Nicholas Roy
Samsung sIII

Journal #3 - Samsung s III

Concluding Trial 1

On my last attempt, I tried flashing directly to the phone using hiemdall in Ubuntu terminal. What I didn't know at the time was the specifics on the command I was using.

Flash boot image:
$ sudo heimdall flash --BOOT /image/path/boot.img --no-reboot

Flash system image:
$ sudo heimdall flash --SYSTEM /image/path/system.img --no-reboot

What I previously didn't understand about these commands was the significance of the *BOOT* and *SYSTEM* tags, as well as the *RECOVERY* tag I tried briefly with the same results.

After further digging, I looked back into the *heimdall --print-pit* command and realized that the PIT (Partition Information File), which contained 28 entries, were all different partitions of the mobile device I connected to the computer. By using the *BOOT* and *SYSTEM* tags, I was targeting the files to be flashed into the *boot* and *system* partitions respectfully. In theory, this should've allowed me to directly flash the correct partitions while the phone was in download mode.

This answered my suspicion of needing to specify the pit file in the flash command, which was not necessary.

Still, I got the same empty bulk transfer error message as before, and decided to give up on this method.

Trial 2 - Using flash SD card through TWRP

After this, I wanted to test if an SD card was even necessary. Logically, I expected there to be the flashing of a boot and system partition onto an SD card and then to the phone, much like how a flash USB drive works.

First, I needed to wipe the SD card. The method I used was a command to turn all data to zeros without changing any formatting. This maybe was a little stupid but it was a quick and easy fix. I lied. It took 2 hours. I do not recommend wiping this way, as I'm sure there are faster and more efficient ways. In case you're a masochist, here's the command.

$ sudo dd if=/dev/zero of=/dev/sdi bs=1M count=10

Next, I used *fdisk* to delete all existing partitions (there were none), then ‘n’ in the *fdisk* menu to create 2 new ones. 1 is for boot and should be around 500MB, and the other 12GB for the system. I labeled them sdi1 and sdi2 respectively.

Next I flashed the files to their partition using the dd command:

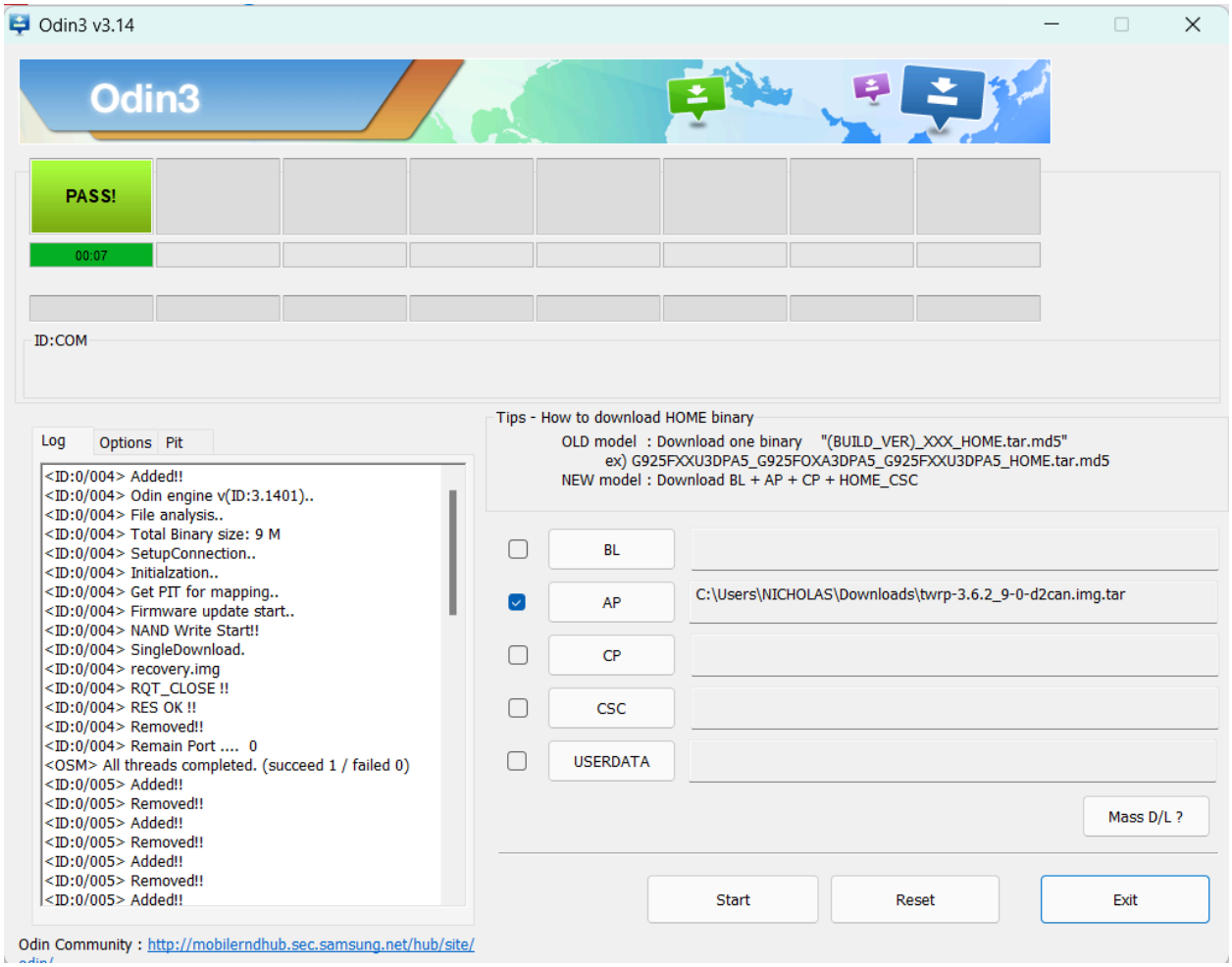
$ sudo dd if=/path/to/file/boot.img of=/dev/sdi1 bs=4M status=progress
$ sudo dd if=/path/to/file/system.img of=/dev/sdi2 bs=4M status=progress

I made sure the boot partition was formatted as FAT32.
And the remaining space was formatted as ext4.

These appeared to work, however I am beyond silly as it doesn’t seem to be possible to flash the phone from an SD with my current arrangement.

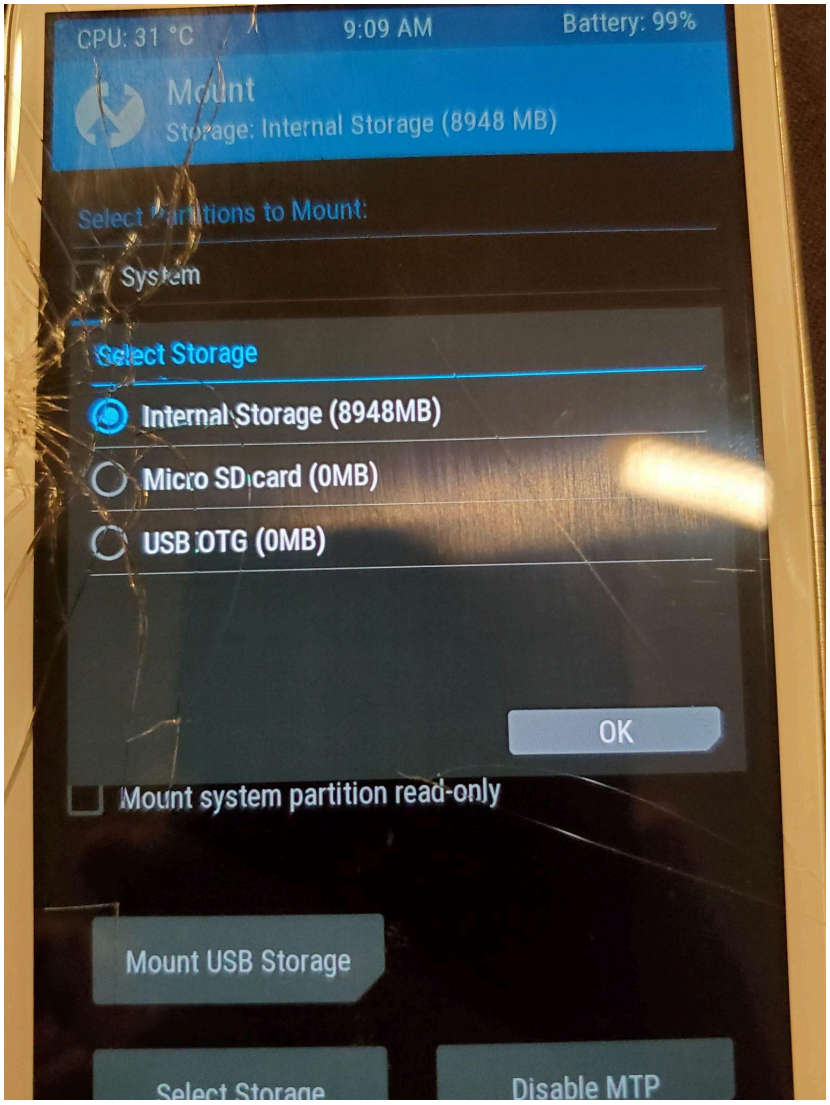
From here, I flashed TWRP to my phone using ODIN. This was how I hoped to configure my android phone to boot and install from the SD card, as this procedure is already pretty invasive.

