文档编号：AN5166

文档名称：使用STM32 MC SDK V5.0控制和定制电源板指南

用ST设计自己的PMSM电机控制板，

可参考文档：

•AN2834：如何在STM32微控制器中获得最佳ADC精度

•UM2392：STM32电机控制SDK v5.0.0固件

•UM2380：STM32电机控制SDK v5. 0工具

**1基本信息**

|  |  |
| --- | --- |
| Acronym | Description |
| HAL | Hardware abstraction layer硬件抽象层 |
| HFI | High frequency injection高频注入法 |
| ICS | Isolated current sensor隔离电流传感器 |
| LL | Low-level |
| PFC | Maximum torque per Ampere每安培最大扭矩 |
| PFC | Power factor correction功率因数校正 |
| SPL | Standard peripheral libraries 标准外围库 |

**2参考设计**

从ST MC开始设计新硬件板的最简单方法是执行以下步骤：

1.选择相同的三相逆变器拓扑

a）相同的STM32 MCU线路（例如STM32F303xx）

b）相同的电流检测拓扑（例如1分流拓扑）

c）类似的电机驱动器（例如STSPIN230作为三个半桥的驱动器）

d）相似功率（例如X-NUCLEO-IHM11M1为低电压/低电流）

2.适应新硬件

a）电阻分流器计算

b）电阻器网络增益计算

c）外部或内部运算放大器组件的使用

d）外部或内部比较器的使用

e）输入比较器的保护阈值源的计算

3.修改STM32 Workbench项目（PC软件工具）

a）查找选定的ST示例

b）使用所有硬件适配进行更新

4.从STM32CubeMX项目（PC软件工具）进行修改

c）需要时，使用兼容的线路更改MCU线路

d）完成应用程序配置

**2.1硬件设置**

三相电机的控制依赖于：

•强制性功能：

- 用于执行FW的微控制器

- 用于控制电机电压和电流的三相逆变器

- 用于反馈调节的电流感应采集

•可选功能（本文档中仅讨论了其中一些功能）：

- 用于降低电机启动期间峰值电流的浪涌电流限制器(In-rush current limiter)

- 降低能耗的功率因数校正（PFC）机制

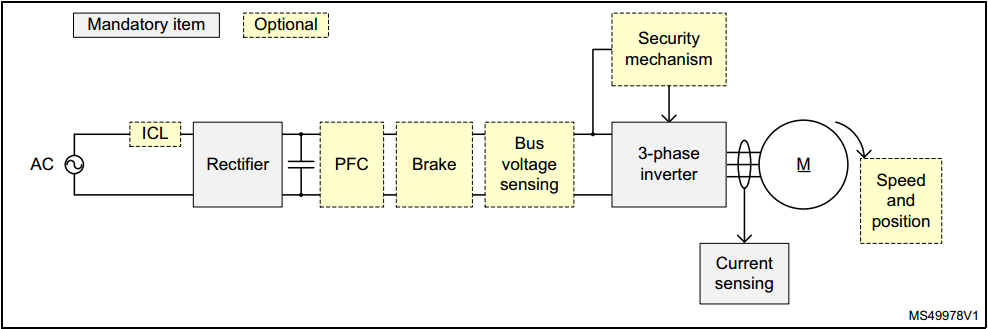
- 释放电机的制动机制能量 (brake mechanism)刹车

- 用于监控输入电压的总线电压检测 (bus voltage sensing)

- 保护人员和硬件的安全机制 (security mechanism )

- 从速度和位置传感器获取数据，用于调节反馈(speed and position sensors)

