



ST MC SDK 5.x WB应用指南和固件详解

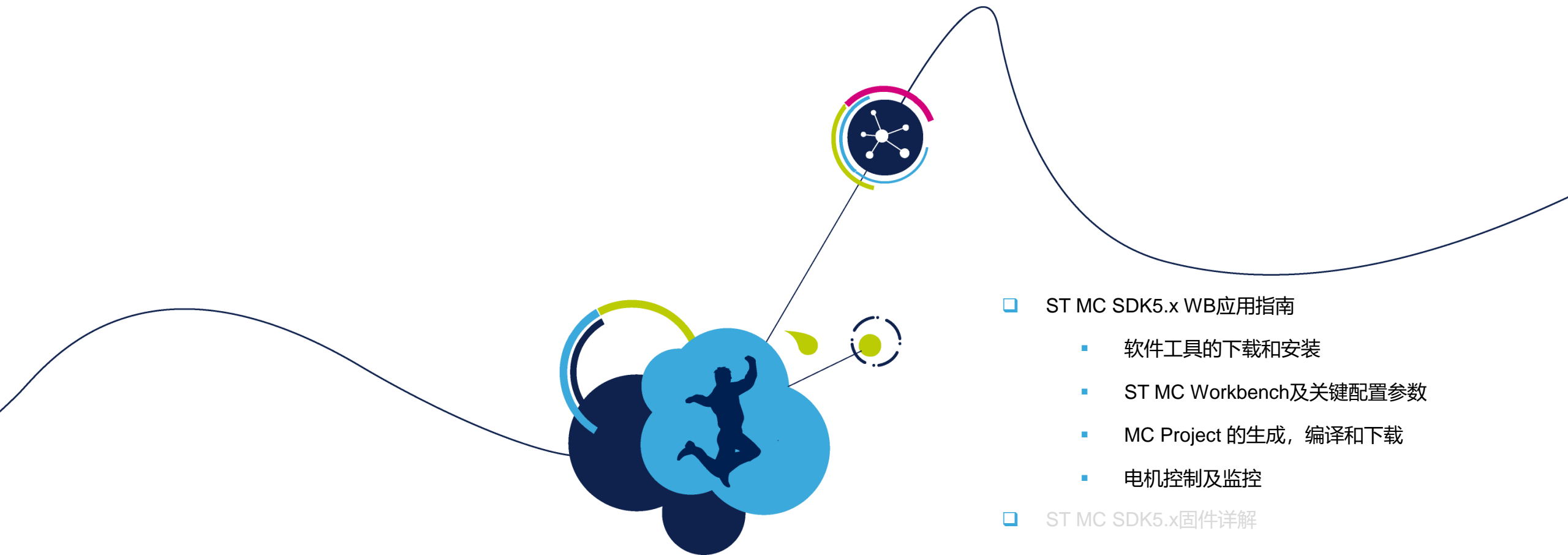
STM32电动机控制应用系列讲座之四

□ ST MC SDK5.x WB应用指南

- 软件工具的下载和安装
- ST MC Workbench及关键配置参数
- MC Project 的生成, 编译和下载
- 电机控制及监控

□ ST MC SDK5.x固件详解

- 程序架构
- 组件
- 例程代码讲解
- 开发实战
 - 如何向例程中添加外设和自己的代码
 - Step-by-Step添加一段闪灯代码




ST MC SDK5.x WB应用指南

- ST MC SDK5.x WB应用指南
 - 软件工具的下载和安装
 - ST MC Workbench及关键配置参数
 - MC Project 的生成, 编译和下载
 - 电机控制及监控

- ST MC SDK5.x固件详解
 - 程序架构
 - 组件
 - 例程代码讲解
 - 开发实战
 - 如何向例程中添加外设和自己的代码
 - Step-by-Step添加一段闪灯代码

请预先安装下列PC软件工具：

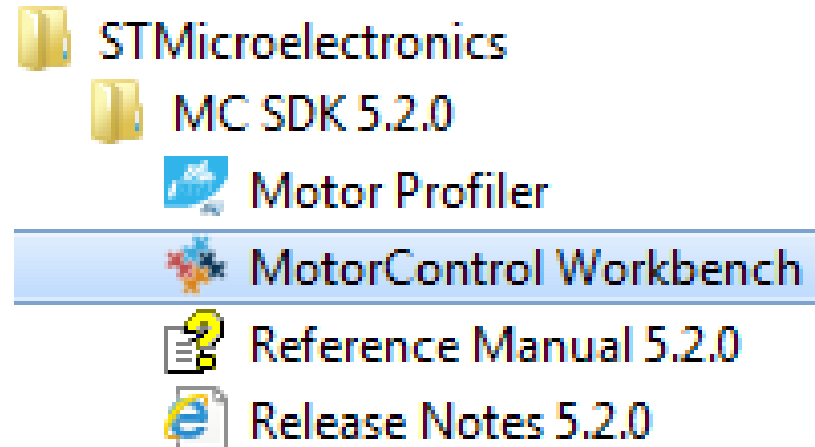
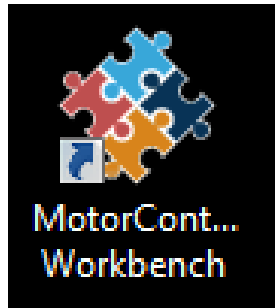
- X-CUBE-MCSDK 或 X-CUBE-MCSDK-FUL
- STM32CubeMX (v4.25.1/v4.26.1)及固件库
- ST-LINK/V2 (v4.2.0)
- IDE:
 - IAR Embedded Workbench for Arm (v7.80.4/v8.20.2)
 - μ Vision® IDE for Arm® (Keil® MDK) v5.25
 - Atollic TrueSTUDIO for STM32 version 9.0.0



STM32CubeMX请
不要安装在中文路
径下！！

下列方式可启动ST MC Workbench软件工具：

- 单击其图标
- 从安装文件夹路径直接启动



ST Motor Control Workbench

File Tools Help Documentation

New Project Load Project About Help 1 Motor Profiler
Motion Control Suite 1 2 3

Recent Projects

| Filename | Type | MCUs | control board | power board | motor |
|--|--------|----------------------------------|---|--|-----------|
| refrigerator solution f030-2018 0416.stmcx | SINGLE | STM32F030x | Custom | Custom | Custom |
| SDK43x-P-NUCLEO-IHM001-BullRunning.stmcx | SINGLE | STM32F301x6/8 - STM32F302x6/8 | P-NUCLEO-IHM001 3Sh - board: NUCLEO-F302R8 | P-NUCLEO-IHM001 3Sh - board: X-NUCLEO-IHM07M1 | Bull Runn |
| Noname.stmcx | SINGLE | STM32F072x | NUCLEO-F072RB | X-NUCLEO-IHM07M1 | BullRunni |
| Noname.stmcx | SINGLE | STM32F051x | Custom | Custom | Shinano I |
| refrigerator solution f030-2018 0416.stmcx | SINGLE | STM32F030x | Custom | Custom | Custom |
| P-NUCLEO-IHM001-BullRunning11.stmcx | SINGLE | STM32F301x6/8 - STM32F302x6/8 | P-NUCLEO-IHM001 3Sh - board: NUCLEO-F302R8 | P-NUCLEO-IHM001 3Sh - board: X-NUCLEO-IHM07M1 | Bull Runn |
| 0303shunt.stmcx | SINGLE | STM32F030x | NUCLEO-F030R8 | X-NUCLEO-IHM07M1 | BullRunni |

Example Projects

| Filename | Type | MCUs | control board | power board | motor |
|---|--------|--------------------------|-----------------|------------------|-----------|
| NUCLEO_L452RE_IHM07M1_SHINANO_1S_PLL | SINGLE | STM32L452xx | NUCLEO-L452RE | X-NUCLEO-IHM07M1 | Shinano I |
| NUCLEO_L452RE_IHM07M1_SHINANO_3S_PLL | SINGLE | STM32L452xx | NUCLEO-L452RE | X-NUCLEO-IHM07M1 | Shinano I |
| NUCLEO_L476RG_IHM07M1_SHINANO_3S_PLL | SINGLE | STM32L476xx | NUCLEO-L476RG | X-NUCLEO-IHM07M1 | Shinano I |
| NUCLEO-F746ZG_IHM07M1_SHINANO_3S_PLL | SINGLE | STM32F476xx | NUCLEO-F746ZG | X-NUCLEO-IHM07M1 | Shinano I |
| STM32F476_IHM07M1_1S_SHINANO_PLL | SINGLE | STM32L476xx | STM32L476G-EVAL | X-NUCLEO-IHM07M1 | Shinano I |
| NUCLEO-F303RE-X-NUCLEO-IHM16M1-GimBal | SINGLE | STM32F303xE | NUCLEO-F303RE | X-NUCLEO-IHM16M1 | GimBal |
| NUCLEO-F446RE_IHM07M1_BULLRUNNING_3S_PLL | SINGLE | STM32F446xC-xE | NUCLEO-F446RE | X-NUCLEO-IHM07M1 | BullRunni |
| NUCLEO-F446RE_IHM07M1_SHINANO_1S_CORDIC | SINGLE | STM32F446xC-xE | NUCLEO-F446RE | X-NUCLEO-IHM07M1 | Shinano I |
| NUCLEO-F446RE_IHM07M1_SHINANO_3S_CORDIC | SINGLE | STM32F446xC-xE | NUCLEO-F446RE | X-NUCLEO-IHM07M1 | Shinano I |
| NUCLEO_F103RB_LD_IHM07M1_SHINANO_1S_STO_PLL | SINGLE | STM32F103 Low Density | NUCLEO-F103RB | X-NUCLEO-IHM07M1 | Shinano I |
| NUCLEO_F103RB_MD_IHM07M1_SHINANO_3S_STO_PLL | SINGLE | STM32F103 Medium Density | NUCLEO-F103RB | X-NUCLEO-IHM07M1 | Shinano I |
| STM3210E-EVAL_IHM07M1_SHINANO_1S_ENC | SINGLE | STM32F103 High Density | STM3210E-EVAL | X-NUCLEO-IHM07M1 | Shinano I |
| STM3210E-EVAL_IHM07M1_SHINANO_1S_HALL | SINGLE | STM32F103 High Density | STM3210E-EVAL | X-NUCLEO-IHM07M1 | Shinano I |
| STM3210E-EVAL_IHM07M1_SHINANO_3S_ENC_FF | SINGLE | STM32F103 High Density | STM3210E-EVAL | X-NUCLEO-IHM07M1 | Shinano I |

life.augmented

1. 用户按钮区用于创建新项目，加载已有项目或启动ST电机参数测量工具。
2. 最近的项目区用于加载近期的项目。
3. 例程区用于加载项目示例。

New Project

1 **Application type**
Custom

2 **System**
 Single Motor Dual Motors

3 **Select Boards:** Inverter MC Kit Power & Control

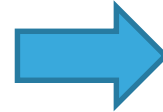
Control
custom board Control board where the control stage parameters have to customized by the user

Power
custom board Power board where the power stage parameters have to customized by the user

4 **Motor**
Generic Low voltage <= 50V
Motor low voltage

Magnetic structure Surface Mounted
Pole Pairs 2
Nominal Speed 4000 rpm
Nominal Voltage 24 V
Nominal Current 1.8 Apk

OK Cancel



New Project Info

i The motor was profiled with the following operation condition:

Bull Running BR2804-1700kv

| | |
|---------------------------------|--------------------------------------|
| Start up parameters | PWM Frequency: 30000 Hz |
| Nominal Current: 1.2 Apk | FOC Rate: 1 PWM periods |
| Nominal Voltage: 11 V | Cut off Frequency: 6000 rad/s |

These values have been imported into the project

OK

The screenshot displays the ST Motor Control Workbench interface. At the top, a menu bar (File, Tools, Help, Documentation) and a toolbar with various icons are visible. Below this, a status bar shows the current configuration: Motor: BullRunning, Control Board: P-NUCLEO-IHM001/002 3Sh, board: NUCLEO-F302R8, Power Board: P-NUCLEO-IHM001/002 3Sh, board: X-NUCLEO-IH. The main workspace contains a detailed circuit diagram of the motor control system, including an AC input stage with a PFC converter, a three-phase inverter, and a motor (M). Various sensing and protection blocks are connected to the system, such as Bus Voltage Sensing, Temperature Sensing, Current Sensing, and Over Current Protection. A 'Control Unit' block is shown on the left, containing sub-modules like Firmware Drive Management, MCU and Clock Freq., Digital I/O, DAC functionality, and Analog Input and Protection. A 'User Interface' block is also present. On the right side of the workspace, there are several checkboxes for enabling features like Bus Voltage Sensing, Dissipative Brake, Temperature Sensing, Current Sensing, Over Current Protection, and Speed Sensing. At the bottom, there are two panels: '主要的硬件配置' (Main Hardware Configuration) and '用户信息' (User Information). The '主要的硬件配置' panel shows a table of variables and their values. The '用户信息' panel shows a log of messages.

图标和菜单区 (Icon and Menu Area)

硬件信息 (Hardware Information)

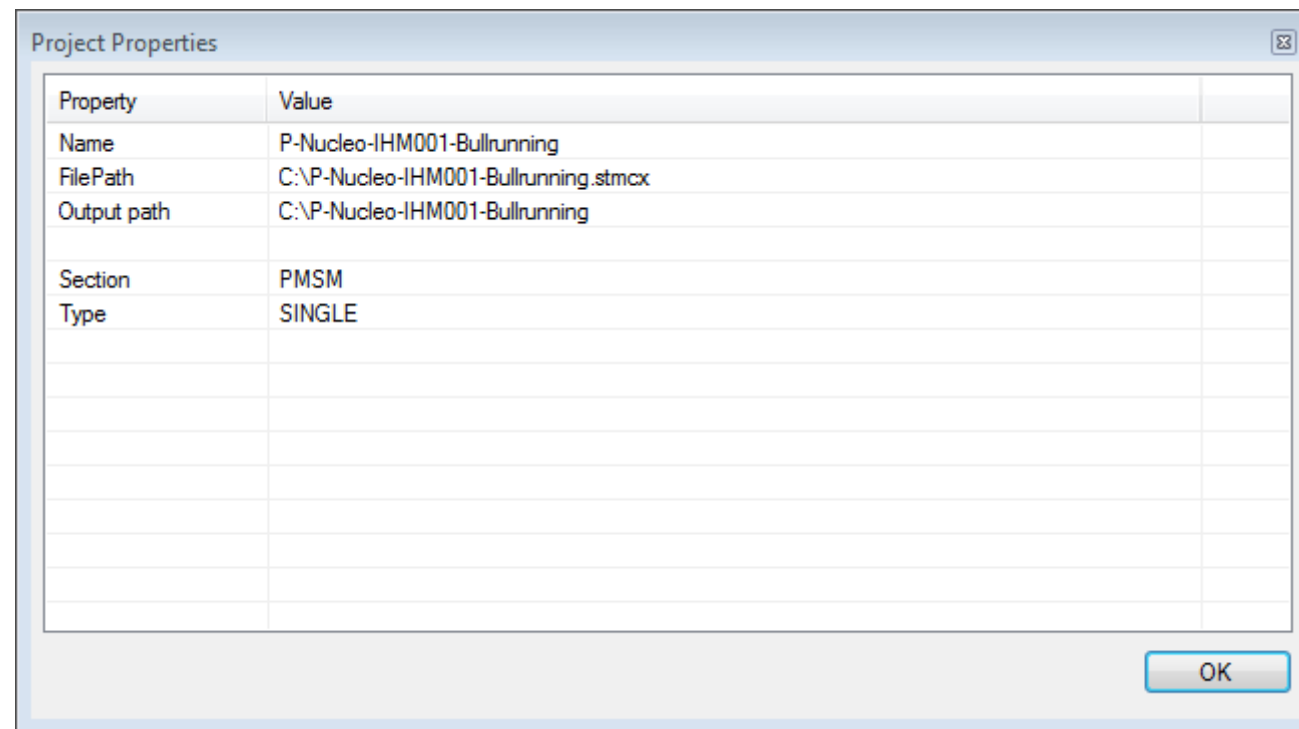
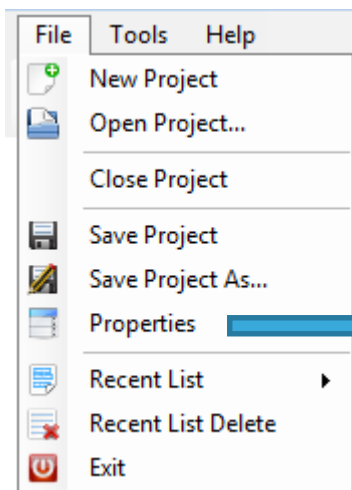
硬件细节设定按钮区 (Hardware Detail Setting Button Area)

主要的硬件配置 (Main Hardware Configuration)

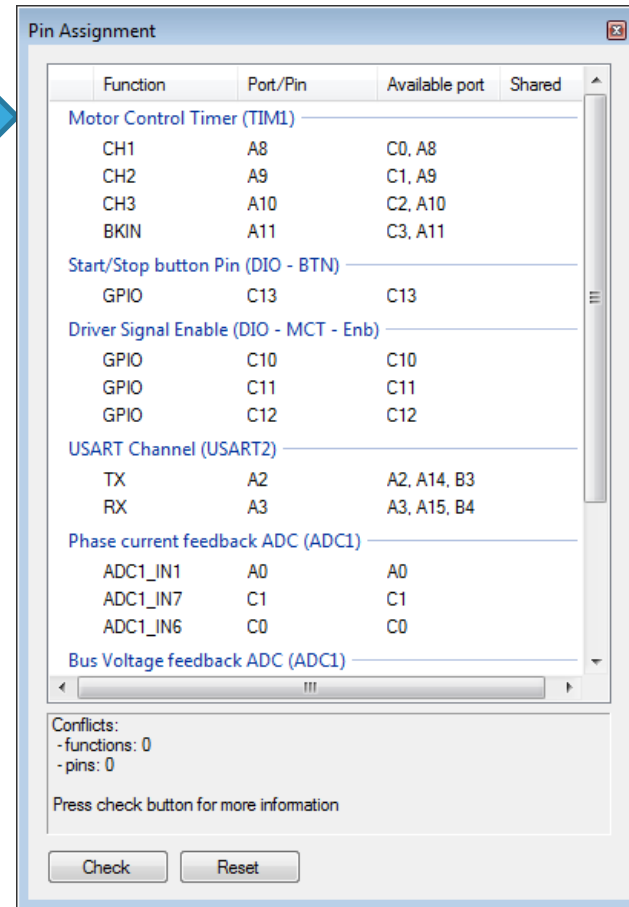
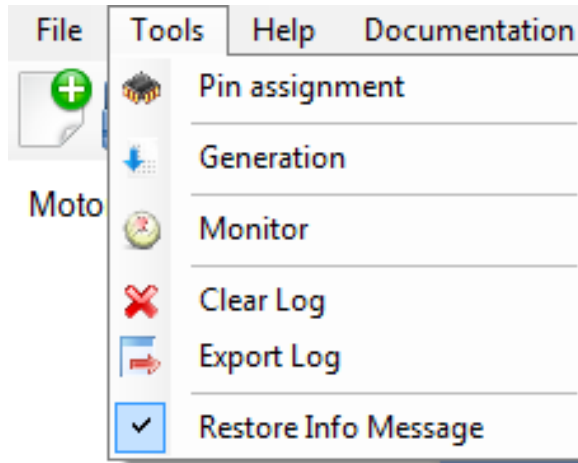
用户信息 (User Information)

| Variable | Motor | Unit |
|------------------------------|-------------------|------|
| PWM frequency | 30000 | Hz |
| Sensor selection main | Sensor-less (O... | |
| Sensor selection aux | Sensor-less (O... | |
| Torque&Flux - Execution rate | 1 | PW |
| Bus voltage sensing | true | |

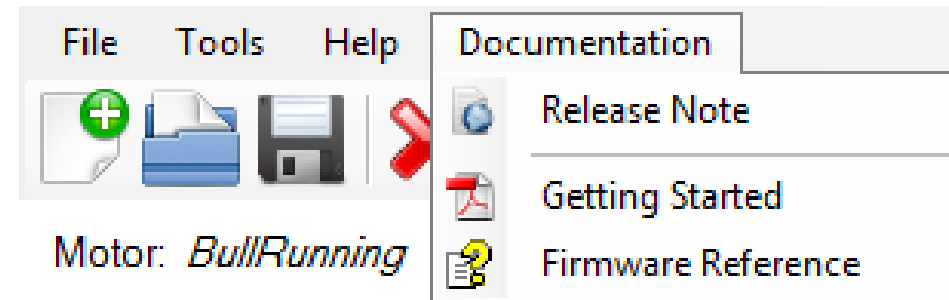
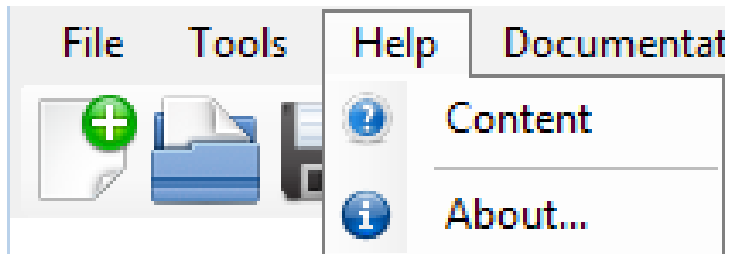
| Time | Motor | Id | Message |
|----------|-------|----|---|
| 04:19:08 | | | The 'PFC' is not supported in the FW for SDK5.x excepted for SDK for 'STM32F103 High Density'. All parameters will be |
| 04:19:08 | | | The 'Sensor-less (HFI+Observer)' is not supported in the FW for SDK5.x. All parameters will be disabled. |
| 04:19:08 | | | F2 mcus are not supported in the FW for SDK5.x |

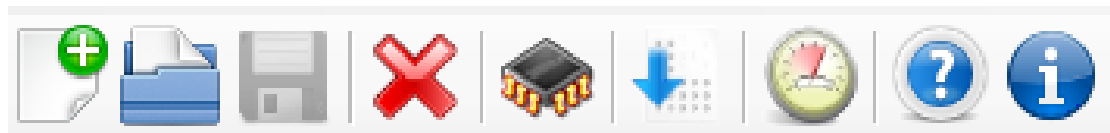











- ✓ New Project: 创建一个新项目
- ✓ Open Project...: 打开一个现有的项目
- ✓ Close Project: 关闭现在的项目
- ✓ Save Project: 使用相同文件名保存打开的项目
- ✓ Save Project As...: 将打开的项目保存为指定的文件名
- ✓ Properties: 查看项目属性
- ✓ Recent List: 从最近打开过的项目列表中加载一个现有项目
- ✓ Recent List Delete: 删除最近的项目列表
- ✓ Exit: 从硬件配置窗口退出



- ✓ Pin assignment: 检查MCU的引脚分配和剩余的可用引脚
- ✓ Generation: 根据所选的IDE, 生成MC应用工程文件
- ✓ Monitor: 监控并转动电机
- ✓ Clear Log: 清除用户信息表
- ✓ Export Log: 将用户信息表以文本格式导出到日志文件
- ✓ Restore Info Message: 需要时显示用户信息表





- ✓  New: 创建一个新项目
- ✓  Load: 打开一个现有的项目
- ✓  Save: 使用相同文件名保存打开的项目
- ✓  Clear Log: 清除用户信息表
- ✓  Pin assignment : 检查MCU的引脚分配和剩余的可用引脚
- ✓  Generation: 根据所选的IDE, 生成MC应用工程文件.
- ✓  Click to open monitor: 监控并转动电机
- ✓  Help: 提供在线帮助文件的访问接口
- ✓  About: 显示该应用的版本号

根据客户需求快速设置电机库

The screenshot displays the ST Motor Control Workbench interface. At the top, it shows the project name 'ST Motor Control Workbench [P-Nucleo-IHM001-Bullrunning]' and a menu bar with 'File', 'Tools', 'Help', and 'Documentation'. Below the menu bar is a toolbar with various icons. The main workspace shows a detailed schematic of a motor control system. On the left, there's an 'AC Input Info' block with an 'Inrush Current Limiter'. This is followed by a 'PFC' (Power Factor Correction) block. The 'Rated Bus Voltage' is set to '11 V (5 - 36) V'. To the right, there are 'Bus Voltage Sensing' and 'Dissipative Brake' blocks. In the center, the 'Control Unit' is shown, containing 'Firmware Drive Management', 'MCU and Clock Freq.', 'Digital I/O', 'DAC functionality', and 'Analog Input and Protection'. It is connected to 'Drivers' for 'Phase U', 'Phase V', and 'Phase W'. A 'User Interface' block is also connected to the control unit. On the right, there's a motor 'M' with 'Temperature Sensing', 'Current Sensing', 'Over Current Protection', and 'Speed Sensing' blocks connected to it. The bottom of the interface features a 'Variable' table and a message log.

驱动控制管理

单片机相关

电机参数

硬件驱动

| Variable | Motor | Uni |
|------------------------------|-------------------|-----|
| PWM frequency | 30000 | Hz |
| Sensor selection main | Sensor-less (O... | |
| Sensor selection aux | Sensor-less (O... | |
| Torque&Flux - Execution rate | 1 | PW |
| Bus voltage sensing | true | |

| Time | Motor | Id | Message |
|----------|-------|----|---|
| 11:56:29 | | | The 'PFC' is not supported in the FW for SDK5.x excepted for SDK for 'STM32F103 High Density'. All parameters will be disa... |
| 11:56:29 | | | The 'Sensor-less (HFI+Observer)' is not supported in the FW for SDK5.x. All parameters will be disabled. |
| 11:56:29 | | | F2 mcus are not supported in the FW for SDK5.x |
| 11:56:29 | | | F103 High Density in dual Motor mcus are not supported in the FW for SDK5.x |

The screenshot displays the ST Motor Control Workbench interface. At the top, the window title is "ST Motor Control Workbench [P-Nucleo-IHM001-Bullrunning]". The main area shows a schematic diagram of a motor control system, including an AC input stage with an inrush current limiter and PFC, a bus with a 11V (5-36V) rated voltage, and a motor (M) with U, V, and W terminals. A "Control Unit" block is also visible, containing options for Firmware Drive Management, MCU and Clock Freq., Digital I/O, DAC functionality, and Analog Input and Protection.

Two "Motor - Parameters" windows are open, showing the configuration for a "Surface Mounted PMSM" motor. The left window shows the "Electrical parameters" tab with the following values:

| Parameter | Value | Unit |
|------------------------|-------|---------------------|
| Pole Pairs | 7 | |
| Max. Application Speed | 15000 | rpm |
| Nominal Current | 1.20 | Apk |
| Nominal DC Voltage | 11.0 | V |
| Rs | 0.11 | Ohm |
| Ls | 0.018 | mH |
| B-Emf constant | 0.4 | Vms/krpm |
| Inertia | 0.348 | uN*m*s ² |
| Friction | 0.437 | uN*m*s |

The right window shows the "Sensors" tab with the following values:

| Sensor Type | Parameter | Value | Unit |
|--------------------|----------------------------------|-------|------|
| Hall sensors | Sensors displacement | 120 | deg |
| | Placement electrical angle | 300 | deg |
| Quadrature encoder | Pulses per mechanical revolution | 400 | |

Yellow arrows indicate the flow of information from the motor schematic to the parameter windows.

