 3) 板级配置;
- 4) 编译方法;
- 5) 辅助工具的使用, 如调试和烧录工具等;
- 6) 板级配置信息和文件简介;



## 2 搭建开发环境

Android 开发环境的搭建大同小异，基本上与 Android 官方网站介绍的一致，详细的 Android 官方编译环境建立方法，可参考 Android 的官方网站：<http://source.android.com/source/initializing.html>

下面是以 Ubuntu12.04\_64 位系统为例（其他 Ubuntu 版本环境搭建，可参考上面 Android 官方网站介绍），介绍君正开发环境搭建流程，与 Android 官网安装区别主要在于：

- 1) 同样 64 位的 Linux 系统，君正额外安装了兼容 32 位的相应插件和 gawk；
- 2) 君正提供的交叉编译工具；

### 2.1 准备 Ubuntu 12.04\_64 位系统的基础环境

安装所需包：

```
$ sudo apt-get install cpp-4.6 g++-4.6 gcc-4.6 gcc-4.6-multilib gcc g++ cpp  
gcc-multilib g++-4.6-multilib git-core git gnupg flex bison gperf  
build-essential zip curl libc6-dev libncurses5-dev:i386 x11proto-core-dev  
libx11-dev:i386 libreadline6-dev:i386 libgll-mesa-glx:i386 libgll-mesa-dev  
g++-multilib mingw32 tofrodos python-markdown libxml2-utils xsltproc  
zlib1g-dev:i386 ia32-libs gawk qt4-dev-tools libgll-mesa-dri:i386  
libglapi-mesa:i386 libncurses5-dev libqt3-mt-dev u-boot-tools;
```

### 2.2 安装 JDK

默认的 Ubuntu 环境中预装的 JAVA 环境是 openjdk，而 Android 编译必须用 oracle(sun)的 JAVA。请将 Ubuntu 上自带的 openjdk 先完全卸载，随后安装 JAVA 官网上下载的最新的对应于 64 位机器的 JDK 安装包，推荐安装 JDK 版本为 jdk-6u37-linux-x64，下载页面为：

<http://www.oracle.com/technetwork/java/javase/downloads/java-archive-downloads-javase6-419409.html>


















| Java SE Development Kit 6u37  |           |  |
|---|-----------|--|
| You must accept the Oracle Binary Code License Agreement for Java SE to download this software.           |           |  |
| <input type="radio"/> Accept License Agreement <input checked="" type="radio"/> Decline License Agreement |           |  |
| Product / File Description  | File Size | Download   |
| Linux x86   | 65.43 MB  |  <a href="#">jdk-6u37-linux-i586-rpm.bin</a>    |
| Linux x86   | 68.44 MB  |  <a href="#">jdk-6u37-linux-i586.bin</a>        |
| Linux x64   | 65.65 MB  |  <a href="#">jdk-6u37-linux-x64-rpm.bin</a>     |
| Linux x64   | 68.71 MB  |  <a href="#">jdk-6u37-linux-x64.bin</a>         |
| Solaris x86   | 68.35 MB  |  <a href="#">jdk-6u37-solaris-i586.sh</a>       |
| Solaris x86   | 119.94 MB |  <a href="#">jdk-6u37-solaris-i586.tar.Z</a>    |
| Solaris SPARC   | 73.36 MB  |  <a href="#">jdk-6u37-solaris-sparc.sh</a>      |
| Solaris SPARC   | 124.71 MB |  <a href="#">jdk-6u37-solaris-sparc.tar.Z</a>   |
| Solaris SPARC 64-bit  | 12.13 MB  |  <a href="#">jdk-6u37-solaris-sparcv9.sh</a>    |
| Solaris SPARC 64-bit  | 15.42 MB  |  <a href="#">jdk-6u37-solaris-sparcv9.tar.Z</a> |
| Solaris x64   | 8.45 MB   |  <a href="#">jdk-6u37-solaris-x64.sh</a>       |
| Solaris x64   | 12.18 MB  |  <a href="#">jdk-6u37-solaris-x64.tar.Z</a>   |
| Windows x86   | 69.72 MB  |  <a href="#">jdk-6u37-windows-i586.exe</a>    |
| Windows x64   | 59.73 MB  |  <a href="#">jdk-6u37-windows-x64.exe</a>     |
| Linux Intel Itanium   | 53.95 MB  |  <a href="#">jdk-6u37-linux-ia64-rpm.bin</a>  |
| Linux Intel Itanium   | 60.67 MB  |  <a href="#">jdk-6u37-linux-ia64.bin</a>      |
| Windows Intel Itanium   | 57.89 MB  |  <a href="#">jdk-6u37-windows-ia64.exe</a>    |
| <a href="#">Back to top</a>   |           |  |

图 2-1 Java JDK 下载页面

选择下载jdk-6u37-linux-x64.bin JDK安装步骤:

- a) 拷贝jdk-6u37-linux-x64.bin到/usr/java/目录中;
- b) 修改可执行权限: `$ sudo chmod u+x /usr/java/jdk-6u37-linux-x64.bin`
- c) 执行下面命令开始安装: `$ sudo /usr/java/jdk-6u37-linux-x64.bin`
- d) 安装完毕, 配置系统环境变量。在这里介绍两种方法:

方法1 使用超级用户权限在/etc/environment中增加:

```
PATH="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/java/jdk1.6.0_37/bin"
JAVA_HOME=/usr/java/jdk1.6.0_37
JRE_HOME=/usr/java/jdk1.6.0_37/jre
CLASSPATH=$CLASSPATH:$JAVA_HOME/lib:$JRE_HOME/lib
```

保存后退出。重新登录用户, 环境变量会自动生效。



方法2. 使用超级用户权限在/etc/profile中增加:

```
export JAVA_HOME=/usr/java/jdk1.6.0_37 export
JRE_HOME=/usr/java/jdk1.6.0_37/jre export
CLASSPATH=$CLASSPATH:$JAVA_HOME/lib:$JRE_HOME/lib export
PATH=$JAVA_HOME/bin:$JAVA_HOME/jre/bin:$PATH
```

保存后退出。重新登录用户，环境变量会自动生效。

#### e) 查看Java版本

重新登陆后，在任一目录下执行下面的命令查看Java版本，显示下图所示信息，说明已经配置成功。

```
root@Ubuntu-1204:~# java -version
java version "1.6.0_37"
Java(TM) SE Runtime Environment (build 1.6.0_37-b06)
Java HotSpot(TM) 64-Bit Server VM (build 20.12-b01, mixed mode)
root@Ubuntu-1204:~#
```

图 2-2 Java JDK 版本信息

## 2.3 交叉编译工具链

当标准的 android 编译环境准备好之后，还需要相应的交叉编译工具链。君正的交叉编译工具链已经包含在 Android 源码中，具体位置：`prebuilts/gcc/linux-x86/mips/mipsel-linux-android-4.7`。编译前，则需要在 Android 目录下执行以下命令：

```
$ ./build/envsetup.sh 或者 source ./build/envsetup.sh
$ lunch dorado-userdebug (lunch 之后默认交叉工具链配置完成)
```

## 2.4 下载源码

君正开放了完整的 Android4.4 源代码 git 库，并使用 repo 进行管理。

1. 下载 repo 脚本；
 

```
$ mkdir android-ingenic
$ cd android-ingenic
$ wget http://git.ingenic.cn:8082/bj/repo
$ chmod +x repo
```
2. 下载 Android 4.4 源码；
 

```
$ ./repo init -u http://git.ingenic.cn:8082/gerrit/AOSP/platform/manif
est -b android-4.4.4_r2-ingenic
$ ./repo sync
```
3. 同步到 tag

## 搭建开发环境

```
$ ./repo forall -c "git reset --hard ingenic-android4.4.4_r2-kernel3.10.14-dorado-vx.x-yyyyymmdd"
```

“ingenic-android4.4.4\_r2-kernel3.10.14-dorado-vx.x-yyyyymmdd”代表君正 Android 源码的 tag，二种方法获取君正最新 tag:

- a) 进入已下载源码中的 kernel 目录，运行“git tag | grep dorado”命令进行查看，查找与“ingenic-android4.4.4\_r2-kernel3.10.14-dorado”相关的 tag。
- b) 关注君正“研发动态”信息发布，网址: <http://www.ingenic.com/?news/tp/228.html>。

## 4. 下载出错:

在下载时，有时会出现一些错误，请大家按照如下方法检查:

- a) 网络是否连接? ping git.ingenic.cn
- b) 是否成功安装了 git?
- c) 使用 nslookup git.ingenic.cn 是否成功解析? 如果不能解析，请尝试使用 ip 地址 60.173.247.205 进行下载，repo 脚本的地址也要手动修改。
- d) 重新创建目录，下载 repo，下载一份干净的代码。
- e) 重新执行，repo 支持断点续传。(适用于下载过程中因网络不稳定而出现下载错误的客户)  
\$ ./repo sync
- f) 其他问题请反馈给 [support@ingenic.com](mailto:support@ingenic.com)

## 3 板级配置

进入 Android 工程目录：执行

```
$ ls
Makefile      art           bootable     cts          developers
device        external     hardware     libcore     prebuilts
ndk           pdk          result       system      sdk
abi           bionic       build        dalvik      development
frameworks   kernel-3.10.14  libnativehelper  packages    tools
docs
```

### 3.1 参考设备

DORADO\_V2.2, M200 参考开发板，屏幕分辨率 480x800（竖屏，型号：BYD8991）。

### 3.2 配置 Bootloader

Bootloader 的源码位于 Android 工程的 bootable/bootloader/uboot,

|                            |  |
|----------------------------|--|
| ./Makefile                 | 编译信息和配置相关信息，例如：<br>dorado_v22_android_msc0_config                |
| ./include/configs/dorado.h | 板级设置信息：memory, nandflash, CPU frequency,<br>UART, binary, GPIO 等 |

