

# Google Chromecast 4.0

© Copyright by 3bird Projects 2022, <http://edukacja.3bird.pl>

## Ogólne

IP (Wi-Fi): DHCP;

Pamięć RAM: 2 GB;

Pamięć ROM: 4,4 GB;

Procesor: ARM Cortex A55 1,9MHz;

GPU: Mali G31;

System: Android 10;

Aplikacja inicjująca: *Google Home for Android*;

Formaty przesyłania: **mp4**, avi (z problemami).

## Jak wysłać obraz ze smartfona

Należy zainstalować aplikację **Google Home** ze sklepu *Google Play*. Zawiera ona przycisk „Prześlij ekran”.

## Jak wysłać obraz z Linuksa?

**1.** W systemie *Linux* nie trzeba w chwili obecnej instalować żadnej aplikacji. Jest już wbudowana w przeglądarkę **Google Chrome** jako domyślna wtyczka: *Ustawienia* : / *Przesyłaj...*

Możliwe jest przesyłanie karty przeglądarki lub całego ekranu (przesyłanie głosu możliwe jest jednak tylko w trybie karty).

**2.** Materiały video można także przysyłać za pomocą aplikacji **VLC Player: Playback / Render / TV**. Aplikacja powinna być skompilowana z udziałem flag: *chromecast, microdns* (dobrze jest aktywować je globalnie).

**3.** Wysłanie za pomocą skryptu Python.

# **emerge -vp pchromecast**

Konieczne jest uruchomienie lekkiego serwera www na laptopie. Rozwiązanie składa się z dwóch autorskich plików:

# **./chromecast-start** (skrypt w bash'u)

Jego zawartość:

```
#!/bin/bash
# Uruchamia w tle Pythonowy serwer WWW :
python -m http.server 7777 --directory /home/user/filmy/ > /dev/null 2>&1 &
echo "Serwer WWW został uruchomiony w tle..."
python chromecast.py
printf "\n\nAby zamknąć serwer, użyj polecenia 'kill PID':\n\n"
ps aux | grep http.server
```

Zawartość pythonowego skryptu *chromecast.py*:

```
import time
import pychromecast

film = input('\n\nPodaj nazwę filmu (razem z rozszerzeniem): ')
print('\n\nCzekaj...')
print('\033[1;31;40m\n\nUwaga! Jeśli film nie zostanie automatycznie wyświetlony,\nnależy w
Chromecast aktywować go w: Konto / Default Media Receiver.\033[0m\n')

# Znajduje urządzenia Chromecast w lokalnej sieci, ale nie łączy się z nimi:
services, browser = pychromecast.discovery.discover_chromecasts()

# Kończy wyszukiwanie urządzeń Chromecast:
pychromecast.discovery.stop_discovery(browser)

# Wykrywa i łączy się z urządzeniem Chromecast o nazwie "Salon":
chromecasts, browser
pychromecast.get_listed_chromecasts(friendly_names=["Salon"])
[cc.device.friendly_name for cc in chromecasts]

# Będzie nadawał do pierwszego wykrytego Chromecasta:
cast = chromecasts[0]
# Jeśli powyższe nie zadziała, można na zdefiniować Chromecasta po jego IP:
# cast = pychromecast.Chromecast("192.168.0.22")

# Zaczyna emisję i czeka, aż urządzenie będzie gotowe:
cast.wait()

# Można wyświetlić właściwości Chromecasta (Device Status):
# print(cast.device)

# Można wyświetlić status połączenia (sesji):
# print(cast.status)

# Zmienna "mc" jest skrótem od "MediaController":
mc = cast.media_controller

# Musi tu być adres http, a materiał video musi być w odpowiednim formacie (mp4):
# mc.play_media('http://192.168.0.2:7777/film.mp4', 'video/mp4')
mc.play_media(f"http://192.168.0.2:7777/{film}", 'video/mp4')

mc.block_until_active()

# Można wyświetlić informacje na temat przesyłanego materiału video (Media Status):
# print(mc.status)

mc.pause()
```

```
time.sleep(5)
mc.play()

# Zamknij wykrywanie urządzeń Chromecast:
pychromecast.discovery.stop_discovery(browser)
```

## Firewall

Urządzenie *Chromecast* ma otwarte następujące porty:

```
8008/tcp      http
8009/tcp      ajp13
8443/tcp      https-alt
9000/tcp      cslistener
9080/tcp      glrpc
10001/tcp     scp-config
```

W czasie połączenia, otwierane są też w naszym komputerze losowe porty z zakresu 32768:61000 (UDP), port 5353 (UDP), oraz port TCP, który określiliśmy przy otwieraniu naszego pythonowego serwera www. Aby zobaczyć zestawienie tych portów, należy wydać polecenie:

```
# netstat -tulpn
```

Aby urządzenie mogło być wykryte w sieci (5G) należy wyłączyć firewalla lub wprowadzić poniższe reguły. Reguła najbardziej ogólna (zawiera IP Chromecast'a):

```
# iptables -A INPUT -s 192.168.0.22/24 -j ACCEPT
```

Reguły bardziej szczegółowe (port 7777 to port naszego pythonowego serwera http.server):

```
# iptables -A INPUT -s 192.168.0.22/24 -p tcp --dport 7777 -j ACCEPT
# iptables -A INPUT -s 192.168.0.22/24 -p udp --dport 5353 -j ACCEPT
# iptables -A INPUT -s 192.168.0.22/24 -p udp --dport 32768:61000 -j ACCEPT
```

## Problemy i Tipsy

Komunikat błędu:

```
cast = chromecasts[0]
IndexError: list index out of range
```

Rozwiązanie: Połączenie blokowane jest przez firewalla.