

# RTSP real-time image transmission

---

## RTSP real-time image transmission

Routine explanation

Introduction

Prepare

Startup routine

Open the player

Routine code

## Introduction to RTSP (Real Time Streaming Protocol)

What is RTSP?

Main Features

Common application scenarios

RTSP vs other streaming protocols

Protocol Comparison

RTSP Workflow

RTSP Architecture

RTSP transmission mode

Advantages

Disadvantages

## Routine explanation

---

### Introduction

In this section, we will demonstrate how to push the images captured by K230 in the form of RTSP stream. We can use a video player that supports playing RTSP streams to play the real-time transmitted images.

Note: The WIFI chip on the K230 has limited performance. Please use the image transmission near a WIFI signal hotspot.

\*This example is experimental and may not run with high quality in different network environments.

If you want to improve the video fluency, you can modify the `bit_rate` parameter in this line of code. The recommended value range of `bit_rate` is 10 ~ 300

**The smaller the `bit_rate` value, the better the fluency, but the lower the clarity.**

```
# 创建编码器 / Create encoder
chnAttr = ChnAttrStr(self.encoder.PAYLOAD_TYPE_H264,
self.encoder.H264_PROFILE_MAIN, width, height, bit_rate=10,
dst_frame_rate=5, src_frame_rate=5)
```

## Prepare

We can download a player that supports playing RTSP streams. The options for Windows systems are

1. VLC player (vlc-3.0.21-win64.exe in the supporting information, double-click to install)

Advantages: good compatibility, powerful functions

Disadvantages: The configuration is numerous and complicated. If you don't know how to configure it, the display effect will be poor.

2. YAHBOOM Rtsp Player (built into the companion gadget program)

Advantages: No configuration required, one-click use, low latency

Disadvantages: Large size

It is recommended to download both. In general, use YAHBOOM Rtsp Player. If YAHBOOM Rtsp Player fails, use Vlc player to debug.

I haven't tried it on other systems yet. You can search for RTSP players supported by xxx systems. I have tested that Vlc for Android can play RTSP videos on Android phones, but the delay is very high. I haven't found a configuration method yet.

## Startup routine

RTSP image transmission has high requirements on network quality. Please try to choose a WIFI with few devices mounted on the router and strong signal.

**For the image transmission function, it is recommended to use the mobile phone to open the hotspot (WIFI and data can be turned off), select the 2.4GHz frequency, and place the K230 next to the mobile phone.**

It is normal for the image transmission to freeze, and it can be optimized in the following ways:

1. Check if the WIFI is using 2.4Ghz. The WIFI chip on the K230 only supports 2.4Ghz wireless signals, not 5Ghz or 2.4/5Ghz mixed signals.
2. Check if there are too many devices mounted on the router

Please put K230 and computer in the same LAN (connect to the same WIFI), and modify the parameters of this line to the WIFI name and password.

```
# The WIFI name I connect to is test, and the password is 12345678
isConnected = Connect_WIFI("test","12345678")
```

Then click the Run button, wait for WIFI to connect and start the RTSP server

FAQ:

1. No response after clicking the Run button

Solution: Turn off the power of K230, wait for 5 seconds, and then plug it in and run it. K230 needs to be restarted after each RTSP routine is completed.

2. Error: run connect failed.

Solution: Turn off the power of K230, wait for 5 seconds, and then plug it in and run it. Under normal circumstances, repeat this operation up to three times to restore it to normal.

If the error message "run connect failed" is displayed repeatedly, the WIFI chip is probably damaged.

