

Only the original [README](#) is guaranteed to be up-to-date.

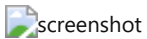
只有原版的 [README](#) 是保證最新的。

本文件翻譯時點: [521f2fe](#)

scrcpy (v1.15)

Scrcpy 可以透過 USB、或是 [TCP/IP](#) 來顯示或控制 Android 裝置。且 scrcpy 不需要 *root* 權限。

Scrcpy 目前支援 *GNU/Linux*、*Windows* 和 *macOS*。



特色:

- **輕量** (只顯示裝置螢幕)
- **效能** (30~60fps)
- **品質** (1920×1080 或更高)
- **低延遲** ([35~70ms](#))
- **快速啟動** (~1 秒就可以顯示第一個畫面)
- **非侵入性** (不安裝、留下任何東西在裝置上)

需求

Android 裝置必須是 API 21+ (Android 5.0+)。

請確認裝置上的 [adb 偵錯](#) ([通常位於開發者模式內](#)) 已啟用。

在部分的裝置上，你也必須啟用[特定的額外選項](#)才能使用鍵盤和滑鼠控制。

下載/獲取軟體

Linux

Debian (目前支援 *testing* 和 *sid*) 和 Ubuntu (20.04):

```
apt install scrcpy
```

[Snap](#) 上也可以下載: [scrcpy](#) .

在 Fedora 上也可以使用 [COPR](#) 下載: [scrcpy](#) .

在 Arch Linux 上也可以使用 [AUR](#) 下載: [scrcpy](#) .

在 Gentoo 上也可以使用 [Ebuild](#) 下載: [scrcpy/](#) .

你也可以自己[編譯 Scrcpy](#)。別擔心，並沒有想像中的難。

Windows

為了保持簡單，Windows 用戶可以下載一個包含所有必需軟體 (包含 `adb`) 的壓縮包:

- [README](#)

[Chocolatey](#) 上也可以下載:

```
choco install scrcpy
choco install adb # 如果你還沒有安裝的話
```

[Scoop](#) 上也可以下載:

```
scoop install scrcpy
scoop install adb # 如果你還沒有安裝的話
```

你也可以自己[編譯 Scrcpy](#)。

macOS

Scrcpy 可以在 [Homebrew](#) 上直接安裝:

```
brew install scrcpy
```

由於執行期間需要可以藉由 `PATH` 存取 `adb` 。如果還沒有安裝 `adb` 可以使用下列方式安裝:

```
brew cask install android-platform-tools
```

你也可以自己[編譯 Scrcpy](#)。

執行

將電腦和你的 Android 裝置連線，然後執行:

```
scrcpy
```

Scrcpy 可以接受命令列參數。輸入下列指令就可以瀏覽可以使用的命令列參數:

```
scrcpy --help
```

功能

以下說明中，有關快捷鍵的說明可能會出現 `MOD` 按鈕。相關說明請參見[快捷鍵](#)內的說明。

畫面擷取

縮小尺寸

使用比較低的解析度來投放 Android 裝置在某些情況可以提升效能。

限制寬和高的最大值(例如: 1024):

```
scrcpy --max-size 1024
scrcpy -m 1024 # 縮短版本
```

比較小的參數會根據螢幕比例重新計算。根據上面的範例，1920x1080 會被縮小成 1024x576。

更改 bit-rate

預設的 bit-rate 是 8 Mbps。如果要更改 bit-rate (例如: 2 Mbps):

```
scrcpy --bit-rate 2M
scrcpy -b 2M # 縮短版本
```

限制 FPS

限制畫面最高的 FPS:

```
scrcpy --max-fps 15
```

僅在 Android 10 後正式支援，不過也有可能可以在 Android 10 以前的版本使用。

裁切

裝置的螢幕可以裁切。如此一來，鏡像出來的螢幕就只會是原本的一部份。

假如只要鏡像 Oculus Go 的其中一隻眼睛:

```
scrcpy --crop 1224:1440:0:0 # 位於 (0,0), 大小1224x1440
```

如果 `--max-size` 也有指定的話，裁切後才會縮放。

鎖定影像方向

如果要鎖定鏡像影像方向:

```
scrcpy --lock-video-orientation 0 # 原本的方向
scrcpy --lock-video-orientation 1 # 逆轉 90°
scrcpy --lock-video-orientation 2 # 180°
scrcpy --lock-video-orientation 3 # 順轉 90°
```

這會影響錄影結果的影像方向。

錄影

鏡像投放螢幕的同時也可以錄影:

```
scrcpy --record file.mp4
scrcpy -r file.mkv
```

如果只要錄影，不要投放螢幕鏡像的話:

```
scrcpy --no-display --record file.mp4
scrcpy -Nr file.mkv
# 用 Ctrl+C 停止錄影
```

