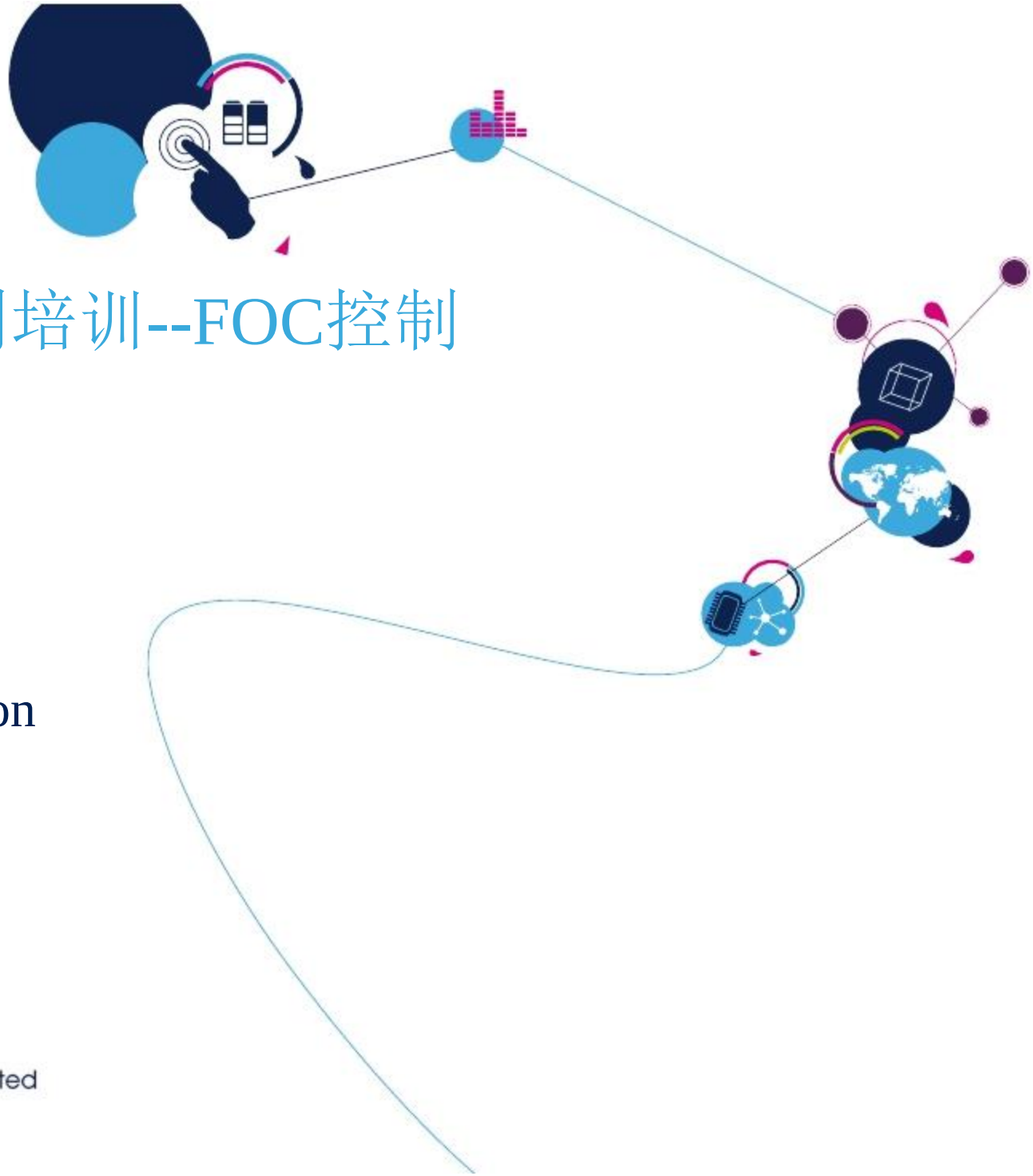


ST电机控制培训--FOC控制

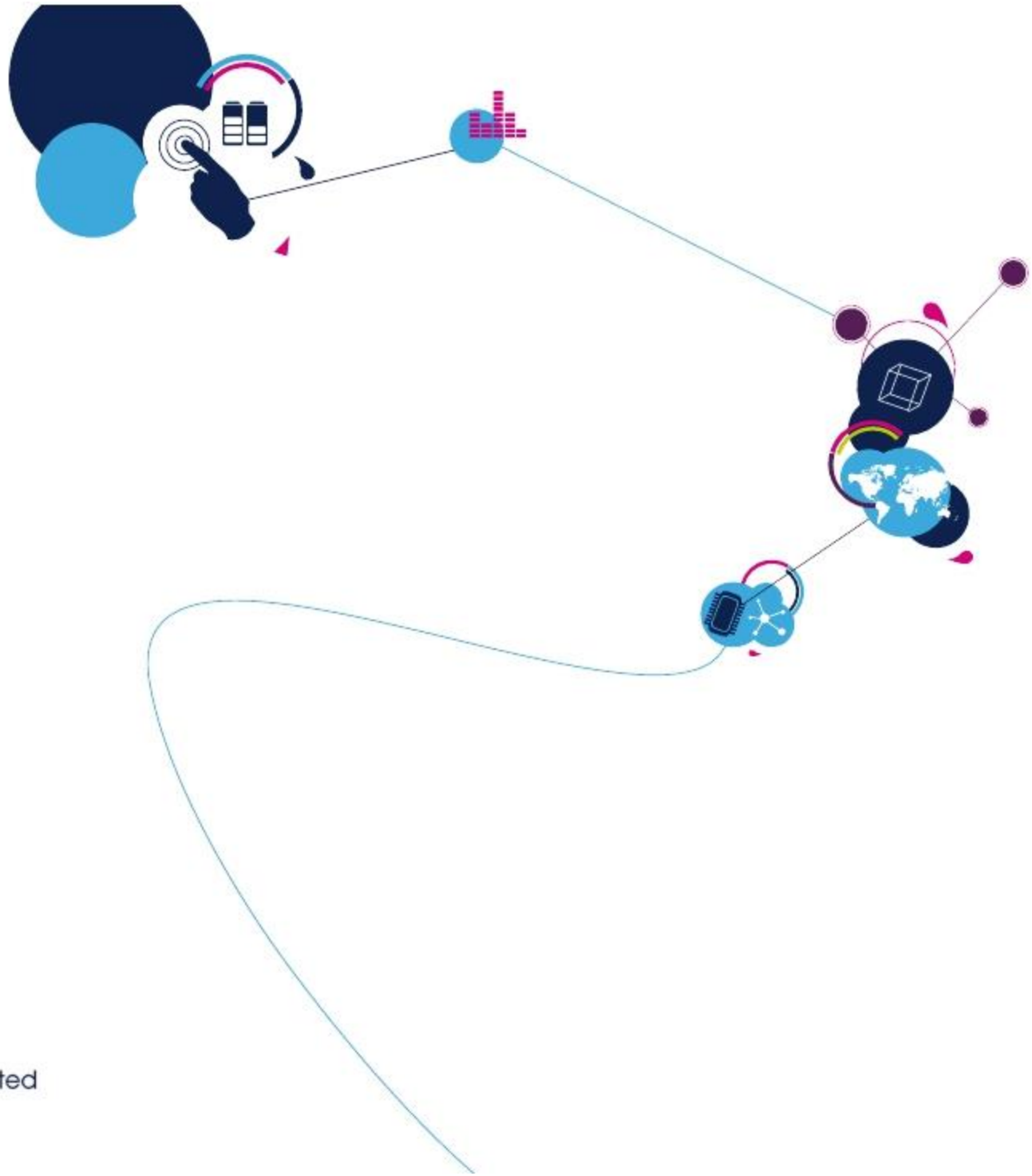
2017.08

MCU Application

life.augmented

Agenda

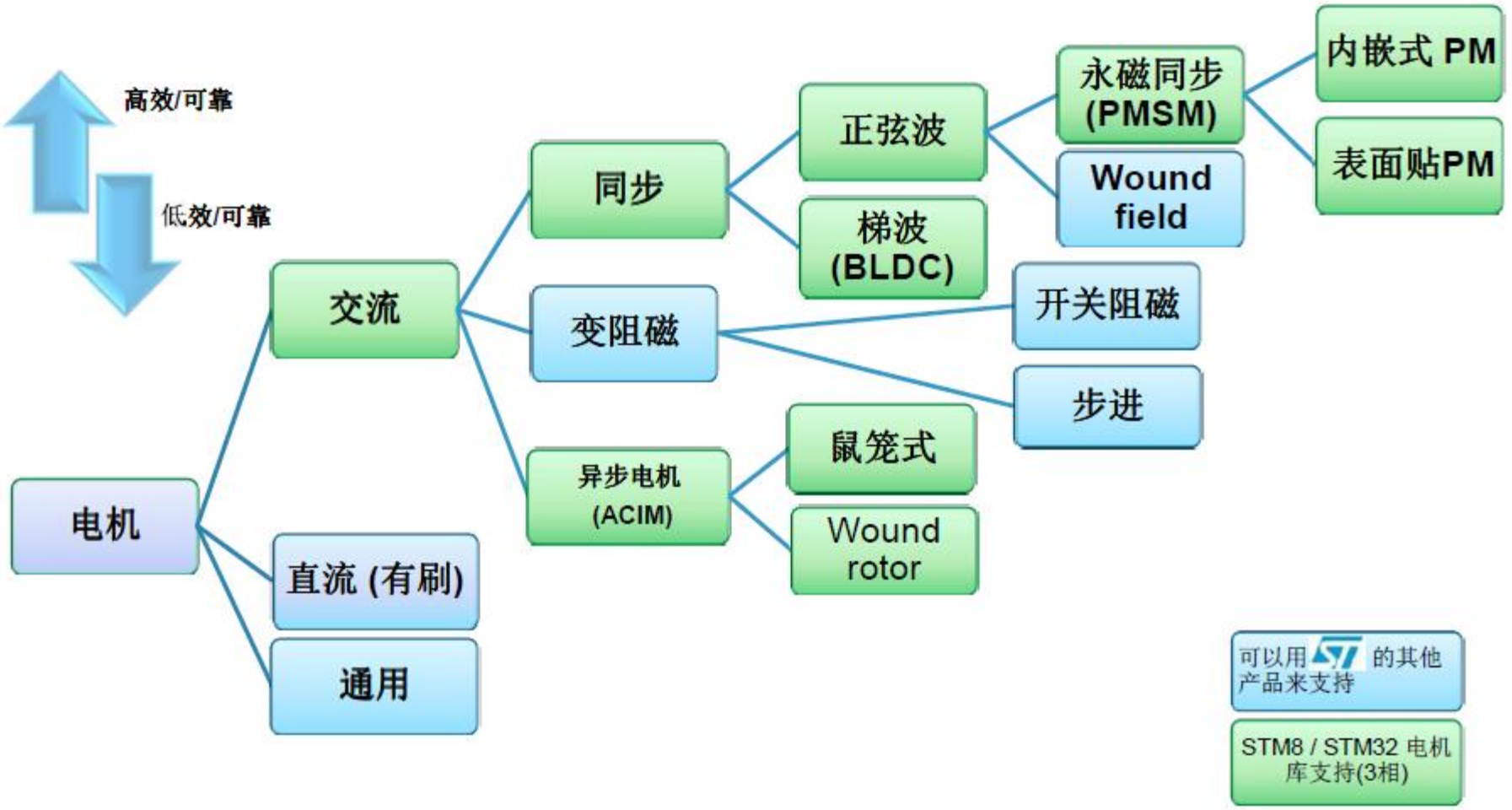
- 电机简介
- 评估工具：IDE，GUI，Demo 板
- STM32 PMSM FOC SDK V4.3概述
  - 试验一：评估工具使用
- FOC控制基础理论
- API使用示例
- 基于电机库开发项目
  - 试验二：API使用



电机简介

life.augmented

电机分类



life.augmented

直流有刷电机

➢ 基本原理：

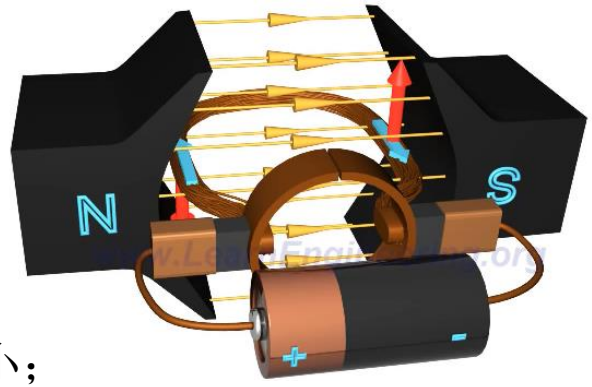
- 通电导体在磁场中受到作用力
- 力的方向符合左手定则
- 力的大小为F = BIL*sinθ

➢ 有刷直流电机构造特点：

- 一般情况下转子是绕组，定子为永磁体
- 电机内部有电刷 - 换向器

➢ 速度控制方式：

- 通直流电，通过控制直流电压大小来控制速度，
- 电压固定时，通过控制PWM波的占空比控制平均电压大小；

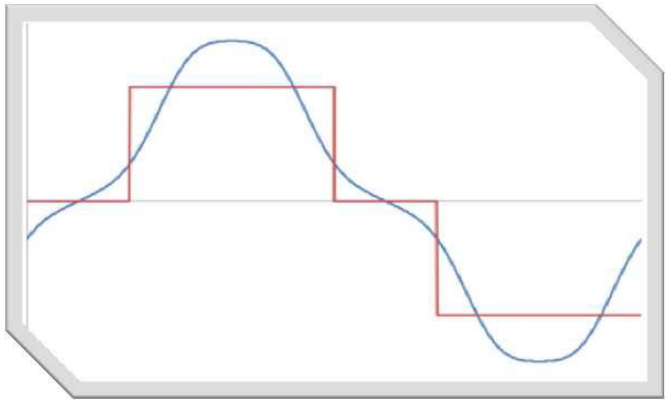


图片来源：http://www.learnengineering.org/

直流无刷电机-BLDC

➢ 基本原理

通电线圈产生磁场，该磁场与永磁体磁场相互作用产生磁力矩，使得转子转动；

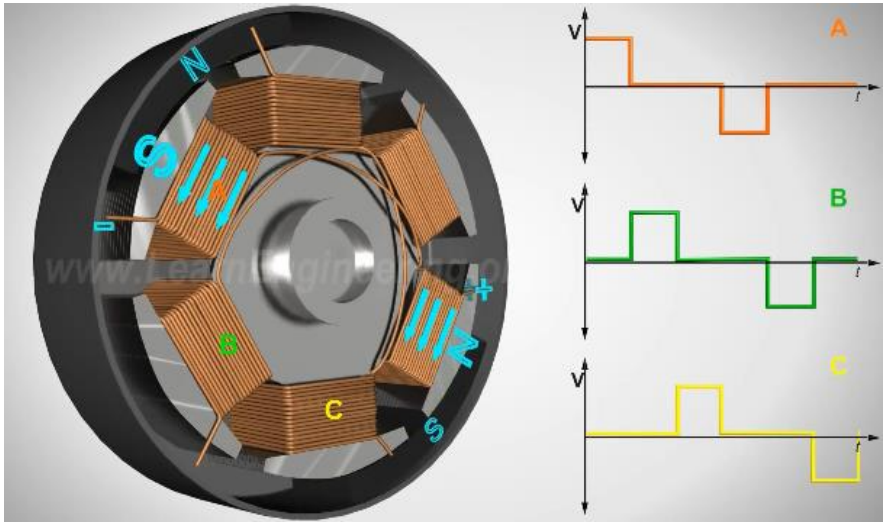


➢ BLDC构造特点：

- 转子是永磁体，定子为绕组
- 永磁体的磁化及其在转子上的分布是经过处理的
- 理想电流为方波电流，理想反电动势为梯形波

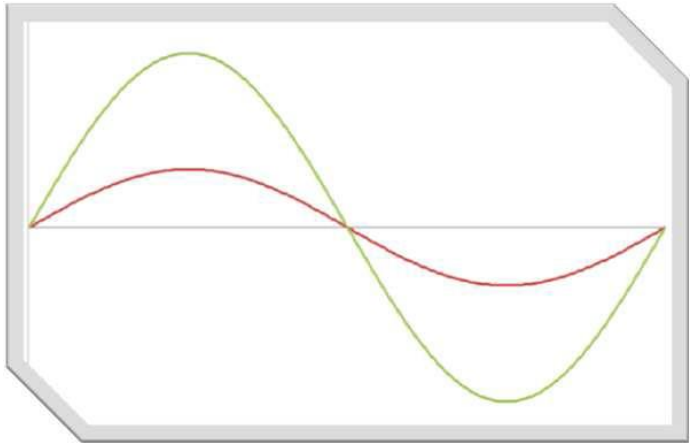
➢ 控制方式：

- 需要控制器产生旋转磁场
- 使用方波电流控制产生最佳效果



永磁同步电机-PMSM

- PMSM构造特点：
  - 转子是永磁体，定子为绕组，同BLDC
  - 理想电流为正弦波电流，理想反电动势为正弦波
- 控制方式：
  - 需要控制器产生矢量旋转磁场
  - 使用正弦波电流控制产生最佳效果



红色：电流　绿色：反电动势

BLDC vs PMSM

- 两者在某些时候可以统一看作是同步电机
- 对比这两种电机需要和控制方法放一起讨论
  - BLDC也可以使用矢量控制
  - PMSM也可以使用方波控制

控制方法	方波控制	矢量控制
控制途径	简单，位置换向	复杂，需要坐标变换
转矩脉动	波动大	波动小
运行噪声	较高	较低
气隙	大，参数稳定	小，参数易饱和
低速性能	差	优
高转速性能	较高	较低

评估工具：评估板，IDE，GUI

life.augmented

马达控制评估板

life.augmented

各种评估板供选择

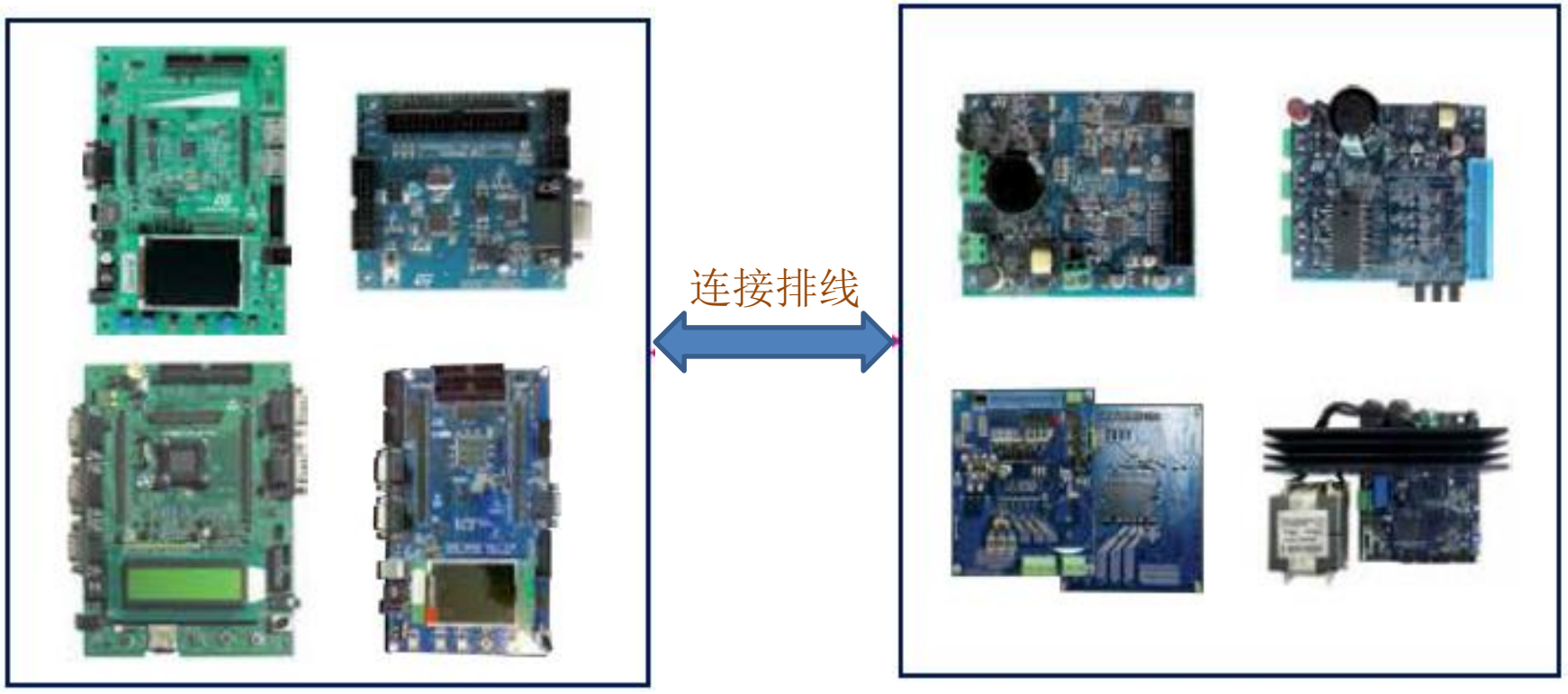
有小电机

Flexible MC Platform	Complete MC drives	STM32 Nucleo + X-NUCLEO	MC KIT
Control stages  MC Connector Power stages			

MC Software Development Kit (SDK)
(FW library, GUI, collateral materials)

EVB+功率板

- 可以灵活搭配MCU，功率板，符合目标开发目的



MCU EVB板

功率板

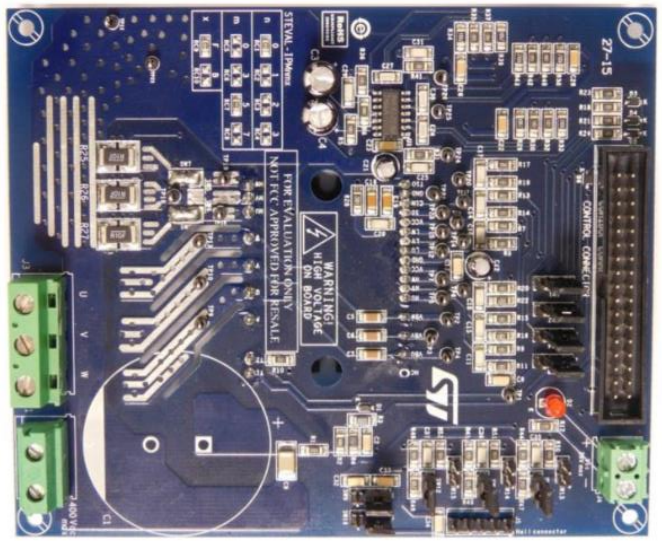
MCU EVB

- MCU EVB 拥有马达控制接口（34-Pin）

Part Number	Description	ST Link on-board	Type
STM32072B-EVAL	Evaluation board with STM32F072VB MCU	Yes	Single drive
STM3210E-EVAL	Evaluation board for STM32 F1 series - with STM32F103 MCU	No	Single drive
STM3220G-EVAL	Evaluation board for STM32 F2 series - with STM32F207IG MCU	Yes	Single drive
STM32303E-EVAL	Evaluation board for STM32F303xx microcontrollers	Yes	Single/Dual drive
STM32446E-EVAL	Evaluation board for STM32F407 line - with STM32F407IG MCU	Yes	Single drive
STEVAL-IHM039V1	Dual motor drive control stage based on the STM32F415ZG microcontroller	No	Single/Dual drive

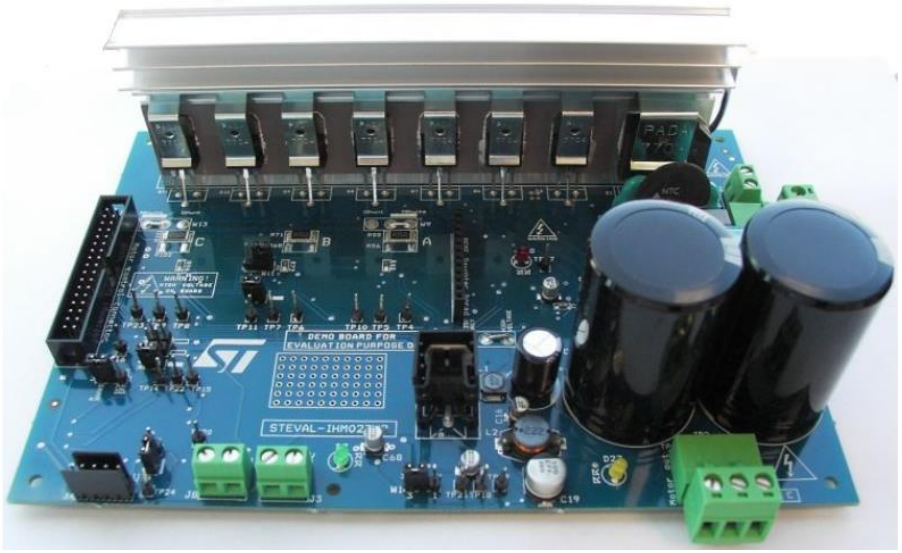
功率板-DC供电

Reference / bundle	Voltage	Power	Motor type / control type *	ST Parts	Application focus
STEVAL-IPM05F	125 – 400 V_{DC}	Up to 700 W (Up to 8A)	PMSM/BLDC FOC/6-step 3-shunt	• 1 x **STGIF5CH60TS-L** • 1x **TSV994**	Power board: water pumps, fans, dish washers and more
STEVAL-IPM07F	125 – 400 V_{DC}	Up to 800 W (Up to 10A)	PMSM/BLDC FOC/6-step Single/3-shunt	• 1 x **STGIF7CH60TS-L** • 1x **TSV994**	Power board: water pumps, fans and more
STEVAL-IPM10F	125 – 400 V_{DC}	Up to 1 kW (Up to 15A)	PMSM/BLDC FOC/6-step	• 1 x **STGIF10CH60TS-L** • 1x **TSV994**	Power board: pumps, compressors, washing machines and more
STEVAL-IPM10B	125 – 400 V_{DC}	Up to 1.2 kW (Up to 15A)	PMSM/BLDC FOC/6-step single/3-shunt	• 1 x **STGIB10CH60TS-L** • 1x **TSV994**	Power board: pumps, compressors, air conditioning and more
STEVAL-IPM15B	125 – 400 V_{DC}	Up to 1.5kW (Up to 20A)	PMSM/BLDC FOC/6-step single/3-shunt	• 1 x **STGIB15CH60TS-L** • 1x **TSV994**	Power board: pumps, compressors, fans, dish washers and more



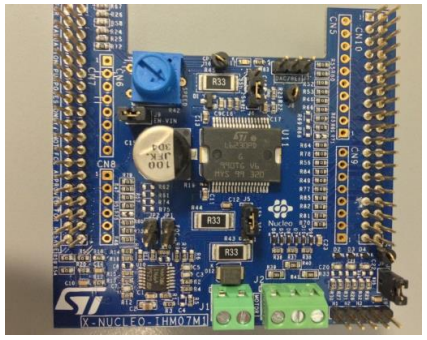
功率板-AC供电

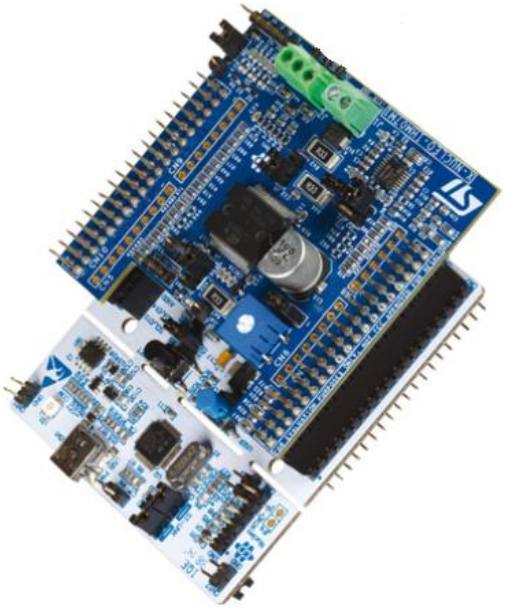
Reference / bundle	Voltage	Power	Motor type / control type *	ST parts	Application focus
STEVAL-IHM021V2	120/230 V_{AC} nominal (60/50 Hz)	Up to 100 W	PMSM/BLDC FOC/6-step 3-shunt	• 3x **L6390** • 1x **Viper12** • 6x **STD5N52U**	Power board: water pumps, fans, dish washers, washing machines
STEVAL-IHM023V3	90 – 285 V_{AC} 125 – 400 V_{DC}	Up to 1 kW	PMSM/BLDC FOC/6-step Single/3-shunt	• 3x **L6390** • 1x **Viper16** • 7x **STGP10H60DF**	Power board: pumps, compressors, washing machines and more
STEVAL-IHM028V2	90 – 285 V_{AC} 125 – 400 V_{DC}	Up to 2 kW	PMSM/BLDC FOC/6-step Single/3-shunt	•1x **STGIPS20C60** • 1x **VIPer26LD** • 1x **STGW35NB60SD**	Power board: pumps, compressors, air conditioning and more
STEVAL-IHM032V1	230 V_{AC} nominal 86 to 260 V_{AC}	Up to 150 W	PMSM/BLDC FOC/6-step Single/3-shunt	• 2x **L6392D** • 1x **L6391D** • 1x **Viper12** • 6 x **STGD3HF60HD**	Power board: pumps, compressors, fans, dish washers and more
STEVAL-IHM035V2	120/230 V_{AC} nominal	Up to 100 W	PMSM/BLDC FOC/6-step single-shunt	• 1x **STGIPN3H60** • 1x **VIPer16L**	Power board: pumps, compressors, fans, dish washers and more
STEVAL-IHM045V1	30 – 270 V_{AC} 40 – 400 V_{DC}	Up to 100 W	PMSM FOC Single/3-shunt	• 1x **STGIPN3H60A** • 1x **VIPer06L** • 1x **TSV994**	Power board: pumps, compressors, fans, dish washers and more



Nucleo功率板

Reference / bundle	Voltage	Power / current	Motor type / control type *	ST Parts	Application focus
X-NUCLEO-IHM07M1	Up to 48V	Up to 2.5A	PMSM/BLDC FOC/6-step Single/3-shunt	• 1x **L6230** • 1x **BAT30KFILM** • 1x**TSV994IPT**	Sewing machines, pumps, drones,
X-NUCLEO-IHM08M1	10 – 48Vdc	Up to 15A	PMSM/BLDC FOC/6-step Single/3-shunt	• 6x**STL220N6F7** • 3x**L6398** • 1x**TSV994IPT**	Drones, e-bikes, drills, pumps, etc.
X-NUCLEO-IHM09M1	-	-	Motor control connector adapter	• **Not silicon devices**	Allow connection of STM32 NUCLEO boards with any ST motor control power boards





电机控制整合板

Part Number	Description	ST Link on-board	Type
STEVAL-IHM034V2	Dual-motor control and PFC demonstration board featuring the STM32F103 and STGIPS20C60	No	Single/Dual drive
STEVAL-IHM036V1	Low-power motor control board featuring the SLLIMM™ STGIPN3H60 and MCU STM32F100C6T6B	No	Single drive
STEVAL-IHM038V1	BLDC ceiling fan controller based on STM32 and SLLIMM-nano	No	Single drive
STEVAL-IHM040V1	BLDC/PMSM driver demonstration board based on STM32 and the SLLIMM-nano	No	Single drive
STEVAL-IHM042V1	Compact, low-voltage dual-motor control board based on the STM32F303 and L6230	Yes	Single/Dual drive
STEVAL-IHM043V1	6-Step BLDC sensorless driver board based on the STM32F051 and L6234	No	Single drive

STEVAL-IHM034V2

STEVAL-IHM036V1

STEVAL-IHM042V1

STEVAL-IHM043V1

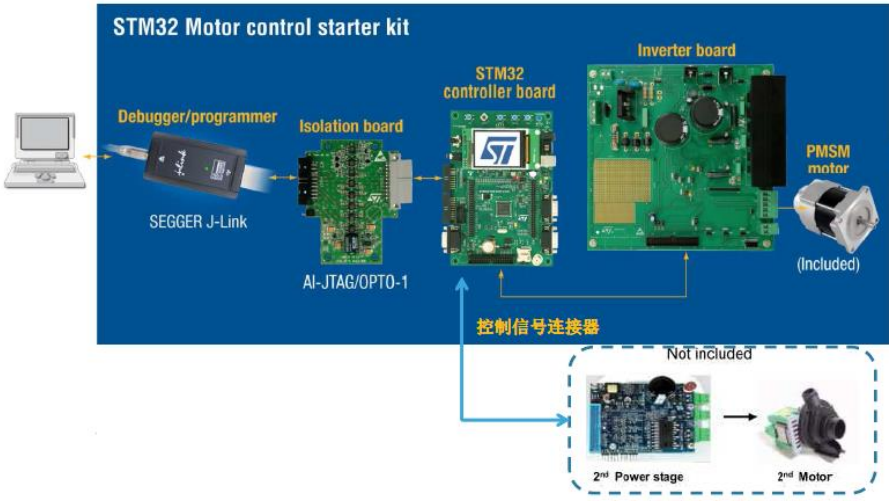
STEVAL-IHM038V1

STEVAL-IHM040V1

life.augmented

电机控制套件(带电机)

Part Number	Description	ST Link on-board	Type
P-NUCLEO-IHM001	STM32 Nucleo Pack FOC and 6-step control for Low voltage 3-ph motors	Yes (embedded)	Single drive
P-NUCLEO-IHM002	with DC Power supply		
STM32100B-MCKIT	Motor control starter kit for STM32F100 (128KB Flash) Value Line MCUs	Yes	Single drive
STM3210B-MCKIT	Motor control starter kit for STM32 (128KB flash) Performance and Access Line microcontrollers	Yes	Single drive



IDE

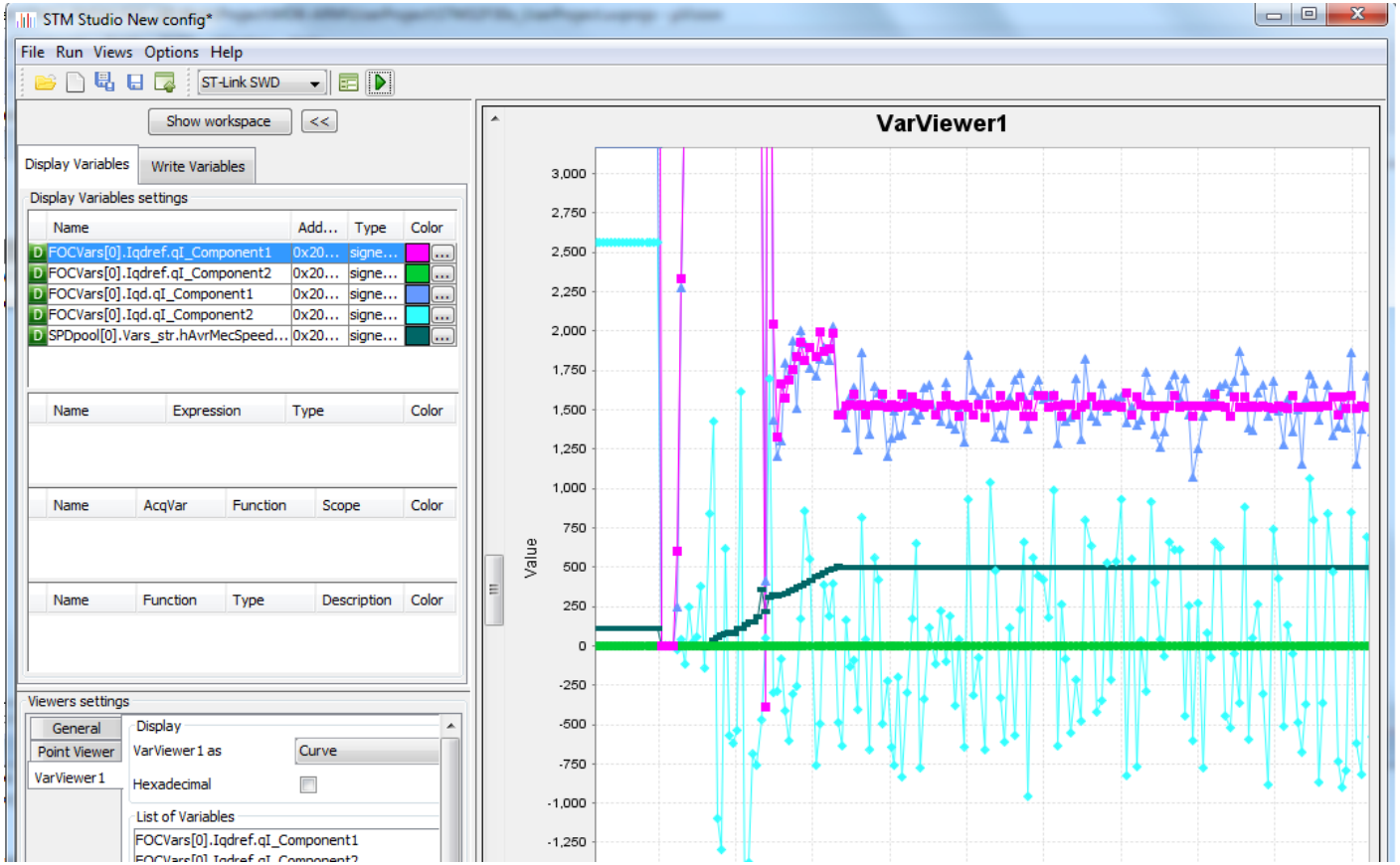
支持多种IDE工具

- *IAR* Embedded Workbench for ARM
  - http://www.iar.com/
- *Keil* Embedded Development Tools for ARM, Cortex-M **...**
  - http://www.keil.com/
- *SW4STM32* : free IDE for STM32on Windows, Linux and OS X
  - http://www.st.com/

注：如果使用的芯片是STM32F0或者STM32L0芯片，Keil有正版无代码限制版本

工具软件 – STM Studio

- 不增加硬件基础上可观测变量变化趋势
- 配合DAC使用效果更佳



电机控制库

- 本次培训所使用的Demo 板可使用的电机库如红色标注的部分

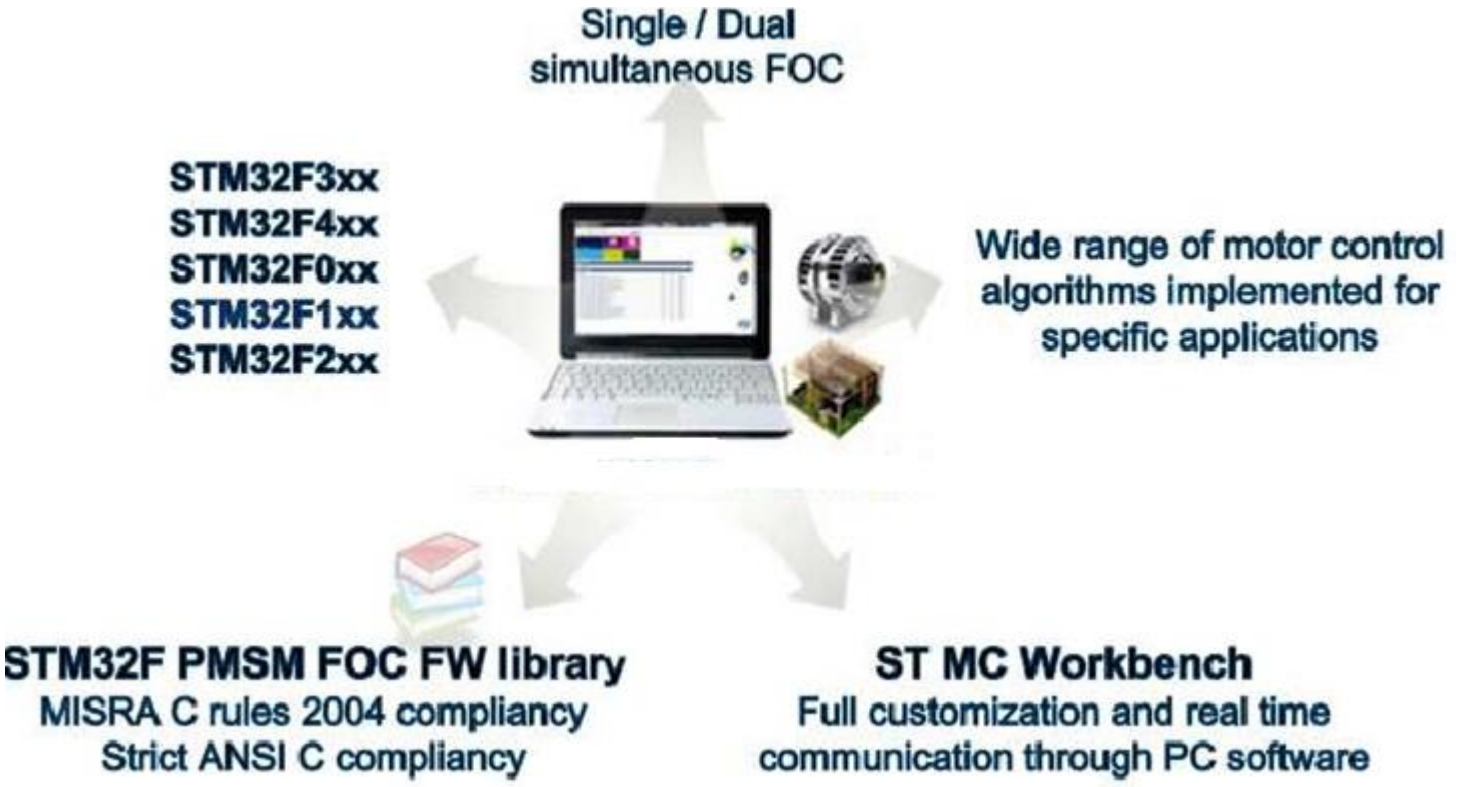
Part Number	General Description
STSW-STM32018	Driving bipolar stepper motors using a medium-density STM32F103xx microcontroller (AN2820)
STSW-STM32056	STM32F1xx motor control firmware for STSPIN L6474
STSW-STM32100	**STM32 PMSM FOC Software Development Kit - MC library (UM1052)**
STSW-STM8020	STM8S and STM8A BLDC and ACIM motor control firmware library V1.0 (UM0708)
X-CUBE-SPN1	Stepper motor driver software expansion for STM32Cube
X-CUBE-SPN11	Low voltage three-phase brushless DC motor driver software expansion for STM32Cube
X-CUBE-SPN12	Low voltage dual brush DC motor driver software expansion for STM32Cube
X-CUBE-SPN13	Low voltage brush DC motor driver software expansion for STM32Cube
X-CUBE-SPN2	Two axes stepper motor driver software expansion for STM32Cube
X-CUBE-SPN3	High power stepper motor driver software expansion for STM32Cube
X-CUBE-SPN4	Dual brush DC motor driver software expansion for STM32Cube
X-CUBE-SPN5	Bipolar stepper motor driver software expansion for STM32Cube
X-CUBE-SPN6	Low voltage stepper motor driver software expansion for STM32Cube
X-CUBE-SPN7	**3-phase brushless DC motor driver software expansion for STM32Cube**
X-CUBE-SPN8	Low-Voltage BLDC motor driver software expansion for STM32Cube

STM32 PMSM FOC SDK V4.3概述

life.augmented

STM32 PMSM FOC SDK V4.3

- SDK V4.3软件包包含：PMSM FOC 固件库和ST MC Workbench(GUI)，允许用户使用STM32进行单或双PMSM马达的FOC的驱动，其支持STM32F0xx, STM32F1xx, STM32F2xx, STM32F3xx及STM32F4xx



整合：软件库 + 产品(MCU、功率器件等) + 应用

高端应用，单/双马达控制

- FOC:高动态性能
- 高频注入法(HFI)无传感器算法
- 低CPU负荷，大部分时间用于应用软件上
- 同时控制2个马达

FOC MC FW lib

中/低端应用

要求：

- FOC矢量控制对马达进行高效控制
- 降低马达的噪音：正弦波电流
- 硬件成本优化：1-shunt电流采样，无传感器算法

低端应用

MC FW lib

- Scalar control
- 成本敏感应用

STM32F4xx
180MHZ,Cortex-M4

STM32F2xx
120MHZ,Cortex-M3

STM32F3xx
72MHZ,Cortex-M4

STM32F103
72MHZ,Cortex-M3

STM32F0xx
48MHZ,Cortex-M0

STM32F100
24MHZ,Cortex-M3

STM8S
24MHZ,ST core

Industrial motor drives

Fitness, wellness and healthcare

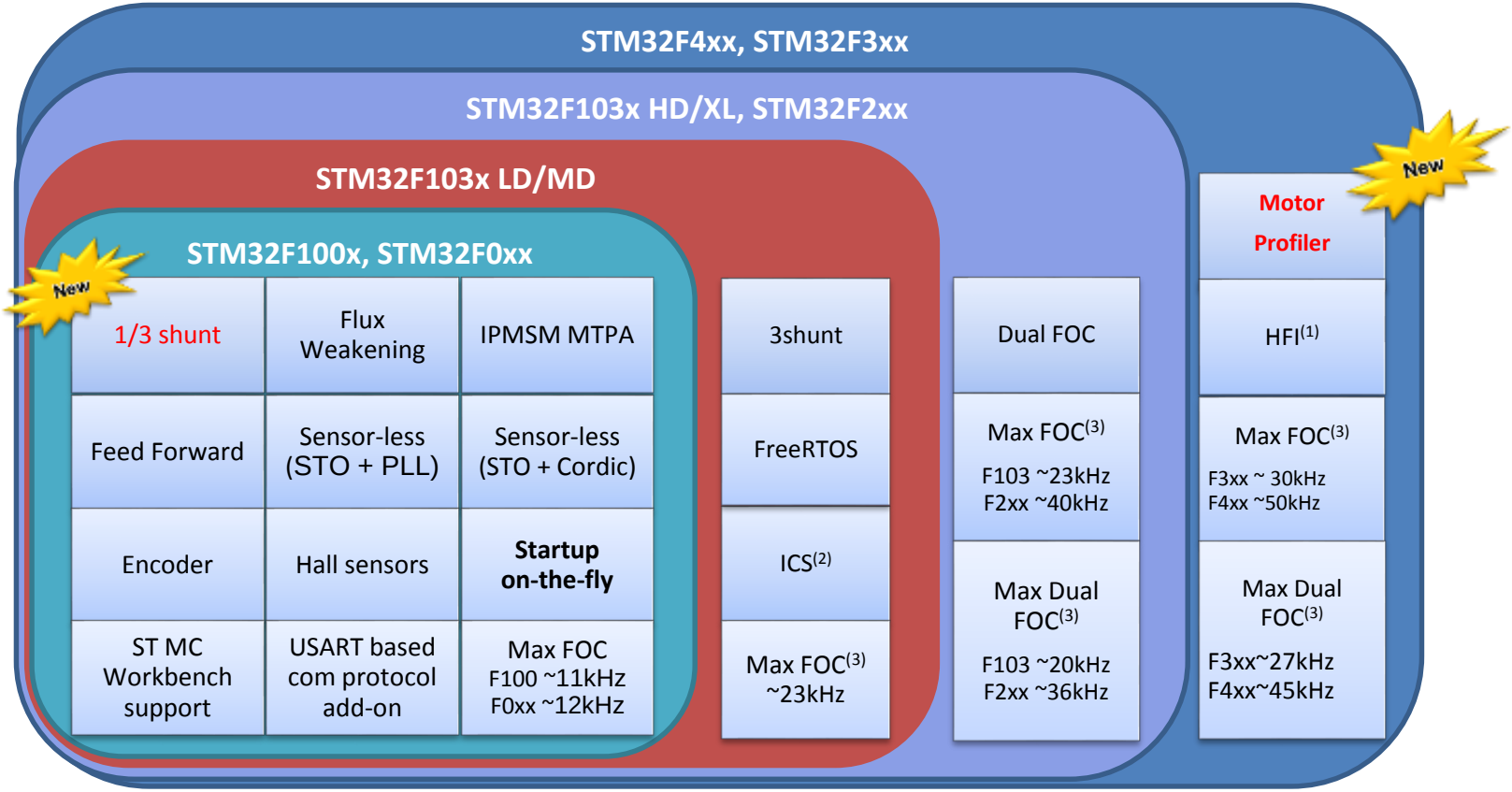
Home appliances

Games

Fan

算法集及MCU支持

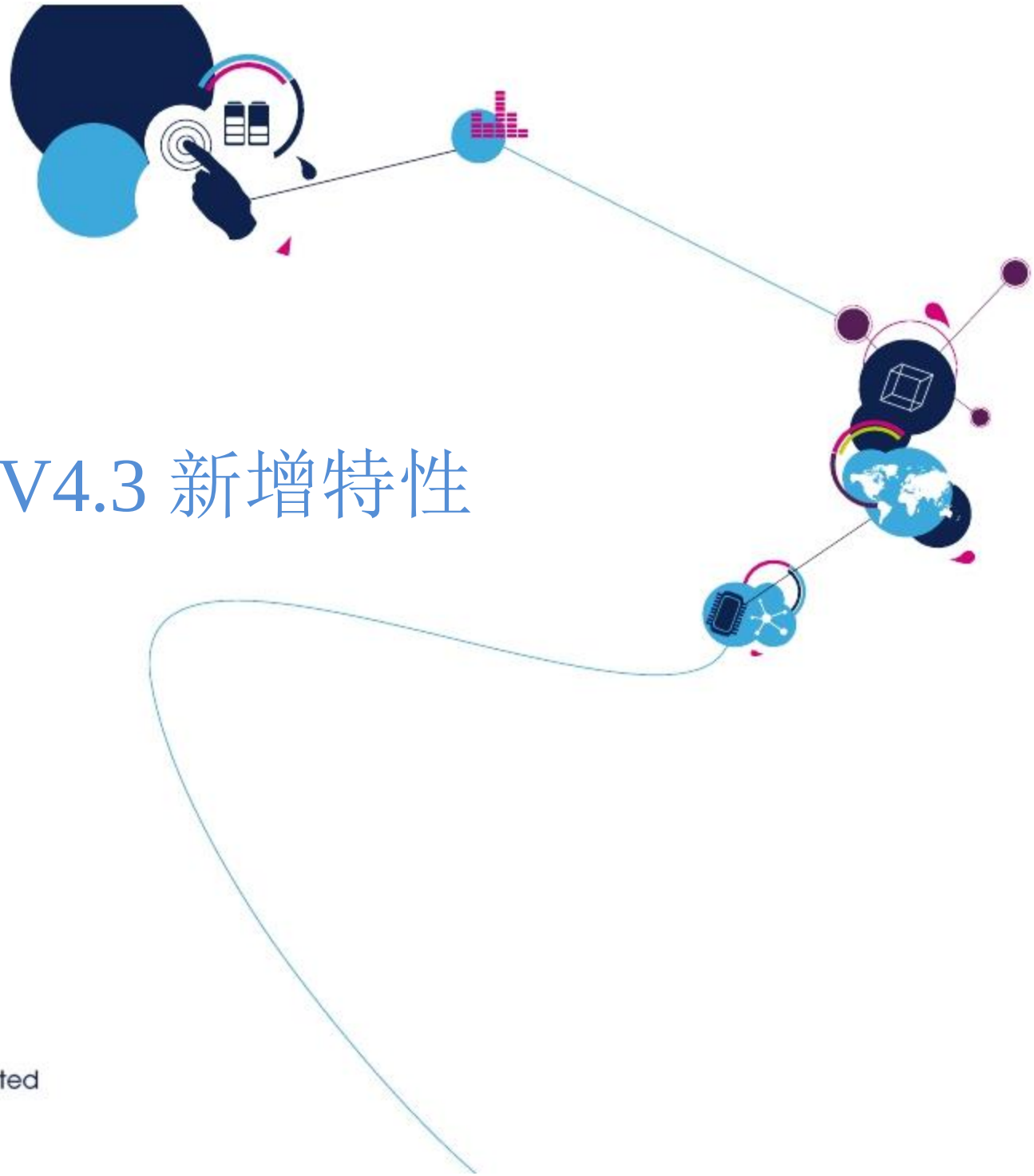
MC FW LIB v4.3



STM32 FOC 性能指标

STM32F	Single motor				Dual motor			
@20kHz PWM / 10kHz FOC								
STM32F	Configuration	Code size (Kb)	RAM usage (Kb)	CPU load (%)	Configuration	Code size (Kb	RAM usage (Kb)	CPU load (%)
STM32F100	1shunt/ sensorless	17.8	2.6	58.0	NA	NA	NA	NA
STM32F05x	**1shunt/ sensorless**	**16.7**	**2.7**	**45.2**	NA	NA	NA	NA
STM32F103x	3shunt/ sensorless	16.2	2.5	21.3	Motor 1&2 3 shunt/ sensorless	17.5	3.9	49.4
STM32F2x	3shunt/ sensorless	15.5	2.6	13.9	Motor 1&2 3 shunt/ sensorless	17.5	3.9	29.0
STM32F30x	3shunt/ sensorless	16.7	2.7	15.2 (*)	**Motor 1&2 3 shunt/ sensorless**	**18.8**	**4.0**	**33.3**
STM32F4x	3shunt/ sensorless	15.5	2.6	11.1	Motor 1&2 3 shunt/ sensorless	17.5	3.9	23.5

(*) +8% if with HFI

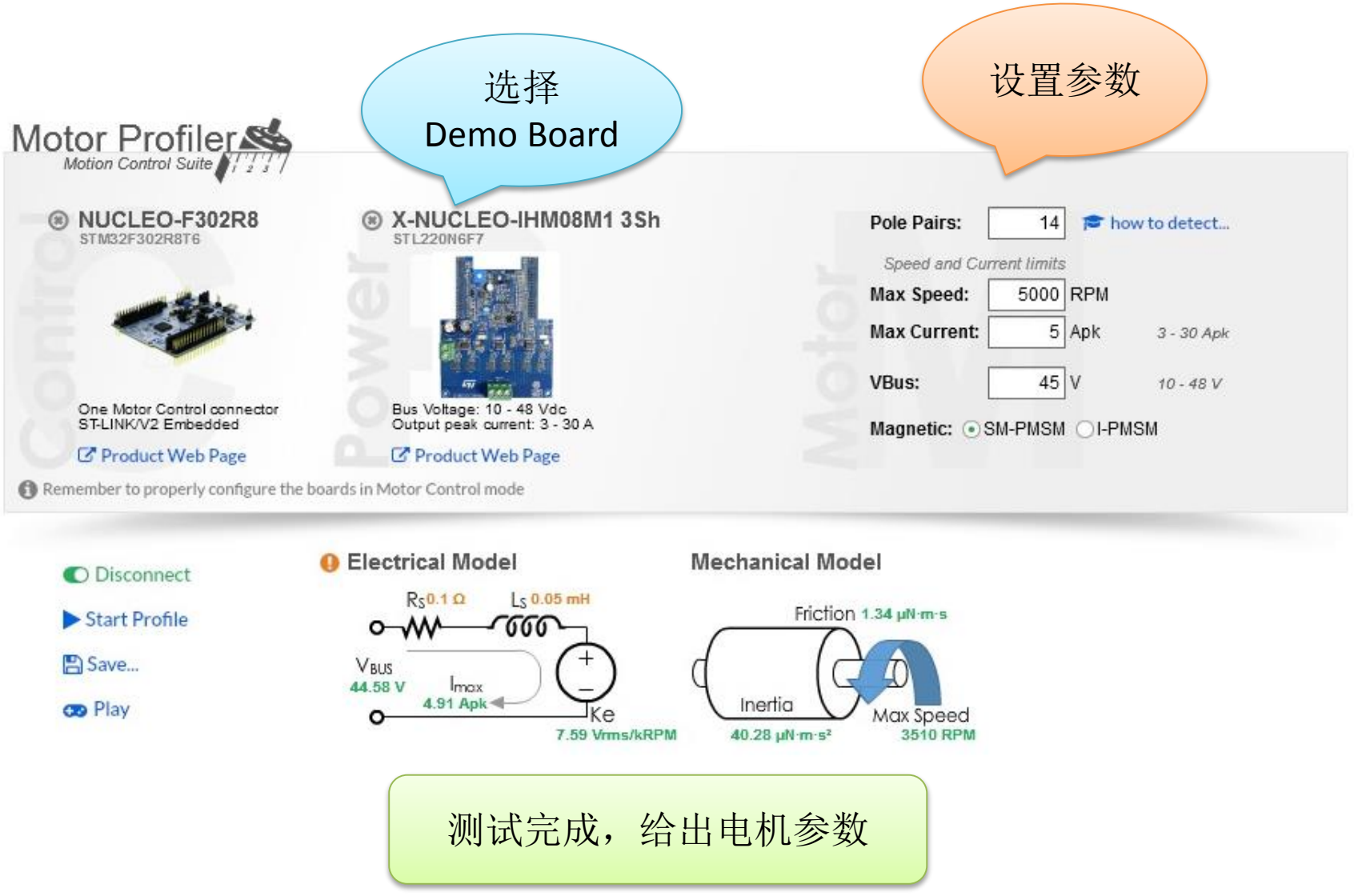


SDK FOC V4.3 新增特性

life.augmented

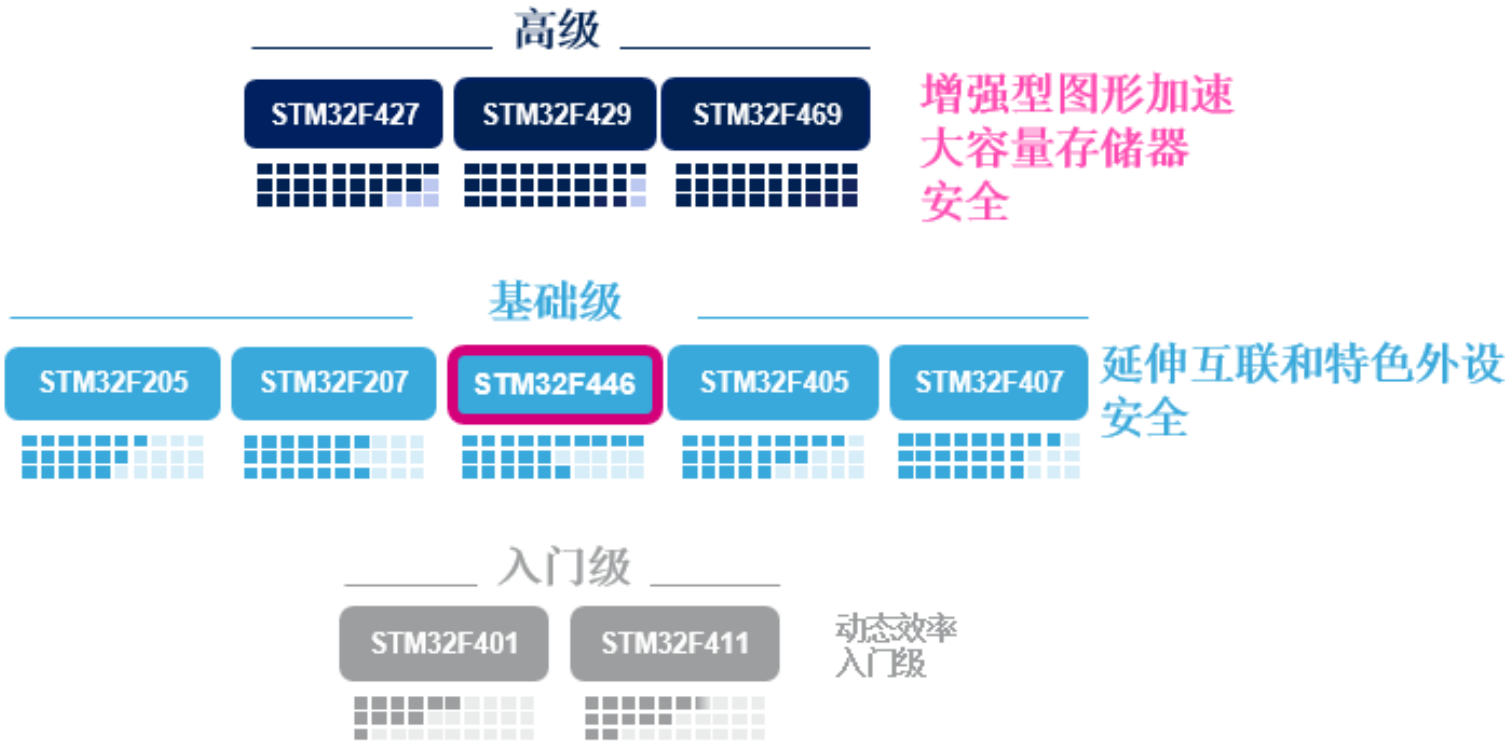
新增工具： STMotorProfiler

- 测试马达参数工具软件
- 需要使用ST Demo板



FOC V4.3库新增支持芯片

- 新增支持芯片STM32F446
  - ARM® 32-bit Cortex®-M4 180MHz
  - 浮点运算单元FPU + DSP指令集
  - 支持QSPI
  - 灵活存储器控制器（FMC)
  - 增强USB，支持Link Power Mode (LPM)



