

机器人工控机软件接口

日期	版本	维护日志	维护人
2019-06-28	V2.0	V2.0 首版发布	赖晗
2019-07-03	V2.0.1	1. 机器人状态控制交由 robot_control 实现，控制模式由 control_status 发送至 status_control，再通过 robot_status 广播给其他模块 2. 3D 障碍物告警直接由 move_base 来产生和发布	赖晗
2019-07-17	V2.0.2	1. 修改 VoiceCmd 部分字段和定义 2. 修改障碍物检测接口 3. 删除 move_status 接口 4. move_base 订阅 control_cmd	赖晗
2019-08-27	V2.0.3	1. 修改云台接口，增加同时设置和查询水平俯仰角度的功能	赖晗
2019-09-17	V2.0.4	1. 修改障碍物接口，直接发送障碍物状态到运动控制模块	赖晗
2019-09-17	V2.0.5	1. 修改 control_cmd，增加打开和关闭软急停控制命令	赖晗
2019-12-13	V2.1.0	1. 增加机械臂控制接口	赖晗
2019-12-26	V2.1.1	1. 增加切换防碰撞/防跌落/障碍物告警命令	赖晗
2020-01-09	V3.0.0	1. 扩展 control_status 字段，增加更多告警信息 2. 同步修改 robot_status 字段，确保二者一致 3. 增加 location_cmd 消息定义和相关说明	赖晗
2020-01-15	V3.0.1	1. 更新 control_status 和 robot_status 部分字段 bit 定义 2. control_status 和 robot_status 增加单独温度字段 3. preset_scale_cmd 增加 extra_data 保存额外信息	赖晗

1. 基本架构

1.1. 适用范围

该协议主要用于自动电力巡检机器人导航模块和后台控制软件上位机进行通信，以及控制模块和导航模块之间的通信。

1.2. 网络架构

系统采用 ros（Robot Operation System）系统的 topic、service 和 actionlib 接口，软件层面采用 c++ 和 python 编程实现，所有服务器和机器人作为 ros 系统的节点存在，进行 topic 话题的发布/订阅 topic 以及提供 service 供其他节点调用。

ros master 部署在机器人工控机上运行，同时启动一个 ros_brigde_server 节点，使用 websocket 方式来实现后台 server 和其他 ros 节点进行通信。

1.3. 系统框图

整个系统所有的模块框图和交互信息如下所示：

2.1.1. server_status

server_status 消息用于服务器向机器下发心跳消息供其内容定义如下：

字段	ROS 类型	含义	取值
header	Header	命令消息头	ROS 消息头
sender	string	发送者	消息发送者
receiver	string	接收者	消息接收者
trans_id	uint32	会话号	命令和应答保持一致
data	string	心跳检测信息	暂未使用

2.1.2. robot_status

robot_status 用于机器人向 server 反馈当前状态，其内容定义如下：

字段	ROS 类型	含义	取值
header	Header	状态消息头	ROS 消息头
sender	string	发送者	消息发送者
receiver	string	接收者	消息接收者
trans_id	uint32	会话号	命令和应答保持一致
offline_warn	uint8	在线/离线状态	0: 正常在线 1: 离线告警
mode	uint8	控制模式	0: 任务模式 1: 后台遥控 2: 手柄控制
velocity_x	float64	机器人当前的线速度	单位: m/s
velocity_yaw	float64	机器人当前的角速度	单位: rad/s
position	float64[]	当前位置坐标点	所在位置 x, y 坐标点
orientation	float64	机器人方向	地图 x 方向为 0, [-pi,pi]
nav_status	uint8	导航告警标志位	bit0: 导航偏轨告警 bit1: 定位失效告警 bit2: 3D 障碍物检测告警
sensor_status	uint32	传感器状态告警	bit0: 可见光告警 bit1: 红外热像仪告警
wheel_status	uint8	车轮转向状态	bit0: 直行模式 bit1: 转向模式 bit2: 方向切换中 bit3: 步进电机故障清除 bit4: 伺服电机故障清除 bit5: 关闭障碍物检测 bit6: 关闭防碰撞检测 bit7: 关闭防跌落检测
light_status	uint8	指示灯状态	bit0: 充电继电器开关 bit1: 指示灯红 bit2: 指示灯绿 bit3: 指示灯蓝 bit4: 软急停使能状态 bit5: 硬急停开关状态

字段	ROS 类型	含义	取值
			bit6: 防撞条状态
stop_status	uint8	停止告警标志位	bit0: 前防碰撞告警 bit1: 预留 bit2: 预留 bit3: 预留 bit4: 预留 bit5: 障碍物告警 bit6: 左侧防跌落告警 bit7: 右侧防跌落告警
circuit_status	uint8	线路异常告警	bit0: 步进底板通信告警 bit1: 手柄通讯中断 bit2: 手柄线路异常 bit3: 传感器板线路异常 bit4: 防碰撞线路异常 bit5: 左侧 TOF 线路异常 bit6: 右侧 TOF 线路异常 bit7: 电池电量极低告警
other_status	uint32[]	其他状态	预留
battery_voltage	int32	电池电压	单位: 0.1V
battery_current	int32	电池电流	单位: 0.1A
battery_quantity	int32	电池电量	单位: 1%
battery_status	uint8	电池告警	bit0: 欠压 bit1: 过压 bit2: 低温 bit3: 高温 bit4: 充电中 bit5: 放电中 bit6: 预留 bit7: 电池线路异常
motor_status	uint8[]	电机告警标志位 数组长度 8, 分别表示 左前伺服、右前伺服、 右后伺服、左后伺服、 左前步进、右前步进、 右后步进、左后步进	伺服电机: bit0: 断线 bit4: 故障 步进电机: bit0: 错误
motor_info	uint8[]	电机告警码 数组长度 8, 分别表示 左前伺服、右前伺服、 右后伺服、左后伺服、 左前步进、右前步进、 右后步进、左后步进	bit0: 当前状态断线状态 bit 1: 当前状态移动中丢步 bit 2: 当前状态到位丢步 bit 3: 当前状态矫正 bit 4: 前一状态断线状态 bit 5: 前一状态移动中丢步 bit 6: 前一状态到位丢步 bit 7: 前一状态矫正 bit 8: 当前状态误差大需归零 bit 12: 前一状态误差大需归零 bit 16~bit31: 步进电机驱动器故障说明
temperature	int32[]	采集到的温度	单位: 0.1℃, -100℃表示未采集 数组长度为 13,含义如下

字段	ROS 类型	含义	取值
			0: 电池温度 1: 左前伺服电机温度 2: 右前伺服电机温度 3: 右后伺服电机温度 4: 左后伺服电机温度 5: 左前步进电机温度 6: 右前步进电机温度 7: 右后步进电机温度 8: 左后步进电机温度 9: 工控机温度 10: 运动控制卡温度 11: 吊仓内部温度 12: 吊仓外部温度