➤ 马达参数配置

极对数

最大转速

最大电流

额定电压

电机相电阻

电机电感

电机发电常数

电机转动惯量

电机阻力系数

Motor - Parameters

Motor Sensors

Magnetic structure: Surface Mounted PMSM

Electrical parameters

| | | |
|------------------------|-------|---------------------|
| Pole Pairs | 7 | |
| Max. Application Speed | 15000 | rpm |
| Nominal Current | 1.20 | Apk |
| Nominal DC Voltage | 11.0 | V |
| Rs | 0.11 | Ohm |
| Ls | 0.018 | mH |
| B-Emf constant | 0.4 | Vms/krpm |
| Inertia | 0.348 | uN*m*s ² |
| Friction | 0.437 | uN*m*s |

Save parameters

Motor - Parameters

Motor Sensors

Magnetic structure: Internal PMSM

Electrical parameters

| | | |
|------------------------|-------|---------------------|
| Pole Pairs | 7 | |
| Max. Application Speed | 15000 | rpm |
| Nominal Current | 1.20 | Apk |
| Nominal DC Voltage | 11.0 | V |
| Rs | 0.11 | Ohm |
| Ld | 0.018 | mH |
| Lq | 0.018 | mH |
| Ld/Lq ratio | 1.000 | |
| B-Emf constant | 0.4 | Vms/krpm |
| Inertia | 0.348 | uN*m*s ² |
| Friction | 0.437 | uN*m*s |

Save parameters

Done

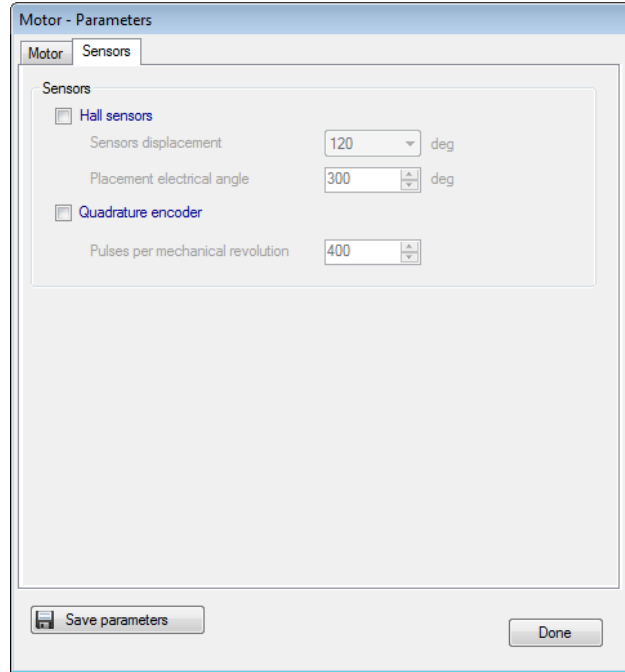
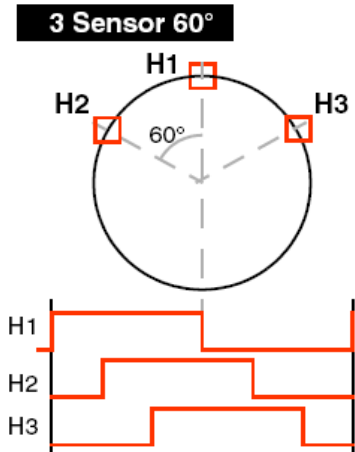
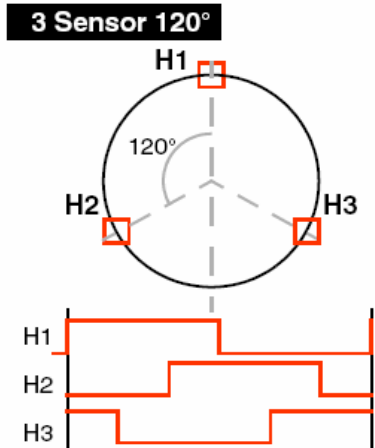
电机d轴电感

电机q轴电感

d轴电感和q轴电感比率

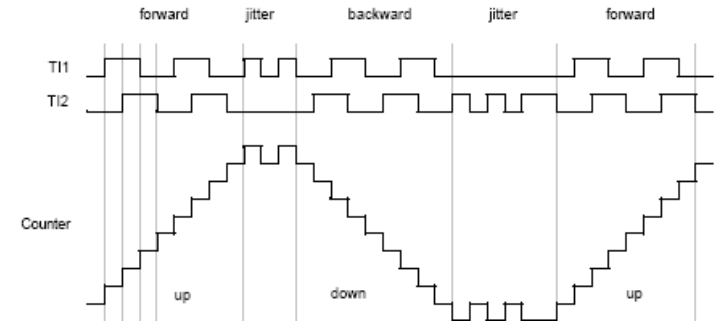
➤ Hall同步电角度:

- 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相反电动势最高点作为电角度的0度;



➤ 编码器参数

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



The screenshot displays the ST Motor Control Workbench interface. The main window shows a power stage diagram with components like AC Input Info, Inrush Current Limiter, PFC, and Rated Bus Voltage (11 V (5 - 36) V). Two dialog boxes are open:

- AC Input Info - Hardware Settings:**
 - Voltage: custom
 - Minimum: 160 Vms
 - Maximum: 340 Vms
 - Nominal: 230 Vms
 - Frequency: 50 Hz (selected), 60 Hz
- Power Stage - Rated Bus Voltage Info:**
 - Rated Voltage:
 - Min rated voltage: 5 V
 - Max rated voltage: 36 V
 - Nominal voltage: 11 V

Red arrows point from the 'AC Input Info' and 'Rated Bus Voltage' labels in the diagram to their respective dialog boxes. A 'Done' button is visible at the bottom of both dialog boxes.

The screenshot displays the ST Motor Control Workbench interface. The main window shows a power stage diagram with components like AC Input Info, Inrush Current Limiter, PFC, Drivers (Phase U, V, W), and various sensing blocks (Bus Voltage Sensing, Temperature Sensing, Current Sensing, Over Current Protection, Speed Sensing). A red arrow points from the 'Bus Voltage Sensing' block to a detailed inset window titled 'Power Stage - Bus Voltage Sensing'.

The inset window shows a circuit diagram divided into 'Power Stage' and 'Control Stage'. It features a voltage divider with resistors R1 (169.00 kOhm) and R2 (9.30 kOhm) connected to the V_{Bus} line, and a feedback resistor R3 connected to the V_{Bus} feedback line. The calculated bus voltage divider value is shown as 19.17.

At the bottom of the main window, there is a variable declaration table and a message log.

| Variable | Motor | Uni |
|------------------------------|-------------------|-----|
| PWM frequency | 30000 | Hz |
| Sensor selection main | Sensor-less (O... | |
| Sensor selection aux | Sensor-less (O... | |
| Torque&Flux - Execution rate | 1 | PW |
| Bus voltage sensing | true | |

| Time | Motor | Id | Message |
|----------|-------|----|---|
| 04:19:08 | | | F2 mcus are not supported in the FW for SDK5.x |
| 04:19:08 | | | F103 High Density in dual Motor mcus are not supported in the FW for SDK5.x |
| 04:25:38 | | | Project: 'P-Nucleo-IHM001-Bullrunning' saved successfully. |

The screenshot displays the ST Motor Control Workbench interface. The main workspace shows a power electronics schematic with components like AC Input Info, Inrush Current Limiter, PFC, Bus Voltage Sensing, Dissipative Brake, Temperature Sensing, Current Sensing, Over Current Protection, and Speed Sensing. Two dialog boxes are open: 'Inrush Current Limiter' and 'Power Stage - Dissipative Brake'. The 'Inrush Current Limiter' dialog has 'Active high' selected for Polarity and 'enable' checked under Additional Features. The 'Power Stage - Dissipative Brake' dialog also has 'Active high' selected for Polarity. A status bar at the bottom indicates the project was saved successfully.



ST Motor Control Workbench [P-Nucleo-IHM001-Bullrunning]

Motor: BullRunning - Control Board: P-NUCLEO-IHM001/002 3Sh - board: NUCLEO-F302R8 - Power Board: P-NUCLEO-IHM001/002 3Sh

Rated Bus Voltage: 11 V (5 - 36) V

Power Factor Correction - Hardware Settings

| | | |
|--|--------------|-----|
| Nominal power | 1000 | W |
| Nominal current | 6.149 | Apk |
| Shunt resistor value | 0.220 | ohm |
| <input checked="" type="checkbox"/> OPAMP on power stage | | |
| <input checked="" type="checkbox"/> use OPAMP for Current Protection | | |
| Comparator threshold | 1.20 | V |
| Overall network gain | 2.76 | |
| Expected Over Current threshold | 1.976 | Apk |
| Max. power transistor current | 10.000 | Apk |
| AC voltage sensing divider 1/... | 116 | |
| Ton propagation delay | 2550 | ns |
| Toff propagation delay | 2550 | ns |
| Driving signal polarity | Active low | |
| Overcurrent signal polarity | Active low | |
| AC Mains sync signal polarity | Falling edge | |

Power Factor Correction - PFC Parameters

Enabling feature

Current Regulation

| | | | | | | |
|-----------------------------------|-------|-------------|------|---|-----|---|
| PWM frequency | 40000 | Hz | 1000 | / | 16 | P |
| Current regulation execution rate | 1 | PWM periods | 700 | / | 256 | I |
| SW Current Limitation | 4.999 | Apk | | | | |

Voltage Regulation

| | | | | | | |
|--|-----|------|------|---|-----|---|
| Output voltage reference | 350 | V | 1000 | / | 16 | P |
| PFC over-voltage threshold | 370 | V | 700 | / | 256 | I |
| Voltage regulation frequency | 100 | Hz | | | | |
| Soft Start Duration | 300 | ms | | | | |
| Switch-on Power level | 250 | W | | | | |
| Switch-off Power level | 50 | W | | | | |
| Digital filter duration on AC sync pin | 1.3 | usec | | | | |
| Digital filter duration on OCP pin | 1.3 | usec | | | | |

ST Motor Control Workbench [P-Nucleo-IHM001-Bullrunning]

Motor: BullRunning - Control Board: P-NUCLEO-IHM001/002 3Sh - board: NUCLEO-F302R8 - Power Board: P-NUCLEO-IHM001/002 3Sh - board: X-NUCLEO-IH

Rated Bus Voltage: 11 V (5 - 36) V

Control Unit:

- Firmware Drive Management
- MCU and Clock Freq.
- Digital I/O
- DAC functionality
- Analog Input and Protection
- User Interface

Drivers: Phase U, Phase V, Phase W

Sensing and Protection:

- Bus Voltage Sensing
- Dissipative Brake
- Temperature Sensing
- Current Sensing
- Over Current Protection
- Speed Sensing

Temperature Sensing Hardware Settings:

- Temperature sensing - V0: 1055 mV
- Temperature sensing - T0: 25.0 °C
- ΔV/ΔT: 22.7 mV/°C
- Max working temperature on sensor: 110 °C

Firmware protection:

- Enable
- Over-Temperature:
 - Set intervention threshold to power stage max working temperature
 - Over-temperature threshold: 110 °C
 - Hysteresis: 10 °C

Variable Table:

| Variable | Motor | Uni |
|------------------------------|-------------------|-----|
| PWM frequency | 30000 | Hz |
| Sensor selection main | Sensor-less (O... | |
| Sensor selection aux | Sensor-less (O... | |
| Torque&Flux - Execution rate | 1 | PW |
| Bus voltage sensing | true | |

Log:

| Time | Motor | Id | Message |
|----------|-------|----|---|
| 04:19:08 | | | F2 mcus are not supported in the FW for SDK5.x |
| 04:19:08 | | | F103 High Density in dual Motor mcus are not supported in the FW for SDK5.x |
| 04:25:38 | | | Project: 'P-Nucleo-IHM001-Bullrunning' saved successfully. |

ST Motor Control Workbench [P-Nucleo-IHM001-Bullrunning]

File Tools Help Documentation

Motor: BullRunning - Control Board: P-NUCLEO-IHM001/002 3Sh - board: NUCLEO-F302R8 - Power Board: P-NUCLEO-IHM001/002 3Sh - board: X-NUCLEO-IH

life.augmented

Rated Bus Voltage: 1 V (5-36) V

Bus Voltage Sensing

Disjunctive

Power Stage - Driving Signals Polarity - U Driver

High side driving signal

Polarity: Active high

Low side driving signal

Complemented from high side:

Polarity: Active high

HW inserted dead time: 800 ns

Driver enabling signal

signal:

Polarity: Active high

Force same values for U,V,W Driver

Share signal enable

Done

Power Stage - Power Switches

Min dead-time: 800 ns

Max switching frequency: 50 kHz

Done

Drivers

Phase U

Phase V

Phase W

Temperature Sensing

Current Sensing

Over Current Protection

Speed Sensing

Sensorless Main

Message

F2 mcus are not supported in the FW for SDK5x

F103 High Density in dual Motor mcus are not supported in the FW for SDK5x

Project: 'P-Nucleo-IHM001-Bullrunning' saved successfully.

ST Motor Control Workbench [P-Nucleo-IHM001-Bullrunning]

Motor: BullRunning - Control Board: P-NUCLEO-IHM001/002 3Sh - board: NUCLEO-F302R8 - Power Board: P-NUCLEO-IHM001/002 3Sh - board: X-NUCLEO-IH

Rated Bus Voltage: 11 V (5 - 36) V

Control Unit:

- Firmware Drive Management
- MCU and Clock Freq.
- Digital I/O
- DAC functionality
- Analog Input and Protection
- User Interface

Power Stage - Current Sensing:

Current sensor and signal conditioning

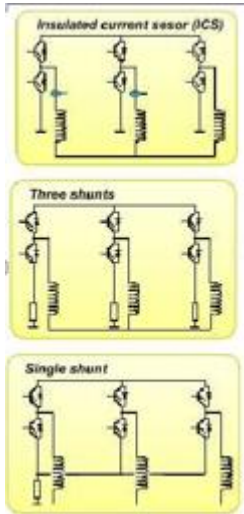
- Current reading topology: Three Shunt Resistors
- ICS gain: 1.000 V/A
- Shunt resistor(s) value: 0.330 ohm
- Amplification on board:
- Amplifying network gain: 1.53
- T-rise: 700 ns
- T-noise: 700 ns
- Max Readable Current: 3.268 A

| Variable | Motor | Uni |
|------------------------------|-------------------|-----|
| PWM frequency | 30000 | Hz |
| Sensor selection main | Sensor-less (O... | |
| Sensor selection aux | Sensor-less (O... | |
| Torque&Flux - Execution rate | 1 | PW |
| Bus voltage sensing | true | |

| Time | Motor | Id | Message |
|----------|-------|----|--|
| 04:19:08 | | | F2 mcus are not supported in the FW for SDK5x |
| 04:19:08 | | | F103 High Density in dual Motor mcus are not supported in the FW for S |
| 04:25:38 | | | Project: 'P-Nucleo-IHM001-Bullrunning' saved successfully. |

电流采样拓扑结构选择:

$$(1 + R5 / R4) \cdot \frac{(R2 // R3)}{((R2 // R3) + R1)}$$



Power Stage - Current Sensing

Current sensor and signal conditioning

Current reading topology: **Three Shunt Resistors**

ICS gain: 1.000 V/A

Shunt resistor(s) value: 0.330 ohm

Amplification on board:

Amplifying network gain: 1.53

T-rise: 700 ns

T-noise: 700 ns

Max Readable Current: 3.268 A

Calculate

Done

Amplifying Network Gain Calculator

Parameters

| Name | Value | Unit | Description |
|------------------|-------|------|------------------------------------|
| I _{max} | 2.80 | A | Desired maximum current |
| V _m | 3.3 | V | Reference voltage |
| R-shunt | 0.330 | Ohm | Shunt resistor value |
| P-rating | 0.5 | W | Shunt resistor power rating |
| R1 | 680 | Ohm | Resistance of the feedback network |
| R2 | 2200 | Ohm | Resistance of the feedback network |
| R4 | 2200 | Ohm | "Non-inverting" feedback resistor |
| R5 | 2200 | Ohm | "Non-inverting" feedback resistor |
| V _{mcu} | 3.3 | V | MCU voltage |

Result

| Name | Value | Unit | Description |
|---------------------------------|-------|------|--|
| Overall gain | 1.528 | | Overall gain of the network |
| Offset network attenuation | 0.764 | | Signal attenuation due to offset network |
| Op-amp gain | 2.000 | | Gain of the "Non-inverting" op-amp |
| V _{out} (polarization) | 1.559 | V | Opamp output voltage (polarization) |
| Min V _{out} | 0.147 | V | Opamp output voltage (minimum) |
| Max V _{out} | 2.971 | V | Opamp output voltage (maximum) |
| Power dissipated on shunt | 0.647 | W | Power dissipated on shunt resistor |

Export

Confirm Cancel

The screenshot displays the ST Motor Control Workbench interface for configuring a power stage. The main window title is "ST Motor Control Workbench [P-Nucleo-IHM001-Bullrunning]". The menu bar includes "File", "Tools", "Help", and "Documentation". The status bar shows: "Motor: BullRunning - Control Board: P-NUCLEO-IHM001/002 3Sh - board: NUCLEO-F302R8 - Power Board: P-NUCLEO-IHM001/002 3Sh - board: X-NUCLEO-IH...".

Two "Power Stage - Over Current Protection" dialog boxes are open, showing the following configuration parameters:

- Over Current Protection
- Comparator threshold: 0.50 V
- Over current network offset: 0.00 V
- Over current network gain: 0.1800 V/A
- Expected over-current threshold: 2.7778 A
- Over-current feedback signal polarity: Active low
- Over-current protection disabling network
- Over-current protection disabling network polarity: Active high

The background shows a block diagram with components: Bus Voltage Sensing, Dissipative Brake, Temperature Sensing, Current Sensing, Over Current Protection, and Speed Sensing. Red arrows point from the "Over Current Protection" block in the diagram to the configuration dialog boxes.

At the bottom, a status bar shows the time "04:25:38" and a message: "Project: 'P-Nucleo-IHM001-Bullrunning' saved successfully." A "Change Log" button is also visible.

ST Motor Control Workbench [P-Nucleo-IHM001-Bullrunning]*

File Tools Help Documentation

Motor: BullRunning - Control Board: P-NUCLEO-IHM001/002 3Sh - board: NUCLEO-F302R8 - Power Board: P-NUCLEO-IHM001/002 3Sh - board: X-NUCLEO-IHM07M1

Control Unit

- Firmware Drive Management
- MCU and Clock Freq.
- Digital I/O
- DAC functionality
- Analog Input and Protection

Drivers

- Phase U
- Phase V

Start-up parameters

- Drive Setting
- Additional Features
- Sensing and FW protection

Speed Sensing

| Variable | Motor | Uni |
|------------------------------|-------------------|-----|
| PWM frequency | 30000 | Hz |
| Sensor selection main | Sensor-less (O... | |
| Sensor selection aux | Sensor-less (O... | |
| Torque&Flux - Execution rate | 1 | PW |
| Bus voltage sensing | true | |

| Time | Motor | Id | Message |
|----------|-------|----|--|
| 11:56:29 | | | The 'PFC' is not supported in the FW for SDK5x excepted for SDK for 'STM32F103 High Density'. All parameters will be disa... |
| 11:56:29 | | | The 'Sensor-less (HFI+Observer)' is not supported in the FW for SDK5x. All parameters will be disabled. |
| 11:56:29 | | | F2 mcus are not supported in the FW for SDK5x |
| 11:56:29 | | | F103 High Density in dual Motor mcus are not supported in the FW for SDK5x |

Info / Errors / Warnings Change Log

Drive Management - Speed Position Feedback Management

Main sensor | Auxiliary sensor

Sensor selection: Sensorless (Observer+PLL)

Max measurement errors number before fault: 3

Observer+PLL

Variance threshold: 10.00 %

Average speed depth for speed loop: 64

Average speed depth for observer equations: 64

B-emf consistency tolerance: 100.00 %

B-emf consistency gain: 100.00 %

Manual editing enabled

Observer

G1: -1380

G2: 2895

Back compatibility

PLL

2234 / 16384 P

52 / 65536 I

Done

Drive Management - Speed Position Feedback Management

Main sensor | Auxiliary sensor

Sensor selection: Hall sensors

Max measurement errors number before fault: 3

Hall Sensors

Average speed FIFO depth: 6

Input Capture filter duration: 1.3 usec

Drive Management - Speed Position Feedback Management

Main sensor | Auxiliary sensor

Sensor selection: Quadrature encoder

Max measurement errors number before fault: 3

Quadrature Encoder

Average speed FIFO depth: 16

Input Capture filter duration: 0.7 usec

Reverse counting direction:

Drive Management - Start-up parameters

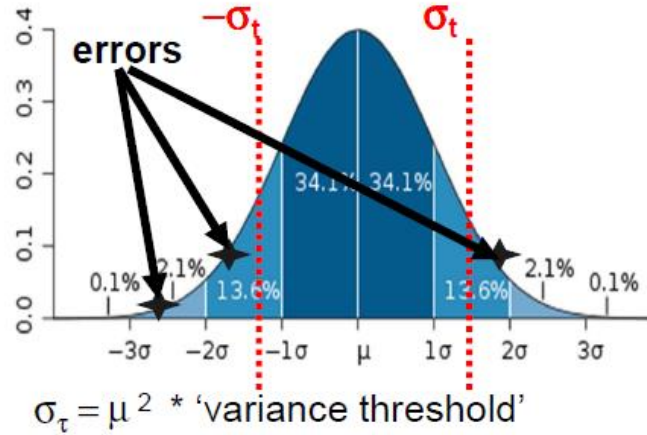
Encoder alignment settings

Duration: 700 ms

Alignment electrical angle: 90 deg

Final current ramp value: 1.95 A

Done



Speed Average value

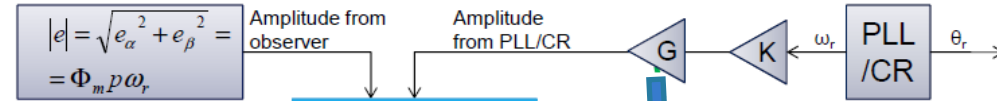
$$\mu = \frac{\sum_{i=1}^{64} \omega_i}{64}$$

Speed Variance

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

- 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



Observed bemfs are consistent with physical model if match is found

Drive Management - Speed Position Feedback Management

Main sensor | Auxiliary sensor

Sensor selection: Sensor-less (Observer+PLL)

Max measurement error number before fault: 3

Observer+PLL

Variance threshold: 30.00 %

Average speed depth for speed loop: 64

Average speed depth for observer equations: 64

B-emf consistency tolerance: 100.00 %

B-emf consistency gain: 100.00 %

Manual editing enabled

Observer: G1: -12064, G2: 27061

PLL: 266 / 16384 P, 11 / 65536 I

Back compatibility

Drive Management - Speed Position Feedback Management

Main sensor | Auxiliary sensor

Enable auxiliary sensor

Sensor selection: Sensor-less (Observer+Cordic)

Max measurement errors number before fault: 3

Observer+Cordic

- Variance threshold: 400.0 %
- Average speed FIFO depth for speed loop: 64
- Average speed FIFO depth for observer equations: 64
- B-emf consistency tolerance: 100.00 %
- B-emf consistency gain: 100.00 %
- Maximum application acceleration: 6000 rpm/s
- B-emf quality factor: 0.017

Manual editing enabled

Observer

- G1: -1380
- G2: 3473

Back compatibility

Done

Drive Management - Speed Position Feedback Management

Main sensor | Auxiliary sensor

Enable auxiliary sensor

Sensor selection: Quadrature encoder

Max measurement errors number before fault: 3

Quadrature Encoder

- Average speed FIFO depth: 16
- Input Capture filter duration: 0.7 usec
- Reverse counting direction:

Done

Drive Management - Speed Position Feedback Management

Main sensor | Auxiliary sensor

Enable auxiliary sensor

Sensor selection: Hall sensors

Max measurement errors number before fault: 3

Hall Sensors

- Average speed FIFO depth: 6
- Input Capture filter duration: 1.3 usec

Done

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

Start up on Fly

Detection Duration 1000 ms

Braking Duration 1000 ms

Sensor-less rev-up settings

On-the-Fly startup

Profile

Basic

Advanced customized

Initial electrical angle 90 deg

| | Duration (ms) | Final speed (rpm) | Final current (A) |
|----|---------------|-------------------|-------------------|
| 1) | 700 | 2000 | 0.70 |
| 2) | 700 | 2000 | 0.70 |
| 3) | 700 | 2000 | 0.70 |
| 4) | 700 | 2000 | 0.70 |
| 5) | 700 | 2000 | 0.70 |

Execute sensor-less algorithm starting from 1

Consecutive successful start-up output tests 2

Minimum start-up output speed 1000 rpm

Estimated speed Band tolerance upper limit 106.25 %

Estimated speed Band tolerance lower limit 93.75 %

Rev-up to FOC switch-over

Enable

Duration 25 ms

Speed (rpm) vs Duration (ms) graph showing a ramp up to 2000 rpm and a constant current of 0.70 A.

