

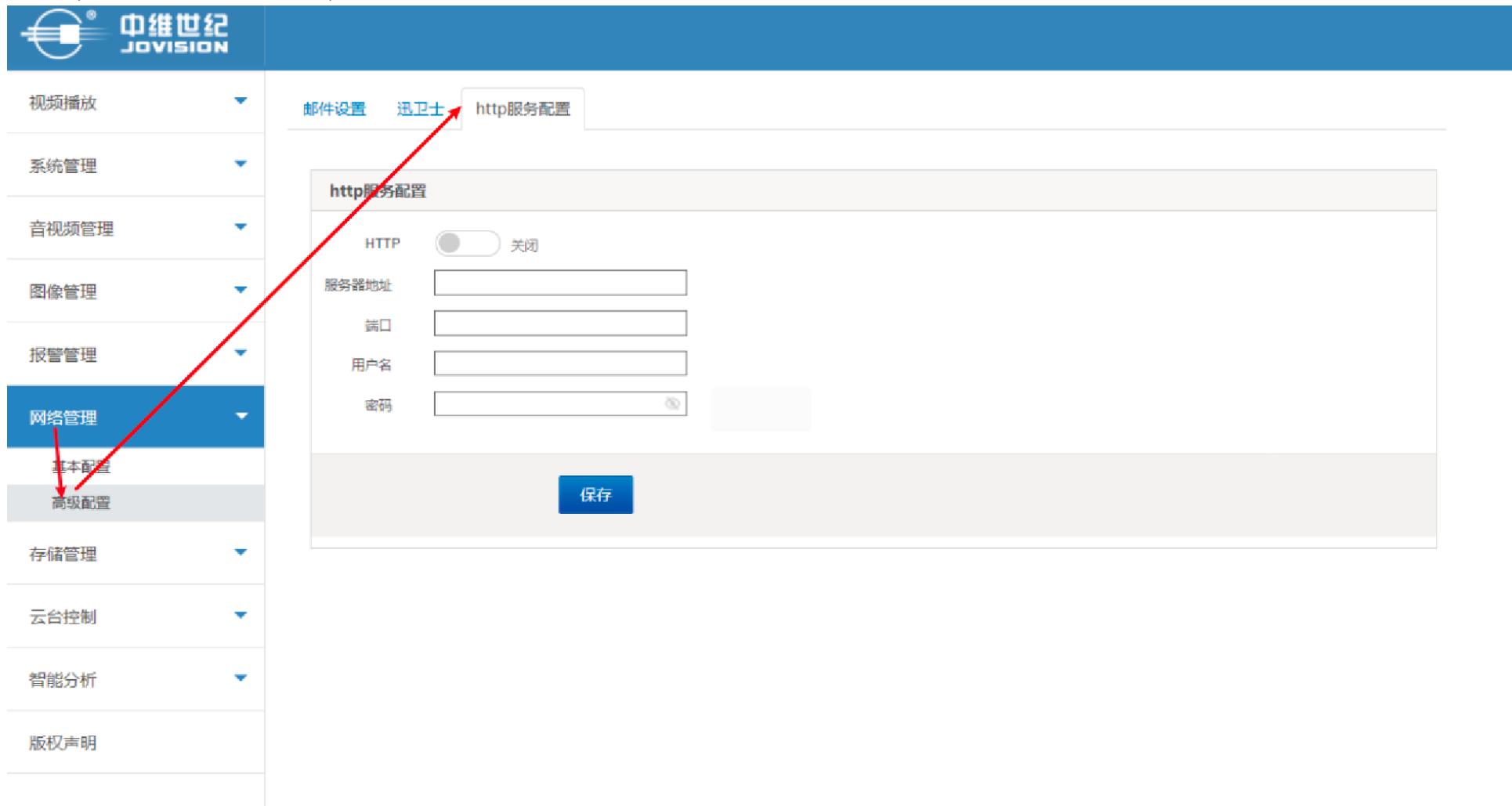
HTTP 推送接口说明

1 概述

摄像机有很多功能，需要通过 http 推送到服务器，本文统一定义具体方案和协议。

2 UI 界面

