Requirements

1. An SD Card of at least 8GB. (It must be an SD Card for this to work, no USB.)
2. A computer (not the Chromebook) that has Linux System, Linux VM, or Windows Subsystem for Linux (WSL) Installation
    a. WSL Installation
    b. If you plan to use WSL, you must connect your SD Card to the WSL Subsystem by following https://devblogs.microsoft.com/commandline/connecting-usb-devices-to-wsl/
    c. If you use a Linux VM, you must connect your SD Card to the VM Directly.

Setup before Creating the SD Card

1. On your Linux VM you must install Python3.6+, OpenSSL, Git, and pip3.
    a. On Ubuntu, this would be running `sudo apt update && sudo apt install python3 python3-pip openssl git`
2. Using pip3, run `pip3 install --user pmbootstrap`
3. Allow to finish.
4. Connect your SD Card at this time.
5. Figure out what drive designation is your SD Card (in the form of /dev/sdX or /dev/mmcblkX)
    a. To do this, you may install gparted (sudo apt install gparted). Run it and see what's the letter designation. **USING THE WRONG DRIVE DESIGNATION WILL NUKE YOUR OWN DEVICE. MAKE SURE YOU USE THE RIGHT DESIGNATION**.

Initializing the "Setup" SD Card (You only have to do this once)

- Note: For the sake of not going insane, I'll be using the designation of `**/dev/sd9**` for anywhere I need to input the raw drive path to the SD Card. **DO NOT USE THIS AS A COPY PASTE. YOU MUST SUBSTITUTE THE TEMPLATIZED DESIGNATION WITH THE ONE ON YOUR SYSTEM.**
- **If you have problems running these commands because you have the error "Allocating guest commpage: Out of memory", please follow the instructions here.**

1. As your user account, run `pmbootstrap init`
        i. If you are unable to run this because it was not found, try `~/.local/bin/pmbootstrap init`
    b. When prompted for the "Work Path", leave it as default.

        i. If you do not have enough space on your local drive for this work path, you may select a location that has more space by typing out that location here.
    c. When prompted for “Channel”, leave it as default (‘edge’)
    d. When prompted for “Vendor”, use “google”
    e. When prompted for “Device Codename”, use “snow”
    f. When prompted for “Kernel”, use “rev5”
    g. When promoted for the package `device-google-snow-nonfree-firmware`, leave it as default (‘y’)
    h. When promoted for the Username, use `eqoptech` (this is for our setup disk)
    i. When promoted for the “User Interface”, use `mate`
    j. When promoted for the “Additional Options”, leave as default (‘n’)
    k. When promoted for the “Additional Packages”, type `firefox, py3-pip,python3,git,openssl,procps,cryptsetup,util-linux,parted,pmbootstrap`
    l. When prompted for the “Timezone”, leave default (`y`)
    m. When promoted for the “Default Locale”, type `en_US.UTF-8`
    n. When promoted for the Device Hostname, leave as default (`google-snow`)
    o. When promoted for the SSH Keys, leave default (`n`)
    p. When prompted for “Build Outdated Packages”, leave as default (`y`)
2. After you see the “DONE!” in Green letters, you can now run `pmbootstrap install --sdcard=/dev/sd9`
    a. This will now install your Setup Disk to your SD Card. It’ll take about 15 minutes.
    b. When prompted for sudo passwords, enter your actual sudo password for your device.
    c. When prompted for “New Password”, enter `eqoptech` (this is the password for the SD Card, not your local system)
    d. When prompted for “EVERYTHING ON /dev/sd9 WILL BE ERASED!”, make sure that it’s the correct device, and proceed with ‘y’
3. After the install is complete, flush everything out via `pmbootstrap shutdown`

Fixing the SD Card so that we can Boot to GUI (Only Once)

Due to an upstream package breakage for certain libraries, we must manually downgrade certain packages so that we can get to a UI. We won’t have Internet on our Chromebook until we get into UI, so we need to get those packages now.

