



—— Mr. Deng ——

灯哥FOC ODrive V1.0



///

Instruction Manual
使用说明书

CONTENT

目录

目录

一、概述	01
二、简介	03
三、检查	05
四、环境配置	09
五、开始使用	16
六、参数说明	24
七、常见问题及其解决	30

一、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），并能够实时检测驱动器温度。
- ◆ Deng FOC预留了多种接口，包括USB、CAN、UART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！

高性能 高性价比

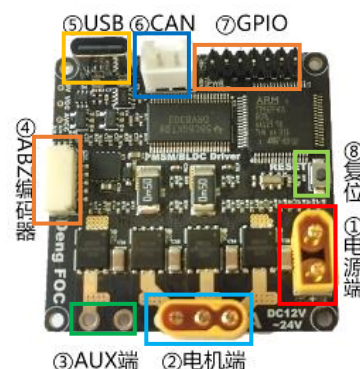
二、简介

1. 总体参数

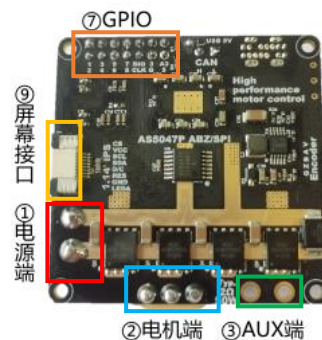
输入电压	12V-24V
工作电流	30A
峰值电流	50A
支持电机类型	直流无刷电机
尺寸	42mm*42mm
制动方式	功率耗散电阻
板载编码器类型	AS5047P
支持编码器信号	ABI SPI PWM HALL
通信接口	USB CAN UART Step/Dir 模拟输入
其他	SWD程序下载/调试 电源指示VCC DRV故障状态指示(nFault)

2. 接口说明

a 正面



b 反面



①XT30电源输入接口

②MR30电机端输出接口

③耗散电阻AUX接口

④GH1.25 ABZ编码器接口

⑤Type-C USB接口

⑥XH2.54 2P CAN接口

⑦2*7P 2.00 GPIO接口

⑧复位按键

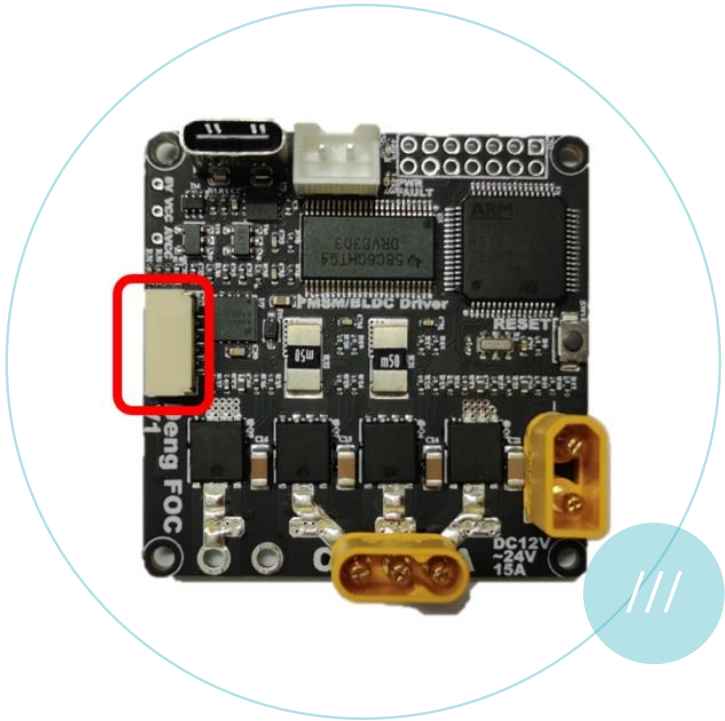
⑨FPC 8P 0.5mm 屏幕接口

其中，⑦的2*7P GPIO引脚示意图如下：

Pin1	Pin3	Pin5	Pin7
UART.TX Step	数字输入 PWM输入	数字输入	
Pin2	Pin4	Pin6	Pin8
UART.RX Dir	数字输入 PWM输入		
DIO	3	A3	
SWDIO	数字3.3V	模拟3.3V	
CLK	G	5	
SWCLK	GND	5V	

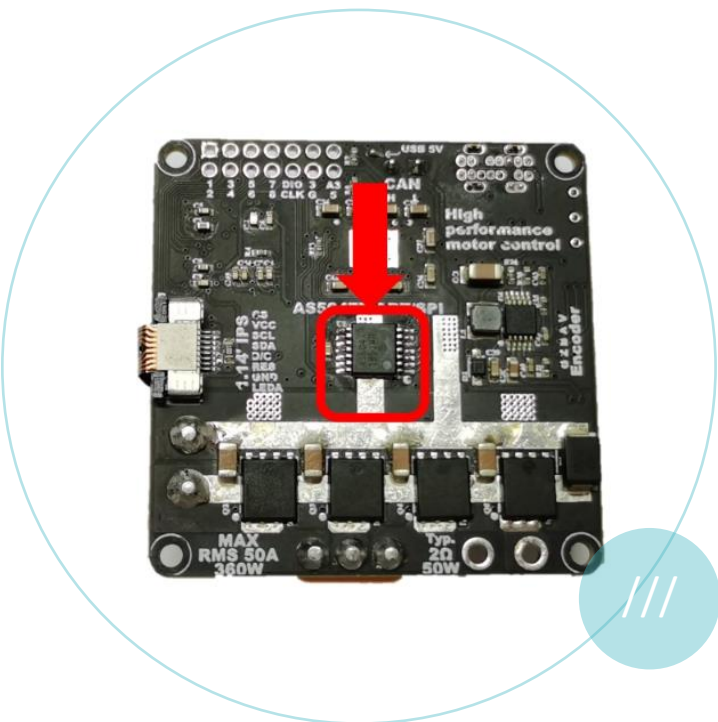
3. 编码器说明

对于编码器，Deng FOC 有两种型号，可在
淘宝页面选购。



外接编码器

Deng FOC在板侧留有编码器接口，支持的输入方式为ABZ/HALL。



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。



!! 注意!!

