

# FOC电机控制软硬件设计及动手试验

Ricky Wen  
Motor control AE  
2021.11.01



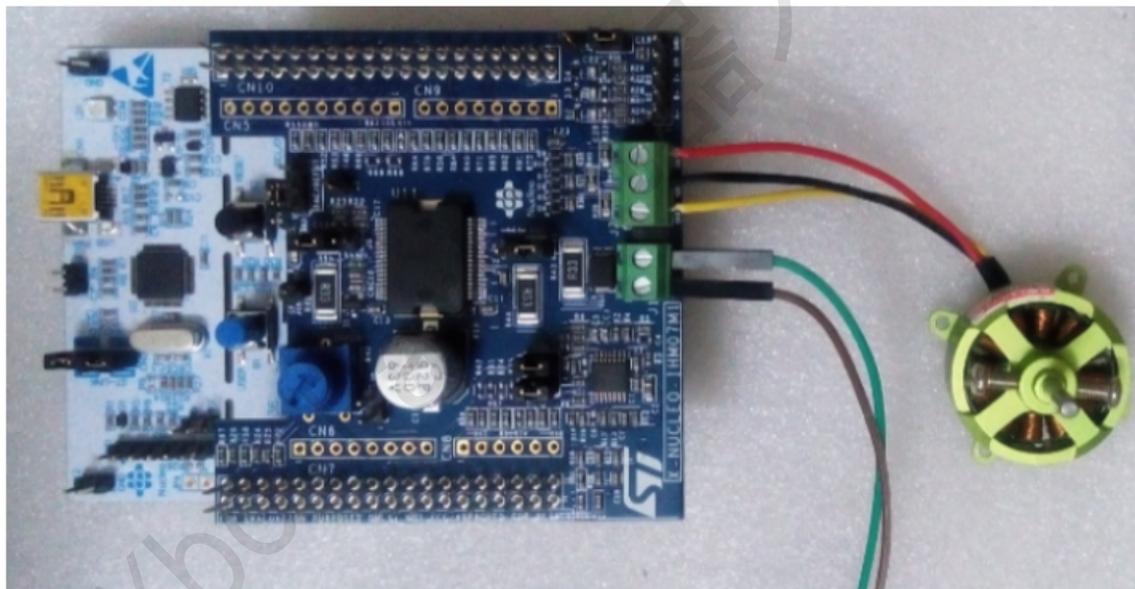
# 目录

培训内容及主要时间流程:

- 1、简单介绍*P-NUCLEO-IHM002*
- 2、*P-NUCLEO-IHM002*电机驱动硬件设计原理图讲解
- 3、*Demo*操作演示
- 4、*Motor Control Workbench*软件调试经验分享
- 5、测试电机参数
- 6、*FOC*控制电机调试技巧补充说明
- 7、动手实践

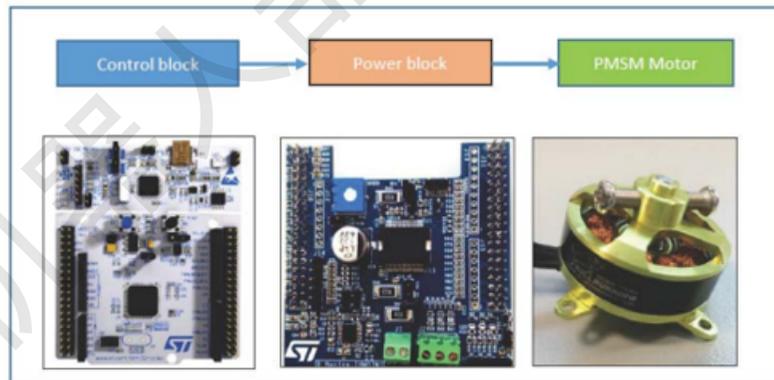
# P-NUCLEO-IHM002

STM32 Nucleo 包 P-NUCLEO-IHM002 是一个套件，包括：X-NUCLEO-IHM07M1 板、NUCLEO-F302R8 板、电机和直流电源（参见下图）



# STM32 Nucleo包的系统架构

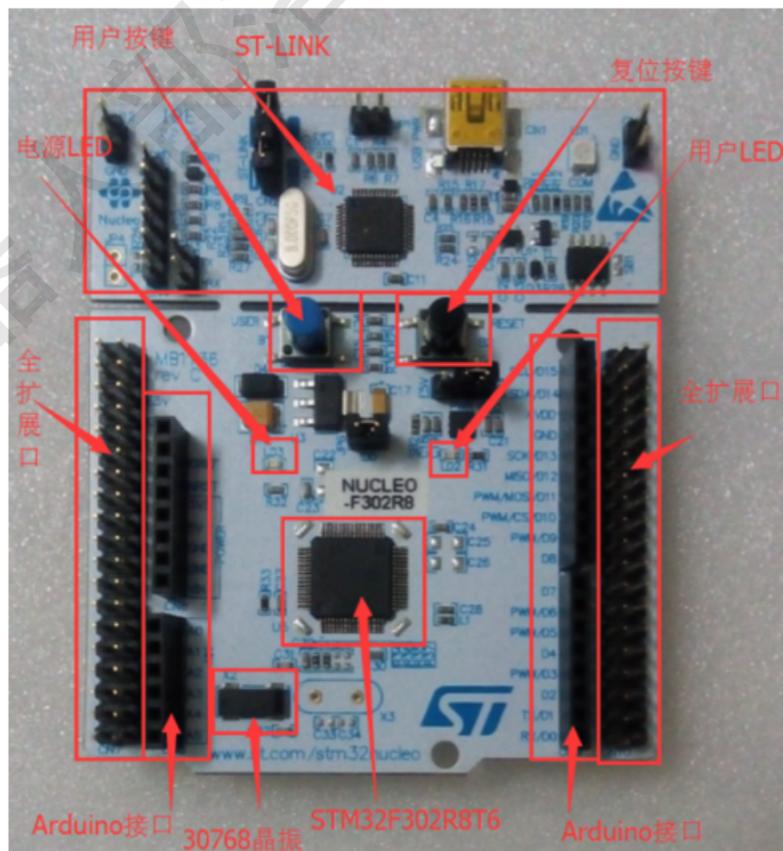
- The P-NUCLEO-IHM002 pack is composed of four main blocks (see *Figure 2*):
- Control block: NUCLEO-F302R8 MCU board
- Power block: X-NUCLEO-IHM07M1
- PMSM motor: Bull-Running BR2804-1700 kV
- Power supply unit (12 V, 2 A)



# NUCLEO-F302R8

NUCLEO板子最上面是一个ST-Link，通过SWD方式连接目标微控制器，下部的主控微控制器和接口(Arduino和Morpho接口)，以及复位按钮、用户LED和用户按钮。

右图是具体NUCLEO-F302R8核心板功能分布：



# NUCLEO-F302R8核心板的主要功能特点:

(1)STM32F302R8T6微控制器。

(2)支持Arduino UNO R3 Shield扩展板，微控制器所有IO口引脚通过排针座引出。

(3)ST-LINK/V2-1调试器。

(4)3个LED：一个USB通讯LED、一个电源LED、一个用户LED。

(5)2个机械按键：复位RESET、用户USER B1。

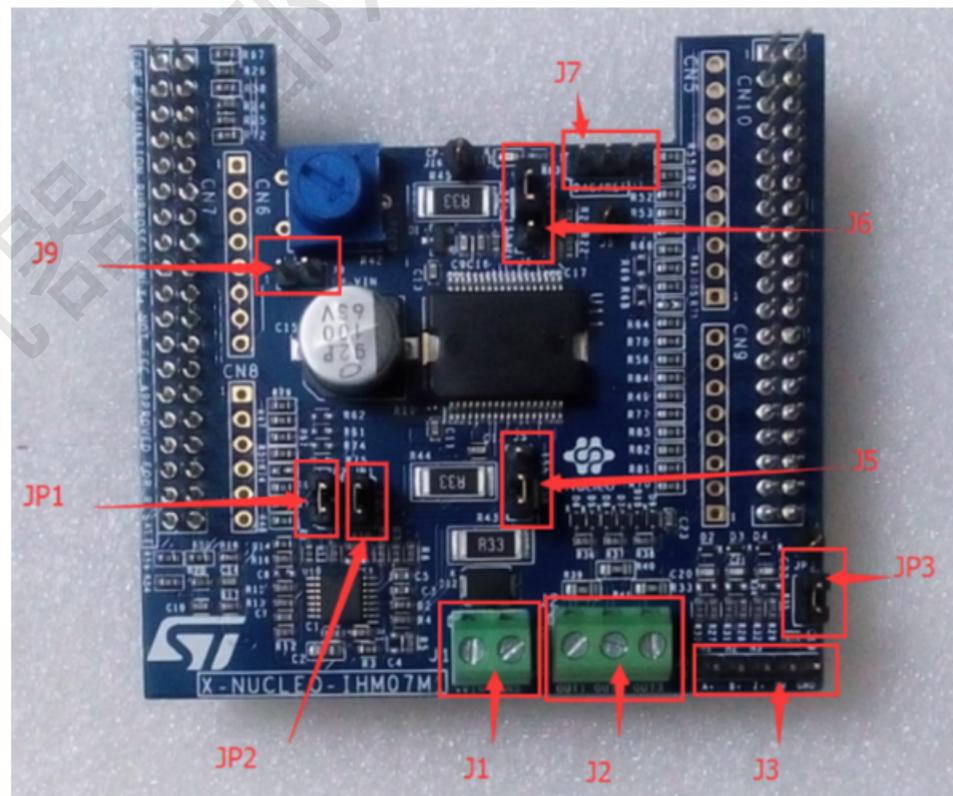
(6)USB接口的3个不同功能：虚拟串口、容量存储、调试接口。

(7)3种不同供电方式：mini USB接口供电;IO引脚用电;通过Arduino UNO R3 Shield接口供电。

(8)支持Keil、IAR、embed在线IDE的设计工具。

# X-NUCLEO-IHM07M1扩展板

X-NUCLEO-IHM07M1是ST公司专门为STM32 Nucleo核心板设计的基于L6230驱动器的三相无刷直流电动机驱动扩展板。它提供一种在STM32 Nucleo工程中的低成本、简单易用的三相无刷直流电动机控制解决方案。



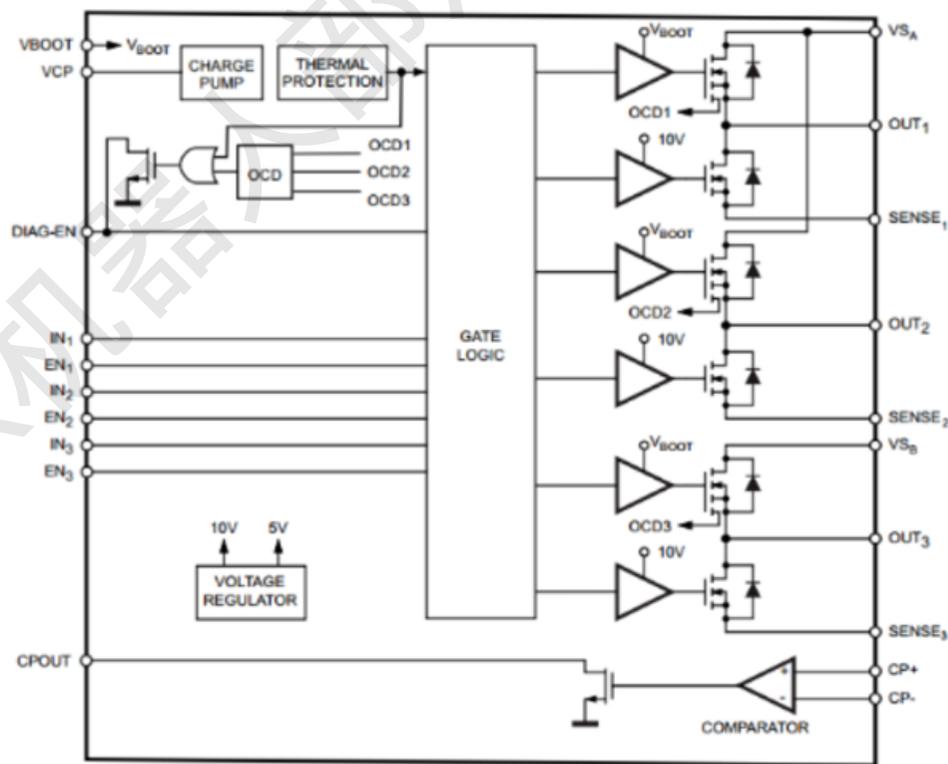
# X-NUCLEO-IHM07M1扩展板主要参数如下:

- (a)无刷直流电动机和永磁同步电动机的三相驱动板
- (b)通常的工作电压范围在8-48V直流电
- (c)2.8A的输出峰值电流(1.4A均方根电流)
- (d)操作频率高达100KHz
- (e)非耗散过流检测和保护
- (f)热检测和过热保护
- (g)与ST的梯形6步算法和 矢量控制算法完全兼容
- (h)完全支持有传感器和无传感器模式
- (i)为电动机电流检测可配置为3电阻或1电阻检测的跳线
- (j)提供霍尔/编码器形式的电机传感器接口和电路
- (k)提供DAC、GPIO等的调试接口
- (l)板载电位器接口可控制电机速度
- (m)具有预留的测试点
- (n)兼容STM32 Nucleo核心板， 配备ST morpho接口

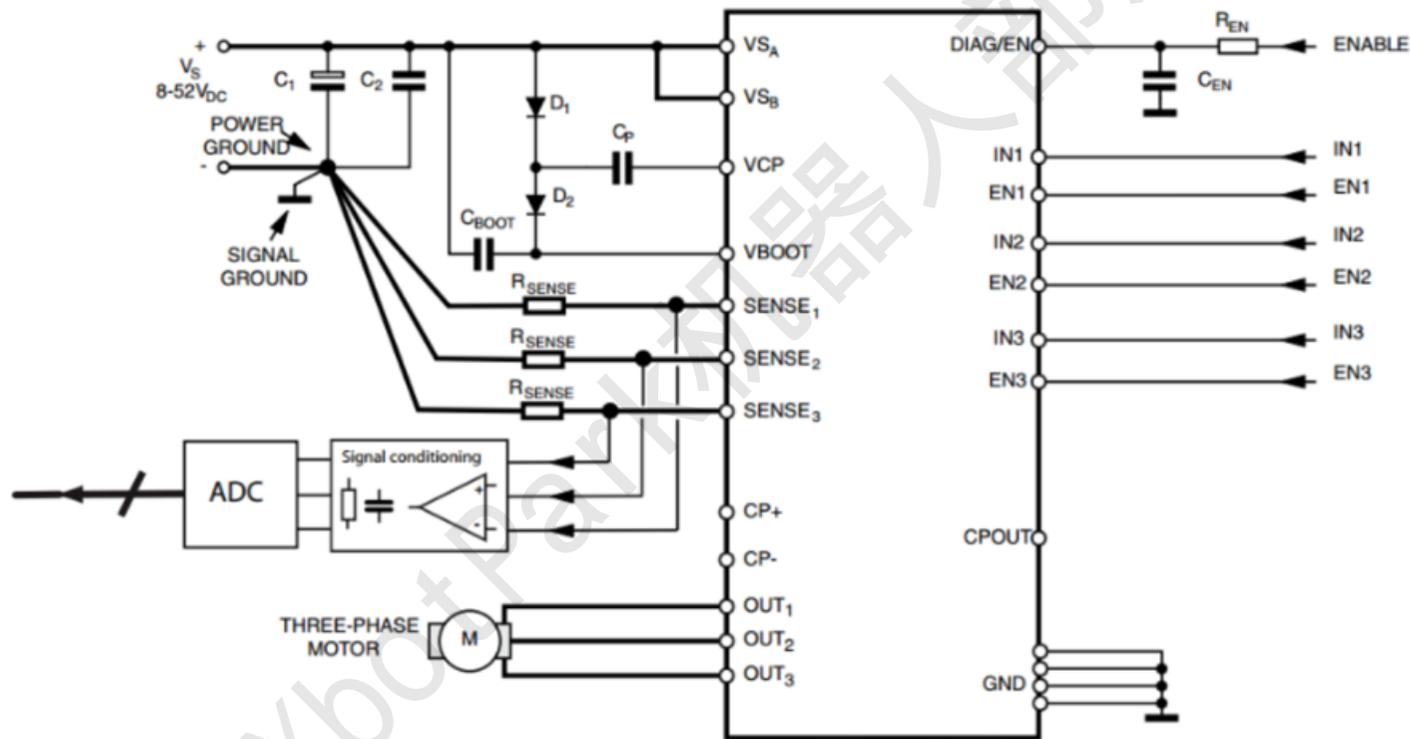
# L6230在矢量控制算法中的典型电路

其主要特点为:

- a) 工作电源电压为8至52 V
- b) 2.8 A输出峰值电流(1.4 A RMS)
- c)  $R_{DS(ON)}$  0.73 $\Omega$ (典型值)。
- d) 集成快速续流二极管
- e) 工作频率高达100 kHz
- f) 非耗散过流检测和保护

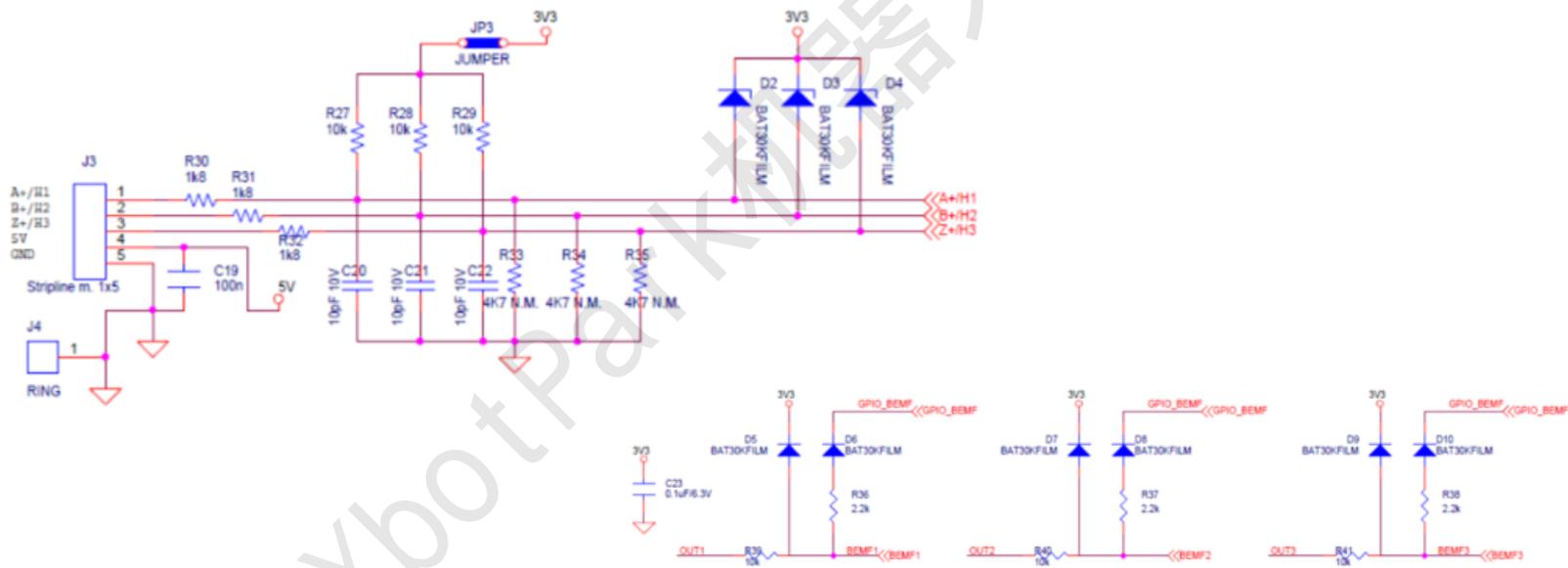


# L6230在矢量控制算法中的典型电路

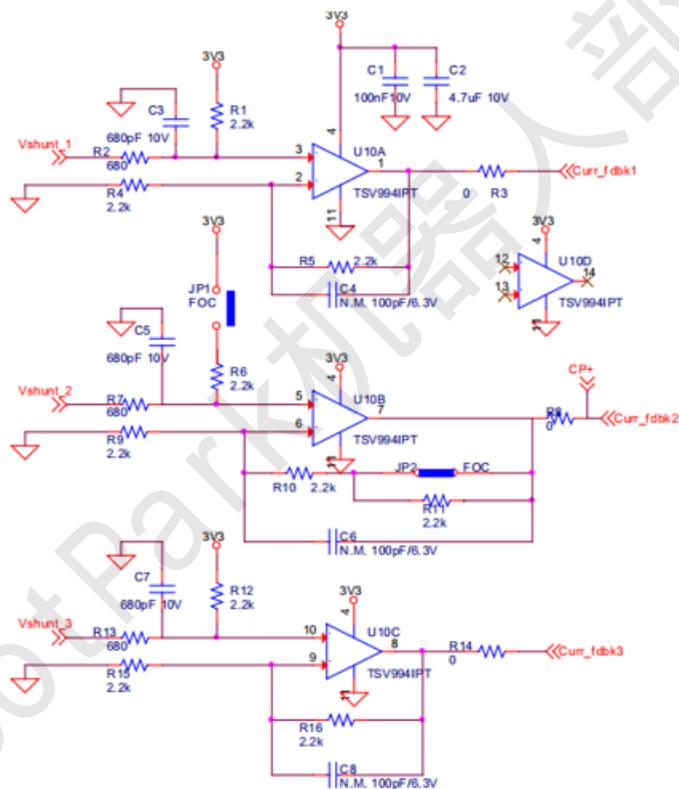


# 电机位置检测电路

X-NUCLEO-IHM07M 1扩展板上提供了电动机位置检测电路，可以选择2中方法，一种是有传感器方案(使用霍尔/编码器电机的转速传感器)，另一种是无传感器方案



# 电机电流采样电路



# PMSM Motor

P-NUCLEO-IHM001套件选择了奔牛公司的BR2804-1700Kv-1型三相无刷直流电动机，可用于航模使用，十分小巧。

Model	KV (rpm/v)	Volate (v)	Motor Weight (g) AppROX	NoLoad Current (A)	NoLoad Speed (rpm)	Load Current (A)	Load Speed (rpm)	Prop	Pull(g)	G/A
2804- 2300	2300	11.1	20	0.8	24830	10.8	14970	7035	380	35
2804- 1700	1700	11.1	23	0.6	18000	5.6	12540	7035	257	46

# Demo操作演示

XbotPark机器人部落

# MotorControl Workbench软件调试经验分享

MotorControl Workbench软件：

- 1、通过选择相应的电机开发套件（板子）或者自定义，配置或修改硬件接口，软件相关参数等；后台调用CUBE软件等来生成响应的工程代码；生成后的工程文件可以直接用IAR或KEIL MDK打开；也可以通过CUBE软件来修改相关的底层接口并由CUBE软件重新生成代码；
- 2、监控及显示FOC的相关参数，及显示实测速度和目标速度的相关波形数据等；
- 3、主要用来测试电机参数，调试电机驱动板等；

# 实验平台

硬件:

NUCLEO-F302(或F303)底板;

驱动板:

IHM07M1 3Sh(低压3A)或 )

请事先下载安装软件:

1)STM32 PMSM FOC Software Development Kit - MC library

(5.3.2及以上版本);

2)STM32CubeMX 4.26及以上 版本;

3) KEIL MDK 5.0 及以上版本



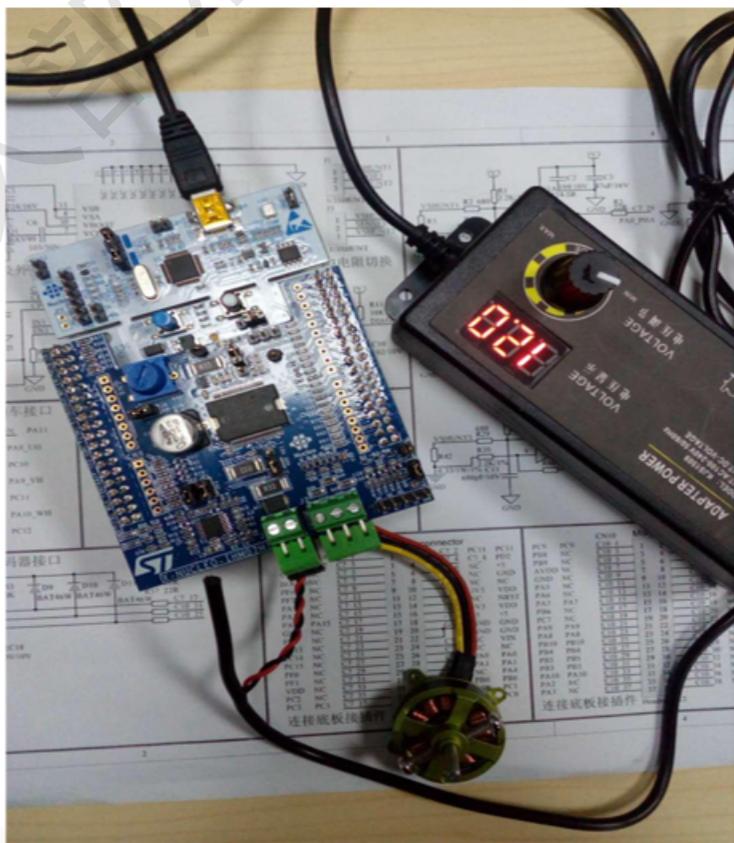
## Demo接线

按照图示接线：

左边接插件接电源（左正右负），  
接电源之前最好将电源调整到12V左右（根据对应电机选电压）；

右边接插件：接电机线（电机线任意接，接错的话电机正反转方向可能是反的）；

首次拆开开发套件（板子已预装程序）：上电后且接上USB线，按套件底板的蓝色按钮，电机正常转，再按下电机停止运行（如果电机不能转，请先按下黑色复位按钮，再按蓝色运行。



# 打开Workbench如下图所示

ST Motor Control Workbench

File Tools Help Documentation

New Project Load Project About Help

Motor Profiler  
Motor Control Suite

Recent Projects

Filename	Type	SDK	MCUs	control board	power board	motor
Noname003.stmxc	SINGLE	4.2.0	STM32F103 High Density	Custom	Custom	Custom
Noname003.stmxc	SINGLE	4.2.0	STM32F103 High Density	Custom	Custom	Custom
two motor foc4 hall.stmxc	DUAL	4.2.0	STM32F103 High Density	STEVAL-IHM039V1	MB459	Shinano Motor
two motor foc4 hall.stmxc	DUAL	4.2.0	STM32F103 High Density	STEVAL-IHM039V1	MB459	Shinano Motor
Noname003.stmxc	SINGLE	4.2.0	STM32F103 High Density	Custom	Custom	Custom
two motor foc4 hall.stmxc	DUAL	4.2.0	STM32F103 High Density	STEVAL-IHM039V1	MB459	Shinano Motor

Example Projects

Filename	Type	SDK	MCUs	control board	power board	motor
P-NUCLEO-HM001 BullRunning kit, 3-Shunt, Sensorless	SINGLE	5.1.0	STM32F301x8/6 - STM32F302x8/6	P-NUCLEO-HM001 3Sh - board: NUCLEO-F302R8	P-NUCLEO-HM001 3Sh - board: X-NUCLEO-HM07M1	Bull Running E
NUCLEO-F303RE + IHM07M1 + BullRunning motor	SINGLE	5.1.0	STM32F303xE	NUCLEO-F303RE	X-NUCLEO-HM07M1	Bull Running E
NUCLEO-F303RE + IHM08M1 + Shinano motor	SINGLE	5.1.0	STM32F303xE	NUCLEO-F303RE	X-NUCLEO-HM08M1	Shinano LA05
STEVAL-IHM042V1 inverter + 2 Shinano motors	DUAL	5.3.0	STM32F303CC	STEVAL-IHM042V1	STEVAL-IHM042V1	Shinano LA05
Speed ramp with Potentiometer on P-NUCLEO-HM001 kit	SINGLE	5.1.0	STM32F301x8/6 - STM32F302x8/6	P-NUCLEO-HM001 3Sh - board: NUCLEO-F302R8	P-NUCLEO-HM001 3Sh - board: X-NUCLEO-HM07M1	Bull Running E
Speed ramp with Potentiometer on P-NUCLEO-HM001 kit -- Advanced	SINGLE	5.1.2	STM32F301x8/6 - STM32F302x8/6	P-NUCLEO-HM001 3Sh - board: NUCLEO-F302R8	P-NUCLEO-HM001 3Sh - board: X-NUCLEO-HM07M1	Bull Running E
Gimbal motor	SINGLE	5.1.0	STM32F303xE	NUCLEO-F303RE	X-NUCLEO-HM16M1	GimBal motor, GBM2804H-10
Saw tooth speed ramp	SINGLE	5.1.0	STM32F303xE	NUCLEO-F303RE	X-NUCLEO-HM07M1	Bull Running E
Speed ramp and CCMRAM	SINGLE	5.1.0	STM32F303xE	NUCLEO-F303RE	X-NUCLEO-HM07M1	Bull Running E
Dual Drive and CCMRAM	DUAL	5.1.0	STM32F303xE	STM32303E-EVAL	STEVAL-IHM045V1	Shinano LA05
Free RTOS on P-NUCLEO-HM001 kit	SINGLE	5.2.0	STM32F301x8/6 - STM32F302x8/6	NUCLEO-F302R8	X-NUCLEO-HM07M1	BullRunning
Power Factor Correction	SINGLE	5.1.0	STM32F103 High Density	STEVAL-IHM034V2	STEVAL-IHM034V2	Allen Bradley TL-A220P-HJ
STM32G0R1 based, Single-Shunt configuration using HALL	SINGLE	5.3.0	STM32G0R1BR	STM32G0R1BR-EVAL	STEVAL-IHM023V3	Shinano LA05

life. augmented

# 为什么要先对电机参数进行识别？

FOC的电流环的PI参数，主要根据电机参数、机械参数、转动惯量等进行计算的。

电机主要参数：

额定电压

额定电流

最高转速

极对数

电阻

电感

反电动势常数

# 电机主要参数手动测试方法

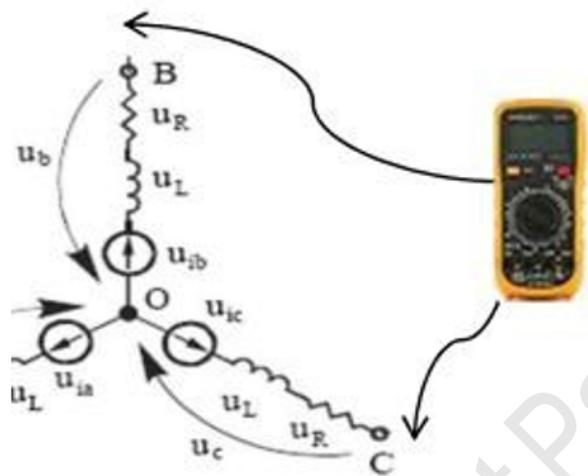
电机参数最好找电机厂要，然后自己在厂家基础上修改，或自己手动测试或用软件测试。

电机主要参数手动测试方法：

- ◆ **额定电压**：一般指母线电压（直流电压），电机的正常工作电压，航模电机一般是12V，普通直流无刷电机一般是24V或48V；高压电机一般是110V、220V、310V等；
- ◆ **额定电流**：一般根据功率除电流即可，或选额定电流，最大电流；
- ◆ **最高转速**：电机的最高转速，普通电机一般3000转左右，航模电机根据KV值表示，如1500KV，则最高转速为 $1500 \times 10 = 15000$ 转；
- ◆ **极对数P**：电机极数（一般指转子磁极数，磁铁个数）除2，如16极电机，则极对数P为8；一般极数比槽数（一般指定子绕组，线圈）多2，比如航模电机槽数为12，则极数为14，极对数是7。

