

# AN1292与电机控制应用程序框架对比

---

# 1 AN1292与MC应用程序框架对比

## 1.1 前言

文档该部分对AN1292软件与motorBench™开发套件随附的MC应用程序框架之间的差异进行了分析，旨在回顾全新MC应用程序框架代码相对于AN1292的参考应用笔记软件的改进和限制。

### 1.1.1 自调试

自调试是motorBench™开发套件的一个组成部分，在专用dsPIC®软件的协助下，可确定PMSM电机的以下参数：

1. 定子电阻 (R)
2. 定子电感 ( $L_d$ 和 $L_q$ )
3. 反电动势常数 ( $K_e$ )
4. 摩擦转矩 ( $T_f$ )
5. 粘性阻尼 (B)
6. 惯性 (J)

### 1.1.2 自动调整

自动调整是motorBench™开发套件的一个组成部分，可根据电机和硬件参数计算特定硬件系统的PI控制器增益。

### 1.1.3 MC应用程序框架

电机控制应用程序框架 (Motor Control Application Framework, MCAF) 是一个固件包，其中包括多个文件，可用于为任何所需MPLAB® X项目生成代码。MC应用程序框架通过motorBench™开发套件“呈现”到应用程序源代码的特定实例中，它根据配置信息、自调试测量值以及由此得出的计算结果来调整电机控制环。

## 1.2 比较的方面

该部分对AN1292软件与motorBench™开发套件随附的和MC应用程序框架进行了一系列主题比较分析。

### 1.2.1 变更汇总

下表高度总结了AN1292软件与电机控制应用程序框架（MC应用程序框架，即MCAF）之间的详细对比信息。

特性	AN1292软件	MC应用程序框架
峰值电机转矩	+ 在小型电机上呈现出较高的转矩 - 可能会损坏硬件	+ 转矩受电机规范限制 - 呈现出较低的转矩
调整工作	- 需要手动调整 + 可针对特定应用进行优化	+ 自动调整，只需有限的用户控制 + 适用于各种应用 - 调整仅限于恒定负载（即速度变化时粘性阻尼保持恒定）
轴抓取测试	+ 似乎很难停止电机 + 通常针对手抓测试进行调整 - 无过载/停转检测	- 由于采用新的停转检测功能，电机似乎很容易停止 - 针对通用负载测试进行调整
电机过载后重新启动	- 不可用	+ 自动尝试启动电机“n”次（可以禁止/更改）
可视用户界面	- 不可用	+ LED指示正常操作（心跳闪烁）或错误代码
电机停止方法	+ 电机快速停止 - 电机关闭时的相电流较高；这会 增加总线电压或损坏硬件	+ 电机缓慢减速 + 电机关闭时的相电流可忽略不计 - 电机需要更多时间才能停止下来
诊断和测试	- 受RTDM/DMCI性能限制 + DMCI插件可供公众使用	+ 支持林茨机电一体化中心的 <a href="#">X2C-Scope</a> + 也支持RTDM/DMCI + 支持高级测试模式
MC代码的CPU执行 开销时间	+ 需要较少的CPU开销	- 由于新增功能而导致CPU开销增大
模块化编码架构	- 仅限于传统代码架构	+ 更容易测试并移植到新硬件 + 更易于阅读代码 - 新架构需要学习曲线 - 编译代码的时间稍微延长
代码库和发布架构	+ 为每个硬件组合发布单独的代码库 - 难以维护多个代码库 - 没有HAL，难以针对其他硬件移植	+ 统一代码库，支持多种硬件配置 + HAL允许轻松移植到其他硬件 - 需要motorBench™开发套件
电机启动	+ 简单的算法 + 通过谨慎调整启动参数可与多个电机配合使用 - 大多数情况下会发生速度过冲 - 如果调整不当，可能会发生过流故障	+ 平稳电机启动，无速度过冲/电流浪涌 + 在多台电机上成功通过测试 + 针对特定的电机和机械负载自动调整 - 需要确定电机和负载的机械参数

\* 在上表中，(+) 表示优点，(-) 表示限制。

## 1.2.2 电机电流限制

电机和电机控制板数据手册通常指定保证其安全运行的最大连续电流。如果电机电流超过电机或电机控制板的规范值，则会增大由于过热的风险。除了连续电流额定值之外，一些电机数据手册和电机控制板数据手册还指定了电机的峰值电流额定值，这是电机/电机控制板在不损坏的前提下短时间内可以处理的最大电流。

理想情况下，设计为与特定电机和特定电机控制板一起工作的电机驱动软件应将最大连续工作电流限制在连续电机电流以及安全连续控制板电流的最低值以内。

		Part/Model Number
Specification	Units	A0421046NC
Supply Voltage	VDC	48
Continuous Stall Torque	oz-in	17
	Nm	0.12
Speed @ Cont. Torque	RPM	4000
Current @ Cont. Torque	Amps (A)	2.75
Continuous Output Power	Watts (W)	50.26
Motor Constant	oz-in/sqrt W	3.89
	Nm/sqrt W	0.027
Torque Constant	oz-in/A	6.179
	Nm/A	0.044
Voltage Constant	V/krpm	4.57
	V/rad/s	0.044
Terminal Resistance	Ohms	2.52
Inductance	mH	1.66
Max. Speed	RPM	4500
Peak Current	Amps (A)	8.25
Peak Torque	oz-in	51
	Nm	0.3601

图1: 典型电机数据手册中指定的峰值和连续电流额定值

### 1.2.2.1 传统AN1292软件如何实现这一点？

AN1292的传统软件将开环电机启动期间的电机电流限制为由`userparams.h`文件中的宏`Q_CURRENT_REF_OPENLOOP`定义的值。一旦电机完成启动阶段并进入闭环正常工作模式，电机电流便受速度控制器输出的限制。由于速度控制器输出预设为在`0x5000`处饱和（见`userparams.h`文件中的宏`SPEEDCNTR_OUTMAX`）， $I_{Qsat}$ 的参考电流限制为通过公式(1)计算的 $I_{Qsat}$ 。

$$I_{Qsat} = 0x5000 \times \frac{I_{peak}}{2^{15}} \dots\dots\dots (1)$$

其中，

$I_{peak}$ 是ADC可测量的峰值电流

对于峰值电流能力设计为4.4A的MCLV2开发板，基于公式(1)计算出的电机电流 $I_{Qsat}$ 的最大值约为 $2.75A_{peak} = 1.94A_{RMS}$ 。如果最终应用需要不同的电流饱和和限制，则需要将赋给`userparams.h`文件中的宏`SPEEDCNTR_OUTMAX`的值更新为通过公式(1)计算的所需值。

### 1.2.2.2 如何在MC应用程序框架/motorBench™开发套件中执行此操作？

MC应用程序框架将最大电机电流限制为以下两项中的较小值：

1. 电机的连续电流额定值
2. 控制板的连续电流额定值。

这可防止控制板和电机发生损坏。

### 1.2.2.3 MC应用程序框架的显著优点

电机电流限制可防止损坏电机和电机控制板。

### 1.2.2.4 MC应用程序框架的显著缺点

对电机电流施加硬限制时，即使电机/控制板能够在短时间内提供高很多的输出，也会有效地将电机产生的峰值机械转矩限制为一个保守值。

### 1.2.2.5 改进范围

对电机电流施加软限制（而不是硬限制），为电机电流在短时间内超出连续电机/控制板规范值留有一定的裕量。

## 1.2.3 可轻松为新电机调整应用程序代码

通常情况下，Microchip应用笔记软件项目针对特定电机组进行调整。例如：Hurst DMA0204024B101、Hurst DMB0224C10002和Motor-80等。虽然这些软件项目适用于演示目的，但在用于运行不同的电机之前，需要重新调整。假定硬件相同的情况下，重新调整过程需要更新应用笔记的以下主要组成部分：

