

# 用 户 手 册

RMP PLUS 401

# 安全

不正确使用 RMP PLUS 401，可能会导致 RMP PLUS 401 出现失控、碰撞或者跌落等情况，从而导致财产损失，人身伤害，甚至死亡。因此，为了减少风险，避免伤害，请阅读并遵循本手册中的所有说明和警告。

本文档中使用了以下安全消息传递约定：

警告！	警告您可能导致伤害甚至死亡的操作
谨慎！	警告您可能导致轻微或中等伤害的操作
请注意	表示重要信息，但不涉及人身伤害

## 警告！

- 不要坐、站或骑在 RMP PLUS 401 上。这样做可能会导致伤害。
- 不要控制 RMP PLUS 401 撞人或动物。碰撞可能导致伤害。
- 当 RMP PLUS 401 运行时，要时刻提醒附近的人。与 RMP PLUS 401 的意外碰撞可能导致伤害。
- RMP PLUS 401 可以快速加速，建议客户使用低速进行练习，直到用户熟悉控制 RMP PLUS 401，RMP PLUS 401 的意外移动可能会造成伤害。
- 请勿试图拆解电池，这样可能会导致触电、烧伤甚至引起火灾。尝试打开电池外壳会损坏电池外壳并释放有毒有害物质，也会导致电池无法使用。
- 与所有可充电电池一样，不要在易燃材料附近充电，可能会引发火灾。
- 如果电池外壳破损或电池发出异味、烟雾、过热或泄漏，请不要继续使用电池，避免接触从电池渗出的任何物质，以免造成中毒。

- 严格遵守并遵循电池上的警告标签上的所有安全信息。不这样做可能导致伤害甚至死亡。
- 请勿使用已经磨损严重或已经损坏的电缆，这样可能会电击自己或损坏 RMP PLUS 401。

### 谨慎！

- 要正确谨慎地设置性能参数。RMP PLUS 401 遵循发布给它的命令，用户有责任执行正确、安全的性能参数。
- 不给电池充电可能会对电池造成永久性的损坏。
- 只能使用 RMP PLUS 401 配套的充电器对电池进行充电。
- 在操作 RMP PLUS 401 之前，请务必阅读用户手册，熟悉 RMP PLUS 401 的操作与各个注意事项。

### 请注意

- 用户在与我公司沟通的情况下，若私自对底盘进行改造，导致出现事故，我公司不承担任何责任。

# 目录

---

1 产品介绍 .....	6
1.1 产品示意图 .....	6
1.2 部件说明 .....	7
1.3 遥控器 .....	8
1.3.1 遥控器示意图 .....	8
1.3.2 接收机对码 .....	9
1.3.3 遥控器控车说明 .....	9
1.3.4 上位机控车说明 .....	10
2 软件介绍 .....	13
2.1 给用户提供的文件 .....	13
2.2 接口功能介绍 .....	13
2.2.1 C/C++接口介绍 .....	13
2.2.2 ROS 接口介绍—SmartCar .....	16
2.2.3 故障代码信息表 .....	20
2.2.4 原地自转功能介绍 .....	22
3 固件升级和版本升级 .....	23
3.1 固件升级 .....	23
3.2 版本升级 .....	25
附录一 系统参数及模式切换逻辑 .....	29

