

# ODriveTool 0.5.1.post0 指令大全

## 版本说明

ODriveTool 版本: **0.5.1.post0** (pip install odrive==0.5.1.post0)

ODrive 硬件版本: [v3.6-24V](#) [v3.6-56V](#)

ODrive 硬件内部固件版本: [fw-v0.5.1](#)

## 指令前缀说明

- [RO] 只读属性
  - 读: 直接输入指令回车即可返回属性值
- [RW] 可读可写属性
  - 读: 直接输入指令回车即可返回当前属性值
  - 写: 输入 `指令 = [值]` `Enter` 即可对属性进行赋值操作。如:  
`odrv0.config.brake_resistance = 2` `Enter`
- [OP] 执行函数
  - 输入指令和函数输入参数回车即可执行。如: `odrv0.get_adc_voltage(2)` `Enter`

## ODriveTool 0.5.1.post0 指令大全

### odrv0

[RO] `odrv0.vbus_voltage`  
[RO] `odrv0.ibus`  
[RW] `odrv0.ibus_report_filter_k`  
[RO] `odrv0.serial_number`  
[RO] `odrv0.hw_version_major`  
[RO] `odrv0.hw_version_minor`  
[RO] `odrv0.hw_version_variant`  
[RO] `odrv0.fw_version_major`  
[RO] `odrv0.fw_version_minor`  
[RO] `odrv0.fw_version_revision`  
[RO] `odrv0.fw_version_unreleased`  
[RO] `odrv0.brake_resistor_armed`  
[RO] `odrv0.brake_resistor_saturated`  
[RO] `odrv0.user_config_loaded`  
[RW] `odrv0.test_property`  
[OP] `odrv0.test_function(delta: int32_t)`  
[OP] `odrv0.get_adc_voltage(gpio: int)`  
[OP] `odrv0.save_configuration()`  
[OP] `odrv0.erase_configuration()`  
[OP] `odrv0.reboot()`  
[OP] `odrv0.enter_dfu_mode()`  
`odrv0.system_stats`  
[RO] `odrv0.system_stats.uptime`  
[RO] `odrv0.system_stats.min_heap_space`  
[RO] `odrv0.system_stats.min_stack_space_axis0`  
[RO] `odrv0.system_stats.min_stack_space_axis1`  
[RO] `odrv0.system_stats.min_stack_space_comms`  
[RO] `odrv0.system_stats.min_stack_space_usb`  
[RO] `odrv0.system_stats.min_stack_space_uart`  
[RO] `odrv0.system_stats.min_stack_space_can`

- [RO] odrv0.system\_stats.min\_stack\_space\_usb\_irq
- [RO] odrv0.system\_stats.min\_stack\_space\_startup
- [RO] odrv0.system\_stats.stack\_usage\_axis0
- [RO] odrv0.system\_stats.stack\_usage\_axis1
- [RO] odrv0.system\_stats.stack\_usage\_comms
- [RO] odrv0.system\_stats.stack\_usage\_usb
- [RO] odrv0.system\_stats.stack\_usage\_uart
- [RO] odrv0.system\_stats.stack\_usage\_can
- [RO] odrv0.system\_stats.stack\_usage\_usb\_irq
- [RO] odrv0.system\_stats.stack\_usage\_startup
- odrv0.system\_stats.usb
  - [RO] odrv0.system\_stats.usb.rx\_cnt
  - [RO] odrv0.system\_stats.usb.tx\_cnt
  - [RO] odrv0.system\_stats.usb.tx\_overrun\_cnt
- odrv0.system\_stats.i2c
  - [RO] odrv0.system\_stats.i2c.addr
  - [RO] odrv0.system\_stats.i2c.addr\_match\_cnt
  - [RO] odrv0.system\_stats.i2c.rx\_cnt
  - [RO] odrv0.system\_stats.i2c.error\_cnt
- odrv0.can
  - [RW] odrv0.can.error
- odrv0.can.config
  - [RO] odrv0.can.config.baud\_rate
  - [RW] odrv0.can.config.protocol
  - [OP] odrv0.can.set\_baud\_rate(baudRate: uint32)
- odrv0.config
  - [RW] odrv0.config.enable\_uart
  - [RW] odrv0.config.uart\_baudrate
  - [RW] odrv0.config.enable\_i2c\_instead\_of\_can
  - [RW] odrv0.config.enable\_ascii\_protocol\_on\_usb
  - [RW] odrv0.config.max\_regen\_current
  - [RW] odrv0.config.brake\_resistance
  - [RW] odrv0.config.dc\_bus\_undervoltage\_trip\_level
  - [RW] odrv0.config.dc\_bus\_overvoltage\_trip\_level
  - [RW] odrv0.config.enable\_dc\_bus\_overvoltage\_ramp
  - [RW] odrv0.config.dc\_bus\_overvoltage\_ramp\_start
  - [RW] odrv0.config.dc\_bus\_overvoltage\_ramp\_end
  - [RW] odrv0.config.dc\_max\_positive\_current
  - [RW] odrv0.config.dc\_max\_negative\_current
- odrv0.config.gpio1\_pwm\_mapping
  - [RW] odrv0.config.gpio1\_pwm\_mapping.endpoint
  - [RW] odrv0.config.gpio1\_pwm\_mapping.min
  - [RW] odrv0.config.gpio1\_pwm\_mapping.max
- odrv0.config.gpio2\_pwm\_mapping
  - [RW] odrv0.config.gpio2\_pwm\_mapping.endpoint
  - [RW] odrv0.config.gpio2\_pwm\_mapping.min
  - [RW] odrv0.config.gpio2\_pwm\_mapping.max
- odrv0.config.gpio3\_pwm\_mapping
  - [RW] odrv0.config.gpio3\_pwm\_mapping.endpoint
  - [RW] odrv0.config.gpio3\_pwm\_mapping.min
  - [RW] odrv0.config.gpio3\_pwm\_mapping.max
- odrv0.config.gpio4\_pwm\_mapping
  - [RW] odrv0.config.gpio4\_pwm\_mapping.endpoint
  - [RW] odrv0.config.gpio4\_pwm\_mapping.min
  - [RW] odrv0.config.gpio4\_pwm\_mapping.max
- odrv0.config.gpio3\_analog\_mapping
  - [RW] odrv0.config.gpio3\_analog\_mapping.endpoint
  - [RW] odrv0.config.gpio3\_analog\_mapping.min

- [RW] odrv0.config.gpio3\_analog\_mapping.max
- odrv0.config.gpio4\_analog\_mapping
  - [RW] odrv0.config.gpio4\_analog\_mapping.endpoint
  - [RW] odrv0.config.gpio4\_analog\_mapping.min
  - [RW] odrv0.config.gpio4\_analog\_mapping.max
- odrv0.axis0
  - [RW] odrv0.axis0.error
  - [RO] odrv0.axis0.step\_dir\_active
  - [RO] odrv0.axis0.current\_state
  - [RW] odrv0.axis0.requested\_state
  - [RO] odrv0.axis0.loop\_counter
  - [RO] odrv0.axis0.lockin\_state
  - [RO] odrv0.axis0.is\_homed
  - [OP] odrv0.axis0.watchdog\_feed()
  - [OP] odrv0.axis0.clear\_errors()
  - odrv0.axis0.config
    - [RW] odrv0.axis0.config.startup\_motor\_calibration
    - [RW] odrv0.axis0.config.startup\_encoder\_index\_search
    - [RW] odrv0.axis0.config.startup\_encoder\_offset\_calibration
    - [RW] odrv0.axis0.config.startup\_closed\_loop\_control
    - [RW] odrv0.axis0.config.startup\_sensorless\_control
    - [RW] odrv0.axis0.config.startup\_homing
    - [RW] odrv0.axis0.config.enable\_step\_dir
    - [RW] odrv0.axis0.config.step\_dir\_always\_on
    - [RW] odrv0.axis0.config.turns\_per\_step
    - [RW] odrv0.axis0.config.watchdog\_timeout
    - [RW] odrv0.axis0.config.enable\_watchdog
    - [RW] odrv0.axis0.config.step\_gpio\_pin
    - [RW] odrv0.axis0.config.dir\_gpio\_pin
    - [RW] odrv0.axis0.config.can\_node\_id
    - [RW] odrv0.axis0.config.can\_node\_id\_extended
    - [RW] odrv0.axis0.config.can\_heartbeat\_rate\_ms
  - odrv0.axis0.config.calibration\_lockin
    - [RW] odrv0.axis0.config.calibration\_lockin.current
    - [RW] odrv0.axis0.config.calibration\_lockin.ramp\_time
    - [RW] odrv0.axis0.config.calibration\_lockin.ramp\_distance
    - [RW] odrv0.axis0.config.calibration\_lockin.accel
    - [RW] odrv0.axis0.config.calibration\_lockin.vel
  - odrv0.axis0.config.sensorless\_ramp
    - [RW] odrv0.axis0.config.sensorless\_ramp.current
    - [RW] odrv0.axis0.config.sensorless\_ramp.ramp\_time
    - [RW] odrv0.axis0.config.sensorless\_ramp.ramp\_distance
    - [RW] odrv0.axis0.config.sensorless\_ramp.accel
    - [RW] odrv0.axis0.config.sensorless\_ramp.vel
    - [RW] odrv0.axis0.config.sensorless\_ramp.finish\_distance
    - [RW] odrv0.axis0.config.sensorless\_ramp.finish\_on\_vel
    - [RW] odrv0.axis0.config.sensorless\_ramp.finish\_on\_distance
    - [RW] odrv0.axis0.config.sensorless\_ramp.finish\_on\_enc\_idx
  - odrv0.axis0.config.general\_lockin
    - [RW] odrv0.axis0.config.general\_lockin.current
    - [RW] odrv0.axis0.config.general\_lockin.ramp\_time
    - [RW] odrv0.axis0.config.general\_lockin.ramp\_distance
    - [RW] odrv0.axis0.config.general\_lockin.accel
    - [RW] odrv0.axis0.config.general\_lockin.vel
    - [RW] odrv0.axis0.config.general\_lockin.finish\_distance
    - [RW] odrv0.axis0.config.general\_lockin.finish\_on\_vel
    - [RW] odrv0.axis0.config.general\_lockin.finish\_on\_distance
    - [RW] odrv0.axis0.config.general\_lockin.finish\_on\_enc\_idx

```

odrv0.axis0.fet_thermistor
[RW] odrv0.axis0.fet_thermistor.error
[RO] odrv0.axis0.fet_thermistor.temperature
odrv0.axis0.fet_thermistor.config
[RW] odrv0.axis0.fet_thermistor.config.temp_limit_lower
[RW] odrv0.axis0.fet_thermistor.config.temp_limit_upper
[RW] odrv0.axis0.fet_thermistor.config.enabled
odrv0.axis0.motor_thermistor
[RW] odrv0.axis0.motor_thermistor.error
[RO] odrv0.axis0.motor_thermistor.temperature
odrv0.axis0.motor_thermistor.config
[RW] odrv0.axis0.motor_thermistor.config.gpio_pin
[RW] odrv0.axis0.motor_thermistor.config.poly_coefficient_0
[RW] odrv0.axis0.motor_thermistor.config.poly_coefficient_1
[RW] odrv0.axis0.motor_thermistor.config.poly_coefficient_2
[RW] odrv0.axis0.motor_thermistor.config.poly_coefficient_3
[RW] odrv0.axis0.motor_thermistor.config.temp_limit_lower
[RW] odrv0.axis0.motor_thermistor.config.temp_limit_upper
[RW] odrv0.axis0.motor_thermistor.config.enabled
odrv0.axis0.motor
[RW] odrv0.axis0.motor.error
[RO] odrv0.axis0.motor.armed_state
[RO] odrv0.axis0.motor.is_calibrated
[RO] odrv0.axis0.motor.current_meas_phB
[RO] odrv0.axis0.motor.current_meas_phC
[RW] odrv0.axis0.motor.DC_calib_phB
[RW] odrv0.axis0.motor.DC_calib_phC
[RW] odrv0.axis0.motor.phase_current_rev_gain
[RO] odrv0.axis0.motor.effective_current_lim
odrv0.axis0.motor.current_control
[RW] odrv0.axis0.motor.current_control.p_gain
[RW] odrv0.axis0.motor.current_control.i_gain
[RW] odrv0.axis0.motor.current_control.v_current_control_integral_d
[RW] odrv0.axis0.motor.current_control.v_current_control_integral_q
[RW] odrv0.axis0.motor.current_control.lbus
[RW] odrv0.axis0.motor.current_control.final_v_alpha
[RW] odrv0.axis0.motor.current_control.final_v_beta
[RW] odrv0.axis0.motor.current_control.Id_setpoint
[RW] odrv0.axis0.motor.current_control.Iq_setpoint
[RW] odrv0.axis0.motor.current_control.Iq_measured
[RW] odrv0.axis0.motor.current_control.Id_measured
[RW] odrv0.axis0.motor.current_control.I_measured_report_filter_k
[RO] odrv0.axis0.motor.current_control.max_allowed_current
[RO] odrv0.axis0.motor.current_control.overcurrent_trip_level
[RW] odrv0.axis0.motor.current_control.acim_rotor_flux
[RO] odrv0.axis0.motor.current_control.async_phase_vel
[RW] odrv0.axis0.motor.current_control.async_phase_offset
odrv0.axis0.motor.gate_driver
[RO] odrv0.axis0.motor.gate_driver.drv_fault
odrv0.axis0.motor.config
[RW] odrv0.axis0.motor.config.pre_calibrated
[RW] odrv0.axis0.motor.config.pole_pairs
[RW] odrv0.axis0.motor.config.calibration_current
[RW] odrv0.axis0.motor.config.resistance_calib_max_voltage
[RW] odrv0.axis0.motor.config.phase_inductance
[RW] odrv0.axis0.motor.config.phase_resistance
[RW] odrv0.axis0.motor.config.direction
[RW] odrv0.axis0.motor.config.motor_type

