

ASTRALL

ASTRALL

璇玑动力

## 璇玑·屹励 T 01 SDK 软件接口手册(Beta)

V1.0.6 2026.01.30



# 版本说明

| 版本号   | 修改时间     | 修改内容                                                                                                                                                                                                                                                              |
|-------|----------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1.0.0 | 20251017 | 首次发布                                                                                                                                                                                                                                                              |
| 1.0.1 | 20251101 | 1、修改了 SDK 使用设备的 IP 地址；<br>2、增加了轮速数据的订阅。                                                                                                                                                                                                                           |
| 1.0.2 | 20251110 | 1、系统状态中将原错误码分离错误码和警告码；<br>2、系统状态中的 version 和 serialNumber 移除，新增 AstrallGetDeviceInfo 获取设备信息接口；<br>3、AstrallSdkInit 传参中添加 SDK 状态回调接口；<br>4、接口修改为阻塞式，直至响应或超时；<br>5、修改控制权机制，详见关于控制权章节；<br>6、未连接时调用 AstrallGetSportStatus 接口时 将返回未知 (ASTRALL_SPORT_STATUS_UNKNOWN)状态； |
| 1.0.3 | 20251113 | 1、增加了订阅头部相机 RGB 数据的接口；                                                                                                                                                                                                                                            |
| 1.0.4 | 20251114 | 1、增加向遥控器消息窗口推送消息的接口 AstrallSendMessage；<br>2、增加订阅摇杆数据；                                                                                                                                                                                                            |
| 1.0.5 | 20251125 | 1、修改了控制权的获取逻辑；<br>2、增加了自动充电任务的相关接口和状态；<br>3、增加了站立中和趴下中两个状态；<br>4、增加了当速度指令不下发时，自动停止，现在同时支持发 0 停止和自动停止；<br>5、在 imu 的数据结构中加入了里程计数据 x，y；<br>6、增加了机动模式切换的相关描述；                                                                                                         |
| 1.0.6 | 20260129 | 修改排版及布局                                                                                                                                                                                                                                                           |
|       |          |                                                                                                                                                                                                                                                                   |

※最终解释权归深圳璇玑动力科技有限公司所有

## 目录

|       |                |    |
|-------|----------------|----|
| 1     | SDK 概述         | 6  |
| 2     | 调试说明           | 8  |
| 2.1   | 协议总则           | 8  |
| 2.2   | IP 地址及端口号说明    | 8  |
| 2.3   | 交互机制           | 8  |
| 3     | 硬件参数           | 9  |
| 3.1   | 机器人本体坐标系与关节坐标系 | 9  |
| 3.2   | 机器人身体参数        | 10 |
| 3.3   | 机器人腿部参数        | 10 |
| 3.4   | 关节参数           | 11 |
| 3.5   | 传感器外参          | 11 |
| 4     | Daedalus 获取    | 12 |
| 5     | 软件服务接口         | 12 |
| 5.1   | 函数调用返回参数表      | 12 |
| 5.2   | 初始化            | 12 |
| 5.2.1 | 函数入参           | 13 |
| 5.2.2 | 函数返回值          | 13 |
| 5.2.3 | 详细说明           | 13 |
| 5.3   | 反初始化           | 13 |
| 5.3.1 | 函数入参           | 14 |
| 5.3.2 | 函数返回值          | 14 |
| 5.3.3 | 详细说明           | 14 |
| 5.4   | 心跳接口           | 14 |
| 5.4.1 | 函数入参           | 14 |
| 5.4.2 | 函数返回值          | 14 |
| 5.4.3 | 详细说明           | 14 |
| 5.5   | 设备信息接口         | 14 |
| 5.5.1 | 函数入参           | 15 |

|        |                |    |
|--------|----------------|----|
| 5.5.2  | 函数返回值          | 15 |
| 5.5.3  | 详细说明           | 15 |
| 5.6    | 系统信息接口         | 15 |
| 5.6.1  | 函数入参           | 15 |
| 5.6.2  | 函数返回值          | 17 |
| 5.6.3  | 详细说明           | 17 |
| 5.7    | 电池信息接口         | 17 |
| 5.7.1  | 函数入参           | 18 |
| 5.7.2  | 函数返回值          | 18 |
| 5.7.3  | 详细说明           | 18 |
| 5.8    | 当前模式信息接口       | 18 |
| 5.8.1  | 函数入参           | 18 |
| 5.8.2  | 函数返回值          | 19 |
| 5.8.3  | 详细说明           | 19 |
| 5.9    | 传感器信息订阅接口      | 20 |
| 5.9.1  | 函数入参           | 20 |
| 5.9.2  | 函数返回值          | 20 |
| 5.9.3  | 详细说明           | 21 |
| 5.10   | 向遥控器消息窗口发送消息接口 | 22 |
| 5.10.1 | 函数入参           | 22 |
| 5.10.2 | 函数返回值          | 23 |
| 5.10.3 | 详细说明           | 23 |
| 5.11   | 运动模式切换接口       | 23 |
| 5.11.1 | 函数入参           | 23 |
| 5.11.2 | 函数返回值          | 23 |
| 5.11.3 | 详细说明           | 23 |
| 5.12   | 控制权申请接口        | 24 |
| 5.12.1 | 函数入参           | 24 |
| 5.12.2 | 函数返回值          | 24 |
| 5.12.3 | 详细说明           | 24 |

|        |                |    |
|--------|----------------|----|
| 5.13   | 移动接口 .....     | 25 |
| 5.13.1 | 函数入参 .....     | 25 |
| 5.13.2 | 函数返回值 .....    | 26 |
| 5.13.3 | 详细说明 .....     | 26 |
| 5.14   | 头部灯光控制接口 ..... | 26 |
| 5.14.1 | 函数入参 .....     | 26 |
| 5.14.2 | 函数返回值 .....    | 26 |
| 5.14.3 | 详细说明 .....     | 26 |

## 1 SDK 概述

璇玑·屹励 T 01 软件开发工具包 Daedalus-SDK（下文统称为 Daedalus）提供了与机器人本体运动控制模块的交互接口，开发者可以通过 Daedalus 控制机器人完成运动、姿态切换以及获取机器人的状态信息，Daedalus 通过 UDP 与机器人进行通讯，支持多平台使用。