# 电机主要参数手动测试方法

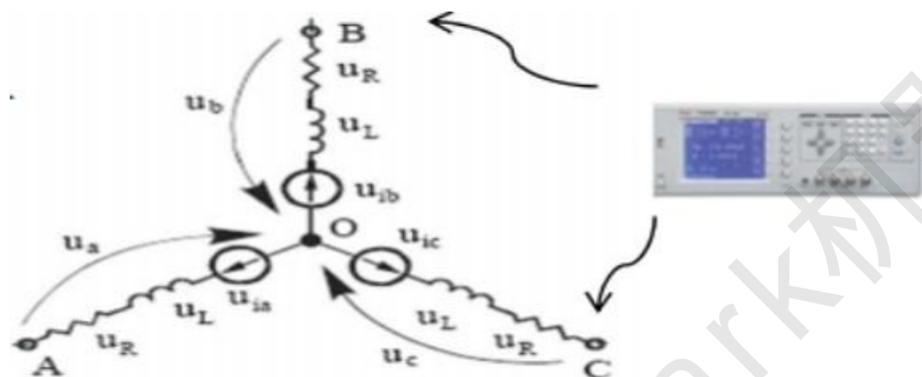
电阻（相电阻 $R_s$ ）：万用表测量电机2相线电阻再除2



如上图有图所示则相电阻 $R_s=1.1/2=0.55$ 欧

# 电机主要参数手动测试方法

电感（相电感 $L_s$ ）：电桥以**30K**频率测**2**相线电感值再除**2**



如上图有图所示则相电感 $L_s=594/2=297\mu\text{H}=0.297\text{mH}$

电桥频率选**30K**左右或**20K**（因为驱动正常工作PWM频率为10到30K）

# 电机主要参数手动测试方法

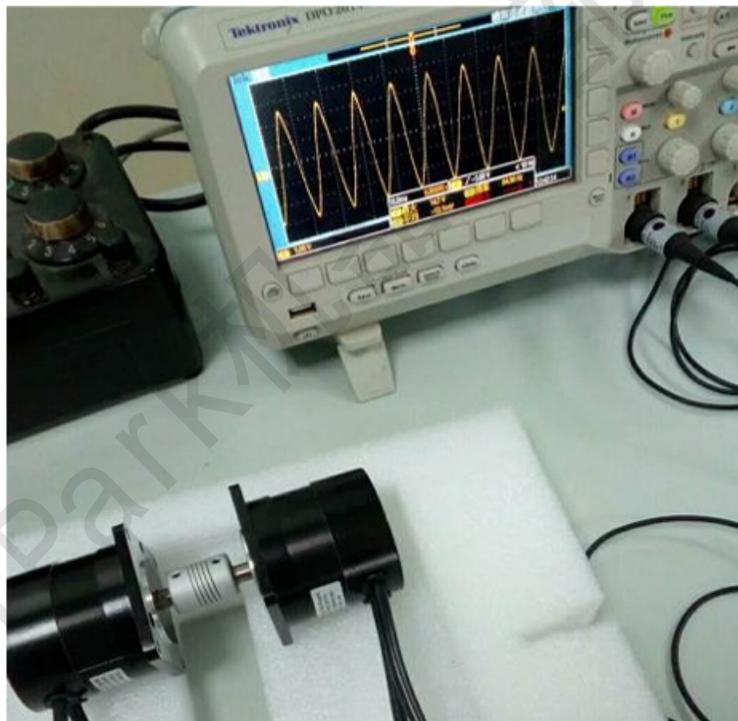
## ◆反电动势常数（Ke）

用联轴器，一个电机带动另一个电机（被测电机）以**1000转**转速运转，或用电动工具手枪钻来转；示波器探头接电机一相，示波器探头地接 另外任意一相，转动负载，测出反向电动势波形。取中间的任意一个 正弦波，测出其峰峰值**Vpp**和频率**f**。

$$K_e = 1000 * P * \frac{V_{pp}}{2 * 1.732 * 60 * f}$$

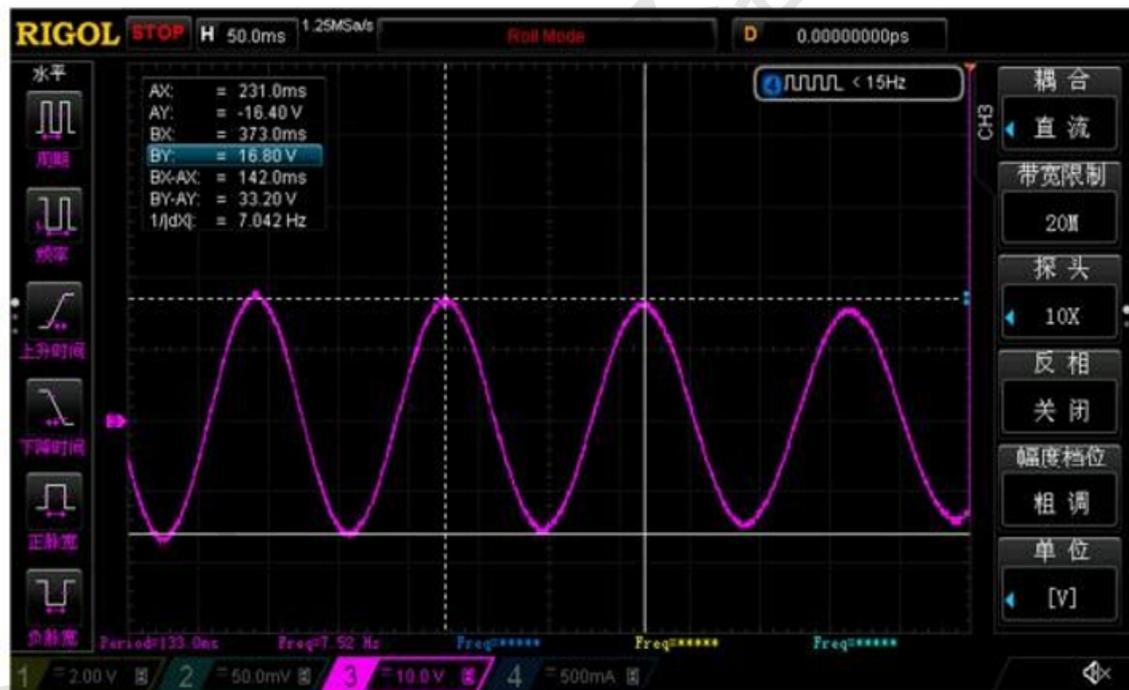
# 电机主要参数手动测试方法

## 测反电动势常数 ( $K_e$ )



# 电机主要参数手动测试方法：

## 测反电动势常数（Ke）



测量峰峰值 $V_{pp}$ 为33.2V，频率 $f$ 为7.042Hz，极对数为4，则：反电动势 $K_e=90.73$

$$K_e = 1000 * 4 * \frac{33.2}{2 * 1.732 * 60 * 7.042}$$

# MC SDK5.3电机参数识别

File Tools Help Documentation

New Project Load Project About Help

Motor Profiler  
Motion Control Suite

**Recent Projects**

Filename	Type	SDK	MCUs	control board	power board	motor
Noname003.stmcx	SINGLE	4.2.0	STM32F103 High Density	Custom	Custom	Custom
Noname003.stmcx	SINGLE	4.2.0	STM32F103 High Density	Custom	Custom	Custom
two motor foc4 hall.stmcx	DUAL	4.2.0	STM32F103 High Density	STEVAL-IHM039V1	MB459	Shinano Motor
two motor foc4 hall.stmcx	DUAL	4.2.0	STM32F103 High Density	STEVAL-IHM039V1	MB459	Shinano Motor
Noname003.stmcx	SINGLE	4.2.0	STM32F103 High Density	Custom	Custom	Custom
two motor foc4 hall.stmcx	DUAL	4.2.0	STM32F103 High Density	STEVAL-IHM039V1	MB459	Shinano Motor

**Example Projects**

Filename	Type	SDK	MCUs	control board	power board	motor
P-NUCLEO-IHM001 BullRunning kit, 3-Shunt, Sensorless	SINGLE	5.1.0	STM32F301x6/8 - STM32F302x6/8	P-NUCLEO-IHM001 3Sh - board: NUCLEO-F302R8	P-NUCLEO-IHM001 3Sh - board: X-NUCLEO-IHM07M1	Bull Running E
NUCLEO-F303RE + IHM07M1 + BullRunning motor	SINGLE	5.1.0	STM32F303xE	NUCLEO-F303RE	X-NUCLEO-IHM07M1	Bull Running E
NUCLEO-F303RE + IHM08M1 + Shinano motor	SINGLE	5.1.0	STM32F303xE	NUCLEO-F303RE	X-NUCLEO-IHM08M1	Shinano LA05
STEVAL-IHM042V1 inverter + 2 Shinano motors	DUAL	5.3.0	STM32F303CC	STEVAL-IHM042V1	STEVAL-IHM042V1	Shinano LA05
Speed ramp with Potentiometer on P-NUCLEO-IHM001 kit	SINGLE	5.1.0	STM32F301x6/8 - STM32F302x6/8	P-NUCLEO-IHM001 3Sh - board: NUCLEO-F302R8	P-NUCLEO-IHM001 3Sh - board: X-NUCLEO-IHM07M1	Bull Running E

# Motor Profiler

ST Motor Profiler

ST life.augmented

Motor Profiler  
Motion Control Suite

Select Boards

Pole Pairs:  [how to detect...](#)

Speed and Current limits

Max Speed:  RPM

Max Current:  Apk 0.1 - 30 Apk

VBus:  V 7 - 400 V

Magnetic:  SM-PMSM  I-PMSM

Connect...  
Start Profile  
Save...  
Play

Electrical Model

Mechanical Model

# 点击SelectBoards选择相应的板子，选择：NUCLEO-F302R8（或303）+X-NUCLEO-IHM07M1 3Sh

The screenshot shows the 'ST Motor Profiler' SelectBoards interface. At the top, there are search filters: 'Hide obsolete boards' (checked) and 'Hide boards with warning' (unchecked). Search boxes for 'Control Board' and 'Power Board' are present. A 'Cancel' button is in the top right. The main area displays eight board options in a grid:

- NUCLEO-F302R8** (STM32F302R8T6): Active. One Motor Control connector, ST-LINK/V2 Embedded. Product Web Page.
- STEVAL-IPM05F 3Sh** (STGIF5CH60): Active. Bus voltage: 125 - 400 Vdc, DC Input voltage: 125 - 400 Vdc, Output peak current: 0.8 - 8 A. Product Web Page.
- NUCLEO-F302R8** (STM32F302R8T6): Active. One Motor Control connector, ST-LINK/V2 Embedded. Product Web Page.
- STEVAL-IPM10B 3Sh** (STGIB10CH60T5-L): Active. Bus voltage: 125 - 400 Vdc, DC Input voltage: 125 - 400 Vdc, Output peak current: 1.5 - 13 A. Product Web Page.
- NUCLEO-F302R8** (STM32F302R8T6): Active. One Motor Control connector, ST-LINK/V2 Embedded. Product Web Page.
- STEVAL-IPM15B 3Sh** (STGIB15CH60T5-L): Active. Bus voltage: 125 - 400 Vdc, DC Input voltage: 125 - 400 Vdc, Output peak current: 2 - 20 A. Product Web Page.
- NUCLEO-F302R8** (STM32F302R8T6): Active. One Motor Control connector, ST-LINK/V2 Embedded. Product Web Page.
- X-NUCLEO-IHM07M1 3Sh** (L6230PD): Active. Bus voltage: 8 - 48 Vdc, DC Input voltage: 8 - 48 Vdc, Output peak current: 0.28 - 2.8 A. Product Web Page.

打开后如下图，请选择**Connect**先连接（如连接不上则按下板子的黑色复位按钮）

Motor Profiler  
Motion Control Suite

NUCLEO-F302R8  
STM32F302R8T6



One Motor Control connector  
ST-LINK/V2 Embedded

[Product Web Page](#)

X-NUCLEO-IHM07M1 3Sh  
L6230PD



Bus Voltage: 8 - 48 Vdc  
Output peak current: 0.28 - 2.8 A

[Product Web Page](#)

Pole Pairs:  [how to detect...](#)

Speed and Current limits

Max Speed:  RPM

Max Current:  Apk      0.28 - 2.8 Apk

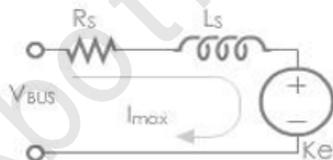
VBus:  V      8 - 48 V

Magnetic:  SM-PMSM  I-PMSM

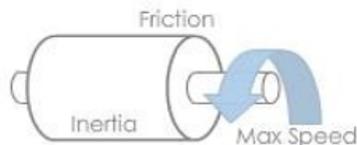
**i** Remember to properly configure the boards in Motor Control mode

- [Connect...](#)
- [Start Profile](#)
- [Save...](#)
- [Play](#)

Electrical Model



Mechanical Model



## SM-PMSM参数示例

The image shows a configuration interface for a motor. It includes several input fields and radio buttons, with red arrows pointing to them from Chinese text on the right. The parameters are:

- Pole Pairs:** An empty input box. A red arrow points to it from the text "电机极对数 (必填)".
- Speed and Current limits:** A section header.
- Max Speed:** An input box containing "16000" followed by "RPM". A red arrow points to it from the text "电机最大转速 (可选)".
- Max Current:** An input box containing "2.8" followed by "Apk". To its right is the range "0.28 - 2.8 Apk". A red arrow points to it from the text "电机允许的最大电流 (可选)".
- VBus:** An input box containing "48" followed by "V". To its right is the range "8 - 48 V". A red arrow points to it from the text "额定总线电压 (可选)".
- Magnetic:** Two radio buttons: "SM-PMSM" (which is selected) and "I-PMSM". A red arrow points to it from the text "磁体内置类型 (必填)".

Watermarks "Motor" and "XbotPark 机器人部落" are visible in the background.

# 点击测试电机参数后等待测试完成（也可停止测试）

**Motor Profiler**  
Motion Control Suite

NUCLEO-F302R8  
STM32F302R8T6

X-NUCLEO-IHM07M1 3Sh  
L6230PD

One Motor Control connector  
ST-LINK/V2 Embedded  
[Product Web Page](#)

Bus Voltage: 8 - 48 Vdc  
Output peak current: 0.28 - 2.8 A  
[Product Web Page](#)

Pole Pairs:  [how to detect...](#)

Speed and Current limits

Max Speed:  RPM

Max Current:  Apk 0.28 - 2.8 Apk

VBus:  V 8 - 48 V

Magnetic:  SM-PMSM  I-PMSM

**Remember to properly configure the boards in Motor Control mode.**

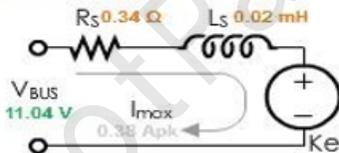
Disconnect

Stop Profile

Save...

Play

**Electrical Model** 44%



**Mechanical Model**



**Faults**

- Over voltage
- Under voltage
- Overheat
- Startup failure
- Speed feedback
- Over current

测试完成后如下图所示（测试不成功的话就重新测试下），测试成功的话**Save**和**Play**按钮功能有效，测试完成后同个电机每次的参数有些差异，但差别不大

Motor Profiler  
Motion Control Suite

④ NUCLEO-F302R8  
STM32F302R8T6



One Motor Control connector  
ST-LINK/V2 Embedded

[Product Web Page](#)

④ X-NUCLEO-IHM07M1 3Sh  
L6230PD



Bus Voltage: 8 - 48 Vdc  
Output peak current: 0.28 - 2.8 A

[Product Web Page](#)

Pole Pairs:  [how to detect...](#)

Speed and Current limits

Max Speed:  RPM

Max Current:  Apk    0.28 - 2.8 Apk

VBus:  V    8 - 48 V

Magnetic:  SM-PMSM  I-PMSM

Remember to properly configure the boards in Motor Control mode

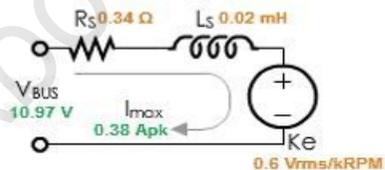
Disconnect

Start Profile

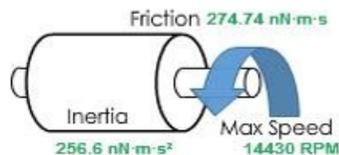
Save...