FOC V4.3库新增支持芯片

- 新增支持芯片STM32F07x
  - ARM® 32-bit Cortex®-M0 48MHz
  - 12个Timer
  - 有CAN Bus
  - DAC（两个通道输出）
  - 两个电压比较器（COMP）
  - HDMI消费电子控制器 (CEC)

life.augmented

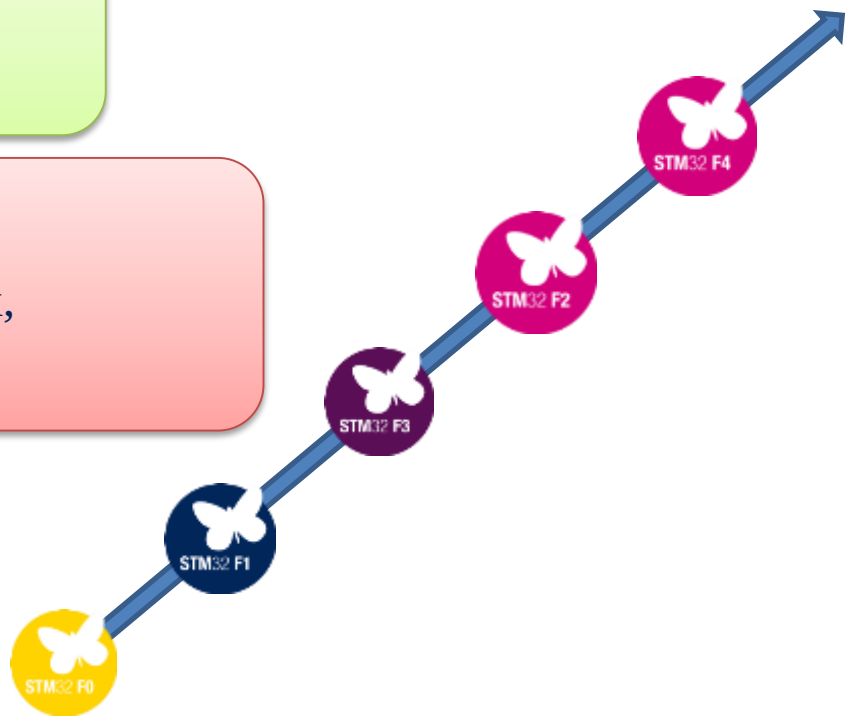
FOC4.3电机库支持的芯片型号

- STM32F030C6/C8/K6/R8
- STM32F051C6/C8/K6/K8/R6/R8
- STM32F072x

- STM32F100, STM32F103
- STM32F103

- STM32F302xB/C, STM32F303xB/C, STM32F302x6/x8
- STM32F301C6/C8/K6/K8/R6/R8

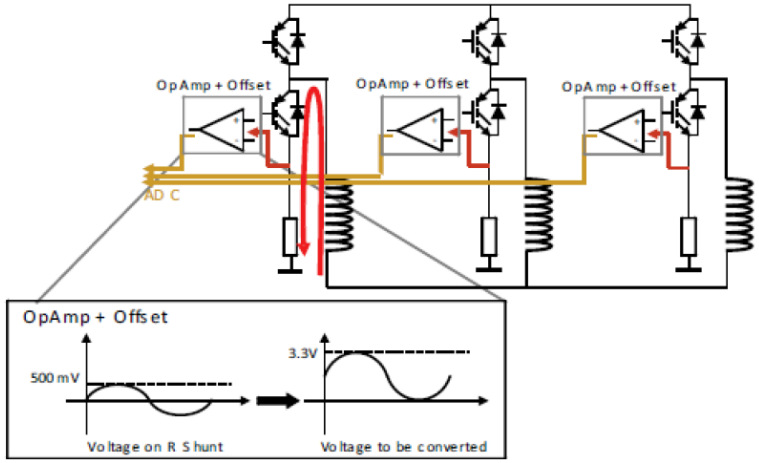
- STM32F2 series
- STM32F405xx, STM32F407xx, STM32F415xx, STM32F417xx, STM32F446xx

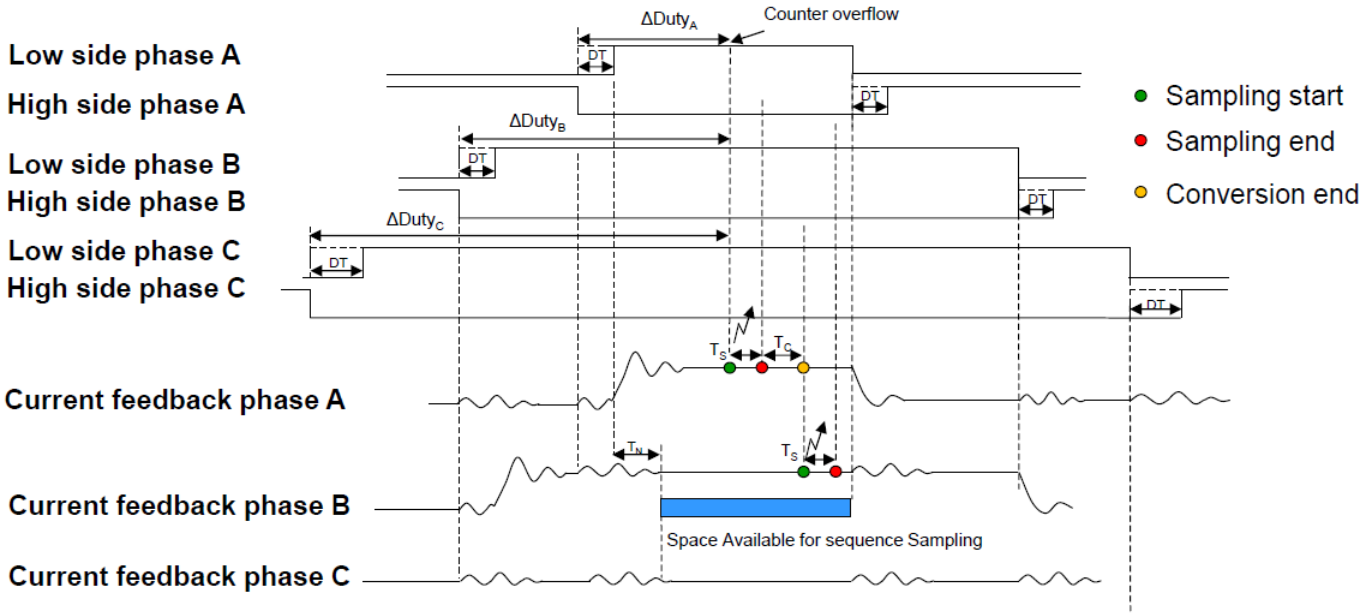


life.augmented

新增STM32F0的3-shunt算法支持

- 使用1个ADC模块采样两相电流数据
- 采样点需要符合2Ts + Tc < DT + max(TN,TR)
- 得益于STM32的ADC灵活的硬件触发机制
  - ADC触发点可以在波形上任意平移





支持AC6 Workspace

- 新增STM32F3xx，STM32F4xx 的Free RTOS工程
- 新增支持AC6 Workspace

6 for IAR EWARM with simple time base

STM32F0xx
STM32F10x
STM32F10x_Example
STM32F2xx
STM32F30x
STM32F4xx

5 for KEIL µVision with simple time base

STM32F0xx
STM32F10x
STM32F2xx
STM32F30x
STM32F4xx

4 for IAR EWARM with FreeRTOS

STM32F10x
STM32F2xx
STM32F3xx
STM32F4xx

5 for AC6

STM32F0xx
STM32F10x
STM32F2xx
STM32F30x
STM32F4xx

新增

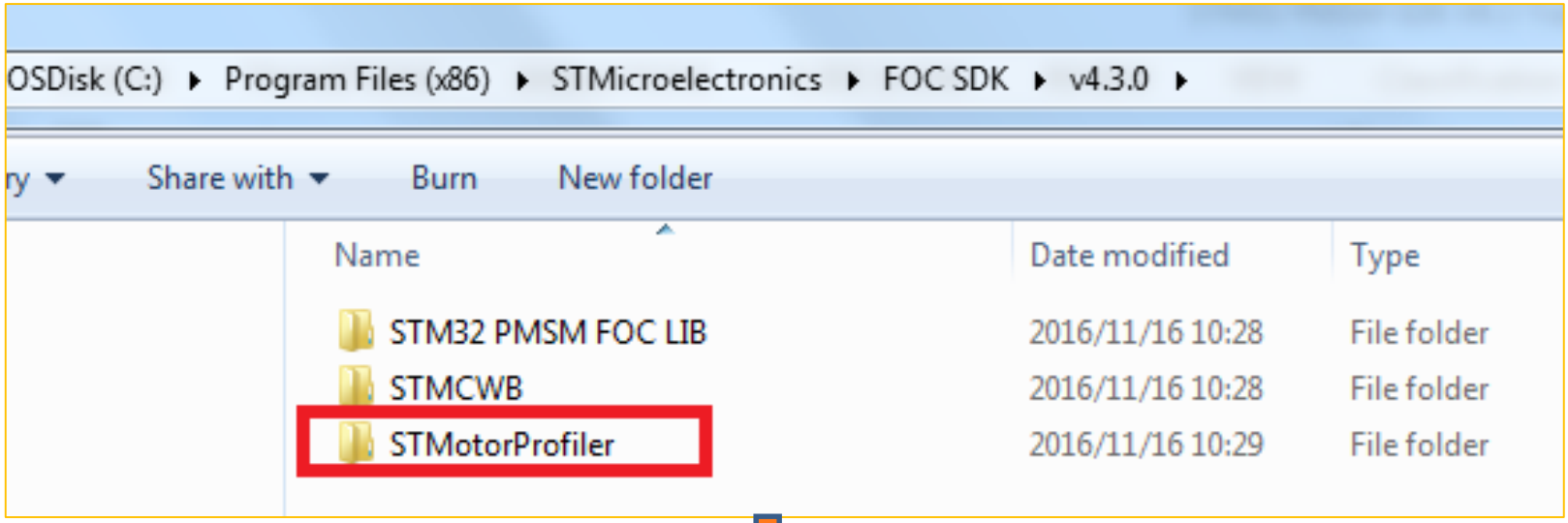
STMotorProfiler 实际操作

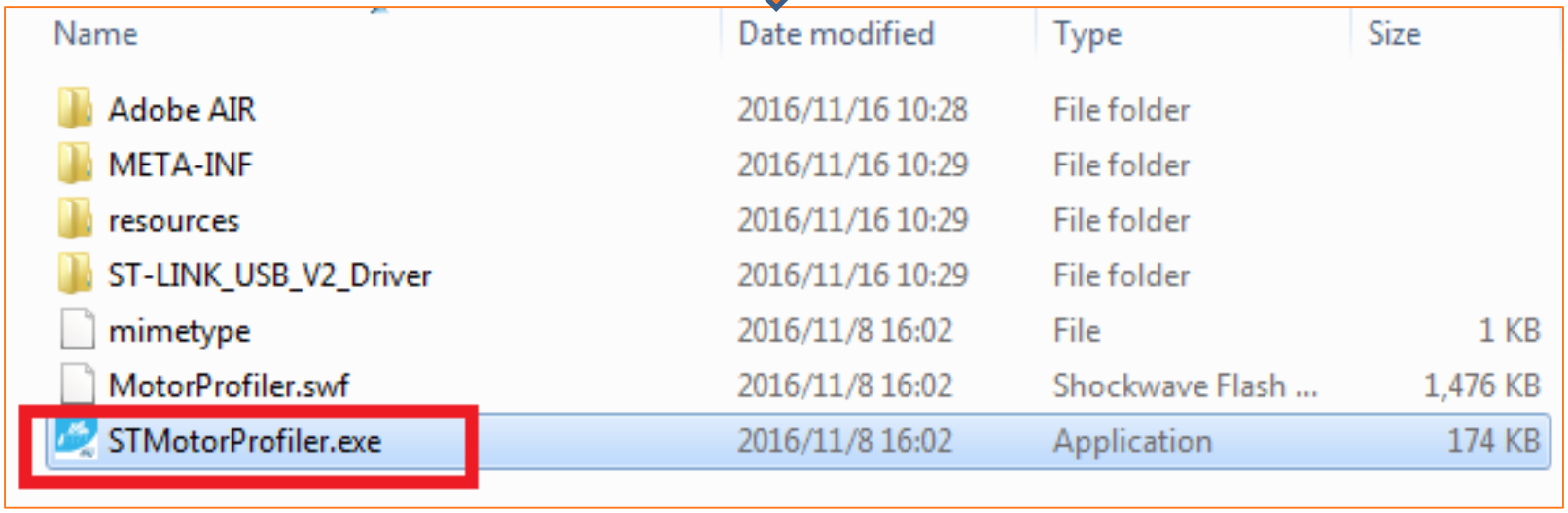
10~15min

life.augmented

操作说明1/4

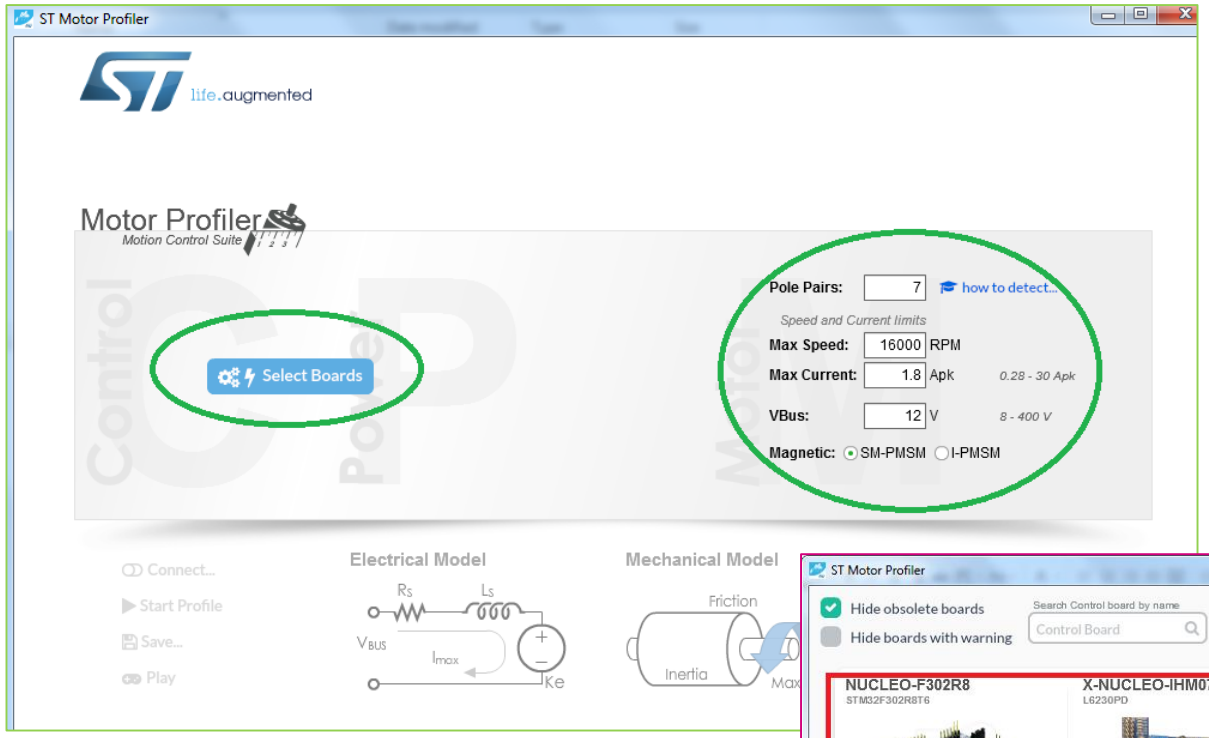
- 找到MotorProfiler软件所在位置，双击打开

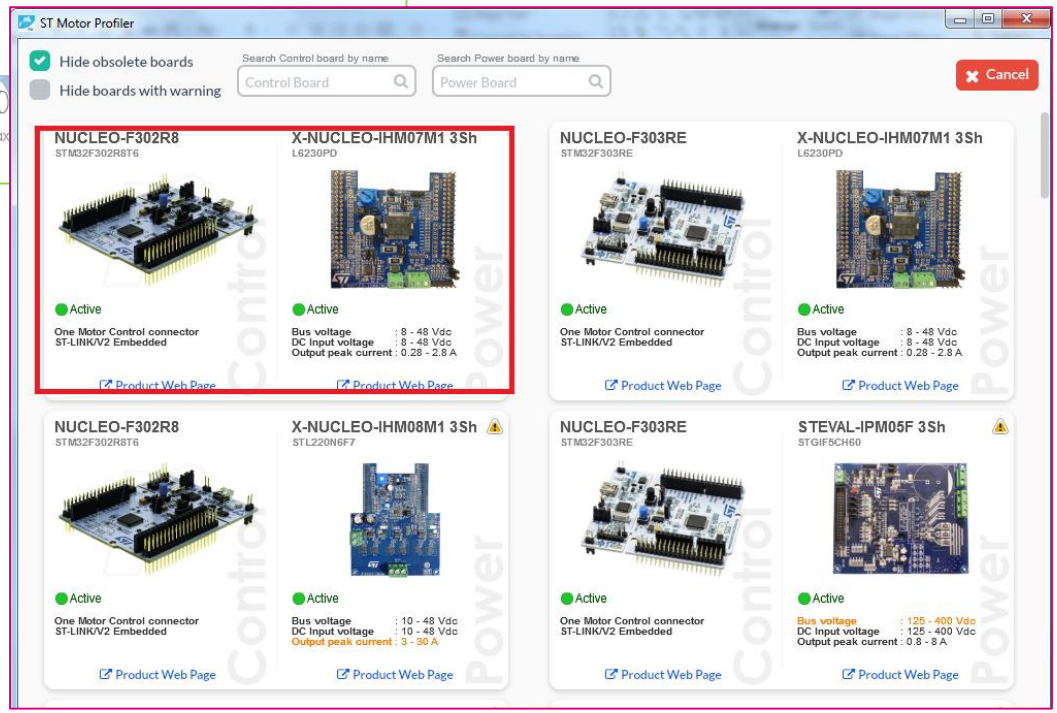




操作说明2/4

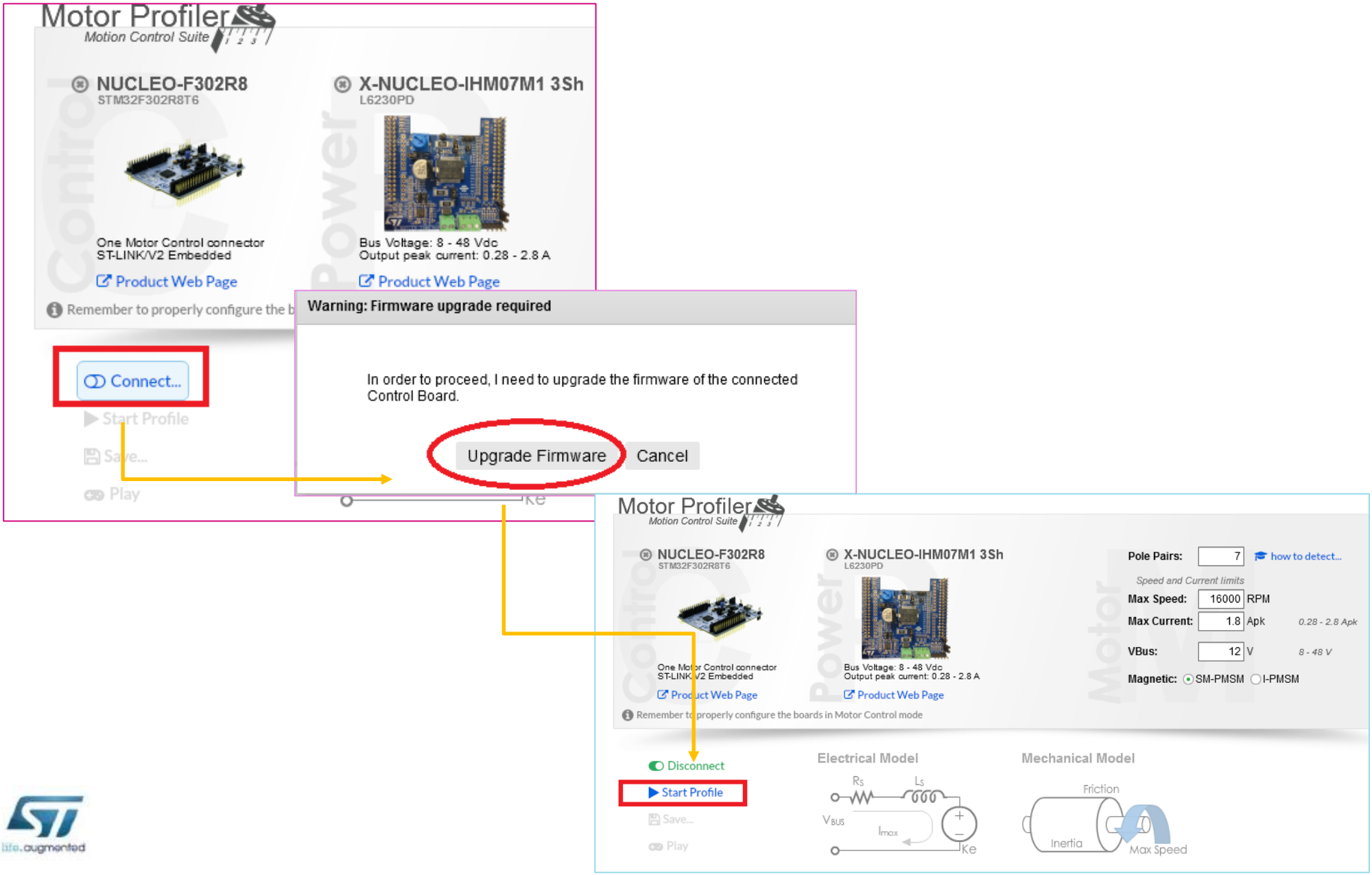
- 选择电机控制套件，并配置基础参数





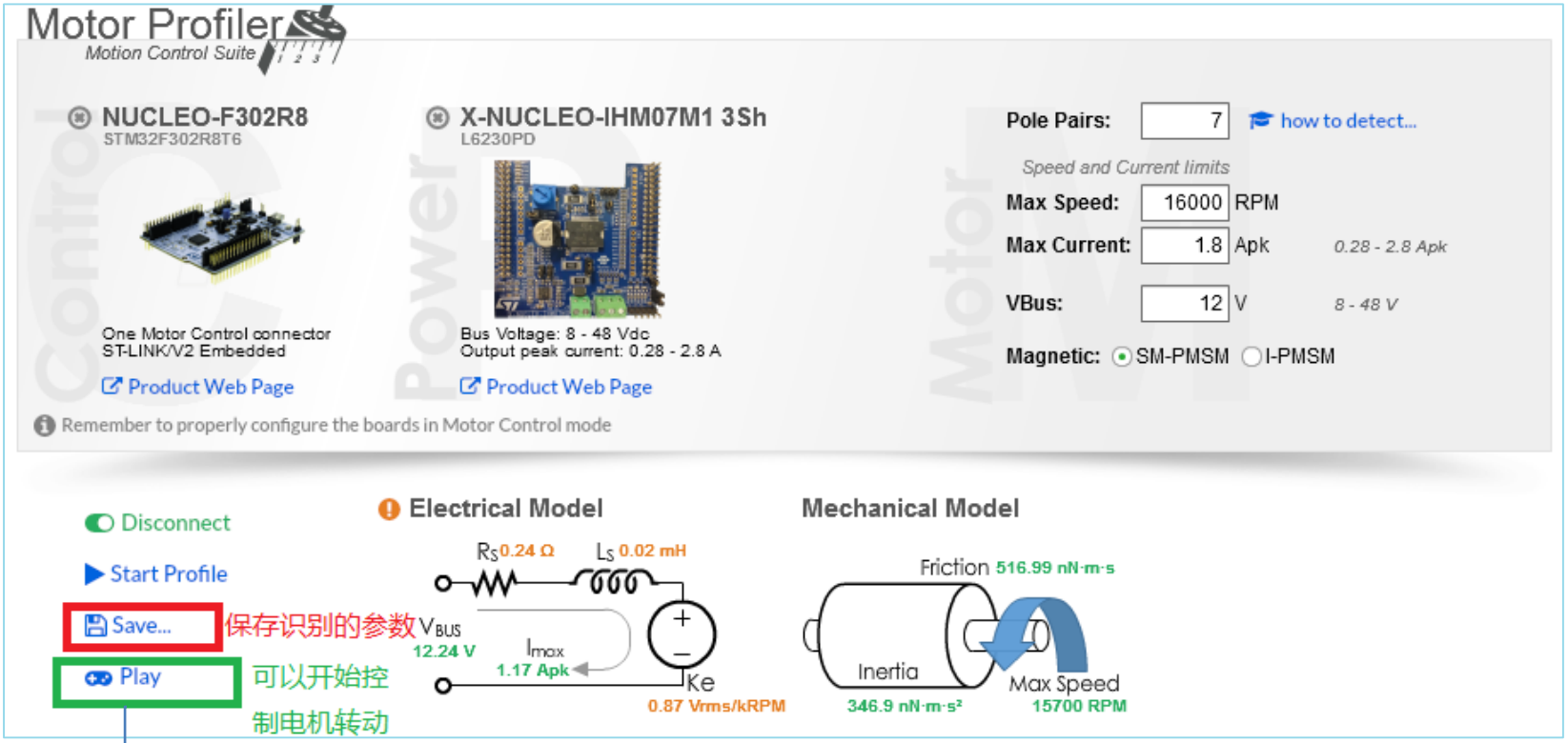
操作说明3/4

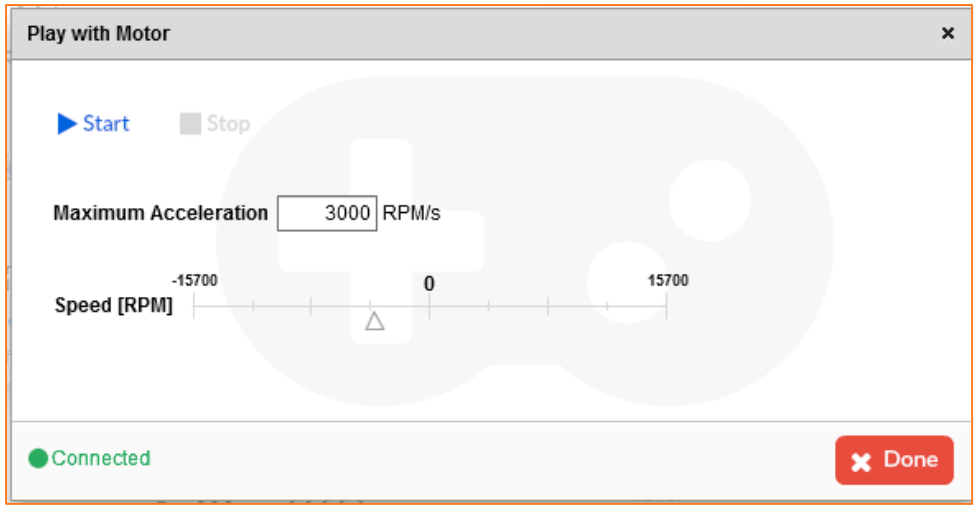
- 连接下载程序后，点击开始参数识别 – Start Profile

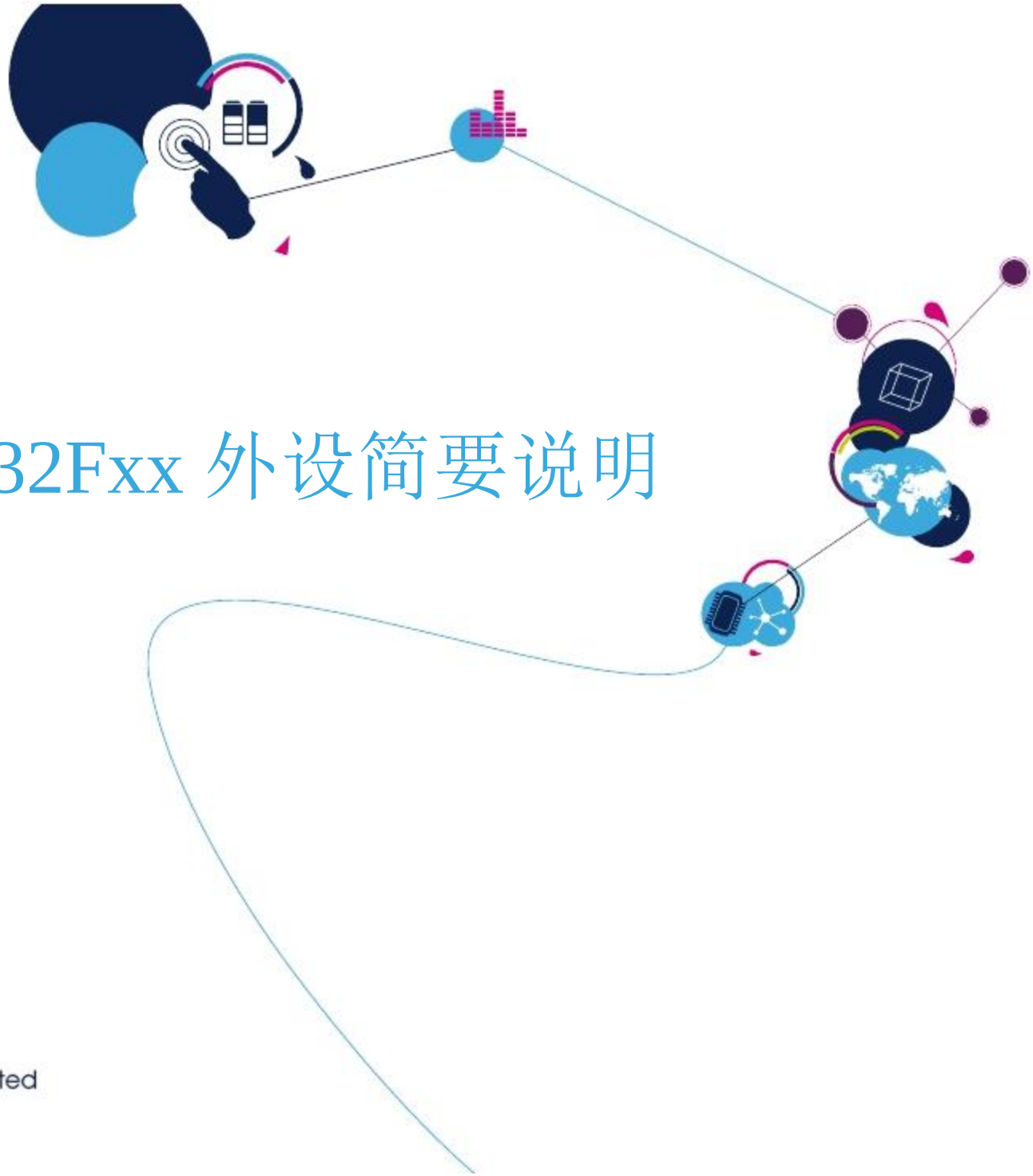


操作说明4/4

- 参数识别成功后，可以保存参数以及运行电机





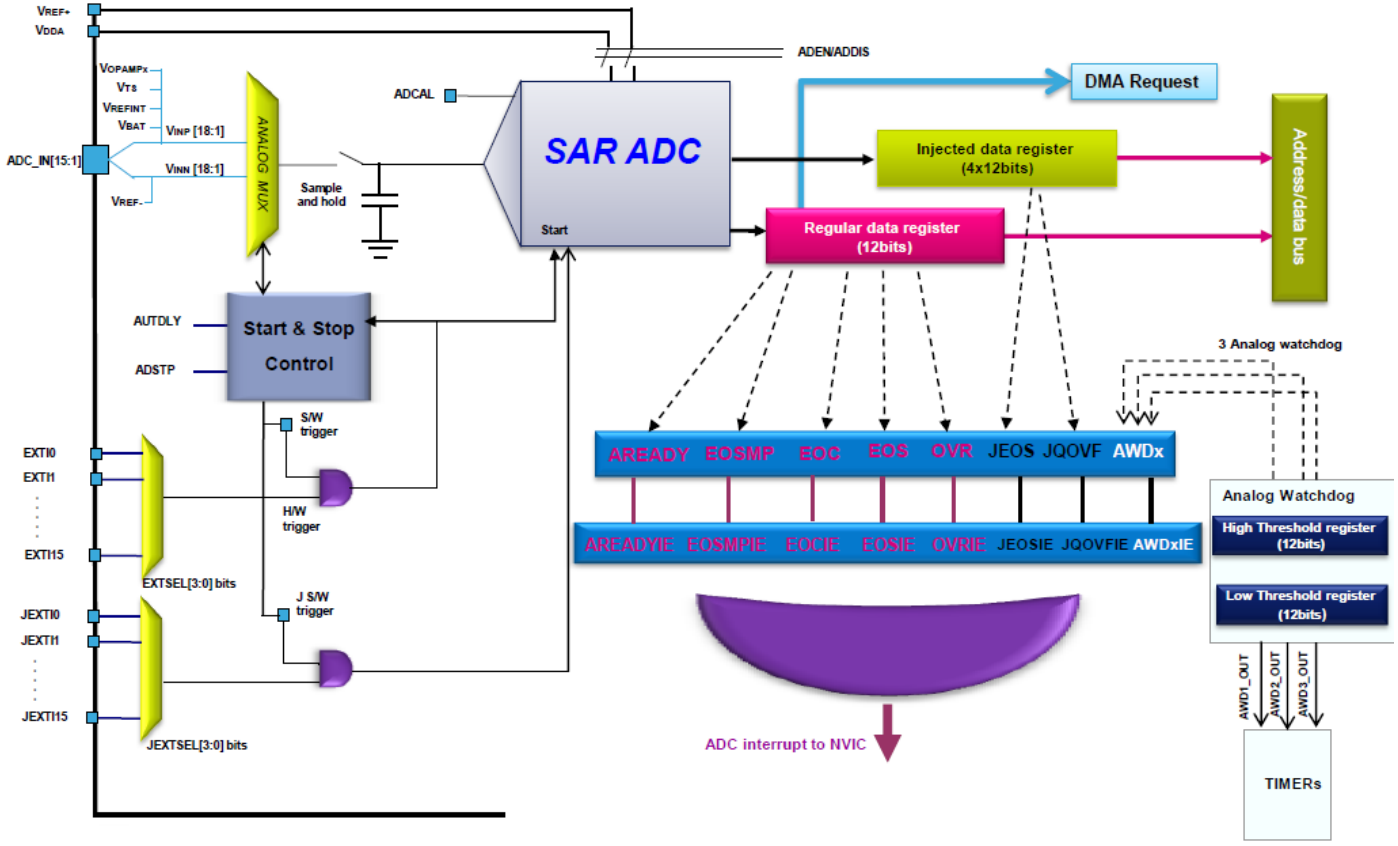


马达相关STM32Fxx 外设简要说明

life.augmented

ADC模块

- 时间组成：采样时间 + 转换时间
  - 采样可以选择不同cycle进行，cycle对应ADC时钟频率
  - 转换时间固定，比如12-bit下为12.5cycle，10-bit下为10.5cycle
- 拥有Inject和Regular两种通道转化组（除STM32F0系列）
  - Regular多个通道ADC转化需要使用DMA
  - Inject通道有四个独立转化结果寄存器
- 可编程通道转化顺序，如CH1，CH5，CH2…..
- 多个触发源，灵活配置ADC开始转换的时刻



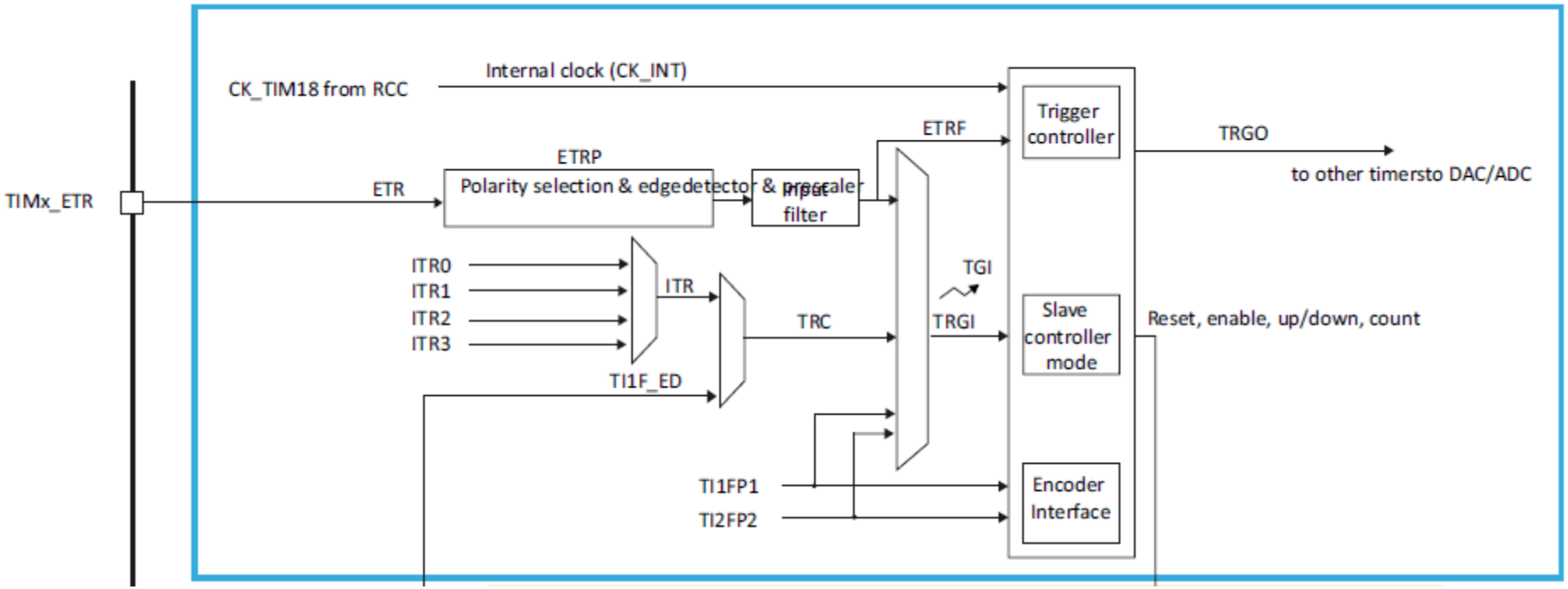
ADC相关培训文档

描述	下载地址
STM32F0模数转换模块（ADC）介绍	下载地址
STM32F1其余模块（包括ADC和计数器）介绍	下载地址
STM32F2模数转换模块（ADC）介绍	下载地址
STM32F3模数转换模块（ADC）介绍	下载地址
How to get the best ADC accuracy in STM32Fx Series and STM32L1 Series devices	下载地址
STM32™'s ADC modes and their applications	下载地址
Improving STM32F1x and STM32L1x ADC resolution by oversampling	下载地址
使用STM32F2xx 和STM32F4xx 微控制器时如何提高ADC 测量精度	下载地址
STM32F30x ADC modes and application	下载地址

life.augmented

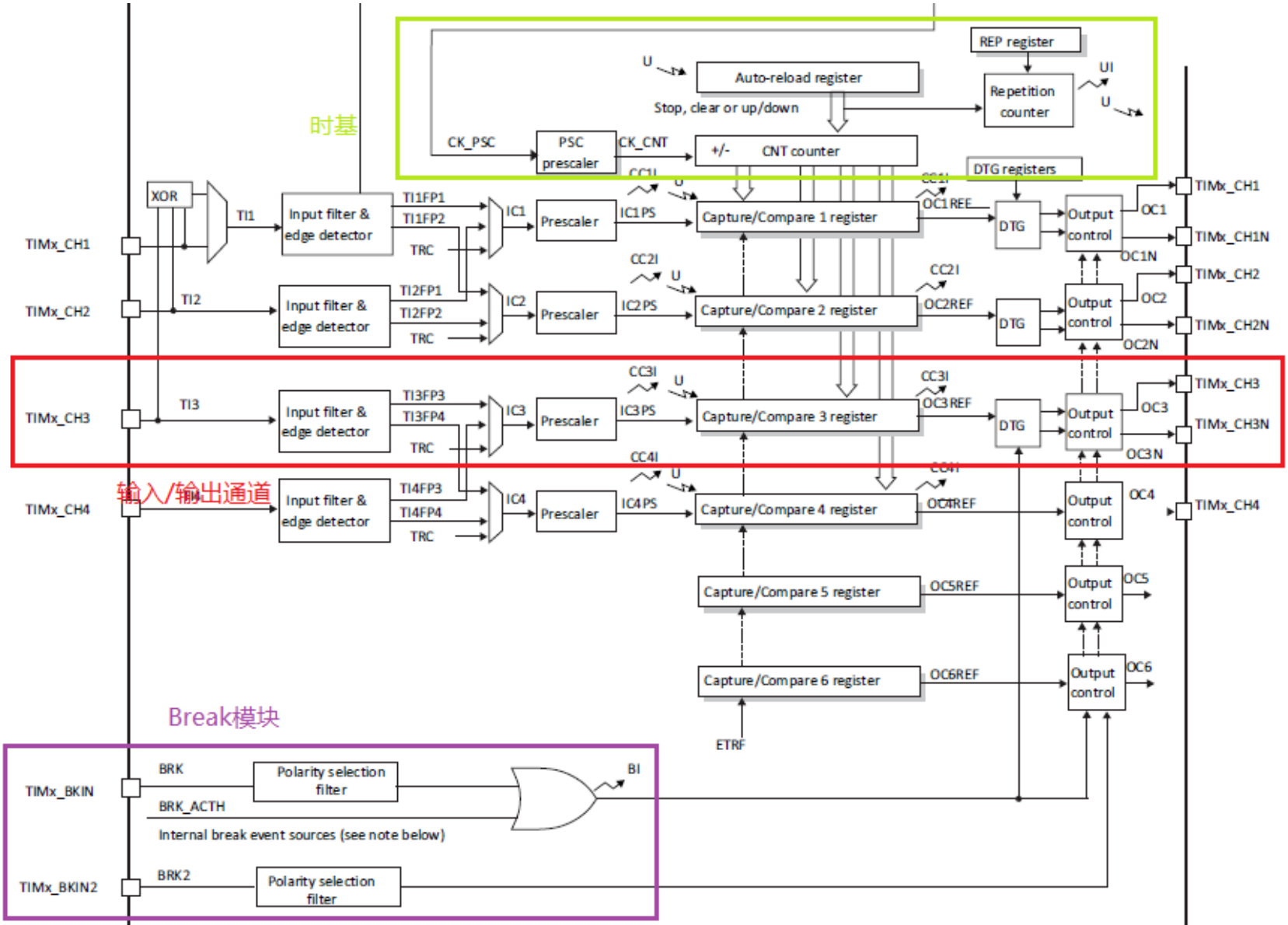
Advanced Timer

- Master/Slave 控制模块
- 外部触发控制
- 触发或门控模式



Advanced Timer

- 时基，向上/向下，中心计数模式
- 输入/输出通道，捕获输入/ PWM输出
- Break模块，电流保护，电压保护

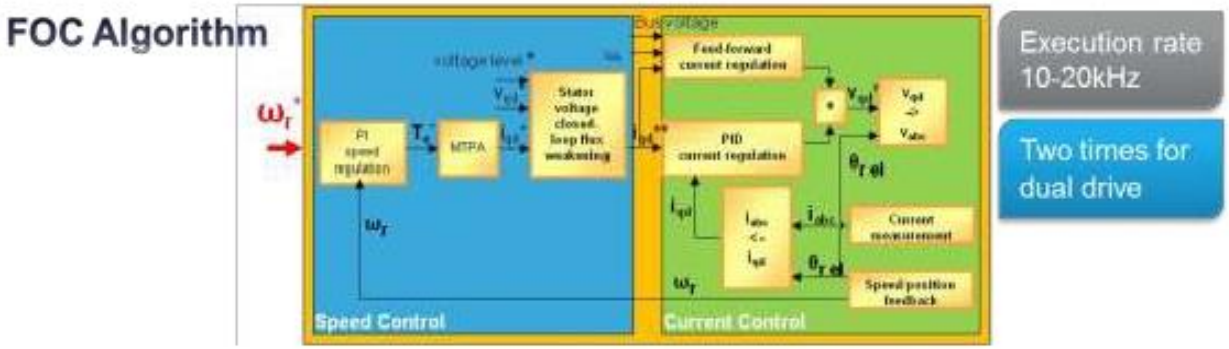


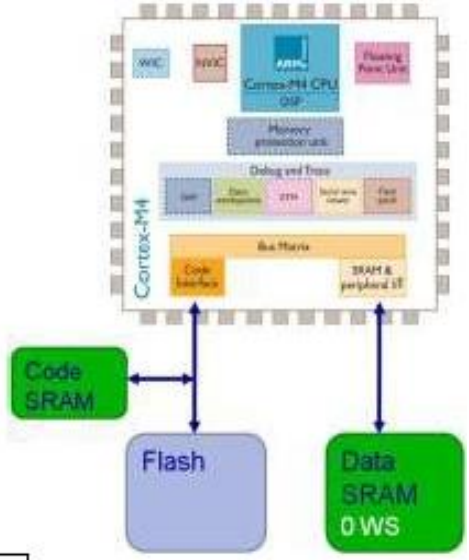
Timer参考资料

描述	链接
STM32F0定时器模块（TIM）介绍	下载地址
STM32F1其余模块（包括ADC和计数器）介绍	下载地址
STM32F2定时器模块（TIM）介绍	下载地址
STM32F3通用定时器模块（TIM）介绍	下载地址
STM32F3定时器模块新增功能介绍	下载地址
General-purpose timer cookbook	下载地址
STM32F1xx、STM32F2xx、STM32F4xx、STM32L1xx、 STM32F30/31/37/38x 定时器概览	下载地址

life.augmented

Cortex M4F - CCMRAM





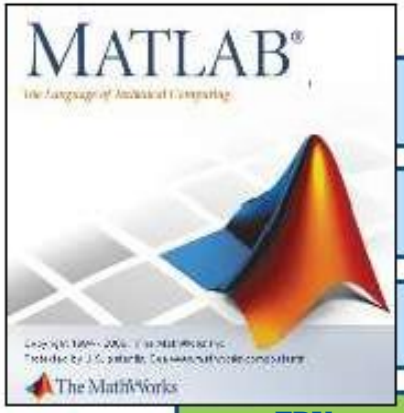
Cortex-M0
"8/16-bit" applications

Cortex-M3
"16/32-bit" applications
MCU

Cortex-M4
"32-bit/DSC" applications

90 Dhrystone MIPS

Binary and tool compatible



High level approach
Matrix, mathematical equations

Meta language tools
Matlab ,Scilab…etc…

C code generation
Floating point numbers (`float`)

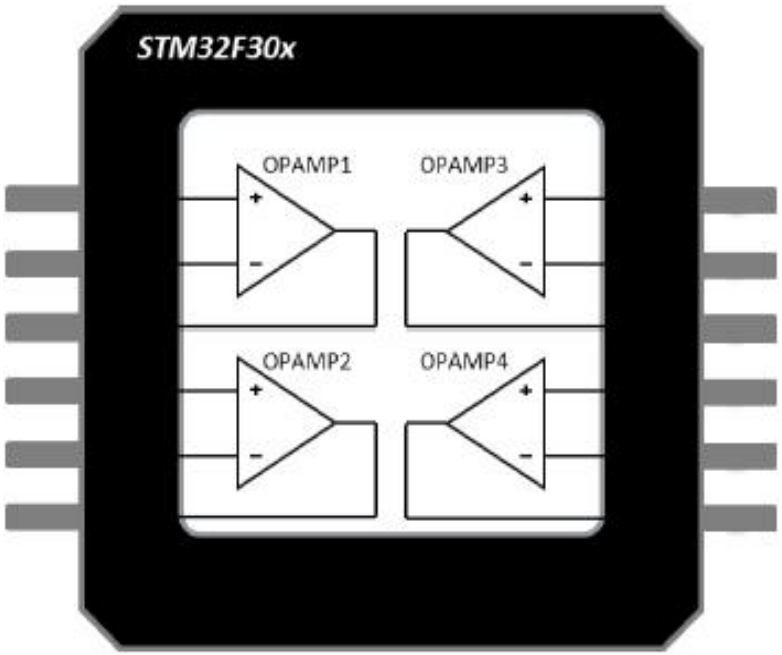
FPU
Direct mapping
No code modification
High performance
Optimal code efficiency

No FPU
Usage of SW lib
No code modification
Low performance
Medium code efficiency

No FPU
Usage of integer based format
Code modification
Corner case behavior to be checked (saturation, scaling)
Medium/high performance
Medium code efficiency

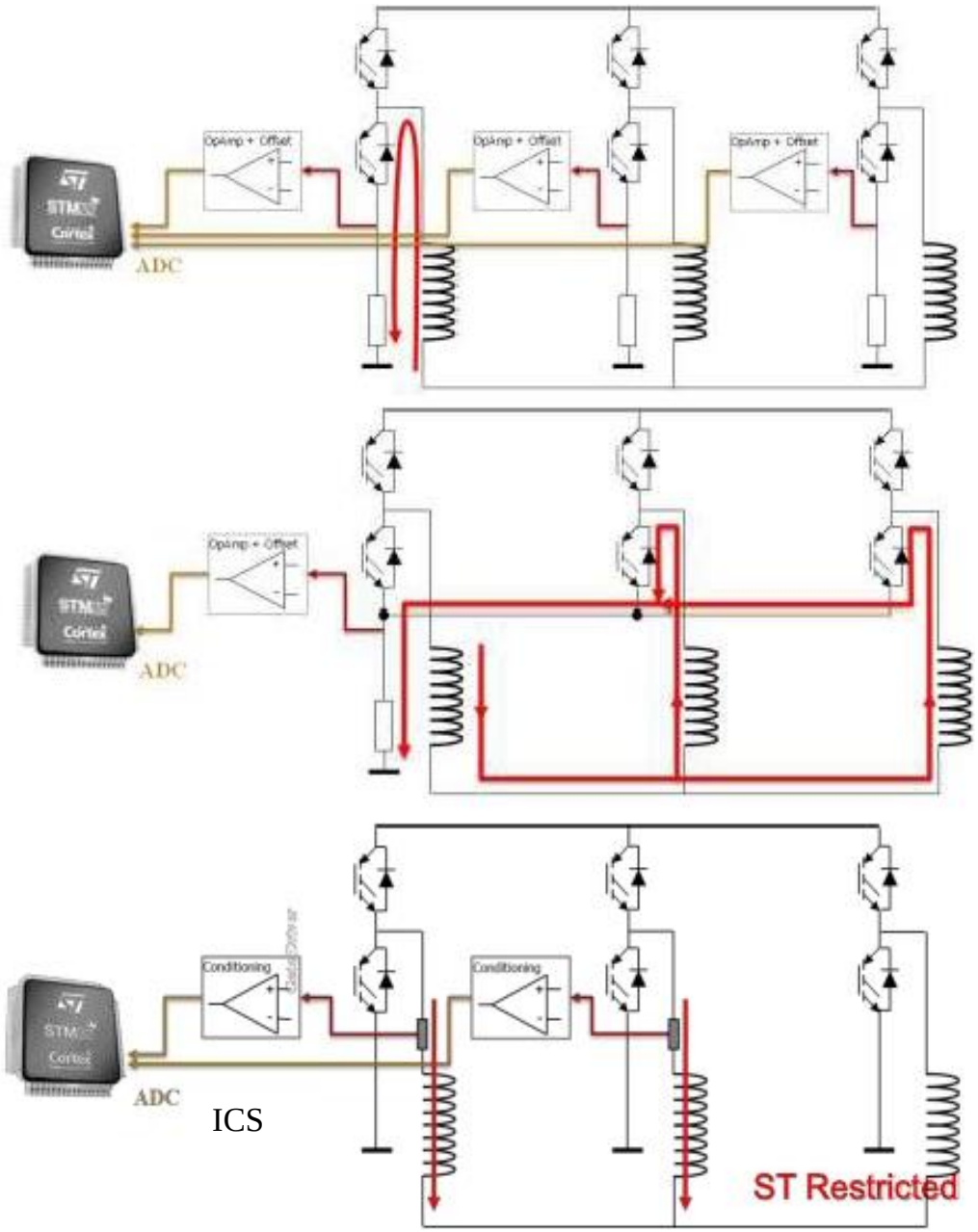
life.augmented

集成运放

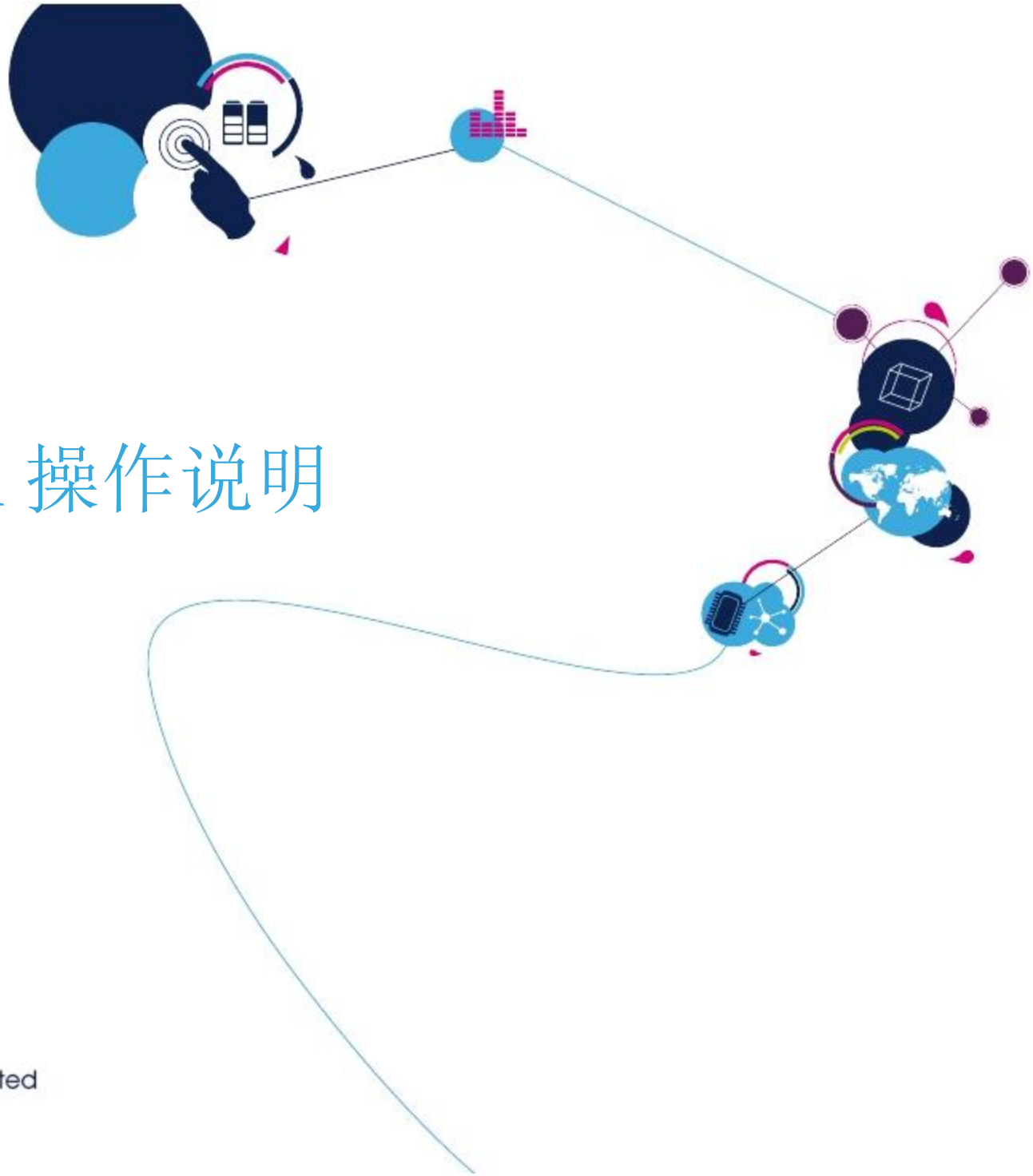


Characteristics

- 8.2 MHz bandwidth
- 4.7V/us slew rate
- 0.5 mA output capability
- Rail-to-rail input/output
- In PGA mode, the gain can be programmed to be x2, x4, x8 or x16.