Daedalus 主要功能包括：

- 1) 运动控制：前进、后退、左转、右转、左平移、右平移；
- 2) 姿态控制：锁定关节站立、锁定关节趴下、软急停、全地形行走、自主充电等；
- 3) 状态获取：轮速，电池信息，当前运动模式，本体状态信息等；
- 4) 图传接口：UDP、H264 格式的 RGB 视频流、深度伪彩色视频流或 IR 灰度视频流（其中 Depth 和 IR 根据设备配置实现，非标配）。

在使用 Daedalus 与机器人进行通讯时，请确保机器人处于阻尼模式或锁定关节趴下状态（见下图 1-1）。

当 Daedalus 与机器人建立连接后，机器人会优先执行 Daedalus 下发的控制指令，当 Daedalus 与机器人本体的通讯间隔超过一定时间后，机器人会判定与 Daedalus 断开连接，清空 Daedalus 之前下发的指令，并在一段时间后进入阻尼模式。

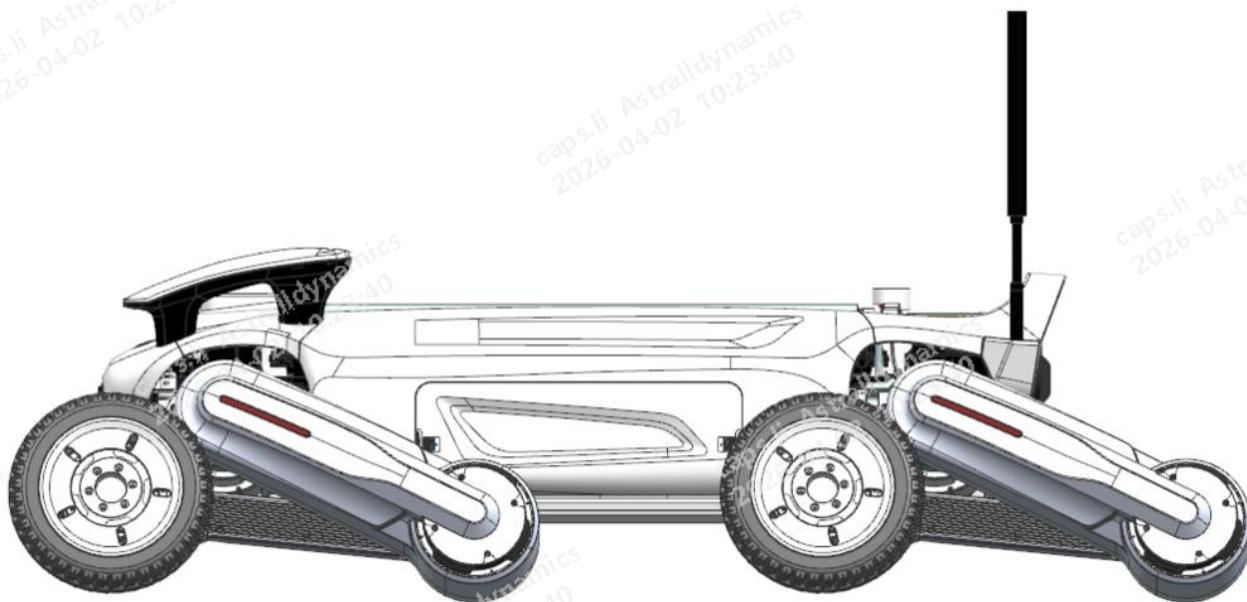


图 1-1 阻尼模式/锁定关节趴下姿态

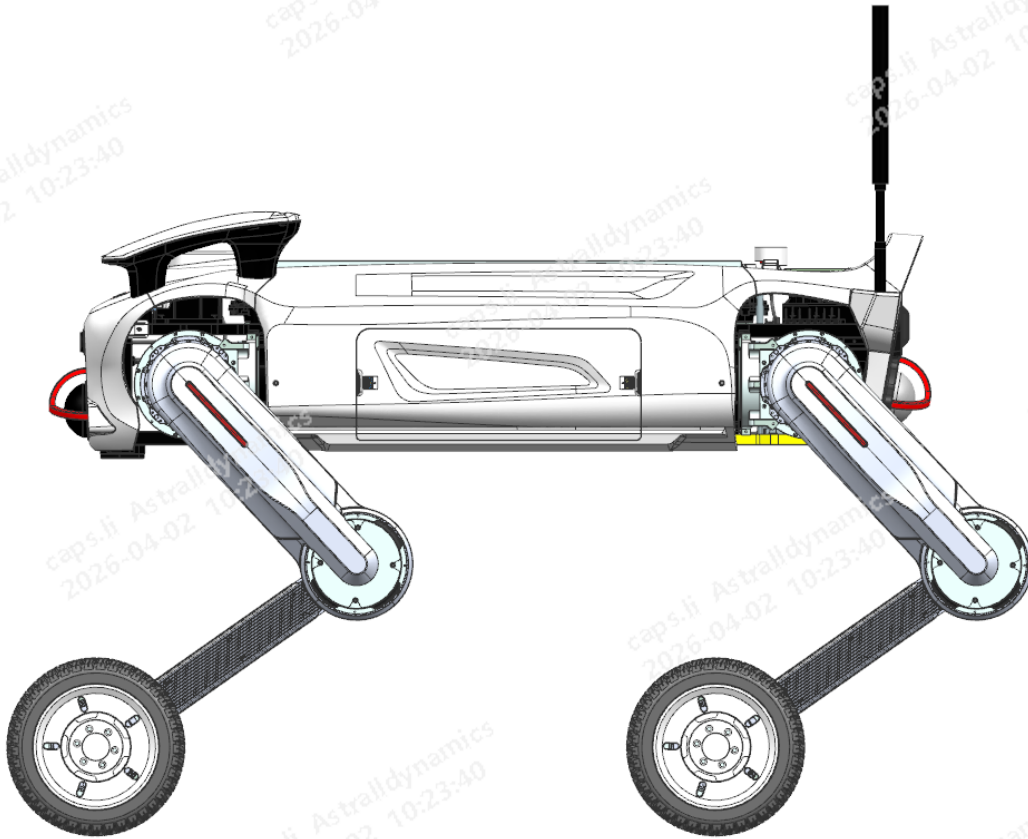


图 1-2 锁定关节站立姿态

注意：机器人趴下姿态整体尺寸为  $1.13\text{m} \times 0.755\text{m} \times 0.32\text{m}$ （长宽高）

机器人站立姿态整体尺寸为  $1.11\text{m} \times 0.61\text{m} \times 0.745\text{m}$ （长宽高）

## 2 调试说明

### 2.1 协议总则

Daedalus 采用 UDP 通讯方式，适用于与机器人本体通讯，可采用外部板卡、上位机或系统使用 Daedalus 进行二次开发。

### 2.2 IP 地址及端口号说明

机器人本体为服务端，默认 IP 地址为：10.18.0.100

调用 Daedalus 的设备 IP 地址应为：10.18.0.200。

调用 Daedalus 的设备端口号：3600(请提前确认该端口没有被占用)