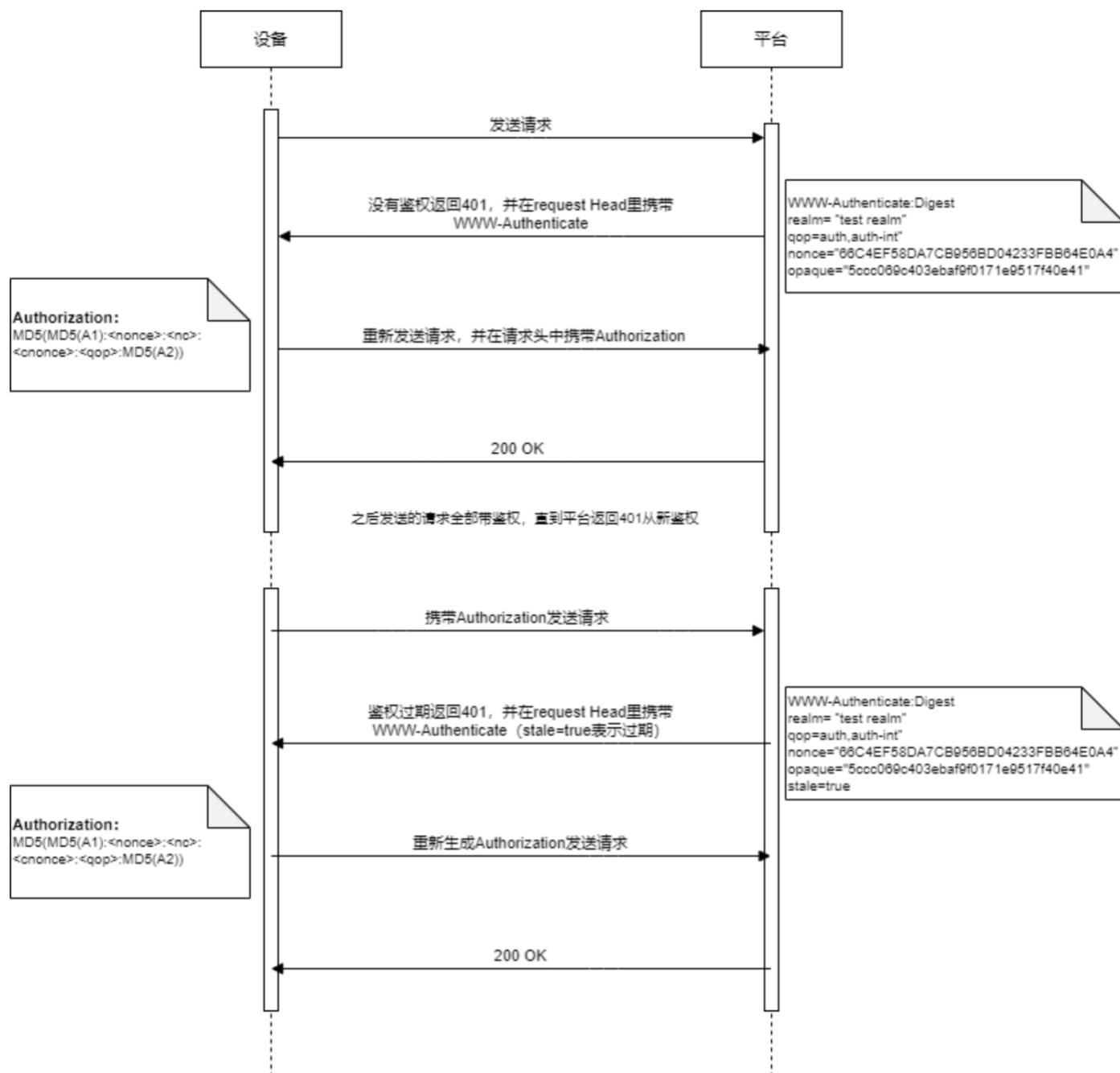
在【网络配置】->【高级配置】->【http 服务配置】，配置中只有一个地址配置，所有 http 推送统一使用该地址，每个业务定义自己的接口名称（即 url 的 filename 部分），配置界面如下：