ICS



Workbench 操作说明

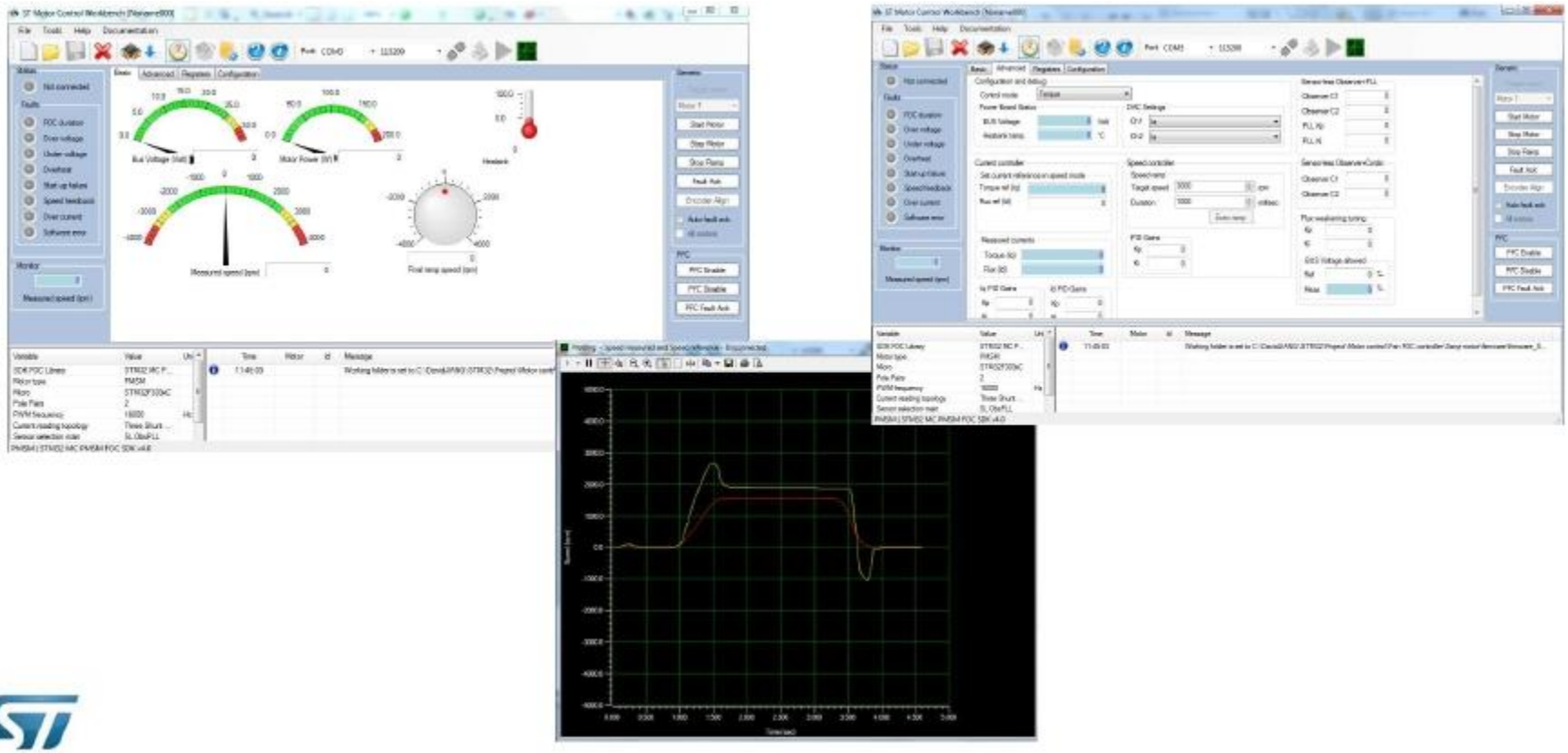
life.augmented

工具：ST MC Workbench 1/2

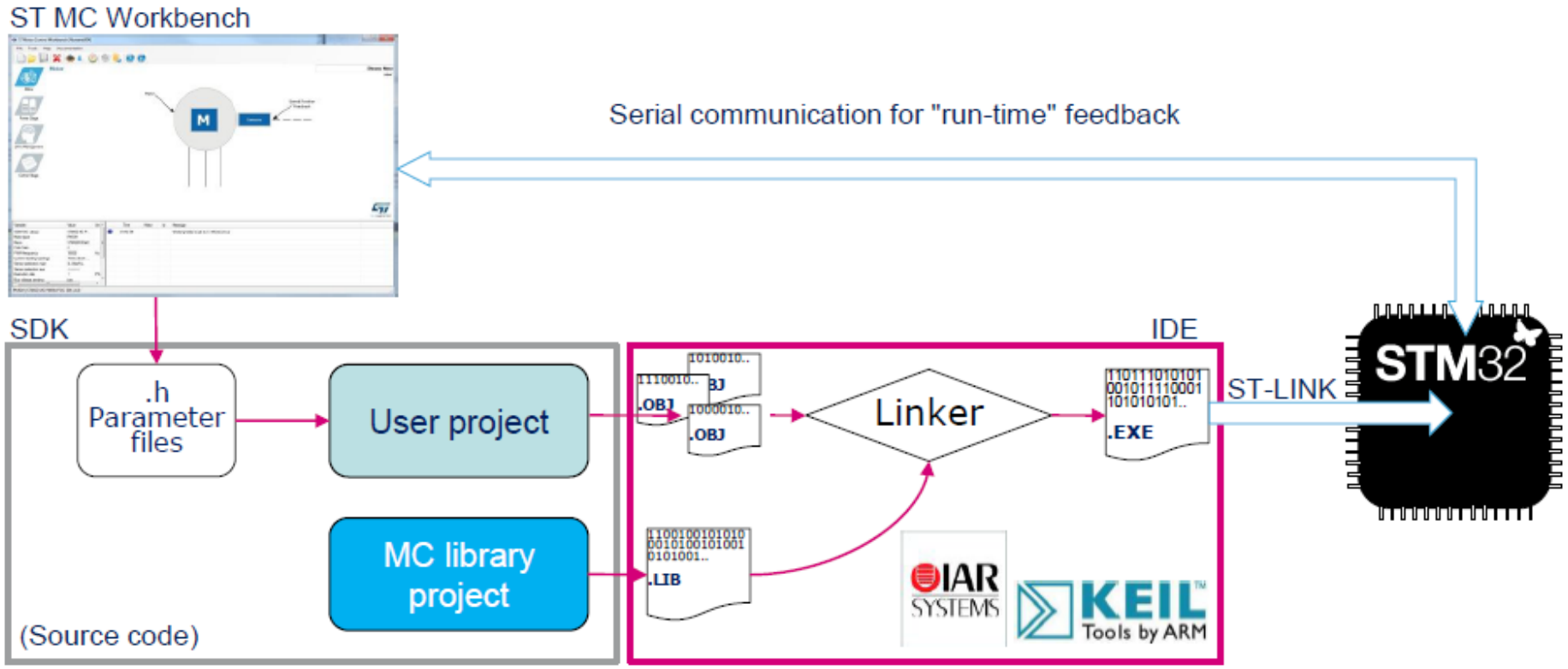
- STMCWB PC软件通过图形界面产生软件库的参数头文件，方便用户对软件库的配置及马达的调试

工具：ST MC Workbench 2/2

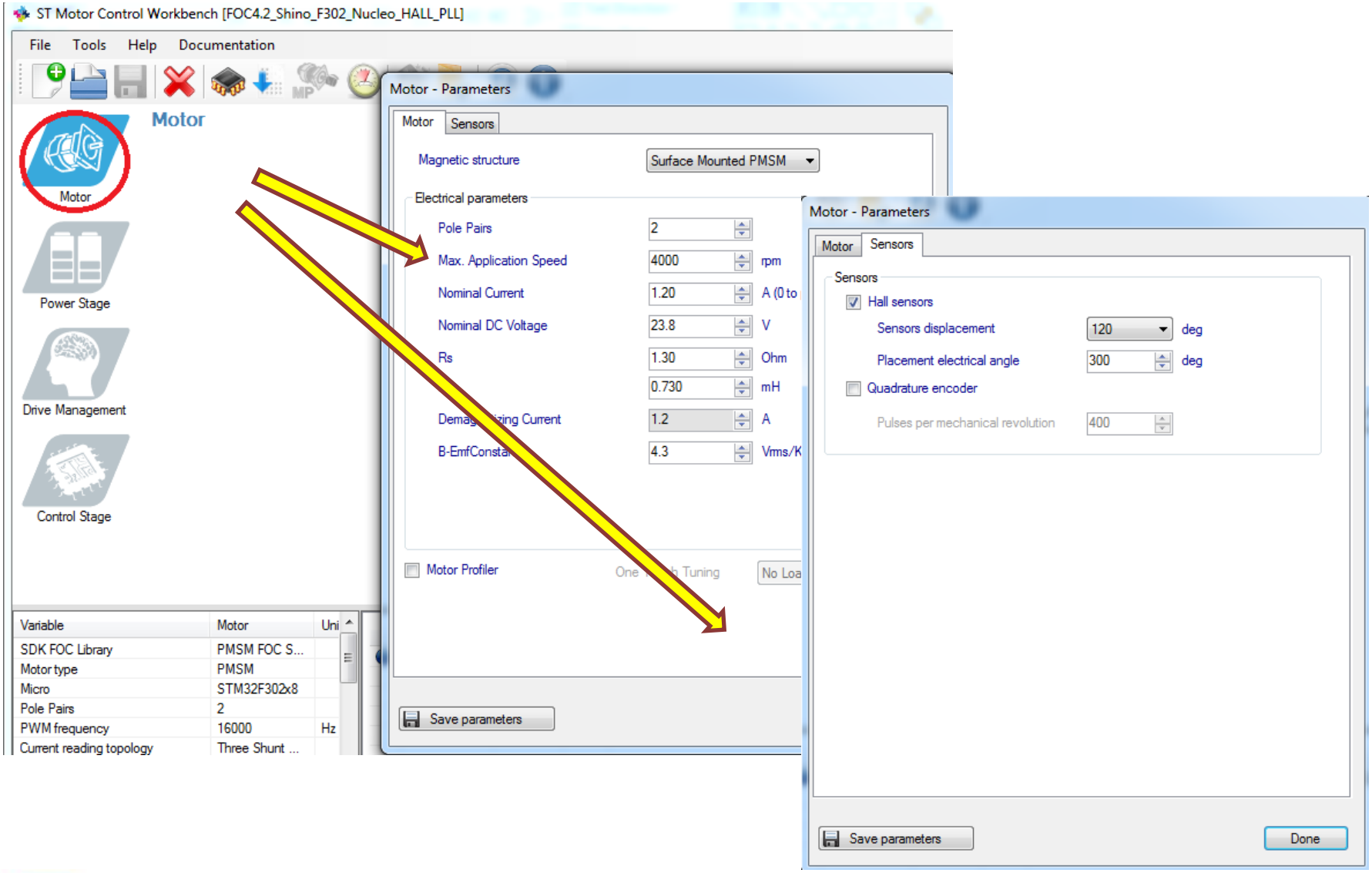
- 实时串口通讯
  - 与PC实时通讯，实现马达的启动/停止，或速度的加减速
  - 调试和监控软件参数
  - 实时画出马达的参考及实际速度曲线



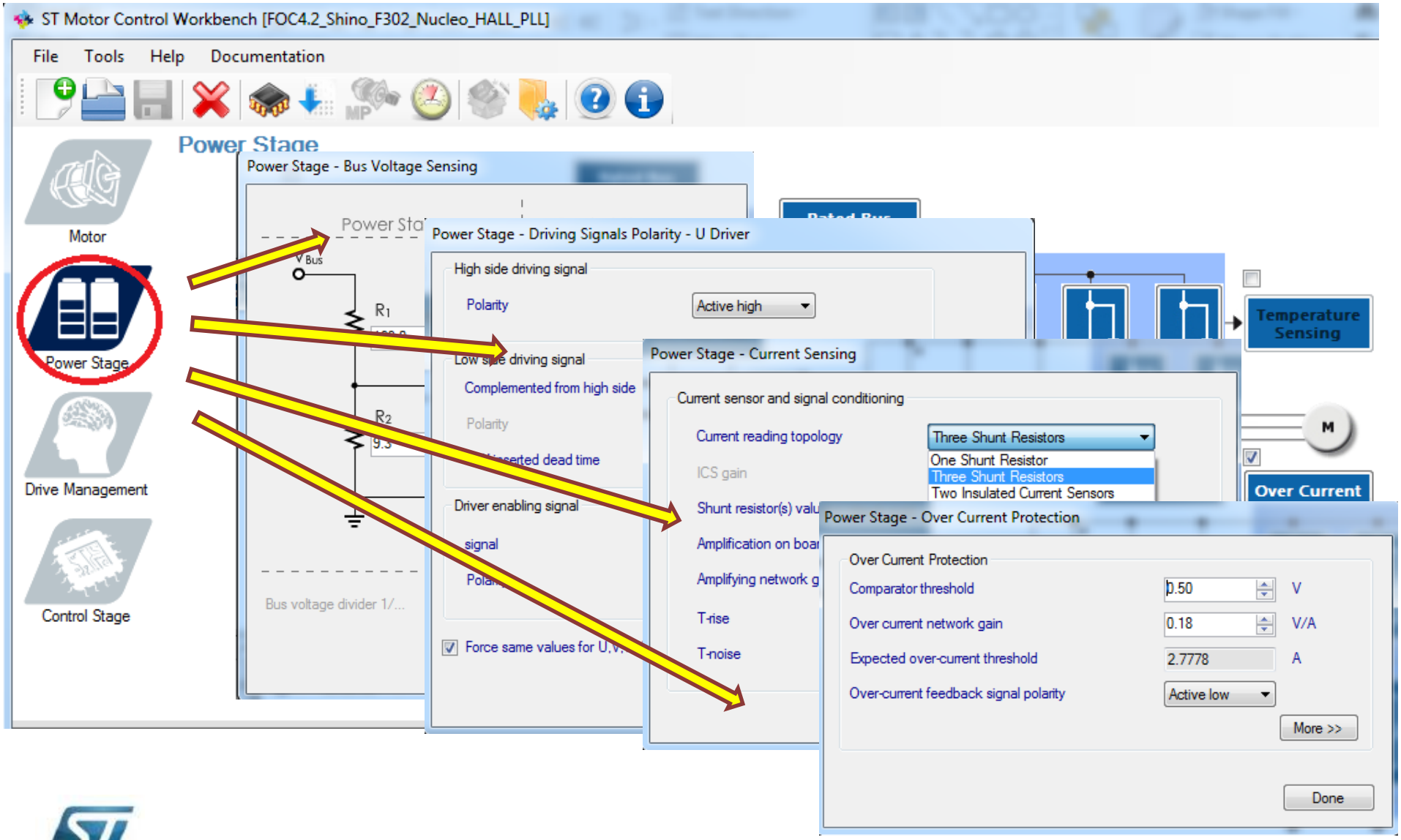
FOC V4.3 软件库操作步骤



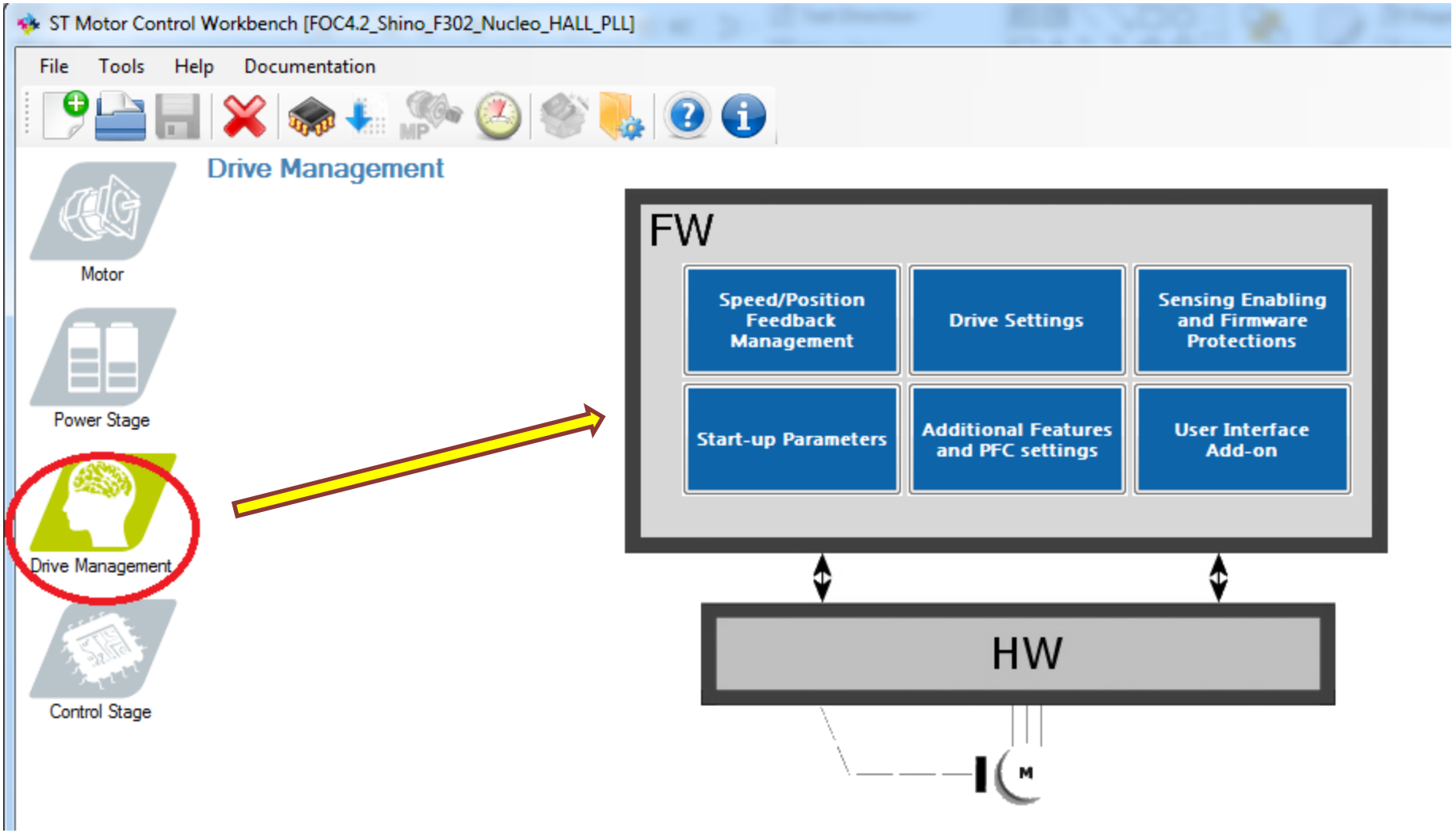
MC Workbench配置（1）



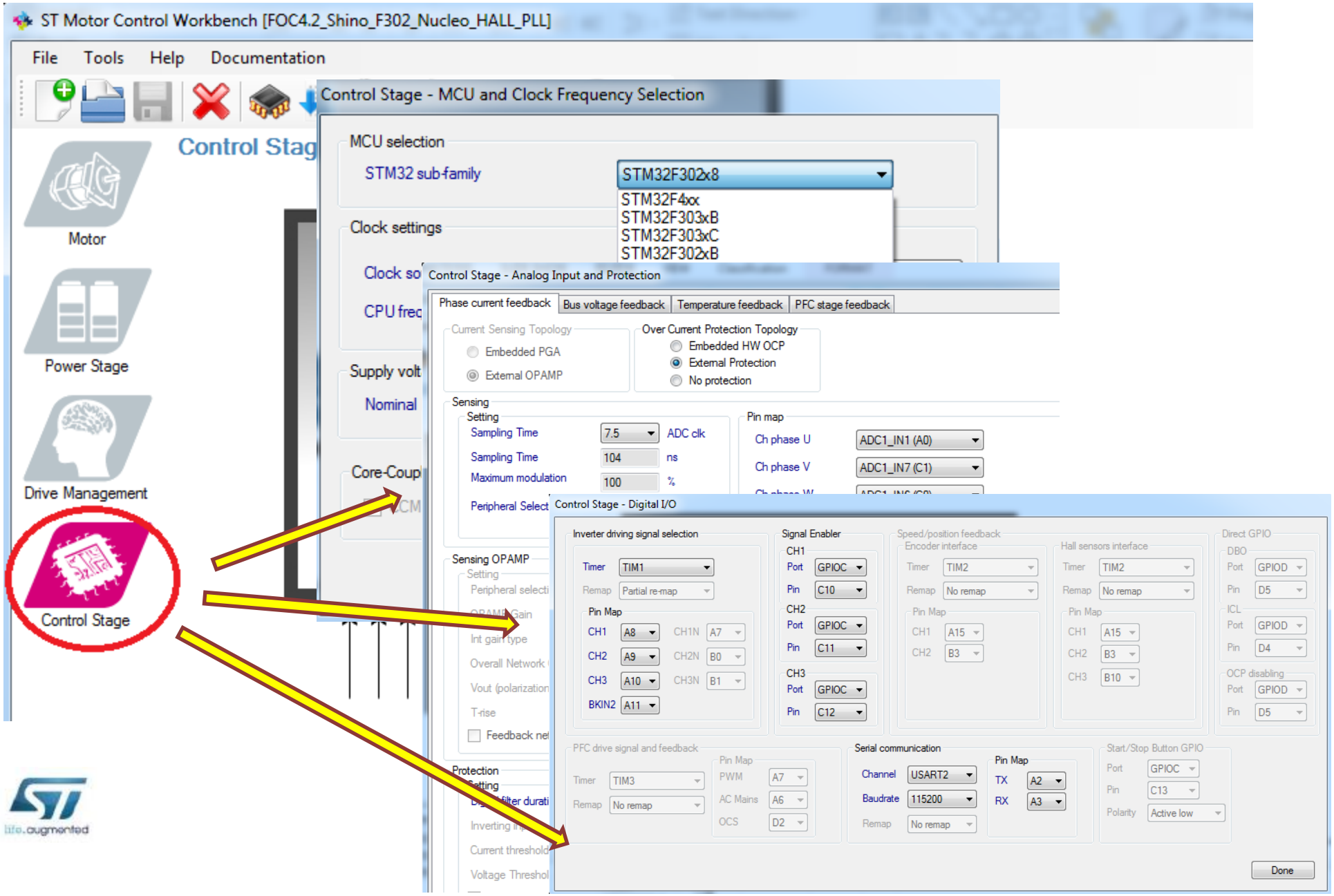
MC Workbench配置（2）



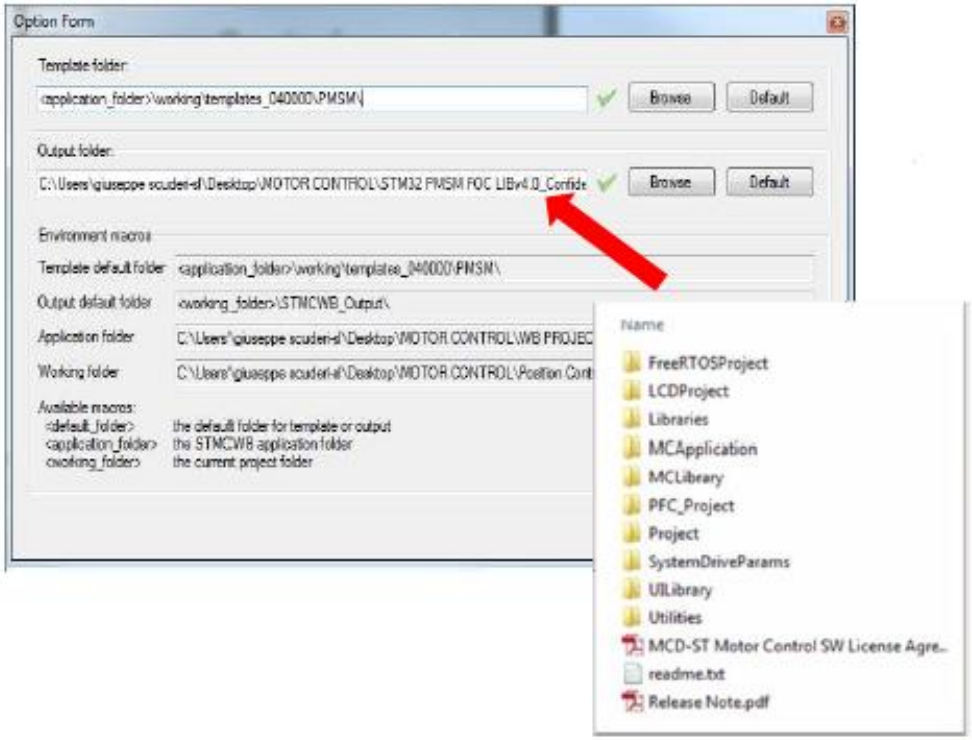
MC Workbench配置（3）



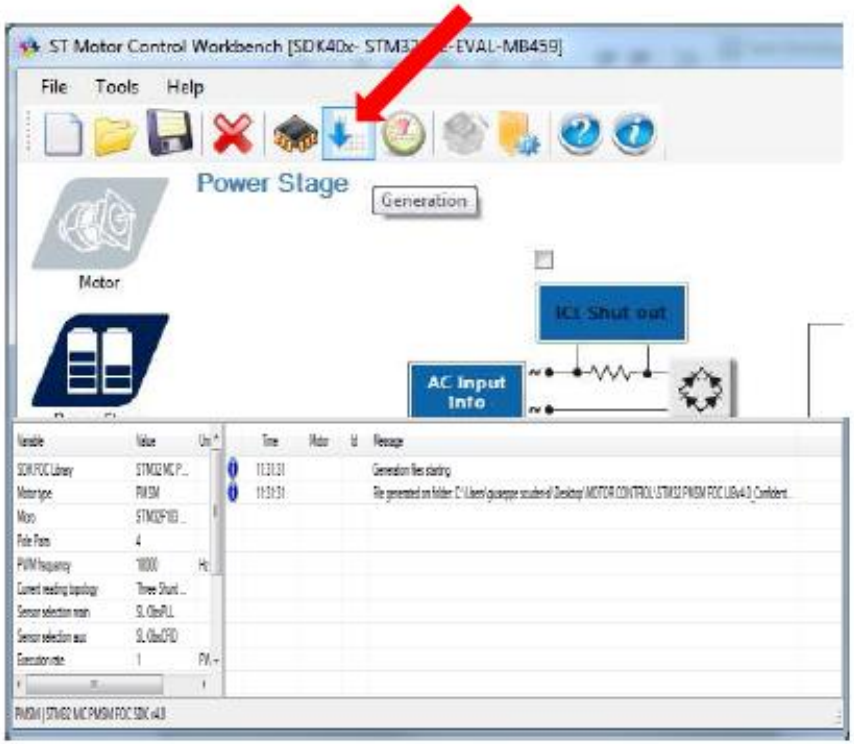
MC Workbench配置（4）



生成头配置文件

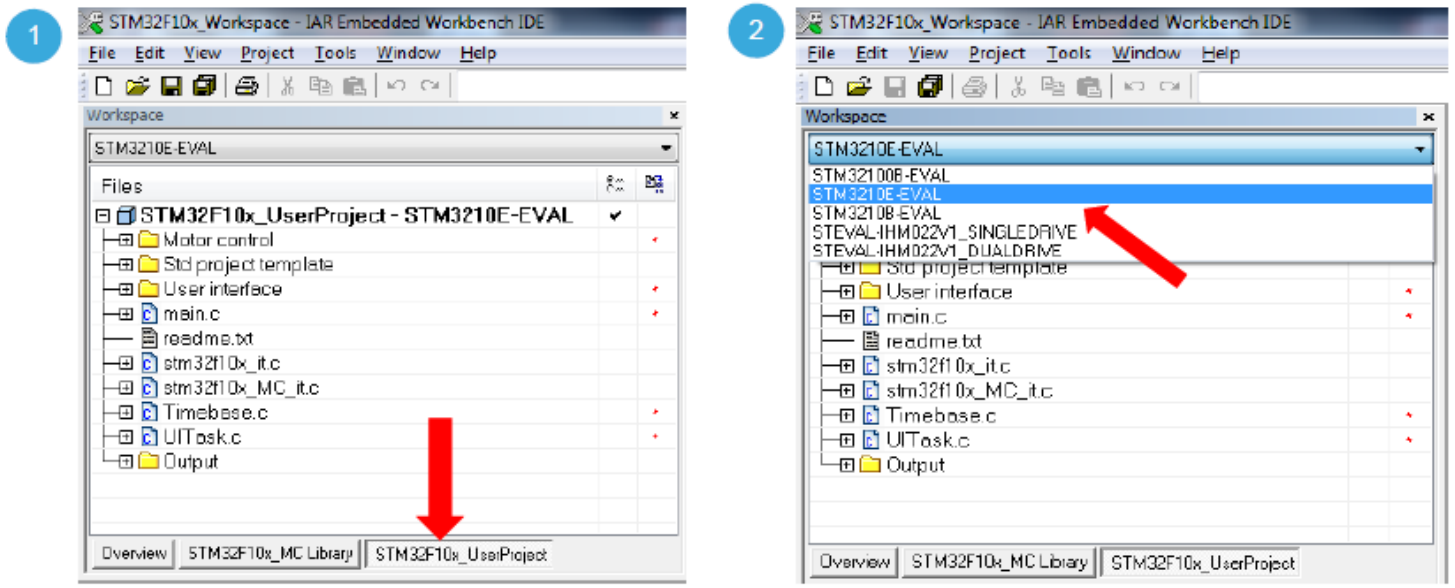


选择头文件保存的文件夹…\SystemDriveParams\



配置完毕后生成头文件

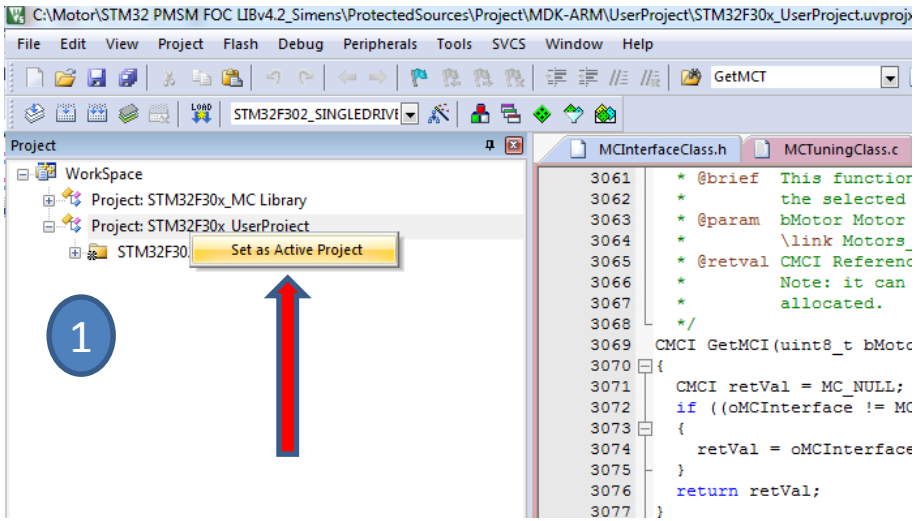
IAR编译链接下载

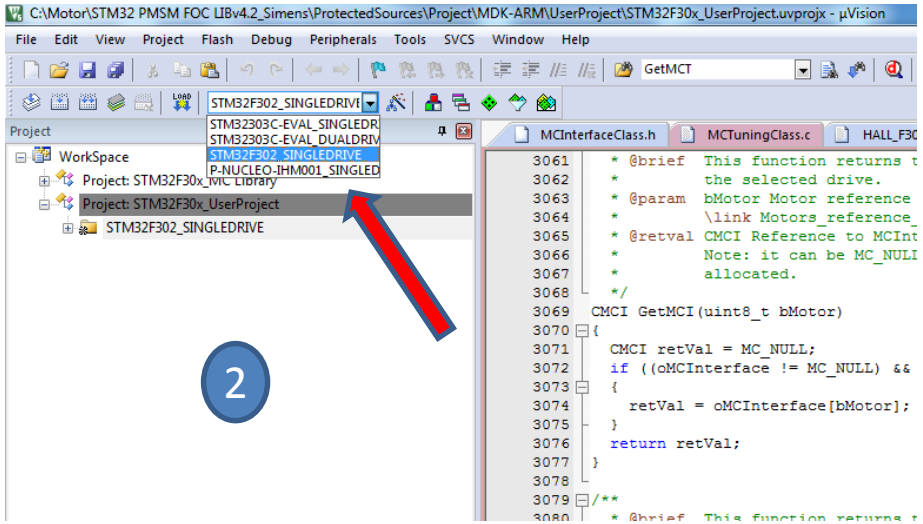


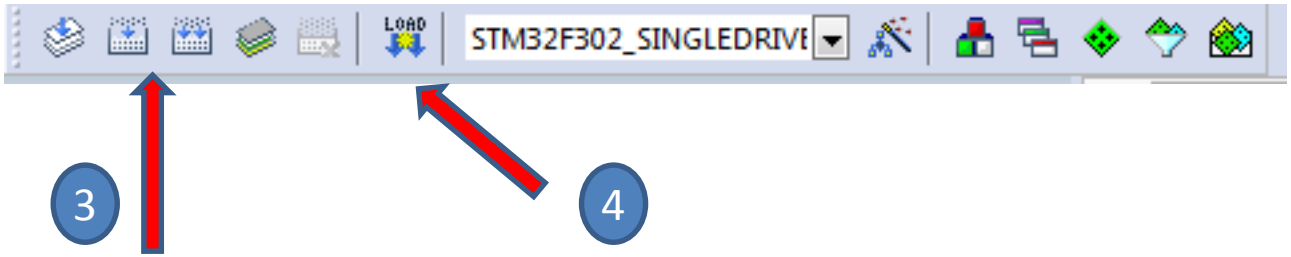
Click on Make 3

4 Click on Download

Keil编译链接下载


C:\Motor\STM32 PMSM FOC LIBv4.2_Simens\ProtectedSources\Project\MDK-ARM\UserProject\STM32F30x_UserProject.uvproj
File Edit View Project Flash Debug Peripherals Tools SVCS Window Help
GetMCT
STM32F302_SINGLEDRIVE
Project
WorkSpace
Project: STM32F30x_MC Library
Project: STM32F30x_UserProject
STM32F30
Set as Active Project
MCInterfaceClass.h
MCTuningClass.c
1


C:\Motor\STM32 PMSM FOC LIBv4.2_Simens\ProtectedSources\Project\MDK-ARM\UserProject\STM32F30x_UserProject.uvprojx - µVision
File Edit View Project Flash Debug Peripherals Tools SVCS Window Help
GetMCT
STM32F302_SINGLEDRIVE
STM32303C-EVAL_SINGLEDR
STM32303C-EVAL_DUALDRIV
STM32F302_SINGLEDRIVE
P-NUCLEO-IHM001_SINGLED
Project
WorkSpace
Project: STM32F30x_MC Library
Project: STM32F30x_UserProject
STM32F302_SINGLEDRIVE
MCInterfaceClass.h
MCTuningClass.c
2


LOAD
STM32F302_SINGLEDRIVE
3
4

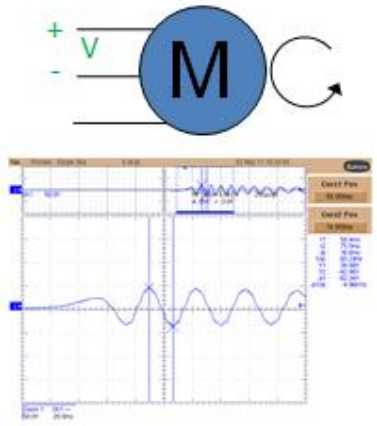
使用GUI串口调试电机

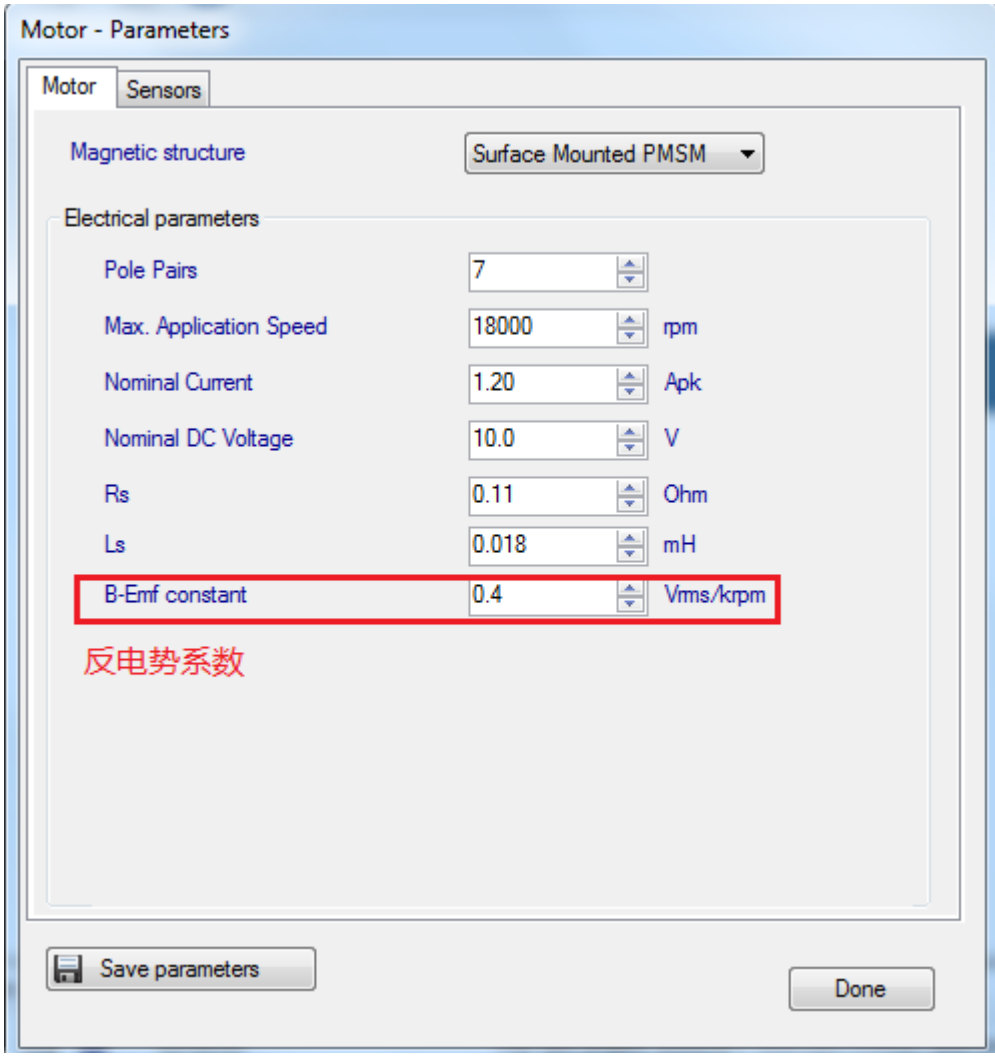
**注意高电压，高电流如果使用电脑调试电机需要加隔离保护，谨慎设置断点

Workbench关键配置参数说明

➢ 马达参数配置

- 极对数
- 最大转速
- 最大电流
- 供电电压
- 电机电阻
- 电机电感
- 电机反电势常数





$$K_e = \frac{V_{Bemf-A}\ [V\ peak-to-peak]\cdot pole\ pairs\ number\cdot 1000}{2\cdot\sqrt{2}\cdot F_{Bemf}\ [Hz]\cdot 60}$$

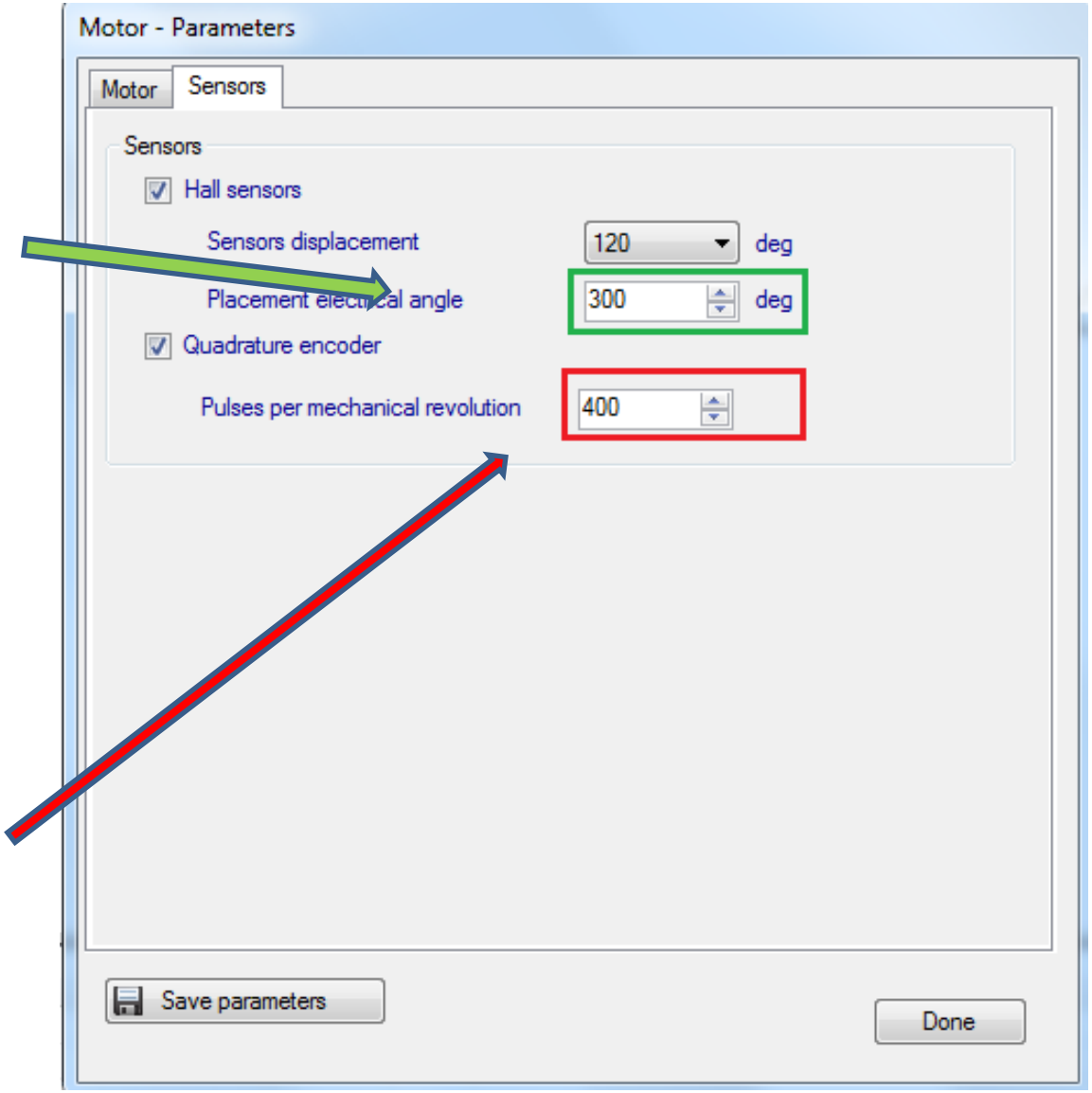
Workbench关键配置参数说明

➢Hall同步电角度：

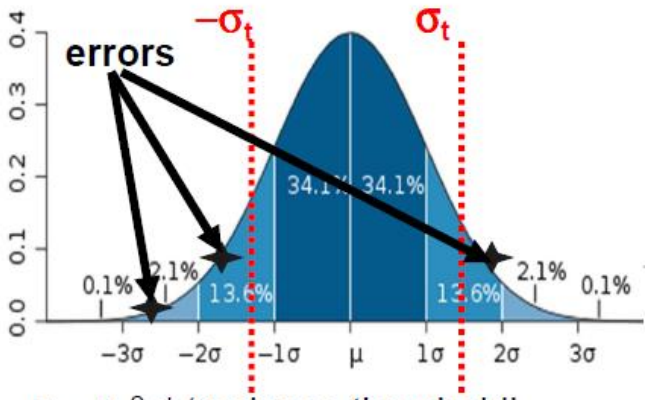
- 电机Hall A的上升沿到电机A相反电动势最高点的延迟角度。
- 默认电机A相的反电动势最高点作为电角度的0度；

➢编码器参数

- 这边Resolution似乎更清晰，分辨率，单位：Pulse/Round
- 旋转一圈的编码器脉冲个数



Workbench关键配置参数说明



$\sigma_\tau = \mu^2$ * 'variance threshold'

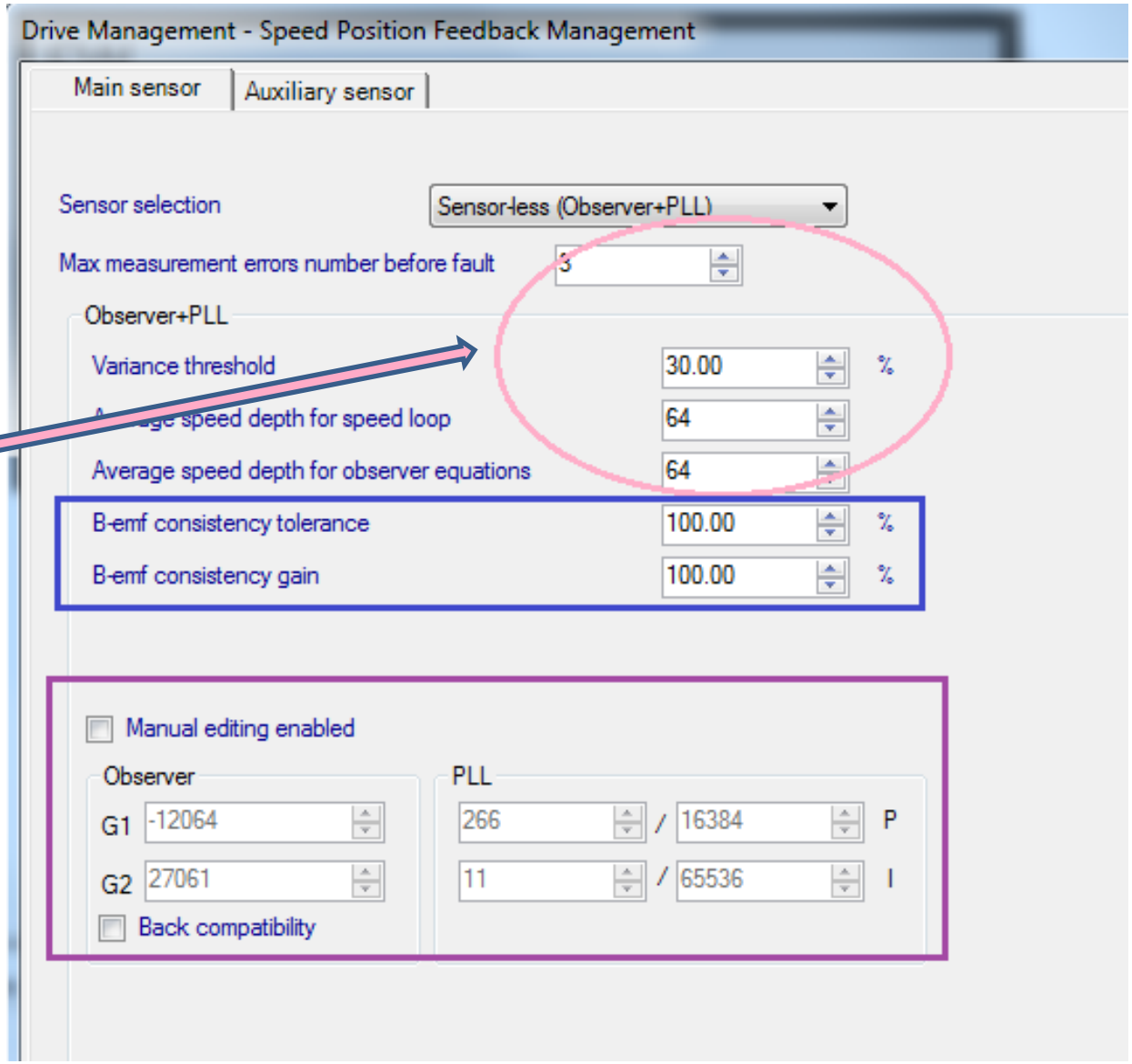
$\sigma 2 \geq \mu 2$ * 'variance threshold'