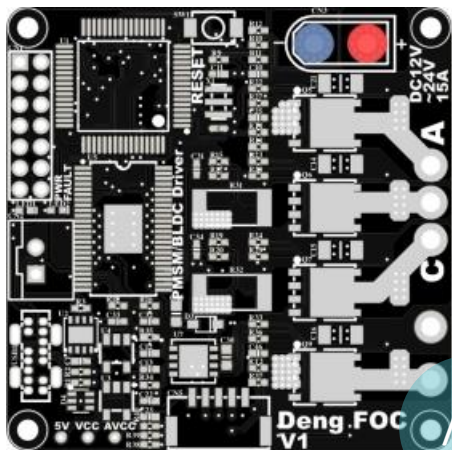
如果板子为板载编码器型，则编码器接口失效。请勿在含有编码器的板子上再外接编码器，否则会造成后续编码器校准出错。

三、检查

1. 使用前检查

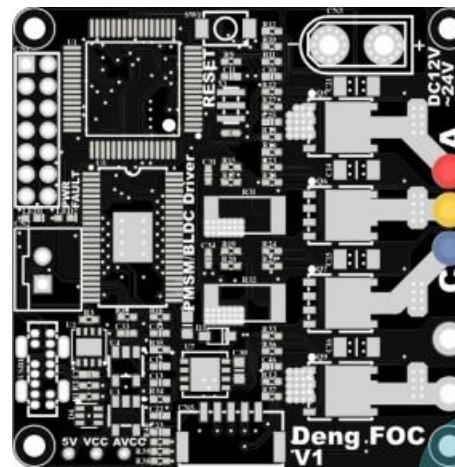
每块Deng FOC发货前都经过仔细的外观检查以及例程测试，因此一般都不会出现问题，但在包装运输过程中难免可能会产生损伤，因此建议收到后在上电前进行一般检查。

◆ DC_BUS 短路检查



检查电源输入的正负极是否短路

◆ 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

2. 硬件连接与检测

【使用前检查】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。

◆ 硬件准备

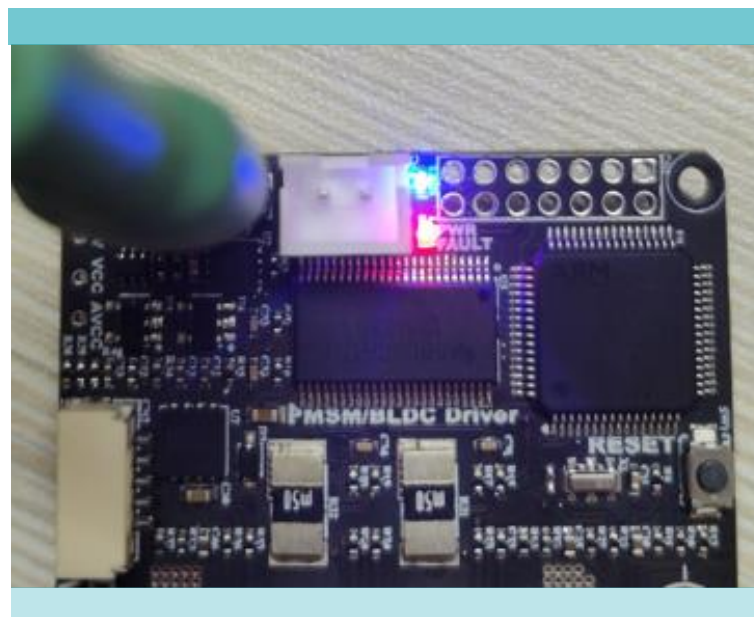
硬件清单

- ✓ Deng FOC
- ✓ 12V-24V DC直流稳压源
- ✓ Type-C 数据线 (**注意区分数据线和充电线**)
- ✓ 1.14寸LCD显示屏 (可选)
- ✓ 耗散电阻 (可选)