附录二 连接器焊接说明 .....	31
附录三 连接器 Pin 角定义说明 .....	34
附录四 C/C++ API 参考文档 .....	35

# 1 产品介绍

## 1.1 产品示意图

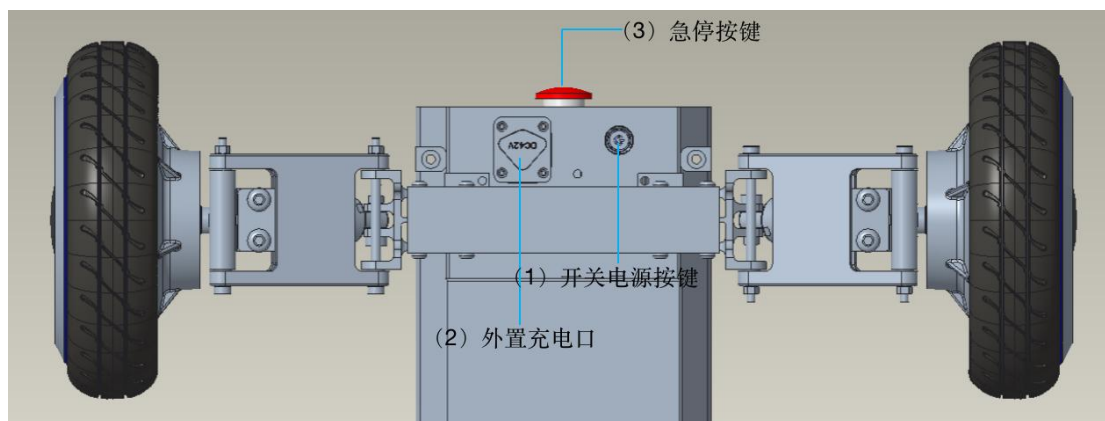


图 1

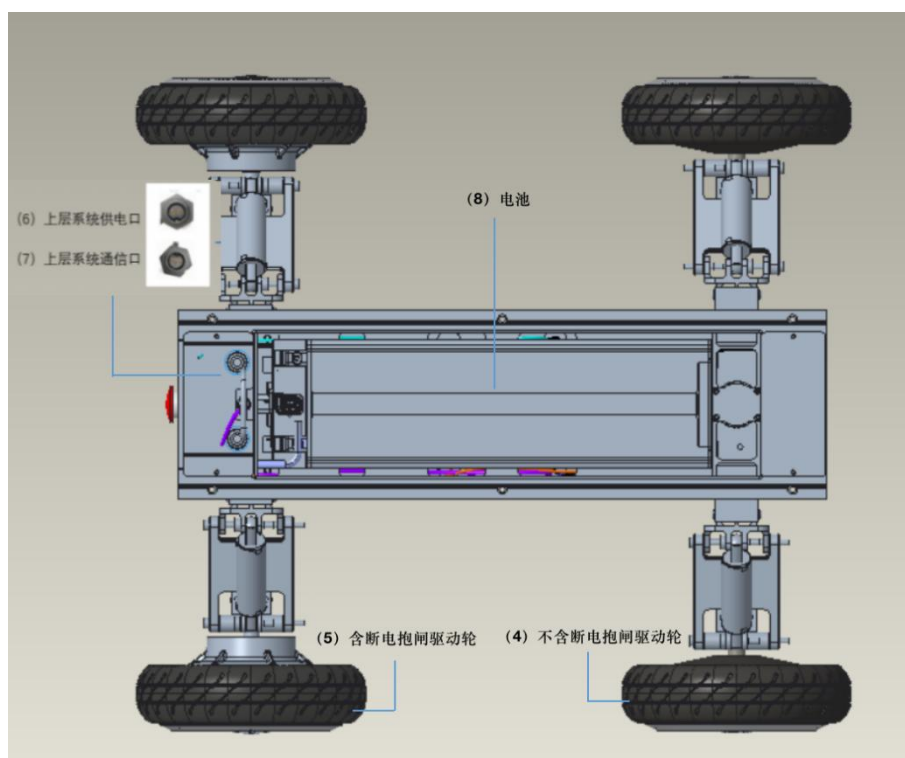


图 2

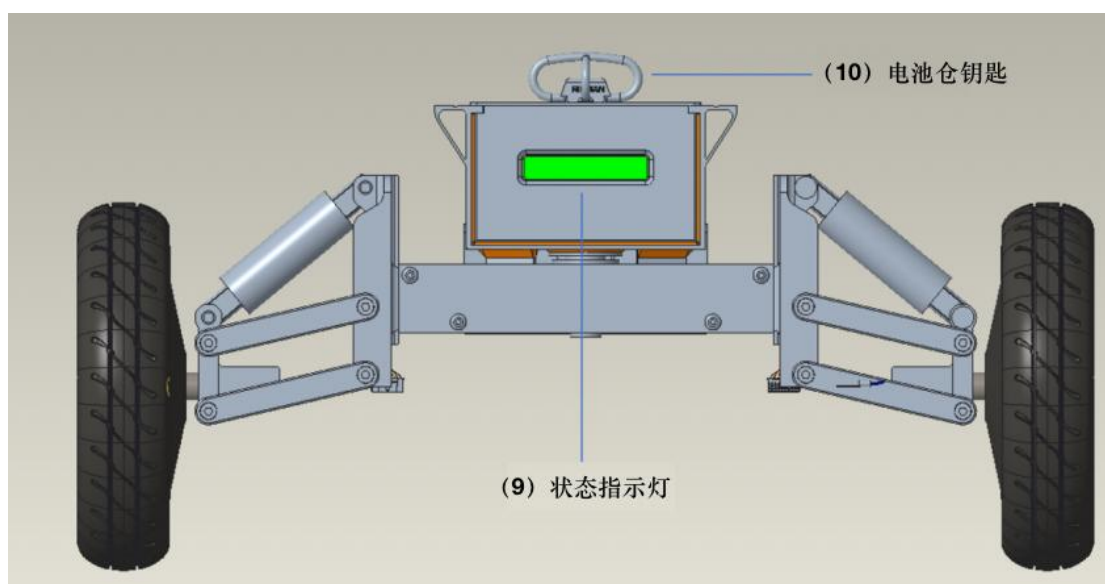


图 3

## 1.2 部件说明

表 1

序号	部件名称	描述
1	电源开关按键	<p>开机：长按至开关键和指示灯常亮并伴随有提示音时，底盘开机成功，此时底盘处于锁车模式，指示灯黄色常亮</p> <p>关机：长按开关键至提示音响起，松开开关键，底盘关机成功，此时开关键灯和指示灯全部熄灭</p>
2	外置充电口	连接充电器给设备进行充电
3	急停按键	用于紧急状态下将底盘切换为急停模式
4	不含断电抱闸驱动轮	8 英寸轮胎
5	含断电抱闸驱动轮	8 英寸轮胎，带抱闸
6	上层系统供电口	为上层系统供电（电压为电池电压）
7	上层系统通信口	包含 CAN、串口以及遥控器接收器
8	电池	给整个系统供电
9	状态指示灯	指示灯颜色和状态代表产品的不同模式
10	电池仓钥匙	用于打开固定电池的锁

## 1.3 遥控器

### 1.3.1 遥控器示意图



图 4

\*遥控器控制输入（油门或方向）的正向和反向可通过拨动 T8FB 下方的相位开关来实现。



图 5

\*遥控器报警电压自适应 2S, 3S, 4S 锂电池及 4 节镍氢电池，即如果 T8FB 用 2S, 3S, 4S 锂电池或者 4 节镍氢电池供电，接上电池后，T8FB 将根据电池类型自动设置低电压报警值。



### 1.3.2 接收机对码

每一台发射机都有独立的 ID 编码，开始使用设备前，接收机必须与发射机对码，对码完成后，ID 编码则储存在接收机内，且不需要再次对码，除非接收机再次与另外一台发射机配套使用。当您购买了新的接收机，必须要重新对码，否则接收机将无法正常使用。

- (1) 将遥控器和接收机水平放置，两者间距 50 厘米左右；
- (2) 打开遥控器电源开关，给接收机供电，接收机 LED 灯开始慢闪；
- (3) 按下接收机侧面的对码键（ID SET）1 秒钟以上，接收机的 LED 灯开始快闪，表示正在对码，接收机将寻找与之最近的遥控器进行对码；
- (4) 将接收机 LED 灯停止闪烁，即表示对码完成，如果接收机 LED 灯慢闪，表示对码失败，需要再次对码。

### 1.3.3 遥控器控车说明

- (1) RMP PLUS 401 底盘开机：按下 RMP PLUS 401 开关键；  
  
注：请检查 RMP PLUS 401 状态。长按至蜂鸣器响且无持续滴滴长鸣，指示灯黄色常亮。
- (2) 打开遥控器：向上推动遥控器电源开关；  
  
注：保证遥控器非急停状态，且进入使能状态。即急停开关不在下方，使能开关由上方拨往下方。
- (3) RMP PLUS 401 此时处于 Normal 模式，具体操作见下表：

表 2 控车与遥控器操作

控车	遥控器操作
左右转动	方向舵操作杆左右拨动
前后移动	油门操作杆前后拨动
急停/退出急停	遥控器急停开关：最上方——退出急停；最下方——开启急停
调节角速度的最大值	遥控器角速度最大值调节旋钮。左旋，角速度最大值降低，右旋反之。
调节线速度的最大值	遥控器线速度最大值调节旋钮。左旋，线速度最大值降低，右旋反之。
使能开启/关闭	使能开关由上方拨往下方，使能成功；使能开关由下方拨往上方，使能关闭。

### 1.3.4 上位机控车说明

上位机为 PC 端的控制计算机，可以直接发出操控命令，屏幕上显示各种信息变化。上位机控制下位机并为下位机提供一些必须的运行环境，并延伸下位机所能提供的人机控制或演示功能。上位机具有领导管理、协调资源、监控代办、控制 RMP PLUS 401 的特点。

(1) RMP PLUS 401 底盘开机：按下 RMP PLUS 401 开关键；

注 1：请检查 RMP PLUS 401 状态。长按至蜂鸣器响且无持续滴滴长鸣，指示灯黄色常亮。

注 2：上位机控车时，不能打开遥控器。或遥控器打开后，将其使能开关拨往上方。

(2) 确保 RMP PLUS 401 串口线或 CAN 线与上位机连接；

(3) 上位机中，给/sdcard/segway/hardware\_log/文件夹加权限，否则新建 log 文件时会失败；给 /catkin\_ws/src/RosCode/segwayRMP PLUS 401/lib/目录下所有文件加权限（首次设置后无需重设）：

‘ ’

```
`cd /sdcard/segway/hardware_log`

`sudo chmod 777 /sdcard/segway/hardware_log/`

`cd $PRO_HOME$/catkin_ws/src/RosCode/segwayRMP PLUS
401/lib/`

`sudo chmod 777 * `

`,`
```

(4) catkin\_ws/src/RosCode/segwayRMP PLUS 401/Cmakelists.txt 文档中，根据上位机在 x86\_64 还是 arm 平台下选择编译选项，如下是在 x86\_64 平台下编译时通过符号#来注释掉 libctrl\_arm64-v8a.so，（首次设置后无需重设）：

```
`,`

`target_link_libraries(SmartCar`

`${catkin_LIBRARIES}`

`${PROJECT_SOURCE_DIR}/lib/libctrl_arm64-v8a.so //x86_64 下，
注释该行，arm 下解除该行注释`

`${PROJECT_SOURCE_DIR}/lib/libctrl_x86_64.so //arm 下，注
释该行，x86_64 下解除该行注释`

`,`
```

(5) 进入 ros 的工作空间，运行如下命令,编译 segway\_msgs 包消息。

```
`,`

cd catkin_ws
```

```
catkin_make
```

```
-DCATKIN_WHITELIST_PACKAGES='segway_msgs'
```

```
‘ ’
```

(6) 进入 ros 的工作空间，运行如下命令,编译 segwayrmp 包。

```
‘ ’
```

```
cd catkin_ws
```

```
catkin_make -DCATKIN_WHITELIST_PACKAGES='segwayrmp'
```

```
‘ ’
```

(7) ros 中执行控车：

1) 新建 terminal，如下命令

```
‘ ’
```

```
cd catkin_ws
```

```
roscore
```

```
‘ ’
```

2) 新建 terminal，如下命令，运行 SmarCar 节点

```
‘ ’
```

```
cd catkin_ws
```

```
source devel/setup.bash
```

```
roslaunch segwayrmp SmartCar
```

```
‘ ’
```

3) 新建 terminal，如下命令，运行例程测试节点

‘ ’

```
cd catkin_ws
```

```
source devel/setup.bash
```

```
roslaunch segwayrmp drive_segway_sample
```

‘ ’

## 2 软件介绍

本章介绍 RMP PLUS 401 提供的相关文件、软件接口功能、故障代码信息。

### 2.1 给用户提供的文件

表 3 提供的文件

文件	功能
Libctrl_x86_64.so	提供 x86 平台 C/C++底盘相关的接口
Libctrl_arm64-v8a.so	提供 arm 平台 C/C++底盘相关的接口
Comm_ctrl_navigation.h	C/C++ API 接口头文件
ROS 包	提供底盘控制的 ROS 节点