Speed Average value

$$\mu = \frac{\sum_{i=1}^{64} \omega_i}{64}$$

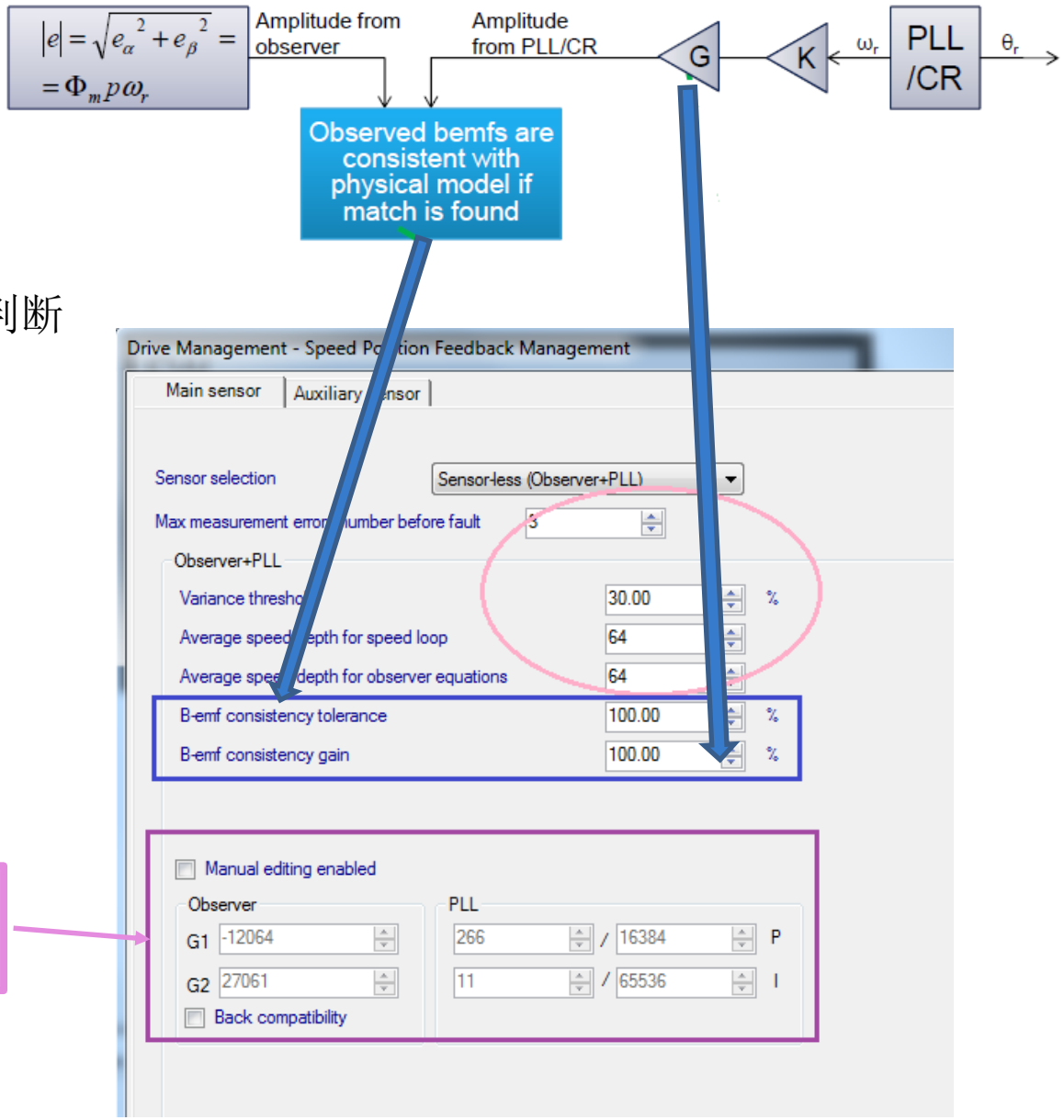
Speed Variance

$$\sigma^2 = \frac{\sum_{i=1}^{64} (\omega_i - \mu)^2}{64}$$



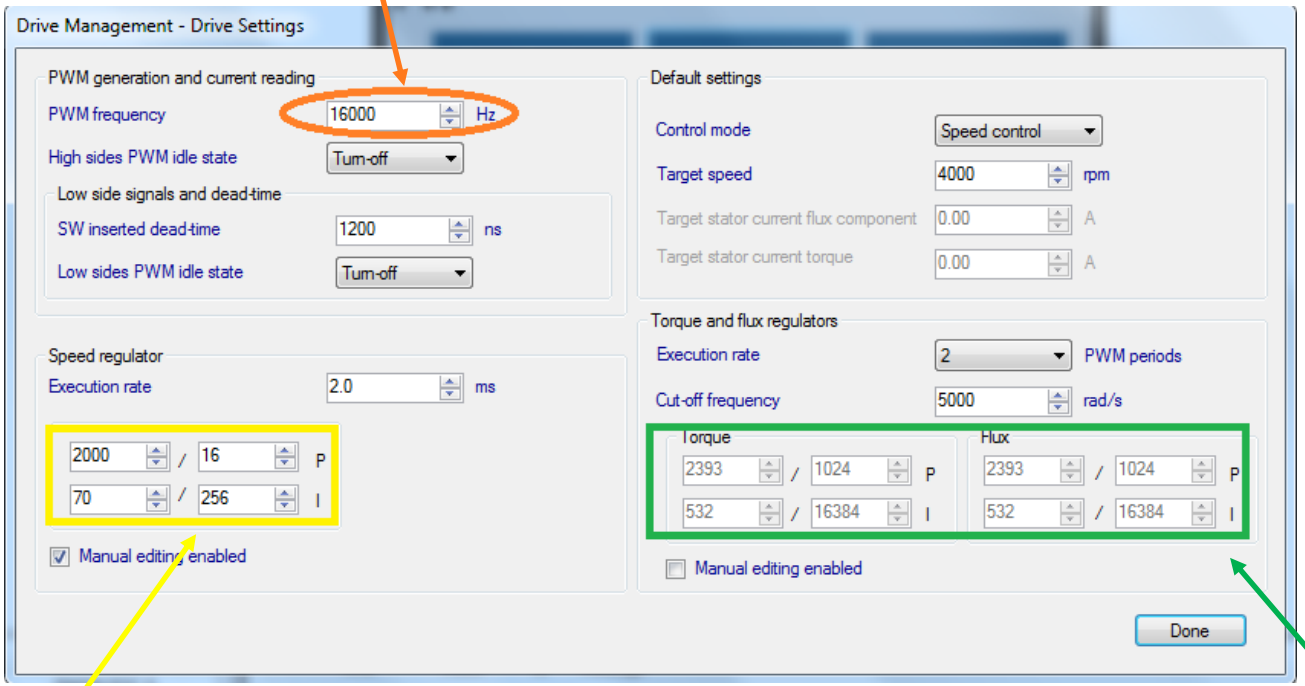
Workbench关键配置参数说明

- BEMF收敛：
  - 一般用于堵转判断
  - 设定为100%时，忽略此判断



根据参数计算得到的系数，一般情况下无需修改，保持为默认数据

Workbench关键配置参数说明



Workbench关键配置参数说明

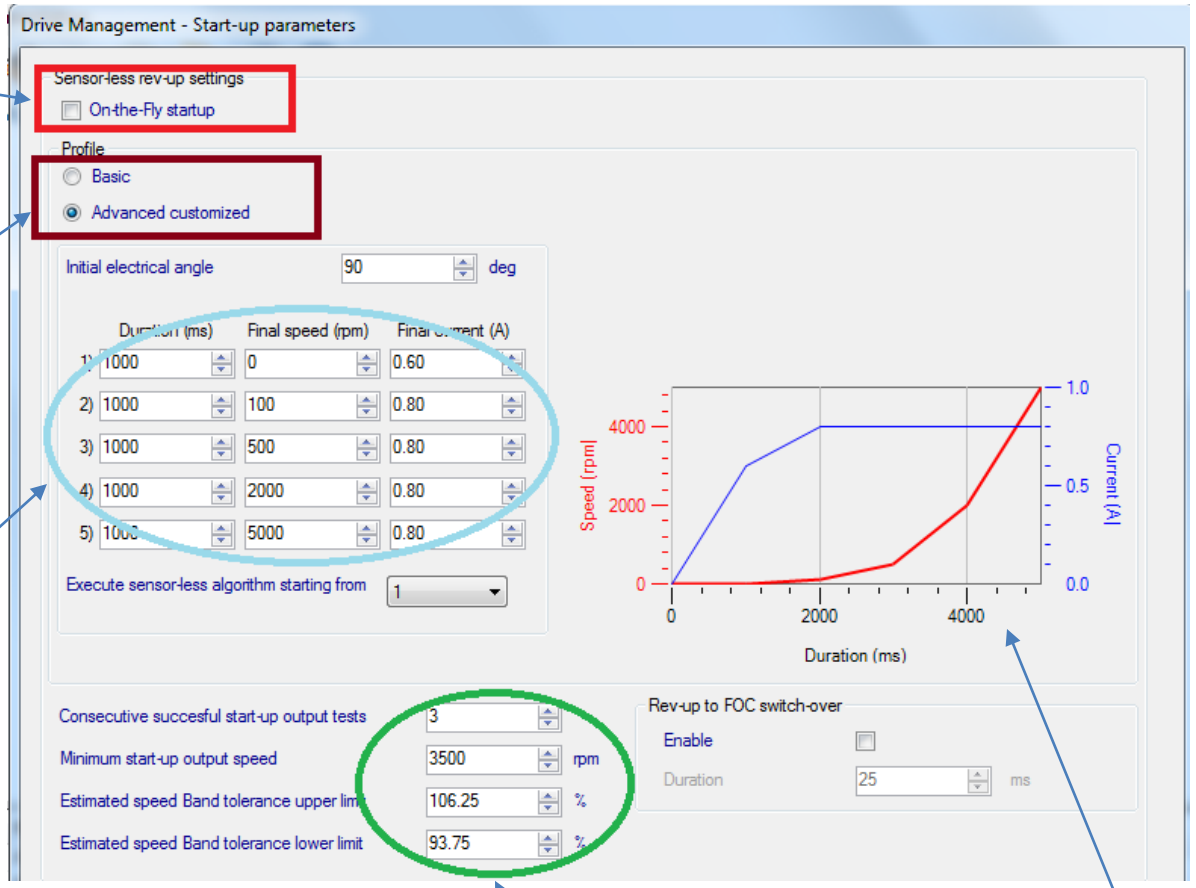
➢ 配置无传感器开环启动参数

- 有传感器不用配置

顺逆风启动检测使能

折线或曲线启动选择

启动曲线配置，包含速度，电流，时间

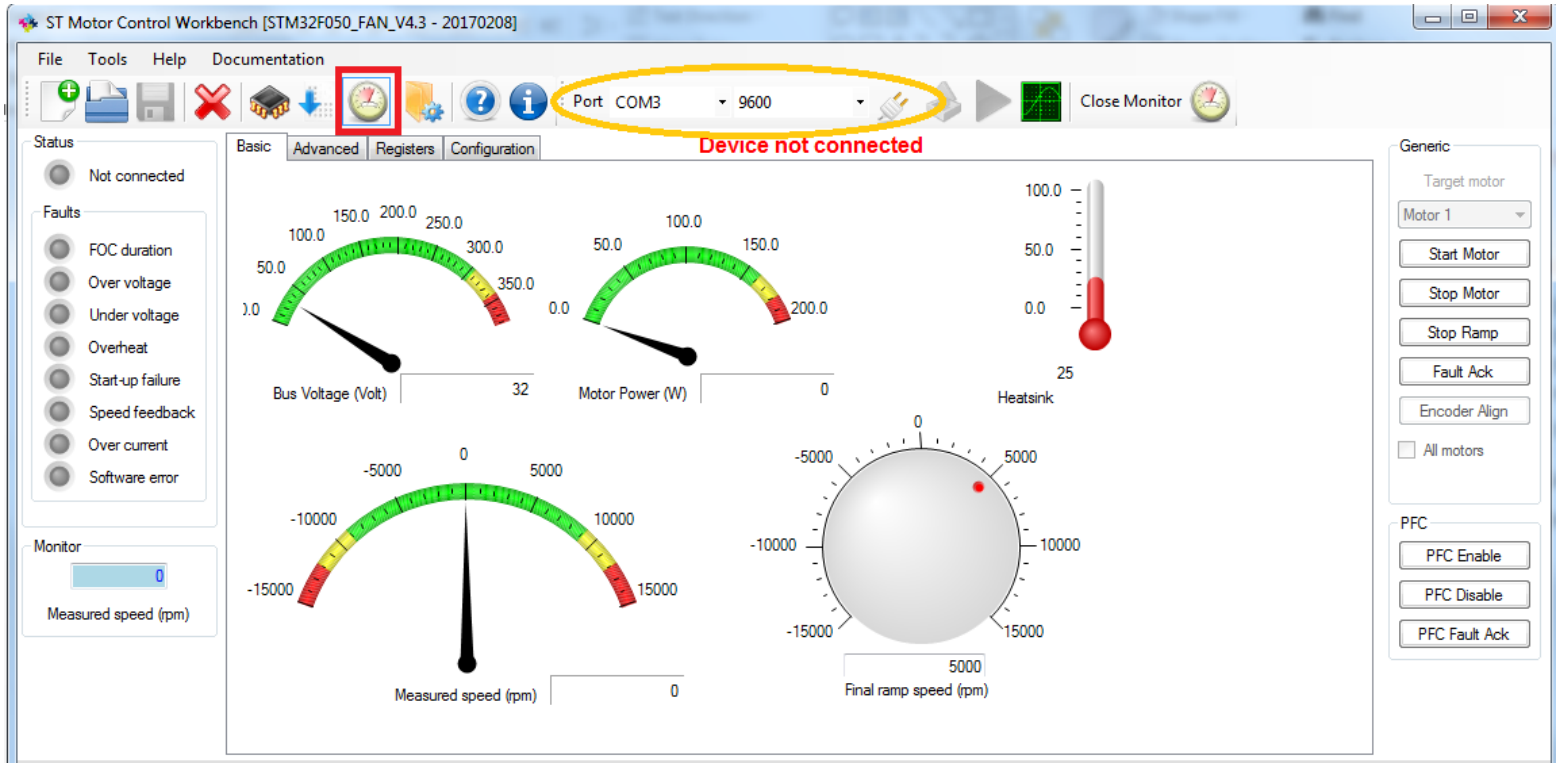


切入闭环条件判定

启动曲线

Workbench调试界面

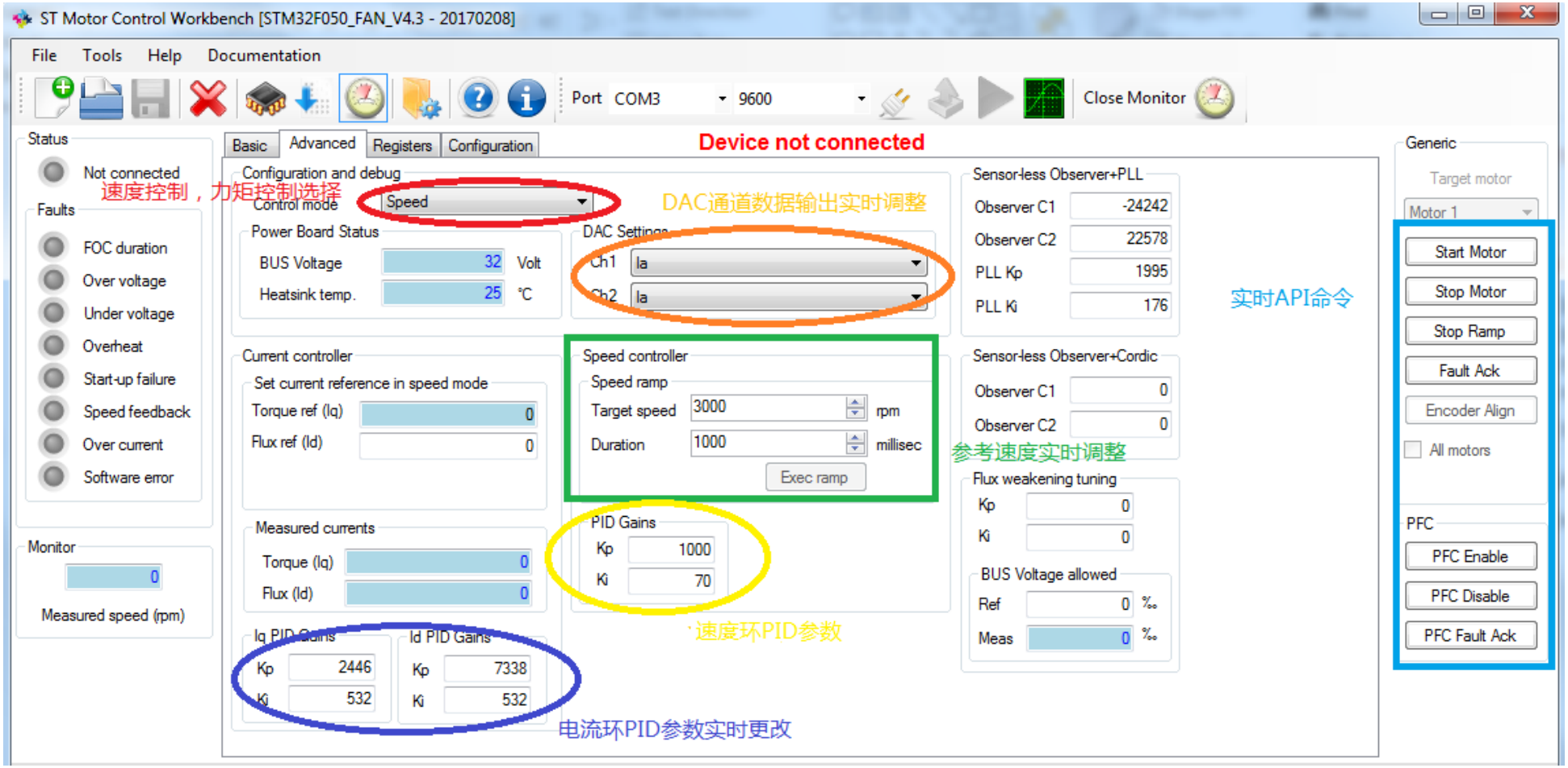
- 通过串口进行实时调试和监控马达运行



Workbench调试界面

➢ 提供实时数据修改界面

- 这边修改的RAM数据，如果对调试参数满意，可以更改前面参数配置数据，重新下载到Flash中。



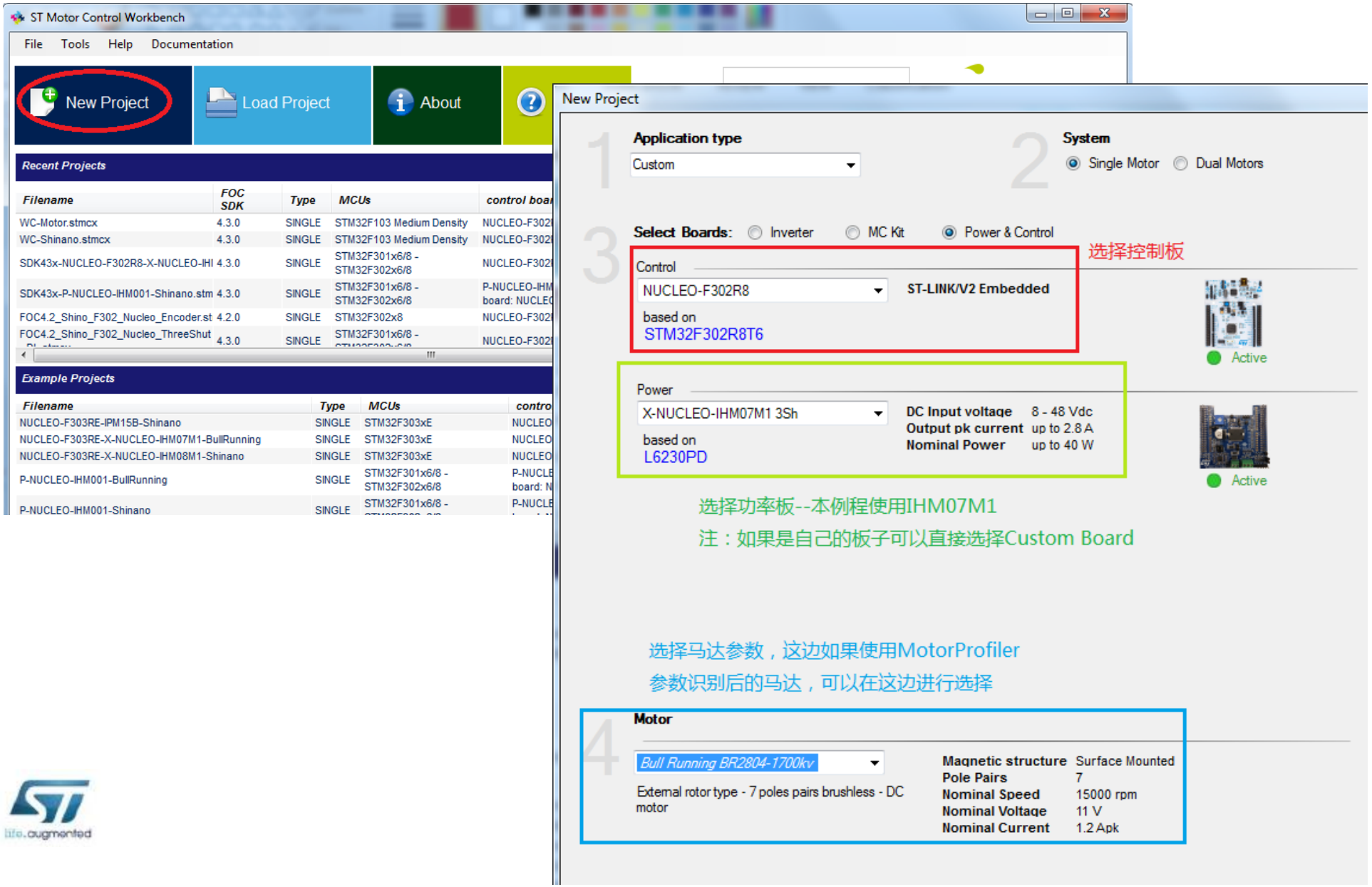
实际操作环节

开始使用MC Workbench，让电机转起来！

20~30 minutes

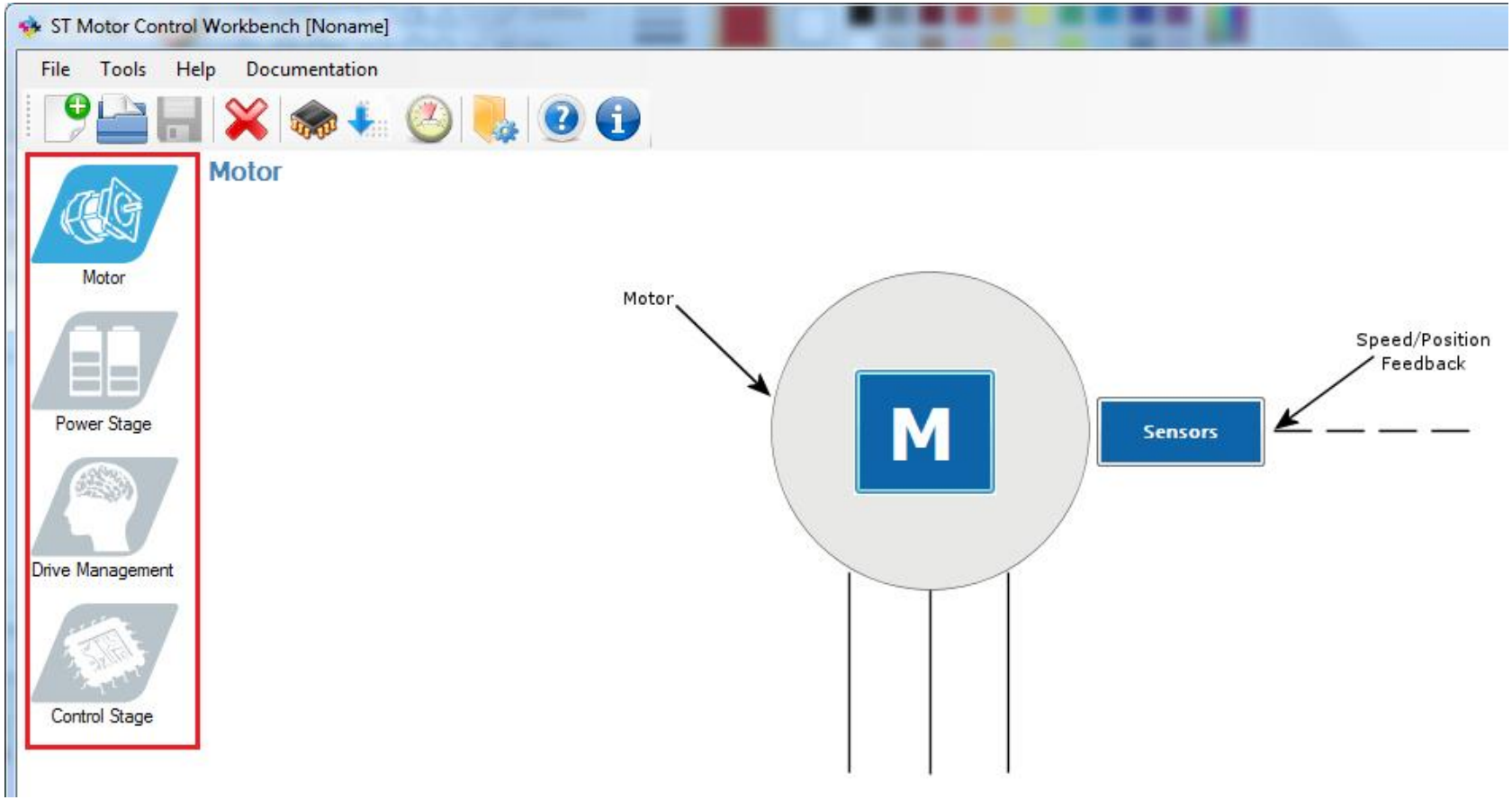
操作说明1/5

- 新建工程，选择控制板，功率板，马达参数
- 如果是自己板子，这边可以选择Custom Board



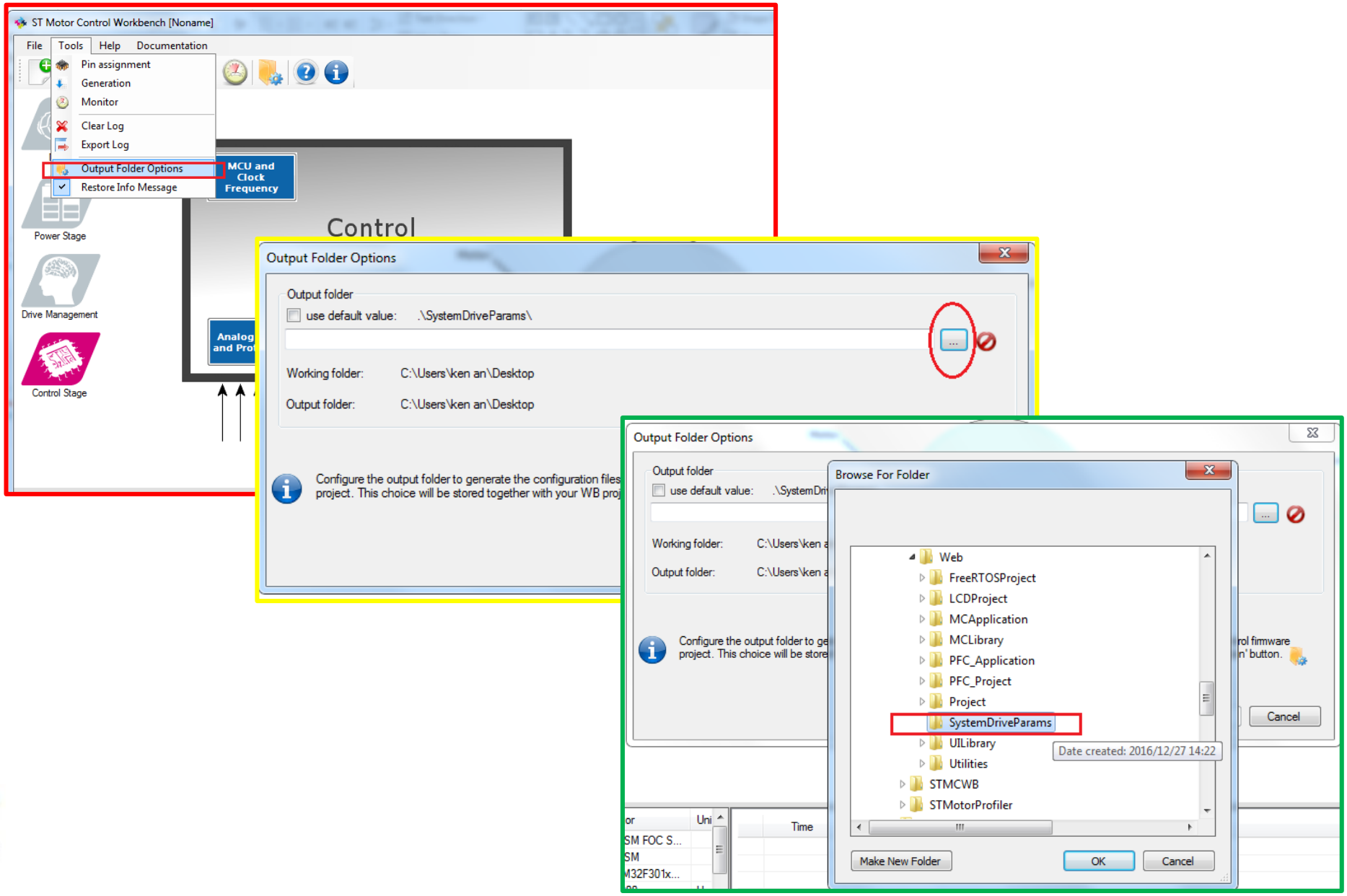
操作说明2/5

- 点击最左边的几个图标，察看参数配置情况，进行调整
- 分别是马达参数，功率板参数，驱动参数，控制板参数（MCU配置）



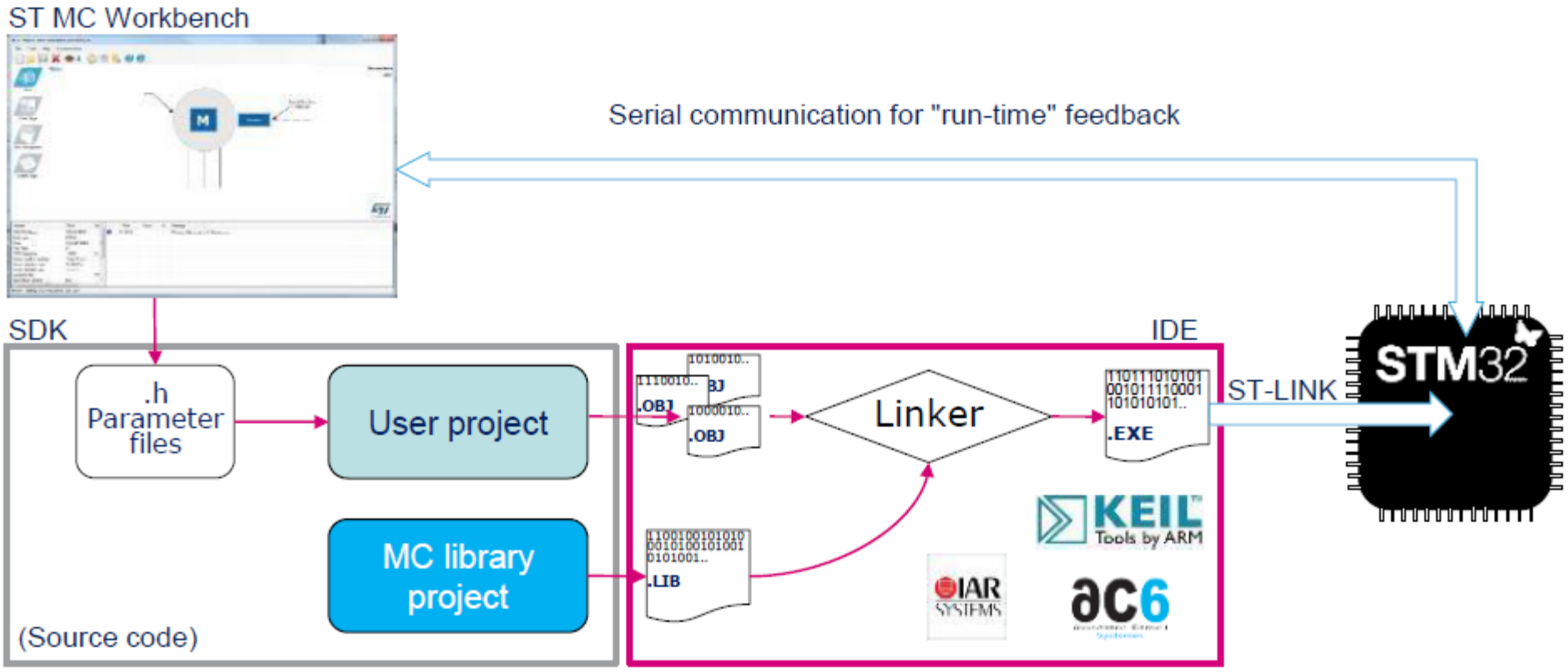
操作说明3/5

- 配置头文件保存的目录---需要放到库文件的\STM32 PMSM FOC LIB\Web\SystemDriveParams文件夹下



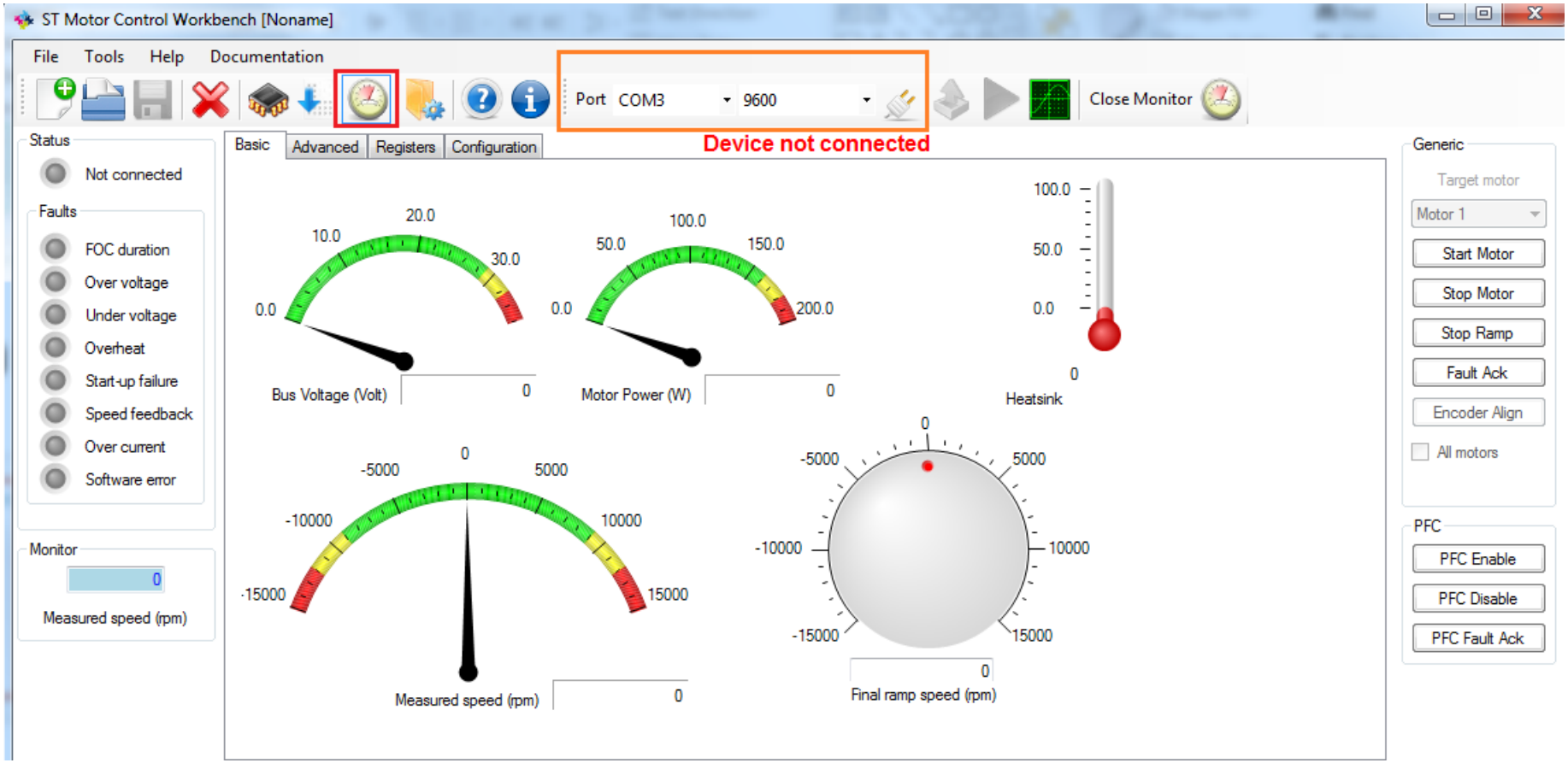
操作说明4/5

➢ 打开相应工程，编译下载到MCU中



操作说明5/5

- 切换到串口调试模式，开始使用GUI对马达进行控制

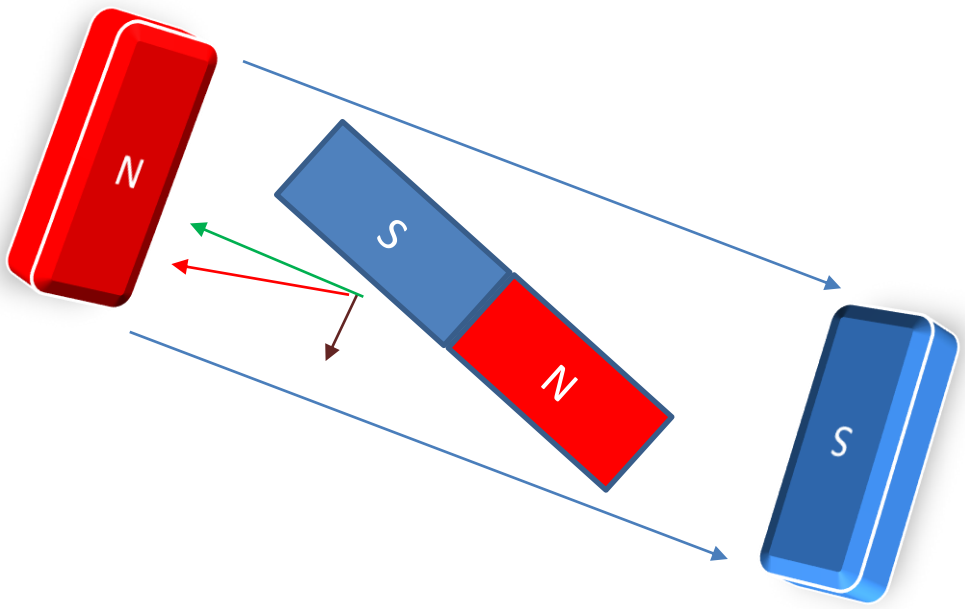


STM32 PMSM FOC
控制理论基础

life.augmented

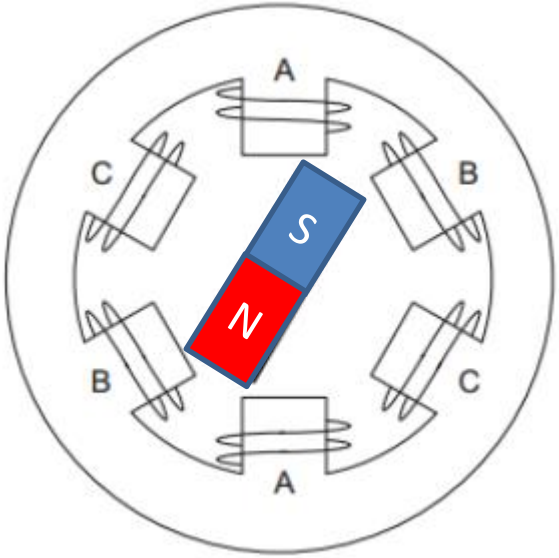
直流电机

- 从磁电角度
  - 通电线圈在磁场中受到力的作用
- 从磁场角度
  - 永磁铁（PM）与电磁铁（线圈）吸引力/排斥力工作
- 转动线圈的力来自于垂直于磁场方向的电流
- 平行于磁场的电流产生叠加磁场



PMSM电机

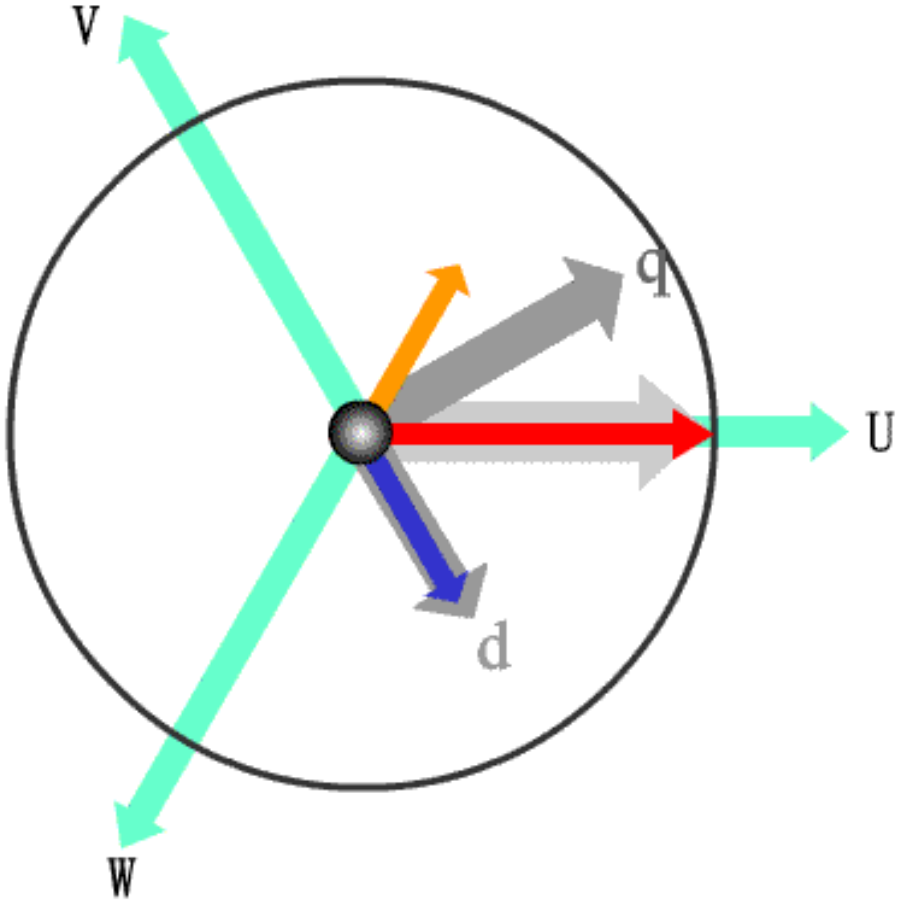
- 转子为永磁体
- 三相通电时电机定子磁场是三相电流产生磁场的叠加
- 涉及到三个磁场耦合，分析起来比较困难



life.augmented

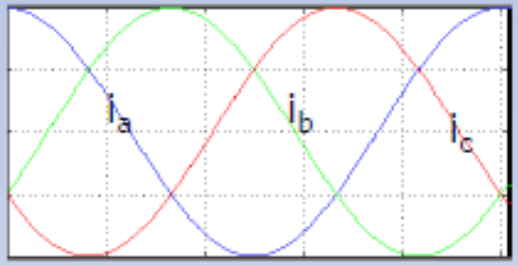
坐标变换

- 坐标变换实现三轴(Ia,Ib,Ic)到两轴(Iα,Iβ)
- 两轴(Iα,Iβ)到动轴(Id,Iq)的变换
- 完美解决三相解耦问题



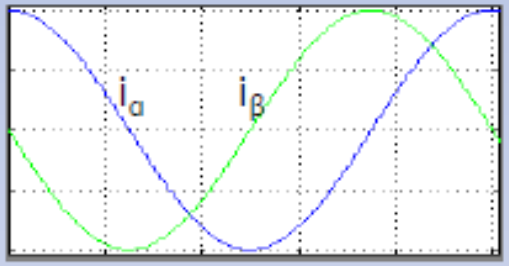
坐标变换理论

- Clarke变换， 三轴ia,ib,ic (120°) 到两轴 iα,iβ (90°)

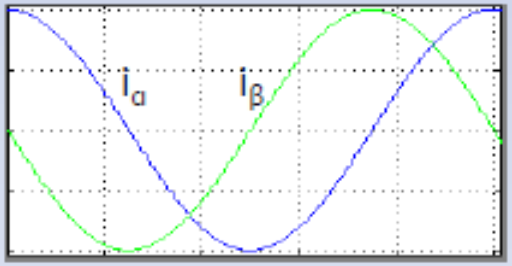


$$i_\alpha = i_{as}$$

$$i_\beta = -\frac{i_{as} + 2i_{bs}}{\sqrt{3}}$$

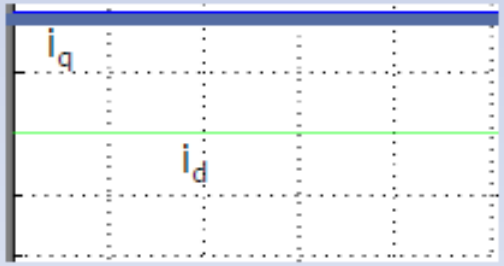


- Park变换， 两轴 iα,iβ (90°)到动轴iq,id (90°)

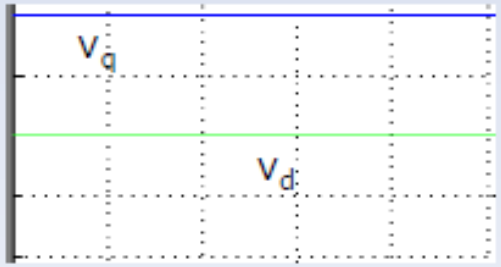


$$i_{qs} = i_\alpha \cos\theta_r - i_\beta \sin\theta_r$$

$$i_{ds} = i_a \sin\theta_r + i_\beta \cos\theta_r$$

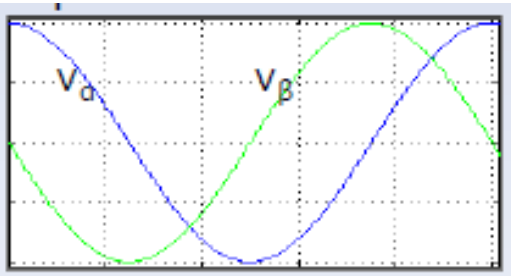


- 反Park变换， 得到Vα,Vβ



$$v_\alpha = v_{qs} \cos\theta_r + v_{ds} \sin\theta_r$$

$$v_\beta = -v_{qs} \sin\theta_r + v_{ds} \cos\theta_r$$



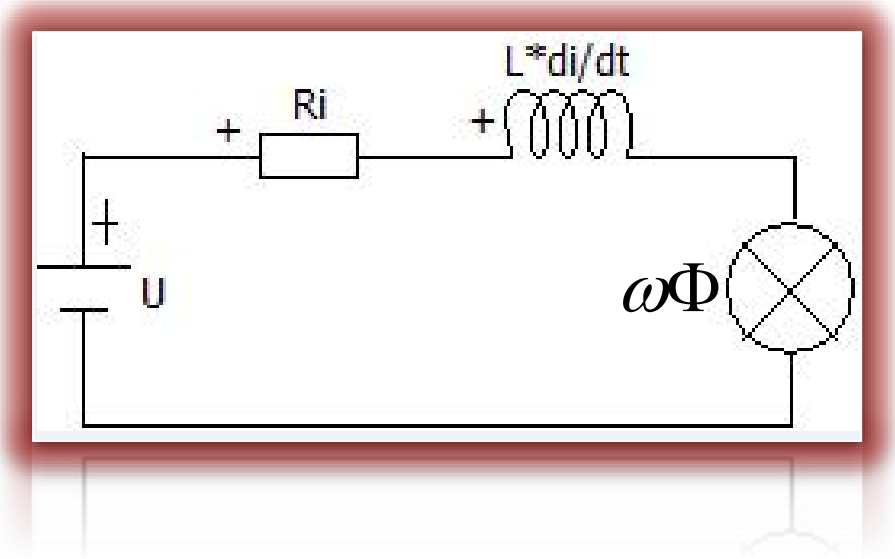
相关PMSM控制方程

- 磁链方程

$$\begin{cases} \Phi_d = L_d i_d + \Phi_f \text{(永磁体磁链)} \\ \Phi_q = L_q i_q \end{cases}$$

- PMSM定子电压方程

$$\begin{cases} U_d = R i_d + L_d \dfrac{d_i}{d_t} - \omega \Phi_q \\ U_q = R i_d + L_q \dfrac{d_i}{d_t} + \omega \Phi_d \end{cases}$$



相关PMSM控制方程

- 电磁转矩方程

$$T_e = \frac{3}{2} p(\Phi_d i_q - \Phi_q i_d) = \frac{3}{2} p(\Phi_f i_q + (L_d - L_q) i_d i_q)$$

- 运动平衡方程

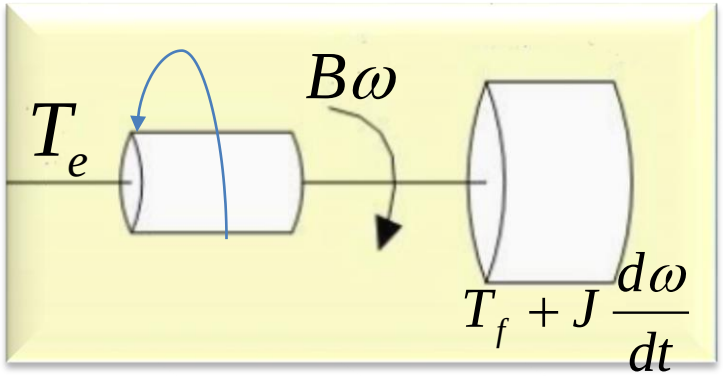
$$J \frac{d\omega}{dt} = T_e - T_f - B\omega$$

J - 转动惯量

ω - 机械角速度

T_f - 转子负载转矩

B - 阻尼系数



磁场定向控制 (FOC): 概述

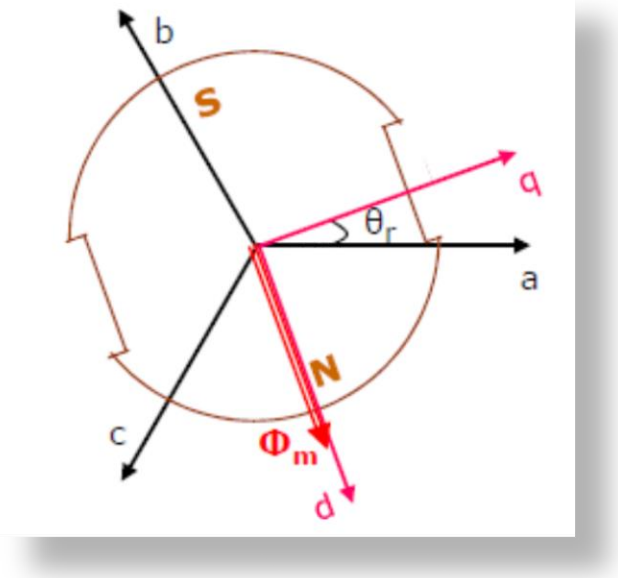
- 数学理论应用于三相马达磁链及转矩的解耦控制

- 定子电流被分解成:
  直轴电流Ids：用于产生磁场，与转子的磁场叠加
  交轴电流Iqs：用于控制转矩（其作用等同于直流电机的电枢电流）。

- 优点:
  - 电流相位与转子位置同步，有利于提高马达效率
  - 电流的励磁和力矩在解耦后可分别直接控制，马达转速可对负载的变化做出精确而快速的反应
  - 可进行位置控制（通过瞬时转矩控制）
  - 由于相电流为正弦波，可降低马达的运行噪音

矢量控制原理

- 通过电流解耦后得到的正交电流Id，Iq
- 和直流电机相似，固定Id后，可通过控制Iq可控制转矩
  - 对于表贴电机，Ld = Lq；另Id* = 0，转矩达到最大
  - 对于凸极电机，Ld < Lq; 也可以固定Id*进行控制
- 速度控制的实质是转矩Te控制
  - 根据运动平衡方程，负载固定，转矩和速度成正比
- 矢量控制实际上是对定子电流矢量相位和幅值的控制



矢量控制 – 电压极限

- 电机稳定运行时，电压矢量的幅值

$$V_s = \sqrt{V_d^{\,2} + V_q^{\,2}} \le V_{\lim}$$

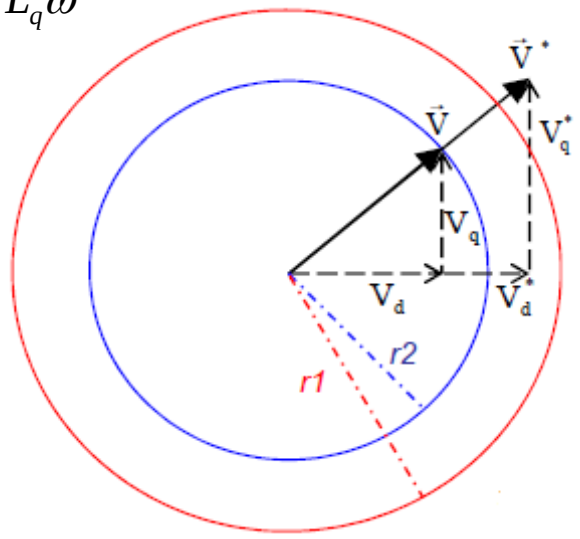
- 带入电压方程，设定稳定运行，忽略定子电阻压降下，得到电压极限方程

$$(L_q i_q)^2 + (L_d i_d + \Phi_f)^2 = (V_{\lim} / \omega)^2$$

$$V_{\lim} = \frac{V_{dc}}{\sqrt{3}}$$

$Ld = Lq$时，圆心在$(-\Phi_f / L_q, 0)$，半径为$\frac{V_{\lim}}{L_q \omega}$的圆

$Ld \neq Lq$时，为椭圆方程



矢量控制 – 电流极限

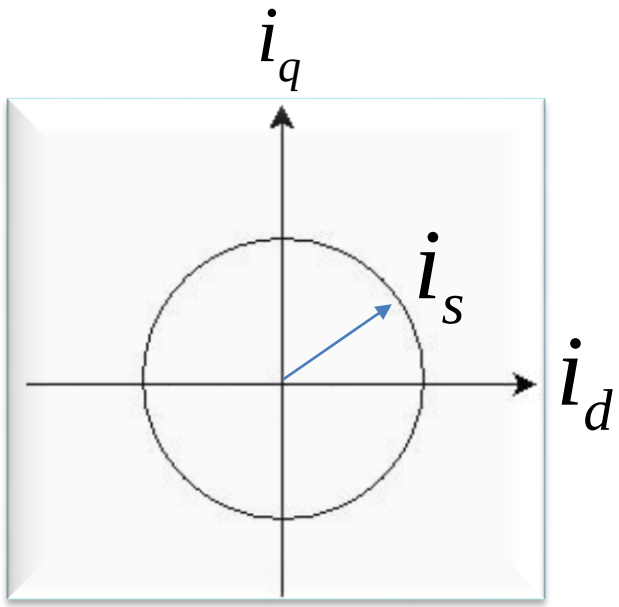
- 电机稳定运行时，电流矢量的幅值

$$i_s = \sqrt{i_d^{\,2} + i_q^{\,2}} \le i_{\lim}$$

$i_{\lim} = \sqrt{3} I_{\lim}$

$I_{\lim}$为电机最大相电流的有效值

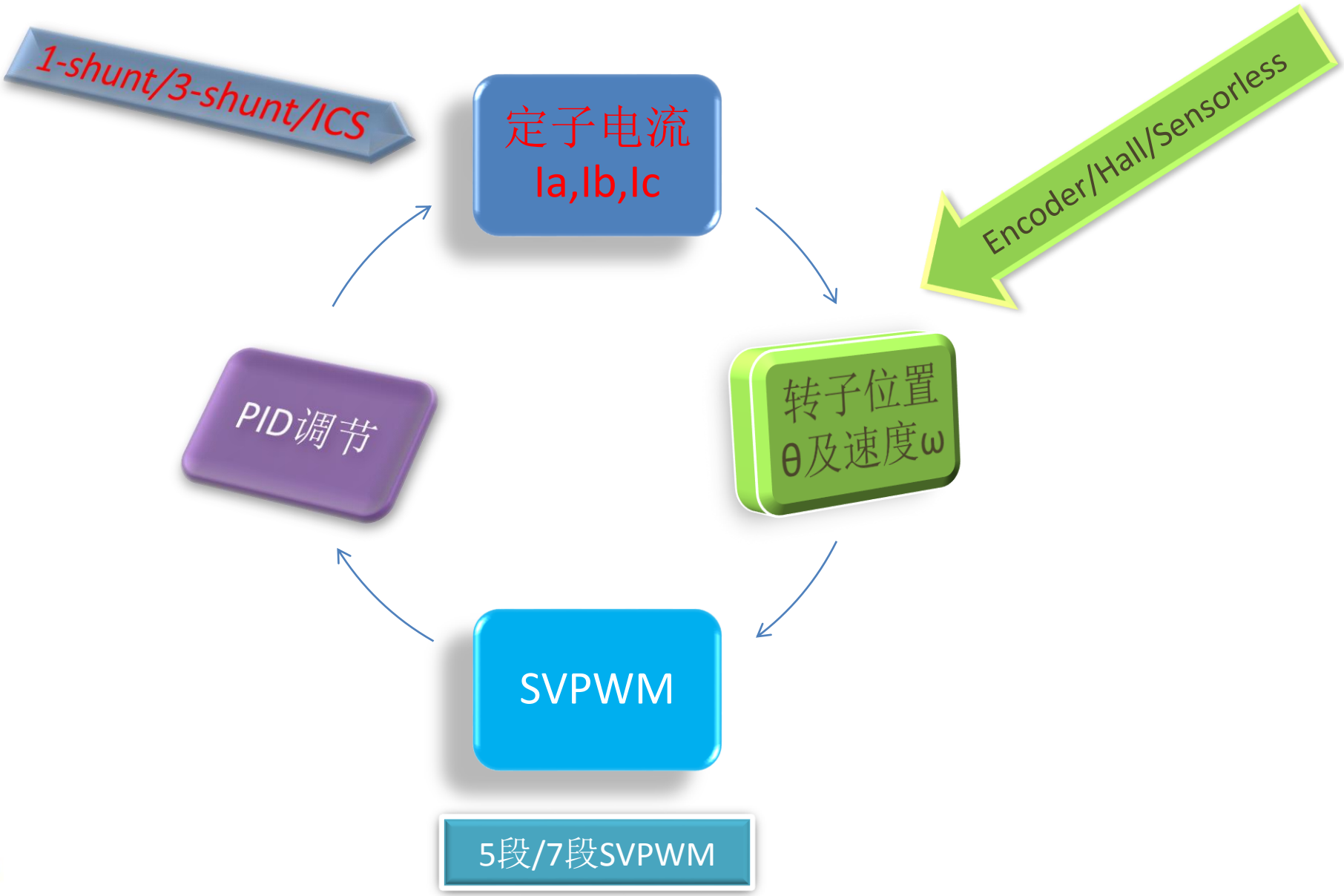
- 电机稳定运行时，电流矢量为一个圆形轨迹

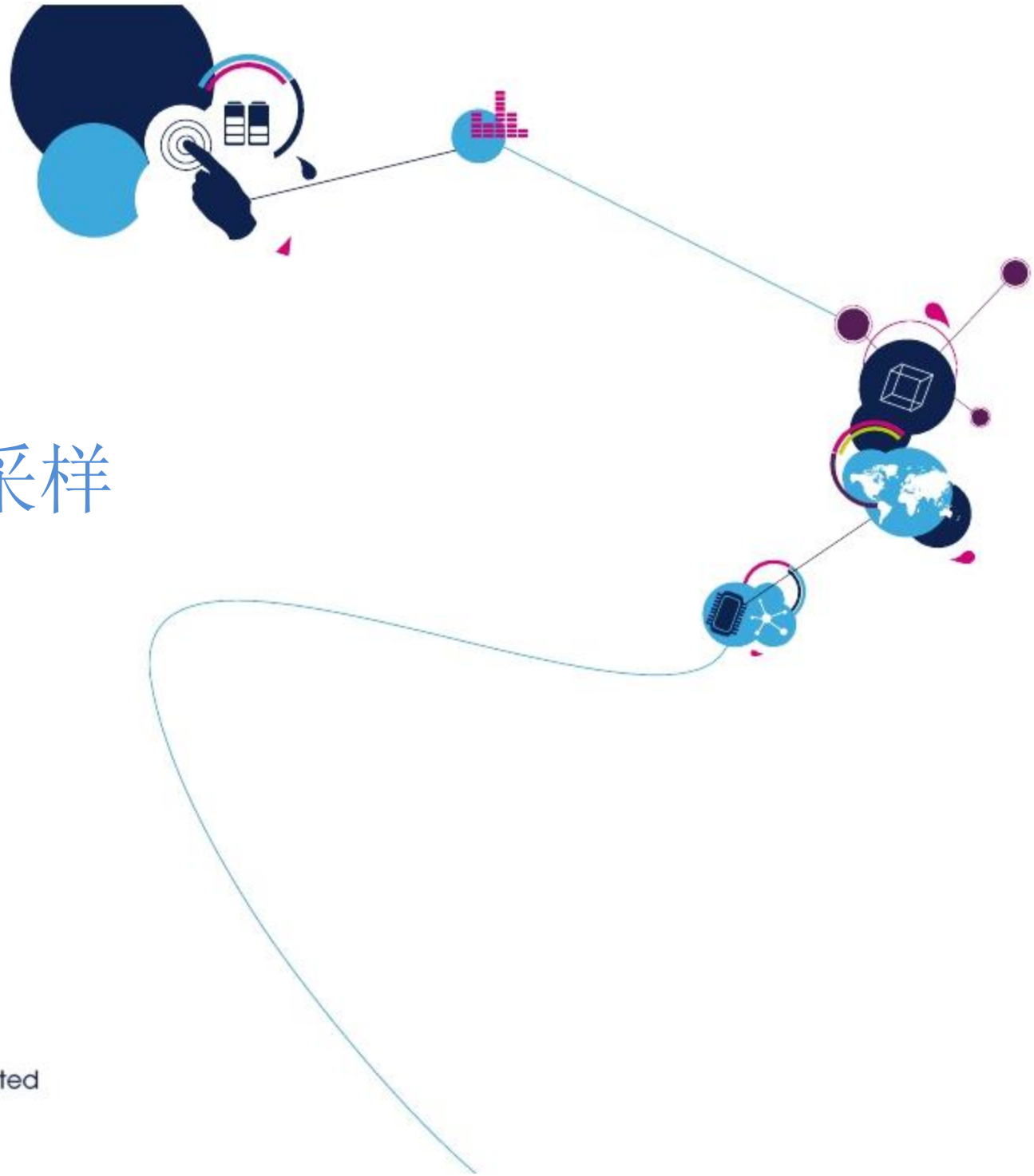


实现FOC控制的具体操作

life.augmented

构成FOC控制要素


1-shunt/3-shunt/ICS
定子电流
Ia,Ib,Ic
Encoder/Hall/Sensorless
转子位置
θ及速度ω
SVPWM
PID调节
5段/7段SVPWM

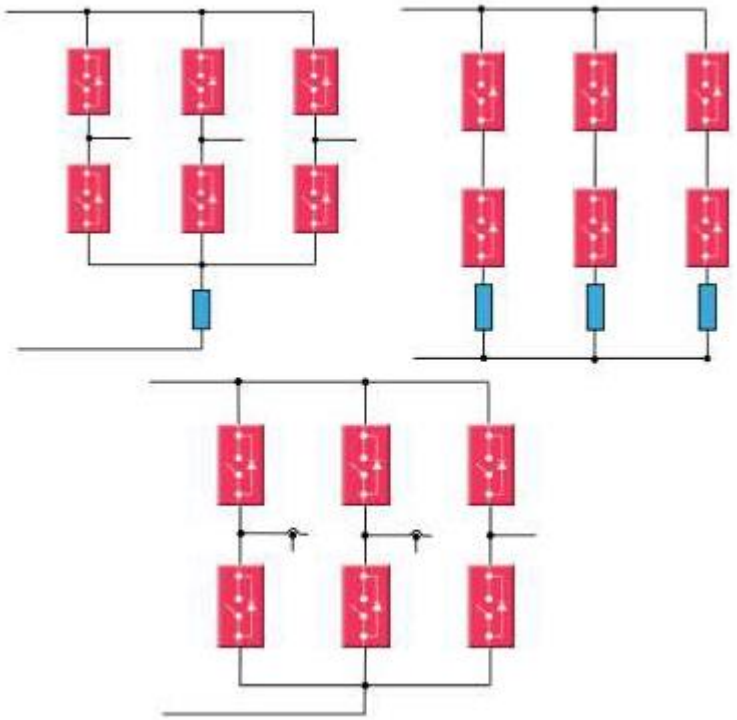


定子电流采样

life.augmented

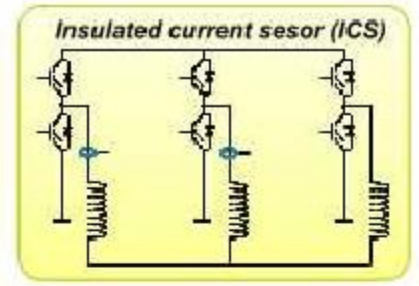
定子电流采样 – 采样方式

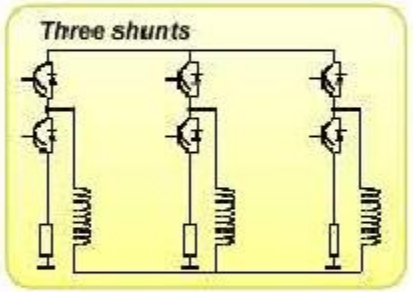
- 1-SHUNT：采样电阻放在DC BUS上
  - ST专利的算法
  - 仅需要1个电阻/运放：成本较低
  - 电流采样算法可能会带来力矩纹波
- 3-SHUNT：采样电阻放在3个下桥臂上
  - 电流采样精度高
  - 不会有电流纹波产生
- ICS：2个隔离的电流传感器
  - 放在A/B相绕组上
  - 适用于相电流较大的场合：无功耗
  - 成本较高

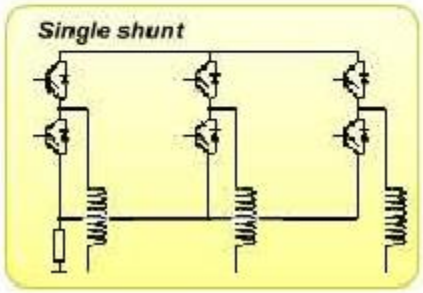


定子电流采样 - Workbench的设定

- 相电流采样拓扑结构选择：在workbench中的设定

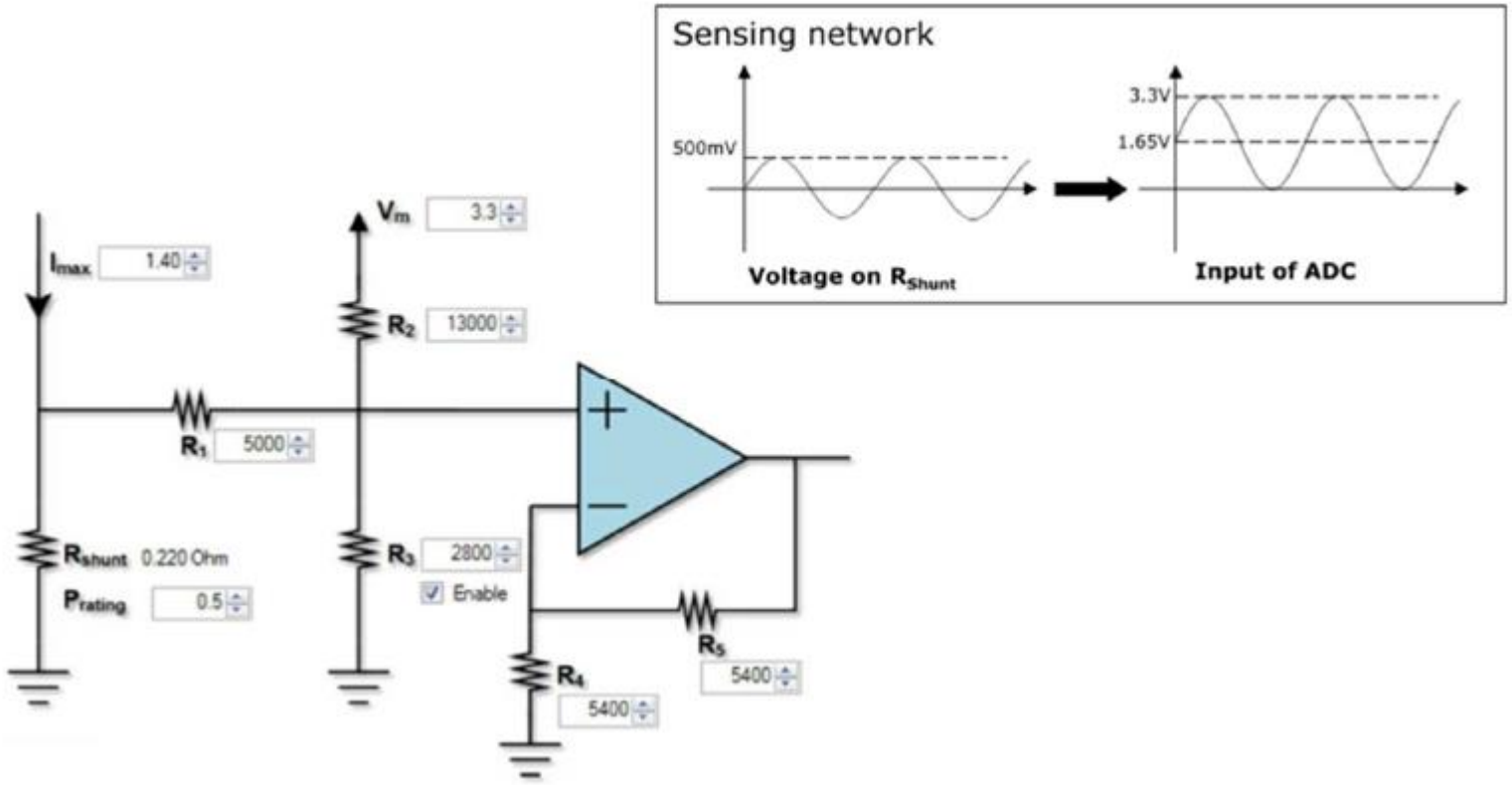






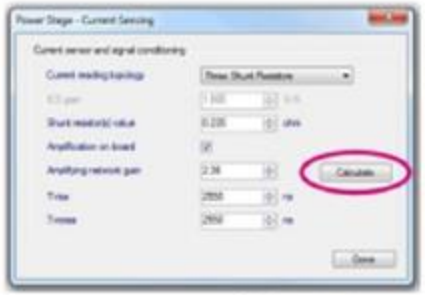
定子电流采样 – 偏置放大电路

- 基于shunt的采样方式要求：运放电路 – 电流信号的放大及偏置电压



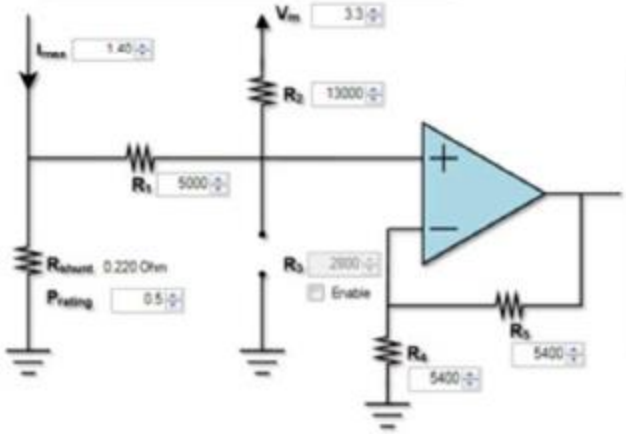
定子电流采样 – 运放外围电路配置

- 必须合理配置运放电路的增益及偏置电压
- Workbench中专门设计了一个放大电路的设计工具



$$V_o = \left(1+\frac{R_5}{R_4}\right)V_M\frac{R_1 \parallel R_3}{R_2+(R_1 \parallel R_3)} + \left(1+\frac{R_5}{R_4}\right)\frac{R_2 \parallel R_3}{R_1+(R_2 \parallel R_3)}R_S I$$

$$V_o = V_{offset} + A_{OP}R_S I$$



Parameters

Name	Value	Unit	Description
Imax	1.40	A	Desired maximu...
Vm	3.3	V	Reference volta...
R-shunt	0.220	Ohm	Shunt resistor va...
P-rating	0.5	W	Shunt resistor m...
R1	5000	Ohm	Resistance of th...
R2	13000	Ohm	Resistance of th...
R4	5400	Ohm	"Non-inverting" ...
R5	5400	Ohm	"Non-inverting" ...
Vmcu	3.3	V	MCU voltage

