

灯哥 FOC ODrive V1.0

使用说明文档 v1

目录

1 概述.....	1
2 简介.....	1
2.1 总体参数	1
2.2 接口说明	2
2.3 编码器说明	3
3 检查.....	5
3.1 使用前检查	5
3.2 硬件连接与检测.....	6
4 环境配置.....	9
4.1 OdriveTool 安装	9
4.2 Zadig 配置.....	13
4.3 环境测试	15
5 开始使用	16
5.1 一键配置	18
5.2 电机校准	22
5.2 控制	24
6 参数说明	25
6.1 Deng FOC 参数设置	25
6.2 电机参数设置.....	27
6.3 编码器参数设置.....	28
6.4 控制器参数设置.....	29
7 常见问题及其解决	32

7.1 连接直流电源后红灯闪灭并伴随咔哒声	32
7.2 复位后，USB 连接正常，连接直流电源后失去响应.....	33
7.3 复位后 USB 连接异常	33
7.4 电源电压测量值始终为 12.0V	34
7.5 电机校准过程声音嘶哑以及编码器校准过程电机卡顿、往复 非正常运动	34
7.6 电机进入闭环模式后不受控制地往复运动	34

1 概述

Deng FOC 是适用于三相无刷直流电机和永磁同步电机的高集成度 FOC 驱动器。基于功能强大、性能稳定的开源项目 ODrive，我们进一步优化电路、缩小体积，优化接口，提高驱动器整体的可扩展性。Deng FOC 驱动器支持驱动大电流电机，可以实现 FOC 位置、速度、力矩闭环控制，编码器方面则支持 ABZ 模式和 SPI 模式（板载），能够实时检测驱动器温度。同时 Deng FOC 预留了多种接口，包括 USB、CAN、GPIO 扩展以及 FPC 串口接口（可实现屏幕显示或无线控制），让大家尽可能自由开发。更重要的是，我们也进一步压缩成本，致力于打造性价比最高，大家都能一起来玩的高性能 FOC 电机驱动器！

2 简介

2.1 总体参数

目前灯哥 ODrive 的具体技术性能指标如下：

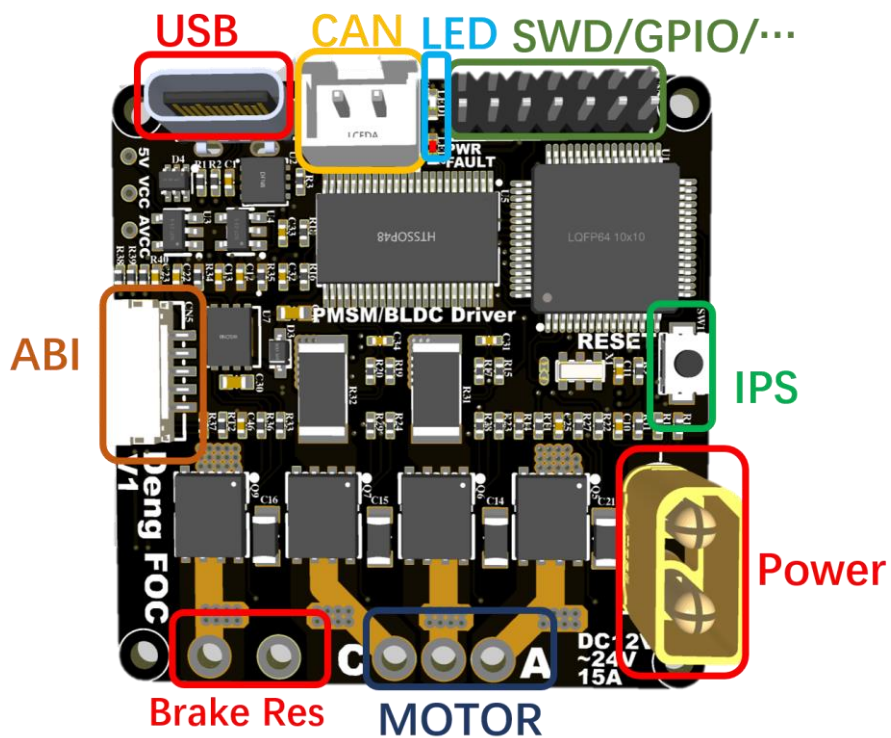
输入电压	12V-24V
工作电流	30
峰值电流	50A
支持电机类型	BLDC 无刷电机
尺寸	42mm*42mm
制动方式	功率耗散电阻
板载编码器类型	AS5047P

支持编码器信号	ABI SPI
通信接口	USB/CAN/UART/模拟输入/PWM
其他	SWD 程序下载/调试 电源指示 VCC DRV 故障状态指示 (nFault)

2.2 接口说明

Deng FOC 包含以下接口：

- XT30 连接器电源输入接口
- MR30 电机输出接口
- XT30 连接器功率电阻接口
- XH2.54 2P CAN 接口
- GH1.25 ABZ 编码器接口
- FPC 8P 屏幕接口
- 2*7P 2.0mm 排针拓展接口，8 个具有模拟输入功能

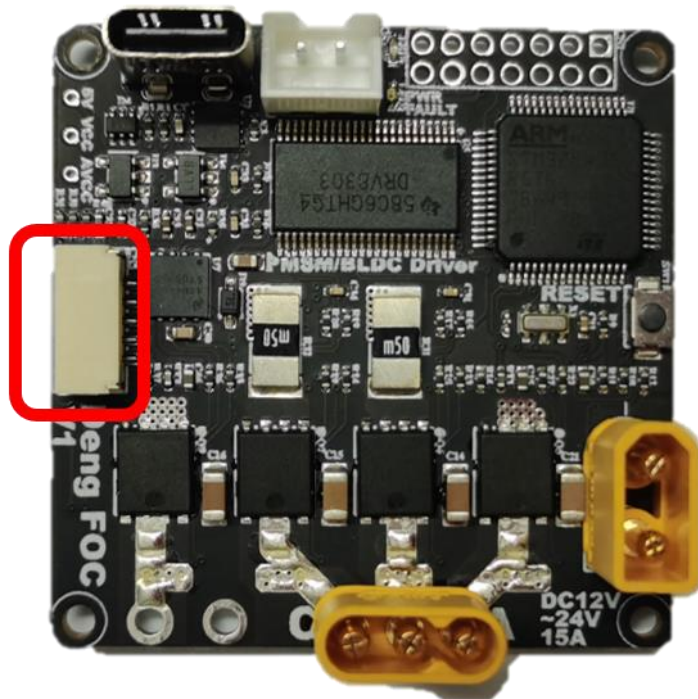


2.3 编码器说明

对于编码器，Deng FOC 有两种使用方式。

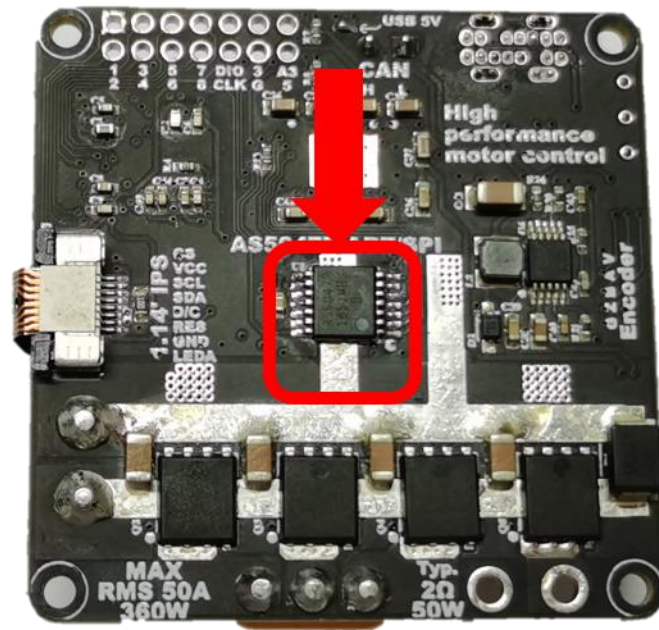
2.3.1 外接编码器

Deng FOC 在板侧留有编码器接口，支持的角度输入方式为 ABI。



2.3.2 板载编码器

Deng FOC 在购买时可选配焊接背面的编码器，该编码器型号为 AS5047P。AS5047P 是一款高分辨率、高速（可达 28krpm）的位置传感器，具有革命性的集成动态角误差补偿（DAEC™），延迟几乎为零。在 Deng FOC 上支持 ABI 和 SPI 的角度输出方式。