### 2.2 接口功能介绍

#### 2.2.1 C/C++接口介绍

表 4 callback 数据类型

回调类型	回调索引	功能描述	数据结构
Chassis_Data_Motors_Speed	1	底盘 4 个轮子速度信息	typedef struct{ int16_t fl_speed; int16_t fr_speed; int16_t rl_speed; int16_t rr_speed; }chassis_motors_speed_data_t;
Chassis_Data_Car_Speed	2	获取底盘速度信息	typedef struct{ int16_t car_speed; int16_t turn_speed; }chassis_car_speed_data_t;
Chassis_Data_Front_Ticks	3	底盘前两轮编码器信息	typedef struct{ int32_t fl_ticks; int32_t fr_ticks; }front_motors_ticks_t;
Chassis_Data_Rear_Ticks	4	底盘后两轮编码器信息	typedef struct{ int32_t rl_ticks; int32_t rr_ticks; }rear_motors_ticks_t;
Chassis_Data_Odom_Pose_xy	5	Odom pose 信息	typedef struct{ float pos_x; float pos_y; }odom_pos_xy_t;
Chassis_Data_Odom_Euler_xy	6	Odom Euler x/y 轴信息	typedef struct{ float euler_x; float euler_y; }odom_euler_xy_t;
Chassis_Data_Odom_Euler_z	7	Odom Euler z 轴信息	typedef struct{ float euler_z; }odom_euler_z_t;
Chassis_Data_Odom_Linevel_xy	8	Odom speed x/y 轴信息	typedef struct{ float vel_line_x; float vel_line_y; }odom_vel_line_xy_t;
Chassis_Data_Imu_Gyr	9	陀螺仪数据	typedef struct{ int16_t gyr[3]; }imu_gyr_original_data_;
Chassis_Data_Imu_Acc	10	加速度计数据	typedef struct{ int16_t acc[3];

			}imu_acc_original_data_;
--	--	--	--------------------------

注 1：Odom 数据：开机默认航向角为 0 度。

注 2：IMU（陀螺仪和加速度计）数据：载体坐标系 xyz 对应右、前、上。

表 5 event 定义

事件类型	事件索引	功能描述
ChassisBootReadyEvent	1	底盘中控板启动完成
PadPowerOffEvent	2	底盘关机
OnEmergeStopEvent	3	进入急停
OutEmergeStopEvent	4	退出急停
OnLockedRotorProtectEvent	5	发生堵转错误现象
OutLockedRotorProtectEvent	6	堵转现象消除
OnLostCtrlProtectEvent	7	发生飞转现象
OutLostCtrlProtectEvent	8	飞转现象消除
CalibrateGyroSuccess	9	校准陀螺仪成功
CalibrateGyroFail	10	校准陀螺仪失败
CalibratePasheCurrentSuccess	11	校准相电流成功
CalibratePasheCurrentFail	12	校准相电流失败
ChassisLockRotorWarning	13	发生堵转告警现象

表 6 get/set 接口

接口名称	接口描述
get_err_state	获取上位机/中控板/电机板/电池的错误码
get_bat_soc	获取电池剩余电量百分比
get_bat_charging	获取电池充电状态（1：充电；0：非充电）
get_bat_mvol	获取电池电压（单位：毫伏）
get_bat_mcurrent	获取电池电流（单位：毫安）
get_bat_temp	获取电池温度（单位：摄氏度）
get_chassis_work_model	获取底盘工作状态（0：泄力；1：加力）
get_chassis_load_state	获取底盘负载参数设置（0：空载；1：满载）
get_chassis_mode	获取底盘状态机（0：锁车；1：控车；2：推行；3：急停；4：错误）
get_ctrl_cmd_src	获取底盘当前控制源（0：遥控器；1：上位机）
get_vehicle_meter	获取底盘里程（单位：米）
get_host_version	获取上位机版本号
get_chassis_central_version	获取中控板版本号
get_chassis_motor_version	获取电机板版本号（预留）
get_line_forward_max_vel_fb	获取底盘前向速度限制值（单位：米每小时）
get_line_backward_max_vel_fb	获取底盘后向速度限制值（单位：米每小时）

get_angular_max_vel_fb	获取底盘角速度限制值（单位：毫弧度每秒）
getlapProgress	获取 IAP 进度
iapCentralBoard	对中控板 IAP 升级
iapMotorBoard	对电机板 IAP 升级
iapBrakeBoard	对抱闸板 IAP 升级（带抱闸版）
isHostlapOver	查询是否 IAP 结束
getHostlapResult	获取 IAP 结果（3：完成；4：失败；5：中断；0：无意义）
getHostlapErrorCode	获取 IAP 错误码
get_calibrate_mid_value_status	查询是否已经校正过前轮中值（1：已校正； 0：未校正）
get_chassis_central_SN	获取底盘中控板 MCU 的唯一的码值
get_rotate_switch_stat	查询底盘当前是否支持原地自转
set_cmd_vel	设置底盘线速度和角速度（单位：米每秒和弧度每秒）
set_line_forward_max_vel	设置底盘前向速度限幅（单位米每秒）
set_line_backward_max_vel	设置底盘后向速度限幅（单位米每秒）
set_angular_max_vel	设置底盘角速度限幅（单位弧度每秒）
set_enable_ctrl	设置底盘上位机控车使能状态（1：使能；0 失能）
init_control_ctrl	底盘初始化接口
exit_control_ctrl	底盘去初始化接口
set_smart_car_serial	设置上位机动态库使用的串口端口名称
set_comu_interface	设置与底盘通信的通信接口（0：串口；1：CAN）
set_chassis_load_state	设置底盘负载参数（0：空载；1：满载）
set_chassis_poweroff	下发底盘关机命令
setHostlapCanceled	取消上位机 IAP 命令
set_calibrate_mid_value	下发校正前两轮角度中值的命令
reset_host_power_time_s	下发命令使底盘复位上位机电源（单位：秒；最大间隔：65535 秒）
clear_chassis_error_code	清除底盘错误码，不包括告警、异常、电池错误。 <b>慎重使用</b>
enable_rotate_switch	使能底盘原地自转功能

## 2.2.2 ROS 接口介绍—SmartCar

表 7 消息发布

Topic Name	功能描述	Message Type	Message Type Info	频率
Bms_fb	电池相关信息	Segway_msgs/ Bms_fb	int16 bat_soc int16 bat_charging int32 bat_vol int32 bat_current int16 bat_temp	1
Chassis_ctrl_src_fb	底盘控制命令来源	Segway_msgs/ Chassis_ctrl_src_fb	uint16 chassis_ctrl_cmd_src	1



Chassis_mileage_meter_fb	底盘里程	Segway_msgs/ Chassis_mileage_meter_fb	uint32 vehicle_meters	1
Chassis_mode_fb	底盘状态机	Segway_msgs/ Chassis_mode_fb	uint16 chassis_mode	1
Error_code_fb	底盘错误码	Segway_msgs/ Error_code_fb	uint32 host_error uint32 central_error uint32 front_left_motor_error uint32 front_right_motor_error uint32 rear_left_motor_error uint32 rear_right_motor_error uint32 bms_error	1
Motor_work_mode_fb	底盘工作状态	Segway_msgs/ Motor_work_mode_fb	uint16 motor_work_mode #0: no output torque 1: output torque	1
Speed_fb	底盘速度	Segway_msgs/ Speed_fb	float32 car_speed float32 turn_speed float32 fl_speed float32 fr_speed float32 rl_speed float32 rr_speed uint64 speed_timestamp	40
Ticks_fb	底盘编码器信息	Segway_msgs/ Ticks_fb	int32 fl_ticks int32 fr_ticks int32 rl_ticks int32 rr_ticks uint64 ticks_timestamp	40
Odom	Odom 数据	Nav_msgs/odom		40
Imu	Imu 数据	Sensor_msgs/imu		40

表 8 消息订阅

TopicName	功能描述	Message Type	Message Type Info
Cmd_vel	控制底盘运动	Geometry_msgs/twist	Angular.z //rad/s Linear.x //m/s

表 9 服务 client

Service name	功能描述	Message type	Message type info
chassis_send_event_srv	发送事件号	Segway_msgs/chassis_send_event	chassis_send_event_id--- ros_is_received