3 鉴权

设备支持 http Digest 认证机制

- 设备在发送请求时，如果服务端返回 401 错误并携带 WWW-Authenticate，设备则会生成保存 Authorization，并添加在请求头上重新发送此次请求。
- 多次请求时，nonce 的三种使用策略：(本协议使用第三个方案，前两个只做介绍)
 - 单次使用：服务器维护一个 nonce 的列表，每次返回 401 时生成的 nonce 插入到列表里，客户端携带认证信息请求时候，如列表中存在 nonce 则执行认证，认证成功后从列表中删除 nonce，如列表中不存在则认证失败（防重放攻击），缺点是每个请求都要质询-响应，浪费时间和性能。
 - 每次请求回复中都携带 Authenticate-Info: nextnonce=xxxx，下次请求使用 nextnonce，这样每次都是用新的，就不需要质询流程，缺点是只能串行发送请求，必须等请求回复后才可以下次请求。
 - nonce 在一段时间内允许重复，服务端设置过期时间比如 1 小时，过期时间内允许使用同一个 nonce，过期后返回 401 携带 WWW-Authenticate 中指定新的 nonce 值，并指定 stale=true，此时客户端使用新的 nonce 重新请求，是一个折中方案。
- 设备开机时会向服务端发送保活包，默认保活间隔 1 分钟一次，保活间隔时间也可以由返回值控制，具体字段查看下方接口章节



4 接口

4.1 请求头

请求头	注释
Content-Type: application/json;charset=utf-8 Content-Length: 1024 Authorization:Digest username="admin",realm="no auth",nonce="C40Hr6NCxnvamwrO71o1zw==",uri="/v1/device/keepalive",response="42ad032d98035f15fde4b2013930ec06",algorithm="MD5",qop=auth,nc=42,cnonce="bc9d806106d901b02b909a0b93538ef0" User-Identify:12221S55G3X8	Authorization: 如果使用 Digest 认证就会携带 User-Identify : 设备唯一 ID (云视通号)

4.2 保活

4.2.1 请求体

功能	请求方法	URI	请求体	注释
设备心跳保活	GET	/v1/device/keepalive	{ "method": "", "param": { "deviceId": "12421HHTCS", "channelNum": 0, "deviceName": "HD IPC", } }	method : 接口名 deviceId : 设备唯一 ID, 云视通号 channelNum : 通道号 deviceName : 设备名 time : 时间 格式:

			"time": "2021-07-16 00:00:00" } }
			YYYY-MM-DD hh:mm:ss

4.2.2 响应体

响应体	注释
{ "method": "keepalive", "keepinterval" : 60, "error": { "errorcode": 0, "errormsg": "" }}	method : 接口名 keepinterval : 心跳间隔 (非必要) errorcode : 错误码 0 正常 <0 异常 errormsg : 错误信息

4.3 数据上报

应用产生数据后，调用不同的 URL 上报对应的消息

4.3.1 请求体

序号	功能	请求方法	URI	请求体	注释
1	报警上报 (老架构)	POST	/v1/uns/reviceMonitorPointRecord/alarm_report	{ "method": "alarm_report", "param": { "device_id": "12421HHTCS", "device_name": "HD IPC", "channel_id": 1, "alarm_type": "ebike_detection", "time": "2021-07-16 00:00:00", "status": "start,stop,pulse", "jpeg_base64": "jpeg_base64", "alarm_uuid": "XXXXXXXX", "alarm_msg": "msg", "extends": "XXXXXXX" } }	device_id : 设备 ID, 云视通号 device_name : 设备名称 channel_id : 通道 id alarm_type : 报警检测算法类型 电动车检测"ebike_detection",高空抛物检测"falling_detection" time : 报警时间 status : 报警状态:开始, 停止, 脉冲(无停止) jpeg_base64 : base64 编码的 jpeg 图片 alarm_uuid : 报警唯一 ID ([device_id]/[channel_id]/[alarm_type]/[发生报警时间]/[alarm_cnt]) alarm_msg:显示信息 extends:扩展字段, 用于扩展属性信息, 如人脸识别报警携带人脸识别信息等, 格式: base64(json), json 格式需要具体定义
2	事件上报 (JES)	POST	/v1/device/event_report	{ "method": "event_report", "param": { "device_id": "", "device_name": "", "channel_type": 0, "channel_id": 0, "event": 0, "type": "ebike_det", "typeName": "", "time": "2024-01-01 08:08:00", "status": 0, "extern": "", "jpeg_base64": "" } }	device_id: 设备 id device_name: 设备名称 channel_type: 0 视频输入, 1 外部输入, 2 设备本身 channel_id: 通道号, IPC 固定是 0 event: 事件大类: 1 报警事件, 2 无用, 3 定时抓拍事件 type: 事件类型: ebike_det 电动车检测报警 typeName: 事件名称, 翻译后的可读消息, 可直接用于消息显示 time: //发生时间 status: 获取报警状态: 0stop, 1start, 2pulse