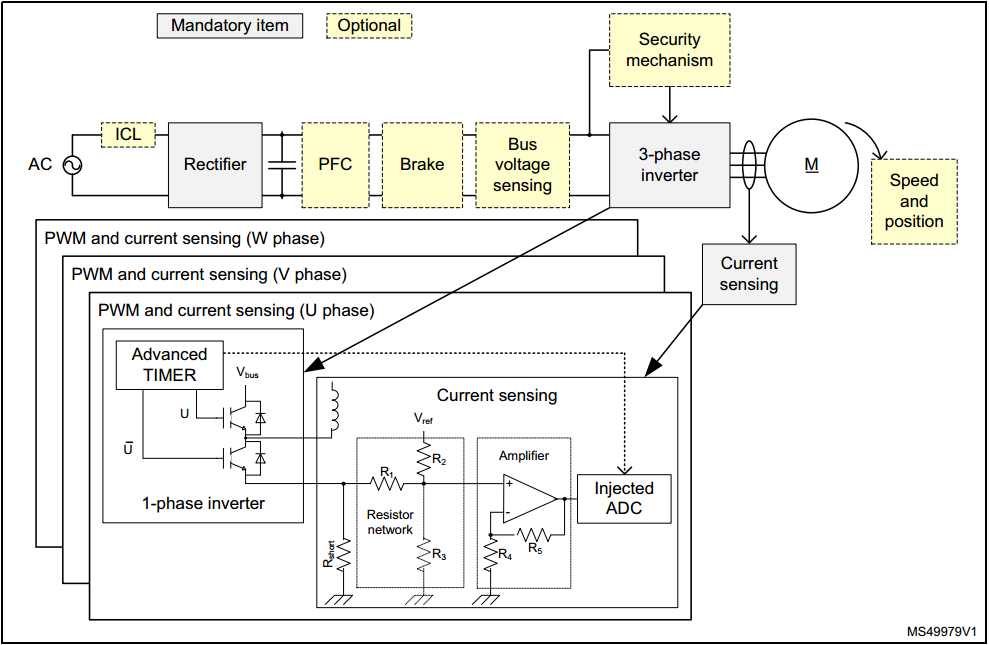
图1.典型的MC硬件设置



**2.1.1三相逆变器(3-phase inverter)**

电机控制子系统使用一个带PWM数字输出的高级定时器来驱动三相逆变器

图2.典型的MC PWM和电流检测



在ST板上，功率级实现了高侧和低侧开关，以驱动三相。根据功率元件和每个电机相位，支持的PWM数字输出（或通道）为：

•高端和低端驱动

- 硬件激活所需的PWM数字输出及其互补信号。

- 用户必须使用STM32 MC Workbench PC软件工具配置PWM开关死区时间，以避免逆变器发生短路。

•仅限高端驱动器

- 硬件启用驱动器并激活所需的PWM数字输出。

- 用户必须验证所使用的硬件组件是否管理PWM切换死区时间，以避免逆变器中的短路。

从固件的角度来看，使用STM32 MC Workbench PC软件可以轻松执行驱动程序桥的配置。用户根据其HW实现，通过Power阶段区域内的专用Phase Driver块配置驱动器桥接死区时间值及其活动极性。

根据实现的电流检测拓扑结构，除了使用的PWM通道外，还需要其他定时器通道来触发ADC的采样时间（参见表2）。

表2.高级计时器使用情况

|  |  |  |  |
| --- | --- | --- | --- |
| Supported MCUs | Topology | | |
| 1-shunt | 3-shunt | Dual ICS |
| STM32F030RC/R8 STM32F031C6 STM32F051R8/C8 STM32F072VB/RB STM32F1xx STM32F302VB/VC STM32F303VB/VC STM32F303ZE/VE/RE STM32F302R8 STM32F415ZG STM32F417IG STM32F407IG STM32F446ZE/RE | -每相驱动PWM -3通道  -SVPWM效率触发 - 1通道  -ADC使用触发-2通道 | - 每相驱动的PWM - 3个通道  - ADC使用触发 - 1个通道 | – PWM for each phase drive – 3 channels  – Trigger for ADC usage – 1 channel |

**2.1.2电流检测**

出于准确性原因，当前测量值由高级定时器触发（由于其内部功能）。

电机控制子系统需要使用ADC。根据所使用的ADC和电流测量拓扑，支持三种实现。

•3分流电流读取模式需要：

- 电阻网络（R1，R2和R3），用于在通过运算放大器及其电阻网络进行放大之前极化测量电压（充当电机的电流消耗图像）（ R4和R5）。需要针对每个阶段实施此设置（总共三次）。

