

---

# **Chromebooks Developer Cookbook Documentation**

*Release 1.0*

**Enric Balletbo i Serra**

**Feb 15, 2022**



---

## Contents:

---

|          |   |           |
|----------|---|-----------|
| <b>1</b> | <b>About the Chromebooks Developer Cookbook</b>   | <b>1</b>  |
| <b>2</b> | <b>Chromebook developer tool</b>                  | <b>3</b>  |
| 2.1      | Switch to developer mode . . . . .                | 3         |
| 2.2      | Enable booting from external storage . . . . .    | 3         |
| 2.3      | Create a USB or SD for dual booting . . . . .     | 4         |
| <b>3</b> | <b>Kernel Status Matrix</b>                       | <b>5</b>  |
| 3.1      | Acer Chromebook R13 . . . . .                     | 5         |
| 3.2      | Asus Chromebook Flip C100P . . . . .              | 6         |
| 3.3      | ASUS Chromebook Tablet CT100PA . . . . .          | 7         |
| 3.4      | Google Chromebook Pixel 2015 . . . . .            | 8         |
| 3.5      | Google Chromebook Pixel Book . . . . .            | 9         |
| 3.6      | Google Chromebook Pixel Slate . . . . .           | 10        |
| 3.7      | Samsung Chromebook Plus . . . . .                 | 11        |
| <b>4</b> | <b>GNOME3 Rootfs Demo</b>                         | <b>13</b> |
| 4.1      | The Debian way to build the demo rootfs . . . . . | 13        |
| 4.2      | Appendix . . . . .                                | 14        |
| <b>5</b> | <b>Weston Rockchip Demo</b>                       | <b>17</b> |
| 5.1      | Supported and tested hardware . . . . .           | 17        |
| 5.2      | The Debian way to build the demo rootfs . . . . . | 18        |
| 5.3      | Quick steps to create a SD-card . . . . .         | 19        |
| 5.4      | Connect the Wiimote . . . . .                     | 19        |
| 5.5      | Known issues . . . . .                            | 19        |
| 5.6      | Appendix . . . . .                                | 19        |
| <b>6</b> | <b>Indices and tables</b>                         | <b>21</b> |



---

## About the Chromebooks Developer Cookbook

---

This cookbook is a collection of useful scripts and information to create Debian-based images with the latest kernel that can be flashed on different Chromebook models.

The documentation, like the chromebook developer tool itself, is very much a work in progress. Please note that improvements to the documentation are welcome; create a github account and fork the project if you want to help out.

The latest version of the ‘Chromebooks Developer Cookbook’ can be found here:

- <https://chromebooks.readthedocs.io/en/latest/>



---

## Chromebook developer tool

---

These instructions will create a dual-booting environment where you can switch between booting Debian and the stock ChromeOS. No changes are made to the internal eMMC drive, and your new Debian install will run completely from external storage. This is the recommended setup for those that just want to take a test drive, or don't want to give up ChromeOS.

You must be running the latest ChromeOS prior to installation.

### 2.1 Switch to developer mode

1. Turn off the laptop.
2. To invoke Recovery mode, you hold down the ESC and Refresh keys and poke the Power button.
3. At the Recovery screen press Ctrl-D (there's no prompt - you have to know to do it).
4. Confirm switching to developer mode by pressing enter, and the laptop will reboot and reset the system. This takes about 10-15 minutes.

Note: After enabling developer mode, you will need to press Ctrl-D each time you boot, or wait 30 seconds to continue booting.

### 2.2 Enable booting from external storage

1. After booting into developer mode, hold Ctrl and Alt and poke the F2 key. This will open up the developer console.
2. Type root to the login screen.
3. Then type this to enable USB booting:

```
$ enable_dev_usb_boot
```

1. Reboot the system to allow the change to take effect.

## 2.3 Create a USB or SD for dual booting

```
$ ./chromebook-setup.sh help
```

For example, to create bootable SD card for the Samsung Chromebook Plus (arm64):

```
$ ./chromebook-setup.sh do_everything --architecture=arm64 --storage=/dev/sdX
```

The default username and password for the image is debian in lowercase.