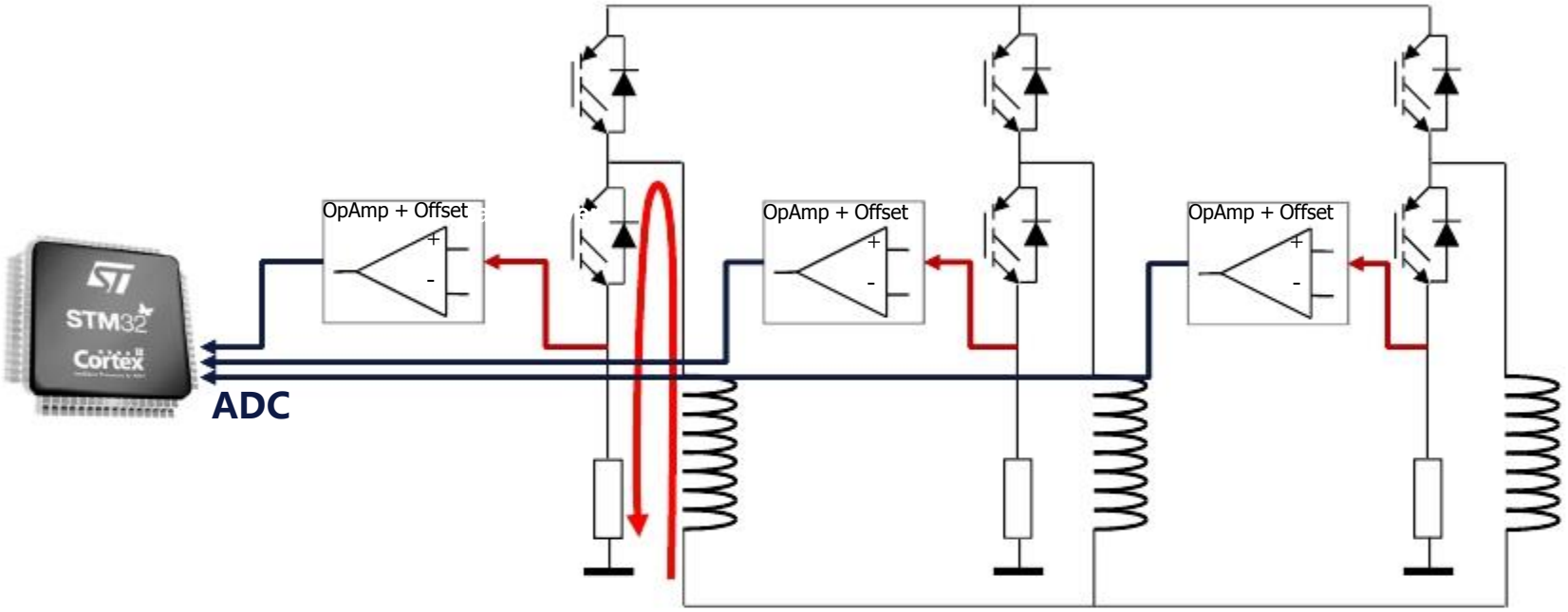
Result

Name	Value	Unit	Description
Overall gain	1.444		Overall gain of t...
Offset network attenuation	0.722		Signal attenuatio...
Op-amp gain	2.000		Gain of the "No...
Vout (polarization)	1.833	V	Opamp output v...
Min Vout	1.389	V	Opamp output v...
Max Vout	2.278	V	Opamp output v...
Power dissipated on shunt	0.108	W	Power dissipate...

life.augmented

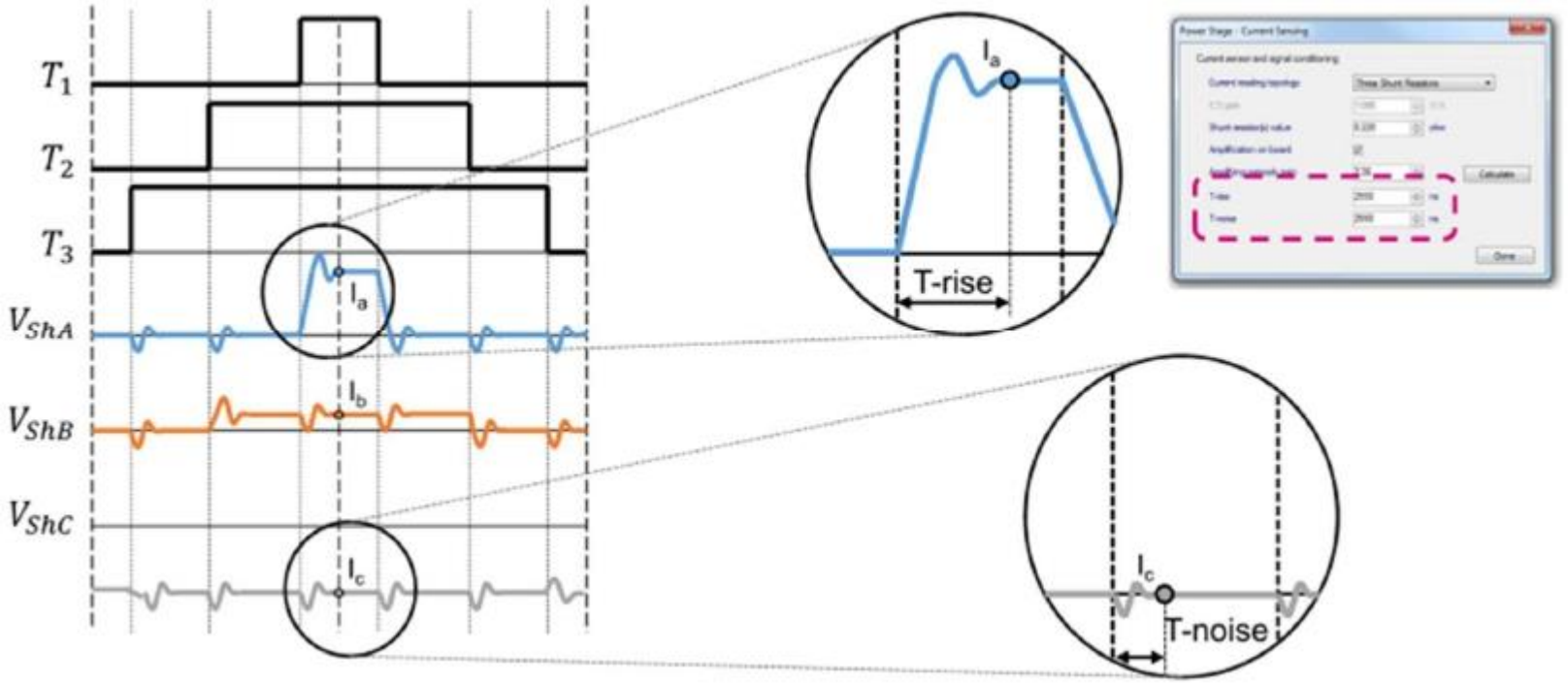
定子电流采样 – 3shunt采样

- 3-shunt：必须在下桥臂打开时才能采样到相电流
- 同时只需要采样两路电流信号，利用Ia+Ib+Ic = 0得到第三路电流

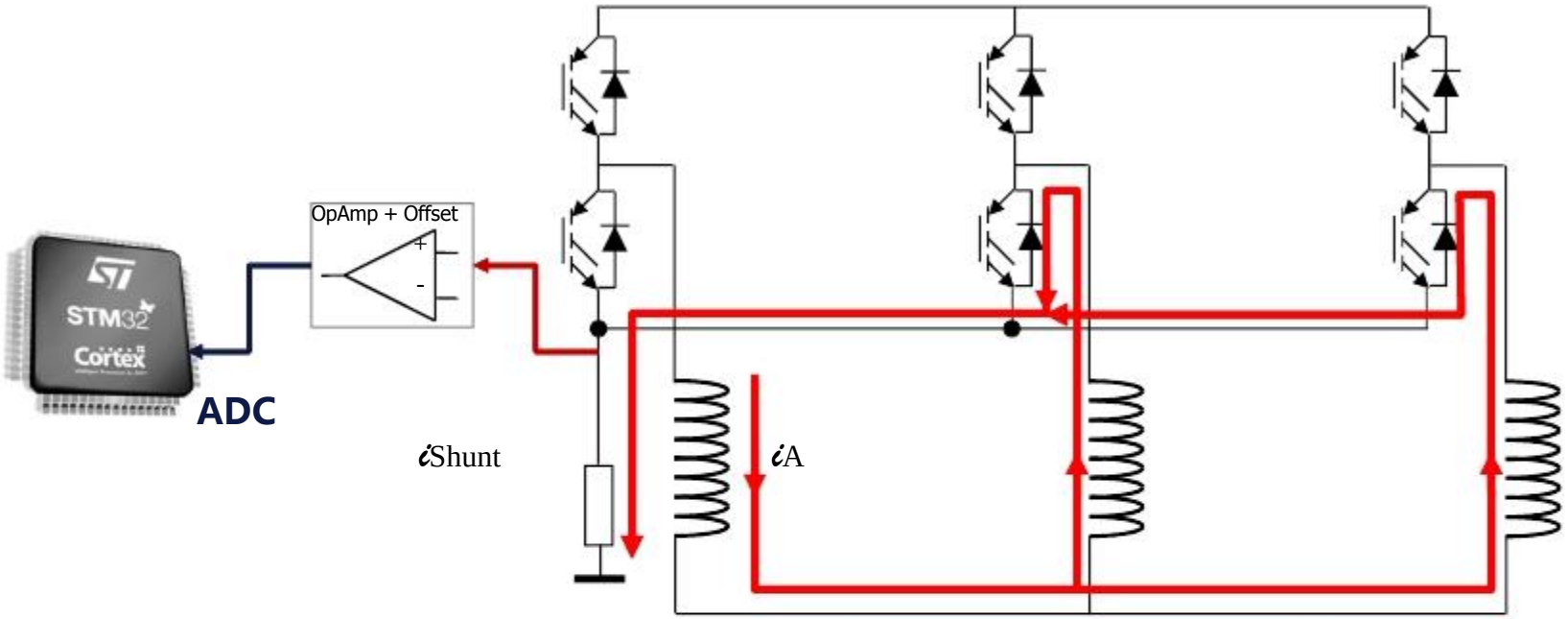


定子电流采样 – ADC采样点选择

- 得益于STM32 灵活的ADC采样硬件触发机制，采样点可以在波形上任意平移
- T_{rise}：ADC通道对应的下桥臂打开时刻到采样信号稳态之间的时间
- T_{noise}：其它桥臂开关时在ADC通道引起的纹波时间



定子电流采样 – 1shunt采样

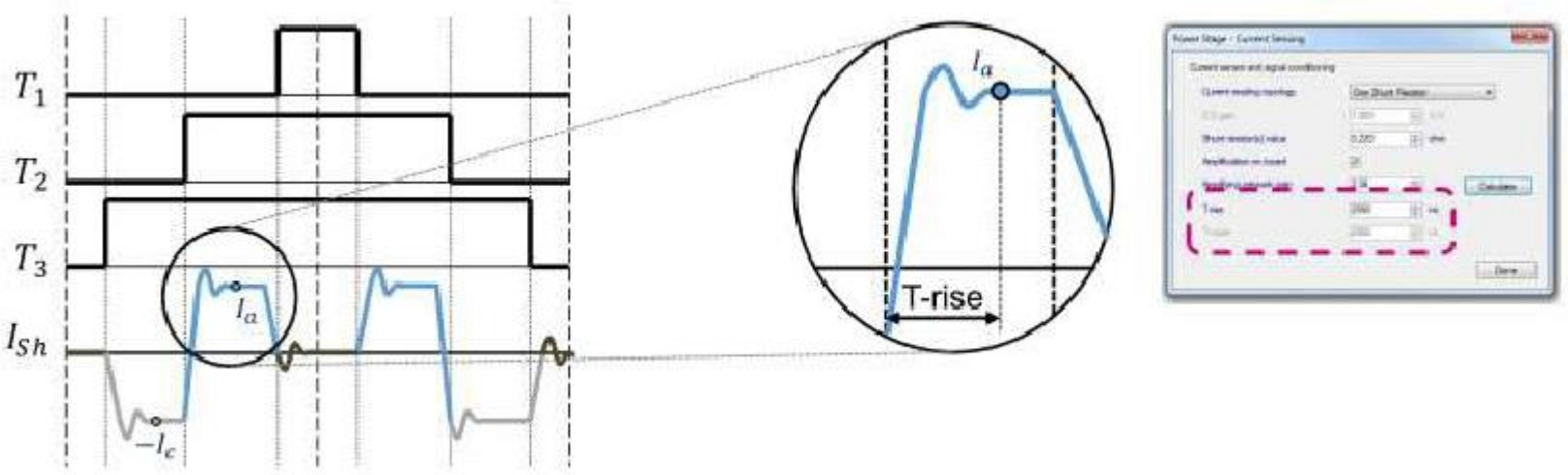


- 下桥臂的不同状态下，流过shunt的电流如右表

AL	BL	CL	i_{Shunt}
Open	Open	Open	0
Open	Close	Close	i_A
Open	Open	Close	$-i_C$
Close	Open	Close	i_B
Close	Open	Open	$-i_A$
Close	Close	Open	i_C
Open	Close	Open	$-i_B$
Close	Close	Close	0

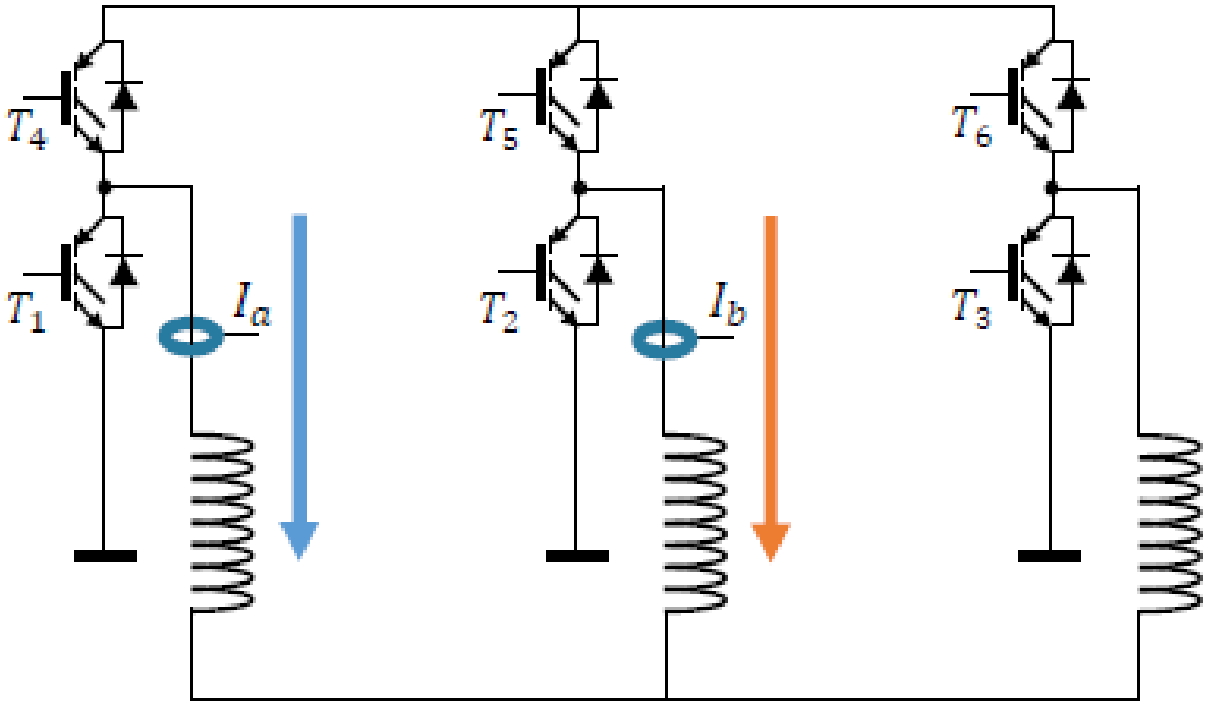
定子电流采样 – ADC采样点选择

- 在一个周期或者两个周期内ADC需要被触发两次进行采样
- Trise：ADC通道对应的下桥臂打开时刻到采样信号稳态之间的时间
- Tnoise：无意义



定子电流采样 - ICS

- 电流传感器串接在相线上
- 在PWM波形中间进行Ia，Ib采样
- Ic电流通过Ia+Ib+Ic=0进行计算



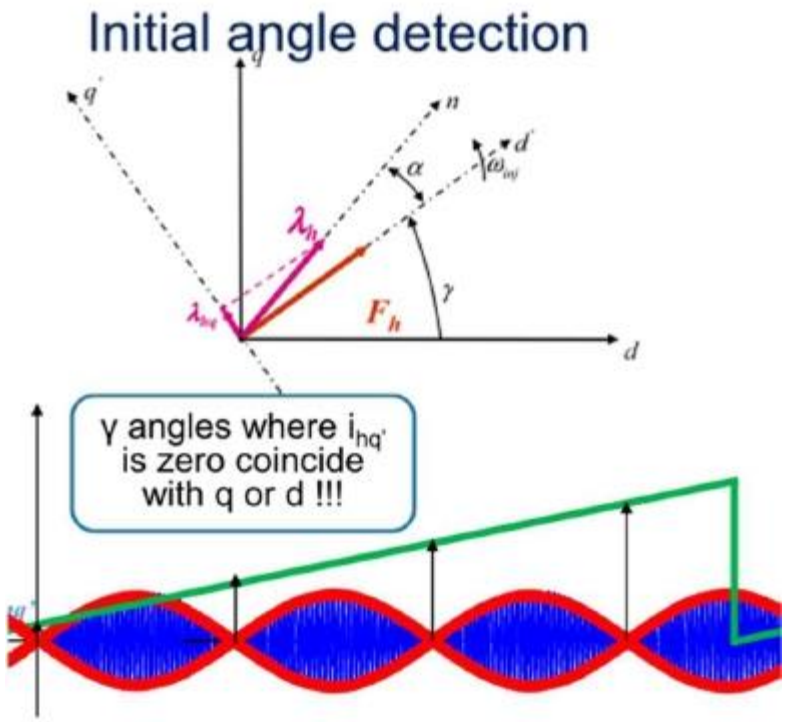
转子位置以及速度检测

life.augmented

转子位置及速度检测

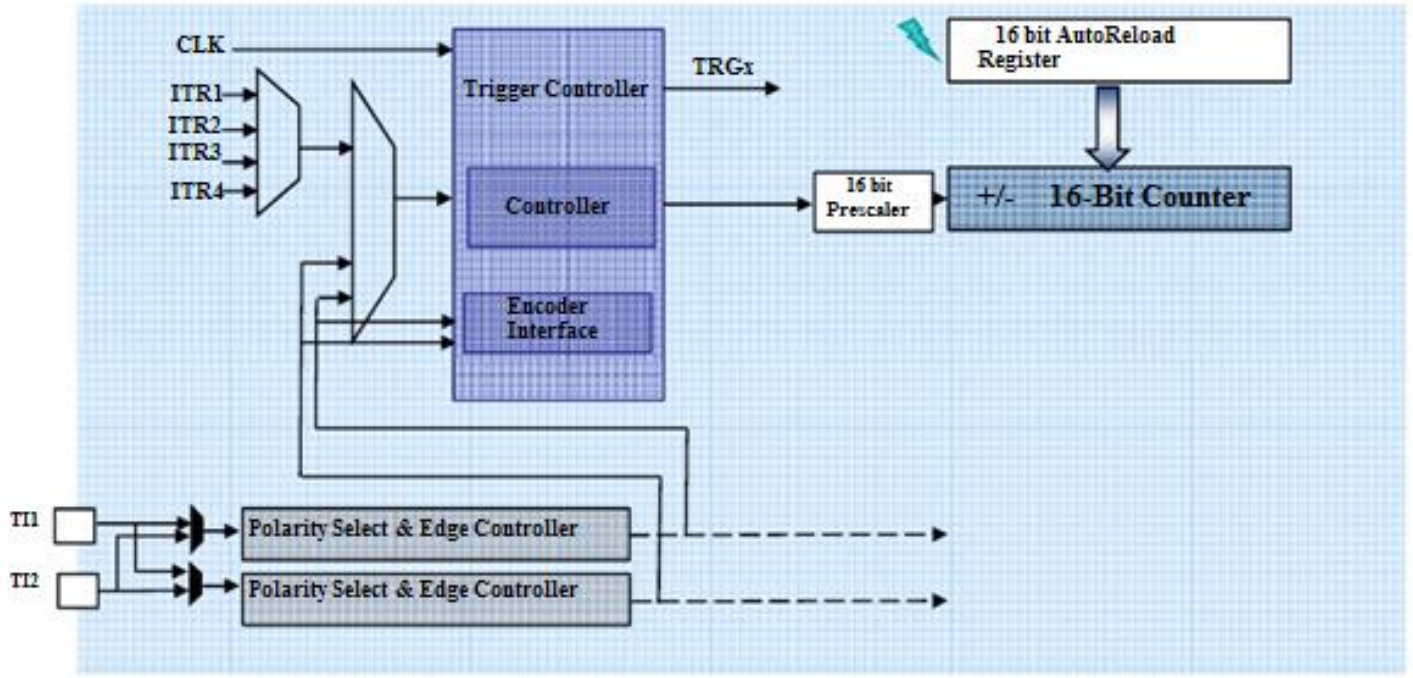
➢ 速度及位置检测支持：

- Encoder
  - 成本较高，一般适用于伺服控制
- Hall
  - 成本较低，一般适用于马达静止或低速下也要求额定扭矩的应用
- Sensorless
  - 高频注入算法 – HFI
    - 适用于凸极马达（IPMSM，Ld<Lq）
    - 能实现马达转子位置的精确检测，即使在静止或低速下
    - 仅STM32F3和STM32F4系列支持
  - State Observer + PLL
    - 基于马达的BEMF，使用相电流及相电压估计马达转子的位置
    - 适用于马达的转速范围：额定转速的5% - 100%
  - State Observer + CORDIC



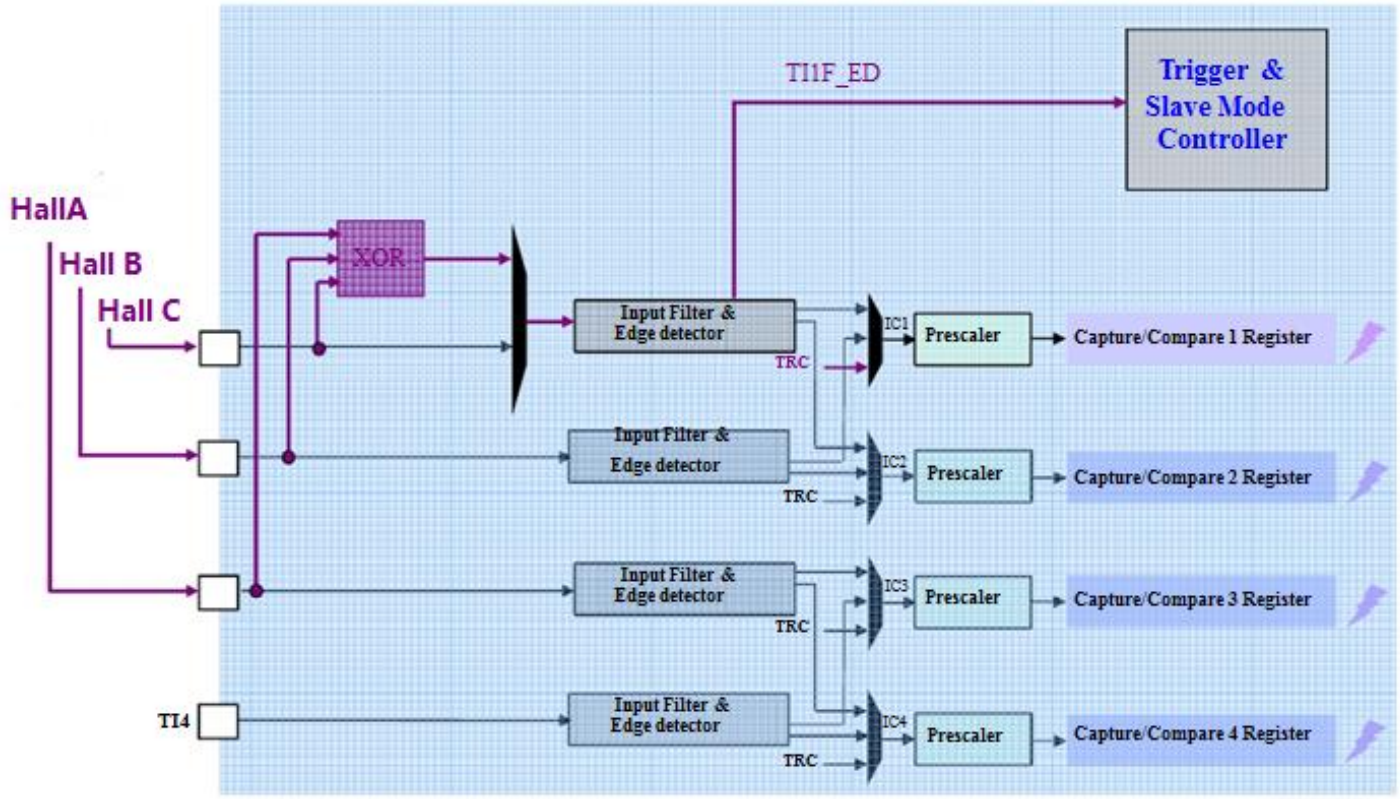
Encoder接口

- STM32提供编码器A，B信号接口
- 无Z信号接口，需要使用外部中断口连接Z信号
- 读取方向以及脉冲数据
- 运转前需要进行定位操作
- 绝对编码器目前电机库未支持，可以根据实际需要添加代码



Hall信号接口

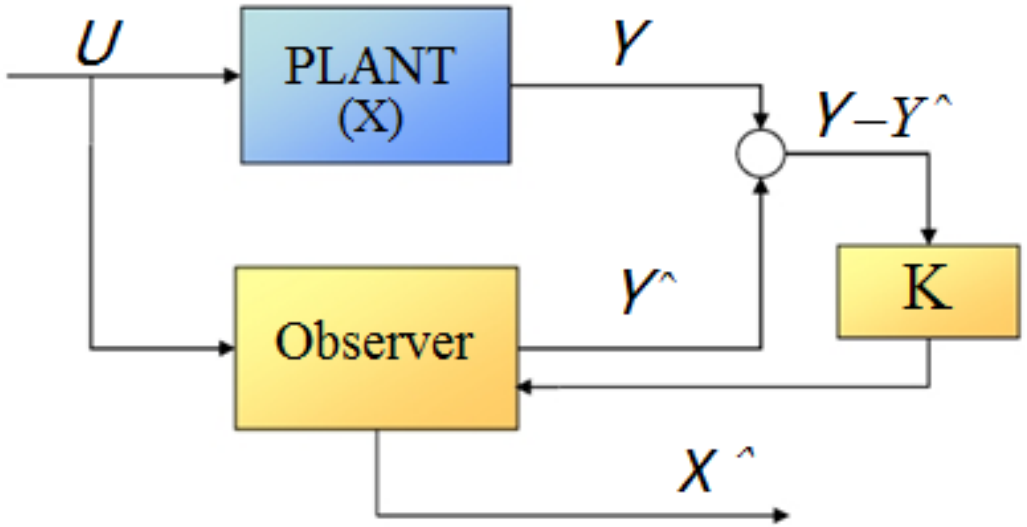
- STM32提供Hall信号接口，每个信号跳变沿都可探测
- 电机库动态调整分频得到准确速度数据
- 每60度电角度同步一次角度数据



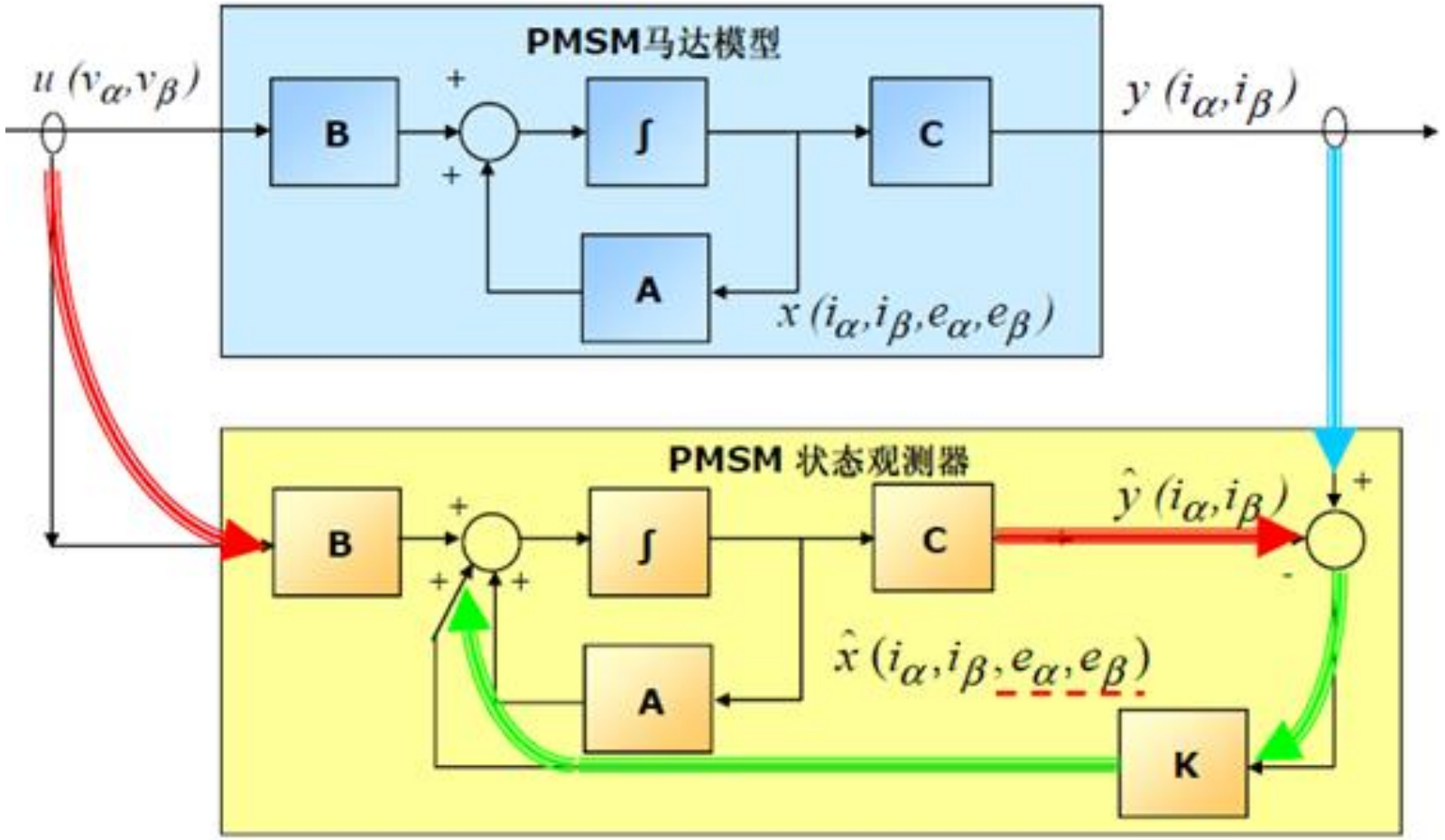
无传感观测器

- 根据控制理论，如果一个系统能够完全通过其检测到的输出值来重构其系统状态，则认为该系统是可观测的；

- 状态观测器根据所观测系统的输入及输出值估计其内部状态。



Luenberger观测器 1/2



Luenberger观测器 2/2

状态模型则表示为：

PMSM 马达模型

$$\begin{cases} \dot{x}(t) = Ax(t) + Bu(t) \\ y(t) = Cx(t) \end{cases}$$

$$\begin{cases} \dfrac{di_\alpha}{dt} = -\dfrac{r_s i_\alpha}{L_s} - \dfrac{e_\alpha}{L_s} + \dfrac{v_\alpha}{L_s} \\ \dfrac{di_\beta}{dt} = -\dfrac{r_s i_\beta}{L_s} - \dfrac{e_\beta}{L_s} + \dfrac{v_\beta}{L_s} \\ \dfrac{de_\alpha}{dt} = p\omega_r e_\beta \\ \dfrac{de_\beta}{dt} = -p\omega_r e_\alpha \end{cases}$$

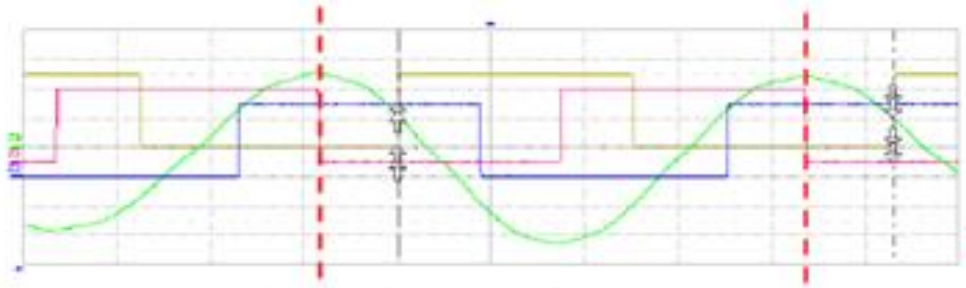
PMSM 状态观测器

$$\begin{cases} \dot{\hat{x}}(t) = A\hat{x}(t) + Bu(t) + K(y - \hat{y}) \\ \hat{y}(t) = C\hat{x}(t) \end{cases}$$

$$\begin{cases} \dfrac{d\hat{i}_\alpha}{dt} = -\dfrac{r_s \hat{i}_\alpha}{L_s} - \dfrac{\hat{e}_\alpha}{L_s} + \dfrac{v_\alpha}{L_s} + K_1(\hat{i}_\alpha - i_\alpha) \\ \dfrac{d\hat{i}_\beta}{dt} = -\dfrac{r_s \hat{i}_\beta}{L_s} - \dfrac{\hat{e}_\beta}{L_s} + \dfrac{v_\beta}{L_s} + K_1(\hat{i}_\beta - i_\beta) \\ \dfrac{d\hat{e}_\alpha}{dt} = p\omega_r \hat{e}_\beta + K_2(\hat{i}_\alpha - i_\alpha) \\ \dfrac{d\hat{e}_\beta}{dt} = -p\omega_r \hat{e}_\alpha + K_2(\hat{i}_\beta - i_\beta) \end{cases}$$

life.augmented

状态信号捕捉



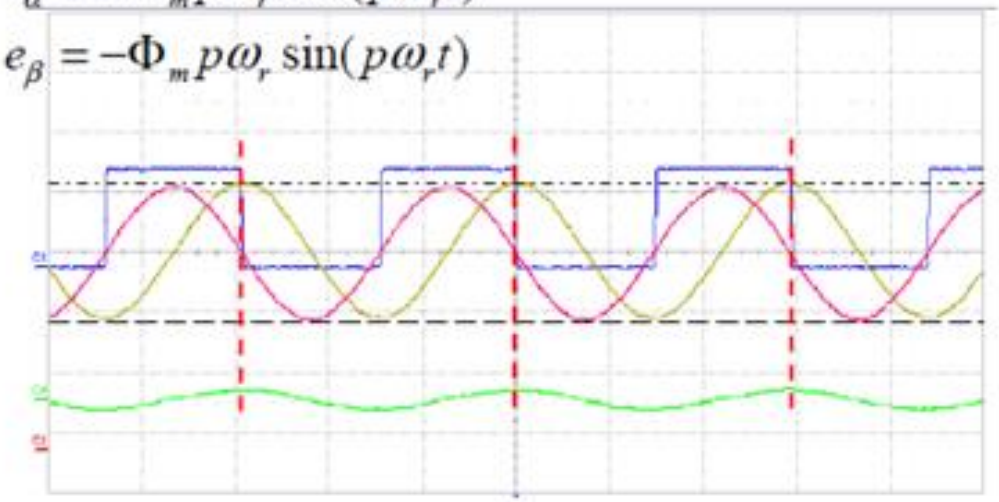
C1 C2 C3: Hall sensors
C4: back-emf phase A-0

正方向手动转动马达

$$v_\alpha = r_s i_\alpha + \frac{d(L_s i_\alpha + \Phi_m \sin\theta_r)}{dt} \longrightarrow v_\alpha = e_\alpha = \Phi_m \frac{d\theta_r}{dt}\cos\theta_r$$

$$e_\alpha = +\Phi_m p\omega_r \cos(p\omega_r t)$$

$$e_\beta = -\Phi_m p\omega_r \sin(p\omega_r t)$$



C1: observed e_α
C2: observed e_β
C3: Hall sensor C2
C4: phase current i_a

转子位置/角速度计算 1/2

- 在得到反电动势的alpha及Beta分量后，可从中解析出转子的位置角；
- 由 e_α 及 e_β 的定义：

$$e_\alpha = \Phi_m p\omega_r \cos(p\omega_r t)$$

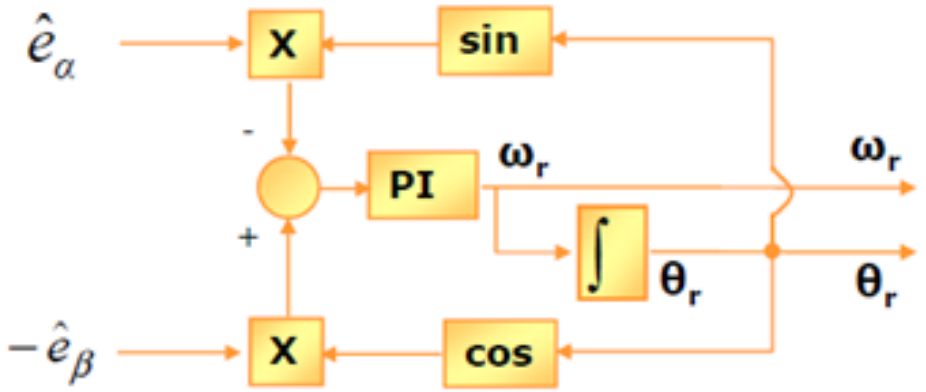
$$e_\beta = -\Phi_m p\omega_r \sin(p\omega_r t)$$

两者相除，可得到转子的位置角为：

$$\theta_r = p\omega_r t = arctg\left(-\frac{\hat{e}_\beta}{\hat{e}_\alpha}\right)$$

这个方法是开环的，它对耦合在反电动势里的干扰非常敏感(它会使反电动势成为非正弦信号！)

转子位置/角速度计算 2/2 锁相环PLL



令 $A = \Phi_m p \omega_r$，则：

$$\hat{e}_\alpha(kT) = A\cos(\hat{\theta}_k) \qquad -\hat{e}_\beta(kT) = A\sin(\hat{\theta}_k)$$

那么：

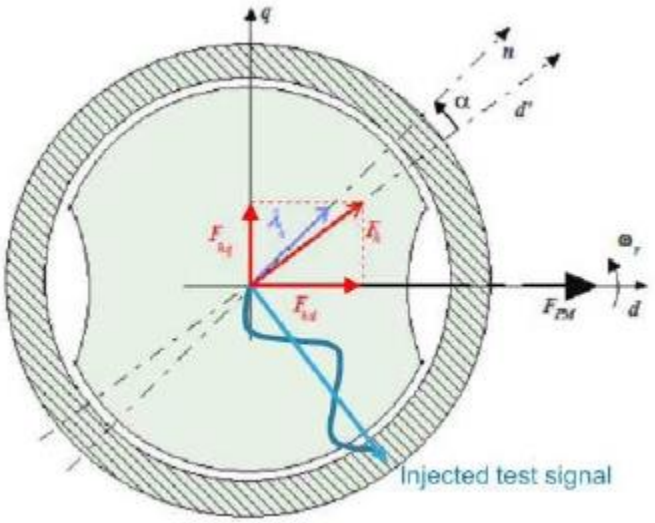
$$-\hat{e}_\alpha(kT)\cdot\sin(\theta_{k-1}) - \hat{e}_\beta(kT)\cdot\cos(\theta_{k-1}) \equiv A(\hat{\theta}_k - \theta_{k-1})$$

PI闭环作用：调节ω_r以保持其输入为0（电角度领先反电动势一个采样时间：$e_\alpha = \Phi_m p\omega_r \cos(p\omega_r t)$，$e_\beta = -\Phi_m p\omega_r \sin(p\omega_r t)$）

STM32高频注入 HFI

- HFI：针对I-PMSM马达的凸极特性，能够检测马达在静止或低速状态下的转子位置

- 假设转子位于某个角度上，在定子中注入一个200HZ~1KHZ的高频正弦信号，该信号位于假定的d’轴上



由于I-PMSM马达的凸极效应，Lq>Ld, 给定的轴上生成磁势Fh与实际磁链λh存在相位差，q’轴上会出现λh的分量---磁链偏离现象

STM32 HFI

- 无论d’轴的方向如何， λ_h在q’轴上的分量都存在。但是有2种情况例外：即d’轴与d轴或q轴重叠时，此时λ_h在q’轴上的分量为零
- 如下图，γ（注入信号相位角，即：d’轴与d轴的夹角）与其它物理量的关系：

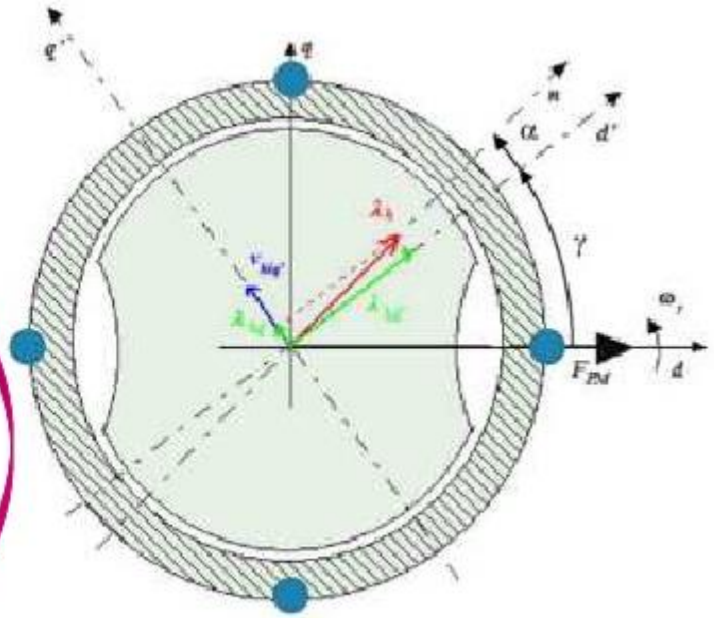
$$\alpha = \tan^{-1}\left(\frac{\lambda_{hq}}{\lambda_{hd}}\right) - \tan^{-1}\left(\frac{F_{hq}}{F_{hd}}\right) = \tan^{-1}\left\{\frac{(L_q - L_d)\sin 2\gamma}{2\left[L_d + (L_q - L_d)\sin^2 2\gamma\right]}\right\}$$

$$|\lambda_h| = I\sin(\omega_h t)\sqrt{L_q^2 + (L_q^2 - L_d^2)\sin^2\gamma}$$

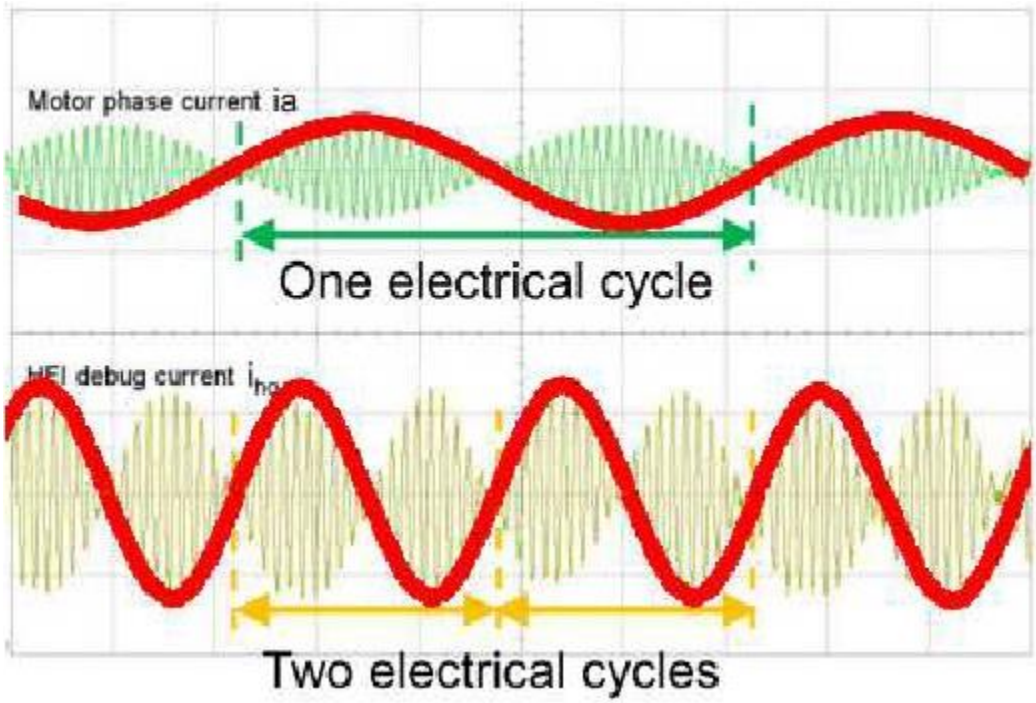
Deviation periodicity is half that of the magnetic structure

Both phase and amplitude are modulated by the motor magnetic structure

$$\lambda_{hq'} = |\lambda_h| \sin\alpha$$

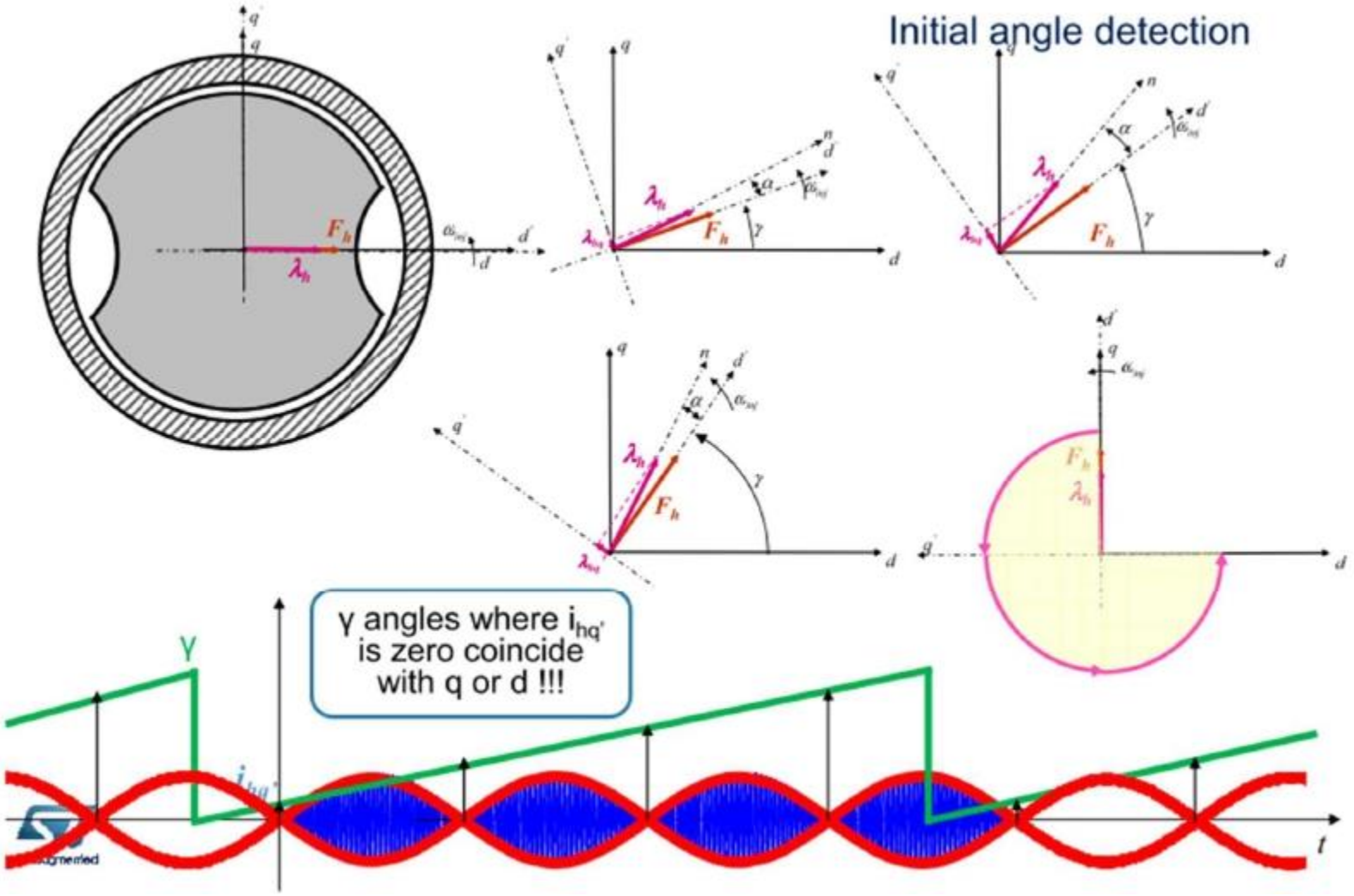


HFI 高频电流



HFI debug current i_{qh}的频率为马达相电流的2倍

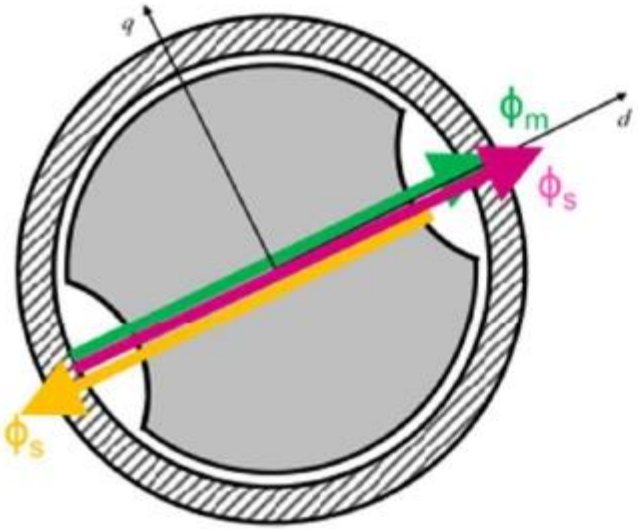
STM32 HFI – 初始位置

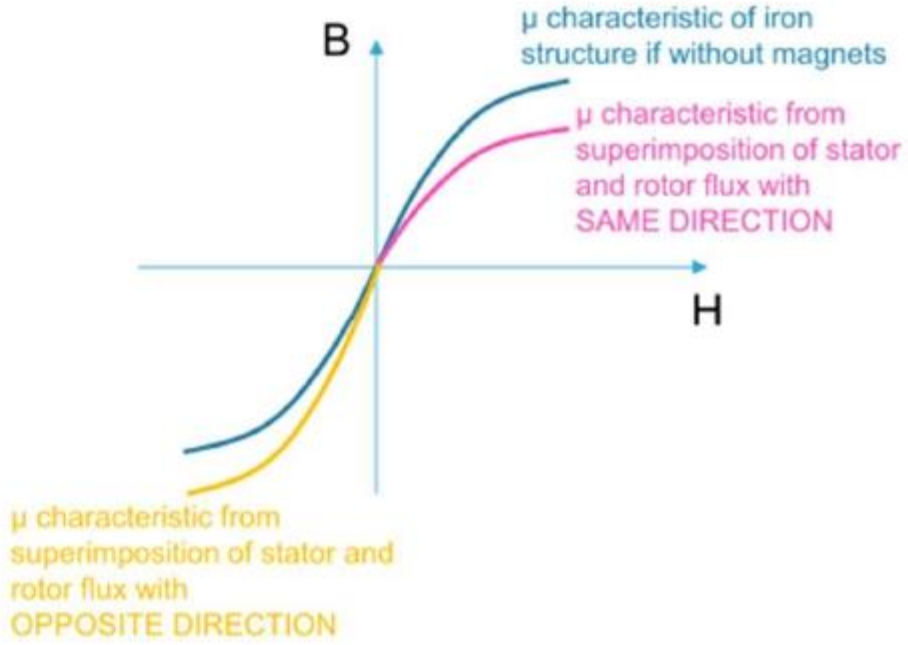


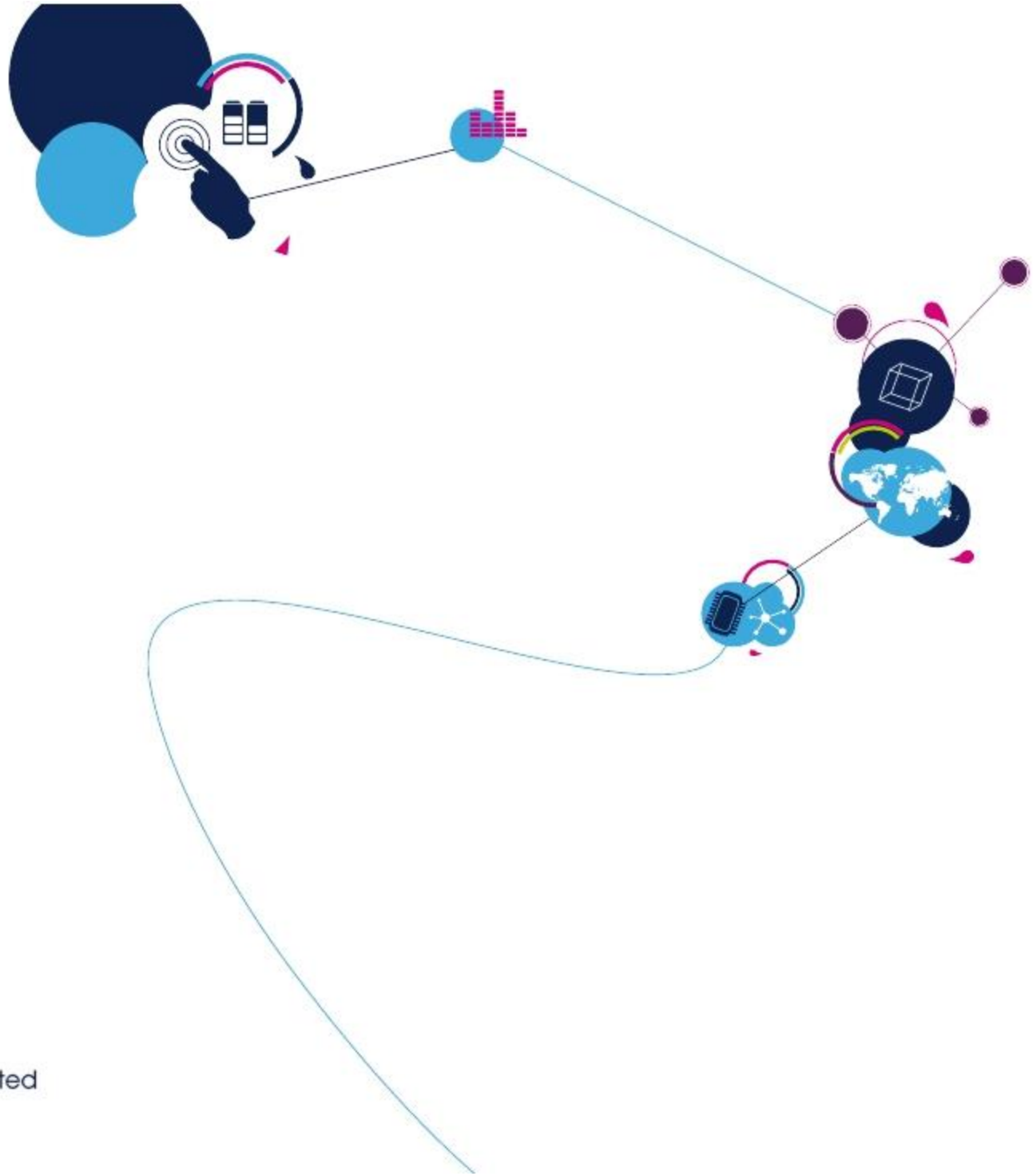
STM32 HFI – 初始位置
磁链极性检测

- 当d轴找到后，必须辨别转子磁链的极性，即：南/北极

- 方法：由于气隙磁场是定子和转子磁链的叠加，当定子的脉动磁链分别在南极和北极时，会导致马达导磁率μ不同，通过μ的不对称性的检测从而确定南/北极的方向





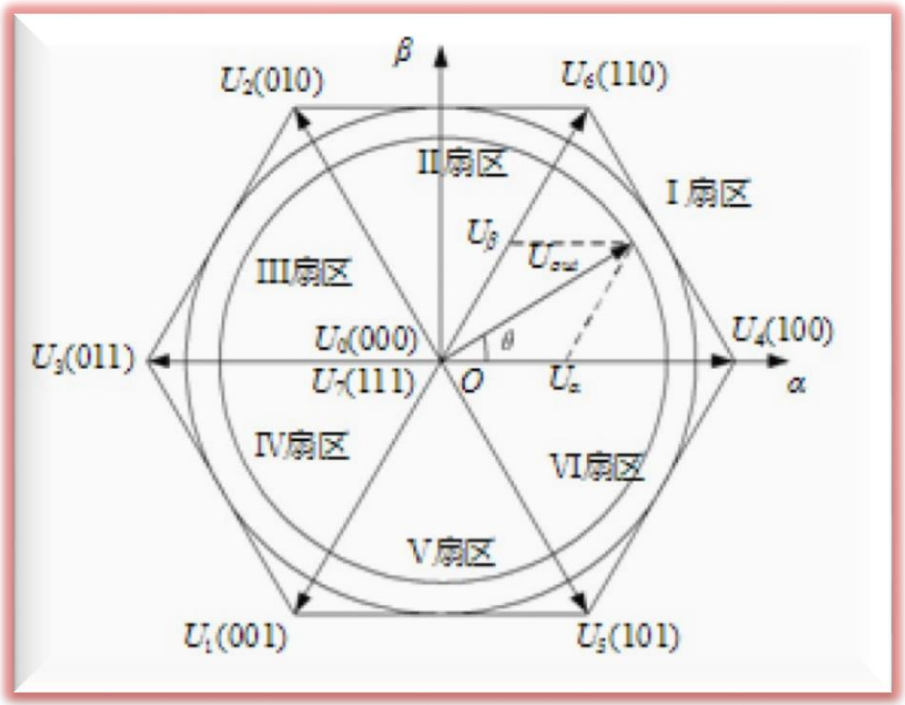


SVPWM

life.augmented

SVPWM

- SVPWM:电压空间矢量调制
- 改变各相导通时间的分配来形成所需的任意空间矢量。
- 电机获得幅值恒定的圆形旋转磁场
- 电压利用率高，谐波小，易于数字化实现
- 根据三相通电顺序组合，形成8种组合
- 矢量由这八种组合接力合成

