

大然机器人

智能一体化关节

C 语言库函数说明



大然机器人

Dr Robot

2024 年 8 月 12 日修订

目录

第一章 简介.....	3
1.1 简介.....	3
第二章 设置关节 ID 编号	4
2.1 设置关节 ID 编号函数: set_id()	4
第三章 运动控制.....	5
3.1 单个关节绝对角度控制函数: set_angle()	5
3.2 多个关节绝对角度控制函数: set_angles().....	5
3.3 单个关节相对角度控制函数: step_angle()	6
3.4 多个关节相对角度控制函数: step_angles().....	7
3.5 单个关节力位混合（自适应）绝对角度控制函数: set_angle_adaptive()	8
3.6 多个关节力位混合（自适应）绝对角度控制函数: set_angles_adaptive()	9
3.7 单个关节阻抗控制函数: impedance_control().....	9
3.8 多个关节阻抗控制函数: impedance_control_multi()	10
3.9 单个关节运动助力函数: motion_aid()	11
3.10 多个关节运动助力函数: motion_aid_multi().....	11
3.11 检查并等待单个关节转动到目标角度函数: position_done().....	12
3.12 检查并等待多个关节转动到目标角度函数: positions_done()	13
3.13 单个关节转速控制函数: set_speed().....	13
3.14 多个关节转速控制函数: set_speeds()	14
3.15 单个关节力矩控制函数: set_torque().....	14
3.16 多个关节力矩控制函数: set_torques()	15
3.17 急停函数: estop ().....	16
第四章 参数回读.....	17
4.1 回读单个关节 ID 编号函数: get_id().....	17
4.2 回读多个关节 ID 编号函数: get_ids()	17
4.3 回读关节当前角度函数: get_angle().....	17
4.4 回读关节当前转速函数: get_speed()	18
4.5 同时回读关节当前角度和转速函数: get_state()	18
4.6 回读关节当前输出力矩函数: get_torque()	18
4.7 回读总线电压和 FOC q 轴电流函数: get_vol_cur()	19
4.8 开启角度、转速、力矩实时反馈函数: enable_angle_speed_torque_state ()	19
4.9 设置角度、转速、力矩反馈时间间隔函数: set_state_feedback_rate_ms().....	20
4.10 单个关节角度、转速、力矩实时反馈函数: angle_speed_torque_state ()	20
4.11 多个关节角度、转速、力矩实时反馈函数: angle_speed_torque_states ()	21
4.12 关闭角度、转速、力矩实时反馈函数: disable_angle_speed_torque_state ()	21
4.13 回读控制环位置增益 P、积分增益 I、转速增益 D 函数: get_pid()	22
4.14 回读配置参数函数: read_property().....	22
第五章 参数设置.....	24
5.1 设置当前角度为零点（断电重启后丢失）函数: set_zero_position_temp()	24
5.2 设置当前角度为零点（断电重启后不丢失）函数: set_zero_position()	24
5.3 设置本次运行期间关节角度限位函数: set_angle_range()	24
5.4 关闭本次运行期间关节角度限位函数: disable_angle_range().....	25

5.5	设置关节角度限位属性函数: set_angle_range_config()	25
5.6	关闭关节角度限位属性函数: disable_angle_range_config()	26
5.7	设置本次运行期间转速限制函数: set_speed_limit()	27
5.8	设置本次运行期间力矩限制函数: set_torque_limit()	27
5.9	设置本次运行期间力位混合（自适应）转速函数: set_speed_adaptive()	28
5.10	设置本次运行期间力位混合（自适应）力矩函数: set_torque_adaptive()	28
5.11	设置本次运行期间位置增益 P、积分增益 I、转速增益 D 函数: set_pid()	29
5.12	设置关节待机或闭环控制函数: set_mode()	29
5.13	设置 CAN 通信波特率函数: set_can_baud_rate()	30
5.14	写入关节配置参数函数: write_property()	30
5.15	保存关节配置参数函数: save_config()	31
第六章 辅助功能.....		32
6.1	重启关节函数: reboot()	32
6.2	恢复出厂设置函数: init_config()	32
附录 开放参数表.....		33



第一章 简介

1.1 简介

DrEmpower 智能一体化关节作为一款智能驱动器，不仅满足运动学控制的各项需求，同时也可满足动力学控制实践，使得用其开发的机器人产品更加贴近应用水平。

关节产品包括谐波减速器系列（HSA）、行星减速器系列（PDA）、中空系列（HCA）。具备运动智能特点，具备四大技能，包括运动控制、参数回读、参数设置和自主决策。

运动控制方面，除角度、转速、力矩控制外，还可进行运动助力控制、力位混合（自适应）控制、阻抗控制、急停控制。更多控制请见后续详情；

参数回读方面，可实时回读角度、转速、力矩。回读控制相关参数，如 PID 等。回读环境参数，如温度。更多参数请见后续详情；

参数设置方面，控制环中所有参数均可设置，也包括一些功能的使能设置。更多参数请见后续详情；

自主决策方面，包括碰撞感知、堵转/过流/温度保护、角度/转速/力矩限制等；

第二章 设置关节 ID 编号

2.1 设置关节 ID 编号函数: set_id()

设置关节 ID 编号函数 set_id(), 改变一体化关节 ID 编号（一次设定，关机后依然保存）。其原型及参数解释如表 2-1 和表 2-2。

表2-1 设置关节 ID 编号函数原型说明

函数原型说明	
函数名	set_id
函数原型	void set_id(uint8_t id_num, int new_id)
返回值	无

表2-2 设置关节 ID 编号函数参数解释说明

参数	解释
id_num	需要重新设置 ID 编号的一体化关节的 ID 编号，如果不知道当前一体化关节 ID 编号，可以用 0 广播。但是这时总线上只能连一个一体化关节，否则多个一体化关节会被设置成相同编号。
new_id	新一体化关节编号，一体化关节 ID 号范围为 1~63 内整数。

特别提醒：

- (1) 使用一体化关节前请先将其设置独有的 ID 号，以免在总线中出现相同 ID 号的多个关节，造成通信混乱；
- (2) 该函数最好在正式使用关节之前使用，预先设置 ID 以便确定控制目标。

第三章 运动控制

3.1 单个关节绝对角度控制函数: set_angle()

单个关节角度控制函数 `set_angle()` 用于控制指定 ID 编号的一体化关节按照指定的转速转动到指定的角度（绝对角度，相对于用户设定的零点角度）。其原型及参数解释如表 3-1 和表 3-2。

表3-1 单个关节角度控制函数原型说明

函数原型说明	
函数名	<code>set_angle</code>
函数原型	<code>void set_angle(uint8_t id_num, float angle, float speed, float param, int mode)</code>
返回值	无。

表3-2 单个关节角度控制函数参数解释说明

参数	解释
<code>id_num</code>	一体化关节 ID 编号，如果不知道当前一体化关节 ID，可以用 0 广播，此时如果总线上有多个一体化关节，则多个一体化关节都会执行该操作。
<code>angle</code>	一体化关节绝对角度（°）。
<code>speed</code>	一体化关节转速（r/min），具体含义由 <code>mode</code> 的取值决定， <code>mode=1</code> : 目标转速； <code>mode=0/2</code> : 前馈转速。
<code>param</code>	运动参数，由 <code>mode</code> 取值决定， <code>mode=0</code> : 角度输入滤波带宽（<300）； <code>mode=1</code> : 启动和停止阶段角加转速（r/min/s）； <code>mode=2</code> : 前馈力矩 <code>torque</code> （Nm）。
<code>mode</code>	角度控制模式，一体化关节支持三种角度控制模式，由 <code>mode</code> 取值决定， <code>mode = 0</code> : 轨迹跟踪模式，适合多个轨迹点输入后进行平滑控制，角度输入滤波带宽参数需设置为指令发送频率的一半； <code>mode = 1</code> : 梯形轨迹模式，这种模式下可以指定运动过程中的目标转速和启停加转速； <code>mode = 2</code> : 前馈控制模式，这种模式下的 <code>speed</code> 和 <code>torque</code> 分别为前馈控制量。前馈控制在原有 PID 控制基础上加入转速和力矩前馈，提高系统的响应特性和减少静态误差。