头部雷达 IP：10.18.0.120 MSOP 端口：6699 DIFOP 端口：7788 IMU 获取端口：6688

尾部雷达 IP：10.18.0.121 MSOP 端口：6969 DIFOP 端口：7878 IMU 获取端口：6868

### 2.3 交互机制

Daedalus 与机器人本体交互，主要使用请求/应答和订阅/推送两种交互机制，Daedalus 可以主动发起数据或状态查询请求，机器人本体通过应答回复部分请求，也可以订阅某些传感器数据，机器人本体按照订阅周期进行推送数据。

### 3 硬件参数

#### 3.1 机器人本体坐标系与关节坐标系

机器人机身坐标系（本体系）：X 轴朝向头部，Y 轴朝向机身左侧，Z 轴沿 XxY 朝向机器人上方。

各个关节坐标系定义：各个关节的坐标系与本体系定义一致，其中，机身关机的旋转轴为 x 轴，大腿关节(髌关节)、小腿关节(膝关节)和轮的旋转轴为 y 轴，旋转正方向符合右手定则。

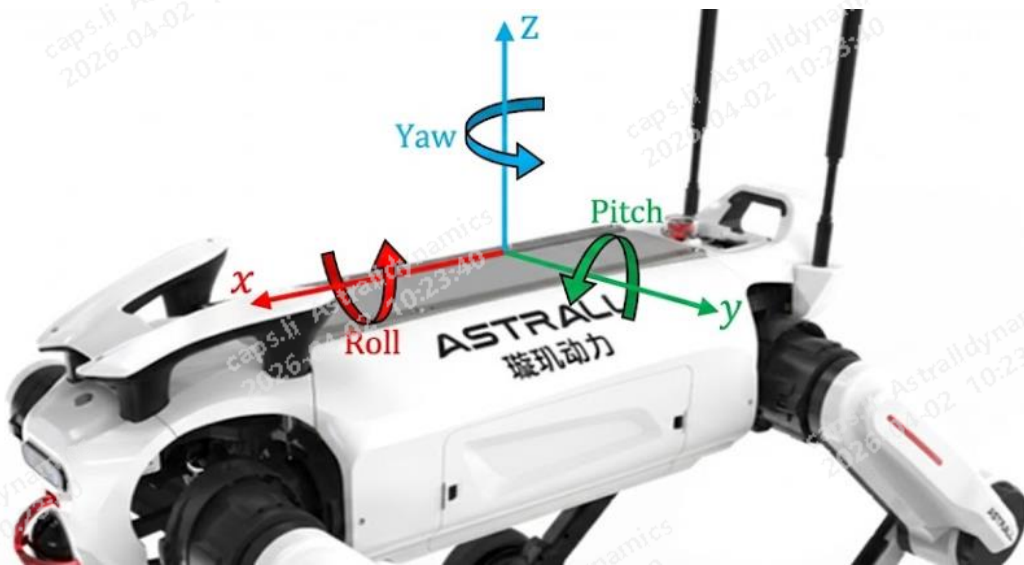


图 3-1 机器人本体坐标系

### 3.2 机器人身体参数

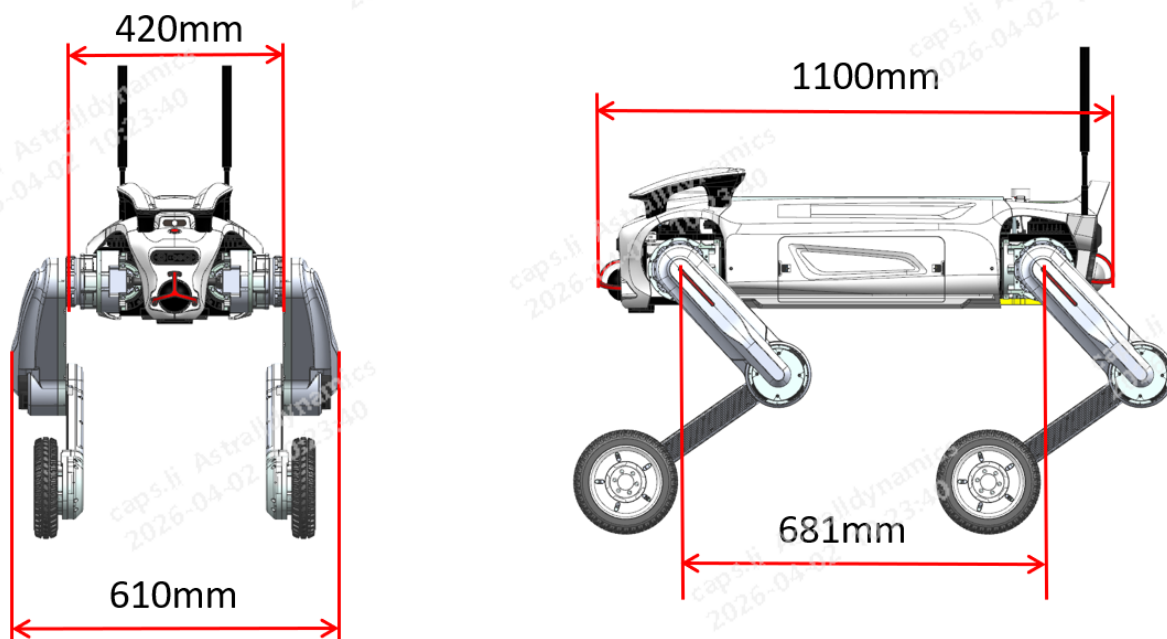


图 3-2 机器人本体参数

表 3-2 机器人本体参数

| 参数      | 数值     | 说明       |
|---------|--------|----------|
| 长度      | 1.10m  | 身体总长度    |
| 腿平面左右间距 | 0.681m | 腿平面左右的距离 |
| 宽度      | 0.61m  | 身体总宽度    |

