

└─0x877b0 [0,0]

Contents: [Dobrica Pavlinu's random unstructured stuff]

- [Dobrica Pavlinu's random unstructured stuff \(Links\)](#)
- [Dobrica Pavlinu's random unstructured stuff \(Samsung ARM working\)](#)
- [Dobrica Pavlinu's random unstructured stuff \(bricked. recovery doesn't help\)](#)
- [Dobrica Pavlinu's random unstructured stuff \(arch\)](#)
- [Dobrica Pavlinu's random unstructured stuff \(custom firmware\)](#)
- [Dobrica Pavlinu's random unstructured stuff \(flash using raspberry pi\)](#)
 - ◆ [Dobrica Pavlinu's random unstructured stuff \(ch341a attempt\)](#)
- [Dobrica Pavlinu's random unstructured stuff \(u-boot\)](#)
 - ◆ [Dobrica Pavlinu's random unstructured stuff \(chromiumos\)](#)
- [Dobrica Pavlinu's random unstructured stuff \(servo debug header\)](#)
- [Dobrica Pavlinu's random unstructured stuff \(spi flash layout\)](#)
- [Dobrica Pavlinu's random unstructured stuff \(enable development mode from recovery\)](#)
- [Dobrica Pavlinu's random unstructured stuff \(upstream u-boot flashing\)](#)
- [Dobrica Pavlinu's random unstructured stuff \(building chromiumos\)](#)
- [Dobrica Pavlinu's random unstructured stuff \(building u-boot\)](#)
- [Dobrica Pavlinu's random unstructured stuff \(boot resistors \(boot from sd card\)\)](#)
- [Dobrica Pavlinu's random unstructured stuff \(coreboot\)](#)

Links

- <https://github.com/hugegreenbug/xf86-input-cmt>
- <https://www.chromium.org/chromium-os/developer-information-for-chrome-os-devices/samsung-arm>

I have two chromebooks, one is working and one is bricked

Samsung ARM Chromebook (working)

```
localhost ~ # cat /proc/cpuinfo
processor       : 0
model name     : ARMv7 Processor rev 4 (v7l)
BogoMIPS      : 48.00
Features       : swp half thumb fastmult vfp edsp thumbee neon vfpv3 tls vfpv4 idiva idivt
CPU implementer : 0x41
CPU architecture: 7
CPU variant    : 0x0
CPU part       : 0xc0f
CPU revision   : 4

processor       : 1
model name     : ARMv7 Processor rev 4 (v7l)
BogoMIPS      : 48.00
Features       : swp half thumb fastmult vfp edsp thumbee neon vfpv3 tls vfpv4 idiva idivt
CPU implementer : 0x41
CPU architecture: 7
CPU variant    : 0x0
CPU part       : 0xc0f
CPU revision   : 4

Hardware       : SAMSUNG EXYNOS5 (Flattened Device Tree)
Revision      : 0000
```

```
Serial          : 0000000000000000
```

```
localhost ~ # flashrom --flash-name  
flashrom v0.9.4 : 244249c : Dec 09 2016 03:49:59 UTC on Linux 3.8.11 (armv7l)  
flashrom v0.9.4 : 244249c : Dec 09 2016 03:49:59 UTC on Linux 3.8.11 (armv7l)  
vendor="Winbond" name="W25Q32DW"
```

<https://media.digikey.com/pdf/Data%20Sheets/Winbond%20PDFs/W25Q32DW.pdf>

```
localhost ~ # flashrom --get-size  
4194304
```

```
localhost ~ # flashrom --wp-status  
WP: status: 0x00b8  
WP: status.srp0: 1  
WP: status.srp1: 0  
WP: write protect is enabled.  
WP: write protect range: start=0x00000000, len=0x00200000
```

```
localhost Downloads # flashrom -r chromebook-spi.rom  
Block protection could not be disabled!  
Reading flash... SUCCESS
```

bricked, recovery doesn't help

This chromebook reports that it wants to do recovery, but inserting (few different) usb sticks generated using <http://www.google.com/chromeos/recovery> instructions doesn't help

https://dl.google.com/dl/edgedl/chromeos/recovery/linux_recovery.sh

arch

[https://wiki.archlinux.org/index.php/Samsung_Chromebook_\(ARM\)](https://wiki.archlinux.org/index.php/Samsung_Chromebook_(ARM))

```
setenv bootargs root=/dev/mmcblk1p2 rootfstype=jfs rootwait rw  
mmc dev 1  
ext2load mmc 1:1 42000000 vmlinux.uimg  
bootm 42000000
```

