Hall Speed Mode and PWM Control

1.0 Hardware List

ODESC driver board(24V/56V)
DC power(8V-24V/8-56V)
6354 brushless motor
USB data cable
Power resistance
PWM remote control and receiver

ODESC purchase link：https://sequremall.com/collections/odesc

2.0 Hardware Wiring

2.1 Connect the power resistor
2.2 Motor cable connection M0
2.3 Power connection ODESC
2.4 Computer connected to ODESC
2.5 The power red line of the motor hall line is connected to the hall interface 5V of M0.
The ground black wire of the motor hall wire is connected to the hall interface GND of M0.
The other 3 wires of the motor hall wire are randomly connected to the hall interface ABZ of M0.
Note: Some motor hall lines have a temperature sensor line, which does not need to be connected to the ODESC.
2.6 If you use PWM remote control, you need to connect the receiver to the ODESC accordingly.

3.0 Motherboard ODESC Configuration

3.1 Refer to the hardware wiring diagram to complete the hardware connection. Open the terminal, enter odrivetool, ODESC is successfully connected to the computer.

```
odrivetool
```

3.2 Enter odrv0.erase_configuration() and press Enter. Erase old configuration and restore default parameters.

```
odrv0.erase_configuration()
```

3.3 Set the power resistance value.

Note: The power resistance parameter is 50W 2Ω, then the resistance value is configured as 2.

```
odrv0.config.brake_resistance = 2
```

3.4 Set the current value of the power supply overcurrent protection, that is, the maximum current that the power supply device can output.

```
odrv0.config.dc_max_positive_current = 30
```

3.5 Set the overcurrent protection value of the reverse current, the maximum current that the power supply can absorb in the reverse direction, generally a negative value.

Note: This parameter defaults to a conservative value of 10mA. If you use power resistors and get the error DC_BUS_OVER_REGEN_CURRENT, increase this parameter value slightly. If you are not using a power resistor and intend to send braking current back to the power supply, set it to the safe range of the power supply. In this case, this parameter should be higher than the motor's current limit + current limit headroom.

```
odrv0.config.dc_max_negative_current = -2.0
```

3.6 Set the current value of the brake backflow, the current value that the power supply can recover charging.

Note: It is powered by switching power supply, does not have the function of recycling and charging, and the parameter is set to 0. It is powered by battery and can be set according to the actual recharge current that the battery can withstand.

```
odrv0.config.max_regen_current = 0
```

3.7 Set the undervoltage protection value. For this test, use the default parameters and skip this setting step.

Note: Battery powered, to prevent battery over-discharge, you can set this parameter to the safe voltage value of the battery.

```
odrv0.config.dc_bus_undervoltage_trip_level
```

3.8 Set the overvoltage protection value. For this test, use the default parameters and skip this setting step.
Note: If you need to set it, set it according to the voltage version of the ODESC, there are different versions of 24V/56V.
```
odrv0.config.dc_bus_overvoltage_trip_level
```

3.9 Save configuration parameters.
```
odrv0.save_configuration()
```

4.0 Hall Speed M0 Configuration

4.1 Set the number of pole pairs of the motor. The number of pole pairs = the number of permanent magnets of the motor/2. You can check the number of permanent magnets of the motor through the motor data manual.
Note: The number of permanent magnets in the 6354 motor is 14, and the number of pole pairs = 14/2 = 7
```
odrv0.axis0.motor.config.pole_pairs = 7
```

4.2 Set the motor type, 6354 is a high current motor.
```
odrv0.axis0.motor.config.motor_type = MOTOR_TYPE_HIGH_CURRENT
```

4.3 Sets the calibration voltage for the motor.
```
odrv0.axis0.motor.config.resistance_calib_max_voltage = 4
```

4.4 Sets the calibration current for the motor.
```
odrv0.axis0.motor.config.calibration_current = 10
```

4.5 Set the maximum running current of the motor, the unit is [A].
Note: Combined with the actual use of the motor, set the maximum current limit of the motor.
```
odrv0.axis0.motor.config.current_lim = 20
```