注：在 `mode=1` 梯形轨迹模式中，`speed` 和 `accel` 都要大于 0；`mode=0` 时 `speed` 不起作用。

3.2 多个关节绝对角度控制函数: set_angles()

多个关节绝对角度控制函数: `set_angles()`，用于控制多个编号的一体化关节按照一定转速转动到指定的角度。其原型及参数解释如表 3-3 和表 3-4。

表3-3多个关节绝对角度控制函数原型说明

函数原型说明	
函数名	set_angles
函数原型	void set_angles(uint8_t *id_list, float *angle_list, float speed, float param, int mode, size_t n)
返回值	无。

表3-4多个关节绝对角度控制函数参数解释说明

参数	解释
id_list	一体化关节 ID 编号组成的列表。
angle_list	目标角度 (°) 组成的列表。
speed	指定转速 (r/min)，由 mode 取值决定： mode=1，多个一体化关节中转速最大者的转速； mode=0/2，或前馈转速。
param	运动参数，由 mode 取值决定： mode=0，角度输入滤波带宽 (<300)； mode=1，启动和停止阶段加转速 (r/min/s)； mode=2，前馈力矩 (Nm)。
mode	控制模式选择，一体化关节支持三种角度控制模式，由 mode 取值决定： mode=0：多个一体化关节轨迹跟踪模式，适合多个轨迹点输入后进行平滑控制，角度输入带宽参数需设置为指令发送频率的一半； mode=1：多个一体化关节梯形轨迹模式，此时 speed 为这些一体化关节的最快转速 (r/min)，param 为目标加转速 (r/min/s)； mode=2：前馈控制模式，这种模式下的 speed 和 torque 分别为前馈控制量。前馈控制在原有 PID 控制基础上加入转速和力矩前馈，提高系统的响应特性和减少静态误差。
n	总线上关节的数量

3.3 单个关节相对角度控制函数：step_angle()

单个关节相对角度控制函数 step_angle()，控制指定编号的一体化关节按照指定的转速相对转动指定的角度（相对角度，相对于发送该指令时的角度）。其原型及参数解释如表 3-5 和表 3-6。

表3-5 单个关节相对角度控制函数原型说明

函数原型说明	
函数名	step_angle
函数原型	void step_angle(uint8_t id_num, float angle, float speed, float param, int mode)
返回值	无。

表3-6 单个关节相对角度控制函数参数解释说明

参数	解释
id_num	一体化关节 ID 编号, 如果不知道当前一体化关节 ID, 可以用 0 广播, 此时如果总线上有多个一体化关节, 则多个一体化关节都会执行该操作。
angle	一体化关节相对角度 (°)
speed	指定转速 (r/min), 由 mode 取值决定: mode=1, 目标转速; mode=0/2, 前馈转速 (r/min)。
param	运动参数, 由 mode 取值决定, mode=0, 角度输入滤波带宽 (<300); mode=1, 启动和停止阶段加转速 (r/min/s); mode=2, 前馈力矩 torque (Nm)。
mode	角度控制模式, 一体化关节支持三种角度控制模式, 由 mode 取值决定, mode=0: 轨迹跟踪模式, 适合多个轨迹点输入后进行平滑控制, 角度输入滤波带宽参数需设置为指令发送频率的一半; mode=1: 梯形轨迹模式, 这种模式下可以指定运动过程中的目标转速和启停加转速; mode=2: 前馈控制模式, 这种模式下的 speed 和 torque 分别为前馈控制量, 前馈控制在原有 PID 控制基础上加入转速和力矩前馈, 提高系统的响应特性和减少静态误差。

3.4 多个关节相对角度控制函数: step_angles()

多个关节相对角度控制函数 step_angles(), 控制多个一体化关节按照指定的时间先对转动给定角度。其原型及参数解释如表 3-7 和表 3-8。

表3-7 多个关节相对角度控制函数原型说明

函数原型说明	
函数名	step_angles
函数原型	void step_angles(uint8_t *id_list, float *angle_list, float speed, float param, int mode, size_t n)
返回值	无。

表3-8 多个关节相对角度控制函数参数解释说明

参数	解释
id_list	一体化关节 ID 编号组成的列表。
angle_list	相对目标角度组成的列表。
speed	指定转速 (r/min), 由 mode 取值决定: mode=1, 目标转速; mode=0/2, 前馈转速 (r/min)。
param	运动参数, 由 mode 取值决定,

参数	解释
id_list	一体化关节 ID 编号组成的列表。
mode	mode=0, 角度输入滤波带宽 (<300) ; mode=1, 启动和停止阶段加转速 (r/min/s) ; mode=2, 前馈力矩 torque (Nm) 。
n	总线上关节的数量

3.5 单个关节力位混合（自适应）绝对角度控制函数: set_angle_adaptive()

单个关节力位混合（自适应）绝对角度控制函数 `set_angle_adaptive()`, 控制指定 ID 编号的一体化关节按照限定的转速和力矩转动到指定的角度(绝对角度, 相对于用户设定的零点角度)。注: 当设置的转速相对于力矩过大, 或力矩相对于转速过小, 则关节无法在短时间内提供足够的加速度使得转速将为 0, 此时若关节未遇阻力会出现在目标角度过冲现象, 此为物理规律, 暂时没有好的解决办法。其原型及参数解释如表 3-9 和表 3-10。

表3-9 单个关节力位混合（自适应）绝对角度控制函数原型说明

函数原型说明	
函数名	<code>set_angle_adaptive</code>
函数原型	<code>void set_angle_adaptive(uint8_t id_num, float angle, float speed, float torque)</code>
返回值	无。

表3-10 单个关节力位混合（自适应）绝对角度控制函数参数解释说明

参数	解释
id_num	一体化关节 ID 编号, 如果不知道当前一体化关节 ID, 可以用 0 广播, 此时如果总线上有多个一体化关节, 则多个一体化关节都会执行该操作。
angle	一体化关节角度 (°)。
speed	限定转速值 (r/min)。
torque	限定力矩值 (Nm)。

3.6 多个关节力位混合（自适应）绝对角度控制函数：set_angles_adaptive()

多个关节力位混合（自适应）绝对角度控制函数：set_angles_adaptive()，控制多个一体化关节按照限定的转速和力矩转动到指定的角度（绝对角度，相对于用户设定的零点角度）。注：当设置的转速相对于力矩过大，或力矩相对于转速过小，则关节无法在短时间内提供足够的加速度使得转速将为 0，此时若关节未遇阻力会出现在目标角度过冲现象，此为物理规律，暂时没有好的解决办法。其原型及参数解释如表 3-11 和表 3-12。

表3-11 多个关节力位混合（自适应）绝对角度控制函数原型说明

函数原型说明	
函数名	set_angles_adaptive
函数原型	void set_angles_adaptive(uint8_t id_list[], float angle_list[], float speed_list[], float torque_list[], size_t n)
返回值	无。

表3-12 多个关节力位混合（自适应）绝对角度控制函数参数解释说明

参数	解释
id_list	一体化关节 ID 编号组成的列表。
angle_list	一体化关节角度（°）组成的列表。
speed_list	限定转速值（r/min）组成的列表。
torque_list	限定力矩值（Nm）组成的列表。
n	总线上关节的数量

3.7 单个关节阻抗控制函数：impedance_control()

单个关节阻抗控制函数 impedance_control()，对指定 ID 编号的一体化关节进行阻抗控制。该函数执行结束后关节会停在目标角度 angle，并对外表现出一定柔性。其原型及参数解释表 3-13 和表 3-14。

表3-13 单个关节阻抗控制函数原型说明

函数原型说明	
函数名	impedance_control
函数原型	void impedance_control(uint8_t id_num, float pos, float vel, float tff, float kp, float kd, int mode)
返回值	无。

表3-14 单个关节阻抗控制函数参数解释说明

参数	解释
id_num	一体化关节 ID 编号，如果不知道当前一体化关节 ID，可以用 0 广播，此时如果总线上有多个一体化关节，则多个一体化关节都会执行该操作。