1. 无传感器估算器参数
2. 电流控制环的PI控制器参数
3. 速度控制环的PI控制器参数
4. 电机启动参数

### 1.2.3.1 为新电机调整传统AN1292软件

为了使用传统AN1292软件运行新电机，最终用户将必须基于应用笔记调整指南文档中所述的新电机参数手动更新`userparams.h`文件。涉及的高级操作步骤汇总如下：

1. 从新电机的数据手册中获取其电气参数。如果电机数据手册不可用/不准确/不完整，请使用数字万用表（Digital Multimeter, DMM）、示波器和手钻等实验室设备来获取电机的电气参数
2. 使用电机电气参数更新软件项目中包含的`tuning_params.xls`电子表格
3. 将一组自动计算的值从`tuning_params.xls`电子表格手动复制到`userparams.h`文件。在某些情况下，电子表格和软件中的换算/预除因子也可能需要进行调整
4. 切换到开环模式并更新电机启动参数
5. 手动调整 $I_a$ 和 $I_b$ 电流控制环的PI控制器参数。然后，切换到闭环模式并调整速度控制环的PI控制器参数
6. 基于电机及其机械负载更新速度的变化率（`userparams.h`文件中的`SPEEDREFRAMP`）

### 1.2.3.2 为新电机调整motorBench™开发套件

为了使用motorBench™开发套件运行新电机，用户只需输入电机的基本铭牌详细信息。motorBench™开发套件将基于该信息运行自调试过程以确定电机的电气和机械参数。之后，它会自动计算所有应用笔记参数，并生成最终的应用笔记软件，该软件可立即用于运行新电机。

### 1.2.3.3 motorBench™开发套件的显著优点

由于motorBench™开发套件可以自动生成已经过调整、可立即操作新电机的应用程序软件，因此与手动对传统AN1292软件执行相同调整相比，这可节省大量时间和精力。

### 1.2.3.4 motorBench™开发套件的显著缺点

尽管motorBench™开发套件背后的意图和设计旨在支持大多数电机类型和机械负载，但它本身不能支持最终用户可能需要使用的所有独特硬件组合。

在这些特殊情况下，最终用户可首先在motorBench™开发套件中选择最能代表实际硬件的电机和负载选项。此后，最终用户可以使用motorBench™开发套件中提供的界面对调整和应用参数进行微调，以实现所需的调整性能。除此之外，最终用户还可以完全访问和控制最终生成的软件，以对软件和调整参数进行必要的修改。

## 1.2.4 速度和电流控制器调整特性

电流和速度控制器的PI控制器增益控制PI控制器对扰动和速度/电流参考输入变化的响应。这些参数必须进行优化调整，以防止相关控制呈现不稳定性，同时仍保持稳健、可预测的干扰响应。

### 1.2.4.1 传统AN1292软件如何实现这一点？

传统AN1292软件中的PI控制器参数基于以下硬件配置手动调整：

- MCLV2开发板
- Hurst电机DMB0224C10002
- $V_{BUS} = 24V$ ，PWM死区时间 = 2  $\mu s$ ，PWM频率 = 20 kHz

默认调整在迭代测试之后结束，以确保电机可靠启动，在弱磁模式下平稳运行，并为手动装载的电机轴提供最佳响应（也称为“轴抓取”测试）。

### 1.2.4.2 如何在MC应用程序框架/motorBench™开发套件中执行此操作？

MC应用程序框架中电流和速度控制环的PI控制器参数由motorBench™开发套件基于电机参数以及交叉频率下相位裕量和PI相位的预配置自动计算。默认情况下，这些增益的调整较为保守，以确保在各种电机中提供可预测的行为。

表1总结了传统AN1292软件中使用的PI控制器参数与motorBench™开发套件生成的PI控制器参数的对比情况，该对比采用相同的硬件配置以及交叉频率下的相位裕量和PI相位的默认值。

控制环	参数	值		特性
		传统AN1292软件	motorBench™开发套件	
电流	带宽 $\omega_c$ [rad/s]	2.84	1278	AN1292的带宽非常低，积分项对PI增益的影响很大。
	相位裕量[°]	103.50	79.99	
	交叉频率下的PI相位[°]	88.70	45.01	
速度	带宽 $\omega_c$ [rad/s]	182.60	120.3	积分项对AN1292中PI控制器增益的影响很小。
	相位裕量[°]	42.54	50	
	交叉频率下的PI相位[°]	3.92	10	

表1: PI控制器参数对比

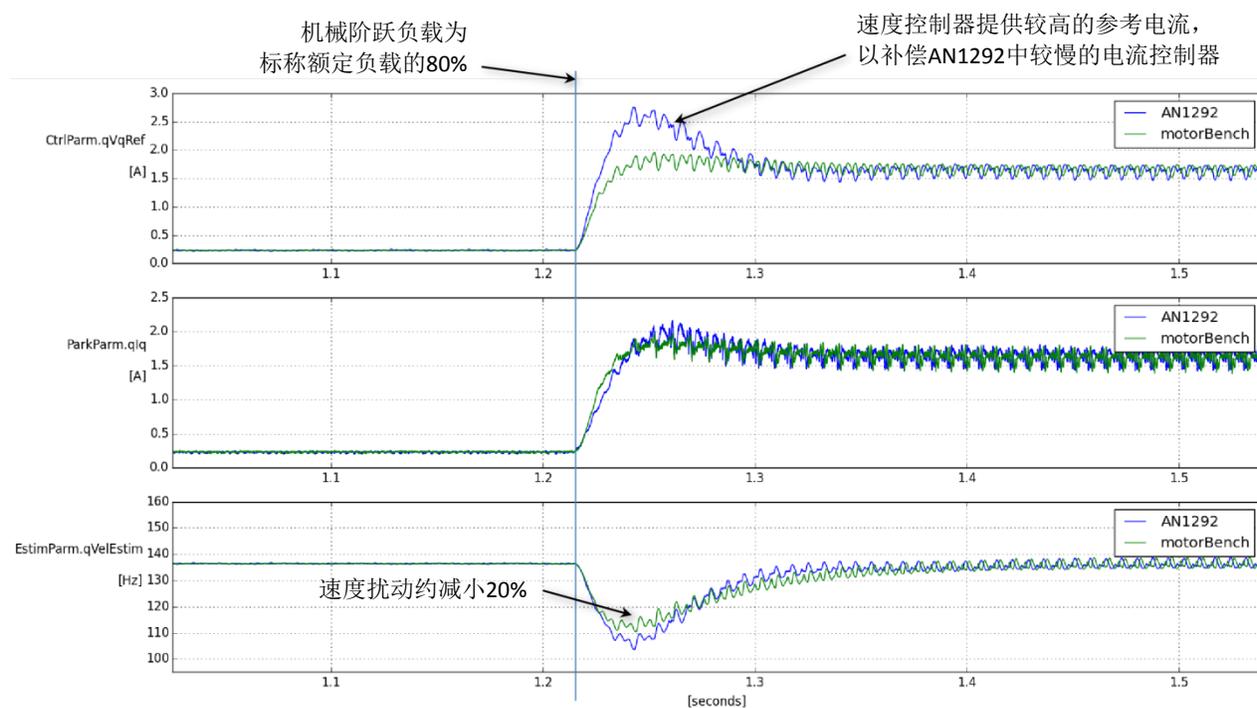


图2: 机械阶跃负载为标称额定负载的80%时电流和速度控制器响应对比图

图2所示为传统AN1292软件和经motorBench™开发套件调整的AN1292软件的电流和速度控制器响应图。在这两种情况下，Hurst电机DMB0224C10002均面临完全相同的测试条件——机械阶跃负载为标称额定负载的80%。该测试使用低惯性台式测力计，以降低附加阻尼系数和惯性对速度控制环解调的影响。