### 3.3 机器人腿部参数

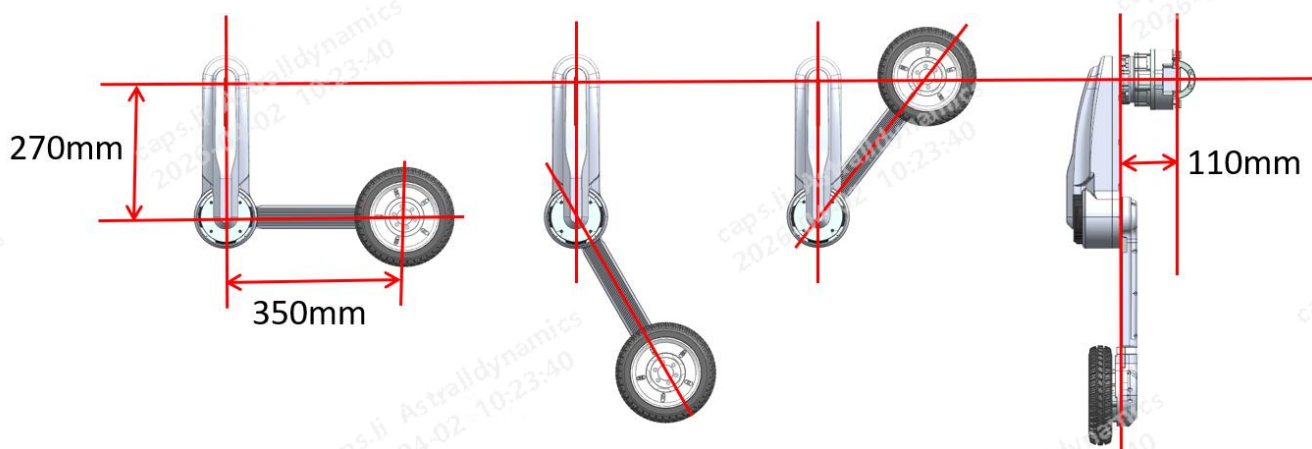


图 3-3 机器人腿部参数

表 3-3 机器人腿部参数

| 参数          | 数值      | 说明              |
|-------------|---------|-----------------|
| 腿平面与髋侧摆关节距离 | 0.11m   | 髋侧摆关节与腿平面距离     |
| 大腿长度        | 0.27m   | 髋前摆关节中心与膝关节中心距离 |
| 小腿长度        | 0.35m   | 膝关节中心与足底圆心距离    |
| 轮半径         | 0.0925m | 轮半径             |

### 3.4 关节参数

表 3-4 机器人关节参数

| 关节名称        | 运动范围       | 额定扭矩  | 额定转速       | 峰值扭矩   |
|-------------|------------|-------|------------|--------|
| 髋关节(Hip)    | -15°~15°   | 50N·m | 100rpm±10% | 300N·m |
| 大腿关节(Thigh) | -170°~170° |       |            |        |
| 小腿关节(Calf)  | -148°~148° |       |            |        |
| 轮(Wheel)    | ~          |       |            |        |

### 3.5 传感器外参

传感器坐标相对于质心坐标在机体系(4.1 节中定义)下的位置，质心为原点，坐标 (0, 0, 0)。

表 3-5 传感器相对机器人质心坐标

| 传感器名称 | 坐标           |
|-------|--------------|
| 前激光雷达 | 444, 0, -32  |
| 后激光雷达 | -420, 0, 19  |
| 双目相机  | 447, 0, 41   |
| IMU   | 127, -85, 82 |
| GPS   | 331, 0, 114  |

注意：其他动力学参数可以在提供的 URDF 文件中获取。

## 4 Daedalus 获取

请联系销售人员获取。

## 5 软件服务接口

Daedalus 提供订阅/推送、请求/应答两种模式。

- 订阅/推送：接收方通过 Daedalus 订阅某个消息，机器人通过订阅内容向接收方推送消息，用于中高频数据(如 IMU)或持续的数据交互；

- 请求/应答：问答模式，通过请求实现模型切换、姿态控制等操作，用于低频或功能切换时的数据交互。

Daedalus 对大部分功能进行了优化封装，实现函数式调用。

### 5.1 函数调用返回参数表

表 5-1 函数调用返回参数表

| 变量原型                      | 数值(Hex) | 含义说明           |
|---------------------------|---------|----------------|
| ASTRALL_RES_FAILED        | 0x8004  | 执行失败/申请失败/订阅失败 |
| ASTRALL_RES_TIMEOUT       | 0x8005  | 超时             |
| ASTRALL_RES_RUNNING       | 0x8006  | 执行中            |
| ASTRALL_RES_SUCCEEDED     | 0x8008  | 执行成功           |
| ASTRALL_RES_INVALID_PARAM | 0x8010  | 无效参数           |
| ASTRALL_RES_RC_NORELEASE  | 0x8020  | 遥控器抢夺未释放控制权    |
| ASTRALL_RES_BEEN_OBTAINED | 0x8021  | 已经有设备获得控制权未释放  |
| ASTRALL_RES_WITHOUT_AUTH  | 0x8022  | 没有控制权          |

### 5.2 初始化

|      |                                                               |
|------|---------------------------------------------------------------|
| 函数名  | AstrallSdkInit                                                |
| 函数原型 | uint16_t AstrallSdkInit(AstrallConfig &cfg, uint32_t timeout) |
| 功能描述 | 阻塞式调用，直到建立连接或超时                                               |

### 5.2.1 函数入参

输入参数 1: AstrallConfig &cfg:

```
1. struct AstrallConfig
2. {
3.     // 心跳回调函数，收到返回的心跳ACK 后执行此函数。
4.     // data: 无
5.     std::function<void(void *data, uint16_t len)> heartbeatCb;
6.     // Daedalus 状态回调函数
7.     // data: 见下 AstrallSdkStatus 结构体数据
8.     std::function<void(void *data, uint16_t len)> sdkStatusCb;
9. };
10.
11. struct AstrallSdkStatus
12. {
13.     uint8_t link : 1;           // 连接状态
14.     uint8_t ctrlAuthority : 1; // 控制权状态
15.     uint8_t reserve : 6;
16.
17.     AstrallSdkStatus()
18.     {
19.         link = 0;
20.         ctrlAuthority = 0;
21.         reserve = 0;
22.     }
23. };
```