◆ 连接USB

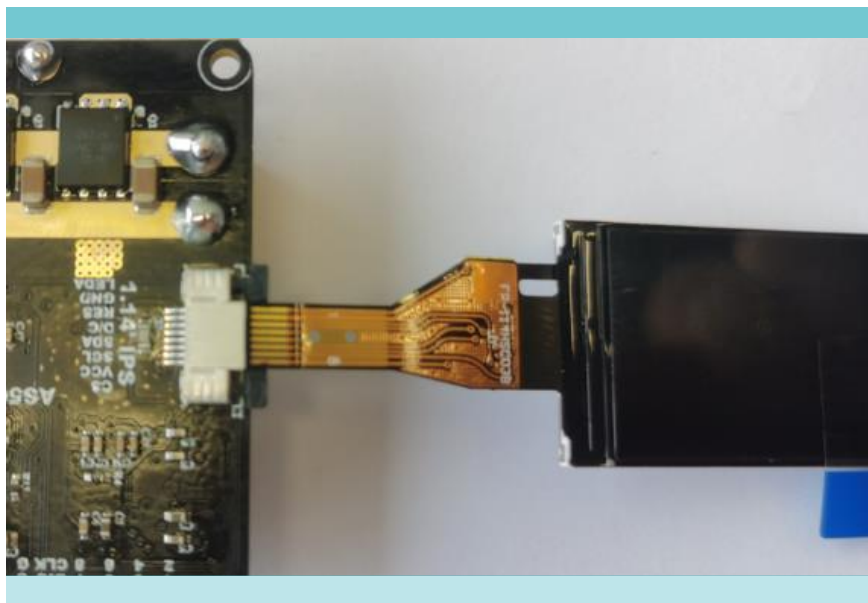
第一步

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。



第二步

如果选配屏幕，在驱动板背面的FPC连接器上连接屏幕。将FPC连接器两边的黑色锁扣轻轻往外拉出，按照下图的方向插入屏幕排线，确保金手指完全进入连接器后，再将锁扣按进连接器。



屏幕详情信息如下：



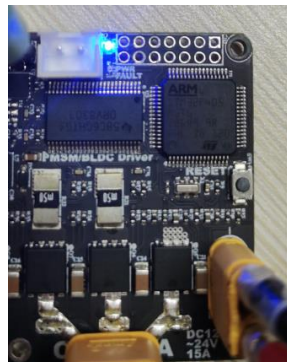
屏幕
显示
信息：

- | | |
|---------|---------|
| ■ 状态 | ■ 电压 |
| ■ 模式 | ■ 电流 |
| ■ 力矩 | ■ MOS温度 |
| ■ 电流q/d | ■ 当前值 |
| ■ 速度 | ■ 目标值 |
| ■ 位置 | |

◆ 连接电源

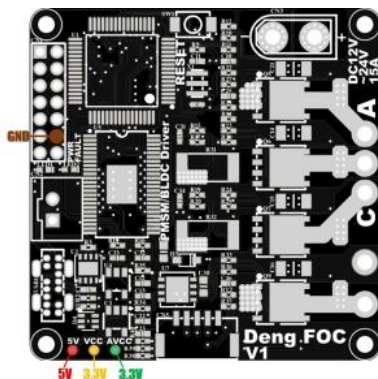
第一步

接入DC_BUS电源（先插入XT30接口，再打开电源开关），红灯熄灭。



第二步

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



四、环境配置

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。

◆ 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。

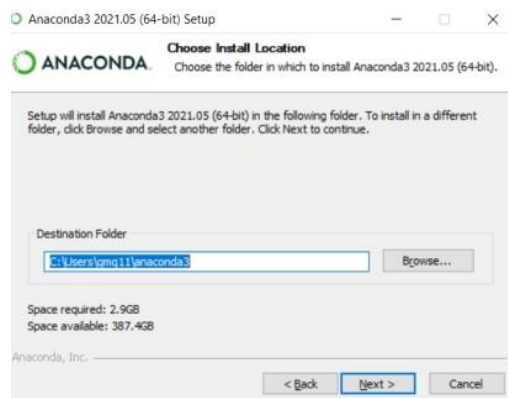
以下以安装anaconda为例：

A 下载anaconda官网网址:

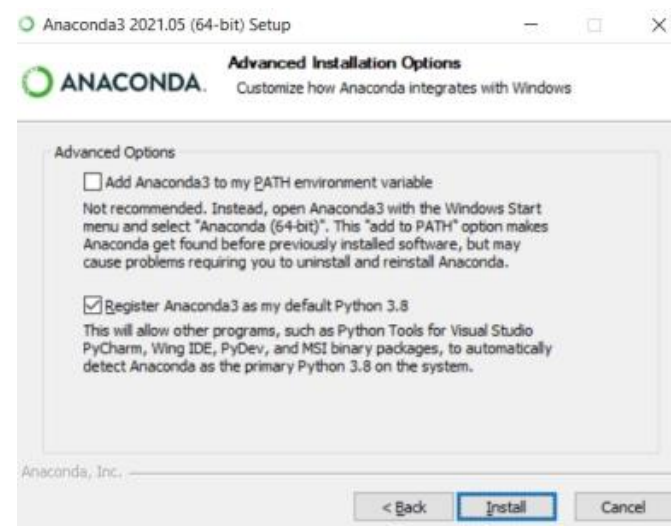
<https://www.anaconda.com/products/individual>, 选择适合自己的版本。这里选择的是64-Bit Graphical Installer (477 MB)



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。



未完, 接后页.....