参数	解释
id_num	一体化关节 ID 编号, 如果不知道当前一体化关节 ID, 可以用 0 广播, 此时如果总线上有多个一体化关节, 则多个一体化关节都会执行该操作。
angle	一体化关节目标角度 (°)。
speed	一体化关节目标转速 (r/min)。
tff	前馈力矩 (Nm)。
kp	角度刚度系数 (Nm/°), 需大于 0。
kd	转速阻尼系数 (Nm/(r/min)), 需大于 0。
mode	模式选择, 等于 1 则以角度为控制目标, 等于 0 则以力矩为控制目标

注: 该函数直接控制关节输出力矩, 其目标输出力矩计算公式如下:

$$\text{torque} = \text{kp} * (\text{angle} - \text{angle}_-) + \text{tff} + \text{kd} * (\text{speed} - \text{speed}_-)$$

其中 angle_ 和 speed_ 分别为输出轴当前实际角度 (°) 和当前实际转速 (r/min), kp 和 kd 为刚度系数和阻尼系数。

3.8 多个关节阻抗控制函数: impedance_control_multi()

多个关节阻抗控制函数 impedance_control_multi(), 对多个一体化关节进行阻抗控制。该函数执行结束后被指定的各个关节会停在目标角度 angle, 并对外表现出一定柔性。其原型及参数解释如表 3-15 和表 3-16。

表3-15 多个关节阻抗控制函数原型说明

函数原型说明	
函数名	impedance_control_multi
函数原型	void impedance_control_multi(uint8_t id_list[], float angle_list[], float speed_list[], float tff_list[], float kp_list[], float kd_list[], size_t n)
返回值	无。

表3-16 多个关节阻抗控制函数参数解释说明

参数	解释
id_list	一体化关节 ID 编号组成的列表。
angle_list	一体化关节目标角度组成的列表 (°)。
speed_list	一体化关节目标转速组成的列表 (r/min)。
tff_list	前馈力矩组成的列表 (Nm)。
kp_list	角度刚度系数组成的列表 (Nm/°), 每个元素均需大于 0。
kd_list	转速阻尼系数组成的列表 (Nm/(r/min)), 每个元素均需大于 0。
mode	模式选择, 等于 1 则以角度为控制目标, 等于 0 则以力矩为控制目标
n	总线上关节的数量

注: 该函数直接控制关节输出力矩, 其目标输出力矩计算公式如下:

```
torque = kp_list[i] * (angle_list[i] - angle_[i]) + tff_list[i] + kd_list[i] * (speed_list[i] - speed_[i])
```

其中 `angle_[i]` 和 `speed_[i]` 分别为对应关节输出轴当前实际角度（度）和当前实际转速（r/min），`kp_list[i]` 和 `kd_list[i]` 为刚度系数和阻尼系数。

3.9 单个关节运动助力函数： motion_aid()

单个关节运动助力函数 `motion_aid()`，指定 ID 编号的一体化关节进行运动助力。当关节在停止状态下检测到角度差 `angle_err` 和转速差 `speed_err` 时向目标角度 `angle` 方向提供力矩大小为 `torque` 的助力，并在到达 `angle` 后停止并保持位置。其原型及参数解释如表 3-17 和表 3-18。

表3-17 单个关节运动助力函数原型说明

函数原型说明	
函数名	<code>motion_aid</code>
函数原型	<code>void motion_aid(uint8_t id_num, float angle, float speed, float angle_err, float speed_err, float torque)</code>
返回值	无。

表3-18 单个关节运动助力函数参数解释说明

参数	解释
<code>id_num</code>	一体化关节 ID 编号，如果不知道当前一体化关节 ID，可以用 0 广播，此时总线上有多个一体化关节，则多个一体化关节都会执行该操作。
<code>angle</code>	助力目标角度（°），该值减去关节当前角度即为助力行程。
<code>speed</code>	限定转速（r/min），即助力的限定转速，防止助力力矩引起的加速导致转速过快
<code>angle_err</code>	角度差值（°），表示运动助力的角度灵敏度。
<code>speed_err</code>	转速差值（r/min），表示运动助力的转速灵敏度。
<code>torque</code>	助力力矩（Nm）。

注：

- a、当助力与外部驱动力之和大于阻力，关节会持续转动；
- b、当助力与外部驱动力之和小于阻力，关键开始减速，当转速小于 2 倍转速差值 `speed_err` 时，关节停止输出助力；
- c、一般情况下，该功能为人进行助力，强烈建议用户将助力力矩设置在人力所能及的范围内，即人力可使关节停止转动；
- d、若必须设置超出人力的力矩，则必须在合理位置设置牢固的机械限位，以避免超出运动范围给人或物体带来损伤。

3.10 多个关节运动助力函数： motion_aid_multi()

多个关节运动助力函数 `motion_aid_multi()`，指定多个一体化关节进行运动助力。当关节在停止状态下检测到角度差 `angle_err_list[i]` 和转速差 `speed_err_list[i]`

时向目标角度 angle_list[i] 方向提供力矩大小为 torque_list[i] 的助力，并在到达 angle_list[i] 后停止并保持位置。其原型及参数解释如表 3-19 和表 3-20。

表3-19 多个关节运动助力函数原型说明

函数原型说明	
函数名	motion_aid_multi
函数原型	void motion_aid_multi(uint8_t id_list[], float angle_list[], float speed_list[], float angle_err_list[], float speed_err_list[], float torque_list[], size_t n)
返回值	无。

表3-20 多个关节运动助力函数参数解释说明

参数	解释
id_list	一体化关节 ID 编号组成的列表。
angle_list	助力目标角度组成的列表（°），该值减去关节当前角度即为助力行程。
speed_list	限定转速组成的列表（r/min），即助力的限定转速，防止助力力矩引起的加速导致转速过快。
angle_err_list	角度差值（°）组成的列表，表示运动助力的角度灵敏度。
speed_err_list	转速差值（r/min）组成的列表，表示运动助力的转速灵敏度。
torque_list	助力力矩（Nm）组成的列表。
n	总线上关节的数量

注：

- a、当助力与外部驱动力之和大于阻力，关节会持续转动；
- b、当助力与外部驱动力之和小于阻力，关键开始减速，当转速小于 2 倍转速差值 speed_err_list[i] 时，关节停止输出助力；
- c、一般情况下，该功能为人进行助力，强烈建议用户将助力力矩设置在人力所能及的范围内，即人力可使关节停止转动；
- d、若必须设置超出人力的力矩，则必须在合理位置设置牢固的机械限位，以避免超出运动范围给人或物体带来损伤。

3.11 检查并等待单个关节转动到目标角度函数：position_done()

检查并等待单个关节转动到目标角度函数 position_done()，检查并等待单个一体化关节是否转动到指定角度。其原型及参数解释如表 3-21 和表 3-22。

表3-21 检查并等待单个关节转动到目标角度函数原型说明

函数原型说明	
函数名	position_done
函数原型	void position_done(uint8_t id_num)
返回值	无。

表3-22 检查并等待单个关节转动到目标角度函数原型说明

参数	解释
id_num	一体化关节 ID 编号, 如果不知道当前一体化关节 ID, 可以用 0 广播, 此时如果总线上有多个一体化关节, 则多个一体化关节都会执行该操作。

3.12 检查并等待多个关节转动到目标角度函数: **positions_done()**

检查并等待多个关节转动到目标角度函数 **positions_done()**, 检查并等待单个一体化关节是否转动到指定角度。其原型及参数解释如表 3-23 和表 3-24。

表3-23 检查并等待多个关节转动到目标角度函数原型说明

函数原型说明	
函数名	positions_done
函数原型	void positions_done(uint8_t *id_list,size_t n)
返回值	无。

表3-24 检查并等待多个关节转动到目标角度函数原型说明

参数	解释
id_list	一体化关节 ID 编号组成的列表。
n	总线上关节的数量

3.13 单个关节转速控制函数: **set_speed()**

单个关节转速控制函数 **set_speed()**, 控制指定 ID 编号的一体化关节按照指定的转速连续整周转动 (转动到关节支持的极限角度后自动停止)。其原型及参数解释如表 3-25 和表 3-26。

表3-25 单个关节转速控制函数原型说明

函数原型说明	
函数名	set_speed
函数原型	void set_speed(uint8_t id_num, float speed, float param, int mode)
返回值	无。

