
picast Documentation

Release 0.1

Hiroshi Miura

Apr 21, 2020

Contents:

1	A Simple Wireless Display Receiver on Raspberry Pi	1
1.1	Description	1
1.2	Dependency	1
1.3	Installation and run	1
1.4	Development	2
1.5	Preparation	2
1.6	Usage	2
1.7	Autostart	2
1.8	Debug	2
1.9	Known issues	3
1.10	License and copyright	3
2	Design	5
3	Modules	7
4	GStreamer	9
5	Threads	11
6	Tests	13
7	Logging	15
8	Settings	17
8.1	Section [logging]	17
8.2	Section [network]	17
8.3	Section [p2p]	18
8.4	Section [player]	18
8.5	Section [gst]	18
8.6	Section [wfd_parameter]	18
9	Resolutions data	19
10	authors	21
11	changeLog	23
11.1	Unreleased	23

11.2	v0.3.2	23
11.3	v0.3.1	24
11.4	v0.3	24
11.5	v0.2.4	24
11.6	v0.2.3	24
11.7	v0.2.2	24
11.8	v0.2.1	25
11.9	v0.2	25
11.10	v0.1	25
11.11	v0.0.1	26
12	Indices and tables	27

A Simple Wireless Display Receiver on Raspberry Pi

1.1 Description

picast is a simple wifi display receiver written by Python3 on Raspberry Pi.

1.2 Dependency

picast depends several external utilities and some are only on Raspbian.

- tvservice (raspbian only)
- udhcpd (debian/ubuntu only)
- wpa_supplicant
- wpa_cli
- vlc

1.3 Installation and run

Run apt install command on Raspbian(buster or later) / Raspberry Pi Zero W/WH, RaPi 3B+, RaPi 4.

```
$ sudo apt install net-tools python3 udhcpd python-gst-1.0 libgtk-3-dev python3-gi_
↳ girl.2-gtk-3.0
$ sudo apt install girl.2-gstreamer-1.0 girl.2-gst-plugins-base-1.0
$ sudo apt install gstreamer1.0-plugins-good gstreamer1.0-gtk3 gstreamer1.0-plugins-
↳ rtp
$ sudo apt install gstreamer1.0-omx-rpi gstreamer1.0-omx-rpi-config vlc
$ sudo apt install --no-install-recommends lxde
```

```
$ python3 -m pip install picast
$ picast
```

1.4 Development

```
$ sudo apt install gstreamer1.0-tools
$ git clone https://github.com/miurahr/picast.git picast
$ cd picast
$ python3 -m venv venv
$ source venv/bin/activate
$ pip install -U pip
$ pip install -r requirements.txt
$ picast --debug
```

1.5 Preparation

Increase GPU memory for decoding fullHD video stream. add `gpu_mem=128` to `/boot/config.txt`

1.6 Usage

Picast search for the wireless display named “picast” on the source device you want to cast. Use “12345678” for a WPS PIN number. It is recommended to initiate the termination of the receiver on the source side. After Pi connects to the source, it has an IP address of 192.168.173.80.

These parameters are configured in `settings.ini`.

1.7 Autostart

Edit `/home/pi/.config/lxsessions/LXDE/autostart`

```
@xscreensaver -no-splash
@lxterminal -l -e /home/pi/picast/bin/picast
```

1.8 Debug

There is a debug log at `/var/tmp/picast.log`. It is configured in `logging.ini`.

1.9 Known issues

- Latency: Limited by the implementation of the RTP player used.
- WiFi: The on-board WiFi chip on Pi 3/Zero W only supports 2.4GHz. Due to the overcrowded nature of the 2.4GHz spectrum and the use of unreliable rtp transmission, you may experience some video glitching/audio stuttering. It may be better with Pi 3B+/4 that support IEEE802.11ac 5GHz.
- HDCP(content protection): Neither the key nor the hardware is available on Pi and therefore is not supported.

1.10 License and copyright

- Copyright 2019 Hiroshi Miura
- Copyright 2018 Hsun-Wei Cho

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program. If not, see <http://www.gnu.org/licenses/>.

CHAPTER 2

Design

picast project try to provide miracast display sink/reciever functionarity on Raspberry Pi Zero W. There is no certification from WiFi Alliance, and it is built upon try-and-error basis with existent devices, there is no gurantee to compliance to WiFi alliance specifications.

picast consist with several modules.

- *picast.dhcpd* provide dhcp server function by running external *udhcpd* daemon with custom configuration.
- *picast.discovery* provide an interface to mDNS/SD network to register and query display sink and source.
- *picast.player* provide a RTP movie and audio player, supporting H.264 and AC3/AAC/LPCM audio codecs.
- *picast.settings* is an ini file loader and configuration provider.
- *picast.video* checks platform and provide proper wfd-video-formats parameter for miracast negotiation.
- *picast.wifip2p* setup *wpa_suplicant* to accept connection from miracast source.
- *picast.wpacli* provide methods to configure *wpa_suplicant* through *wpa-cli* command line.
- *picast.picast* is a core module to define controller and communicator with miracast source through RTSP.

