ChromeOS PowerControl

Hardware Control Suite for ChromeOS

Visit ChromeGeek.com

Overview

ChromeOS_PowerControl is a suite of lightweight shell scripts providing comprehensive hardware control in ChromeOS. This powerful toolkit enables users to manage CPU performance, battery charging, fan curves, GPU frequencies, and sleep behavior to optimize their ChromeOS experience.

PowerControl

Control CPU clockspeed in relation to temperature, enabling lower temperatures and longer battery life under load.

BatteryControl

Control battery charging limit instead of relying on Adaptive Charging to maximize battery longevity.

FanControl

Control fan curve in relation to temperature with built-in hysteresis and 0% RPM mode.

GPUControl

Control GPU clockspeed below its default maximum, enabling longer battery life under load.

SleepControl

Control how long ChromeOS can remain idle before sleep, with display dimming support.

Requirements

Requires Developer Mode - Supports AMD, ARM, and Intel architectures.

Features global commands for ease of use, a unified config file, and the ability to change settings in real-time. Includes a feature-rich installer, an uninstaller that cleans up after itself, and logs stored in /var/log/ for statistics.

Optionally have BatteryControl, PowerControl, FanControl, GPUControl, and SleepControl start on boot if user has rootfs verification disabled.

Installation

Step 1: Open crosh shell and run:

```
bash <(curl -s
"https://raw.githubusercontent.com/shadowed1/ChromeOS_PowerControl/main/ChromeOS_Power
Control_Downloader.sh?$(date +%s)")
```

The installer will be placed at:

```
/home/chronos/ChromeOS_PowerControl/ChromeOS_PowerControl_Installer.sh
```

Step 2: In VT-2 or crosh shell with sudo enabled, run:

```
sudo mkdir -p /usr/local/bin
sudo mv /home/chronos/ChromeOS_PowerControl_Installer.sh /usr/local/bin
sudo bash /usr/local/bin/ChromeOS_PowerControl_Installer.sh
```

The installer has prompts to customize installation. PowerControl, BatteryControl, FanControl, and SleepControl can run in the background and can be adjusted in real-time.

Commands Reference

PowerControl

```
sudo powercontrol  # Show status
sudo powercontrol start  # Throttle CPU based on temperature curve
sudo powercontrol stop  # Restore default CPU settings
sudo powercontrol no_turbo 1  # 0 = Enable, 1 = Disable Turbo Boost
sudo powercontrol max_perf_pct 75  # Set max performance percentage
sudo powercontrol min_perf_pct 50  # Set minimum performance at max temp
sudo powercontrol max_temp 86  # Max temperature threshold - Limit is 90°C
sudo powercontrol min_temp 60  # Min temperature threshold
sudo powercontrol hotzone 78  # Temperature threshold for aggressive thermal management
sudo powercontrol ramp_up 15  # % in steps CPU will increase in clockspeed per second
sudo powercontrol ramp_down 20  # % in steps CPU will decrease in clockspeed per second
sudo powercontrol monitor  # Toggle live temperature monitoring
sudo powercontrol startup  # Copy/Remove no_turbo.conf & powercontrol.conf at: /etc/init/
sudo powercontrol version  # Check PowerControl version
sudo powercontrol help  # Help menu
```

BatteryControl

```
sudo batterycontrol  # Check BatteryControl status
sudo batterycontrol start  # Start BatteryControl
sudo batterycontrol stop  # Stop BatteryControl
sudo batterycontrol 77  # Charge limit set to 77% - minimum of 14% allowed
sudo batterycontrol startup  # Copy/Remove batterycontrol.conf at: /etc/init/
sudo batterycontrol help  # Help menu
```

FanControl

```
sudo fancontrol  # Show FanControl status
sudo fancontrol start  # Start FanControl
sudo fancontrol stop  # Stop FanControl
sudo fancontrol fan_min_temp 48  # Min temp threshold
sudo fancontrol fan_max_temp 81  # Max temp threshold - Limit is 90°C
sudo fancontrol min_fan 0  # Min fan speed %
sudo fancontrol max_fan 100  # Max fan speed %
sudo fancontrol step_up 20  # Fan step-up %
sudo fancontrol step_down 1  # Fan step-down %
sudo fancontrol monitor  # Toggle on/off live monitoring in terminal
sudo fancontrol startup  # Copy/Remove fancontrol.conf at: /etc/init/
sudo fancontrol help  # Help menu
```