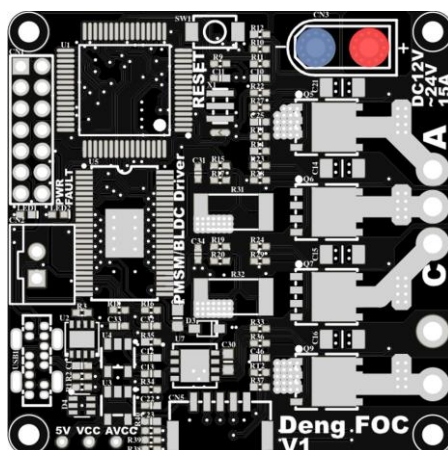
3 检查

3.1 使用前检查

每块 Deng FOC 发货前都经过仔细的外观检查以及例程测试，因此一般都不会出现问题，但在包装运输过程中难免可能会产生损伤，因此建议收到后在上电前进行一般检查。

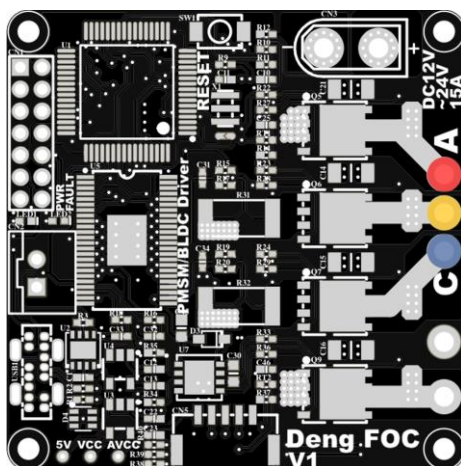
3.1.1 DC_BUS 短路检查

检查电源输入的正负极是否短路。



3.1.2 电机三相输出端短路检查

检查电机输出的三相端是否短路。



3.2 硬件连接与检测

按照 3.2.1 检测完成后，就可以开始准备后文所述的硬件进行首次的上电检测。

3.2.1 硬件准备

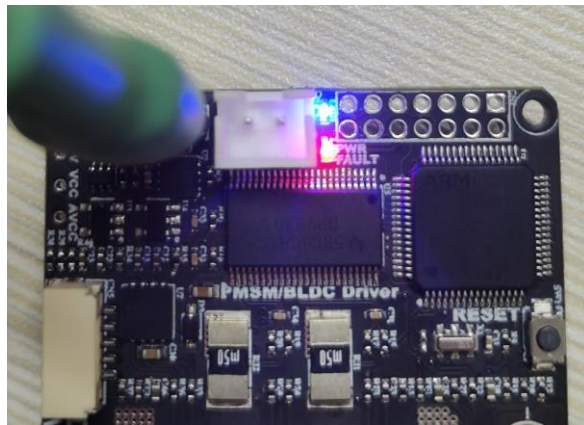
准备硬件清单：

- Deng FOC

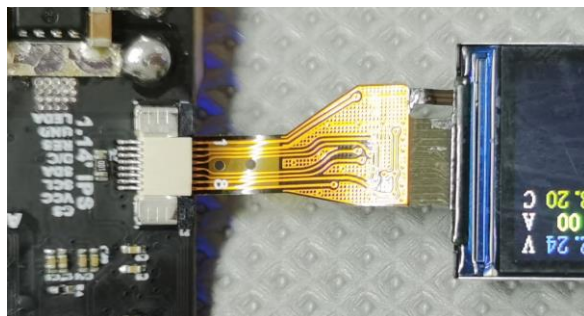
- 12V DC 直流稳压源
- Type-C 数据线
- 1.14 寸 LCD 显示屏（可选）
- 耗散电阻（可选）

3.2.2 连接 USB

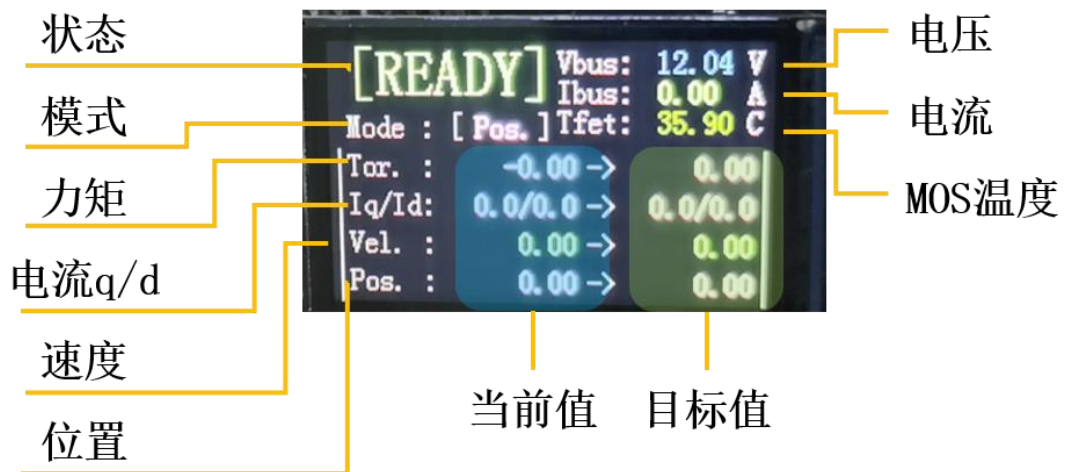
接入 Type-C 数据线，板上蓝灯（电源指示灯）和红灯（DRV 故障指示灯）均亮起。