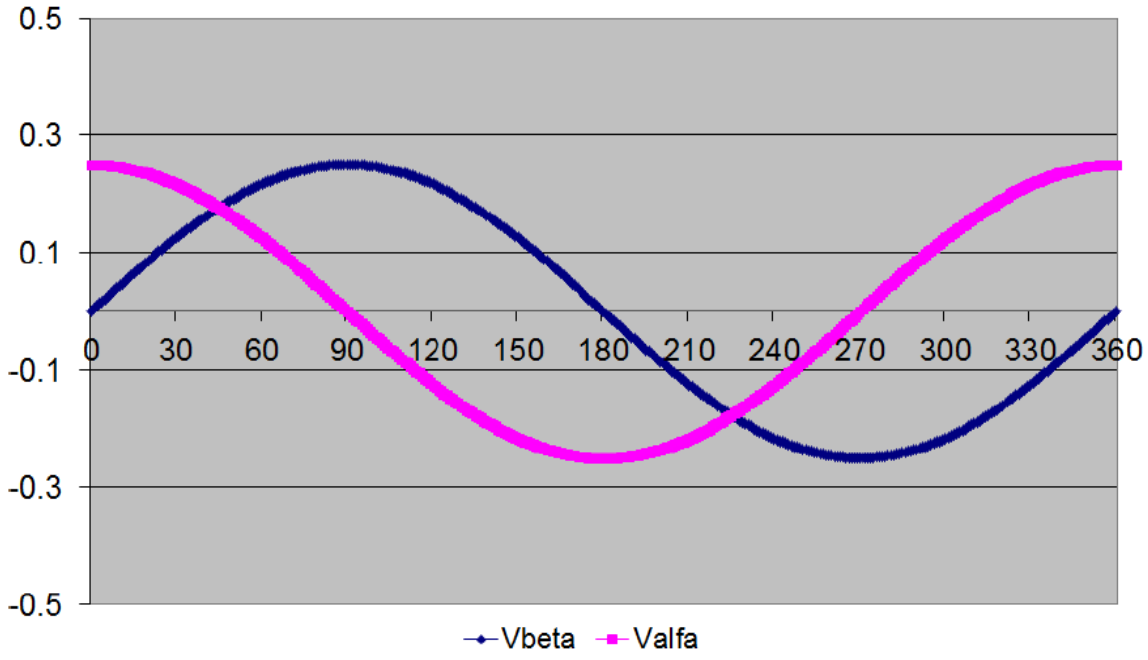


扇区判断

$$U_\alpha = \sqrt{3} \times T \times V_\alpha \ , \quad U_\beta = -T \times V_\beta$$

$$X = U_\beta , \qquad Y = \frac{U_\alpha + U_\beta}{2} \qquad Z = \frac{U_\beta - U_\alpha}{2}$$

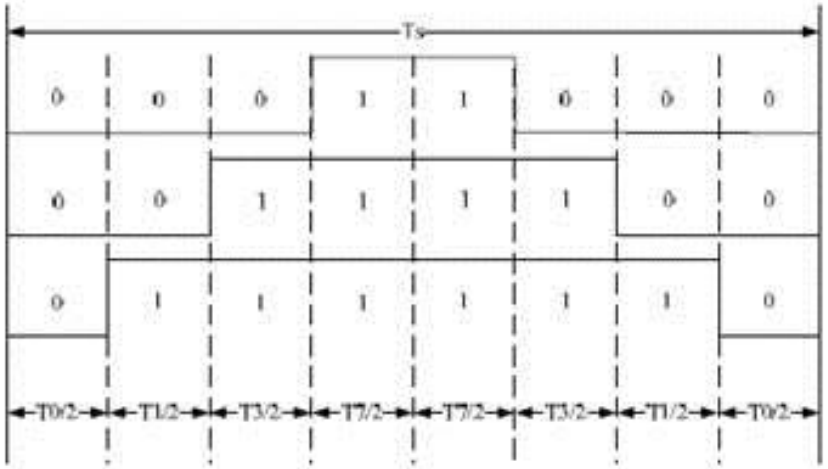
	Y < 0			Y ≥ 0		
	Z < 0	Z ≥ 0		Z < 0		Z ≥ 0
		X ≤ 0	X > 0	X ≤ 0	X > 0	
Sector	V	IV	III	VI	I	II



7段式PWM波占空比计算

$$T_0 = T_7 = (T_s - T_{Ux} - T_{Ux+60})/2$$

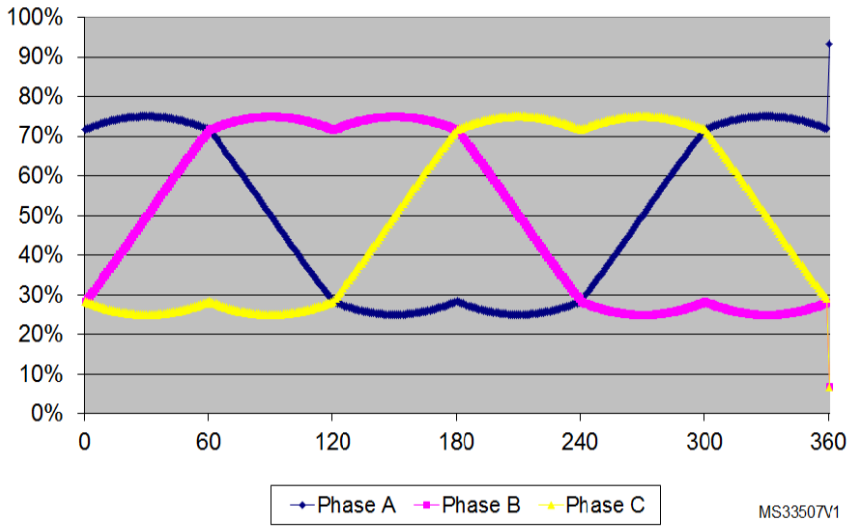
其中T_{Ux}，T_{Ux+60}代表相邻的两个基本电压空间矢量



$$Sector I, IV: t_A = \frac{T + X - Z}{2}, t_B = t_A + Z, t_C = t_B - X$$

$$Sector II, V: t_A = \frac{T + Y - Z}{2}, t_B = t_A + Z, t_C = t_A - Y$$

$$Sector III, VI: t_A = \frac{T - X + Y}{2}, t_B = t_C + X, t_C = t_A - Y$$

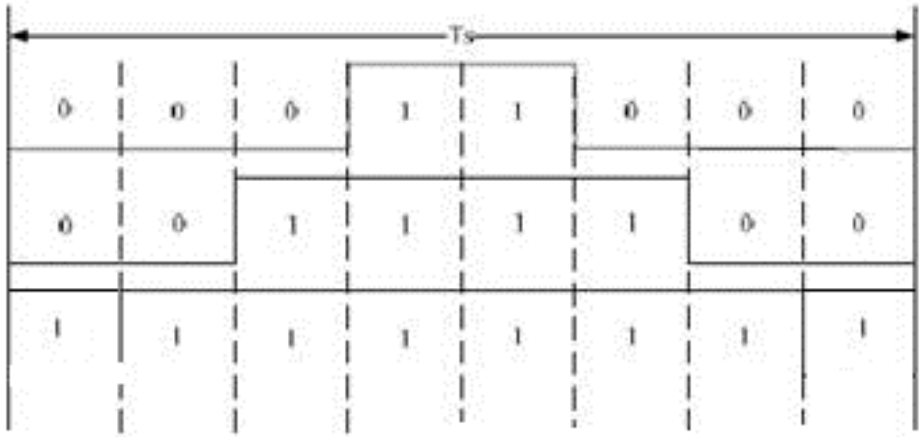


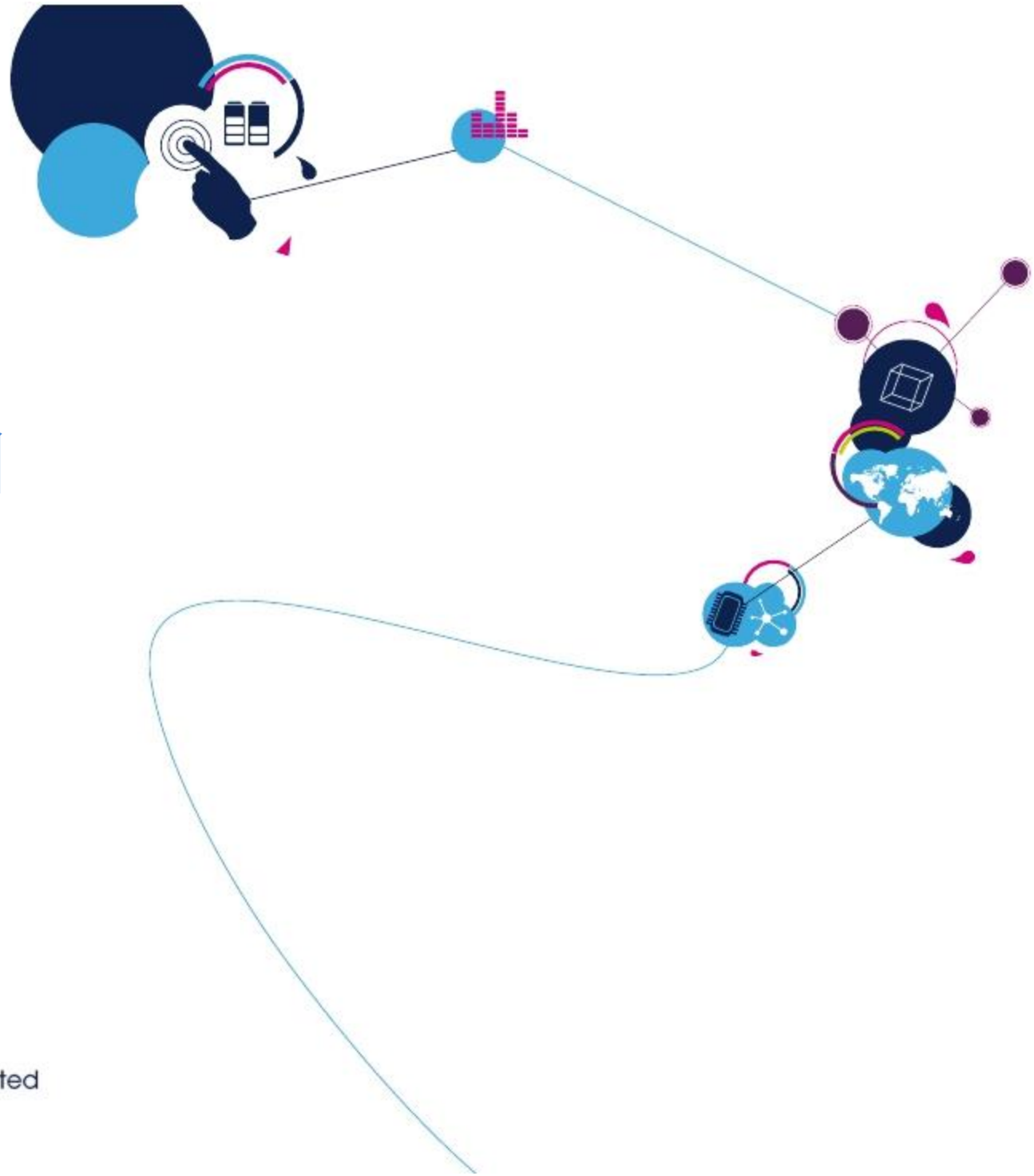
7段式 Vs 5段式

$$T_0(或T_7) = T_s - T_{Ux} - T_{Ux+60}$$

其中T_{Ux}，T_{Ux+60}代表相邻的两个基本电压空间矢量

- 在五段式SVPWM 中，有一相的相电压在一个PWM 周期内不发生翻转，
  - 占空比恒为0或恒为1
- 7段式开关损耗相对大
- 5段式谐波含量相对大



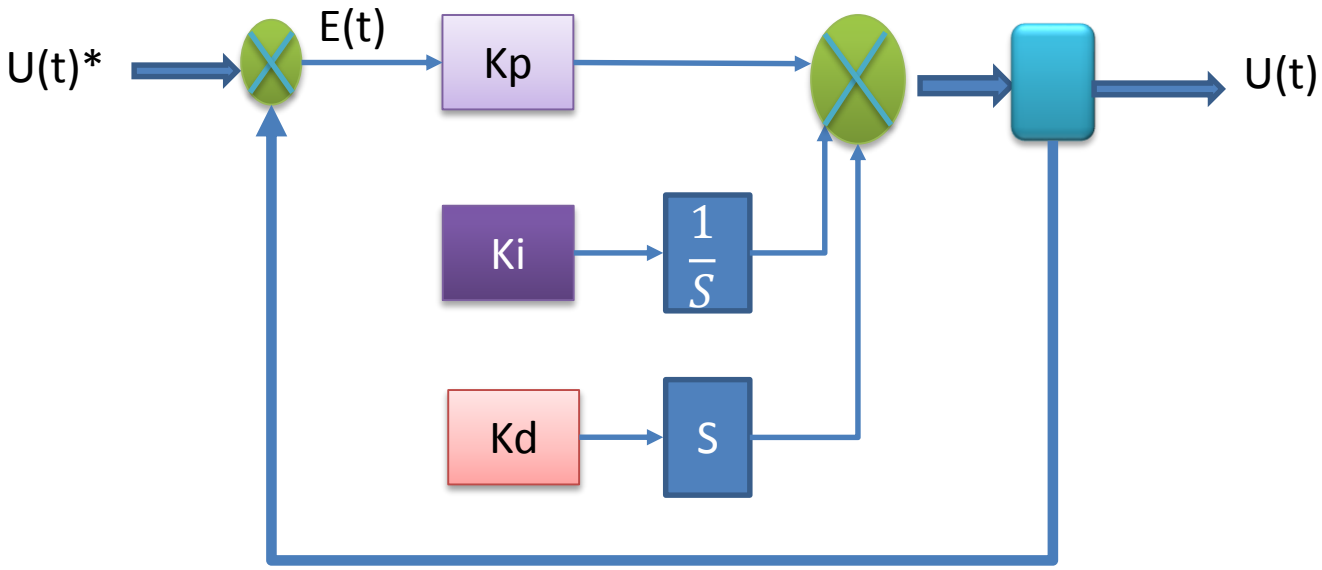


PID控制

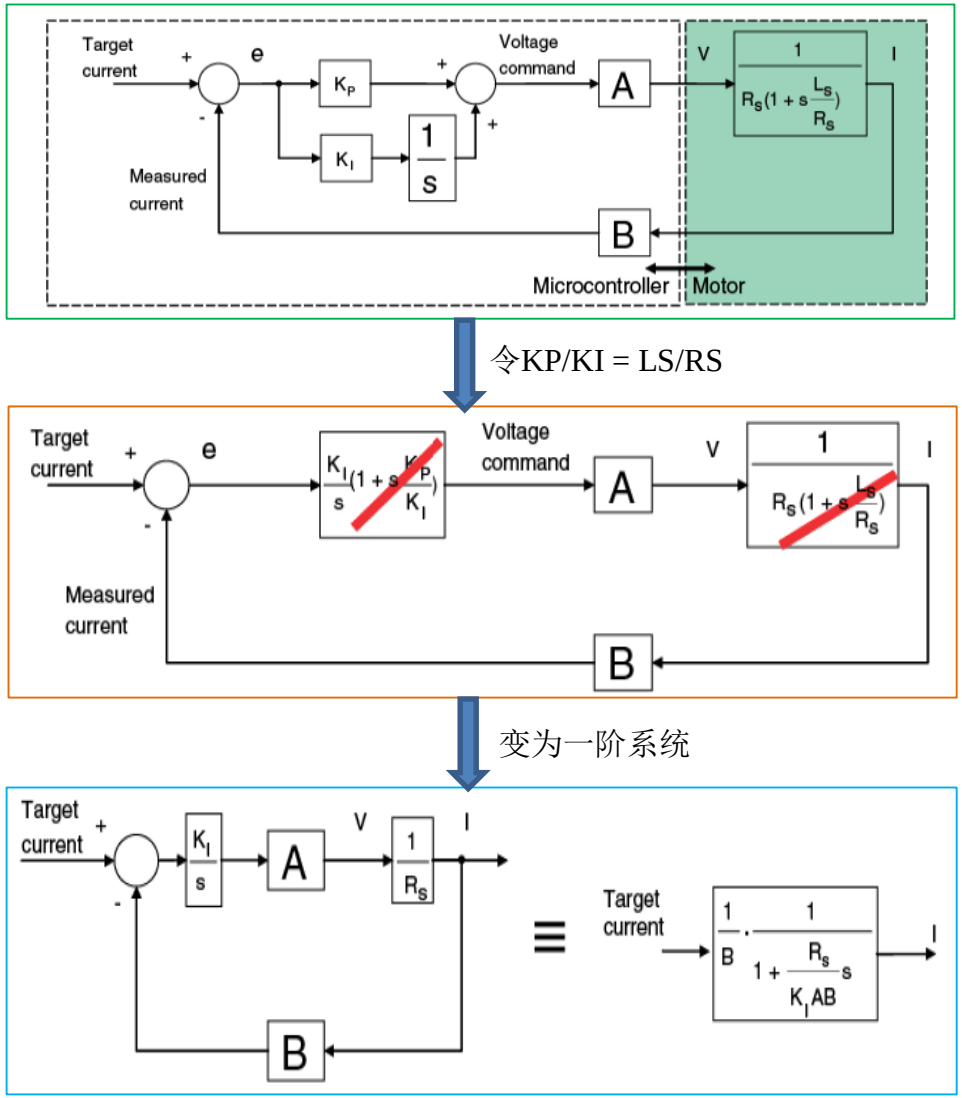
life.augmented

PID控制

$$f(t) = K_p \times Error_{sys}(t) + K_i \times \int_0^t Error_{sys}(t)dt + K_d \times \frac{d}{dt}(Error_{sys}(t))$$



电机电流环PID控制



$$K_P = L_S \frac{\omega_C}{AB}$$

$$K_i = \frac{R_S \cdot \omega_C \quad T}{AB}$$

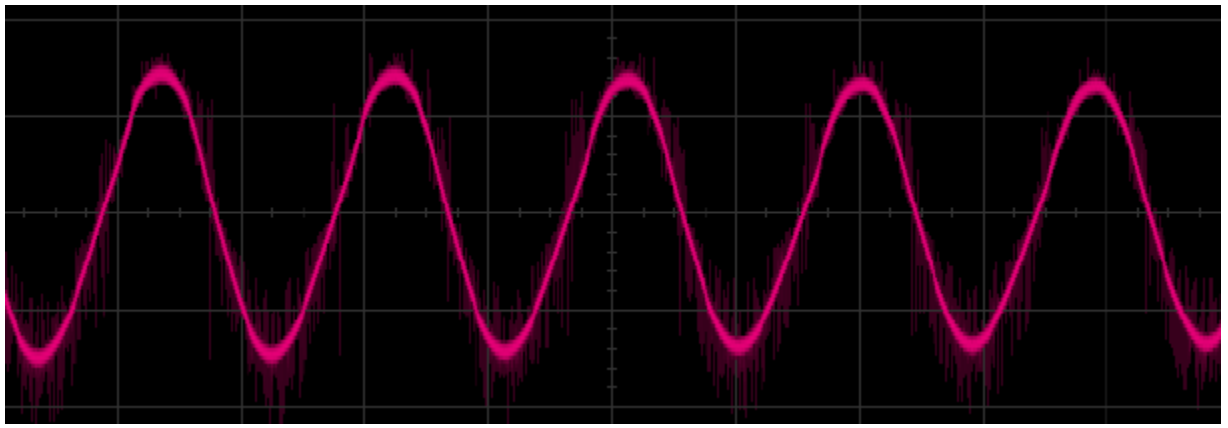
$$AB = \frac{V_{Bus}DC \cdot R_{shunt} \cdot A_{op}}{3.3}$$

$$A = \frac{V_{Bus}DC}{2^{16}}$$

$$B = \frac{R_{shunt}A_{op}2^{16}}{3.3}$$

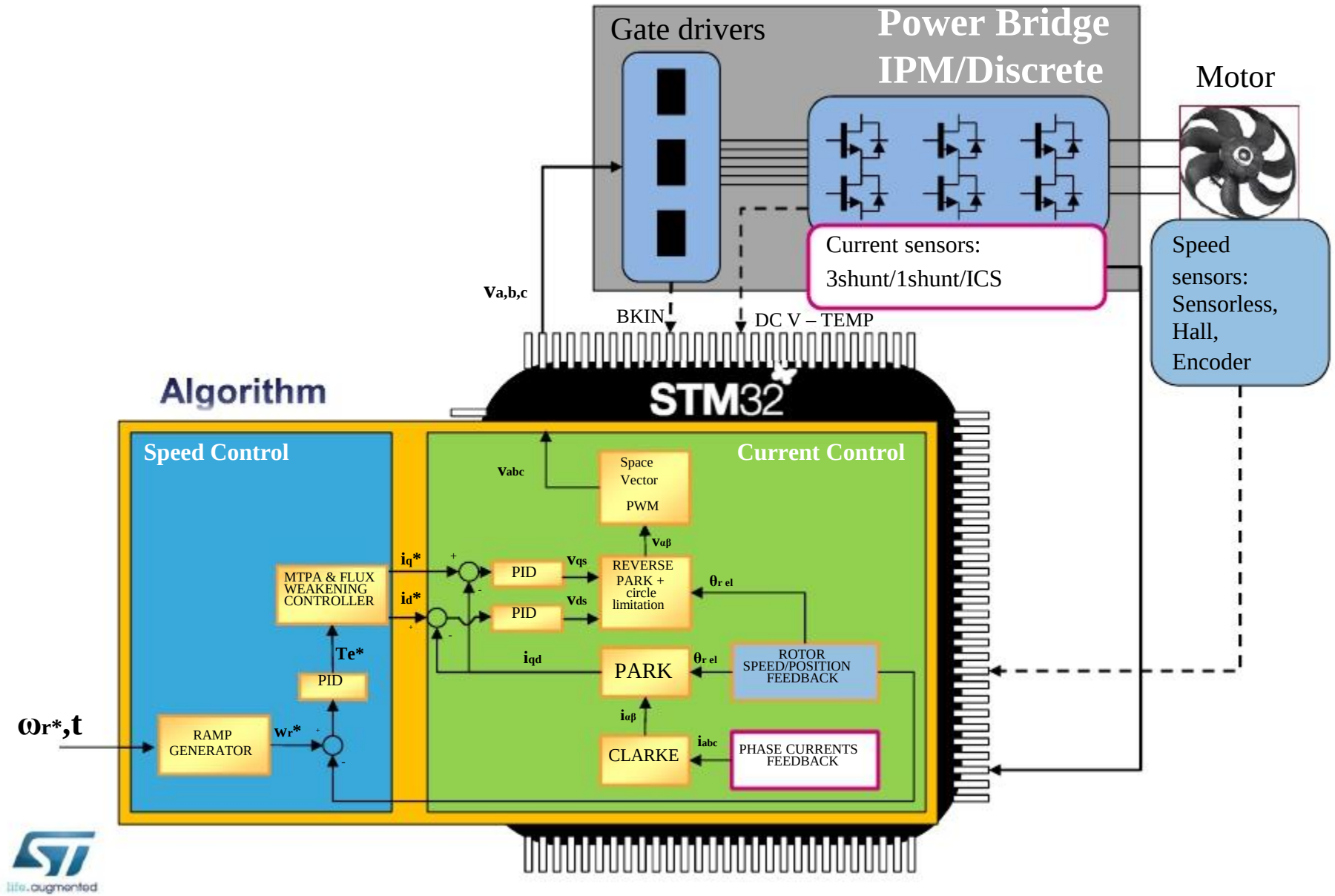
总结

- 通过上面的主要FOC控制的要素的具备，我们已经可以对电机进行FOC控制，当然性能的调试要更进一步的细调
- FOC控制由数学模型进一步实化为电机马达的实际控制
- 得益于当今MCU技术的飞速发展，更多更好的控制算法可以走出实验室，更多的可用于实际应用中

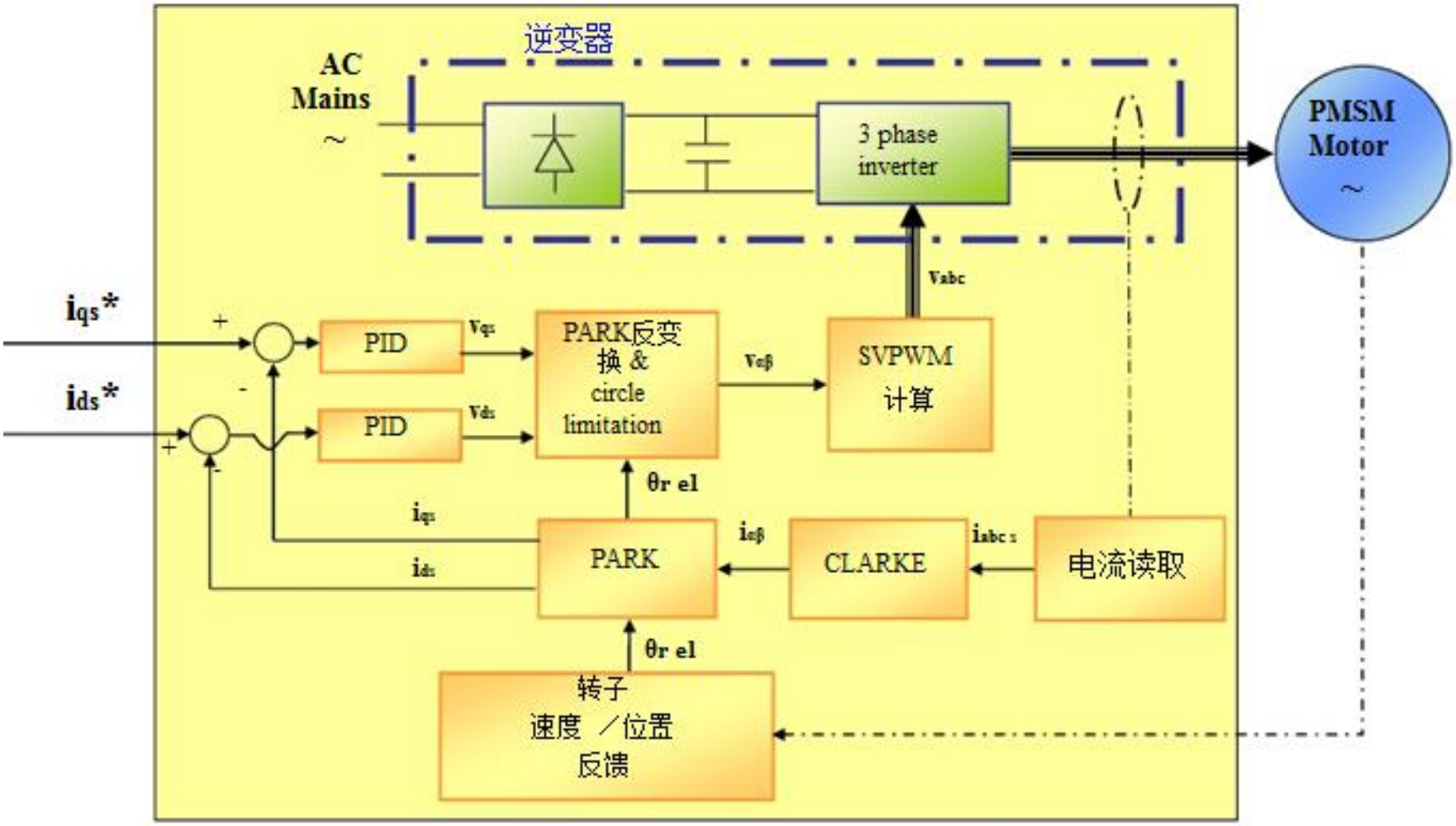


life.augmented

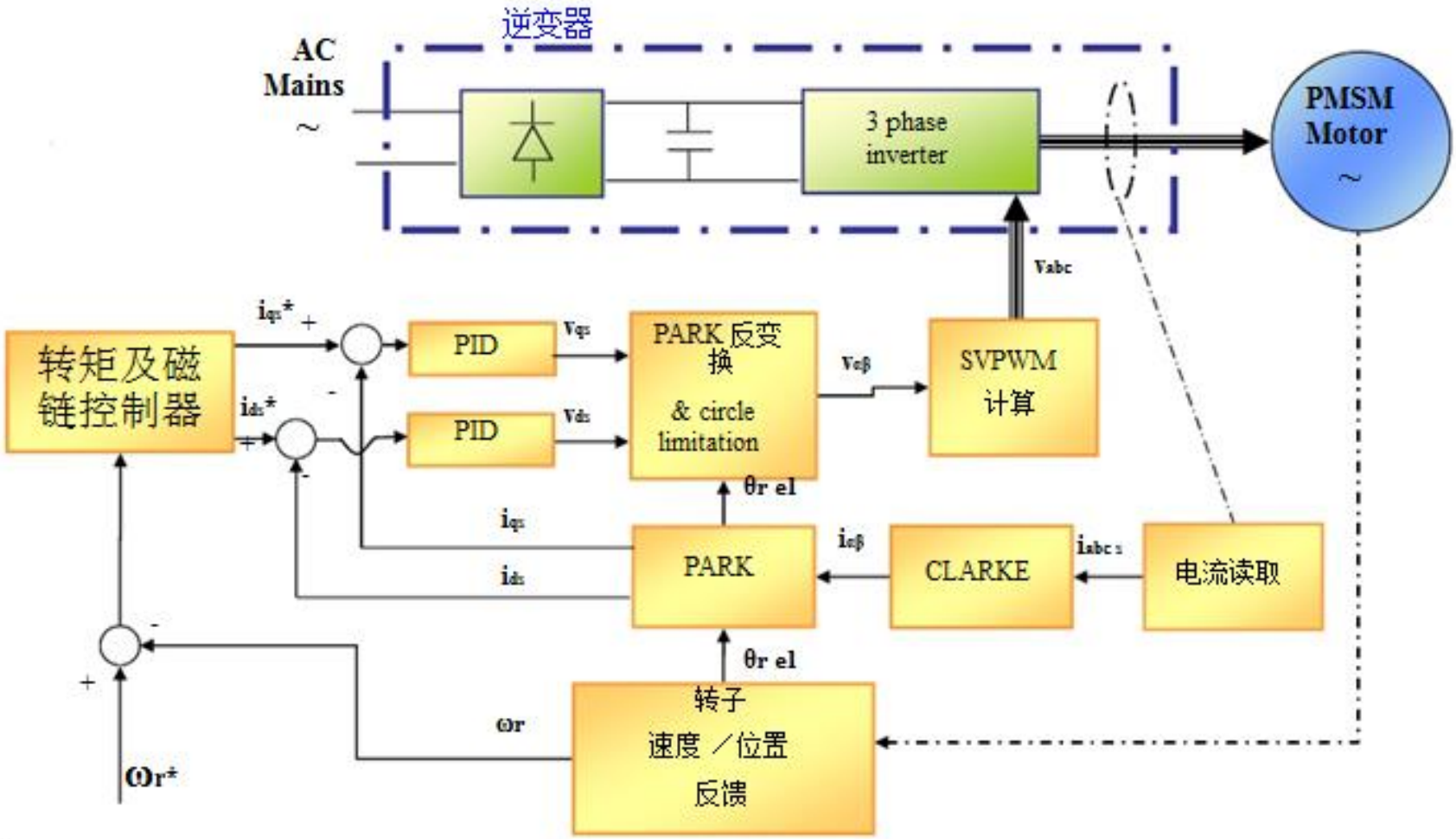
PMSM FOC – 框图


Gate drivers
Power Bridge
IPM/Discrete
Motor
Current sensors:
3shunt/1shunt/ICS
Speed
sensors:
Sensorless,
Hall,
Encoder
Va,b,c
BKIN
DC V – TEMP
Algorithm
STM32
Speed Control
Current Control
Vabc
Space
Vector
PWM
Vαβ
iq*
id*
MTPA & FLUX
WEAKENING
CONTROLLER
PID
PID
Vqs
Vds
REVERSE
PARK +
circle
limitation
θr el
Te*
PID
iqd
PARK
θr el
ROTOR
SPEED/POSITION
FEEDBACK
ωr*,t
RAMP
GENERATOR
Wr*
iαβ
CLARKE
iabc
PHASE CURRENTS
FEEDBACK

FOC转矩控制


逆变器
AC
Mains
~
3 phase
inverter
PMSM
Motor
~
Vabc
iqs*
ids*
+
-
+
-
PID
PID
Vqs
Vds
PARK反变
换 &
circle
limitation
Vαβ
SVPWM
计算
θr el
iqs
ids
PARK
iαβ
CLARKE
iabc s
电流读取
θr el
转子
速度 /位置
反馈

FOC速度控制

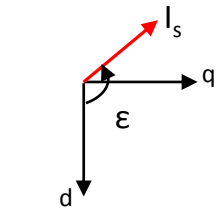

逆变器
AC
Mains
~
3 phase
inverter
PMSM
Motor
~
Vabc
转矩及磁
链控制器
PID
PID
PARK 反变
换
& circle
limitation
SVPWM
计算
θr el
PARK
CLARKE
电流读取
θr el
转子
速度 ／位置
反馈
ωr
ωr*

每安培最大扭矩控制 (MTPA)

- 找到一条电流极限和电压极限交叉的曲线
- 建立id，iq的对应查表数据
- 当马达的速度高于其额定值时， MTPA 控制就由弱磁控制代替

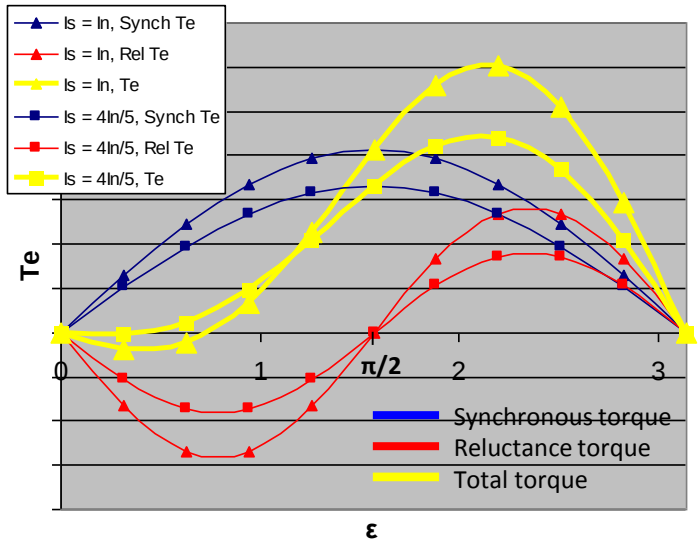
$$T_e = \frac{3}{2} p\left(\lambda_{ds} i_{qs} - \lambda_{qs} i_{ds}\right) = \frac{3}{2} p\left[\Phi_m i_{qs} + \left(L_d - L_q\right) i_{qs} i_{ds}\right]$$

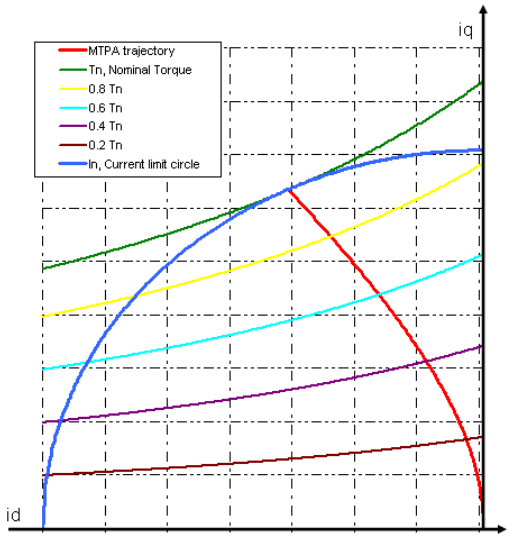
synchronous torque　　reluctance torque



$$i_{qs} = I_s \sin\varepsilon$$
$$i_{ds} = I_s \cos\varepsilon$$

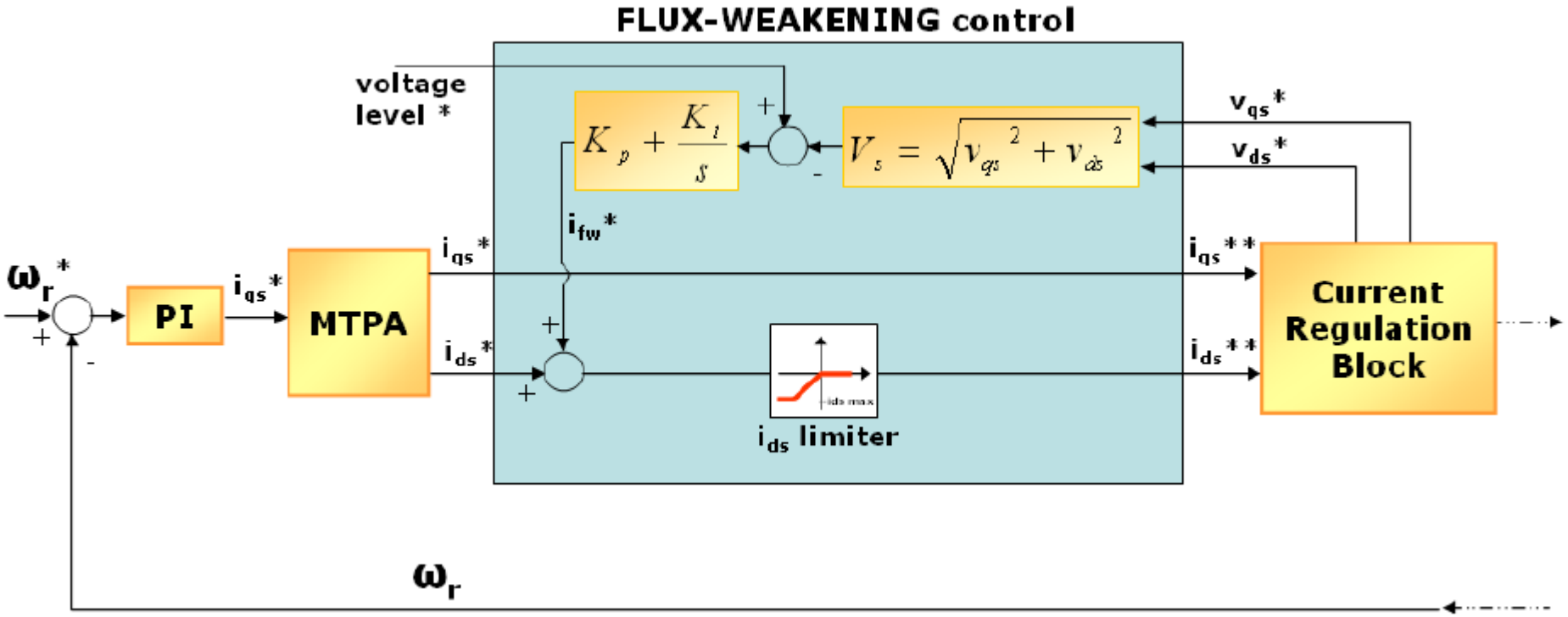
$$\Longrightarrow \quad T_e = \frac{3}{2} p\left(\Phi_m I_s \sin\varepsilon + \frac{I_s^{\,2}}{2}\left(L_d - L_q\right)\sin 2\varepsilon\right)$$





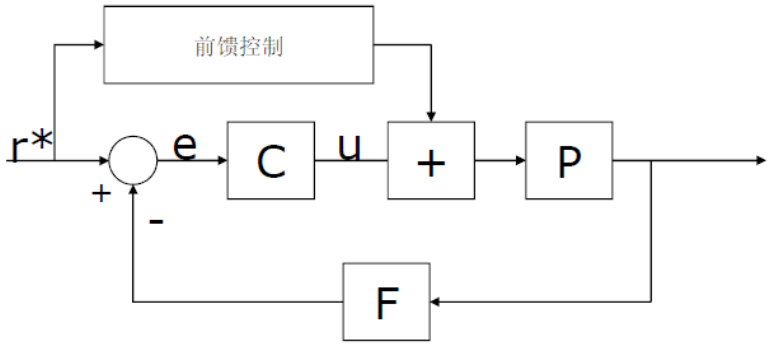
弱磁控制

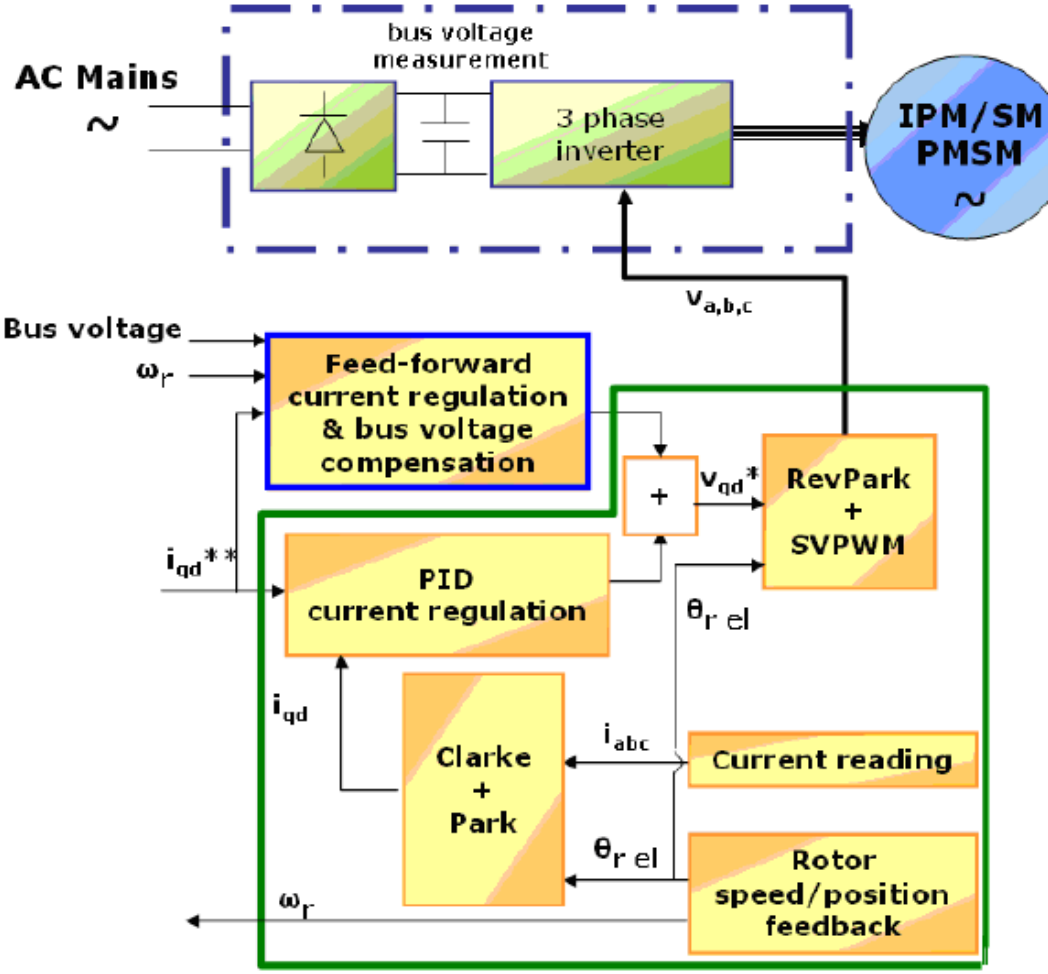
- “闭环” 弱磁控制: 不需要知道电机的精确参数, 这样减少电机参数漂移造成对参数敏感。
- 控制环基于定子电压监测。
- 在一些应用场合，可能需要刹车电阻来及时释放由反向电动势给母线电压上的电容充电的能量，以免导致过压以致损坏硬件。