				}	extern: 扩展数据, base64 编码, 不同事件扩展数据单独定义, 见 报警扩展数据 jpeg_base64: 抓拍的全景大图, base64 编码
3	元数据上报 (JES)	POST	/v1/device/tlv_report	tlv 格式	按照 tlv 格式进行解析
4	modbus 上报 (老架构)	POST	/rs/env/reportData.do	{ "deviceNo": "12221S37GCEU", "time": "2021-12-01 10:30:49", "data": { "@0": 0, "@1": 0, "@2": 0, "@3": 0, "@4": 0, "@5": 0, "@6": 0, "@7": 0, "@8": 0, "@9": 0 }, "alarm": { "@0": 0, "@1": 1 } }	deviceNo : 设备 ID, 云视通号 time : 报警时间 data : modbus 寄存器数据 @0 : 寄存器 0 数据 @1 : 寄存器 1 数据 @2 : 寄存器 2 数据 @3 : 寄存器 3 数据 @4 : 寄存器 4 数据 @5 : 寄存器 5 数据 @6 : 寄存器 6 数据 @7 : 寄存器 7 数据 @8 : 寄存器 8 数据 @9 : 寄存器 9 数据 data : modbus 报警信息 @0 : 寄存器 0 报警: 0 低于设定最低值报警 @1 : 寄存器 1 数据: 1 高于设定最高值报警
5	测温上报 (老架构)	POST	/v1/uns/reviceMonitorPointRecord/report_temper_data	{ "method": "report_temper_data", "param": { "device_id": "12421HHTCS", "device_name": "HD IPC", "channel_id": 1, "temper_rect": [{ "id": 0, "name": "rect 1", "temper": 30.1, "over_threshold": 100 }] } }	device_id : 设备 id, 云视通号 device_name : 设备名称 channel_id : 通道 id temper_rect : 检测区域 id : 区域 ID0,1,2... name : 区域名称 temper : 当前温度 over_threshold : 超温报警阈值
6	抓拍上报 (老架构)	POST	/v1/device/snap_report	{ "mid": 2, "method": "snap_report", "param": { "deviceID": "12221S6MR7KT", "deviceName": "dv2", "name": "T01140300.jpg", "jpegBase64": "" } }	mid: 推送 id, 每次推送会改变 deviceID: 设备 id, 云视通号 deviceName: 设备名称 name: 图片名称 jpegBase64: base64 编码的 jpeg 图片

报警扩展数据

项目	扩展数据内容
人脸抓拍	{ "thumbnail_file_path": "/dev/sda/20221013/A011173757.jpg" // 缩略图, 设备内存储路径 }

双车位	<pre> type: 新能源报警 dual_parking_new_energy 非新能源报警 dual_parking_non_new_energy 离场检测 dual_parking_exit_detected 双车位白名单报警 dual_parking_whitelist 双车位黑名单报警 dual_parking_blacklist 双车位陌生车辆报警 dual_parking_unknown 双车位占位报警 dual_parking_occupied 双车位空闲报警 dual_parking_idle extern: { "parking_id": 0, //0 左车位, 1 右车位 "parking_name": "", //车位名称 "plate_no": "京 J66666", //车牌号 "plate_type": 2, //车牌类型 -1 代表全部类型; 0 无车牌, 1 大型汽车号牌 ,2 小型汽车号牌 ,3 使馆汽车号牌 4 领馆汽 车号牌 ,5 挂车号牌 ,6 教练车号牌 ,7 警车号牌 ,8 香港出入境号牌, 9 澳门出入境号牌, 10 武警号牌, 11 军队号牌, 12 新能源号牌, 13 其他号牌 "plate_color": 1, //车牌颜色 -1 代表全部颜色; 0 未知; 1 蓝牌; 2 黄牌; 3 绿牌; 4 黑牌; 5 白牌 "vehicle_type": 1, //车辆类型 -1 代表全部类型 0 未知; 1 小轿车; 2 SUV; 3 面包车; 4 卡车; 5 自行车; 6 电动 车/摩托车; 7 三轮车; 8 客车; 9 皮卡; 10 货车 "vehicle_color": 4, //车辆颜色 -1 代表全部颜色 0 未知; 1 黑色; 2 蓝色 (蓝、青蓝) ;3 棕色; 4 灰色 (灰、银、深灰) 5 黄色 (黄、橘、金); 6 绿色; 7 紫色; 8 红色; 9 白色 "vehicle_orient": 1 //车辆朝向 -1 代表全部; 0 未知; 1 正面; 2 背面; 3 侧面 "entry_type": "",// whitelist 白名单, blacklist 黑名单, unknown 陌生车辆, new_energy 新能源, non_new_energy 非新能 源 "lock_id": "", // 地锁 ID "entry_plate_no": "" //进场车牌号 } </pre>
人脸属性	<pre>{ "gender":int,//性别;1 男; 2 女 "age":int,//年龄(代表年龄段, 不是具体年龄) 1:(0-10) 2:(10-20) 3:(20-40) 4:(40-60) 5:(60-100) "mask":int, //口罩; 戴口罩; 0 未戴口罩 "expression":int,//表情; 1 微笑; 2 愤怒; 3 悲伤; 4 正常; 5 惊讶 "hat":int,//帽子; 1 有; 0 无 "bread":int,//胡子; 1 有; 0 无 "hair":int,//头发; 1 长发; 2 短发 "glass":int,//眼镜; 0 无; 1 普通眼镜; 2 太阳眼镜 "thumbnail_file_path": "/dev/sda/20221013/A011173757.jpg" }</pre>