Done

切入闭环条件判定

启动曲线

PWM载波频率

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

Speed regulator

Execution rate: 1.0 ms

1000 / 16 P

600 / 256 I

Manual editing enabled

Default settings

Control mode: Speed control

Target speed: 3000 rpm

Target stator current flux component: 0.00 A

Target stator current torque: 0.00 A

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

速度环Kp & Ki

电流环Kp & Ki

ST Motor Control Workbench [P-Nucleo-IHM001-Bullrunning]*

File Tools Help Documentation

Motor: BullRunning - Control Board: P-NUCLEO-IHM001/002 3Sh - board: NUCLEO-F302R8 - Power Board: P-NUCLEO-IHM001/002 3Sh - board: X-NUCLEO-IHM07M1

AC Input Info
Inrush Current Limiter
PFC
Rated Bus Voltage
11 V (5 - 36 V)

Control Unit

- Firmware Drive Management
- MCU and Clock Freq.
- Digital I/O
- DAC functionality
- Analog Input and Protection

Drivers

- Phase U
- Phase V

Sensing

- Over Current Protection
- Speed Sensing

Drive Management - Additional Features and PFC settings

- Flux weakening
- MTPA
- Feed Forward
- Sensorless speed feedback
- Inrush Current Limiter

Flux weakening

3000 / 32768 P
5000 / 32768 I

Voltage limit 98.5 %

Done

| Variable | Motor | Uni |
|------------------------------|-------------------|-----|
| PWM frequency | 30000 | Hz |
| Sensor selection main | Sensor-less (O... | |
| Sensor selection aux | Sensor-less (O... | |
| Torque&Flux - Execution rate | 1 | PW |
| Bus voltage sensing | true | |

| Time | Motor | Id | Message |
|----------|-------|----|--|
| 11:56:29 | | | The 'PFC' is not supported in the FW for SDK5x excepted for SDK for 'STM32F103 High Density'. All parameters will be disa... |
| 11:56:29 | | | The 'Sensor-less (HFI+Observer)' is not supported in the FW for SDK5x. All parameters will be disabled. |
| 11:56:29 | | | F2 mcus are not supported in the FW for SDK5x |
| 11:56:29 | | | F103 High Density in dual Motor mcus are not supported in the FW for SDK5x |

Info / Errors / Warnings Change Log

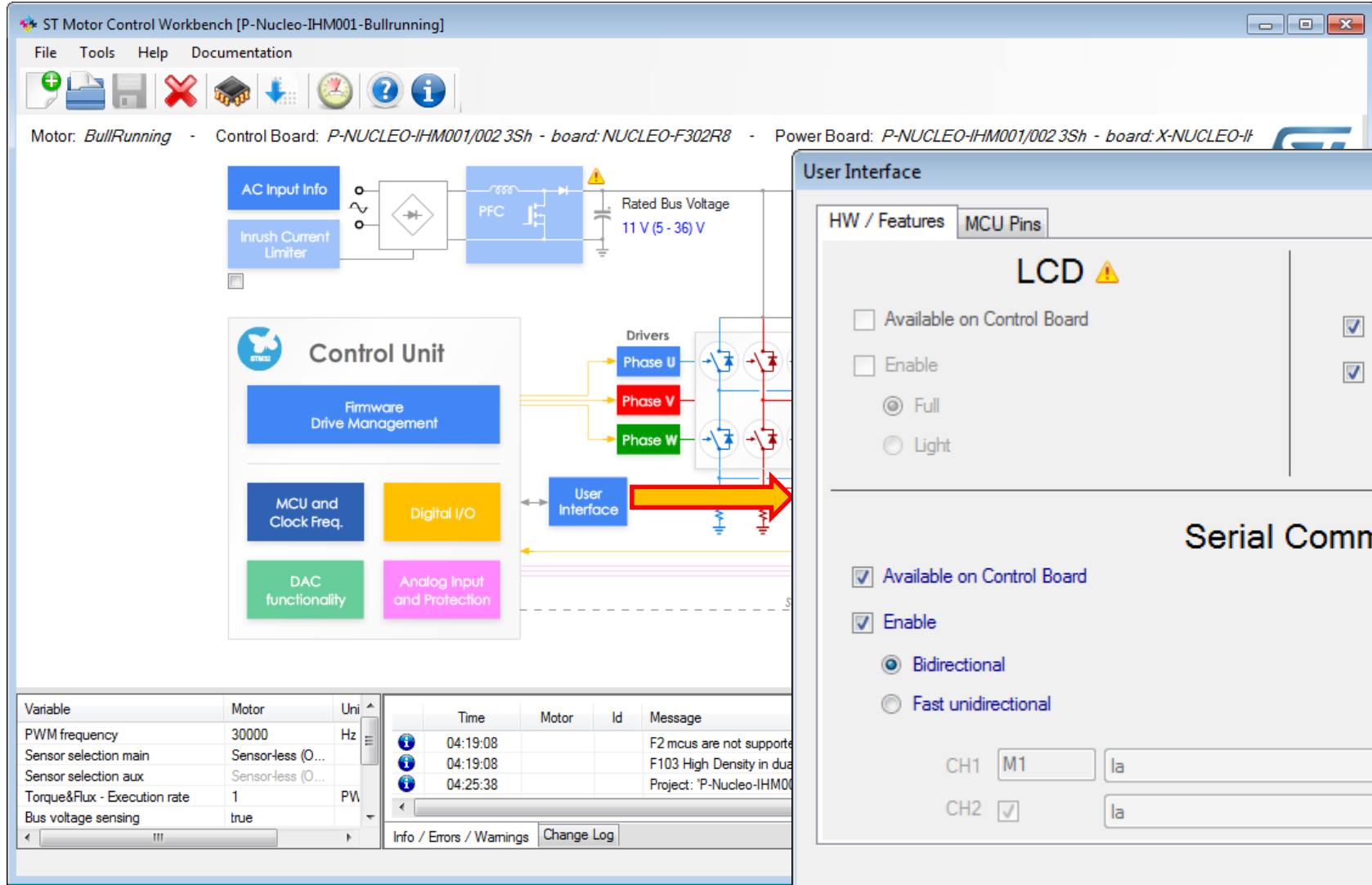
The screenshot displays the ST Motor Control Workbench interface. The main window shows a schematic diagram of the motor control system, including AC input, PFC, and motor drivers. A 'Control Unit' block is highlighted, with a context menu open over the 'Sensing and FW protection' option. A secondary window, 'Drive Management - Sensing Enabling and Firmware Protections', is open, showing the following settings:

- DC Bus voltage sensing:**
 - Enable
 - Over-voltage:**
 - Motor control
 - Enable
 - Set intervention threshold to power stage max rated voltage
 - Over-voltage threshold: 36 V
 - Over-voltage protection: Disable PWM generation
 - On over-voltage, disable over-current protection by HW
 - Under-voltage:**
 - Enable
 - Set intervention threshold to power stage min rated voltage
 - Under-voltage threshold: 5 V

Buttons for 'Temperature Sensing', 'AC Input', and 'Done' are visible at the bottom of the settings window. A red arrow points from the 'Sensing and FW protection' menu item to the 'Over-voltage' section of the settings window.

| Variable | Motor | Unit |
|------------------------------|-------------------|------|
| PWM frequency | 30000 | Hz |
| Sensor selection main | Sensor-less (O... | |
| Sensor selection aux | Sensor-less (O... | |
| Torque&Flux - Execution rate | 1 | PW |
| Bus voltage sensing | true | |

| Time | Motor | Id | Message |
|----------|-------|----|---|
| 11:56:29 | | | The 'PFC' is not supported in the FW for SDK5.x excepted for SDK for 'STM32F103 High Density'. All parameters will be disa... |
| 11:56:29 | | | The 'Sensor-less (HFI+Observer)' is not supported in the FW for SDK5.x. All parameters will be disabled. |
| 11:56:29 | | | F2 mcus are not supported in the FW for SDK5.x |
| 11:56:29 | | | F103 High Density in dual Motor mcus are not supported in the FW for SDK5.x |



User Interface

HW / Features | MCU Pins

LCD ⚠

- Available on Control Board
- Enable
- Full
- Light

Start/Stop Button

- Available on Control Board
- Enable

Serial Communication

- Available on Control Board
- Enable
- Bidirectional
- Fast unidirectional

CH1 M1

CH2

ST Motor Control Workbench [P-Nucleo-IHM001-Bullrunning]

Motor: BullRunning - Control Board: P-NUCLEO-IHM001/002 3Sh - board: NUCLEO-F302R8 - Power Board: P-NUCLEO-IHM001/002 3Sh - board: X-NUCLEO-IH

Control Unit

- Firmware Drive Management
- MCU and Clock Freq.
- Digital I/O
- User Interface
- DAC functionality
- Analog Input and Protection

Control Stage - MCU and Clock Frequency Selection

MCU selection

MCU TYPE: STM32F301x6/8 - STM32F302x6/8

MCU: STM32F302R8

package: LQFP64

Clock settings

Clock source: 8MHz External crystal/ceramic resonator

CPU frequency: 72 MHz

Supply voltage

Nominal MCU supply voltage: 3.30 V

Done

Workbench中不能选择HSI

| Variable | Motor | Uni |
|------------------------------|-------------------|-----|
| PWM frequency | 30000 | Hz |
| Sensor selection main | Sensor-less (O... | |
| Sensor selection aux | Sensor-less (O... | |
| Torque&Flux - Execution rate | 1 | PW |
| Bus voltage sensing | true | |

| Time | Motor | Id | Message |
|----------|-------|----|--|
| 04:19:08 | | | F2 mcus are not supporte |
| 04:19:08 | | | F103 High Density in dua |
| 04:25:38 | | | Project: 'P-Nucleo-IHM001-Bullrunning' saved successfully. |

Control Stage - Digital I/O

Inverter driving signal selection

Timer: TIM1
Remap: Partial re-map

Pin Map:

| | | | |
|-------|-----|------|----|
| CH1 | A8 | CH1N | A7 |
| CH2 | A9 | CH2N | B0 |
| CH3 | A10 | CH3N | B1 |
| BKin2 | A11 | | |

Signal Enabler

CH1: Port GPIOC, Pin C10
CH2: Port GPIOC, Pin C11
CH3: Port GPIOC, Pin C12

Share signal enable

Speed/position feedback

Encoder interface: Timer TIM2, Remap No remap

Pin Map:

| | |
|-----|-----|
| CH1 | A15 |
| CH2 | B3 |

Hall sensors interface

Timer TIM2, Remap No remap

Pin Map:

| | |
|-----|-----|
| CH1 | A15 |
| CH2 | B3 |
| CH3 | B10 |

Direct GPIO

DBO: Port GPIOC, Pin C11
ICL: Port GPIOC, Pin C10
OCP disabling: Port GPIOC, Pin C11

PFC drive signal and feedback

Timer: [empty]
Pin Map: PWM --, AC Mains --, OCS --

Serial communication

Channel: USART2, Baudrate: 115200, Remap: No remap

Pin Map: TX A2, RX A3

Start/Stop Button GPIO

Port: GPIOC, Pin: C13, Polarity: Active low

Variable table:

| Variable | Motor | Uni |
|------------------------------|-------------------|-----|
| PWM frequency | 30000 | Hz |
| Sensor selection main | Sensor-less (O... | |
| Sensor selection aux | Sensor-less (O... | |
| Torque&Flux - Execution rate | 1 | PW |
| Bus voltage sensing | true | |

Project: 'P-Nucleo-IHM001-Bullrunning' saved successfully.

ST Motor Control Workbench [P-Nucleo-IHM001-Bullrunning]

Motor: BullRunning - Control Board: P-NUCLEO-IHM001/002 3Sh - board: NUCLEO-F302R8 - Power Board: P-NUCLEO-IHM001/002 3Sh - board: X-NUCLEO-IH

Control Unit

- Firmware Drive Management
- MCU and Clock Freq.
- DAC functionality**
- Analog Input and Protection

Control Stage - DAC Functionality

DAC

CH1: Debug | M1 | Ia | A4

CH2: Debug | M1 | Ib | -

Variable | Motor | Uni | Value

| | | | |
|------------------------------|-------------------|----|--|
| PWM frequency | 30000 | Hz | |
| Sensor selection main | Sensor-less (O... | | |
| Sensor selection aux | Sensor-less (O... | | |
| Torque&Flux - Execution rate | 1 | PW | |
| Bus voltage sensing | true | | |

Time | Motor | Id | Message

| | | | |
|----------|--|--|---|
| 04:19:08 | | | F2 mcus are not supported in the FW for SDK5.x |
| 04:19:08 | | | F103 High Density in dual Motor mcus are not supported in the FW for SDK5.x |
| 04:25:38 | | | Project: 'P-Nucleo-IHM001-Bullrunning' saved successfully. |

The screenshot displays the ST Motor Control Workbench interface. The main window shows a schematic diagram of a motor control system, including an AC input stage with an inrush current limiter and a PFC stage, followed by a control unit and three-phase drivers (Phase U, V, W). A red circle highlights the 'Analog Input and Protection' block in the control unit, with a yellow arrow pointing to the configuration panel on the right.

Control Stage - Analog Input and Protection

Phase current feedback | Bus voltage feedback | Temperature feedback | PFC stage feedback

Current Sensing Topology: Embedded PGA, External OPAMP

Over Current Protection Topology: Embedded HW OCP, External Protection, No protection

Sensing

Setting: Sampling Time: 1.5 ADC clk, Sampling Time: 21 ns, Maximum modulation: 97 %, Peripheral Selection: ADC1

Pin map: Ch phase U: ADC1_IN1 (A0), Ch phase V: ADC1_IN11 (B0), Ch phase W: ADC1_IN6 (C0)

Sensing OPAMP

Setting: Peripheral selection: OPAMP1/OPAMP2, OPAMP Gain: Internal, Int gain type: 2, Overall Network Gain: 1.44, Calculate, Vout (polarization): 1.559 V, T-rise: 2550 ns, Feedback net filtering:

Pin map: Not inverting: Ch U: A1, Ch V: A7, Ch W: B0; Inverting: OPAMP1: A3, OPAMP2: C5; Output: OPAMP1: A2, OPAMP2: A6

Protection

Setting: Digital filter duration: 64 clock, Inverting input: Internal, Current threshold: 1.668 Apk, Voltage Threshold: 1.2 V, Output enable:

Pin map: Inverting input: none; Not inverting: Ch U: A1, Ch V: A7, Ch W: D14; COMP1: COMP1, COMP2: COMP2, COMP3: COMP3; Output: Ch U: A0, Ch V: A2, Ch W: C8

Variable

| Variable | Motor | Uni |
|------------------------------|-------------------|-----|
| PWM frequency | 30000 | Hz |
| Sensor selection main | Sensor-less (O... | |
| Sensor selection aux | Sensor-less (O... | |
| Torque&Flux - Execution rate | 1 | PW |
| Bus voltage sensing | true | |

Time

| Time | Motor | Id | Message |
|----------|-------|----|--|
| 04:19:08 | | | F2 mcus are not supported in the FW f... |
| 04:19:08 | | | F103 High Density in dual Motor mcus... |
| 04:25:38 | | | Project: 'P-Nucleo-IHM001-Bullrunning' |

The screenshot displays the ST Motor Control Workbench interface. The main window shows a schematic diagram of a motor control system. On the left, there is an 'AC Input Info' block and an 'Inrush Current Limiter' block. The central part of the schematic includes a 'PFC' (Power Factor Correction) stage and a 'Rated Bus Voltage' of 11 V (5 - 36 V). Below this, the 'Control Unit' is shown, containing 'Firmware Drive Management', 'MCU and Clock Freq.', 'Digital I/O', 'DAC functionality', and 'Analog Input and Protection'. The 'Analog Input and Protection' block is circled in red, and a yellow arrow points from it to the 'Control Stage - Analog Input and Protection' configuration window on the right.

The 'Control Stage - Analog Input and Protection' window has several tabs: 'Phase current feedback', 'Bus voltage feedback', 'Temperature feedback', and 'PFC stage feedback'. The 'Bus voltage feedback' tab is selected. Under 'Sensing Setting', there are dropdown menus for 'Sampling Time' (61.5) and 'Peripheral selection' (ADC1). The 'Pin map' section shows 'ADC Channel' set to 'ADC1_IN2 (A1)'. There is a checkbox for 'use Input Resistance (R3)' and a field for 'Input Resistance' set to 100.0 kOhm. A 'Bus Voltage Partitioning ...' button is also visible.

At the bottom of the main window, there is a 'Variable' table and a log window.

| Variable | Motor | Uni |
|------------------------------|-------------------|-----|
| PWM frequency | 30000 | Hz |
| Sensor selection main | Sensor-less (O... | |
| Sensor selection aux | Sensor-less (O... | |
| Torque&Flux - Execution rate | 1 | PW |
| Bus voltage sensing | true | |

| Time | Motor | Id | Message |
|----------|-------|----|------------------------------------|
| 04:19:08 | | | F2 mcus are not supported in the f |
| 04:19:08 | | | F103 High Density in dual Motor m |
| 04:25:38 | | | Project: 'P-Nucleo-IHM001-Bullrun |

The screenshot displays the ST Motor Control Workbench interface. The main workspace shows a schematic diagram of a motor control system. On the left, there is an AC input section with 'AC Input Info' and 'Inrush Current Limiter' blocks. This is followed by a PFC (Power Factor Correction) stage. The 'Control Unit' block contains 'Firmware Drive Management', 'MCU and Clock Freq.', 'Digital I/O', 'DAC functionality', and 'Analog Input and Protection'. The 'Analog Input and Protection' block is circled in red, and a yellow arrow points from it to the 'Control Stage - Analog Input and Protection' configuration window on the right.

The 'Control Stage - Analog Input and Protection' window has several tabs: 'Phase current feedback', 'Bus voltage feedback', 'Temperature feedback', and 'PFC stage feedback'. The 'Sensing Setting' section includes:

- Sampling Time: 61.5 (dropdown)
- ADC clk: ADC1
- Peripheral selection: ADC1
- Pin map: ADC Channel: ADC1_IN8 (C2)

At the bottom of the interface, there is a variable declaration table and a log window.

| Variable | Motor | Uni |
|------------------------------|-------------------|-----|
| PWM frequency | 30000 | Hz |
| Sensor selection main | Sensor-less (O... | |
| Sensor selection aux | Sensor-less (O... | |
| Torque&Flux - Execution rate | 1 | PW |
| Bus voltage sensing | true | |

| Time | Motor | Id | Message |
|----------|-------|----|--|
| 04:19:08 | | | F2 mcus are not supported in the FW fo |
| 04:19:08 | | | F103 High Density in dual Motor mcus |
| 04:25:38 | | | Project: 'P-Nucleo-IHM001-Bullrunning' |

The screenshot displays the ST Motor Control Workbench interface. The main workspace shows a schematic diagram of a motor control system. On the left, there is an 'AC Input Info' block and an 'Inrush Current Limiter' block. The central part of the schematic includes a 'PFC' (Power Factor Correction) stage and a 'Rated Bus Voltage' of 11 V (5 - 36 V). Below this, the 'Control Unit' is shown with several functional blocks: 'Firmware Drive Management', 'MCU and Clock Freq.', 'Digital I/O', 'DAC functionality', and 'Analog Input and Protection'. The 'Analog Input and Protection' block is circled in red, and a large yellow arrow points from it to the right-hand configuration window.

The configuration window, titled 'Control Stage - Analog Input and Protection', has four tabs: 'Phase current feedback', 'Bus voltage feedback', 'Temperature feedback', and 'PFC stage feedback'. The 'Bus voltage feedback' tab is active. It contains two main sections:

- Current sensing:**
 - Setting: Sampling Time (1.5), ADC clk, Peripheral selection (ADC2)
 - Pin map: ADC Channel (ADC12_IN13 (C3))
- AC voltage sensing:**
 - Setting: Sampling Time (1.5), ADC clk, PFC ACVolt Sens (ADC2)
 - Pin map: ADC Channel (ADC12_IN3 (A3))

At the bottom of the interface, there is a 'Variable' table and a log window.

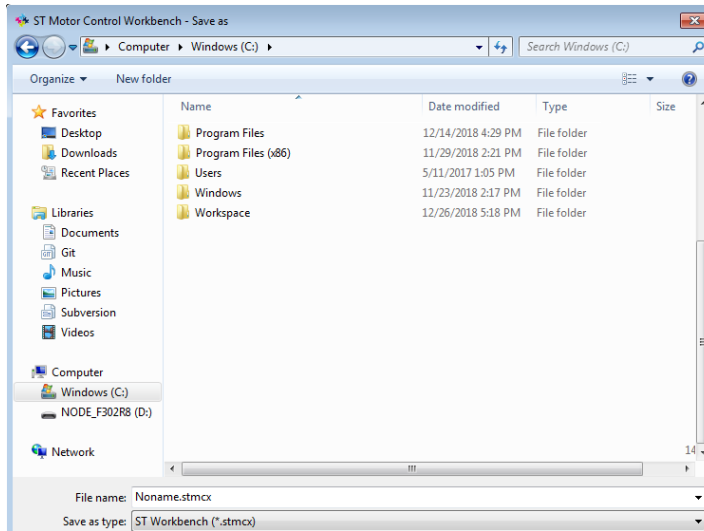
| Variable | Motor | Uni |
|------------------------------|-------------------|-----|
| PWM frequency | 30000 | Hz |
| Sensor selection main | Sensor-less (O... | |
| Sensor selection aux | Sensor-less (O... | |
| Torque&Flux - Execution rate | 1 | PW |
| Bus voltage sensing | true | |

| Time | Motor | Id | Message |
|----------|-------|----|--|
| 04:19:08 | | | F2 mcus are not supported in the FW f... |
| 04:19:08 | | | F103 High Density in dual Motor mcus... |
| 04:25:38 | | | Project: 'P-Nucleo-IHM001-Bullrunning' |

所有的参数配置完成后，点击生成图标，可根据所选的IDE生成MC应用工程：

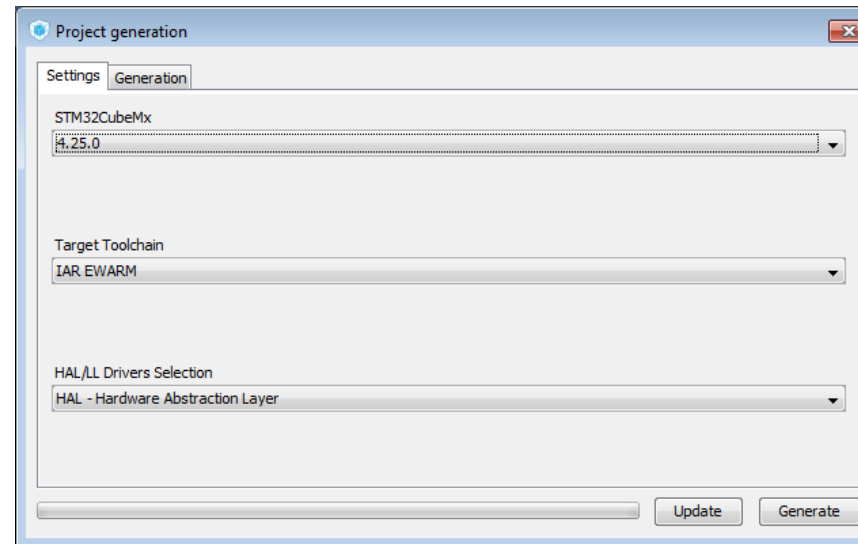


- 用户不再需要将MC Workbench输出文件拷贝到其工程下。
- STM32CubeMX作为接口被MC Workbench后台调用，来生成选择的IDE项目工作框架。

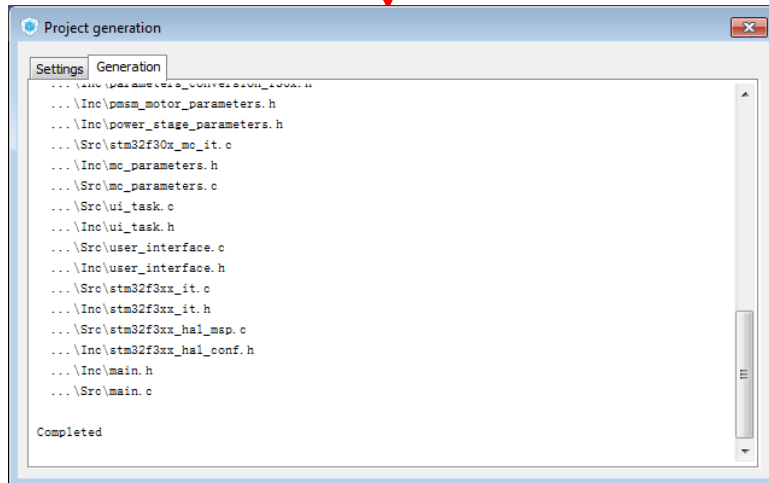


如果当前工程尚未保存，会显示一个文件管理器窗口，询问是否将当前工程设置保存为一个新项目。

如果当前工程已保存过，会弹出右侧的窗口，进行CubeMX版本，IDE的选择和HAL/LL的选择。



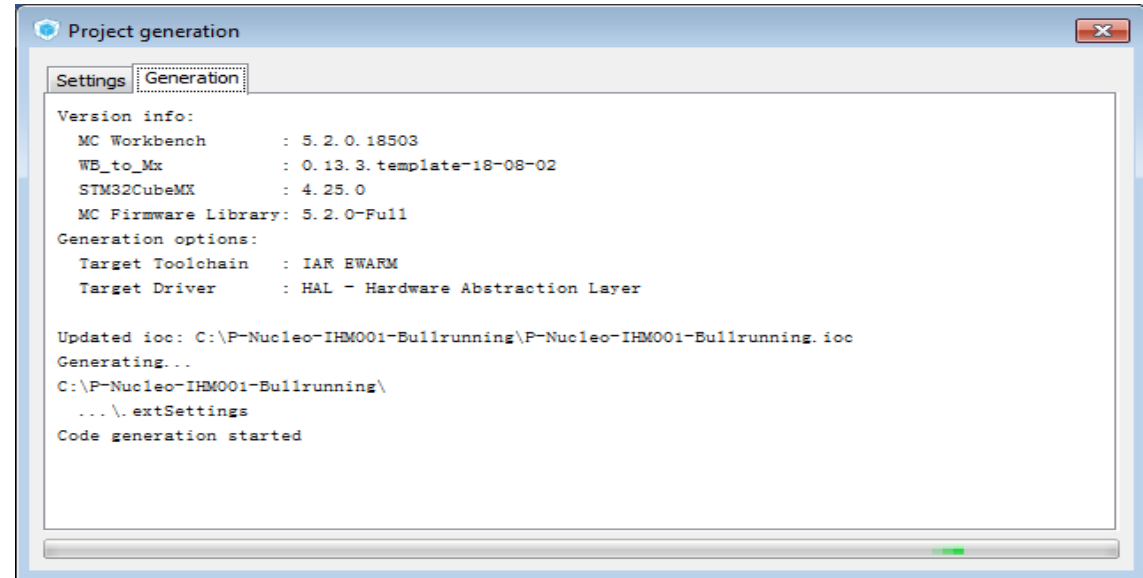
开始生成时，会显示一个进程窗口，告知用户脚本在执行，完成后的提示窗口如下图所示，可关闭该窗口。用户信息表也随之更新。



```

... \Src\parameters_conversion_1000.h
... \Ino\pmsm_motor_parameters.h
... \Ino\power_stage_parameters.h
... \Src\stm32f30x_mo_it.c
... \Ino\mo_parameters.h
... \Src\mo_parameters.c
... \Src\ui_task.c
... \Ino\ui_task.h
... \Src\user_interface.c
... \Ino\user_interface.h
... \Src\stm32f3xx_it.c
... \Ino\stm32f3xx_it.h
... \Src\stm32f3xx_hal_msp.c
... \Ino\stm32f3xx_hal_conf.h
... \Ino\main.h
... \Src\main.c

Completed
  
```



```

Project generation

Settings  Generation

Version info:
MC Workbench      : 5.2.0.18503
WB_to_Mx          : 0.13.3.template-18-08-02
STM32CubeMX       : 4.25.0
MC Firmware Library: 5.2.0-Full

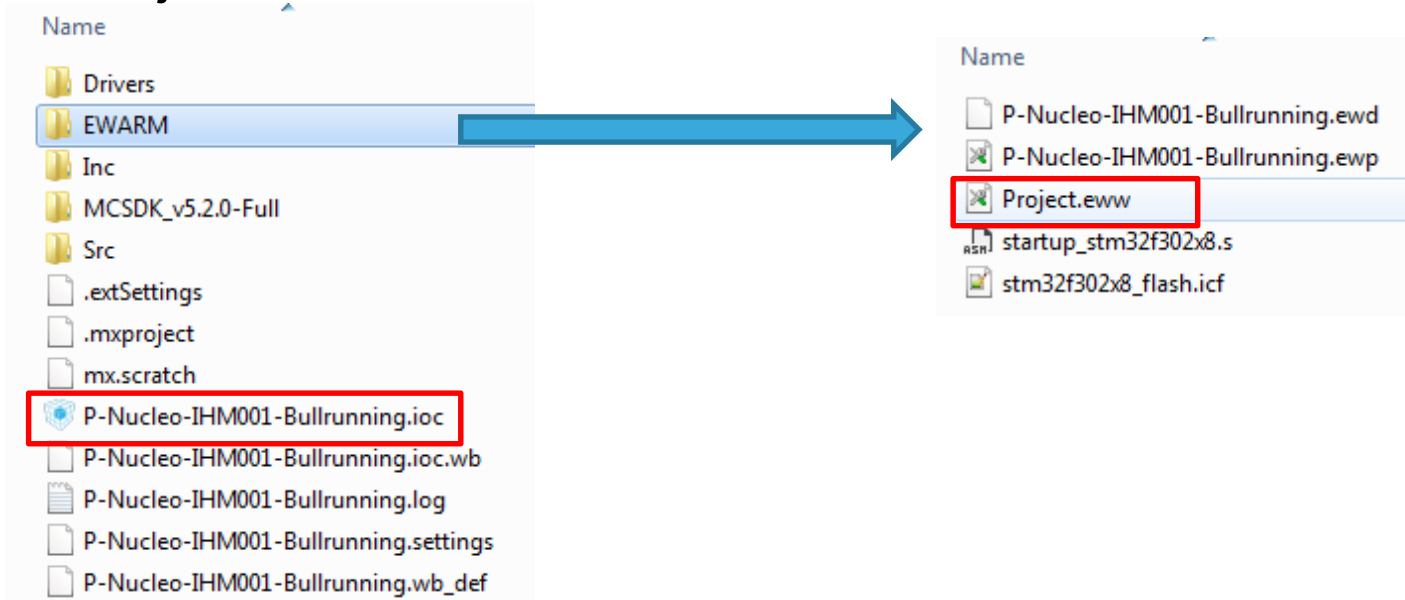
Generation options:
Target Toolchain  : IAR EWARM
Target Driver     : HAL - Hardware Abstraction Layer

Updated ioc: C:\P-Nucleo-IHM001-Bullrunning\P-Nucleo-IHM001-Bullrunning.ioc
Generating...
C:\P-Nucleo-IHM001-Bullrunning\
... \.extSettings
Code generation started
  
```

| | Time | Motor | Id | Message |
|--|----------|-------|----|--|
| | 03:01:28 | | | Generation files starting |
| | 03:03:51 | | | Project files generated on folder:'C:\P-Nucleo-IHM001-Bullrunning' |
| | | | | |
| | | | | |