如果选配屏幕，在驱动板背面的 FPC 连接器上连接屏幕。

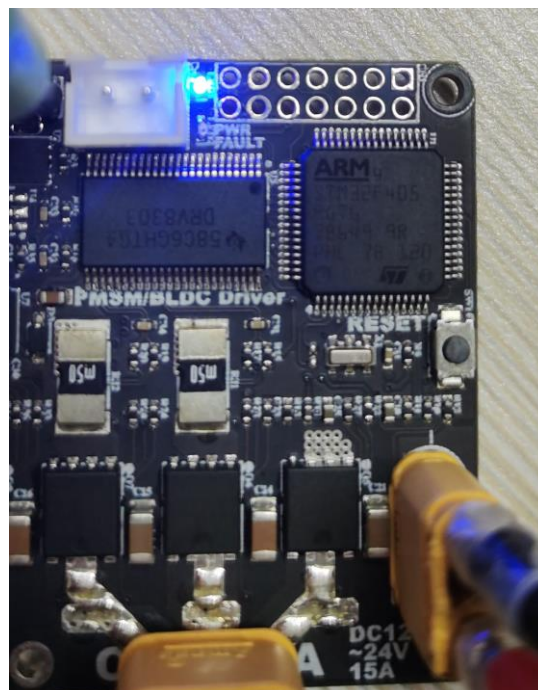


屏幕详情信息如下：

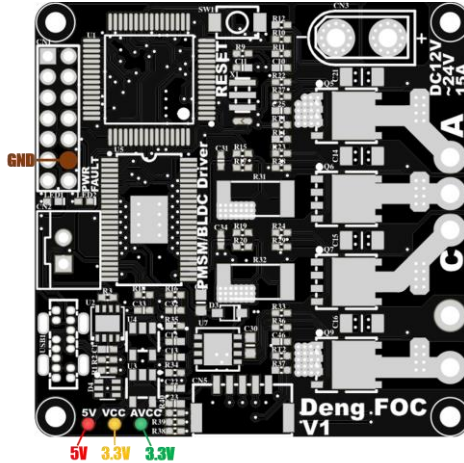


3.3.3 连接电源

接入 DC_BUS 电源（建议先插入 XT30 接口，再打开电源开关），红灯熄灭。



检查稳压电路的输出 5V，以及数字供电 3.3V 和模拟供电 3.3V



4 环境配置

4.1 OdriveTool 安装

Odrivetool 是 python 的一个模块，可以用来配置和调试 odrive 和 Deng FOC。在使用它之前，需要先配置好 python 的环境。

4.1.1 python 环境安装

已安装好可以跳过这一步。我们推荐使用 **anaconda**，因为他包含了很多工具，并且用虚拟环境配置不至于弄乱原有的环境。

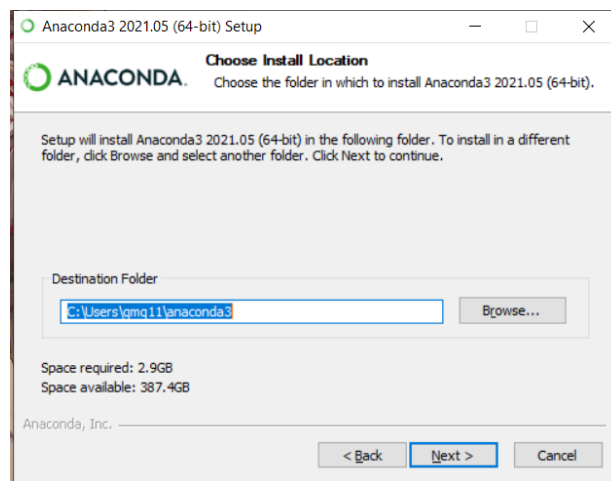
以下以安装 **anaconda** 为例：

A. 下载 **anaconda**。官网网址：

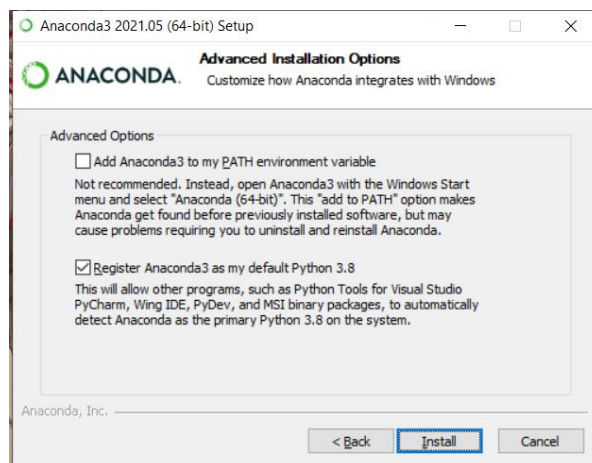
<https://www.anaconda.com/products/individual>，选择适合自己的版本。这里选择的是 64-Bit Graphical Installer (477 MB)。



B. 选择安装路径



C. 注意要取消勾选“Add Anaconda to my PATH environment variable”，等后面安装完成后手动配置。



D. 安装完成后，在开始界面找到并打开 Anaconda Prompt（Anaconda3），输入“conda info”验证是否安装成功。

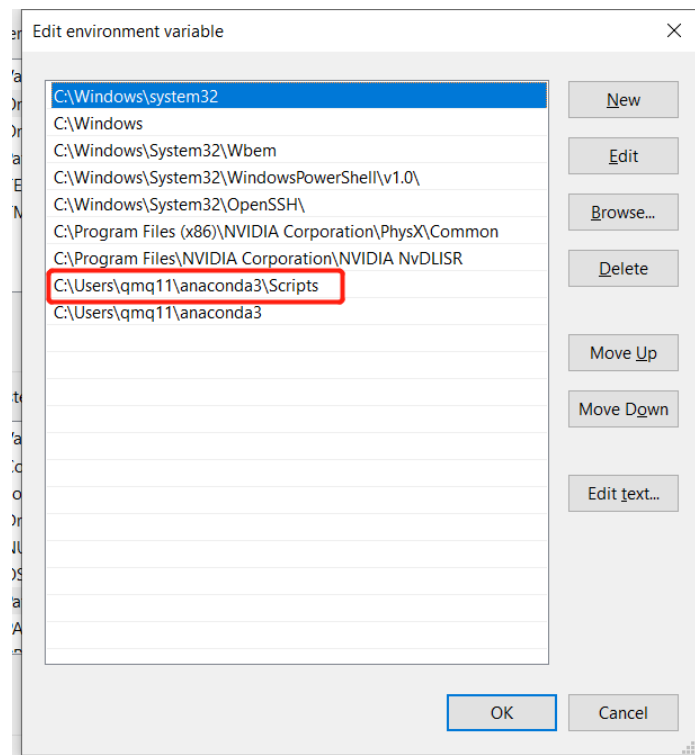
```
Select Anaconda Prompt (anaconda3)

(base) C:\Users\qmq11>conda info

active environment : base
active env location : C:\Users\qmq11\anaconda3
shell level : 1
user config file : C:\Users\qmq11\.condarc
populated config files :
  conda version : 4.10.1
  conda-build version : 3.21.4
  python version : 3.8.8.final.0
  virtual packages : __cuda=11.2=0
                    __win=0=0
                    __archspec=1=x86_64
base environment : C:\Users\qmq11\anaconda3 (writable)
conda av data dir : C:\Users\qmq11\anaconda3\etc\conda
conda av metadata url : https://repo.anaconda.com/pkgs/main
channel URLs : https://repo.anaconda.com/pkgs/main/win-64
               https://repo.anaconda.com/pkgs/main/noarch
               https://repo.anaconda.com/pkgs/r/win-64
               https://repo.anaconda.com/pkgs/r/noarch
               https://repo.anaconda.com/pkgs/msys2/win-64
               https://repo.anaconda.com/pkgs/msys2/noarch
package cache : C:\Users\qmq11\anaconda3\pkgs
                 C:\Users\qmq11\.conda\pkgs
                 C:\Users\qmq11\AppData\Local\conda\conda\pkgs
envs directories : C:\Users\qmq11\anaconda3\envs
                   C:\Users\qmq11\.conda\envs
                   C:\Users\qmq11\AppData\Local\conda\conda\envs
platform : win-64
```

E. 配置环境变量。

控制面板->系统和安全->系统->高级系统设置->环境变量->用户变量->PATH 中添加 anaconda 的安装目录的 Scripts 文件夹。
查看自己安装 anaconda 的路径，找到 Scripts 文件夹，我这里的
路径是（anaconda 安装路径）\Scripts。