1. Remove the SD card and insert it back again. The SD card should then be mounted, e.g. to /media/$USER/pmOS_root. Verify this with:

    ```
    $ df -h
    ```

2. Download the working version of the MESA Packages (21.3.8-r1):
    a. `cd </path/to/mounted/sdcard>/home/eqoptech`
    b. `sudo wget https://pastebin.com/raw/nLiAbmhp -O get.sh`

```
c. sed -e "s/^M//g" get.sh > ~/get.sh
    i.   To enter ^M, type CTRL-V, then CTRL-M. That is, hold
         down the CTRL key then press V and M in succession.
d. sudo rm get.sh
e. sudo mv ~/get.sh get.sh
f. sudo chmod +x get.sh
g. sudo chown 10000:10000 get.sh
h. ./get.sh
i. sudo chown 10000:10000 *.apk
j. cd ~
k. sudo umount </path/to/mounted/sdcard>
```

Getting the Chromebook Ready (Once Per Device)

Requirements

- An XE303C12 Chromebook. (This guide is written for this specific Chromebook)
    - The chromebook cannot be under Google Chrome Device Management Policy. If it is, you must request that the chromebook be deprovisioned from the owning Google Apps for Education Tenant FIRST.
        - You can check whether a chromebook is managed by logging in (use email:eqoptechtesting@gmail.com pw:eqoptechusertesting) and click on the clock in the bottom right. If the popup has the text “Managed”, “This Device is Managed by XXX”, or “Chromebook is Managed by XXX”, This method **WILL NOT WORK.**
            - **If you have a managed device, please notify Terence or Lucy on Slack.**
                - If you are able to, please report with the Serial Number (On the Bottom of the Device) and the domain it is attached to (mvusd.org, mvla.net, lasdk8.org, etc…)

Getting to Developer Mode

1. To get to Developer Mode, you’ll need to do the Three-Finger Salute (Press ESC + Refresh + Power at the same time.)
2. You’ll be presented with a screen saying you are in recovery mode. Press CTRL+D now.
    a. You will be prompted to confirm entering developer mode. Confirm this action by pressing Enter.
3. The Chromebook will Reboot, and present you with a screen saying “OS Verification is Off”. Press CTRL+D Now.

    a. The chromebook will promptly Powerwash itself to “Prepare to go into Developer Mode”
    b. If it complains about policy, you’ll need to get the Device Management cleared. Please ping Randall or Terence on Slack in the #archlinux channel and we will assist you.
4. Once started, sign in back to the ChromeBook. You are now in developer mode.

Tell the Chromebook to Boot From External Media

The chromebook needs to be set up to boot from External Media, since that’s how we are going to install the OS.

1. Press CTRL+ALT+T to open up crosh, The Chrome OS Shell.
2. At the prompt, type “shell” and press enter. You should now see a prompt like `chronos@localhost ~`
3. Run `sudo crossystem dev_boot_usb=1 dev_boot_signed_only=0` to tell the Chromebook to Allow Booting from USB (and unsigned software, which is what we are doing)
4. Run `sudo chromeos-firmwareupdate --mode=todev` to update the firmware. (If you get an error, ignore it. This step is needed for some os versions but not needed for later versions)
5. Reboot the Device.

Booting the Chromebook on the SD Card

1. Plug in your freshly created SD Card to the Left Side of the Chromebook. (There’s a flap that’ll fold down to allow it to enter)
2. Boot the Chromebook
3. At the OS Verification Screen, press CTRL+U.
4. Wait until you get to the MATE Desktop before touching ANYTHING.

Fix pmOS MATE Desktop on SD Card (Only Once per New SD Card)

1. Press Ctrl-Alt-F3 (i.e. Ctrl-Alt-Refresh) to go to a console terminal (In fact, any of Ctrl-Alt-F1 to Ctrl-Alt-F6 should work, too).
2. Install working mesa packages to pmOS on SD card
    a. Log in as eqoptech from the console (eqoptech:eqoptech)
    b. Install the working version of mesa packages (ver 21.3.8-r1) onto the pmOS installation on the SD card:
    ```
    $ cd /home/eqoptech
    $ sudo apk add llvm13-libs*.apk mesa*-21.3.8-r1.apk
    ```
3. Run “`sudo startx`”, ***The MATE desktop should show up.***

Installing to the Actual Chromebook

Checking your CPU Revision

A critical step is that you will have to check your CPU revision to install the correct end-user kernel.

1. Open a Terminal
2. Run `sudo dmesg | grep "ARMv7 Processor"`
3. If your output is:

   [0.000000] CPU: ARMv7 Processor | [] revision 4 (ARMv7), cr=

   You must use “rev4”
4. If your output is:

   [0.000000] CPU: ARMv7 Processor | [] revision 5 (ARMv7), cr=

   You must use “rev5”