表 10 服务 server

Service name	功能描述	Message type	Message type info
ros_clear_chassis_error_code_cmd.srv	清除底盘错误, 不包括告警、异常、电池错误。 <b>非常慎重使用</b>	Segway_msgs/ros_clear_chassis_error_code_cmd	bool clear_chassis_error_code_cmd --- uint8 clear_chassis_error_code_result
ros_enable_chassis_rotate_cmd.srv	使能底盘原地自转模式	Segway_msgs/ros_enable_chassis_rotate_cmd	bool ros_enable_chassis_rotate_cmd --- int16 chassis_enable_rotate_result
ros_get_chassis_rotate_switch_cmd.srv	查询底盘是否处于原地自转	Segway_msgs/ros_get_chassis_rotate_switch_cmd	bool ros_get_chassis_rotate_cmd --- uint8 chassis_rotate_state
ros_get_chassis_SN_cmd.srv	获取底盘中控 MCU 的 SN	Segway_msgs/ros_get_chassis_SN_cmd	bool ros_get_chassis_SN --- string chassis_SN
ros_get_load_param_cmd.srv	获取负载设置	Segway_msgs/ros_get_load_param_cmd	ros_get_load_param --- get_load_param #0:no_load, 1: full_load
ros_get_sw_version_cmd.srv	获取软件版本	Segway_msgs/ros_get_sw_version_cmd	ros_get_sw_version_cmd --- uint16 host_version

			uint16 central_version uint16 motor_version
ros_get_vel_max_feedback_cmd_srv	获取速度限幅	Segway_msgs/ ros_get_vel_max_feedback_cmd	ros_get_vel_max_fb_cmd --- forward_max_vel_fb backward_max_vel_fb angular_max_vel_fb
ros_set_chassis_enable_cmd_srv	下发底盘使能命令	Segway_msgs/ ros_set_chassis_enable_cmd	ros_set_chassis_enable_cmd --- chassis_set_chassis_enable_result
ros_set_chassis_poweroff_cmd_srv	下发底盘关机命令	Segway_msgs/ ros_set_chassis_poweroff_cmd	ros_set_chassis_poweroff_cmd --- chassis_set_poweroff_result
ros_set_load_param_cmd_srv	设置底盘负载	Segway_msgs/ ros_set_load_param_cmd	ros_set_load_param #0:no_load, 1: full_load --- chassis_set_load_param_result
ros_set_vel_max_cmd_srv	设置速度限幅	Segway_msgs/ ros_set_vel_max_cmd_srv	ros_set_forward_max_vel ros_set_backward_max_vel ros_set_angular_max_vel --- chassis_set_max_vel_result
ros_reset_host_power_cmd_srv	底盘复位上位机电源	segway_msgs/ros_reset_host_power_cmd	uint16 reset_interval_time --- uint8 reset_result

表 11 动作 server

Action name	功能描述	Message type	Message type info
ros_set_iap_cmd_action	对单板固件 IAP 升级	Segway_msgs/ ros_set_iap_cmdAction	uint16 board_index_for_iap --- Int16 iap_result #3: iap_state_complete; 4: iap_state_fail; 5: iap_state_abort Int16 error_code #When iap_result value is 4, this value represents the error code --- Int16 iap_percent

## 2.2.3 故障代码信息表

故障代码通过 `uint32_t get_err_state(board_name_e board_name)`

接口获取, 其对应信息如下, (注意: 需非常慎重实施“手动强制清除错误”):

表 12 故障码

Board name	Bit	Error info	底盘动作	处理
host	0x00000000	无错误		
	0x00000001	中控板失联	2 级, 无法控车	检查通信
	0x00000002	串口模块拔出	2 级, 无法控车	插回模块
Central	0x00000000	无错误		
	0x00000001	控车命令通信中断	2 级, 底盘停车, 锁车模式	检查通信
	0x00000002	电机板通信中断	2 级, 底盘停车, 锁车模式	检查电机板通信
	0x00000004	IMU 初始化失败	1 级, 可控车, 角度错误	低速驶回, 检测硬件
	0x00000008	IMU 读数据失败	1 级, 可控车, 角度错误	低速驶回, 检测硬件
	0x00000010	飞转	5 级, 停车, 错误模式	手动清除错误, 低速驶到安全处, 检测硬件
	0x00000020	堵转	3 级, 停车, 错误模式	手动清除错误, 低速驶到安全处, 检测硬件
	0x00000040	标定 IMU 失败	1 级, 可控车, 角度错误	低速驶回, 检测硬件
	0x00000080	Flash 读取失败	3 级, 停车, 错误模式	手动清除错误, 低速驶到安全处, 检测硬件
	0x00000100	IMU 数据更新失败	1 级, 可控车, 角度错误	低速驶回, 检测硬件
	0x00000400	翻车	2 级, 停车, 锁车模式	扶正底盘
	0x00000800	检测到任一电机板重启	2 级, 停车, 锁车模式	检测电机板通信
	0x00001000	前左轮磁编码器故障	5 级, 停车, 错误模式	手动清除错误, 低速驶到安全处, 检测硬件
	0x00002000	前右轮磁编码器故障	5 级, 停车, 错误模式	手动清除错误, 低速驶到安全处, 检测硬件
	0x00004000	电池通信中断	3 级, 停车, 错误模式	手动清除错误, 低速驶到安全处, 检测硬件

				件
	0x00008000	后左轮磁编码器故障	5 级, 停车, 错误模式	手动清除错误, 低速驶到安全处, 检测硬件
	0x00010000	后右轮磁编码器故障	5 级, 停车, 错误模式	手动清除错误, 低速驶到安全处, 检测硬件
	0x00200000	前轮角度收敛超时异常	2 级, 停车, 锁车模式	扶正前轮
Motor	0x00000000	无故障		
	0x00000001	相电流故障	4 级, 停车, 错误模式	手动清除错误, 低速驶到安全处, 检测硬件
	0x00000002	相电压故障	3 级, 停车, 错误模式	手动清除错误, 低速驶到安全处, 检测硬件
	0x00000004	缺相	3 级, 停车, 错误模式	手动清除错误, 低速驶到安全处, 检测硬件
	0x00000008	电压故障	4 级, 停车, 错误模式	手动清除错误, 低速驶到安全处, 检测硬件
	0x00000010	自测故障	3 级, 停车, 错误模式	手动清除错误, 低速驶到安全处, 检测硬件
	0x00000020	过流	4 级, 停车, 错误模式	手动清除错误, 低速驶到安全处, 检测硬件
	0x00000080	堵转	4 级, 停车, 错误模式	手动清除错误, 低速驶到安全处, 检测硬件
	0x00000100	电角度故障	3 级, 停车, 错误模式	手动清除错误, 低速驶到安全处, 检测硬件
	0x00000200	功率过大故障	4 级, 停车, 错误模式	检测硬件
	0x00000400	超速故障	5 级, 停车, 错误模式	手动清除错误, 低速驶到安全处, 检测硬件
	0x00000800	转速传感器故障	3 级, 停车, 错误模式	手动清除错误, 低速驶到安全处, 检测硬件
Battery	0x00000000	无故障		

BrakeSticking	0x00000200	过冲	3 级, 停车, 错误模式	检测硬件
	0x00000400	充电过温	3 级, 停车, 错误模式	检测硬件
	0x00000002	左抱闸开路	3 级, 停车, 错误模式	手动清除错误, 低速 驶到安全处, 检测硬 件
	0x00000004	左抱闸短路	4 级, 停车, 错误模式	手动清除错误, 低速 驶到安全处, 检测硬 件
	0x00000008	左抱闸关闭失败	3 级, 停车, 错误模式	手动清除错误, 低速 驶到安全处, 检测硬 件
	0x00000200	右抱闸开路	3 级, 停车, 错误模式	手动清除错误, 低速 驶到安全处, 检测硬 件
	0x00000400	右抱闸短路	4 级, 停车, 错误模式	手动清除错误, 低速 驶到安全处, 检测硬 件
	0x00000800	右抱闸关闭失败	3 级, 停车, 错误模式	手动清除错误, 低速 驶到安全处, 检测硬 件
	0x00008000	抱闸板通信失联	2 级, 停车, 错误模式	检测抱闸板通信