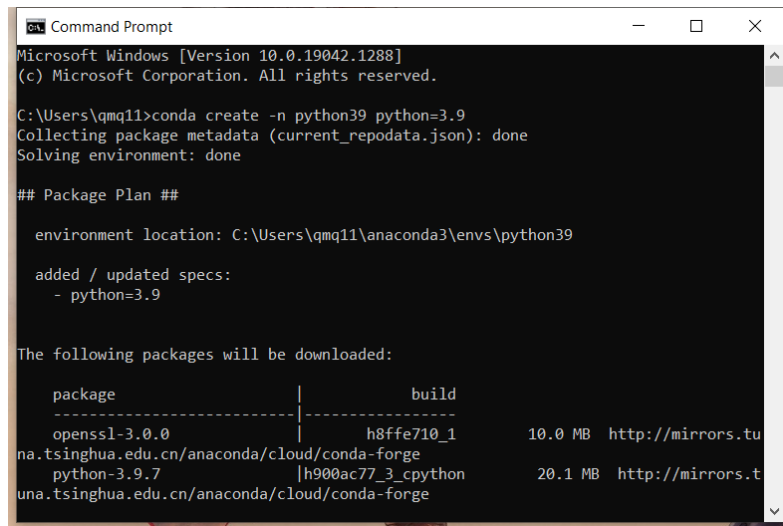


4.1.2 安装 odrivetool

A. 打开命令行 cmd，创建虚拟环境：

```
conda create -n python39 python=3.9
```

其中 python39 为环境名称，python=3.9 为 python 的版本



```
Microsoft Windows [Version 10.0.19042.1288]
(c) Microsoft Corporation. All rights reserved.

C:\Users\qmq11>conda create -n python39 python=3.9
Collecting package metadata (current_repodata.json): done
Solving environment: done

## Package Plan ##

  environment location: C:\Users\qmq11\anaconda3\envs\python39

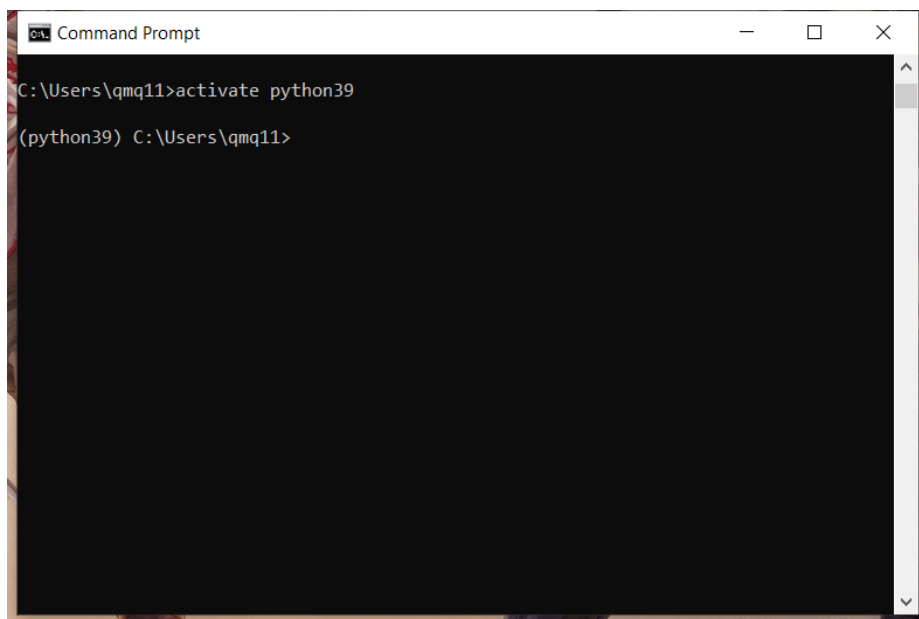
added / updated specs:
- python=3.9

The following packages will be downloaded:

package | build | size | url
-----|-----|-----|-----
openssl-3.0.0 | h8ffe710_1 | 10.0 MB | http://mirrors.tuna.tsinghua.edu.cn/anaconda/cloud/conda-forge/
python-3.9.7 | h900ac77_3_cpython | 20.1 MB | http://mirrors.tuna.tsinghua.edu.cn/anaconda/cloud/conda-forge/
```

B. 激活刚刚新建的环境

```
conda activate python39
```



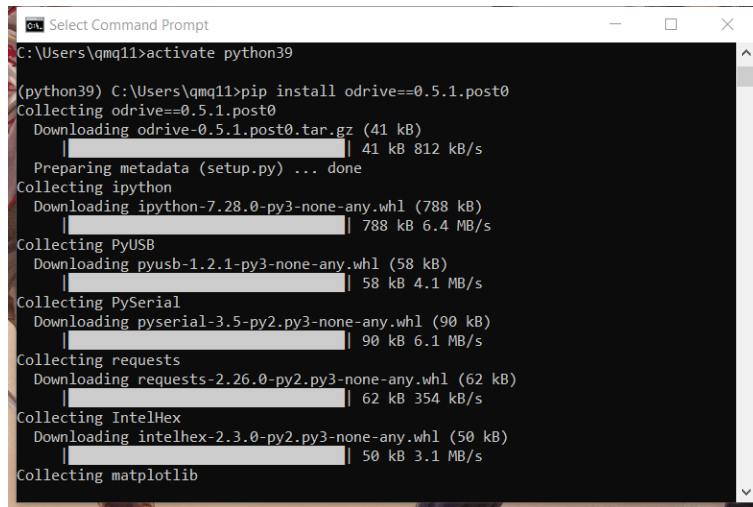
```
Microsoft Windows [Version 10.0.19042.1288]
(c) Microsoft Corporation. All rights reserved.

C:\Users\qmq11>conda activate python39

(python39) C:\Users\qmq11>
```

C. 安装 odrivetool

pip install odrive==0.5.1.post0



D. 输入 odrivetool 测试是否安装成功。

E. 如果出现以下错误

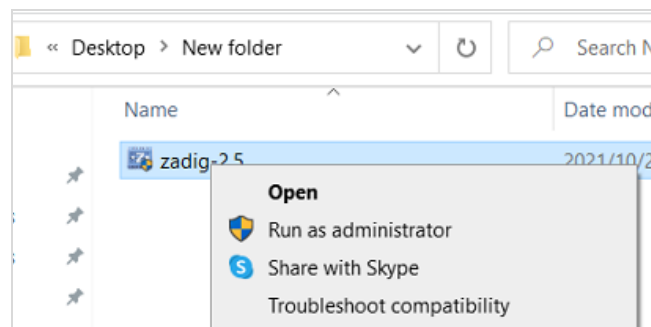
```
~\anaconda3\envs\python39\lib\site-packages\fbre\utils.py in __init__
    159     self._print_lock = threading.Lock()
    160     if platform.system() == 'Windows':
--> 161         self._stdout_buf = win32console.GetStdHandle(win32console.STDOUT)
    162
    163     def indent(self, prefix=' '):
NameError: name 'win32console' is not defined
```

执行

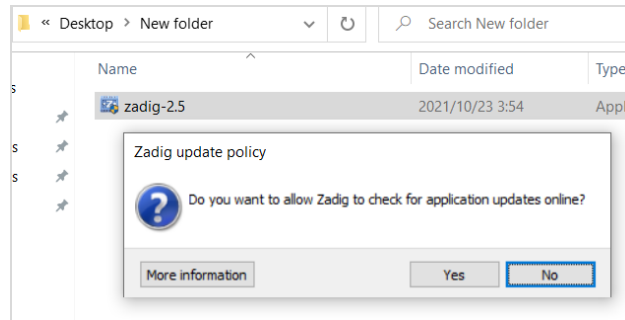
pip install pywin32 == 225

4.2 Zadig 配置

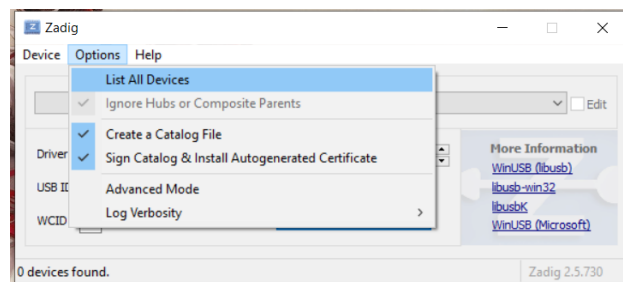
A. 以管理员权限运行 zadig-2.5;