CHAPTER 4

GStreamer

picast depends upon *GStreamer* video/audio handling framework and its python bindings. *picast* uses ‘OpenMAX’ codec API to decode H.264 video container.

CHAPTER 5

Threads

picast uses threading technic to realize a controller communication through RTSP and RTP player control

CHAPTER 6

Tests

There are several unittests for functions but it is not enough and there is possibility to exist defeat.

CHAPTER 7

Logging

picast defines module level logger and output many debug messages.

There are several configurations you can customize in *settings.ini*.

8.1 Section [logging]

logger

'logger' is a name of logging. You can change if you want. It is used for `logging.getLogger(<name>)` as a its <name>.

config

'config' is a filename of logging configuration. It should be ini style configuration that syntax is described in [configuring logging](#) section of python manual.

8.2 Section [network]

myaddress

'myaddress' is an IPv4 address to attach WiFi-P2P wifi device.

peeraddress

'peeraddress' is an IPv4 address to lease to source device by dhcp server.

netmask

'netmask' is an IPv4 netmask when leasing address.

rtsp_port

'rtsp_port' is a port to use for negotiation between sink and source.

rtp_port

'rtp_port' is a port to use displaying source image onto sink monitor.

8.3 Section [p2p]

device_name

‘device_name’ is used to display a name of receiver name on source device.

device_type

‘device_type’ is used to specify a *WiFi-P2P device_type*.

group_name

‘group_name’ is used to specify a *WiFi-P2P Group* name.

wps_mode

‘wps_mode’ is used to specify which mode for WPS configuration. *pbk* is for ‘Push the Button’ and *pin* is for static PIN used.

pin

‘pin’ for wps_mode=pin configuration.

timeout

‘timeout’ for wps_mode=pin configuration.

8.4 Section [player]

name

‘name’ of player, one of *vlc* or *gst* is accepted.

8.5 Section [gst]

decoder

‘decoder’ is a gst plugin name to decode H.264/H.265 video. *omxh264dec* is recommended for Raspberry Pi video chip. It is used when player name is specified as *gst*.

8.6 Section [wfd_parameter]

wfd_audio_codecs

It is used as a value for *wfd_audio_codecs* parameter when negotiation. default value is *AAC 00000001 00, LPCM 00000002 00*

wfd_connector_type

It is used as a value for *wfd_connector_type* parameter when negotiation. HDMI is *05*.

CHAPTER 9

Resolutions data

There is a *resolutions.json* file that describe Raspberry Pi video resolutions data. It is used to determine which video mode is supported in current RaPi device and monitor device.

Resolutions are retrived by *tvservice* command. *code* number is used to determine its output. *mode* is a WiFi Direct video mode corresponding with specified resolution.

If you want to porting picast to another device, you should create a new json file and update video.py module.

CHAPTER 10

authors

PiCast is written and maintained by Hiroshi Miura <miurahr@linux.com>

Original lazycast is written by Hsun-Wei Cho and its contributors

Contributors, listed alphabetically, are:

CHAPTER 11

changeLog

All notable changes to this project will be documented in this file.

11.1 Unreleased

11.1.1 Added

11.1.2 Changed

11.1.3 Fixed

11.1.4 Deprecated

11.1.5 Removed

11.1.6 Security

11.2 v0.3.2

11.2.1 Fixed

- Fix ip address error for zeroconf constructor(#9).

11.3 v0.3.1

11.3.1 Fixed

- Fix settings.ini not found error on distribution.

11.4 v0.3

11.4.1 Changed

- Release artifacts with github actions.
- Use setuptools_scm to automate version.
- Packaging metadata in setup.cfg
- Auto generate binary by build backend.

11.4.2 Fixed

- Several document errors.

11.5 v0.2.4

11.5.1 Added

- Comments for parameter configs.
- Parameters Microsoft V2 extensions.

11.6 v0.2.3

11.6.1 Changed

- Change class name RtspSink to RTSPSink

11.6.2 Fixed

- Watching socket timeout and break connection.

11.7 v0.2.2

11.7.1 Changed

- Update README to use pypi.org for distribution.

- Refactoring RTSPSink class

11.7.2 Fixed

- Fix config loading error.

11.8 v0.2.1

11.8.1 Added

- Introduce [p2p] wps_mode= in settings.ini. can be 'pbc' ie. Push the Button, or 'pin'
- Add documentation for settings.

11.8.2 Changed

- Add p2p_connect() and start_wps_pbc() methods.
- Improve CLI staff.

11.8.3 Fixed

- main() drop unused asyncio staff.

11.9 v0.2

11.9.1 Added

- Test for python 3.7 and 3.8 - Install PyObject through pip. - Add network protocol functional tests.
- Test typing with mypy. - dependency for gst-python-stubs and PyObject-stubs

11.9.2 Changed

- Introduce RTSPTransport class to handle network connection.

11.9.3 Fixed

- Bug: fails to recieve M4 message because M3 reciever can read both M3 and M4.

11.10 v0.1

- First working release.

11.11 v0.0.1

- Forked from lazycast.

CHAPTER 12

Indices and tables

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