GPUControl

```
sudo gpucontrol  # Show current GPU info and frequency
sudo gpucontrol restore  # Restore GPU max frequency to original value
sudo gpucontrol intel 700  # Clamp Intel GPU max frequency to 700 MHz
sudo gpucontrol amd 800  # Clamp AMD GPU max frequency to 800 MHz (rounds down)
sudo gpucontrol adreno 500000  # Clamp Adreno GPU max frequency to 500000 kHz (or 500 MHz)
sudo gpucontrol mali 600000  # Clamp Mali GPU max frequency to 600000 kHz (or 600 MHz)
sudo gpucontrol startup  # Copy/Remove gpucontrol.conf at: /etc/init/
sudo gpucontrol help  # Help menu
```

SleepControl

```
sudo sleepcontrol  # Show SleepControl status
```

```
sudo sleepcontrol start  # Start SleepControl
```

```
sudo sleepcontrol stop  # Stop SleepControl
```

```
sudo sleepcontrol battery 3 7 12  # Dims in 3m, timeout in 7m, and sleeps in 12m on battery
```

```
sudo sleepcontrol power 5 15 30  # Dims in 5m, timeout in 15m and sleeps in 30m when plugged-in
```

```
sudo sleepcontrol battery audio 0  # Disable audio detection on battery; sleep can occur during media playback
```

```
sudo sleepcontrol power audio 1  # Enable audio detection on power; delaying sleep until audio is stopped
```

```
sudo sleepcontrol startup  # Copy or Remove sleepcontrol.conf at: /etc/init/
```

```
sudo sleepcontrol help  # Help menu
```

Maintenance Commands

```
sudo powercontrol reinstall  # Download and reinstall ChromeOS_PowerControl from main branch on Github
```

```
sudo powercontrol uninstall  # Global uninstaller that will clean up after itself
```

```
sudo /usr/local/bin/ChromeOS_PowerControl/Uninstall_ChromeOS_PowerControl.sh  # Alternative uninstall method
```

How It Works

PowerControl

Uses ARM, AMD, and Intel's max_perf_pct for easy user control. Pairs user adjustable max_perf_pct and thermal0 temp sensor to create a user adjustable clockspeed-temperature curve. If $min_temp threshold is below a certain point, the CPU will be able to reach max_perf_pct of its speed. The closer the CPU approaches $max_temp, the closer it is to min_perf_pct. PowerControl will always be stringent regarding thermals and performance versus native behavior.

BatteryControl

Uses ectool's chargecontrol to toggle between normal or idle. Checks ectool usbpdpower to identify which charge port is being used. Recommend turning off adaptive charging in ChromeOS to avoid notification spam. Checks BAT0/capacity to measure when to toggle ectool's chargecontrol. ChromeOS reports slightly higher values than what BatteryControl sets the charge limit to. Charge limit is preserved during sleep unless in deep sleep before reaching limit.

FanControl

Uses ectool's fanduty control and autofanctrl for manual and automatic control. Pairs fanduty with thermal0 temperature sensor for a user adjustable fan-temperature curve. Uses hysteresis formula to attempt a better sounding and performing fan curve than the OEM provides. Uses a kickstart mechanism when fan leaves 0% to enable zero RPM mode for any fan type. Default FanControl behavior has aggressive fan ramp-up behavior with a graceful decrease.

GPUControl

Identifies the GPU (AMD, Adreno, Mali, and Intel) based on the name of the device's path in /sys/class/. Limits control to only below the maximum clock speed for safety and with Chromebooks in mind. Applies a 120s delay on boot if the user is applying a custom clock speed as a precaution.

- Intel GPU's maximum clock speed changed from: /sys/class/drm/card0/gt_max_freq_mhz
- AMD GPU's maximum clockspeed changed from: /sys/class/drm/card0/pp_od_clk_voltage
- Adreno GPU's maximum clockspeed changed from /sys/class/kgsl/kgsl-3d0/max_gpuclk
- Mali GPU's maximum clockspeed changed from: /sys/class/devfreq/mali0/max_freq

SleepControl

By reading powerd.LATEST log, SleepControl monitors when the powerd daemon reports 'User activity stopped'. Parsing strings like 'User activity started' or 'Audio activity started' tells SleepControl the user is active to pause until is reported. If 'User activity stopped' and 'Audio activity stopped' is parsed, SleepControl assumes the user is away and custom sleep timers can begin.

Can turn on or off audio detection to customize sleep during multimedia playback. ChromeOS will report 'User activity stopped' after around 20 seconds of inactivity, so the timers won't be exact. When idle, SleepControl uses dbus to send an empty input every 120s until interrupted/sleeping with the configurable timer. By using epoch timestamps, SleepControl is able to check when its simulated inputs are to be ignored.

ChromeOS PowerControl Documentation

For more ChromeOS tips, tricks, and tools visit:

ChromeGeek.com