

# T/HBSGX

## 湖北省光电显示行业协会团体标准

T/HBSGX 005—2025

### 视频软件接口与数据交换规范

Specification for video software interface and data exchange

2025 - 12 - 16 发布

2025 - 12 - 16 实施

## 前 言

本文件按照GB/T 1.1—2020《标准化工作导则 第1部分：标准化文件的结构和起草规则》的规定起草。

请注意本文件的某些内容可能涉及专利。本文件的发布机构不承担识别专利的责任。

本文件由成都云文未来科技有限公司提出。

本文件由湖北省光电显示行业协会归口。

本文件起草单位：成都云文未来科技有限公司、成都启航新汇科技有限公司、成都颜火科技有限公司、成都智通达诚科技有限公司、成都威尔纳思科技有限公司

本文件主要起草人：王小纲、江苙、田际燕、彭世伟、曹捷。

# 视频软件接口与数据交换规范

## 1 范围

本文件规定了视频软件接口的类型、数据交换规范以及相关的安全要求。

本文件适用于视频采集、编码、存储、传输、播放、处理、分析等环节的软件系统开发、集成和互操作。

## 2 规范性引用文件

下列文件中的内容通过文中的规范性引用而构成本文件必不可少的条款。其中，注日期的引用文件，仅该日期对应的版本适用于本文件；不注日期的引用文件，其最新版本（包括所有的修改单）适用于本文件。

GB/T 22239 信息安全技术 网络安全等级保护基本要求

GB/T 28181 公共安全视频监控联网系统信息传输、交换、控制技术要求

GB/T 39786 信息安全技术 信息系统密码应用基本要求

## 3 术语和定义

下列术语和定义适用于本文件。

### 3.1

**视频软件 video software**

用于对视频数据进行采集、编码、存储、传输、播放、处理、分析等操作的计算机程序及其相关组件的集合。

### 3.2

**接口 interface**

软件模块之间进行通信和交互的连接点，包括函数、方法、应用程序编程接口（API）、消息格式等。

### 3.3

**数据交换 data exchange**

在不同软件系统或模块之间传输和共享信息的过程。

### 3.4

**元数据 metadata**

描述视频数据内容、特征、上下文等信息的结构化数据，例如时间戳、地理位置、摄像机ID、事件类型、分析结果等。

## 4 缩略语

下列缩略语适用于本文件。

API: 应用程序编程接口 (application programming interface)

AVI: 音频视频交错格式 (audio video interleave)

DASH: 基于HTTP协议的自适应流媒体传输标准 (dynamic adaptive streaming over HTTP)

FLV: 流媒体文件格式 (flash video)

FTP: 文件传输协议 (file transfer protocol)

HLS: 基于HTTP的自适应码率流媒体传输协议 (HTTP live streaming)

HTTP: 超文件传输协议 (hypertext transfer protocol)

HTTPS: 安全超文件传输协议 (hypertext transfer protocol secure)

JSON: JavaScript对象表示法 (JavaScript object notation)

MQTT: 消息队列遥测传输 (message queuing telemetry transport)  
 RTCP: 实时传输协议 (RTP) 的配套控制协议 (real-time transport control protocol)  
 RTMP: 实时消息传输协议 (real-time messaging protocol)  
 RTP: 实时传输协议 (real-time transport protocol)  
 RTSP: 实时流传输协议 (real-time Streaming protocol)  
 SFTP: 基于SSH (Secure Shell) 的安全文件传输协议 (SSH File transfer protocol)  
 XML: 可扩展标记语言 (extensible markup language)

## 5 视频软件接口

### 5.1 接口设计原则

#### 5.1.1 接口兼容性

接口应具备向下兼容性, 新版本接口在保留旧版本核心功能的基础上进行扩展, 确保基于旧版本接口开发的软件能够正常运行。同时, 接口应支持与主流视频编码标准 (H.264、H.265等) 及传输协议 (HDMI、以太网等) 的兼容。

#### 5.1.2 可靠性

接口应具备完善的错误处理、异常捕获和重试机制, 确保在网络中断、服务不可用、数据格式错误等异常情况下的稳定性和可靠性。应支持幂等性设计, 避免重复操作引起的不一致。