输入参数 2: uint32\_t timeout: 超时时间 (ms)

### 5.2.2 函数返回值

详见表 5-1 函数调用返回参数表。

### 5.2.3 详细说明

无。

## 5.3 反初始化

|      |                         |
|------|-------------------------|
| 函数名  | AstrallSdkDeinit        |
| 函数原型 | void AstrallSdkDeinit() |
| 功能描述 | 断开连接，清理释放资源             |

### 5.3.1 函数入参

无。

### 5.3.2 函数返回值

详见表 5-1 函数调用返回参数表。

### 5.3.3 详细说明

无。

## 5.4 心跳接口

|      |                                             |
|------|---------------------------------------------|
| 函数名  | AstrallHeartbeat                            |
| 函数原型 | uint16_t AstrallHeartbeat(uint32_t timeout) |
| 功能描述 | 发送心跳包                                       |

### 5.4.1 函数入参

输入参数：uint32\_t timeout：超时时间（ms）

### 5.4.2 函数返回值

详见表 5-1 函数调用返回参数表。

### 5.4.3 详细说明

心跳接收在初始化时的回调中，推荐心跳频率为 10Hz，若 500ms 内机器人本体未收到心跳包，则清除 Daedalus 下发的指令，保持锁定站立姿态；若再接下来的 500ms 内仍未收到心跳包，机器人本体判断 Daedalus 断开连接，清除 Daedalus 连接标识，清除 Daedalus 控制权，保持站立姿态，等待其他控制器指令，若无其他控制器(如遥控器未连接)则机器人自主趴下。

## 5.5 设备信息接口

|      |                                                                          |
|------|--------------------------------------------------------------------------|
| 函数名  | AstrallGetDeviceInfo                                                     |
| 函数原型 | uint16_t AstrallGetDeviceInfo(AstrallDeviceInfo& info, uint32_t timeout) |
| 功能描述 | 获取设备信息                                                                   |

### 5.5.1 函数入参

输入参数 1: AstrallDeviceInfo &info:

```
1. struct AstrallDeviceInfo
2. {
3.     std::string version;
4.     std::string serialNumber;
5.     uint32_t model;
6.
7.     AstrallDeviceInfo()
8.     {
9.         version = "";
10.        serialNumber = "";
11.        model = 0;
12.    }
13.};
```

输入参数 2: uint32\_t timeout: 超时时间 (ms)

### 5.5.2 函数返回值

详见表 5-1 函数调用返回参数表。

### 5.5.3 详细说明

无。

## 5.6 系统信息接口

|      |                                                              |
|------|--------------------------------------------------------------|
| 函数名  | AstrallGetSystemStatus                                       |
| 函数原型 | uint16_t AstrallGetSystemStatus(AstrallSystemStatus& status) |
| 功能描述 | 获取系统状态参数                                                     |

### 5.6.1 函数入参

输入参数 1: AstrallSystemStatus &status:

```
1. enum AstrallSystemCode : uint8_t
2. {
3.     ASTRALL_SYSTEM_CODE_INIT = 0, // Initialization & System Self-Check
4.     ASTRALL_SYSTEM_CODE_STANDBY = 1, // Standby -> Damping Mode
5.     ASTRALL_SYSTEM_CODE_RUNNING = 2, // Normal Operation
6.     ASTRALL_SYSTEM_CODE_WARNING = 3, // Warning
```

```

7.     ASTRALL_SYSTEM_CODE_ERROR = 4,    // Error -> Emergency Shutdown
8.     ASTRALL_SYSTEM_CODE_SHUTDOWN = 5    // Shut Down
9. };
10.
11. enum AstrallErrorCode : uint32_t
12. {
13.     ASTRALL_ERR_NONE = 0,
14.     ASTRALL_ERR_MOTOR_VERSION = (1U << 0), // Motor driver version error
15.     ASTRALL_ERR_MOTOR_COMM_LOSE = (1U << 1), // Motor communication lost
16.     ASTRALL_ERR_MOTOR_STATUS = (1U << 2), // Motor abnormal.
17.     ASTRALL_ERR_MOTOR_OVER_TEMP = (1U << 3), // Motor over temperature
18.     ASTRALL_ERR_STO_TRIGGERED = (1U << 4), // STO triggered
19.     ASTRALL_ERR_IMU_ERROR = (1U << 5), // IMU error
20.     ASTRALL_ERR_LOW_BATTERY_LEVEL = (1U << 6), // Battery level too low
21.     ASTRALL_ERR_OVER_LOAD = (1U << 7), // Motor overload
22.     ASTRALL_ERR_MBUS_DRIVER_FAIL = (1u << 8), // Driver version error
23. };
24.
25. enum AstrallWarnCode : uint32_t
26. {
27.     ASTRALL_WARN_NONE = 0,
28.     ASTRALL_WARN_MOTOR_OVER_TEMP = (1U << 0), // Motor about to overheat
29.     ASTRALL_WARN_JOYSTICK_LOCK = (1U << 1), // RC is locked.
30.     ASTRALL_INFO_JOYSTICK_INTERVENTION = (1U << 2), // RC has obtained control
31.     ASTRALL_WARN_MOTOR_COMM_UNSTABLE = (1U << 3), // Motor comm unstable
32.     ASTRALL_WARN_LOW_BATTERY_LEVEL = (1U << 4), // Low battery level warning
33.     ASTRALL_WARN_JOYSTICK_LOSE = (1U << 5), // Remote controller offline
34.     ASTRALL_WARN_MOTOR_OVER_LOAD = (1U << 6), // Motor about to overload
35.     ASTRALL_WARNING_CONNECT_SDK_LOSE = (1u << 7), // Connected SDK is offline
36.     ASTRALL_WARNING_LOSE_CAMERA = (1u << 8), // No USB camera device
37. };
38.
39. struct AstrallSystemStatus
40. {
41.     AstrallSystemCode sysStatus;    // System status word
42.     AstrallErrorCode errorCode;    // Error Code
43.     AstrallWarnCode warnCode;    // Warn Code
44.
45.     AstrallSystemStatus()
46.     {
47.         sysStatus = AstrallSystemCode::ASTRALL_SYSTEM_CODE_INIT;
48.         errorCode = AstrallErrorCode::ASTRALL_ERR_NONE;
49.         warnCode = AstrallWarnCode::ASTRALL_WARN_NONE;
50.     }
51. };