2.2. 运动控制接口

相关的消息定义如下：

消息名	类型	定义	收发方向
mode_cmd	topic	GeneralCmd	joy_control => robot_control server => robot_control
move_cmd	topic	MoveCmd	joy_control => robot_control move_base => robot_control server => robot_control
move_status	topic	MoveStatus	robot_control => move_base
control_cmd	topic	GeneralCmd	joy_control => robot_control server => robot_control
control_status	topic	ControlStatus	robot_control => status_control
robot_odom	topic	nav_msgs::Odometry	robot_control => move_base

2.2.1. 控制模式

控制模式分为任务模式、手柄控制、后台遥控三种。分别对应导航控制、手柄遥控、后台遥控。

开机默认情况下机器人处于任务模式，mode_control 用于后台、手柄对机器人进行遥控时申请控制权限，控制完成后释放控制权限，机器人再次回到任务模式。

优先级：手柄控制 > 后台遥控 > 任务模式

2.2.2. mode_cmd

mode_cmd 用于确定机器人控制模式，其对应字段定义如下：

字段	ROS 类型	含义	取值
header	Header	状态消息头	ROS 消息头
sender	string	发送者	消息发送者

字段	ROS 类型	含义	取值
receiver	string	接收者	消息接收者
trans_id	uint32	会话号	命令和应答保持一致
cmd	string	控制命令	acquire: 申请控制权限 release: 释放控制权限
data	string	控制参数	见后续说明

cmd 和对应的 data 含义如下:

cmd	cmd data
acquire	joy: 手柄控制 server: 后台遥控
release	--

2.2.3. move_cmd

move_cmd 用于向 robot_control 下发运动命令, 其内容定义如下:

字段	ROS 类型	含义	取值
header	Header	状态消息头	ROS 消息头
sender	string	发送者	消息发送者
receiver	string	接收者	消息接收者
trans_id	uint32	会话号	命令和应答保持一致
velocity_x	float64	机器人线速度	单位: 0.001m/s
velocity_yaw	float64	机器人角速度	单位: 0.001rad/s
wheel_control	uint8	机器人车轮转动控制	0: 直行模式; 1: 转向模式

2.2.4. control_cmd

control_cmd 消息用于向机器下发控制命令, 其内容定义如下:

字段	ROS 类型	含义	取值
header	Header	状态消息头	ROS 消息头
sender	string	发送者	消息发送者
receiver	string	接收者	消息接收者
trans_id	uint32	会话号	命令和应答保持一致
cmd	string	控制命令	disable_collision: 关闭防碰撞告警 enable_collision: 打开防碰撞告警 switch_collision: 切换防碰撞告警 disable_fallen: 关闭防跌落告警 enable_fallen: 打开防跌落告警 switch_fallen: 切换防跌落告警 disable_charge: 关闭充电开关 enable_charge: 打开充电开关 disable_obstacle: 关闭障碍物告警 enable_obstacle: 打开障碍物告警 switch_obstacle: 切换障碍物告警 enable_stop: 打开软急停开关

字段	ROS 类型	含义	取值
			disable_stop: 关闭软急停开关 query_status: 查询当前状态（播放语音提示）
data	string	数据	控制命令对应的数据值

2.2.5. control_status

control_status 消息用于机器人反馈当前控制状态，其内容定义如下：

字段	ROS 类型	含义	取值
header	Header	状态消息头	ROS 消息头
sender	string	发送者	消息发送者
receiver	string	接收者	消息接收者
trans_id	uint32	会话号	命令和应答保持一致
mode	uint8	控制模式	0: 任务模式 1: 后台遥控 2: 手柄控制
wheel_status	uint8	车轮转向状态	bit0: 直行模式 bit1: 转向模式 bit2: 方向切换中 bit3: 步进电机故障清除 bit4: 伺服电机故障清除 bit5: 关闭障碍物检测 bit6: 关闭防碰撞检测 bit7: 关闭防跌落检测
light_status	uint8	指示灯状态	bit0: 充电继电器开关 bit1: 指示灯红 bit2: 指示灯绿 bit3: 指示灯蓝 bit4: 软急停使能状态 bit5: 硬急停开关状态 bit6: 防撞条状态
stop_status	uint8	停止告警标志位	bit0: 前防碰撞告警 bit1: 预留 bit2: 预留 bit3: 预留 bit4: 预留 bit5: 障碍物告警 bit6: 左侧防跌落告警 bit7: 右侧防跌落告警
circuit_status	uint8	线路异常告警	bit0: 步进底板通信告警 bit1: 手柄通讯中断 bit2: 手柄线路异常 bit3: 传感器板线路异常 bit4: 防碰撞线路异常 bit5: 左侧 TOF 线路异常 bit6: 右侧 TOF 线路异常 bit7: 电池电量极低告警

字段	ROS 类型	含义	取值
other_status	uint32[]	其他状态	预留
battery_voltage	int32	电池电压	单位：0.1V
battery_current	int32	电池电流	单位：0.1A
battery_quantity	int32	电池电量	单位：1%
battery_status	uint8	电池告警	bit0: 欠压 bit1: 过压 bit2: 低温 bit3: 高温 bit4: 充电中 bit5: 放电中 bit6: 预留 bit7: 电池线路异常
motor_status	uint8[]	电机告警标志位 数组长度 8，分别表示 左前伺服、右前伺服、 右后伺服、左后伺服、 左前步进、右前步进、 右后步进、左后步进	伺服电机： bit0: 断线 bit4: 故障 步进电机： bit0: 错误
motor_info	uint8[]	电机告警码 数组长度 8，分别表示 左前伺服、右前伺服、 右后伺服、左后伺服、 左前步进、右前步进、 右后步进、左后步进	bit0: 当前状态断线状态 bit 1: 当前状态移动中丢步 bit 2: 当前状态到位丢步 bit 3: 当前状态矫正 bit 4: 前一状态断线状态 bit 5: 前一状态移动中丢步 bit 6: 前一状态到位丢步 bit 7: 前一状态矫正 bit 8: 当前状态误差大需归零 bit 12: 前一状态误差大需归零 bit 16~bit31: 步进电机驱动器故障说明
temperature	int32[]	采集到的温度	单位：0.1℃，-100℃表示未采集 数组长度为 13,含义如下 0: 电池温度 1: 左前伺服电机温度 2: 右前伺服电机温度 3: 右后伺服电机温度 4: 左后伺服电机温度 5: 左前步进电机温度 6: 右前步进电机温度 7: 右后步进电机温度 8: 左后步进电机温度 9: 工控机温度 10: 运动控制卡温度 11: 吊仓内部温度 12: 吊仓外部温度