表3-26 单个关节转速控制函数原型说明

参数	解释
id_num	一体化关节 ID 编号, 如果不知道当前一体化关节 ID, 可以用 0 广播, 此时如果总线上有多个一体化关节, 则多个一体化关节都会执行该操作。
speed	目标转速 (r/min)。

param	运动参数，由 mode 取值决定： mode=0，前馈力矩 (Nm); mode!=0，目标加转速 (r/min/s)。
mode	控制模式选择，由 mode 取值决定： mode=0，转速直接控制模式，将一体化关节目标转速直接设为 speed; mode!=0，匀加速控制模式，一体化关节将按照目标角加速变化到 speed。

3.14 多个关节转速控制函数：set_speeds()

多个关节转速控制函数 set_speeds()，控制多个一体化关节按照指定的转速连续整周转动（转动到关节支持的极限角度后自动停止）。其原型及参数解释如表 3-27 和表 3-28。

表3-27 多个关节转速控制函数原型说明

函数原型说明	
函数名	set_speeds
函数原型	void set_speeds(uint8_t *id_list, float *speed_list, float param, float mode, size_t n)
返回值	无。

表3-28 多个关节转速控制函数原型说明

参数	解释
id_list	一体化关节 ID 编号组成的列表。
speed_list	一体化关节目标转速 (r/min) 组成的列表。
param	运动参数，由 mode 取值决定： mode=0，前馈力矩 (Nm); mode!=0，目标加转速 (r/min/s)。
mode	控制模式选择，由 mode 取值决定： mode=0，转速直接控制模式，将一体化关节目标转速直接设为 speed; mode!=0，匀加速控制模式，一体化关节将按照目标角加速变化到 speed。
n	总线上关节的数量

3.15 单个关节力矩控制函数：set_torque()

单个关节力矩控制函数 set_torque()，控制指定 ID 编号的一体化关节输出指定的力矩 (Nm)，若阻力不足以抵抗该力矩，则关节会持续转动（转动到关节支持的极限角度后自动停止）。其原型及参数解释如表 3-29 和表 3-30。

表3-29 单个关节力矩控制函数原型说明

函数原型说明	
函数名	set_torque

函数原型说明	
函数原型	void set_torque(uint8_t id_num, float torque, float param, int mode)
返回值	无。

表3-30 单个关节力矩控制函数原型说明

参数	解释
id_num	一体化关节 ID 编号, 如果不知道当前一体化关节 ID, 可以用 0 广播, 此时如果总线上有多个一体化关节, 则多个一体化关节都会执行该操作。
torque	目标力矩 (Nm)。
param	运动参数, 由 mode 取值决定: mode=0, 参数不起作用; mode!=0, 力矩在单位时间内的增量 (Nm/s)。
mode	控制模式选择, 由 mode 取值决定: mode=0, 力矩直接控制模式, 将一体化关节目标转速直接设为 torque; mode!=0, 力矩匀速增加模式, 一体化关节将按照指定的单位时间内的增量匀速变化到 torque。

3.16 多个关节力矩控制函数: set_torques()

多个关节力矩控制函数 set_torques(), 控制多个一体化关节输出指定的力矩, 若阻力不足以抵抗该力矩, 则关节会持续转动 (转动到关节支持的极限角度后自动停止)。其原型及参数解释如表 3-31 和表 3-32。

表3-31 单个关节力矩控制函数原型说明

函数原型说明	
函数名	set_torques
函数原型	void set_torques(uint8_t *id_list, float *torque_list, float param, int mode, size_t n)
返回值	无。

表3-32 单个关节力矩控制函数原型说明

参数	解释
id_list	一体化关节 ID 编号组成的列表。
torque_list	一体化关节目标力矩 (Nm) 组成的列表。
param	运动参数, 由 mode 取值决定: mode=0, 参数不起作用; mode!=0, 力矩在单位时间内的增量 (Nm/s)。
mode	控制模式选择, 由 mode 取值决定: mode=0, 力矩直接控制模式, 将一体化关节目标转速直接设为 torque; mode!=0, 力矩匀速增加模式, 一体化关节将按照指定的单位时间内的增量匀速变化到 torque。

n	总线上关节的数量
---	----------

3.17 急停函数: estop()

急停函数 estop(), 控制一体化关节紧急停止。如果想控制多个关节同时急停，则可使 id_num=0。其原型及参数解释如表 3-33 和表 3-34。

表3-33 关节急停函数原型说明

函数原型说明	
函数名	estop
函数原型	void estop(uint8_t id_num)
返回值	无。

表3-34 关节急停函数原型说明

参数	解释
id_num	一体化关节 ID 编号。如果不知道当前一体化关节 ID，可以用 0 广播，此时如果总线上有多个一体化关节，则多个一体化关节都会执行该操作。

第四章 参数回读

4.1 回读单个关节 ID 编号函数: get_id()

回读关节 ID 编号函数 `get_id()`, 读取一体化关节 ID 编号。注意使用该函数时总线上只能接 1 个一体化关节。其原型及参数解释如表 4-1 和表 4-2。

表4-1 回读关节 ID 编号函数原型说明

函数原型说明	
函数名	<code>get_id</code>
函数原型	<code>uint8_t get_id(uint8_t id_num)</code>
返回值	id: 关节 ID 编号。

表4-2 回读关节 ID 编号函数参数解释说明

参数	解释
<code>id_num</code>	通常为 0

4.2 回读多个关节 ID 编号函数: get_ids()

`get_ids()`, 用

函数原型说明	

4.3 回读关节当前角度函数: get_angle()

回读关节当前角度函数 `get_angle()`, 读取一体化关节当前角度, 单位为度(°)。其原型及参数解释如表 4-5 和表 4-6。

表4-5 回读关节当前角度函数原型说明

函数原型说明	
函数名	<code>get_angle</code>
函数原型	<code>float get_angle(uint8_t id_num)</code>
返回值	angle: 当前关节角度 (°)。

表4-6 回读关节当前角度函数参数解释说明

参数	解释
id_num	一体化关节 ID 编号, 如果不知道当前一体化关节 ID 编号, 可以用 0 广播。但此时如果总线上有多个一体化关节, 会造成总线通信干扰, 不可使用 0 广播。

4.4 回读关节当前转速函数: `get_speed()`

回读关节当前转速函数 `get_speed()`, 读取一体化关节当前转速, 单位为转每分钟 (r/min)。其原型及参数解释如表 4-7 和表 4-8。

表4-7 回读关节当前转速函数原型说明

函数原型说明	
函数名	<code>get_speed</code>
函数原型	<code>float get_speed(uint8_t id_num)</code>
返回值	speed: 关节当前转速 (r/min)。

表4-8 回读关节当前转速函数参数解释说明

参数	解释
id_num	一体化关节 ID 编号, 如果不知道当前一体化关节 ID 编号, 可以用 0 广播。但此时如果总线上有多个一体化关节, 会造成总线通信干扰, 不可使用 0 广播。

4.5 同时回读关节当前角度和转速函数: `get_state()`

同时回读关节当前角度和转速函数 `get_state()`, 读取一体化关节当前角度和转速, 单位为度(°)和转每分钟(r/min)。其原型及参数解释如表 4-9 和表 4-10。

表4-9 同时回读关节当前角度和转速函数原型说明

函数原型说明	
函数名	<code>get_state</code>
函数原型	<code>struct servo_state get_state(uint8_t id_num)</code>
返回值	输出轴角度(°)和转速 (r/min) 组成的结构体

表4-10 同时回读关节当前角度和转速函数参数解释说明

参数	解释
id_num	一体化关节 ID 编号, 如果不知道当前一体化关节 ID 编号, 可以用 0 广播。但此时如果总线上有多个一体化关节, 会造成总线通信干扰, 不可使用 0 广播。

4.6 回读关节当前输出力矩函数: `get_torque()`

回读关节当前输出力矩函数 `get_torque()`, 读取一体化关节当前输出力矩, 单位为牛米 (Nm)。其原型及参数解释如表 4-11 和表 4-12。

表4-11 回读关节当前输出力矩函数原型说明

函数原型说明	
函数名	get_torque
函数原型	float get_torque(uint8_t id_num)
返回值	torque: 关节当前输出力矩 (Nm)。

