

# Firefox OS移植

## ファームウェアのビルド

Android 4.2ベースのFirefox OS(ファームウェア全体)のビルドにはUbuntu Server 12.04.5 LTS 64-bitを利用します。ブートローダー・カーネルのビルドはAndroid 4.2 SDKの arm-eabi-4.6(prebuilts/gcc/linux-x86/arm/arm-eabi-4.6/bin/arm-eabi-)があれば可能ですが、ここではビルド環境と同じ環境(Ubuntu Server 12.04.5 LTS 64-bit)を利用する事とします。

## ブートローダー・カーネルのビルド

必要なパッケージをインストールします。

```
$ sudo apt-get install build-essential git ia32-libs-multiarch  
libncurses5-dev lzop
```

コンパイラおよびカーネルのソースを展開します。(TBD:独自のgitリポジトリを用意する)

```
$ tar xf MarsBoard_RK3066_Box_Android_4.2.2_SDK_V1.0.tar.gz  
MarsBoard_RK3066_Box_Android_4.2.2_SDK_V1.0/prebuilts/gcc/linux-x86/a  
rm/arm-eabi-4.6 MarsBoard_RK3066_Box_Android_4.2.2_SDK_V1.0/kernel
```

ブートローダーのソースをAndroid SDKのトップディレクトリに展開します。(TBD:独自のgitリポジトリを用意する)

```
$ cd MarsBoard_RK3066_Box_Android_4.2.2_SDK_V1.0  
$ git clone -b u-boot-rk3066  
https://github.com/linux-rockchip/u-boot-rockchip.git
```

ブートローダーをビルドします。

```
$ cd u-boot-rockchip  
$ make rk30xx  
$ cd ..
```

カーネルをビルドします。

```
$ cd kernel  
$ make rk30_marsboard_defconfig  
$ make kernel.img  
$ cd ..
```

カーネルをカスタマイズする場合、次のコマンドを実行し、再度ビルドします。

```
$ make menuconfig
```

カーネルバイナリのサイズを小さく(圧縮)する場合、次のコマンドでビルドします。

```
$ make zkernel.img
```

## Firefox OSのビルド

必要なパッケージをインストールします。

```
$ sudo apt-get install autoconf2.13 bison bzip2 ccache curl flex gawk  
gcc g++ g++-multilib git ia32-libs lib32ncurses5-dev lib32z1-dev  
libgl1-mesa-dev libx11-dev make zip libxml2-utils libasound2
```

PPAを利用してJDK6をインストールします。

```
$ sudo apt-get install python-software-properties  
$ sudo add-apt-repository ppa:webupd8team/java  
$ sudo apt-get update  
$ sudo apt-get install oracle-java6-installer
```

複数のバージョンのJDKをインストールしている場合、次のコマンドでJDK6を標準のJDKに設定できます。

```
$ sudo apt-get install oracle-java6-set-default
```

ccacheを設定します。(ユーザ毎に行います)

```
$ ccache -M 10G
```

Firefox OSのソースを取得します。(初回のみ。数GBダウンロードします)

```
$ git clone https://github.com/flingone/B2G-FlingOne.git  
$ cd B2G-FlingOne  
$ ./config.sh -j 4 rk30sdk
```

MozOpenHardのリポジトリのソースを取得します。(初回のみ)

```
$ cd .repo  
$ git clone https://github.com/MozOpenHard/local_manifests.git  
$ cd ..  
$ repo sync -d -j 4
```

ソースを更新します。(2回目以降)

```
$ cd B2G-FlingOne  
$ git pull
```

```
$ cd .repo/local_manifests/  
$ git pull  
$ cd ../../  
$ repo sync -d -j 4
```

Firefox OSをビルドします。(マシンの性能により数時間かかる場合があります)

```
$ ./build.sh -j 4  
$ ./flash.sh
```

rockdev/Image/にboot.img, kernel.img, recovery.img, system.imgが生成されます。

## シリアルコンソール

起動メッセージの確認や問題の調査の為、DEBUGポートとPCを接続し、シリアルコンソールにアクセスできるようにします。

### Linux (Ubuntu)

dialoutグループに自分のアカウントを追加します。設定後に一旦ログアウトし、再度ログインします。

```
$ sudo usermod -G dialout -a (アカウント名)
```