就算有些幀為了效能而被跳過，它們還是一樣會被錄製。

裝置上的每一幀都有時間戳記，所以 [封包延遲 \(Packet Delay Variation, PDV\)](#) 並不會影響錄影的檔案。

連線

無線

`Scrcpy` 利用 `adb` 和裝置通訊，而 `adb` 可以[透過 TCP/IP 連結](#)：

1. 讓電腦和裝置連到同一個 Wi-Fi。
2. 獲取手機的 IP 位址(設定 → 關於手機 → 狀態)。
3. 啟用裝置上的 `adb over TCP/IP`：`adb tcpip 5555`。
4. 拔掉裝置上的線。
5. 透過 TCP/IP 連接裝置：`adb connect DEVICE_IP:5555` (把 `DEVICE_IP` 換成裝置的 IP 位址)。
6. 和平常一樣執行 `scrcpy`。

如果效能太差，可以降低 bit-rate 和解析度：

```
scrcpy --bit-rate 2M --max-size 800
scrcpy -b2M -m800 # 縮短版本
```

多裝置

如果 `adb devices` 內有多個裝置，則必須附上 `serial`：

```
scrcpy --serial 0123456789abcdef
scrcpy -s 0123456789abcdef # 縮短版本
```

如果裝置是透過 TCP/IP 連線：

```
scrcpy --serial 192.168.0.1:5555
scrcpy -s 192.168.0.1:5555 # 縮短版本
```

你可以啟用復數個對應不同裝置的 `_scrcpy_`。

裝置連結後自動啟動

你可以使用 [AutoAdb](#)：

```
autoadb scrcpy -s '{}'
```

SSH tunnel

本地的 `adb` 可以連接到遠端的 `adb` 伺服器(假設兩者使用相同版本的 `adb` 通訊協定)，以連接到遠端裝置：

```
adb kill-server # 停止在 Port 5037 的本地 adb 伺服
ssh -CN -L5037:localhost:5037 -R27183:localhost:27183 your_remote_computer
# 保持開啟
```

從另外一個終端機：

```
scrcpy
```

如果要避免啟用 remote port forwarding，你可以強制它使用 forward connection (注意 `-L` 和 `-R` 的差別):

```
adb kill-server      # 停止在 Port 5037 的本地 adb 伺服
ssh -CN -L5037:localhost:5037 -L27183:localhost:27183 your_remote_computer
# 保持開啟
```

從另外一個終端機:

```
scrcpy --force-adb-forward
```

和無線連接一樣，有時候降低品質會比較好:

```
scrcpy -b2M -m800 --max-fps 15
```

視窗調整

標題

預設標題是裝置的型號，不過可以透過以下方式修改:

```
scrcpy --window-title 'My device'
```

位置 & 大小

初始的視窗位置和大小也可以指定:

```
scrcpy --window-x 100 --window-y 100 --window-width 800 --window-height 600
```

無邊框

如果要停用視窗裝飾:

```
scrcpy --window-borderless
```

保持最上層

如果要保持 `scrcpy` 的視窗在最上層:

```
scrcpy --always-on-top
```

全螢幕

這個軟體可以直接在全螢幕模式下起動:

```
scrcpy --fullscreen
scrcpy -f # 縮短版本
```

全螢幕可以使用 `MOD+f` 開關。

旋轉

視窗可以旋轉:

```
scrcpy --rotation 1
```

可用的數值:

- `0` : 不旋轉
- `1` : 90 度逆轉
- `2` : 180 度
- `3` : 90 度順轉

旋轉方向也可以使用 `MOD+←` (左方向鍵) 和 `MOD+→` (右方向鍵) 調整。

`scrcpy` 有 3 種不同的旋轉:

- `MOD+r` 要求裝置在垂直、水平之間旋轉 (目前運行中的 App 有可能會因為不支援而拒絕)。
- `--lock-video-orientation` 修改鏡像的方向 (裝置傳給電腦的影像)。這會影響錄影結果的影像方向。
- `--rotation` (或是 `MOD+←` / `MOD+→`) 只旋轉視窗的內容。這只會影響鏡像結果，不會影響錄影結果。