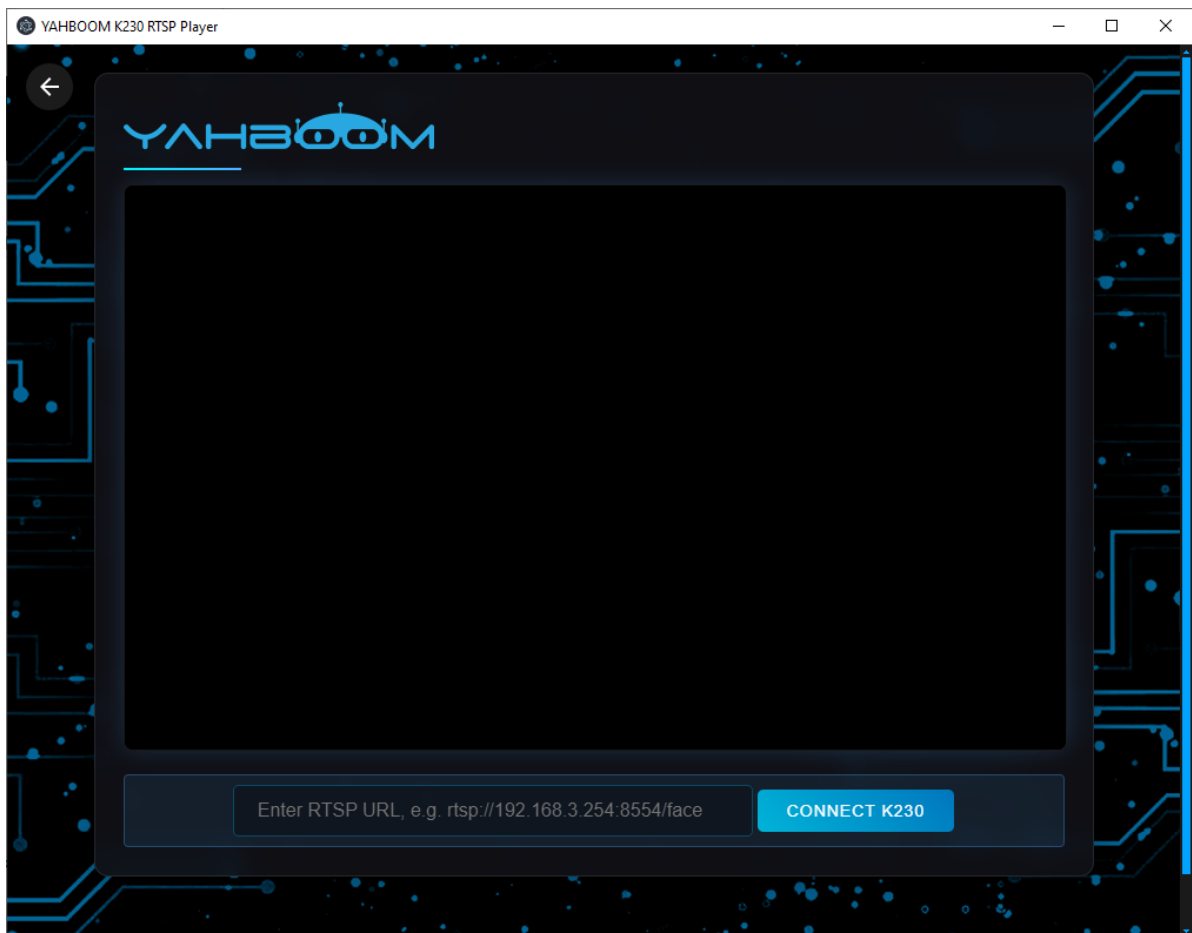
If it starts normally, the K230 serial terminal will output the following

```
[WIFI] 连接网络中 ...  
[WIFI] 连接网络成功  
[RTSP] 启动中 ..  
find sensor gc2093_csi2, type 24, output 1920x1080@60  
buffer pool : 3  
sensor(0), mode 0, buffer_num 4, buffer_size 0  
[RTSP] 启动成功, 地址: rtsp://192.168.207.22:8554/video
```

This address is the address of your RTSP streaming.