Play

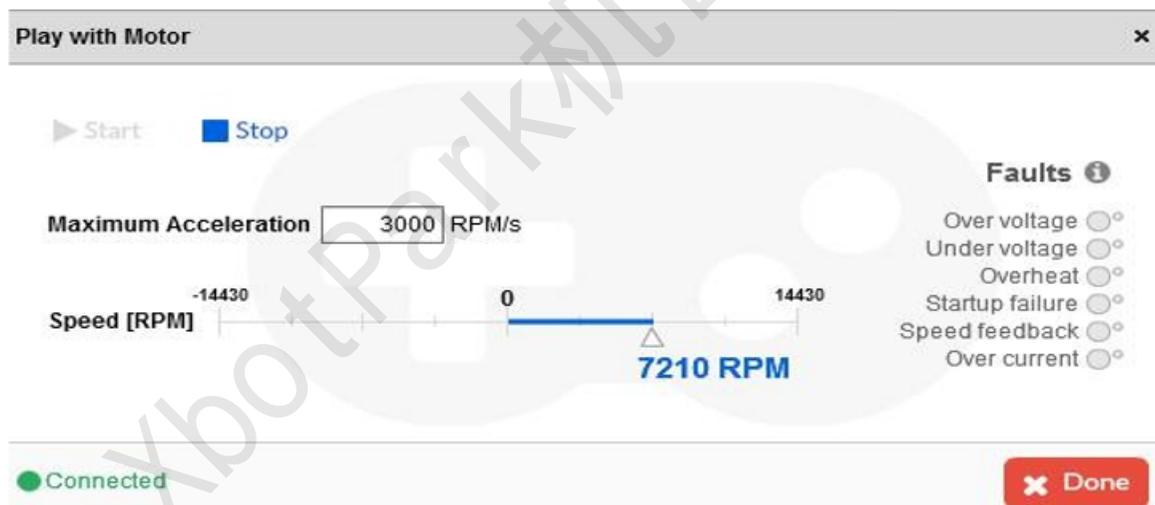
① Electrical Model



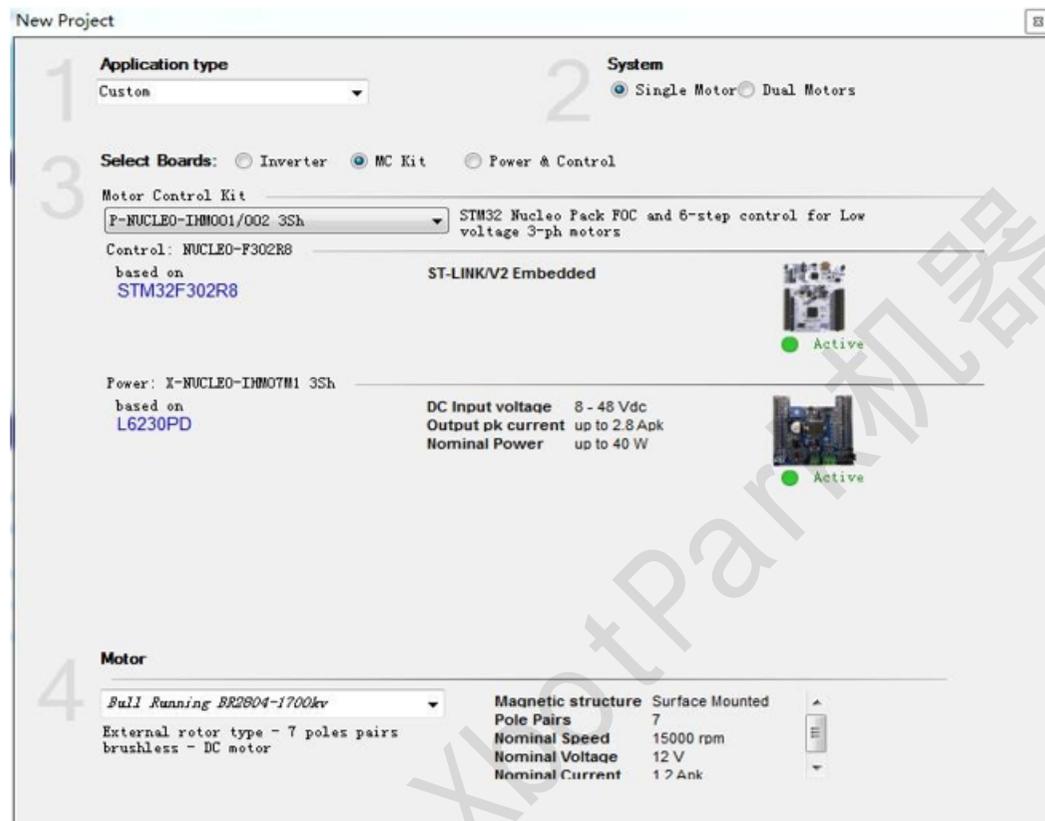
Mechanical Model



点击**Play**后，再点击**Start**即可让电机转动，通过调节中间箭头调节转速，点**Stop**按钮停止转动，点**Done**按钮关闭界面（如电机在转动则停止转动），如遇干扰等原因连接不成功，可点左下角按钮重新连接。由于是高速电机，加速度和目标转速不要设置过低，否则转不起来。



关闭电机并退出电机测试回到Workbench主界面，点击New Project新建工程，选MC Kit并选择相应的板子和配套的电机参数（也可自定义）



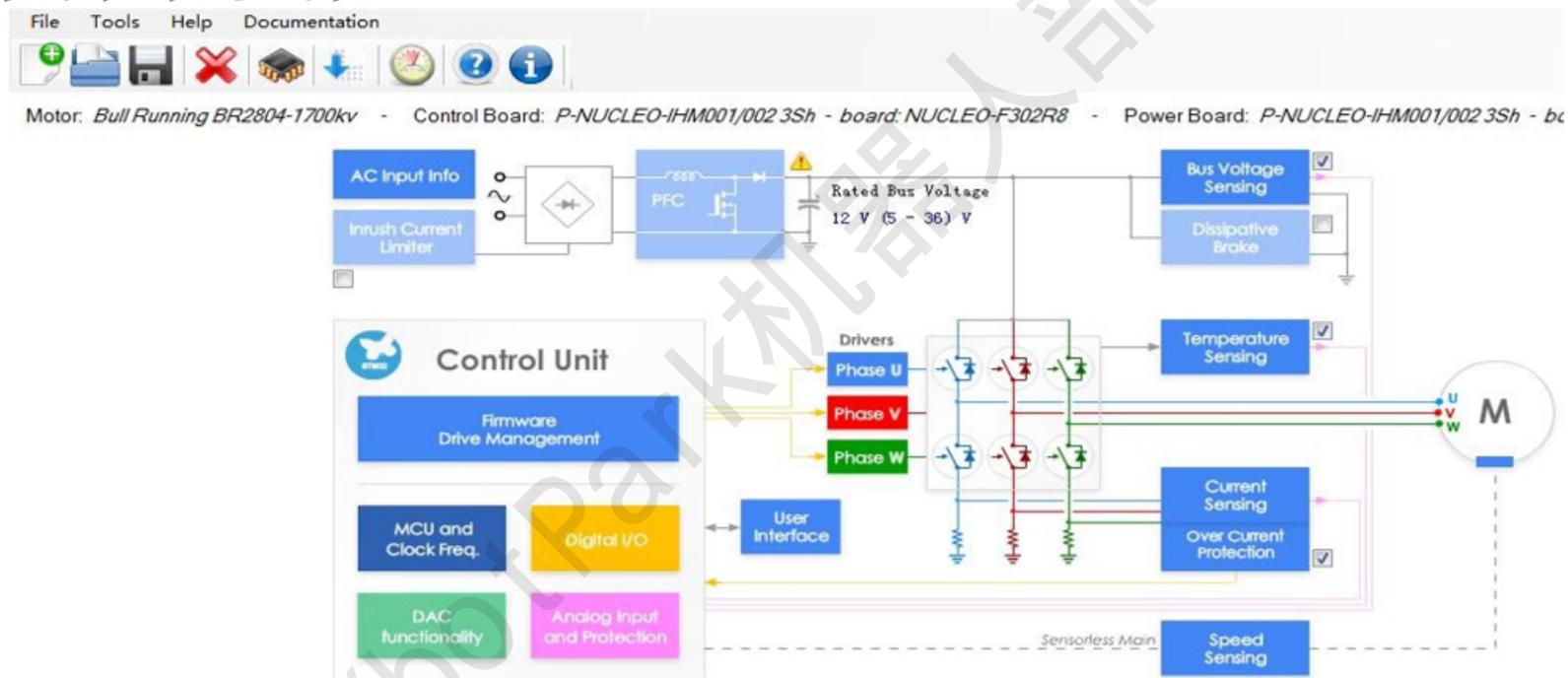
备注:

NUCLEO-F302(或F303)  
底板;

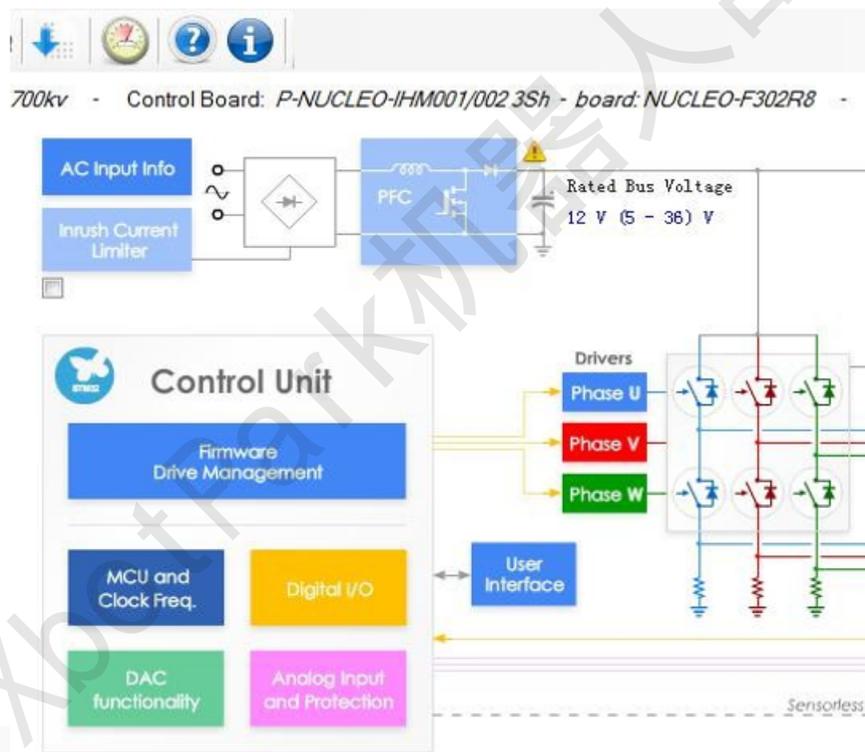
驱动板:

IHM07M1 3Sh(低压3A)  
或IHM08M1 3Sh(低压  
33A)或IPM05F 3Sh(高  
压10A)

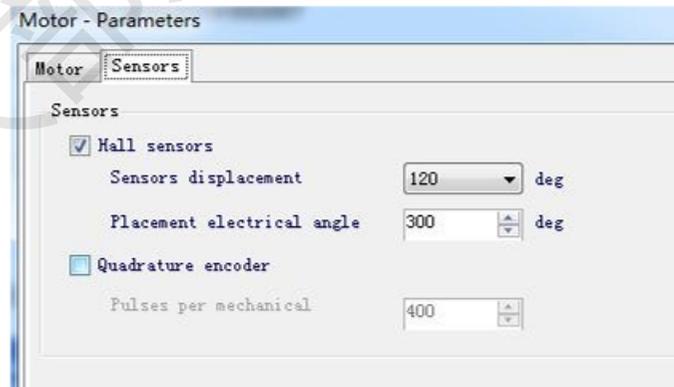
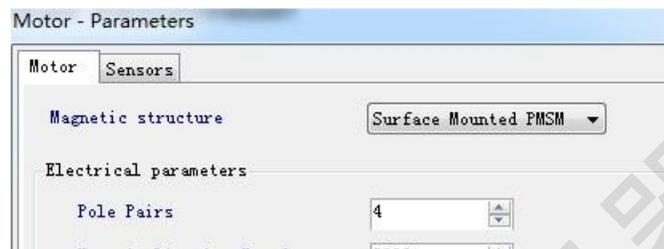
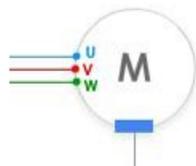
点击**OK**跳到工程硬件配置窗口界面，如下图所示，可更改相关默认参数