#### 1.2.4.3 经motorBench™开发套件调整的PI控制器参数的显著优点

由motorBench™开发套件计算出的PI控制器增益可直接用于各种电机。

#### 1.2.4.4 经motorBench™开发套件调整的PI控制器参数的显著缺点

从motorBench™开发套件获取调整参数无疑是一个很好的切入点，但它并不总是能够实现所有应用程序的独特目标。最终用户将必须手动微调motorBench™开发套件的输出，以获得其所需的精确控制器响应。

### 1.2.5 电机启动

电机启动程序提供了一种关键功能，即以零速启动电机并将其速度提升至能够保证无传感器位置估算器可靠运行的最小值。

#### 1.2.5.1 传统AN1292软件如何实现这一点？

AN1292的电机启动程序采用简单的三步法：

1. 锁定阶段
  - a. 为Iq和Id使能闭环电流控制
  - b. 在Iq上施加等于电机连续电流额定值的直流电流
  - c. 等待一段为电机对齐预先定义的时间
2. 速度上升阶段
  - a. 开始以一定的速率递增所施加电压的相位角（即强制角）以达到所需速度
  - b. 递增速度以使电机速度上升至能够确保无传感器估算器可靠运行的最小速度
3. 转换阶段
  - a. 将换向角从强制角切换为估算角以完成转换
  - b. 减小添加到估算的电机角的角偏移量

图3所示为使用AN1292软件、MCLV2开发板和Hurst电机（DMB0224C10002）且未连接机械负载时启动阶段的电机电流和速度曲线图。

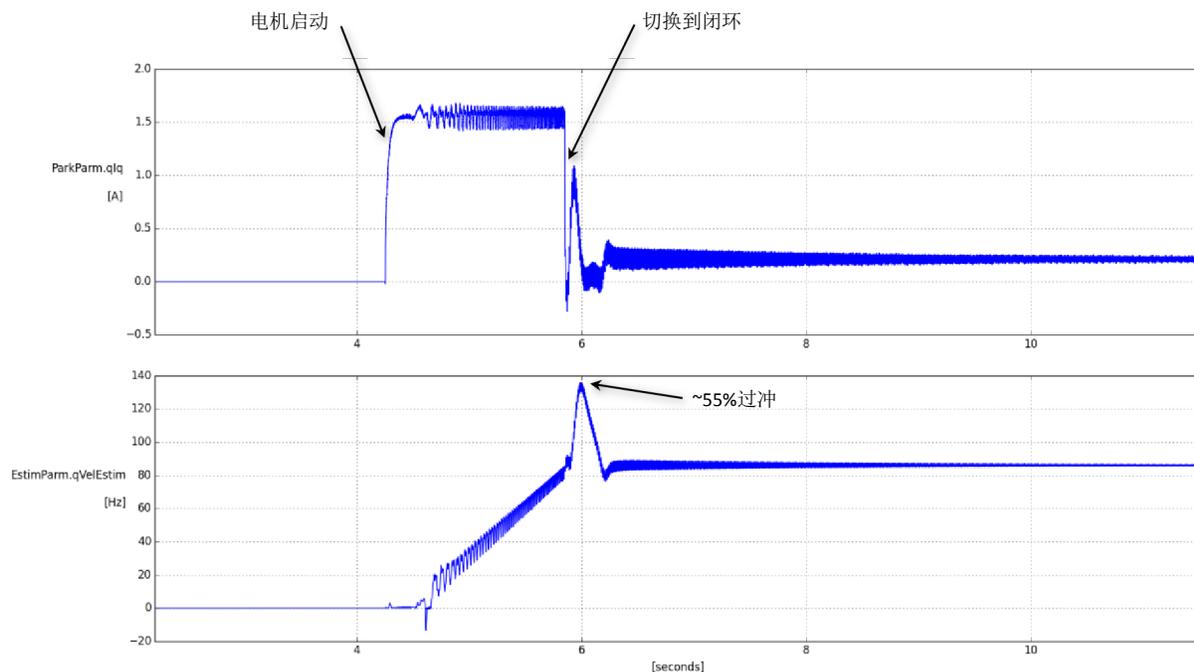


图3：使用AN1292软件且无机械负载时启动阶段的电机电流和速度曲线图

由图3可知，电机电流和速度在启动阶段受到瞬态干扰。图4所示为使用AN1292软件、MCLV2开发板和Hurst电机（DMB0224C10002）且机械负载约占标称值50%时启动阶段的电机电流和速度曲线图。