Setting Up the Install

1. Note: throughout the process, you may be prompted for a sudo password, enter the sudo password set up earlier (`eqoptech`)
2. Connect to the Internet before Installing. This is critical for a future step in the process.
3. As your user account, run `pmbootstrap init`
    i. If you are unable to run this because it was not found, try `~/.local/bin/pmbootstrap init`
    b. When prompted for the “Work Path”, leave it as default.
        i. If you do not have enough space on your local drive for this work path, you may select a location that has more space by typing out that location here.
    c. When prompted for “Channel”, leave it as default (‘edge’)
    d. When prompted for “Vendor”, use “google”
    e. When prompted for “Device Codename”, use “snow”
    f. When prompted for “Kernel”, use whatever you found out in the “Checking your CPU Revision” section
    g. When promoted for the package `device-google-snow-nonfree-firmware`, leave it as default (‘y’)
    h. When promoted for the Username, leave as default (`user`)

i. When promoted for the “User Interface”, use `mate`
j. When promoted for the “Additional Options”, leave as default (‘n’)
k. When promoted for the “Additional Packages”, type `firefox`
l. When promoted for the “Default Locale”, type `en_US.UTF-8`
m. When promoted for the Device Hostname, DO NOT LEAVE AS DEFAULT.
    i. In another terminal window, run `ifconfig` and get the HWAddr from mlan0. Remove all Colons from the Address.
    ii. Set the hostname to be `chromebook-<macaddresshere>`
    iii. An example would be `chromebook-147dc5e2ffba`
n. When prompted for “Build Outdated Packages”, leave as default (`y`)

Actually Installing To The Chromebook (The Point of No Return)

1. After init is complete, run `pmbootstrap install --sdcard=/dev/mmcblk0`
    a. This will now install your Setup Disk to the chromebook. It’ll take about 30 minutes.
    b. When prompted for sudo passwords, enter ‘eqoptech’
    c. When prompted for “New Password”, enter `user` (This is the password for the Chromebook!)
    d. When prompted for “EVERYTHING ON /dev/mmcblk0 WILL BE ERASED!”, make sure that it’s the correct device (it should be), and proceed with ‘y’
2. Remember to do `pmbootstrap shutdown` after the install is complete! But we are not done.

Fixing the Install on the Chromebook

Due to the aforementioned package breakage, we now need to install those packages again onto the actual laptop itself. But instead of having to boot into the laptop, we can do that from within our own current environment.

1. Get the root disk mounted.
    a. `$ sudo mkdir -p /media/rootdisk`
    b. `$ sudo mount /dev/mmcblk0p3 /media/rootdisk`
2. Copy the APKs to the EMMC
    a. `$ sudo mkdir -p /media/rootdisk/apkinstall`
    b. `$ sudo cp /home/eqoptech/llvm13-libs*.apk /media/rootdisk/apkinstall`
    c. `$ sudo cp /home/eqoptech/mesa*-21.3.8-r1.apk /media/rootdisk/apkinstall`
3. CHROOT into the EMMC. (**We are now modifying the chromebook’s flash, tread carefully**)
    a. `$ sudo chroot /media/rootdisk # After this point, our root will be within the actual chromebook emmc flash. Tread carefully.`
4. Install the APKs
    a. `# cd /apkinstall`

    b. `# sudo apk add llvm13-libs*.apk mesa*-21.3.8-r1.apk`
    c. `# cd /`
    d. `# sudo rm -rf /apkinstall # Remove all APKs we left on emmc.`
    e. `# exit`
5. Unmount the EMMC.
    a. `$ cd /`
    b. `$ sudo umount /media/rootdisk`
        i. If this returns an error saying “Device is Busy”, ensure all terminal windows are closed and try again.

Verification

Boot to the Chromebook in PMOS

1. Once the install is complete, Shutdown the chromebook.
2. Remove the SD Card, use it for another Chromebook.
3. Boot the Chromebook
4. Press CTRL+D at the OS Verification Screen.
5. Observe that the Chromebook does boot to the OS.

Setup Time Zone

1. Connect to the Internet
2. Run `sudo setup-timezone`
3. Type “America/Los_Angeles” as the Timezone
4. Run `ntpd -q -p 0.us.pool.ntp.org`

Pin MESA Package Versions

Please pin the MESA Package Versions so that we don’t accidentally brick devices by updating.

1. Open a Terminal from Applications
2. Run the following commands within the terminal
    a. `$ sudo apk update`
    b. `$ sudo apk add mesa=21.3.8-r1`
    c. `$ sudo apk add mesa-dri-gallium=21.3.8-r1`
    d. `$ sudo apk add mesa-egl=21.3.8-r1`
    e. `$ sudo apk add mesa-gbm=21.3.8-r1`
    f. `$ sudo apk add mesa-gl=21.3.8-r1`
    g. `$ sudo apk add mesa-glapi=21.3.8-r1`
    h. `$ sudo apk add mesa-gles=21.3.8-r1`

    i. `$ sudo apk upgrade`
3. Reboot the System Now.

