International Journal of Current Research and Modern Education (IJCRME)
ISSN (Online): 2455 - 5428
(www.rdmodernresearch.com) Volume I, Issue II, 2016

FUTURE DIRECTIONS FOR FIRMWARE FLASHING ON ANDROID DEVICES

Yadunandan Laxman Huded* & S. Srinath**
Department of Computer Science and Engineering, Sri Jayachamarajendra College of Engineering, Mysore, Karnataka

Abstract*:*

This paper explains the process of flashing custom ROM / Firmware of android to any of the android platform present in the market. The paper also involves different commands and tools used to flash the android firmware. In this paper we also explained the step by step approach to be followed to flash the firmware, like identifying the device, initialising the boot loader, stitching the firmware images and flashing the images in appropriate order.

Key Words: Flashing, Boot Loader, Firmware, ROM, Fast Boot Mode, ADB Commands& Stitching

1. Introduction:

Smart phones, which we are using now a days has an internal ROM (Read Only Memory). As name suggests ROM contains the copy of an operating system, which cannot be modified at all during the normal operation of the device. Keeping OS untouched ensures the malfunction free execution of the device.

The read-only operating systems that we just discussed above are also called 'firmware', as they stay firmly in place without modification access to the users of the device. Modification of firmware is still however possible, just not under normal usage. Many devices require specialized hardware to be used for the purpose while other devices have the storage set as read-only through software protection only, which can be removed or overridden without the need for any specialized hardware, just by using software written for the purpose, often but not always requiring connection to a computer. Thus, the terms 'operating system' and 'firmware' both refer to the same thing and can be used interchangeably when applied to such devices.

The ROM memory used in smart phones and tablets etc. is often same as flash memory found in SD cards and USB flash drives, simply optimized for better speed and performance while running the operating system. As explained above, it is read-only under normal usage and requires a special procedure for any modifications to be made to its contents. The procedure of modifying or replacing the contents of such flash memory is known as flashing. Thus, in layman's terms, flashing is essentially the same as installing or modifying the firmware of a device that is stored on its protected flash memory.

This paper explains the technical aspects and steps required to flash an android device.

2. Technology Involved:

This section explains the various technical aspects that are come into act when we need to flash an android board. This involves the explanations of system software like boot loader, also explains techniques like adb debugging, flashing, fast boot mode etc.

Firmware: Firmware [1] refers to the micro programs present on ROM (Read Only Memory) modules, which contain low-level (e.g., hexadecimal, machine code) software. They enable the device on which they are present to take stock of its capabilities and to render those capabilities functional. The information loaded onto the ROM is non-volatile meaning that it is not lost when power is switched off. The most basic example

of firmware would be the BIOS that come with the motherboard of your PC. The firmware also coordinates the activities of the hardware during normal operation and contains programming constructs used to perform those operations. The use of firmware gives more flexibility compared to the use of pure hardware circuitry. For example, in a typical modem, the firmware will be a factor in establishing the modem's data rate and command set recognition. Some firmware are non-rewriteable while others are upgradeable, meaning that it is possible to upgrade the firmware of the device by connecting it to your PC in a particular configuration and then running the software provided by the manufacturer. This process is called "flashing firmware" or simply "flashing". This becomes necessary when the device becomes incompatible with newer operating systems or to simply enhance the performance of the device. For example, CD and DVD drive manufacturers often release firmware updates available that allow the drives to read faster media.

Boot Loader: A boot loader [2] is a system software that loads an operating system (OS) or runtime environment on to the ROM, once all of the self-tests are over. The boot loader configures the device to an initial known state and has a means to select where to start executing the kernel. It can allow you to make this selection, which give you for example the opportunity to start an alternative Linux kernel, or Windows. Because the boot loader is an essential component of the boot process, it is stored in non-volatile memory, such as flash memory.When power is first applied to a processor board, many elements of hardware must be initialized before even the simplest program can run. Each architecture and processor has a set of predefined actions and configurations, which include fetching some initialization code from an on-board storage device (usually Flash memory). This early initialization code is part of the boot loader and is responsible for breathing life into the processor and related hardware components. Most processors have a default address from which the first bytes of code are fetched upon application of power and release of reset.

