User manual

（RTSP Pusher）

Happytimesoft Technology Co., LTD

Declaration

Since the product version upgrade or other reasons, this manual will subsequently be updated. Unless otherwise agreed, this manual only as a guide, this manual all statements, information, recommendations do not constitute any express or implied warranties。

Table of Contents

Chapter 1 Introduction

Happytime RTSP pusher is an rtsp streaming push program that supports RTP OVER UDP, RTP OVER TCP, RTP OVER RTSP three push modes, supports streaming screen, camera, local audio and video files, video payload format supports MPEG4, MJPEG, H264 and H265 The audio load format supports G711, G722, G726, OPUS, AAC. The audio and video parameters pushed can be set through configuration files, such as video resolution, frame rate, audio sampling rate, number of channels, etc., to support simultaneous push of multiple streams. Stable flow, low resource consumption. Support multi-platform, support windows, linux, macos, etc., support cross-compilation, support embedded development, provide live stream stub processing class, embedded developers only need to implement several functions simply Can migrate very quickly to meet project requirements.

Chapter 2 Key features

Support a variety of audio and video files

Support push video from camera and living screen

Support push audio from audio device

Support media stream, include RTSP/RTMP/HTTP MJPEG stream

Support for configuring audio and video output parameters

Small size, suitable for embedded development

Code structure clear, easy to use

Chapter 3 Configuration

3.1 Configuration Templates

```
<?xml version="1.0" encoding="utf-8"?>
<config>
    <log_enable>1</log_enable>
    <log_level>1</log_level>
    <loop_nums>-1</loop_nums>
    <reconn_interval>15</reconn_interval>
    <pusher>
        <src>screenlive</src>
        <transfer>
            <mode>tcp</mode>
            <rtspurl>rtsp://192.168.1.50/pusher</rtspurl>
            <user></user>
            <pass></pass>
            <ip>192.168.1.50</ip>
            <vport>50001</vport>
            <aport>50002</aport>
        </transfer>
        <video>
            <codec>H264</codec>
            <width></width>
            <height></height>
            <framerate></framerate>
            <bitrate></bitrate>
        </video>
        <audio>
            <codec>G711A</codec>
            <samplerate>8000</samplerate>
            <channels>1</channels>
            <bitrate></bitrate>
        </audio>
    </pusher>
    <pusher>
        <src>1.mp4</src>
        <transfer>
            <mode>udp</mode>
```

```
            <rtspurl>rtsp://192.168.1.50/pusher1</rtspurl>
            <user></user>
            <pass></pass>
            <ip>192.168.1.50</ip>
            <vport>50001</vport>
            <aport>50002</aport>
        </transfer>
        <video>
            <codec>H264</codec>
            <width></width>
            <height></height>
            <framerate></framerate>
            <bitrate></bitrate>
        </video>
        <audio>
            <codec>G711A</codec>
            <samplerate>8000</samplerate>
            <channels>1</channels>
            <bitrate></bitrate>
        </audio>
    </pusher>
</config>
```

3.2 Configuring Node Description

3.2.1 System parameters

<log_enable>

Whether enable the log function,0-disable,1-enable

<log_level>

The log level:

TRACE	0
DEBUG	1
INFO	2
WARN	3
ERROR	4
FATAL	5

<loop_nums>

When streaming local media files, specify the number of loop playback, -1 means infinite loop.

<reconn_interval>

When using RTSP push mode, reconnect interval after disconnection.

3.2.2 *Pusher* node

<*Pusher*> : Represents a push stream, specify the audio and video output parameters, it can configure multiple nodes

<src>

The push stream source, it supports the following parameters:

filename: local media file name, the media files need to be placed in the directory where rtsppusher is located

rtsp,rtmp or http mjpeg url: the original rtsp/rtmp/http mjpeg stream address, such as rtsp://user:pass@ip:port/url or rtmp://user:pass@ip:port/app/stream.

screenlive : stream the living screen

videodevice : stream from the camera

audiodevice: stream from the audio device

screenlive+audiodevice: stream the living screen and stream audio from the audio device

videodevice+audiodevice: stream video from the video device and stream audio from the audio device

If your system have multiple video capture device, you can use

videodevice*N, the N to specify the video capture device index, start from 0, such as:*

videodevice ; stream video from the first video device

videodevice1 ; stream video from the second video device

The audio device and screen similar to the video device

<transfer> : Specify the transfer parameters

<mode>

Specify transfer mode, Support the following

UDP: Transfer using UDP protocol

TCP: Transfer using TCP protocol

RTSP: Transfer using RTSP protocol

<rtspurl>

Pushed RTSP destination address

<user>

RTSP destination address authentication username

<pass>

RTSP destination address authentication password

<ip>

Specify the server IP address if using TCP or UDP mode.

<vport>

Specify the video data receiving port if using TCP or UDP mode

<aport>

Specify the audio data receiving port if using TCP or UDP mode

<video> : Specify the video output parameters

<codec>

Specify the video stream codec, it can specify the following value:

H264 : output H264 video stream

H265 : output H265 video stream

MP4: output MP4 video stream

JPEG: output MJPEG video stream

<width>

Specify the output video width, if 0 it use the original video width (living screen use the screen width, camera use the default width)

<height>

Specify the output video height, if 0 it use the original video height (living screen use the screen height, camera use the default height)

<framerate>

Specify the output video framerate, if 0 it use the original video framerate (living screen use the default value 15, camera use the default value 25)

<bitrate>

Specify the output video bit rate, if it is 0, the bit rate is automatically calculated.

<audio> : Specify the audio output parameters

<codec>

Specify the audio stream codec, it can specify the following value:

G711A: output G711 a-law audio stream

G711U: output G711 mu-law audio stream

G722: output G726 audio stream

G726: output G726 audio stream

AAC: output AAC audio stream

OPUS: output OPUS audio stream

<samplerate>

Specify the audio sample rate, it can specify the following values:

8000, 11025, 12000, 16000, 22050, 24000, 32000, 44100, 48000

If 0 it use the original audio sample rate (audio device use the default value 8000)

<channels>

Specify the audio channel number, 1 is mono, 2 is stereo

If 0 it use the original audio channel number (audio device use the default value 2)

<bitrate>

Specify the output audio bit rate, if it is 0, the bit rate is automatically calculated.

Chapter 4 Run RTSP Pusher

The rtsp pusher is a console application.

Windows: to run the server, simply type "rtsppusher".

Linux: to run the rtsp pusher, type "./start.sh", on linux platform, rtsp pusher run as deamon by default.

Note : The demo version has the following limitations

Maximum support two simultaneous pusher stream.