custom firmware

- <https://www.chromium.org/chromium-os/developer-information-for-chrome-os-devices/custom-firmware>
- http://selinuxproject.org/~jmorris/lss2013_slides/safford_chromebook_takeown.pdf
- <https://www.chromium.org/chromium-os/firmware-porting-guide/using-nv-u-boot-on-the-samsung-arm>

```
dpavlin@nuc:/nuc/books/Chromebook$ wget http://commondatastorage.googleapis.com/chromeos-localmirror/2017-01-19/19:05:28--  
http://commondatastorage.googleapis.com/chromeos-localmirror/distfiles/n  
Resolving commondatastorage.googleapis.com (commondatastorage.googleapis.com)... 216.58.206.16, 216.58.206.16  
Connecting to commondatastorage.googleapis.com (commondatastorage.googleapis.com) [216.58.206.16]:
```

```
HTTP request sent, awaiting response... 200 OK
Length: 281844 (275K) [application/octet-stream]
Saving to: â nv_uboot-snow.kpart.bz2â
```

```
nv_uboot-snow.kpart.bz2      100%[=====>] 275.24K  1.
2017-01-19 19:05:28 (1.37 MB/s) - â nv_uboot-snow.kpart.bz2â  saved [281844/281844]
```

This might be upstream bios update, but u-boot doesn't start at beginning, so it's probably some update format and not raw image.

- <https://blogs.s-osg.org/use-mainline-u-boot-non-signed-kernels-exynos-chromebooks/>

flash using raspberry pi

unplug battery before attempting this! rpi3 can power flash memory

```
# backup
```

```
root@rpi3:/home/pi/flashrom-0.9.9# ./flashrom -p linux_spi:dev=/dev/spidev0.0 -r chromeboot-brick
flashrom v0.9.9-r1955 on Linux 4.4.27-v7+ (armv7l)
flashrom is free software, get the source code at https://flashrom.org
```

```
Calibrating delay loop... OK.
Found GigaDevice flash chip "GD25LQ32" (4096 kB, SPI) on linux_spi.
Unsetting lock bit(s) failed.
Reading flash... done.
```

file:///nuc/books/Chromebook/GD25LQ32_Rev1.3.pdf

I repeated this twice and checked md5sum of both files to verify that I have stable connection with programmer

```
# flash version from another chromebook
```

```
root@rpi3:/home/pi# time ./flashrom-0.9.9/flashrom -p linux_spi:dev=/dev/spidev0.0 -w chromebook-
flashrom v0.9.9-r1955 on Linux 4.4.27-v7+ (armv7l)
flashrom is free software, get the source code at https://flashrom.org
```

```
Calibrating delay loop... OK.
Found GigaDevice flash chip "GD25LQ32" (4096 kB, SPI) on linux_spi.
Unsetting lock bit(s) failed.
Reading old flash chip contents... done.
Erasing and writing flash chip... FAILED at 0x00002000! Expected=0xff, Found=0x14, failed byte co
ERASE FAILED!
Reading current flash chip contents... done. Looking for another erase function.
FAILED at 0x00000000! Expected=0xff, Found=0xa3, failed byte count from 0x00000000-0x00007fff: 0x
ERASE FAILED!
Reading current flash chip contents... done. Looking for another erase function.
FAILED at 0x00000000! Expected=0xff, Found=0xa3, failed byte count from 0x00000000-0x0000ffff: 0x
ERASE FAILED!
Reading current flash chip contents... done. Looking for another erase function.
FAILED at 0x00000000! Expected=0xff, Found=0xa3, failed byte count from 0x00000000-0x003fffff: 0x
ERASE FAILED!
Reading current flash chip contents... done. Looking for another erase function.
```

```
No usable erase functions left.
FAILED!
Uh oh. Erase/write failed. Checking if anything has changed.
Reading current flash chip contents... done.
Good, writing to the flash chip apparently didn't do anything.
Please check the connections (especially those to write protection pins) between
the programmer and the flash chip. If you think the error is caused by flashrom
please report this on IRC at chat.freenode.net (channel #flashrom) or
mail flashrom@flashrom.org, thanks!
```

```
real    14m40.938s
user    0m1.600s
sys     0m9.000s
```

```
# hm?!
root@rpi3:/home/pi# time ./flashrom-0.9.9/flashrom -p linux_spi:dev=/dev/spidev0.0 -r chromebook-
flashrom v0.9.9-r1955 on Linux 4.4.27-v7+ (armv7l)
flashrom is free software, get the source code at https://flashrom.org
```

```
Calibrating delay loop... OK.
Found GigaDevice flash chip "GD25LQ32" (4096 kB, SPI) on linux_spi.
Unsetting lock bit(s) failed.
Reading flash... done.
```

```
real    1m19.653s
user    0m1.140s
sys     0m0.700s
root@rpi3:/home/pi# md5sum chromeboot-bricked.rom chromebook-broken.rom.3
1ef1d2230c27624661a86b57064057cb  chromeboot-bricked.rom
1ef1d2230c27624661a86b57064057cb  chromebook-broken.rom.3
```

```
# It really didn't change!
```