Fix Speaker Audio

Copy the files from Appendix (fix-snow-sound.sh, Snow-I2S-MAX98090/HiFi.conf, Snow-I2S-MAX98090/Snow-I2S-MAX98090.conf, Snow-I2S-MAX98095/HiFi.conf, Snow-I2S-MAX98095/Snow-I2S-MAX98095.conf) to “scripts” directory on an SD card or a USB drive, keeping the directory structure.

Insert the SD card or USB drive to the Chromebook and mount it, e.g.

```
$ sudo mount /dev/mmcblk1p3 /media/usb
```

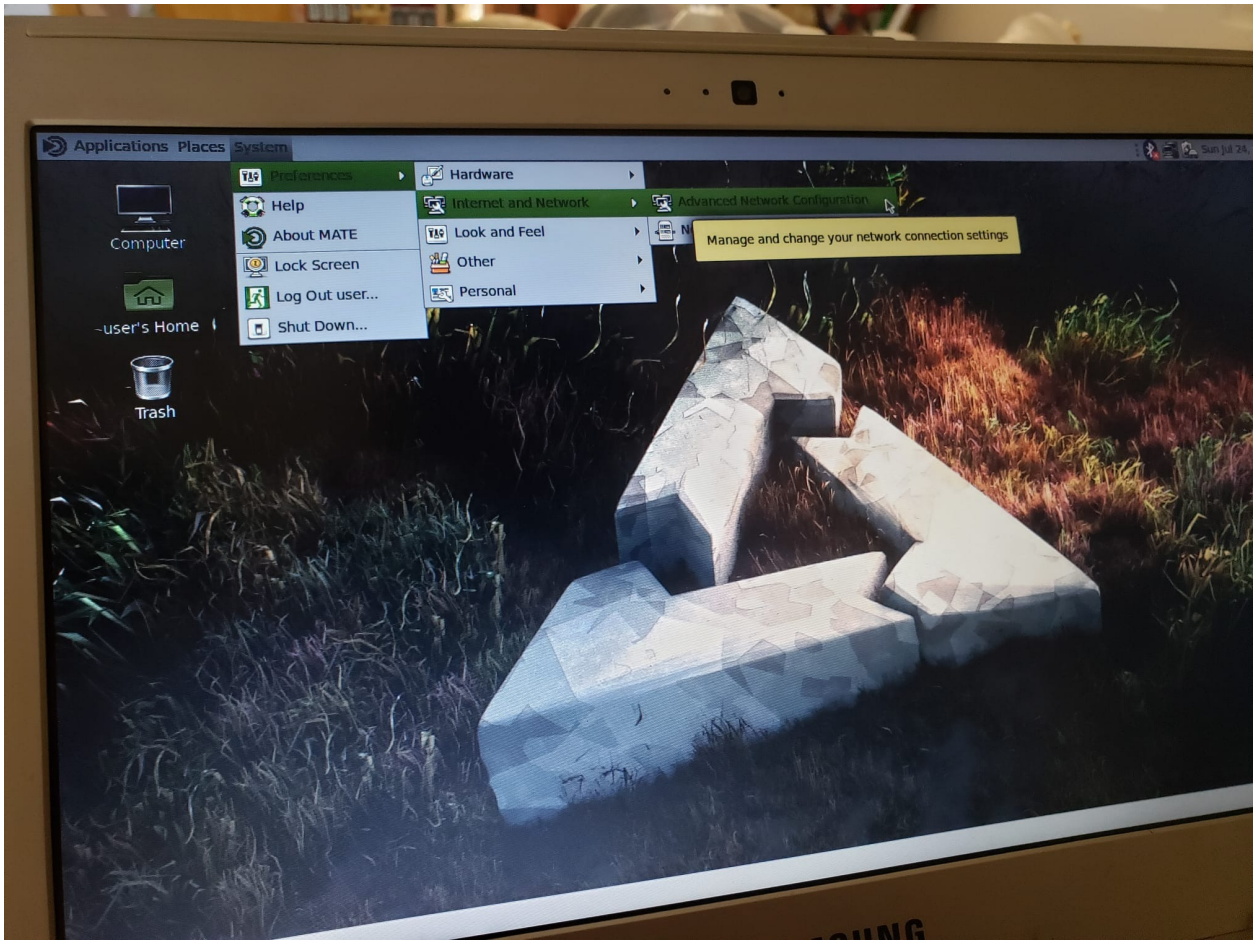
Run fix-snow-sound.sh to fix the speaker audio configuration:

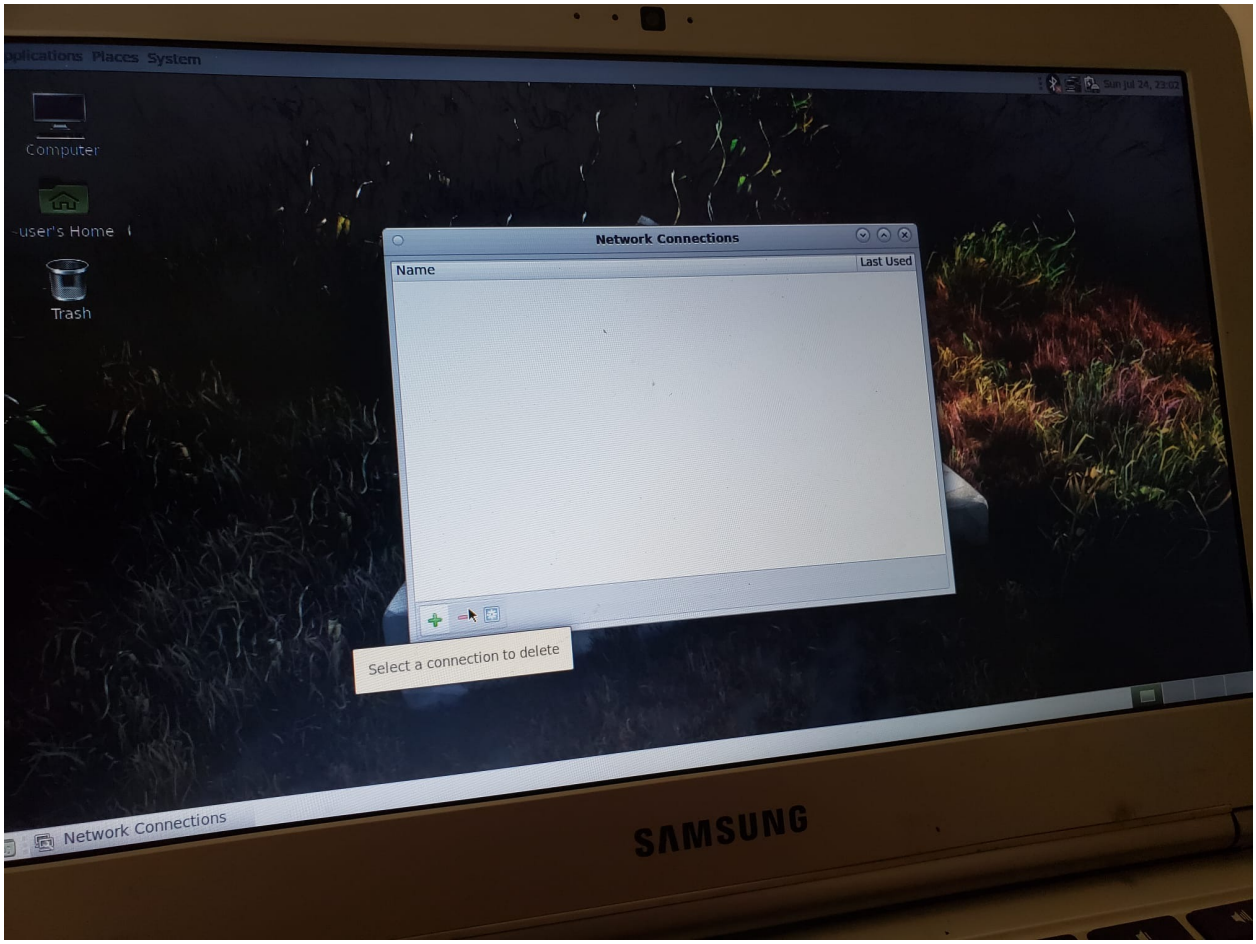
```
$ sudo chmod +x /media/usb/scripts/fix-snow-sound.sh

$ sudo /media/usb/scripts/fix-snow-sound.sh
```

Reboot

Go into settings and forget your wireless network once you are done with the chromebook.


Applications Places System
Preferences
Help
About MATE
Lock Screen
Log Out user...
Shut Down...
Hardware
Internet and Network
Look and Feel
Other
Personal
Advanced Network Configuration
Manage and change your network connection settings
Computer
user's Home
Trash



Video Tutorial

Creating SD Card

Installing OS (Ear Warning, sorry about the Fan Noise. This video is mostly out of date with some steps that are necessary to the process added in. You may use this as a starter guide, but it is not the full instructions in video format.)