#### 5.1.3 实时性

视频数据传输接口, 端到端的传输延迟应不超过50 ms; 对于控制指令接口, 指令的响应时间应不超过10 ms, 确保视频显示的流畅性和控制的及时性。

#### 5.1.4 安全性

接口应支持数据加密传输功能, 对于敏感控制指令和视频数据, 采用加密算法 (如AES) 进行加密处理; 同时, 接口应具备权限验证机制, 防止未授权访问和非法操作。

#### 5.1.5 可扩展性

接口设计应预留扩展接口和参数配置空间, 支持新增视频编码格式、传输协议及功能模块的接入, 无需对原有接口结构进行大规模修改。

### 5.2 接口分类

#### 5.2.1 功能接口

##### 5.2.1.1 视频采集与控制接口

用于控制视频源的启动、停止、参数设置和状态查询:

- 启动/停止采集: 控制视频源的开启和关闭;
- 参数配置: 设置视频分辨率、帧率、码率、编码格式、曝光、白平衡、焦距等参数;
- 状态查询: 查询视频源的连接状态、工作模式、当前配置等信息。

##### 5.2.1.2 视频编码与解码接口

用于调用视频编码器进行视频数据压缩, 或调用解码器进行视频数据解压:

- 编码器调用: 输入原始视频帧, 输出符合特定编码标准的压缩视频数据, 支持 H.264 等主流编码标准;
- 解码器调用: 输入压缩视频数据, 输出原始视频帧;
- 参数设置: 配置编码器 (如 I 帧间隔、B 帧数量、QP 值、码率控制模式) 和解码器参数。

##### 5.2.1.3 视频存储与检索接口

用于视频数据的写入、读取、删除、查询和检索:

- 存储写入：将实时视频流或文件保存至指定存储介质（如磁盘、云存储）；
- 存储读取：从存储介质中读取视频数据；
- 存储删除：删除指定的视频文件或录像片段；
- 查询与检索：基于时间范围、事件类型、元数据标签、设备 ID、存储位置等条件，查询和检索录像片段或视频文件。支持模糊查询和组合查询。

#### 5.2.1.4 视频传输接口

用于实时或非实时视频数据的传输，包括拉流、推流等操作：

- 推流：将视频数据从源端主动推送到目标服务器或客户端；
- 拉流：客户端从服务器请求并接收视频数据；
- 流地址管理：生成、管理和销毁视频流的传输地址（如 RTSP URL、RTMP URL）；
- 传输控制：控制流媒体的开始、停止、暂停、快进、倒退等。

#### 5.2.1.5 视频播放与渲染接口

用于视频数据的播放控制和屏幕渲染：

- 播放控制：控制视频流或文件的播放（开始、暂停、停止、快进、倒退、跳转到指定时间点）；
- 渲染输出：将解码后的视频帧渲染到指定的显示区域；
- 音视频同步：确保播放过程中音视频的同步性；
- 播放状态查询：查询当前播放进度、缓冲状态、音量等信息。

#### 5.2.1.6 视频处理与分析接口

用于调用视频处理算法和视频分析算法：

- 图像处理：提供图像增强、去噪、锐化、色彩校正、水印叠加、区域裁剪等功能；
- 视频分析任务管理：启动、停止、配置和查询视频分析任务（如人脸检测、车辆识别、异常行为分析、区域入侵检测）；
- 分析结果获取：获取视频分析任务产生的结构化元数据（如识别结果、告警信息、目标轨迹）。

#### 5.2.2 管理与配置接口

用于系统级别的配置、日志管理、用户认证和权限管理、系统状态监控等：

- 系统配置：配置系统参数、网络设置、存储策略、告警规则等；
- 用户与权限管理：添加、删除、修改用户账户，分配和管理用户角色及权限；
- 日志管理：查询、导出、清除系统运行日志、操作日志、安全日志；
- 系统监控：获取系统资源使用情况（CPU、内存、带宽）、服务状态、设备健康状况等信息。