表4-12 回读关节当前输出力矩函数参数解释说明

参数	解释
id_num	一体化关节 ID 编号, 如果不知道当前一体化关节 ID 编号, 可以用 0 广播。但此时如果总线上有多个一体化关节, 会造成总线通信干扰, 不可使用 0 广播。

4.7 回读总线电压和 FOC q 轴电流函数: `get_vol_cur()`

回读总线电压和 FOC q 轴电流函数 `get_vol_cur()`, 读取一体化关节当前电压和 q 轴电流, 单位分别为伏 (V) 和安 (A)。其原型及参数解释如表 4-13 和表 4-14。

表4-13 回读总线电压和 FOC q 轴电流函数原型说明

函数原型说明	
函数名	get_volcur
函数原型	struct servo_volcur get_volcur(uint8_t id_num)
返回值	母线电压”float vol”和 FOC q 轴电流”float cur”组成的结构体。

表4-14 回读总线电压和 FOC q 轴电流函数参数解释说明

参数	解释
id_num	一体化关节 ID 编号, 如果不知道当前一体化关节 ID 编号, 可以用 0 广播。但此时如果总线上有多个一体化关节, 会造成总线通信干扰, 不可使用 0 广播。

4.8 开启角度、转速、力矩实时反馈函数: `enable_angle_speed_torque_state()`

开启角度、转速、力矩实时反馈函数 `enable_angle_speed_torque_state()`, 用于开启关节的角度、转速、力矩通过 CAN 总线实时反馈。其原型及参数解释如表 4-15 和表 4-16。

表4-15 开启角度、转速、力矩实时反馈函数原型说明

函数原型说明	
函数名	enable_angle_speed_torque_state
函数原型	void enable_angle_speed_torque_state(uint8_t id_num)
返回值	无。

表4-16 开启角度、转速、力矩实时反馈函数参数解释说明

参数	解释
id_num	一体化关节 ID 编号, 如果不知道当前一体化关节 ID 编号, 可以用 0 广播。但此时如果总线上有多个一体化关节, 会造成总线通信干扰, 不可使用 0 广播。

4.9 设置角度、转速、力矩反馈时间间隔函数: `set_state_feedback_rate_ms()`

设置角度、转速、力矩反馈时间间隔函数: `set_state_feedback_rate_ms()`, 用于设置角度、转速、力矩反馈时间间隔, 单位为 ms, 默认为 2ms。其原型及参数解释如表 4-17 和表 4-18。

表4-17 开启角度、转速、力矩实时反馈函数原型说明

函数原型说明	
函数名	<code>set_state_feedback_rate_ms</code>
函数原型	<code>void set_state_feedback_rate_ms(uint8_t id_num, uint32_t n_ms)</code>
返回值	无。

表4-18 开启角度、转速、力矩实时反馈函数参数解释说明

参数	解释
<code>id_num</code>	一体化关节 ID 编号, 如果不知道当前一体化关节 ID 编号, 可以用 0 广播。但此时如果总线上有多个一体化关节, 会造成总线通信干扰, 不可使用 0 广播。

特别提醒: 当总线中不同 ID 号关节数量为 n 时, 建议将所有关节的将该值统一设置为 2n。

4.10 单个关节角度、转速、力矩实时反馈函数: `angle_speed_torque_state()`

单个关节角度、转速、力矩实时反馈函数 `angle_speed_torque_state()`, 用于指定单个关节的角度(°)、转速(r/min)、力矩(Nm)通过 CAN 总线实时反馈。其原型及参数解释如表 4-19 和表 4-20。

表4-19 角度、转速、力矩实时反馈函数原型说明

函数原型说明	
函数名	<code>angle_speed_torque_state</code>
函数原型	<code>struct angle_speed_torque angle_speed_torque_state(uint8_t id_num)</code>
返回值	关节的角度(°)"float"、转速(r/min) "float"、力矩(Nm) "float"组成的结构体。

表4-20 角度、转速、力矩实时反馈函数参数解释说明

参数	解释
<code>id_num</code>	一体化关节 ID 编号, 注意使用该函数时最好已经将总线中的关节设置为 1~63 号, 并且没有相同 ID 号的关节。

特别提醒:

- 由于总线实时发送数据, 读取的第一组数据可能出错, 因此该函数连续运行第二次之后才能保证数据有效性。
- 运行任何其他回读参数的函数前, 须先运行后面的 `disable_angle_speed_torque_state()` 关闭实时反馈, 否则由于总线实时占用有较大概率无法得到正确的数值。

3. 使用该函数时最好已经将总线中的关节设置为 1~63 号，并且没有相同 ID 号的关节。

4.11 多个关节角度、转速、力矩实时反馈函数: `angle_speed_torque_states()`

4.12 关闭角度、转速、力矩实时反馈函数: `disable_angle_speed_torque_state()`

关闭角度、转速、力矩实时反馈函数 `disable_angle_speed_torque_state()`, 用于关闭关节的角度、转速、力矩通过 CAN 总线实时反馈。其原型及参数解释如表 4-23 和表 4-24。

表4-23 关闭角度、转速、力矩实时反馈函数原型说明

函数原型说明	
函数名	<code>disable_angle_speed_torque_state</code>
函数原型	<code>void disable_angle_speed_torque_state(uint8_t id_num)</code>
返回值	无。

表4-24 关闭角度、转速、力矩实时反馈函数参数解释说明

参数	解释
id_num	一体化关节 ID 编号, 如果不知道当前一体化关节 ID 编号, 可以用 0 广播。但此时如果总线上有多个一体化关节, 会造成总线通信干扰, 不可使用 0 广播。

特别提醒:

- 运行任何其他回读参数的函数前, 须先运行这里的 disable_angle_speed_torque_state() 关闭实时反馈, 否则由于总线实时占用有较大概率无法得到正确的数值。

4.13 回读控制环位置增益 P、积分增益 I、转速增益 D 函数: get_pid()

回读控制环位置增益 P、积分增益 I、转速增益 D 函数: get_pid(), 读取一体化关节控制环的位置增益 P、积分增益 I、转速增益 D。其原型及参数解释如表 4-25 和表 4-26。

表4-25 回读控制环位置增益 P、积分增益 I、转速增益 D 函数原型说明

函数原型说明	
函数名	get_pid
函数原型	struct PID get_pid(uint8_t id_num)
返回值	P"float"、D"float"、I"float"组成的结构体。

表4-26 回读控制环位置增益 P、积分增益 I、转速增益 D 函数参数解释说明

参数	解释
id_num	一体化关节 ID 编号, 如果不知道当前一体化关节 ID 编号, 可以用 0 广播。但此时如果总线上有多个一体化关节, 会造成总线通信干扰, 不可使用 0 广播。

4.14 回读配置参数函数: read_property()

回读配置参数函数 read_property(), 读取一体化关节属性参数, 这里的属性参数为一体化关节控制参数, 详见附录《开放参数表》。其原型及参数解释如表 4-27 和表 4-28。

表4-27 回读配置参数函数原型说明

函数原型说明	
函数名	read_property
函数原型	float read_property(uint8_t id_num,int param_address, int param_type)
返回值	value: 返回对应属性参数的值。

表4-28 回读配置参数函数参数解释说明

参数	解释
id_num	一体化关节 ID 编号, 如果不知道当前一体化关节 ID 编号, 可以用 0 广播。但此时如果总线上有多个一体化关节, 会造成总线通信干扰, 不可使用 0 广播。
address	参数键码, 详见附录《开放参数表》

参数	解释
data_type	参数类型, 详见附录《开放参数表》



第五章 参数设置

5.1 设置当前角度为零点（断电重启后丢失）函数：set_zero_position_temp()

设置当前角度为零点（断电重启后丢失）函数：set_zero_position_temp()，设置当前角度为一体化关节输出轴零点，设置完后当前角度为 0 度。其原型及参数解释如表 5-1 和表 5-2。（中空关节无此功能）注意该函数后不可以使用 save_config()，否则会改变原定的零点位置。

表5-1 设置当前角度为零点（断电重启后丢失）函数原型说明

函数原型说明	
函数名	set_zero_position_temp
函数原型	void set_zero_position_temp(uint8_t id_num)
返回值	无。