```

- [RW] odrv0.axis0.motor.config.current\_lim
- [RW] odrv0.axis0.motor.config.current\_lim\_margin
- [RW] odrv0.axis0.motor.config.torque\_lim
- [RW] odrv0.axis0.motor.config.requested\_current\_range
- [RW] odrv0.axis0.motor.config.current\_control\_bandwidth
- [RW] odrv0.axis0.motor.config.acim\_slip\_velocity
- [RW] odrv0.axis0.motor.config.acim\_gain\_min\_flux
- [RW] odrv0.axis0.motor.config.acim\_autoflux\_min\_id
- [RW] odrv0.axis0.motor.config.acim\_autoflux\_enable
- [RW] odrv0.axis0.motor.config.acim\_autoflux\_attack\_gain
- [RW] odrv0.axis0.motor.config.acim\_autoflux\_decay\_gain
- odrv0.axis0.controller
  - [RW] odrv0.axis0.controller.error
  - [RW] odrv0.axis0.controller.input\_pos
  - [RW] odrv0.axis0.controller.input\_vel
  - [RW] odrv0.axis0.controller.input\_torque
  - [RO] odrv0.axis0.controller.pos\_setpoint
  - [RO] odrv0.axis0.controller.vel\_setpoint
  - [RO] odrv0.axis0.controller.torque\_setpoint
  - [RO] odrv0.axis0.controller.trajectory\_done
  - [RW] odrv0.axis0.controller.vel\_integrator\_torque
  - [RW] odrv0.axis0.controller.anticogging\_valid
  - [OP] odrv0.axis0.controller.move\_incremental(displacement: float, from\_input\_pos: bool)
  - [OP] odrv0.axis0.controller.start\_anticogging\_calibration()
  - odrv0.axis0.controller.config
    - [RW] odrv0.axis0.controller.config.gain\_scheduling\_width
    - [RW] odrv0.axis0.controller.config.enable\_gain\_scheduling
    - [RW] odrv0.axis0.controller.config.enable\_vel\_limit
    - [RW] odrv0.axis0.controller.config.enable\_current\_mode\_vel\_limit
    - [RW] odrv0.axis0.controller.config.enable\_overspeed\_error
    - [RW] odrv0.axis0.controller.config.control\_mode
    - [RW] odrv0.axis0.controller.config.input\_mode
    - [RW] odrv0.axis0.controller.config.pos\_gain
    - [RW] odrv0.axis0.controller.config.vel\_gain
    - [RW] odrv0.axis0.controller.config.vel\_integrator\_gain
    - [RW] odrv0.axis0.controller.config.vel\_limit
    - [RW] odrv0.axis0.controller.config.vel\_limit\_tolerance
    - [RW] odrv0.axis0.controller.config.vel\_ramp\_rate
    - [RW] odrv0.axis0.controller.config.torque\_ramp\_rate
    - [RW] odrv0.axis0.controller.config.circular\_setpoints
    - [RW] odrv0.axis0.controller.config.circular\_setpoint\_range
    - [RW] odrv0.axis0.controller.config.homing\_speed
    - [RW] odrv0.axis0.controller.config.inertia
    - [RW] odrv0.axis0.controller.config.axis\_to\_mirror
    - [RW] odrv0.axis0.controller.config.mirror\_ratio
    - [RW] odrv0.axis0.controller.config.input\_filter\_bandwidth
    - odrv0.axis0.controller.config.anticogging
      - [RO] odrv0.axis0.controller.config.anticogging.index
      - [RW] odrv0.axis0.controller.config.anticogging.pre\_calibrated
      - [RO] odrv0.axis0.controller.config.anticogging.calib\_anticogging
      - [RW] odrv0.axis0.controller.config.anticogging.calib\_pos\_threshold
      - [RW] odrv0.axis0.controller.config.anticogging.calib\_vel\_threshold
      - [RO] odrv0.axis0.controller.config.anticogging.cogging\_ratio
      - [RW] odrv0.axis0.controller.config.anticogging.anticogging\_enabled
- odrv0.axis0.encoder
  - [RW] odrv0.axis0.encoder.error
  - [RO] odrv0.axis0.encoder.is\_ready
  - [RO] odrv0.axis0.encoder.index\_found

- [RO] odrv0.axis0.encoder.shadow\_count
- [RO] odrv0.axis0.encoder.count\_in\_cpr
- [RO] odrv0.axis0.encoder.interpolation
- [RO] odrv0.axis0.encoder.phase
- [RO] odrv0.axis0.encoder.pos\_estimate
- [RO] odrv0.axis0.encoder.pos\_estimate\_counts
- [RO] odrv0.axis0.encoder.pos\_cpr
- [RO] odrv0.axis0.encoder.pos\_cpr\_counts
- [RO] odrv0.axis0.encoder.pos\_circular
- [RO] odrv0.axis0.encoder.hall\_state
- [RO] odrv0.axis0.encoder.vel\_estimate
- [RO] odrv0.axis0.encoder.vel\_estimate\_counts
- [RO] odrv0.axis0.encoder.calib\_scan\_response
- [RW] odrv0.axis0.encoder.pos\_abs
- [RO] odrv0.axis0.encoder.spi\_error\_rate
- odrv0.axis0.encoder.config
  - [RW] odrv0.axis0.encoder.config.mode
  - [RW] odrv0.axis0.encoder.config.use\_index
  - [RW] odrv0.axis0.encoder.config.find\_idx\_on\_lockin\_only
  - [RW] odrv0.axis0.encoder.config.abs\_spi\_cs\_gpio\_pin
  - [RW] odrv0.axis0.encoder.config.zero\_count\_on\_find\_idx
  - [RW] odrv0.axis0.encoder.config.cpr
  - [RW] odrv0.axis0.encoder.config.offset
  - [RW] odrv0.axis0.encoder.config.pre\_calibrated
  - [RW] odrv0.axis0.encoder.config.offset\_float
  - [RW] odrv0.axis0.encoder.config.enable\_phase\_interpolation
  - [RW] odrv0.axis0.encoder.config.bandwidth
  - [RW] odrv0.axis0.encoder.config.calib\_range
  - [RW] odrv0.axis0.encoder.config.calib\_scan\_distance
  - [RW] odrv0.axis0.encoder.config.calib\_scan\_omega
  - [RW] odrv0.axis0.encoder.config.idx\_search\_unidirectional
  - [RW] odrv0.axis0.encoder.config.ignore\_illegal\_hall\_state
  - [RW] odrv0.axis0.encoder.config.sincos\_gpio\_pin\_sin
  - [RW] odrv0.axis0.encoder.config.sincos\_gpio\_pin\_cos
  - [OP] odrv0.axis0.encoder.set\_linear\_count(count: int32)
- odrv0.axis0.sensorless\_estimator
  - [RW] odrv0.axis0.sensorless\_estimator.error
  - [RW] odrv0.axis0.sensorless\_estimator.phase
  - [RW] odrv0.axis0.sensorless\_estimator pll\_pos
  - [RW] odrv0.axis0.sensorless\_estimator.vel\_estimate
  - odrv0.axis0.sensorless\_estimator.config
    - [RW] odrv0.axis0.sensorless\_estimator.config.observer\_gain
    - [RW] odrv0.axis0.sensorless\_estimator.config.pll\_bandwidth
    - [RW] odrv0.axis0.sensorless\_estimator.config.pm\_flux\_linkage
- odrv0.axis0.trap\_traj
  - odrv0.axis0.trap\_traj.config
    - [RW] odrv0.axis0.trap\_traj.config.vel\_limit
    - [RW] odrv0.axis0.trap\_traj.config.accel\_limit
    - [RW] odrv0.axis0.trap\_traj.config.decel\_limit
- odrv0.axis0.min\_endstop
  - [RO] odrv0.axis0.min\_endstop.endstop\_state
  - odrv0.axis0.min\_endstop.config
    - [RW] odrv0.axis0.min\_endstop.config gpio\_num
    - [RW] odrv0.axis0.min\_endstop.config.enabled
    - [RW] odrv0.axis0.min\_endstop.config.offset
    - [RW] odrv0.axis0.min\_endstop.config.is\_active\_high
    - [RW] odrv0.axis0.min\_endstop.config.pullup
    - [RW] odrv0.axis0.min\_endstop.config.debounce\_ms

```
odrv0.axis0.max_endstop
  [RO] odrv0.axis0.max_endstop.endstop_state
  odrv0.axis0.max_endstop.config
    [RW] odrv0.axis0.max_endstop.config.gpio_num
    [RW] odrv0.axis0.max_endstop.config.enabled
    [RW] odrv0.axis0.max_endstop.config.offset
    [RW] odrv0.axis0.max_endstop.config.is_active_high
    [RW] odrv0.axis0.max_endstop.config.pullup
    [RW] odrv0.axis0.max_endstop.config.debounce_ms
odrv0.axis1 (参考 axis0)
```