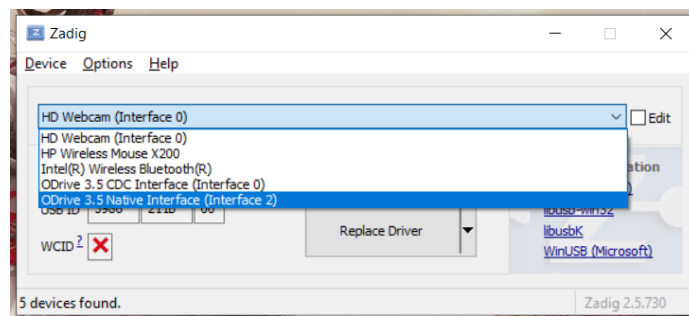
B. 弹出更新提醒，选择 no（不更新）;



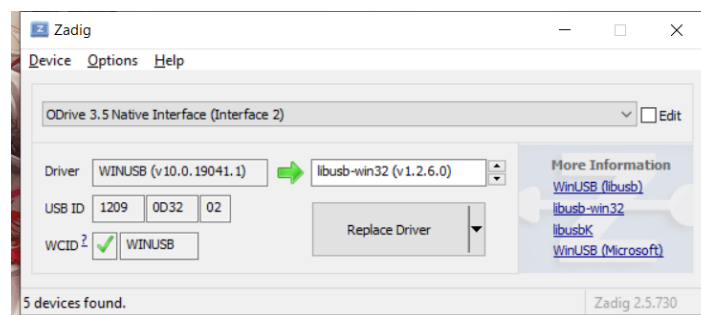
C. 选择 Options 中的 List All Devices;



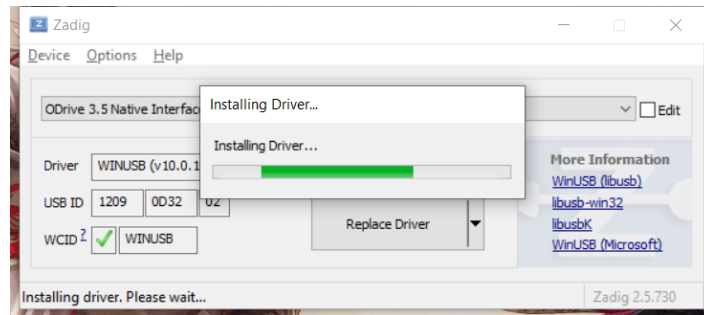
D. 下拉菜单中，选择 Odrive 3.5 Native Interface (Interface 2) ;



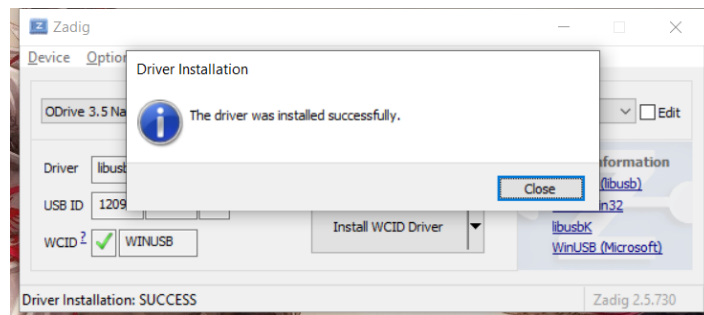
E. 选择上下箭头，选择 libusb-win32(v 1.2.6.0)，点击 Replace Driver;



F. 等待驱动替换完成;



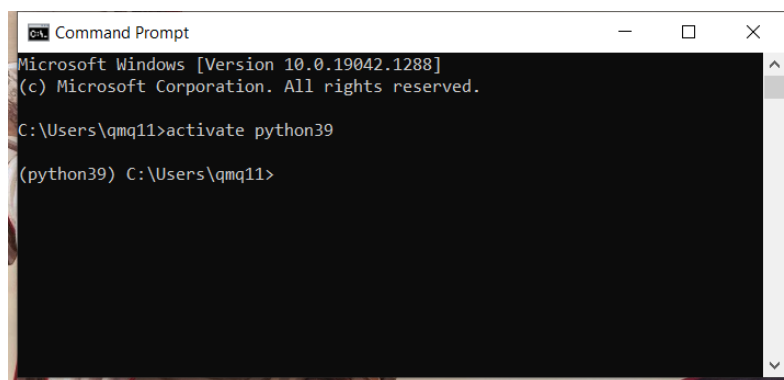
G. 完成后，关闭 Zadig 窗口；



4.3 环境测试

插入 USB，板上的蓝灯亮起。

激活环境（Python39）



在 Python39 环境中连接 odrivetool，输入命令：odrivetool

出现“Connected to ODrive xx as odrv0”，代表连接成功。xx 是 ODrive 的主板序列号

```
Please connect your ODrive.  
You can also type help() or quit().  
  
Downloading json data from ODrive... (this might take a while)  
Connected to ODrive 306335513135 as odrv0  
In [1]: odrv0.vbus_voltage  
Out[1]: 12.0  
  
In [2]: odrv0.axis0.error  
Out[2]: 0  
  
In [3]: odrv0.axis1.error  
Out[3]: 0
```

5 开始使用

准备硬件清单：

- Deng FOC
- 电机
- ABI 编码器（带板载编码器可忽略）
- 12V-24V 直流电源
- Type-C 数据线
- 1.14 寸 LCD 显示屏（可选）
- 耗散电阻（可选）