MR. DENG

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

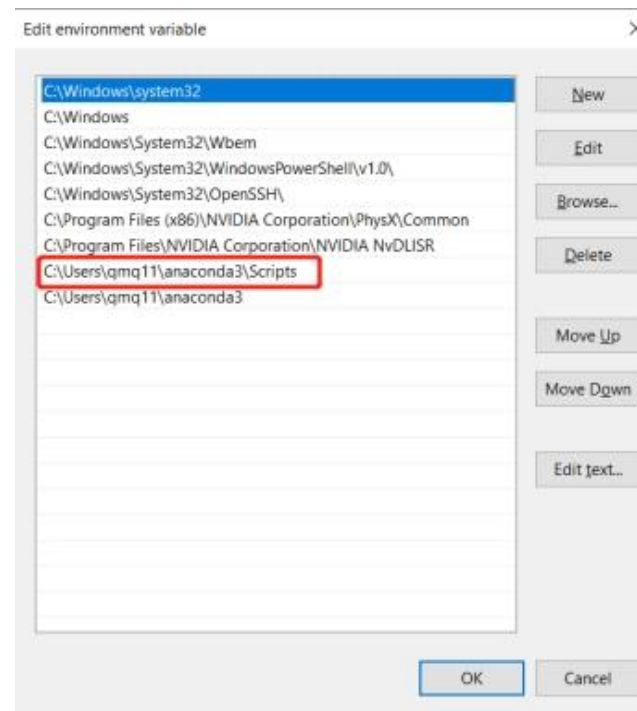
active environment : base
active env location : C:\Users\qmq11\anaconda3
shell level : 1
user config file : C:\Users\qmq11\.condarc
populated config files :
  conda version : 4.10.1
  conda-build version : 3.21.4
  python version : 3.8.8.final.0
  virtual packages :
    __cuda=11.2=0
    __win=0=0
    __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。



详情见右页



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。

◆ 安装 odrivetool

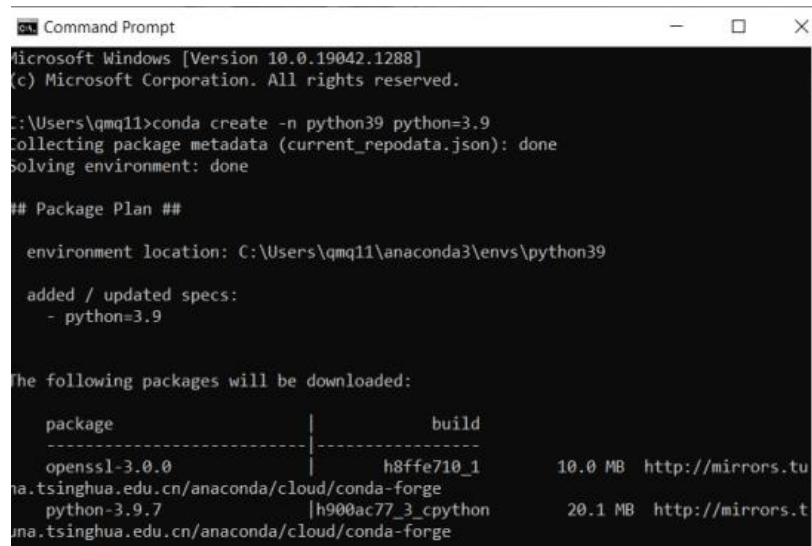
A 打开命令行cmd，创建虚拟环境：

```
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\qqm11>conda create -n python39 python=3.9
Collecting package metadata (current_repodata.json): done
Solving environment: done

## Package Plan ##

  environment location: C:\Users\qqm11\anaconda3\envs\python39

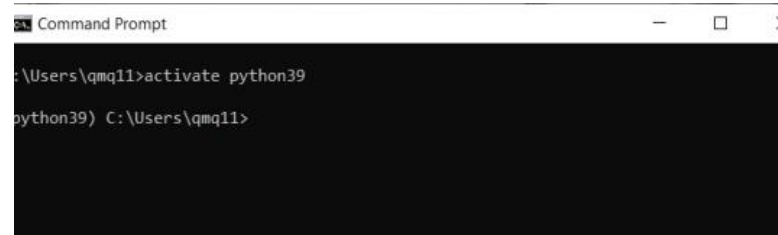
added / updated specs:
- python=3.9

The following packages will be downloaded:

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

B 激活刚刚新建的环境

```
conda activate python39
```

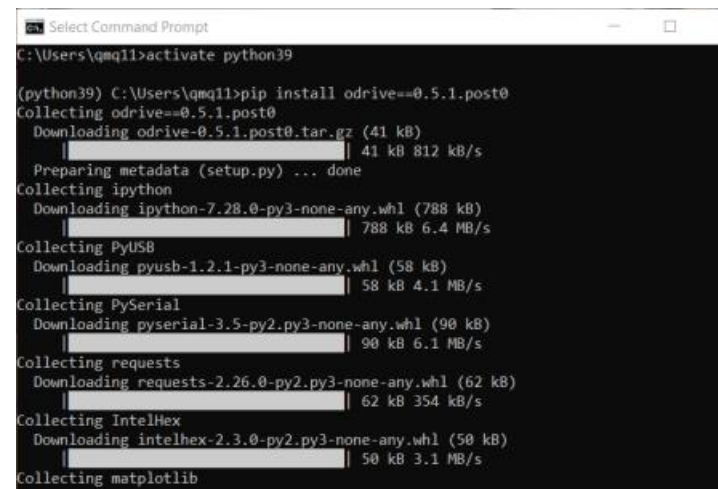


```
Command Prompt

C:\Users\qqm11>conda activate python39
(python39) C:\Users\qqm11>
```

C 安装 odrivetool