其他鏡像選項

唯讀

停用所有控制，包含鍵盤輸入、滑鼠事件、拖放檔案:

```
scrcpy --no-control  
scrcpy -n
```

顯示螢幕

如果裝置有複數個螢幕，可以指定要鏡像哪個螢幕:

```
scrcpy --display 1
```

可以透過下列指令獲取螢幕 ID:

```
adb shell dumpsys display # 找輸出結果中的 "mDisplayId="
```

第二螢幕只有在 Android 10+ 時可以控制。如果不是 Android 10+，螢幕就會在唯讀狀態下投放。

保持清醒

如果要避免裝置在連接狀態下進入睡眠:

```
scrcpy --stay-awake  
scrcpy -w
```

`scrcpy` 關閉後就會回復成原本的設定。

關閉螢幕

鏡像開始時，可以要求裝置關閉螢幕：

```
scrcpy --turn-screen-off  
scrcpy -S
```

或是在任何時候輸入 `MOD+o`。

如果要開啟螢幕，輸入 `MOD+Shift+o`。

在 Android 上，`POWER` 按鈕總是開啟螢幕。

為了方便，如果 `POWER` 是透過 `scrcpy` 轉送 (右鍵 或 `MOD+p`) 的話，螢幕將會在短暫的延遲後關閉。

實際在手機上的 `POWER` 還是會開啟螢幕。

防止裝置進入睡眠狀態：

```
scrcpy --turn-screen-off --stay-awake  
scrcpy -Sw
```

顯示過期的幀

為了降低延遲，`scrcpy` 預設只會顯示最後解碼的幀，並且拋棄所有在這之前的幀。

如果要強制顯示所有的幀 (有可能會拉高延遲)，輸入：

```
scrcpy --render-expired-frames
```

顯示觸控點

對於要報告的人來說，顯示裝置上的實際觸控點有時候是有幫助的。

Android 在「開發者選項」中有提供這個功能。

`Scrcpy` 可以在啟動時啟用這個功能，並且在停止後恢復成原本的設定：

```
scrcpy --show-touches  
scrcpy -t
```

這個選項只會顯示**實際觸碰在裝置上的觸碰點**。

輸入控制

旋轉裝置螢幕

輸入 `MOD+r` 以在垂直、水平之間切換。

如果使用中的程式不支援，則不會切換。

複製/貼上

如果 Android 剪貼簿上的內容有任何更動，電腦的剪貼簿也會一起更動。

任何與 `Ctrl` 相關的快捷鍵事件都會轉送到裝置上。特別來說：

- `Ctrl+c` 通常是複製
- `Ctrl+x` 通常是剪下
- `Ctrl+v` 通常是貼上 (在電腦的剪貼簿與裝置上的剪貼簿同步之後)

這些跟你通常預期的行為一樣。

但是，實際上的行為是根據目前運行中的應用程式而定。

舉例來說，*Termux* 在收到 `Ctrl+c` 後，會傳送 `SIGINT`；而 *K-9 Mail* 則是建立新訊息。

如果在這情況下，要剪下、複製或貼上 (只有在Android 7+時才支援)：

- `MOD+c` 注入 複製
- `MOD+x` 注入 剪下
- `MOD+v` 注入 貼上 (在電腦的剪貼簿與裝置上的剪貼簿同步之後)

另外，`MOD+Shift+v` 則是以一連串的按鍵事件貼上電腦剪貼簿中的內容。當元件不允許文字貼上 (例如 *Termux*) 時，這就很有用。不過，這在非 ASCII 內容上就無法使用。

警告：貼上電腦的剪貼簿內容 (無論是從 `Ctrl+v` 或 `MOD+v`) 時，會複製剪貼簿中的內容至裝置的剪貼簿上。這會讓所有 Android 程式讀取剪貼簿的內容。請避免貼上任何敏感內容 (像是密碼)。

文字輸入偏好

輸入文字時，有兩種[事件](#)會被觸發：

- `_` 鍵盤事件 (key events)_，代表有一個按鍵被按下或放開
- `_` 文字事件 (text events)_，代表有一個文字被輸入

預設上，文字是被以鍵盤事件 (key events) 輸入的，所以鍵盤和遊戲內所預期的一樣 (通常是指 WASD)。

但是這可能造成[一些問題](#)。如果在這輸入這方面遇到了問題，你可以試試：

```
scrcpy --prefer-text
```

(不過遊戲內鍵盤就會不可用)