Info / Errors / Warnings Change Log

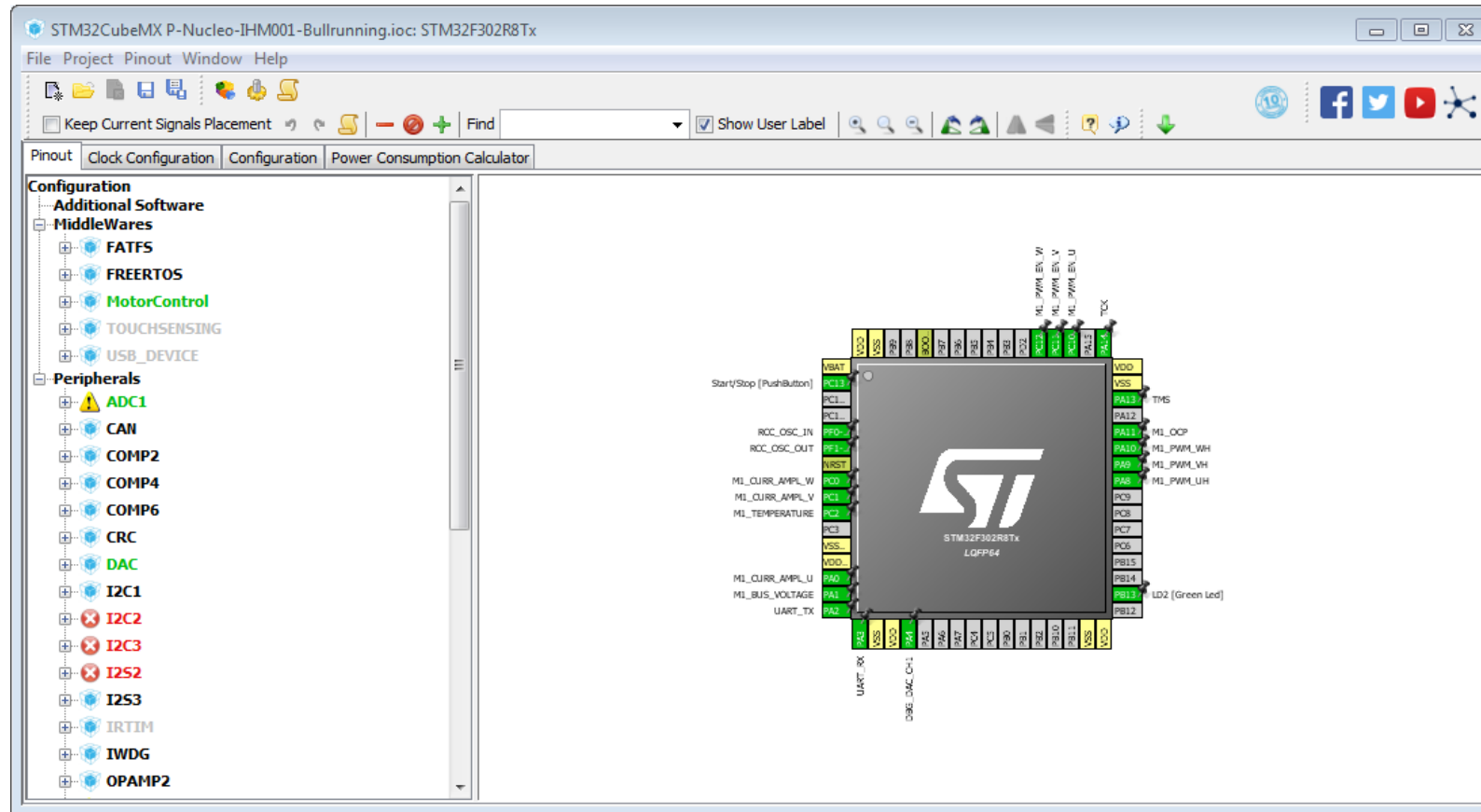
生成的IAR工程文件保存在如下路径：
\$ProjectFolder\EWARM



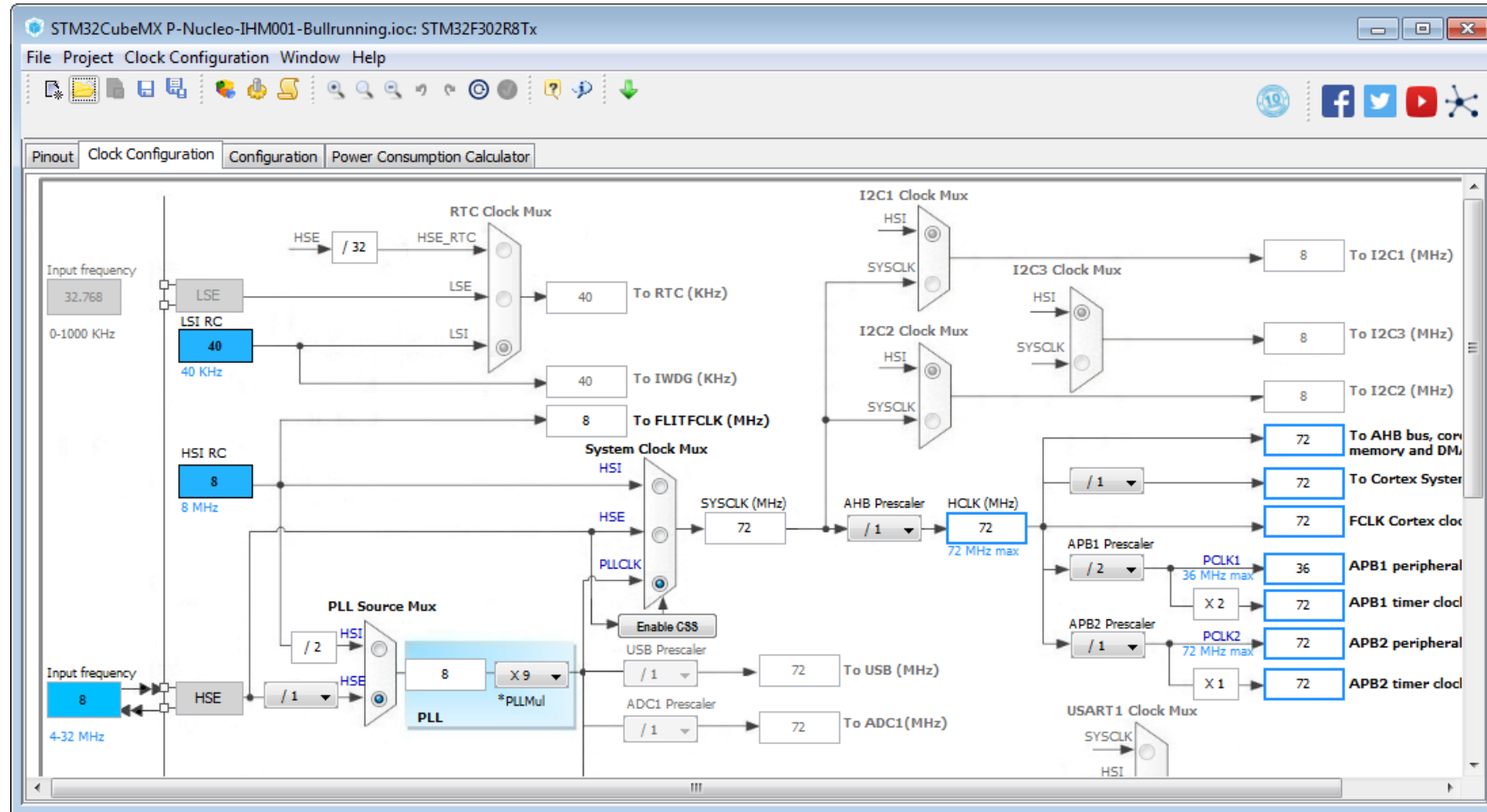
如果不需要配置其他外设，可直接打开.eww文件。

如果需要配置其他外设，请用CubeMX打开.ioc文件进行添加。

打开.ioc文件，STM32CubeMX将被打开。
图形化的引脚分配配置



时钟树结构



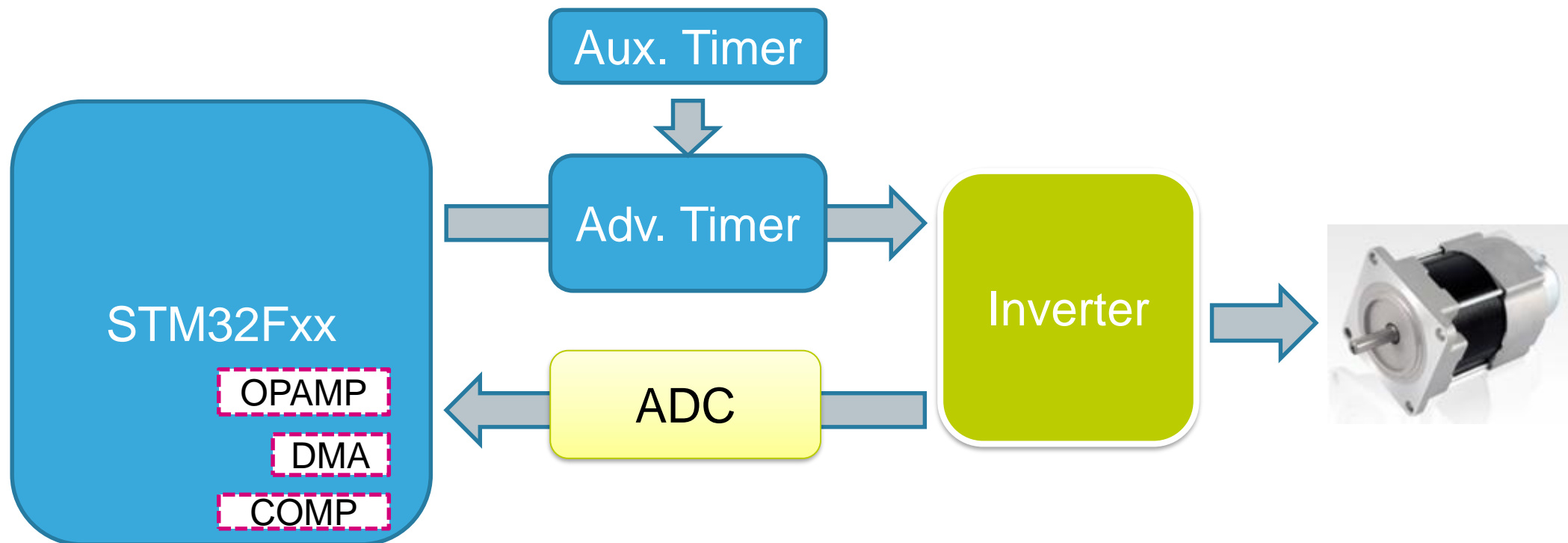
配置

The screenshot displays the STM32CubeMX configuration environment for a STM32F302R8Tx. Two configuration windows are open:

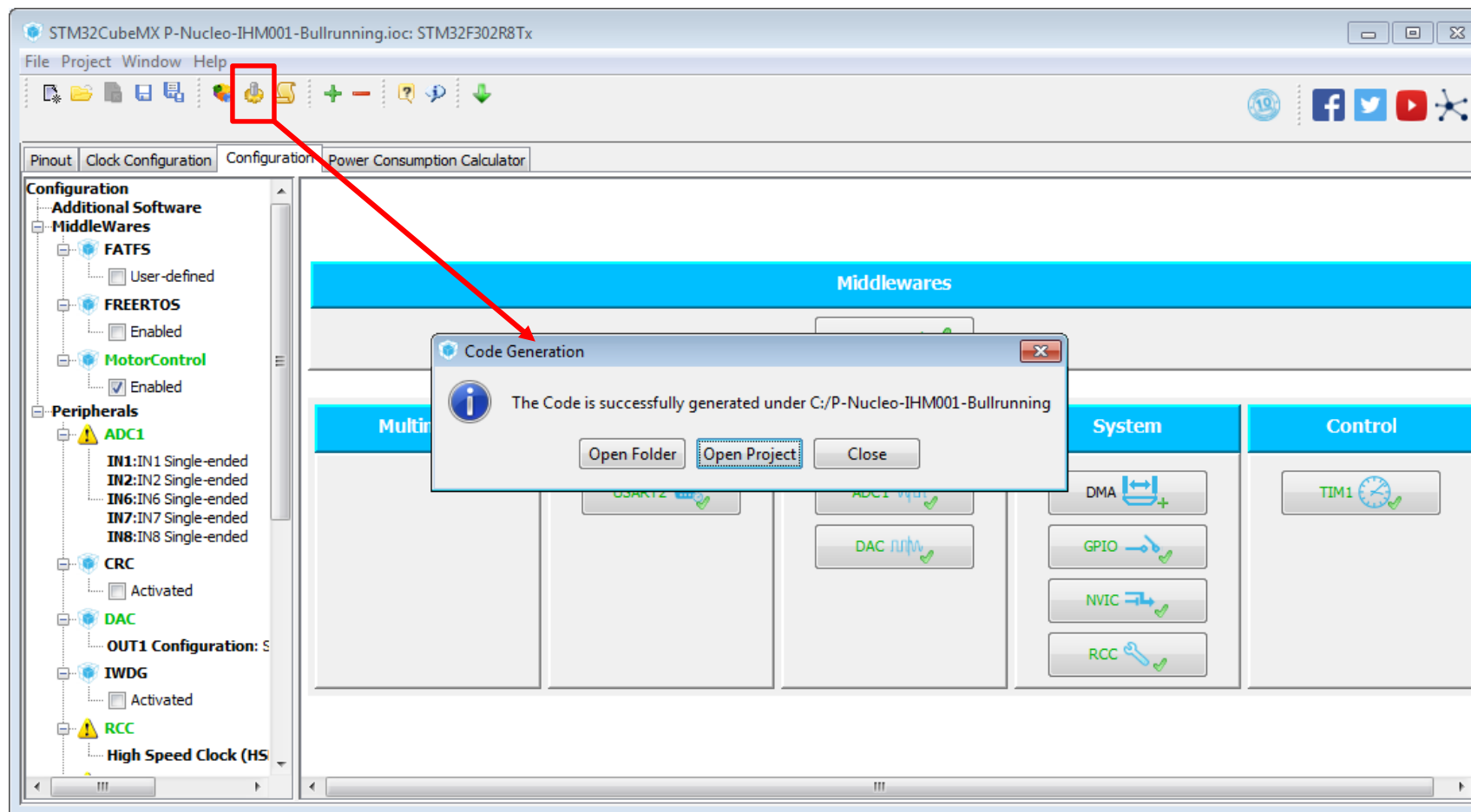
- ADC1 Configuration:** Shows parameters for the ADC1 peripheral. The 'ADC_Settings' section includes:
 - Clock Prescaler: Synchronous clock mode divided by 1
 - Resolution: ADC 12-bit resolution
 - Data Alignment: Left alignment
 - Scan Conversion Mode: Enabled
 - Continuous Conversion Mode: Disabled
 - Discontinuous Conversion Mode: Disabled
 - DMA Continuous Requests: Disabled
 - End Of Conversion Selection: End of single conversion
 - Overrun behaviour: Overrun data preserved
 - Low Power Auto Wait: Disabled
- TIM1 Configuration:** Shows parameters for the TIM1 peripheral. The 'Counter Settings' section includes:
 - Prescaler (PSC - 16 bits value): ((TIM_CLOCK_DIVIDER) - 1)
 - Counter Mode: Center Aligned mode 1
 - Counter Period (AutoReload Register - 16 bits val...): ((PWM_PERIOD_CYCLES) / 2)
 - Internal Clock Division (CKD): Division by 2
 - Repetition Counter (RCR - 16 bits value): (REP_COUNTER)
 - auto-reload preload: Disable
 - Slave Mode Controller: Trigger Mode

The background interface shows various peripheral configuration buttons such as MotorControl, ADC1, DAC, DMA, GPIO, NVIC, and RCC, with a red arrow pointing to the TIM1 button.

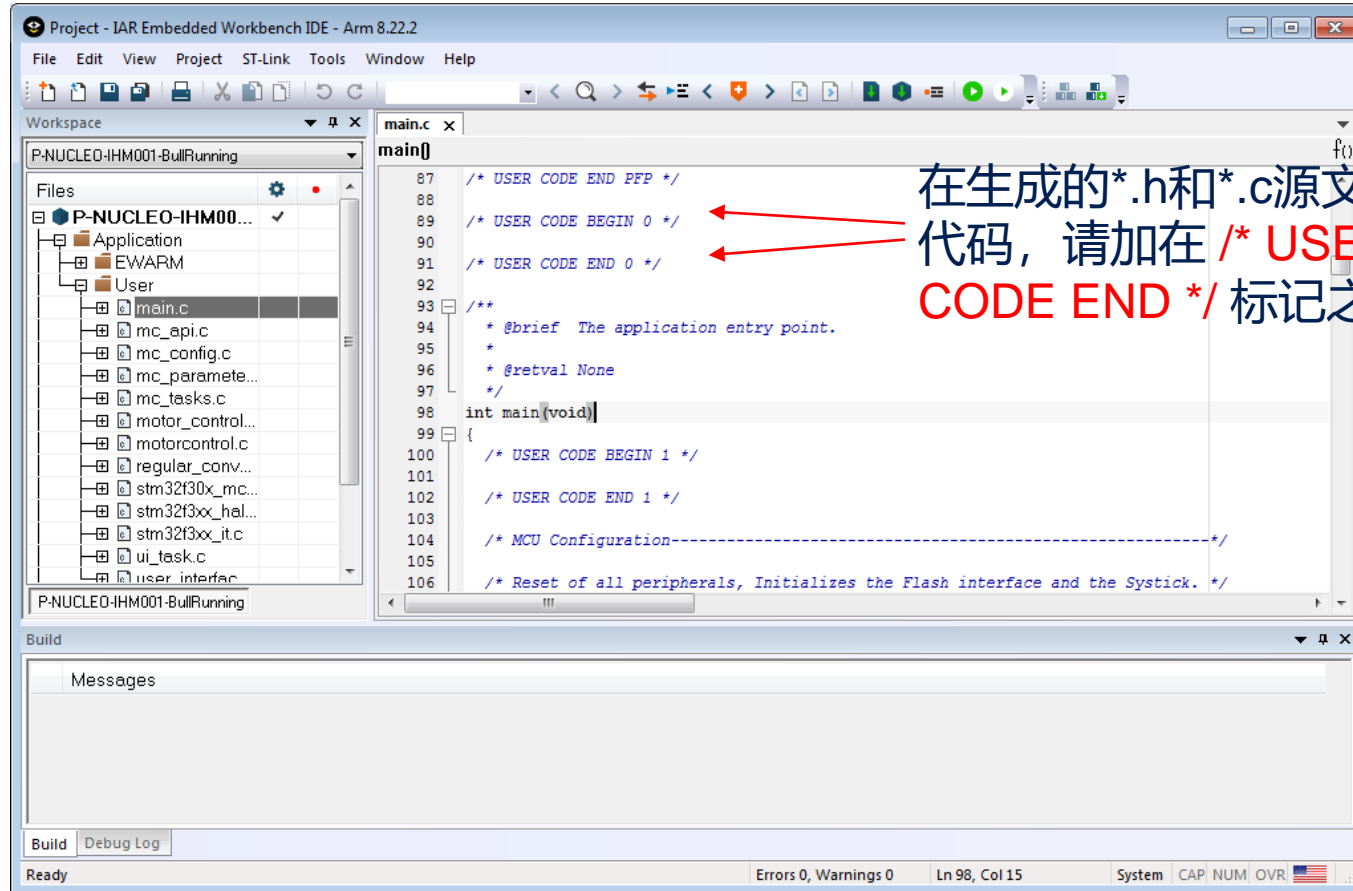
在CubeMX中请不要修改电机控制相关外设的设置!



代码生成:

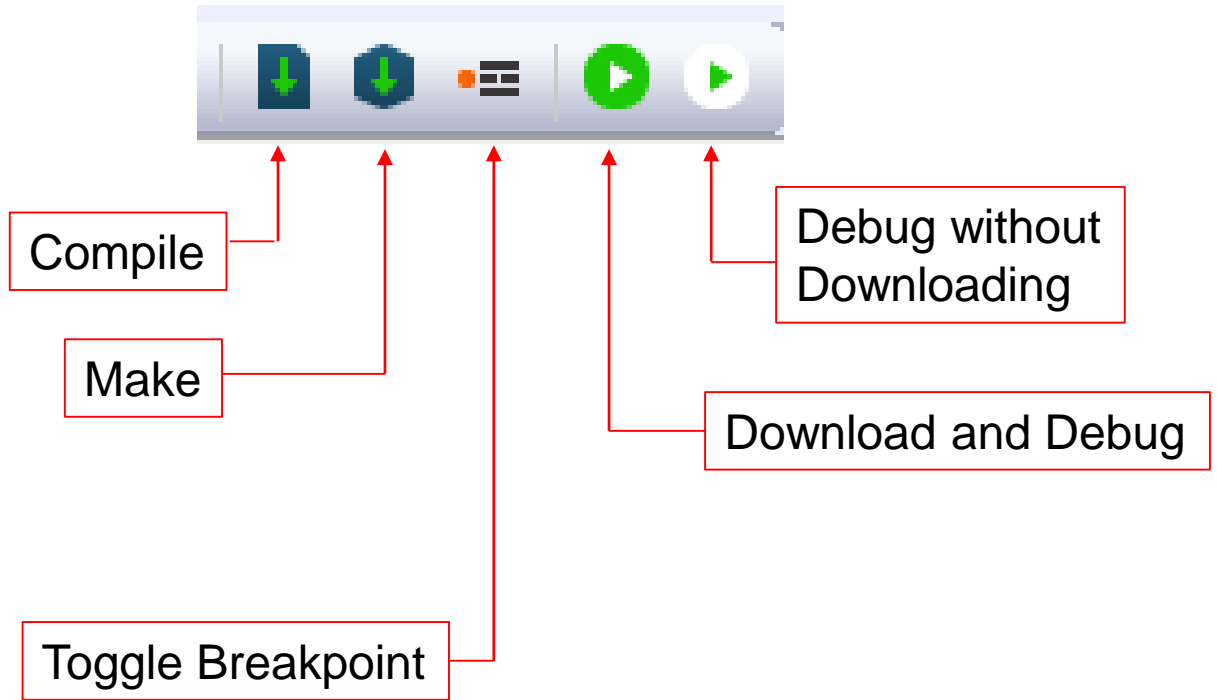
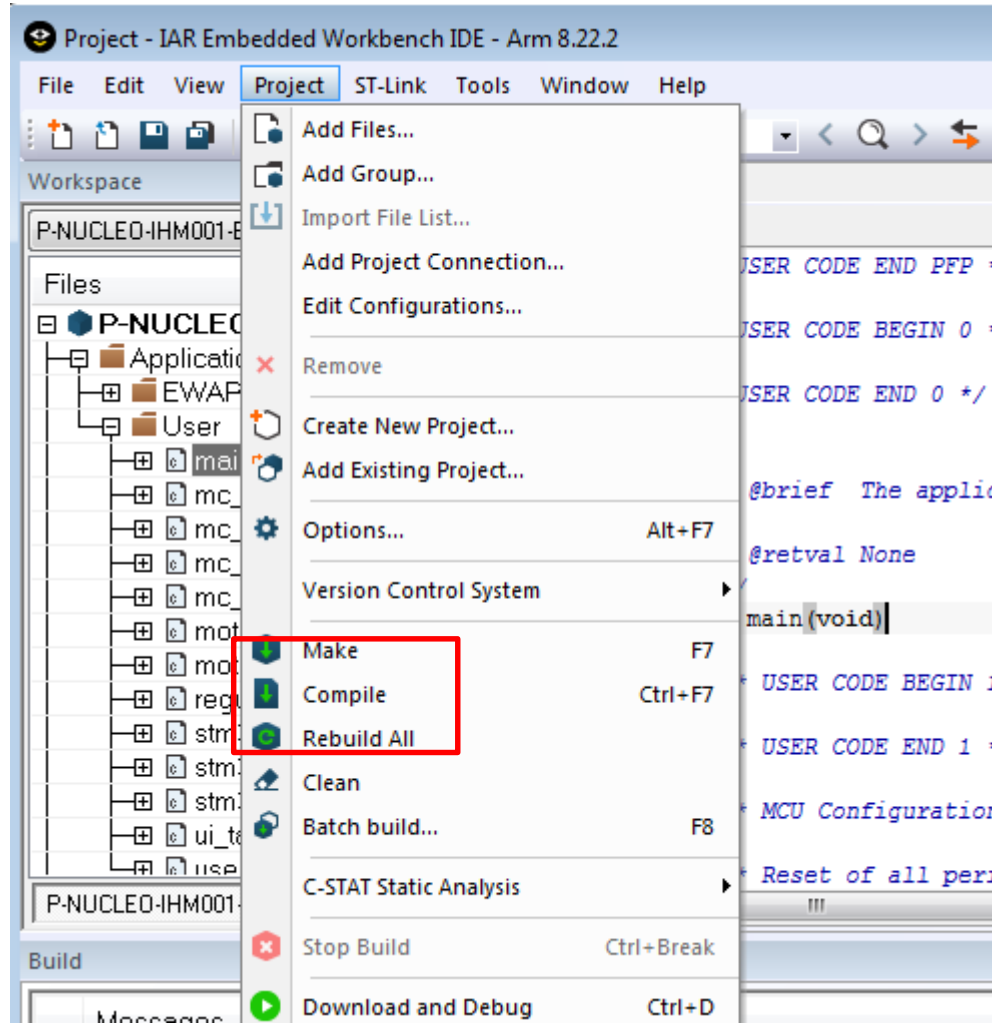


IAR Embedded Workbench for Arm (v8.xx)



在生成的*.h和*.c源文件中，用户如果添加自己的应用代码，请加在 `/* USER CODE BEGIN */`和`/* USER CODE END */` 标记之间。

在IAR Embedded Workbench for Arm v8.xx中，C编译器的优化选择请设定为 **Speed**。



The screenshot displays the ST Motor Control Workbench interface. At the top, a menu bar includes File, Tools, Help, and Documentation. Below it is a toolbar with icons for file operations and a 'Close Monitor' button. A red box highlights the toolbar area, with the label '通信链路' (Communication Link) in red text. The main dashboard area, also outlined in red, contains several gauges: 'Bus Voltage (Volt)' (0.0 to 20.0), 'Motor Power (W)' (0.0 to 200.0), 'Heatsink' (0 to 100.0), 'Measured speed (rpm)' (-15000 to 15000), and 'Final ramp speed (rpm)' (-15000 to 15000). A red label 'Dashboard area' points to this section. On the left, a 'Status' panel shows 'Not connected' and a list of faults including FOC duration, Over voltage, Under voltage, Overheat, Start-up failure, Speed feedback, Over current, and Software error. A 'Monitor' section shows 'Measured speed (rpm)' at 0. On the right, a 'Generic' control panel features buttons for 'Start Motor', 'Stop Motor', 'Stop Ramp', 'Fault Ack', 'Encoder Align', and 'PFC' (Enable, Disable, Fault Ack). A red label '电机控制按钮' (Motor Control Buttons) points to these controls. At the bottom, a 'Variable' table lists parameters like PWM frequency (30000 Hz) and Bus voltage sensing (true). A log window shows messages such as 'The 'PFC' is not supported in the FW for SDK5.x...' and 'F2 mcus are not supported in the FW for SDK5.x'.