```
pip install odrive==0.5.1.post0
```



```
Select Command Prompt

C:\Users\qqm11>activate python39
(python39) C:\Users\qqm11>pip install odrive==0.5.1.post0
Collecting odrive==0.5.1.post0
  Downloading odrive-0.5.1.post0.tar.gz (41 kB)
    |#####| 41 kB 812 kB/s
  Preparing metadata (setup.py) ... done
Collecting ipython
  Downloading ipython-7.28.0-py3-none-any.whl (788 kB)
    |#####| 788 kB 6.4 MB/s
Collecting PyUSB
  Downloading pyusb-1.2.1-py3-none-any.whl (58 kB)
    |#####| 58 kB 4.1 MB/s
Collecting PySerial
  Downloading pyserial-3.5-py2.py3-none-any.whl (90 kB)
    |#####| 90 kB 6.1 MB/s
Collecting requests
  Downloading requests-2.26.0-py2.py3-none-any.whl (62 kB)
    |#####| 62 kB 354 kB/s
Collecting IntelHex
  Downloading intelhex-2.3.0-py2.py3-none-any.whl (50 kB)
    |#####| 50 kB 3.1 MB/s
Collecting matplotlib
```

D 输入 odrivetool 测试是否安装成功。

E 如果出现以下错误：

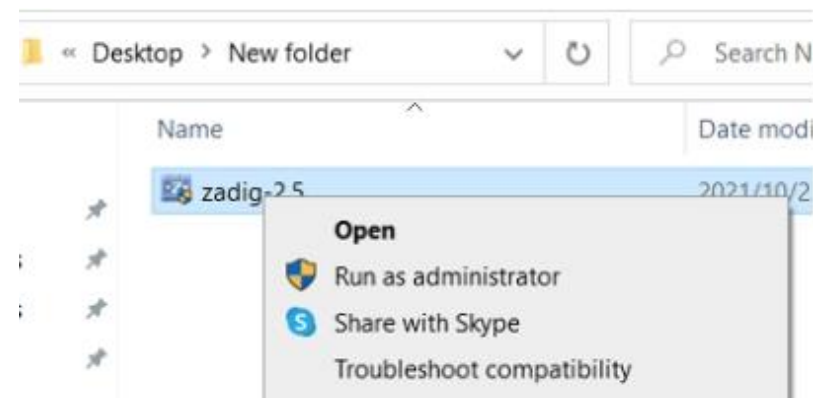
```
~\anaconda3\envs\python39\lib\site-packages\fbre\utils.py in __init_