### 3.3 板级 kernel 配置

Linux kernel 的源码位于 Android 工程项目的 kernel-3.10.14 目录下。

#### 3.3.1 内核 config 文件简介

在 kernel-3.10.14/arch/mips/configs/目录下有 Dorado 的配置文件，请按照以下说明进行配置：

对于 BYD8991（竖屏，型号 480X800）：

```
dorado_v22_android_defconfig;
```

#### 3.3.2 板级文件

1) kernel-3.10.14/arch/mips/xburst/soc-m200/board/dorado/dorado\_v22/目录下，包含板级硬件的设置信息。

board.h: 板级信息头文件，GPIO 定义等

pm.c: 休眠时 GPIO 状态处理，以降低板级功耗；

pmu.h: 电源管理, 有效降低板级功耗;

2) kernel-3.10.14/arch/mips/xburst/soc-m200/board/dorado/common/目录下, 包含板级模块硬件的设置信息

bcm\_power\_control.c: 不同状态下电量控制信息;

board-power-619.c: 电压、电流配置;

board\_base.c: 板级信息;

board\_base.h: 板级信息头文件, GPIO 定义等;

camera.c: Camera 的配置信息;

i2c\_bus.c: 触摸屏配置信息;

sound.c: 耳机/Volume 配置信息;

misc.c: 按键, I2C 设备, 电池, MAC 等配置信息, 及注册各种总线设备;

mmc.c: MMC 接口配置信息;

regulator.c: 电源配置信息, 控制外设上电, 掉电;

## 3.4 Android system 的板级配置

### 3.4.1 板级配置文件简介

以 device/ingenic/dorado 为例。该目录文件如下:

vendorsetup.sh: 将产品信息加入 lunch 列表;

Android.mk: 负责将目录下的其它 makefile 根据\$(TARGET\_DEVICE)包含进编译范围;

AndroidProducts.mk: 将 dorado.mk 加入编译范围;

device.mk: 包含了产品的配置信息;

dorado.mk: 产品的相关定义;

bluetooth/bdroid\_buildcfg.h: 定义了设备在蓝牙中的默认名字。

config/: 一系列的配置文件。

dhcpcd.conf --- DHCP 的配置文件。

fstab.board --- 系统启动时挂载分区的配置文件。

init.board.usb.rc --- 板级的 USB 配置信息。

init.recovery.board.rc --- recovery 使用的配置信息。

wpa\_supplicant.conf --- WPA 程序的配置文件。

audio\_policy.conf --- 音频设备使用的配置文件。

gpio-keys.kl --- 按键的键值定义。

media\_codecs.xml --- 系统中 codec 的注册配置文件。

bt\_addr --- 定义蓝牙地址。

excluded-input-devices.xml --- 注册目前系统中的输入设备的配置文件。

init.board.rc --- 板级的启动配置文件。

media\_profiles.xml --- 多媒体相关的配置文件, 包括 Camera 的配置信息。

ueventd.board.rc --- ueventd 使用的配置文件。

---

overlay/: 存放该设备需要替换或者需要增加的，用来定制的产品需要的配置。

frameworks/base/core/res/res/values/config.xml

frameworks/base/packages/SettingsProvider/res/values/defaults.xml

frameworks/base/packages/SystemUI/res/values/config.xml

recovery/

recovery\_ui.cpp --- 界面相关的自定义程序。

recovery\_updater.c --- 在这里注册自定义命令。

board-info.txt : 文本文件。

Kernel: kernel 的二进制文件 (zImage)。

recovery.fstab: recovery 用的挂载各个分区的信息。



## 4 编译

### 4.1 代码目录说明

代码下载成功后，会得到以下的一个目录结构：

```
$ ls
Makefile      art          bootable    cts          developers
device        external    hardware    libcore      prebuilts
ndk           pdk         result      system       sdk
abi           bionic      build       dalvik        development
frameworks    tools       kernel-3.10.14 docs          packages
libnativehelper
```

其中：

- 1) Bootloader 的源码，位于：/bootable/bootloader/uboot
- 2) Android 的 linux 内核源码，位于：kernel-3.10.14
- 3) 其余为完整 android 的源码。

### 4.2 编译 Android BSP

进入到工程根目录，执行编译命令：

```
$ source build/envsetup.sh
$ lunch dorado-userdebug
$ make
```

整个编译过程需要持续几个小时，具体的时间取决于主机的性能情况。

编译完成后，在 `out/target/product/dorado/` 目录下生成 BSP 文件：`boot.img`，`system.img`。

**请注意：**`boot.img` 是由 `device/ingenic/dorado/kernel` 文件直接生成的，该 `kernel` 文件有可能与 `kernel` 目录下最新的代码不一致，需要再单独重新编译 `kernel` 文件，编译命令如下：

```
$ cd kernel-3.10.14
$ make dorado_v22_android_defconfig
$ make zImage
$ cp arch/mips/boot/compressed/zImage ../device/ingenic/dorado/kernel
$ cd ..
$ make bootimage
```

也可参考后面章节“4.4 单独编译 Kernel”。`system.img` 没有此问题。

### 4.3 编译 u-boot

Bootloader 需要单独编译。

```
$ source build/envsetup.sh
$ lunch dorado-userdebug
$ cd bootable/bootloader/uboot/
$ make clean; make dorado_v22_android_msc0_config; make
```

编译完成后，会在当前目录下生成 `u-boot-with-spl-mbr-gpt.bin` 文件。

### 4.4 单独编译 Kernel

Android 默认编译的 Kernel 是 `device/ingenic/dorado/kernel` 目录下提交的内核，该文件可能不够新或者不满足用户要求，若要生成最新的内核，请通过以下命令编译。

```
$ source build/envsetup.sh
$ lunch dorado-userdebug
$ cd kernel-3.10.14
$ make dorado_v22_android_defconfig
$ make zImage
$ cp arch/mips/boot/compressed/zImage ../device/ingenic/dorado/kernel
$ make bootimage
```

生成 `out/target/product/dorado/boot.img`

### 4.5 单独编译 system

如果想单独生成 `system.img`，请在 Android 根目录下执行

```
$ source build/envsetup.sh
$ lunch dorado-userdebug
$ make systemimage
```

### 4.6 准备烧录文件

所有的文件编译成功后，就可以进行烧录(参考后续章节“辅助工具使用”的相关内容)。如果此时不想编译，也可以在官网下载已经编译好的 BSP 进行烧录。

- `u-boot-with-spl-mbr-gpt.bin`
  - `boot.img`
  - `system.img`



## 5 辅助工具使用

### 5.1 串口调试

君正 dorado v22 开发板，调试底层的 Bootloader 或 kernel 时，可以使用串口进行打印调试，串口波特率：57600 8N1

注意：需要一个交叉串口线连接到 PC：即板子的 RXD，是 PC 的 TXD，板子的 TXD，是 PC 的 RXD。

### 5.2 ADB 的简介及使用方法

#### 5.2.1 Linux 主机 ADB 环境配置方法

1) 添加/etc/udev/rules.d/51-android.rules文件

对于Linux用户，需要添加/etc/udev/rules.d/51-android.rules文件，内容如下：

```
SUBSYSTEM=="usb|usb_device", ATTR{idVendor}=="18d1", MODE="0666",
GROUP="plugdev"
```

2) 下载SDK

从Android开发网站选择合适的Linux版本的SDK并下载，SDK的网址为：

<http://developer.android.com/sdk/index.html>

| ADT Bundle      |  |                 |                                  |
|-----------------|--|-----------------|----------------------------------|
| Platform        | Package  | Size            | MD5 Checksum                     |
| Windows 32-bit  | <a href="#">adt-bundle-windows-x86-20140321.zip</a>    | 535085536 bytes | b61495a6bf591cc374c31bce4fc46ec0 |
| Windows 64-bit  | <a href="#">adt-bundle-windows-x86_64-20140321.zip</a> | 535287324 bytes | a6f4699bbdc5a29b371ed60610535651 |
| Mac OS X 64-bit | <a href="#">adt-bundle-mac-x86_64-20140321.zip</a>     | 501955296 bytes | 4a08649cea9b098cdf7349f452294014 |
| Linux 32-bit    | <a href="#">adt-bundle-linux-x86-20140321.zip</a>      | 527971926 bytes | 943ae4d28fe7c79108c8bf2aafd5e6d2 |
| Linux 64-bit    | <a href="#">adt-bundle-linux-x86_64-20140321.zip</a>   | 528187678 bytes | f2a2153b5c7dbaeb86b550bf4f770c36 |

图 5-1 Android Linux SDK 下载页面

例如，对于64位的Ubuntu系统，选择下载adt-bundle-linux-x86\_64-20140321.zip

3) 安装SDK 将下载的SDK解压安装到任意目录下。然后修改~/.bash\_profile文件, 设置运行环境指向sdk的platform-tools/目录。具体是打开~/.bash\_profile文件(如果没有此文件也可以自行添加), 在里面加入一行: `export PATH=${PATH}:<ANDROID_SDK_INSTALL_PATH>/platform-tools` 设置好以后, 就可以在任意目录下使用adb命令了。

## 5.2.2 Windows 下 ADB 环境配置方法

1) 下载SDK 从Android开发网站选择合适的Windows版本的SDK并下载, SDK的网址为:

<http://developer.android.com/sdk/index.html>

| ADT Bundle      |  |                 |                                  |
|-----------------|--|-----------------|----------------------------------|
| Platform        | Package  | Size            | MD5 Checksum                     |
| Windows 32-bit  | <a href="#">adt-bundle-windows-x86-20140321.zip</a>    | 535085536 bytes | b61495a6bf591cc374c31bce4fc46ec0 |
| Windows 64-bit  | <a href="#">adt-bundle-windows-x86_64-20140321.zip</a> | 535287324 bytes | a6f4699bbdc5a29b371ed60610535651 |
| Mac OS X 64-bit | <a href="#">adt-bundle-mac-x86_64-20140321.zip</a>     | 501955296 bytes | 4a08649cea9b098cdf7349f452294014 |
| Linux 32-bit    | <a href="#">adt-bundle-linux-x86-20140321.zip</a>      | 527971926 bytes | 943ae4d28fe7c79108c8bf2aafd5e6d2 |
| Linux 64-bit    | <a href="#">adt-bundle-linux-x86_64-20140321.zip</a>   | 528187678 bytes | f2a2153b5c7dbaeb86b550bf1f770c36 |

图 5-2 Android Windows SDK 下载页面

例如, 对于Windows 32位系统, 选择下载adt-bundle-windows-x86-20140321.zip, 解压到目录。

2) 设置运行ADB的环境变量 以Windows7为例, 右击“计算机”, 选择“属性”, 点击“高级系统设置”弹出“系统属性”对话框, 选择“高级”, 点击“环境变量”弹出对话框, 在“系统变量(S)”中找到“Path”变量, 然后“编辑”, 将adb所在目录的路径加到“变量值”结尾处, 前面用“;”隔开。

3) 查看ADB版本 运行cmd, 可以用adb version命令查看adb的版本, 若显示下图所示信息, 说明adb已经配置成功。

```
C:\Users\Administrator>adb version
Android Debug Bridge version 1.0.31
```

图 5-3 查看 ADB 的版本

下载地址: <http://developer.android.com/sdk/win-usb.html>

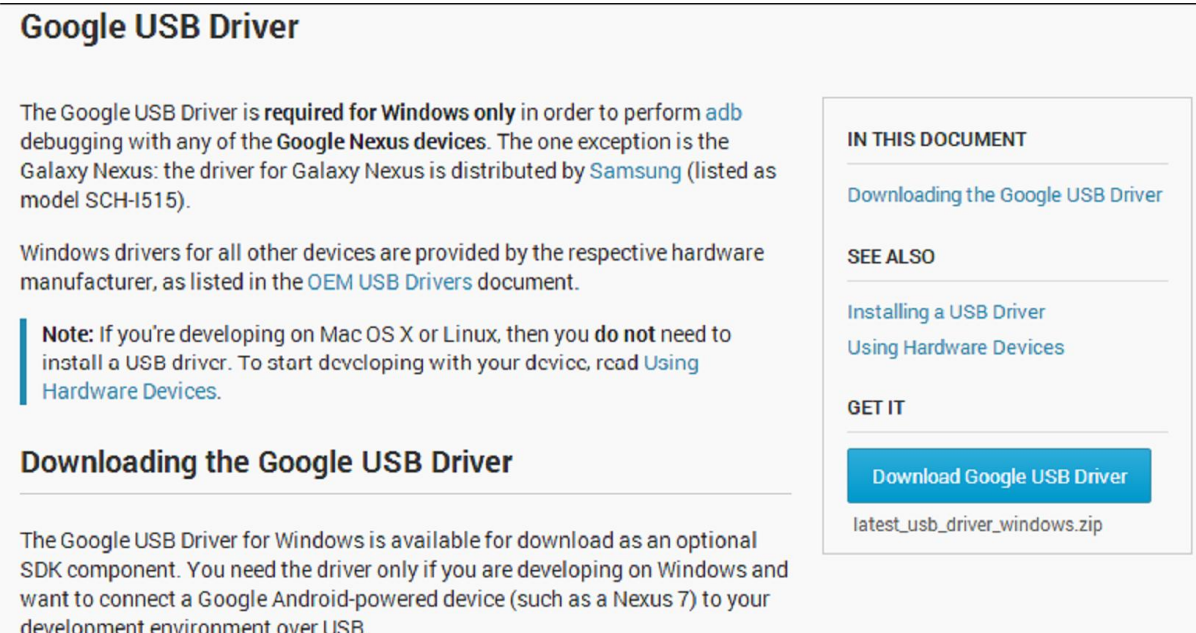


图 5-4 Google USB 驱动下载页面

点击“Download Google USB Driver”下载最新的驱动latest\_usb\_driver\_windows.zip

2) 安装ADB驱动 关于如何安装Windows环境下的ADB驱动，请参考Android官方网站：  
<http://developer.android.com/tools/extras/oem-usb.html#InstallingDriver>

### 5.3 USB 烧录工具

USB 烧录工具可运行在

- 1) Windows (32 位,64 位);
- 2) Ubuntu12.04 (32 位, 64 位);
- 3) Ubuntu14.04 (32 位, 64 位)。

操作系统，分别有对应压缩包，两种方法可获取烧录工具包，如下：

- 1) 官网：<http://www.ingenic.com/?product/id/2/lm/1.html> 对应的 tag 版本下载包里。
- 2) 君正 Android 代码根目录下的：  
device/ingenic/host-tools/BurnerBinary/ClonerAllVersions/0.23.17 目录下。

这里简单介绍 Windows 下烧录过程。

#### 5.3.1 USBCloner 烧录工具使用说明

##### 5.3.1.1 Windows 下首次安装驱动

使板子进入烧录模式，如图 5-0

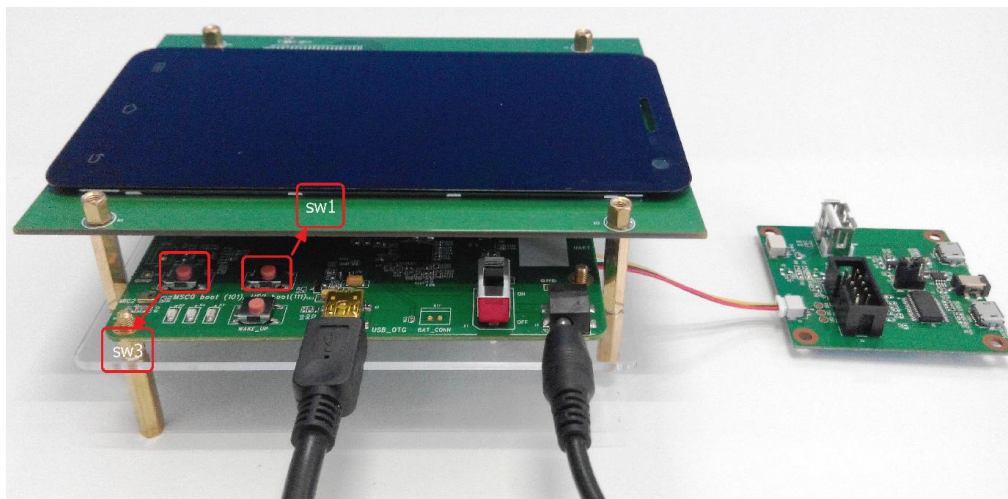


图 5-0

按着“SW1”键，按一下“SW3”键，这时板子如果连接到 PC 上，会弹出硬件安装向导，选择“从列表或指定位置安装”，单击“下一步”如下图 5-1 所示：



图 5-1

- 选择驱动文件。选择“不要搜索，我要自己选择要安装的驱动程序”，点击“下一步”，如图 5-2

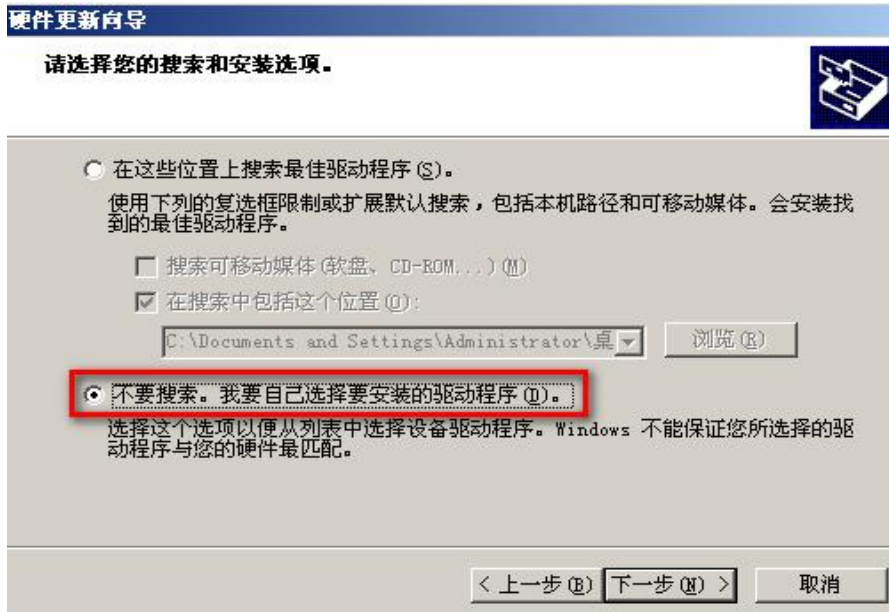


图 5-2

- 选择“从磁盘中安装”，单击“确定”，如图 5-3。



图 5-3

- 然后在“厂商文件复制来源”下选择“浏览”，然后选择要安装的驱动的 inf 文件，点击“确定退出”，如图所示 5-4：





图 5-4

- 点击“下一步”，直至“完成”为止，如图 5-5 所示。



图 5-5

### 5.3.1.2 使用过君正 USBBurnTool 烧录工具（要替换成君正 USBCloner 烧录工具）

如果用户以前使用过旧的烧录工具，那么在使用之前就需要手动更新驱动，需要右击“我的电脑”->“管理”->“设备管理器”，此时会弹出电脑所有的设备。选择“Ingenic Usb Boot Class”下的“Usb Boot Device”，右击鼠标，选择“更新驱动程序”显示如图 5-6:

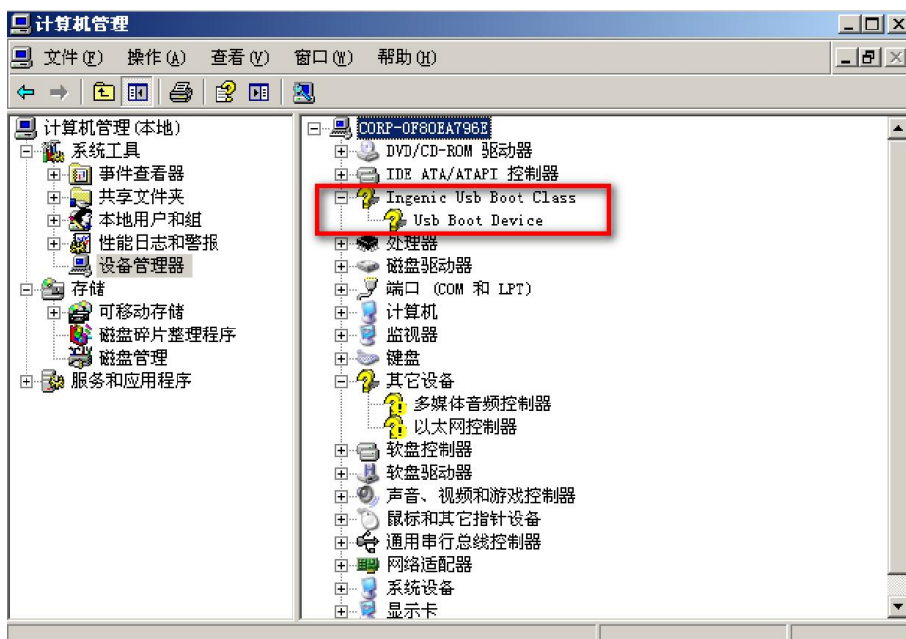


图 5-6

会弹出硬件安装向导，选则“从列表或指定位置安装”，单击“下一步”如下图 5-7 所示:

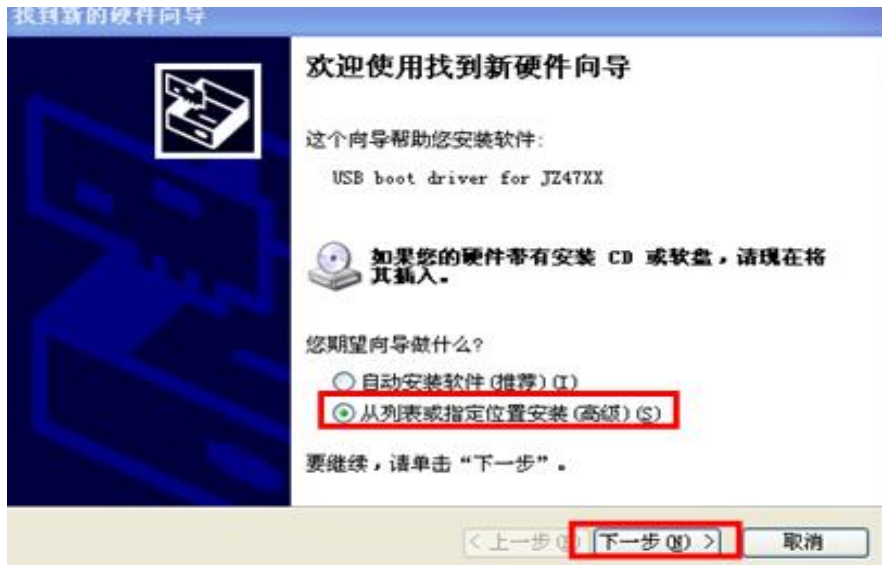


图 5-7

- 选择驱动文件。选择“不要搜索，我要自己选择要安装的驱动程序”，点击“下一步”，如图 5-8

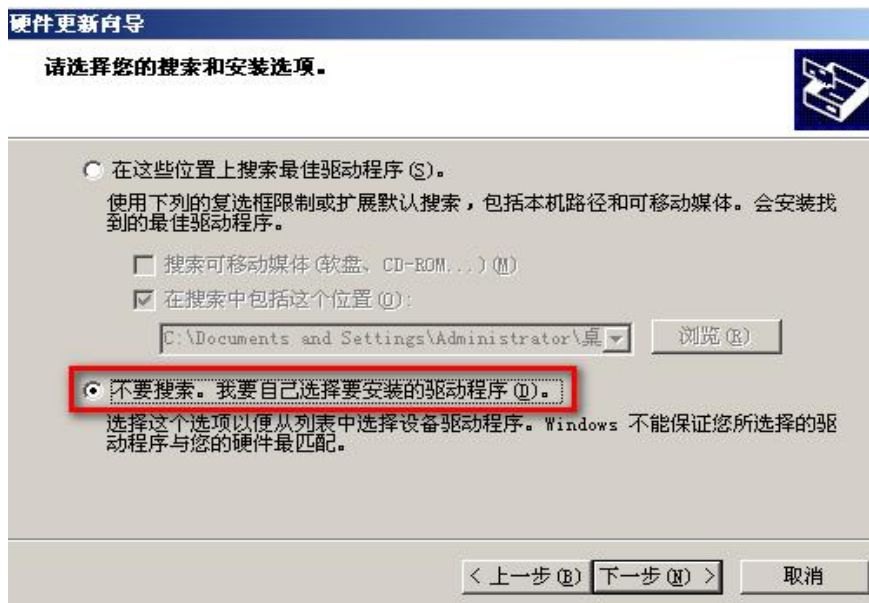


图 5-8

- 选择“从磁盘中安装”，单击“确定”，如图 5-9。





图 5-9

- 然后在“厂商文件复制来源”下选择“浏览”，然后选择要安装的驱动的 inf 文件，点击“确定退出”，如图所示 5-10：



图 5-10

- 点击“下一步”，直至“完成”为止，如图 5-11 所示。



图 5-11

- 安装成功后的设备管理器如图 5-12 所示。

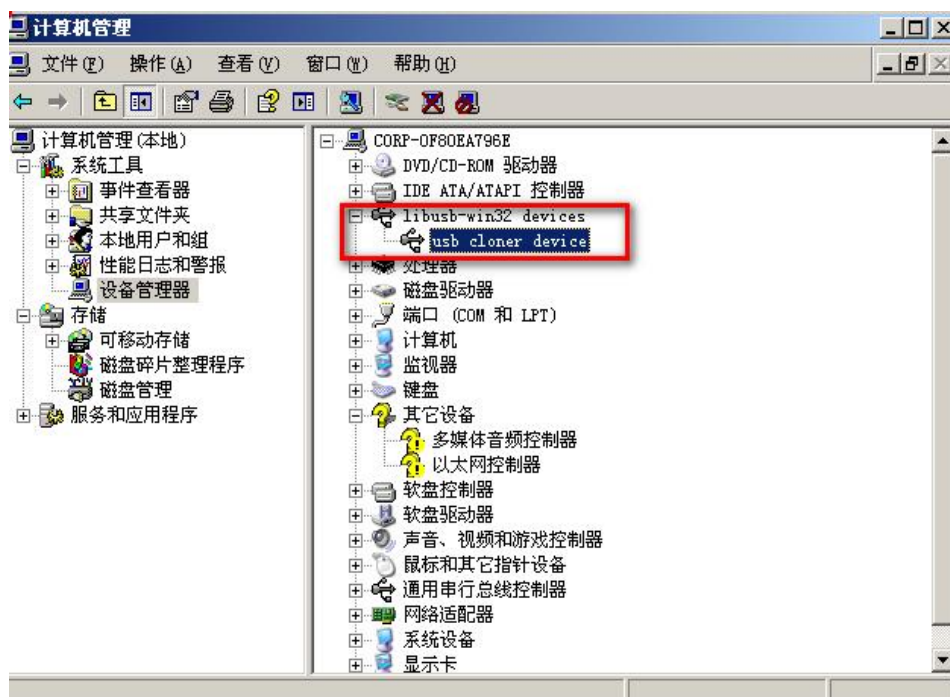


图 5-12

### 5.3.2 配置并烧录

打开 USBCloner 烧录工具以后，界面显示如图 5-13 所示。

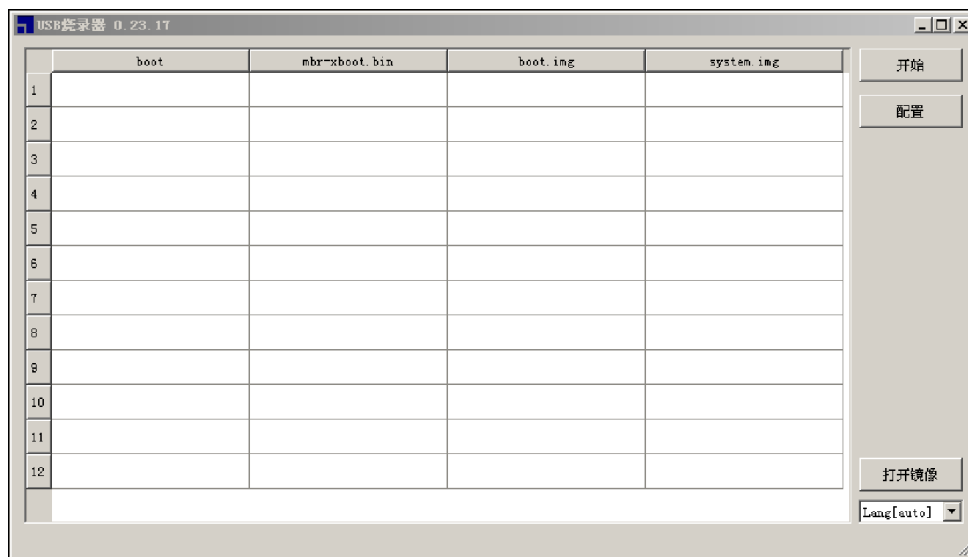


图 5-13

在主界面中，图 5-14 中红色标注所显示的是该 USBCloner 烧录工具的名称以及版本号；图 5-15 红色标注显示的是该烧录工具此次要烧录的文件列表；图 5-16 红色标注显示的烧录工具语言的选择；图 5-17 红色标注的区域表示的是要烧录的第一个设备的所有烧录文件进度条显示区域；图 5-18 红色标注的区域表示的是所有设备在 boot 阶段的进度条显示区域；图 5-19 中红色标注的区域所表示的是所有设备在烧录第一个文件时进度条显示区域；图 5-20 色标注的区域所表示的是所有设备在烧录第二个文件时进度条显示区域；图 5-21 中红色区域所表示的是所有设备在烧录第三个文件时进度条显示区域；图 5-22 中红色区域所表示的是烧录“开始”按钮，配置好烧录工具后，要先点击“开始”按钮，然后使板子进入烧录模式；图 5-23 中红色区域所表示的 USBCloner “配置”按钮。