---

## Kernel Status Matrix

---

### 3.1 Acer Chromebook R13

|            |                        |
|------------|------------------------|
| Board      | Oak Elm                |
| SoC        | Mediatek MT8173 2.1GHz |
| RAM        | 4GB                    |
| Firmware   | Coreboot               |
| Boot media | eMMC, SD or USB        |

### 3.1.1 Kernel Status Report

| Kernel version: 5.3 |                    |          |
|---------------------|--------------------|----------|
| Component           |                    | Status   |
| CPU                 | ARMv8 Cortex-A72   | FAIL     |
| GPU                 | PowerVR GX6250 GPU | Untested |
| Display             | 13.3" @ 1920x1080  | Untested |
| WiFi                |                    | Untested |
| Bluetooth           |                    | Untested |
| Touchpad            |                    | Untested |
| Touscreen           |                    | Untested |
| Camera              |                    | Untested |
| Type-C              | USB 3.0            | Untested |
|                     | USB 2.0            | Untested |
| BUTTONS             | POWER              | Untested |
|                     | VOLUME             | Untested |
| Embedded Controller | Google CrOS EC     | Untested |
| Audio               | Speaker            | Untested |
|                     | Headphone          | Untested |
|                     | Internal MIC       | Untested |
|                     | Headset MIC        | Untested |
|                     | Headset BTN_0      | Untested |
|                     | Headset VOLUME     | Untested |
| Suspend / Resume    |                    | Untested |

### 3.2 Asus Chromebook Flip C100P

|            |                   |
|------------|-------------------|
| Board      | Veyron-Minnie     |
| SoC        | Rockchip RK3288-C |
| RAM        | 4GB               |
| Firmware   | Coreboot          |
| Boot media | eMMC, SD or USB   |

### 3.2.1 Kernel Status Report

| Kernel version: 5.3 |                  |          |
|---------------------|------------------|----------|
| Component           | Model            | Status   |
| CPU                 | ARMv7 Cortex-A17 | Works    |
| GPU                 | Mali-T764        | Works    |
| USB 2.0             |                  | Works    |
| HDMI                |                  | Works    |
| Analog Audio        | Rockchip I2S     | Works    |
| Display             | 11.6" @ 1366x768 | Works    |
| WiFi                | Broadcom         | Works    |
| Bluetooth           |                  | Untested |
| Touchpad            | Elan I2C         | Works    |
| Touscreen           |                  | Works    |
| Camera              | HD UVC WebCam    | Works    |
| Embedded Controller | Google CrOS EC   | Works    |
| Audio               | Speaker          | Works    |
|                     | Headphone        | Works    |
|                     | Internal MIC     | Works    |
|                     | Headset MIC      | Works    |
|                     | Headset BTN_0    | Works    |
|                     | Headset VOLUME   | Works    |

### 3.3 ASUS Chromebook Tablet CT100PA

|            |            |
|------------|------------|
| Board      | Scarlet    |
| SoC        | rk3399     |
| RAM        | 4GB        |
| Firmware   | Coreboot   |
| Boot media | eMMC or SD |

### 3.3.1 Kernel Status Report

| Kernel version: 5.5 |                                   |          |
|---------------------|-----------------------------------|----------|
| Component           | Model                             | Status   |
| CPU                 | Armv8 Cortex-A72 Armv8 Cortex-A53 | Works    |
| GPU                 | Mali-T860MP4                      | Works    |
| Display             | 9.7" @ 1920x1080                  | Works    |
| WiFi                | Qualcomm Atheros                  | Works    |
| Bluetooth           |                                   | Fail     |
| Tousscreen          |                                   | Works    |
| Front Camera        |                                   | Untested |
| Back Camera         |                                   | Untested |
| Type-C              | USB 3.0                           | Works    |
|                     | USB 2.0                           | Works    |
| BUTTONS             | POWER                             | Works    |
|                     | VOLUME                            | Works    |
| Embedded Controller | Google CrOS EC                    | Works    |
| Audio               | Speaker                           | Works    |
|                     | Headphone                         | Fail     |
| Suspend / Resume    |                                   | Works    |

### 3.4 Google Chromebook Pixel 2015

|            |                                   |
|------------|-----------------------------------|
| Board      | Broadwell - Samus                 |
| SoC        | Intel Core-i5 (dual-core 2.2 GHz) |
| RAM        | 8GB                               |
| Firmware   | Coreboot                          |
| Boot media | eMMC, SD or USB                   |