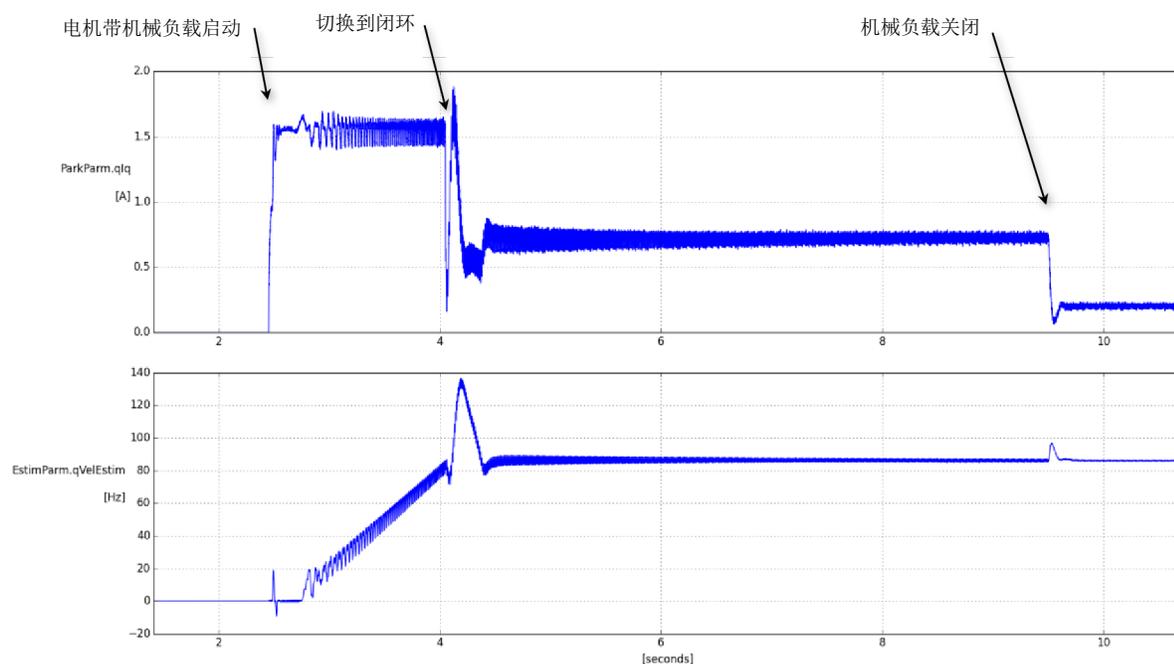


图4：使用AN1292软件且机械负载约占标称值50%时启动阶段的电机电流和速度曲线图

对比图3和图4可知，电机负载占标称值50%时（图4）电机电流瞬态尖峰的幅值相对于无负载条件（图3）增加约200%。

### 1.2.5.2 如何在MC应用程序框架/motorBench™开发套件中执行此操作？

MC应用程序框架实现了新设计的电机启动程序，以便：

1. 消除或减少瞬态干扰
2. 适应启动过程中的负载变化
3. 提高电机启动的可靠性

有关新电机启动算法的更多详细信息，请参见文档中的电机启动部分。

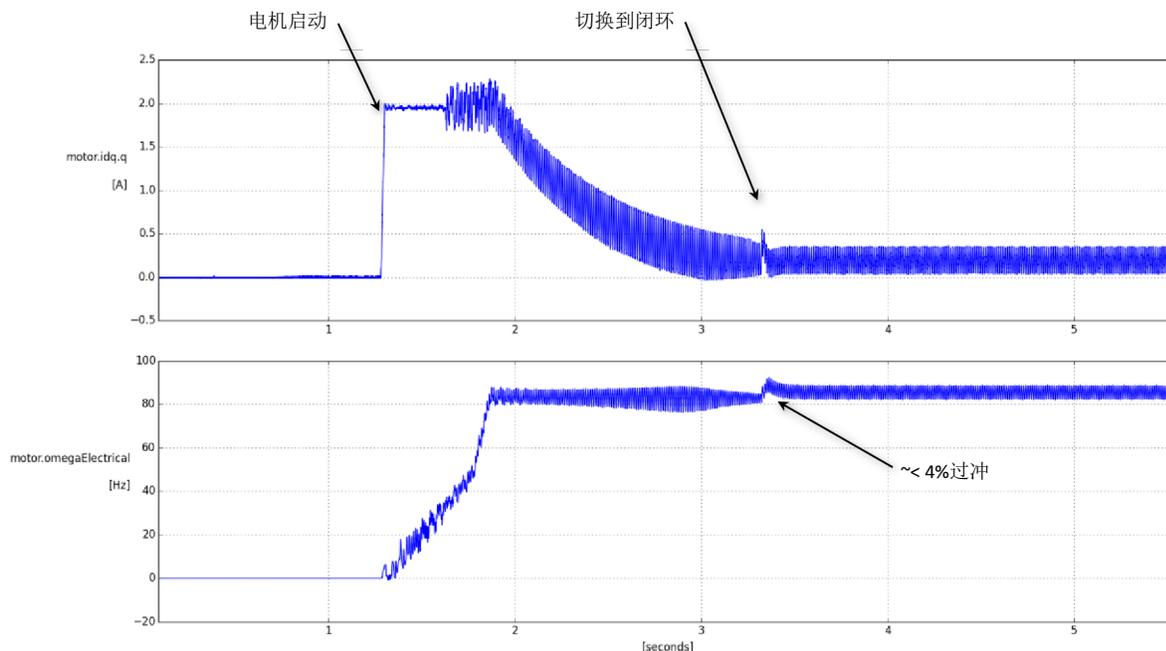


图5：使用MC应用程序框架且无机械负载时启动阶段的电机电流和速度曲线图

图5所示为使用MC应用程序框架软件、MCLV2开发板和Hurst电机（DMB0224C10002）且未连接机械负载时启动阶段的电机电流和速度曲线图。由图5可知，电机启动期间和之后几乎没有瞬态干扰。

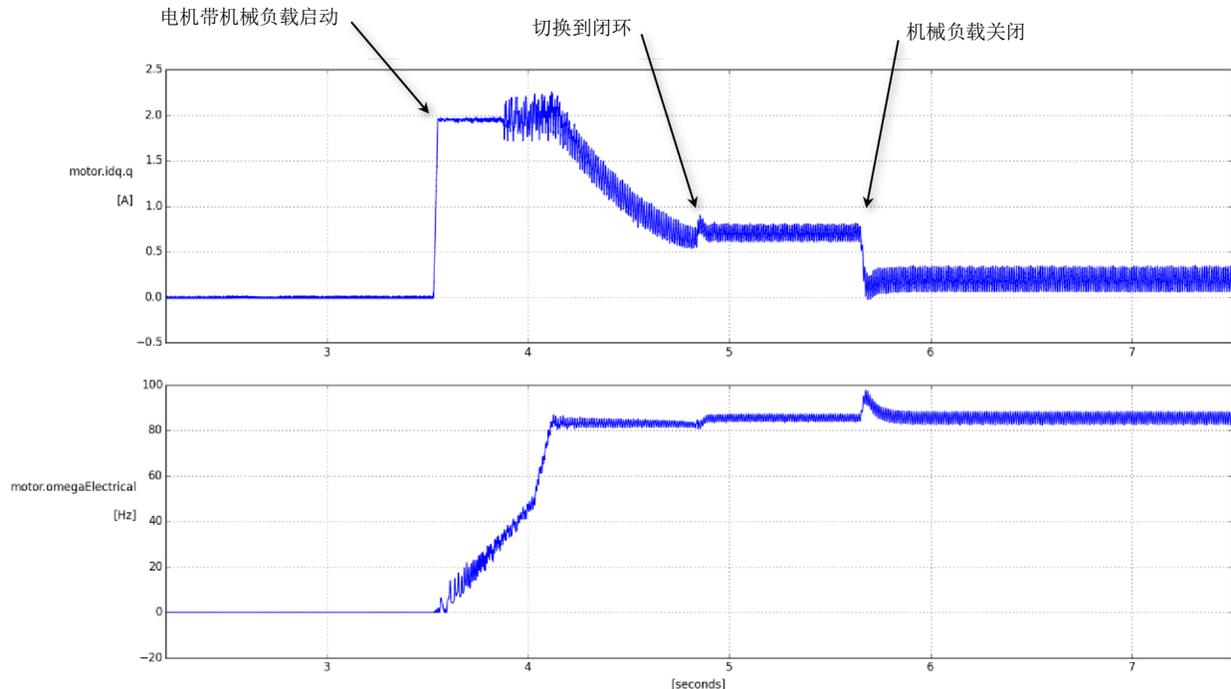


图6：使用MC应用程序框架且机械负载为标称值50%时启动阶段的电机电流和速度曲线图

图6所示为使用MC应用程序框架软件、MCLV2开发板和Hurst电机（DMB0224C10002）且机械负载约占标称值50%时启动阶段的电机电流和速度曲线图。

### 1.2.5.3 motorBench™开发套件方法的显著优点

新设计的电机启动程序可实现速度过冲/电流瞬变最小的平稳电机启动。启动程序参数会针对特定的电机和机械负载组合自动调整。

### 1.2.5.4 motorBench™开发套件方法的显著缺点

新的电机启动程序需要motorBench™开发套件来计算启动参数。这些启动参数基于电机的机械参数和电机启动时使用的机械负载的机械参数。如果电机或负载的机械参数发生变化，则将需要使用motorBench™开发套件重新调整启动参数。

## 1.2.6 电机停转检测

电机停转事件可以定义为以下情况：

1. 估计电机角的精度不足以同步驱动电机
2. 电机转矩、磁通和速度控制环均因此不再有效。

控制算法及时检测电机停转情况并妥善处理电机停转的后果非常重要。如果控制算法无法检测到电机停转情况，可能会导致过流和/或过压故障，进而可能损坏硬件。

### 1.2.6.1 传统AN1292软件中的停转检测逻辑

尽管旧版应用笔记（如AN1078）的软件使用电机速度作为检测电机停转情况的主要信息来源，但AN1292并未实现此特性。

### 1.2.6.2 MC应用程序框架中的停转检测逻辑

MC应用程序框架使用新设计的停转检测算法，该算法使用以下重要控制参数来检测电机停转：

1. 基于PLL的无传感器电机位置估算器的反电动势电压变化
2. 基于PLL的无传感器电机位置估算器的反电动势电压的符号
3. 转矩角
4. 电机转矩电流值，即 $I_q$
5. 电机速度