## Open the player

Take YAHBOOM Rtsp Player as an example. Double-click to run the program. The interface is as follows:

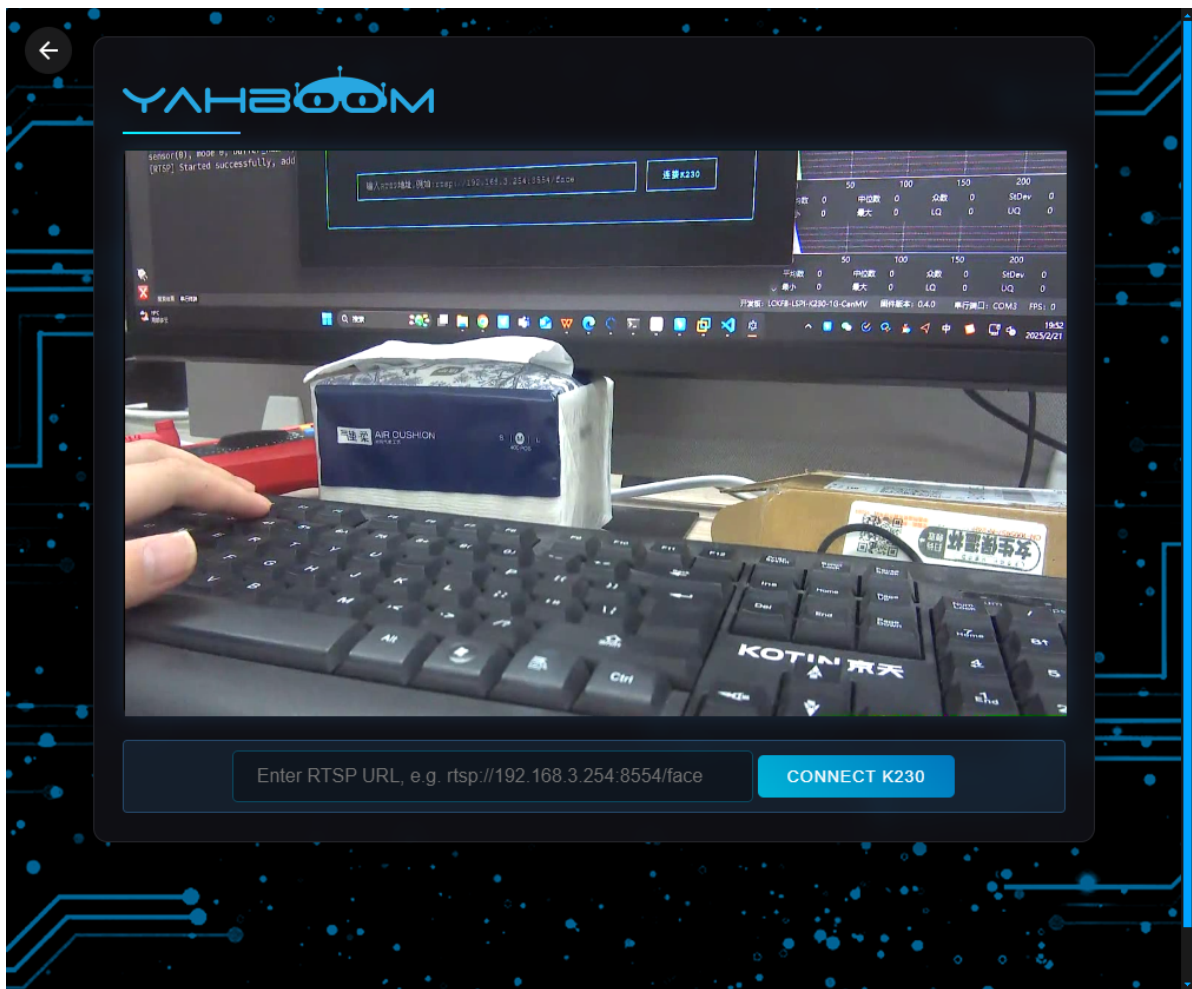


Note: The actual interface may be slightly different, but the overall functionality is the same



After clicking the connection, it will take some time to load. The loading speed is related to the network speed.