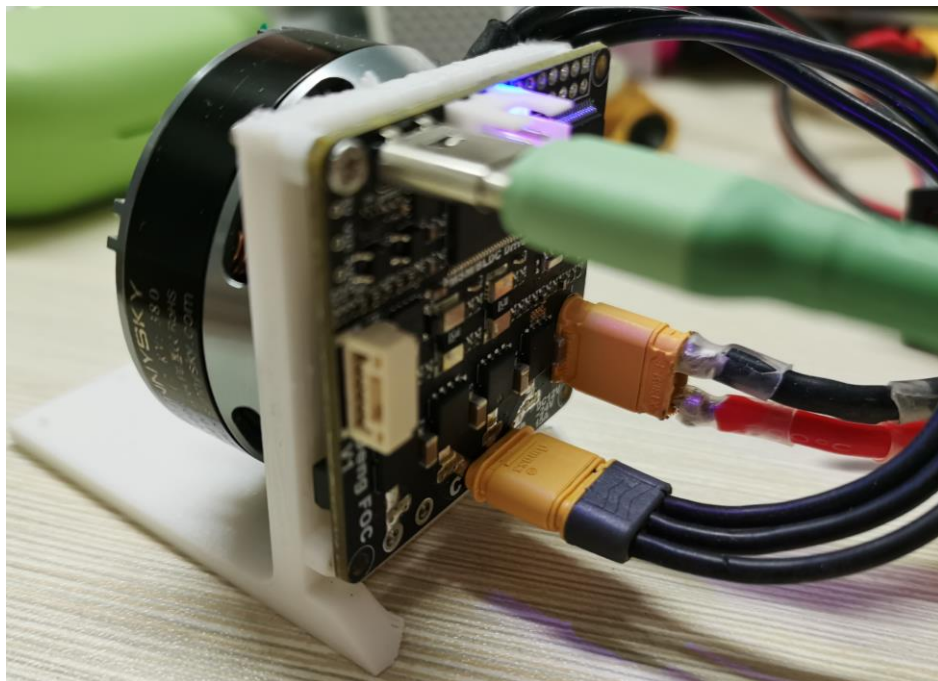
硬件连接

带板载编码器的连接

- 电机轴粘上磁铁：



- 固定 DengFOC，使背面的编码器对准磁铁，间隔 3mm 左右。
- 接上 USB，电机以及电源。



- USB 连接电脑(电源可以先不打开，只需连接 USB)，激活 odrivetool 所在的 conda 环境，输入“odrivetool”。出现连接成功的提示：

```
Downloading json data from ODrive... (this might take a while)
Connected to ODrive 306335513135 as odriv0
```

- 开始新的配置前建议清除现有配置，在 odrivetool 上执行指令：

```
odrv0.erase_configuration()
```

5.1 一键配置

Deng FOC 准备了 python 程序可以一键配置参数。在程序的开头说明并定义了所需的基本参数，运行后即可配置所有参数。

5.1.1 Deng FOC 驱动板的参数设置

根据实际情况修改以下参数（具体的参数说明可以看第 6 章）

```
# Deng FOC参数
board_parameter = {"brake_res": 0, # 耗散电阻值(根据实际接入的功率电阻值输入，不接为0)
                   "dc_max_pos_cur": 35, # 电源的过流保护的电流值
                   "dc_max_neg_cur": -3.0, # 反向电流的过流保护阈值
                   "under_volt_level": 8.0, # 欠压保护阈值
                   "over_volt_level": 30.0, # 过压保护阈值
                   "max_regen_current": 0.0 # 制动回流电流值
}
```

在 main 中执行以下函数：

```
set_board_param(odrv0, board_parameter)
```

5.1.2 电机的参数设置

按照所使用的电机修改具体参数。以 SUNNYSKY V4008 KV380 电机为例，该电机参数为：

电机型号/KV值	V4008/ KV380
槽极数	18N24P
相间内阻	131mΩ
电机尺寸	φ44.3*28.5mm
轴径	4.0mm
电机硅胶线规格	20AWG 600mm
电机重量 (含线)	105g
电机重量 (不含线)	暂无数据
工作额定电压 (Lipo)	4-6S
空载电流	0.5A/10V
最大连续功率	500W
最大连续电流	20A/30s
建议使用电调	30-40A
推荐螺旋桨规格	EOLO CN12*5/CN13*5/CN15*5.5/CN17*6.2

则相应地在配置程序中所设置的参数为：

```
# 电机相关参数
motor_parameter = {"pole_pairs": 12, # 电机的极对数
    "motor_type": MOTOR_TYPE_HIGH_CURRENT, # 电机类型
    "cur_lim": 20, # 电机电流限制 (A)
    "cal_cur": 3, # 电机校准电流限制 (A)
    "cal_vol": 4, # 电机校准电压限制 (V)
    "requested_cur_range": 60, # 电流采样范围 (A)
    "kv": 380 # 电机kv值
}
```

在 main 中执行以下函数：

```
set_motor_param(odrv0, motor_parameter)
```

5.1.3 编码器设置

由于 Deng FOC 的板载编码器型号为 AS5047P，所以以 AS5047P 的 ABI 模式和 SPI 模式为例介绍配置参数。

```
# 编码器参数 for AS5047P
encoder_parameter = {"mode": ["5047_ABI", "5047_SPI"],
                    "encoder_mode": [ENCODER_MODE_INCREMENTAL, ENCODER_MODE_SPI_ABS_AMS], # 编码器模式
                    "cpr": [4000, 2 ** 14], # 编码器cpr
                    "bandwidth": 3000, # 带宽
                    "encoder_cs_pin": 8, # SPI CS引脚
                    "cali_cur": 5, # 配置编码器时校准电机的电流
                    "cali_ramp_dis": 3.1415927410125732, # 配置编码器时电机转动距离
                    "cali_ramp_time": 0.4, # 配置编码器时电流提高的时间
                    "cali_accel": 20, # 配置编码器时电机的加速度
                    "cali_vel": 40, # 配置编码器时电机的速度
                    }
```

另外，因为外接编码器仅支持 ABI 模式，所以如果需要配置其他的编码器，也可以使用上面的“5047_ABI”进行配置，修改相应的 cpr 等参数即可。

在 main 中执行以下函数的一种（分别指代 ABI 和 SPI 模式）：

```
set_encoder_param(odrv0, encoder_parameter, "5047_ABI")
set_encoder_param(odrv0, encoder_parameter, "5047_SPI")
```

值得说明的是，AS5047P-ABI 模式每次上电必须进行运动校准，而使用 AS5047P-SPI 的优势在于可以选择只要校准一次编码器，后续上电就可以不进行编码器运动校准，缩短的校准时间，避免校准运动对其他部件的影响。但使用 AS5047P-SPI 会比较容易出错。关于运动校准的教程后面会详述。

5.1.4 控制器设置

在控制器设置中，首先设置如速度限制的基本参数：

```
# 控制器基本参数
controller_base_parameter = {"vel_lim": 20, # 速度限制 ([turn/s)
                             "vel_gain": 0.02, # 速度环增益
                             "pos_gain": 30, # 位置环增益
                             "vel_integrator": 0.2, # 速度环积分
                             }
```

Deng FOC 可以实现电机的速度/位置/力矩运动控制，每种运动控

制支持多种输入模式，如直接输入，爬升模式等。每种模式会有对应的参数配置，如梯形轨迹的位置模式下需要设置梯形轨迹的速度限制，梯形轨迹的加减速速度限制等。

```
# 位置模式
position_mode_parameter = {"control_mode": CONTROL_MODE_POSITION_CONTROL, # 设置位置控制
                           "input_mode": [INPUT_MODE_PASSTHROUGH, INPUT_MODE_TRAP_TRAJ], # 设置输入模式
                           "trap_traj_vel_lim": 30, # 梯形轨迹速度限制
                           "trap_traj_accel_limit": 5, # 梯形轨迹加速度限制
                           "trap_traj_decel_limit": 5, # 梯形轨迹减速度限制
                           }
)
```

方便起见，程序设置了 6 种模式可以进行配置，分别包括

- 0: 速度直接模式
- 1: 速度爬升模式
- 2: 位置直接模式
- 3: 位置梯形轨迹模式
- 4: 力矩直接模式
- 5: 力矩爬升模式

在 main 中执行以下函数

```
set_control_mode(odrv0, mode)
```

其中 mode 应配置为以上各模式的数字，例如想设置为速度直接模式，则为：

```
set_control_mode(odrv0, 0)
```

5.1.5 一键配置

设置完上面的参数后，并在 main 中加入所需设置部分的函数后，在 conda 激活 odrivetool 所在的虚拟环境，运行一键配置的程序（需退出 odrivetool），执行指令：

```
Python Deng_FOC_Setting.py
```


出现“Done!”后，表示配置完成。

```
Setting board parameters
Setting motor parameters
Setting encoder parameters
Setting controller base param
Setting Deng FOC as velocity mode
Done!
```