### 3.4.1 Kernel Status Report

| Kernel version: 5.3 |                    |        |
|---------------------|--------------------|--------|
| Component           |                    | Status |
| CPU                 | Intel Core-i5      | Works  |
| GPU                 | Intel HD 5500      | Works  |
| Display             | 12.85" @ 2560x1700 | Works  |
| WiFi                | Atheros 802.11n    |        |
| Bluetooth           |                    |        |
| Touchpad            | Atmel mXT I2C      | Works  |
| Tousscreen          | Atmel mXT I2C      | Works  |
| Camera              | HD UVC WebCam      | Works  |
| Type-C              | USB 3.0            | Works  |
|                     | USB 2.0            | Works  |
|                     | DP                 |        |
|                     | HDMI               |        |
| USB-A               | USB 3.0            | Works  |
|                     | USB 2.0            | Works  |
| BUTTONS             | POWER              | Works  |
| Embedded Controller | Google CrOS EC     |        |
| Audio               | Speaker            | Fails  |
|                     | Headphone          | Fails  |
|                     | Internal MIC       | Fails  |
|                     | Headset MIC        | Fails  |
|                     | Headset BTN_0      | Fails  |
|                     | Headset VOLUME     | Fails  |
| Suspend / Resume    |                    | Fails  |

### 3.5 Google Chromebook Pixel Book

|            |               |
|------------|---------------|
| Board      | Eve           |
| SoC        | Intel Core i7 |
| RAM        | 16GB          |
| Firmware   | Coreboot      |
| Boot media | eMMC or USB   |

### 3.5.1 Kernel Status Report

| Kernel version: 5.4.8 |                  |        |
|-----------------------|------------------|--------|
| Component             | Model            | Status |
| CPU                   | Intel core i7    | Works  |
| GPU                   | i915             | Works  |
| Display               | 12.3" @2400x1600 | Works  |
| WiFi                  |                  | Works  |
| Bluetooth             |                  | Works  |
| Touchscreen           |                  | Works  |
| Touchpad              |                  | Fail   |
| Camera                |                  | Works  |
| Type-C                | USB 3.0          | Works  |
|                       | USB 2.0          | Works  |
| BUTTONS               | POWER            | Works  |
|                       | VOLUME           | Works  |
| Embedded Controller   | Google CrOS EC   | Works  |
| Audio                 | Speaker          | Fail   |
|                       | Headphone        | Fail   |
| Suspend / Resume      |                  | Fail   |

## 3.6 Google Chromebook Pixel Slate

|            |             |
|------------|-------------|
| Board      | Nocturne    |
| SoC        | Intel i5    |
| RAM        | 8GB         |
| Firmware   | Coreboot    |
| Boot media | eMMC or USB |

### 3.6.1 Kernel Status Report

| Kernel version: 5.4.8 |                        |          |
|-----------------------|------------------------|----------|
| Component             | Model                  | Status   |
| CPU                   | Intel i5               | Works    |
| GPU                   | Intel UHD Graphics 615 | Works    |
| USB type C            |                        | Works    |
| Display               | 12.3" @ 3000x2000      | Works    |
| WiFi                  |                        | Works    |
| Bluetooth             |                        | Fail     |
| Touchpad              |                        | Fail     |
| Touscreen             |                        | Works    |
| Front Camera          |                        | Fail     |
| Back Camera           |                        | Fail     |
| Embedded Controller   | Google CrOS EC         | Works    |
| Audio                 | Speaker                | Fail     |
|                       | Headphone              | Untested |

### 3.6.2 Notes:

#### camera:

The camera seems to work but are not fully tested. Nocturne uses INTEL IPU3 to get data form camera sensors. In order to test it, we can use libcamera.

Download git camera tool:

```
` git clone git://linuxtv.org/libcamera.git cd libcamera meson build ninja -C build install `
```

You should see 2 devices, imx355 and imx319 with 'cam -l'

You can get data from sensors:

```
` cam -c "imx319 9-0010 0" -C --file="/tmp/libcamframe#.data" -s width=1280, height=720 or cam -c "imx355 10-001a 1" -C --file="/tmp/libcamframe#.data" -s width=1280,height=720 `
```

Next step is to use IMGU device to transform raw data to viewable pictures. This can be done using v4l2n tool available here: <https://github.com/intel/nvt>

## 3.7 Samsung Chromebook Plus

|            |                        |
|------------|------------------------|
| Board      | Gru Kevin              |
| SoC        | Rockchip RK3399 2.0GHz |
| RAM        | 4GB                    |
| Firmware   | Coreboot               |
| Boot media | eMMC, SD or USB        |