状态一览

Dashboard area

电机控制按钮



ST Motor Control Workbench [P-Nucleo-IHM001-Bullrunning]

File Tools Help Documentation

Port COM33 115200 Close Monitor

Firmware: ST MC SDK Ver. 5.2.0

Status: Idle

Faults: FOC duration, Over voltage, Under voltage, Overheat, Start-up failure, Speed feedback, Over current, Software error

Monitor: Measured speed (rpm) 0

Configuration and debug: Control mode Speed, Power Board Status BUS Voltage 12 Volt, Heatsink temp. 29 °C, DAC Settings Ch1 Ia, Ch2 Ib

Current controller: Torque ref (Iq) 0, Flux ref (Id) 0

Speed controller: Speed ramp, Target speed 3000 rpm, Duration 1000 millisecond, Exec ramp

PID Gains: Kp 1000, Ki 600

Sensor-less Observer+PLL: Observer C1 -1380, Observer C2 2895, PLL Kp 2234, PLL Ki 52

Sensor-less Observer+Cordic: Observer C1 0, Observer C2 0

Flux weakening tuning: Kp 0, Ki 0, BUS Voltage allowed Ref 0%, Meas 0%

Generic: Target motor Motor 1, Start Motor, Stop Motor, Stop Ramp, Fault Ack, Encoder Align, All motors

PFC: PFC Enable, PFC Disable, PFC Fault Ack

| Variable | Motor | Unit |
|------------------------------|-------------------|------|
| PWM frequency | 30000 | Hz |
| Sensor selection main | Sensor-less (O... | |
| Sensor selection aux | Sensor-less (O... | |
| Torque&Flux - Execution rate | 1 | PW |
| Bus voltage sensing | true | |

| Time | Motor | Id | Message |
|----------|-------|----|---|
| 03:35:41 | | | The 'PFC' is not supported in the FW for SDK5.x excepted for SDK for 'STM32F103 High Density'. All parameters will be disa... |
| 03:35:41 | | | The 'Sensor-less (HFI+Observer)' is not supported in the FW for SDK5.x. All parameters will be disabled. |
| 03:35:41 | | | F2 mcus are not supported in the FW for SDK5.x |
| 03:35:41 | | | F103 High Density in dual Motor mcus are not supported in the FW for SDK5.x |

Info / Errors / Warnings Change Log



ST Motor Control Workbench [P-Nucleo-IHM001-Bullrunning]

File Tools Help Documentation

Port COM33 115200 Close Monitor

Firmware: STMC SDK Ver. 5.2.0

Status: Idle

Faults: FOC duration, Over voltage, Under voltage, Overheat, Start-up failure, Speed feedback, Over current, Software error

Monitor: Measured speed (rpm) 0

| 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"/> | never |
| 0x02 | Status | | 0 | 0 | 255 | 200 | U8 | R | <input checked="" type="checkbox"/> | never |
| 0x03 | Control mode | | 1 | 0 | 255 | 500 | U8 | RW | <input checked="" type="checkbox"/> | 2018-12-28 15:3... |
| 0x04 | Speed reference | RPM | 3000 | -15000 | 15000 | 200 | S32 | R | <input checked="" type="checkbox"/> | 2018-12-28 15:3... |
| 0x05 | Speed Kp | | 1000 | 0 | 65535 | 0 | U16 | RW | <input checked="" type="checkbox"/> | 2018-12-28 15:3... |
| 0x06 | Speed Ki | | 600 | 0 | 65535 | 0 | U16 | RW | <input checked="" type="checkbox"/> | 2018-12-28 15:3... |
| 0x07 | Speed Kd | | 0 | 0 | 65535 | 0 | U16 | RW | <input checked="" type="checkbox"/> | never |
| 0x08 | Torque reference (Iq) | | 0 | -32768 | 32767 | 0 | S16 | RW | <input checked="" type="checkbox"/> | never |
| 0x09 | Torque Kp | | 1051 | 0 | 65535 | 0 | U16 | RW | <input checked="" type="checkbox"/> | 2018-12-28 15:3... |
| 0x0A | Torque Ki | | 214 | 0 | 65535 | 0 | U16 | RW | <input checked="" type="checkbox"/> | 2018-12-28 15:3... |
| 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 | | 1051 | 0 | 65535 | 0 | U16 | RW | <input checked="" type="checkbox"/> | 2018-12-28 15:3... |
| 0x0E | Flux Ki | | 214 | 0 | 65535 | 0 | U16 | RW | <input checked="" type="checkbox"/> | 2018-12-28 15:3... |
| 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"/> | 2018-12-28 15:3... |
| 0x11 | Observer C2 | | 2895 | -32768 | 32767 | 0 | S16 | RW | <input checked="" type="checkbox"/> | 2018-12-28 15:3... |
| 0x12 | Cordic Observer C1 | | 0 | -32768 | 32767 | 0 | S16 | RW | <input type="checkbox"/> | never |
| 0x13 | Cordic Observer C2 | | 0 | -32768 | 32767 | 0 | S16 | RW | <input type="checkbox"/> | never |


Generic: Target motor Motor 1 Start Motor Stop Motor Stop Ramp Fault Ack Encoder Align All motors

PFC: PFC Enable PFC Disable PFC Fault Ack

Variable: Motor 30000 Hz, Sensor selection main Sensor-less (O...), Sensor selection aux Sensor-less (O...), Torque&Flux - Execution rate 1 PW, Bus voltage sensing true

| Time | Motor | Id | Message |
|----------|-------|----|---|
| 03:35:41 | | | The 'PFC' is not supported in the FW for SDK5.x excepted for SDK for 'STM32F103 High Density'. All parameters will be disa... |
| 03:35:41 | | | The 'Sensor-less (HFI+Observer)' is not supported in the FW for SDK5.x. All parameters will be disabled. |
| 03:35:41 | | | F2 mcus are not supported in the FW for SDK5.x |
| 03:35:41 | | | F103 High Density in dual Motor mcus are not supported in the FW for SDK5.x |

Info / Errors / Warnings Change Log




ST Motor Control Workbench [P-Nucleo-IHM001-Bullrunning]

File Tools Help Documentation

Port COM33 115200 Close Monitor

Status: Idle

Faults: FOC duration, Over voltage, Under voltage, Overheat, Start-up failure, Speed feedback, Over current, Software error

Monitor: Measured speed (rpm) 0

Configuration: Firmware: ST MC SDK Ver. 5.2.0

Board Configuration: Motor available: Single Motor

Motor 1 or any motor: Sensor-less (Obs+PLL), Sensor-less (Obs+Cordic), Sensor-less (HFI+Obs), Quadrature encoder, Hall sensors, Flux weakening, DAC channels

Control mode: Speed
Min speed: -15000
Max speed: 15000
Max bus voltage: 36.0

Motor 2 (if available): Sensor-less (Obs+PLL), Sensor-less (Obs+Cordic), Sensor-less (HFI+Obs), Quadrature encoder, Hall sensors, Flux weakening

Control mode: none
Min speed: -5000

Revup Configuration

| Num | Final Speed (rpm) | Final Torque | Duration (ms) | Last read | Last write |
|-----|-------------------|--------------|---------------|-------------------------|------------|
| 1 | 0 | 0 | 0 | 2018-12-28 15:37:34.292 | never |
| 2 | 0 | 10026 | 300 | 2018-12-28 15:37:34.355 | never |
| 3 | 132 | 12032 | 100 | 2018-12-28 15:37:34.417 | never |
| 4 | 3996 | 12032 | 2900 | 2018-12-28 15:37:34.479 | never |
| 5 | 3996 | 12032 | 0 | 2018-12-28 15:37:34.542 | never |

Generic: Target motor: Motor 1
Start Motor, Stop Motor, Stop Ramp, Fault Ack, Encoder Align, All motors

PFC: PFC Enable, PFC Disable, PFC Fault Ack

ST life.augmented

| Variable | Motor | Uni |
|------------------------------|-------------------|-----|
| PWM frequency | 30000 | Hz |
| Sensor selection main | Sensor-less (O... | |
| Sensor selection aux | Sensor-less (O... | |
| Torque&Flux - Execution rate | 1 | PW |
| Bus voltage sensing | true | |

| Time | Motor | Id | Message |
|----------|-------|----|---|
| 03:35:41 | | | The 'PFC' is not supported in the FW for SDK5.x excepted for SDK for 'STM32F103 High Density'. All parameters will be disa... |
| 03:35:41 | | | The 'Sensor-less (HFI+Observer)' is not supported in the FW for SDK5.x. All parameters will be disabled. |
| 03:35:41 | | | F2 mcus are not supported in the FW for SDK5.x |
| 03:35:41 | | | F103 High Density in dual Motor mcus are not supported in the FW for SDK5.x |

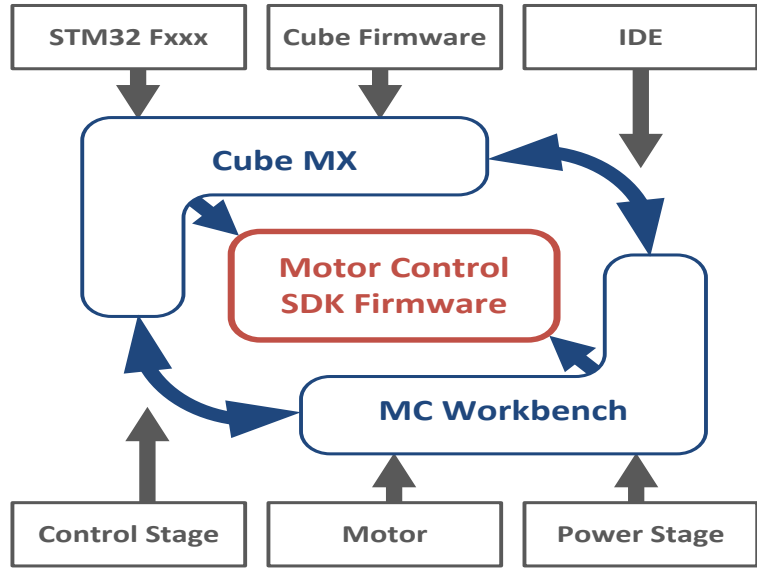
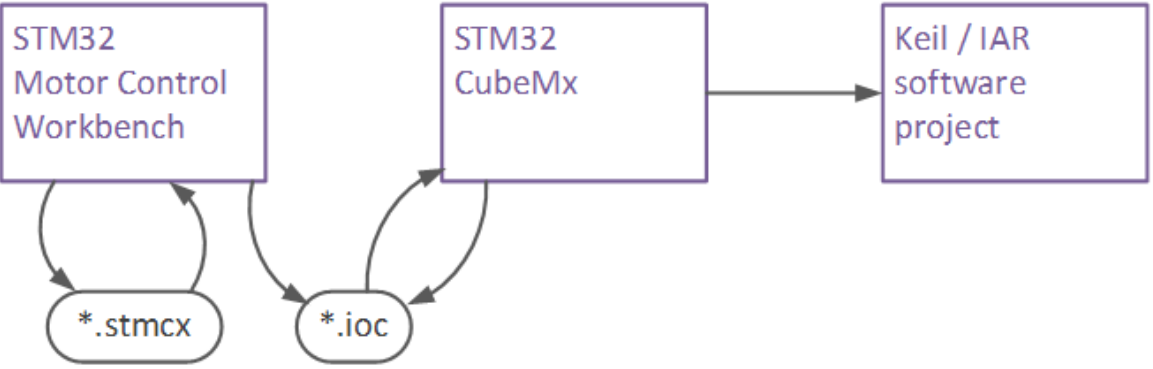
Info / Errors / Warnings Change Log



ST MC SDK5.x固件详解

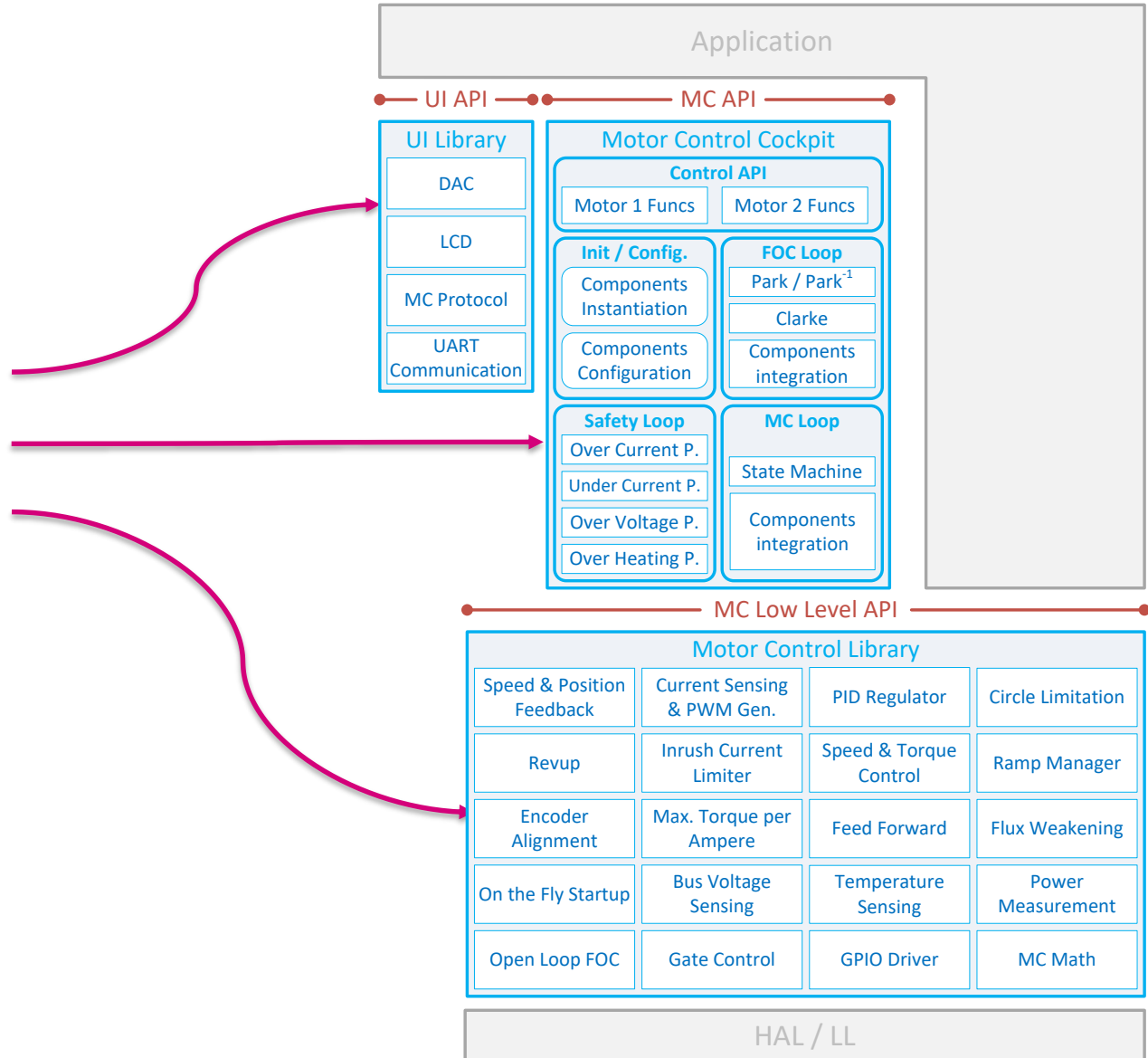
- ST MC SDK5.x WB应用指南
 - 软件工具的下载和安装
 - ST MC Workbench及关键配置参数
 - MC Project 的生成, 编译和下载
 - 电机控制及监控
- ST MC SDK5.x固件详解
 - 程序架构
 - 组件
 - 例程代码讲解
 - 开发实战
 - 如何向例程中添加外设和自己的代码
 - Step-by-Step添加一段闪灯代码

代码生成流程



整个软件库由三部分组成

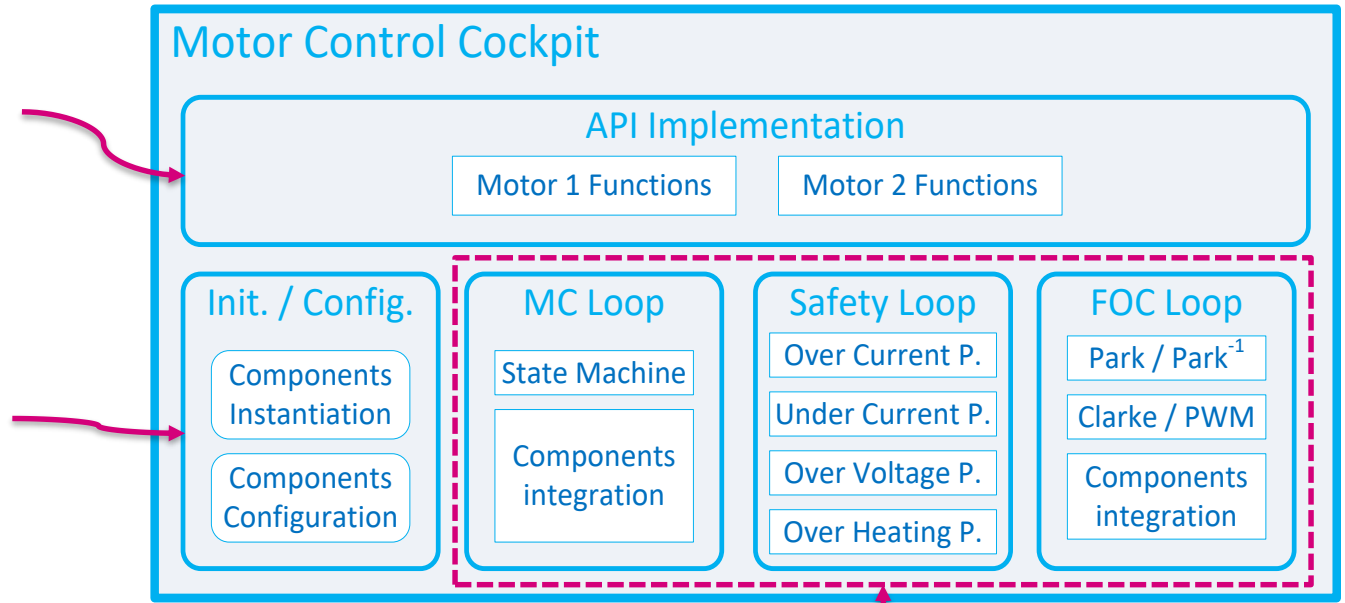
- 用户界面库 UI Library
- 电机驾驶舱 Motor Control Cockpit
- 电机控制库 Motor Control Library



电机驾驶舱 Motor Control Cockpit 由三部分构成

电机控制接口通过 MC API来实现

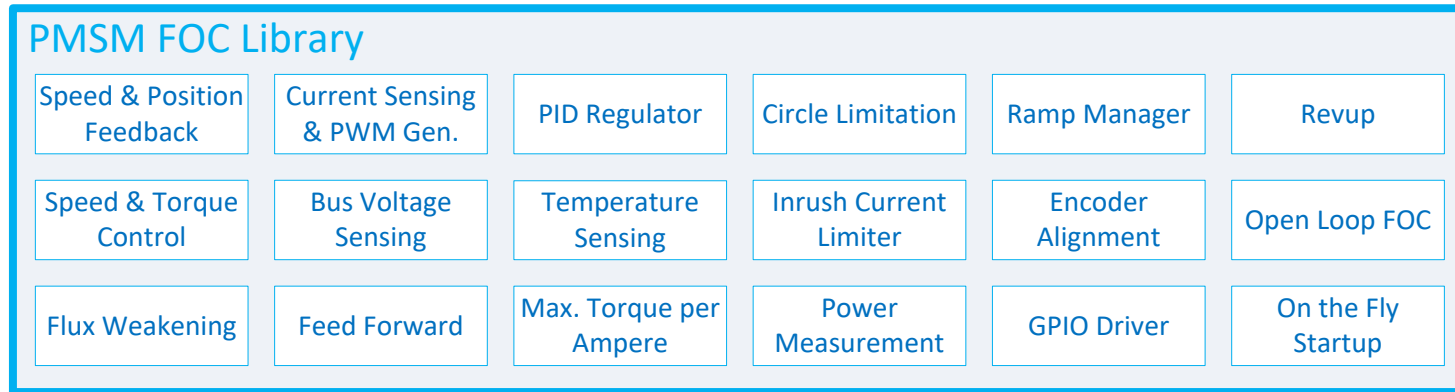
电机控制配置实例化并配置所有需要的组件。



电机控制动态实现对电机的动态控制:

- FOC控制环路(高频任务)
- 电机控制环路(中频任务)
- 安全控制环路 (安全任务)

电机控制库

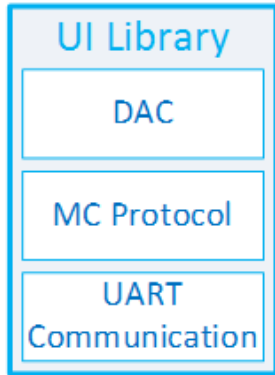


是诸多组件的集合。每一个组件实现电机控制的一个功能例如，速度和位置检测, 电流检测, PID算法等等...

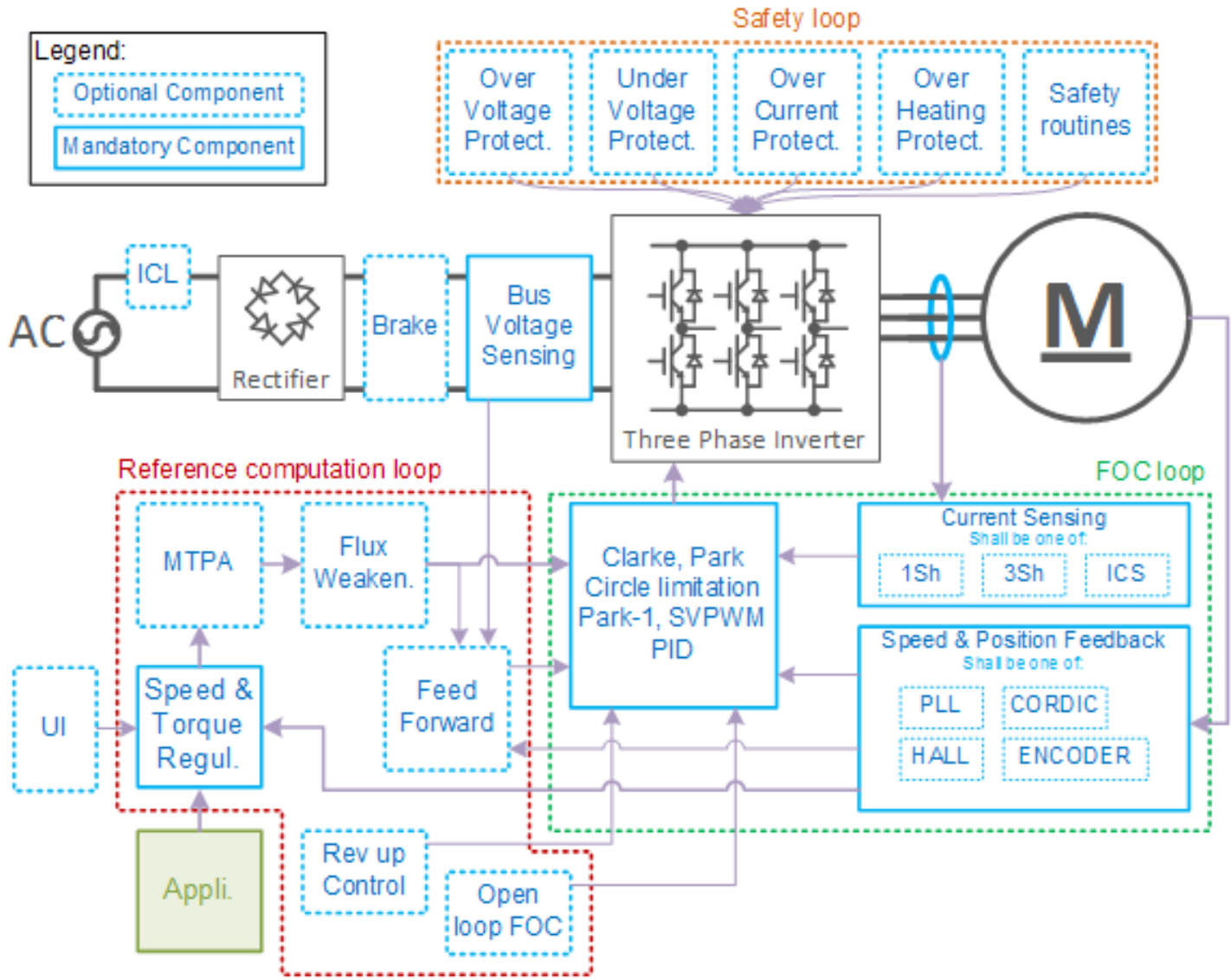
下列组件均以库的形式提供。用户无法看到源码，但可以根据提供的头文件从库里调用相应的函数。

| 组件名称 | 描述 |
|-----------------------------|---------------|
| feed_forward_ctrl.c | 前馈控制 |
| flux_weakening_ctrl.c | 弱磁控制 |
| max_torque_per_ampere.c | 最大转矩控制 |
| sto_cordic_speed_pos_fdbk.c | 速度和位置反馈cordic |
| sto_pll_speed_pos_fdbk.c | 速度和位置反馈PLL |
| revup_ctrl.c | 启动控制 |

用户界面库



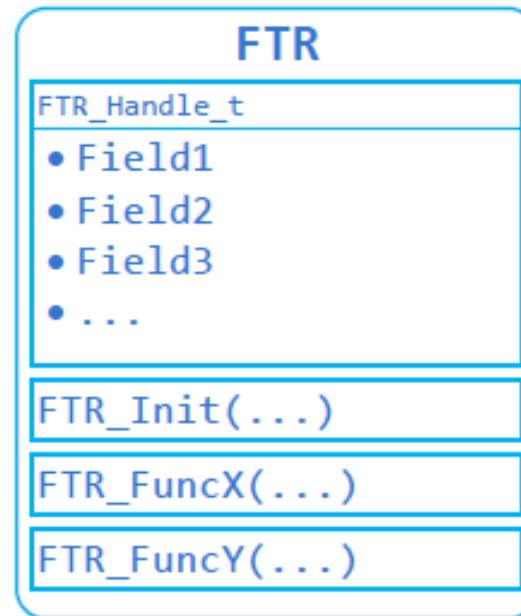
用户界面库包含负责通讯的组件。电机控制代码通过这些组件控制串口和 DAC 与外界通讯。通过这个库我们可以连接 MCU 和 Workbench 。在 Workbench 中实现对电机运行状态的监控。



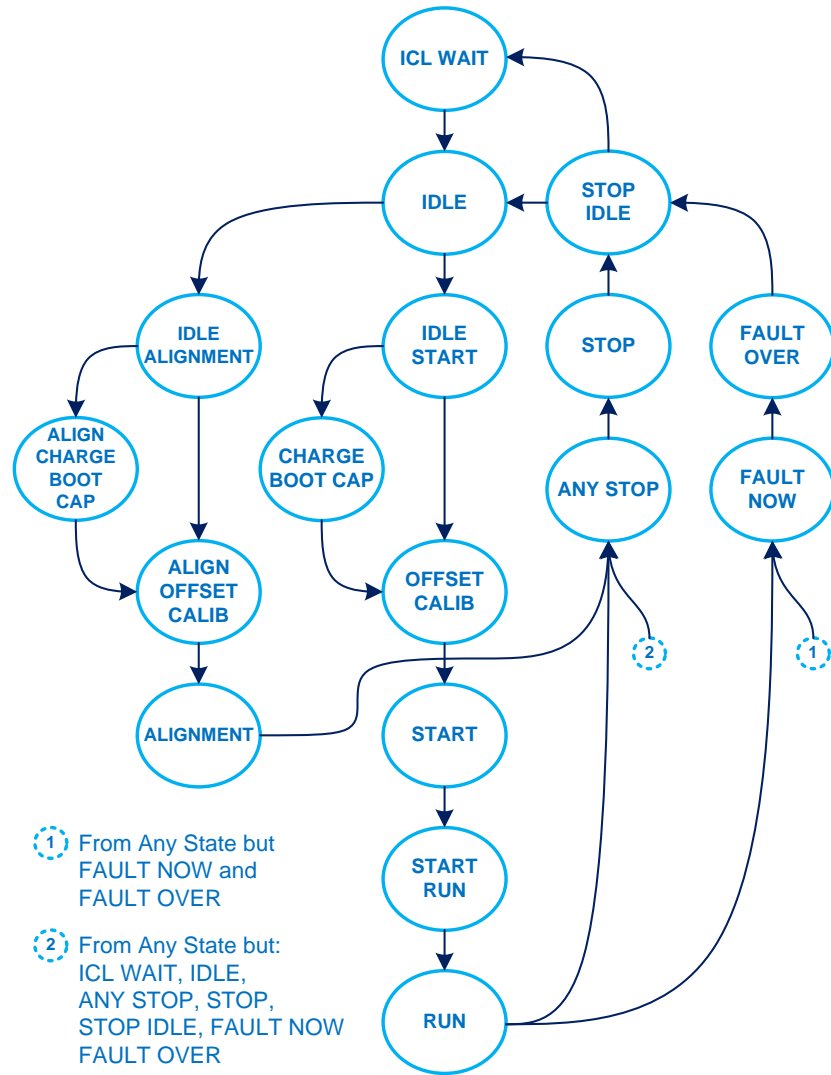
- FOC 控制环路 loop
- 电机控制环路 loop
- 安全控制环路

电机控制库是由我们称之为**组件**(component) 的单元构成的。组件是一个自给自足的软件单元，他包含两个部分：

- 包含一个能满足此组件功能的数据的结构体。
- 一系列的函数，这些函数操作组件的结构体变量以实现组件的功能。



组件通常包含一个 **.c** 文件和一个 **.h** 文件。结构体以 **组件名缩写 + _Handle_t** 命名，函数以 **组件名缩写 + 函数功能命名**