シリアル通信プログラムをインストールします。

```
$ sudo apt-get install picocom
```

DEBUGポートとLinux PCのUSBポートを接続し、

```
$ picocom -b 115200 /dev/ttyUSB0
```

### Windows

TBD

### Mac

TBD

## ローダーモード

TBD

# ファームウェアの更新

## upgrade\_tool (Linux)

plugdevグループに自分のアカウントを追加します。

```
$ sudo usermod -G plugdev -a (アカウント名)
```

設定を反映させる為、設定後に一旦ログアウトし、再度ログインします。

/etc/udev/rules.d/51-android.rulesに次の一行を追加(ファイルがなければ新規作成)します。

```
SUBSYSTEM=="usb", ATTR{idVendor}=="2207", MODE="0666",  
GROUP="plugdev"
```

設定を反映します。

```
$ sudo udevadm control --reload-rules
```

Linux\_Upgrade\_Tool\_v1.2.1.zipを任意の場所に展開し、PATHを設定します。

```
$ unzip Linux_Upgrade_Tool_v1.2.1.zip  
$ chmod a+x Linux_Upgrade_Tool_v1.2.1/upgrade_tool  
$ PATH=${PATH}:\`pwd\`/Linux_Upgrade_Tool_v1.2.1
```

rockdev-rk3066.zipを任意の場所に展開します。

```
$ unzip rockdev-rk3066.zip  
$ cd rockdev-rk3066  
$ chmod a+x afptool rkImageMaker mkupdate.sh unpack.sh
```

ビルドしたboot.img, kernel.img, recovery.img, system.imgをpackage-fileに従い配置し(または配置したファイルに従いpackage-fileを更新し)、mkupdate.shを実行します。

```
$ ./mkupdate.sh
```

rockdev/にupdate.imgが生成されます。

次のコマンドを実行します。

```
$ upgrade_tool uf update.img
```

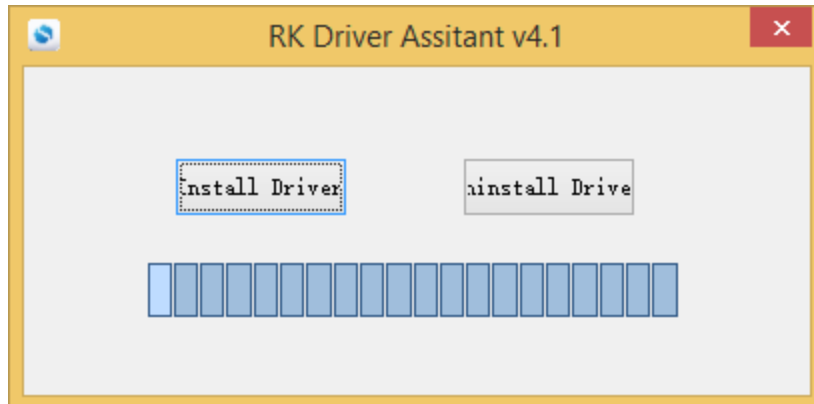
次のように、必要なファイルだけ書き込む事もできます。

```
$ upgrade_tool ul RK3066Loader_miniall.bin  
$ upgrade_tool di -p parameter  
$ upgrade_tool di uboot uboot.img  
$ upgrade_tool di -m misc.img  
$ upgrade_tool di -k kernel.img
```

```
$ upgrade_tool di -b boot.img
$ upgrade_tool di -r recovery.img
$ upgrade_tool di -s system.img
```

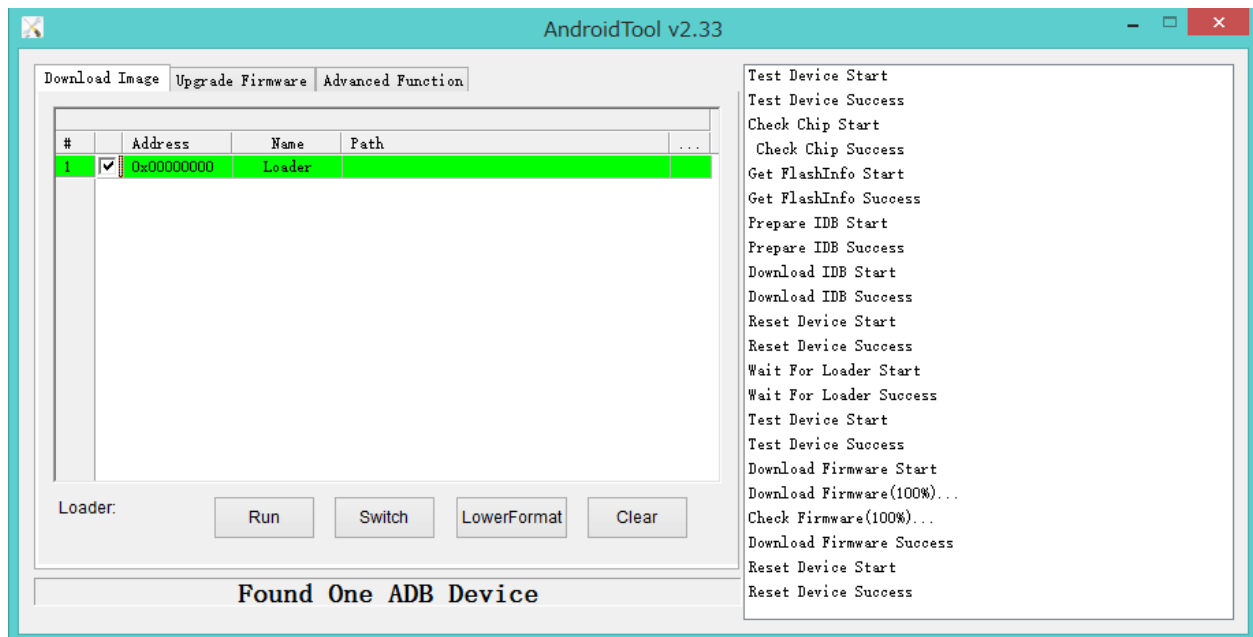
## AndroidTool (Windows)