159     self._print_lock = threading.Lock()
160     if platform.system() == 'Windows':
--> 161         self._stdout_buf = win32console.GetStdHandle(win3
162
163     def indent(self, prefix=' '):
NameError: name 'win32console' is not defined
```

执行：

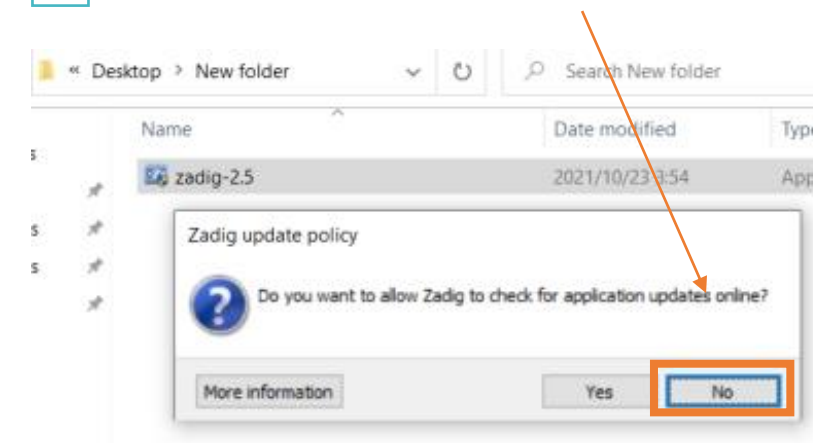
```
pip install pywin32 == 225
```

2. Zadig 配置

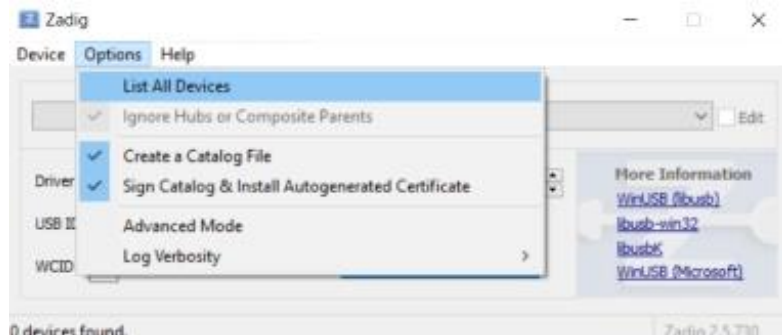
A 以 管理员权限 运行zadig-2.5



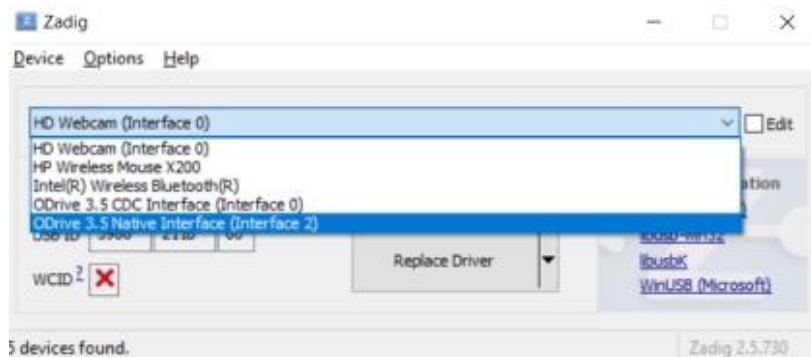
B 弹出更新提醒，选择no（不更新）



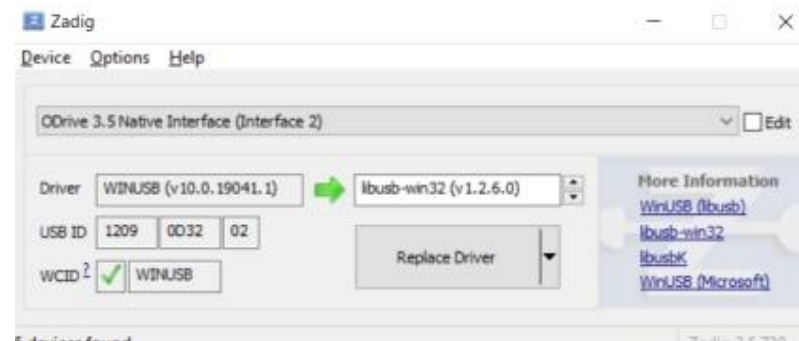
C 选择 Options 中的 List All Devices



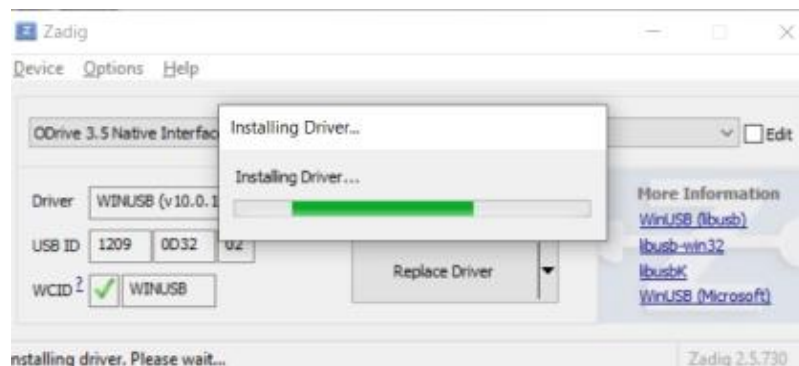
D 下拉菜单中，选择 Odrive 3.5 Native Interface (Interface 2)



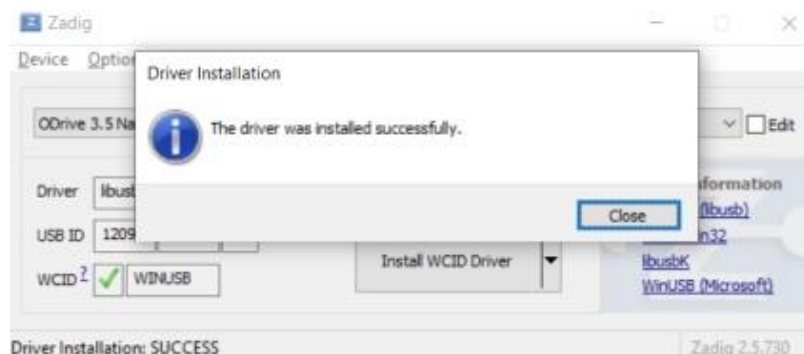
E 选择上下箭头，选择 libusb-win32(v 1.2.6.0)，点击 Replace Driver



F 等待驱动替换完成



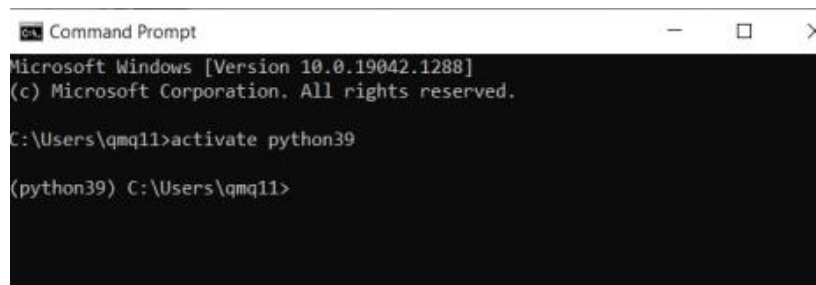
G 完成后，关闭 Zadig 窗口



3. 环境测试

A 插入USB，板上的蓝灯亮起

B 激活环境 (Python39)



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

D 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
```