## odrv0

---

### [RO] odrv0.vbus\_voltage

---

类型为 [float]，单位为 [V]，当前 DC 总线上的电压。

### [RO] odrv0.ibus

---

类型为 [float]，单位为 [A]，当前 DC 总线上的电流。

当此值为正时表示 ODrive 正在消耗能量，此值为负时表示 ODrive 正在反向向电源回冲能量。

此值等于流过电机的电流加上流过制动电阻的电流，流过电机的电流是 ODrive 测量得到的，流过制动电阻的电流是 ODrive 基于制动电阻阻值(odrv0.config.brake\_resistance)计算得到的。

### [RW] odrv0.ibus\_report\_filter\_k

---

类型为 [float]，DC 总线上的电流(odrv0.ibus)滤波增益。

设置范围为 0~1，当设置为小于1时 odrv0.ibus 将更加平滑，越接近 0 得到的曲线越平滑，但是相应的滞后越严重，当设置为1时将不使用滤波。这个参数只影响 odrv0.ibus 的输出，对于 ODrive 内部电流计算没有任何影响。

### [RO] odrv0.serial\_number

---

类型为 [uint64\_t]，ODrive 硬件序列号。

### [RO] odrv0.hw\_version\_major

---

类型为 [uint8\_t]，ODrive 硬件主版本号。

### [RO] odrv0.hw\_version\_minor

---

类型为 [uint8\_t]，ODrive 硬件次版本号。

### [RO] odrv0.hw\_version\_variant

---

类型为 [uint8\_t]，ODrive 硬件变种版本，用来细分硬件电压版本，如 56v 电压版本此值为 56，24v 电压版本此值为 24。

## [RO] odrv0.fw\_version\_major

---

类型为 [uint8\_t]，固件主版本号。

## [RO] odrv0.fw\_version\_minor

---

类型为 [uint8\_t]，固件次版本号。

## [RO] odrv0.fw\_version\_revision

---

类型为 [uint8\_t]，固件修正版本号。

## [RO] odrv0.fw\_version\_unreleased

---

类型为 [uint8\_t]，固件版本是否为官方发布，如果为官方发布版本此值为 0，否则为 1。

## [RO] odrv0.brake\_resistor\_armed

---

类型为 [bool]，用来表示当前制动电阻PWM控制信号是否使能。

## [RO] odrv0.brake\_resistor\_saturated

---

类型为 [bool]，用来表示当前制动电阻控制信号是否饱和，当占空比大于等于 %95 时即认为饱和。

## [RO] odrv0.user\_config\_loaded

---

类型为 [bool]，用来查看启动后配置是否被正确加载，如果配置加载正确此值为 True，否则为 False。

## [RW] odrv0.test\_property

---

类型为 [uint32\_t]，测试用，用来测试属性的读写是否正常。

## [OP] odrv0.test\_function(delta: int32\_t)

---

测试用，用来测试函数执行是否正常，返回值为内部静态变量加上函数输入的 delta 值。

## [OP] odrv0.get\_adc\_voltage(gpio: int)

---

获取GPIO端口采集到的电压，单位为[V]，仅在 GPIO 端口 1、2、3、4 支持 ADC 输入。

## [OP] odrv0.save\_configuration()

---

将配置保存到内部 FLASH 内。

## [OP] odrv0.erase\_configuration()

---

擦除 FLASH 中的配置，下次重新启动后所有配置变为默认配置。

## [OP] odrv0.reboot()

---

重启 ODrive 硬件。



## [OP] odrv0.enter\_dfu\_mode()

---

使 ODrive 进入 DFU 模式。

## odrv0.system\_stats

---

### [RO] odrv0.system\_stats.uptime

类型为 [uint32\_t]，单位为 [ms]，系统启动后运行时间。

### [RO] odrv0.system\_stats.min\_heap\_space

类型为 [uint32\_t]，单位为 [Bytes]，获取未分配的内存堆历史最小值，用来调试堆栈分配。

### [RO] odrv0.system\_stats.min\_stack\_space\_axis0

类型为 [uint32\_t]，单位为 [Bytes]，获取 axis0 线程未分配的栈内存历史最小值，用来调试堆栈分配。

### [RO] odrv0.system\_stats.min\_stack\_space\_axis1

类型为 [uint32\_t]，单位为 [Bytes]，获取 axis1 线程未分配的栈内存历史最小值，用来调试堆栈分配。

### [RO] odrv0.system\_stats.min\_stack\_space\_comms

类型为 [uint32\_t]，单位为 [Bytes]，获取通讯线程未分配的栈内存历史最小值，用来调试堆栈分配。

### [RO] odrv0.system\_stats.min\_stack\_space\_usb

类型为 [uint32\_t]，单位为 [Bytes]，获取USB线程未分配的栈内存历史最小值，用来调试堆栈分配。

### [RO] odrv0.system\_stats.min\_stack\_space\_uart

类型为 [uint32\_t]，单位为 [Bytes]，获取UART线程未分配的栈内存历史最小值，用来调试堆栈分配。

### [RO] odrv0.system\_stats.min\_stack\_space\_can

类型为 [uint32\_t]，单位为 [Bytes]，获取CAN线程未分配的栈内存历史最小值，用来调试堆栈分配。

### [RO] odrv0.system\_stats.min\_stack\_space\_usb\_irq

类型为 [uint32\_t]，单位为 [Bytes]，获取USB IRQ线程未分配的栈内存历史最小值，用来调试堆栈分配。

### [RO] odrv0.system\_stats.min\_stack\_space\_startup

类型为 [uint32\_t]，单位为 [Bytes]，获取startup线程未分配的栈内存历史最小值，用来调试堆栈分配。

### [RO] odrv0.system\_stats.stack\_usage\_axis0

类型为 [uint32\_t]，单位为 [Bytes]，获取 axis0 线程的栈内存用量，用来调试堆栈分配。

### [RO] odrv0.system\_stats.stack\_usage\_axis1

类型为 [uint32\_t]，单位为 [Bytes]，获取 axis1 线程的栈内存用量，用来调试堆栈分配。

## **[RO] odrv0.system\_stats.stack\_usage\_comms**

类型为 [uint32\_t], 单位为 [Bytes], 获取通讯线程的栈内存用量, 用来调试堆栈分配。

## **[RO] odrv0.system\_stats.stack\_usage\_usb**

类型为 [uint32\_t], 单位为 [Bytes], 获取USB线程的栈内存用量, 用来调试堆栈分配。

## **[RO] odrv0.system\_stats.stack\_usage\_uart**

类型为 [uint32\_t], 单位为 [Bytes], 获取UART线程的栈内存用量, 用来调试堆栈分配。

## **[RO] odrv0.system\_stats.stack\_usage\_can**

类型为 [uint32\_t], 单位为 [Bytes], 获取CAN线程的栈内存用量, 用来调试堆栈分配。

## **[RO] odrv0.system\_stats.stack\_usage\_usb\_irq**

类型为 [uint32\_t], 单位为 [Bytes], 获取USB IRQ线程的栈内存用量, 用来调试堆栈分配。

## **[RO] odrv0.system\_stats.stack\_usage\_startup**

类型为 [uint32\_t], 单位为 [Bytes], 获取startup线程的栈内存用量, 用来调试堆栈分配。

## **odrv0.system\_stats.usb**

### **[RO] odrv0.system\_stats.usb.rx\_cnt**

类型为 [uint32\_t], usb 接收数据包计数。

### **[RO] odrv0.system\_stats.usb.tx\_cnt**

类型为 [uint32\_t], usb 发送数据包计数。

### **[RO] odrv0.system\_stats.usb.tx\_overrun\_cnt**

类型为 [uint32\_t], usb 发送数据包过载计数。

## **odrv0.system\_stats.i2c**

### **[RO] odrv0.system\_stats.i2c.addr**

类型为 [uint8\_t], i2c 地址。

### **[RO] odrv0.system\_stats.i2c.addr\_match\_cnt**

类型为 [uint32\_t], i2c 地址匹配计数。

### **[RO] odrv0.system\_stats.i2c.rx\_cnt**

类型为 [uint32\_t], i2c 数据包接收计数。

### **[RO] odrv0.system\_stats.i2c.error\_cnt**

类型为 [uint32\_t], i2c 错误计数。

## **odrv0.can**

---

## [RW] odrv0.can.error

类型为 [enum], can 错误标志。

```
ERROR_NONE = 0x00000000,  
ERROR_DUPLICATE_CAN_IDS = 0x00000001,
```