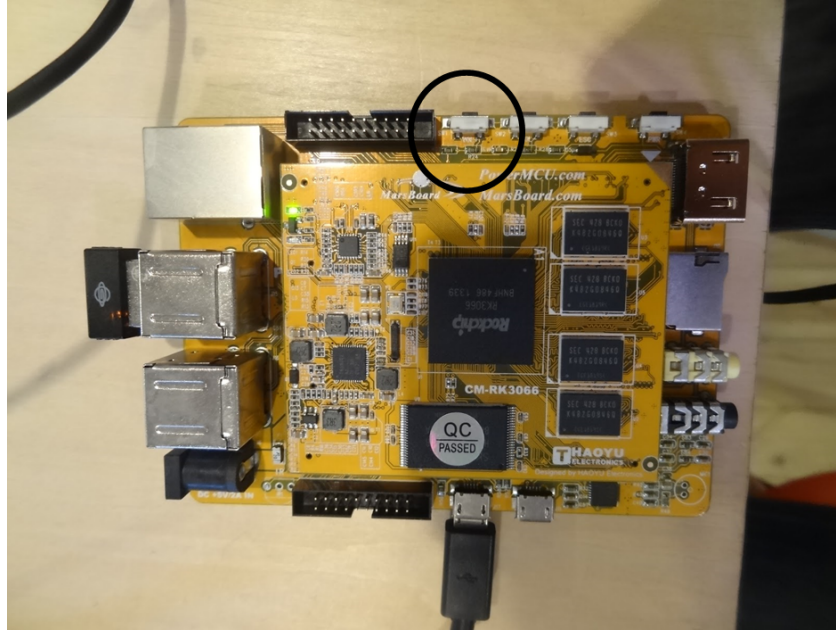
DriverAssistant\_v4.1.1.zipを展開し、DriverInstall.exeを実行します。



左側の“Install Driver”ボタンを押してRockusbドライバおよびADBドライバをインストールします。

AndroidTool\_v2.33.zipを展開して、AndroidTool.exeを実行します。MarsBoardとPCを接続すると、ウインドウに“Found One ADB Device”と表示されます。Switchを押すと、“Found One LOADER Device”と表示されます。(MarsBoardの一番左のボタン(VOL+)を押しながら起動すると自動でLOADERモードになります)