ch341a attempt

```
root@rpi3:/home/pi# flashrom --programmer ch341a_spi -r ch341a-snow-broken-spi.rom
flashrom v0.9.9-r1955 on Linux 4.4.27-v7+ (armv7l)
flashrom is free software, get the source code at https://flashrom.org
```

```
Calibrating delay loop... OK.
Found GigaDevice flash chip "GD25LQ32" (4096 kB, SPI) on ch341a_spi.
Reading flash... done.
```

```
root@rpi3:/home/pi# md5sum ch341a-snow-broken-spi.rom
1ef1d2230c27624661a86b57064057cb  ch341a-snow-broken-spi.rom
```

```
root@rpi3:/home/pi# md5sum *.rom
1ef1d2230c27624661a86b57064057cb  ch341a-snow-broken-spi.rom
11e616f5dcf18d775f6484b78953ada0  chromebook-spi.rom
1ef1d2230c27624661a86b57064057cb  chromeboot-bricked.rom
```

```
root@rpi3:/home/pi# time flashrom --programmer ch341a_spi -w chromebook-spi.rom
flashrom v0.9.9-r1955 on Linux 4.4.27-v7+ (armv7l)
flashrom is free software, get the source code at https://flashrom.org
```

```
Calibrating delay loop... OK.
Found GigaDevice flash chip "GD25LQ32" (4096 kB, SPI) on ch341a_spi.
Reading old flash chip contents... done.
Erasing and writing flash chip... Erase/write done.
Verifying flash... VERIFIED.
```

```
real    5m21.170s
```

```
user    0m48.300s
sys     0m18.330s
```

https://dl.google.com/dl/edgedl/chromeos/recovery/linux_recovery.sh

https://dl.google.com/dl/edgedl/chromeos/recovery/chromeos_8743.85.0_daisy_recovery_stable-channel_s

u-boot

```
dpavlin@klin:/klin$ git clone git://git.denx.de/u-boot.git
Cloning into 'u-boot'...
remote: Counting objects: 442911, done.
remote: Compressing objects: 100% (77033/77033), done.
remote: Total 442911 (delta 360952), reused 440666 (delta 358805)
Receiving objects: 100% (442911/442911), 93.64 MiB | 11.14 MiB/s, done.
Resolving deltas: 100% (360952/360952), done.
```

```
git remote add u-boot-samsung git://git.denx.de/u-boot-samsung.git
git checkout u-boot-samsung/master -b u-boot-samsung/master
```

```
dpavlin@klin:/klin/u-boot$ find . -name 'snow*'
./include/configs/snow.h
./configs/snow_defconfig
```

```
export CROSS_COMPILE="arm-none-eabi-" ARCH=arm
```

```
dpavlin@klin:/klin/u-boot$ make snow_defconfig
HOSTCC  scripts/basic/fixdep
HOSTCC  scripts/kconfig/conf.o
SHIPPED scripts/kconfig/zconf.tab.c
SHIPPED scripts/kconfig/zconf.lex.c
SHIPPED scripts/kconfig/zconf.hash.c
HOSTCC  scripts/kconfig/zconf.tab.o
HOSTLD  scripts/kconfig/conf
#
# configuration written to .config
#
```