5.2 电机校准

初次配置电机后，需要进行电机的校准和编码器的校准，并设置上电闭环模式（可选）。

依然进入 odrivetool 所在的虚拟环境，输入“odrivetool”，出现连接成功的提示：

```
Downloading json data from ODrive... (this might take a while)
Connected to ODrive 306335513135 as odrv0
```

接入 12-24V 电源，输入 odrv0.vbus_voltage，检查电源电压是否正确，就可以进入校准步骤。

```
Downloading json data from ODrive... (this might take a while)
Connected to ODrive 306335513135 as odrv0
In [1]: odrv0.vbus_voltage
Out[1]: 12.0
```

5.2.1 一般校准

一般校准时，每次上电后都自动校准编码器，然后进入闭环模式。

ABI 编码器只能使用一般校准。

在 odrivetool 上执行：

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

指令执行两秒后电机会“哔”，这个校准会测量电机的电气特性（包括相电阻和相电感）。

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

设置开机自动校准电机。

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

指令执行后，电机慢慢地来回转动，测量电机相位和编码器的偏移。

```
odrv0.axis0.config.startup_encoder_offset_calibration = True
```

设置开机自动进行编码器校准。

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

指令执行后，电机进入闭环模式，电机保持当前的位置。

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

设置开机自动进入闭环。

```
odrv0.save_configuration()  
odrv0.reboot()
```

保存配置并重启。

5.2.2 免编码器校准

SPI 模式下的编码器可以选择免编码器校准，上电后只校准电机。

在 `odrivetool` 上执行：

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

指令执行两秒后电机“哔”，这个校准会测量电机的电气特性（包括相电阻和相电感）。

```
odrv0.axis0.config.startup_motor_calibration = True
```

设置开机自动校准电机

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

```
    AXIS_STATE_ENCODER_OFFSET_CALIBRATION
```

指令执行后，电机慢慢地来回转动，测量电机相位和编码器的偏移。

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

设置编码器的预校准，设置后会保存本次的校准值，后续上电会自动校准。

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

指令执行后，电机进入闭环模式，电机保持当前的位置。

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

设置开机自动进入闭环。

```
odrv0.save_configuration()
```

```
odrv0.reboot()
```

保存配置并重启。

最后，在进行任何配置后，记得执行 `odrv0.save_configuration()` 保存设置。

5.2 控制

5.2.1 闭环速度模式

设置速度直接或速度爬升模式，运行一键配置程序并校准完电机后，打开在 conda 中打开 odrivetool，输入

```
odrv0.axis0.controller.input_vel = [速度目标值]
```

实现转速控制，同时可以在屏幕上看到实时变化。。

5.2.2 闭环位置模式

设置位置直接或梯形轨迹位置模式，运行一键配置程序并校准完电机后，打开在 conda 中打开 odrivetool，输入

```
odrv0.axis0.controller.input_pos = [位置目标值]
```

实现位置控制，同时可以在屏幕上看到实时变化。。

5.2.3 力矩模式

设置力矩直接或力矩爬升置模式，运行一键配置程序并校准完电机后，打开在 conda 中打开 odrivetool，输入

```
odrv0.axis0.controller.input_torque = [位置目标值]
```

实现力矩控制，同时可以在屏幕上看到实时变化。

6 参数说明

Deng FOC 可以通过 ODrive Tool，对 axis0 进行参数配置。

6.1 Deng FOC 参数设置

■ 使能耗散电阻

```
odrv0.config.enable_brake_resistor
```

如果要使用耗散电阻，就将其设置为 **True**。用电池电源的话一般可以不设置。

■ 设置耗散电阻值

```
odrv0.config.brake_resistance [Ω]
```

如果不想使用，保持为默认值即可。另外因为接线和端子可能也会存在一些电阻，所以如果刹车时有什么问题，可以试试将这个参数增加 $0.05\ \Omega$ 。

■ 设置电源的过流保护的电流值

```
odrv0.config.dc_max_positive_current[A]
```

■ 设置反向电流的过流保护阈值

```
odrv0.config.dc_max_negative_current[A]
```

即电源可以吸收的最大电流值。

一般为负值。默认保守设置为 10mA。如果在用功率电阻的情况下出现 **DC_BUS_OVER_REGEN_CURRENT** 的错误，就稍微提高这个参数。而在不用功率电阻时想将电流回流到电源的话，就将这个参数设置为电源的安全范围内，这个时候该参数应该高于电机的电流限制+电流限制余量。

■ 设置欠压，过压保护。

```
odrv0.config.dc_bus_undervoltage_trip_level [V]  
odrv0.config.dc_bus_overvoltage_trip_level [V]
```

Deng FOC 支持电压范围是 12-24V，**建议设置为 26V。**

■ 设置制动回流的电流值。

```
odrv0.config.max_regen_current[A]
```

如果用的是直流稳压源，不具备回收电能，就配置为 0。如果是电池供电，可以按照电池实际承受的回流电流进行设置。高于该参数的回流电流会由耗散电阻消耗掉。

6.2 电机参数设置

■ 设置电机的极对数

```
odrv0.axis0.motor.config.pole_pairs
```

电机的磁极数除以 2。

■ 设置电机类型

```
odrv0.axis0.motor.config.motor_type
```

目前支持两种类型：

◆ 大电流电机 **MOTOR_TYPE_HIGH_CURRENT**

◆ 云台电机 **MOTOR_TYPE_GIMBAL**

注意：不要在非云台电机上选择 **MOTOR TYPE GIMBAL**，可能导致电机或者 Deng FOC 过热。

电机类型的选择：

◆ 如果 **100mA** 的电流噪声对电机来说算“小”的，那就可以选择大电流电机

◆ 如果 **100mA** 的电流噪声对电机来说算“大”，同时电机转速不需要很高，电机内阻大于 1Ω ，就选云台电机

■ 设置电机的力矩常数

```
odrv0.axis0.motor.config.torque_constant
```

通常**设置为 8.27/KV 值**，这是每 **1A** 电流流到电机所产生的力矩的比值。如果你想以电流 **A** 为单位设置力矩，就将此参数设置为 **1**。

■ 设置电机的电流限制

```
odrv0.axis0.motor.config.current_lim [A].
```

出于安全的考虑，可以先开始设置 10A 的电流限制，确保 Deng FOC 稳定运行。后面配置好后，再修改合适的电机电流限制。（注意：电机电流和电源电流并不是一回事，简单来说电机的转速越高，电源电流和电机电流就越接近）。

■ 设置校准电流

```
odrv0.axis0.motor.config.calibration_current [A]
```

设置最大的校准电流。如果电机有负载，设太小电机不够力去旋转校准，就需要加大校准电流来实现完整正确的电机校准。

■ 设置电机校准时的电压

```
odrv0.axis0.motor.config.resistance_calib_max_voltage[V]
```

电机相电阻小，该参数设太高就会导致过流保护的错误。具体可以通过欧姆定律来大概设置。

■ 设置电机的速度限制

```
odrv0.axis0.controller.config.vel_limit [turn/s].
```

顾名思义，用于限制电机的速度。

■ 设置电机电流采样范围

```
odrv0.axis0.motor.config.requested_current_range[A]
```

6.3 编码器参数设置

以 AMS5047P 的 ABI 模式和 SPI 模式的设置来说明。

	指令	ABI 模式	SPI 模式
--	----	--------	--------

编码器类型	<div>odrv0.axis0.encoder.conf</div> <div>g.mode</div>	ENCODER_MODE_INCREMENTAL	MODE_SPI_ABS_AMS
编码器CPR	<div>odrv0.axis0.encoder.conf</div> <div>g.cpr</div>	4000	16384
SPI模式CS引脚	<div>odrv0.axis0.encoder.conf</div> <div>g.abs_spi_cs_gpio_pin</div>		8
带宽	<div>odrv0.axis0.encoder.conf</div> <div>g.bandwidth</div>	3000	