4.3.2 响应体

响应体	注释
<pre>{ "method": "people_count_report", "error": { "errorcode": 0, "errormsg": "" } }</pre>	<p>method : 接口名 errorcode : 错误码 0 正常 <0 异常 errormsg : 错误信息</p>

4.4 元数据上报

4.4.1 TLV 整体说明

4.4.1.1 TLV 的基本格式

采用网络字节序

T 类型, 占 4 字节

L 长度, 占 4 字节, 其内容表示 V 的长度

V 数据, 要传输的数据内容

4.4.1.2 TLV 结构

分为两层

第一层, 每个 TLV 表示一条数据

类型	长度	取值
T	4	META: 智能元数据 ALAM: 报警数据

		AREA: 检测框数据
L	4	有效数据长度, 即 V 的长度
V	N	第二层 TLV 数据的堆叠

第二层, 多个 TLV 结构并列存在, 不同的 T 代表不同的内容

4.4.1.3 目标类型

```
typedef enum {
    META_TARGET_FACE = 1,      ///<人脸
    META_TARGET_BODY = 2,      ///<人体
    META_TARGET_MOTOR_VEHICLE = 3,  ///<机动车
    META_TARGET_NONMOTOR_VEHICLE = 4,  ///<非机动车
    META_TARGET_CUSTOMER_COUNTER = 5,  ///<客流统计
    META_TARGET_HOT_MAP = 6,      ///<热度图
    META_TARGET_HOT_AREA = 7,      ///<热区
    META_TARGET_CUSTOMER_COUNTER_BASIC = 8,  ///<客流统计基础数据
    META_TARGET_HOT_AREA_BASIC = 9,  ///<热区基础数据
    META_TARGET_EXTERNAL_SENSOR = 10,  ///<外接传感器数据
    META_TARGET_COMMON = 11,  ///<通用内容, 比如目标检测
    META_TARGET_HUMAN_QUEUE_LENGTH = 13,  ///<排队长度
    META_TARGET_LICENSE_PLATE_RECOGNITION = 14,  ///<车牌识别
    META_TARGET_OCR = 15,  ///<字符识别
    META_TARGET_CUSTOMER_GROUP_COUNTER = 16,  ///<客群统计
} MetaTarget_e;
```