- 两个具有两个通道的ADC，用于同时测量电压，此时从两个使能相位（最高精度），或者一个带有三个通道的ADC在两个不同时间测量电压，从两个启用阶段（精度较低）。

•1分流电流读取模式需要：

- 电阻网络（R1，R2和R3），用于在通过运算放大器及其电阻网络进行放大之前极化测量电压（充当电机的电流消耗图像）（ R4和R5）。此设置仅需一次。

- 一个具有一个通道的ADC，用于在采样时间（每个周期两次）测量电压，此时一个（或两个）相位处于活动状态。

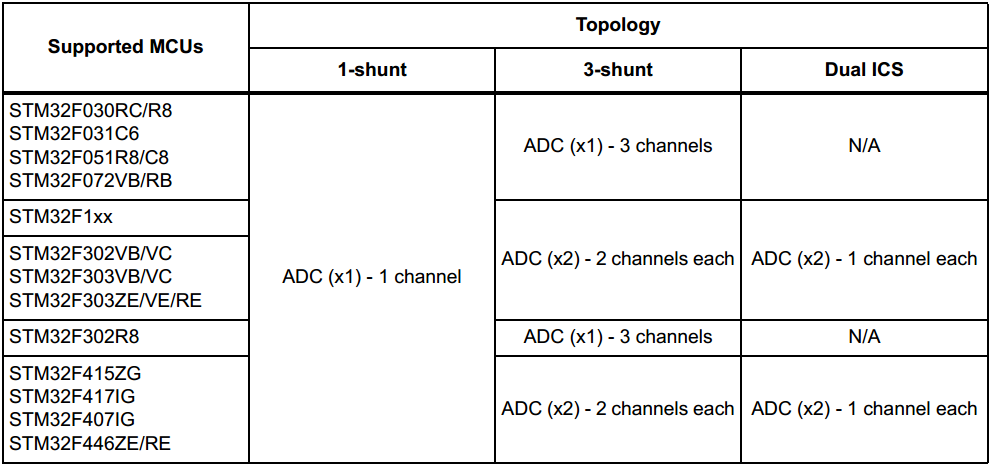
•双ICS读取模式需要：

- 需要时，电阻网络（R1，R2和R3）通过运算放大器及其电阻器在放大之前极化测量电压（充当相位的电流消耗图像）网络（R4和R5）。必须针对三个阶段中的至少两个实施该设置。

- 两个ADC，每个ADC有一个通道，用于测量两个硬连接相的电压（在相同的采样时间）

从固件的角度来看，使用STM32 MC Workbench PC软件可以轻松地从一种拓扑读取模式转换到另一种拓扑读取模式。用户必须根据其硬件实现，通过功率级区域内的电流感测块来调整电阻网络。

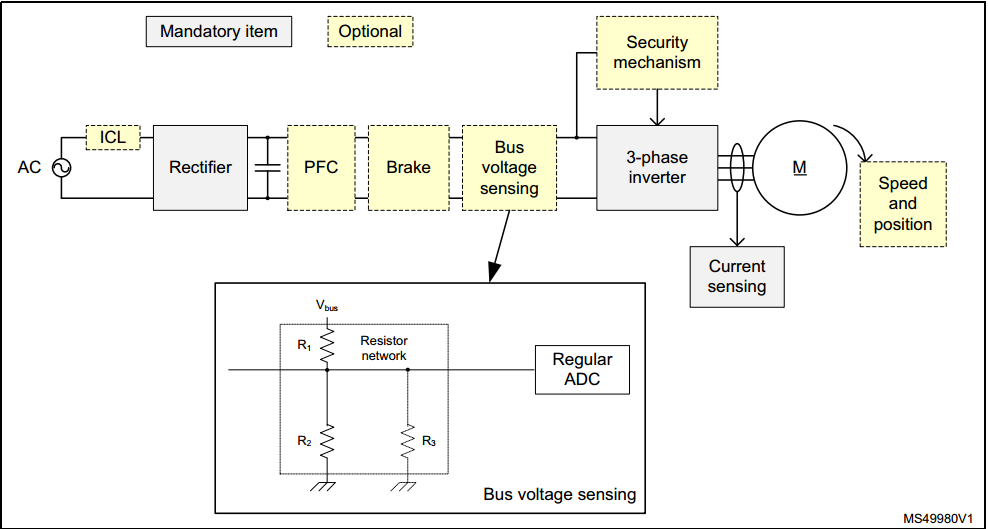
表3.电流检测



**2.1.3总线电压检测**

实现电阻网络（R1，R2和R3）以使ADC输入范围适应总线电压DC范围（见图3）。

图3.典型的MC总线电压检测



从固件的角度来看，使用STM32 MC Workbench PC软件可以轻松地调整电阻值。用户通过功率级区域内的总线电压感测块根据其HW实现来配置电阻器网络。

使用一个具有一个通道的ADC定期测量总线电压（仅限常规转换）。但是，由于MC固件的实时约束（电流感应采集优先级），可以延迟常规转换。禁用此选项时，MC固件始终使用标称额定总线电压作为固定值，在STM32 MC Workbench PC软件中定义。

