notes

BV

January 23, 2016

Contents

1 Info

See this page to understand the partitions.

- chrome OS ping-pongs between A and B
- it refers to `/dev/sda` but I see `/dev/mmcblk0`

See this page for ChrUbuntu install.

2 Get Debian booting from eMMC

- Get developer mode with esc-swirly-power key combo.
- ctrl-alt-f2 (top-row right arrow)
- Login `chronos` no password

```
$ sudo -s
# chromeos-firmwareupdate --mode=todev
# cgpt show /dev/mmcblk0
```

Find STATE. I have (after applying full recovery)

start 8,671,232

size 22,073,344

- Give space for a "KERN-C" partition taken from STATE

```
# cgpt add -i 1 -b 8671232 -s 22040576 -t data -l STATE /dev/mmcblk0
# cgpt add -i 6 -b 30711808 -s 32768 -t kernel -l KERN-C /dev/mmcblk0
```

Need to reboot now to make these changes stick.

```
# reboot
```

Will get spinning wheel "Your system is repairing itself. Please wait". ETA 15 minutes.

```
$ sudo -s
#
```

- check partitions again to find which has highest priority (KERN-A in my case) and that KERN-C is 32k

```
# cgpt show /dev/mmcblk0
...
2 Label: "KERN-A"
Attr: priority=1
4 Label: "KERN-B"
Attr: priority=0
6 Label: "KERN-C"
Attr: priority=0
```

- download full image from `https://wiki.debian.org/InstallingDebianOn/Samsung/ARMChromebook` and `dd` to 8GB SD
- insert SD and mount

```
# mkdir /tmp/sd-root
# mount /dev/mmcblk1p3 /tmp/sd-root
# cd /tmp/sd-root/root
# echo 'console=tty1 debug verbose root=/dev/mmcblk1p1 rootwait rw lsm.module_locking=
# dd if=/dev/mmcblk0p2 of=kern-a
# vbutil_kernel --repack newkern \
--keyblock /usr/share/vboot/devkeys/kernel.keyblock \
--version 1 \
--signprivate \
    /usr/share/vboot/devkeys/kernel_data_key.vbprivk \
--config=config \
--oldblob kern-a
# dd if=newkern of=/dev/mmcblk0p6
# cgpt add -i 6 -S 1 -P 5 /dev/mmcblk0
```

This `newkern` is only 4.5MB.
Copy from chromeos

```
# cd /tmp/sd-root
# cp -ar /lib/firmware lib/
# cp -ar /lib/modules lib/
# cp -ar /usr/share/alsa/ucm usr/share/alsa/
# cp -ar /usr/bin/alsaucm usr/bin
# chmod 755 usr/bin/alsaucm
```

Take care with the slashes.

3 ChrUbuntu

Besides being hard to spell, ChrUbuntu appears to be the easiest way to get Debian onto the internal flash.

- ChrUbuntu github
- ChrUbuntu blog with links to releases of the script
- This post with s9ryd was used initially local copy

What follows are the steps I took.

3.1 Prep on Chrome OS side

- Get into developer mode
- Create a Chromebook Recovery Media following these instructions which boil down to downloading and running $\text{linux}_{\text{recovery}}$.sh.
- Boot the recovery to apply it. This wipes system and may not be needed, but I do it just to start from a known point.
- Get through initial Chrome OS setup just enough to set network and click through to the sign in screen.
- Stop there and hit `ctrl-alt-f2` (`f2` = forward arrow) to get a console.
- Log in as user `chronos`, no pass.
- At some point in the initial playing around I did

```
chronos$ sudo chromeos-firmwareupdate --mode=todev
chronos$ sudo crossystem dev_boot_usb=1 dev_boot_legacy=1 dev_boot_signed_only=0
```

3.2 Installing ChrUbuntu

Despite what the blog says about `s9ryd` not supporting newer releases it actually is clever enough to find 14.04 when asked for "lts". This time, around let's try to install 14.10 directly. Also, I eventually want a desktop environment using MATE+Sawfish so start without any extra packages for a DE I will not use and that means `ubuntu-standard`.

```
chronos$ curl -L -O http://goo.gl/s9ryd
chronos$ sudo bash s9ryd ubuntu-standard 14.10
```

3.3 Choosing the partition size

The script asks for a storage size to reserve for Ubuntu. It uses the 3rd pair of boot/root partitions available, leaving Chrome OS's first two pairs alone. This pair is essentially empty and so it takes space from the largest partition normally used for user files. I mostly want to use this laptop for Ubuntu but want to keep the option to boot Chrome OS so I choose the recommended max of 9GB. This gives ample space for the OS and a small home area.

As soon as the answer is given the laptop reboots. Hit `ctrl-d` to get past the scary screen and it will go in to "Your system is repairing itself. Please wait". It takes a couple of minutes. Boots again back to scary boot screen. `ctrl-d` to get past.

3.4 One more time

Go though the setup screen one more time just to get network. Then `ctrl-alt-f2` and repeat the download and running:

```
chronos$ curl -L -O http://goo.gl/s9ryd
chronos$ sudo bash s9ryd ubuntu-standard 14.10
...
Chome device model is: SNOW LATITUDEAG AA-J 5406
...
```

Hit enter to continue and it goes to town. While it churns I survey the `s9ryd` script:

- partitions and formats the flash
- produces the base OS via a tarball from Ubuntu
- sets up this area as a `chroot`
- finishes up the install with `apt-get`
- copies Chrome OS kernel modules and firmware
- makes `user/user`.

Hit enter and system reboots to Ubuntu. Let's make this permanent following the last bit of guidance the `s9ryd` script gives us

```
chrubuntu$ sudo cgpt add -i 6 -P 5 -S 1 /dev/mmcblk0
```

Okay, immediate show stopper. No `wpa_supplicant` in the `s9ryd`. Let's try this again.

Fork jay0lee's repo and follow this guidance on making an easy to type URL. Commit and push and browse the latest version to get the "raw" URL then do:

```
$  curl -i http://git.io -F "url=https://raw.githubusercontent.com/brettviren/chrubunt
...
Location: http://git.io/F5zJ
...
```

GitHub is so cool!

Set the boot record to go back into ChrUbuntu:

```
$ sudo cgpt add -i 6 -P 0 -S 1 /dev/mmcblk0
$ sudo reboot
```

Then in chrome OS:

```
chronos$ curl -L -O http://git.io/F5zJ
chronos$ sudo bash F5zJ ubuntu-standard 14.10
```

Rerunning this skips the partitioning step and goes straight to formatting the root partition, downloading the core tarball and installing the system.

Okay, rebooting seems to have bricked that partition. It just goes to a black screen. Stick recovery flash in and redo the dance. This time using `s9ryd` and the xubuntu option.