



 hitbot@hitbot.cc
www.hitbot.cc

Z-Arm- API-LINUX-CPP

领先的轻量型协作机械臂提供商

主营: 工业机器人 / 协作机器人 / 电动夹爪 / 电缸模组

慧灵科技（深圳）有限公司
Huiling-tech Robotic Co., LTD

发布说明

日期	版本	发布说明
2019.01.19	1.0.0.20191019_beta	第一版
2020.02.28	2.1.0.20200228_release	1-优化通讯协议，解决手臂连接断开问题 2-添加 movej_xyz_lr 等函数 3-其他常规更新
2020.08.14	2.1.0.20200814_release	1, 常规更新 2, 同步 windows 版本新添加功能

目录

1、准备工作.....	5
2、SDK 接口.....	5
2.1 hitbot_interface.h:	5
2.1.1 导出函数:	5
2.1.1.1 net_port_initial.....	5
2.1.1.2 card_number_connect.....	6
2.1.1.3 get_robot.....	6
2.2 ControlBeanEx 类:	6
2.2.1 ControlBeanEx 成员变量:	7
2.2.1.1 机械臂当前状态:	7
2.2.1.2 位置预判.....	8
2.2.2 ControlBeanEx 成员函数:	8
2.2.2.1 initial.....	8
2.2.2.2 get_scara_param.....	9
2.2.2.3 set_arm_length(过时，不推荐使用).....	9
2.2.2.4 int unlock_position.....	10
2.2.2.5 is_connected.....	11
2.2.2.6 get_joint_state.....	11
2.2.2.7 set_drag_teach.....	12
2.2.2.8 get_drag_teach.....	13
2.2.2.9 set_cooperation_fun_state.....	13
2.2.2.10 get_cooperation_fun_state.....	14
2.2.2.11 is_collision.....	14
2.2.2.12 set_position_move(过时，推荐使用 new_movej_xyz_lr).....	14
2.2.2.13 set_angle_move(过时，推荐使用 new_movej_xyz_lr).....	16
2.2.2.14 xyz_move(过时，推荐使用 new_movej_xyz_lr).....	18
2.2.2.15 single_joint_move(过时，推荐使用 new_movej_xyz_lr).....	19
2.2.2.16 trail_move.....	20
2.2.2.17 stop_move.....	21
2.2.2.18 is_robot_goto_target.....	22
2.2.2.19 set_allow_offset_at_target_position.....	22
2.2.2.20 set_catch_or_release_accuracy.....	23

2.2.2.21 judge_in_range	23
2.2.2.22 judge_position_gesture (过时, 不推荐使用).....	24
2.2.2.23 joint_home	24
2.2.2.24 moveL_xyz	25
2.2.2.25 moveJ_xyz (过时, 推荐使用 new_moveJ_xyz_lr).....	26
2.2.2.26 moveJ_angle (过时, 推荐使用 new_moveJ_xyz_lr).....	28
2.2.2.27 change_attitude	30
2.2.2.28 wait_stop	30
2.2.2.29 pause_move	31
2.2.2.30 resume_move	31
2.2.2.31 set_digital_out	31
2.2.2.32 get_digital_out	32
2.2.2.33 get_digital_in	32
2.2.2.34 set_efg_state	33
2.2.2.35 get_efg_state	34
2.2.2.36 moveJ_xyz_lr	34
3.2.2.37 new_moveJ_xyz_lr	36
3.2.2.38 new_set_acc	38
3.2.2.39 j5_motor_zero	38
3.2.2.40 set_j5_motor_pos	39
3.2.2.41 get_j5_parameter	40
3.2.2.42 moveJ_j5	40
3.2.2.43 get_j5_positon (过时, 建议使用 get_j5_paramter).....	41
3.2.2.44 get_efg_state_dji	41
3.2.2.45 set_efg_state_dji	42
3.2.2.46 new_stop_move	42
3.2.2.47 new_moveJ_angle	43

目前可支持 Ubunut 16.04 - 18.04 X64 平台。

1、准备工作

- 打开终端输入以下命令，复制库文件到系统库目录,: sudo cp libsmall_scara_interface.so libsmall_scara_interface.so.1 libsmall_scara_interface.so.1.0 libsmall_scara_interface.so.1.0.0 /usr/lib
sudo ldconfig
- 工程中引用 hitbot_interface.h 和 ControlBean.h 两个头文件。

2、SDK 接口

2.1 [hitbot_interface.h](#):

头文件	hitbot_interface.h
导出函数功能	申请内存和初始化网络服务器，获得机械臂控制实例对象等。

2.1.1 导出函数:

2.1.1.1 [net_port_initial](#)

功能	申请内存和初始化网络服务器
注意	调用其他 api 之前，需要先调用该函数，否则将导致程序崩溃，函数返回后，建议延迟 3s
函数定义	int net_port_initial()
参数	无
返回	1 开启成功

2 开启失败，一般是端口号被占用

2.1.1.2 `card_number_connect`

功能	查询指定的机械臂是否在线
注意	传入参数范围应 $>=0, <=255$
函数定义	<code>int card_number_connect(int card_number)</code>
参数	<code>int card_number</code> : 机械臂 ip 地址的第四位
返回	<ul style="list-style-type: none">• 0: 未连接• 1: 已连接• 2: 传入参数错误• 101: 传入参数 NOT A NUMBER