#### 5.2.3 消息通知接口

用于事件触发、告警信息、系统状态变更等异步消息的发布和订阅：

- 事件订阅/取消订阅：允许客户端订阅或取消订阅特定类型的事件（如移动侦测告警、设备上/下线通知、分析结果通知）；
- 消息发布：系统将发生的事件和告警信息以标准格式发布到消息队列或通过 WebSocket 推送给订阅者；
- 消息确认：对于关键消息，支持消息确认机制。

### 5.3 基本要求

#### 5.3.1 视频输入接口

##### 5.3.1.1 数字视频输入接口

支持 HDMI、LVDS 等主流数字视频输入协议，接口参数应符合对应协议的标准要求，包括但不限于：

- HDMI 接口：支持 HDMI 2.1 以下版本，并支持 HDCP 2.3 加密协议；

——LVDS 接口：支持单通道或多通道传输，单通道传输速率宜不低于 1.5Gbps，支持分辨率最高至 4K@60Hz。

### 5.3.1.2 模拟视频输入接口

支持复合视频（CVBS）、S-视频等模拟视频输入格式。

## 5.3.2 视频输出接口

### 5.3.2.1 显示驱动接口

应支持RGB、MIPI DSI等显示驱动接口，用于连接显示面板，接口规范包括但不限于：

——RGB 接口：支持 8bit/16bit/24bit 色彩深度，像素时钟频率范围为 10 MHz~200 MHz；

——MIPI DSI 接口：支持 MIPI DSI 1.3 及以上版本，数据通道数量为 1~4 通道可配置，并支持命令模式和视频模式两种工作模式。

### 5.3.2.2 视频转发接口

支持以太网、Wi-Fi等网络视频转发接口，用于将接收的视频数据转发至其他设备，接口规范包括但不限于：

——以太网接口：支持 10/100/1000Mbps 自适应速率，支持 TCP/UDP 协议；

——Wi-Fi 接口：支持 IEEE 802.11a/b/g/n/ac/ax 标准。

## 5.3.3 控制接口

### 5.3.3.1 本地控制接口

支持UART、SPI、I2C等本地控制接口，用于连接本地控制模块（如按键、旋钮、MCU等），接口规范包括但不限于：

——UART 接口：支持波特率为 9600/19200/38400/115200bps 可配置，数据位为 8 位，停止位为 1 位，校验位可选择无校验/奇校验/偶校验；

——I2C 接口：支持标准模式和快速模式；

——SPI 接口：支持主模式/从模式，全双工通信。

### 5.3.4 远程控制接口

支持HTTP/HTTPS、MQTT等远程控制协议，用于通过网络接收远程控制指令，接口规范包括：

——HTTP/HTTPS 接口：支持 HTTP 1.1/2.0 及 HTTPS 1.2/1.3 协议，接口宜采用 RESTful 风格设计，请求方法支持 GET、POST、PUT、DELETE，响应格式支持 JSON/XML；