电机控制状态机

- 程序用状态机来控制电机状态。
- 电机在如图的状态之间来回切换。
- SDK 5.x 中用 `state_machine` 组件来实现状态机的功能。

- `state_machine` 这个组件包含两个文件 `state_machine.c` 和 `state_machine.h`。
- 结构体以 `STM_Handle_t` 命名。
- 初始化函数以 `STM_Init` 命名。函数的参数是指向结构体变量 `STM_Handle_t` 的指针 `pHandle`。

```
typedef struct
{
    State_t    bState;

    uint16_t   hFaultNow;

    uint16_t   hFaultOccurred;

} STM_Handle_t;
```

结构体

```
void STM_Init( STM_Handle_t * pHandle );

bool STM_NextState( STM_Handle_t * pHandle, State_t bState );

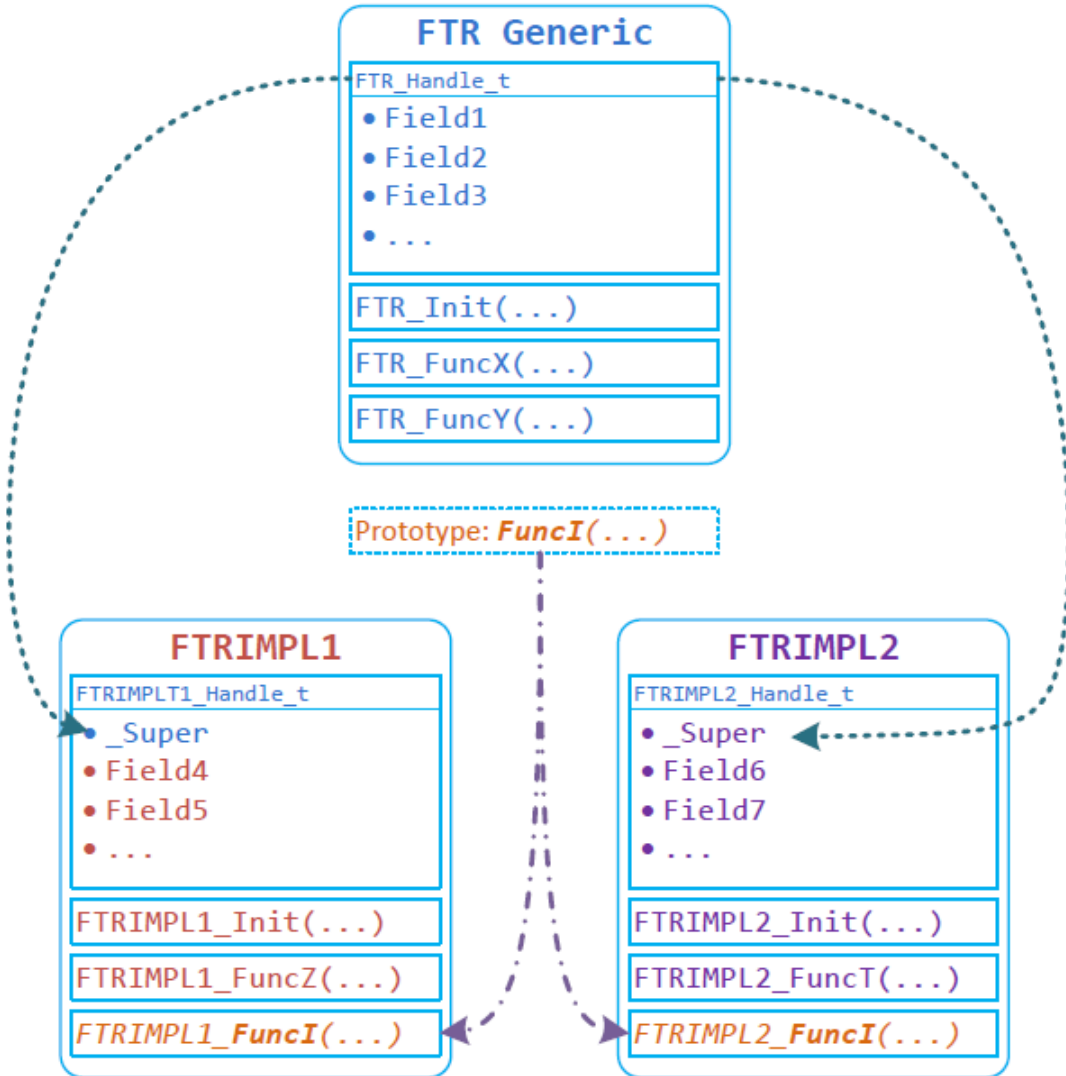
State_t STM_FaultProcessing( STM_Handle_t * pHandle, uint16_t hSetErrors, uint16_t
                             hResetErrors );

State_t STM_GetState( STM_Handle_t * pHandle );

bool STM_FaultAcknowledged( STM_Handle_t * pHandle );

uint32_t STM_GetFaultState( STM_Handle_t * pHandle );
```

函数



- 同一个功能有很多不同的具体实现方式，这些具体实现方式既包含共性的部分又有各自的个性。
- SDK 5.x 用**通用组件**来定义共性的部分，用**执行组件**来定义具体实现。
 - 通用组件包含公共的数据、函数和函数原型。
 - 执行组件包含共性加个性的部分。
- 图中的 **FTR** 是实现某个功能的通用组件。**FRTIMPL1** 和 **FRTIMPL2** 是这个功能的执行组件。
- 这两个执行组件的结构体中都有一个成员叫 `_Super`。执行组件通过成员 `_Super` 实现对 FTR 所定义的公共资源的复用。
- 通用组件 FTR 中定义一系列的函数原型如 `FuncI(...)`，并在其结构体中定义了这个函数原型的函数指针。
- FRTIMPL1 和 FRTIMPL2 分别定义函数 `FTRIMPL1_FuncI(...)` 和 `FTRIMPL2_FuncI(...)` 来具体实现通用组件定义的函数原型。

下面以 SDK 5.x 自带的例程代码 **P-NUCLEO-IHM001-BullRunning** 中的 PWM 和电流采样功能为例，进一步说明通用组件及其执行组件。

- PWM 和电流采样功能可以采用单电阻，3电阻或者电流传感器来实现
- SDK 5.x 定义了通用组件 **PWMC** 来实现这个功能的通用部分。
- 通用组件 PWMC 有很多执行组件。我们以其中的 **R3_1_F30X** 和 **R1F30X** 这两个执行组件为例。

通用组件 **PWMC** 中定义的结构体 **PWMC_Handle_t**。这个结构体的成员中不仅有数据变量，还有函数指针。如下图中的 **pFctGetPhaseCurrents**

```
typedef struct PWMC_Handle PWMC_Handle_t;

struct PWMC_Handle
{
    /** @{ */
    PWMC_IrqHandler_Cb_t pFctIrqHandler;          /**< pointer on the interrupt handling function. */
    PWMC_GetPhaseCurr_Cb_t
    pFctGetPhaseCurrents;                        /**< pointer on the function the component instance uses to retrieve phase currents */
    PWMC_Generic_Cb_t
    pFctSwitchOffPwm;                            /**< pointer on the function the component instance uses to switch PWM off */
    /** @} */
};
```

通用组件定义了这个函数指针的函数原型 **PWMC_GetPhaseCurr_Cb_t**。

```
typedef void ( *PWMC_GetPhaseCurr_Cb_t ) ( PWMC_Handle_t * pHandle, Curr_Components * pStator_Currents );
```

```
typedef struct
{
    PwMCHandle_t_Super;
    uint32_t wPhaseAOffset;
    uint32_t wPhaseBOffset;
    uint32_t wPhaseCOffset;
    uint16_t Half_PWMPeriod;
    uint16_t hRegConv;

    .
    .
    .
    .
    .
    .

    volatile uint8_t bIndex;
    uint16_t ADC_ExternalTriggerInjected;
    /*!< Trigger selection for ADC peripheral*/
    bool OverCurrentFlag; /*!< This flag*/
    bool OverVoltageFlag; /*!< This flag*/
    bool BrakeActionLock; /*!< This flag*/
    interrupt

    R3_1_F30XParams_t const * pParams_str;
} PwMCHandle_R3_1_F3_Handle_t;
```

```
typedef struct
{
    PwMCHandle_t_Super;
    uint16_t hDmaBuff[2];
    uint16_t hCntSmp1;
    uint16_t hCntSmp2;
    uint8_t sampCur1;
    uint8_t sampCur2;

    .
    .
    .
    .
    .
    .

    DMA_Channel_TypeDef * PreloadDMAy_Chx; /*
    DMA_Channel_TypeDef * DistortionDMAy_Chx; /*
    bool OverCurrentFlag; /*!< This flag*/
    bool OverVoltageFlag; /*!< This flag*/
    bool BrakeActionLock; /*!< This flag*/

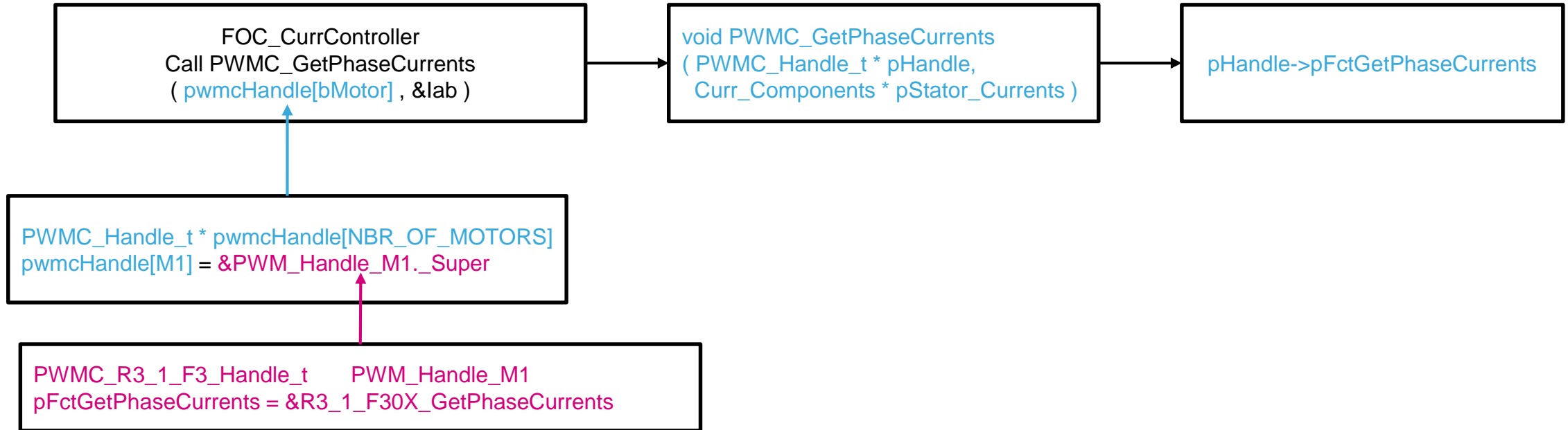
    R1_F30XParams_t const * pParams_str;
} PwMCHandle_R1_F3_Handle_t;
```

- 组件 **R3_1_F30X** 和 **R1F30X** 的结构体中都有一个成员 **_Super**。通过 **_Super** 的类型是通用组件 **PwMCHandle_t** 定义的结构体 **PwMCHandle_t**。
- **R3_1_F30X** 和 **R1F30X** 的实例会把 **_Super** 中的 **pFctGetPhaseCurrents** 指向各自的实现函数。


```
/* Change back function to be executed in ADCx_ISR */
pHandle->_Super.pFctGetPhaseCurrents = &R3_1_F30X_GetPhaseCurrents;

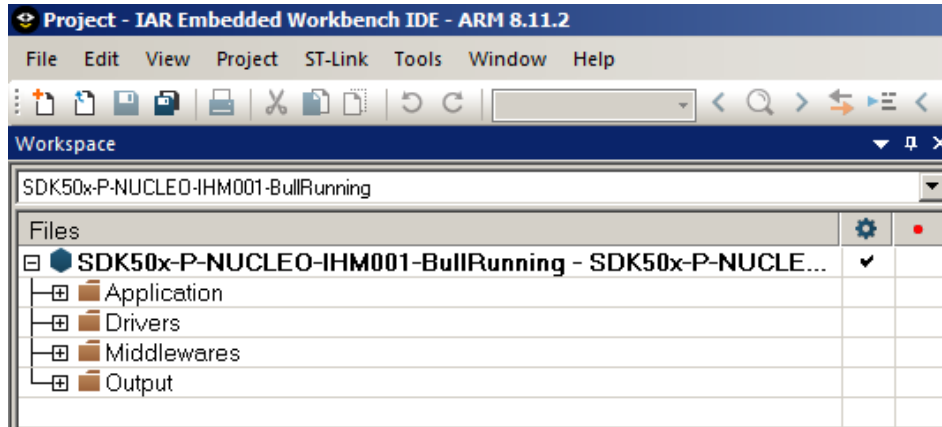
/* Change back function to be executed in ADCx_ISR */
pHandle->_Super.pFctGetPhaseCurrents = &R1F30X_GetPhaseCurrents;
```
- **R3_1_F30X** 和 **R1F30X** 除了通用组件中公有的部分还各自定义了自己独有的数据。

以读取电流值这个功能为例，说明例程代码 P-NUCLEO-IHM001-BullRunning 如何使用通用组件 **PWMC** 和执行组件 **R3_1_F30X**



下列组件是SDK5.x中通用组件的结构体及其执行组件结构体

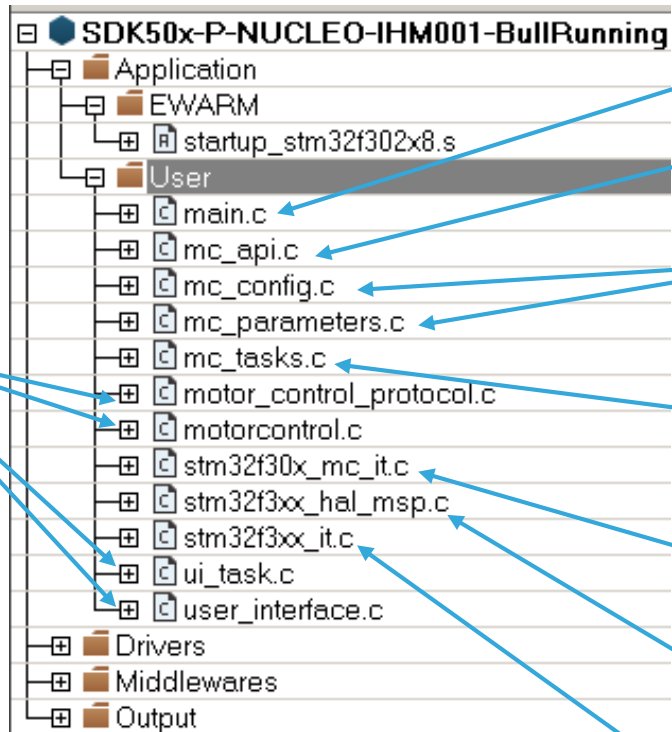
| 通用组件结构体 | 执行组件结构体 | 执行组件结构体 | 执行组件结构体 |
|---------------------------|-----------------------|----------------------------------|------------------------|
| BusVoltageSensor_Handle_t | RDivider_Handle_t | VirtualBusVoltageSensor_Handle_t | |
| SpeednPosFdbk_Handle_t | ENCODER_Handle_t | HALL_Handle_t | STO_Handle_t |
| | STO_CR_Handle_t | VirtualSpeedSensor_Handle_t | STO_PLL_Handle_t |
| UI_Handle_t | DAC_UI_Handle_t | MCP_Handle_t | UDFastCom_Handle_t |
| FCP_Handle_t | UFCP_Handle_t | | |
| PWMC_Handle_t | PWMC_ICS_F3_Handle_t | PWMC_ICS_F4_Handle_t | PWMC_ICS_F7_Handle_t |
| | PWMC_ICS_HD2_Handle_t | PWMC_ICS_L4_Handle_t | PWMC_ICS_LM1_Handle_t |
| | PWMC_R1_F0_Handle_t | PWMC_R1_F3_Handle_t | PWMC_R1_F4_Handle_t |
| | PWMC_R1_F7_Handle_t | PWMC_R1_HD2_Handle_t | PWMC_R1_L4_Handle_t |
| | PWMC_R1_VL1_Handle_t | PWMC_R3_1_F3_Handle_t | PWMC_R3_1_F4_Handle_t |
| | PWMC_R3_1_F7_Handle_t | PWMC_R3_1_L4_Handle_t | PWMC_R3_2_F3_Handle_t |
| | PWMC_R3_2_F7_Handle_t | PWMC_R3_2_L4_Handle_t | PWMC_R3_32_G4_Handle_t |
| | PWMC_R3_3_G4_Handle_t | PWMC_R3_4_F3_Handle_t | PWMC_R3_F0_Handle_t |
| | PWMC_R3_F4_Handle_t | PWMC_R3_HD2_Handle_t | PWMC_R3_LM1_Handle_t |



打开 **SDK 5.x** 自带的例程代码 **P-NUCLEO-IHM001-BullRunning** 的 **IAR** 工程。在 **IAR** 的 **workspace** 窗口，我们可以看到在工程下有四个文件夹。

- **Application**
和客户的应用紧密相关的程序。通常客户只需要更改这一部分的代码。我们在后面会有更详细的说明。
- **Drivers**
MCU 的外设驱动代码，由 **CubeMX** 生成。另外还包含 **ARM** 的 **CMSIS** 库。
- **Middleware**
电机控制库中的各个组件的代码和库文件。
- **Output**
链接器输出的文件如 **.out** 文件, **.map** 文件等。

UI 相关



main 函数。

用户与 SDK 交互的接口函数。

实例化各组件的结构体，并初始化。

电机控制的核心环路。如电流环，转矩和速度环，状态机等。

电机控制相关的外设中断服务程序。

CubeMX 生成的各外设的 IO 配置。

用户自定义的外设中断服务程序。

```
97 int main(void)
98 {
99     /* USER CODE BEGIN 1 */
100
101     /* USER CODE END 1 */
102
103     /* MCU Configuration-----
104
105     /* Reset of all peripherals, Initializes th
106     HAL_Init();
107
108     /* USER CODE BEGIN Init */
109
110     /* USER CODE END Init */
111
112     /* Configure the system clock */
113     SystemClock_Config();
114
115     /* USER CODE BEGIN SysInit */
116
117     /* USER CODE END SysInit */
118
119     /* Initialize all configured peripherals */
120     MX_GPIO_Init();
121     MX_ADC1_Init();
122     MX_DAC_Init();
123     MX_TIM1_Init();
124     MX_USART2_UART_Init();
125     MX_MotorControl_Init();
126
127     /* Initialize interrupts */
128     MX_NVIC_Init();
129     /* USER CODE BEGIN 2 */
130
131     /* USER CODE END 2 */
132
133     /* Infinite loop */
134     /* USER CODE BEGIN WHILE */
135     while (1)
136     {
137
138     /* USER CODE END WHILE */
139
140     /* USER CODE BEGIN 3 */
141
142     }
143     /* USER CODE END 3 */
144
145 }
```

在 `main.c` 文件中，函数 `int main(void)` 包含以下内容：

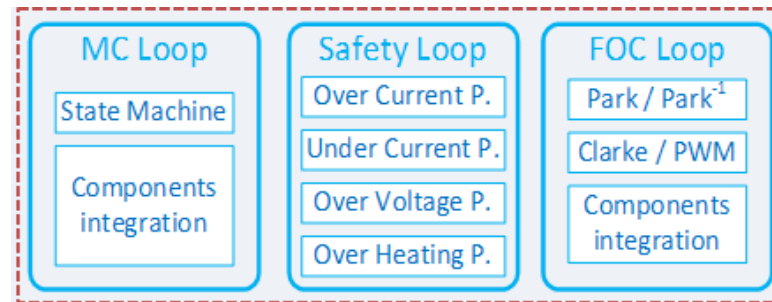
- System configure
- Peripherals configure
- Motor Control configure
- Interrupt configure
- Main loop

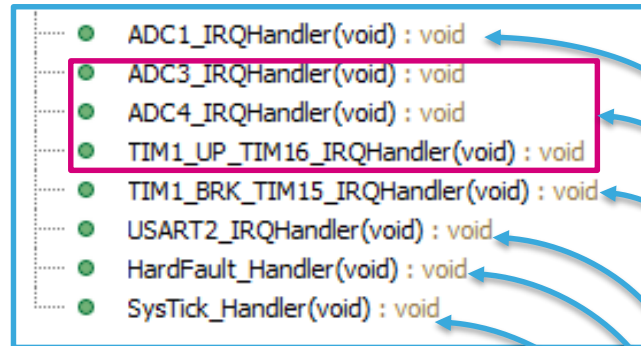
```

● MCboot(MCI_Handle_t*, MCT_Handle_t*) : void
● MC_Scheduler(void) : void
● TSK_MediumFrequencyTaskM1(void) : void
● FOC_Clear(uint8_t) : void
● FOC_InitAdditionalMethods(uint8_t) : void
● FOC_CalcCurrRef(uint8_t) : void
● TSK_SetChargeBootCapDelayM1(uint16_t) : void
● TSK_ChargeBootCapDelayHasElapsedM1(void)
● TSK_SetStopPermanencyTimeM1(uint16_t) : void
● TSK_StopPermanencyTimeHasElapsedM1(void)
● TSK_HighFrequencyTask(void) : uint8_t
● FOC_CurrController(uint8_t) : uint16_t
● MC_RequestRegularConv(uint8_t, uint8_t) : void
● MC_GetRegularConv(void) : uint16_t
● MC_RegularConvState(void) : UPRC_State_t
● TSK_SafetyTask(void) : void
● TSK_SafetyTask_PWMOFF(uint8_t) : void
● GetMCI(uint8_t) : MCI_Handle_t*
● GetMCT(uint8_t) : MCT_Handle_t*
● TSK_HardwareFaultTask(void) : void
● mc_lock_pins(void) : void
    
```

在文件 `mc_tasks.c` 中核心的是下面的几个环路:

- MC loop
- Torque or speed loop
- Current loop
- Safety loop





```
309 void SysTick_Handler(void)
310 {
311     /* USER CODE BEGIN SysTick_IRQn 0 */
312
313     /* USER CODE END SysTick_IRQn 0 */
314     HAL_IncTick();
315     HAL_SYSTICK_IRQHandler();
316     /* USER CODE BEGIN SysTick_IRQn 1 */
317     /* USER CODE END SysTick_IRQn 1 */
318     TB_Scheduler();
319     /* USER CODE BEGIN SysTick_IRQn 2 */
320     /* USER CODE END SysTick_IRQn 2 */
321 }
322
323 /* USER CODE BEGIN 1 */
324
325 /* USER CODE END 1 */
```

在文件 `stm32f30x_mc_it.c` 中有以下的中断服务程序：

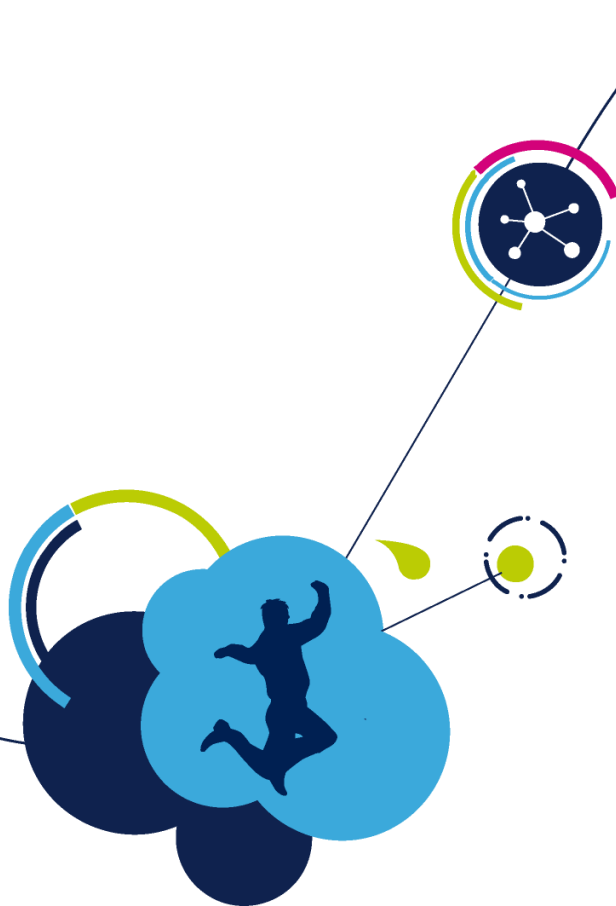
ADC 中断: HighFrequencyTask() 在这个函数中被调用。调用的频率可以在 `workbench` 中设定。最大的频率可以和 **PWM** 频率相同。

- 此例程中未启用。
- **TIM1 Break** 中断。
- **UART** 中断。
- **System hard fault** 中断。
- **System Tick** 中断：周期为500us。

在 `mc_api.c` 中有一系列的函数来实现与对电机的控制，我们称之为 **MC API**。这些函数是用户和 **SDK** 之间的桥梁。下面列出了一些最常用的函数：