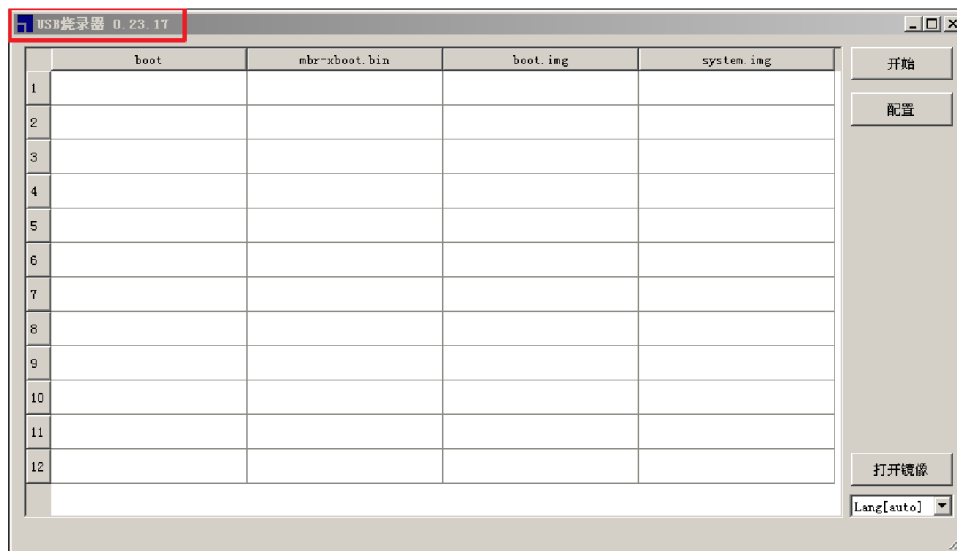


图 5-14 烧录工具名称及版本号

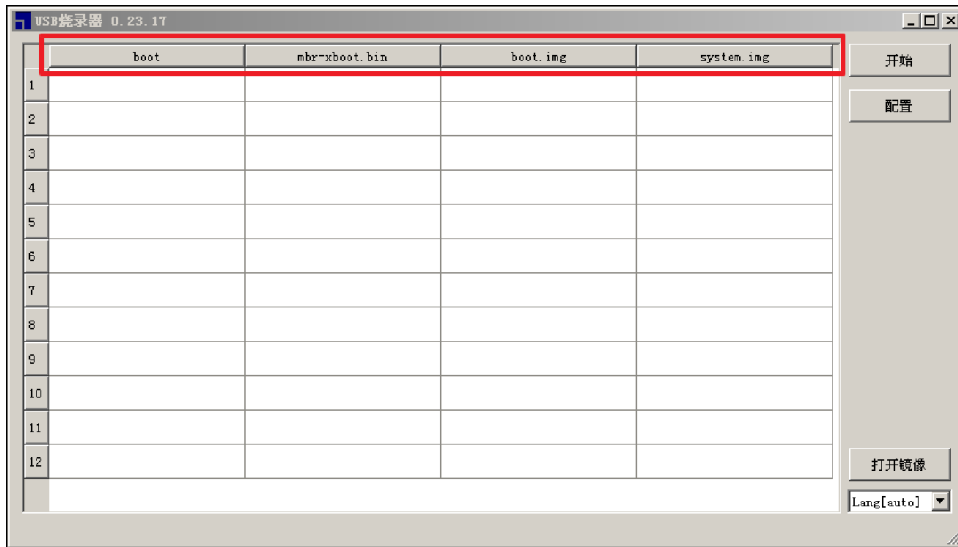


图 5-15 烧录工具要烧录的文件显示区域

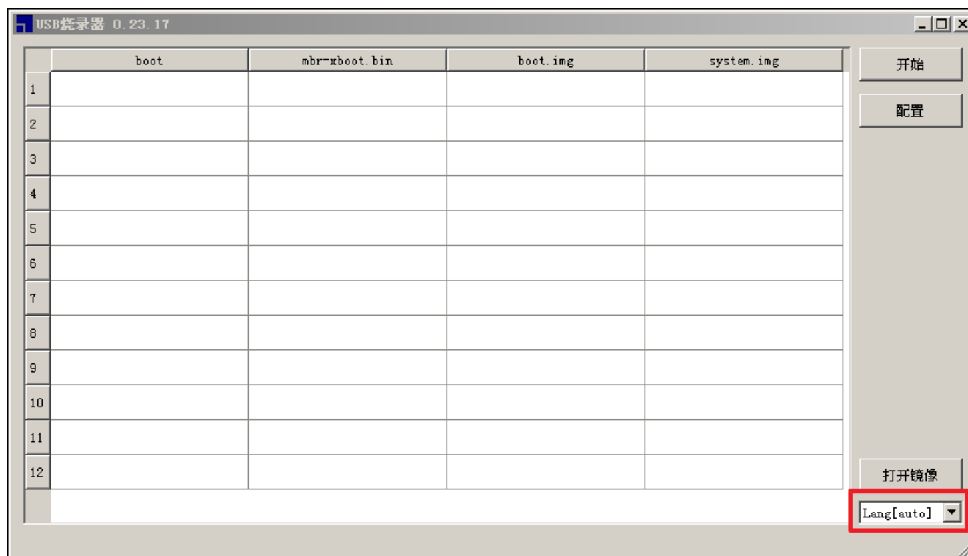


图 5-16 烧录工具语言选择复选框

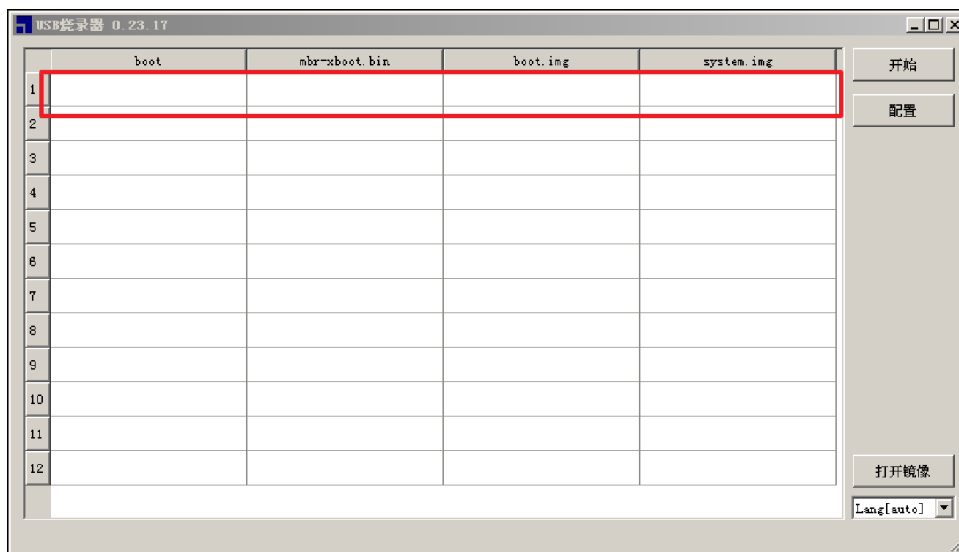


图 5-17 第一个被烧录设备的进度条显示区域

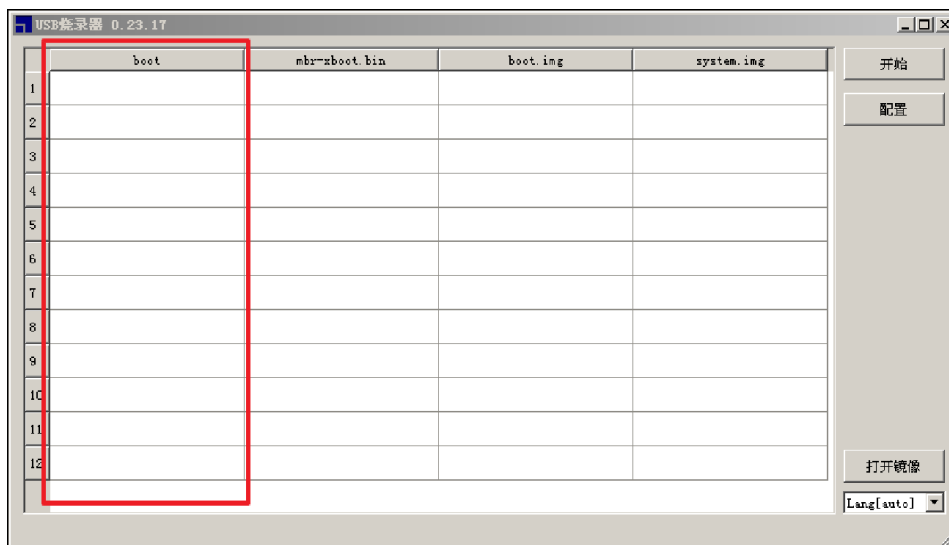


图 5-18 各个设备 boot 阶段进度条显示区域

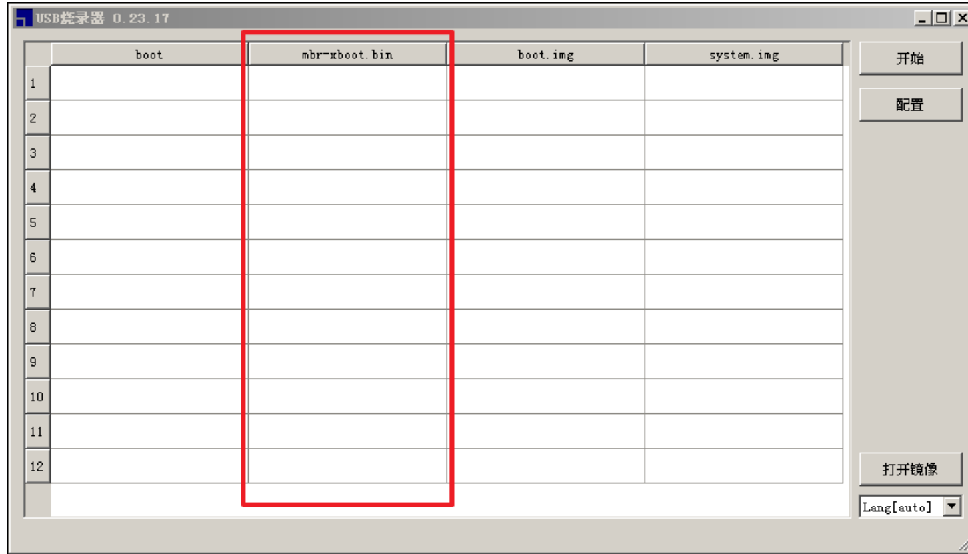