表5-2 设置当前角度为零点（断电重启后丢失）函数参数解释说明

参数	解释
id_num	一体化关节 ID 编号，如果不知道当前一体化关节 ID，可以用 0 广播，此时如果总线上有多个一体化关节，则多个一体化关节都会执行该操作。

5.2 设置当前角度为零点（断电重启后不丢失）函数：set_zero_position()

设置当前角度为零点（断电重启后不丢失）函数：set_zero_position()，设置当前角度为一体化关节输出轴零点，设置完后当前角度为 0 度。其原型及参数解释如表 5-3 和表 5-4。注意最好在修改其他参数前使用该函数，因该函数中包含 save_config() 函数，会将其他参数一并保存。

表5-3 设置当前角度为零点（断电重启后不丢失）函数原型说明

函数原型说明	
函数原型	void set_zero_position(uint8_t id_num)
返回值	无。

表5-4 设置当前角度为零点（断电重启后不丢失）函数参数解释说明

参数	解释
id_num	一体化关节 ID 编号，如果不知道当前一体化关节 ID，可以用 0 广播，此时如果总线上有多个一体化关节，则多个一体化关节都会执行该操作。

5.3 设置本次运行期间关节角度限位函数：set_angle_range()

设置本次运行期间关节角度限位函数 set_angle_range()，设置一体化关节运行过程中的极限角度，设置成功后一体化关节的可控制的转动角度将限定在 [angle_min, angle_max] 范围内。关机重启后失效。其原型及参数解释如表 5-5 和表 5-6。

表5-5 设置本次运行期间关节角度限位函数原型说明

函数原型说明	
函数名	set_angle_range
函数原型	int8_t set_angle_range(uint8_t id_num, float angle_min, float angle_max)
返回值	1: 设置成功; 0: 设置失败。

表5-6 设置本次运行期间关节角度限位函数参数解释说明

参数	解释
id_num	一体化关节 ID 编号, 如果不知道当前一体化关节 ID, 可以用 0 广播, 此时如果总线上有多个一体化关节, 则多个一体化关节都会执行该操作。
angle_min	最小限位角度 (°)。
angle_max	最大限位角度 (°)。

注:

- a、使用该函数时输出轴角度必须在[angle_min, angle_max]范围内, 否则将设置失败;
- b、该功能只在本次开机运行过程中有效, 对应地将在关节重启后失效;
- c、关节本身还有一个极限角度属性, 默认生效范围为[-180.5°, 180.5°], 该属性不受该函数影响, 且每次开机重启均有效, 如需重新设置或取消该属性, 请使用 set_angle_range_config() 和 disable_angle_range_config(), 详见对应函数说明。

5.4 关闭本次运行期间关节角度限位函数: `disable_angle_range()`

关闭本次运行期间的关节角度限位函数 `disable_angle_range()`, 取消本次运行期间一体化关节运行过程中的角度限位。其原型及参数解释如表 5-7 和表 5-8。

表5-7 关闭本次运行期间的关节角度限位函数原型说明

函数原型说明	
函数名	disable_angle_range
函数原型	int8_t disable_angle_range(uint8_t id_num)
返回值	1: 取消成功; 0: 取消失败。

表5-8 关闭本次运行期间的关节角度限位函数参数解释说明

参数	解释
id_num	一体化关节 ID 编号, 如果不知道当前一体化关节 ID, 可以用 0 广播, 此时如果总线上有多个一体化关节, 则多个一体化关节都会执行该操作。

5.5 设置关节角度限位属性函数: `set_angle_range_config()`

设置关节角度限位属性函数 `set_angle_range_config()`, 设置一体化关节极限角度属性, 设置成功后一体化关节的可控制的转动角度将限定在[angle_min, angle_max]范围内; 如该函数执行完成后再执行 `save_config()` 函数则该设置将

会保存，每次开机重启均默认有效；如不执行 `save_config()` 函数则该设置将在关机重启后失效。其原型及参数解释如表 5-9 和表 5-10。

表5-9 设置本次运行期间关节角度限位函数原型说明

函数原型说明	
函数名	<code>set_angle_range_config</code>
函数原型	<code>int8_t set_angle_range_config(uint8_t id_num, float angle_min, float angle_max)</code>
返回值	1：设置成功；0：设置失败。

表5-10 设置本次运行期间关节角度限位函数参数解释说明

参数	解释
<code>id_num</code>	一体化关节 ID 编号，如果不知道当前一体化关节 ID，可以用 0 广播，此时如果总线上有多个一体化关节，则多个一体化关节都会执行该操作。
<code>angle_min</code>	最小限位角度（°）。
<code>angle_max</code>	最大限位角度（°）。

注：

- a、使用该函数时输出轴角度必须在`[angle_min, angle_max]`范围内，否则将设置失败；
- b、限位范围设置成功后，再执行 `save_config()` 函数，则每次开机重启后均有效；
- c、如需取消该属性影响，请使用 `disable_angle_range_config()` 函数将该属性关闭，则本次开机该属性不起作用；若随后使用 `save_config()` 则该属性将永久失去，如需找回该属性，则再次使用本函数即可。

5.6 关闭关节角度限位属性函数：`disable_angle_range_config()`

关闭关节角度限位属性函数 `disable_angle_range_config()`，取消一体化关节角度限位属性；如该函数执行完成后再执行 `save_config()` 函数则该设置将会保存，每次开机重启均默认有效；如不执行 `save_config()` 函数则该设置将在关机重启后失效。其原型及参数解释如表 5-11 和表 5-12。

表5-11 关闭关节角度限位属性函数原型说明

函数原型说明	
函数名	<code>disable_angle_range_config</code>
函数原型	<code>int8_t disable_angle_range_config(uint8_t id_num)</code>
返回值	1：取消成功；2：取消失败。

表5-12 关闭关节角度限位属性函数参数解释说明

参数	解释
<code>id_num</code>	一体化关节 ID 编号，如果不知道当前一体化关节 ID，可以用 0 广播，此时如果总线上有多个一体化关节，则多个一体化关节都会执行该操作。

5.7 设置本次运行期间转速限制函数: set_speed_limit()

设置本次运行期间转速限制函数 `set_speed_limit()`, 设置一体化关节转速限制 `speed_limit` (r/min), 此后关节转速绝对值不超过 `speed_limit`。该函数执行完转速限制在本次开机运行期间有效, 关机或重启后将失效。如决定永久保持转速限制 `speed_limit`, 请紧接着使用 `save_config` 函数 (慎用)。其原型及参数解释如表 5-13 和表 5-14。

表5-13 设置本次运行期间转速限制函数原型说明

函数原型说明	
函数名	<code>set_speed_limit</code>
函数原型	<code>void set_speed_limit(uint8_t id_num, float speed_limit)</code>
返回值	无。

表5-14 设置本次运行期间转速限制函数参数解释说明

参数	解释
<code>id_num</code>	一体化关节 ID 编号, 如果不知道当前一体化关节 ID, 可以用 0 广播, 此时如果总线上有多个一体化关节, 则多个一体化关节都会执行该操作。
<code>speed_limit</code>	转速限制 (r/min) (必须大于 0)。

特别提醒:

- 若想关机重启前取消该限制, 只需设置一个非常大的数值, 比如令 `speed_limit = 100000`。

5.8 设置本次运行期间力矩限制函数: set_torque_limit()

设置本次运行期间力矩限制函数 `set_torque_limit()`, 设置一体化关节力矩限制 `torque_limit` (Nm), 此后关节力矩绝对值不超过 `torque_limit`。该函数执行完力矩限制在本次开机运行期间有效, 关机或重启后将失效。如决定永久保持力矩限制 `torque_limit`, 请紧接着使用 `save_config` 函数 (慎用)。其原型及参数解释如表 5-15 和表 5-16。

表5-15 设置本次运行期间力矩限制函数原型说明

函数原型说明	
函数名	<code>set_torque_limit</code>
函数原型	<code>void set_torque_limit(uint8_t id_num, float torque_limit)</code>
返回值	无。