2.1.1.3 `get_robot`

功能	获取机械臂实例对象
注意	传入参数范围应 $>=0, <=255$
函数定义	<code>ControlBeanEx * get_robot(int card_number)</code>
参数	<code>int card_number</code> : 机械臂 ip 地址的第四位

2.2 `ControlBeanEx` 类:

类名	<code>ControlBeanEx</code>
----	----------------------------

功能	通过成员函数和成员变量控制机械臂和获取机械臂当前状态。
----	-----------------------------

2.2.1 ControlBeanEx 成员变量：

2.2.1.1 机械臂当前状态：

调用 <code>get_scara_param</code> 更新	
变量定义	功能
<code>float x</code>	x 坐标 (mm)
<code>float y</code>	y 坐标 (mm)
<code>float z</code>	z 坐标 (mm)
<code>float rotation</code>	轴 4 角度 (deg)
<code>float angle1</code>	轴 1 角度 (deg)
<code>float angle2</code>	轴 2 角度 (deg)
<code>bool communicate_success</code>	<ul style="list-style-type: none">• true: 不在线• false: 在线
<code>bool initial_finish</code>	<ul style="list-style-type: none">• true: 已初始化• false: 未初始化
<code>bool move_flag</code>	<ul style="list-style-type: none">• true: 正在运动• false: 静止状态

2.2.1.2 位置预判

调用 <code>judge_position_gesture</code> 返回 <code>true</code> 更新	
变量定义	功能
<code>bool isReach_after_judge</code>	<ul style="list-style-type: none"> • <code>true</code>: <code>judge_position_gesture</code> 预判的坐标点位可以到达 • <code>false</code>: <code>judge_position_gesture</code> 预判的坐标点位不在机器人的工作范围内
<code>float angle1_after_judge</code>	机器人在预判点位的轴 1 角度坐标 (deg)
<code>float angle2_after_judge</code>	机器人在预判点位的轴 2 角度坐标 (deg)

2.2.2 ControlBeanEx 成员函数:

2.2.2.1 `initial`

功能	初始化机械臂的各项参数，机械臂进入工作状态
注意	<ul style="list-style-type: none"> • 机械臂在运动时调用该函数，会造成机械臂抖动 • 返回 3 时，可以使用 <code>joint_home</code> 使关节强制回零
函数定义	<pre>int initial(int generation, float z_travel)</pre>
参数	<ul style="list-style-type: none"> • <code>int generation</code>: 400mm 臂展系列传入 1, 320mm 臂展系

	列传入 5
	<ul style="list-style-type: none">• <code>float z_travel</code>: 上下关节有效行程 (mm)
返回	<ul style="list-style-type: none">• 0: 机械臂不在线• 1: 初始化成功• 2: <code>generation</code> 参数错误• 3: 机械臂当前位置不在限定范围, 可以使用 <code>joint_home</code> 使关节强制回零• 12: <code>z_travel</code> 传参错误• 101: 传入参数 NOT A NUMBER• 105: 存在某一个关节失效• >= 10000, pid 自检异常

2.2.2.2 `get_scara_param`

功能	更新 x, y, z 等成员变量
注意	机械臂不在线时, 坐标参数无意义
函数定义	<code>void get_scara_param()</code>
参数	无
返回	无

2.2.2.3 `set_arm_length`(过时, 不推荐使用)

功能	设置机械臂轴 1 和轴 2 长度
----	------------------

注意	<code>initial</code> 函数中会按照设定机型自动设置该参数, 如果臂长没有变化, 无需重复调用
函数定义	<pre>void set_arm_length(float l1, float l2)</pre>
参数	<ul style="list-style-type: none">• <code>float l1</code>: 轴 1 长度 (mm)• <code>float l2</code>: 轴 2 长度 (mm)
返回	<ul style="list-style-type: none">• 0: 机械臂不在线• 1: 初始化成功• 2: generation 参数错误• 3: 机械臂当前位置不在限定范围• 11: 手机端正在控制, 本次初始化不成功• 12: z_travel 传参错误• 101: 传入参数 NOT A NUMBER

2.2.2.4 int unlock_position

功能	解锁机械臂, 使机械臂可以接受运动指令
注意	一般初始化成功后立刻调用
函数定义	<code>int unlock_position()</code>
参数	无
返回	<ul style="list-style-type: none">• 0: 机械臂不在线

- 1: 成功

2.2.2.5 `is_connected`

功能	查询机械臂是否在线
注意	无
函数定义	<code>bool is_connected()</code>
参数	无
返回	<ul style="list-style-type: none">• false: 机械臂不在线• true: 机械臂在线

2.2.2.6 `get_joint_state`

功能	获取机械臂关节状态
注意	无
函数定义	<code>int get_joint_state(int joint_num)</code>
参数	<code>int joint_num</code> : 轴号
返回	<ul style="list-style-type: none">• 0: 轴发生复位, 需要重新初始化• 1: 关节正常• 2: 传入参数超范围• 3: 未初始化• 4: 轴状态获取失败• 5: 发生碰撞;