图 5-19 各个设备烧录第一个文件进度条显示区域

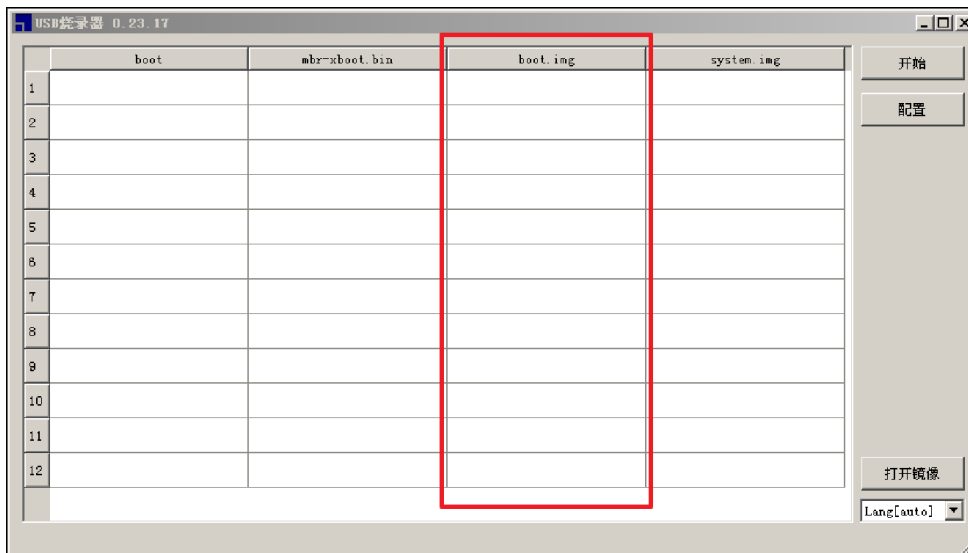


图 5-20 各个设备烧录第二个文件进度条显示区域

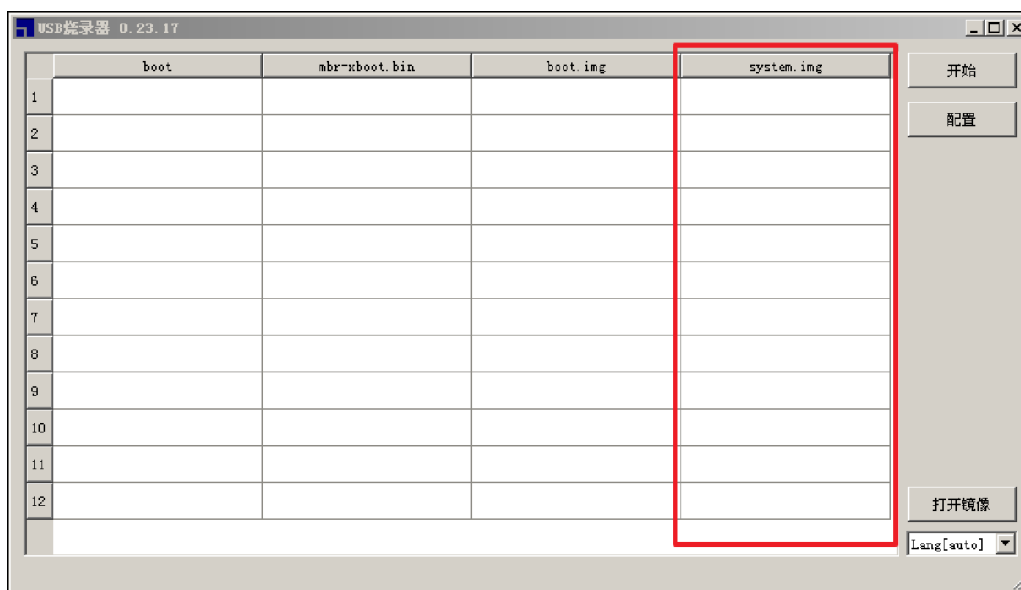


图 5-21 各个设备烧录第三个文件进度条显示区域

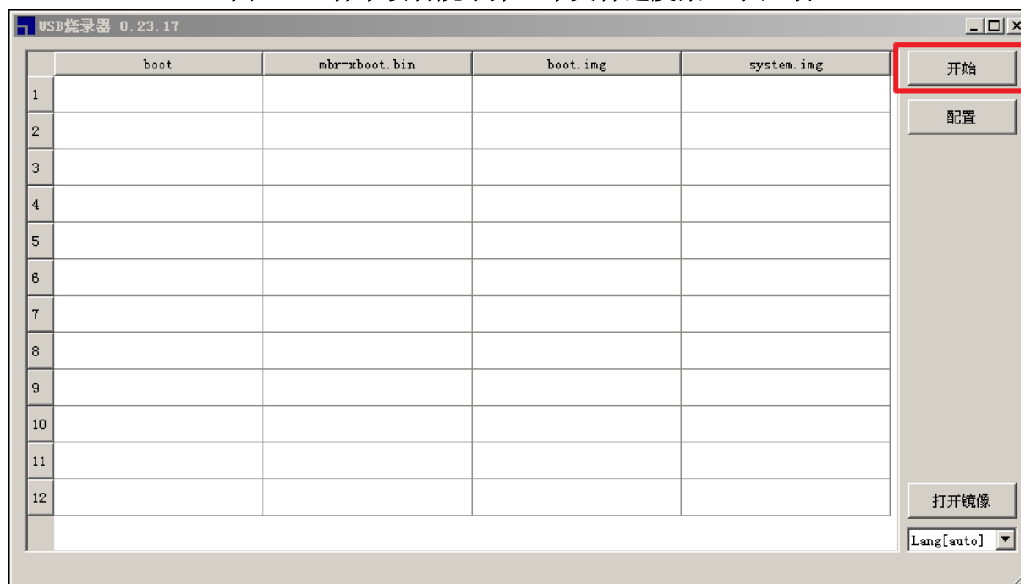


图 5-22 烧录工具的开始烧录按钮



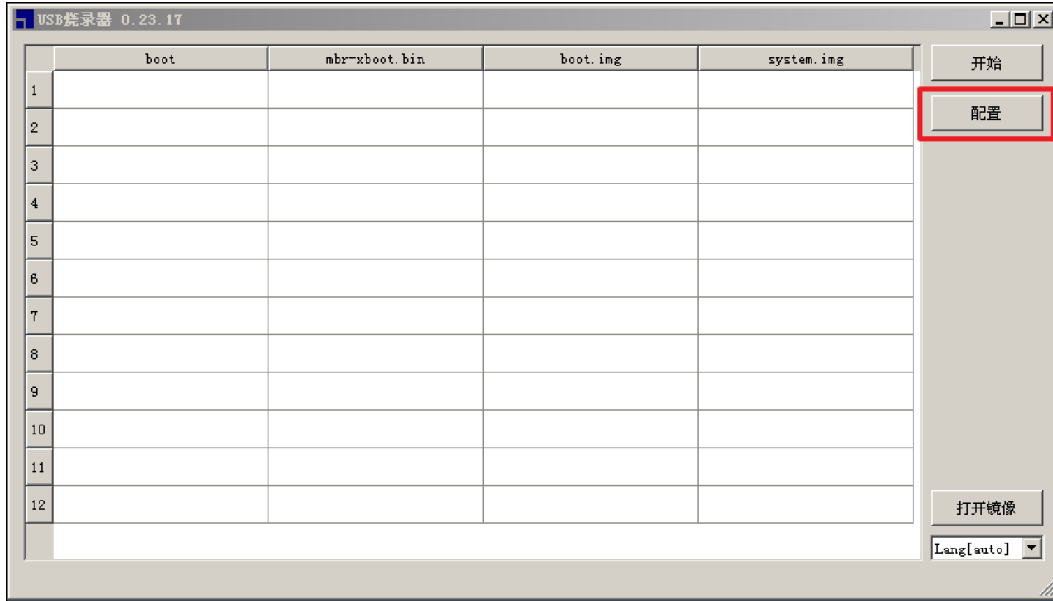


图 5-23 烧录工具的配置按钮

如果需要重新配置一些选项，点击主界面上的“config”按钮（中文显示为“配置”），即点击“配置”按钮进入配置界面，选择“信息”配置，选择平台（M200），板级等如图 5-24：