更多详细信息，请参见文档中的“电机停转检测算法”部分。

### 1.2.6.3 显著优点

MC应用程序框架中使用的新停转检测算法已在多台带有或不带机械负载的电机上进行了测试，以确保可靠地检测电机停转情况。该功能从应用程序角度提供了高度的控制可靠性，并且还可以防止由于不受控制的电机停转状况导致硬件损坏。

### 1.2.6.4 显著缺点

在没有任何停转检测算法时，某些情况下，使用传统AN1292软件运行的电机在超出其额定转矩时会发生“周跳”。当发生这种情况时，转子将停止旋转并以较大的相电流振动。此时如果减小电机负载，电机可从此停转状态恢复并继续旋转。由于这种行为与交流感应电机相似，因此与MC应用程序框架相比可能对未经训练的观测器更有利。

## 1.2.7 从电机停转状态恢复

一旦检测到电机停转状态，控制算法必须采取进一步的措施以便：

1. 在应用需要时关闭然后重新启动电机
2. 谨慎地修正停转状态并恢复对电机的控制

在某些应用中，电机在停转状态后自动重启可能不合需要。因此，从电机停转状态恢复的决定和实际恢复行为通常取决于最终应用指示的所需逻辑。

### 1.2.7.1 传统AN1292软件中的恢复逻辑

传统AN1292软件不支持从电机停转状态恢复。但是，根据第1.2.6.4节所讨论的内容，可能会出现电机从停转状态自动恢复的意外情况。

### 1.2.7.2 MC应用程序框架中的恢复逻辑

MC应用程序框架实现了一个自动恢复逻辑，旨在关闭电机，等待一段预定时间以供电机缓慢减速，然后尝试重新启动电机。恢复逻辑预编程为在标记故障状态之前连续重试恢复序列指定的次数。应用程序设计人员可以更改尝试进行恢复重试的次数。如果应用程序不希望从电机停转状态自动恢复，则该数字也可以设置为“0”。

### 1.2.7.3 显著优点

在MC应用程序框架中设计的恢复逻辑可以使最终应用程序免受从电机停转状态恢复过程中的复杂情况的影响，同时仍能为其提供对停转检测和恢复过程的抽象访问。

### 1.2.7.4 显著缺点

[无]

## 1.2.8 外部用户界面 (UI)

用户界面 (User Interface, UI) 指的是应用程序中的固件逻辑，它使用LED、开关、测试点跟踪输出等人机界面元素向用户/应用程序设计人员提供有关应用程序状态的反馈信息。应用笔记软件应该利用外部界面来帮助应用程序设计人员完成开发和调试阶段。可以通过这些界面显示有关状态、故障和其他事件的宝贵实时信息，以便应用程序设计人员可以避免使用更多的时间和资源密集型调试工具。

### 1.2.8.1 传统AN1292软件中的外部UI支持

在开发传统AN1292软件时，Microchip电机控制开发板 (MCHV、MCLV和MCSM板) 上不提供LED等UI元素。因此，传统AN1292软件支持的惟一UI元素是电位器和两个按钮。

从第二代电机控制开发板 (MCHV-2、MCLV-2和双电机控制板) 开始提供LED和附加测试点等UI元素。

### 1.2.8.2 MC应用程序框架中的外部UI

MC应用程序框架利用按钮、电位器和LED提供支持以下功能的全新UI界面：

1. “心跳”指示 ↔ 固件处于工作状态并正常运行
2. 电机方向指示
3. 使用基本4 LED闪烁序列进行故障代码显示
4. 使用按钮界面进行故障处理/清除

有关应用程序界面中提供的外部UI的更多详细信息，请参见文档的外部界面部分。

### 1.2.8.3 MC应用程序框架中的外部UI的显著优点

在使用MC应用程序框架时，应用程序设计人员无需使用外部调试工具即可轻松确定固件的状态。

### 1.2.8.4 显著缺点

[无]

## 1.2.9 电机停止技术

电机属于机电系统，凭借自身所具备的转动惯量，可以储存动能以及产生用于驱动机械负载的转矩。除此之外，机械负载本身可以在其转动惯量中存储大量的能量。因此，在停止电机旋转状态的同时，电机控制应用程序将必须通过以下方式之一消耗该动能：

1. 最简单、最缓慢且最安全的方法：关闭控制电压，使电机自行缓慢减速。在这种情况下，能量会以电机和负载内热量的形式逐渐消耗。相对而言，该技术在电机关闭阶段对电机控制硬件施加的应力最小。
2. 快速但不计后果的方法：通过短接所有下桥臂或所有上桥臂逆变器开关，将电机相电流重新循环回电机绕组。这将很快消耗掉动能：进入电机绕组和逆变器开关的部分以热量形式消耗，进入定子叠片的部分以涡流损耗的形式消耗。但是，这样会产生大幅值的电机电流，进而可能损坏电机控制硬件和/或电机本身。
3. 动态制动方法：在下桥臂逆变器开关上施加一个受控的PWM占空比以充当斩波器电路并将能量泵入逆变器电源。如果得不到谨慎控制，可能会增加逆变器总线电压并可能损坏电机控制硬件。
4. 受控制动/再生方法：使用速度控制环将电机速度降至零。这也会导致再生和逆变器总线电压增加。因此，为了防止由于逆变器总线电压的增加而对电机控制硬件造成过大的压力，必须根据电动机+负载的总转动惯量将速度下降速率控制在安全值。

#### 1.2.9.1 传统AN1292软件中的电机停止技术

为了关闭电机，传统AN1292软件通过短接所有下桥臂逆变器开关，将电机相电流重新循环回电机绕组（第1.2.9节中讨论的技术2）。在电机和负载转动惯量较大的应用（例如：洗衣机）中，使用这种技术关闭电机会导致较大的电机相电流，进而导致电机控制硬件和电机本身所承受的压力过大。

#### 1.2.9.2 如何在MC应用程序框架/motorBench™开发套件中执行此操作？

要关闭电机，MC应用程序框架需关闭控制电压，使电机自行缓慢减速（即类似于第1.2.9节中讨论的技术1）。除此之外，所有三个下桥臂PWM开关都维持小占空比，以使自举电容保持充电状态。

#### 1.2.9.3 MC应用程序框架方法的显著优点

该方法对电机控制硬件施加的压力最小。

图7所示为分别使用传统AN1292软件和MC应用程序框架关闭电机时电机相电流的对比图。这种情况下的测试设置包括MCLV2开发板和Hurst电机——DMB0224C10002，其中惯性负载为 $39.01 \mu\text{Nms}^2$ ， $V_{\text{BUS}} = 24\text{V}$ ，PWM死区时间 =  $2 \mu\text{s}$ ，PWM频率 =  $20 \text{kHz}$ 。由图7可知，当电机关闭时，AN1292软件会引起大电流浪涌，而MC应用程序框架在类似测试条件下不会产生任何电机相电流。