2.3. 导航控制接口

导航相关的消息定义如下：

消息名	类型	定义	收发方向
move_action	actionlib	GeneralAction	task_manager => move_base charge_control => move_base
nav_status	topic	NavStatus	move_base => status_control move_base => preset_scale move_base => find_scale
location_cmd	topic	GeneralCmd	robot_control => move_base server => move_base

2.3.1. move_action

mode_control 采用通用 action 消息定义，其对应字段定义如下：

action 类型	字段	ROS 类型	含义	取值
goal	header	Header	状态消息头	ROS 消息头
	sender	string	发送者	消息发送者
	receiver	string	接收者	消息接收者
	trans_id	uint32	会话号	命令和应答保持一致
	type	string	请求类型	见后续说明
	data	string	json 数据	见后续说明
feedback	event_code	string	事件信息	见后续说明
	data	string	json 数据	见后续说明
result	ret_code	string	响应码	见后续说明
	data	string	json 数据	见后续说明

各字段取值含义如下：

action 类型	字段	取值	
goal	type	move: 移动	
	data	type="move"	<pre>{ "with_direction": 0, // 带方向? 1-是 0-否 "precision": 0, // 精确移动? 1-是 0-否 "move_style": 0, // 移动方式: 0-前进 1-后退 "velocity_x": 1.00, // 移动线速度 单位:m/s "velocity_yaw": 2.00, // 移动角速度 单位:rad/s "goal_pose": // 移动目标点位姿 { "x": 1.0, "y": 2.0, "yaw": 3.0 } }</pre>
feedback	event_code	0: 路径规划中 1: 路径规划完成 2: 移动中 3: 告警	

action 类型	字段	取值	
	data	event_code="0"	
		event_code="1"	
		event_code="2"	
		event_code="3"	
result	ret_code	0: 成功	
		1: 路径规划失败	
		2: 定位失效	
		3: 障碍物阻挡	
	data	event_code="0"	
		event_code="1"	
		event_code="2"	
		event_code="3"	

2.3.2. nav_status

nav_status 命令消息用于 move_base 反馈导航状态，其内容定义如下：

字段	ROS 类型	含义	取值
header	Header	命令消息头	ROS 消息头
sender	string	发送者	消息发送者
receiver	string	接收者	消息接收者
trans_id	uint32	会话号	命令和应答保持一致
nav_status	uint8	导航告警使能	bit0: 导航偏轨告警使能 bit1: 定位失效告警使能 bit2: 障碍物检测告警使能
nav_warn	uint8	导航告警标志位	bit0: 导航偏轨告警 bit1: 定位失效告警 bit2: 障碍物检测告警
position	float64[]	机器人当前坐标	机器人 x, y 坐标
orientation	float64	机器人当前朝向	机器人方向, [-pi,pi]

2.3.3. location_cmd

location_cmd 命令消息用于通知 move_base 重新设置机器人位置，在机器人重启和定位失效后手工指定机器人位置的情况下使用。其内容定义如下：

字段	ROS 类型	含义	取值
header	Header	状态消息头	ROS 消息头
sender	string	发送者	消息发送者
receiver	string	接收者	消息接收者
trans_id	uint32	会话号	命令和应答保持一致
cmd	string	控制命令	set_location: 设置机器人当前位置
data	string	控制参数	init_pose: 设置为初始位置
			last_pose: 设置为上一次位置

2.4. 云台控制接口

云台控制模块通过调用底层的 pelco-d 协议完成对云台的控制，云台模块采用 ros service 方式来实现控制命令下发和完成情况反馈。

云台分别提供 service 和 topic，名为 terrace_control 和 terrace_cmd/terrace_ack。实际使用中根据实际情况选用。

云台控制相关的消息定义如下：

消息名	类型	定义	收发方向
terrace_control	service	ControlService	preset_scale => cloud_terrace find_scale => cloud_terrace
terrace_cmd	topic	GeneralCmd	preset_scale => cloud_terrace find_scale => cloud_terrace joy_control => cloud_terrace
terrace_ack	topic	GeneralAck	cloud_terrace => preset_scale cloud_terrace => find_scale

2.4.1. pelco-d 协议

云台和工控机之间采用 RS485 串口直接连接和控制，采用 pelco-d 协议。

pelco-d 的协议消息内容如下：

字节 1	字节 2	字节 3	字节 4	字节 5	字节 6	字节 7
同步字节	地址码	指令码 1	指令码 2	数据码 1	数据码 2	校验码

所有指令的通用字段如下：

字段	取值
同步字节	0xff
地址码	0x01

针对不同命令，其对应的指令码、数据码如下：

命令	指令码 1	指令码 2	数据码 1	数据码 2
设置水平角度	0x00	0x4b	msb	lsb
设置俯仰角度	0x00	0x4d	msb	lsb
查询水平角度	0x00	0x51	0x00	0x00
查询俯仰角度	0x00	0x53	0x00	0x00
水平角度反馈	0x00	0x59	msb	lsb
俯仰角度反馈	0x00	0x5b	msb	lsb
打开雨刷	0x00	0x09	0x00	0x01
关闭雨刷	0x00	0x0b	0x00	0x01
打开补光灯	0x00	0x09	0x00	0x02
关闭补光灯	0x00	0x0b	0x00	0x02

设置和查询水平/俯仰角度时，遵循如下规则：

云台角度 $\times 100 = \text{msb} \times 256 + \text{lsb}$

也即是：读写的角度时，将实际度数乘以 100 并且转化为 16 进制数据，高 8 位作为在数据码 1，低 8 位作为数据码 2。

水平角度范围：0~359 度，正前方为 0 度，顺时针方向为正
俯仰角度范围：俯视为 0~90 度，水平为 0 度，向下为正；
仰视 270~359 度，正上方为 270 度，向下为正。