## odrv0.can.config

### [RO] odrv0.can.config.baud\_rate

类型为 [uint32\_t], 当前can波特率。

### [RW] odrv0.can.config.protocol

类型为 [enum], can通讯所使用的协议。

```
PROTOCOL_SIMPLE = 0
```

## [OP] odrv0.can.set\_baud\_rate(baudRate: uint32)

设置 can 通讯波特率。目前支持以下波特率：

- 125000
- 250000
- 500000
- 1000000

## odrv0.config

### [RW] odrv0.config.enable\_uart

类型为 [bool], 是否启用 uart 通讯。

**注意: 更改后需要保存并重新启动 ODrive 才能生效**

### [RW] odrv0.config uart\_baudrate

类型为 [uint32\_t], uart 通讯波特率。

### [RW] odrv0.config.enable\_i2c\_instead\_of\_can

类型为 [bool], 是否启用 i2c 通讯而不是 can 通讯, 设置为 True 使用 i2c, False 使用 can。

**注意: 更改后需要保存并重新启动 ODrive 才能生效**

### [RW] odrv0.config.enable\_ascii\_protocol\_on\_usb

类型为 [bool], 是否启用 usb 支持 ascii 协议解析。

### [RW] odrv0.config.max\_regen\_current

类型为 [float], 单位为 [A], 最大再生电流, 此值应为正值, 当反向电流高于此值时开始启动制动电阻进行反向电流的消耗。

**注意: 此值必须大于等于 0, 不能为负值, 否则制动电阻将持续消耗能量, 会烧坏制动电阻**

## [RW] odrv0.config.brake\_resistance

类型为 [float]，单位为 [ohm]，制动电阻的阻值，如果不使用或不接制动电阻设置为 0 即可。

## [RW] odrv0.config.dc\_bus\_undervoltage\_trip\_level

类型为 [float]，单位为 [V]，设置低电压报警阈值，当 DC 电压低于此值电机将停止并报错。

## [RW] odrv0.config.dc\_bus\_overvoltage\_trip\_level

类型为 [float]，单位为 [V]，设置过压报警阈值，当 DC 电压高于此值电机将停止并报错。

## [RW] odrv0.config.enable\_dc\_bus\_overvoltage\_ramp

类型为 [bool]，是否启用过压爬升特性。如果启用此特性，ODrive 将会把更多的能量消耗在制动电阻上，来降低母线上的电压。当母线电压大于 `odrv0.config.dc_bus_overvoltage_ramp_start` 时开始按比例增加制动电阻的占空比，当母线电压大于等于 `odrv0.config.dc_bus_overvoltage_ramp_end` 时制动电阻上的占空比将增加 100%。

```
* vbus_voltage == dc_bus_overvoltage_ramp_start => brake_duty_cycle += 0%
```

```
* vbus_voltage == dc_bus_overvoltage_ramp_end => brake_duty_cycle += 100%
```

**注意:** 当此特性启用时，电机不运行时依然会起作用

**注意:** 当不使用制动电阻时，此特性不起作用

## [RW] odrv0.config.dc\_bus\_overvoltage\_ramp\_start

类型为 [float]，单位为 [V]，设置过压爬升特性的开始阈值。

**注意:** 此值应高于母线供电电压，如果低于供电电压值，制动电阻将持续消耗能量，会烧坏制动电阻

## [RW] odrv0.config.dc\_bus\_overvoltage\_ramp\_end

类型为 [float]，单位为 [V]，设置过压爬升特性的结束阈值。

**注意:** 此值应高于 `odrv0.config.dc_bus_overvoltage_ramp_start`

## [RW] odrv0.config.dc\_max\_positive\_current

类型为 [float]，单位为 [A]，供电设备能够输出的最大电流。

## [RW] odrv0.config.dc\_max\_negative\_current

类型为 [float]，单位为 [A]，供电设备能够反向吸收的最大电流。此值应当为负值，当 `odrv0.ibus` 小于此值时 ODrive 将报错，当设为负无穷大时将不进行检查报错。

## odrv0.config.gpio1\_pwm\_mapping

### [RW] odrv0.config.gpio1\_pwm\_mapping.endpoint

设置 GPIO 1 作为 PWM 输入时和被控制量的映射，如果希望 PWM 输入控制 axis0 电机的转速

```
odrv0.config.gpio1_pwm_mapping.endpoint =
```

```
odrv0.axis0.controller._remote_attributes['input_vel']。
```

如果希望解除映射可以输入 `odrv0.config.gpio1_pwm_mapping.endpoint = None`。

## **[RW] odrv0.config.gpio1\_pwm\_mapping.min**

映射被控制量的最小值，当 PWM duty 为 1 ms 时，被控量为此值。

## **[RW] odrv0.config.gpio1\_pwm\_mapping.max**

映射被控制量的最大值，当 PWM duty 为 2 ms 时，被控量为此值。

## **odrv0.config.gpio2\_pwm\_mapping**

### **[RW] odrv0.config.gpio2\_pwm\_mapping.endpoint**

设置 GPIO 2 作为 PWM 输入时和被控制量的映射，如果希望 PWM 输入控制 axis0 电机的转速  
`odrv0.config.gpio2_pwm_mapping.endpoint =`  
`odrv0.axis0.controller._remote_attributes['input_vel']`。

如果希望解除映射可以输入 `odrv0.config.gpio2_pwm_mapping.endpoint = None`。

### **[RW] odrv0.config.gpio2\_pwm\_mapping.min**

映射被控制量的最小值，当 PWM duty 为 1 ms 时，被控量为此值。

### **[RW] odrv0.config.gpio2\_pwm\_mapping.max**

映射被控制量的最大值，当 PWM duty 为 2 ms 时，被控量为此值。

## **odrv0.config.gpio3\_pwm\_mapping**

### **[RW] odrv0.config.gpio3\_pwm\_mapping.endpoint**

设置 GPIO 3 作为 PWM 输入时和被控制量的映射，如果希望 PWM 输入控制 axis0 电机的转速输  
`odrv0.config.gpio3_pwm_mapping.endpoint =`  
`odrv0.axis0.controller._remote_attributes['input_vel']`。

如果希望解除映射可以输入 `odrv0.config.gpio3_pwm_mapping.endpoint = None`。

### **[RW] odrv0.config.gpio3\_pwm\_mapping.min**

映射被控制量的最小值，当 PWM duty 为 1 ms 时，被控量为此值。

### **[RW] odrv0.config.gpio3\_pwm\_mapping.max**

映射被控制量的最大值，当 PWM duty 为 2 ms 时，被控量为此值。

## **odrv0.config.gpio4\_pwm\_mapping**

### **[RW] odrv0.config.gpio4\_pwm\_mapping.endpoint**

设置 GPIO 4 作为 PWM 输入时和被控制量的映射，如果希望 PWM 输入控制 axis0 电机的转速  
`odrv0.config.gpio4_pwm_mapping.endpoint =`  
`odrv0.axis0.controller._remote_attributes['input_vel']`。

如果希望解除映射可以输入 `odrv0.config.gpio4_pwm_mapping.endpoint = None`。

### **[RW] odrv0.config.gpio4\_pwm\_mapping.min**

映射被控制量的最小值，当 PWM duty 为 1 ms 时，被控量为此值。

### **[RW] odrv0.config.gpio4\_pwm\_mapping.max**

映射被控制量的最大值，当 PWM duty 为 2 ms 时，被控量为此值。

## odrv0.config.gpio3\_analog\_mapping

### [RW] odrv0.config.gpio3\_analog\_mapping.endpoint

设置 GPIO 3 作为 模拟输入时和被控制量的映射，如果希望模拟输入控制 axis0 电机的转速  
odrv0.config.gpio3\_analog\_mapping.endpoint =  
odrv0.axis0.controller.\_remote\_attributes['input\_vel']。

如果希望解除映射可以输入 odrv0.config.gpio3\_analog\_mapping.endpoint = None。

### [RW] odrv0.config.gpio3\_analog\_mapping.min

映射被控制量的最小值，当输入电压为 0 v 时，被控量为此值。

### [RW] odrv0.config.gpio3\_analog\_mapping.max

映射被控制量的最大值，当输入电压为 3.3 v 时，被控量为此值。

## odrv0.config.gpio4\_analog\_mapping

### [RW] odrv0.config.gpio4\_analog\_mapping.endpoint

设置 GPIO 4 作为 模拟输入时和被控制量的映射，如果希望模拟输入控制 axis0 电机的转速。  
odrv0.config.gpio4\_analog\_mapping.endpoint =  
odrv0.axis0.controller.\_remote\_attributes['input\_vel']。

如果希望解除映射可以输入 odrv0.config.gpio4\_analog\_mapping.endpoint = None。

### [RW] odrv0.config.gpio4\_analog\_mapping.min

映射被控制量的最小值，当输入电压为 0 v 时，被控量为此值。

### [RW] odrv0.config.gpio4\_analog\_mapping.max

映射被控制量的最大值，当输入电压为 3.3 v 时，被控量为此值。

## odrv0.axis0

### [RW] odrv0.axis0.error

类型为 [enum]，axis0 错误代码。

ERROR_NONE	= 0x00000000,
ERROR_INVALID_STATE	= 0x00000001,
ERROR_DC_BUS_UNDER_VOLTAGE	= 0x00000002,
ERROR_DC_BUS_OVER_VOLTAGE	= 0x00000004,
ERROR_CURRENT_MEASUREMENT_TIMEOUT	= 0x00000008,
ERROR_BRAKE_RESISTOR_DISARMED	= 0x00000010,
ERROR_MOTOR_DISARMED	= 0x00000020,
ERROR_MOTOR_FAILED	= 0x00000040,
ERROR_SENSORLESS_ESTIMATOR_FAILED	= 0x00000080,
ERROR_ENCODER_FAILED	= 0x00000100,
ERROR_CONTROLLER_FAILED	= 0x00000200,
ERROR_POS_CTRL_DURING_SENSORLESS	= 0x00000400,
ERROR_WATCHDOG_TIMER_EXPIRED	= 0x00000800,
ERROR_MIN_ENDSTOP_PRESSED	= 0x00001000,

```
ERROR_MAX_ENDSTOP_PRESSED      = 0x00002000,
ERROR_ESTOP_REQUESTED          = 0x00004000,
ERROR_HOMING_WITHOUT_ENDSTOP    = 0x00020000,
ERROR_OVER_TEMP                 = 0x00040000,
```

## [RO] odrv0.axis0.step\_dir\_active

类型为 [bool], step/dir 是否使能。

## [RO] odrv0.axis0.current\_state

类型为 [enum], axis0 当前状态机。

```
AXIS_STATE_UNDEFINED           = 0,
AXIS_STATE_IDLE                = 1,
AXIS_STATE_STARTUP_SEQUENCE    = 2,
AXIS_STATE_FULL_CALIBRATION_SEQUENCE = 3,
AXIS_STATE_MOTOR_CALIBRATION   = 4,
AXIS_STATE_SENSORLESS_CONTROL  = 5,
AXIS_STATE_ENCODER_INDEX_SEARCH = 6,
AXIS_STATE_ENCODER_OFFSET_CALIBRATION = 7,
AXIS_STATE_CLOSED_LOOP_CONTROL = 8,
AXIS_STATE_LOCKIN_SPIN         = 9,
AXIS_STATE_ENCODER_DIR_FIND     = 10,
AXIS_STATE_HOMING              = 11,
```