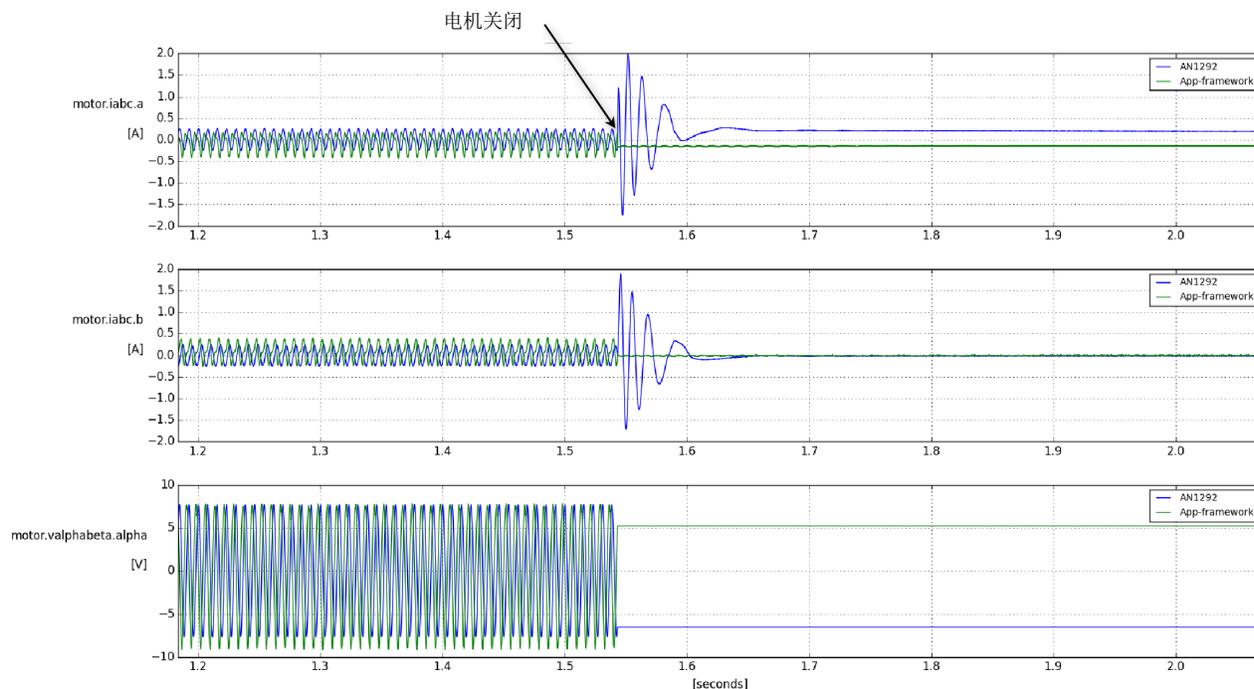


图7：电机关闭时的电机相电流对比图

#### 1.2.9.4 MC应用程序框架方法的显著缺点

与其他电机停止技术相比，此方法需要较长的时间才能使电机完全停止。

### 1.2.10 模块化代码架构

模块化代码架构是指一种源代码合理组织成多个独立模块且每个模块都有特定用途的软件架构。由于以下重要原因，应用笔记软件必须实现模块化：

1. 应用笔记软件的最终用户在将其用于最终应用之前通常会对其进行修改。模块化软件允许最终用户轻松识别和修改软件的各个模块，而无需搜索源代码的特定部分
2. 模块化源代码方便最终用户维护和阅读
3. 最终用户可以轻松地对他们所做的任何修改运行单元测试

#### 1.2.10.1 传统AN1292软件中的代码架构

##### 1.2.10.1.1 数据结构和数据访问

传统AN1292软件设计为具有一组八个托管应用笔记所需的所有数据变量的核心数据结构以及一组隐性地将这些结构用作全局变量的应用笔记特定函数，如图8所示。其中一些函数采用汇编语言编写，这些汇编函数使用包含在相应\*.inc文件中的ASM30 API访问核心数据结构。

##### 1.2.10.1.2 硬件访问

根据模块化编码架构的构思，所有dsPIC函数的初始化程序都包含在单个源文件（即`periph.c`）中。但是，软件项目其余部分的许多应用笔记函数直接访问dsPIC特殊功能寄存器，因此这些函数在设计上具有硬件依赖性。

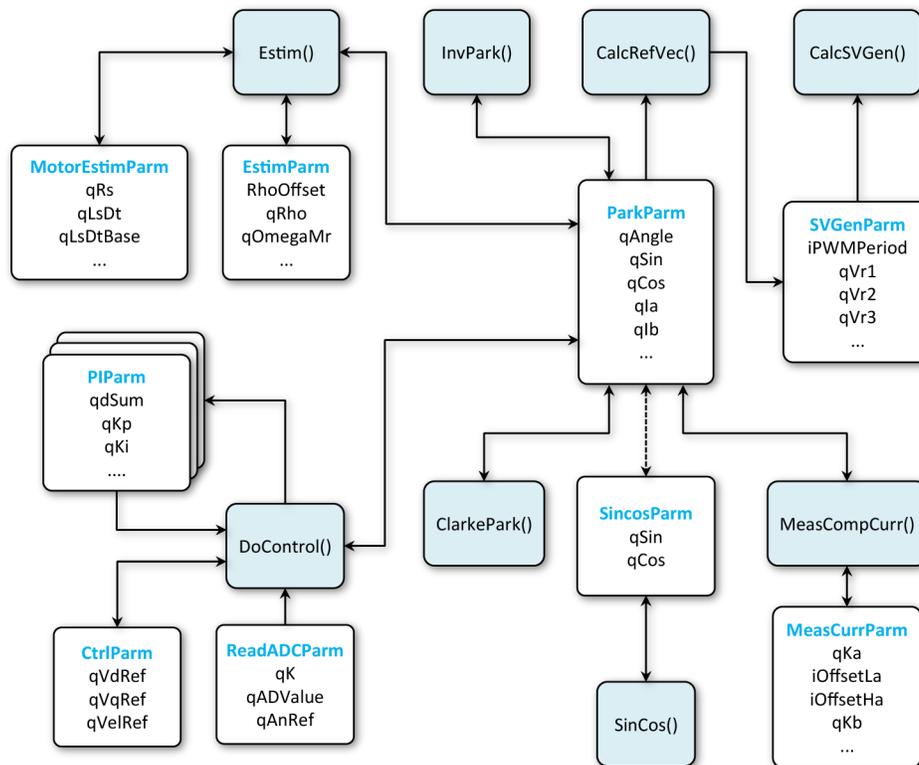


图8: AN1292软件中核心数据结构和访问的架构

### 1.2.10.1.3 源代码和文件组织

除main()之外，**pmsm.c**文件还托管CalculateParkAngle()、DoControl()、InitControlParameters和ADC ISR。其余应用笔记函数由各个文件托管。

### 1.2.10.1.4 用户界面的状态逻辑处理

处理用户界面操作（例如开/关电机以及使电机速度加倍）的控制逻辑均使用软件位标志来实现，并且在main()中的阻塞while循环内进行控制。

### 1.2.10.1.5 添加/删除/禁止模块

传统AN1292软件的模块化程度不够，无法在不破坏代码的情况下添加/删除/禁止估算器、启动程序和SVM程序等核心模块。

## 1.2.10.2 MC应用程序框架中的代码架构

有关MC应用程序框架的架构的更多详细信息，请参见第3节“架构”。

### 1.2.10.2.1 数据结构和数据访问

MC应用程序框架使用分层方法将数据存储在各个结构中，然后使用更小的独立数据结构来专门托管函数特定的变量。例如，

- **idqCmd**保存为 $I_d$ 和 $I_q$ 提供参考值的变量
- **vabc**保存提供所需相电压 $V_A$ 、 $V_B$ 和 $V_C$ 的变量
- **estimator**保存与位置和速度估算器程序相关的变量。