以上是飞跃云台的物理角度，实际使用中，将所有角度进行了标准化处理，处理后的
标准化角度如下：

水平角度范围：-179.99~180.00 度，正前方为 0 度，逆时针方向为正；
俯仰角度范围：-90.00~90.00 度，水平方向为 0 度，向下为正。

所有 terrace_control、terrace_cmd、terrace_ack 消息中的读数，都应当使用标准化处理
之后的读数。

2.4.2. terrace_control

云台控制的 service 定义如下：

service 类型	名称	ROS 类型	含义	取值
request	header	Header	命令消息头	ROS 消息头
	sender	string	发送者	消息发送者
	receiver	string	接收者	消息接收者
	trans_id	uint32	会话号	命令和应答保持一致
	cmd	string	控制命令	见后续说明
	data	string	命令数据	见后续说明
response	trans_id	uint32	会话号	命令和应答保持一致
	ack	string	控制响应	见后续说明
	data	string	响应数据	见后续说明

terrace_control 提供的各种服务，其请求、响应和对应数据的含义如下：

服务类型	cmd	cmd_data	ack	ack_data
设置水平角度	set_pan_angle	角度值*100	success/failed	--
查询水平角度	get_pan_angle	--	success/failed	角度值*100/failed
设置俯仰角度	set_tilt_angle	角度值*100	success/failed	--
查询俯仰角度	get_tilt_angle	--	success/failed	角度值*100/failed
设置水平俯仰角度	set_angle	水平俯仰角度值*100	success/failed	水平俯仰角度值*100
查询水平俯仰角度	get_angle	--	success/failed	
云台左转	pan_left	移动速度	success/failed	--
云台右转	pan_right	移动速度	success/failed	--
云台上转	tilt_up	移动速度	success/failed	--
云台下转	tilt_down	移动速度	success/failed	--
云台停止	terrace_stop	--	success/failed	--
打开雨刷	brush_on	--	success/failed	--
关闭雨刷	brush_off	--	success/failed	--
打开补光灯	light_on	--	success/failed	--
关闭补光灯	light_off	--	success/failed	--

说明：

- 1.如果云台正在接受手柄的控制，则所有 service 都立即返回 true，并且 response_data 值为 busy
- 2.云台移动速度有高、中、低三挡，分别在 request.data 中用 string 类型 high_speed、mid_speed、low_speed 表示
- 3. 设置和查询水平俯仰角度值，data 部分同时包含水平俯仰两个角度，先水平角度，后俯仰角度中间用英文逗号隔开，例如“12300,-300”表示水平 123 度，俯仰-3 度

2.4.3. terrace_cmd

该消息用于其他模块向 cloud_terrace 模块下发云台控制命令。其 msg 文件名称为：TerraceCmd.msg。其内容定义如下：

字段	ROS 类型	含义	取值
header	Header	状态消息头	ROS 消息头
sender	string	发送者	消息发送者
receiver	string	接收者	消息接收者
trans_id	uint32	会话号	命令和应答保持一致
cmd	string	控制命令	云台的控制命令
data	string	控制参数	控制命令附带的参数

云台控制命令和对应参数，和 terrace_control 中 reqes 以及 request data 保持一致。

2.4.4. terrace_ack

该消息用于 cloud_terrace 模块向其他模块发送命令响应。其 msg 文件名称为：TerraceAck.msg。其内容定义如下：

字段	ROS 类型	含义	取值
header	Header	状态消息头	ROS 消息头
sender	string	发送者	消息发送者
receiver	string	接收者	消息接收者
trans_id	uint32	会话号	命令和应答保持一致
ack	string	应答消息	云台控制应答命令
data	string	应答结果	云台控制应答的结果

云台响应命令和应答数据结果，和 terrace_control 中 response 以及 response data 保持一致。

2.5. 机械臂控制接口

机械臂控制模块 robot_arm 用于控制机械臂完成相关旋转和移动指令。
模块分别提供 service 和 topic，命名为 arm_control 和 arm_cmd/arm_ack。实际使用时，可以根据需要选择一种来调用。

机械臂控制相关的消息定义如下：

消息名	类型	定义	收发方向
-----	----	----	------

消息名	类型	定义	收发方向
arm_control	service	ControlService	preset_scale => arm_control find_scale => arm_control
arm_cmd	topic	GeneralCmd	preset_scale => arm_control find_scale => arm_control joy_control => arm_control
arm_ack	topic	GeneralAck	arm_control => preset_scale arm_control => find_scale

2.5.1. arm_control

机械臂控制的 service 定义如下：

service 类型	名称	ROS 类型	含义	取值
request	header	Header	命令消息头	ROS 消息头
	sender	string	发送者	消息发送者
	receiver	string	接收者	消息接收者
	trans_id	uint32	会话号	命令和应答保持一致
	cmd	string	控制命令	见后续说明
	data	string	命令数据	见后续说明
response	trans_id	uint32	会话号	命令和应答保持一致
	ack	string	控制响应	见后续说明
	data	string	响应数据	见后续说明

arm_control 提供的各种服务，其请求、响应和对应数据的含义如下：

服务类型	cmd	cmd_data	ack	ack_data
记录轨迹	record_orbit	轨迹文件全路径+文件名	success/failed	失败原因
停止记录	stop_orbit	轨迹文件全路径+文件名	success/failed	失败原因
播放轨迹	play_orbit	轨迹文件全路径+文件名，方向 +: 正向； -: 反向	success/failed	失败原因
旋转	rotate	末端: roll, pitch, yaw 0: 不旋转；+: 正向； -: 逆向	success/failed	失败原因
停止旋转	stop_rotate	末端: roll, pitch, yaw 0: 停止；+: 继续	success/failed	失败原因
查询姿态	get_orientation	--	success/failed	末端: roll, pitch, yaw 绝对角度值
设置姿态	set_orientation	末端: roll, pitch, yaw 绝对角度值	success/failed	失败原因
移动	move	末端: x, y, z 0: 不移动；+: 正向； -: 逆向	success/failed	失败原因
停止移动	stop_move	末端: x, y, z 0: 停止；+: 继续	success/failed	失败原因
查询位置	get_position	--	success/failed	末端: x, y, z 绝对坐标值
设置位置	set_position	末端: x, y, z 绝对坐标值	success/failed	失败原因
机械臂复位	reset	--	success/failed	失败原因

说明：

- (1) 所有旋转和移动时，x、y、z、roll、pitch、yaw 正负方向采用 ros 坐标系标准，即：
x 向前为正，向后为负；
y 向左为正，向右为负；
z 向上为正，向下为负；
roll 顺时针为正，逆时针为负；
pitch 向下为正，向上为负；
yaw 逆时针为正，顺时针为负。
- (2) 旋转时以机械臂末端为参照系进行旋转；查询/设置姿态时，以机械臂原点为参照系，查询/设置末端的欧拉角角度值。
- (3) 移动时以机械臂末端为参照系进行移动，查询/设置位置时，以机械臂原点为参照系，查询/设置末端的笛卡尔坐标值。
- (4) cmd 和 ack 中的 data 数据值如果有多项，中间用英文逗号分割，无空格。
- (5) 复位指令要求位置、姿态全部归零，机械臂回到零位。
- (6) 如果记录或者播放轨迹的命令中，轨迹文件名为空，则 robot_arm 节点使用默认的一个轨迹文件名来记录轨迹。该文件是一个固定位置和名称的临时文件，只保留一份，后面录制的轨迹会覆盖前面的轨迹。默认的轨迹文件名和路径在 robot_arm 节点中通过 ros 参数配置。
- (7) 记录轨迹或者播放轨迹过程中，如果发生异常，robot_arm 节点会主动发送一个 arm_ack 消息，告知发送者记录/播放轨迹失败以及失败原因。