## [RW] odrv0.axis0.requested\_state

类型为 [enum], 命令 axis0 进入某个状态。

```
AXIS_STATE_UNDEFINED           = 0,
AXIS_STATE_IDLE                = 1,
AXIS_STATE_STARTUP_SEQUENCE    = 2,
AXIS_STATE_FULL_CALIBRATION_SEQUENCE = 3,
AXIS_STATE_MOTOR_CALIBRATION   = 4,
AXIS_STATE_SENSORLESS_CONTROL  = 5,
AXIS_STATE_ENCODER_INDEX_SEARCH = 6,
AXIS_STATE_ENCODER_OFFSET_CALIBRATION = 7,
AXIS_STATE_CLOSED_LOOP_CONTROL = 8,
AXIS_STATE_LOCKIN_SPIN         = 9,
AXIS_STATE_ENCODER_DIR_FIND     = 10,
AXIS_STATE_HOMING              = 11,
```

## [RO] odrv0.axis0.loop\_counter

类型为 [uint32\_t], axis0 内部循环计数。

## [RO] odrv0.axis0.lockin\_state

类型为 [enum], 电机锁定运行状态。

```
LOCKIN_STATE_INACTIVE      = 0,
LOCKIN_STATE_RAMP          = 1,
LOCKIN_STATE_ACCELERATE    = 2,
LOCKIN_STATE_CONST_VEL     = 3,
```