所有参数设置完成后或不设置，点击生成按钮生成相应的工程文件（向下箭头）

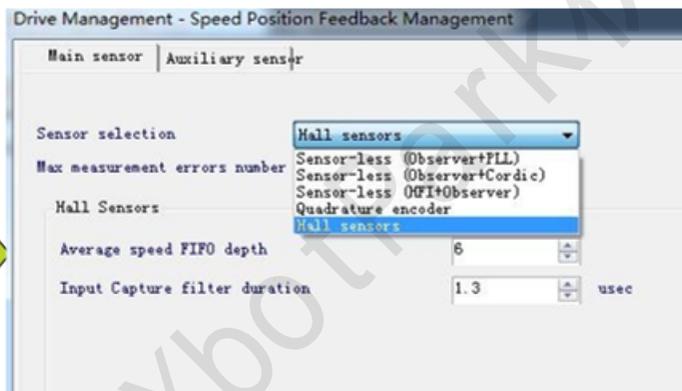


# HALL 或编码器添加（两处），无感控制可跳过

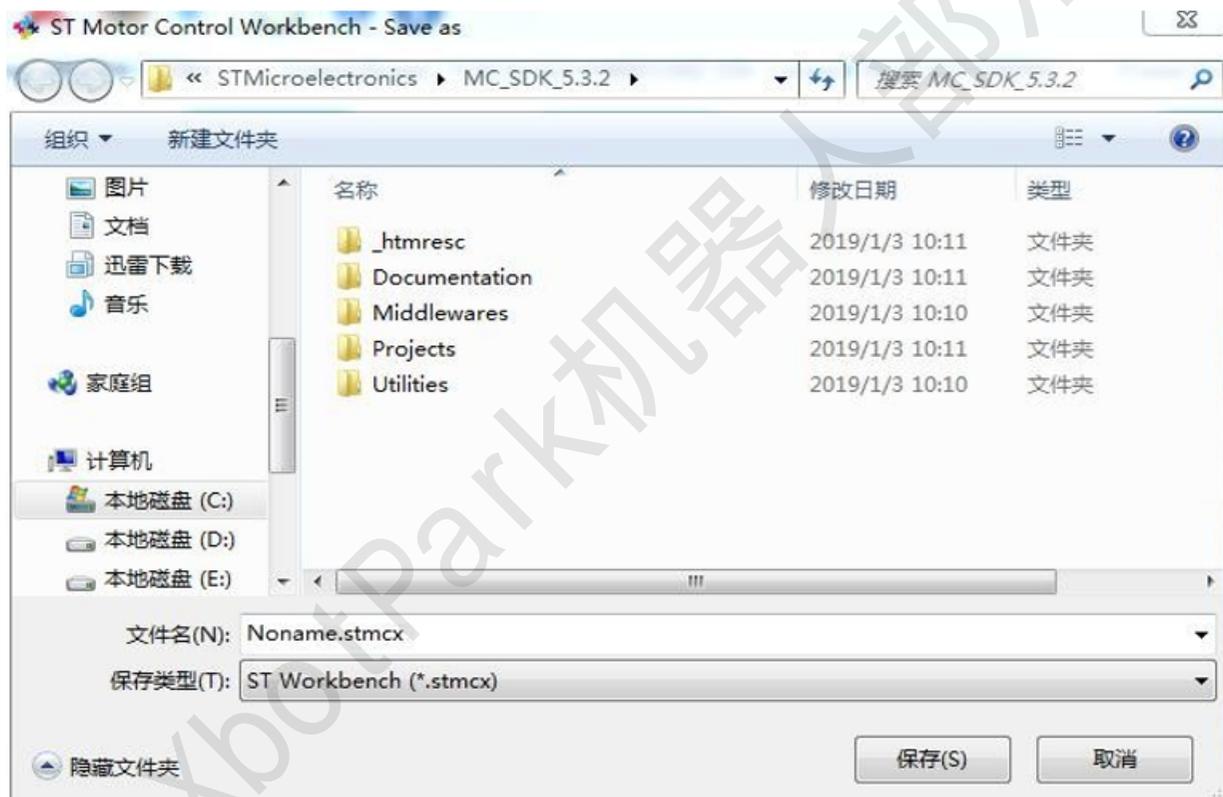


Main

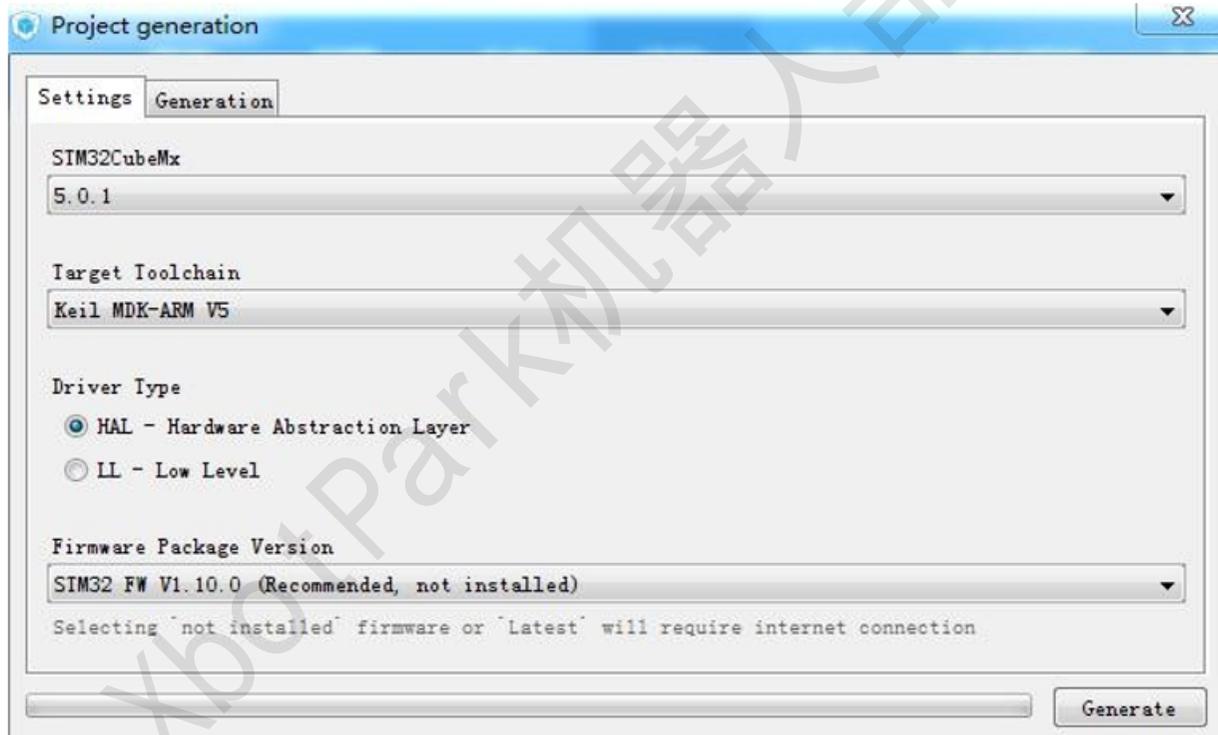
Speed Sensing



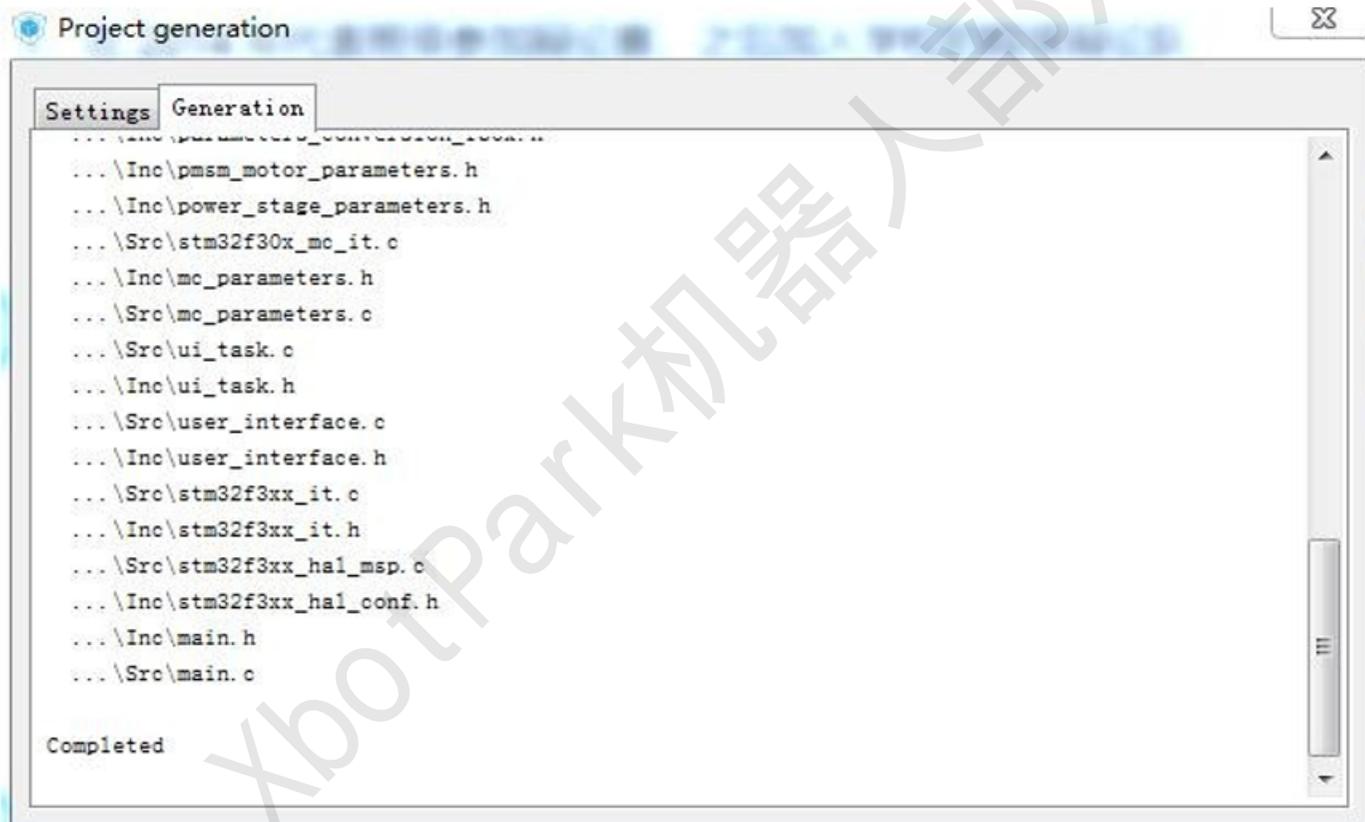
点击**OK**按钮后如下图所示，可修改工程名和选型路径等



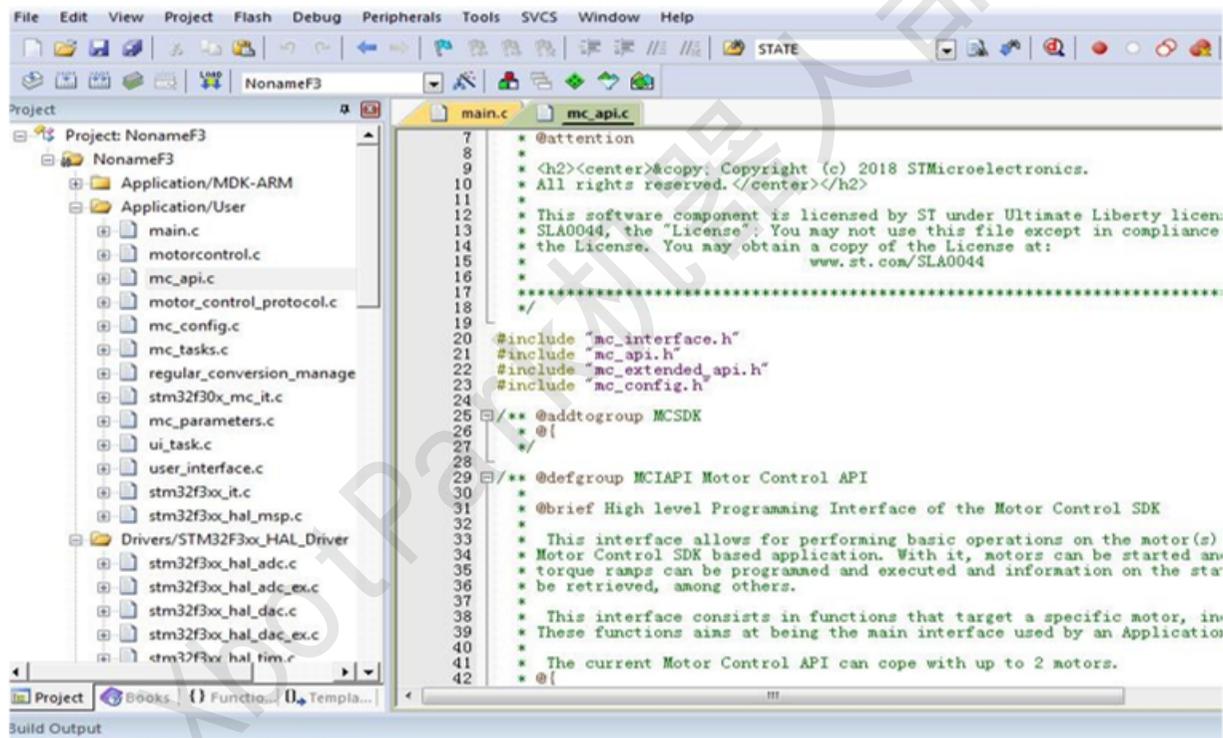
点击保存按钮后如下图所示，可生成相关的软件工程  
(MDK或IAR)



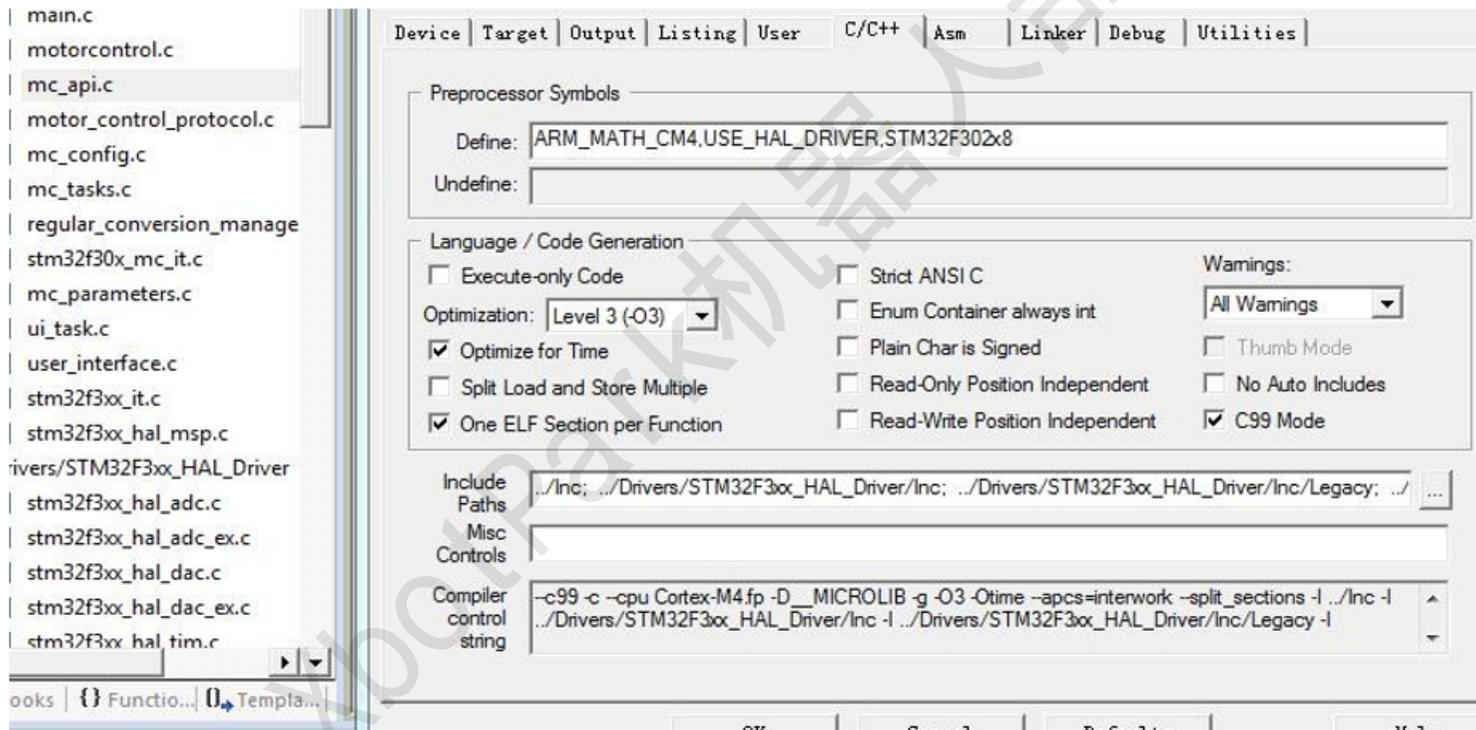
生成完后如下图所示，会显示**Completed**



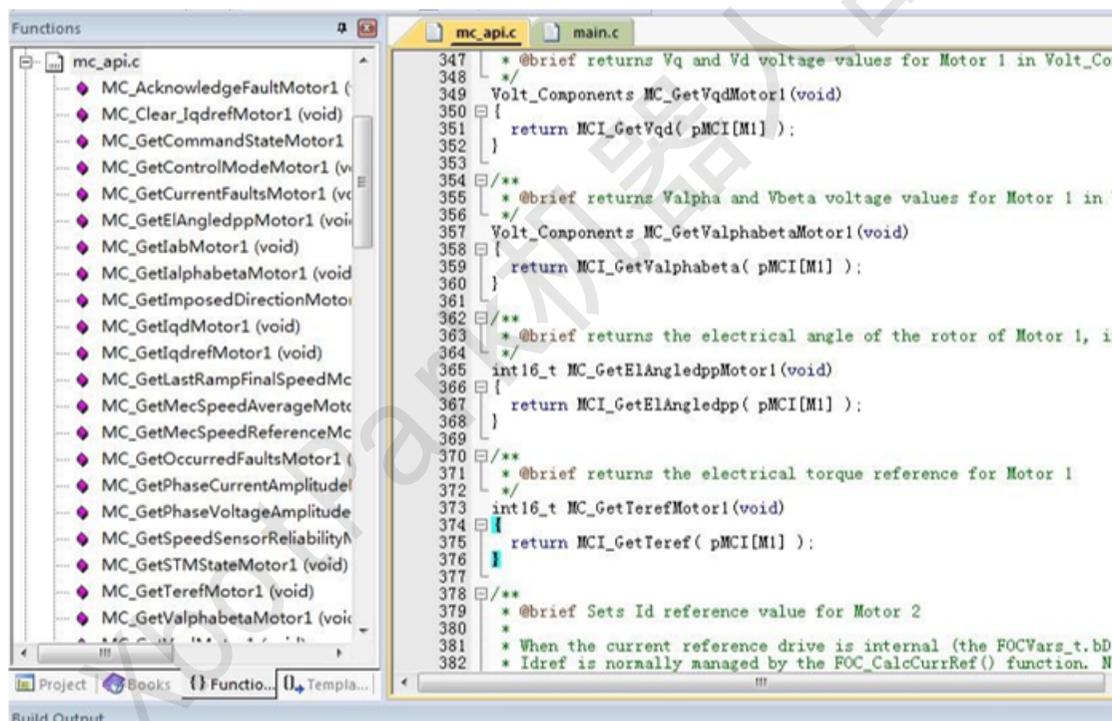
找到刚才生成的文件夹下的工程文件，打开**MDK**工程，如下图所示



优化等级设置成最高（设的太低，**FOC**库会运行报错，导致电机转不起来），并编译下工程



在mc\_api.c中有一系列函数来实现对电机的控制，主要调用mc\_api.c里的一些函数



The screenshot shows an IDE window with two panes. The left pane, titled 'Functions', displays a list of functions from the 'mc\_api.c' file, including:

- MC\_AcknowledgeFaultMotor1
- MC\_Clear\_IqdrefMotor1 (void)
- MC\_GetCommandStateMotor1
- MC\_GetControlModeMotor1 (void)
- MC\_GetCurrentFaultsMotor1 (void)
- MC\_GetElAngledppMotor1 (void)
- MC\_GetIabMotor1 (void)
- MC\_GetIalambdaMotor1 (void)
- MC\_GetIalphabetaMotor1 (void)
- MC\_GetImposedDirectionMotor1
- MC\_GetIqdMotor1 (void)
- MC\_GetIqdrefMotor1 (void)
- MC\_GetLastRampFinalSpeedMc
- MC\_GetMecSpeedAverageMotc
- MC\_GetMecSpeedReferenceMc
- MC\_GetOccurredFaultsMotor1
- MC\_GetPhaseCurrentAmplitude
- MC\_GetPhaseVoltageAmplitude
- MC\_GetSpeedSensorReliabilityM
- MC\_GetSTMStateMotor1 (void)
- MC\_GetTerefMotor1 (void)
- MC\_GetValphabetaMotor1 (void)

The right pane shows the source code for 'mc\_api.c', displaying several function definitions with their respective comments and code blocks:

```
347  * @brief returns Vq and Vd voltage values for Motor 1 in Volt_Co
348  */
349  Volt_Components MC_GetVqdMotor1(void)
350  {
351      return MCI_GetVqd( pMCI[M1] );
352  }
353
354  /**
355  * @brief returns Valpha and Vbeta voltage values for Motor 1 in
356  */
357  Volt_Components MC_GetValphabetaMotor1(void)
358  {
359      return MCI_GetValphabeta( pMCI[M1] );
360  }
361
362  /**
363  * @brief returns the electrical angle of the rotor of Motor 1, i
364  */
365  int16_t MC_GetElAngledppMotor1(void)
366  {
367      return MCI_GetElAngledpp( pMCI[M1] );
368  }
369
370  /**
371  * @brief returns the electrical torque reference for Motor 1
372  */
373  int16_t MC_GetTerefMotor1(void)
374  {
375      return MCI_GetTeref( pMCI[M1] );
376  }
377
378  /**
379  * @brief Sets Id reference value for Motor 2
380  *
381  * When the current reference drive is internal (the FOCVars_t.bd
382  * Idref is normally managed by the FOC_CalcCurrRef() function. N
```

## mc\_api.c中常用函数

- MC\_StartMotor1
- MC\_StopMotor1
- MC\_ProgramSpeedRampMotor1
- MC\_ProgramTorqueRampMotor1
- MC\_GetMecSpeedReferenceMotor1
- MC\_GetMecSpeedAverageMotor1
- MC\_GetSTMStateMotor1
- MC\_GetOccurredFaultsMotor1
- MC\_AcknowledgeFaultMotor1
- MC\_GetImposedDirectionMotor1

MC\_StartMotor1(); //电机启动

如果Motor1为IDLE状态，该命令立即执行启动电机，返回值true，反之返回命令丢弃返回false。

```
bool MC_StartMotor1(void)
{
    return MCI_StartMotor( pMCI[M1] );
}
```

MC\_StopMotor1(); //电机停止

如果Motor1为RUN或START状态，该命令立即执行停止电机，反之 被丢弃。

```
bool MC_StopMotor1(void)
{
    return MCI_StopMotor( pMCI[M1] );
}
```

MC\_ProgramSpeedRampMotor1(3000/6,2000); //设置为速度模式，设置速度指令调用此函数后hDurationms设置时间内当前速度变化到hFinalSpeed设置的目标速度

```
void MC_ProgramSpeedRampMotor1( int16_t hFinalSpeed, uint16_t  
hDurationms )  
{  
    MCI_ExecSpeedRamp( pMCI[M1], hFinalSpeed, hDurationms );  
}
```

MC\_ProgramTorqueRampMotor1(2000,2000); //设置为转矩模式，设置转矩指令调用此函数后hDurationms设置时间内当前转矩变化到hFinalTorque设置的目标转矩

```
void MC_ProgramTorqueRampMotor1( int16_t hFinalTorque, uint16_t  
hDurationms )  
{  
    MCI_ExecTorqueRamp( pMCI[M1], hFinalTorque, hDurationms );  
}
```

```
Motor1Speed=MC_GetMecSpeedReferenceMotor1();  
//获取Motor1当前指令的机械转速，数字量1代表0.1HZ  
int16_t MC_GetMecSpeedReferenceMotor1(void)  
{  
    return MCI_GetMecSpeedRef01Hz( pMCI[M1] );  
}
```

```
Motor1SpeedAverage=MC_GetMecSpeedAverageMotor1();  
//获取Motor1当前指令的平均机械转速，数字量1代表0.1HZ  
int16_t MC_GetMecSpeedAverageMotor1(void)  
{  
    return MCI_GetAvrgMecSpeed01Hz( pMCI[M1] );  
}
```

```
Motor1State=MC_GetSTMStateMotor1();
```

```
//获取Motor1状态机的电机 状态
```

```
State_t MC_GetSTMStateMotor1(void)
```

```
{
```

```
    return MCI_GetSTMState( pMCI[M1] );
```

```
}
```

Frequently used state

State	Code	Simple Description
IDLE	0	Idle
START	4	In start up process
START RUN	5	In start up process
RUN	6	Run normal
ANY_STOP	7	Stop
STOP	8	Stop
STOP_IDLE	9	Stop
FAULT_NOW	10	Fault
FAULT_OVER	11	Fault

```

Motor1Faults=MC_GetOccurredFaultsMotor1();
//获取Motor1发生过的 故障代码
uint16_t MC_GetOccurredFaultsMotor1(void)
{
    return MCI_GetOccurredFaults( pMCI[M1] );
}

```

Fault List

State	Code	Simple Description
MC_NO_ERROR	0x0000	No fault
MC_NO_FAULTS	0x0000	No fault
MC_FOC_DURATION	0x0001	FOC calculation time out
MC_OVER_VOLT	0x0002	Bus over voltage
MC_UNDER_VOLT	0x0004	Bus under voltage
MC_OVER_TEMP	0x0008	Over temperature
MC_START_UP	0x0010	Start up failure
MC_SPEED_FDBK	0x0020	Speed feedback is not reliable
MC_BREAK_IN	0x0040	Hardware break in
MC_SW_ERROR	0x0080	Software fault

`MC_AcknowledgeFaultMotor1();`//清除电机故障错误

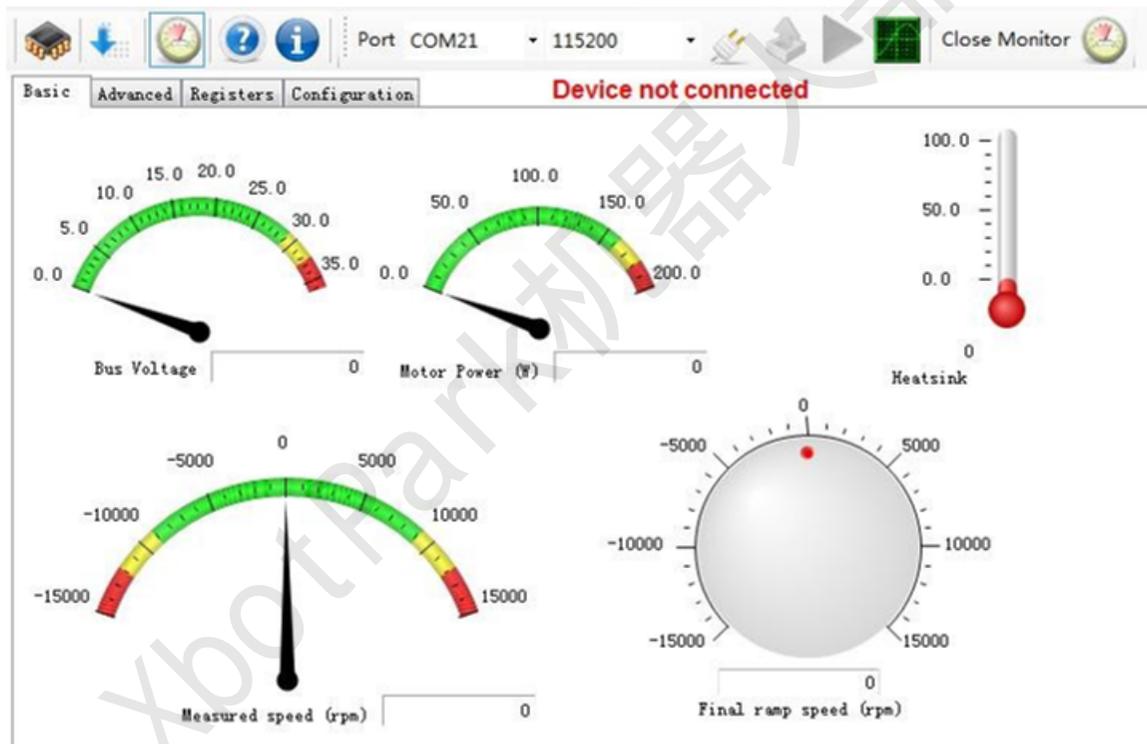
调用此函数前，如果电机发生故障。电机停留在**FAULT\_OVER**状态，并保留故障代码。调用后，状态机清除故障代码记录，并恢复到**IDLE**状态。

```
bool MC_AcknowledgeFaultMotor1( void )
{
    return MCI_FaultAcknowledged( pMCI[M1] );
}
```

`Motor1Dir=MC_GetImposedDirectionMotor1();`//返回最后一个指令设置的电机方向，如果最终的速度或转矩指令为负数返回-1，否则返回1

```
int16_t MC_GetImposedDirectionMotor1(void)
{
    return MCI_GetImposedMotorDirection( pMCI[M1] );
}
```

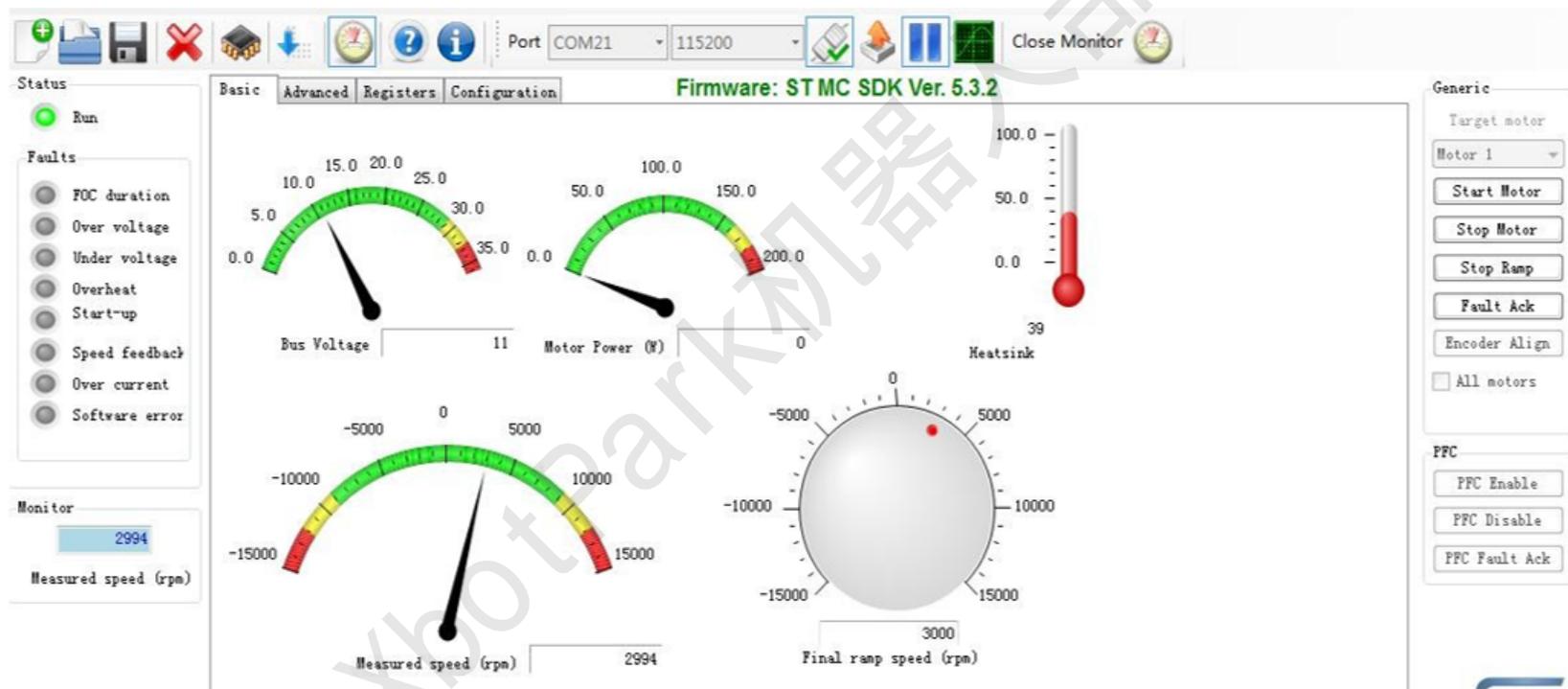
# 打开MotorControl Workbench观察相关参数和电机运行曲线 点Monitor按钮观察电机相关参数



点击**connect**按钮先进行连接，如连接不上按下板子上黑色复位按钮或更换串口端口号先进行测试。连接后如下图所示（可以看到**SDK 5.3**版本）



点**Basic**界面，可以看到板子的一些参数，也可以对板子进行启动、停止、清除故障等操作



点**Advanced**界面，可以看到**PID**一些参数以及速度或转矩模式，也可以对板子进行启动、停止、清除故障及模式切换等操作

The screenshot displays the STM32 Advanced Configuration and Debug Monitor (ACD) interface. At the top, it shows the port configuration (COM21, 115200) and the firmware version (STM32 SDK Ver. 5.3.2). The interface is divided into several sections:

- Configuration and debug:** Control mode is set to Speed. Power Board Status shows BUS Voltage at 11 Volt and Heatsink temp. at 39 °C. DAC Settings are configured for Ch1 (Ia) and Ch2 (Ib).
- Current controller:** Set current reference in speed mode. Torque ref (Iq) is 986 and Flux ref (Id) is 0.
- Measured currents:** Torque (Iq) is -799 and Flux (Id) is 0.
- Iq PID Gains:** Kp is 964 and Ki is 198.
- Id PID Gains:** Kp is 964 and Ki is 198.
- Speed controller:** Speed ramp with Target speed at 3000 rpm and Duration at 1000 millise. An Exec ramp button is present.
- PID Gains:** Kp is 3955 and Ki is 1271.
- Sensor-less Observer+PLL:** Observer is -1380, PLL Kp is 1862, and PLL Ki is 44.
- Sensor-less Observer+Cordic:** Observer is 0.
- Flux weakening tuning:** Kp and Ki are both 0.
- BUS Voltage allowed:** Ref is 0 % and Meas is 0 %.