Flash Memory: Most mobile devices are provisioned with an internal flash storage, an external SD card slot, and a limited amount of RAM. In addition, some devices also have a non-removable SD partition inside the phone; such storage is still treated as external. Figure 2 shows the internal raw NAND and external flash storage. The internal flash storage contains all the important system partitions, including partitions for the boot loader and kernel, recovery, system settings, pre-installed system applications, and user-installed application data. The external storage is primarily used for storing user content such as media files (i.e., songs, movies, and photographs), documents, and backup images

Android Debug Bridge (ADB): The Android Debug Bridge (ADB) is used to keep track of all Android devices and emulators instances connected to or running on a given host developer machine and to implement various control commands(e.g. "adb shell", "adb pull", etc..) for the benefit of clients (command-line users, or helper programs like DDMS). These commands are what is called a 'service' in ADB. It is a client-server program that includes three components mainly a client, which runs on your development machine. You can invoke a client from a shell by issuing an adb command. Other Android tools such as the ADT plugin and DDMS also create adb clients. A server, which runs as a background process on your development machine. The server manages communication between the client and the adb daemon running on an emulator or device. A daemon, which runs as a background process on each emulator or device instance.

Fast Boot Mode: In Android, fast boot is a special diagnostic and engineering protocol that you can boot your Android device into. While in fast boot, you can modify the file system images from a computer over a USB connection. A powerful, nerdy tool deserves to be broken down into terms we all can understand. Fast boot is three different things with the same name: A protocol for communication between your phone hardware and a computer, software that runs on the phone when in fast boot mode and the executable file on the computer you use to make them talk to each other. Not all phones have a fast boot mode that the user can access. It's turned on with Nexus devices by default (as well as a few other phones and tablets) and has been enabled by independent Android developers and enthusiasts on some other phones. It also requires software from the Android SDK, and different USB drivers for Windows computers. Fast boot runs on Windows, Mac, and Linux and all the information about setting it up can be found in the forums if you're interested. Once set up, you boot your phone to fast boot and you can flash image files to your phone's internal memory. Flashing a custom recovery (after unlocking the boot loader) is a popular use case, as is resetting it all back using factory images after we're done breaking things by flashing the factory images. With an unlocked boot loader, the images you flash don't need to be signed with a particular key, so just about anything will try to flash even if it shouldn't be used, so use care. There are other commands you can use with fast boot, and they are a bit more advanced. Things like erasing partitions and overriding kernel command line options can be done, and this makes the tool very useful for developing hardware and software solutions that may need customized booting procedures. With a little bit of knowledge, and the right Android hardware, fast boot can be a great tool.

3. Proposed Work:

This section explains how we can flash the firmware to the android platform. In this section, we also explained different steps involved in the process of flashing. One of the best things about the openness of the Android platform is that if you are unhappy with the stock OS, you can install one of many modified versions of Android (called Firm wares) on your device. A new firmware can bring you the latest version of Android before your manufacturer does, or it can replace your manufacturer mode version of Android with a clean, stock version. Or, it can take your existing version and just beef it up with awesome new features. The following flowchart shows the different steps of flashing.

To boot any new firmware we need not to root it again from scratch, but by unlocking the boot loader and by custom recovery, we can flash new firm ware. The figure 1 represents the flowchart of the flashing process. The flow chart consists of the different steps involved in the flashing process like identifying the device using adb commands, putting the device in fast boot mode, specifying the USB port and baud rate at which the firmware will be flashed, selecting the suitable firmware, stitching the selected firmware in proper order and lastly flashing the device with the stitched images.

Algorithm:

The detailed algorithm with all the technical aspects are given below.

Step 1: Connect the device which to be flashed with the firmware to the host system and communicate with the device using adb commands.

- ✓ Make sure the device is recognized using the following command 1.
  - ❖ Command 1: “adb devices”
- ✓ Put the device in boot loader mode by using the command 2.