2.5.2. arm_cmd

该消息用于其他模块向 robot_arm 模块下发摄像头控制命令。其 msg 文件名称为：CameraCmd.msg。其内容定义如下：

字段	ROS 类型	含义	取值
header	Header	状态消息头	ROS 消息头
sender	string	发送者	消息发送者
receiver	string	接收者	消息接收者
trans_id	uint32	会话号	命令和应答保持一致
cmd	string	控制命令	机械臂控制命令
data	string	控制参数	控制命令附带的参数

摄像头控制命令和对应参数，和 arm_control 中 reques 以及 request data 保持一致。

2.5.3. arm_ack

该消息用于 robot_arm 模块向其他模块发送命令响应消息。其 msg 文件名称为：CameraAck.msg。其内容定义如下：

字段	ROS 类型	含义	取值
header	Header	状态消息头	ROS 消息头
sender	string	发送者	消息发送者
receiver	string	接收者	消息接收者
trans_id	uint32	会话号	命令和应答保持一致
ack	string	应答消息	机械臂命令响应

字段	ROS 类型	含义	取值
data	string	应答结果	机械臂响应结果

摄像头响应命令和响应结果数据，和 arm_control 中 response 以及 response data 保持一致。

2.6. 摄像头控制接口

摄像头控制模块 camera_node 采用海康威视公司提供的 SDK 完成对摄像头的控制，实现镜头抓图、变焦、变倍等操作，采用 ros service 方式实现。

摄像头控制分别提供 service 和 topic，命名为 camera_control 和 camera_cmd/camera_ack。实际使用时，可以根据需要选择一种来调用。

摄像头控制相关的消息定义如下：

消息名	类型	定义	收发方向
camera_control	service	ControlService	preset_scale => camera_node find_scale => camera_node
camera_cmd	topic	GeneralCmd	preset_scale => camera_node find_scale => camera_node joy_control => camera_node
camera_ack	topic	GeneralAck	camera_node => preset_scale camera_node => find_scale

2.6.1. camera_control

摄像头控制的 service 定义如下：

service 类型	名称	ROS 类型	含义	取值
request	header	Header	命令消息头	ROS 消息头
	sender	string	发送者	消息发送者
	receiver	string	接收者	消息接收者
	trans_id	uint32	会话号	命令和应答保持一致
	cmd	string	控制命令	见后续说明
	data	string	命令数据	见后续说明
response	trans_id	uint32	会话号	命令和应答保持一致
	ack	string	控制响应	见后续说明
	data	string	响应数据	见后续说明

camera_control 提供的各种服务，其请求、响应和对应数据的含义如下：

服务类型	cmd	cmd_data	ack	ack_data
设置变倍值	set_zoom_level	放大倍数	success/failed	--
查询变倍值	get_zoom_level	--	success/failed	放大倍数
截图	capture_picture	截图文件路径	success/failed	--
设置聚焦模式	set_focus_mode	auto: 自动; manual: 手动; semiauto: 半自动	success/failed	--
查询聚焦模式	get_focus_mode	--	success/failed	聚焦模式

服务类型	cmd	cmd_data	ack	ack_data
查询焦点位置	get_focus_position	--	success/failed	焦点位置
设置焦点位置	set_focus_position	0x1000~0xc000	success/failed	--
变倍放大	zoom_in	执行时间 (ms) 0: 持续放大	success/failed	--
变倍缩小	zoom_out	执行时间 (ms) 0: 持续缩小	success/failed	--
聚焦拉远	focus_far	执行时间 (ms) 0: 持续拉远	success/failed	--
聚焦拉近	focus_near	执行时间 (ms) 0: 持续拉近	success/failed	--
停止变倍/聚焦	camera_stop	--	success/failed	--
osd 显示设置	set_osd_mode	0: 不显示; 1: 显示	success/failed	--
设置最小聚焦距离	set_min_focus_distance	10,30,100,150,300,600,1000,2000,65535 单位: cm	success/failed	--
查询最小聚焦距离	get_min_focus_distance	--	success/failed	最小聚焦距离
设置宽动态等级	set_wdr_level	0~100	success/failed	--
查询宽动态等级	get_wdr_level	--	success/failed	宽动态等级
设置宽动态值	set_wdr_mode	0: 不启用 1: 启用	success/failed	--
查询宽动态值	get_wdr_mode	--	success/failed	宽动态值
设置曝光模式	set_exposure_mode	manual: 手动曝光 auto: 自动曝光	success/failed	--
查询曝光模式	get_exposure_mode	--	success/failed	曝光模式
设置光圈大小 (手动曝光模式下)	set_iris	160,200,240,280, 340,400,480,560, 680,960,1100,1400, 1600,1900,2200	success/failed	--
查询光圈大小 (手动曝光模式下)	get_iris	--	success/failed	光圈大小

如果摄像头正在接受手柄的控制，则所有 service 都立即返回 true，并且 response_data 值为 busy。

2.6.2. camera_cmd

该消息用于其他模块向 camera_node 模块下发摄像头控制命令。其 msg 文件名称为：CameraCmd.msg。其内容定义如下：

字段	ROS 类型	含义	取值
header	Header	状态消息头	ROS 消息头
sender	string	发送者	消息发送者
receiver	string	接收者	消息接收者
trans_id	uint32	会话号	命令和应答保持一致
cmd	string	控制命令	摄像头控制命令
data	string	控制参数	控制命令附带的参数

摄像头控制命令和对应参数，和 camera_control 中 reqs 以及 request data 保持一致。

2.6.3. camera_ack

该消息用于 camera_node 模块向其他模块发送命令响应消息。其 msg 文件名称为：CameraAck.msg。其内容定义如下：

字段	ROS 类型	含义	取值
header	Header	状态消息头	ROS 消息头
sender	string	发送者	消息发送者
receiver	string	接收者	消息接收者
trans_id	uint32	会话号	命令和应答保持一致
ack	string	应答消息	摄像头命令响应
data	string	应答结果	摄像头响应结果

摄像头响应命令和响应结果数据，和 camera_control 中 response 以及 response data 保持一致。

2.7. 热像仪接口

热像仪模块根据不同设备（目前支持 flier、巨哥、飞普乐 3 种），通过调用设备底层的 sdk 包和控制协议完成相应的控制。所有热像仪模块都提供相同的接口和调用方式，但根据不同的热像仪厂家提供的 sdk 和控制协议，不同热像仪可能只支持部分接口和功能。