- 6: 处于拖动模式
- 7: 关节缺轴/失效
- 8: 编码器错误
- 9: 电机堵转/过电流保护
- 10: 过压保护
- 11: 手臂收到 PC 发送的错误数据
- 12: 手臂主控与驱动通讯异常
- 21: MOS ntc 开路
- 22: MOS ntc 短路
- 23: 电机 ntc 开路
- 24: 电机 ntc 短路
- 25: MOS ntc 温度过高
- 26: MOS ntc 温度过低
- 27: 电机 ntc 温度过高
- 28: 电机 ntc 温度过低

2.2.2.7 `set_drag_teach`

功能	设定拖动示教功能是否开启
注意	<ul style="list-style-type: none">• 仅协作机型支持开启此功能• 拖动示教功能开启后，轴 3 不支持拖动，需要通过调用运动函数控制轴 3，

函数定义	<code>bool set_drag_teach(bool state)</code>
参数	<code>bool state</code> : true 开启, false 关闭
返回	<ul style="list-style-type: none">• false: 失败• true: 成功

2.2.2.8 `get_drag_teach`

功能	查询是否处于拖动示教状态
注意	无
函数定义	<code>bool get_drag_teach()</code>
参数	无
返回	<ul style="list-style-type: none">• false: 开启• true: 关闭

2.2.2.9 `set_cooperation_fun_state`

功能	设定协作功能是否开启, 开启后机械臂遇到障碍物会停止运动, 并上报此状态
注意	仅协作机型支持开启此功能
函数定义	<code>bool set_cooperation_fun_state(bool state)</code>
参数	<code>bool state</code> : true 开启, false 关闭
返回	<ul style="list-style-type: none">• false: 成功• true: 失败

2.2.2.10 `get_cooperation_fun_state`

功能	查询机械臂是否开启协作功能
注意	
函数定义	<code>bool get_cooperation_fun_state()</code>
参数	无
返回	<ul style="list-style-type: none">• <code>false</code>: 开启• <code>true</code>: 关闭

2.2.2.11 `is_collision`

功能	查询机械臂是否发生碰撞
注意	无
函数定义	<code>bool is_collision()</code>
参数	无
返回	<ul style="list-style-type: none">• <code>false</code>: 未发生碰撞• <code>true</code>: 发生碰撞

2.2.2.12 `set_position_move`(过时, 推荐使用 `new_movej_xyz_lr`)

功能	驱动机械臂到达设定的 xyzr 坐标点
注意	<ul style="list-style-type: none">• <code>move_flag</code> 等于 <code>false</code> 时调用有效

	<ul style="list-style-type: none">• movej 模式下，机械臂优先以当前姿态到达目标点
函数定义	<pre>int set_position_move(float goal_x, float goal_y, float goal_z, float goal_r, float speed, float acceleration, int interpolation, int move_mode)</pre>
参数	<ul style="list-style-type: none">• float goal_x: 目标点的 x 坐标值 (mm)• float goal_y: 目标点的 y 坐标值 (mm)• float goal_z: 目标点的 z 坐标值 (mm)• float goal_r: 目标点的 r 坐标值 (deg)• float speed: xyz 点移动距离小于等于 1mm, 轴 4 转动角度大于 0.0.1deg 时, speed 为轴 4 旋转速度 (deg/s), 其他关节做协同运动, 其他情况 speed 为 xyz 点线速度 (mm/s), 轴 4 做协同运动• float acceleration: T 型插补时的加速度 (mm/s²)• int interpolation: 1 S 型插补, 2 T 型插补• int move_mode: 1 角度运动 (MOVEJ) ,2 直线运动 (MOVEL)
返回	<ul style="list-style-type: none">• -2: 不支持该运动函数

- 0: 正在执行其他指令，本次指令无效
- 1: 本次指令生效，机械臂开始运动
- 2: 设置速度小于等于零
- 3: 未初始化
- 4: MOVEJ 运动模式下，目标点无法到达
- 6: 伺服未开启
- 7: MOVEL 运动方式下，存在中间过程点无法以机械臂当前姿态（手系）达到
- 8: 加速度小于等于零
- 9: 插补方式参数错误
- 10: 移动方式错误
- 11: 手机端在控制
- 101: 传入参数 NOT A NUMBER
- 102: 发生碰撞，本次指令无效
- 103: 轴发生复位，需要重新初始化，本次指令无效

2.2.2.13 set_angle_move(过时，推荐使用 new_movej_xyz_lr)

功能	驱动机械臂到达设定的 angle1, angle2, z, r 坐标点
注意	move_flag 等于 false 时调用有效
函数定义	<pre>int set_angle_move(float goal_angle1, float goal_angle2,</pre>

	<pre>float goal_z, float goal_r, float speed)</pre>
参数	<ul style="list-style-type: none">• float goal_angle1: 目标点轴 1 角度坐标值 (deg)• float goal_angle2: 目标点轴 2 角度坐标值 (deg)• float goal_z: 目标点 z 坐标值 (mm)• float goal_r: 目标点 r 坐标值 (deg)• float speed: xyz 点移动距离小于等于 1mm, 轴 4 转动角度大于 0.01deg 时, speed 为轴 4 旋转速度 (deg/s), 其他关节做协同运动, 其他情况 speed 为 xyz 点线速度 (mm/s), 轴 4 做协同运动
返回	<ul style="list-style-type: none">• -2: 不支持该运动函数• 0: 正在执行其他指令, 本次指令无效• 1: 本次指令生效, 机械臂开始运动• 2: 设置速度小于等于零• 3: 未初始化• 4: MOVEJ 运动模式下, 目标点无法到达• 6: 伺服未开启• 11: 手机端在控制• 101: 传入参数 NOT A NUMBER• 102: 发生碰撞, 本次指令无效• 103: 轴发生复位, 需要重新初始化, 本次指令无效