```

注意：当有任何一个错误码出现，机器人整机立即趴下，进入被动控制模式，不响应任何控制指令。

输入参数 2: uint32\_t timeout: 超时时间 (ms)

### 5.6.2 函数返回值

详见表 5-1 函数调用返回参数表。

### 5.6.3 详细说明

Daedalus 初始化成功后，返回的 AstrallSdkStatus 状态中的 link 为连接状态时，返回数据才有效，否则数据无效。

AstrallErrorCode 为机器人错误码，AstrallWarnCode 为机器人警告码。

关于 AstrallWarnCode 的详细说明，见下表 5-4-3：

表 5-4-3 AstrallWarnCode 详细说明

| 变量原型                               | 含义        | 说明                                                     |
|------------------------------------|-----------|--------------------------------------------------------|
| ASTRALL_WARN_MOTOR_OVER_TEMP       | 电机即将过温    | 电机温度即将超过 85℃或驱动温度超过 70℃，继续大负载动作可能会触发电机高温保护（电机 120 摄氏度） |
| ASTRALL_WARN_JOYSTICK_LOCK         | 遥控器已锁定    | 不影响使用，将遥控器开关拨至三档即可解锁                                   |
| ASTRALL_INFO_JOYSTICK_INTERVENTION | 遥控器已获取控制权 | 如需要再次获取控制权，可以重新申请                                      |
| ASTRALL_WARN_MOTOR_COMM_UNSTABLE   | 电机通讯不稳定   | 不影响使用，电机丢失 2ms 数据且连续不超过 6ms                            |
| ASTRALL_WARN_LOW_BATTERY_LEVEL     | 电池电量不足警告  | 提示用户电量即将不足，不影响使用                                       |
| ASTRALL_WARN_JOYSTICK_LOSE         | 遥控器离线     | 不影响使用                                                  |
| ASTRALL_WARN_MOTOR_OVER_LOAD       | 电机即将过载    | 提示用户电机即将过载，继续进行大负重运动、上楼梯、爬坡等动作可能会造成电机过载。               |

## 5.7 电池信息接口

|      |                                                           |
|------|-----------------------------------------------------------|
| 函数名  | AstrallGetPowerStatus                                     |
| 函数原型 | uint16_t AstrallGetPowerStatus(AstrallPowerStatus& power) |
| 功能描述 | 获取电源参数                                                    |

### 5.7.1 函数入参

输入参数: AstrallPowerStatus &power:

```
1. struct AstrallPowerStatus
2. {
3.     float soc;                // Remaining battery power: %
4.     float temp;              // Battery temperature
5.     float voltage;           // Battery voltage
6.     uint16_t cycleCount;     // Number of charge cycles
7.     uint16_t charged;        // Chargestatus 0:unknown 1:discharged 2:stand 3:fast
8.
9.     AstrallPowerStatus()
10.    {
11.        soc = 0.0f;
12.        temp = 0.0f;
13.        voltage = 0.0f;
14.        cycleCount = 0;
15.        charged = 0;
16.    }
17.};
```

### 5.7.2 函数返回值

详见表 5-1 函数调用返回参数表。

### 5.7.3 详细说明

Daedalus 初始化成功后, 返回的 AstrallSdkStatus 状态中的 link 为连接状态时, 返回数据才有效, 否则数据无效。

## 5.8 当前模式信息接口

|      |                                  |
|------|----------------------------------|
| 函数名  | AstrallGetSportStatus            |
| 函数原型 | uint16_t AstrallGetSportStatus() |
| 功能描述 | 获取机器人本体当前运动状态                    |

### 5.8.1 函数入参

无。

### 5.8.2 函数返回值

函数返回参数：AstrallSportStatus:

```
1. enum AstrallSportStatus : uint16_t
2. {
3.
4.     ASTRALL_SPORT_STATUS_UNKNOWN = 0,           // 未知状态
5.     ASTRALL_SPORT_STATUS_INIT = 0xB100,         // 初始化中
6.     ASTRALL_SPORT_STATUS_DAMPING = 0xB101,       // 阻尼状态
7.     ASTRALL_SPORT_STATUS_FIXEDSTAND = 0xB102,    // 锁定站立状态
8.     ASTRALL_SPORT_STATUS_FIXEDDOWN = 0xB103,     // 锁定趴下状态
9.     ASTRALL_SPORT_STATUS_MOVE = 0xB104,          // 行走(RL 步态全地形覆盖) 中
10.
11.     ASTRALL_SPORT_STATUS_UPSTAIRS = 0xB105,      // 爬楼梯中
12.     ASTRALL_SPORT_STATUS_CAR = 0xB106,           // 负重小车模式
13.     ASTRALL_SPORT_STATUS_AUTOCHARGE = 0xB107,    // 准备执行自动充电
14.     ASTRALL_SPORT_STATUS_SEARCHCHARGING = 0xB108, // 寻找充电桩中
15.     ASTRALL_SPORT_STATUS_PLUGGINGCHARGING = 0xB109, // 已找到, 正在上桩
16.     ASTRALL_SPORT_STATUS_ATCHARGING = 0xB10A,    // 在充电桩上
17.     ASTRALL_SPORT_STATUS_EXITCHARGING = 0xB10B,  // 退出充电桩中
18.
19.     ASTRALL_SPORT_STATUS_CHARGING = 0xB10C,      // 充电中
20.
21.     ASTRALL_SPORT_STATUS_FIXEDSTANDING = 0xB10D, // 执行站立中
22.     ASTRALL_SPORT_STATUS_FIXEDDOWNING = 0xB10E, // 执行趴下中
23.     ASTRALL_SPORT_STATUS_GET_RIGHT = 0xB1FF     // 摔倒自主恢复站立
24. };
```

### 5.8.3 详细说明

Daedalus 初始化成功后, 返回的 AstrallSdkStatus 状态中的 link 为连接状态时, 返回数据才有效, 否则数据无效。

## 5.9 传感器信息订阅接口

| 函数名  | AstrallSubscriptionData                                                                                                                                                   |
|------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 函数原型 | uint16_t AstrallSubscriptionData(AstrallSubscribeTopicId topicId,<br>AstrallSubscribeFreq freq,<br>std::function<void(void *data, uint16_t len)> cb,<br>uint32_t timeout) |
| 功能描述 | 订阅传感器数据                                                                                                                                                                   |