**2.1.4安全机制**

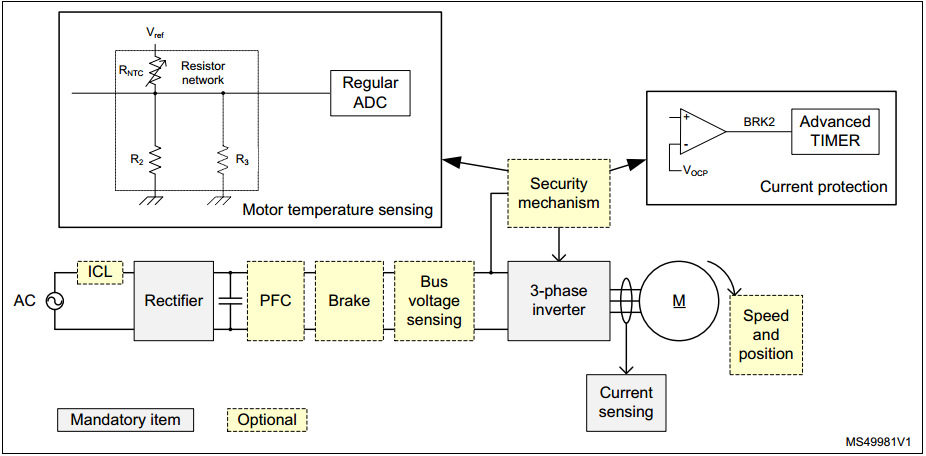
一些安全机制（见图4）是通过硬件实现的（由于内部功能）。支持的硬件安全机制包括：

•过压保护 可由硬件完成，但由固件管理（由于总线电压检测采集）。

•过流保护 将参考电压（MCU内部或外部）转换为参考电流作为最大上限，然后与测量电流进行比较。然后，比较器输出可以触发高级定时器的Break或Break2输入，以立即停止PWM生成

•电机温度 检测电阻网络（RNTC，R2和R3）用于极化测量电压（用作电子驱动器的温度图像）。一个具有一个通道的ADC可以定期测量电压.

图4.典型的MC安全机制



从固件的角度来看，使用STM32 MC Workbench PC软件可以轻松设置过流（或电机温度）保护阈值。用户根据其HW实现，通过功率级区域内的过电流保护模块（或温度感测模块）配置这些阈值。

**2.1.5MCU**

当设计源自ST示例板时，用户可能仅在同一MCU线内（例如从STM32F030RC到STM32F030K6）进行更改。

更换MCU时，用户必须仔细检查所有相同外设的可用性（如前几段所述），即：

•带有（可用）通道的ADC

•高级定时器

•运算放大器

•比较器

•时钟频率。

更换微控制器之前，请参见第2.3节：STM32 MC工作台和第2.4节：STM32CubeMX。

**2.2 STM32 MC Workbench和STM32CubeMX之间的交互**

STM32 MC Workbench在\* .ioc文件中存储两组数据：

1。与为电机控制应用程序选择的MCU相关的配置信息和使用的外围设备（硬件IP）

2。纯电机控制参数及其值。

第一组允许STM32CubeMX为MCU及其外设生成初始化代码。由此，当电机控制软件子系统开始初始化时，电机控制应用所需的所有外围设备都可以使用。

第二组提供STM32CubeMX生成电机控制驾驶舱所需的信息，并选择适当的电机控制源或库文件以包含在生成的项目中。其中一些数据也可以在生成的源文件中访问固件。固件将它们用于自己的初始化以及操作期间。

用户可以从STM32CubeMX进一步修改STM32 MC Workbench生成的\* .ioc文件。这样做时，需要直接从STM32CubeMX再次生成软件项目。但请注意，STM32 MC Workbench无法完全读取\* .ioc文件，因此可能会覆盖对STM32CubeMX中的\* .ioc文件所做的任何更改。