2.2.2.14 xyz_move(过时，推荐使用 new_movej_xyz_lr)

功能	驱动机械臂沿某一方向运动指定距离
注意	<ul style="list-style-type: none">move_flag 等于 false 时调用有效运动轨迹为直线
函数定义	<pre>int xyz_move(int direction, float distance, float speed)</pre>
参数	<ul style="list-style-type: none">int direction: 1 x 轴正方向, 2 y 轴正方向, 3 z 轴正方向float distance: 移动距离 (mm)float speed : 移动速度 (mm/s)
返回	<ul style="list-style-type: none">-2: 不支持该运动函数0: 正在执行其他指令, 本次指令无效1: 本次指令生效, 机械臂开始运动2: 设置速度小于等于零3: 未初始化5: 方向参数错误6: 伺服未开启7: 某些过程点无法到达11: 手机端在控制

- 101: 传入参数 NOT A NUMBER
- 102: 发生碰撞，本次指令无效
- 103: 轴发生复位，需要重新初始化，本次指令无效

2.2.2.15 `single_joint_move`(过时，推荐使用 `new_movej_xyz_lr`)

功能	驱动机械臂某一个轴旋转指定角度
注意	<ul style="list-style-type: none">• <code>move_flag</code> 等于 <code>false</code> 时调用有效
函数定义	<pre>int single_joint_move(int axis, float distance, float speed)</pre>
参数	<ul style="list-style-type: none">• <code>int axis</code>: 1 轴 1 正方向, 2 轴 2 正方向, 3 轴 3 正方向, 4 轴 4 正方向• <code>float distance</code>: 旋转角度 (deg)• <code>float speed</code> : 轴 3 为线速度 (mm/s), 其他为旋转速度 (deg/s)
返回	<ul style="list-style-type: none">• -2: 不支持该运动函数• 0: 正在执行其他指令，本次指令无效• 1: 本次指令生效，机械臂开始运动• 2: 设置速度小于等于零• 3: 未初始化

- 5: 轴号参数错误
- 6: 伺服未开启
- 11: 手机端在控制
- 101: 传入参数 NOT A NUMBER
- 102: 发生碰撞, 本次指令无效
- 103: 轴发生复位, 需要重新初始化, 本次指令无效

2.2.2.16 trail_move

功能	驱动机械臂沿一条特定的轨迹运行
注意	<ul style="list-style-type: none">• move_flag 等于 false 时调用有效• 轨迹坐标 xyzr 分别保存在四个 float 类型的数组中• 轨迹中相邻两点的距离应为 1mm
函数定义	<pre>int trail_move(int point_number, float *x, float *y, float *z, float *r, float speed)</pre>
参数	<ul style="list-style-type: none">• int point_number: 总点数• float *x: 存放 x 坐标数组的首地址• float *y: 存放 y 坐标数组的首地址

	<ul style="list-style-type: none">• float *z: 存放 z 坐标数组的首地址• float *r: 存放 r 坐标数组的首地址• float speed:
返回	<ul style="list-style-type: none">• 0: 正在执行其他指令，本次指令无效• 1: 本次指令生效，机械臂开始运动• 2: 设置速度小于等于零• 3: 未初始化• 4: 轨迹起点无法到达• 6: 伺服未开启• 11: 手机端在控制• 99: 急停中• 101: 传入参数 NOT A NUMBER• 102: 发生碰撞，本次指令无效• 103: 轴发生复位，需要重新初始化，本次指令无效

2.2.2.17 **stop_move**

功能	立即结束当前正在执行的指令
注意	无
函数定义	void stop_move()
参数	无
返回	无

2.2.2.18 `is_robot_goto_target`

功能	查询机械臂是否到达目标点
注意	无
函数定义	<code>bool is_robot_goto_target()</code>
参数	无
返回	<ul style="list-style-type: none">• <code>false</code>: 未到达• <code>true</code>: 已到达

2.2.2.19 `set_allow_offset_at_target_position`

功能	设定 <code>is_robot_goto_target</code> 的评判标准
注意	无
函数定义	<pre>void set_allow_offset_at_target_position(float x_distance float y_distance float z_distance float r_distance)</pre>
参数	<ul style="list-style-type: none">• <code>float x_distance</code>: x 轴方向允许误差 (mm)• <code>float y_distance</code>: y 轴方向允许误差 (mm)• <code>float z_distance</code>: z 轴方向允许误差 (mm)• <code>float r_distance</code>: r 轴方向允许误差 (deg)
返回	无

2.2.2.20 `set_catch_or_release_accuracy`

功能	设定机械臂沿 z 轴竖直向下运动时，是否到达目标点的判断标准
注意	无
函数定义	<code>void set_catch_or_release_accuracy(float accuracy);</code>
参数	<ul style="list-style-type: none">• <code>float accuracy</code>: 判断标准 (mm)
返回	无