重複輸入

通常來說，長時間按住一個按鍵會重複觸發按鍵事件。這會在一些遊戲中造成效能問題，而且這個重複的按鍵事件是沒有意義的。

如果不要轉送這些重複的按鍵事件：

```
scrcpy --no-key-repeat
```

檔案

安裝 APK

如果要安裝 APK，拖放一個 APK 檔案 (以 `.apk` 為副檔名) 到 *scrcpy* 的視窗上。

視窗上不會有任何反饋；結果會顯示在命令列中。

推送檔案至裝置

如果要推送檔案到裝置上的 `/sdcard/`，拖放一個非 APK 檔案 (不以 `.apk` 為副檔名) 到 `scrcpy` 的視窗上。

視窗上不會有任何反饋；結果會顯示在命令列中。

推送檔案的目標路徑可以在啟動時指定：

```
scrcpy --push-target /sdcard/foo/bar/
```

音訊轉送

`scrcpy` 不轉送音訊。請使用 [sndcpy](#)。另外，參見 [issue #14](#)。

快捷鍵

在以下的清單中，`MOD` 是快捷鍵的特殊按鍵。通常來說，這個按鍵是 (左) `Alt` 或是 (左) `Super`。

這個是可以使用 `--shortcut-mod` 更改的。可以用的選項有：

- `lctrl`：左邊的 `Ctrl`
- `rctrl`：右邊的 `Ctrl`
- `lalt`：左邊的 `Alt`
- `ralt`：右邊的 `Alt`
- `lsuper`：左邊的 `Super`
- `rsuper`：右邊的 `Super`

```
# 以 右邊的 Ctrl 為快捷鍵特殊按鍵
scrcpy --shortcut-mod=rctrl

# 以 左邊的 Ctrl 和左邊的 Alt 或是 左邊的 Super 為快捷鍵特殊按鍵
scrcpy --shortcut-mod=lctrl+lalt,lsuper
```

[Super](#) 通常是 *Windows* 或 *Cmd* 鍵。

Action	Shortcut
切換至全螢幕	<code>MOD+f</code>
左旋顯示螢幕	<code>MOD+←</code> (左)
右旋顯示螢幕	<code>MOD+→</code> (右)
縮放視窗成 1:1 (pixel-perfect)	<code>MOD+g</code>
縮放視窗到沒有黑邊框為止	<code>MOD+w</code> 雙擊 ¹
按下 首頁 鍵	<code>MOD+h</code> 中鍵
按下 返回 鍵	<code>MOD+b</code> 右鍵 ²
按下 切換 APP 鍵	<code>MOD+s</code>
按下 選單 鍵 (或解鎖螢幕)	<code>MOD+m</code>

按下 音量+ 鍵	MOD+↑ (上)
按下 音量- 鍵	MOD+↓ (下)
按下 電源 鍵	MOD+p
開啟	右鍵 ²
關閉裝置螢幕(持續鏡像)	MOD+o
開啟裝置螢幕	MOD+Shift+o
旋轉裝置螢幕	MOD+r
開啟通知列	MOD+n
關閉通知列	MOD+Shift+n
複製至剪貼簿 ³	MOD+c
剪下至剪貼簿 ³	MOD+x
同步剪貼簿並貼上 ³	MOD+v
複製電腦剪貼簿中的文字至裝置並貼上	MOD+Shift+v
啟用/停用 FPS 計數器(顯示於 stdout - 通常是命令列)	MOD+i

¹在黑邊框上雙擊以移除它們。

²右鍵會返回。如果螢幕是關閉狀態，則會打開螢幕。

³只支援 Android 7+。

所有 `Ctrl+按鍵` 快捷鍵都會傳送到裝置上，所以它們是由目前運作的應用程式處理的。

自訂路徑

如果要使用特定的 `adb`，將它設定到環境變數中的 `ADB`：

```
ADB=/path/to/adb scrcpy
```

如果要覆寫 `scrcpy-server` 檔案的路徑，則將路徑設定到環境變數中的 `SCRCPY_SERVER_PATH`。

[相關連結](#)

為何叫 *scrcpy*？

有一個同事要我找一個跟 [gnirehtet](#) 一樣難念的名字。

[strcpy](#) 複製一個字串 (**s**tring)；`scrcpy` 複製一個螢幕 (**s**creen)。

如何編譯？

請看[這份文件\(英文\)](#)。

常見問題

請看[這份文件 \(英文\)](#)。

開發者文件

請看[這個頁面 \(英文\)](#)。

Licence

```
Copyright (C) 2018 Genymobile
```

```
Copyright (C) 2018-2022 Romain Vimont
```

```
Licensed under the Apache License, Version 2.0 (the "License");  
you may not use this file except in compliance with the License.  
You may obtain a copy of the License at
```

```
    http://www.apache.org/licenses/LICENSE-2.0
```

```
Unless required by applicable law or agreed to in writing, software  
distributed under the License is distributed on an "AS IS" BASIS,  
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.  
See the License for the specific language governing permissions and  
limitations under the License.
```

相關文章

- [Scrcpy 簡介 \(英文\)](#)
- [Scrcpy 可以無線連線了 \(英文\)](#)