点Registers界面，可以看到PID、寄存器等相关参数以及最后读取参数时间等

Basic Advanced Registers Configuration Firmware: ST MC SDK Ver. 5.3.2

	Id	Name	Unit	Value	Min	Max	Period	Type	Mode	Enable	Last read
▶	0x00	Target motor		0	0	255	0	U8	RW	<input checked="" type="checkbox"/>	never
	0x01	Flags		0	0	4294967...	200	U32	R	<input checked="" type="checkbox"/>	2019-01-06 14:1...
	0x02	Status		4	0	255	200	U8	R	<input checked="" type="checkbox"/>	2019-01-06 14:1...
	0x03	Control mode		1	0	255	500	U8	RW	<input checked="" type="checkbox"/>	2019-01-06 14:0...
	0x04	Speed reference	RPM	-3000	-15000	15000	200	S32	R	<input checked="" type="checkbox"/>	2019-01-06 14:1...
	0x05	Speed Kp		3955	0	65535	0	U16	RW	<input checked="" type="checkbox"/>	2019-01-06 14:0...
	0x06	Speed Ki		1271	0	65535	0	U16	RW	<input checked="" type="checkbox"/>	2019-01-06 14:0...
	0x07	Speed Kd		0	0	65535	0	U16	RW	<input checked="" type="checkbox"/>	never
	0x08	Torque reference (Iq)		986	-32768	32767	0	S16	RW	<input checked="" type="checkbox"/>	2019-01-06 14:0...
	0x09	Torque Kp		964	0	65535	0	U16	RW	<input checked="" type="checkbox"/>	2019-01-06 14:0...
	0x0A	Torque Ki		196	0	65535	0	U16	RW	<input checked="" type="checkbox"/>	2019-01-06 14:0...
	0x0B	Torque Kd		0	0	65535	0	U16	RW	<input checked="" type="checkbox"/>	never
	0x0C	Flux reference (Id)		0	-32768	32767	0	S16	RW	<input checked="" type="checkbox"/>	never
	0x0D	Flux Kp		964	0	65535	0	U16	RW	<input checked="" type="checkbox"/>	2019-01-06 14:0...
	0x0E	Flux Ki		196	0	65535	0	U16	RW	<input checked="" type="checkbox"/>	2019-01-06 14:0...
	0x0F	Flux Kd		0	0	65535	0	U16	RW	<input checked="" type="checkbox"/>	never
	0x10	Observer C1		-1380	-32768	32767	0	S16	RW	<input checked="" type="checkbox"/>	2019-01-06 14:0...
	0x11	Observer C2		3473	-32768	32767	0	S16	RW	<input checked="" type="checkbox"/>	2019-01-06 14:0...

# 点Configuration界面，可以对一些参数进行重新配置及重新装载

Port COM21 115200 Close Monitor

Firmware: STMC SDK Ver. 5.3.2

Board Configuration

Import from

Motor available: Single Mot

Motor 1 or any motor

- Sensor-less (Obs+PLL)
- Sensor-less (Obs+Cordic)
- Sensor-less (OPI+Obs)
- Quadrature encoder
- Hall sensors
- Flux weakening
- DAC channels

Control mode: Speed

Min speed: -15000

Max speed: 15000

Max bus: 38.0

Motor 2 (if available)

- Sensor-less (Obs+PLL)
- Sensor-less (Obs+Cordic)
- Sensor-less (OPI+Obs)
- Quadrature encoder
- Hall sensors
- Flux weakening

Control mode: none

Min speed: -5000

Max speed: 5000

Max bus: 500.0

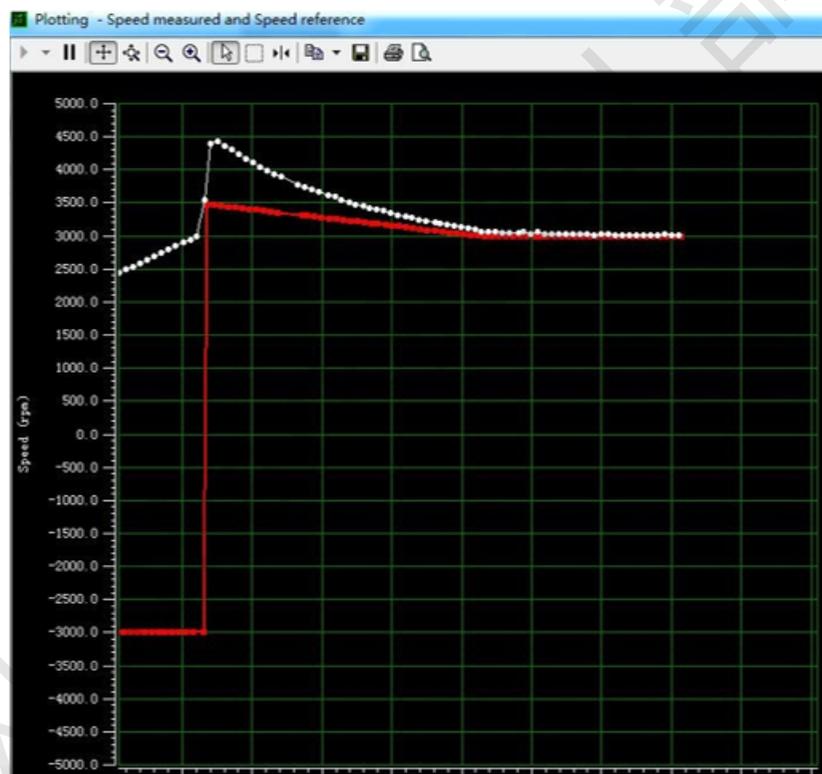
Revup Configuration

	Num	Final Speed (rpm)	Final Torque	Duration (ms)	Last read	Last write
▶	1	0	12032	1000	2019-01-06 14:00:49.601	never
	2	3330	12032	3333	2019-01-06 14:00:49.609	never
	3	3330	12032	0	2019-01-06 14:00:49.617	never
	4	3330	12032	0	2019-01-06 14:00:49.625	never
	5	3330	12032	0	2019-01-06 14:00:49.633	never

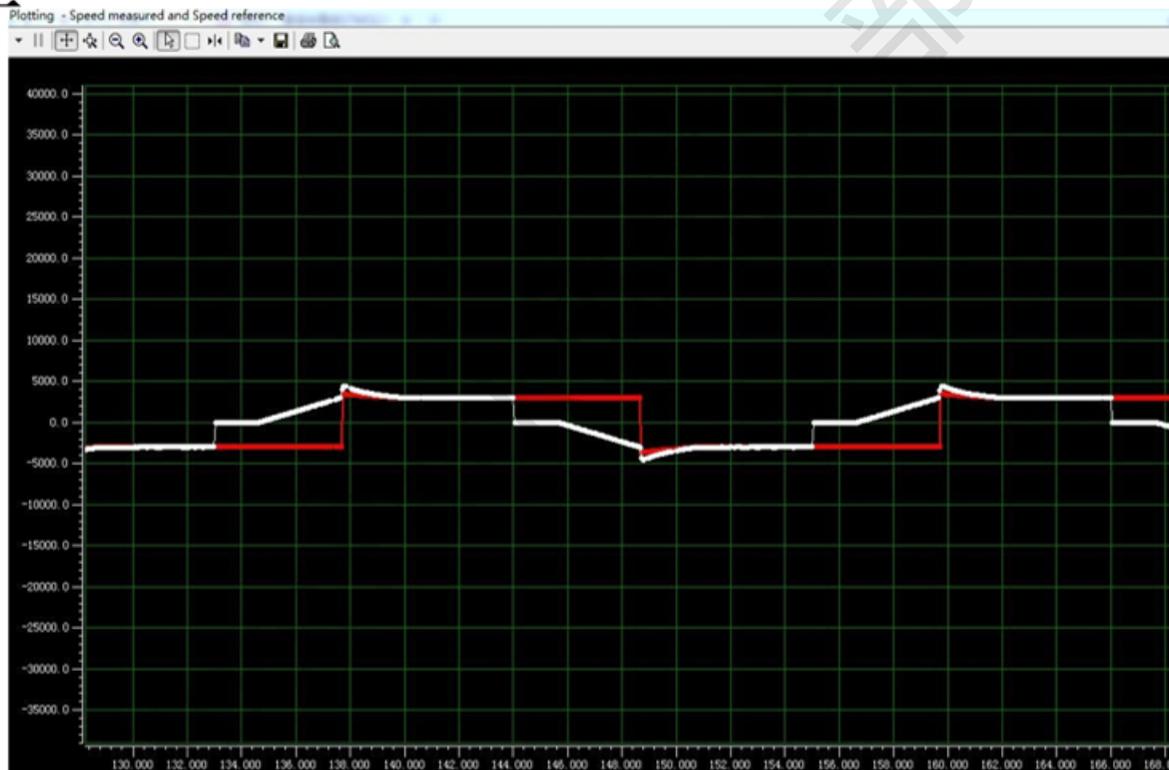
Reload

# 点Plotter按钮，观察电机的转速运行曲线

打开后如下图所示



可以通过放大、缩小、拖动等操作对电机的运行曲线看的更清晰些



# FOC控制电机调试技巧补充说明

## ◆ 关于电机参数测试Motor Profiler:

目前主要支持302及303相关的一些板子;

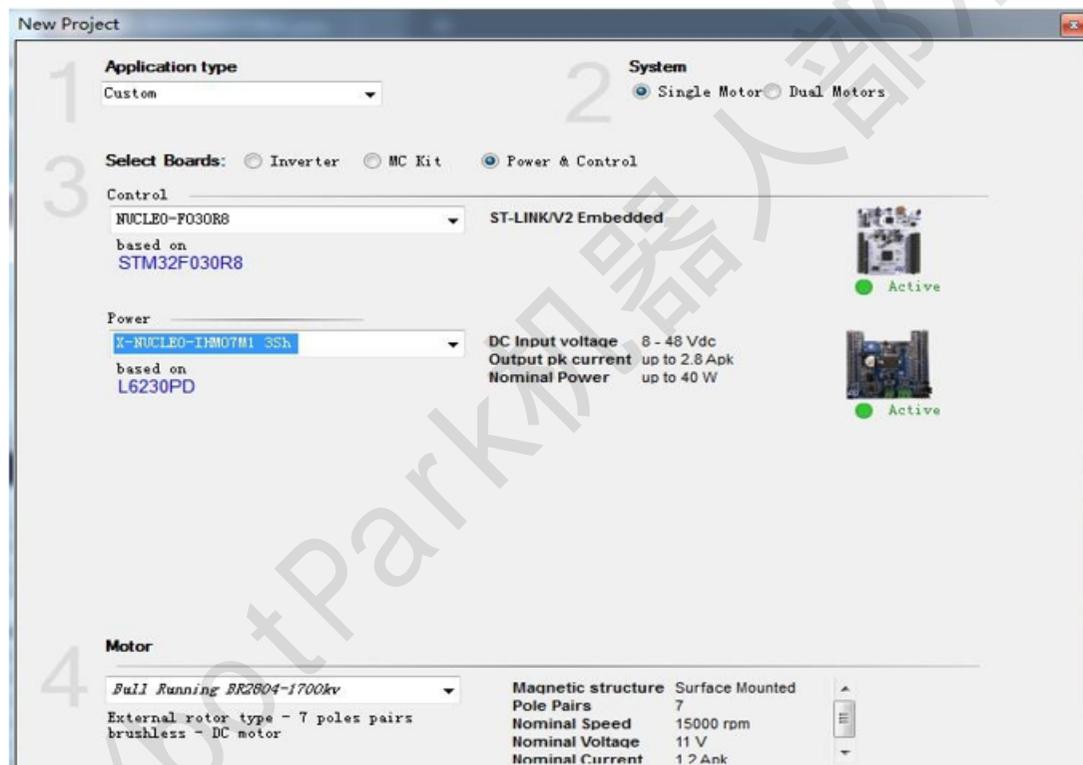
电机参数测试有一定的限制, 主要是测试1欧以内的电机绕组电阻, 电阻太大或太小都无法测试(比如有些航模电机就测不了);

对于无法测试的电机参数, 相关参数最好找电机长要; 或可以用LR测试仪以30K频率的对L和R进行测试, 其它参数再进行理论计算; 或用相近的电机参数, 自己调节电流环的PI参数;

◆关于电流环、速度环、转矩环等PI参数调节: PI参数调小点, 先调P, 再调I, 从小往大调选择合适的值;

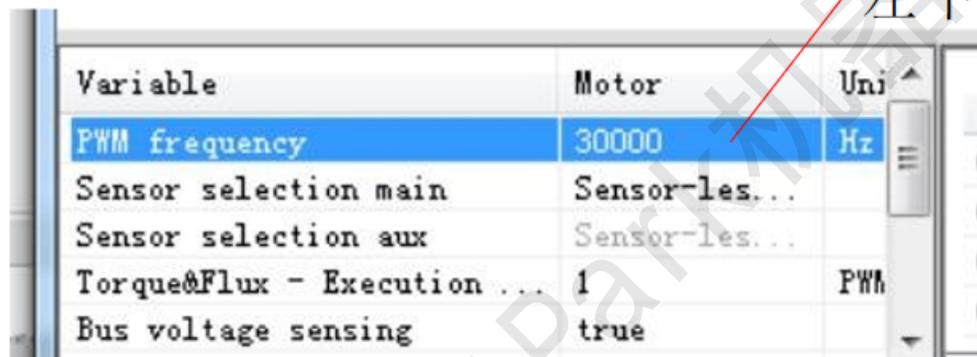
◆关于不支持的芯片或64脚以下的芯片: 可以选择相近的芯片, 通过软件生成代码, 自己修改头文件进行移植。

# STM32F030R8生成工程注意事项



# STM32F030R8生成工程注意事项

新建工程后双击  
左下角的30000



Variable	Motor	Uni
PWM frequency	30000	Hz
Sensor selection main	Sensor-les...	
Sensor selection aux	Sensor-les...	
Torque@Flux - Execution ...	1	PWM
Bus voltage sensing	true	