Nicholas Roy
Samsung sIII



With TWRP on my phone, I loaded it into recovery mode (power + up volume + home, immediately after exiting download mode). This is where I realized the reason for the little to no documentation online for the flash SD card method, as my SD card was not being read, likely because of its strange formatting properties as a flash drive. This caused it to appear as empty or non-existent from the TWRP mount menu.

Nicholas Roy
Samsung sIII



Nicholas Roy
Samsung sIII

Trial 3 - Installing postmarketOS thought terminal

My final trial did not involve TWRP as I thought it would, since I still saw the value in the software. However, my potential solution broke before I could get there.

I opened a linux terminal and attempted to install through pmbootstrap.

$ git clone https://gitlab.postmarketos.org/postmarketOS/pmbootstrap.git
$ cd pmbootstrap
$ mkdir -p ~/.local/bin
$ ln -s "$PWD/pmbootstrap.py" ~/.local/bin/pmbootstrap
$ pmbootstrap --version (to make sure it works)

Once installed, I tried initializing multiple working, installable versions using different configurations each time. Including gnome and phosh interface options, as well as using edge versions and latest stable distros. After each, I ran:

$ pmbootstrap install

Which always managed to break at the same point in progress. That being a failure while creating the devices rootfs, with the exit code being 31. Looking into the log.txt file, it appears that pmbootstrap couldn't locate the release folder at some point (edge or v24.06 in my case). This error is unexpected and not documented anywhere. I tried

$ pmbootstrap zap -p

… to wipe and recreate those files.

As of writing this, I'm waiting for any results. It's taking a while…

Apparently it worked!! WE ARE SO BACK!!

$ pmbootstrap flasher flash_rootfs

On 2 separate devices, the command mentioned that my partition size is too small for given devices. Unfortunately this is all I have time to do for now. Because it broke on both devices, and because there were configuration errors in the past, I'll assume it is some sort of configuration error. That or I must correctly partition my phones for this. But that doesn't seem like a topic ever discussed online when it comes to installing custom boots like this.

Nicholas Roy
Samsung sIII

To Conclude for now

What we are doing is a very messy and invasive process. What sucks is that the documentation online is very minimal, and makes it difficult when many edge case issues appear in the procedure. Many issues arose during these procedures, many that I've seen and were able to be identified, and likely many more that weren't identified at all. A simple, easily, resolvable issue that goes unnoticed in the beginning could disrupt the entire process. I made sure to stick to only using 1 distribution for all trials, as to hopefully get a full understanding of my process as a whole, however now I am left with nothing to show for my work. For now, I was not able to successfully flash a postmarketOS system to a samsung galaxy s3, and any other distros will likely face the same fate as this one.