- Command 2: "adb reboot bootloader"

Flow Chart:

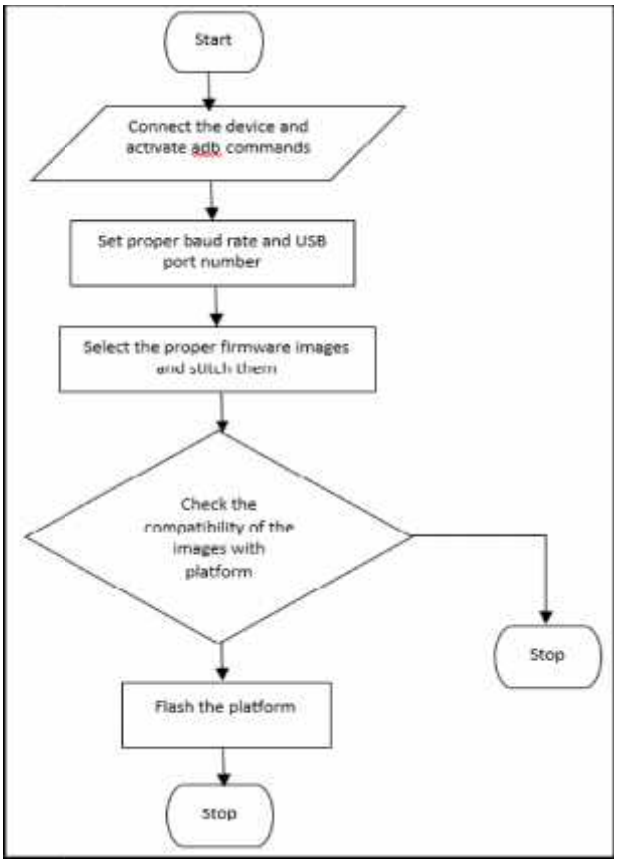

Figure 1: Steps involved in flashing

The above adb commands can be used to communicate with the connected android device.

Step 2: Once the device gets identified fix the baud rate and USB port number.

Step 3: Select the proper firmware images and stich them according to their addresses and we need to order them based on their flashing priority. i.e recovery fls, systemfls, boot fls

The following figure shows the order in which the .fls files are stitched.

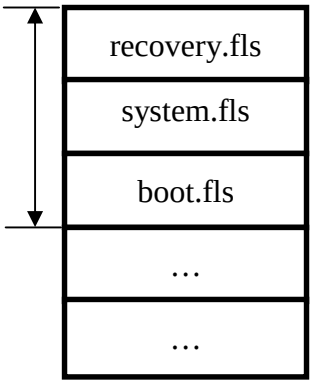

Figure 2: Ordering and Stitching the Firm Wares

The figure 2 shows the order in which the firmware images must be stitched.

Step 4: Once the stitching is complete the total stitched file can be flashed completely using the following commands.

- ✓ Flash recovery image first using the command 3.
  - ❖ Command 3: "fastboot flash recovery recovery.fls
- ✓ Flash system.fls using the following command 4.
  - ❖ Command 4: "fastboot flash system system.fls
- ✓ Flash the boot fls using the command 5.
  - ❖ Command 5: "fastboot flash boot boot.fls"

Step 5: Once the mandatory images are flashed, we can proceed further the flash the updated complete firmware. To do this we need to use the following command.

- ❖ Command 6: "adbsideload update.zip"

Using this .zip we flash the complete firmware.

4. Conclusion:

In this paper, we explained how we flash the android custom ROM/Firmware. By using the above-mentioned steps, we can flash the android platform with required firm wares. Irrespective of type of architecture, we can use the same method to flash all the available android platforms. This flexibility makes the present method very user friendly and effective.

5. References:

1. Whitson Gordon, "Lifeshacker", Jan 2014, <http://lifehacker.com/how-to-flash-a-rom-to-your-android-phone>
2. Intel wiki, <https://www.employeeportal/intelwiki/bootloader>
3. Android debug bridge, <https://androiddevelopers/adb>
4. Hyojun Kim, Nitin Agrawal, Cristian Ungureanu"Revisiting Storage for Smartphones",<https://www.usenix.org/legacy/event/fast12/tech/full_papers/Kim.pdf>