### 5.9.1 函数入参

输入参数 1: AstrallSubscribeTopicId topicId:

```
1. enum AstrallSubscribeTopicId : uint16_t
2. {
3.     ASTRALL_SUB_TOPIC_ID_IMU           = 0x0001U,    // Subscribe IMU Data
4.     ASTRALL_SUB_TOPIC_ID_SPORT         = 0x0002U,    // Subscribe -> wheel speed
5.     ASTRALL_SUB_TOPIC_ID_RGB           = 0x0003U,    // Subscribe RGB to point:6000
6.     ASTRALL_SUB_TOPIC_ID_JOYSTICK      = 0x0010U,    // Subscribe joystick data
7. };
```

输入参数 2: AstrallSubscribeFreq freq:

```
1. enum AstrallSubscribeFreq : uint16_t
2. {
3.     ASTRALL_SUB_FREQ_CLOSE             = 0U,
4.     ASTRALL_SUB_FREQ_1HZ,
5.     ASTRALL_SUB_FREQ_25HZ,
6.     ASTRALL_SUB_FREQ_50HZ,
7.     ASTRALL_SUB_FREQ_125HZ,
8.     ASTRALL_SUB_FREQ_250HZ
9. };
```

输入参数 3: 回调函数 std::function<void(void \*data, uint16\_t len)> cb;

输入参数 4: uint32\_t timeout: 超时时间 (ms)。

### 5.9.2 函数返回值

详见表 5-1 函数调用返回参数表。

## 5.9.3 详细说明

表 5-4-3 函数调用返回参数表

| TOPIC_ID                      | 支持频率 (Hz)           | 回调 data 数据结构        | 说明                                                                                      |
|-------------------------------|---------------------|---------------------|-----------------------------------------------------------------------------------------|
| ASTRALL_SUB_TOPIC_ID_IMU      | 1、25、50、<br>125、250 | AstrallImuData      |                                                                                         |
| ASTRALL_SUB_TOPIC_ID_SPORT    | 1、25、50、<br>125、250 | AstrallSportData    |                                                                                         |
| ASTRALL_SUB_TOPIC_ID_RGB      | --                  | --                  | 订阅频率填除 0 外的任意字段, (填 0 为取消订阅), 订阅成功后会通过 UDP 向 Daedalus 启动设备的端口 6000 推送相机 H264 格式 RGB 码流。 |
| ASTRALL_SUB_TOPIC_ID_JOYSTICK | 50                  | AstrallJoystickData |                                                                                         |

对应的数据结构格式如下:

```
1. struct AstrallImuData
2. {
3.     long long timestamp;           // IMU timestamp
4.     float accelerometer[3];        // Acceleration: m/(s*s)
5.     float gyroscope[3];           // Angular velocity: rad/s
6.     float quaternion[4];          // Quaternion
7.     float pitch;                   // Pitch angle: rad
8.     float roll;                    // Roll angle: rad
9.     float yaw;                     // Heading angle: rad
10.    float odomX;
11.    float odomY;
12.
13.    AstrallImuData()
14.    {
15.        timestamp = 0;
16.        memset(accelerometer, 0, sizeof(accelerometer));
17.        memset(gyroscope, 0, sizeof(gyroscope));
18.        memset(quaternion, 0, sizeof(quaternion));
19.        pitch = 0.0f;
20.        yaw = 0.0f;
21.        roll = 0.0f;
22.        odomX = 0.0f;
23.        odomY = 0.0f;
24.    }
```

```
25. };
26.
27. struct AstrallSportData
28. {
29.     long long timestamp;
30.     float wheelSpeed[4];
31.
32.     AstrallSportData()
33.     {
34.         timestamp = 0;
35.         memset(wheelSpeed, 0, sizeof(wheelSpeed));
36.     }
37. };
38.
39. struct AstrallJoystickData
40. {
41.     long long timestamp;
42.     int16_t rocker[6];
43.     uint16_t l1 : 1;
44.     uint16_t l2 : 1;
45.     uint16_t l3 : 1;
46.     uint16_t r1 : 1;
47.     uint16_t r2 : 1;
48.     uint16_t r3 : 1;
49.     uint16_t sl : 1;
50.     uint16_t sr : 1;
51.     uint16_t h : 1;
52.     uint16_t sw : 2;
53.     uint16_t reserve : 5;
54. };
```

注意：雷达数据需要 调用厂家 SDK 或通过 UDP 在对应端口获取点云、雷达内置 IMU 数据。

## 5.10 向遥控器消息窗口发送消息接口

| 函数名  | AstrallSendMessage                                    |
|------|-------------------------------------------------------|
| 函数原型 | uint16_t AstrallSendMessage(char *data, uint16_t len) |
| 功能描述 | 向遥控器消息窗口发送消息                                          |

### 5.10.1 函数入参

输入参数 1: char \*data: 文本消息。

输入参数 2: uint16\_t len: 消息长度。

### 5.10.2 函数返回值

详见表 5-1 函数调用返回参数表。

### 5.10.3 详细说明

该函数调用频率应小于 5 Hz，否则返回失败。

## 5.11 运动模式切换接口

|      |                                                                              |
|------|------------------------------------------------------------------------------|
| 函数名  | AstrallSportModeControl                                                      |
| 函数原型 | uint16_t AstrallSportModeControl(AstrallSportModeCmd mode, uint32_t timeout) |
| 功能描述 | 控制机器人切换不同的模式                                                                 |

### 5.11.1 函数入参

输入参数 1: AstrallSportModeCmd mode: 结构及内容见下:

```
1. enum AstrallSportModeCmd : uint16_t
2. {
3.     ASTRALL_SPORT_MODE_CMD_DAMPING      = 0xA101,    // A101H Damping
4.     ASTRALL_SPORT_MODE_CMD_FIXEDSTAND    = 0xA102,    // A102H Fixed Stand
5.     ASTRALL_SPORT_MODE_CMD_FIXEDDOWN     = 0xA103,    // A103H Fixed Down
6.     ASTRALL_SPORT_MODE_CMD_MOVE          = 0xA104,    // A104H Move
7.     ASTRALL_SPORT_MODE_CMD_AUTOCHARGE    = 0xA105,    // A105 Auto Charge
8.     ASTRALL_SPORT_MODE_CMD_EXITCHARGE    = 0xA106,    // A106 Exit Charge
9.     ASTRALL_SPORT_MODE_CMD_GET_RIGHT     = 0xA1FF      // A1FFH Recover the ability to stand independently after falling down
10.};
```