——MQTT 接口：支持 MQTT 3.1.1/5.0 协议，支持 QoS 0/1/2 三种服务质量等级，支持遗留消息和保留消息功能。

### 5.3.5 状态反馈接口

状态反馈接口用于向外部设备或系统反馈嵌入式显示终端视频软件的运行状态，包括接口连接状态、视频信号质量、软件运行状态等，接口规范如下：

——状态信息类型：包括设备在线状态、视频输入/输出接口连接状态、视频码流参数（分辨率、帧率、码率）、软件错误信息、资源占用率（CPU、内存）等；

——反馈频率：常规状态信息反馈频率为 1 次/秒，异常状态信息应在状态变化后 100 ms 内立即反馈；

——数据格式：状态信息采用 JSON 格式封装。

## 6 数据交换规范

### 6.1 数据格式

#### 6.1.1 视频数据格式

### 6.1.1.1 原始视频数据

未编码的视频帧数据格式，主要用于内部处理或与编解码器交互。

- YUV 格式：用于视频处理和编码输入；
- RGB 格式：用于显示渲染。

### 6.1.1.2 编码视频数据

符合特定编码标准的压缩视频数据格式。

- H.264 (AVC)；
- H.265 (HEVC)；
- VP8/VP9；
- AVI。

### 6.1.1.3 封装格式

包含视频、音频和元数据等的多媒体封装格式，用于存储和传输，宜采用MP4、TS、MKV或AVI格式。

## 6.1.2 元数据格式

### 6.1.2.1 通用元数据

宜采用JSON或XML格式进行结构化描述。

### 6.1.2.2 业务元数据

根据具体应用场景定义的业务相关元数据，如识别结果、事件类型、告警级别、目标坐标、分析模型ID等。宜采用JSON或XML格式。

## 6.1.3 控制命令与响应格式

宜建议采用JSON或XML格式。

- 请求体：客户端发送给服务器的控制命令参数；
- 响应体：服务器返回给客户端的操作结果、状态信息或查询数据。

## 6.2 数据交换协议

### 6.2.1 实时流媒体协议

应根据应用场景选用相应的协议：

- RTSP，主要用于控制音视频流的传输，本身不传输媒体数据，需配合 RTP/RTCP 完成数据传输与质量监控；
- WebRTC，支持浏览器/客户端间的实时音视频和数据传输，无需额外插件；
- RTMP，采用握手-连接-流传输的流程，支持将音视频封装为 FLV 格式传输，可实现低延迟推流和拉流；
- HLS，基于 HTTP，可穿透防火墙，兼容性极强（支持所有浏览器和移动设备），适合点播和非实时性直播（如赛事回放）；
- DASH，适配不同网络带宽自动切换码率，多用于广电、高端视频平台。

### 6.2.2 控制与管理协议

#### 6.2.2.1 HTTP/HTTPS 协议

HTTP用于RESTful API的调用，实现各种控制命令和状态查询。

#### 6.2.2.2 WebSocket 协议

用于建立客户端和服务端之间的实时双向通信通道，适用于需要长连接和实时推送的场景。

#### 6.2.2.3 MQTT 协议

轻量级的消息发布/订阅协议，适用于物联网、移动互联网和低带宽、高延迟环境下的消息交换，特别适合设备状态上报、告警通知等。

### 6.2.3 文件传输协议

#### 6.2.3.1 FTP

用于视频文件或录像的上传下载。

#### 6.2.3.2 SFTP

基于SSH协议的安全文件传输协议，提供加密和认证，推荐用于敏感文件传输。

## 7 安全要求

### 7.1 认证与授权

#### 7.1.1 用户认证

应支持多种可靠的用户身份认证机制，防止未经授权的访问。

- 用户名/密码：基本认证方式，密码应进行加密存储和传输，并支持多因素认证。
- API 密钥：用于程序间调用，密钥应具备安全生成、分发、存储和轮换机制。
- 数字证书/PKI 体系：高安全性要求场景下，可采用基于 X.509 证书的身份认证。

#### 7.1.2 权限控制

实现基于角色的访问控制或基于属性的访问控制，对不同用户、角色或系统授予不同操作权限。

- 最小权限原则：仅授予完成任务所需的最小权限；
- 权限粒度 权限控制应细化到 API 操作级别；
- 权限管理：提供完善的权限配置、查询和审计功能。

### 7.2 数据安全

7.2.1 视频信令和视频流数据的会话协议和媒体传输协议对接应符合 GB/T 28181 的规定。

7.2.2 数据交换应符合 GB/T 22239 和 GB/T 39786 的规定。

7.2.3 采用哈希校验、数字签名等机制确保数据在传输过程中的完整性，防止数据在传输过程中被篡改。

### 7.3 安全审计与日志

#### 7.3.1 访问日志

记录接口的访问时间、来源IP、操作类型、操作对象、操作结果、用户身份等信息，用于安全审计和追踪。

- 日志内容：应包含时间戳、源 IP、用户 ID、操作类型、URI、HTTP 方法、状态码、响应时间。
- 日志存储：日志应进行安全存储，防止篡改和丢失，并设置合理的保留期限。

#### 7.3.2 异常日志

异常日志应记录系统错误、安全事件、资源耗尽等异常信息，便于及时发现和处理安全问题。