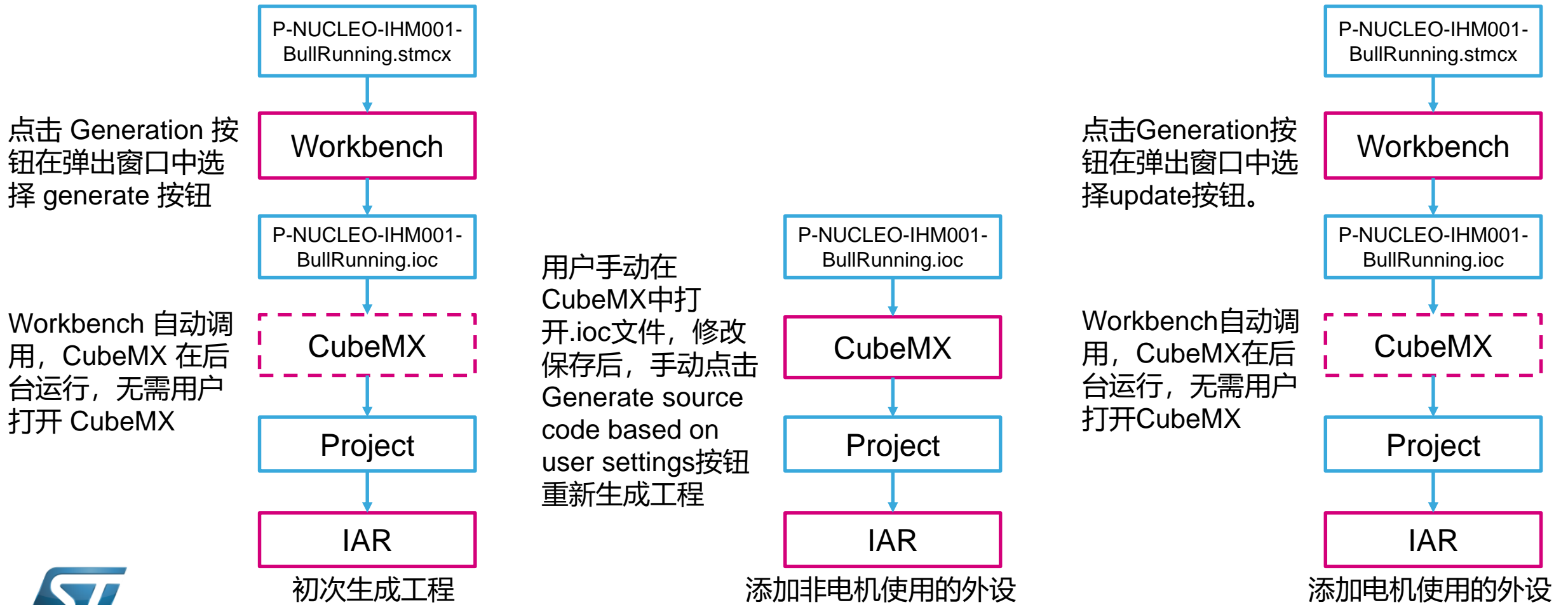
- `MC_StartMotor1`
- `MC_StopMotor1`
- `MC_ProgramSpeedRampMotor1`
- `MC_ProgramTorqueRampMotor1`
- `MC_GetMecSpeedReferenceMotor1`
- `MC_GetMecSpeedAverageMotor1`
- `MC_GetSTMStateMotor1`
- `MC_GetOccurredFaultsMotor1`
- `MC_AcknowledgeFaultMotor1`
- `MC_GetImposedDirectionMotor1`

开发实战



- ST MC SDK5.x WB应用指南
 - 软件工具的下载和安装
 - ST MC Workbench及关键配置参数
 - MC Project 的生成, 编译和下载
 - 电机控制及监控
- ST MC SDK5.x固件详解
 - 程序架构
 - 组件
 - 例程代码讲解
 - 开发实战
 - 如何向例程中添加外设和自己的代码
 - Step-by-Step添加一段闪灯代码

下图说明开发者在初次生成代码，修改代码配置一个非电机使用的外设，修改一个电机使用外设三种应用场景下的流程。




```
int main(void)
{
    /* USER CODE BEGIN 1 */

    /* USER CODE END 1 */

    /* MCU Configuration----- */

    /* Reset of all peripherals, Initializes the
    HAL_Init();

    /* USER CODE BEGIN Init */

    /* USER CODE END Init */

    /* Configure the system clock */
    SystemClock_Config();

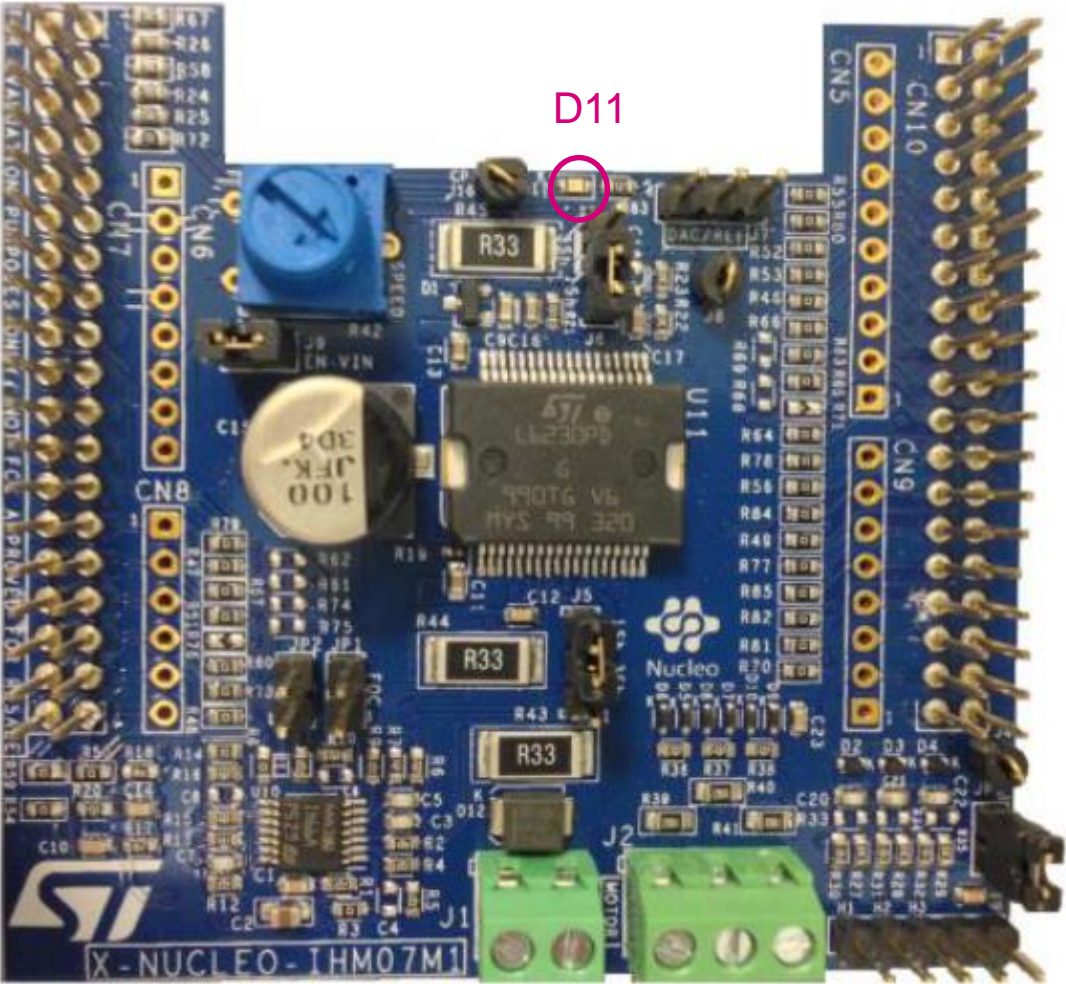
    /* USER CODE BEGIN SysInit */

    /* USER CODE END SysInit */

    /* Initialize all configured peripherals */
    MX_GPIO_Init();
    MX_ADC1_Init();
    MX_DAC_Init();
    MX_TIM1_Init();
    MX_USART2_UART_Init();
    MX_MotorControl_Init();
}
```

因为 **Workbench** 生成的工程中，绝大部分的代码是以源码的形式提供给大家的。所以大家可以很方便的修改和添加自己的应用代码。

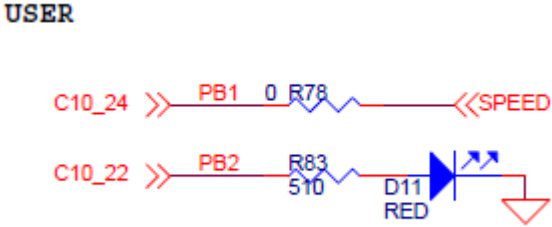
左图是例程代码 **P-NUCLEO-IHM001-BullRunning** 中的 **main** 函数。我们可以看到很多注释如 **/* USER CODE BEGIN 1 */** 和 **/* USER CODE END 1 */**。开发者应把自己的应用代码添加在 **USER CODE BEGIN** 和 **USER CODE END** 之间。这样当我们需要修改外设配置重新生成代码时，我们之前添加的代码不会被修改掉。否则代码将被覆盖为 SDK 的默认代码。



SDK5.x 中自带代码例程 **P-NUCLEO-IHM001-BullRunning** 。
与之配套的硬件是控制板 **NUCLEO F302R8** 和功率板 **X-NUCLEO-IHM07M1**。

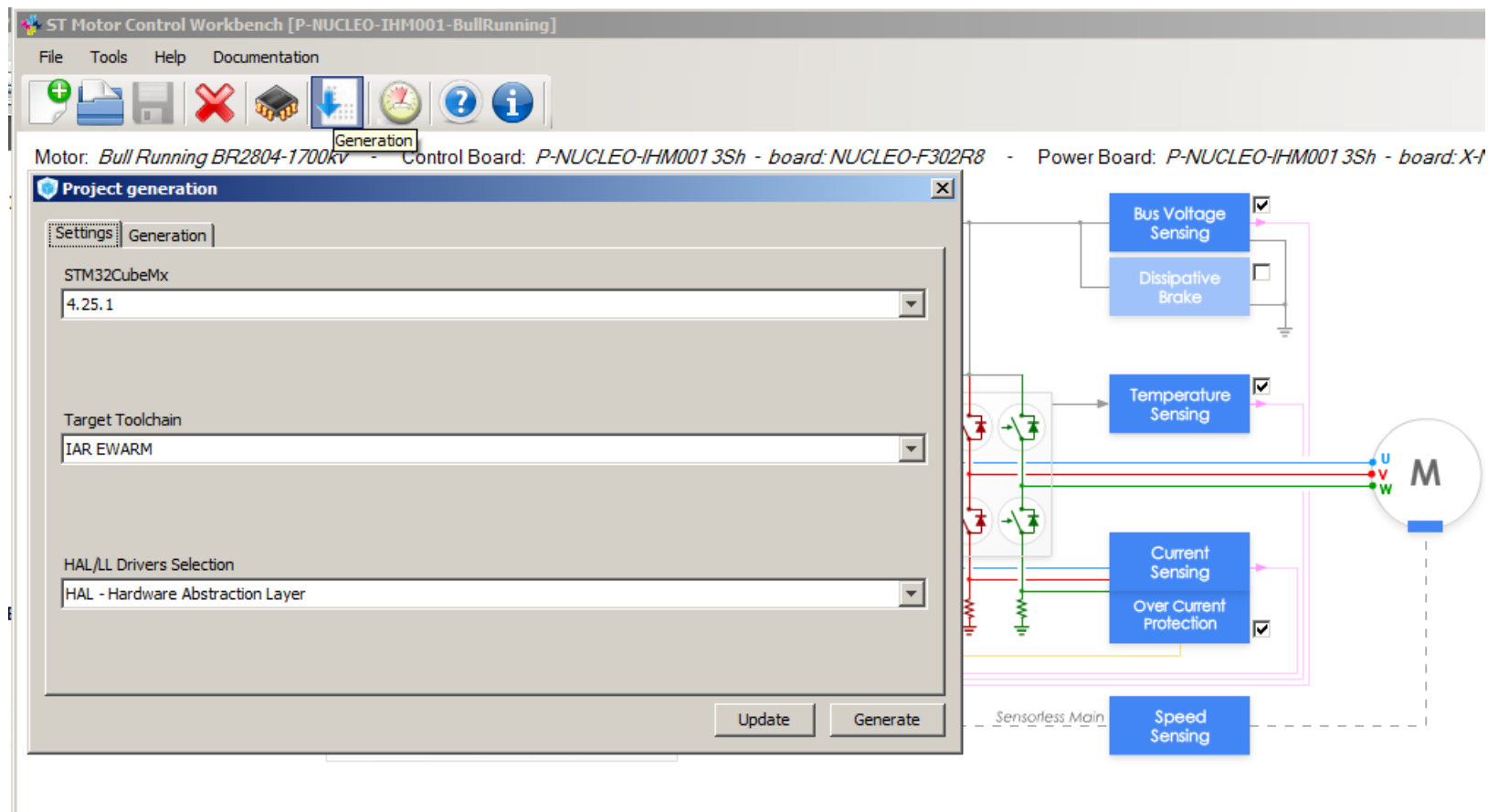
下面我们将一步一步修改代码,实现 LED 灯 **D11** 在电机进入正常运行状态后以1.0s的频率闪烁。

D11 连到了 **STM32F302R8** 的 PB2 引脚。



Step 1: Motor Control Workbench 初次生成代码

- 用 **Workbench** 打开 **SDK** 自带的 **P-NUCLEO-IHM001-BullRunning.stmcx**。
- 点击下图的 **Generation** 按钮，会跳出 **Project generation** 窗口。
- 在窗口中点击 **Generate** 按钮。



Step 1: Motor Control Workbench 初次生成代码

- 打开 IAR 工程，我们可以看到如下的 GPIO 初始化代码。

```
static void MX_GPIO_Init(void)
{
    GPIO_InitTypeDef GPIO_InitStructure;

    /* GPIO Ports Clock Enable */
    __HAL_RCC_GPIOC_CLK_ENABLE();
    __HAL_RCC_GPIOF_CLK_ENABLE();
    __HAL_RCC_GPIOA_CLK_ENABLE();

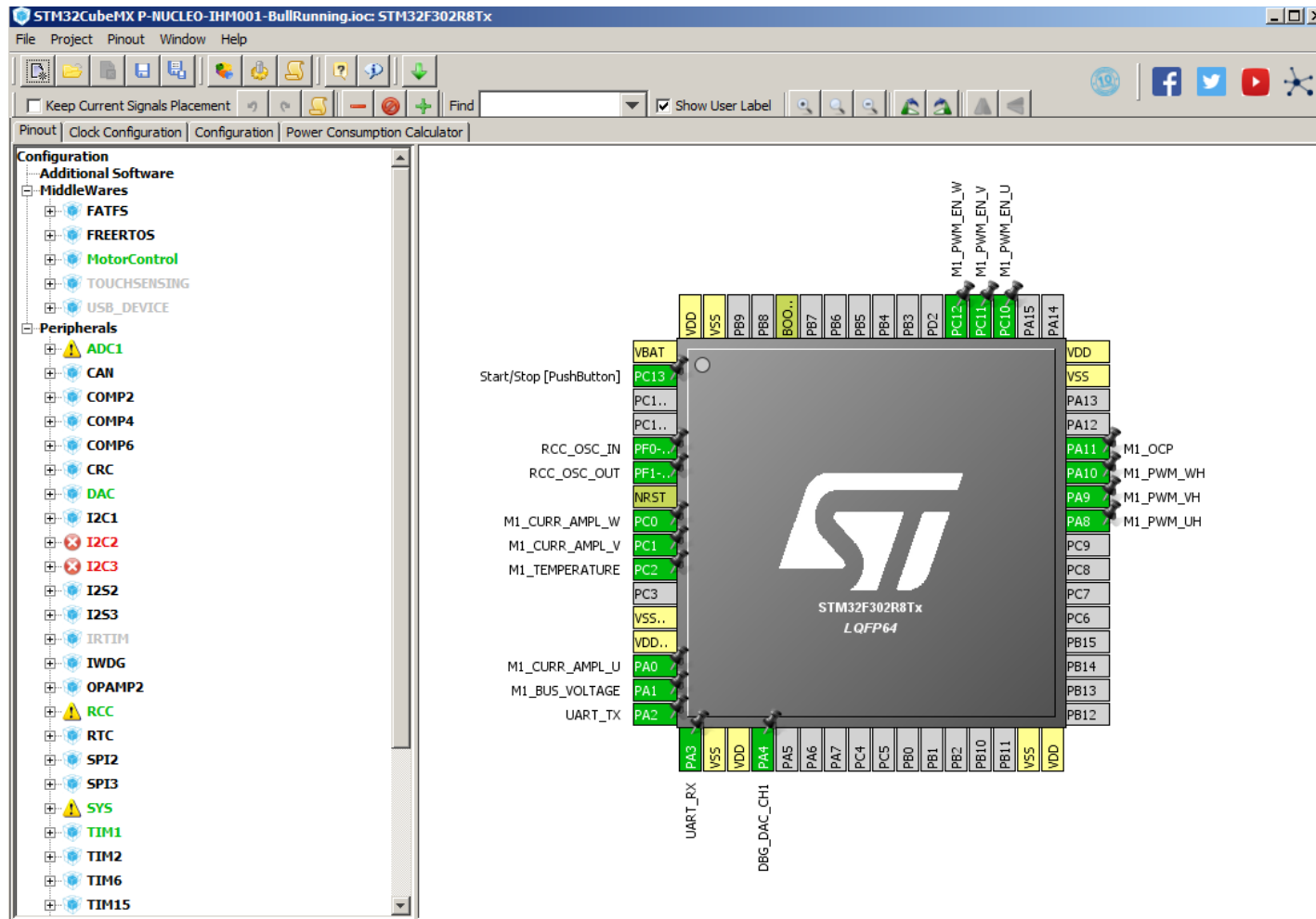
    /*Configure GPIO pin Output Level */
    HAL_GPIO_WritePin(GPIOC, M1_PWM_EN_U_Pin|M1_PWM_EN_V_Pin|M1_PWM_EN_W_Pin, GPIO_PIN_RESET);

    /*Configure GPIO pin : Start_Stop_Pin */
    GPIO_InitStructure.Pin = Start_Stop_Pin;
    GPIO_InitStructure.Mode = GPIO_MODE_IT_FALLING;
    GPIO_InitStructure.Pull = GPIO_NOPULL;
    HAL_GPIO_Init(Start_Stop_GPIO_Port, &GPIO_InitStructure);

    /*Configure GPIO pins : M1_PWM_EN_U_Pin M1_PWM_EN_V_Pin M1_PWM_EN_W_Pin */
    GPIO_InitStructure.Pin = M1_PWM_EN_U_Pin|M1_PWM_EN_V_Pin|M1_PWM_EN_W_Pin;
    GPIO_InitStructure.Mode = GPIO_MODE_OUTPUT_PP;
    GPIO_InitStructure.Pull = GPIO_PULLDOWN;
    GPIO_InitStructure.Speed = GPIO_SPEED_FREQ_HIGH;
    HAL_GPIO_Init(GPIOC, &GPIO_InitStructure);
}
```

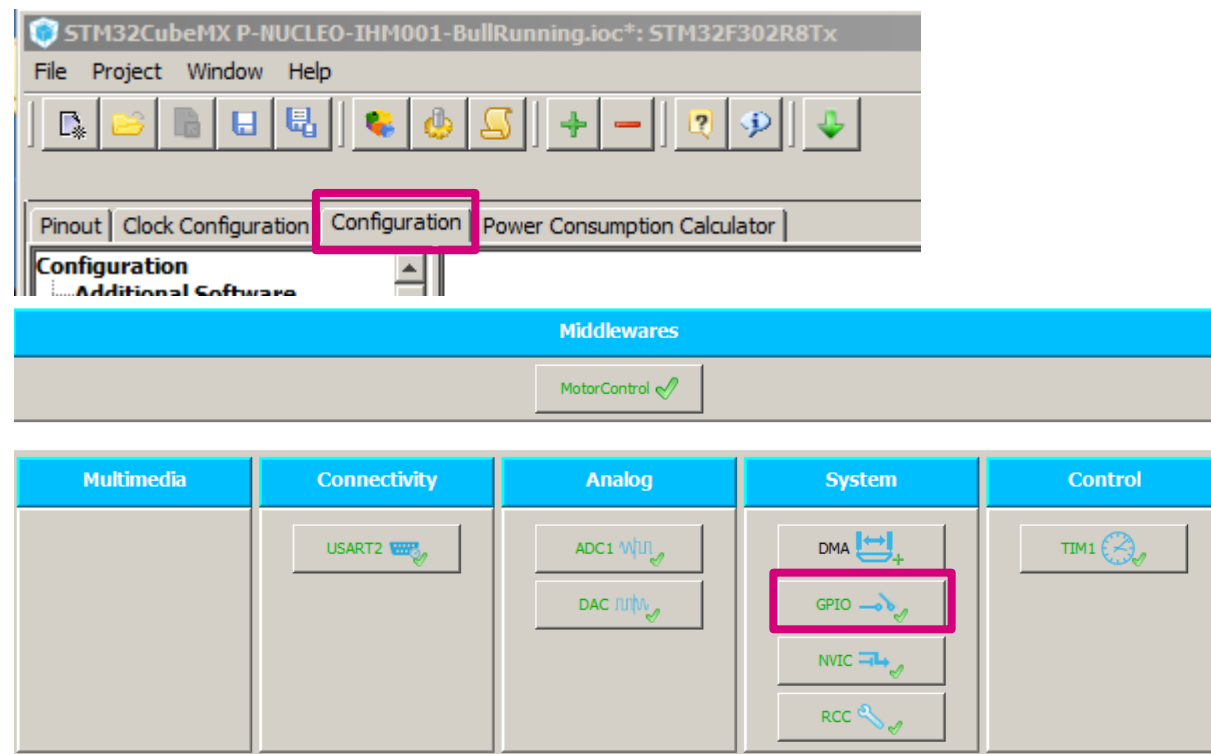
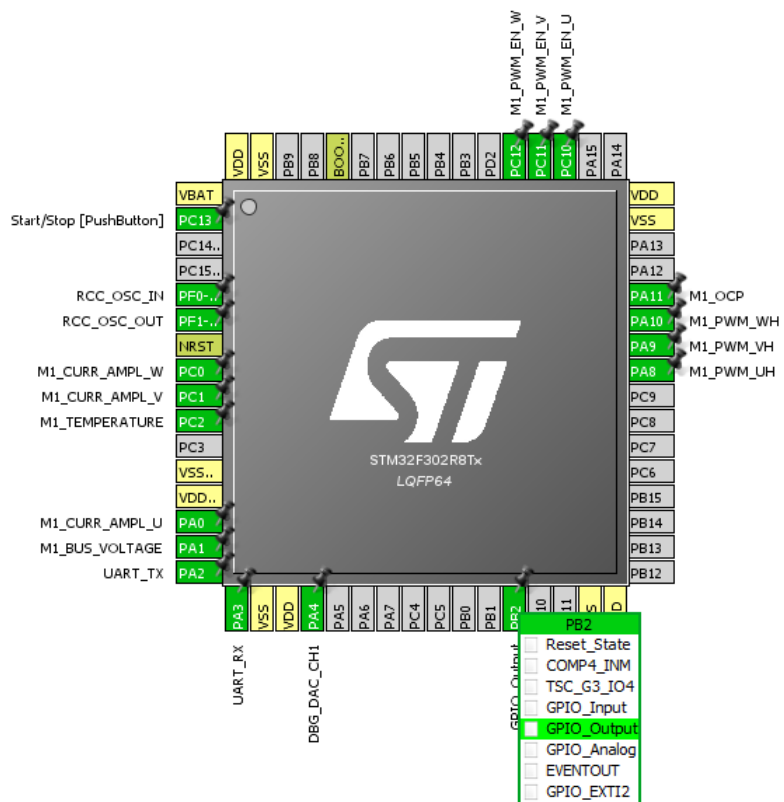
Step 2: 在 CubeMX 中添加一个 GPIO

- 打开 **P-NUCLEO-IHM001-BullRunning.ioc** , 在 **CubeMX** 里配置 **GPIO** 的 **PB2** 引脚。
- 从下图我们可以看到 **PB2** 尚未被占用。



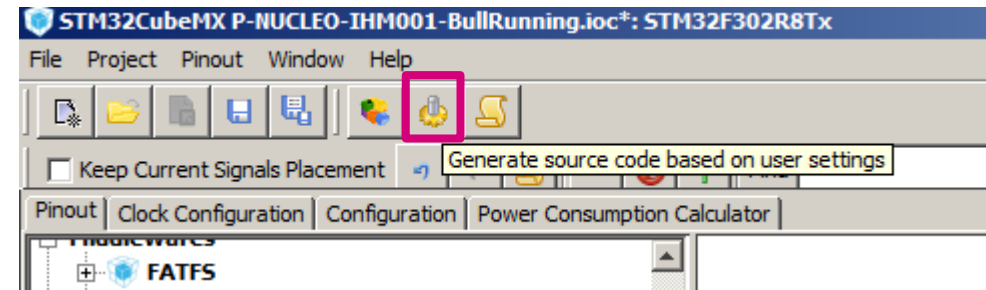
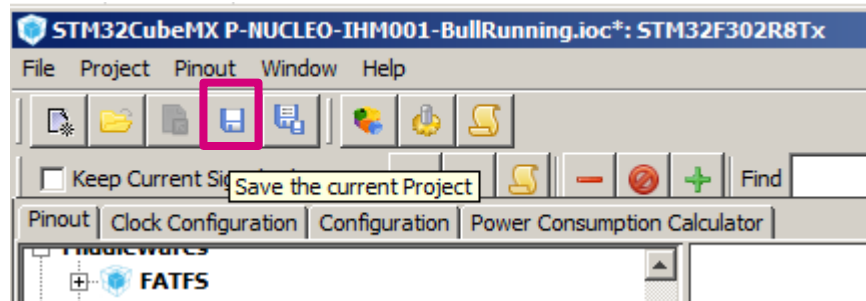
Step 2: 在 CubeMX 中添加一个 GPIO

- 点击左图中的 **PB2** 引脚，选择 **GPIO_Output**。
- 如右图，在 **CubeMX** 的窗口中选择 **Configuration** 选项卡。在窗口中我们可以看到 **System** 区域内有 **GPIO** 的按钮。点击设置 **PB2** 的初始配置。我们把 PB2 初始化为输出高电平，来点亮 LED。



Step 3: 保存 P-NUCLEO-IHM001-BullRunning.ioc, 重新生成代码

- 点击 **Save the current Project** 按钮。保存对 **PB2** 的设置到 **P-NUCLEO-IHM001-BullRunning.ioc**。
- 点击 **Generate source code based on user settings** 。



Step 3: 保存 P-NUCLEO-IHM001-BullRunning.ioc, 重新生成代码

- 用 IAR 打开新生成的代码, 我们可以看到在 `main.h` 中增加了 `PB2` 的宏定义。
- 在 `main.c` 的函数 `static void MX_GPIO_Init(void)` 中也新增了初始化代码。
- 运行新生成的代码, 我们可以看到功率板 `X-NUCLEO-IHM07M1` 上的 `LED` 亮起来。

```
main.h x
81 #define LED_D11_Pin GPIO_PIN_2
82 #define LED_D11_GPIO_Port GPIOB

main.c x
MX_GPIO_Init()
457 /*
458 static void MX_GPIO_Init(void)
459 {
460
461     GPIO_InitTypeDef GPIO_InitStruct;
462
463     /* GPIO Ports Clock Enable */
464     __HAL_RCC_GPIOC_CLK_ENABLE();
465     __HAL_RCC_GPIOF_CLK_ENABLE();
466     __HAL_RCC_GPIOA_CLK_ENABLE();
467     __HAL_RCC_GPIOB_CLK_ENABLE();
468
469     /*Configure GPIO pin Output Level */
470     HAL_GPIO_WritePin(LED_D11_GPIO_Port, LED_D11_Pin, GPIO_PIN_SET);
471
472     /*Configure GPIO pin Output Level */
473     HAL_GPIO_WritePin(GPIOC, M1_PWM_EN_U_Pin|M1_PWM_EN_V_Pin|M1_PWM_EN_W_Pin, GPIO_PIN_RESET);
474
475     /*Configure GPIO pin : Start_Stop_Pin */
476     GPIO_InitStruct.Pin = Start_Stop_Pin;
477     GPIO_InitStruct.Mode = GPIO_MODE_IT_FALLING;
478     GPIO_InitStruct.Pull = GPIO_NOPULL;
479     HAL_GPIO_Init(Start_Stop_GPIO_Port, &GPIO_InitStruct);
480
481     /*Configure GPIO pin : LED_D11_Pin */
482     GPIO_InitStruct.Pin = LED_D11_Pin;
483     GPIO_InitStruct.Mode = GPIO_MODE_OUTPUT_PP;
484     GPIO_InitStruct.Pull = GPIO_NOPULL;
485     GPIO_InitStruct.Speed = GPIO_SPEED_FREQ_LOW;
486     HAL_GPIO_Init(LED_D11_GPIO_Port, &GPIO_InitStruct);
```