If there is still no response after waiting for 20 seconds, try playing it with VLC Player. If there is still no response, unplug the USB cable to power off, plug it in again after 10 seconds, and re-run the example code



## Routine code

For the complete code, please refer to the file: [Source Code/11.Network/07.rtsp/rtsp\_no\_ai.py]

1. Overall structure and function:

- The code mainly implements video capture, encoding and RTSP streaming functions
- Contains two main parts: WiFi connection module and RTSP server class
- Use multithreading to process video stream data

1. Key classes and methods:

The core members of the RtspServer class are:

```
def __init__(self):
    self.session_name = session_name # 会话名称 Session name
    self.video_type = video_type     # 视频类型(H.264/H.265) video type
    self.enable_audio = enable_audio # 是否启用音频 whether to enable audio
    self.port = port                  # RTSP 端口号 RTSP port number
    self.rtspserver = mm.rtsp_server() # RTSP服务器实例 RTSP server instance
    self.venc_chn = VENC_CHN_ID_0     # 视频编码通道 Video encoding channel
```

The main methods include:

- start(): Start the server
- stop(): Stop the server
- \_init\_stream(): Initialize the video stream

- `_do_rtsp_stream()`: The core thread for processing video streams

#### 1. Video processing flow:

##### a) Collection phase:

```
# Collect images through sensor
# 通过sensor采集图像
rtsp_show_img = self.sensor.snapshot()

# Set frame information
# 设置帧信息
frame_info.v_frame.width = rtsp_show_img.width()
frame_info.v_frame.height = rtsp_show_img.height()
frame_info.v_frame.pixel_format = Sensor.YUV420SP
```

##### b) Coding stage:

```
# Send frames to encoder
# 发送帧到编码器
self.encoder.SendFrame(self.venc_chn, frame_info)
# Get the encoded stream
# 获取编码后的流
self.encoder.GetStream(self.venc_chn, streamData)
```

##### c) Streaming stage:

```
# Send the encoded data to the RTSP server
# 将编码数据发送到RTSP服务器
self.rtspserver.rtspserver_sendvideodata(self.session_name, stream_data,
streamData.data_size[pack_idx], 1000)
```

#### 1. Optimized design:

- Use multithreading to process video streams to avoid blocking
- Aggressive garbage collection
- Set up the video buffer
- Support image processing of different resolutions

#### 1. Error handling:

- WiFi connection failure handling
- Video stream exception handling
- Resource release mechanism

#### 1. Example usage:

```
if __name__ == "__main__":  
    # Connect to WiFi  
    # 连接WiFi  
    isConnected = Connect_WIFI("test", "password")  
  
    # Create and start the RTSP server  
    # 创建并启动RTSP服务器  
    rtspserver = RtspServer()  
    rtspserver.start()  
  
    # Get and print RTSP address  
    # 获取并打印RTSP地址  
    rtsp_address = rtspserver.get_rtsp_url()
```

# Introduction to RTSP (Real Time Streaming Protocol)

---

## What is RTSP?

---

RTSP is a network application layer protocol for controlling streaming media servers, defined by RFC 2326. It establishes and controls one or more time-synchronized continuous media streams.

## Main Features

---

- Strong real-time performance
- Support on-demand and live broadcast
- Support streaming media control (play, pause, fast forward, etc.)
- Works at the application layer, using port 554 by default