### 3.7.1 Kernel Status Report

| Kernel version: 5.5 |                  |        |
|---------------------|------------------|--------|
| Component           |                  | Status |
| CPU                 | ARMv8 Cortex-A72 | Works  |
| GPU                 | Mali-T860MP4     | Works  |
| Display             | 11.6" @ 1366x768 | Works  |
| WiFi                | Broadcom         | Works  |
| Bluetooth           |                  | Works  |
| Touchpad            | Elan I2C         | Works  |
| Touscreen           |                  | Works  |
| Camera              | HD UVC WebCam    | Works  |
| Type-C              | USB 3.0          | Works  |
|                     | USB 2.0          | Works  |
| BUTTONS             | POWER            | Works  |
|                     | VOLUME           | Works  |
| Embedded Controller | Google CrOS EC   | Works  |
| Audio               | Speaker          | Works  |
|                     | Headphone        | FAIL   |
|                     | Internal MIC     | FAIL   |
|                     | Headset MIC      | FAIL   |
|                     | Headset BTN_0    | Works  |
|                     | Headset VOLUME   | Works  |
| Suspend / Resume    |                  | Works  |



---

## GNOME3 Rootfs Demo

---

This is the rootfs used by default to build the images for the Chromebooks. The main purpose of this rootfs is show current support in mainline for different boards. The demo runs a *Debian* based image with GNOME3 and accelerated graphics when possible.

The *Debian* images are assembled using the `debos` utility, which uses the *Debian* package feed beneath. Stuff not available in official *Debian* packages will be built from sources or downloaded into the final image.

### 4.1 The Debian way to build the demo rootfs

To install `debos` you can do the following steps:

```
$ sudo apt install golang git libglib2.0-dev libostree-dev qemu-system-x86 qemu-user-  
→static debootstrap systemd-container xz-utils bmap-tools  
$ export GOPATH=`pwd`/gocode  
$ go get -u github.com/go-debos/debos/cmd/debos
```

First, make sure you have KVM installed:

```
$ sudo apt install qemu-kvm ovmf  
  
And then run:  
  
$ $GOPATH/bin/debos -m 4G <debos-image.yaml>
```

Now that `debos` is installed, let's create the demos images, run:

```
Export the architecture of your device (arm or arm64 or amd64):  
  
$ export architecture=arm64  
  
And then run:
```

(continues on next page)

(continued from previous page)

```
$ $GOPATH/bin/debos -m 4G -t architecture:$architecture tools/debos/images/gnome-  
↳desktop/gnome-desktop.yaml
```

Will create the following output:

- debian-gnome-desktop-sid-\$architecture.tar.gz, a tarball with the Debian GNOME3 based filesystem.

After that, to create an image run (**note** this is only supported for arm and arm64 architectures):

```
$ $GOPATH/bin/debos -m 4G -t architecture:$architecture tools/debos/images/gnome-  
↳desktop/chromebook-image.yaml
```

Will create the following output:

- debian-gnome-desktop-sid-\$architecture.img.gz, a gz-compressed image file for a Chromebook.
- debian-gnome-desktop-sid-\$architecture.img.gz.md5, the image checksum.
- debian-gnome-desktop-sid-\$architecture.img.bmap, a bitmap summary for faster flashing via bmaptools.

To flash it, assuming your SD card is /dev/mmcblk0, use:

```
$ bmaptool copy debian-gnome-desktop-sid-$architecture.img.gz /dev/mmcblk0
```

The bmap file is automatically looked for in the current directory.

Note that the credentials to login are debian:debian.

## 4.2 Appendix

### 4.2.1 The Docker way to build the demo rootfs

**NOT TESTED YET**

This is really simple as an official container is provided for it:

```
$ docker pull godebos/debos
```

To build the image run:

```
$ docker run --rm --interactive --tty --device /dev/kvm --user $(id -u) --workdir /  
↳recipes --mount "type=bind,source=$(pwd),destination=/recipes" --security-opt_  
↳label=disable godebos/debos <debos-image.yaml>
```

### 4.2.2 Extend the rootfs partition to fill available space

What you need to do is to run growpart against you physical disk and partition.

For example your disk is /dev/mmcblk0 and your partition is 2. What growpart does is actually extending the size of your partition to the maximum allowed physical disk size.

You can run below command:

```
$ growpart /dev/mmcblk0 2
```

Once it finish you will see growpart has extend your partition table to the maximum available disk size.