## **[RO] odrv0.axis0.is\_homed**

类型为 [bool]，是否已找回零点。

## **[OP] odrv0.axis0.watchdog\_feed()**

axis0 看门狗喂狗。

## **[OP] odrv0.axis0.clear\_errors()**

清除 axis0 上的错误。

## **odrv0.axis0.config**

### **[RW] odrv0.axis0.config.startup\_motor\_calibration**

类型为 [bool]，ODrive 上电后是否自动进行电机校准。

### **[RW] odrv0.axis0.config.startup\_encoder\_index\_search**

类型为 [bool]，ODrive 上电后是否自动进行编码器索引搜索，只有在启用编码器索引信号时才生效。

### **[RW] odrv0.axis0.config.startup\_encoder\_offset\_calibration**

类型为 [bool]，ODrive 上电后是否自动进行编码器偏移校准。

### **[RW] odrv0.axis0.config.startup\_closed\_loop\_control**

类型为 [bool]，ODrive 上电后是否自动进入闭环控制。

### **[RW] odrv0.axis0.config.startup\_sensorless\_control**

类型为 [bool]，ODrive 上电后是否自动进入无感控制模式。

### **[RW] odrv0.axis0.config.startup\_homing**

类型为 [bool]，ODrive 上电后是否自动进行找零点。

### **[RW] odrv0.axis0.config.enable\_step\_dir**

类型为 [bool]，是否启用 step/dir 输入控制。

**注意：**当 odrv.config.enable\_uart 为 True 时 M0 电机无法使用 step/dir 控制

### **[RW] odrv0.axis0.config.step\_dir\_always\_on**

类型为 [bool]，step/dir 输入控制在电机处于非闭环状态下是否依然有效。

### **[RW] odrv0.axis0.config.turns\_per\_step**

类型为 [float]，step/dir 每步对应电机转动圈数。

### **[RW] odrv0.axis0.config.watchdog\_timeout**



类型为 [float], 单位为 [s], axis0 内部看门狗超时时间设置。

### **[RW] odrv0.axis0.config.enable\_watchdog**

类型为 [bool], 是否启用 axis0 内部看门狗。

### **[RW] odrv0.axis0.config.step\_gpio\_pin**

类型为 [uint16\_t], axis0 step 对应的 GPIO。

### **[RW] odrv0.axis0.config.dir\_gpio\_pin**

类型为 [uint16\_t], axis0 dir 对应的 GPIO。

### **[RW] odrv0.axis0.config.can\_node\_id**

类型为 [uint32\_t], axis0 的 can 通讯端点 ID。

### **[RW] odrv0.axis0.config.can\_node\_id\_extended**

类型为 [bool], axis0 的 can 通讯是否为扩展帧。

### **[RW] odrv0.axis0.config.can\_heartbeat\_rate\_ms**

类型为 [uint32\_t], 单位为 [ms], axis0 的 can 通讯发送心跳时间间隔。

## **odrv0.axis0.config.calibration\_lockin**

### **[RW] odrv0.axis0.config.calibration\_lockin.current**

类型为 [float], 单位为 [A], 电机开环旋转校准编码器时的电机电流。

### **[RW] odrv0.axis0.config.calibration\_lockin.ramp\_time**

类型为 [float], 单位为 [s], 电机开环旋转校准编码器时电流爬升到设置的电机电流所需的时间。

### **[RW] odrv0.axis0.config.calibration\_lockin.ramp\_distance**

类型为 [float], 单位为 [rad], 电机开环旋转校准编码器时转速爬升的距离。

### **[RW] odrv0.axis0.config.calibration\_lockin.accel**

类型为 [float], 单位为 [rad/s<sup>2</sup>], 电机开环旋转校准编码器时转速爬升的加速度。

### **[RW] odrv0.axis0.config.calibration\_lockin.vel**

类型为 [float], 单位为 [rad/s], 电机开环旋转校准编码器时设定的旋转速度。

## **odrv0.axis0.config.sensorless\_ramp**

### **[RW] odrv0.axis0.config.sensorless\_ramp.current**

类型为 [float], 单位为 [A], 电机无感模式开环启动的电机电流。

### **[RW] odrv0.axis0.config.sensorless\_ramp.ramp\_time**

类型为 [float], 单位为 [s], 电机无感模式开环启动的电流爬升时间。

### **[RW] odrv0.axis0.config.sensorless\_ramp.ramp\_distance**

类型为 [float], 单位为 [rad], 电机无感模式开环启动的爬升距离。

### **[RW] odrv0.axis0.config.sensorless\_ramp.accel**

类型为 [float], 单位为 [rad/s^2], 电机无感模式开环启动的加速度。

#### **[RW] odrv0.axis0.config.sensorless\_ramp.vel**

类型为 [float], 单位为 [rad/s], 电机无感模式开环启动的目标转速。

#### **[RW] odrv0.axis0.config.sensorless\_ramp.finish\_distance**

类型为 [float], 单位为 [rad], 电机无感模式开环启动的运行距离。

#### **[RW] odrv0.axis0.config.sensorless\_ramp.finish\_on\_vel**

类型为 [bool], 是否将达到目标转速作为开环启动结束标志。

#### **[RW] odrv0.axis0.config.sensorless\_ramp.finish\_on\_distance**

类型为 [bool], 是否将达到目标距离作为开环启动结束标志。

#### **[RW] odrv0.axis0.config.sensorless\_ramp.finish\_on\_enc\_idx**

类型为 [bool], 是否将检测到编码器索引信号作为开环启动结束标志。

### **odrv0.axis0.config.general\_lockin**

#### **[RW] odrv0.axis0.config.general\_lockin.current**

类型为 [float], 单位为 [A], 电机开环运行模式的电机电流。

#### **[RW] odrv0.axis0.config.general\_lockin.ramp\_time**

类型为 [float], 单位为 [s], 电机开环运行模式的电流爬升时间。

#### **[RW] odrv0.axis0.config.general\_lockin.ramp\_distance**

类型为 [float], 单位为 [rad], 电机开环运行模式的爬升距离。

#### **[RW] odrv0.axis0.config.general\_lockin.accel**

类型为 [float], 单位为 [rad/s^2], 电机开环运行模式的加速度。

#### **[RW] odrv0.axis0.config.general\_lockin.vel**

类型为 [float], 单位为 [rad/s], 电机开环运行模式的目标转速。

#### **[RW] odrv0.axis0.config.general\_lockin.finish\_distance**

类型为 [float], 单位为 [rad], 电机开环运行模式的运行距离。

#### **[RW] odrv0.axis0.config.general\_lockin.finish\_on\_vel**

类型为 [bool], 是否将达到目标转速作为开环运行的结束标志。

#### **[RW] odrv0.axis0.config.general\_lockin.finish\_on\_distance**

类型为 [bool], 是否将达到目标距离作为开环运行的结束标志。

#### **[RW] odrv0.axis0.config.general\_lockin.finish\_on\_enc\_idx**

类型为 [bool], 是否将检测到编码器索引信号作为开环运行的结束标志。

### **odrv0.axis0.fet\_thermistor**

#### **[RW] odrv0.axis0.fet\_thermistor.error**

类型为 [enum], axis0 MOSFET 温度错误标志。

ERROR_NONE	= 0x00000000,
ERROR_OVER_TEMP	= 0x00000001,

## [RO] odrv0.axis0.fet\_thermistor.temperature

类型为 [float], 单位为 [°C], 当前测量得到的 axis0 MOSFET 温度。

## odrv0.axis0.fet\_thermistor.config

### [RW] odrv0.axis0.fet\_thermistor.config.temp\_limit\_lower

类型为 [float], 单位为 [°C], 当 axis0 MOSFET 温度高于此值时 ODrive 开始减小电流输出。

### [RW] odrv0.axis0.fet\_thermistor.config.temp\_limit\_upper

类型为 [float], 单位为 [°C], 当 axis0 MOSFET 温度高于此值时 ODrive 将停止电流输出并报出电机温度过高错误。

### [RW] odrv0.axis0.fet\_thermistor.config.enabled

类型为 [bool], 是否启用 axis0 MOSFET 温度测量。

## odrv0.axis0.motor\_thermistor

### [RW] odrv0.axis0.motor\_thermistor.error

类型为 [enum], 电机温度错误标志。

ERROR_NONE	= 0x00000000,
ERROR_OVER_TEMP	= 0x00000001,

## [RO] odrv0.axis0.motor\_thermistor.temperature

类型为 [float], 单位为 [°C], 当前测量得到的电机温度。

## odrv0.axis0.motor\_thermistor.config

### [RW] odrv0.axis0.motor\_thermistor.config.gpio\_pin

类型为 [uint32\_t], 测量电机温度所使用的 GPIO。

### [RW] odrv0.axis0.motor\_thermistor.config.poly\_coefficient\_0

类型为 [float], 测量电机温度所使用的 PTC 温度系数 0。

### [RW] odrv0.axis0.motor\_thermistor.config.poly\_coefficient\_1

类型为 [float], 测量电机温度所使用的 PTC 温度系数 1。

### [RW] odrv0.axis0.motor\_thermistor.config.poly\_coefficient\_2

类型为 [float], 测量电机温度所使用的 PTC 温度系数 2。

### [RW] odrv0.axis0.motor\_thermistor.config.poly\_coefficient\_3

类型为 [float], 测量电机温度所使用的 PTC 温度系数 3。

### [RW] odrv0.axis0.motor\_thermistor.config.temp\_limit\_lower

类型为 [float], 单位为 [°C], 当电机温度高于此值时 ODrive 开始减小电流输出。

#### **[RW] odrv0.axis0.motor\_thermistor.config.temp\_limit\_upper**

类型为 [float], 单位为 [°C], 当电机温度高于此值时 ODrive 将停止电流输出并报出电机温度过高错误。

#### **[RW] odrv0.axis0.motor\_thermistor.config.enabled**

类型为 [bool], 是否启用电机温度测量。

### **odrv0.axis0.motor**

#### **[RW] odrv0.axis0.motor.error**

类型为 [enum], 电机错误代码。

```
ERROR_NONE = 0x00000000,
ERROR_PHASE_RESISTANCE_OUT_OF_RANGE = 0x00000001,
ERROR_PHASE_INDUCTANCE_OUT_OF_RANGE = 0x00000002,
ERROR_ADC_FAILED = 0x00000004,
ERROR_DRV_FAULT = 0x00000008,
ERROR_CONTROL_DEADLINE_MISSED = 0x00000010,
ERROR_NOT_IMPLEMENTED_MOTOR_TYPE = 0x00000020,
ERROR_BRAKE_CURRENT_OUT_OF_RANGE = 0x00000040,
ERROR_MODULATION_MAGNITUDE = 0x00000080,
ERROR_BRAKE_DEADTIME_VIOLATION = 0x00000100,
ERROR_UNEXPECTED_TIMER_CALLBACK = 0x00000200,
ERROR_CURRENT_SENSE_SATURATION = 0x00000400,
ERROR_CURRENT_LIMIT_VIOLATION = 0x00001000,
ERROR_BRAKE_DUTY_CYCLE_NAN = 0x00002000,
ERROR_DC_BUS_OVER_REGEN_CURRENT = 0x00004000,
ERROR_DC_BUS_OVER_CURRENT = 0x00008000,
```

#### **[RO] odrv0.axis0.motor.armed\_state**

类型为 [enum], PWM输出状态。

```
ARMED_STATE_DISARMED = 0,
ARMED_STATE_WAITING_FOR_TIMINGS = 1,
ARMED_STATE_WAITING_FOR_UPDATE = 2,
ARMED_STATE_ARMED = 3,
```

#### **[RO] odrv0.axis0.motor.is\_calibrated**

类型为 [bool], 电机是否已校准。

#### **[RO] odrv0.axis0.motor.current\_meas\_phB**

类型为 [float], 单位为 [A], 电机 B 相测量的电流。

#### **[RO] odrv0.axis0.motor.current\_meas\_phC**

类型为 [float], 单位为 [A], 电机 C 相测量的电流。

#### **[RW] odrv0.axis0.motor.DC\_calib\_phB**

类型为 [float], 单位为 [A], 电机 B 相电流偏移量。

## **[RW] odrv0.axis0.motor.DC\_calib\_phC**

类型为 [float]，单位为 [A]，电机 C 相电流偏移量。

## **[RW] odrv0.axis0.motor.phase\_current\_rev\_gain**

类型为 [float]，DRV8301 电流采样运放放大倍数的倒数。

## **[RO] odrv0.axis0.motor.effective\_current\_lim**

类型为 [float]，单位为 [A]，由于温度升高或设定的最大电流而计算得到的当前电流限制值。

## **odrv0.axis0.motor.current\_control**

### **[RW] odrv0.axis0.motor.current\_control.p\_gain**

类型为 [float]，电流PI环的 P 值，此值根据电机相电感，相电阻自动计算生成，也可手动在此处调整，但是无法保存到FLASH，重启后手动设置的值将会消失。

### **[RW] odrv0.axis0.motor.current\_control.i\_gain**

类型为 [float]，电流PI环的 I 值，此值根据电机相电感，相电阻自动计算生成，也可手动在此处调整，但是无法保存到FLASH，重启后手动设置的值将会消失。

### **[RW] odrv0.axis0.motor.current\_control.v\_current\_control\_integral\_d**

类型为 [float]，直轴电流环PI控制中的积分项，一般用来查看调试使用。

### **[RW] odrv0.axis0.motor.current\_control.v\_current\_control\_integral\_q**

类型为 [float]，交轴电流环PI控制中的积分项，一般用来查看调试使用。

### **[RW] odrv0.axis0.motor.current\_control.lbus**

类型为 [float]，单位为 [A]，axis0 电机电流，此值为  $I_q + I_d$ 。

### **[RW] odrv0.axis0.motor.current\_control.final\_v\_alpha**

类型为 [float]，单位为 [V]，最终输出到 SVM 的  $V_{\alpha}$ 。

### **[RW] odrv0.axis0.motor.current\_control.final\_v\_beta**

类型为 [float]，单位为 [V]，最终输出到 SVM 的  $V_{\beta}$ 。

### **[RW] odrv0.axis0.motor.current\_control.Id\_setpoint**

类型为 [float]，单位为 [A]，电流环控制输入的目标直轴电流值。

### **[RW] odrv0.axis0.motor.current\_control.Iq\_setpoint**

类型为 [float]，单位为 [A]，电流环控制输入的目标交轴电流值。

### **[RW] odrv0.axis0.motor.current\_control.Iq\_measured**

类型为 [float]，单位为 [A]，通过电流采样获取的交轴电流值。

### **[RW] odrv0.axis0.motor.current\_control.Id\_measured**

类型为 [float]，单位为 [A]，通过电流采样获取的直轴电流值。

### **[RW] odrv0.axis0.motor.current\_control.I\_measured\_report\_filter\_k**

类型为 [float]，电流采样后对直轴交轴的电流低通滤波系数，默认为 1 不启用低通滤波，设置为 0 将完全被过滤，此值无法保存到 FLASH，重启后恢复为默认值 1。

#### **[RO] odrv0.axis0.motor.current\_control.max\_allowed\_current**

类型为 [float]，单位为 [A]，根据设定的最大电流值和当前温度自动计算得出的当前最大允许电流。

#### **[RO] odrv0.axis0.motor.current\_control.overcurrent\_trip\_level**

类型为 [float]，单位为 [A]，根据实时计算的最大允许电流和设定的允许波动范围计算出的过流阈值，超过此值将停止电机并报错 ERROR\_CURRENT\_SENSE\_SATURATION。

#### **[RW] odrv0.axis0.motor.current\_control.acim\_rotor\_flux**

类型为 [float]，单位为 [A]，交流感应电机转子磁通强度，用电流来表示。

#### **[RO] odrv0.axis0.motor.current\_control.async\_phase\_vel**

类型为 [float]，单位为 [rad/s]，交流感应电机相速度。

#### **[RW] odrv0.axis0.motor.current\_control.async\_phase\_offset**

类型为 [float]，单位为 [rad]，交流感应电机相移。

### **odrv0.axis0.motor.gate\_driver**

#### **[RO] odrv0.axis0.motor.gate\_driver.drv\_fault**

类型为 [enum]，axis0 DRV8301 芯片内部错误代码。

```
DRV8301_FaultType_NoFault = (0 << 0),    //!< No fault
DRV8301_FaultType_FETLC_OC = (1 << 0),    //!< FET Low side, Phase C Over Current fault
DRV8301_FaultType_FETHC_OC = (1 << 1),    //!< FET High side, Phase C Over Current fault
DRV8301_FaultType_FETLB_OC = (1 << 2),    //!< FET Low side, Phase B Over Current fault
DRV8301_FaultType_FETHB_OC = (1 << 3),    //!< FET High side, Phase B Over Current fault
DRV8301_FaultType_FETLA_OC = (1 << 4),    //!< FET Low side, Phase A Over Current fault
DRV8301_FaultType_FETHA_OC = (1 << 5),    //!< FET High side, Phase A Over Current fault
DRV8301_FaultType_OTW      = (1 << 6),    //!< Over Temperature Warning fault
DRV8301_FaultType_OTSD     = (1 << 7),    //!< Over Temperature Shut Down fault
DRV8301_FaultType_PVDD_UV  = (1 << 8),    //!< Power supply Vdd Under Voltage fault
DRV8301_FaultType_GVDD_UV  = (1 << 9),    //!< DRV8301 Vdd Under Voltage fault
DRV8301_FaultType_GVDD_OV  = (1 << 10)    //!< DRV8301 Vdd Over Voltage fault
```

### **odrv0.axis0.motor.config**

#### **[RW] odrv0.axis0.motor.config.pre\_calibrated**

类型为 [bool]，电机是否已经校准，设置为 True 时认为以下电机参数有效可用，如果不想每次重启重新校准电机参数可以设置为 True

#### **[RW] odrv0.axis0.motor.config.pole\_pairs**

类型为 [int32\_t], 电机极对数, 可以查看电机数据手册或者数一下永磁体个数来获取极对数, 极对数 = 电机永磁体个数 / 2

#### **[RW] odrv0.axis0.motor.config.calibration\_current**

类型为 [float], 电机校准时的电流大小, 单位为 [A]

#### **[RW] odrv0.axis0.motor.config.resistance\_calib\_max\_voltage**

类型为 [float], 电机校准自动检测相电阻时最大电压, 单位为 [V]

#### **[RW] odrv0.axis0.motor.config.phase\_inductance**

类型为 [float], 电机相电感, 单位为 [H], 此值会在执行电机校准后自动更新, 如果您有详细的电机参数可以手动设置此值

#### **[RW] odrv0.axis0.motor.config.phase\_resistance**

类型为 [float], 电机相电阻, 单位为 [Ohm], 此值会在执行电机校准后自动更新, 如果您有详细的电机参数可以手动设置此值

#### **[RW] odrv0.axis0.motor.config.direction**

类型为 [int32\_t], 电机运行方向, 0 无效, 1 和编码器方向一致 -1 和编码器方向相反, 无感模式下需要手动设置方向, 有感模式下会在编码器校准中自动设置

#### **[RW] odrv0.axis0.motor.config.motor\_type**

类型为 [enum], 电机类型

```
MOTOR_TYPE_HIGH_CURRENT      = 0,
MOTOR_TYPE_GIMBAL            = 2,
MOTOR_TYPE_ACIM              = 3,
```

#### **[RW] odrv0.axis0.motor.config.current\_lim**

类型为 [float], 电机最大运行电流, 单位为 [A]