## Common application scenarios

---

### 1. Video surveillance system

- Security Camera
- Industrial Monitoring
- Traffic monitoring

### 2. Live Streaming

- videoconference
- Online Education
- Telemedicine

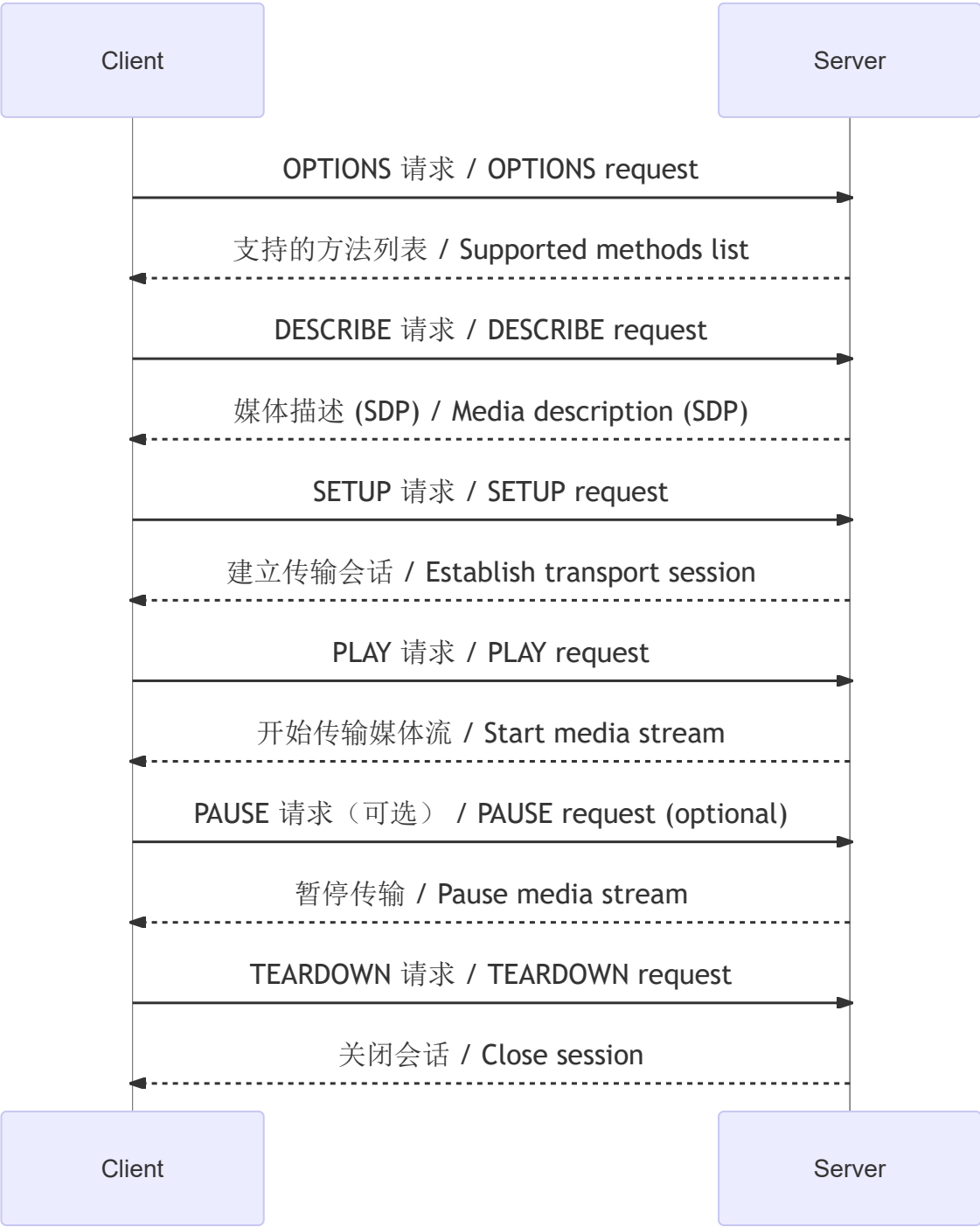
## RTSP vs other streaming protocols

---

## Protocol Comparison

protocol	Delay	reliability	Interactivity	Application Scenario
RTSP	Low	middle	high	Video surveillance, on-demand
HLS	high	high	Low	Live broadcast, on-demand
RTMP	Low	middle	middle	Live streaming
WebRTC	Very low	middle	high	Real-time communication