2.2.2.21 `judge_in_range`

功能	判断点位是否可以到达
注意	无
函数定义	<pre>bool judge_in_range(float x, float y, float z, float rotation)</pre>
参数	<ul style="list-style-type: none">• <code>float x</code>: x 坐标 (mm)• <code>float y</code>: y 坐标 (mm)• <code>float z</code>: z 坐标 (mm)• <code>float rotation</code>: r 坐标 (deg)
返回	<ul style="list-style-type: none">• <code>false</code>: 超出范围无法到达

- **true**: 可以到达

2.2.2.22 judge_position_gesture(过时, 不推荐使用)

功能	判断在当前位置调用 <code>set_position_move</code> 以 movej 模式到达 xy 时机械臂的姿态
注意	<ul style="list-style-type: none">• move_flag 等于 false 时调用有效• 返回 true 后, <code>isReach_after_judge</code>, <code>angle1_after_judge</code>, <code>angle2_after_judge</code> 将会被更新
函数定义	<pre>bool judge_position_gesture(float x, float y)</pre>
参数	<ul style="list-style-type: none">• float x: x 坐标 (mm)• float y: y 坐标 (mm)
返回	<ul style="list-style-type: none">• false: 机械臂正在运动, 判定无效• true: 判定成功,

2.2.2.23 joint_home

功能	使轴回到零位
注意	无
函数定义	<code>int joint_home(int joint_num)</code>

参数	<ul style="list-style-type: none"> • <code>int joint_num</code>: 轴号
返回	<ul style="list-style-type: none"> • 0: 未连接 • 1: 调用成功 • 2: 传入参数错误 • 3: 机械臂正在初始

2.2.2.24 `moveL_xyz`

功能	以 <code>moveL</code> 的运动方式到达目标点
注意	<ul style="list-style-type: none"> • <code>move_flag</code> 等于 <code>false</code> 时, 进行首次调用, 返回 1 后, 可以重复调用该函数设定新的目标点位 • 距离当前目标点有一定距离时, 再次设定新的目标, 机械臂将以一定的速度通过当前目标点, 而后去往新的目标点。
函数定义	<pre>int moveL_xyz(float goal_x, float goal_y, float goal_z, float goal_r, float speed)</pre>
参数	<ul style="list-style-type: none"> • <code>float goal_x</code>: 目标点 x 坐标 (mm) • <code>float goal_y</code>: 目标点 y 坐标 (mm) • <code>float goal_z</code>: 目标点 z 坐标 (mm) • <code>float goal_r</code>: 目标点 r 坐标 (deg)

	<ul style="list-style-type: none">• float speed: xyz 点的线速度 (mm/s) 或轴 4 的旋转角速度 (deg/s)
返回	<ul style="list-style-type: none">• 0: 正在执行其他指令，本次指令无效• 1: 本次指令生效，机械臂开始运动• 2: 设置速度小于等于零• 3: 未初始化• 4: 过程点无法到达• 6: 伺服未开启• 7: 存在中间过程点无法以机械臂当前姿态（手系）达到• 11: 手机端在控制• 99: 急停中• 101: 传入参数 NOT A NUMBER• 102: 发生碰撞，本次指令无效• 103: 轴发生复位，需要重新初始化，本次指令无效

2.2.2.25 movej_xyz(过时，推荐使用 new_movej_xyz_lr)

功能	以 movej 的运动方式到达目标点
注意	<ul style="list-style-type: none">• move_flag 等于 false 时，进行首次调用，返回 1 后，可以重复调用该函数设定新的目标点位• 距离当前目标点有一定距离时，并且当前目标点 roughly 大于 0，再次设定新的目标，机械臂将以一定的速度在当前目

	标点附近通过，而后去往新的目标点。
函数定义	<pre>int movej_xyz(float goal_x, float goal_y, float goal_z, float goal_r, float speed, float roughly)</pre>
参数	<ul style="list-style-type: none">• float goal_x: 目标点 x 坐标 (mm)• float goal_y: 目标点 y 坐标 (mm)• float goal_z: 目标点 z 坐标 (mm)• float goal_r: 目标点 r 坐标 (deg)• float speed: xyz 点的线速度 (mm/s) 或轴 4 的旋转角速度 (deg/s)• float roughly: 0 先运动到前目标点，并且在当前目标点的速度等于 0，而后在去往新的目标点，1 在有新目标点时，将在当前目标点附近以最大速度通过，此参数取值范围 0 到 1，越大通过速度越快，但是通过点距离当前目标点越远。
返回	<ul style="list-style-type: none">• -2: 不支持该运动函数• 0: 正在执行其他指令，本次指令无效• 1: 本次指令生效，机械臂开始运动• 2: 设置速度小于等于零• 3: 未初始化

- 4: 目标点无法到达
- 6: 伺服未开启
- 11: 手机端在控制
- 101: 传入参数 NOT A NUMBER
- 102: 发生碰撞, 本次指令无效
- 103: 轴发生复位, 需要重新初始化, 本次指令无效

2.2.2.26 movej_angle(过时, 推荐使用 new_movej_xyz_lr)