#### **[RW] odrv0.axis0.motor.config.current\_lim\_margin**

类型为 [float], 超过电机最大运行电流的容忍度, 如: 此值设置为 3 表示当电机电流超过限制电流 3A 时停止电机并报错

#### **[RW] odrv0.axis0.motor.config.torque\_lim**

类型为 [float], 单位为 [Nm], 电机输出最大扭矩

#### **[RW] odrv0.axis0.motor.config.requested\_current\_range**

类型为 [float], 单位为 [A], 电机运行的电流范围, 根据此值来自动调整电流采样运放的增益, 设置后需要保存配置并重启才能生效

#### **[RW] odrv0.axis0.motor.config.current\_control\_bandwidth**

类型为 [float], 电流控制环的控制带宽, 用来自动计算电流环 PI 参数, 加减速比较缓慢的电机此参数应该越低

#### **[RW] odrv0.axis0.motor.config.acim\_slip\_velocity**

类型为 [float], 单位为 [rad/s], 交流感应电机的 slip velocity。

#### **[RW] odrv0.axis0.motor.config.acim\_gain\_min\_flux**

类型为 [float], 单位为 [A], acim\_gain\_min\_flux。

#### **[RW] odrv0.axis0.motor.config.acim\_autoflux\_min\_id**

类型为 [float], 单位为 [A], acim\_autoflux\_min\_id。

#### **[RW] odrv0.axis0.motor.config.acim\_autoflux\_enable**

类型为 [bool], 是否启用交流感应电机的自动弱磁功能。

#### **[RW] odrv0.axis0.motor.config.acim\_autoflux\_attack\_gain**

类型为 [float], acim\_autoflux\_attack\_gain。

#### **[RW] odrv0.axis0.motor.config.acim\_autoflux\_decay\_gain**

类型为 [float], acim\_autoflux\_decay\_gain。

## **odrv0.axis0.controller**

### **[RW] odrv0.axis0.controller.error**

类型为 [enum], axis0 控制器错误代码

ERROR_NONE	= 0x00000000,
ERROR_OVERSPEED	= 0x00000001,
ERROR_INVALID_INPUT_MODE	= 0x00000002,
ERROR_UNSTABLE_GAIN	= 0x00000004,
ERROR_INVALID_MIRROR_AXIS	= 0x00000008,
ERROR_INVALID_LOAD_ENCODER	= 0x00000010,
ERROR_INVALID_ESTIMATE	= 0x00000020,

### **[RW] odrv0.axis0.controller.input\_pos**

类型为 [float], 单位为 [turn], 输入的电机目标位置, 如: 设置为 0.1 即表示电机轴转动到 36° (360° \* 0.1 = 36°) 。

### **[RW] odrv0.axis0.controller.input\_vel**

类型为 [float], 单位为 [turn/s], 输入的电机目标转速, 如: 设置为 1 即表示电机将以 1转/秒的速度旋转。

### **[RW] odrv0.axis0.controller.input\_torque**

类型为 [float], 单位为 [Nm], 输入的电机输出的力矩大小。

### **[RO] odrv0.axis0.controller.pos\_setpoint**

类型为 [float], 单位为 [turn], 设定的电机目标位置。

### **[RO] odrv0.axis0.controller.vel\_setpoint**

类型为 [float], 单位为 [turn/s], 设定的电机目标转速。

### **[RO] odrv0.axis0.controller.torque\_setpoint**

类型为 [float], 单位为 [Nm], 设定的电机输出的力矩大小。

### **[RO] odrv0.axis0.controller.trajectory\_done**



类型为 [bool]，轨迹模式下是否已运行到设定位置。

### **[RW] odrv0.axis0.controller.vel\_integrator\_torque**

类型为 [float]，单位为 [Nm]，维持当前角度或位置需要输出的扭矩大小。

### **[RW] odrv0.axis0.controller.anticogging\_valid**

类型为 [bool]，anticogging 数据是否有效。

### **[OP] odrv0.axis0.controller.move\_incremental(displacement: float, from\_input\_pos: bool)**

命令电机转动一定角度，displacement 想要转动多少转，from\_input\_pos 是否以 input\_pos 作为参考，如果设置为 False 则相对于 pos\_setpoint。

### **[OP] odrv0.axis0.controller.start\_anticogging\_calibration()**

执行 anticogging 校准。

## **odrv0.axis0.controller.config**

### **[RW] odrv0.axis0.controller.config.gain\_scheduling\_width**

类型为 [float]，速度环增益规划功能启用的位置误差范围。

### **[RW] odrv0.axis0.controller.config.enable\_gain\_scheduling**

类型为 [bool]，是否启用增益规划功能，增益规划功能有利于增加在目标位置的鲁棒性。原理为当位置误差小于 odrv0.axis0.controller.config.gain\_scheduling\_width 时按照比例调整速度环增益大小，来增强非线性区域控制的鲁棒性。

### **[RW] odrv0.axis0.controller.config.enable\_vel\_limit**

类型为 [bool]，是否启用转速限制功能。

### **[RW] odrv0.axis0.controller.config.enable\_current\_mode\_vel\_limit**

类型为 [bool]，是否启用在电流控制模式下转速限制功能。

### **[RW] odrv0.axis0.controller.config.enable\_overspeed\_error**

类型为 [bool]，当转速超过设置限制值时是否停止电机并报错。

### **[RW] odrv0.axis0.controller.config.control\_mode**

类型为 [enum]，控制模式。

```
CONTROL_MODE_VOLTAGE_CONTROL    = 0,
CONTROL_MODE_TORQUE_CONTROL     = 1,
CONTROL_MODE_VELOCITY_CONTROL   = 2,
CONTROL_MODE_POSITION_CONTROL   = 3,
```

### **[RW] odrv0.axis0.controller.config.input\_mode**

类型为 [enum]，控制信号输入模式。

```
INPUT_MODE_INACTIVE           = 0,
INPUT_MODE_PASSTHROUGH       = 1,
INPUT_MODE_VEL_RAMP          = 2,
INPUT_MODE_POS_FILTER        = 3,
INPUT_MODE_MIX_CHANNELS      = 4,
INPUT_MODE_TRAP_TRAJ         = 5,
INPUT_MODE_TORQUE_RAMP       = 6,
INPUT_MODE_MIRROR            = 7,
```

#### **[RW] odrv0.axis0.controller.config.pos\_gain**

类型为 [float], 单位为 [(turn/s) / turn], 位置环增益。

#### **[RW] odrv0.axis0.controller.config.vel\_gain**

类型为 [float], 单位为 [Nm/(turn/s)], 速度环增益。

#### **[RW] odrv0.axis0.controller.config.vel\_integrator\_gain**

类型为 [float], 单位为 [Nm/(turn/s \* s)], 速度环积分增益。

#### **[RW] odrv0.axis0.controller.config.vel\_limit**

类型为 [float], 单位为 [turn/s], 设定的最大转速限制。

#### **[RW] odrv0.axis0.controller.config.vel\_limit\_tolerance**

类型为 [float], 最大转速波动容忍度, 如: 此值设定为 1.2 时表示当转速超过设定的最大转速值的 1.2 倍才会触发 ERROR\_OVERSPEED。

#### **[RW] odrv0.axis0.controller.config.vel\_ramp\_rate**

类型为 [float], 单位为 [(turn/s) / s], 转速爬升速度。用于当控制信号输入模式设置为 INPUT\_MODE\_VEL\_RAMP 时。

#### **[RW] odrv0.axis0.controller.config.torque\_ramp\_rate**

类型为 [float], 单位为 [Nm/s], 转矩爬升速度。用于当控制信号输入模式设置为 INPUT\_MODE\_TORQUE\_RAMP 时。

#### **[RW] odrv0.axis0.controller.config.circular\_setpoints**

类型为 [bool], 是否启用环形控制模式, 此模式对于连续的增量位置移动很有用。例如, 机器人会无限期滚动, 或者挤出机马达或传送带会以受控的增量无限期地移动。在常规位置模式下, input\_pos 将增长到非常大的值, 并且由于浮点舍入而失去精度。

#### **[RW] odrv0.axis0.controller.config.circular\_setpoint\_range**

类型为 [bool], 单位为 [turn], 环形控制模式下可输入转动范围。

#### **[RW] odrv0.axis0.controller.config.homing\_speed**

类型为 [float], 单位为 [turn/s], 寻找零点时的转动速度。

#### **[RW] odrv0.axis0.controller.config.inertia**

类型为 [float], 单位为 [Nm/(turn/s^2)], 电机转动惯量。

#### **[RW] odrv0.axis0.controller.config.axis\_to\_mirror**

类型为 [uint8\_t], 控制信号输入模式为 INPUT\_MODE\_MIRROR 时哪个电机作为输入。

### **[RW] odrv0.axis0.controller.config.mirror\_ratio**

类型为 [float], 控制信号输入模式为 INPUT\_MODE\_MIRROR 时, 输入电机和输出电机转动的比例。

### **[RW] odrv0.axis0.controller.config.input\_filter\_bandwidth**

类型为 [float], 单位为 [1/s], 控制信号输入的滤波带宽。

### **odrv0.axis0.controller.config.anticogging**

#### **[RO] odrv0.axis0.controller.config.anticogging.index**

类型为 [uint32\_t], anticogging 当前索引。

#### **[RW] odrv0.axis0.controller.config.anticogging.pre\_calibrated**

类型为 [bool], anticogging 是否已经校准完成。

#### **[RO] odrv0.axis0.controller.config.anticogging.calib\_anticogging**

类型为 [bool], anticogging 是否正在校准中。

#### **[RW] odrv0.axis0.controller.config.anticogging.calib\_pos\_threshold**

类型为 [float], 单位为 [count], anticogging 时位置误差的阈值。

#### **[RW] odrv0.axis0.controller.config.anticogging.calib\_vel\_threshold**

类型为 [float], 单位为 [count], anticogging 时转速误差的阈值。

#### **[RO] odrv0.axis0.controller.config.anticogging.cogging\_ratio**

类型为 [float], 暂未使用。

#### **[RW] odrv0.axis0.controller.config.anticogging.anticogging\_enabled**

类型为 [bool], 是否启用 anticogging。

## **odrv0.axis0.encoder**

### **[RW] odrv0.axis0.encoder.error**

类型为 [enum], 编码器错误代码。

ERROR_NONE	= 0x00000000,
ERROR_UNSTABLE_GAIN	= 0x00000001,
ERROR_CPR_POLEPAIRS_MISMATCH	= 0x00000002,
ERROR_NO_RESPONSE	= 0x00000004,
ERROR_UNSUPPORTED_ENCODER_MODE	= 0x00000008,
ERROR_ILLEGAL_HALL_STATE	= 0x00000010,
ERROR_INDEX_NOT_FOUND_YET	= 0x00000020,
ERROR_ABS_SPI_TIMEOUT	= 0x00000040,
ERROR_ABS_SPI_COM_FAIL	= 0x00000080,
ERROR_ABS_SPI_NOT_READY	= 0x00000100,