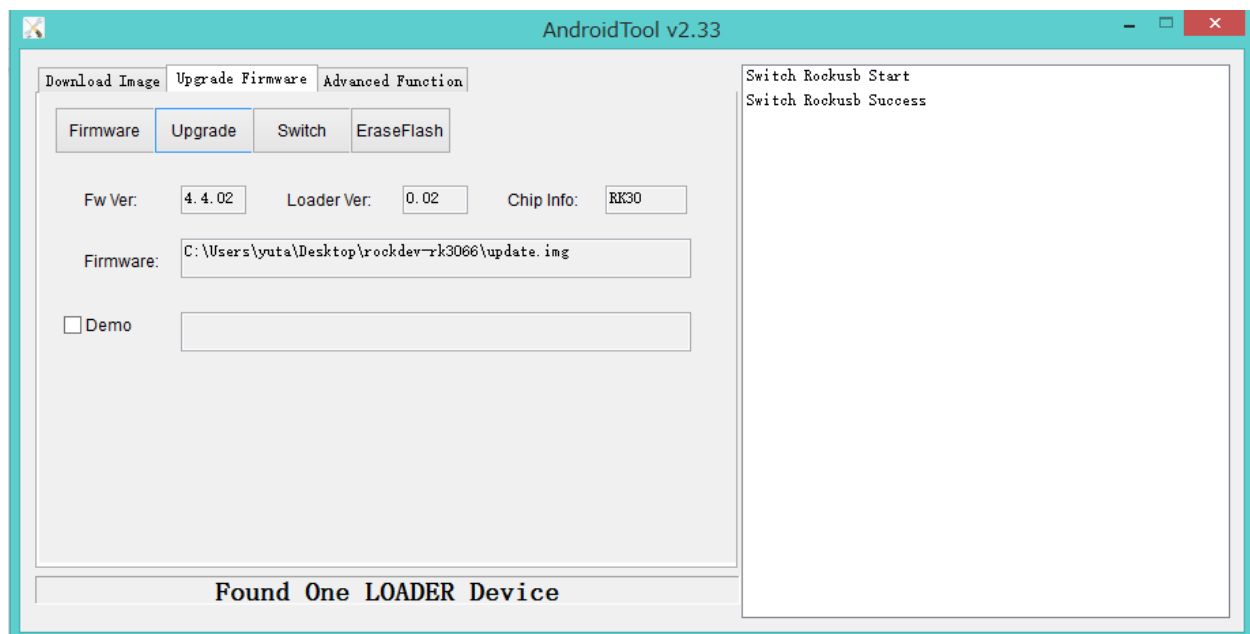




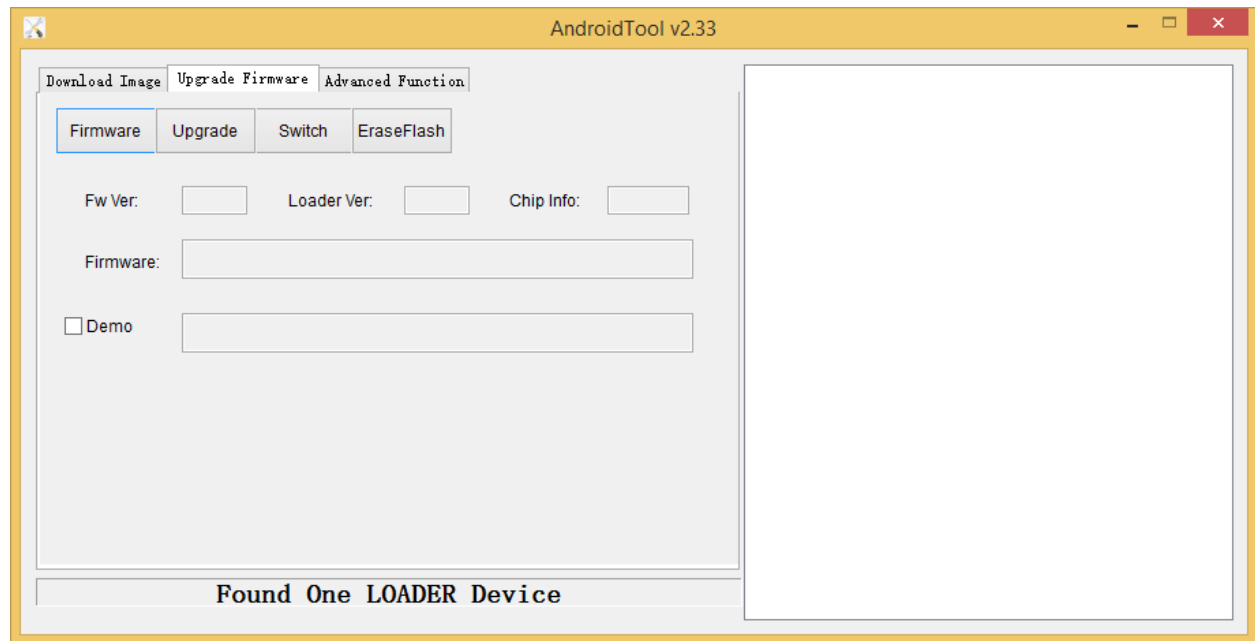
Upgrade Firmwareのタブをクリックします。Firmwareをクリックしてimgファイルを選択します。

Upgradeのタブをクリックすると、書き込みが始まります。書き込みが終了すると自動でOSが起動します。

(LinuxとWindowsでimgファイルが異なるので注意が必要です。2015/02/08現在)



## fastboot (Linux/Windows/Mac)



TBD

## rkflashtool (Linux/etc)

TBD