## 2.2.4 原地自转功能介绍

原地自转功能是底盘特殊模式下实现的功能, 当线速度为 0 同时角速度不为 0 时底盘使用此功能, 使用此功能时会导致后轮电流过大, 从而可能引起底盘和电机异常, 建议非必要不轻易使用。

当使用底盘原地自转时, 如出现底盘被卡转不动而堵转时, 底盘会有以下动作:

- 1、当堵转 5 秒左右后, 底盘取消对原地自转功能的支持, 同时发送堵转告警事件到上位机。此时需要停止继续原地自转, 使底盘有一个前后运动的分量从而释放底盘积累的电能。
- 2、当堵转发生 10 秒左右后, 底盘仍然无法释放电能时, 底盘自动释放积累的电能, 此时需防止底盘处于有坡度地面。
- 3、当堵转发生, 导致硬件损坏无法释放电能, 15 秒左右后底盘发送堵转错误事件到上位机, 同时底盘转到错误模式, 断电抱闸抱死。

底盘释放电能恢复正常后, 如确实非常需要继续使用原地自转功能, 则需如 2.2.2 节所述上位机需要通过 ROS 的“使能底盘原地自转模式”服务以使底盘再次支持原地自转。同时需要注意为了保护底盘, 底盘停止支持原地自转后 30 秒内无法使用上述服务使能底盘对原地自转功能的支持。

## 3 固件升级和版本升级

IAP 是系统的一个软件功能模块，即在应用编程，也就是在线对单片机程序进行升级。本功能通过上位机在程序运行时对单片机（包括中控板、电机驱动板等）进行新版本 bin 文件烧写：针对 V2.02.3 之前的发布版本，前提是需要把待烧写的单片机 bin 文件按照上位机的要求进行命名(如 central.bin), 然后放置到上位机的 /sdcard/firmware/ 路径下；针对 V2.02.3 及之后的发布版本，需要把待烧写的单片机 bin 文件按照上位机要求命名(如 central\_v2023.bin)，然后放置到 lib 库和 ctrl\_x86\_64 所在的文件夹下的 segway\_firmware 文件夹下；之后在终端通过命令进行在线升级。

### 3.1 固件升级

在进行固件升级之前需对上位机与各下位机的数据通信进行测试，查看是否正常通信。在 shell 终端利用命令进行测试。

#### （1）查看上位机程序路径

进入上位机程序所在路径，查看上位机可执行文件是否存在。如下图所示，分别是 arm 可执行文件、x86 可执行文件、arm 动态库、x86 动态库：

```
ubuntu@ubuntu:/home/project/EE_PROJECT_RMP/Project/RMP_panda/ROS/src/segwayrmp/lib$ ll
总用量 1742
drwxrwxrwx 1 root root 4096 6月 30 10:11 /
drwxrwxrwx 1 root root 4096 6月 16 17:29 /
-rwxrwxrwx 1 root root 93115 4月 20 17:50 adb*
-rwxrwxrwx 1 root root 414192 6月 30 10:11 ctrl_arm64-v8a*
-rwxrwxrwx 1 root root 386280 6月 30 10:11 ctrl_x86_64*
-rwxrwxrwx 1 root root 446544 6月 30 10:11 libctrl_arm64-v8a.so*
-rwxrwxrwx 1 root root 433816 6月 30 10:11 libctrl_x86_64.so*
ubuntu@ubuntu:/home/project/EE_PROJECT_RMP/Project/RMP_panda/ROS/src/segwayrmp/lib$
```

图 6

## (2) 查看各下位机板卡软件版本

查看下位机软件版本，此步可同时测试上位机与下位机的通信状况，若能通过上位机查看下位机各个板块的软件版本即表明通信良好。

中控板测试命令：`./ctrl_x86_64 s -test central`

1) 首次连接时，如果串口的 USB 端口没有赋予执行权限的话，程序要求获取 root 权限修改串口 USB 端口的可执行权限，此时需要输入系统的登录密码，输入后敲击回车键，如下图所示：

```
ubuntu@ubuntu: /home/project/EE_PROJECT_RMP/Project/RMP_panda/RUS/src/segwayrmp/ltb$ ./ctrl_x86_64 s -test central
.....Start Comucore!.....
host version build date:[21-06-29]
host version build time:[20:39:35]
Communication interface adding SERIAL_INTERFACE
Use the serial port[/dev/ttyUSB0]
Please enter the administrator permission login password:
[sudo] ubuntu 的密码:
```

图 7

2) 通信失败时，版本号为 0xFFFF，如下图所示：

```
ubuntu@ubuntu: /home/project/EE_PROJECT_RMP/Project/RMP_panda/RUS/src/segwayrmp/ltb$ ./ctrl_x86_64 s -test central
.....Start Comucore!.....
host version build date:[21-06-29]
host version build time:[20:39:35]
Communication interface adding SERIAL_INTERFACE
Use the serial port[/dev/ttyUSB0]
Please enter the administrator permission login password:
[sudo] ubuntu 的密码:
serial open success! serial port:/dev/ttyUSB0, baud:921600
Scheduler Num 0 Start. Task Num = 1. Period = 100000
Scheduler Num 1 Start. Task Num = 1. Period = 50000
Scheduler Num 2 Start. Task Num = 1. Period = 20000
当前测试RMP版本: 1.0.0

central board test started.....
get_chassis_central_version: 0xFFFF
get_chassis_Err_Status: 0x00000000
get_chassis_central_version: 0xFFFF
get_chassis_Err_Status: 0x00000000
get_chassis_central_version: 0xFFFF
get_chassis_Err_Status: 0x00000000
```

图 8

3) 通信成功时，版本号打印如下，并且查看该版本号是非 0xFFFF 数字，此时上位机与单片机通信正常，可进行在线升级：



```
ubuntu@ubuntu: /home/project/EE_PROJECT_RMP/Project/RMP_panda/ROS/src/segwayrmp/lib$ ./ctrl_x86_64 s -test central
.....Start Comucore!.....
host version build date:[21-06-29]
host version build time:[20:39:35]
Communication interface adding SERIAL_INTERFACE
Use the serial port[/dev/ttyUSB0]
Please enter the administrator permission login password:
serial open success! serial port:/dev/ttyUSB0, baud:921600
Scheduler Num 1 Start. Task Num = 1. Period = 50000
Scheduler Num 0 Start. Task Num = 1. Period = 100000
Scheduler Num 2 Start. Task Num = 1. Period = 20000
当前测试RMP版本: 1.0.0

central board test started.....
get_chassis_central_version: 0x1000
get_chassis_Err_Status: 0x00000000
get_chassis_central_version: 0x1000
get_chassis_Err_Status: 0x00000000
```

图 9

## 3.2 版本升级

### (1) 单片机 bin 文件放置

**V2.02.3 之前的发布版**，将待升级板卡软件的 bin 文件放到上位机

/sdcard/firmware 路径下，中控板软件 bin 文件 central.bin，前轮电机板 bin 文件 motor\_front.bin，后轮电机板 bin 文件 motor\_rear.bin，抱闸板 bin 文件 brake.bin（带抱闸版底盘特有），如下图文件位置所示

```
ubuntu@ubuntu: /sdcard/firmware$ ll
总用量 272
drwxrwxrwx 2 root root 4096 1月 7 10:33 ./
drwxrwxrwx 5 root root 4096 7月 26 16:28 ../
-rwxrwxrwx 1 root root 28708 11月 16 10:39 brake.bin*
-rwxrwxrwx 1 ubuntu ubuntu 91368 12月 31 14:59 central.bin*
-rwxrwxrwx 1 root root 67788 11月 12 20:49 motor_front.bin*
-rwxrwxrwx 1 root root 67788 11月 12 20:49 motor_rear.bin*
-rwxrwxrwx 1 root root 40 10月 13 10:09 password.txt*
ubuntu@ubuntu: /sdcard/firmware$
```