与特定电机的控制和功能相关的各个数据结构被分组在一个高级结构下。如图9所示，对于单电机系统来说，此高级结构就是motor结构。同样，用于系统级任务的所有更小的各个结构都维护在一个systemData结构下。

Name	Type	Name	Type
motor	MCAPP_MOTOR_DATA	systemData	MCAPP_SYSTEM_DATA
idqCmdRaw	MC_DQ_T	vDC	int16_t
d	int16_t	vDCTarget	int16_t
q	int16_t	fieldWeakeningParar	MCAPP_FIELD_WEAK_PARAM
idqCmd	MC_DQ_T	onSpeed	int16_t
iabc	MC_ABC_T	curve	int[16]
ialphabeta	MC_ALPHABETA_T	debugCounters	tagDebugCounters
idq	MC_DQ_T	reset	uint16_t
idCtrl	MCAPP_PISTATE_NEW_T	stop	uint16_t
iqCtrl	MCAPP_PISTATE_NEW_T	testing	MCAPP_SYSTEM_TEST_MANAGER
vdqCmd	MC_DQ_T	guard	tagGuard
vdq	MC_DQ_T		
valphabeta	MC_ALPHABETA_T		
vabc	MC_ABC_T		
pwmDutycycle	MC_DUTYCYCLEOUT_T		
thetaElectrical	int16_t		
omegaElectrical	int16_t		
sincos	MC_SINCOS_T		
estimator	MCAPP_ESTIMATOR_T		
smo	MCAPP_SMO_ESTIMATOR_T		
pll	MCAPP_PLL_ESTIMATOR_T		
deltaT	int16_t		
rho	int16_t		
rhoStateVar	int32_t		
omegaMr	int16_t		
esalphabeta	MC_ALPHABETA_T		
esdq	MC_DQ_T		
diCounter	int16_t		

图9：MC应用程序框架中的数据结构层级

图9所示的详细信息仅供参考，可能会有变动。有关变量名称和数据结构组织的更准确详细信息，请参见最新的软件文档。

#### 1.2.10.2.2 硬件访问

在MC应用程序框架中对硬件（即dsPIC SFR）的所有访问均通过硬件抽象层（Hardware Abstraction Layer, HAL）进行。

#### 1.2.10.2.3 源代码和文件组织

MC应用程序框架使用各个模块（= 一个\*.c文件 + 一个\*.h文件）来托管函数及其相关的支持元素（如常量、类型定义和函数声明）。

#### 1.2.10.2.4 用户界面的状态逻辑处理

MC应用程序框架使用状态机方法来实现与用户界面相关的逻辑。

#### 1.2.10.2.5 添加/删除/禁止模块

MC应用程序框架旨在允许将其大多数单独的模块/功能禁止或替换成其他变型，前提是这些模块/功能共用相同的界面。例如，MC应用程序框架支持估算器的多种变型——每种变型都可以单独选择、禁止或替换成其他变型。

### 1.2.10.3 MC应用程序框架的显著优点

与传统AN1292相比，MC应用程序框架具有以下优点

1. MC应用程序框架更易于阅读和理解，这要归因于下列特性：
  - a. 标准化的编码和代码注释指南
  - b. 高度模块化的架构
  - c. 一致的函数/变量/常量命名体系
2. MC应用程序框架更便于最终用户进行修改和维护
3. 硬件抽象层允许轻松移植到客户硬件

### 1.2.10.4 MC应用程序框架的显著缺点

MC应用程序框架为最终用户引入了与传统AN1292软件截然不同的软件架构。这一变化对于已经熟悉AN1292软件的最终用户而言，意味着最初使用时需要投入额外的时间和精力学习一番。

### 1.2.11 参数计算

应用笔记源代码项目通常由两个重要部分组成：

1. 实现电机控制算法的实际源代码
2. 在头文件中定义为常量的算法配置参数，有助于算法源代码根据用户设置的特定系统条件进行调整。

例如，实现电机启动算法的源代码需要获知以下参数：

1. 电机的极数
2. 电机锁定时间
3. 启动结束速度
4. 电机启动标称电流

这些参数通常在软件复位时使用头文件中的常量进行初始化。这些参数常量可能会从一组计算中得出。在这种情况下，有两种选择：

1. 提前执行参数计算并将最终整数分配给相应的参数常量

例如：`#define KIP 5584 // Q15(0.17041)`

2. 使用C预处理程序在编译时执行参数计算

例如：`#define KIP Q15(0.17041) // Q15() is another pre-processor macro`

#### 1.2.11.1 传统AN1292软件如何处理参数计算？

传统AN1292软件使用C预处理程序来执行 `userparams.h` 文件中的大多数参数计算。

### 1.2.11.2 MC应用程序框架/motorBench™开发套件如何处理参数计算？

motorBench™开发套件计算参数值并使用模板引擎将含有最终整数值的参数常量呈现到各种头文件中，这些头文件位于MC应用程序框架中的`parameters`文件夹。除此之外，它还将包含实际值作为行注释。

### 1.2.11.3 采用motorBench™开发套件/MC应用程序框架方法的优点

由于参数的有效值通过motorBench™开发套件直接呈现到各种参数头文件中，因此不存在编译时发生整数溢出错误或不当类型转换错误的风险。

### 1.2.11.4 采用motorBench™开发套件/MC应用程序框架方法的缺点

在某些情况下，最终用户可能很难确定在motorBench™开发套件中如何得出参数常量的最终数字。包含实际计算的模板头文件中的行注释在很大程度上有助于缓解这种担忧。

## 1.2.12 统一的源代码库

我们目前在网络上为基于磁场定向控制（Field Oriented Control，FOC）的应用笔记提供的大部分源代码都使用彼此通用的源代码组件。即使是基于非FOC的应用笔记，也可以与基于FOC的应用笔记共用它们的一些源代码组件。原则上，这种通用性可为我们提供*统一的源代码库*，应用笔记的每个变型都可以从中获得其源代码。

### 1.2.12.1 传统应用笔记软件

我们的传统应用笔记软件为PIM和电机控制板的每种硬件组合提供了一个源代码项目副本，如图10所示。

**AN1292****Title:** Sensorless Field Oriented Control (FOC) for a Permanent Magnet Synchronous Motor (PMSM) Using a PLL Estimator and Field Weakening (FW)**Name:** AN1292**Date:** 06/30/2011**Author:** Mihai Cheles**Description:** This application note describes a sensorless FOC algorithm for PMSM motors using a Phase-Locked Loop (PLL) position and speed estimator.**Keywords:** Sensorless, FOC, dsPIC, DSC, Digital Signal Control, DSP, 16-bit, Sensorless, BLDC, PMSM, Brushless, PLL, Estimator, Vector Control, FW, Field Weakening, Flux Weakening