4.6 Set the motor current sampling range.
```
odrv0.axis0.motor.config.requested_current_range = 30
```

4.7 Save configuration parameters.
```
odrv0.save_configuration()
```

4.8 Set the hall mode.
```
odrv0.axis0.encoder.config.mode = ENCODER_MODE_HALL
```

4.9 Set to speed control mode.

```
odrv0.axis0.controller.config.control_mode = CONTROL_MODE_VELOCITY_CONTROL
```

4.10 In Hall mode, CPR = number of pole pairs * 6 = 7*6 = 42.

```
odrv0.axis0.encoder.config.cpr = 42
```

4.11 Set the position loop gain.

```
odrv0.axis0.controller.config.pos_gain = 20
```

Set the speed loop gain.

```
odrv0.axis0.controller.config.vel_gain = 0.02
```

Set the speed loop integral gain.

```
odrv0.axis0.controller.config.vel_integrator_gain = 0.1
```

Note: There are unstable situations such as jitter in the closed-loop state of the motor, and generally these three gain parameters need to be debugged.

4.12 Set the maximum speed of the motor, the unit is [turn/s].

```
odrv0.axis0.controller.config.vel_limit = 100
```

4.13 Save configuration parameters.

```
odrv0.save_configuration()
```

4.14 Calibrate the motor and wait for the motor to make a "squeak" sound.

```
odrv0.axis0.requested_state = AXIS_STATE_MOTOR_CALIBRATION
```

4.15 Check the motor error, the return value is "0", the calibration is successful, otherwise solve the problem according to the error prompt.

```
odrv0.axis0.motor
```

4.16 Set the motor has been calibrated, the parameter is True, each restart does not need to re-calibrate the motor.

```
odrv0.axis0.motor.config.pre_calibrated = True
```

4.17 The required accuracy of the encoder CPR check, and an error is reported when the error value is exceeded.

Note: The error ERROR_CPR_OUT_OF_RANGE is reported. After restarting, increase the value and recalibrate the encoder.

```
odrv0.axis0.encoder.config.calib_range = 0.05
```

4.18 To calibrate the encoder, the motor rotates 1 or 1/2 turn in the forward and reverse directions.

```
odrv0.axis0.requested_state = AXIS_STATE_ENCODER_OFFSET_CALIBRATION
```

4.19 Check the encoder error, the return value is "0", the calibration is successful, otherwise solve the problem according to the error prompt.

Note: If the error "0x02" is returned, after restart, set odrv0.axis0.encoder.config.calib_range increase the value and recalibrate the encoder.

```
odrv0.axis0.encoder
```

4.20 Set the encoder calibration parameter to True.

```
odrv0.axis0.encoder.config.pre_calibrated = True
```

4.21 Set to enter closed loop control.

```
odrv0.axis0.requested_state = AXIS_STATE_CLOSED_LOOP_CONTROL
```

4.22 Automatically enter closed-loop control after power-on.

```
odrv0.axis0.config.startup_closed_loop_control = True
```

4.23 Save configuration parameters.

```
odrv0.save_configuration()
```

4.24 Reboot.

```
odrv0.reboot()
```

4.25 To test the motor, enter the target speed of the motor in [turn/s].

Note: Set to a negative number to change the running direction of the motor, and "0" to stop. The target speed setting value and the actual motor running condition are related to the value of position loop gain, speed loop gain and speed loop integral gain.

```
odrv0.axis0.controller.input_vel = 5
```

```
odrv0.axis0.controller.input_vel = -5
```

4.26 Estimate the actual speed of the current motor, the unit is [turn/s].

```
odrv0.axis0.encoder.vel_estimate
```

5.0 If Using PWM Control

5.1 Set up to use the RC remote.

```
odrv0.config.gpio3_pwm_mapping.min = -20
```

```
odrv0.config.gpio3_pwm_mapping.max = 20
```

```
odrv0.config.gpio3_pwm_mapping.endpoint = odrv0.axis0.controller._remote_attributes['input_vel']
```

5.2 Save configuration parameters.

```
odrv0.save_configuration()
```

5.3 Reboot.

```
odrv0.reboot()
```

5.4 Operate the RC remote control to control the motor.

The basic configuration test has been completed.