热像仪同时提供 topic 和 service 两种控制方式，实际根据需要选用。

云台控制相关的消息定义如下：

消息名	类型	定义	收发方向
infrared_control	service	ControlService	preset_scale =>infrared_imager find_scale =>infrared_imager
infrared_cmd	topic	GeneralCmd	preset_scale =>infrared_imager find_scale =>infrared_imager
infrared_ack	topic	GeneralAck	infrared_imager => preset_scale infrared_imager => find_scale

2.7.1. infrared_control

热像仪控制 service 定义如下：

service 类型	名称	ROS 类型	含义	取值
request	header	Header	命令消息头	ROS 消息头
	sender	string	发送者	消息发送者
	receiver	string	接收者	消息接收者
	trans_id	uint32	会话号	命令和应答保持一致
	cmd	string	控制命令	见后续说明
	data	string	命令数据	见后续说明
response	trans_id	uint32	会话号	命令和应答保持一致
	ack	string	控制响应	见后续说明
	data	string	响应数据	见后续说明

terrace_control 提供的各种服务，其请求、响应和对应数据的含义如下：

服务类型	cmd	cmd_data	ack	ack_data
抓取温度图	save_temperature		success/failed	--
抓图	capture_picture	抓图文件名	success/failed	--
最高温度（人工）	get_max_temperature	--	success/failed	最高温度
标记点温度	get_spot_temperature	标记点坐标(x,y)	success/failed	标记点温度
最高温度（自动）	get_high_temperature	--	success/failed	最高温度
最低温度	get_low_temperature	--	success/failed	最低温度
自动聚焦	auto_focus	--	success/failed	--
设置标记点坐标	set_measure_spot	标记点坐标(x,y)	success/failed	--

说明：

1.如果云台正在接受手柄的控制，则所有 service 都立即返回 true，并且 response_data 值为 busy。

2.7.2. infrared_cmd

命令消息用于服务器向机器下发命令，其内容定义如下：

字段	ROS 类型	含义	取值
header	Header	状态消息头	ROS 消息头
sender	string	发送者	消息发送者
receiver	string	接收者	消息接收者
trans_id	uint32	会话号	命令和应答保持一致
cmd	string	热像仪控制命令	见下表
data	string	控制数据	见下表

2.7.3. infrared_ack

命令消息用于服务器向机器下发命令，其内容定义如下：

字段	ROS 类型	含义	取值
header	Header	状态消息头	ROS 消息头
sender	string	发送者	消息发送者
receiver	string	接收者	消息接收者
trans_id	uint32	会话号	命令和应答保持一致
ack	string	热像仪应答消息	见下表
data	string	应答数据	见下表

热像仪响应命令和响应结果数据，和 infrared_control 中 response 以及 response data 保持一致。

2.8. 录表接口

预置位录入接口用于进行预置位录入操作。云台调整好角度，摄像头调整好缩放倍数并完成聚焦后，由后台服务模块发送 `preset_scale_cmd` 命令至 `preset_scale` 模块，`preset_scale` 模块处理并记录相关数据后回应 `preset_scale_ack` 至后台服务模块。

找表相关的消息定义如下：

消息名	类型	定义	收发方向
<code>preset_scale_cmd</code>	topic	<code>PresetScaleCmd</code>	server => <code>preset_scale</code>
<code>preset_scale_ack</code>	topic	<code>PresetScaleAck</code>	<code>preset_scale</code> => server
<code>nav_status</code>	topic	<code>NavStatus</code>	move_base => <code>preset_scale</code>

2.8.1. preset_scale_cmd

该消息用于 server 向 `preset_scale` 模块下发录表命令。其 msg 文件名称为：`PresetCmd.msg`。其内容定义如下：

字段	ROS 类型	含义	取值
<code>header</code>	<code>Header</code>	状态消息头	ROS 消息头
<code>sender</code>	<code>string</code>	发送者	消息发送者
<code>receiver</code>	<code>string</code>	接收者	消息接收者
<code>trans_id</code>	<code>uint32</code>	会话号	命令和应答保持一致
<code>preset_id</code>	<code>string</code>	预置位编号	预置位 id 号
<code>infrared</code>	<code>bool</code>	是否红外	true: 红外拍摄; false: 可见光拍摄
<code>type</code>	<code>string</code>	表计类型	用于说明表计类型
<code>extra_data</code>	<code>string</code>	其他信息 中用逗号分隔	字段 0: 机械臂录表是否保存轨迹文件 0: 不保存轨迹文件 1: 保存轨迹文件

2.8.2. preset_scale_ack

该消息用于 `preset_scale` 模块向 server 返回录表结果。其 msg 文件名称为：`PresetAck.msg`。其内容定义如下：

字段	ROS 类型	含义	取值
<code>header</code>	<code>Header</code>	状态消息头	ROS 消息头
<code>sender</code>	<code>string</code>	发送者	消息发送者
<code>receiver</code>	<code>string</code>	接收者	消息接收者
<code>trans_id</code>	<code>uint32</code>	会话号	命令和应答保持一致
<code>result</code>	<code>string</code>	执行结果	表示操作是否成功
<code>directory</code>	<code>string</code>	目录名	存储预置位信息的目录名

2.8.3. 其他说明

1. 录表模块同时也订阅 `heartbeat_ack` 消息，用于获取机器人当前的方向角，记录到预置位信息文件中，供找表时计算云台方向用

2. 录表模块将预置位信息保存在特定目录中，目录名定义：配置路径/预置位编号/创建时间/。

例如：`home/ros/preset/000001/20171219_100123/`。

3. 预置位目录中包含的文件：

(1) `zoomX.jpg`：预置位图片，包含不同放大倍数的图片。其中 `X` 为 0~6，`zoom0.jpg` 是最终放大倍数的图片，根据缩放倍数不同，最终保存的图片数量也有差异。

(2) `preset.ini`：预置位相关配置信息，由录表模块生成

(3) `zoomX_yyyymmdd_hhmmss_xx.jpg`：找表临时文件，由找表模块生成。其中 `yyymmdd_hhmmss` 分别表示年月日-时分秒，`xx` 表示第几次移动云台。

(4) `preset_calib_info.ini`：表计标定信息，录表完成后标定表计时由标定工具生成

(5) `zoomX.xml`：用于保存 kcf 算法需要的预置框位置信息，录表完成后由预置框标定工具生成

2.9. 找表接口

找表相关的消息定义如下：

消息名	类型	定义	收发方向
<code>find_scale_action</code>	<code>actionlib</code>	<code>GeneralAction</code>	<code>task_manager => find_scale</code>
<code>nav_status</code>	<code>topic</code>	<code>NavStatus</code>	<code>move_base => find_scale</code>