图 10a

**V2.02.3 及之后的发布版**，将待升级板卡固件 bin 文件放到上位机 libctrl\_x86\_64.so 所在的文件夹下的 segway\_firmware 文件夹下，如中控板 bin 文件 central\_v2023.bin，前轮电机板 bin 文件

motor\_front\_v2023.bin, 后轮电机板 bin 文件 motor\_rear\_v2023.bin,  
抱闸板 bin 文件 brake\_v2023.bin (带抱闸板底盘特有), 如下图所示:

```
ubuntu@ubuntu:/home/project/git_pro/EE_PROJECT_RMP/Project/RMP_panda/ROS/src/segwayrmp/lib/segway_firmware$ ll
总用量 268
drwxrwxr-x 2 ubuntu ubuntu 4096 3月 30 15:25 ./
drwxrwxr-x 3 ubuntu ubuntu 4096 3月 30 15:25 ../
-rw-rw-r-- 1 ubuntu ubuntu 28952 3月 30 15:25 brake_v2024.bin
-rw-rw-r-- 1 ubuntu ubuntu 93516 3月 30 15:25 central_v2024.bin
-rw-rw-r-- 1 ubuntu ubuntu 68672 3月 30 15:25 motor_front_v2024.bin
-rw-rw-r-- 1 ubuntu ubuntu 68672 3月 30 15:25 motor_rear_v2024.bin
ubuntu@ubuntu:/home/project/git_pro/EE_PROJECT_RMP/Project/RMP_panda/ROS/src/segwayrmp/lib/segway_firmware$
```

图 10b

## (2) 下位机 bin 文件在线烧写

进入上位机程序可执行文件 ctrl\_x86\_64 或 ctrl\_arm64-v8a 所在的路径, 如下:

```
ubuntu@ubuntu:/home/project/EE_PROJECT_RMP/Project/RMP_panda/ROS/src/segwayrmp/lib$ ll
总用量 1763
drwxrwxrwx 1 root root 4096 6月 30 10:11 ./
drwxrwxrwx 1 root root 4096 6月 16 17:29 ../
-rwxrwxrwx 1 root root 114264 6月 30 14:13 adb*
-rwxrwxrwx 1 root root 414192 6月 30 10:11 ctrl_arm64-v8a*
-rwxrwxrwx 1 root root 386280 6月 30 10:11 ctrl_x86_64*
-rwxrwxrwx 1 root root 446544 6月 30 10:11 libctrl_arm64-v8a.so*
-rwxrwxrwx 1 root root 433816 6月 30 10:11 libctrl_x86_64.so*
ubuntu@ubuntu:/home/project/EE_PROJECT_RMP/Project/RMP_panda/ROS/src/segwayrmp/lib$
```

图 11

在线升级各板的命令如下, 进入上位机程序路径下后执行下列命令 (使用串口时使用 's'; 使用 CAN 口时使用 'c'):

中控板升级命令 : `./ctrl_arm64-v8a c -iap central`

前轮电机板升级命令 : `./ctrl_arm64-v8a c -iap motor_front`

后轮电机板升级命令 : `./ctrl_arm64-v8a c -iap motor_rear`

抱闸板升级命令 : `./ctrl_arm64-v8a c -iap brake`(带抱闸版  
底盘特有)

以中控板为例，在此输入命令 `./ctrl_x86_64 s -iap central` 进行  
升级，如下图所示：

```
ubuntu@ubuntu: /home/project/EE_PROJECT_RMP/Project/RMP_panda/ROS/src/segwayrmp/lib$ ./ctrl_x86_64 s -iap central
.....Start Comucore!.....

host version build date:[21-06-29]
host version build time:[20:39:35]
Communication interface adding SERIAL_INTERFACE
Use the serial port[/dev/ttyUSB0]
Please enter the administrator permission login password:
serial open success! serial port:/dev/ttyUSB0, baud:921600
Scheduler Num 0 Start. Task Num = 1. Period = 100000
Scheduler Num 1 Start. Task Num = 1. Period = 50000
Scheduler Num 2 Start. Task Num = 1. Period = 20000
当前测试RMP版本: 1.0.0

IAP Start! path:/sdcard/firmware/central.bin id: 38 version:2.01
Id:0x38 version:2.01 Iap Progress 0: status: 2
Id:0x38 version:2.01 Iap Progress 0: status: 2
Id:0x38 version:2.01 Iap Progress 0: status: 2
Id:0x38 version:2.01 Iap Progress 0: status: 2
Id:0x38 version:2.01 Iap Progress 0: status: 2
Id:0x38 version:2.01 Iap Progress 0: status: 2
Id:0x38 version:2.01 Iap Progress 0: status: 2
Id:0x38 version:2.01 Iap Progress 0: status: 2
Id:0x38 version:2.01 Iap Progress 0: status: 2
Id:0x38 version:2.01 Iap Progress 0: status: 4
Id:0x38 version:2.01 Iap Progress 0: status: 4
Id:0x38 version:2.01 Iap Progress 0: status: 4
Id:0x38 version:2.01 Iap Progress 1: status: 4
Id:0x38 version:2.01 Iap Progress 1: status: 4
Id:0x38 version:2.01 Iap Progress 1: status: 4
Id:0x38 version:2.01 Iap Progress 2: status: 4
```

图 12

升级过程中可以查看升级进度，Progress 表示 IAP 升级进度的百分比，  
当 Progress 值达到 100 时，即表示已经把路由板 bin 文件烧写到中控板芯  
片中。如下图所示：

```
Id:0x38 version:2.01 Iap Progress 97: status: 4
Id:0x38 version:2.01 Iap Progress 98: status: 4
Id:0x38 version:2.01 Iap Progress 98: status: 4
Id:0x38 version:2.01 Iap Progress 98: status: 4
Id:0x38 version:2.01 Iap Progress 98: status: 4
Id:0x38 version:2.01 Iap Progress 99: status: 4
Id:0x38 version:2.01 Iap Progress 99: status: 4
Id:0x38 version:2.01 Iap Progress 99: status: 4
Id:0x38 version:2.01 Iap Progress 99: status: 4
Id:0x38 version:2.01 Iap Progress 99: status: 8
Id:0x38 version:2.01 Iap Progress 99: status: 8
Id:0x38 version:2.01 Iap Progress 99: status: 8
Id:0x38 version:2.01 Iap Progress 99: status: 8
Id:0x38 version:2.01 Iap Progress 99: status: 8
Id:0x38 version:2.01 Iap Progress 99: status: 8
Id:0x38 version:2.01 Iap Progress 99: status: 8
Id:0x38 version:2.01 Iap Progress 99: status: 8
Id:0x38 version:2.01 Iap Progress 100: status:10
Iap_success!
ubuntu@ubuntu: /home/project/EE_PROJECT_RMP/Project/RMP_panda/ROS/src/segwayrmp/lib$
```

图 13

### (3) 测试 IAP 版本在线升级结果:

执行步骤一，测试查看软件版本号，在上位机程序所在路径，输入命令./

*ctrl\_x86\_64 s -test central*，如下图所示：

此时得到中控板软件版本号为 xxxx 与新固件版本一致，说明在线升级已经成功，并且上位机与中控板通信状况良好。

```
ubuntu@ubuntu: /home/project/EE_PROJECT_RMP/Project/RMP_panda/ROS/src/segwayrmp/lib$ ./ctrl_x86_64 s -test central
.....Start Conucore!.....
host version build date:[21-06-29]
host version build time:[20:39:35]
Communication interface adding SERIAL_INTERFACE
Use the serial port[/dev/ttyUSB0]
Please enter the administrator permission login password:
serial open success! serial port:/dev/ttyUSB0, baud:921600
Scheduler Num 1 Start. Task Num = 1. Period = 50000
Scheduler Num 0 Start. Task Num = 1. Period = 100000
Scheduler Num 2 Start. Task Num = 1. Period = 20000
当前测试RMP版本: 1.0.0

central board test started.....
get_chassis_central_version: 0x1000
get_chassis_Err_Status: 0x00000000
get_chassis_central_version: 0x1000
```

图 14

## 附录一 系统参数及模式切换逻辑

表 1 系统参数

结构参数	尺寸	长*宽*高 (mm) 672*617*274
	结构参数	轴距*轮距*离地间隙 (mm) : 456*545*58 (底盘压缩到最低点)
	轮胎尺寸	8.5 英寸 (216mm) 轮毂电机
	自重	28kg
	标称载重	28kg
	越障	5cm/10°斜坡/减速带
	悬挂	独立悬挂
	防护等级	IPX5
性能参数	最大速度	3.56m/s
	最大转向速度	2rad/s
	最小转弯半径	1.36m
	刹车距离	满载 3.56m/s 约 1m
	控制方式	遥控控制、上位机控制
	刹车方式	电刹、机械刹车
通信	通信接口	UART, CAN
	API	ROS1、ROS2
	反馈数据	磁编、IMU
电池	续航	551Wh 满载 2m/s 续航约 40km
	电池	36V 15.3Ah
	充电方式	手动线充/快速换电池
交互	按键	急停按键、开关机按键
	状态指示	开关机状态指示灯, 底盘状态指示灯, 控制源指示, 电量显示, 充电状态显示