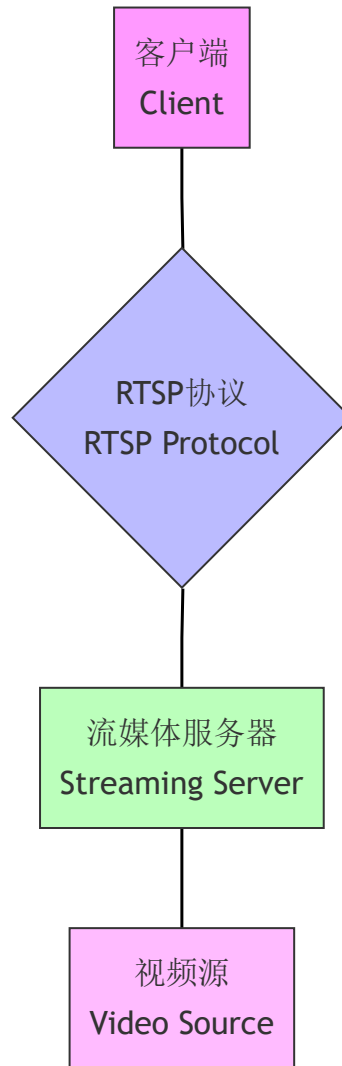
## RTSP Workflow





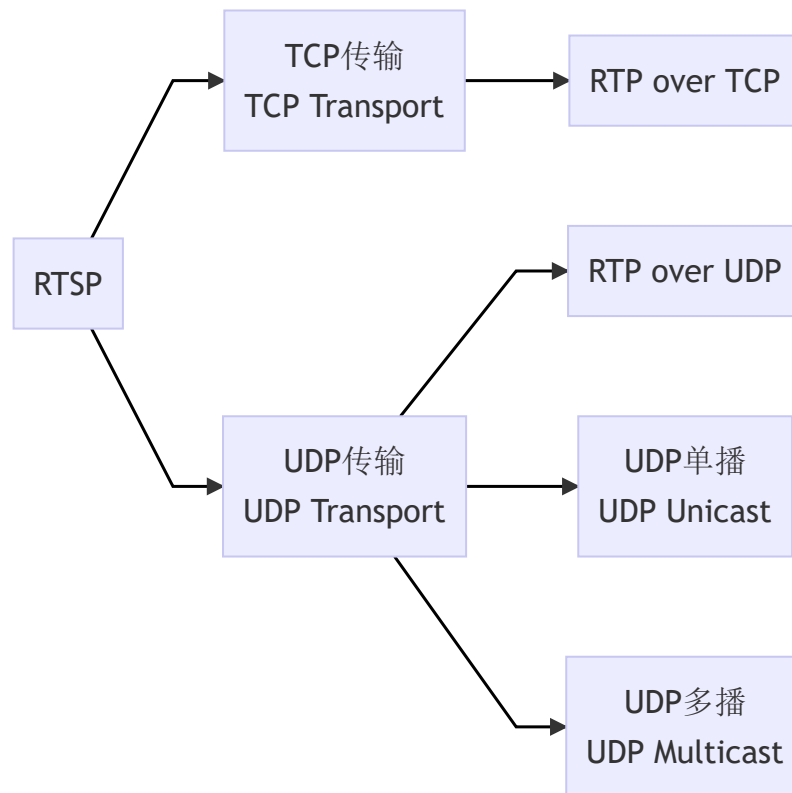
## RTSP Architecture

---



## RTSP transmission mode

---



## Advantages

---

1. Good real-time performance and low latency
2. Support streaming media control function
3. The bandwidth usage is relatively small
4. Suitable for on-demand scenarios

## Disadvantages

---

1. Requires a dedicated streaming server
2. Might be blocked by a firewall
3. Does not support adaptive bitrate
4. Web browsers do not natively support