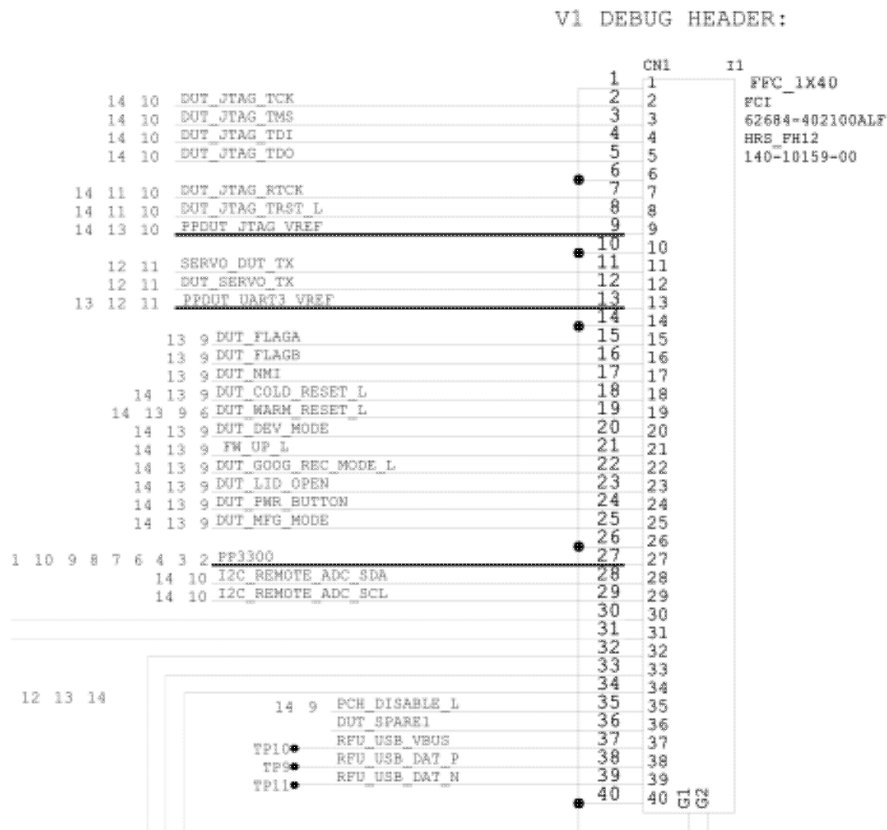
```
make
```

chromiumos

```
dpavlin@klin:/klin/chromebook/u-boot$ git remote -v
origin  https://chromium.googlesource.com/chromiumos/third_party/u-boot/ (fetch)
origin  https://chromium.googlesource.com/chromiumos/third_party/u-boot/ (push)
```

```
dpavlin@klin:/klin/chromebook/u-boot$ git checkout -b firmware-snow-2695.B remotes/origin/firmware-snow-2695.B
Branch firmware-snow-2695.B set up to track remote branch firmware-snow-2695.B from origin.
```

servo debug header



spi flash layout

```
dpavlin@nuc:/nuc/flashmap$ git remote -v
origin https://github.com/dhendrix/flashmap (fetch)
origin https://github.com/dhendrix/flashmap (push)
```

```
dpavlin@nuc:/nuc/chromebook/flashmap$ ./fmap_decode /nuc/books/Chromebook/spi/snow-spi.rom | sort
area_offset="0x00000000" area_size="0x00002000" area_name="BL1 PRE_BOOT" area_flags_raw="0x01" ar
area_offset="0x00002000" area_size="0x00004000" area_name="BL2 SPL" area_flags_raw="0x01" area_fl
area_offset="0x00006000" area_size="0x0009a000" area_name="U_BOOT" area_flags_raw="0x01" area fla
area_offset="0x000a0000" area_size="0x00001000" area_name="FMAP" area_flags_raw="0x01" area_flags
area_offset="0x000b0000" area_size="0x000ef000" area_name="GBB" area_flags_raw="0x01" area_flags=
area_offset="0x00200000" area_size="0x00002000" area_name="VBLOCK_A" area_flags_raw="0x01" area_f
area_offset="0x00202000" area_size="0x000edf00" area_name="FW_MAIN_A" area_flags_raw="0x01" area_
area_offset="0x00300000" area_size="0x00002000" area_name="VBLOCK_B" area_flags_raw="0x01" area_f
area_offset="0x00302000" area_size="0x000edf00" area_name="FW_MAIN_B" area_flags_raw="0x01" area_
area_offset="0x003f8000" area_size="0x00004000" area_name="SHARED_DATA" area_flags_raw="0x01" are
fmap_signature="0x5f5f464d41505f5f" fmap_ver_major="1" fmap_ver_minor="0" fmap_base="0x0000000000
```

enable development mode from recovery

<https://www.chromium.org/chromium-os/developer-information-for-chrome-os-devices/workaround-for-batter>

```
(parted) unit b
(parted) print
Model: MEM Drive Mini Metal (scsi)
Disk /dev/sdc: 2021654528B
Sector size (logical/physical): 512B/512B
Partition Table: gpt
Disk Flags:
```