电流前馈

- 相对于单反馈控制而言，当一个大的扰动在其未影响系统输出的情况下能被测到，则前馈控制联合反馈控制可以明显地改善控制性能





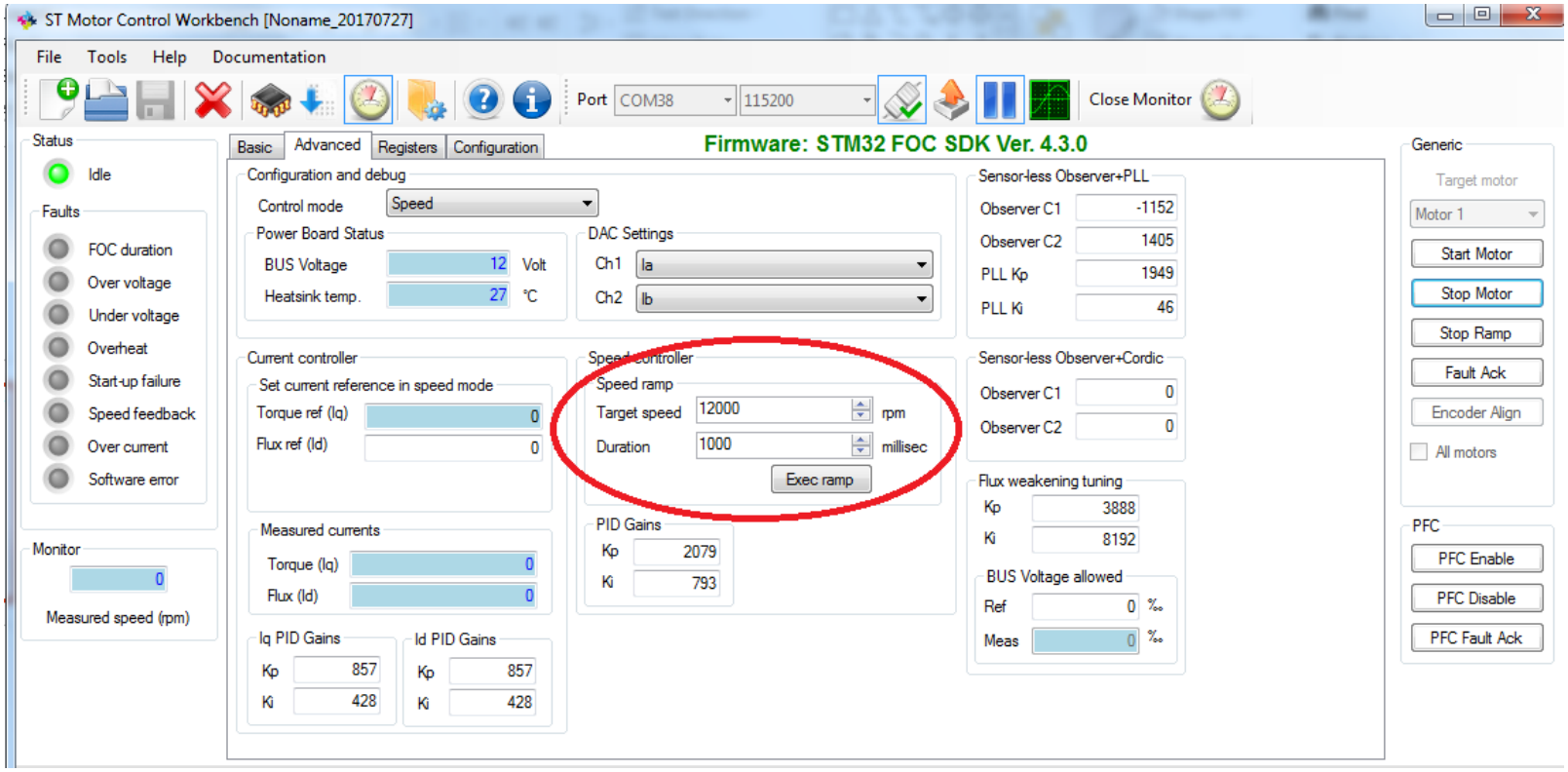
实际操作环节 – 力矩与速度

5~10min

life.augmented

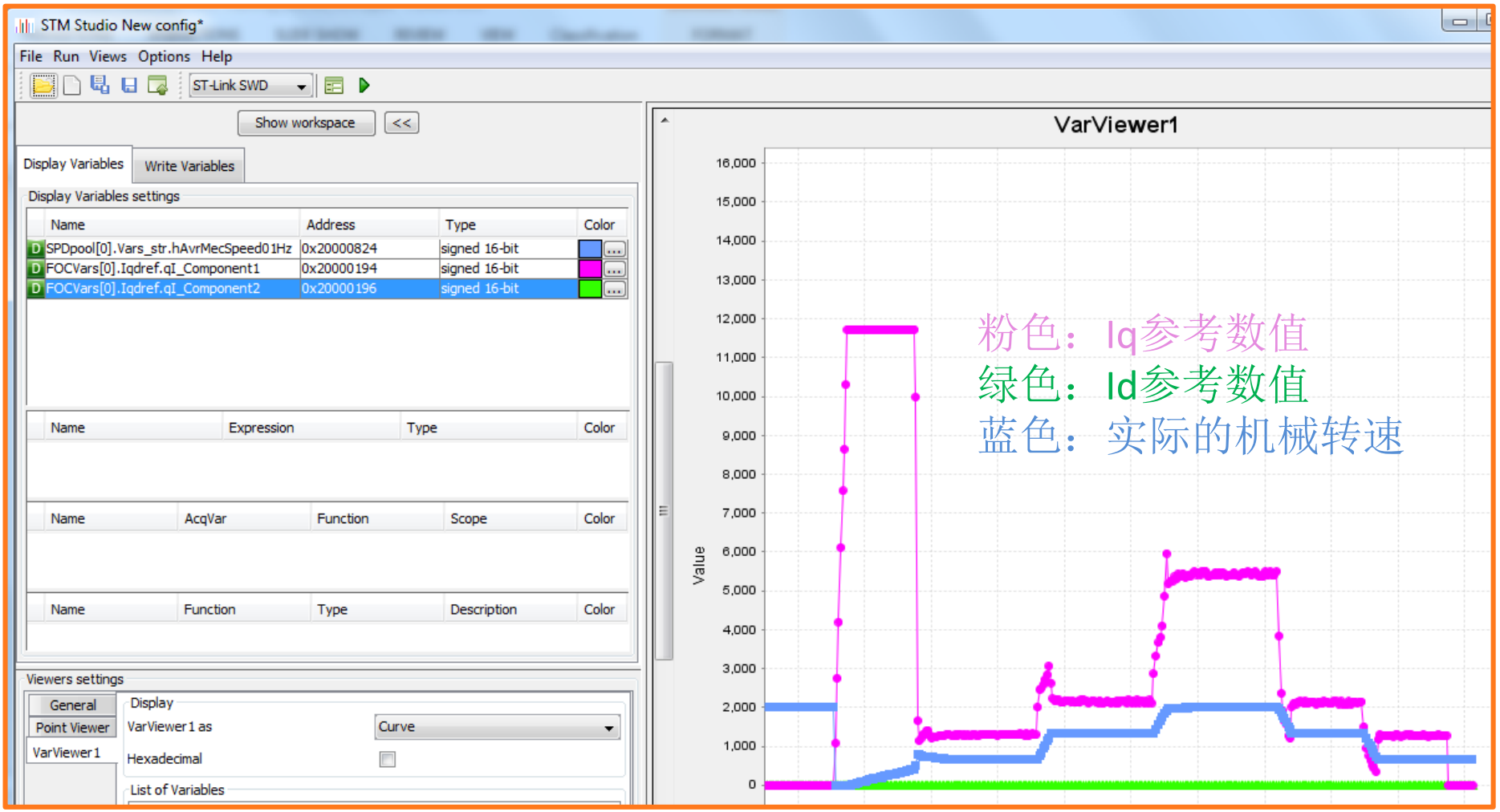
操作说明1/2

- 使用Workbench不断更改速度 4000RPM→800RPM→12000RPM



操作说明2/2

➢ 使用STM Studio查看波形

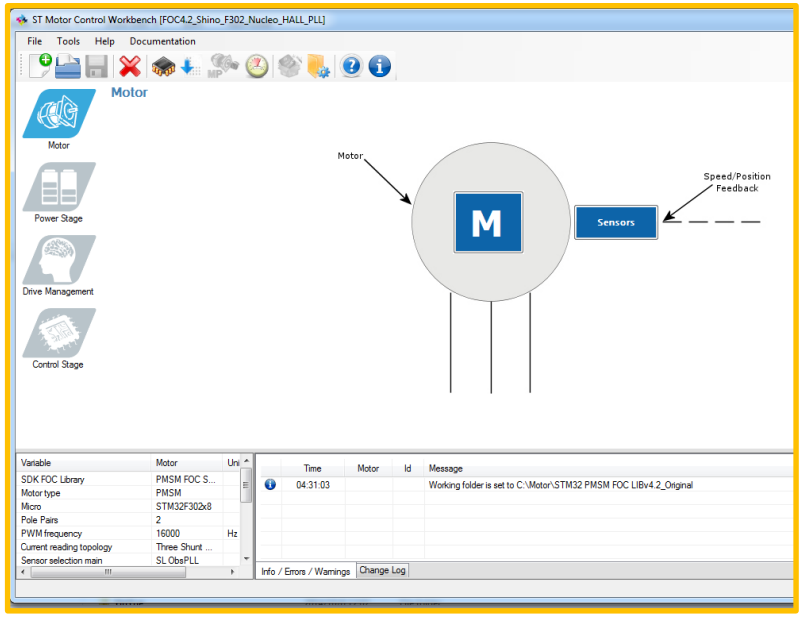


认识ST FOC SDK V4.3电机库

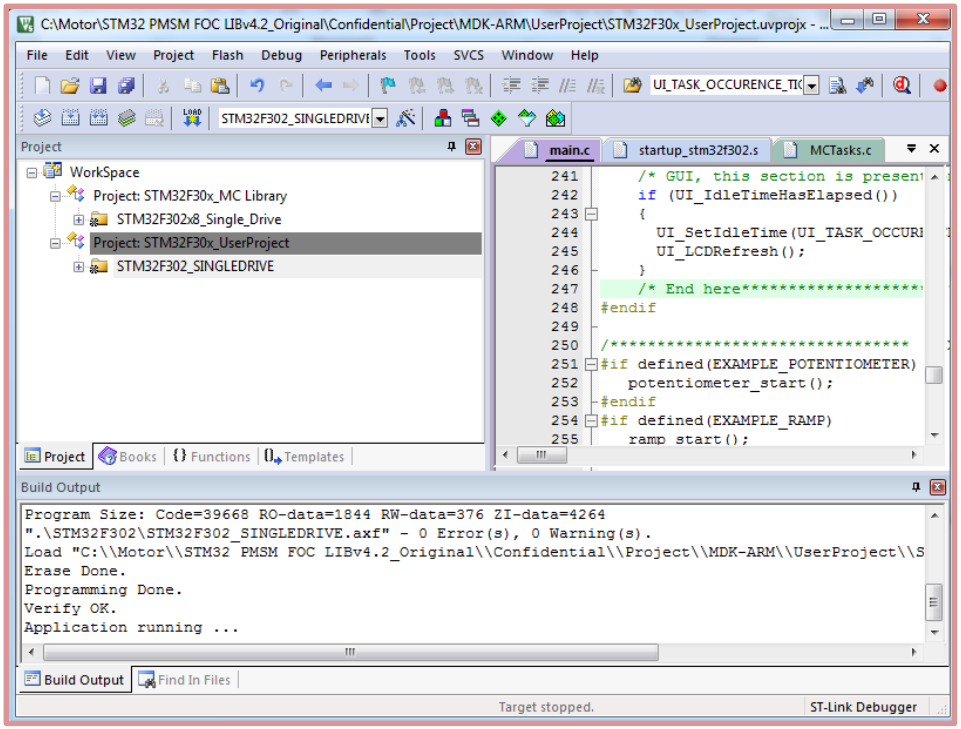
life.augmented

电机库构成

- 包含源代码程序STM32 PMSM FOC LIBv4.3
- 以及PC端配置软件ST MC Workbench GUI

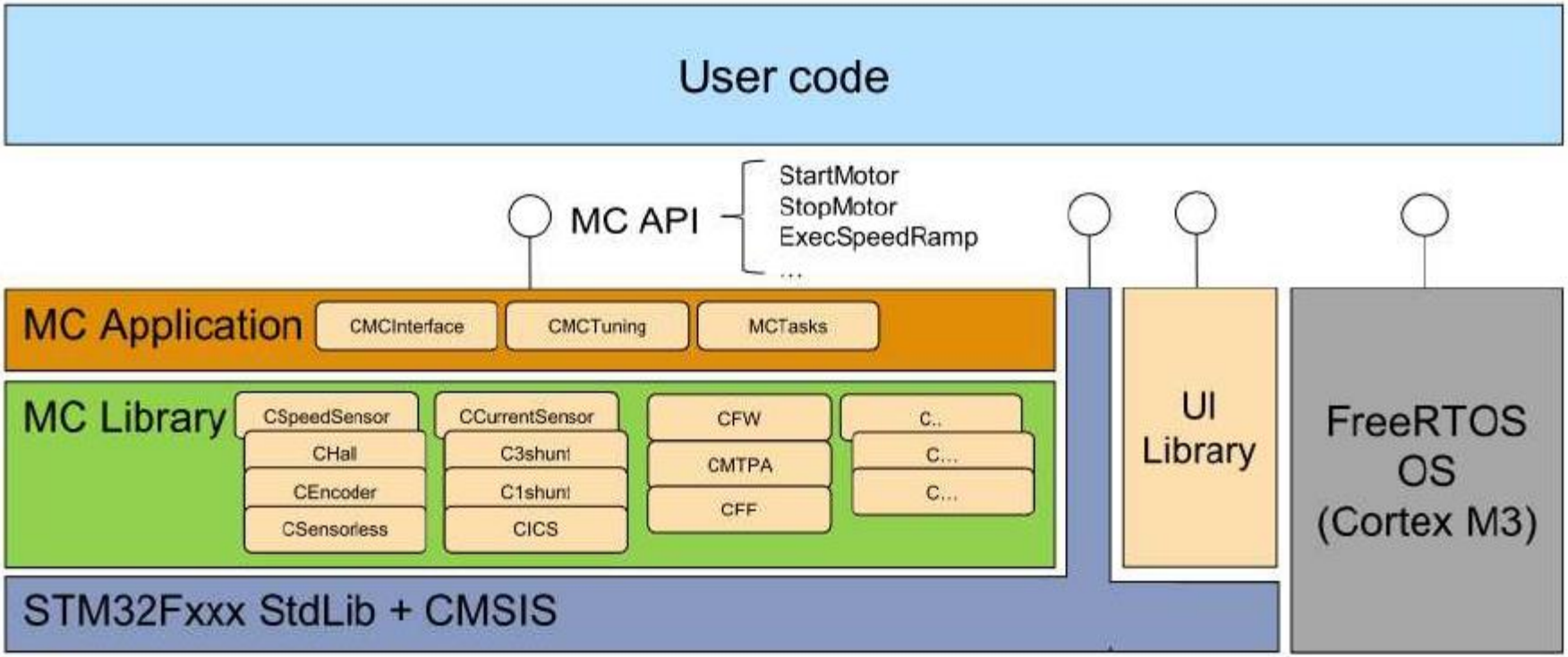


*.h文件



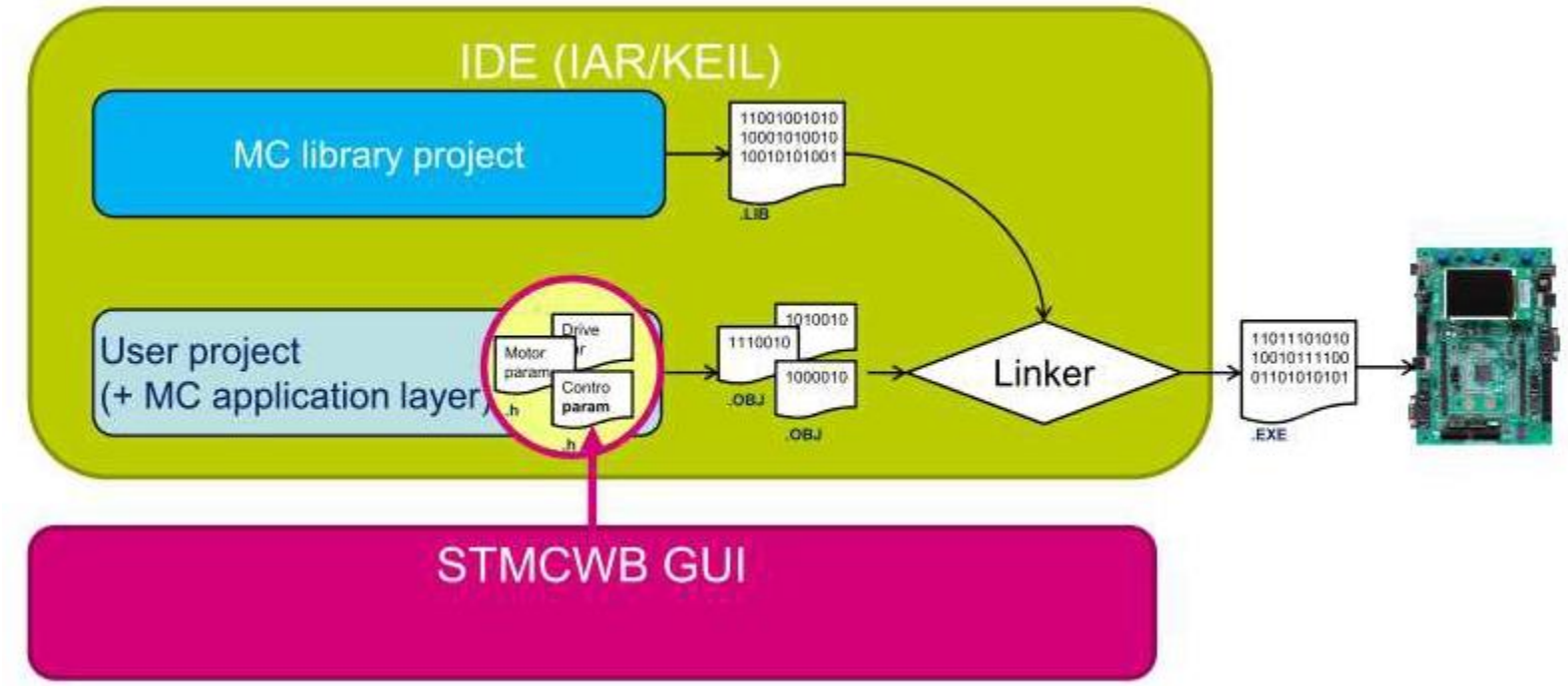
新的架构

- SDK软件库分为若干个软件层，分别为：MCU标准外设软件库、马达控制库及马达控制应用层
- 还包含用户界面层及FreeRTOS模块



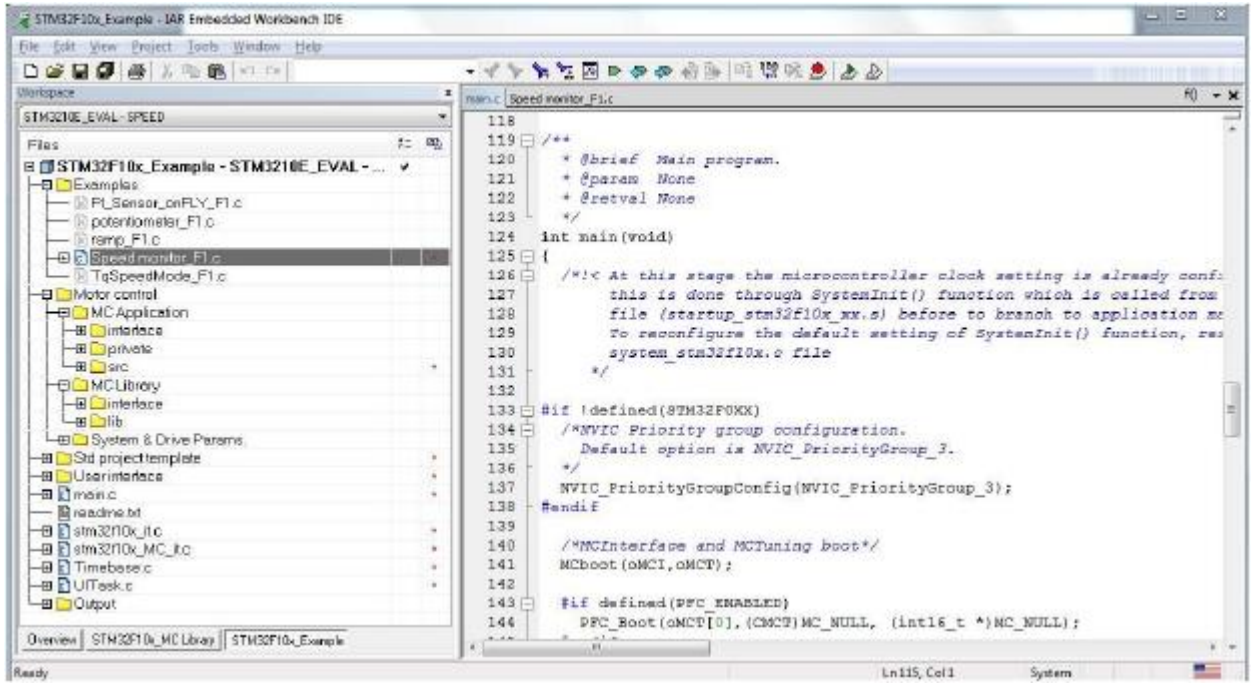
Workspaces and projects

- 每一个马达控制软件的workspace包含有2个project:
  - MC Library project
  - User project(包含MC Application层)：即在V4.3中，User project层和MC Application层被放到了同一层
- 每个project都需要分别编译，并最终连接生成可执行代码



软件库使用例程

- 软件库包含若干个例程，用于指导用户如何在SDK的基础上开发一个新的项目，并帮助用户更好地理解MC API函数
- 仅仅支持IAR IDE： \Project\EWARM \STM32F10x_Example.eww
- 会在后续的培训内容中详述



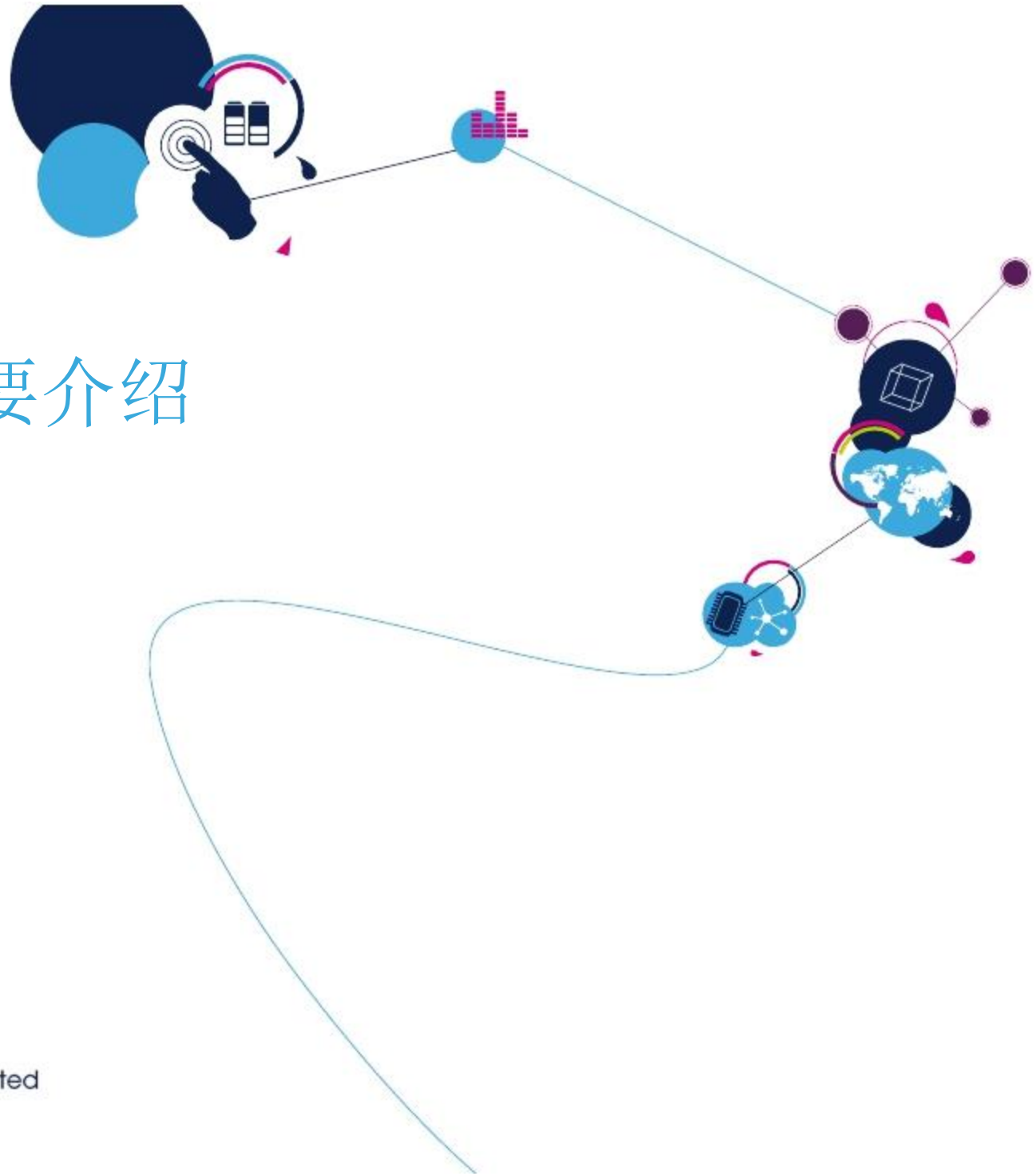
MC Application – User Tasks

- Safety task,medium frequency task,UI task及User task用systick的500us中断作为时基

Scope	Name	Main role	Priority	Frequency (Period)
MC tasks	**High frequency task**	Execute the current regulation loop	IRQ	$\frac{PWM\ frequency}{FOC\ rate}$
	Safety task	Over voltage management Switch off PWM if fault occurs User ADC conversions	SysTick	2kHz (500µs)
	Medium frequency task	Execute the speed regulation loop State machine management	SysTick	500Hz* (2ms)*
User tasks	**UI task**	LCD screen refresh	Main	10Hz (100ms)
	User task	Time base for example projects	Main	10Hz (100ms)

*：缺省值，可在workbench的Driver management - Drive settings – Speed regular – Execution rate中更改

life.augmented

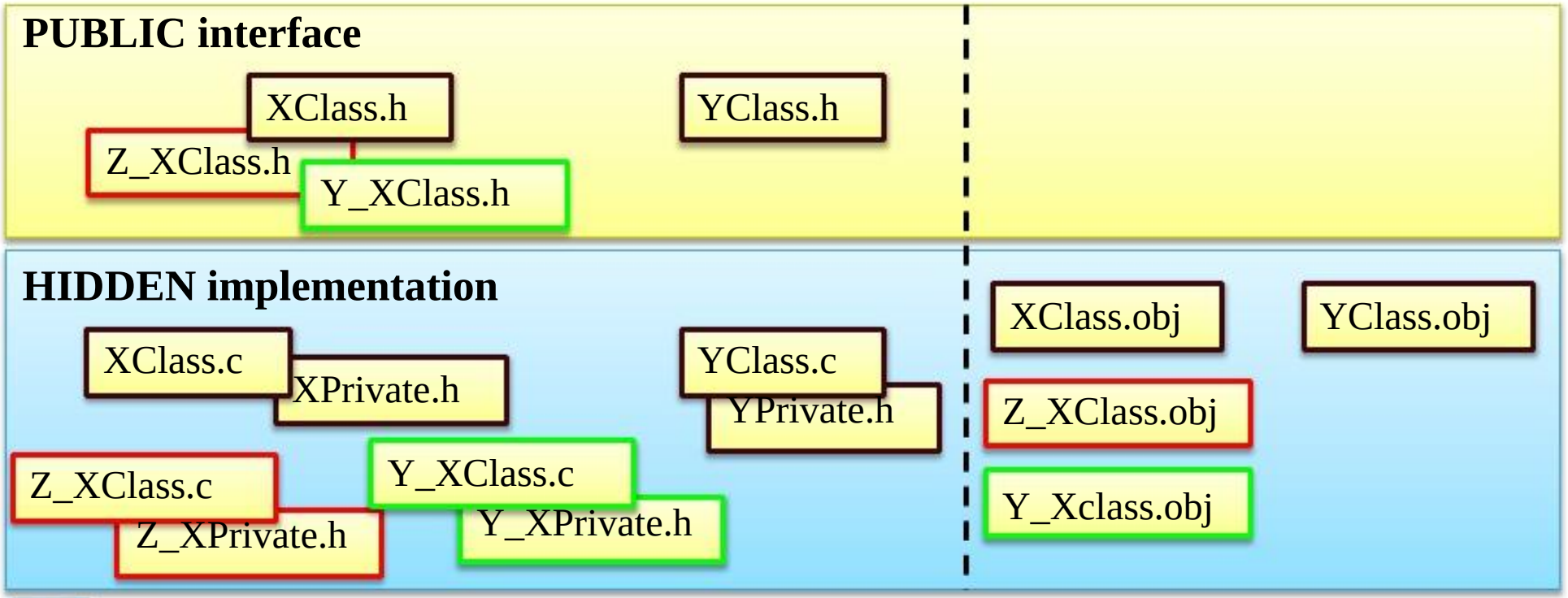


电机库简要介绍

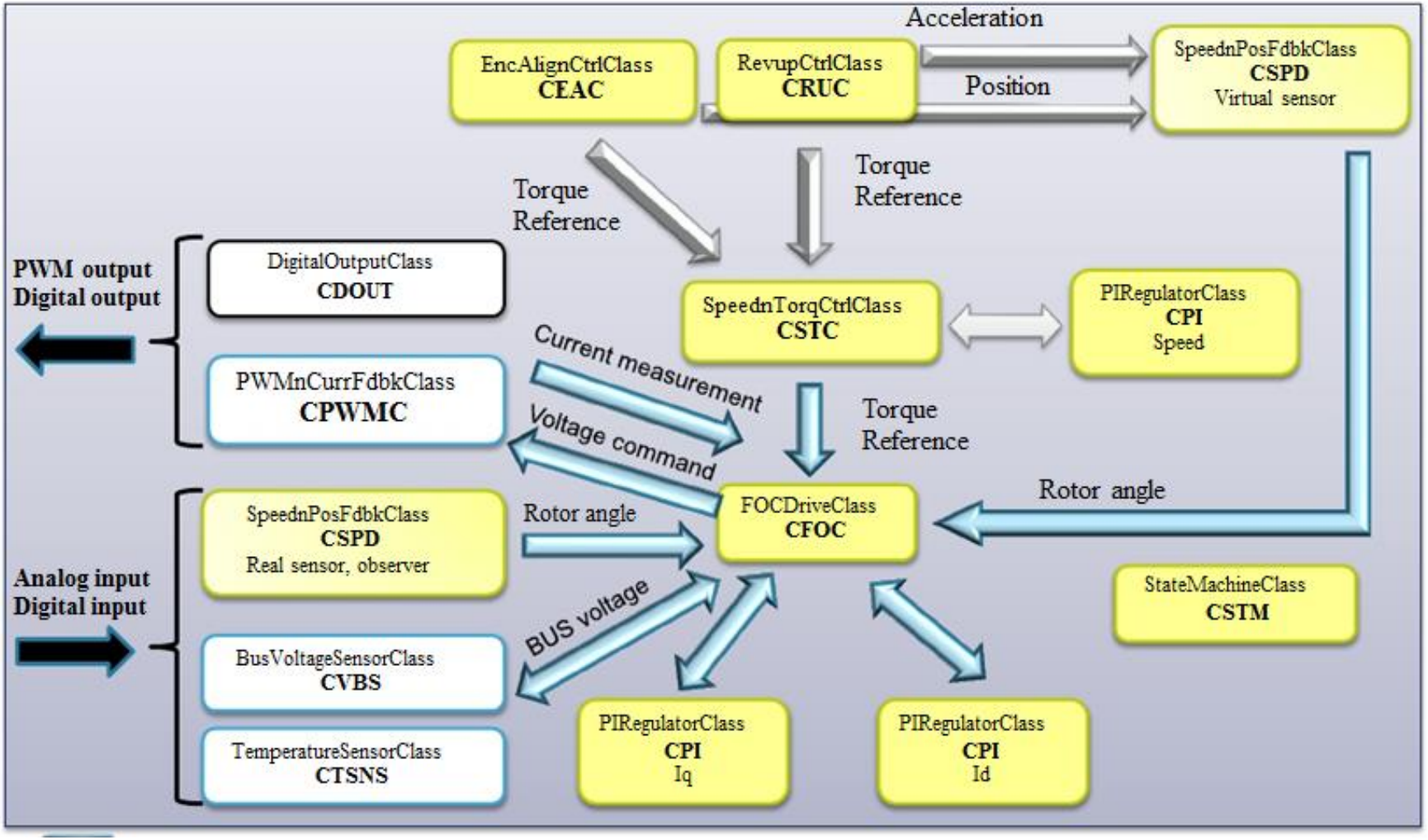
life.augmented

电机库OOP架构

- C语言编写的仿制面向对象程序，并非真正C++程序
- 每个类包含三个文件：
  - xClass.h　用户可使用的必要资源的头文件
  - xPrivate.h 私有化头文件（包含对象变量，数据类型）
  - xClass.c　私有化的.c文件，对象真正的操作代码

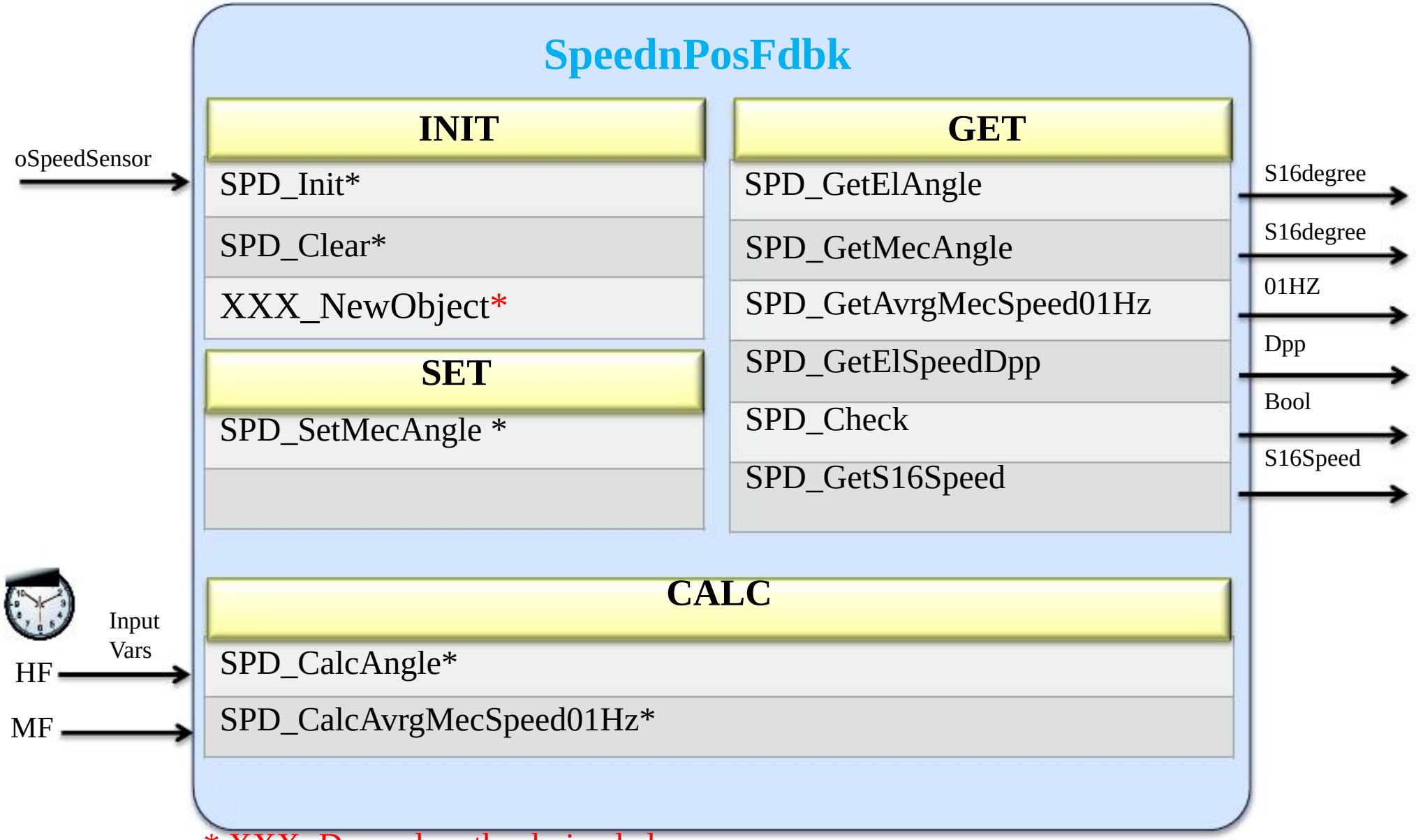


电机库类关系图


Acceleration
EncAlignCtrlClass
CEAC
RevupCtrlClass
CRUC
Position
SpeednPosFdbkClass
CSPD
Virtual sensor
Torque
Reference
Torque
Reference
PWM output
Digital output
DigitalOutputClass
CDOUT
SpeednTorqCtrlClass
CSTC
PIRegulatorClass
CPI
Speed
Current measurement
PWMnCurrFdbkClass
CPWMC
Voltage command
Torque
Reference
Rotor angle
SpeednPosFdbkClass
CSPD
Real sensor, observer
Rotor angle
FOCDriveClass
CFOC
Analog input
Digital input
StateMachineClass
CSTM
BUS voltage
BusVoltageSensorClass
CVBS
PIRegulatorClass
CPI
Iq
PIRegulatorClass
CPI
Id
TemperatureSensorClass
CTSNS

举例说明

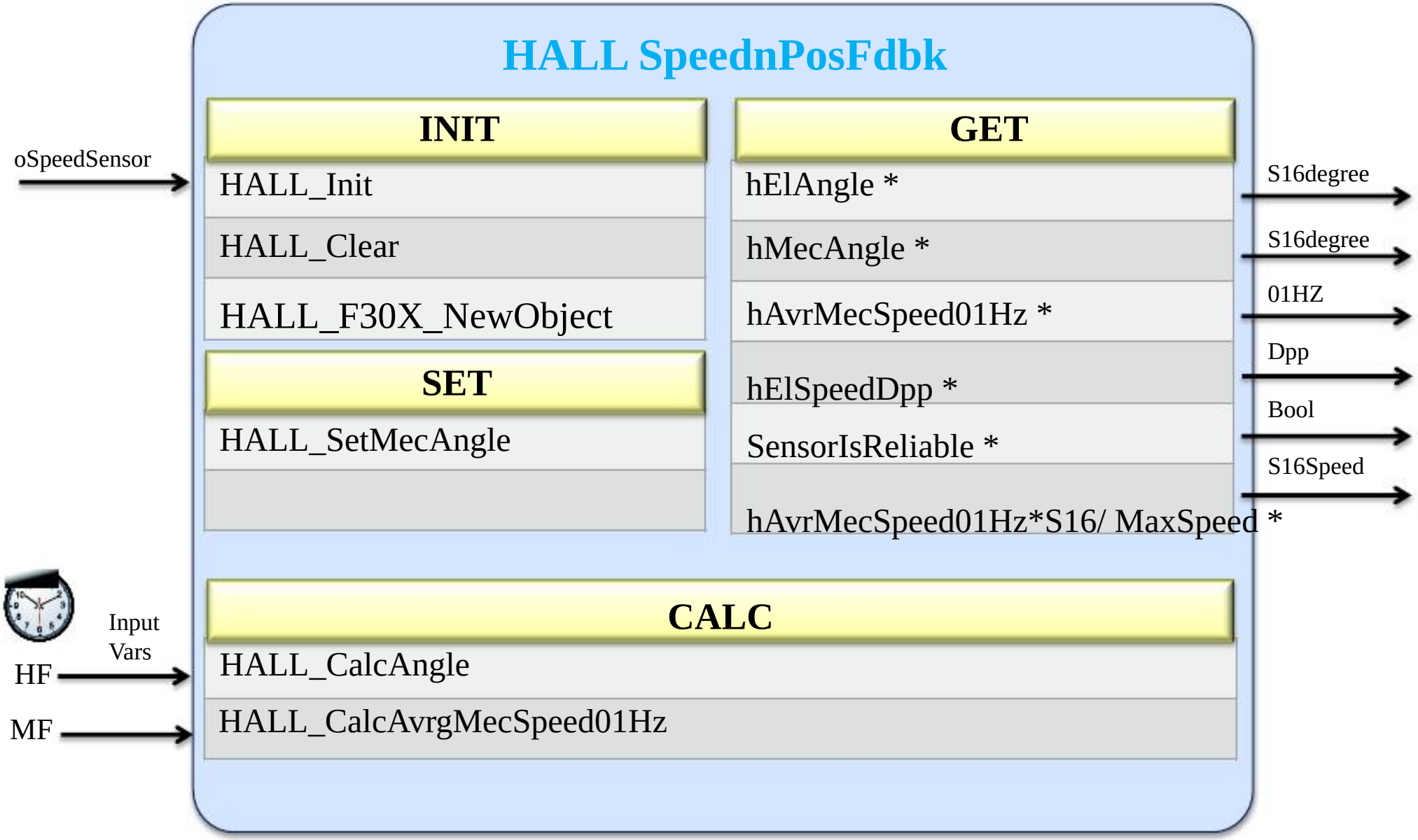
速度和位置反馈类



* XXX: Depend on the derived class

举例说明

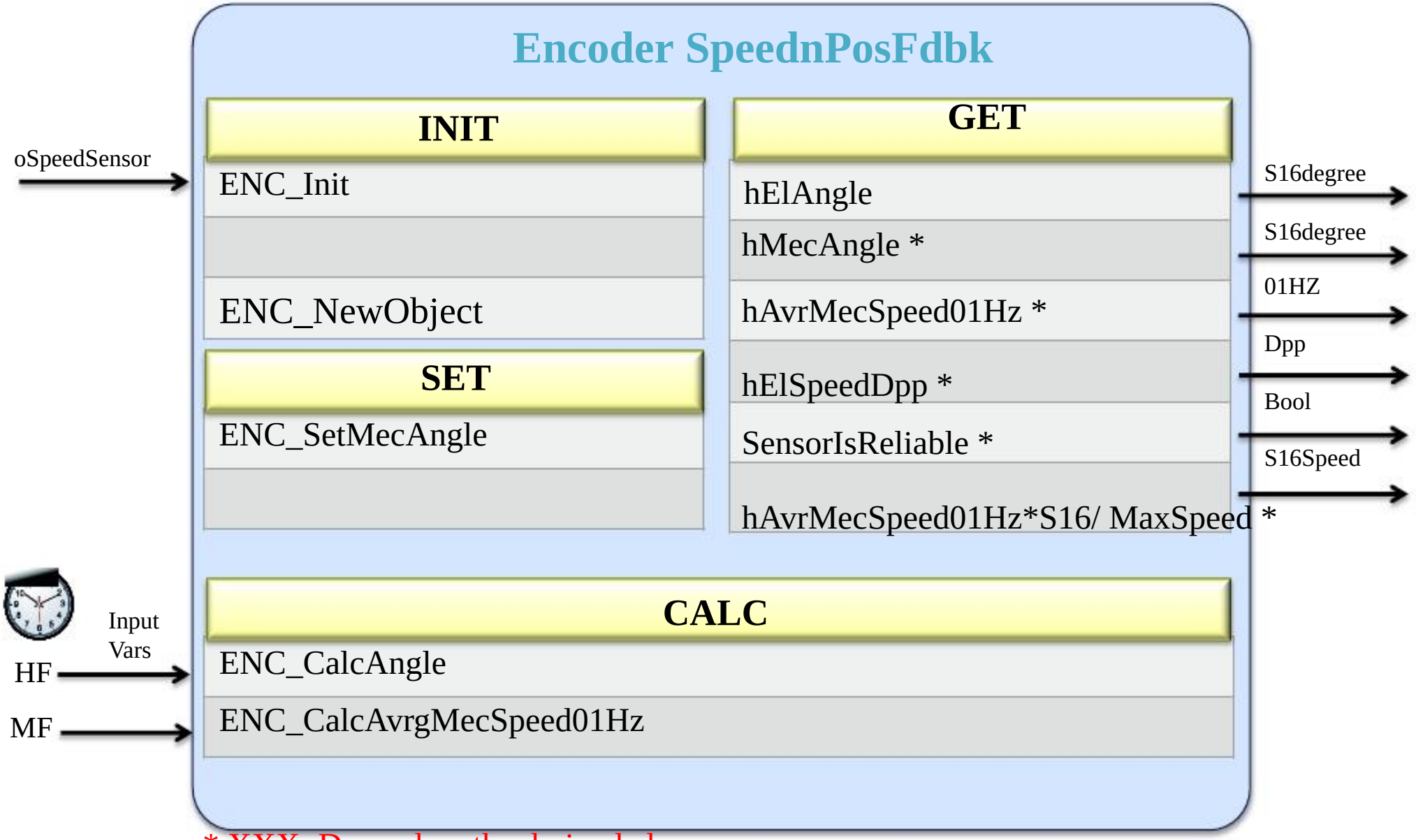
Hall传感器



*：返回的变量数据

举例说明

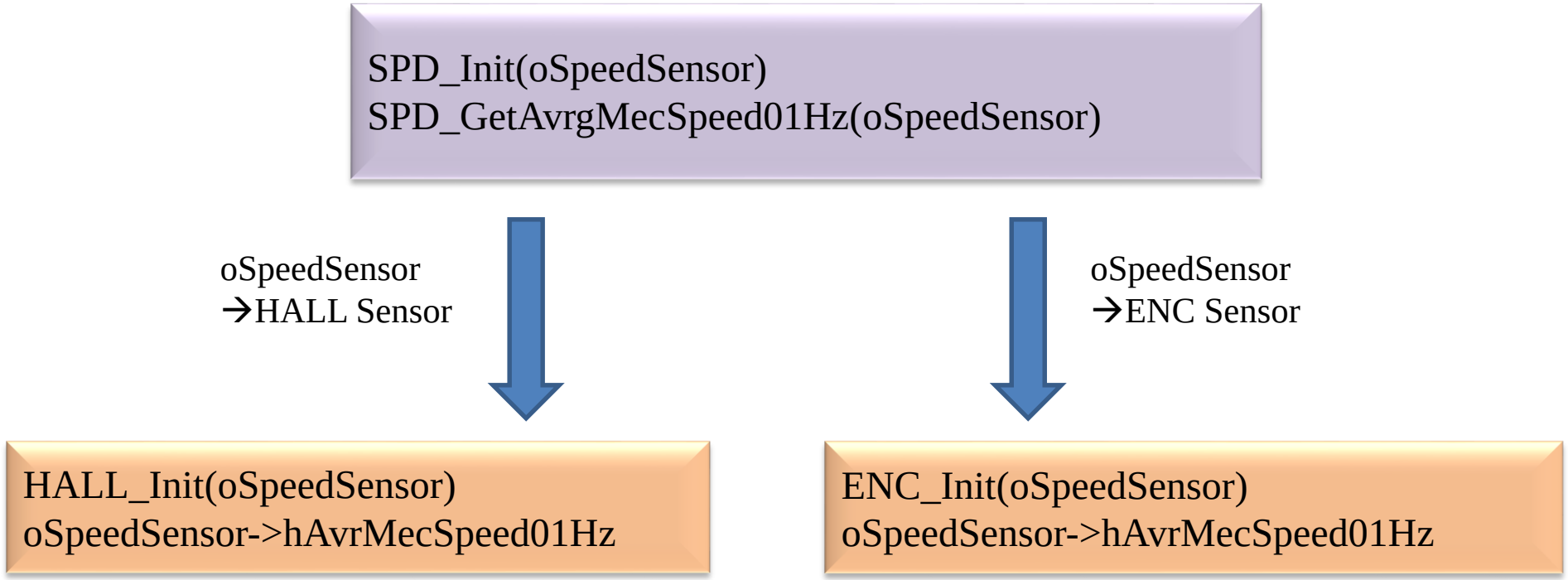
Encoder传感器



* XXX: Depend on the derived class

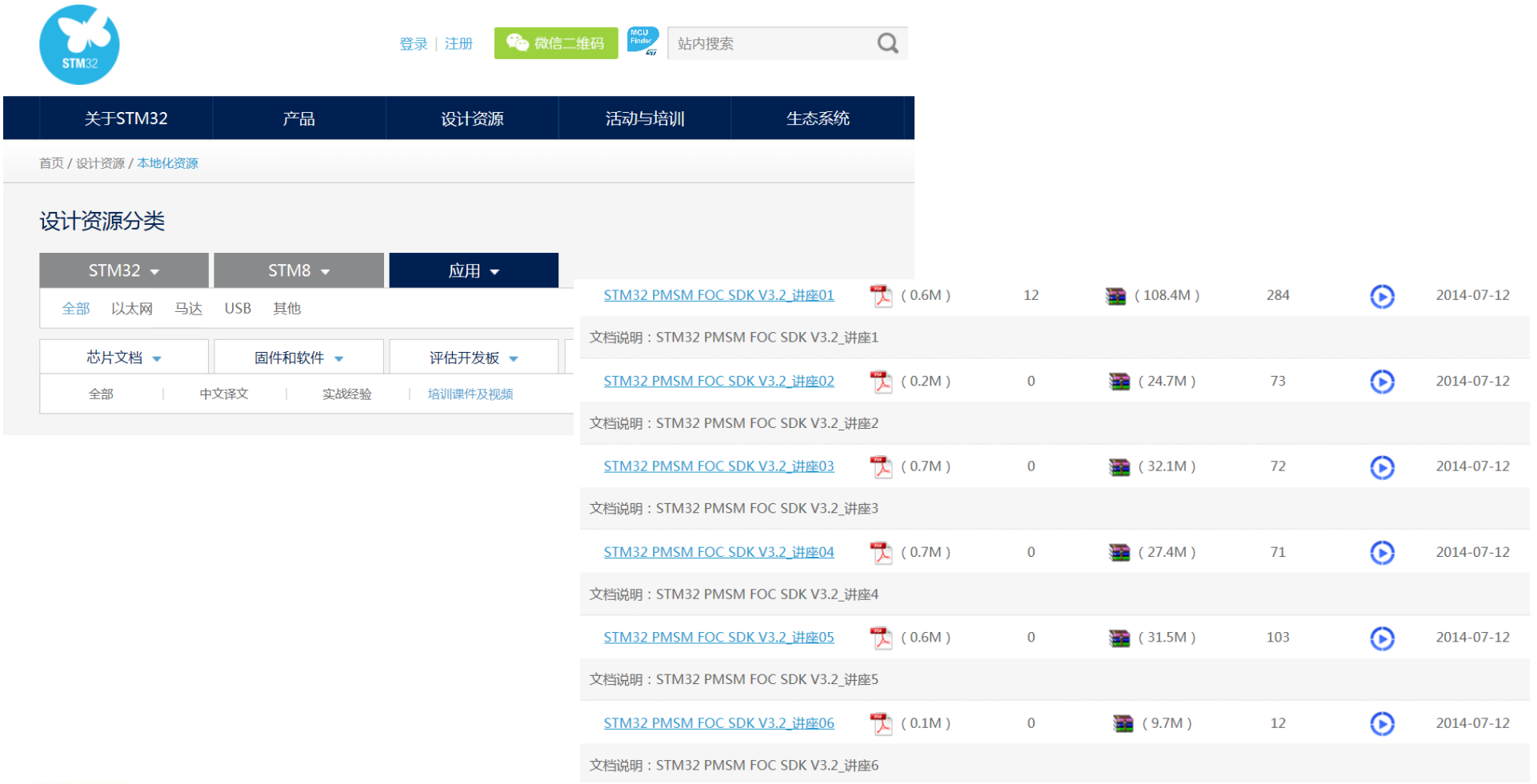
举例说明

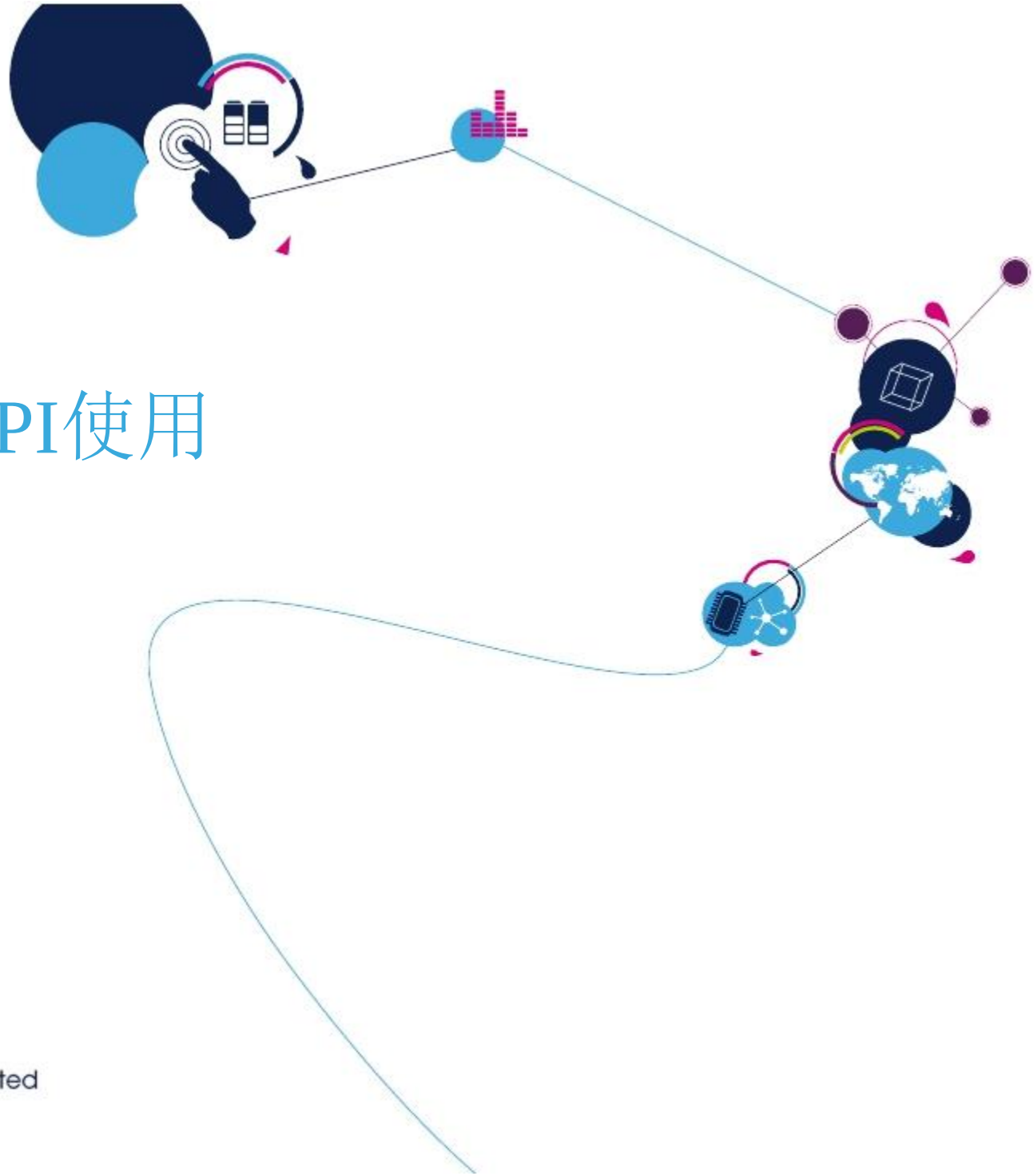
- Hall传感器类以及Encoder传感器为速度和位置反馈类的子类



详细介绍视频以及资料

http://www.stmcu.com.cn



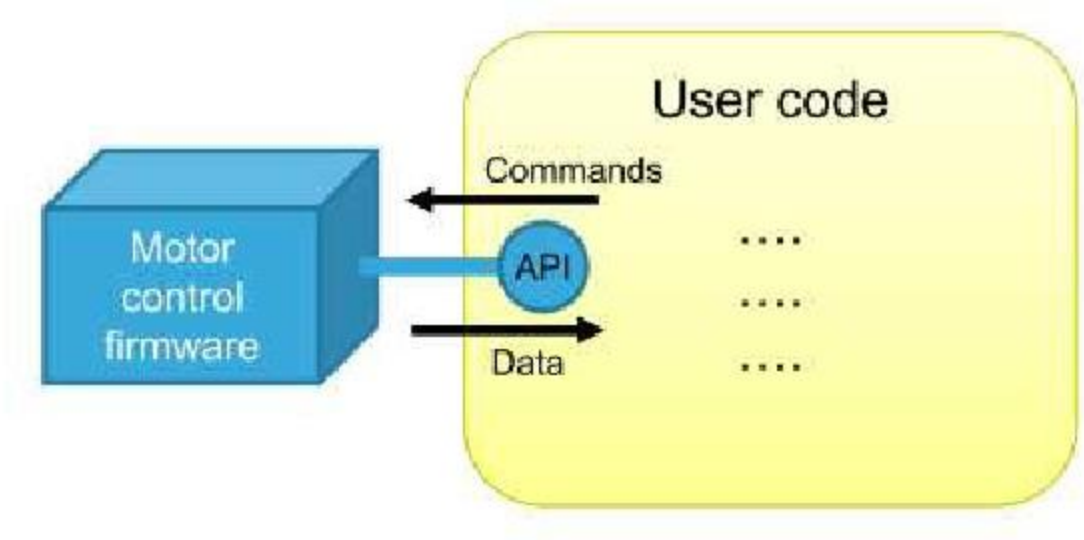


电机库API使用

life.augmented

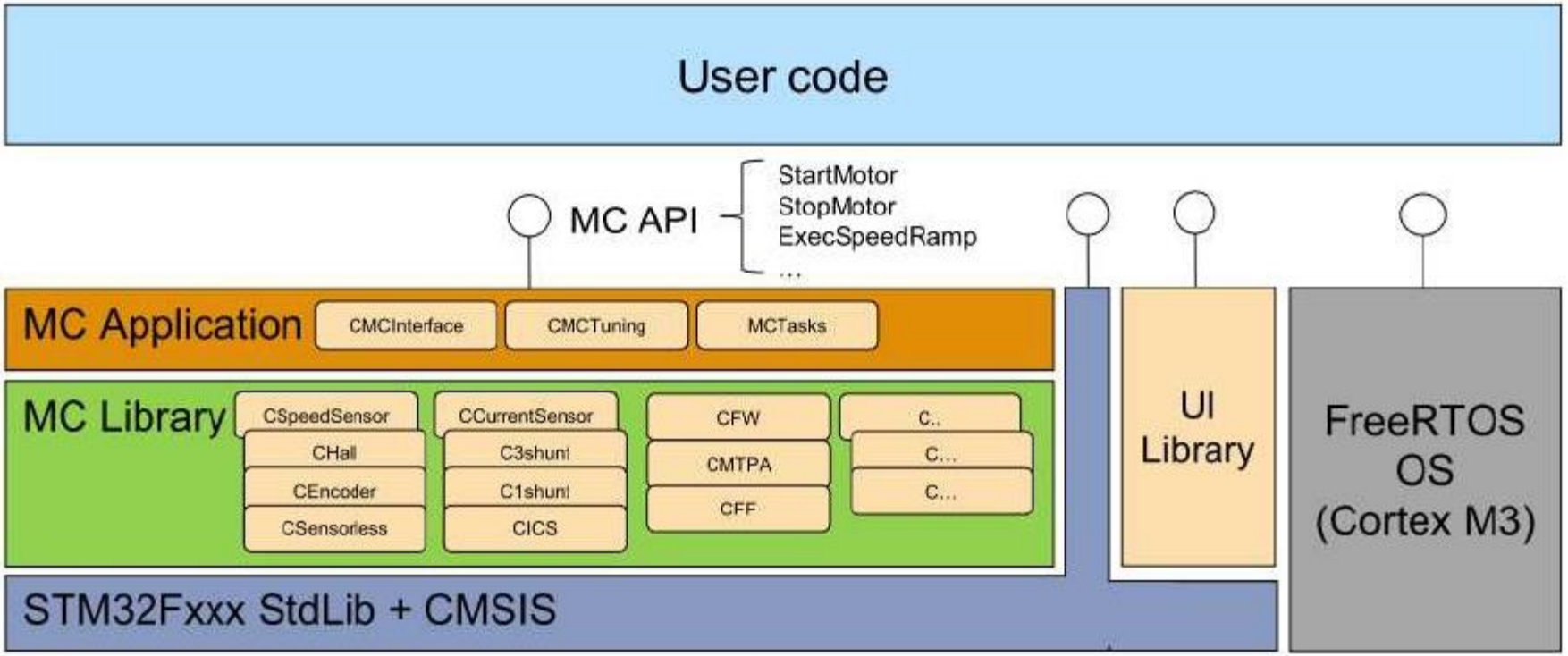
FOC SDK API

- 什么是API
  - API – Application Programming Interface：是SDK底层软件与用户软件的接口
  - 它提供了一个访问底层资源的工具
  - 它采用结构化的设计
  - 它允许软件开发者通过已定义好的数据结构来访问底层软件功能



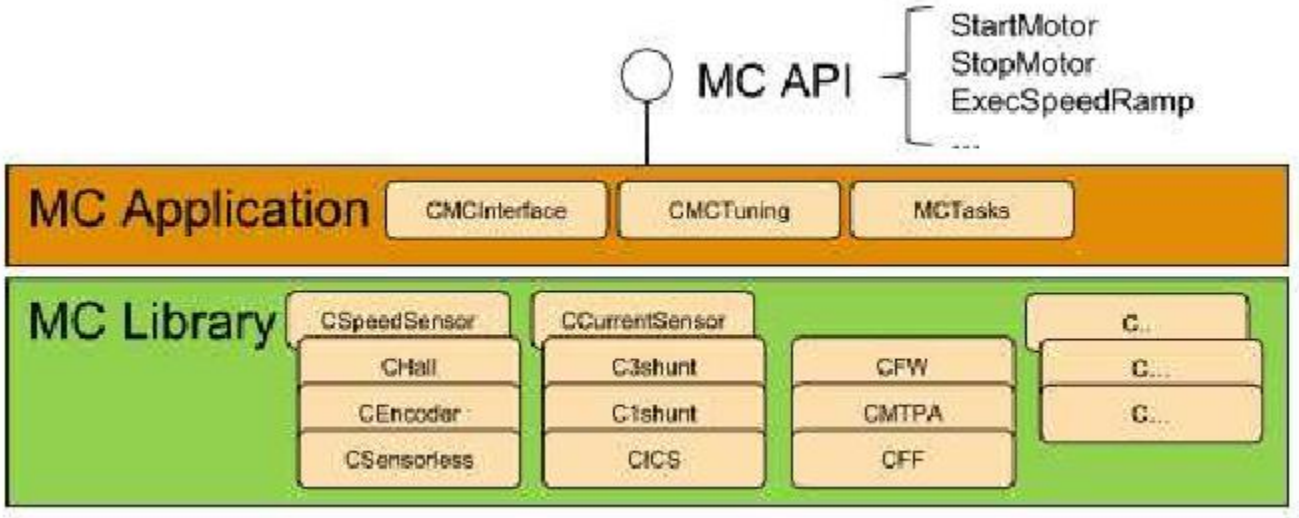
软件架构

- SDK软件库分为若干个软件层，分别为：MCU标准外设软件库、马达控制库及马达控制应用层
- 还包含用户界面层及FreeRTOS模块



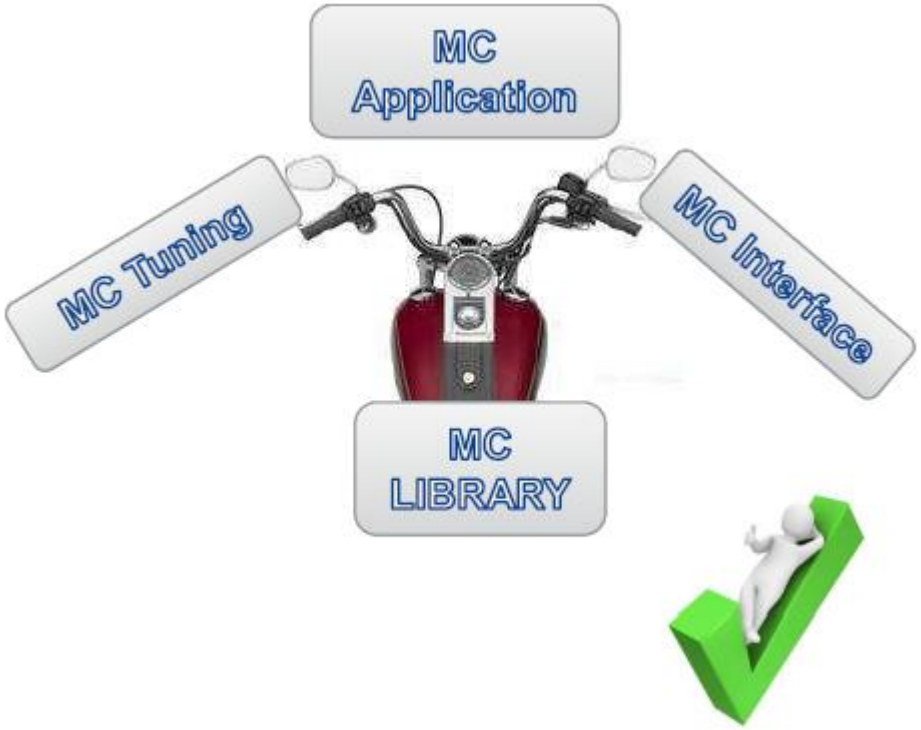
MC Applicatin层

- MC application是MC Library层的上一层，它主要执行如下任务
- MC Boot：根据用户在Workbench中的配置，实例化所有需要的对象
- MC Tasks：根据状态机管理所有的对象
- MC API：为用户提供访问MC Library层的API函数



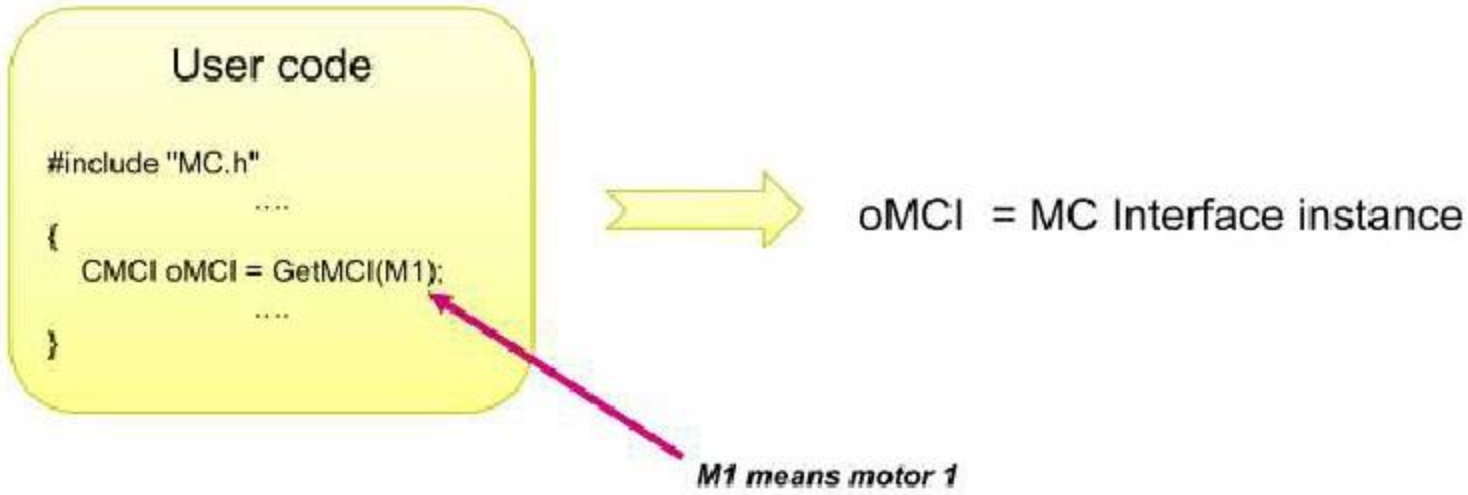
MC API

- MC API是基于SDK V4.3进行用户应用开发的起点，它主要分为2个部分：
  - MC Interface：电机控制接口类
  - MC Tuning：　电机控制调试类



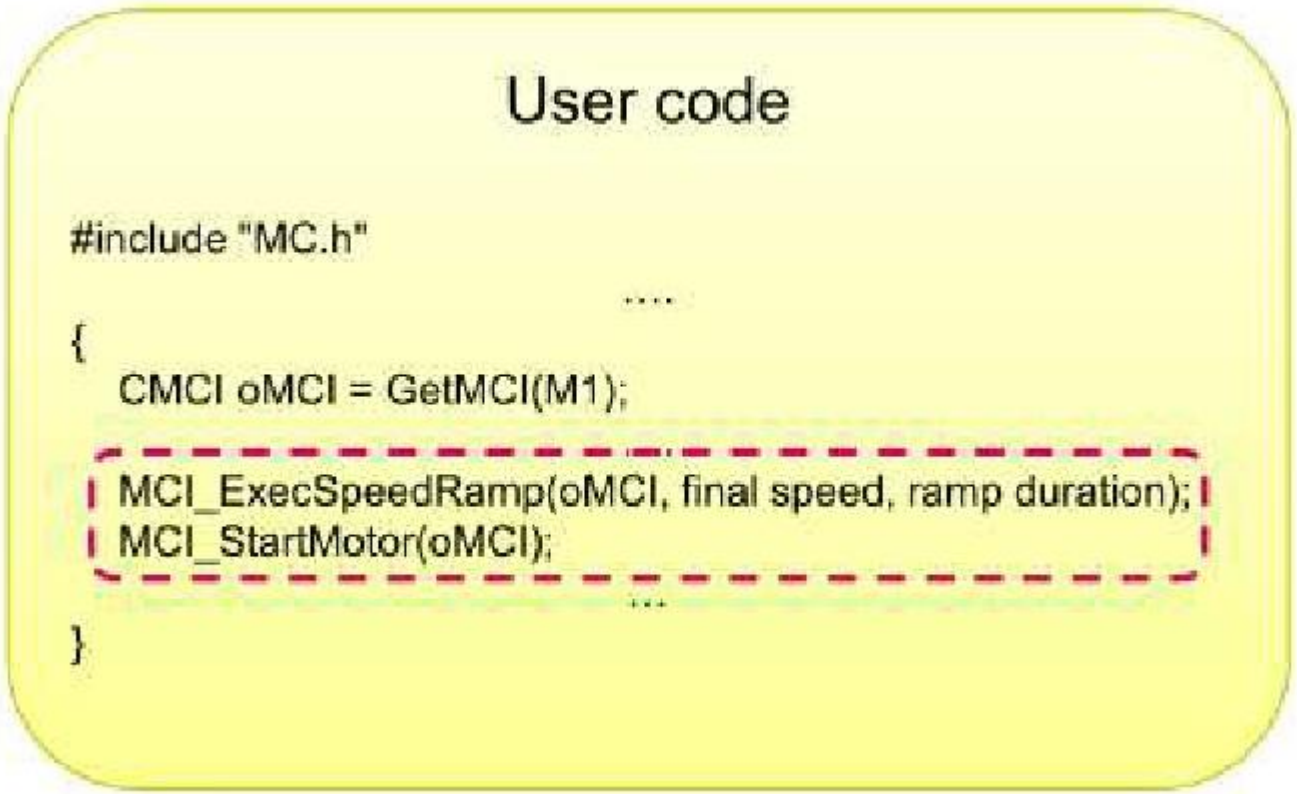
访问MC Interface

- #include "MC.h“
- CMCI oMCI：定义一个MCI对象
- oMCI = GetMCI(M1)：调用函数GetMCI()，使得oMCI指向目标马达的MCI对象



MC Interface函数

- 任何一个MCI的函数的第一个形参必须为oMCI



MC Interface命令

- MC Interface命令分为缓冲型的（Buffered）和非缓冲型的（Not Buffered）：
  - 缓冲型的命令在调用时并不马上执行，而是在状态机变为“RUN”时执行
  - 非缓冲型的命令在调用时，如果状态机允许的话马上执行

	Behavior	Example functions
Buffered commands	Command is buffered and executed when the state macchine reach the RUN state.	MCI_ExecSpeedRamp MCI_ExecTorqueRamp MCI_SetCurrentReferences
Not buffered commands	Command is executed instantaneously if the state machine is in the proper state otherwise it is discarded.	MCI_StartMotor MCI_StopMotor MCI_FaultAcknowledged

非缓冲型的命令

- 非缓冲型的命令：在状态机允许的条件下,用户需要立即执行的命令
  - 例如MCI_StartMotor()命令：如果状态机在真确的状态（IDLE），则会立即执行起动马达的过程
- 返回值：如果命令被执行则返回TRUE，否则返回FALSE
  - 例如MCI_StartMotor()命令：如果返回值为TRUE，只表示在执行该命令前马达处于IDLE状态，而并不是说执行该命令后马达已经在运行了
  - 如果要确认马达已经在运行了，则需要检查马达的状态机（RUN状态）

缓冲型的命令

- 缓冲型的命令只有在状态机处于RUN时才执行
  - 例如MCI_ExecSpeedRamp()：马达在停止状态时并不会得到执行
- 如果在马达的状态机变为RUN前调用了几个缓冲型的命令，则只有最后的一个才会得到执行
- 用户可以通过调用函数MCI_IsCommandAcknowledged()来检查该命令是否已得到执行

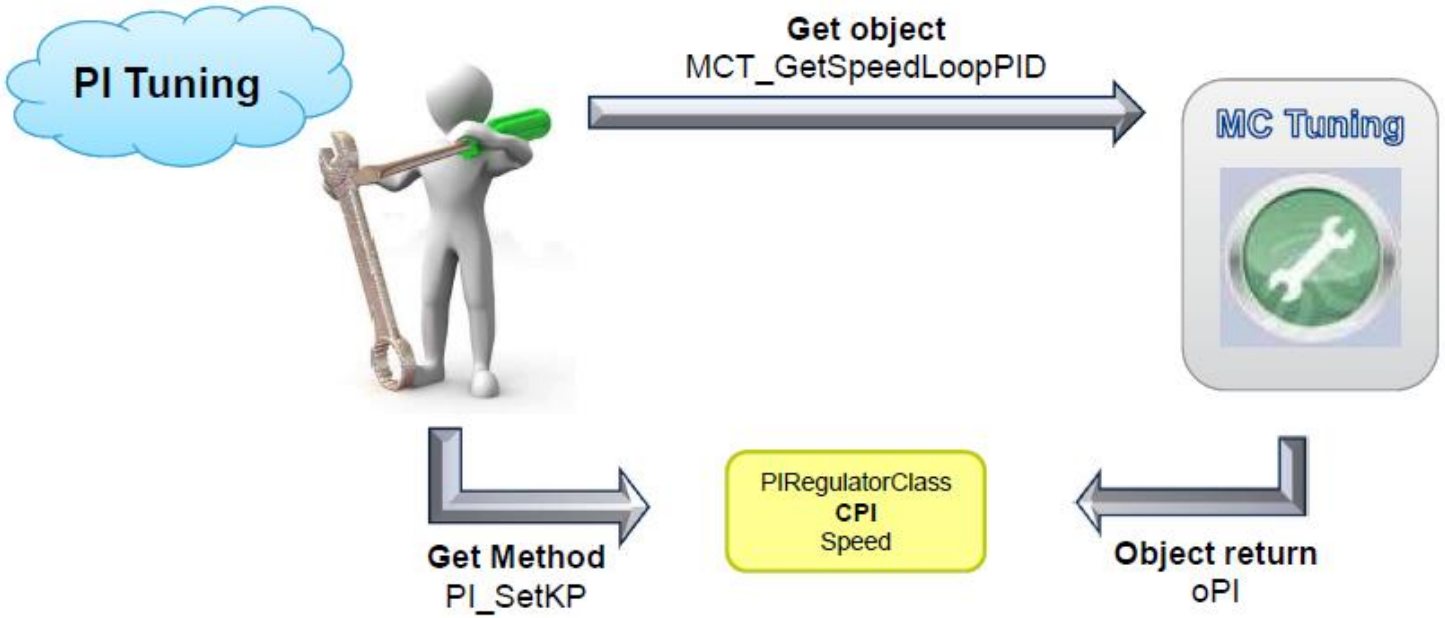
Return value	Description
MCI_BUFFER_EMPTY	No buffered command has been called.
MCI_COMMAND_NOT_ALREADY_EXECUTED	One buffered command has been called but RUN state hasn't yet reached.
MCI_COMMAND_EXECUTED_SUCCESFULLY	the buffered command has been executed successfully. In this case calling this function reset the buffered command state to MCI_BUFFER_EMPTY
MCI_COMMAND_EXECUTED_UNSUCCESFULLY	if the buffered command has been executed unsuccessfully. In this case calling this function reset the command state to MCI_BUFFER_EMPTY.