表5-16 设置本次运行期间力矩限制函数参数解释说明

参数	解释
<code>id_num</code>	一体化关节 ID 编号, 如果不知道当前一体化关节 ID, 可以用 0 广播, 此时如果总线上有多个一体化关节, 则多个一体化关节都会执行该操作。

参数	解释
torque_limit	力矩限制 (Nm) (必须大于 0)。

特别提醒：

- 若想关机重启前取消该限制，只需设置一个非常大的数值，比如令 `torque_limit = 100000`。

5.9 设置本次运行期间力位混合（自适应）转速函数：set_speed_adaptive()

设置本次运行期间力位混合（自适应）转速函数 `set_speed_adaptive()`，设置一体化关节力位混合（自适应）转速限制 `speed_adaptive` (r/min)，此后关节力位混合（自适应）转速绝对值不超过 `speed_adaptive`。该函数将在除 `set_angle_adaptive()`、`set_angles_adaptive()`、`motion_aid()`、`motion_aid_multi()` 之外的其他运动控制函数之后失效，关机或重启后也将失效。其原型及参数解释如表 5-17 和表 5-18。

表5-17 设置本次运行期间力位混合（自适应）转速函数原型说明

函数原型说明	
函数名	<code>set_speed_adaptive</code>
函数原型	<code>void set_speed_adaptive(uint8_t id_num, float speed_adaptive)</code>
返回值	无。

表5-18 设置本次运行期间力位混合（自适应）转速函数参数解释说明

参数	解释
<code>id_num</code>	一体化关节 ID 编号，如果不知道当前一体化关节 ID，可以用 0 广播，此时如果总线上有多个一体化关节，则多个一体化关节都会执行该操作。
<code>speed_adaptive</code>	力位混合（自适应）转速 (r/min) (必须大于 0)。

特别提醒：本函数需在运动指令之后运行。

5.10 设置本次运行期间力位混合（自适应）力矩函数：set_torque_adaptive()

设置本次运行期间力位混合（自适应）力矩函数 `set_torque_adaptive()`，设置一体化关节力位混合（自适应）力矩限制 `torque_adaptive` (Nm)，此后关节力位混合（自适应）转速绝对值不超过 `torque_adaptive`。该函数将在除 `set_angle_adaptive()`、`set_angles_adaptive()`、`motion_aid()`、`motion_aid_multi()` 之外的其他运动控制函数之后失效，关机或重启后也将失效。其原型及参数解释如表 5-19 和表 5-20。

表5-19 设置本次运行期间力位混合（自适应）力矩函数原型说明

函数原型说明	
函数名	<code>set_torque_adaptive</code>

函数原型说明	
函数名	set_torque_adaptive
函数原型	void set_torque_adaptive(uint8_t id_num, float torque_adaptive)
返回值	无。

表5-20 设置本次运行期间力位混合（自适应）力矩函数参数解释说明

参数	解释
id_num	一体化关节 ID 编号，如果不知道当前一体化关节 ID，可以用 0 广播，此时如果总线上有多个一体化关节，则多个一体化关节都会执行该操作。
torque_adaptive	力位混合（自适应）力矩（Nm）（必须大于 0）。

特别提醒：本函数需在运动指令之后运行。

5.11 设置本次运行期间位置增益 P、积分增益 I、转速增益 D 函数: set_pid()

设置本次运行期间位置增益 P、积分增益 I、转速增益 D 函数 set_pid()，设置一体化关节控制环的位置增益 PD、积分增益 I、转速增益，以便实现调整关节控制性能的目的。该函数执行完 PID 的值在本次开机运行期间有效，关机或重启后将失效。如决定永久使用某组 PID 则可在使用该函数设置 PID 后，紧接着使用 save_config 函数。其原型及参数解释如表 5-21 和表 5-22。

表5-21 设置本次运行期间位置增益 P、转速增益 D、积分增益 I 函数原型说明

函数原型说明	
函数名	set_pid
函数原型	void set_pid(uint8_t id_num, float P, float I, float D)
返回值	无。

表5-22 设置本次运行期间位置增益 P、积分增益 I、转速增益 D 函数参数解释说明

参数	解释
id_num	一体化关节 ID 编号，如果不知道当前一体化关节 ID，可以用 0 广播，此时如果总线上有多个一体化关节，则多个一体化关节都会执行该操作。

5.12 设置关节待机或闭环控制函数: set_mode()

设置关节模式函数 set_mode()，设置一体化关节进入不同的控制模式。其原型及参数解释如表 5-23 和表 5-24。

表5-23 设置关节模式函数原型说明

函数原型说明	
函数名	set_mode
函数原型	void set_mode(uint8_t id_num, int mode)
返回值	无。

表5-24 设置关节模式函数参数解释说明

参数	解释
id_num	一体化关节 ID 编号, 如果不知道当前一体化关节 ID, 可以用 0 广播, 此时如果总线上有多个一体化关节, 则多个一体化关节都会执行该操作。
mode	一体化关节模式编号, mode=1: 待机模式, 一体化关节卸载; mode=2: 闭环控制模式, 运动控制函数必须在闭环控制模式下才能进行控制。(一体化关节上电后的默认模式)

5.13 设置 CAN 通信波特率函数: set_can_baud_rate()

设置 CAN 通信波特率函数 `set_can_baud_rate()`, 设置 CAN 波特率; 该函数执行后不会在本次开机生效, 需接着执行 `save_config()` 函数将配置保存, 再重新开机后生效。其原型及参数解释如表 5-25 和表 5-26。

表5-25 设置 CAN 通信波特率函数原型说明

函数原型说明	
函数名	<code>set_can_baud_rate</code>
函数原型	<code>void set_can_baud_rate(uint8_t id_num, int baud_rate)</code>
返回值	无。

表5-26 设置 CAN 通信波特率函数参数解释说明

参数	解释
id_num	一体化关节 ID 编号, 如果不知道当前一体化关节 ID, 可以用 0 广播, 此时如果总线上有多个一体化关节, 则多个一体化关节都会执行该操作。
baud_rate	CAN 波特率, 支持 125k、250k、500k、1M 中任意一种, 修改成功后需手动将上位机 CAN 波特率也修改为相同值。

5.14 写入关节配置参数函数: write_property()

写入关节配置参数函数 `write_property()`, 修改一体化关节属性参数, 这里的属性参数为一体化关节控制参数, 详见附录《开放参数表》。其原型及参数解释如表 5-27 和表 5-28。

表5-27 写入关节配置参数函数原型说明

函数原型说明	
函数名	<code>write_property</code>
函数原型	<code>void write_property(uint8_t id_num, unsigned short param_address,int8_t param_type, float value)</code>
返回值	无。

表5-28 写入关节配置参数函数参数解释说明

参数	解释
id_num	一体化关节 ID 编号, 如果不知道当前一体化关节 ID, 可以用 0 广播, 此时如果总

参数	解释
	线上有多个一体化关节，则多个一体化关节都会执行该操作。
address	参数键码，详见附录《开放参数表》
data_type	参数类型，详见附录《开放参数表》
value	对应参数的目标值。

5.15 保存关节配置参数函数：save_config()

保存关节配置参数函数 `save_config()`，保存一体化关节属性参数，这里的属性参数详见附录《开放参数表》。一般通过 `write_property` 函数修改的属性一体化关节关机或重启之后，会恢复为修改前的值；如果想永久保存，需要用 `save_config` 函数将相关参数保存到 `flash` 中，关机或重启后不丢失。其原型及参数解释如表 5-29 和表 5-30。`save_config()` 函数是将所有参数整体一次性保存，所以只需将所有目标参数全部修改后再执行一次 `save_config()` 函数，便可将所有参数保存。

表5-29 保存关节配置参数函数原型说明

函数原型说明	
函数名	<code>save_config</code>
函数原型	<code>void save_config(uint8_t id_num)</code>
返回值	无。

表5-30 保存关节配置参数函数参数解释说明

参数	解释
<code>id_num</code>	一体化关节 ID 编号，如果不知道当前一体化关节 ID，可以用 0 广播，此时如果总线上有多个一体化关节，则多个一体化关节都会执行该操作。

第六章 辅助功能

6.1 重启关节函数: reboot()