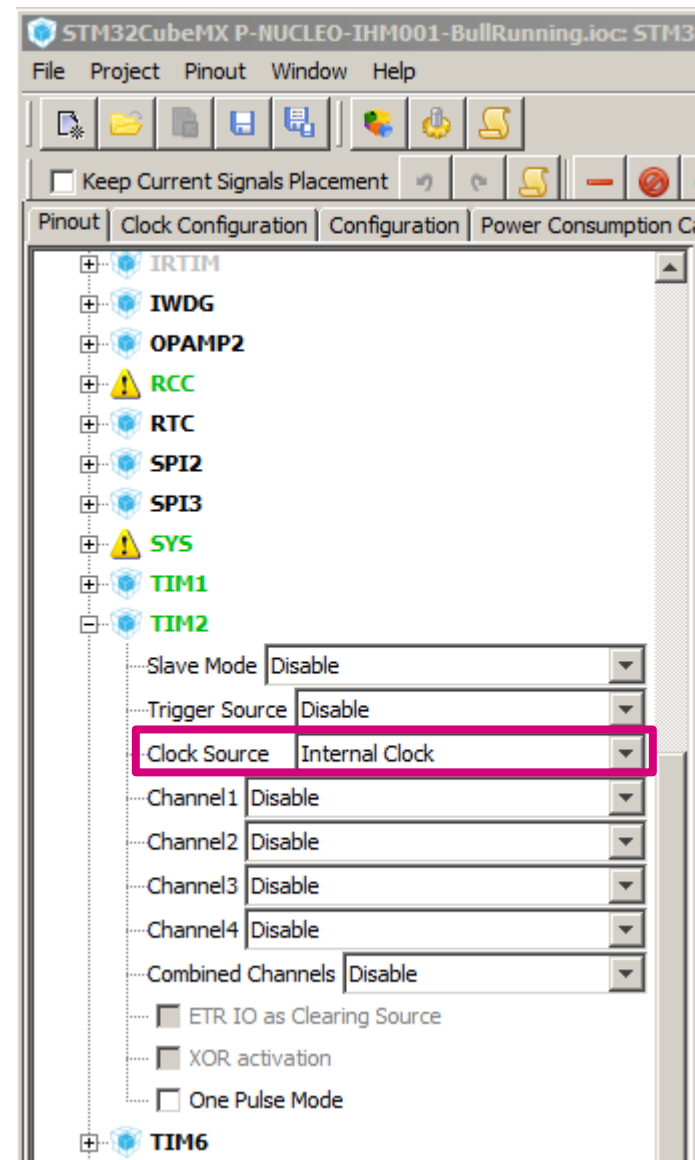
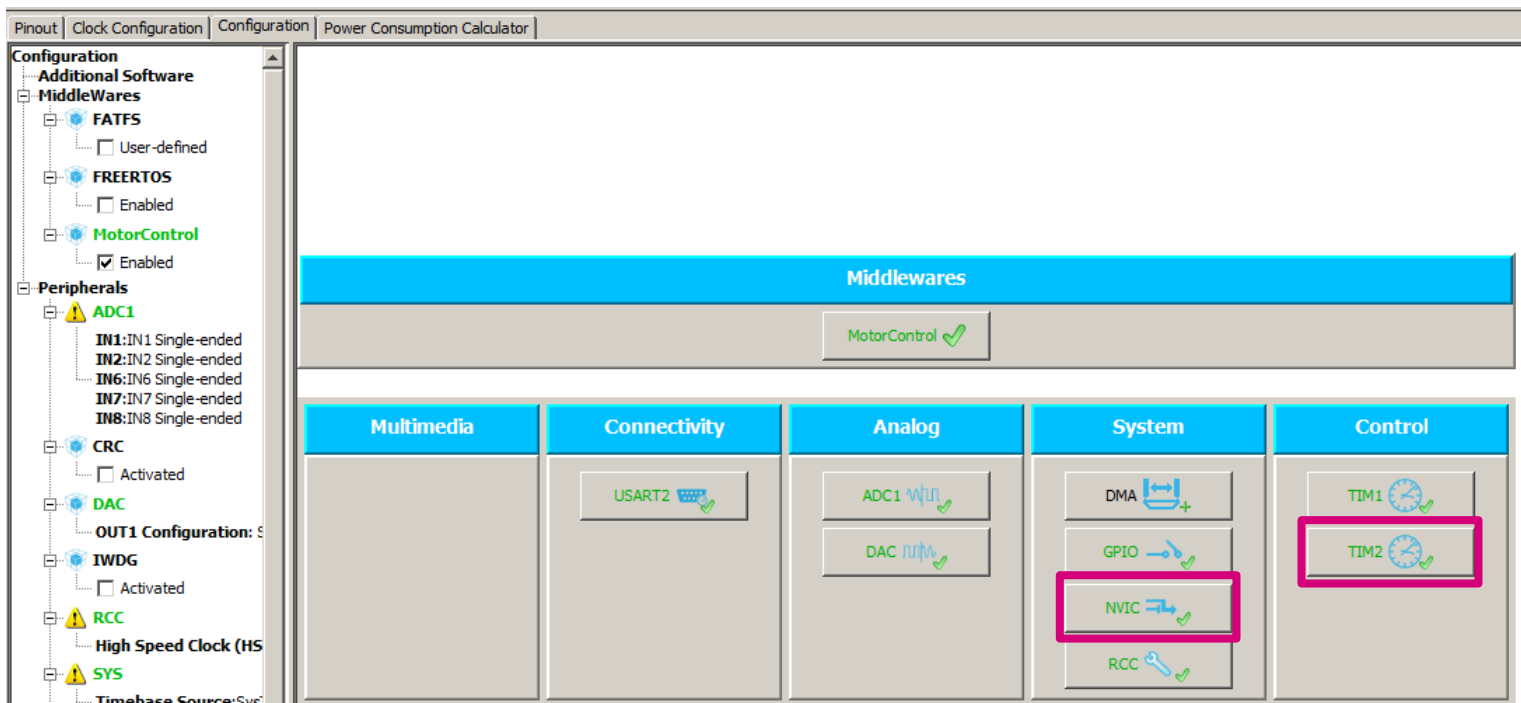

Step 4: 添加代码使 LED 根据电机的运行状态亮灭

- 在 `mc_api.c` 中查找用户 API 后，我们发现可以调用函数 `State_t MC_GetSTMStateMotor1(void)` 得到电机状态。
- 通过查询 `State_t` 的定义，我们可以在 `state_machine.h` 找到电机的各个状态。各状态的含义见程序注释。当电机正常运行的时候，对应的状态因为 `RUN = 6`。
- 在程序中添加如图代码，实现在电机进入RUN后点亮LED。注意所添加的代码都在注释 `USER CODE BEGIN` 和 `USER CODE END` 之间。除了红框中的一行代码。
- 通过以上的修改我们实现了让 LED 根据电机状态点亮和熄灭。

```
main.c x
135  /* Infinite loop */
136  HAL_GPIO_WritePin(LED_D11_GPIO_Port, LED_D11_Pin, GPIO_PIN_RESET);
137  /* USER CODE BEGIN WHILE */
138  while (1)
139  {
140
141  /* USER CODE END WHILE */
142
143  /* USER CODE BEGIN 3 */
144  if (MC_GetSTMStateMotor1() == RUN)
145  {
146  if (LEDIndicationFlag == 0)
147  {
148  LEDIndicationFlag = 1;
149  HAL_GPIO_WritePin(LED_D11_GPIO_Port, LED_D11_Pin, GPIO_PIN_SET);
150  }
151  }
152  else
153  {
154  if (LEDIndicationFlag == 1)
155  {
156  LEDIndicationFlag = 0;
157  HAL_GPIO_WritePin(LED_D11_GPIO_Port, LED_D11_Pin, GPIO_PIN_RESET);
158  }
159  }
160  }
161  /* USER CODE END 3 */
162
163 }
```

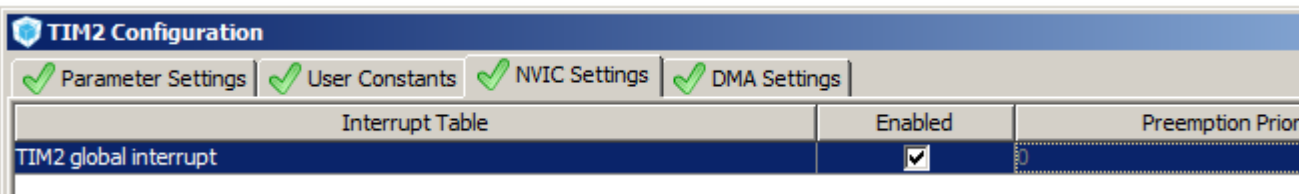
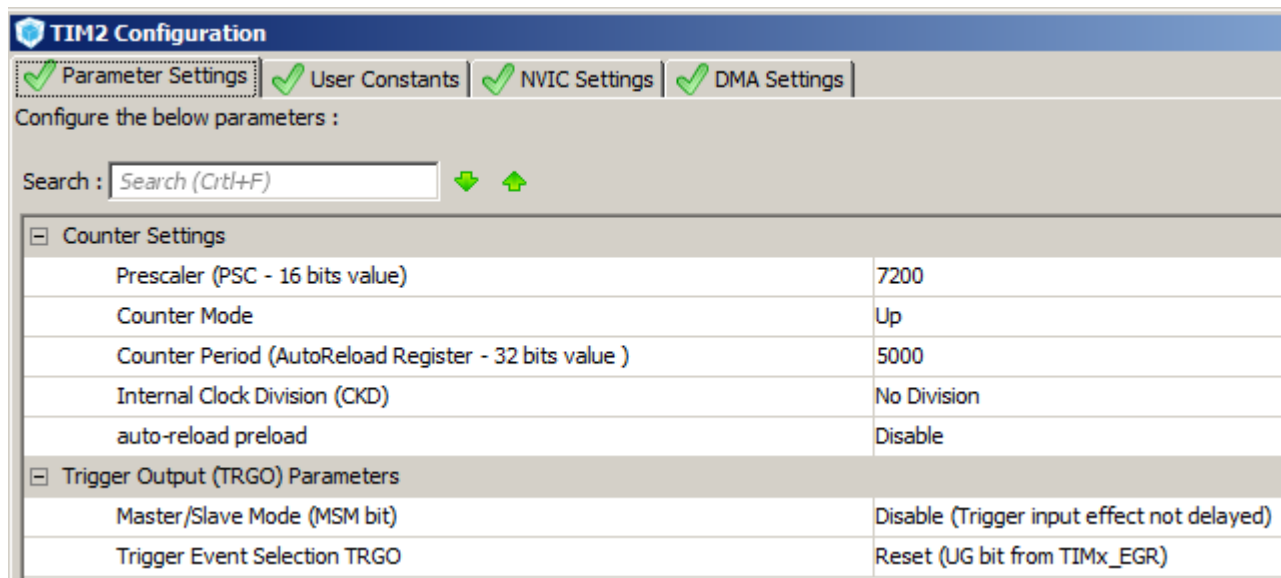
Step 5: 添加一个定时器 TIM2 来实现500ms的定时中断

- 打开 **P-NUCLEO-IHM001-BullRunning.ioc** 在右图所示的 Pinout 窗口中点击 TIM2。为选择 TIM2 选择时钟源。
- 在如下图在窗口 **Configuration** 中有对各外设的具体配置。我们需要点击下图中的 **TIM2** 和 **NVIC** 这两个按钮，分别配置 **TIM2** 及其中断。



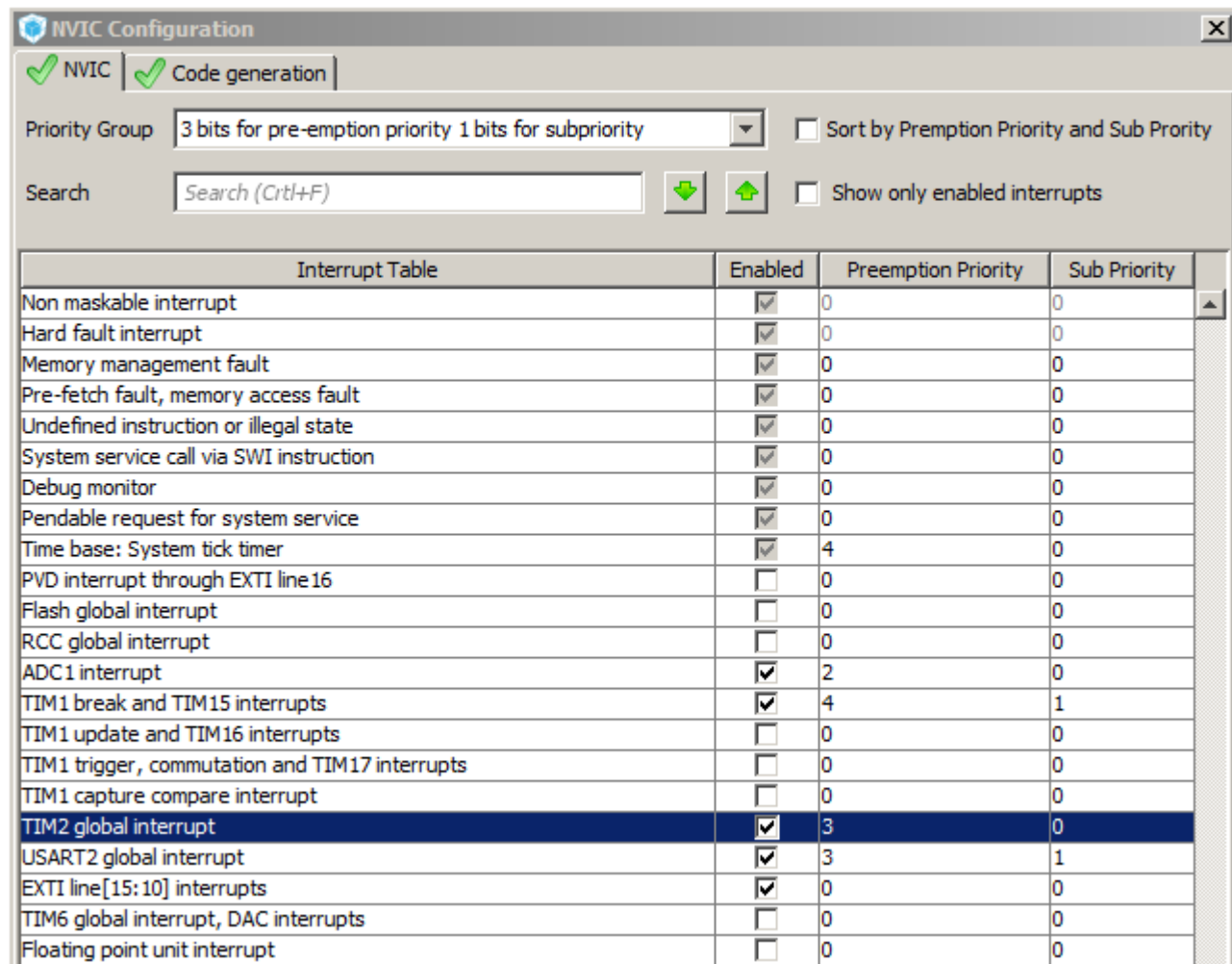
Step 5: 添加一个定时器 TIM2 来实现500ms的定时中断

- 点击 **TIM2** 后弹出下图所示的 **TIM2 Configuration** 窗口。
- 在窗口内的 **Parameter** 页面设置时钟的周期为500ms。
- 在窗口内的 **NVIC Settings** 页面使能 **TIM2 global interrupt**。



Step 5: 添加一个定时器 TIM2 来实现500ms的定时中断

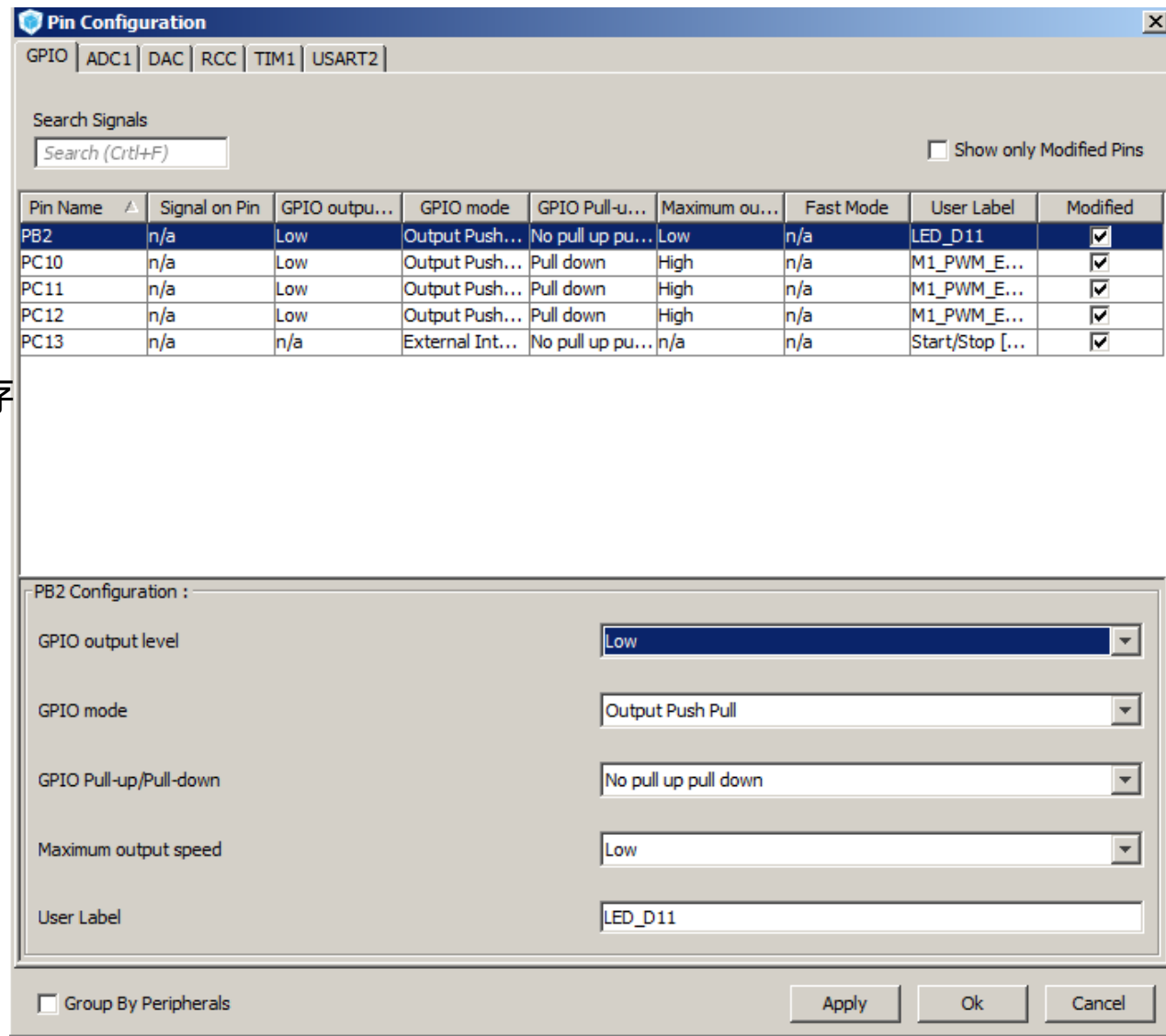
- 关闭 **TIM2 Configuration** 窗口，回到 **Configuration** 窗口，点击 **NVIC** 按钮。
- 在 **NVIC Configuration** 窗口选择 **TIM2 global interrupt** 的中断优先级。



Step 6: 修改PB2的初始输出状态

- 在 **Configuration** 窗口点击 **GPIO** 按钮。出现如下 **Pin Configuration** 窗口。
- 在 Pin Configuration 窗口中设置 **GPIO output level** 为 **low**。

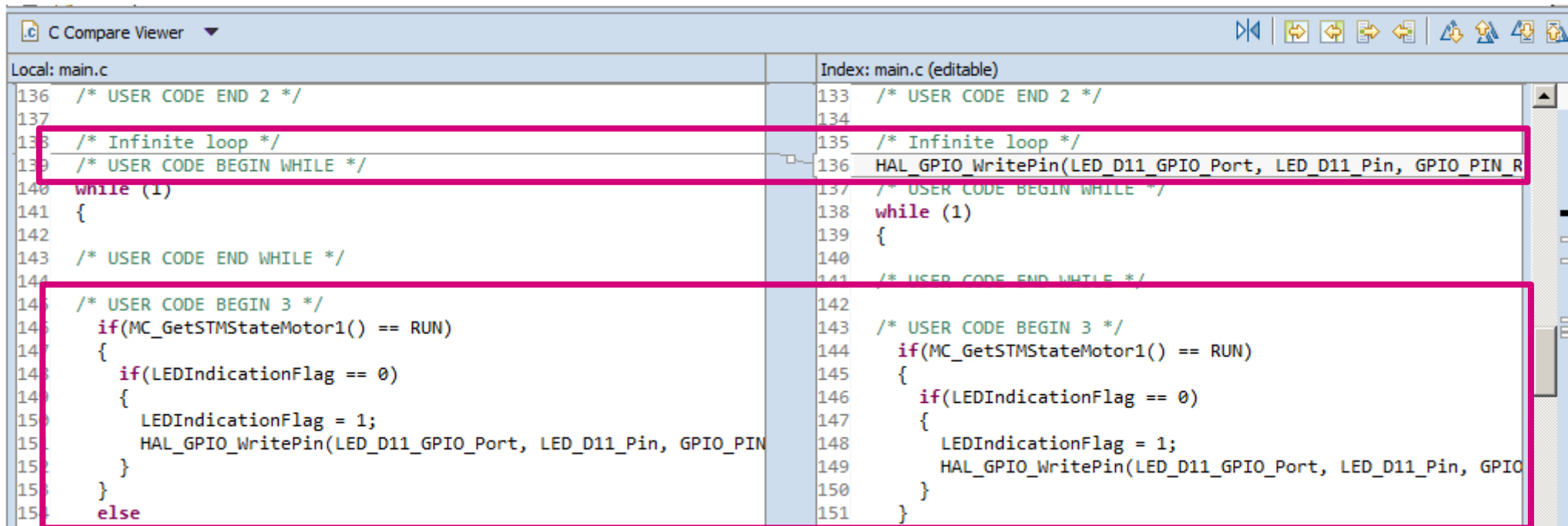
到这一步我们已经完成了所有的CubeMX的设置。保存并重新生成代码。



Step 7: 修改代码

对比新生成的代码我们看到下面的文件发生了变 [main.c](#) , [stm32f3xx_hal_msp.c](#) , [stm32f3xx_it.c](#) , [stm32f3xx_it.h](#)

这里着重提示一点：从下图中可以看到之前写在 **USER CODE BEGIN** 和 **USER CODE END** 之间的代码仍然存在。**写在USER CODE BEGIN 和 USER CODE END之外的代码被覆盖。**



```
C Compare Viewer
Local: main.c
Index: main.c (editable)

136 /* USER CODE END 2 */
137
138 /* Infinite loop */
139 /* USER CODE BEGIN WHILE */
140 while (1)
141 {
142
143 /* USER CODE END WHILE */
144
145 /* USER CODE BEGIN 3 */
146 if(MC_GetSTMStateMotor1() == RUN)
147 {
148     if(LEDIndicationFlag == 0)
149     {
150         LEDIndicationFlag = 1;
151         HAL_GPIO_WritePin(LED_D11_GPIO_Port, LED_D11_Pin, GPIO_PIN_R
152     }
153 }
154 else
155

133 /* USER CODE END 2 */
134
135 /* Infinite loop */
136 HAL_GPIO_WritePin(LED_D11_GPIO_Port, LED_D11_Pin, GPIO_PIN_R
137 /* USER CODE BEGIN WHILE */
138 while (1)
139 {
140
141 /* USER CODE END WHILE */
142
143 /* USER CODE BEGIN 3 */
144 if(MC_GetSTMStateMotor1() == RUN)
145 {
146     if(LEDIndicationFlag == 0)
147     {
148         LEDIndicationFlag = 1;
149         HAL_GPIO_WritePin(LED_D11_GPIO_Port, LED_D11_Pin, GPIO
150     }
151 }
```

Step 7: 修改代码

添加 **TIM2** 启动的代码。CubeMX 只是配置了 TIM2，要启动 TIM2 还需要添加如下代码。

```
main.c x
main()
130     MX_TIM2_Init();
131
132     /* Initialize interrupts */
133     MX_NVIC_Init();
134     /* USER CODE BEGIN 2 */
135     HAL_TIM_Base_Start_IT(&htim2);
136     /* USER CODE END 2 */
137
```

我们在 `main.c` 中的 `main` 函数配置完 TIM2 和中断之后调用了函数 `HAL_StatusTypeDef HAL_TIM_Base_Start_IT(TIM_HandleTypeDef *htim)`。这个函数启动 TIM2 的计数器并使能 `update` 中断。函数的具体定义如下

```
HAL_StatusTypeDef HAL_TIM_Base_Start_IT(TIM_HandleTypeDef *htim)
{
    /* Check the parameters */
    assert_param(IS_TIM_INSTANCE(htim->Instance));

    /* Enable the TIM Update interrupt */
    __HAL_TIM_ENABLE_IT(htim, TIM_IT_UPDATE);

    /* Enable the Peripheral */
    __HAL_TIM_ENABLE(htim);

    /* Return function status */
    return HAL_OK;
}
```

Step 7: 修改代码

在右图 **TIM2** 的中断服务程序中将变量 **Flag500ms** 置为1。以此来标识 **500ms** 的时间周期。这个中断服务程序可以由 **CubeMX** 生成，我们只需要在函数中 **USER CODE BEGIN** 和 **USER CODE END** 之间添加自己的应用代码。

另外我们对在 **main** 函数中的用户代码做了如右图的修改。

```
/**
 * @brief This function handles TIM2 global interrupt.
 */
void TIM2_IRQHandler(void)
{
    /* USER CODE BEGIN TIM2_IRQn 0 */
    Flag500ms = 1;
    /* USER CODE END TIM2_IRQn 0 */
    HAL_TIM_IRQHandler(&htim2);
    /* USER CODE BEGIN TIM2_IRQn 1 */

    /* USER CODE END TIM2_IRQn 1 */
}
```

```
/* USER CODE BEGIN 3 */
if(MC_GetSTMStateMotor1() == RUN)
{
    if(Flag500ms == 1)
    {
        Flag500ms = 0;
        if(LEDIndicationFlag == 0)
        {
            LEDIndicationFlag = 1;
            HAL_GPIO_WritePin(LED_D11_GPIO_Port, LED_D11_Pin, GPIO_PIN_SET);
        }
        else
        {
            LEDIndicationFlag = 0;
            HAL_GPIO_WritePin(LED_D11_GPIO_Port, LED_D11_Pin, GPIO_PIN_RESET);
        }
    }
}
else
```


小结

- **Motor Control Workbench** 初次生成代码和用于 **CubeMX** 的 **.ioc**文件。
- 添加非电机驱动的外设配置可以打开由 **Workbench** 生成的 **.ioc** 文件, 然后在 **CubeMX** 中配置并重新生成代码。
- 用户代码要写在 **USER CODE BEGIN** 和 **USER CODE END** 之间。

Releasing your creativity



- Thank you -

重要通知 - 请仔细阅读

意法半导体公司及其子公司（“ST”）保留随时对ST 产品和/ 或本文档进行变更、更正、增强、修改和改进的权利，恕不另行通知。买方在订货之前应获取关于ST 产品的最新信息。ST 产品的销售依照订单确认时的相关ST 销售条款。

买方自行负责对ST 产品的选择和使用， ST 概不承担与应用协助或买方产品设计相关的任何责任。

ST 不对任何知识产权进行任何明示或默示的授权或许可。

转售的ST 产品如有不同于此处提供的信息的规定，将导致ST 针对该产品授予的任何保证失效。

ST 和ST 徽标是ST 的商标。所有其他产品或服务名称均为其各自所有者的财产。

本文档中的信息取代本文档所有早期版本中提供的信息。

版权声明

本文档为意法半导体公司及其子公司（“ST”）版权所有，未经ST允许不得复制、修改、转发或应用于商业目的。