输入参数 2: uint32\_t timeout: 超时时间 (单位: ms)。

### 5.11.2 函数返回值

详见表 5-1 函数调用返回参数表。

### 5.11.3 详细说明

- 1) 阻尼模式相当于软急停，具有最高优先级，可以在任意状态立即切换至阻尼模式；
- 2) 模式切换过程中必须等待上一个模式切换完成才能切换下一个模式，否则返回失败；
- 3) 全地形行走模式(Move)，只能在机器人完成站立后切换，否则返回失败；
- 4) 自动充电指令只能在站立模式或全地形行走模式中才能执行，否则返回失败；

- 5) 退出充电桩指令，只能在机器人状态处于充电中或在充电桩上才能执行，否则返回失败；
- 6) 机器人的当前状态处于在桩上或充电中不能执行趴下动作；
- 7) 机器人处于在桩上或充电中不响应如趴下、站立等指令，只响应退出充电桩指令；
- 8) 机器人在任意模式中都可以执行急停，进入阻尼模式；
- 9) 机器人在充电桩上重启后，可以发送退桩指令，也可以直接执行站立，通过指令完成退桩。

## 5.12 控制权申请接口

|      |                                                                 |
|------|-----------------------------------------------------------------|
| 函数名  | AstrallAuthControl                                              |
| 函数原型 | uint16_t AstrallAuthControl(AstrallAuth auth, uint32_t timeout) |
| 功能描述 | 申请控制权接口                                                         |

### 5.12.1 函数入参

输入参数 1: AstrallAuth auth: 控制权 ID:

```
1. enum AstrallAuth : uint16_t
2. {
3.     ASTRALL_AUTH_SDK           = 1U,           // 1U SDK control operation
4.     ASTRALL_AUTH_JOYSTICK      = 2U           // 2U Remote control operation
5. };
```

输入参数 2: uint32\_t timeout: 超时时间 (单位: ms)。

### 5.12.2 函数返回值

详见表 5-1 函数调用返回参数表。

### 5.12.3 详细说明

控制权说明:

控制权: 是指当前具备控制机器人进行运动的权限, 控制对象有 Daedalus、PSDK、遥控器等, 控制权允许申请、抢夺和主动释放。

- 1) 在调用机器人控制接口前, 需要先调用 AstrallAuthControl() 接口申请控制权, 在获得机器人控制权之前, 机器人的控制接口将无法生效。

2) 失控行为：当机器人在控制过程中由于各种突发因素导致的不受控状态（如遥控器断联、SDK 程序崩溃、机器人低电量等场景）。进入失控模式后，机器人的控制权会发生变化并执行对应的预设失控行为（如进入锁定关节站立、趴下、整机急停等）。

3) 控制权主动切换的场景：Daedalus 主动释放控制权、遥控器抢夺控制权、遥控器主动释放。

4) 控制权被动切换的场景：低电量、Daedalus 异常断联、在没有任何设备连接时的遥控器断联、关节电机异常(包括过温、过载、编码器错误等)、ST0 被按下、保护模式。

5) Daedalus 连接成功后，默认没有控制权，可以通过初始化时注册的回调接口 (sdkStatusCb) 查看控制权情况和调用 AstrallAuthControl() 接口申请控制权。

6) 当有多个设备申请控制权时，第一个申请控制权的设备会获得控制权，其他设备返回申请失败。

7) 当 Daedalus 拥有控制权时，此时操作遥控器进行抢夺控制权操作(三档开关由 1 调至 3)，遥控器会获得控制权，Daedalus 失去控制权，仅失去控制权，不影响获取之前订阅的信息，但是不能订阅新的信息。

8) 当遥控器拥有控制权时，其他设备无法再申请控制权，需要遥控器主动释放控制权（将三档开关拨回 1），释放控制权后，其他设备可以申请控制权。

9) 机器人允许无遥控器开机，在无遥控器开机后，Daedalus 可以直接申请控制权，不需要遥控器切换。

## 5.13 移动接口

| 函数名  | AstrallMove                                                            |
|------|------------------------------------------------------------------------|
| 函数原型 | uint16_t AstrallMove(float vx, float vy, float vyaw, uint32_t timeout) |
| 功能描述 | 机器人移动控制接口                                                              |

### 5.13.1 函数入参

输入参数 1: float vx: 前进速度（单位： m/s）；

输入参数 2: float vy: 平移速度（单位： m/s）；

输入参数 3: float vyaw: 旋转速度（单位： rad/s）；

输入参数 4: uint32\_t timeout: 超时时间（单位： ms）。

### 5.13.2 函数返回值

详见表 5-1 函数调用返回参数表。

### 5.13.3 详细说明

速度控制 停止时需要发 (0.0, 0.0, 0.0)

正在移动中如果超 50ms 未接收到移动命令将自动停止。

参数范围 vx[-1.0, +1.0] vy[-1.0, +1.0] vyaw[-1.0, +1.0].

## 5.14 头部灯光控制接口

|      |                                                                     |
|------|---------------------------------------------------------------------|
| 函数名  | AstrallLightControl                                                 |
| 函数原型 | uint16_t AstrallLightControl(AstrallLightCmd cmd, uint32_t timeout) |
| 功能描述 | 控制机器人头部探照灯                                                          |

### 5.14.1 函数入参

输入参数 1: AstrallLightCmd cmd: 开关, 参考 AstrallLightCmd 类型:

```
1. enum AstrallLightCmd : uint16_t
2. {
3.     ASTRALL_CMD_LIGHT_OPEN      = 1U, // 1U turn on
4.     ASTRALL_CMD_LIGHT_CLOSE     = 2U  // 2U turn off
5. };
```

输入参数 2: uint32\_t timeout: 超时时间 (单位: ms)。

### 5.14.2 函数返回值

详见表 5-1 函数调用返回参数表。

### 5.14.3 详细说明

无。