五、开始使用

硬件清单

- ✓ Deng FOC
- ✓ 电机
- ✓ **ABI编码器（带板载编码器型忽略）**
- ✓ 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）

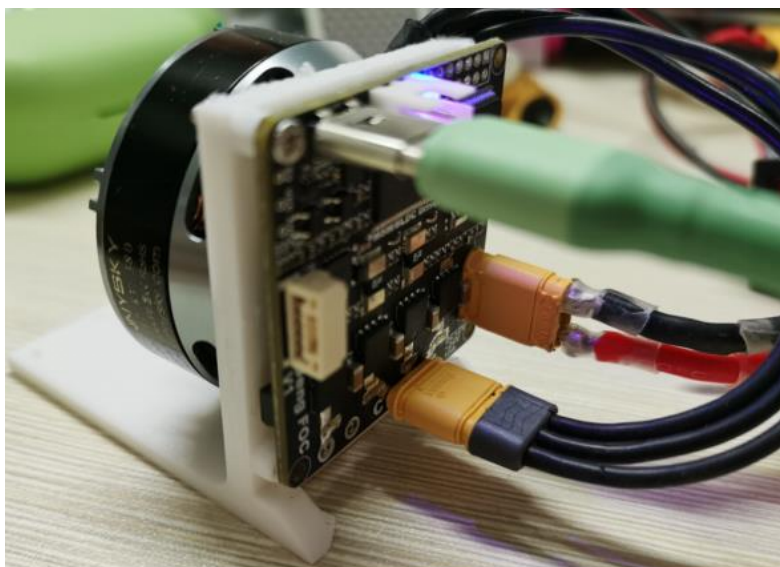
1. 硬件连接

- ◆ 电机轴粘上磁铁：（磁铁必须为 **径向磁铁**，已随板附赠）



- ❖ 若使用的是**板载编码器型**，固定DengFOC板子和电机，使DengFOC背面的编码器对准磁铁，间隔1-3mm左右。
- ❖ 若使用的是**外接编码器型**，则固定编码器模块和电机，使编码器芯片对准磁铁，同样距离1-3mm左右。

- ◆ 接上USB，电机以及电源。下图为板载编码器型的连接示意。图中的模型可在售后群下载自行打印。



MR. DENG

- ◆ 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()
```

MR. DENG

2. 一键配置

Deng FOC 准备了 python 程序可以一键配置参数和校准。在程序的开头说明并定义了所需的基本参数可按实际情况修改，运行后即可配置所有参数，并进行校准。

针对不同电机，主要需要设置耗散电阻值、制动回流电流值、电机的极对数，KV值以及编码器的位置读取模式、CPR值（暂时只支持选择 AS5047P 的ABI 模式和 SPI 模式）。其他参数可详细参考 5.2部分及第六章。程序中参数设置的位置如右图：

///

```
# 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": 30, # 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": -3 # 制动回流电流值
                  }

# 电机相关参数
motor_parameter = {"pole_pairs": 12, # 电机的极对
                  "motor_type": MOTOR_TYPE_HIGH_CURRENT, # 电机类型
                  "cur_lim": 40, # 电机电流限制 (A)
                  "cal_cur": 10, # 电机校准电流限制 (A)
                  "cal_vol": 1, # 电机校准电压限制 (v)
                  "requested_cur_range": 80, # 电流采样范围 (A)
                  "kv": 340 # 电机kv值
                  }

# 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": 5, # 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
                    }
```

///

///

在设置完所需的参数后，我们就可以对电机进行参数配置和参数校准了。

A 在终端中激活 4.1 中安装 odrive 的 python 环境

```
D:\>conda activate python39
```

B 在 python 程序所在的目录下运行程序

```
D:\Deng_FOC>python "Deng FOC Setting and calibrating.py"
```

C ABI模式下按照提示进行参数配置与校准，如右图所示

```
是否需要设置参数? (Y/N)y
正在设置电机参数
选择编码器模式:
1.AS5047P-ABI(每次上电必须来回转以校准编码器)
2.AS5047P-SPI
1
正在设置编码器为ABI模式
Setting controller base param
需要设置该电机为:
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
3
设置电机为位置模式
是否需要校准电机? (Y/N)y
正在校准电机.....
校准电机成功
是否需要校准编码器? (Y/N)y
正在校准编码器.....
校准编码器成功
是否需要上电进入闭环模式? (Y/N)y
正在测试闭环.....
检查是否进入闭环模式? (Y/N)y
是否保存设置? (Y/N)y
是否需要重启? (Y/N)y
```


D 针对于板载编码器型：由于ODrive 对 SPI 编码器的支持存在一点问题，如果需要使用AS5047P 的 SPI 模式，则需要按照以下步骤执行：

第一步：执行程序

```
是否需要设置参数？ (Y/N)y
正在设置电机参数
选择编码器模式：
1.AS5047P-ABI(每次上电必须来回转以校准编码器)
2.AS5047P-SPI
2
正在设置编码器为SPI模式
Setting controller base param
需要设置该电机为：
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
3
设置电机为位置模式
是否需要校准电机？ (Y/N)n
是否需要校准编码器？ (Y/N)n
是否需要上电进入闭环模式？ (Y/N)n
是否保存设置？ (Y/N)y
是否需要重启？ (Y/N)y
```

首先先配置参数，但不要校准电机、编码器以及进入闭环模式，然后保存设置，重启。

第二步：再次执行程序

```
是否需要设置参数？ (Y/N)n
是否需要校准电机？ (Y/N)y
正在校准电机.....
校准电机成功
是否需要校准编码器？ (Y/N)y
正在校准编码器.....
校准编码器成功
是否需要上电进入闭环模式？ (Y/N)y
正在测试闭环.....
检查是否进入闭环模式？ (Y/N)y
是否保存设置？ (Y/N)y
是否需要重启？ (Y/N)y
```

不再次配置参数，而是进行校准步骤。

E 如果在参数设置过程中出现错误，程序会停止操作并打印出错误原因，如下图当极对数设置不正确，导致编码器校准出错。

```
是否需要校准编码器? (Y/N)y
正在校准编码器.....
axis0
  axis: Error(s):
    AXIS_ERROR_ENCODER_FAILED
  motor: no error
  fet_thermistor: no error
  motor_thermistor: no error
  encoder: Error(s):
    ENCODER_ERROR_CPR_POLEPAIRS_MISMATCH
  controller: no error
axis1
  axis: no error
  motor: no error
  fet_thermistor: no error
  motor_thermistor: no error
  encoder: no error
  controller: no error
```

3. Deng FOC 参数设置

◆ Deng FOC 驱动板参数设置

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
}
```