4.4.2 人脸抓拍/人脸属性

人脸抓拍上报 meta 数据中 FACE_INFO 为 0, 不包含人脸属性, 人脸属性应用上报 meta 数据中 FACE_INFO 为 1, 包含属性信息。

模块	字段名称	取值	长度(字节)	说明
基础信息	TARGET	0x1	4	目标类型; 4 字节; 取值 @see MetaTarget_e::META_TARGET_FACE
	AI_NAME	0x2	N	算法名字, 用于区分不同厂家的算法
	DEVICEID	0x3	N	通道 ID, 字符串形式
	CHANNELID	0x4	4	通道号; 4 字节
	PTS	0x5	8	时间戳; 8 字节; 1970 年以来的毫秒数 (注意: 不是标准的 mktime 的秒数, 而是忽略了时区的墙钟时间的秒数)
	PIC_SIZE	0x6	4	图像宽高; 4 字节; 宽、高各占 2 字节
	OBJ_RECT	0x7	8	抓图位置和大小; 8 字节; xywh 各占 2 字节
	OBJ_ID	0x8	8	目标 ID uint64_t
	OBJ_PIC	0x9	N	目标抠图, 包括人脸, 人体, 机非人; 二进制图像
	SNAPSHOT	0xA	N	全景抓图; 二进制图像
人脸信息	OBJ_NAME	0x10	N	检测目标名称
	FACE_INFO	0x10001	1	是否有人脸属性; 1 字节; 1 有; 0 无
	FACE_GENDER	0x10002	1	性别; 1 字节; 1 男; 2 女
	FACE_AGEGROUP	0x10003	1	年龄段; 1 儿童; 2 少年; 3 青年; 4 中年; 5 老年;
	FACE_MASK	0x10004	1	口罩; 1 字节; 1 戴口罩; 0 未戴口罩
	FACE_EXPRESSION	0x10005	1	表情; 1 字节; 1 微笑; 2 愤怒; 3 悲伤; 4 正常; 5 惊讶
	FACE_HAT	0x10006	1	帽子; 1 字节; 1 有; 0 无
	FACE_BEARD	0x10007	1	胡子; 1 字节; 1 有; 0 无
	FACE_HAIR	0x10008	1	头发; 1 字节; 1 长发; 2 短发
	FACE_GLASS	0x10009	1	眼镜; 1 字节; 0 无; 1 普通眼镜; 2 太阳眼镜
	FACE_FEATURE	0x1000A	N	人脸识别特征值; N 字节; 二进制数据
	FACE_SIMILARITY	0x1000B	2	相似度; 2 字节 取值 0-10000, 除以 100 为百分比
	FACE_LIVENESS	0x1000C	1	是否为活体; 1 字节; 1 是; 0 否
	FACE_ID	0x1000D	N	人脸 ID; N 字节; 二进制数据。人脸库中一个人脸的唯一标识
扩展数据	FACE_TEMPERATURE	0x1000F	2	人脸温度; 2 字节; 人脸摄氏温度乘以 100
	-	-	-	[{"thumbnail_file_path": "/dev/sda/20221013/A011173757.jpg"} // 缩略图, 设备内存储路径]

4.4.3 客流量数据

模块	字段名称	取值	长度(字节)	说明
基础信息	TARGET	0x1	4	目标类型; 4 字节; 取值 @see MetaTargetType_e
	AI_NAME	0x2	N	算法名字, 用于区分不同厂家的算法
	DEVICEID	0x3	N	通道 ID, 字符串形式
	CHANNELID	0x4	4	通道号; 4 字节
	PTS	0x5	8	时间戳; 8 字节; 1970 年以来的毫秒数 (注意: 不是标准的 mktime 的秒数, 而是忽略了时区的墙钟时间的秒数)
客流统计数据	CUSTOMER_COUNT_IN	0x40001	4	进店人数; 4 字节;
	CUSTOMER_COUNT_OUT	0x40002	4	出店人数; 4 字节;

	CUSTOMER_PASS	0x40003	4	过店人数; 4 字节;
	CUSTOMER_COUNT_KEEP	0x40004	4	留存人数; 4 字节;

4.4.4 字符提取

模块	字段名称	取值	长度(字节)	说明
基础信息	TARGET	0x1	4	目标类型; 4 字节; 取值 @see MetaTargetType_e
	AI_NAME	0x2	N	算法名字, 用于区分不同厂家的算法
	DEVICEID	0x3	N	设备 ID, 字符串形式
	PTS	0x5	8	时间戳; 8 字节; 1970 年以来的毫秒数 (注意: 不是标准的 mktime 的秒数, 而是忽略了时区的墙钟时间的秒数)
	PIC_SIZE	0x6	4	图像宽高; 4 字节; 宽、高各占 2 字节
	SNAPSHOT	0xA	N	全景抓图; 二进制图像
字符识别	META_TYPE_OCR_REGION_NUM	0xa0001	1	区域个数; 1 字节, 每个区域包含 0xa0002-0xa0006
	META_TYPE_OCR_REGION_POINTS	0xa0002	16	字符区域的 4 个顶点, 16 字节, x,y 各占 2 字节
	META_TYPE_OCR_REGION_NAME	0xa0003	N	区域名称; 字符串
	META_TYPE_OCR_TEXT	0xa0004	N	识别结果; 字符串
	META_TYPE_OCR_CONFIDENCE	0xa0005	2	置信度; 2 字节, 取值 0-10000, 除以 100 为百分比
	META_TYPE_OCR_EACH_CONFIDENCE	0xa0006	N	每个字符置信度; N 项, 每项 2 字节, 取值 0-10000, 除以 100 为百分比