Application Notes & Source Code	Last Updated	Size	
AN1292	06/30/2011	440KB	
AN1292 Source Code for DV330100 using dsPIC33EP512GM710	12/22/2014	2376KB	
AN1292 Source Code for dsPIC33EP512MU810 and dsPICDEM MCHV	11/07/2012	1978KB	
AN1292 Source Code for dsPIC33EP512MU810 and dsPICDEM MCLV	11/07/2012	1584KB	
AN1292 Source Code for MCLV-2 using dsPIC33EP256MC506 External OpAmp PIM	08/29/2012	1479KB	
AN1292 Source Code for MCLV-2 using dsPIC33EP256MC506 Internal OpAmp PIM	08/29/2012	1569KB	
AN1292 Source Code for MCHV-2 using dsPIC33EP256MC506 External OpAmp PIM	08/29/2012	1496KB	
AN1292 Source Code for MCHV-2 using dsPIC33EP256MC506 Internal OpAmp PIM	08/29/2012	1684KB	
AN1292 Source Code for DV330100 using dsPIC33EV256GM106	02/12/2015	2298KB	
AN1292 Source Code for dsPIC33FJ32MC204 and dsPICDEM MCLV	12/15/2009	835KB	
AN1292 Source Code for dsPIC33EP64MC504 and dsPICDEM MCLV	11/10/2011	1505KB	

图10: 含有多个源代码项目副本的AN1292的应用笔记网页

**1.2.12.2 MC应用程序框架/motorBench™开发套件中的统一代码库方法**

motorBench™开发套件为所有需要支持一组应用笔记和软件功能的源代码模块维护单一副本。然后根据统一代码库中特定的模块选择，结合硬件抽象层（HAL）文件和由模板过程生成的参数头文件，呈现最终系统特定的代码。图11所示为此呈现过程的两个示例。

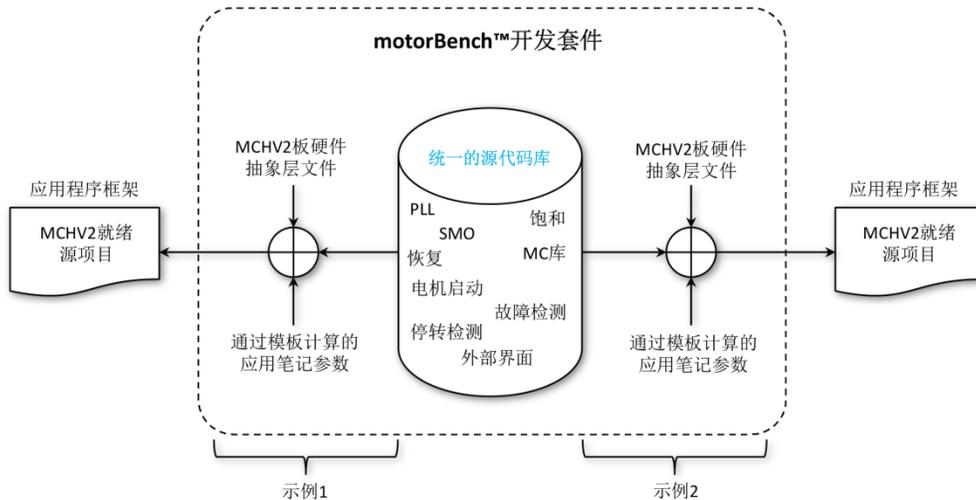


图11: 通过motorBench™开发套件呈现系统特定代码的示例

### 1.2.12.3 统一代码库方法的显著优点

统一代码库方法可以高效地维护源代码。对统一代码库的缺陷修复会自动传播到所呈现输出的所有变型。

### 1.2.12.4 统一代码库方法的显著缺点

最终用户将必须搭配motorBench™开发套件（或备用模板脚本）才能使用统一代码库方法。

## 1.2.13 诊断和测试界面

理想情况下，应用笔记软件必须为应用程序开发人员提供调试界面，以便实时探测/写入变量、检查系统状态和记录变量，而无需使用时间和资源密集型调试工具。

### 1.2.13.1 传统AN1292软件中的诊断和测试界面

传统AN1292软件提供以下界面功能以帮助应用程序开发人员进行测试、调试和诊断：

1. RTDM/DMCI，用于读取和写入变量，绘制最多四个变量（数据吞吐量有限）
2. 基本测试模式，有助于实现调整过程：
  - a. 软件斜坡——使用`userparams.h`文件中的宏TUNING
  - b. 开环模式——使用`userparams.h`文件中的宏OPEN\_LOOP\_FUNCTIONING
  - c. 转矩控制模式——使用`userparams.h`文件中的宏TORQUE\_MODE

### 1.2.13.2 MC应用程序框架中的诊断和测试界面

MC应用程序框架提供以下界面功能，以帮助应用程序开发人员进行测试、调试和诊断：

1. 用户界面LED和按钮，用于传达有关状态和故障的信息

2. 林茨机电一体化中心的[X2C-Scope](#)
3. 具有测试保护的软件测试模式；支持
  - a. 多种操作模式——可用于增生测试（一次一个阶段）
  - b. 手动换相频率改写
  - c. 给定速度、电流或电压的方波扰动
  - d. 用给定固定速度改写速度输入

更多详细信息，请参见MC应用程序框架文档的外部界面部分。

### 1.2.13.3 显著优点

MC应用程序框架为应用程序开发人员提供了更多用于诊断和测试的功能。

### 1.2.13.4 显著缺点

MC应用程序框架支持的一些诊断和测试功能需要专门的PC软件工具，这些工具尚未公开发布。

## 1.2.14 饱和和处理

所有实际电机控制系统可承受的最大/最小电压和电流都有所限制。因此，应用笔记对速度和电流控制器的最大/最小输出值有限制。在速度的参考值和测量值导致输出电流超出给定系统有效范围的情况下，会发生电流饱和。同样，如果电流的参考值和测量值导致输出电压超出给定系统的有效范围，则会发生电压饱和。

### 1.2.14.1 传统AN1292软件中的饱和和处理

在传统AN1292软件中，速度控制器设置为在0x5000处饱和（见`userparams.h`文件中的宏SPEEDCNTR\_OUTMAX）， $I_Q$ 的参考电流限制为通过公式(1)计算的 $I_{Qsat}$ 。

$$I_{Qsat} = 0x5000 \times \frac{I_{peak}}{2^{15}} \dots\dots\dots (2)$$

其中，

$I_{peak}$ 是ADC可测量的峰值电流

对于峰值电流能力为4.4A的MCLV2开发板，基于公式(2)计算出的电机电流 $I_{Qsat}$ 的最大值约为 $2.75A_{peak} = 1.94A_{RMS}$ 。

电流控制器设置为在最大Q15值（即32767）处饱和。这对应于逆变器上可用的最大电压。

一旦达到饱和状态，速度和电流控制器都将其超限增益（ $K_c$ ）设置为最大值。因此，积分项将立即归零，而不会出现饱和。

另外，如果两个控制器（即速度和电流）中的一个或两个进入饱和状态，则二者不进行通信并且表现出不同的行为。

### 1.2.14.2 MC应用程序框架中的饱和和处理

MC应用程序框架具有更复杂的饱和和处理系统，支持以下功能：

1. 电流和速度控制器中的一个或两个处于饱和状态时，二者相互通信并且作出不同的反应
2. 电流和速度控制器使用饱和窗口而不是单个阈值
3. 电流和速度控制器中使用的饱和窗口限值是基于电机、电机控制板和电源规范推导得出。

更多详细信息，请参见文档第1.0节“饱和”。

#### **1.2.14.3 MC应用程序框架方法的显著优点**

使用多个PI控制器协同合作的方法实现速度和电流控制环有助于使电机角估计值保持在安全运行边界内，从而提高效率。

将饱和窗口限值与电机、电机控制板和电源规范相关联可防止由于过载/过热而导致硬件损坏。

#### **1.2.14.4 显著缺点**

更新新电机/控制板的PI控制器饱和限值需要使用motorBench™开发套件。

### 1.3 版本历史

版本	日期	作者	说明
1	2017年2月24日	Srikar Deshmukh - C14317	本文档的第一个版本