功能	以 movej 的运动方式到达目标点
注意	<ul style="list-style-type: none">• move_flag 等于 false 时, 进行首次调用, 返回 1 后, 可以重复调用该函数设定新的目标点位• 距离当前目标点有一定距离时, 并且当前目标点 roughly 大于 0, 再次设定新的目标, 机械臂将以一定的速度在当前目标点附近通过, 而后去往新的目标点。
函数定义	<pre>int movej_angle(float angle1, float angle2, float goal_z, float goal_r, float speed, float roughly)</pre>
参数	<ul style="list-style-type: none">• float angle1: 目标点轴 1 角度坐标 (deg)

- **float angle2:** 目标点轴 2 角度坐标 (deg)
- **float goal_z:** 目标点 z 坐标 (mm)
- **float goal_r:** 目标点 r 坐标 (deg)
- **float speed:** xyz 点的线速度 (mm/s) 或轴 4 的旋转角速度 (deg/s)
- **float roughly:** 0 先运动到前目标点，并且在当前目标点的速度等于 0，而后在去往新的目标点，1 在有新目标点时，将在当前目标点附近以最大速度通过，此参数取值范围 0 到 1，越大通过速度越快，但是通过点距离当前目标点越远。

- -2: 不支持该运动函数
- 0: 正在执行其他指令，本次指令无效
- 1: 本次指令生效，机械臂开始运动
- 2: 设置速度小于等于零
- 3: 未初始化
- 4: 目标点无法到达
- 6: 伺服未开启
- 11: 手机端在控制
- 101: 传入参数 NOT A NUMBER
- 102: 发生碰撞，本次指令无效
- 103: 轴发生复位，需要重新初始化，本次指令无效

返回

2.2.2.27 `change_attitude`

功能	切换手系
注意	
函数定义	<code>int change_attitude(float speed)</code>
参数	<code>float speed</code> : 线速度 (mm/s)
返回	<ul style="list-style-type: none">• 0: 机械臂正在运行其他指令，本次制定无效• 1: 本次指令生效，机械臂开始移动• 2: 传入的速度小于等于 0• 3: 尚未初始化• 4: 另一姿态无法到达• 6: 伺服未开启• 11: 手机端正在控制• 101: 传入参数 NAN• 102: 发生碰撞，无法运动• 103: 关节发生复位，需要重新初始化

2.2.2.28 `wait_stop`

功能	用于判断运动是否结束
注意	<ul style="list-style-type: none">• 阻塞函数，直到运动结束才会返回
函数定义	<code>bool wait_stop()</code>
参数	无

返回	• 默认返回 true
----	-------------

2.2.2.29 pause_move

功能	暂停机械臂的运动
注意	函数返回后，经过一定延时（几百毫秒），机械臂停止运动
函数定义	int pause_move()
参数	无
返回	• 默认返回 1

2.2.2.30 resume_move

功能	恢复机械臂的运动
注意	无
函数定义	int resume_move()
参数	无
返回	• 默认返回 1

2.2.2.31 set_digital_out

功能	设定 io 输出点状态
注意	
函数定义	bool set_digital_out(int io_number,

	<code>bool value)</code>
参数	<ul style="list-style-type: none">• <code>int io_number</code>: io 输出点编号• <code>bool value</code>: 设定值
返回	<ul style="list-style-type: none">• 0: <code>io_number</code> 参数错误• 1: 设置成功• 3: 未初始化;

2.2.2.32 `get_digital_out`

功能	获取 io 输出点状态
注意	
函数定义	<code>int get_digital_out(int io_out_num)</code>
参数	<code>int io_number</code> : io 输出点编号
返回	<ul style="list-style-type: none">• -1: <code>io_number</code> 参数错误• 0: io 口输出状态为断开• 1: io 口输出状态为导通• 3: 未初始化

2.2.2.33 `get_digital_in`

功能	获取 io 输入点状态
注意	

函数定义	<code>int get_digital_in(int io_in_number)</code>
参数	<code>int io_number</code> : io 输入点编号
返回	<ul style="list-style-type: none"> -1: <code>io_number</code> 参数错误 0: io 输入点没有被触发 1: io 输入点被触发 3: 未初始化

2.2.2.34 `set_efg_state`

功能	设定电动夹爪状态
注意	<ul style="list-style-type: none"> 切换控制类型需要重启机械臂和 sdk
函数定义	<pre>int set_efg_state(int type, float distance)</pre>
参数	<ul style="list-style-type: none"> <code>int type</code>: 8 EFG-8, 20 EFG-20 <code>float distance</code>: <p>Type==8:</p> <ul style="list-style-type: none"> 0 张开 1 闭合 <p>Type==20: 实际行程 (mm)</p>
返回	<ul style="list-style-type: none"> -1: 控制参数发生变化 0: <code>type</code> 参数错误

- 1: 设置成功
- 3: 未初始化;

2.2.2.35 get_efg_state

功能	获取电动夹爪状态
注意	<ul style="list-style-type: none">• 1632 机型不支持在运动过程中调用
函数定义	<pre>int get_efg_state (int* type, float* distance)</pre>
参数	<ul style="list-style-type: none">• int* type: 电爪类型指针• float* distance: 电爪当前状态指针
返回	<ul style="list-style-type: none">• 1: 成功• 3: 失败, 未初始化;