**2.3 STM32 MC Workbench**

**2.3.1修改电机控制参数**

STM32 MC Workbench用于配置硬件外设设置及其固件参数，并通过STM32CubeMX生成电机控制项目。

请注意，电机控制参数主要由STM32 MC Workbench中的“驱动管理”阶段配置。

修改电机控制参数的正确方法包括使用STM32 MC Workbench进行修改，然后使用“更新”功能重新生成项目。

**2.3.2 Using another MCU**

STM32 MC Workbench支持STM32 MCU的子集，主要是STM32 MC SDK v5.0支持的Nucleo和Eval板上的子集。

由于STM32CubeMX的导入功能，用户可以更改其电机控制应用中使用的MCU。此功能允许用户在新的STM32CubeMX项目中导入现有的\* .ioc文件，其中只选择了MCU。此功能受到一些限制（有关更多信息，请参阅UM1718“用于STM32配置和初始化C代码生成的STM32CubeMX”，可在www.st.com上获得）

**2.3.3时钟树配置**

时钟配置是电机控制软件子系统性能的敏感因素。 STM32 MC Workbench从电机控制角度将其设置为最佳值，STM32CubeMX不得更改它。

特别是，这适用于内核时钟，也适用于ADC的时钟和电机控制软件子系统使用的定时器。

**2.3.4处理中断**

出于性能原因，STM32 MC FoC固件实现了它使用的中断处理程序。因此，STM32 MC Workbench会阻止STM32CubeMX自行生成这些处理程序。

此外，STM32 MC Workbench选择这些中断进行初始化序列排序。中断初始化的顺序本身并不重要，此功能可确保在电机控制软件子系统之后，STM32CubeMX生成的初始化序列中最后执行这些中断的初始化。

在最后阶段，STM32 MC Workbench配置NVIC外设的优先级组功能以及电机控制子系统使用的中断的优先级和子优先级。必须在\* .ioc文件中保持这三个项目不变。不得使用STM32CubeMX更改它们。有关处理电机控制子系统中的中断优先级的更多详细信息，请参阅UM2392。

**2.3.5更改专用引脚分配**

STM32 MC Workbench为电机控制应用程序使用的各种输入和输出引脚生成初始配置。只要有可能，用户就可以使用STM32 MC Workbench选择引脚分配。但是，如果确实需要，用户可以从STM32CubeMX更改引脚分配。

某些引脚具有明确的名称，分别以电机1或电机2相关引脚的M1\_或M2\_开头，以及两个电机共用的引脚的M1M2\_或M2M1\_。电机控制驾驶舱代码在内部使用这些名称来初始化电机控制子系统。电机控制引脚名称的示例是M1\_PWM\_VH，它指的是电机1的V相的PWM输出。

要更改引脚分配，用户需要为要移动的功能选择合适的备用引脚，根据移动的功能配置此新引脚，然后将名称从前一个引脚准确传输到新引脚。

更改引脚后，必须使用“更新”功能保存使用STM32 MC Workbench进行的任何进一步修改，以保留\* .ioc文件中的新分配。

**2.4 STM32CubeMX**

STM32CubeMX可用于编辑和修改所选MCU中的硬件外设配置，以适应最终用户应用。

但是，用户需要检查新外设的配置是否不会更改先前由STM32 MC Workbench为电机控制外设设置的时钟树参数。

**2.4.1用户项目配置**

除硬件外设配置外，STM32CubeMX还允许用户配置其软件项目的生成方式。但是，建议不要在生成的\* .ioc文件中更改从STM32 MC Workbench中选择的项目设置。

**2.4.2修改电机控制参数**

更改电机控制应用程序已使用的外围设备配置非常棘手，如果绝对需要，必须非常小心。这里区分两种情况：

•更改STM32 MC Workbench设置的参数值。这可能会导致应用程序出现故障，原因与第2.2节：STM32 MC Workbench和STM32CubeMX之间的交互中突出显示的原因相同。

•使用电机控制外设的自由通道。这不是问题，只要对外围设备的其余部分没有影响，特别是对其一般参数没有影响。例如，只要触发源，从动模式和自动重载寄存器保持不变，就可以使用用于电机控制的定时器的附加通道。但是，在使用ADC的空闲通道时必须小心外围。应用程序可以根据自己的需要预定定期转换（不是注入转换）。但是，它应该使用API​​函数（参见UM2392）。

**2.4.3添加新引脚或更改引脚分配**

配置电机控制应用程序未使用的外围设备通常不是问题，可以自由完成。

但是，请注意，在使用它并更新ioc文件时，将覆盖在STM32 MC Workbench外部执行的电机控制专用引脚的更改。更改引脚分配的首选方法显然是通过STM32 MC Workbench。