◆ 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
                   "motor_type": MOTOR_TYPE_HIGH_CURRENT, # 电机类型
                   "cur_lim": 40, # 电机电流限制 (A)
                   "cal_cur": 10, # 电机校准电流限制 (A)
                   "cal_vol": 1, # 电机校准电压限制 (v)
                   "requested_cur_range": 80, # 电流采样范围 (A)
                   "kv": 340 # 电机kv值
}
```

◆ 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。

需要说明的是，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。

///

◆ 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

4. Deng FOC电机校准

一键配置和校准程序已经包含了电机的校准的程序。在首次使用某种电机时或清除配置后，必须进行校准。

校准程序包括分别执行的电机的校准、编码器校准，以及电机位置闭环测试，如下：

```
是否需要校准电机？ (Y/N)y  
正在校准电机.....  
校准电机成功  
是否需要校准编码器？ (Y/N)y  
正在校准编码器.....  
校准编码器成功  
是否需要上电进入闭环模式？ (Y/N)y  
正在测试闭环.....  
检查是否进入闭环模式？ (Y/N)y
```

- ◆ 校准程序同样默认保存了不同的编码器读取模式下不同的开机启动流程。
- ◆ 编码器读取模式为ABI模式时，上电开机时必须进行编码器校准，所以所设置的开机模式为电机预校准（即无“哔”一声），开机校准编码器（即开机校准后电机来回转一下0），然后进去位置闭环模式。

- ◆ 编码器读取模式为SPI绝对编码器模式时，在首次使用时校准过一次后，后续则不再需要重新校准编码器，所以所设计的开机模式为开机校准电机（即“哔”一声），预校准编码器（即不会来回转），然后进入位置闭环模式。

5. 控制

◆ 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。

◆ 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。

◆ 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。

六、参数说明

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

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 0.05Ω 。

◆ 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。

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]
```

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	odrv0.axis0.encoder.config.mode	ENCODER_MODE_INCREMENTAL	MODE_SPI_ABS_AMS
编码器CPR	odrv0.axis0.encoder.config.cpr	4000	16384
SPI模式CS引脚	odrv0.axis0.encoder.config.abs_spi_cs_gpio_pin		8
带宽	odrv0.axis0.encoder.config.bandwidth	3000	

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

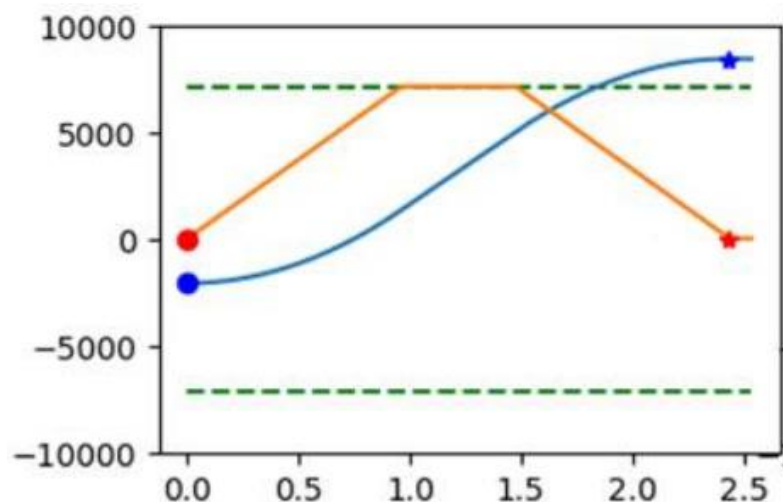
```
odrv0.axis0.controller.config.vel_ramp_rate[turn/sec]
```

■ 负载惯量

```
odrv0.axis0.controller.config.inertia[Nm/(turn/s²)]
```

❖ 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.jerk_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
```

七、常见问题及其解决

1. 连接直流电源后红灯闪灭并伴随咔哒声
2. 复位后，USB连接正常，连接直流电源后失去响应
3. 复位后USB连接异常
4. 电源电压测量值始终为12.0V
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
6. 电机进入闭环模式后不受控制地往复运动
7. 电机校准失败，报错
8. 编码器校准失败，报错

问题一

连接直流电源后红灯闪灭并伴随咔哒声

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

- 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
- 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

问题二


复位后，USB连接正常，连接直流电源后失去响应

 可能原因

逆变电路MOS G极虚焊

问题三

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 22Ω ，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

问题四

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


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

问题五

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

电机接触不良

 定位问题&参考解决方法

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

问题六

电机进入闭环模式后不受控制地往复运动

 可能原因

LP5907损坏

✓ 定位问题&参考解决方法

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

问题七

电机校准失败，报错

“ERROR_PHASE_RESISTANCE_OUT_OF_RANGE”

 可能原因

一般为电机端接线接触不良

✓ 定位问题&参考解决方法

检查三相线与板子是否接触良好。

问题八

编码器校准失败，报错

“ERROR_CPR_POLEPAIRS_MISMATCH”

 可能原因

- (1) 电机极对数设置错误
- (2) 编码器模式或CPR设置错误
- (3) 使用SPI模式没有按照5.2配置
- (4) 电机磁铁距离编码器过远，要求距离为1-3mm

✓ 定位问题&参考解决方法

按照上述可能原因逐一排查。