2.2.2.36 movej_xyz_lr

功能	以 movej 的方式运动，并以指定姿态到达目标点
注意	<ul style="list-style-type: none">• move_flag 等于 false 时，进行首次调用，返回 1 后，可以重复调用该函数设定新的目标点位• 距离当前目标点有一定距离时，并且当前目标点 roughly 大于 0，再次设定新的目标，机械臂将以一定的速度在当前目标点附近通过，而后去往新的目标点。

	<pre>int movej_xyz(float goal_x, float goal_y, float goal_z, float goal_r, float speed, float roughly, int lr)</pre>
参数	<ul style="list-style-type: none">• float goal_x: 目标点 x 坐标 (mm)• float goal_y: 目标点 y 坐标 (mm)• float goal_z: 目标点 z 坐标 (mm)• float goal_r: 目标点 r 坐标 (deg)• float speed: xyz 点的线速度 (mm/s) 或轴 4 的旋转角速度 (deg/s)• float roughly: 0 先运动到前目标点，并且在当前目标点的速度等于 0，而后在去往新的目标点，1 在有新目标点时，将在当前目标点附近以最大速度通过，此参数取值范围 0 到 1，越大通过速度越快，但是通过点距离当前目标点越远。• int lr: 1 右手系，-1 左手洗，angle2>0 右手系，否则为左手系
返回	<ul style="list-style-type: none">• -2: 不支持该运动函数• 0: 正在执行其他指令，本次指令无效• 1: 本次指令生效，机械臂开始运动

- 2: 设置速度小于等于零
- 3: 未初始化
- 4: 目标点无法到达
- 6: 伺服未开启
- 7: 无法以指定姿态到达目标点位
- 11: 手机端在控制
- 101: 传入参数 NOT A NUMBER
- 102: 发生碰撞，本次指令无效
- 103: 轴发生复位，需要重新初始化，本次指令无效

3.2.2.37 new_movej_xyz_lr

功能	以 movej 的运动方式，并以指定手系达目标点
注意	<ul style="list-style-type: none">• move_flag 等于 false 时，进行首次调用，返回 1 后，可以重复调用该函数设定新的目标点位• 距离当前目标点有一定距离时，并且当前目标点 roughly 大于 0，再次设定新的目标，机械臂将以一定的速度在当前目标点附近通过，而后去往新的目标点。
函数定义	<pre>int new_movej_xyz_lr(float goal_x, float goal_y, float goal_z, float goal_r, float speed, float roughly,</pre>

	<pre>int lr) • float goal_x: 目标点 x 坐标 (mm) • float goal_y: 目标点 y 坐标 (mm) • float goal_z: 目标点 z 坐标 (mm) • float goal_r: 目标点 r 坐标 (deg) • float speed: xyz 点的线速度 (mm/s) 或轴 4 的旋转角速度 (deg/s)</pre>
参数	<p>• float roughly: 0 先运动到前目标点，并且在当前目标点的速度等于 0，而后在去往新的目标点，1 在有新目标点时，将在当前目标点附近以最大速度通过，此参数取值范围 0 到 1，越大通过速度越快，但是通过点距离当前目标点越远。</p> <p>• int lr: 1 右手系，-1 左手洗，angle2>0 右手系，否则为左手系</p>
返回	<ul style="list-style-type: none">• 0: 正在执行其他指令，本次指令无效• 1: 本次指令生效，机械臂开始运动• 2: 设置速度小于等于零• 3: 未初始化• 4: 目标点无法到达• 6: 伺服未开启• 7: 无法以设定手系到达目标点位• 11: 手机端在控制

	<ul style="list-style-type: none"> • 99：急停中 • 101：传入参数 NOT A NUMBER • 102：发生碰撞，本次指令无效 • 103：轴发生复位，需要重新初始化，本次指令无效
--	--

3.2.2.38 new_set_acc

功能	设定 new_movej_xyz_lr 运动模型的关节加速度百分比
注意	无，范围 $>=30, <=220$
函数定义	<pre>int new_set_acc(int j1_max_acc, int j2_max_acc, int j3_max_acc, int j4_max_acc)</pre>
参数	<ul style="list-style-type: none"> • <code>int j1_max_acc</code>: 关节 1 加速度百分比 • <code>int j2_max_acc</code>: 关节 2 加速度百分比 • <code>int j3_max_acc</code>: 关节 3 加速度百分比 • <code>int j4_max_acc</code>: 关节 4 加速度百分比
返回	• 1：默认返回 1

3.2.2.39 j5_motor_zero

功能	1) 搭配步进电机扩展模块使用，调用后会将当前位置作为零位
----	-------------------------------

	2) 搭配第五轴使用时, 无效
注意	无
函数定义	<code>int j5_motor_zero()</code>
参数	无
返回	<ul style="list-style-type: none"> • 0: 手臂未连接 • 1: 设定成功 • 3: 手臂未初始化 • 4: 不支持该功能

3.2.2.40 `set_j5_motor_pos`

功能	1) 搭配步进电机扩展模块使用, 设定步进电机的当前位置, 范围限制正负 1000 圈 2) 搭配第五轴使用时, 范围正负 150 度
注意	手臂本体四个关节在运动时, 禁止调用此接口
函数定义	<code>int set_j5_motor_pos(float deg, float speed);</code>
参数	<ul style="list-style-type: none"> • <code>float deg</code>: 目标角度 单位 deg • <code>float speed</code>: 平均速度 单位 deg/s
返回	<ul style="list-style-type: none"> • 0: 其他运动函数调用中 • 1: 设定成功 • 2: speed 小于等于零 • 3: 未初始化 • 4: 手臂当前 xyzr 超出运动范围