Now, you need to reboot your machine. When the machine comes live again, you can run `resize2fs` command to extend the filesystem. Below is the sample command.

```
$ resize2fs /dev/mmcblk0p2
```



---

## Weston Rockchip Demo

---

The purpose of this demo is show current support in mainline for different Rockchip boards. The demo runs a *Debian* based image with Weston and accelerated graphics using Panfrost.

The *Debian* images are assembled using the `debos` utility, which uses the *Debian* package feed beneath. Stuff not available in official *Debian* packages will be built from sources or downloaded into the final image.

### 5.1 Supported and tested hardware

#### 5.1.1 Samsung Chromebook Plus (kevin)

#### 5.1.2 Samsung Chromebook Plus (kevin)

The Samsung Chromebook Plus is a convertible touchscreen laptop powered by an ARMv8 Rockchip RK3399 hexa-core processor and 4GB RAM, measuring 11.04" x 8.72" x 0.55" and weighing 2.38 lbs.

Features:

- Rockchip RK3399 (OP1) dual-core 2.0GHz Cortex-A72 and quad-core 1.4GHz Cortex-A53 processor
- 4GB LPDDR3 RAM
- 12.3" 2400x1600 LED display
- Mali T860MP4 GPU
- 32GB eMMC
- 5140 mAh battery
- 2x USB 3.0 Type-C ports
- Built-in stylus

### 5.1.3 ASUS Chromebook Flip C100P (veyron\_minnie)

### 5.1.4 ASUS Chromebook Flip C100P (veyron\_minnie)

The ASUS Chromebook Flip C100P is a convertible touchscreen laptop powered by an ARMv7 Rockchip RK3288 processor and 4GB RAM, measuring 262.8 x 182.4 x 15.6 mm (WxDxH) and weighing 0.89 kg.

Features:

- Rockchip RK3288 1.8GHz
- 2GB/4GB LPDDR3 RAM
- 10,1" - 25,65 cm 16:10 WXGA (1280x800) LED display
- Mali T764 GPU
- 16GB/32GB eMMC
- 2Cells 31 Whrs nattery
- 2x USB 2.0 ports

## 5.2 The Debian way to build the demo rootfs

To install `debos` you can do the following steps:

```
$ sudo apt install golang git libglib2.0-dev libostree-dev qemu-system-x86 qemu-user-  
↳static debootstrap systemd-container xz-utils bmap-tools  
$ export GOPATH=`pwd`/gocode  
$ go get -u github.com/go-debos/debos/cmd/debos
```

First, make sure you have KVM installed:

```
$ sudo apt install qemu-kvm ovmf  
  
And then run:  
  
$ $GOPATH/bin/debos -m 4G <debos-image.yaml>
```

Now that `debos` is installed, let's create the demos images, run:

```
Export the architecture of your device:  
  
$ export architecture=arm64  
or  
$ export architecture=armhf  
  
And then run:  
$ $GOPATH/bin/debos -m 4G -t architecture:$architecture tools/debos/images/weston-  
↳desktop/weston-desktop.yaml
```

Will create the following output:

- `debian-weston-desktop-sid-$architecture.tar.gz`, a tarball with the debian weston based filesystem.

## 5.3 Quick steps to create a SD-card

```
$ ./chromebook-setup.sh do_everything --architecture=<arm64|arm> --storage=/dev/
↳mmcblkX
```

## 5.4 Connect the Wiimote

First you need to make sure to load the uinput module:

```
$ modprobe uinput
```

Thanks to cwiid you can scan for your Wiimote now:

```
(press the 1 and 2 buttons on your Wiimote)
$ bluetoothctl scan on
Scanning ...
    <MAC address>          Nintendo RVL-CNT-01
```

The Wiimote can act as a regular input device like a mouse using wminput, simply run:

```
$ wminput -w
```

## 5.5 Known issues

### 5.5.1 1. Ethernet is down by default

For some reason the ethernet interface (through a docking USB-C) is down, to enable it run:

```
$ ip link set enx0050b6213e94 up
```

## 5.6 Appendix

### 5.6.1 The Docker way to build the demo rootfs

#### NOT TESTED YET

This is really simple as an official container is provided for it:

```
$ docker pull godebos/debos
```

To build the image run:

```
$ docker run --rm --interactive --tty --device /dev/kvm --user $(id -u) --workdir /
↳recipes --mount "type=bind,source=$(pwd),destination=/recipes" --security-opt_
↳label=disable godebos/debos <debos-image.yaml>
```





## CHAPTER 6

---

### Indices and tables

---

- `genindex`
- `modindex`
- `search`