6.4 控制器参数设置

■ 控制模式

```
odrv0.axis0.controller.config.control_mode
```

常用的控制模式包括：

- ◆ 力矩控制-**CONTROL_MODE_TORQUE_CONTROL**
- ◆ 速度控制-**CONTROL_MODE_VELOCITY_CONTROL**
- ◆ 位置控制-**CONTROL_MODE_POSITION_CONTROL**

■ 输入模式

```
odrv0.axis0.controller.config.control_mode
```

输入模式有很多种，以下列举常用的输入模式，以及说明每种输入模式需要相应设置的参数：

- ◆ 关闭输入- **INPUT_MODE_INACTIVE**
- ◆ 直接控制- **INPUT_MODE_PASSTHROUGH**

在这种输入模式下，根据所设置的控制模式 `control_mode` 来控制电机，指令包括 `input_pos/input_vel/input_torque`，输入指令后电机直接运转至目标值。

- ◆ 速度爬升- **INPUT_MODE_VEL_RAMP**

这种输入模式下，驱动器需要在**速度控制模式**下，并会从当前的速度值逐渐爬升/下降到输入到 `input_vel` 的目标速度值。在这种模式下，需要配置

爬 升 速 率 -

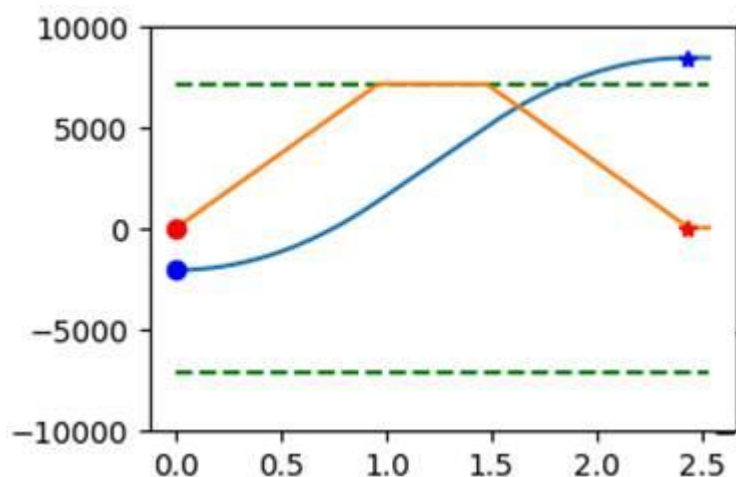
```
odrv0.axis0.controller.config.vel_ramp_rate[turn/sec]
```

负载惯量 - `odrv0.axis0.controller.config.inertia[Nm/(turn/s`

²)]

◆ 梯形轨迹- **INPUT_MODE_TRAP_TRAJ**

这种输入模式下，驱动器需要在**位置控制模式**下，通过 `input_pos` 输入目标位置值，电机按照加速-匀速-减速到达设定位置，即速度变化轨迹为梯形，位置变化接近曲线。



在这种模式下，需要配置

梯形轨迹下的速度限制- `odrv0.axis0.trap_traj.config.vel_limit`

梯形轨迹下的加速度限制- `odrv0.axis0.trap_traj.config.accel_limit`

梯形轨迹下的加加速度限制- `odrv0.axis0.trap_traj.config.dccel_limit`

◆ 力矩爬升- **INPUT_MODE_TORQUE_RAMP**

这种输入模式下，驱动器需要在**力矩控制模式**下，并会从当前的力矩值逐渐爬升/下降到输入到 `input_tor` 的目标速度值。在这种模式下，需要配置

力 矩 爬 升 速 度 -

`odrv0.axis0.controller.config.torque_ramp_rate`

■ PID 参数

速度环增益- `odrv0.axis0.controller.config.vel_gain`

速度环积分- `odrv0.axis0.controller.config.vel_integrator_gain`

位置环增益- `odrv0.axis0.controller.config.pos_gain`

位置环积分- `odrv0.axis0.controller.config.pos_integrator_gain`

7 常见问题及其解决

7.1 连接直流电源后红灯闪灭并伴随咋哒声

可能原因：

- (1) LM5109 损坏
- (2) AUX_H 和 AUX_L 短路
- (3) 功率耗散电路下管损坏
- (4) 逆变电路 MOS 管损坏

定位问题&参考解决方法：

(1) 断电测量 AUX_H 和 AUX_L 是否短路，短路则问题为 AUX_H 和 AUX_L 短路

(2) 若无问题，断电测量下管 DS 是否导通，导通则问题为功率耗散电路下管损坏

(3) 若无问题，拆除耗散电路两只 MOS 管，上电后依旧闪灭则问题为逆变电路故障，若不再闪灭则问题为功率耗散电路故障

(3.1) 若依旧闪灭，则按组拆除逆变电路 A/B/C 三相的上下管，直到不再闪灭，更换该组 MOS 即可

(3.2) 若不再闪灭，同时测量耗散电路上管下管 G 极信号，观察是否会导致上下管同时导通，或更换耗散电路 MOS 管，若存在同

时导通的可能或者依旧闪灭，则问题为 LM5109 损坏，更换 LM5109 即可

7.2 复位后，USB 连接正常，连接直流电源后失去响应

可能原因：

- （1）逆变电路 MOS G 极虚焊

7.3 复位后 USB 连接异常

可能原因：

- （1）USB 母座焊接问题
- （2）CAN 收发器损坏
- （3）晶振损坏

定位问题&参考解决方法：

- （1）测试 USB 母座 DP 与 DM 是否短接，测试 USB 母座 DM 与 DP、DM/DP 与单片机 PA11/PA12 之间的电阻，短路、阻值远大于 $22\ \Omega$ ，则问题为 USB 母座焊接问题
- （2）使用示波器观察晶振是否起振，若未能观察到 8MHz 的正弦波，则判定晶振损坏，应尝试更换晶振，若问题依旧，应检查 VCC/AVCC 是否为 3.3V，若电压正常，则继续更换晶振
- （3）使用 VS code 进行调试，调用堆栈为“MX_FREERTOS_Init”函数、“construct_objects”函数、“MX_CAN1_Init”函数，问题定位至 CAN

收发器损坏，更换 CAN 收发器即可

7.4 电源电压测量值始终为 12.0V

可能原因：

- (1) DRV8303 损坏
- (2) DEV8303 的 EN_GATE 虚焊

定位问题&参考解决方法：

- (1) 测量 GVDD/AVDD/DVDD，观察其是否分别为 11V/6V/3.3V 左右，若近乎为 0，则应检查 EN_GATE 虚焊是否虚焊，若无虚焊，应尝试更换 DRV8303 后重新测试

7.5 电机校准过程声音嘶哑以及编码器校准过程电机卡顿、往复非正常运动

可能原因：

- (1) 电机接触不良

定位问题&参考解决方法：

- (1) 用手压实连接器重新进行电机校准和编码器校准，若有所改善，则问题为电机接触不良，更换连接器或者直接焊线即可

7.6 电机进入闭环模式后不受控制地往复运动

可能原因：

- (1) LP5907 损坏

定位问题&参考解决方法:

- (1) 使用示波器观察 AVCC，若纹波过大，这判定为 LP5907 损坏，
更换 LP5907 即可

校准失败

检查编码器的接线。检查电机的磁铁和编码器之间的距离是否为
1-3mm