### **[RO] odrv0.axis0.encoder.is\_ready**

类型为 [bool], 编码器是否准备就绪。

### **[RO] odrv0.axis0.encoder.index\_found**

类型为 [bool]，是否已完成编码器索引校准。

### **[RO] odrv0.axis0.encoder.shadow\_count**

类型为 [int32\_t]，编码器累计计数，当编码器计数增加或减少时此值将累加编码器计数值的变化量。

### **[RO] odrv0.axis0.encoder.count\_in\_cpr**

类型为 [int32\_t]，编码器当前计数值。

### **[RO] odrv0.axis0.encoder.interpolation**

类型为 [float]，编码器当前插补值。

### **[RO] odrv0.axis0.encoder.phase**

类型为 [float]，通过编码器计算得到的当前电角度，范围为  $-\pi \sim +\pi$ 。

### **[RO] odrv0.axis0.encoder.pos\_estimate**

类型为 [float]，单位为 [turn]，当前预测到的位置值。

### **[RO] odrv0.axis0.encoder.pos\_estimate\_counts**

类型为 [float]，单位为 [count]，当前预测到的位置值。

### **[RO] odrv0.axis0.encoder.pos\_cpr**

类型为 [float]，单位为 [turn]，当前约束在 cpr 范围内的位置值。

### **[RO] odrv0.axis0.encoder.pos\_cpr\_counts**

类型为 [float]，单位为 [count]，当前约束在 cpr 范围内的位置值。

### **[RO] odrv0.axis0.encoder.pos\_circular**

类型为 [float]，单位为 [turn]，环形位置模式下当前位置值。

### **[RO] odrv0.axis0.encoder.hall\_state**

类型为 [uint8\_t]，当前霍尔信号状态，位码表示。

### **[RO] odrv0.axis0.encoder.vel\_estimate**

类型为 [float]，单位为 [turn/s]，当前估算转速。

### **[RO] odrv0.axis0.encoder.vel\_estimate\_counts**

类型为 [float]，单位为 [count/s]，当前估算转速。

### **[RO] odrv0.axis0.encoder.calib\_scan\_response**

类型为 [float]，执行编码器偏移校准时的中间值，用来调试使用。

### **[RW] odrv0.axis0.encoder.pos\_abs**

类型为 [int32\_t]，绝对值编码器的绝对位置。

### **[RO] odrv0.axis0.encoder.spi\_error\_rate**

类型为 [float]，spi 绝对值编码器通讯误码率，当大于 0.005 时将报错 ERROR\_ABS\_SPI\_COM\_FAIL。

## odrv0.axis0.encoder.config

### [RW] odrv0.axis0.encoder.config.mode

类型为 [enum], 所使用的编码器类型。

```
MODE_INCREMENTAL      = 0,
MODE_HALL              = 1,
MODE_SINCOS            = 2,
MODE_SPI_ABS_CUI       = 256,
MODE_SPI_ABS_AMS       = 257,
MODE_SPI_ABS_AEAT      = 258,
MODE_SPI_ABS_RLS       = 259,
```

### [RW] odrv0.axis0.encoder.config.use\_index

类型为 [bool], 是否使用编码器索引信号。

### [RW] odrv0.axis0.encoder.config.find\_idx\_on\_lockin\_only

类型为 [bool], 是否只在 lockin 匀速运行时检测索引信号。

### [RW] odrv0.axis0.encoder.config.abs\_spi\_cs\_gpio\_pin

类型为 [uint16\_t], 绝对值编码器spi通讯的片选信号使用的GPIO。

### [RW] odrv0.axis0.encoder.config.zero\_count\_on\_find\_idx

类型为 [bool], 是否将索引信号位置设置为编码器计数0的位置。

### [RW] odrv0.axis0.encoder.config.cpr

类型为 [int32\_t], 电机轴转动每圈编码器产生的脉冲数, 注意和编码器手册中的每转产生的脉冲数 (PPR)区分,  $cpr = PPR * 4$ 。

### [RW] odrv0.axis0.encoder.config.offset

类型为 [int32\_t], 编码器和转子电角度之间的相位差, 进行编码器偏移校准之后此项自动更新。

### [RW] odrv0.axis0.encoder.config.pre\_calibrated

类型为 [bool], 编码器是否事先已被校准。

### [RW] odrv0.axis0.encoder.config.offset\_float

类型为 [float], 编码器和转子电角度之间的相位差浮点部分, 进行编码器偏移校准之后此项自动更新。

### [RW] odrv0.axis0.encoder.config.enable\_phase\_interpolation

类型为 [bool], 是否启用根据转速插补编码器相位。

### [RW] odrv0.axis0.encoder.config.bandwidth

类型为 [float], 编码器更新带宽, 设置后立即生效, 使用的编码器分辨率越高此项应该约高。

### [RW] odrv0.axis0.encoder.config.calib\_range

类型为 [float], 通过编码器cpr检查所需的精度, 用于在执行编码器偏移校准时编码器的实际输出计算的移动距离和电机开环步进移动距离之间允许的最大误差, 超过此误差将报错 ERROR\_CPR\_OUT\_OF\_RANGE。

### [RW] odrv0.axis0.encoder.config.calib\_scan\_distance

类型为 [float]，单位为 [rad]，编码器偏移校准时转动的距离，此值越大校准精度相对越高，但是同时校准所需的时间越长。

#### **[RW] odrv0.axis0.encoder.config.calib\_scan\_omega**

类型为 [float]，单位为 [rad/s]，编码器偏移校准时的转动速度。

#### **[RW] odrv0.axis0.encoder.config.idx\_search\_unidirectional**

类型为 [bool]，是否只在方向被设置的情况下运行编码器索引搜索。

#### **[RW] odrv0.axis0.encoder.config.ignore\_illegal\_hall\_state**

类型为 [bool]，是否忽略错误的霍尔信号。

#### **[RW] odrv0.axis0.encoder.config.sincos\_gpio\_pin\_sin**

类型为 [uint16\_t]，sin cos 编码器sin使用的GPIO。

#### **[RW] odrv0.axis0.encoder.config.sincos\_gpio\_pin\_cos**

类型为 [uint16\_t]，sin cos 编码器cos使用的GPIO。

#### **[OP] odrv0.axis0.encoder.set\_linear\_count(count: int32)**

手动设置当前编码器计数值。

### **odrv0.axis0.sensorless\_estimator**

#### **[RW] odrv0.axis0.sensorless\_estimator.error**

类型为 [enum]，无感观测器错误代码。

ERROR_NONE	= 0x00000000,
ERROR_UNSTABLE_GAIN	= 0x00000004,

#### **[RW] odrv0.axis0.sensorless\_estimator.phase**

类型为 [float]，单位为 [rad]，无感观测器输出相位。

#### **[RW] odrv0.axis0.sensorless\_estimator pll\_pos**

类型为 [float]，单位为 [rad]，无感观测器锁相环输出的位置。

#### **[RW] odrv0.axis0.sensorless\_estimator.vel\_estimate**

类型为 [float]，单位为 [turn/s]，无感观测器估算的转速。

### **odrv0.axis0.sensorless\_estimator.config**

#### **[RW] odrv0.axis0.sensorless\_estimator.config.observer\_gain**

类型为 [float]，单位为 [rad/s]，无感观测器的增益。

#### **[RW] odrv0.axis0.sensorless\_estimator.config pll\_bandwidth**

类型为 [float]，单位为 [rad/s]，无感观测器的锁相环带宽。

#### **[RW] odrv0.axis0.sensorless\_estimator.config.pm\_flux\_linkage**

类型为 [float], 单位为  $[V / (\text{rad/s})]$ , 电机磁链, 观测器需要使用此参数进行观测, 可以通过公式  $\{5.51328895422 / (* <\text{rpm/v}>)\}$  计算得出。

## **odrv0.axis0.traj\_traj**

### **odrv0.axis0.traj\_traj.config**

#### **[RW] odrv0.axis0.traj\_traj.config.vel\_limit**

类型为 [float], 单位为  $[\text{turn/s}]$ , 梯形轨迹控制模式下的最高转速。

#### **[RW] odrv0.axis0.traj\_traj.config.accel\_limit**

类型为 [float], 单位为  $[\text{turn/s}^2]$ , 梯形轨迹控制模式下加速时的加速度。

#### **[RW] odrv0.axis0.traj\_traj.config.decel\_limit**

类型为 [float], 单位为  $[\text{turn/s}^2]$ , 梯形轨迹控制模式下减速时的加速度。

## **odrv0.axis0.min\_endstop**

### **[RO] odrv0.axis0.min\_endstop.endstop\_state**

类型为 [bool], 当前最小限位GPIO信号输入状态。

### **odrv0.axis0.min\_endstop.config**

#### **[RW] odrv0.axis0.min\_endstop.config.gpio\_num**

类型为 [uint16\_t], 最小限位信号输入使用的 GPIO。

#### **[RW] odrv0.axis0.min\_endstop.config.enabled**

类型为 [bool], 是否启用最小限位信号输入。

#### **[RW] odrv0.axis0.min\_endstop.config.offset**

类型为 [float], 单位为  $[\text{turn}]$ , 最小限位信号触发和零点位置偏移量。

#### **[RW] odrv0.axis0.min\_endstop.config.is\_active\_high**

类型为 [bool], 最小限位信号有效电平是否为高电平。

#### **[RW] odrv0.axis0.min\_endstop.config.pullup**

类型为 [bool], 最小限位信号是否启用内部上拉, 如果外部配置上拉电阻则设置为 False。

#### **[RW] odrv0.axis0.min\_endstop.config.debounce\_ms**

类型为 [uint32\_t], 单位为  $[\text{ms}]$ , 最小限位信号消抖时间。

## **odrv0.axis0.max\_endstop**

### **[RO] odrv0.axis0.max\_endstop.endstop\_state**

类型为 [bool], 当前最大限位GPIO信号输入状态。

### **odrv0.axis0.max\_endstop.config**

#### **[RW] odrv0.axis0.max\_endstop.config.gpio\_num**

类型为 [uint16\_t], 最大限位信号输入使用的 GPIO。

**[RW] odrv0.axis0.max\_endstop.config.enabled**

类型为 [bool]，是否启用最大限位信号输入。

**[RW] odrv0.axis0.max\_endstop.config.offset**

类型为 [float]，单位为 [turn]，此项设置无意义，只在 max\_endstop 中有效。

**[RW] odrv0.axis0.max\_endstop.config.is\_active\_high**

类型为 [bool]，最小限位信号有效电平是否为高电平。

**[RW] odrv0.axis0.max\_endstop.config.pullup**

类型为 [bool]，最小限位信号是否启用内部上拉，如果外部配置上拉电阻则设置为 False。

**[RW] odrv0.axis0.max\_endstop.config.debounce\_ms**

类型为 [uint32\_t]，单位为 [ms]，最小限位信号消抖时间。

## **odrv0.axis1 (参考 axis0)**

---