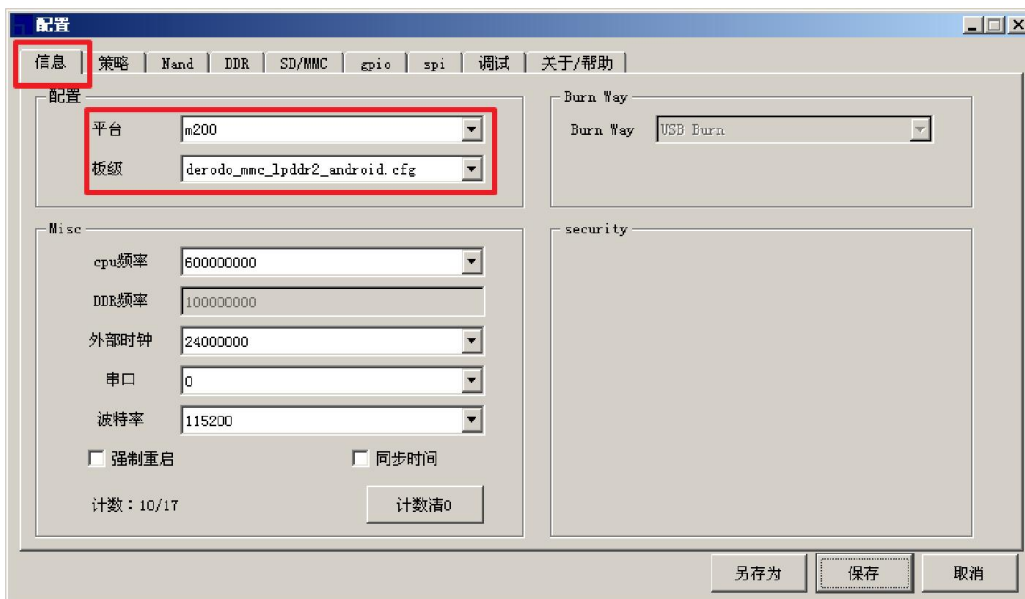


图 5-24 烧录工具的信息配置

选择“策略”配置，勾选所要烧录的镜像文件，**注意：烧录文件的偏移值**，如图 5-25：

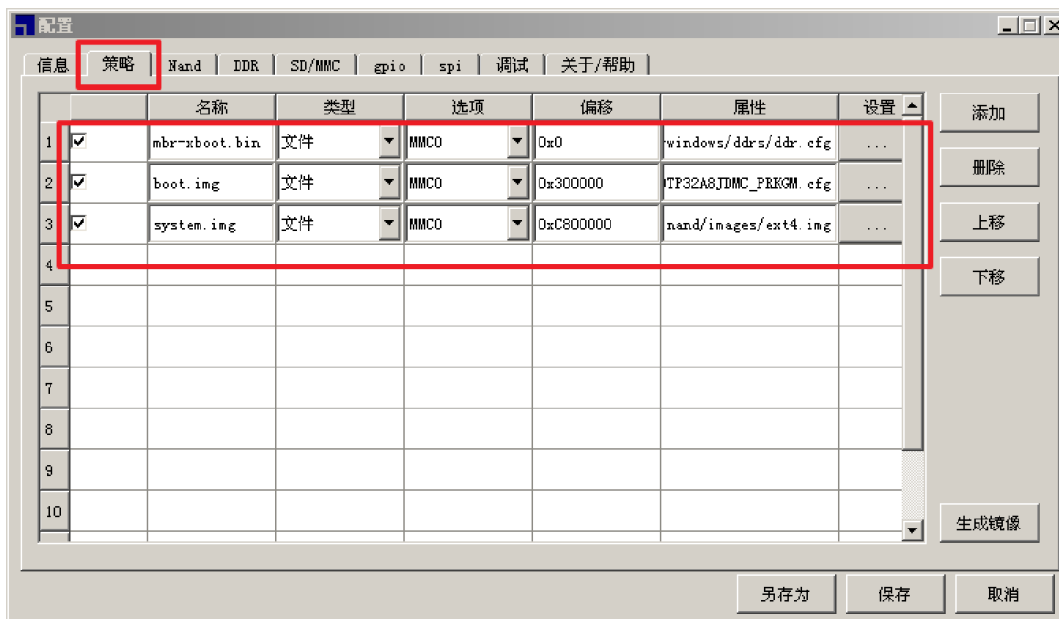


图 5-25 烧录工具的策略配置界面

选择“SD/MMC”配置，确认烧录前是否擦除，如图 5-26:

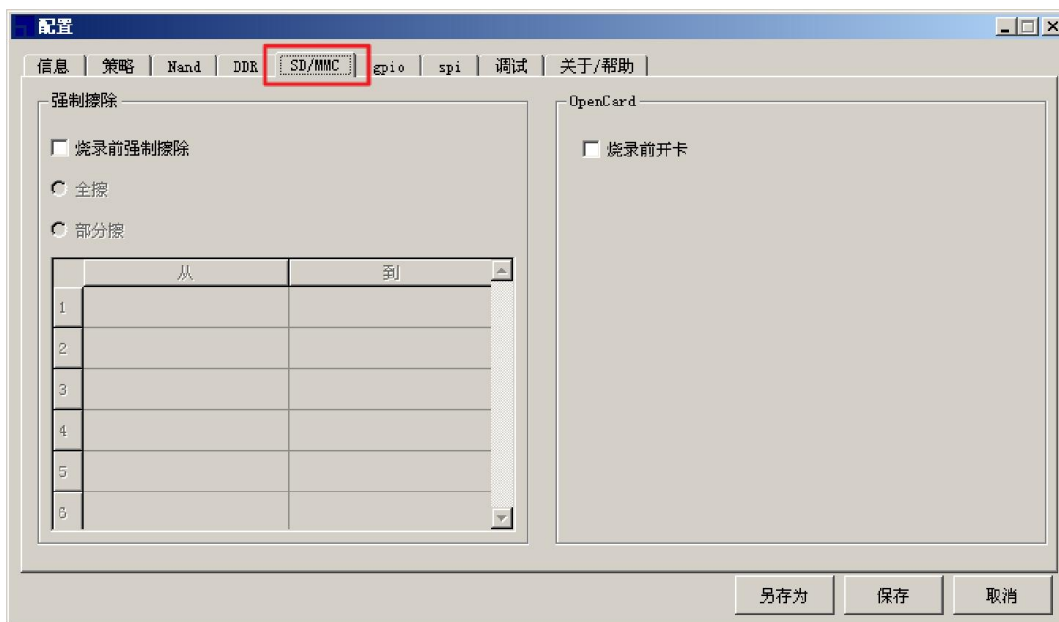


图 5-26 烧录工具的擦除配置界面

若烧录前需要将 uboot、boot、system 等全部擦除，只需同时勾选“烧录前强制擦除”与“全擦”，如下图 5-27:

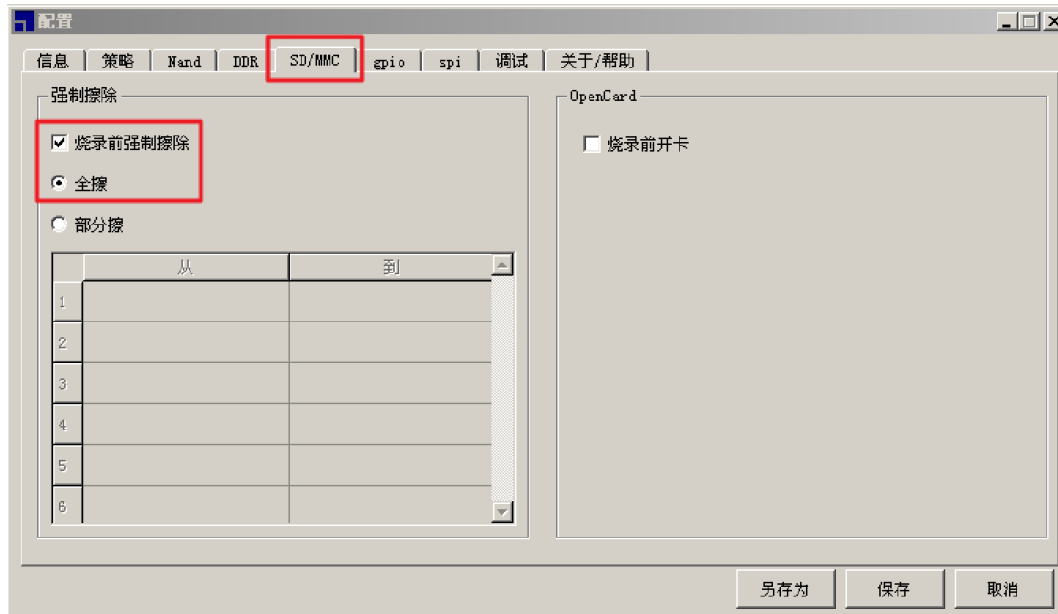


图 5-27 烧录工具的全部擦除配置界面

若烧录前只需要擦除 uboot、boot、system 等某个文件，只需同时勾选“烧录前强制擦除”与“部分擦”，如下 5-28 图：

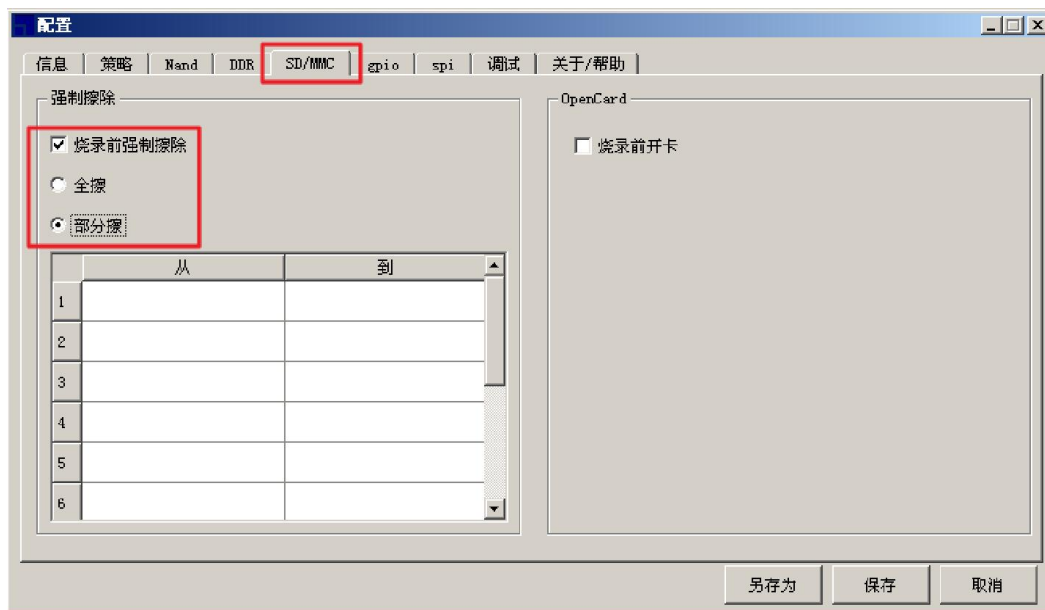


图 5-28 烧录工具的部分擦除配置界面

同时烧录工具策略界面，只需勾选需要烧录的文件。

完成后，一定要点击“保存”，保存后以上的配置才可被保存到 config 里，否则没有完成配置。按着“SW1”键，按一下“SW3”键，使板子进入烧录模式（具体方法，参考：图 5-0 介绍）。进入烧录后，烧录工具界面，如下图 5-26：

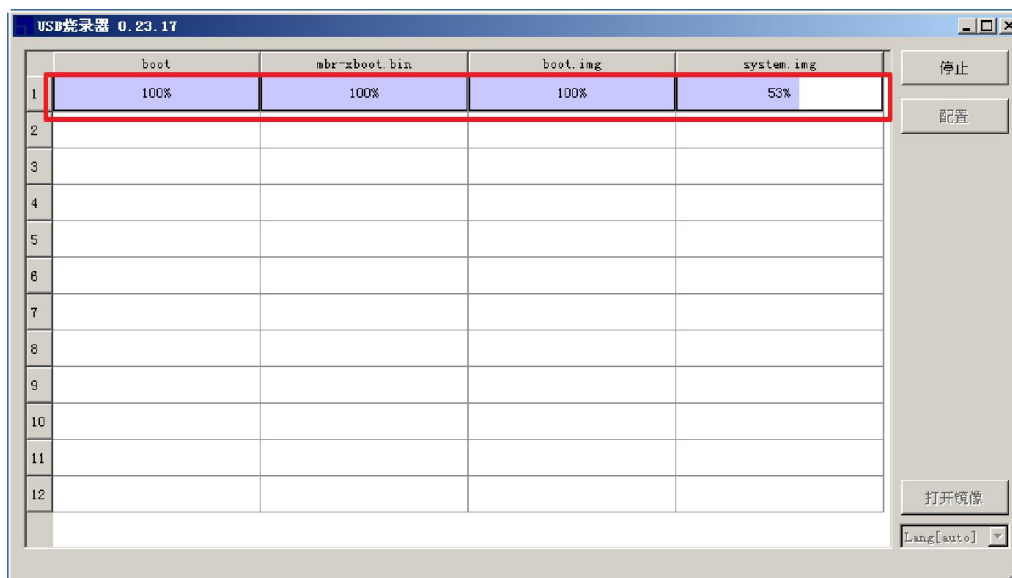


图 5-26 烧录工具烧录界面

等待几分钟，烧录完成，如下图 5-27:

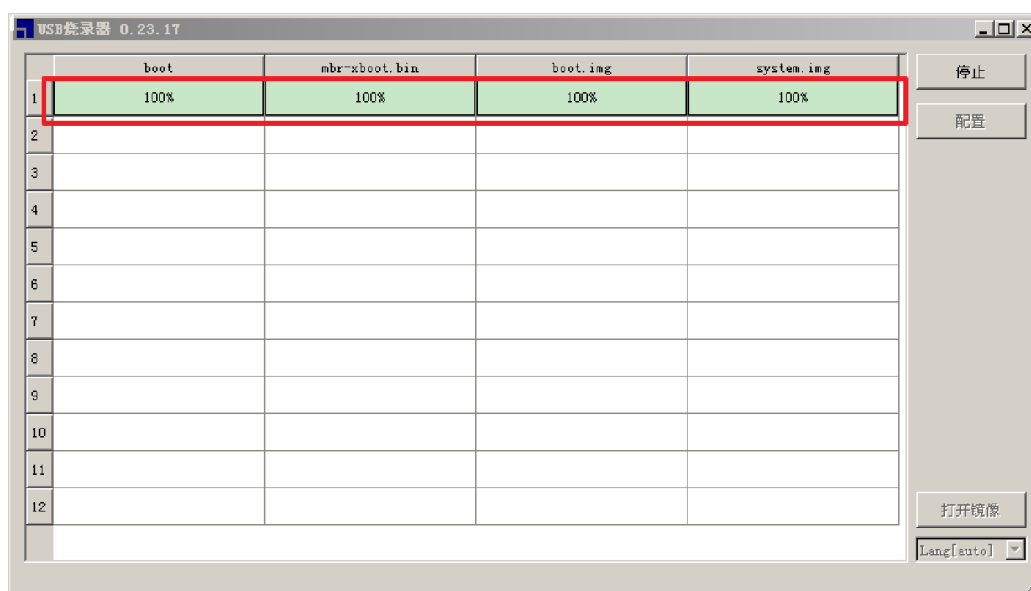


图 5-27 烧录完成界面

### 5.3.3 Linux 下使用 USBCloner 烧录工具

解压 USBCloner 烧录工具压缩包，然后进入 USBCloner 烧录工具目录，首先为“cloner.exe”文件赋予可执行权限，然后用**超级用户权限**执行./cloner 即 (**sudo./cloner**)，其界面上使用方法与 windows 下使用方法类同，请参考。

## 6 个性化定制

### 6.1 开机动画

#### 6.1.1 替换已经制作好的动画

如果有已经做好的动画 `bootanimation.zip`，将其拷贝到 `/system/media/bootanimation.zip` 或 `/data/local/bootanimation.zip`，即可实现修改开机动画。

需要注意的是，在 `init.rc` 中需要启动 `bootanimation` 的 `service`，例如 `device/ingenic/mensa/config/init.rc` 启动服务代码如下：

```
service bootanim /system/bin/bootanimation
    class main
    user graphics
    group graphics
    disabled
    oneshot
```

#### 6.1.2 制作 bootanimation.zip

1) 根据屏幕的分辨率制作同样的图片，动画有几帧，就做对应的几张 `jpg` 图片，按顺序命名 `all_0001.jpg`，`all_0002.jpg` ..... 等。

2) 压缩包里的文件目录结构

两个目录 `part1`，`part2`，分别是第一部分的动画，和等待系统启动时的循环动画部分，一个文件 `desc.txt` 的配置文件。

`part1` 中的文件为第一步做出来的各个图片，`part2` 中的文件为第一步做出来的各个图片。

`desc.txt` 文件内容为：

第一行：图片分辨率宽 图片分辨率高 播放下一帧的间隔时间（ms）

例如：800 480 5

第二行：p 1 0 part1

第三行：p 0 0 part2

3) Zip 包的制作

在 windows 下使用 `rar` 工具进行压缩，注意删除所有目录下隐藏的 `Thumbs.db` 等。

鼠标选中 `part1`，`part2` 和 `desc.txt`，点击右键选择“添加到压缩文件”，输入压缩文件名字 `bootanimation.zip`，压缩文件格式选中“ZIP”，压缩方式选择“存储”，如下图所示。点击确定，成功压缩后，`bootanimation.zip` 制作成功。



图 6-1 压缩选项

## 7 关注君正

欢迎关注君正的微博和微信，第一时间获得君正最新动态。

### 7.1 新浪微博号：北京君正官微



图 7-1 北京君正新浪微博

### 7.2 微信号：北京君正



图 7-2 北京君正微信

### 7.3 北京君正官网

官网主页: <http://www.ingenic.com>

芯片产品主页: <http://www.ingenic.com/?product/id/2.html>

开发支持主页: <http://www.ingenic.com/?product/id/2/lm/3.html>

如有技术方面问题和建议, 请联系君正技术支持:

邮箱: [support@ingenic.com](mailto:support@ingenic.com)

电话: 4008-900-223