# STM32F030R8生成工程注意事项

修改为16000（具体数值可后续调试）

Drive Management - Drive Settings

**PWM generation and current reading**

PWM frequency: 30000 Hz

High sides PWM idle state: Turn-off

Low side signals and dead-time

SW inserted dead-time: 800 ns

Low sides PWM idle state: Turn-off

**Default settings**

Control mode: Speed control

Target speed: 3000 rpm

Target stator current flux: 0.00 A

Target stator current torque: 0.00 A

**Speed regulator**

Execution rate: 1.0 ms

3955 / 64 P

1271 / 16384 I

Manual editing

Please, set the friction and inertia values in the Motor

**Torque and flux regulators**

Execution rate: 1 PWM periods

Cut-off frequency: 6000 rad/s

**Torque**

1051 / 16384 P

214 / 16384 I

**Flux**

1051 / 16384 P

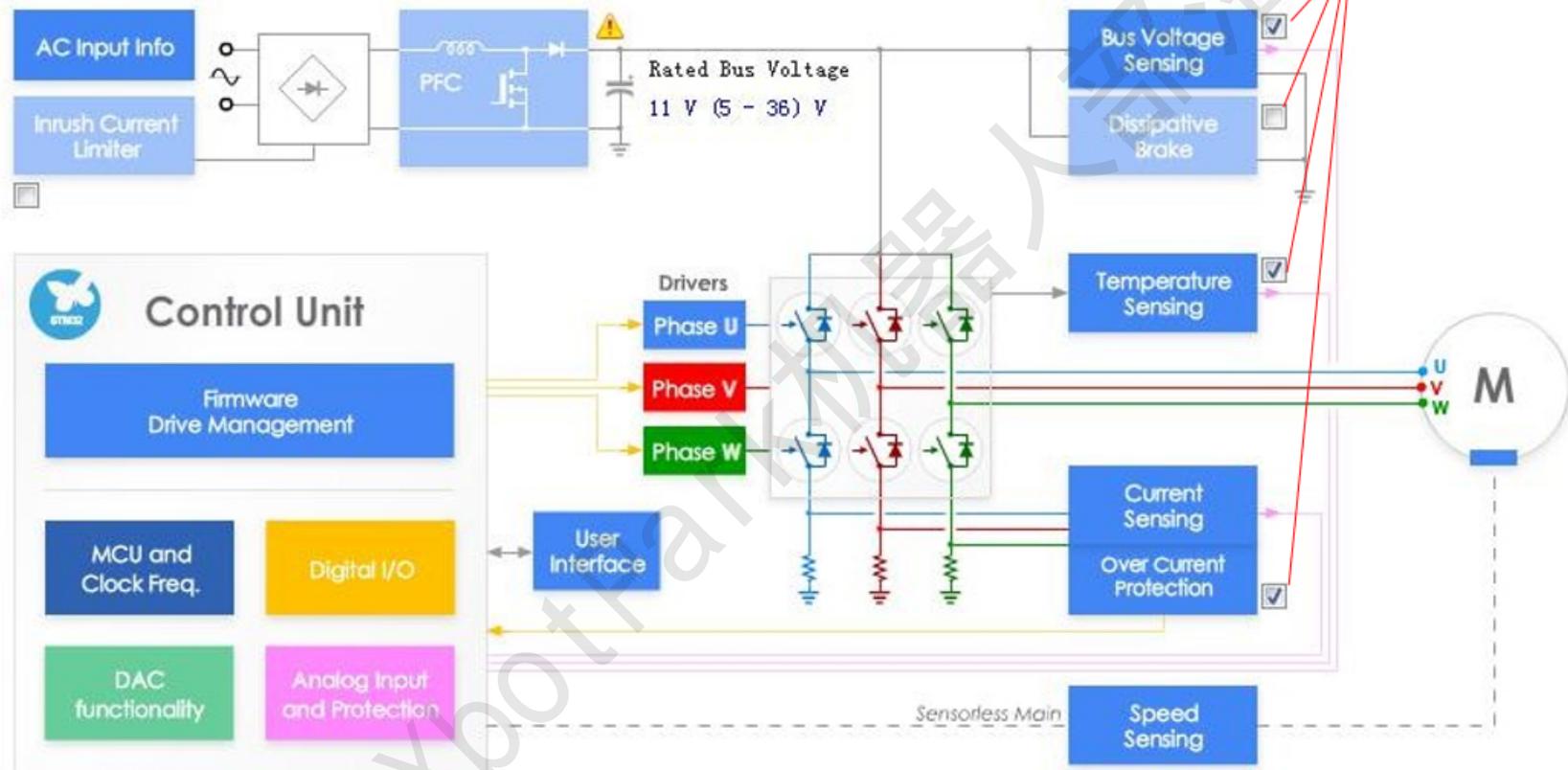
214 / 16384 I

Manual editing enabled

Done

# STM32F030R8生成工程注意事项

可根据需求勾选



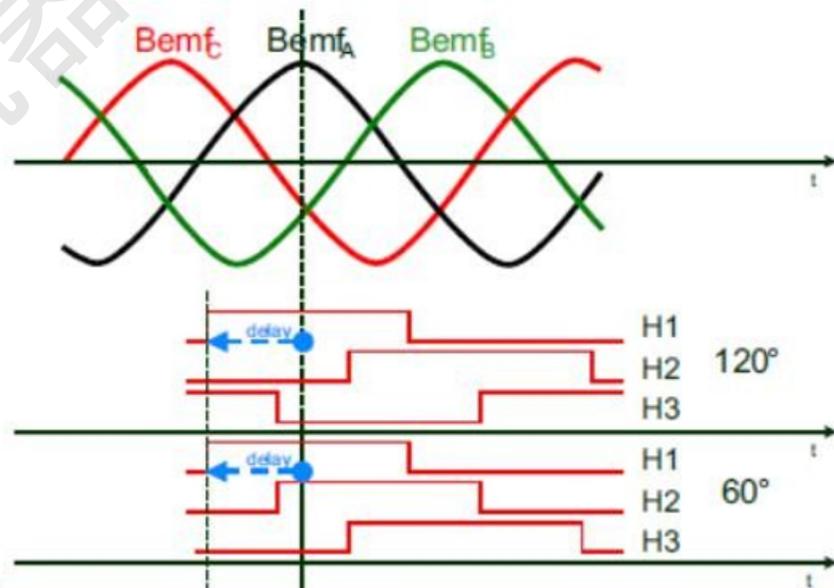
## 电机控制同步电角度测试说明

- 在使用 ST FOC电机库时，当使用Hall信号作为位置信号时，需要输入同步电角度数据，这个数据根据当前使用电机的特性进行输入，会在每次Hall信号变化时同步电角度，如果角度偏差较大时会影响控制效果，可能带来效率或者电机的震荡，初始测试还是有必要的。

- ST FOC电机库电角度约定：

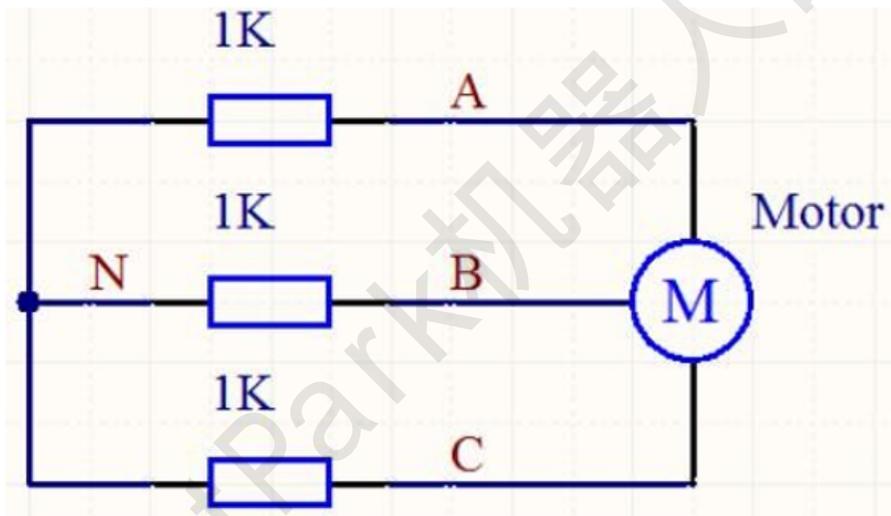
默认电机A相反电动势最高点作为电角度的0度；

电机Hall A的上升沿到电机A相反电动势最高点的延迟角度为同步电角度；



## 测试准备

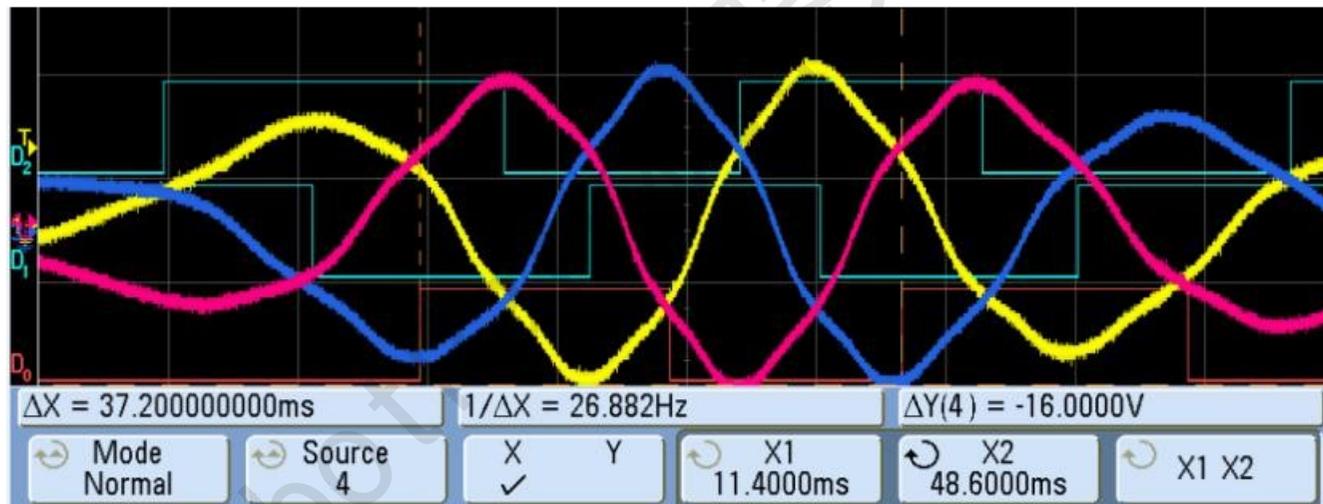
- 如果电机没有虚拟中点接出，需要连接三个相同阻值电阻到电机的三相接线 上，电阻另外一端连接到一起作为虚拟中点；



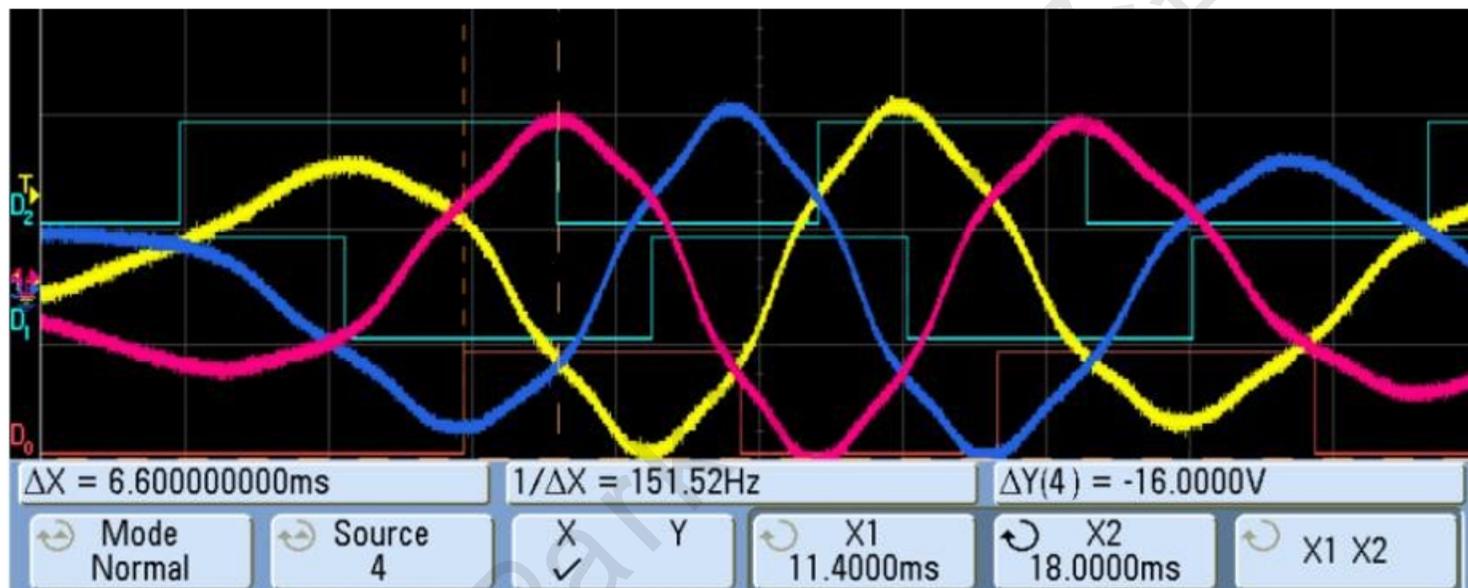
- 将 Hall信号接入5V电, 并且在H1上接入上拉电阻；
- 接入示波器, 转动电机, 测试反向电动势信号以及Hall信号；

# 波形测试及计算结果

- 下面是举例说明电机测试波形  
测试一个电周期的时间，这个周期对应360度；



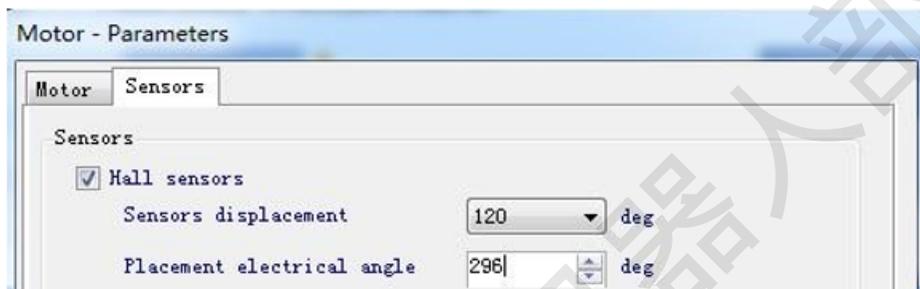
- 测试电机 A 相反向电动势最高点到 H1 的时间



- 上图中粉色为电机 A 相反向电动势，红色数字端口 D0 为 H1 信号；
- 该电机同步电角度：
$$\theta = \frac{37.2 - 6.6}{37.2} * 360^\circ = 296^\circ$$

## 同步角度添加到代码

如果使用 *Workbench* 的话，添加到电机参数中，如下图所示：



如果直接写入程序中，则将数据写入 *pmsm\_motor\_parameters.h* 文件参数中

```
/** Hall sensors */
#define HALL_SENSORS_PLACEMENT DEGREES_120 /*!<Define here the
mechanical position of the sensors
withreference to an electrical cycle.
It can be either DEGREES_120 or
DEGREES_60 */

#define HALL_PHASE_SHIFT 300 /*!< Define here in degrees
the electrical phase shift between
the low to high transition of
signal H1 and the maximum of
the Bemf induced on phase A */
```

# 软件调试经验分享

将测试的环境复杂性降到更低

- 关闭外部的控制器的功能
- 当第一次拿板子/算法测试
- 当问题出现

逐步从底层往外围进行功能测试



# PI参数调整方法

## 常用的调整方法:

- 1. 保持  $K_i = 0$  慢慢增加  $K_p$ , 直到系统达到最小静差

注意: 比例系数过大会使超调量增大, 振荡次数增加, 调节时间加长, 动态性能变坏, 甚至会使闭环系统不稳定。

2. 缓慢增加  $K_i$ , 例如:  $K_i = 0.1, 0.3, 0.5$  直到静差消除

注意:  $K_i$  越大变化速度越快, 积分作用越强。

若系统超调量太大, 经过多次振荡才能稳定或者根本不稳定, 应减小  $K_p$  和  $K_i$  系数。

# 电机调试技巧：1/2

- 首先，需要再次确认ST workbench中所有设置的参数是否和实际的硬件参数一致：如电机的相关参数，驱动部分的参数，单片机IO设置等。
  - 如果有其中任意一个参数设置错误，可能导致电机永远也无法正确启动。
  - 如果有需要，可以让电机运行在开环模式，来测量Tnoise和Trise相关参数。
- 如果启动后立即出现硬件过流保护，可能由以下原因导致：
  - 选择了错误的电流采样方式
  - 选择了错误的电流采样参数：如取样电阻值，放大倍数，ICS增益，Tnoise，Trise等。
  - 电流环的调节带宽过高：3电阻采样建议为2000rad/s，单电阻采样建议为1000rad/s
  - 由于布线受到干扰而导致误触发硬件过流保护，需要检查硬件设计。
- 如果出现电机只动一下，但是没有加速动作：
  - 这种问题一般是因为开环电流不够大导致无法拖起转子加速，有时出现开环启动完成，但报启动失败故障，这时：
    - 需要减低加速率，或提高开环启动电流
    - 如果以上方法可以解决，但是不能保证100%有效，请尝试增加定位功能。

## 电机调试技巧：2/2

- 如果转子可以转动并且有加速动作，但是还是会停止并且报“速度反馈失败”错误，可能由以下原因导致：
  - 启动成功的限制条件过于宽松导致过早切入闭环。
  - 如下的方法可以解决这样的问题：
    - 提高“连续成功启动输出测试”值，正常情况下请不要大于5。
    - 提高最小启动输出速度。
- 如果采用以上方法导致开环的最终速度过高，或没有解决问题，可以尝试以下方法：
  - 减少观测器的增益G2,它可以降低扰动对速度反馈的影响。
    - 通常G2应该按照/2,/4,/6,/8方式来减少。
  - 放宽观测器的收敛条件，这样使观测器更容易收敛：
    - 使用新的电机库，可以设置速度变化波动为80% (PLL) ,或400% (Cordic)。
    - 这种情况下需要增加反向电动势幅度与估算速度一致性的检查。
  - 更改速度/扭矩的爬升率：根据实际负载和转子的惯性等情况，让加速度更加柔和，防止突然加速导致对反向电动势估算的扰动。

# 电机启动波形



CH2: 反向电动势  
(DAC)

CH1: 电角度  
(DAC)

CH4: 相电流  
(电流探头)

# 电机运行时波形



CH2: 反向电动势  
(DAC)

CH1: 电角度  
(DAC)

CH4: 相电流U  
(电流探头)

## FOC控制电机调试技巧补充说明

- 关于电机示波器及仿真调试说明 对于高压电机驱动板：

电源最好使用隔离变压器或带隔离电源调试；

示波器的三芯电源线电源插头的，中间接地线最好不要接，以防烧坏示波器；最好使用带隔离的仿真器，隔离**USB**或隔离调试接口，以防烧坏电脑；

调试的时候电机运转时，程序最好不要设置断点。

# 动手试验

XbotPark机器人部落