重启关节函数: `reboot()`, 设置一体化关节软件重启, 效果与重新上电类似。其原型及参数解释如表 6-1 和表 6-2。

表6-1重启关节函数原型说明

函数原型说明	
函数名	<code>reboot</code>
函数原型	<code>void reboot(uint8_t id_num)</code>
返回值	无。

表6-2重启关节函数参数解释说明

参数	解释
<code>id_num</code>	一体化关节 ID 编号, 如果不知道当前一体化关节 ID 编号, 可以用 0 广播, 此时如果总线上有多个一体化关节, 则多个一体化关节都会执行该操作。

6.2 恢复出厂设置函数: init_config()

恢复出厂设置函数 `init_config()`, 恢复出厂时的参数配置, 不改变用户配置的 ID 号。其原型及参数解释如表 6-3 和表 6-4。

表6-3恢复出厂设置函数原型说明

函数原型说明	
函数名	<code>init_config</code>
函数原型	<code>void init_config(id_num=0)</code>
返回值	无。

表6-4恢复出厂设置函数参数解释说明

参数	解释
<code>id_num</code>	一体化关节 ID 编号, 如果不知道当前一体化关节 ID 编号, 可以用 0 广播, 此时如果总线上有多个一体化关节, 则多个一体化关节都会执行该操作。

附录 开放参数表

注，对于以下参数：

- (1) 读写列 1 表示可读可写，0 表示只读；
- (2) 名称中带 config 的参数均可由 save_config() 函数保存并在重启后依然有效；
- (3) 使用 read_property() 函数读取；
- (4) 使用 write_property() 函数改写
- (5) 数据类型代号：float (0); uint16 (1); int16 (2); uint32 (3); int32 (4)。

序号	含义	名称	单位	默认值	类型	键码	读写	备注
1	电压	dr.voltage	V		float	1	0	
2	电流	dr.i	A		float	2	0	
3	CAN 总线波特率	dr.can.config.baud_rate	bit/s	1M	uint32	21001	1	
4	是否开启 CAN 实时状态反馈	dr.can.enable_state_feedback		0	uint32	22001	1	1 表示开启，0 表示不开启，版本号 240627 及以后不可由 save_config() 函数保存
5	固件版本日期	dr.version_date			uint32	30004	0	固件版本号为 231207 及以后版本可用
6	CAN 总线 ID 号	dr.config.can_id		7	uint32	31001	1	取值范围：1~63
7	CAN 总线实时状态反馈时间间隔	dr.config.state_feedback_rate_ms	ms	2	uint32	31002	1	比如设置 3ms，则每隔 3ms 反馈一次状态
8	关节型号	dr.config.product_model			uint32	31003	0	参照手册中的关节型号代码
9	是否开启角度限制属性	dr.config.enable_angle_limit		0	uint32	31201	1	1 表示开启，0 表示不开启
10	最小角度限位	dr.config.angle_min	°	-180	float	31202	1	

序号	含义	名称	单位	默认值	类型	键码	读写	备注
11	最大角度限位	dr.config.angle_max	°	180	float	31203	1	
12	减速比	dr.config.gear_ratio			float	31204	0	
13	堵转电流	dr.config.stall_current_limit	A	50	float	31205	1	
14	是否开启碰撞检测	dr.config.enable_crash_detect		1	uint32	31206	1	
15	碰撞检测灵敏度	dr.config.crash_detect_sensitivity		150	float	31207	1	该值需大于 0, 越小越灵敏
16	是否开启多圈计数角度限制	dr.config.enable_encoder_circular_limit		1	uint32	31208	1	超出该限制关节将无法记住重启前的角度, 需重新设置零点。若关闭该限制则务必把编码器供电线拔掉, 即默认系统不需要关机后零点位置。 该参数只针对具有多圈计数功能关节有效。
17	是否运动到指定位置	dr.controller.position_done		1	uint32	32002	0	为 1 表示到达指定位置, 为 0 则未到达
18	判定运动到位的定位精度	dr.controller.position_precision	°	0.036	float	32003	1	固件版本号为 231207 及以后版本可用
19	是否启用转速限制	dr.controller.config.enable_speed_limit		1	uint32	32101	1	1 表示启用, 0 表示不启用
20	位置增益 P	dr.controller.config.angle_gain			float	32102	1	
21	转速增益 D	dr.controller.config.speed_gain			float	32103	1	
22	积分增益 I	dr.controller.config.speed_integrator_gain			float	32104	1	
23	最大限制转速	dr.controller.config.speed_limit	r/min	+∞	float	32105	1	该值针对电机, 输出端需乘以减速比
24	超速容忍度	dr.controller.config.speed_limit_tolerance	%	1.2	float	32106	1	比如设为 1.2 则超出 20% 后才会报错
25	负载转动惯量	dr.controller.config.inertia		0	float	32107	1	默认为 0
26	输入滤波带宽	dr.controller.config.input_filter_bandwidth		2	float	32108	1	
27	电机极对数	dr.motor.config.pole_pairs			int32	33101	0	
28	电机相电感	dr.motor.config.phase_inductance			float	33102	0	

序号	含义	名称	单位	默认值	类型	键码	读写	备注
29	电机相电阻	dr.motor.config.phase_resistance			float	33103	0	
30	电机力矩常数	dr.motor.config.torque_constant			float	33104	0	
31	最大限制电流	dr.motor.config.current_limit			float	33105	1	
32	过流容忍度	dr.motor.config.current_limit_margin	A		float	33106	1	比如设为 1.2 则超出 1.2A 后才会报错
33	最大力矩限制值	dr.motor.config.torque_limit	Nm	+∞	float	33107	1	该值对象是电机，输出端需除以减速比
34	电流控制带宽	dr.motor.config.current_control_bandwidth			float	33108	1	
35	FOC Q 轴电流	dr.motor.Iq_measured	A		float	33201	0	
36	FOC D 轴电流	dr.motor.Id_measured	A		float	33202	0	
37	设置零点后输出轴编码器零位值	dr.encoder.pos_zero_output			int32	34101	0	中空关节专属
38	刚启动时读取到的输出轴编码器值	dr.encoder.abs_pos_power_on			int32	34102	0	中空关节专属
39	绝对位置误差的前半个周期长度	dr.encoder.first_half_T			float	34103	1	中空关节专属，标定后误差补偿使用
40	绝对位置误差的前半个周期幅值	dr.encoder.first_half_M			float	34104	1	中空关节专属，标定后误差补偿使用
41	绝对位置误差的后半个周期幅值	dr.encoder.second_half_M			float	34105	1	中空关节专属，标定后误差补偿使用
42	绝对位置误差截距	dr.encoder.intercept			float	34106	1	中空关节专属，标定后误差补偿使用
43	绝对位置误差前半个周期幅值是否向上	dr.encoder.first_half_up			uint32	34107	1	中空关节专属，标定后误差补偿使用
44	驱动板温度	dr.board_temperature	°C		float	36002	0	
45	是否开启驱动板温度保护	dr.board_temperature.config.enabled		1	uint32	36101	1	
46	驱动板温度保护下限	dr.board_temperature.config.temp_limit_lower	°C	100	float	36102	1	

序号	含义	名称	单位	默认值	类型	键码	读写	备注
47	驱动板温度保护上限	dr.board_temperature.config.temp_limit_upper	°C	120	float	36103	1	
48	电机温度	dr.motor_temperature	°C		float	37002	0	
49	是否开启电机温度保护	dr.motor_temperature.config.enabled		1	uint32	37101	1	
50	电机温度保护下限	dr.motor_temperature.config.temp_limit_lower	°C	80	float	37102	1	
51	电机温度保护上限	dr.motor_temperature.config.temp_limit_upper	°C	100	float	37103	1	
52	输出轴角度	dr.output_shaft.angle	°		float	38001	0	
53	输出轴转速	dr.output_shaft.speed	r/min		float	38002	0	
54	输出轴力矩	dr.output_shaft.torque	Nm		float	38003	0	
55	输出轴最小临时角度限位	dr.output_shaft.angle_min	°	-180	float	38004	1	
56	输出轴最大临时角度限位	dr.output_shaft.angle_max	°	180	float	38005	1	
57	是否开启临时角度限位	dr.output_shaft.enable_angle_limit		0	uint32	38006	1	

DR TECH
大然机器人