- 99: 急停中
- 102: 手臂发生碰撞
- 103: 关节复位
- 105: 关节异常
- 106: 不支持该功能

3.2.2.41 get_j5_parameter

功能	搭配步进电机扩展模块或第五轴使用，返回电机电机当前位置
注意	无
函数定义	<code>int get_j5_parameter();</code>
参数	无
返回	<ul style="list-style-type: none">• 返回五轴当前位置

3.2.2.42 movej_j5

功能	设定第五轴舵机位置
注意	<ul style="list-style-type: none">• 1632 机型不支持
函数定义	<pre>int movej_j5(float j5_pos, float speed)</pre>
参数	<ul style="list-style-type: none">• <code>float j5_pos</code>: 舵机位置 deg• <code>float speed</code>: 舵机转速 deg/s

	<ul style="list-style-type: none"> • 0: 正在执行其他指令，本次指令无效 • 1: 本次指令生效，机械臂开始运动 • 2: 设置速度小于等于零 • 3: 未初始化
返回	<ul style="list-style-type: none"> • 6: 伺服未开启 • 11: 手机端在控制 • 101: 传入参数 NOT A NUMBER • 102: 发生碰撞，本次指令无效 • 103: 轴发生复位，需要重新初始化，本次指令无效

3.2.2.43 `get_j5_positon` (过时，建议使用 `get_j5_paramter`)

功能	获取第五轴舵机的位置
注意	<ul style="list-style-type: none"> • 1632 机型不支持
函数定义	<code>float get_j5_positon(int type);</code>
参数	<ul style="list-style-type: none"> • <code>int type</code>: 舵机传入 2
返回	<ul style="list-style-type: none"> • 返回舵机位置 deg

3.2.2.44 `get_efg_state_dji`

功能	搭配 EFG-20 上电夹指不动版本使用，用于查询夹爪位置
注意	仅用于控制上电夹指不动 EFG-20 电动夹爪，上电位置为 0
函数定义	<pre>int get_efg_state_dji (int* type, float* distance</pre>

)
参数	<ul style="list-style-type: none"> int* type: 电爪类型指针 float* distance: 电爪当前状态指针
返回	<ul style="list-style-type: none"> 1: 成功 3: 失败, 未初始化;

3.2.2.45 set_efg_state_dji

功能	搭配 EFG-20 上电夹指不动版本使用, 用于控制夹爪移动
注意	仅用于控制上电夹指不动 EFG-20 电动夹爪, 上电位置为 0
函数定义	<pre>int set_efg_state(int type, float distance)</pre>
参数	<ul style="list-style-type: none"> int type: 传入 20 float distance: type==20: 实际行程 (mm), 范围正负 40
返回	<ul style="list-style-type: none"> -1: 控制参数发生变化 0: type 参数错误 1: 设置成功 3: 未初始化;

3.2.2.46 new_stop_move

功能	立即结束目前正在执行的指令
注意	相比 stop_move , 优化了停止算法 , 建议搭配

	new_movej_xyz_lr 使用
函数定义	<code>void new_stop_move()</code>
参数	无
返回	无

3.2.2.47 new_movej_angle

功能	以 movej 的运动方式到达目标点
注意	<ul style="list-style-type: none">move_flag 等于 false 时，进行首次调用，返回 1 后，可以重复调用该函数设定新的目标点位距离当前目标点有一定距离时，并且当前目标点 roughly 大于 0，再次设定新的目标，机械臂将以一定的速度在当前目标点附近通过，而后去往新的目标点。
函数定义	<pre>int new_movej_angle(float angle1, float angle2, float goal_z, float goal_r, float speed, float roughly)</pre>
参数	<ul style="list-style-type: none">float angle1：目标点轴 1 角度坐标 (deg)float angle2：目标点轴 2 角度坐标 (deg)float goal_z：目标点 z 坐标 (mm)

- **float goal_r**: 目标点 r 坐标 (deg)
- **float speed**: xyz 点的线速度 (mm/s) 或轴 4 的旋转角速度 (deg/s)
- **float roughly**: 0 先运动到前目标点，并且在当前目标点的速度等于 0，而后在去往新的目标点，1 在有新目标点时，将在当前目标点附近以最大速度通过，此参数取值范围 0 或 1

- 0: 正在执行其他指令，本次指令无效
- 1: 本次指令生效，机械臂开始运动
- 2: 设置速度小于等于零
- 3: 未初始化
- 4: 目标点无法到达

返回

- 6: 伺服未开启
- 11: 手机端在控制
- 99: 急停中
- 101: 传入参数 NOT A NUMBER
- 102: 发生碰撞，本次指令无效
- 103: 轴发生复位，需要重新初始化，本次指令无效



HITBOT

慧灵科技(深圳)有限公司
Huiling-tech Robotic Co.,Ltd

电话 0755-36382405
邮箱 hitbot@hitbot.cc
网址 www.hitbot.cc
地址 广东省深圳市宝安区西乡街道南昌社区航城大道
华丰国际机器人产业园B栋六层601-605