2.9.1. find_scale_action

`find_scale_action` 采用通用 action 消息定义，其对应字段定义如下：

action 类型	字段	ROS 类型	含义	取值
goal	header	Header	状态消息头	ROS 消息头
	sender	string	发送者	消息发送者
	receiver	string	接收者	消息接收者
	trans_id	uint32	会话号	命令和应答保持一致
	type	string	请求类型	见后续说明
	data	string	json 数据	见后续说明
feedback	event_code	string	事件信息	见后续说明
	data	string	json 数据	见后续说明
result	ret_code	string	响应码	见后续说明
	data	string	json 数据	见后续说明

各字段取值含义如下：

action 类型	字段	取值
goal	type	<code>pre_find</code> ：找表预动作 <code>find_scale</code> ：找表

action 类型	字段	取值	
		find_foreign: 找异物 find_both: 找表同时找异物	
	data	type="pre_find" type="find_scale" type="find_foreign" type="find_both"	{ "preset_path":/home/ros/preset/xxxx, // 预置位目录 }
feedback	event_code	0: 预置位设置完成 1: 开始找表	
	data	event_code="0"	
		event_code="1"	
result	ret_code	0: 成功 1: 找表失败	
	data	event_code="0"	{ "filename":/home/ros/preset/xxx.jpg, "infrared_filename":"/home/ros/preset/xxx.jpg", "foreign_filename":/home/ros/preset/xxx.jpg, "roi_vertex": ["x1":1.0, "y1":1.0, "x2":2.0, "y2":2.0,], "infrared_roi_vertex": ["x1":1.0, "y1":1.0, "x2":2.0, "y2":2.0,] }
		event_code="1"	{ "reason":xxx//失败原因 }

2.9.2. 其他说明

1. 如果是可见光找表，则 filename 只包含可见光图片名；如果是红外预置位，则包含可见光和红外两幅图片名

2. 找表模块同时也会订阅 robot_command 消息，并判断其中 directory 是否为空。如果不为空，说明到达的目的地后会有找表动作，此时找表模块会根据预置位目录中保存的信息，提前转动云台节约找表时间

3. 找表模块同时也订阅 heartbeat_ack 消息，用于获取机器人当前的方向角，供找表时计算云台方向用

4. 找表模块返回的矩形框顶点坐标，每 8 个一组，分别表示矩形 4 个定点的 x, y 坐标。4 个定点按照左上、右上、右下、左下的顺序排列。如果有多个方框，则 8 个值一组，依次记录每个方框 4 个定点的 8 个坐标值。

2.10. 读表接口

机器人找表完成后，server 发送 read_scale_cmd 至 read_scale 模块进行读表操作，完成后由 read_scale 模块发送 read_scale_ack 至 server 反馈读表结果。

消息名	类型	定义	收发方向
read_scale_cmd	topic	ReadScaleCmd	server => read_scale
read_scale_ack	topic	ReadScaleAck	read_scale => server

2.10.1. read_scale_cmd

该消息用于服务器向图像处理模块下发读表命令。其 msg 文件名称为：ReadScaleCmd.msg。其内容定义如下：

字段	ROS 类型	含义	取值
header	Header	状态消息头	ROS 消息头
sender	string	发送者	消息发送者
receiver	string	接收者	消息接收者
trans_id	uint32	会话号	命令和应答保持一致
directory	string	预置位目录	找表预置位配置目录
foreign_detect	int32	异物检测	0：只读表，不检测异物； 1：只检测异物，不读表； 2：读表同时也检测异物。
filename	string[]	图片文件名	找表完成后保存的图片路径和文件名
foreign_filename	string[]	异物图片文件名	执行异物检测的图片路径和文件名
roi_vertex	int32[]	可见光目标方框	8 个一组，表示矩形顶点 x, y 坐标
roi_vertex_thermal	int32[]	红外目标方框	8 个一组，表示矩形顶点 x, y 坐标

说明：

1. 图片文件名使用 find_scale_ack 中的值。如果是可见光读表，则 filename 只包含可见光图片名；如果是红外读表，则包含可见光和红外两幅图片名。

2. 矩形框含义和找表模块 find_scale_ack 中一致，由后台直接将找表模块返回结果转发至读表模块。

2.10.2. read_scale_ack

该消息用于图像处理模块向服务器返回读表结果。其 msg 文件名称为：ReadScaleAck.msg。其内容定义如下：

字段	ROS 类型	含义	取值
header	Header	状态消息头	ROS 消息头
sender	string	发送者	消息发送者
receiver	string	接收者	消息接收者
trans_id	uint32	会话号	命令和应答保持一致
result	string	读表结果应答	success/failed
foreign_result	string	异物检测结果	success/failed
scale	float64[]	表计读数	图像处理模块读出的表计读数

字段	ROS 类型	含义	取值
foreign_scale	float64[]	异物读数	异物查找结果读数（预留）
possibility	float64[]	读数置信度	读表结果的置信度百分比（0~100）
foreign_possibility	float64[]	异物查找置信度	异物查找结果置信度百分比（0~100）
picture_path	string[]	图片保存路径	图片存储路径和文件名
foreign_picture_path	string[]	异物图片保存路径	异物图片存储路径和文件名

说明：

1. 图片保存路径形如：配置路径/预置位编号/图片文件名.jpg。其中图片文件名就是找表模块返回的图片文件名。例如：home/ros/picture/0000101/zoom0_201712191001_01.jpg。

2. 如果是可见光读表，则 picture_path 只包含可见光图片名；如果是红外读表，则包含可见光和红外两幅图片名。

2.11. 虚拟障碍物配置接口

虚拟障碍物功能主要用于限制机器人在导航地图上的活动范围，主要原理是在 move_base 路径规划模块在搜索路径时，避开地图上一些标记为无障碍物的地方，将机器人活动范围限制在一定安全区域内。另外诸如挂牌避障等功能也需要借助虚拟障碍物来实现。

虚拟障碍物配置接口主要用于后台对相关虚拟障碍物进行配置，采用 ros topic 方式来实现控制命令下发和完成情况反馈。

消息名	类型	定义	收发方向
virtual_obstacle_cmd	topic	GeneralCmd	server => move_base
virtual_obstacle_ack	topic	GeneralAck	move_base => server

虚拟障碍物的实现方式为在地图上的一组折线段，这些折线段可能是闭合的，有可能没有闭合。在实时地图上能够看到虚拟障碍物，但是定位模块并不会将其视作特征点进行定位特征匹配，而路径规划模块在路径规划时则会将其作为障碍物来处理，这样规划出的导航路径就可以根据需要被限制在一定区域和范围内。