表 2 模式切换逻辑

底盘模式	输入	执行	退出
锁车模式	<ol style="list-style-type: none"> <li>1. 底盘上电的默认模式</li> <li>2. 急停恢复后默认模式</li> <li>3. 控车模式下, 通信超时、通信断链等可恢复性异常发生后进入的模式</li> <li>4. 上位机手动强制清除错误后进入该模式</li> </ol>	不带抱闸版本: 0 速闭环, 屏蔽速度指令, 状态指示灯黄色常亮; 带抱闸版本: 屏蔽速度指令, 状态指示灯黄色常亮, 断电抱闸失能, 抱闸抱死。	<ol style="list-style-type: none"> <li>1. 检测到不可恢复异常 errorcode, 进入错误模式</li> <li>2. 接收到 enable 指令, 进入控车模式</li> <li>3. 急停按键触发, 进入急停模式</li> </ol>
控车模式	<ol style="list-style-type: none"> <li>1. 锁车模式下, 接收到 enable 指令</li> </ol>	闭环, 接受控制指令。遥控器控车-指示灯绿色常闪; 上位机控车-指示灯绿色常亮。(带断电抱闸版本此时抱闸通电打开抱闸)	<ol style="list-style-type: none"> <li>1. 检测到不可恢复异常 errorcode, 进入错误模式</li> <li>2. 检测到通信超时、通信断链等可恢复性异常发生后, 进入锁车模式</li> <li>3. 急停按键触发, 进入急停模式</li> </ol>
急停模式	非异常模式下, 急停按键触发	泄力, 屏蔽速度及使能指令, 状态指示灯红色常闪。(带断电抱闸版本此时失能, 抱闸抱死)	急停按键弹起, 进入锁车模式
错误模式	检测到不可恢复异常 errorcode	刹车, 泄力, 屏蔽速度及使能指令。指示灯红色常亮。(带断电抱闸版本此时失能, 抱闸抱死)	<ol style="list-style-type: none"> <li>1. 重启</li> <li>2. 上位机手动强制清除错误。(需非常慎重实施手动强制清除错误)</li> </ol>

## 附录二 连接器焊接说明

### 一、准备工作

#### 1、工具

电烙铁、焊锡丝

#### 2、材料

8pin 连接器、2pin 连接器、AWG16 的线缆 2 根、AWG26 的线缆 8 根,如图 1 所示。

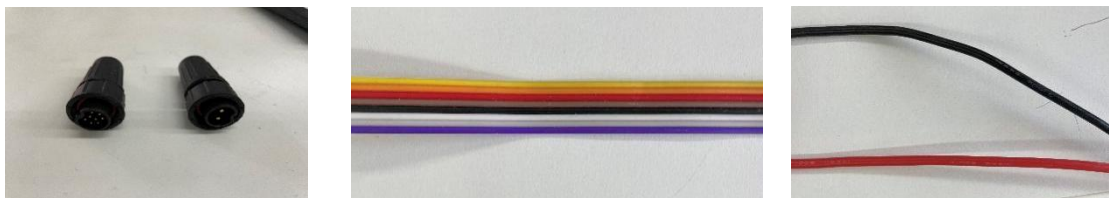


图 1

### 二、 焊接说明（以 8pin 连接器为例）

- 如图 2 所示为客户收到的 8pin 连接器，从红色箭头所示位置拧动连接器，使其拆成图 3 所示状态；



图 2



图 3

- 取出图 4 所示部件，该部件为需要进行焊接的部件；



图 4

3. 如图 5 所示，从部件的一侧可以看到连接器的 pin 角序号，之后将部件旋转 180°，就是需要进行焊接的部位；



图 5

4. 使用 AWG26 的线按照焊接说明书中的 pin 角的定义（详见附录三）进行焊接，焊接完成后，如图 6 所示；

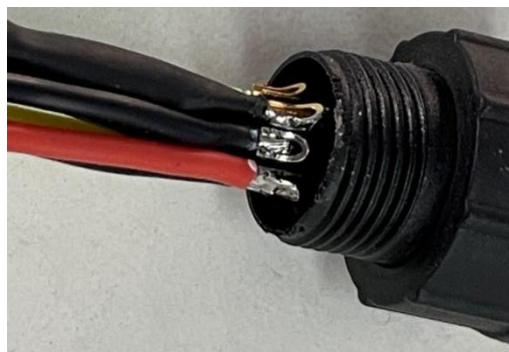


图 6

5. 取出图 7 所示的两个部件，套在焊接好的部件上，如图 8 所示；

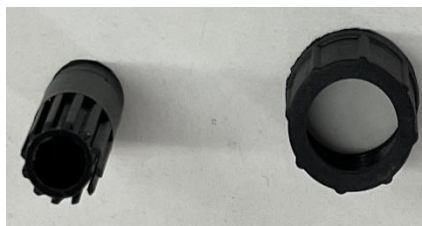


图 7



图 8

6. 取出图 9 所示的部件，套在之前组装好的部件上，拧紧，如图 10 所示；





图 9



图 10

7. 之后连接遥控器接收器以及串口即可，如图 11 所示；

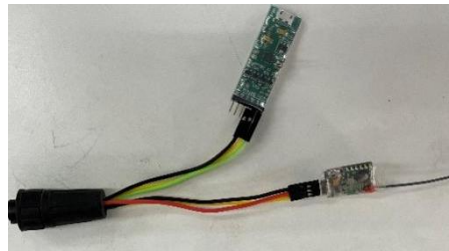


图 11

8. 2pin 连接器焊接方法与 8pin 连接器一致。

## 附录三 连接器 Pin 角定义说明

连接器	Pin 角序号	定义	线号	备注	颜色
8pin 航插	1	CANH	AWG26	CAN 口	红色
	2	CANL	AWG26		灰色
	3	TX	AWG26	串口	蓝色
	4	RX	AWG26		绿色
	5	GND	AWG26		白色
	6	5V	AWG26	遥控器接收器	棕色
	7	GND	AWG26		黑色
	8	S.B PPM	AWG26		黄色
2pin 航插	1	Power+	AWG16	上位机电源	红色
	2	Power-	AWG16		黑色

## 附录四 C/C++ API 参考文档

---

➤ `int Init_Comcore(void)`

Function: initialization of host computer dynamic link library

Parameter: none

Return value: 0: initialization succeed;

Other: initialization fail

➤ `void exit_Comcore(void)`

Function: exit initialization of host computer dynamic link library

Parameter: none

Return value: none

➤ `void aprctrl_datastamped_jni_register(saprctrl_datastamped* f)`

Function: registration via the callback function provided by parameter,  
and this callback function conducts the sensor data processing.

Parameter: f is a struct pointer, and this struct includes the unique function pointer member variables.

Return value: none

➤ `void aprctrl_eventcallback_jni_register(saprctrl_eventt* f)`

Function: registration via the callback function provided by parameter, and this callback function conducts the processing of event code.

Parameter: f is a struct pointer, and this struct includes the unique function pointer member variables.