Number	Start	End	Size	File system	Name	Flags
11	32768B	8421375B	8388608B		RWFW	
6	8421376B	8421887B	512B		KERN-C	
7	8421888B	8422399B	512B		ROOT-C	
9	8422400B	8422911B	512B		reserved	
10	8422912B	8423423B	512B		reserved	
2	10485760B	27262975B	16777216B		KERN-A	
4	27262976B	44040191B	16777216B		KERN-B	
8	44040192B	60817407B	16777216B	ext4	OEM	msftdata
12	127926272B	144703487B	16777216B	fat16	EFI-SYSTEM	boot, esp
5	144703488B	146800639B	2097152B		ROOT-B	
3	146800640B	1499463679B	1352663040B	ext2	ROOT-A	
1	1499463680B	1518338047B	18874368B	ext2	STATE	msftdata

```
(parted) quit
dpavlin@nuc:/nuc/flashmap$ ./enable_rw_mount.sh /dev/sdc 146800640B
./enable_rw_mount.sh: line 9: 146800640B: value too great for base (error token is "146800640B")
enable_rw_mount called on non-ext2 filesystem: /dev/sdc 146800640B
dpavlin@nuc:/nuc/flashmap$ ./enable_rw_mount.sh /dev/sdc 146800640
dpavlin@nuc:/nuc/flashmap$ sudo mount /dev/sdc3 /tmp/sdc3/
```

And this doesn't help to get recovery on broken cromebook working

upstream u-boot flashing

```
oot@rpi3:/home/pi# cp chromebook-spi.rom spi-mix.rom && dd conv=notrunc if=u-boot-nodtb.bin of=spi-mix.rom
1163+1 records in
1163+1 records out
595496 bytes (595 kB) copied, 0.0177353 s, 33.6 MB/s
```

```
root@rpi3:/home/pi# time flashrom --programmer ch341a_spi --layout snow.layout2 --image U_BOOT --
flashrom v0.9.9-r1955 on Linux 4.4.27-v7+ (armv7l)
flashrom is free software, get the source code at https://flashrom.org
```

```
Using region: "U_BOOT".
Calibrating delay loop... OK.
Found GigaDevice flash chip "GD25LQ32" (4096 kB, SPI) on ch341a_spi.
Reading old flash chip contents... done.
Erasing and writing flash chip... Erase/write done.
Verifying flash... VERIFIED.
```

```
real    3m31.555s
user    0m21.310s
sys     0m13.660s
```

building chromiumos

- <https://www.chromium.org/chromium-os/developer-guide>

```
dpavlin@klin:/klin/chromebook$ . env.sh
```

```
dpavlin@klin:/klin/chromebook/chromiumos$ cros_sdk --download
```

```
(cr) ((971c906...)) dpavlin@klin ~/trunk/src/scripts $ export BOARD=daisy
```

```
(cr) ((971c906...)) dpavlin@klin ~/trunk/src/scripts $ ./set_shared_user_password.sh  
Enter password for shared user account: Password set in /etc/shared_user_passwd.txt
```

```
(cr) ((971c906...)) dpavlin@klin ~/trunk/src/scripts $ ./build_packages --board=${BOARD}
```

breaks

building u-boot

- <https://www.chromium.org/chromium-os/firmware-porting-guide/using-nv-u-boot-on-the-samsung-arm>

```
BOARD=daisy  
FDT=snow
```

```
cros_workon --board=${BOARD} start chromeos-u-boot
```

```
emerge-${BOARD} chromeos-ec chromeos-u-boot chromeos-bootimage
```

boot resistors (boot from sd card)

- <https://archlinuxarm.org/forum/viewtopic.php?f=27&t=4016&start=120#p30291>

coreboot

- <https://www.coreboot.org/Exynos5>

```
dpavlin@klin:/klin/coreboot/util/crossgcc$ ./buildgcc -p armv7a-eabi -j 4
```