Chromebook Pixel

This HOWTO covers installing and configuring Slackware (tested with -current/14.2) on a 2015 Chromebook Pixel. Older versions of Slackware (older kernel) and the 2013 Chromebook Pixel may run into issues. There are a couple main sources of information for running Linux on the Pixel (see Sources); for the best experience under Slackware, we'll need to use a combination of each guide.

Preparation

Before installing Slackware on your Chromebook Pixel, you will need to enable developer mode and allow for USB boot devices.

Developer Mode

From ChromeOS, hold down ESC and F3 (the refresh key), and press the power button. This will reboot the device into Recovery Mode. From there, press Ctrl-D, and confirm that you wish to enable developer mode. Your device will reboot.

Once Developer Mode is enabled, you will see a white screen with a warning on every boot. To boot into Chrome OS normally, press Ctrl-D. To view the boot menu, press Ctrl-L. Pressing Space will prompt you to reset the Chromebook and replace Chrome OS.

Legacy Boot

Back in Chrome OS, you'll need to enter shell (Ctrl-Alt-T), and enter the following:

```
> shell
> sudo bash
> crossystem dev_boot_usb=1 dev_boot_legacy=1
```

From here, prepare your USB installer with Slackware and reboot the device. Make sure to press Ctrl-L to enter the boot menu, and choose your USB device.

Installation

When you reboot and choose to boot the Slackware installer, make sure to enter the following at the Slackware boot prompt:

```
huge.s vga=0x340
```

This will ensure a normal display for the installation. From here, follow the standard Slackware installation procedure.

It is recommended to add “libata.force=noncq” to the append section of your lilo.conf. Additionally,

you can add “video=1600×1062” or “video=1920×1275” if you'd prefer a more readable console resolution.

After completing the installation, reboot, press Ctrl-L to view the boot menu, and choose the new partition listed. You should now be able to boot into your Slackware installation.

Drivers

Out of the box, you will find that the touchpad and sound do not work. Wifi, the HiDPI display and touchscreen should all be picked up natively. It is strongly recommended that you build the linux-samus kernel for the best compatibility. You can find the source here: https://github.com/raphael/linux-samus.

Run the following to download, build, and install the new kernel (which is currently 4.4.6, despite being named 4.4.2ph):

```
$ git clone https://github.com/raphael/linux-samus
$ cd linux-samus/build/linux
$ make nconfig
$ make -j4
$ sudo make modules_install
$ sudo make install
```

By default, it will replace the symlinks in /boot/, so simply reboot and you should be all set.

For touchpad, brightness, and X acceleration, using the scripts provided by linux-samus should be sufficient. You will need to enable two-finger scrolling and/or tap-to-click through your desktop environment's settings. By default, natural scrolling and forward/back navigation should work with the touchscreen. Also interesting: the Pixel bar works natively (showing battery on tap, and playing a series of lights when entering the `Konami Code` on the keyboard).

Sound

The Pixel uses bcm-rt5677 for sound, which is not part of the mainline kernel. The custom linux-samus kernel patches this in and enables it for you, though some additional configuration is likely required. There is an all-in-one sound.sh script that should be run initially. Additionally, on every reboot, you will likely need to re-run the following to re-enable sound:

```
$ cd linux-samus/scripts/setup/sound
$ ALSA_CONFIG_UCM=ucm/ alsaucm -c bdw-rt5677 set _verb HiFi
```

You can add this to a startup script (~/.xinitrc for example) to have it automatically enabled in the future.

The handler for ACPI does not work out of the box under Slackware. If you change /etc/acpi/handler.sh to /etc/acpi/acpi_handler.sh in sound.sh, the proper rules will be added, but even so, some custom edits will need to be made. That said, you can run alsamixer or open your preferred volume control and simple change the source from “Speakers” to “Headphones” to switch between them.

Resolution

Natively, the Chromebook Pixel runs at a 2560×1700 resolution, which is picked up by Linux. However, other resolutions typically are not (all other resolutions available are 16:9, and will stretch or black bar your display). In order to use a lower resolution (e.g. 1920×1275 or 1600×1062), you'll need to add them to xrandr. First, get the mode line for your preferred resolution:

```
$ cvt 1920 1275
1920x1275 59.97 Hz (CVT) hsync: 79.28 kHz; pclk: 205.50 MHz
Modeline "1920x1275_60.00"  205.50  1920 2056 2256 2592  1275 1278 1288 1322
-hsync +vsync
```

Copy the last line, starting with the double quotes, and run the following:

```
$ xrandr --addmode eDP1 "1920x1275_60.00"  205.50  1920 2056 2256 2592  1275
1278 1288 1322 -hsync +vsync
```

From there, you can either alter the resolution via your desktop environment's display setting, or using xrandr:

```
$ xrandr --output eDP1 --mode 1920x1275
```

For other resolutions, simply repeat the process. It is recommended to add one (or more) resolution to a startup script to automatically add, or set, the native resolution on startup.

Keyboard

Natively, the search key will register as Super/Mod4 under Slackware. Likewise, the special keys along the top row will all register as F-keys. If you wish to use some of the special keys available on the Chromebook, it is recommended to use your desktop environment's keyboard shortcut settings, or an alternative program like xbindkeys.

Sources

- Slackware64 on the Chromebook Pixel 2015 (no longer maintained)
- Linux for the Chromebook Pixel 2015
- Developer Information for Chromebook Pixel

See Also

- Chromebook Pixel on ArchWiki

howtos, hardware, installation, chromebook, pixel