Return value: none

➤ `uint16_t get_err_state(boardname_e boardname)`

Function: acquire the software/firmware runtime error code

Parameter: board name refers to the software/firmware ID

Parameter is one of the following values:

Host   upper computer node ID

Motor0   front\_left motor node   ID

Motor1 front\_right motor node ID

Motor2 rear\_left motor node ID

Motor3 rear\_right motor node ID

Central central\_board node ID

BMS batter ID

Return value: error code

➤ `int16_t get_bat_soc(void)`

Function: acquire percentage of battery remaining capacity

Parameter: none

Return value: percentage of battery remaining capacity (1~100)

➤ `int16_t get_bat_charging(void)`

Function: inquire whether the battery is in charging state

Parameter: none

Return value: 0: not in charging state

1: in charging state

➤ `int16_t get_bat_mvolt(void)`

Function: acquire real-time voltage of battery

Parameter: none

Return value: voltage value, unit mV

➤ `int16_t get_bat_mcurrent(void)`

Function: acquire real-time current of battery

Parameter: none

Return value: current value, unit mA

➤ `int16_t get_bat_temp(void)`

Function: acquire battery temperature

Parameter: none

Return value: temperature value, unit degree Celsius

➤ `int16_t get_chassis_work_model(void)`

Function: acquire working state of chassis motor

Parameter: none

Return value: 1: motor in augmentation;

0: motor not in augmentation

➤ `int16_t get_chassis_load_state(void)`

Function: acquire setting value of chassis based on controlling parameter of different loading

Parameter: none

Return value: 0: no-load control parameter;

1: full load controlling parameter

➤ `int16_t get_chassis_mode(void)`

Function: acquire working mode of chassis finite state machine (FSM)

Parameter: none

Return value: 0 locking mode;

1 vehicle control mode;

2 pushing mode;

3 emergency stop mode;

4 error mode

➤ `int16_t get_ctrl_cmd_src(void)`

Function: acquire command origin of motor chassis control

Parameter: none

Return value: 0: control vehicle with remote controller;

1: control vehicle with host computer

➤ `int16_t get_vehicle_meter(void)`

Function: acquire the mileage since the chassis is power up

Parameter: none

Return value: mileage value, unit meter



➤ `uint16_t get_host_version(void)`

Function: acquire the host computer software version

Parameter: none

Return value: host computer software version number

➤ `uint16_t get_chassis_central_version (void)`

Function: acquire the central board firmware version

Parameter: none

Return value: the central board firmware version number

➤ `uint16_t get_chassis_motor_version (void)`

Function: acquire the motor board firmware version

Parameter: none

Return value: the motor board firmware version number

➤ `int16_t get_line_forward_max_vel_fb (void)`

Function: acquire the forward speed limiting feedback value of the chassis

Parameter: None

Return value: the forward speed limiting feedback value of the chassis

➤ `int16_t get_line_backward_max_vel_fb (void)`

Function: acquire the backward speed limiting feedback value of the chassis

Parameter: None

Return value: the backward speed limiting feedback value of the chassis

➤ `int16_t get_angular_max_vel_fb (void)`

Function: acquire the angular speed limiting feedback value of the chassis

Parameter: None

Return value: the angular speed limiting feedback value of the chassis

➤ `int16_t getlapProgress (void)`

Function: Get the progress of IAP upgrades

Parameter: None

Return value:

-1: IAP upgrade failed

0: IAP upgrades are idle or started or interrupted

100: IAP upgrade completed

Other: Percentage of IAP upgrade progress

➤ `void iapCentralBoard (void)`

Function: IAP upgrade of the central board firmware of the chassis

Parameter: None

Return value: none

Note: You need to place the central board firmware "central.bin" in the path of "/sdcard/firmware/" in advance.

➤ `void iapMotorBoard (motor_index_e motor_index )`

Function: IAP upgrade of the motor board firmware of the chassis

Parameter: Parameter is one of the following enumerated values:

Motor\_front: Represents the front wheel circuit board

Motor\_rear: Represents the rear wheel circuit board

Return value: none

Note: You need to place the motor board firmware "motor\_front.bin" or "motor\_rear.bin" in the path of "/sdcard/firmware/" in advance.

➤ `void iapBrakeBoard (void )`

Function: IAP upgrade of the brake board firmware of the chassis

Parameter: none

Return value: none

Note: You need to place the brake board firmware "brake.bin" in the path of "/sdcard/firmware/" in advance.

➤ `bool isHostlapOver (void)`

Function: Query if the IAP upgrade process has ended

Parameter: None

Return value: true: the IAP completes or fails or is interrupted

False: IAP not started or in progress

➤ `Int16_t getHostlapResult (void)`

Function: acquire the reason for the end of IAP

Parameter: None

Return value: 3: IAP completes

4: IAP fails

5: IAP is interrupted

Others: IAP not started or in progress

➤ `Int16_t getHostlapErrorCode (void)`

Function: Gets the error code for IAP failure

Parameter: None

Return value: the error code for IAP failure

➤ `int16_t get_calibrate_mid_value_status(void)`

Function: Query if the median headway Angle has been calibrated

Parameter: None

Return value: 1: calibrated;

0: no calibrated.

➤ `uint8_t get_chassis_central_SN(char * SN)`

Function: Get the SN code of MCU of chassis central board. return:

1:success, 0:fail]

Parameter: SN array size>25 bytes;

Return value: 1: Success in obtaining results;

0: Obtaining results failed.

➤ `uint8_t get_rotate_switch_stat(void)`

Function: Query the status of the in-situ rotation switch on che chassis.

return: 1:support , 0: not support]

Parameter: none

Return value: 1: support in situ rotation function;

0: donot support in situ rotation function now.

➤ `void set_cmd_vel(double linearx, double angularz)`

Function: set up the command value of chassis target speed, which needs to be regular transmit once the chassis is enabled. It will be determined as communication failure if the chassis can't receive the command value in continuous 150ms in controlling mode.

Parameter: linear\_x: linear velocity command value, unit m/s;

angular\_z: angular velocity command value, unit rad/s

Return value: none

➤ `void set_line_forward_max_vel(double linearforwardmax_x)`

Function: set up the max forward linear velocity value of chassis.

Parameter: `linearforwardmax_x`: max forward linear velocity value of chassis, unit m/s, range 0 - 2.3

Return value: none

➤ `void set_line_backward_max_vel(double linearbackwardmax_x)`

Function: set up the max backward linear velocity value of chassis.

Parameter: `linearbackwardmax_x`: max backward linear velocity value of chassis, unit m/s, range -0.85 - 0

Return value: none

➤ `void set_angular_max_vel(double angularmax_z)`

Function: set up the max angular velocity command value of chassis.

Parameter: `angularmaxz`: the max angular velocity command value, unit rad/s, range 0 - 2

Return value: none

➤ `void set_enable_ctrl(uint16_t enableflag)`



Function: set up to enable the chassis to control the vehicle.

Parameter: enable\_flag:

1 enable the vehicle control;

0 exit the vehicle control

Return value: none

➤ void set\_smart\_car\_serial(*const char \* serialno*)

Function: set up the terminal name of serial port of host computer, e.g. ttyUSB0.

Parameter: serial\_no: terminal name of serial port, under the path /dev/  
by default, e.g. “ttyUSB0”

Return value: none

➤ void set\_comu\_interface (*comu\_choice\_e comu\_choice*)

Function: Set up the communication interface between the host computer and the chassis, including serial communication and CAN communication

Parameter: *comu\_choice*:

‘comu\_serial’ Use a serial port for communication

‘comu\_can’ Use a CAN port for communication

Return value: none

➤ void set\_chassis\_load\_state(int16\_t newLoadSet)

Function: set up the parameter of chassis control based on the different chassis load.

Parameter: newLoadSet:

0: no-load parameter;

1: full load parameter

Return value: none

➤ void set\_chassis\_poweroff (void)

Function: chassis power off controlled by host computer.

Parameter: none

Return value: none

➤ void setHostlapCanceled (void)

Function: Interrupt the IAP upgrade process.

Parameter: none

Return value: none

➤ uint8\_t set\_calibrate\_mid\_value(void)

Function: Sets the command to calibrate the median Angle of the front wheel of the chassis.

Parameter : none.

Return value: 0: successfully set; other: setup failed.

➤ uint8\_t reset\_host\_power\_time\_s(uint16\_t reset\_time\_s);

Function: Set the reset time after power failure of the upper machine

Parameter : reset\_time\_s: reset interval time.

Range: 0~65535. unit: second.

Return value: 0: successfully; other: failed.

➤ `uint8_t clear_chassis_error_code(void);`

Function: Clear the error code of the chassis, excluding warnings and exceptions

Parameter : none

Return value: 0: successfully; other: failed.

➤ `Int16_t enable_rotate_switch(uint8_t enable);`

Function: Enable or disable the chassis to rotate in place switch,

Parameter : 1: enable ; 0:disable.

Return value:

0: successfully;

other: countdown time (seconds)