虚拟障碍物配置接口就用来对各组虚拟障碍物折线段进行配置，可以增加、删除、修改和查询各个折线段。

2.11.1. virtual_obstacle_cmd

命令消息用于服务器向机器下发命令，配置虚拟障碍物或者查询当前虚拟障碍物，其内容定义如下：

字段	ROS 类型	含义	取值
header	Header	状态消息头	ROS 消息头
sender	string	发送者	消息发送者
receiver	string	接收者	消息接收者
trans_id	uint32	会话号	命令和应答保持一致
cmd	string	操作命令	见后续说明
data	string	命令数据	保留

2.11.2. virtual_obstacle_ack

应答消息用于向服务器反馈增加、删除、修改的虚拟障碍物折线段是否成功，如果是查询虚拟障碍物的请求，则反馈对应折线段的各个点位信息，其内容定义如下：

字段	ROS 类型	含义	取值
header	Header	状态消息头	ROS 消息头
sender	string	发送者	消息发送者
receiver	string	接收者	消息接收者
trans_id	uint32	会话号	命令和应答保持一致
ack	string	命令响应	见后续说明
data	string	响应数据	保留

2.12. 陀螺仪配置接口

陀螺仪控制接口用于配置 imu 进行校准、复位等操作。无论 imu 是集成到运动控制板上，还是独立接到工控机上，都可以通过 imu 配置接口对 imu 进行配置操作。

imu 配置接口采用 ros topic 方式来实现控制命令下发和完成情况反馈。

消息名	类型	定义	收发方向
imu_cmd	topic	GeneralCmd	server => robot_control
imu_ack	topic	GeneralAck	robot_control => server

imu 配置接口主要是通过手动发送配置消息的方式对 imu 进行各种配置，包含校准、复位等。

如果 imu 集成在运动控制板上，则运动控制模块负责 imu_cmd 和 imu_ack 的收发，如果 imu 是单独接在工控机上，则由单独的 imu 模块完成收发。

2.12.1. imu_cmd

命令消息用于向 imu 下发命令，配置 imu，其内容定义如下：

字段	ROS 类型	含义	取值
header	Header	状态消息头	ROS 消息头
sender	string	发送者	消息发送者
receiver	string	接收者	消息接收者
trans_id	uint32	会话号	命令和应答保持一致
cmd	string	操作命令	见后续说明
data	string	操作数据	见后续说明

2.12.2. imu_ack

应答消息用于反馈 imu 配置操作结果，其内容定义如下：

字段	ROS 类型	含义	取值
----	--------	----	----

字段	ROS 类型	含义	取值
header	Header	状态消息头	ROS 消息头
sender	string	发送者	消息发送者
receiver	string	接收者	消息接收者
trans_id	uint32	会话号	命令和应答保持一致
ack	string	命令响应	见后续说明
data	string	响应数据	保留

2.13. 障碍物检测接口

障碍物检测接口用于障碍物检测模块 `obstacle_detector` 实时发布检测到的 3D 障碍物点云信息。供导航、运动控制等模块使用。

障碍物检测接口用于障碍检测模块 `obstacle_detector` 发布当前检测到的障碍物点云信息。topic 消息类型为 `ObstacleStatus`，消息名为 `/obstacle_status`。

topic 名称	发布	订阅	用途
<code>obstacle_status</code>	<code>obstacle_detector</code>	<code>robot_control</code>	发布障碍物点云信息

该 topic 目前有 `robot_control` 模块订阅，作为超声波避障和放跌落的补充，用于控制机器人急停。

当 `obstacle_detector` 运行时不停检测是否出现威胁到机器人运动的障碍物，并通过该 topic 发布给人 `robot_control`。

2.13.1. obstacle_status

该消息用于发布障碍物点云信息，其内容定义如下：

字段	ROS 类型	含义	取值
header	Header	状态消息头	ROS 消息头
sender	string	发送者	消息发送者
receiver	string	接收者	消息接收者
trans_id	uint32	会话号	命令和应答保持一致
obstacle	bool	障碍物状态	true: 有障碍物 false: 无障碍物

说明：

1. 初始化时，所有模块都处于无障碍物状态；
2. 每当障碍物状态发生变化，都需要发布该消息；
3. 当前障碍物状态保持不变时，不需要重复发布。

2.14. 旋转云台接口

旋转云台接口用于录表时直接在后台软件上发送命令控制云台旋转，方便录制预置位。

该接口使用 `ros topic` 的方式实现，发布者是后台服务器，订阅者是 `preset_rotate` 模块。

消息名	类型	定义	收发方向
-----	----	----	------

消息名	类型	定义	收发方向
preset_rotate	topic	PresetRotate	server =>cloud_terrace

录表旋转云台接口用于录表时直接在后台软件上发送命令控制云台旋转，方便录制预置位。后台服务器发布当前画面的宽度、高度像素值，以及需要旋转的点相对于中心点的x、y方向像素偏移值，当 preset_rotate 收到该消息后，通过画面宽高和偏移量的像素值，以及当前摄像头放大倍数，计算出云台的偏转角度，然后控制云台旋转到目标位置。

2.14.1. preset_rotate

旋转云台消息用于服务器向录表旋转模块发布转动命令，其内容定义如下：

字段	ROS 类型	含义	取值
header	Header	状态消息头	ROS 消息头
sender	string	发送者	消息发送者
receiver	string	接收者	消息接收者
trans_id	uint32	会话号	命令和应答保持一致
width	int32	画面宽度	宽度像素值
height	int32	画面高度	高度像素值
delta_x	int32	水平偏移	水平偏移像素值（右正左负）
delta_y	int32	垂直偏移	垂直偏移像素值（上正下负）

2.15. 语音控制接口

机器人语音模块主要用于机器人和网页客户端电脑进行语音对讲控制。通过调用对应的播放器软件来播放推送的语音以及调整音量。

机器人语音模块采用 ros topic 方式来实现控制命令下发和完成情况反馈。

消息名	类型	定义	收发方向
voice_cmd	topic	VoiceCmd	other =>robot_voice

2.15.1. voice_cmd

命令消息用于服务器向机器下发命令，其内容定义如下：

字段	ROS 类型	含义	取值
header	Header	状态消息头	ROS 消息头
sender	string	发送者	消息发送者
receiver	string	接收者	消息接收者
trans_id	uint32	会话号	命令和应答保持一致
language	string	语种	chinese: 中文 english: 英文
data	string[]	语音数据	需要播放的语音数据

2.15.2. 命令说明

voice_cmd 命令中对应 data 的每个 string，根据其是否包含有特殊符号，表明不同含义。
其详细说明如下：

data string	含义	示例
[xx]	播放数值	[123]: 一百二十三
{xx}	播放数字	{123}: 一二三
(xx)	播放温度	(34): 三十四摄氏度
%xx%	播放百分比	%17%: 百分之十七
其他	播放对应的语音文件	init_complete: init_complete.wav