电机控制接口API举例

函数	描述
MCI_StartMotor	启动电机
MCI_StopMotor	停止电机
MCI_ExecSpeedRamp	更新目标速度
MCI_ExecTorqueRamp	更新目标转矩
MCI_SetCurrentReferences	设定参考电流Idref，Iqref
MCI_GetSTMState	返回状态机当前状态
MCI_GetAvrgMecSpeed01Hz	返回当前估算的平均机械转速
MCI_IsCommandAcknowledged	返回缓冲命令执行状态
……	……

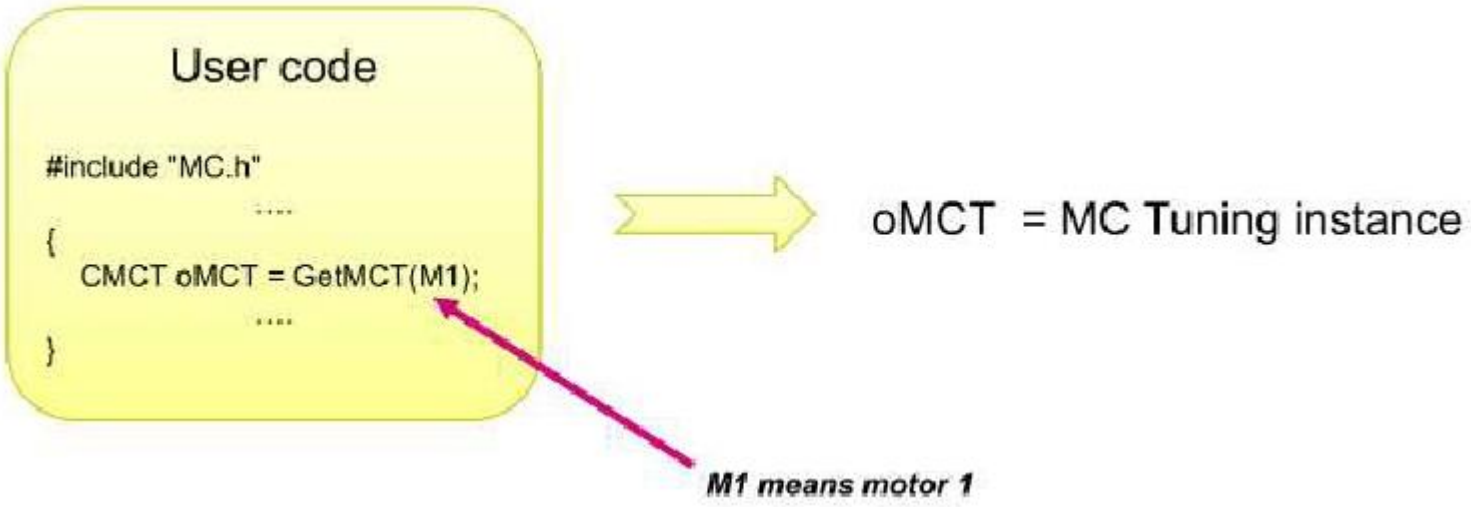
MC Tuning

- MC Tuning 类允许用户获得电机应用层的对象，是对象操作入口



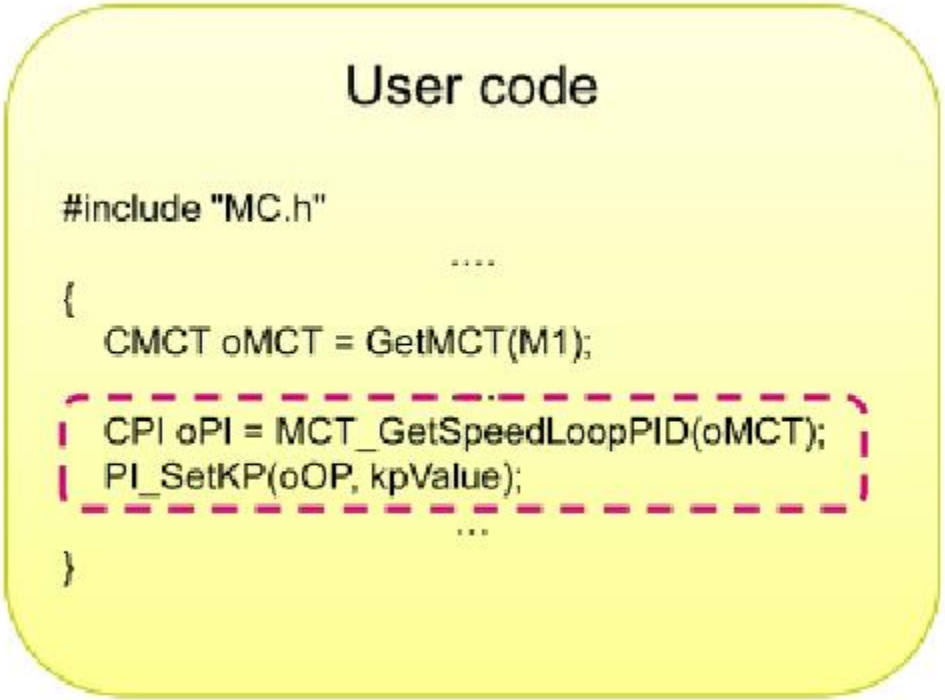
访问MC Tuning

- #include "MC.h“
- CMCT oMCT：定义一个MCT对象
- oMCT = GetMCT(M1)：调用函数GetMCT()，使得oMCT指向目标马达的MCT对象



操作MC Tuning

- 操作MC Tuning分2步进行：
  - 通过MC Tuning的函数得到目标对象
  - 调用该目标对象的操作函数



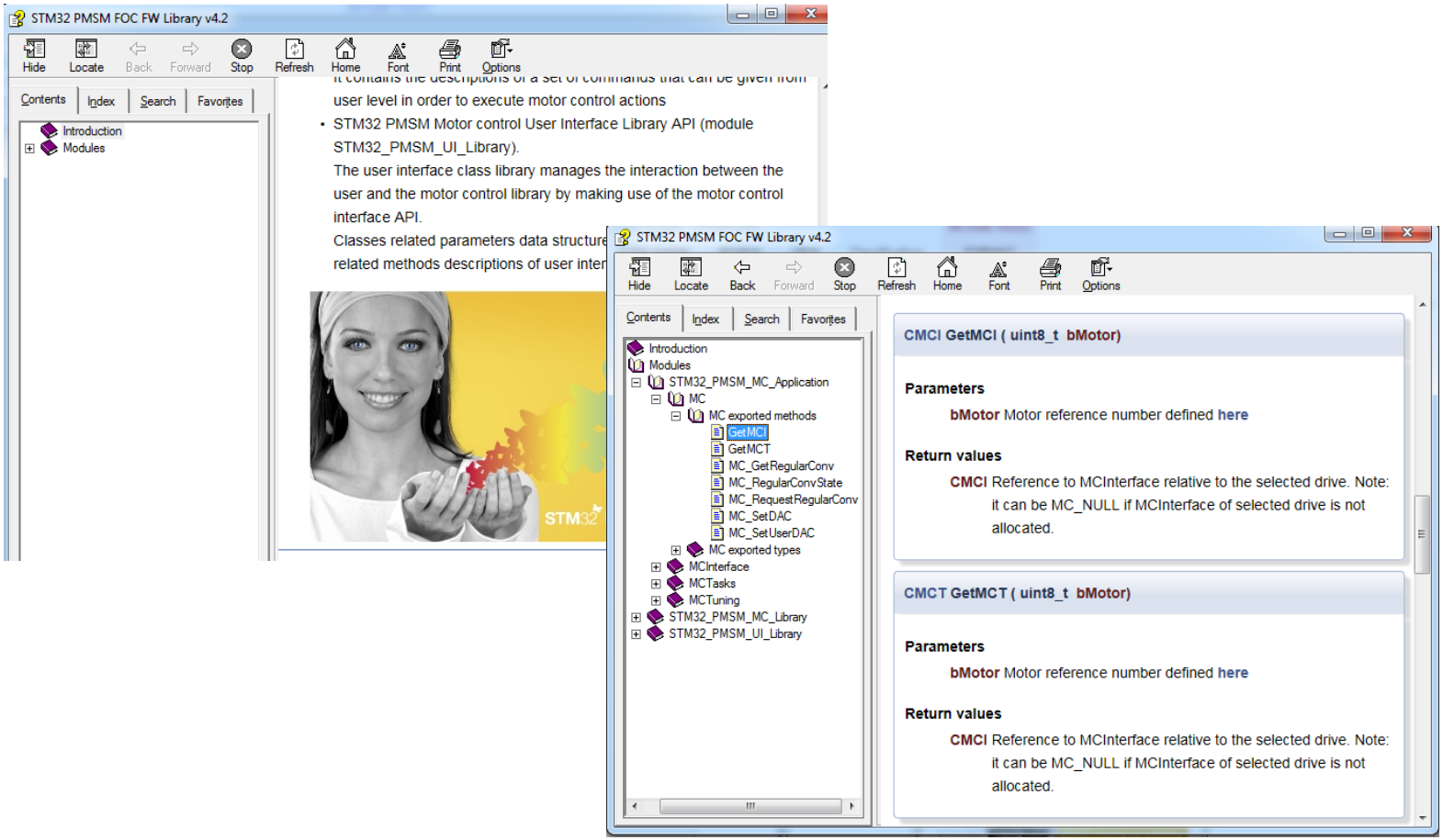
MC Tuning API

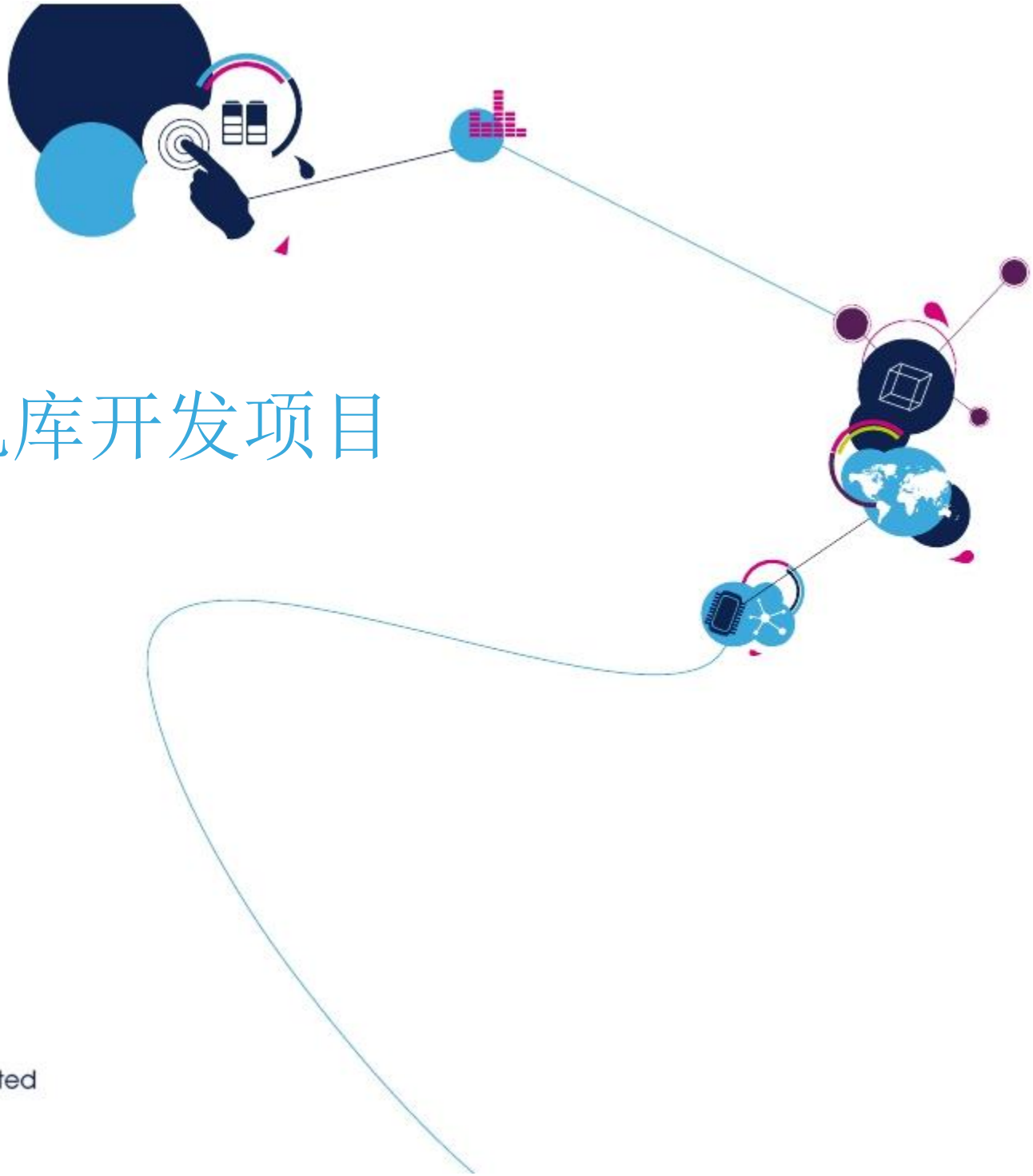
- 任何一个在MC Library层定义的对象都能在MC Tuning中通过相应的函数取得

MCT get object methods	Retrieved object	Type
MCT_GetSpeedLoopPID	Speed control loop PI(D)	CPI
MCT_GetIqLoopPID	Iq current control loop PI(D)	CPI
MCT_GetIdLoopPID	Id current control loop PI(D)	CPI
MCT_GetFluxWeakeningLoopPID	Flux Weakening control loop PI(D)	CPI
MCT_GetPWMnCurrFdbk	PWM generation and current feedback object	CPWMC
MCT_GetRevupCtrl	Rev-up controller for state observer sensor-less startup	CRUC
MCT_GetSpeednPosSensorMain	Main Speed'n Position sensor object.	CSPD
MCT_GetSpeednPosSensorAuxiliary	Auxiliary Speed'n Position sensor object.	CSPD
MCT_GetSpeednPosSensorVirtual	Virtual Speed'n Position sensor used for state observer sensor-less startup	CSPD
MCT_GetSpeednTorqueController	Speed'n Torque Controller	CSTC
MCT_GetStateMachine	State Machine	CSTM
MCT_GetTemperatureSensor	Temperature sensor	CSTM
MCT_GetBusVoltageSensor	Bus Voltage sensor	CVBS
...	...	...

MC API 帮助文件

- 所有的MCI及MCTAPI的具体介绍在SDK安装目录的Docs文件夹中

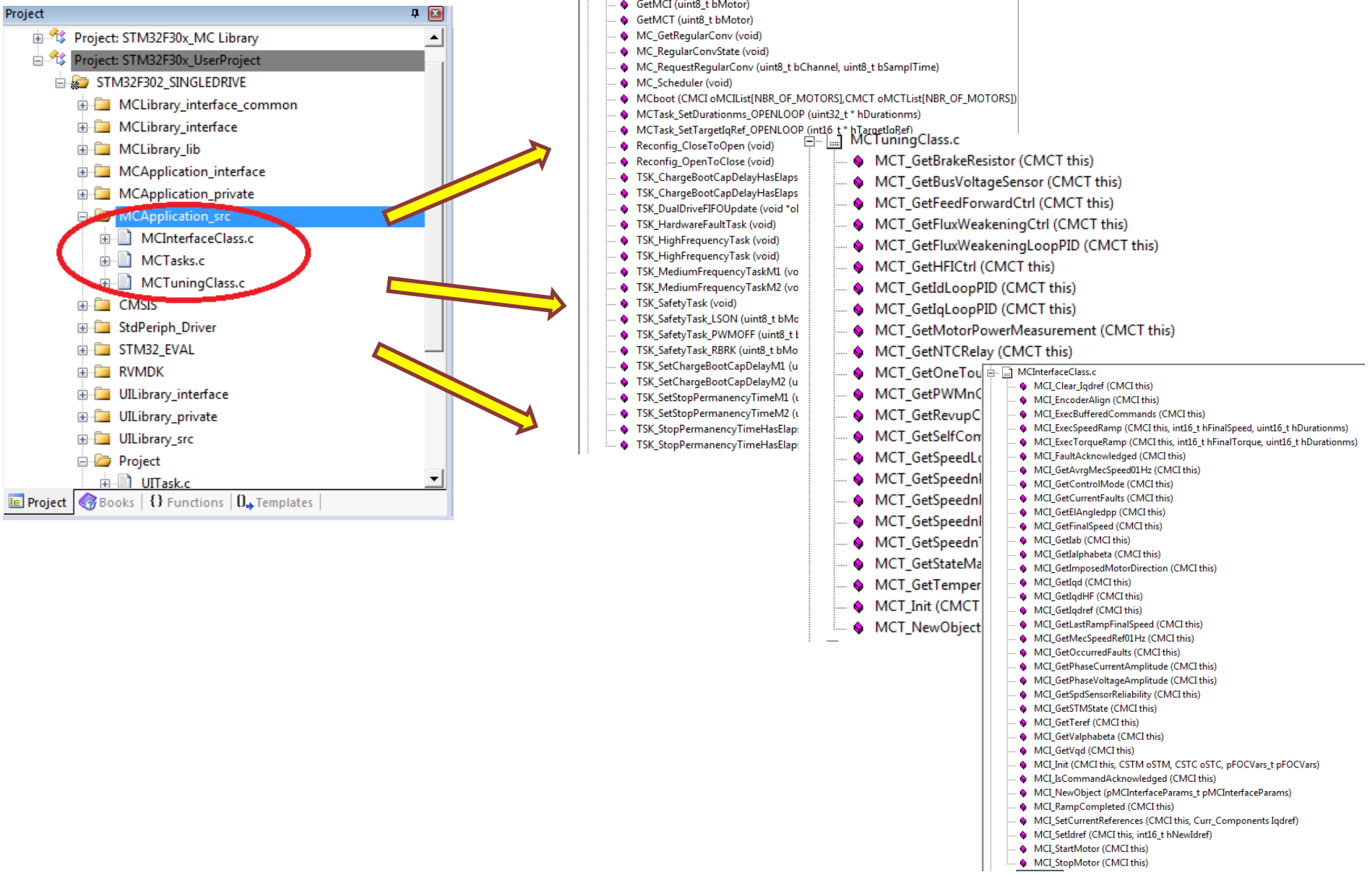




基于电机库开发项目

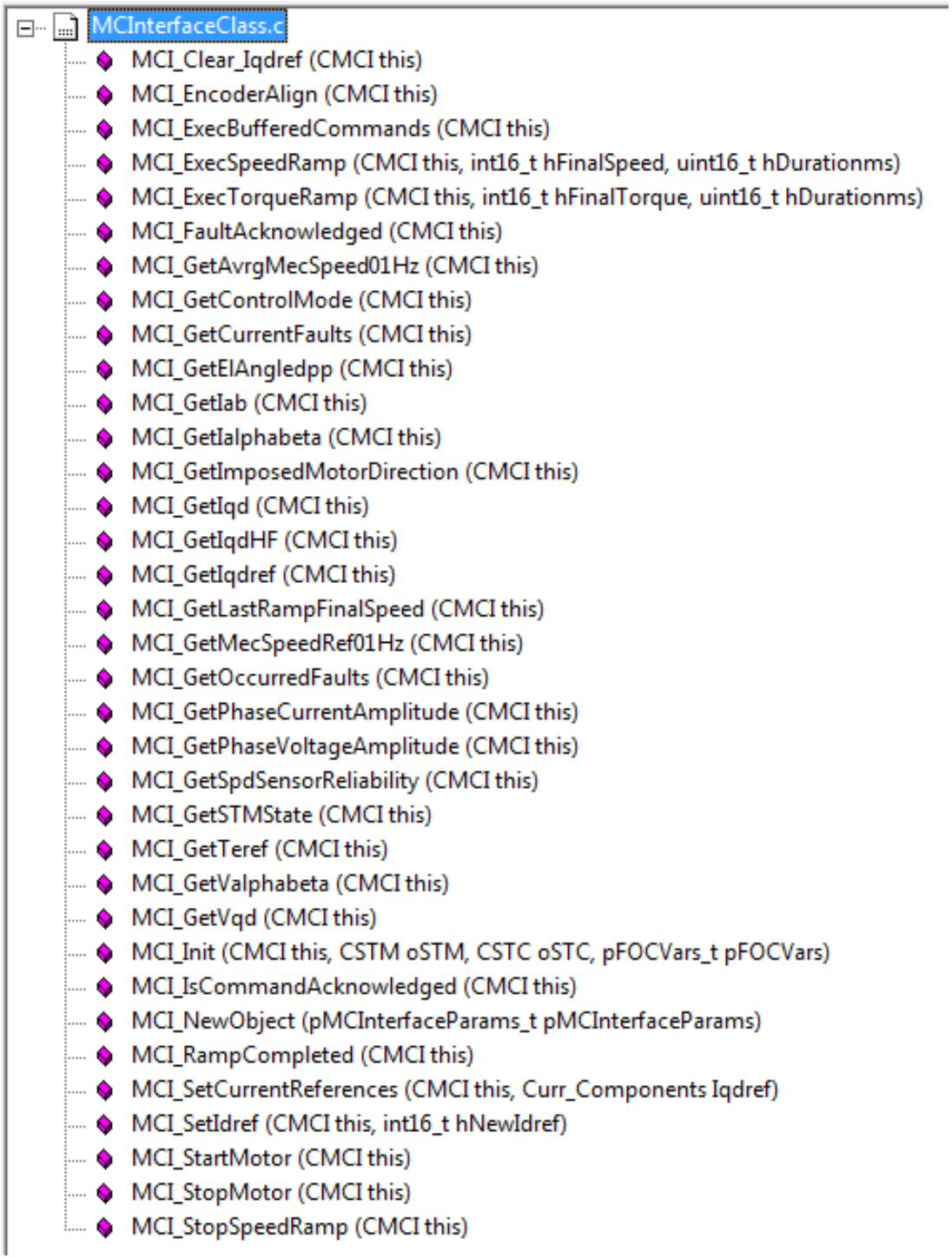
life.augmented

三个重要.c文件


Project
Project: STM32F30x_MC Library
Project: STM32F30x_UserProject
STM32F302_SINGLEDRIVE
MCLibrary_interface_common
MCLibrary_interface
MCLibrary_lib
MCApplication_interface
MCApplication_private
MCApplication_src
MCInterfaceClass.c
MCTasks.c
MCTuningClass.c
CMSIS
StdPeriph_Driver
STM32_EVAL
RVMDK
UILibrary_interface
UILibrary_private
UILibrary_src
Project
UITask.c
Project
Books
Functions
Templates
MCTasks.c
elif (defined(DUALDRIVE)&& (defined(HFINJECTION)|| defined(HFINJECTION2)))
FOC_CalcCurrRef (uint8_t bMotor)
FOC_Clear (uint8_t bMotor)
FOC_CurrController (uint8_t bMotor)
FOC_InitAdditionalMethods (uint8_t bMotor)
GetMCI (uint8_t bMotor)
GetMCT (uint8_t bMotor)
MC_GetRegularConv (void)
MC_RegularConvState (void)
MC_RequestRegularConv (uint8_t bChannel, uint8_t bSamplTime)
MC_Scheduler (void)
MCboot (CMCI oMCIList[NBR_OF_MOTORS],CMCT oMCTList[NBR_OF_MOTORS])
MCTask_SetDurationms_OPENLOOP (uint32_t * hDurationms)
MCTask_SetTargetIqRef_OPENLOOP (int16_t * hTargetIqRef)
Reconfig_CloseToOpen (void)
Reconfig_OpenToClose (void)
TSK_ChargeBootCapDelayHasElaps
TSK_ChargeBootCapDelayHasElaps
TSK_DualDriveFIFOUpdate (void *ol
TSK_HardwareFaultTask (void)
TSK_HighFrequencyTask (void)
TSK_HighFrequencyTask (void)
TSK_MediumFrequencyTaskM1 (vo
TSK_MediumFrequencyTaskM2 (vo
TSK_SafetyTask (void)
TSK_SafetyTask_LSON (uint8_t bMo
TSK_SafetyTask_PWMOFF (uint8_t b
TSK_SafetyTask_RBRK (uint8_t bMo
TSK_SetChargeBootCapDelayM1 (u
TSK_SetChargeBootCapDelayM2 (u
TSK_SetStopPermanencyTimeM1 (
TSK_SetStopPermanencyTimeM2 (
TSK_StopPermanencyTimeHasElap
TSK_StopPermanencyTimeHasElap
MCTuningClass.c
MCT_GetBrakeResistor (CMCT this)
MCT_GetBusVoltageSensor (CMCT this)
MCT_GetFeedForwardCtrl (CMCT this)
MCT_GetFluxWeakeningCtrl (CMCT this)
MCT_GetFluxWeakeningLoopPID (CMCT this)
MCT_GetHFICtrl (CMCT this)
MCT_GetIdLoopPID (CMCT this)
MCT_GetIqLoopPID (CMCT this)
MCT_GetMotorPowerMeasurement (CMCT this)
MCT_GetNTCRelay (CMCT this)
MCT_GetOneTou
MCT_GetPWMnC
MCT_GetRevupC
MCT_GetSelfCon
MCT_GetSpeedLo
MCT_GetSpeednl
MCT_GetSpeednl
MCT_GetSpeednl
MCT_GetSpeedn
MCT_GetStateMa
MCT_GetTemper
MCT_Init (CMCT
MCT_NewObject
MCInterfaceClass.c
MCI_Clear_Iqdref (CMCI this)
MCI_EncoderAlign (CMCI this)
MCI_ExecBufferedCommands (CMCI this)
MCI_ExecSpeedRamp (CMCI this, int16_t hFinalSpeed, uint16_t hDurationms)
MCI_ExecTorqueRamp (CMCI this, int16_t hFinalTorque, uint16_t hDurationms)
MCI_FaultAcknowledged (CMCI this)
MCI_GetAvrgMecSpeed01Hz (CMCI this)
MCI_GetControlMode (CMCI this)
MCI_GetCurrentFaults (CMCI this)
MCI_GetElAngledpp (CMCI this)
MCI_GetFinalSpeed (CMCI this)
MCI_GetIab (CMCI this)
MCI_GetIalphabeta (CMCI this)
MCI_GetImposedMotorDirection (CMCI this)
MCI_GetIqd (CMCI this)
MCI_GetIqdHF (CMCI this)
MCI_GetIqdref (CMCI this)
MCI_GetLastRampFinalSpeed (CMCI this)
MCI_GetMecSpeedRef01Hz (CMCI this)
MCI_GetOccurredFaults (CMCI this)
MCI_GetPhaseCurrentAmplitude (CMCI this)
MCI_GetPhaseVoltageAmplitude (CMCI this)
MCI_GetSpdSensorReliability (CMCI this)
MCI_GetSTMState (CMCI this)
MCI_GetTeref (CMCI this)
MCI_GetValphabeta (CMCI this)
MCI_GetVqd (CMCI this)
MCI_Init (CMCI this, CSTM oSTM, CSTC oSTC, pFOCVars_t pFOCVars)
MCI_IsCommandAcknowledged (CMCI this)
MCI_NewObject (pMCInterfaceParams_t pMCInterfaceParams)
MCI_RampCompleted (CMCI this)
MCI_SetCurrentReferences (CMCI this, Curr_Components Iqdref)
MCI_SetIdref (CMCI this, int16_t hNewIdref)
MCI_StartMotor (CMCI this)
MCI_StopMotor (CMCI this)

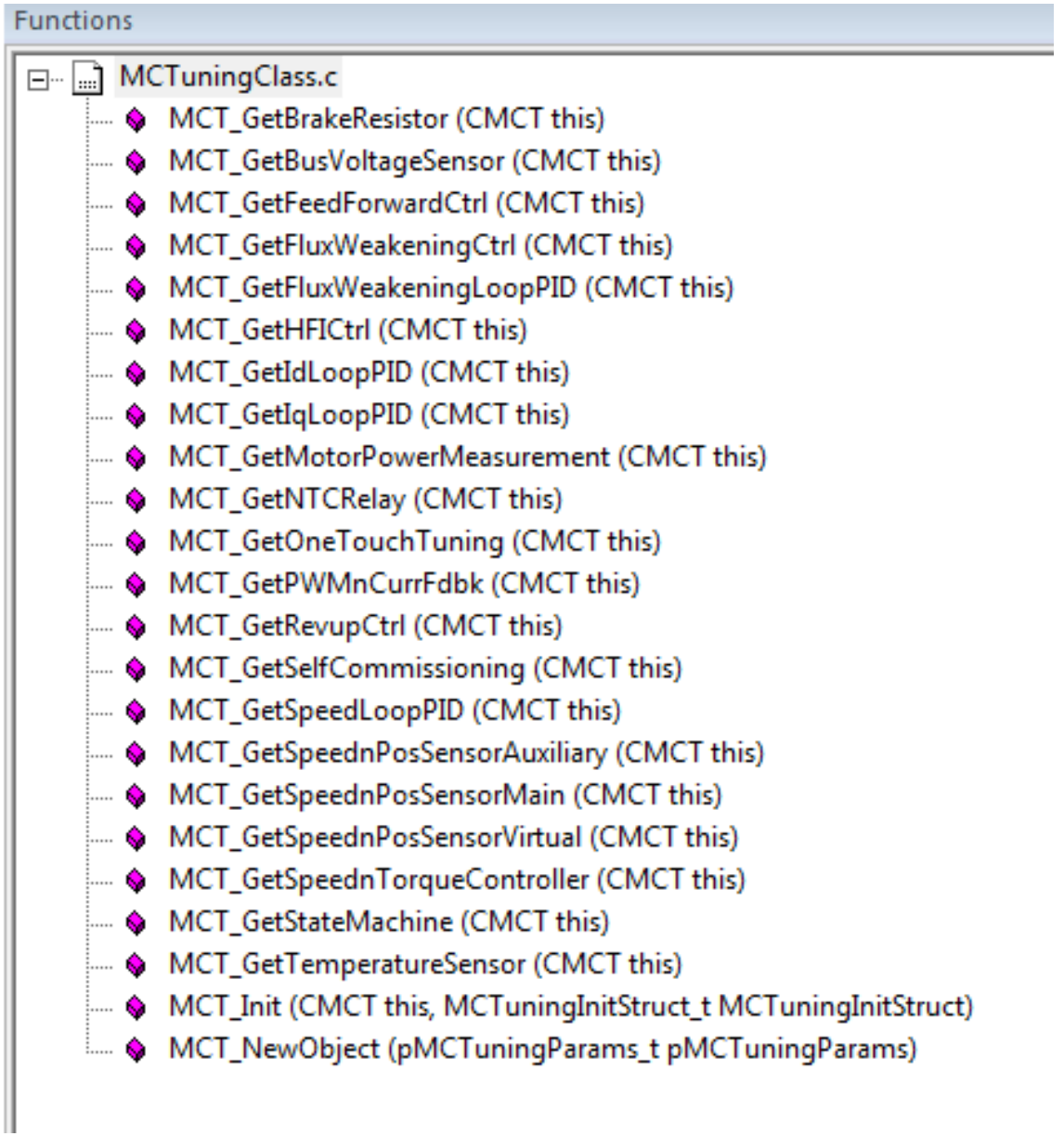
MCInterfaceClass.c

- 提供接口函数
- 获得各种控制变量函数
- 执行电机命令函数



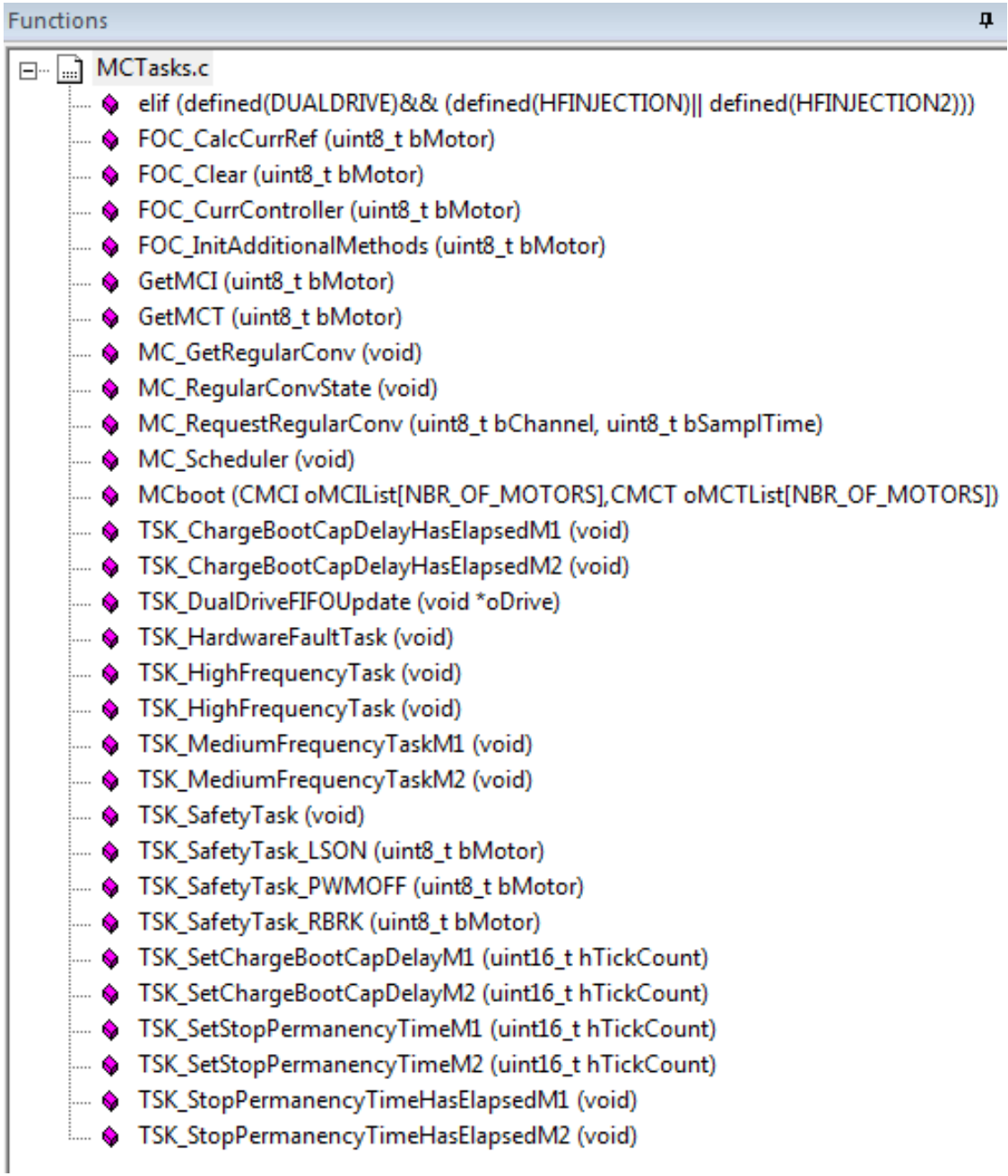
MCTuningClass.c

- MC Tuning函数
- 获得各成员控制的对象



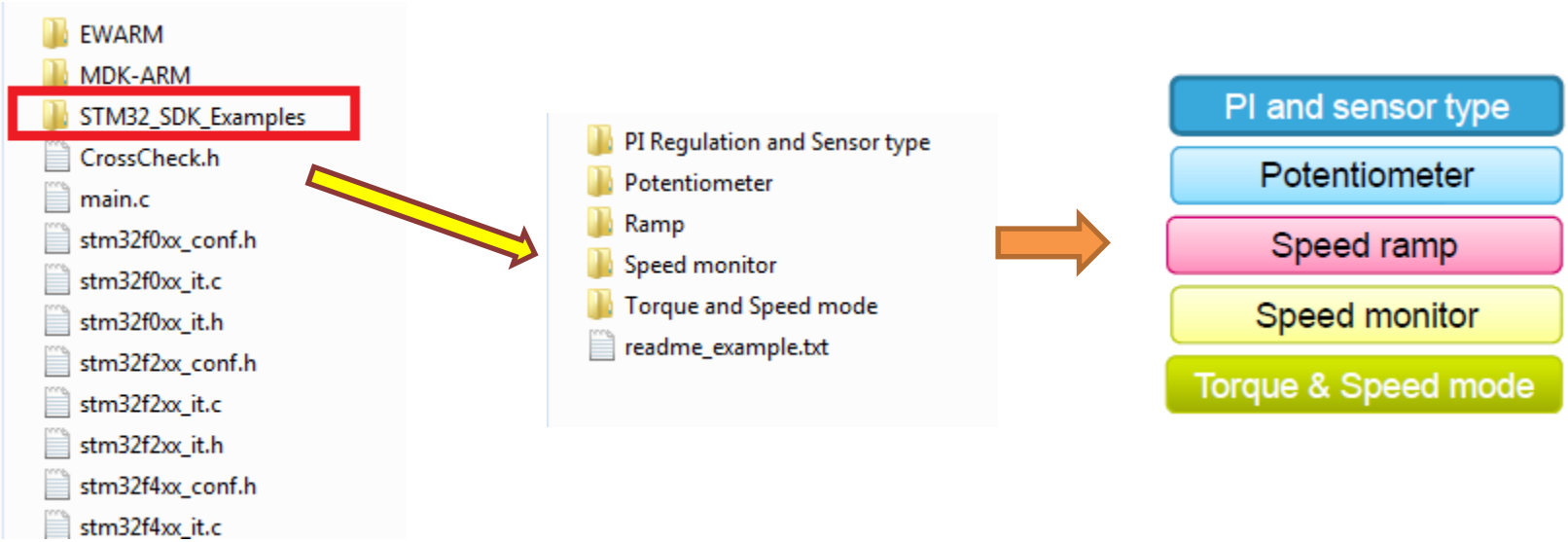
MCTask.c

- 过程控制的主要集中地
- 高，中，低时基控制具体实现
- 可根据实际情况添加程序



SDK提供参考示例代码

- 在…\Project下有示例代码文件夹，五个示例



补充说明

PI_Sensor_onFLY_F1.c | potentiometer_F1.c | ramp_F1.c | Speed monitor_F1.c | TqSpeedMode_F1.c | main.c

```
128         file (startup_stm32f10x_xx.s) before to branch to application main.
129         To reconfigure the default setting of SystemInit() function, refer to
130         system_stm32f10x.c file
131       */
132
133 #if !defined(STM32F0XX)
134   /*NVIC Priority group configuration.
135     Default option is NVIC_PriorityGroup_3.
136   */
137   NVIC_PriorityGroupConfig(NVIC_PriorityGroup_3);
138 #endif
139
140   /*MCInterface and MCTuning boot*/
141   MCboot(oMCI,oMCT);
142
143   #if defined(PFC_ENABLED)
144     PFC_Boot(oMCT[0],(CMCT)MC_NULL, (int16_t *)MC_NULL);
145   #endif
146
147   /*Systick configuration.*/
148   SysTick_Configuration();
149
150   /* Start here ***************************************************************/
151   /* GUI, this section is present only if LCD, DAC or serial communication is */
152   /* enabled.                                                                 */
153 #if (defined(LCD_FUNCTIONALITY) | defined(DAC_FUNCTIONALITY) | defined(SERIAL_COMMUNICATION))
154   UI_TaskInit(UI_INIT_CFG,wConfig,MC_NUM,oMCI,oMCT,s_fwVer);
155 #endif
156   /* End here*****************************************************************/
157
158   while(1)
159   {
160 #ifdef SERIAL_COMMUNICATION
161     /* Start here ***********************************************************/
162     /* GUI, this section is present only if serial communication is enabled.*/
163     if (UI_SerialCommunicationTimeOutHasElapsed())
164     {
```

- 在main.c中要有MCboot用于系统初始化
- SysTick时基配置

life.augmented

基于SDK用户开发程序步骤

1 声明一个数组，类型为CMCI（MC Interface类）

```
CMCI oMCI[MC_NUM];
```

MC Interface类　对象名称　驱动马达的数量（单马达/双马达）

2 声明一个数组，类型为CMCT（MC Tuning类）

```
CMCT oMCT[MC_NUM];
```

MC Tuning类　对象名称　驱动马达的数量（单马达/双马达）

3 开始Boot初始化

```
MCboot(oMCI,oMCT);
```

初始化整个马达控制

可以开始使用MC API命令了！！

基于SDK用户开发程序步骤

4 用户硬件外设，变量初始化

5 调用API进行马达控制的实体化操作

```
MCI_ExecSpeedRamp(oMCI[0], 600/6, 1000);
MCI_StartMotor(oMCI[0]);
MCI_StopMotor(oMCI[0]);

CPI xCPI = MCT_GetSpeedLoopPID (oMCT);
PI_SetKP(xCPI,newKPValue);
value_Speed_RPM = SPD_GetAvrgMecSpeed01Hz(xCSPD)*6;
```

操作总结

- 调用MC_ Interface函数用于电机行为控制
- 通过MC_Tuning函数得到目标对象
- 操作目标对象用于内部变量修改以及内部函数调用
- 可方便添加用户定义函数以及变量
- 使用者可更加关注实际应用行为

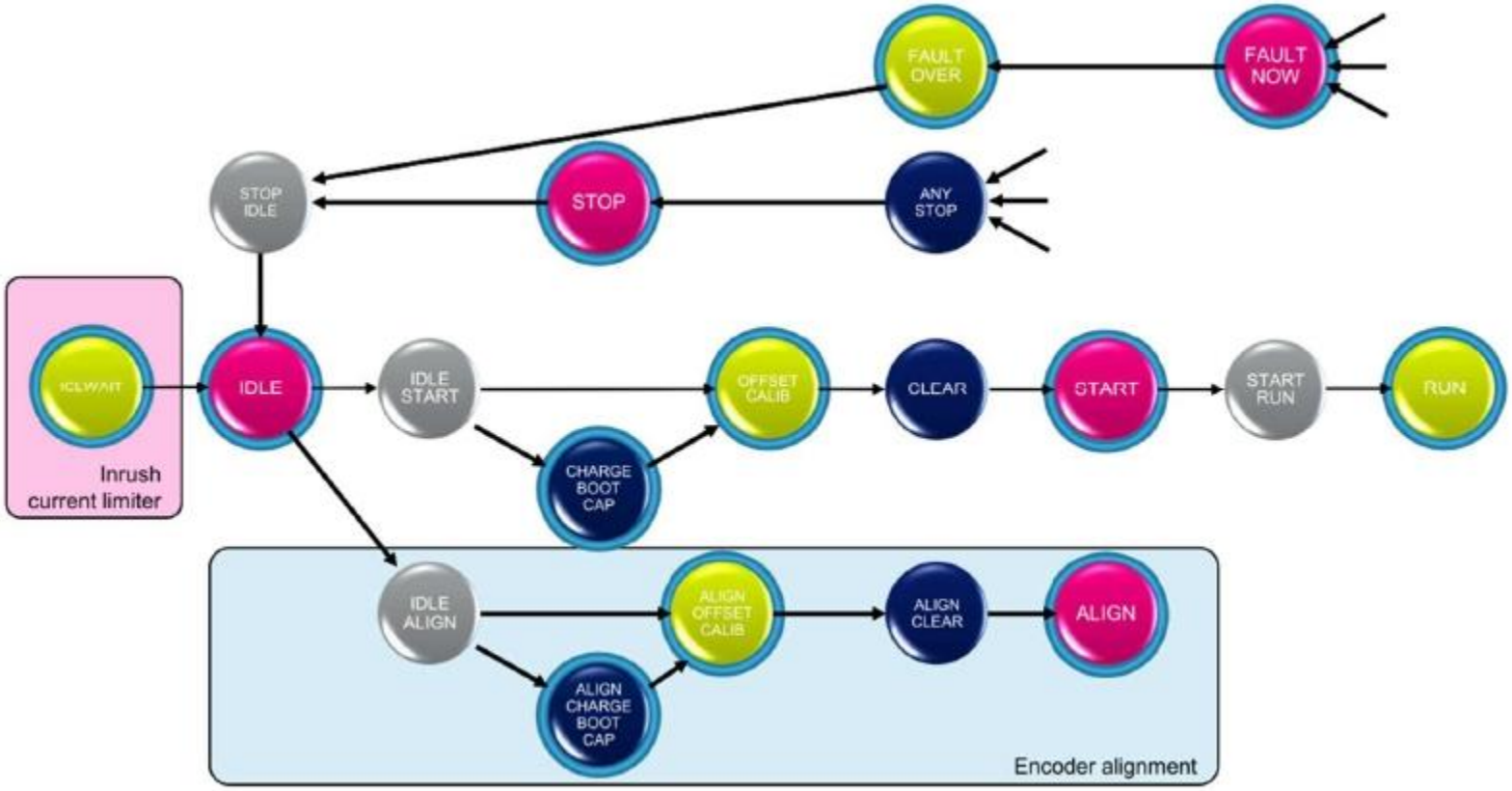
故障查询

```
#define MC_NO_ERROR          (uint16_t)(0x0000u)
#define MC_NO_FAULTS         (uint16_t)(0x0000u)
#define MC_FOC_DURATION      (uint16_t)(0x0001u)
#define MC_OVER_VOLT         (uint16_t)(0x0002u)
#define MC_UNDER_VOLT        (uint16_t)(0x0004u)
#define MC_OVER_TEMP         (uint16_t)(0x0008u)
#define MC_START_UP          (uint16_t)(0x0010u)
#define MC_SPEED_FDBK        (uint16_t)(0x0020u)
#define MC_BREAK_IN          (uint16_t)(0x0040u)
#define MC_SW_ERROR          (uint16_t)(0x0080u)
```

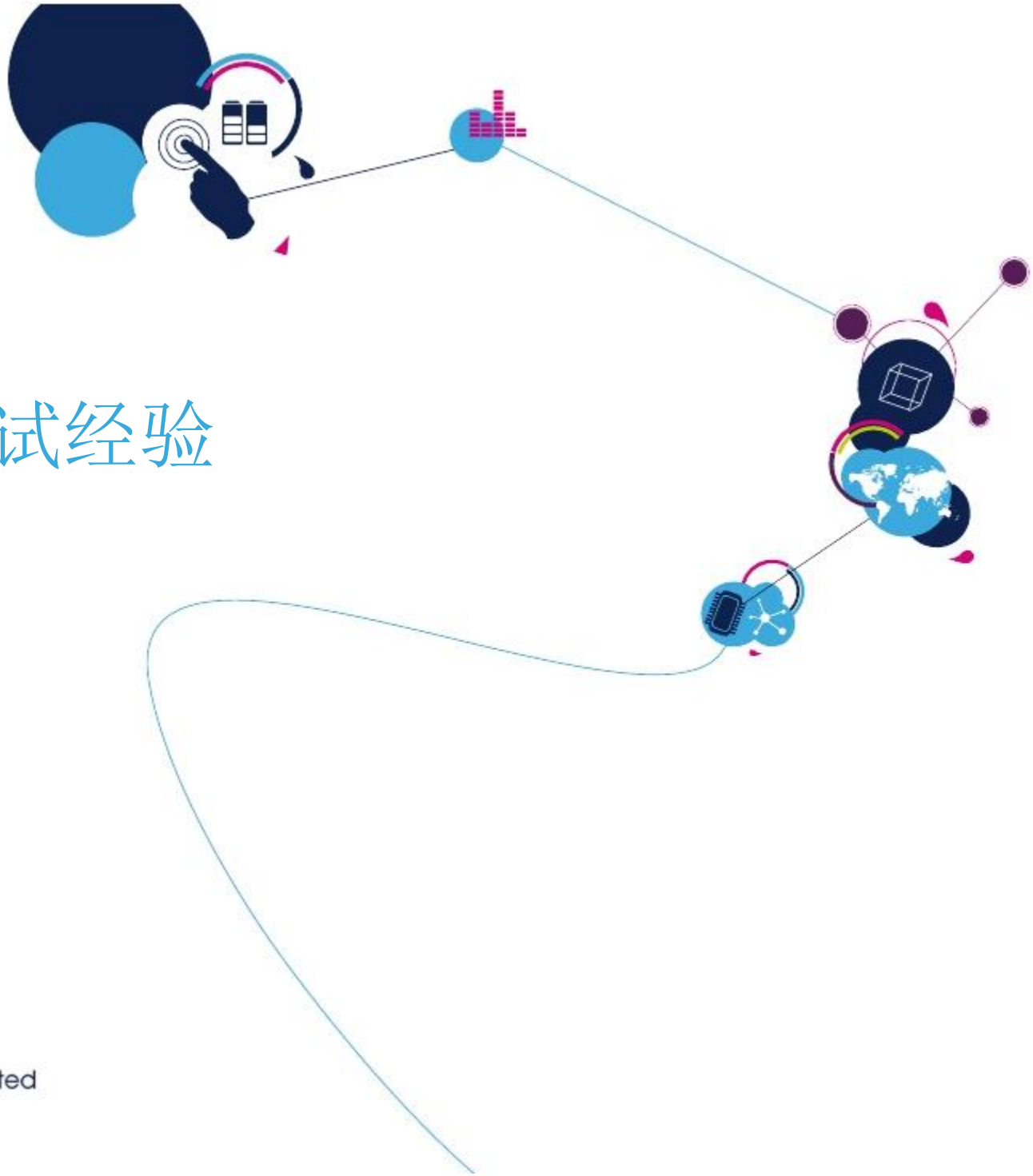
故障代码查询：

```
CSTM oSTM = MCT_GetStateMachine(oMCT[0]);

Fault_Type = (uint16_t)STM_GetFaultState(oSTM);
```

获得状态机




```
CSTM oSTM = MCT_GetStateMachine(oMCT[0]);

BSPState = (int32_t)STM_GetState(oSTM);
```



电机调试经验

life.augmented

电机调试技巧: 启动 1/2

- 首先，需要再次确认**ST MC 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: 启动 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)

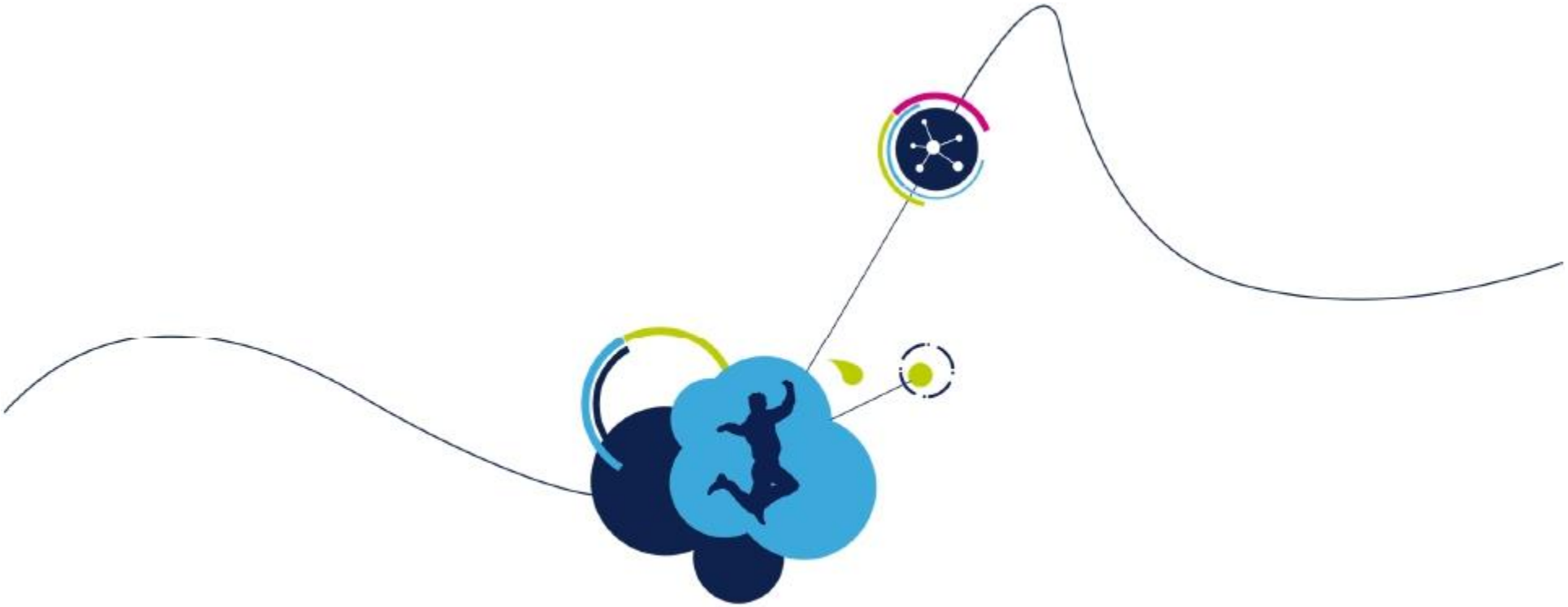
CH4: 相电流 (电流探头)

life.augmented

ST电机库编程试验

20~30min

life.augmented



谢谢！

life.augmented