

采用 MC56F8006 的 BLDC 无传感器 参考设计

支持的器件：
MC56F8006

Document Number: DRM108
Rev. 0
04/2009

第 1 章 简介

1.1	简介	1-5
1.1.1	应用特性和组件	1-6
1.2	Freescale 控制器优势和特性	1-7
1.3	前页	1-9
1.3.1	参考文献	1-9
1.3.2	首字母缩略词和缩写词	1-9
1.3.3	术语词汇表	1-10

第 2 章 控制理论

2.1	无刷直流电机 (BLDC 电机)	2-11
2.2	无传感器控制基础知识和要点	2-13
2.2.1	功率平台—电机系统模型	2-13
2.2.2	反电动势过零检测	2-14
2.3	BLDC 驱动状态	2-16
2.3.1	对齐	2-16
2.3.2	启动	2-17
2.3.3	运行	2-17

第 3 章 系统概念

3.1	系统规范	3-19
3.2	无传感器驱动概念	3-20

第 4 章 硬件

4.1	硬件实现	4-21
4.2	组件介绍	4-22

第 5 章 软件设计

5.1	简介	5-23
5.2	主要软件流程图	5-23
5.3	数据流	5-25
5.4	进程描述	5-26
5.4.1	过零检测进程	5-26
5.4.2	过零电流和速度计算进程	5-26
5.4.3	低速检测	5-26
5.4.4	换相进程	5-26
5.4.5	手动接口检测进程	5-26
5.4.6	应用状态机进程	5-27
5.4.7	速度和电流 PI 控制器进程	5-28
5.4.8	RAMP 进程	5-28
5.4.9	FreeMASTER 进程	5-28
5.4.10	PWM 更新进程	5-28
5.5	应用变量和常量定标	5-29
5.5.1	小数表示	5-29
5.5.2	模拟量调整	5-29

5.6	算法实现	5-31
5.6.1	ADC 采样机制	5-31
5.6.2	反电动势过零检测	5-32
5.7	FreeMASTER 软件	5-34
5.7.1	FreeMASTER 串行通信驱动程序	5-35
5.7.2	FreeMASTER 记录器	5-36
5.7.3	FreeMASTER 控制页面	5-36
5.8	特定电机的软件参数设置	5-37

第 6 章 应用设置

6.1	硬件设置	6-40
6.2	软件设置	6-41

第 7 章 结果和测量

7.1	结论	7-46
-----	----------	------

第 1 章 简介

1.1 简介

本设计参考手册介绍基于 Freescale MC56F8006 专用电机控制器件的无传感器 3 相无刷直流 (BLDC) 电机驱动的设计。

BLDC 电机广泛应用于许多领域。BLDC 电机没有换向器，因此，比直流电机更加可靠。与交流感应电机相比，BLDC 电机同样具有优势。BLDC 电机通过转子磁体生成转子磁通，具有更高的效率，因此，它可用于高端白色家电（例如冰箱、洗衣机、洗碗机）、高端泵、风扇以及其他需要高可靠性和效率的家电。

其应用原理为使用无传感器过零技术实现速度和电流控制闭环 BLDC 驱动。要控制 BLDC 电机，则必须知道转子位置所在角度才能让所施加的电压与反电动势（定子绕组中由于转子永磁体运动而感应产生）对齐。

虽然部分 BLDC 使用传感器进行位置感测，但如今的趋势则是使用无传感器控制。因此，位置通过电压或流入电机的电流来估算。其中一项无传感器技术就是通过反电动势过零技术进行无传感器 BLDC 控制。

这种控制方式的优势如下：

- 节省位置传感器和接线成本。
- 使用范围广泛，尤其是对于无法连接传感器或用传感器代价比较高昂的时候。
- 低成本系统（控制 DSP 功率要求中等）。

它可作为采用 Freescale MC56F8006 数字信号控制器的 BLDC 电机控制设计示例。本参考设计包括基本电机理论、系统设计理念、硬件实现和软件设计，其中包括 FreeMASTER 软件可视化工具。

1.1.1 应用特性和组件

该系统设计用于驱动 3 相 BLDC 电机。应用特性：

- 3 相无传感器 BLDC 电机控制。
- 采用多次采样的过零点检测技术。
- 单极性 PWM 调制技术、自动控制速度和限流。
- 基于 Freescale MC56F8006 控制器。
- 运行于带控制器的 3 相 BLDC/PMSM 低压电机控制驱动板及相关子板上。
- FreeMASTER 软件控制接口和监控器。

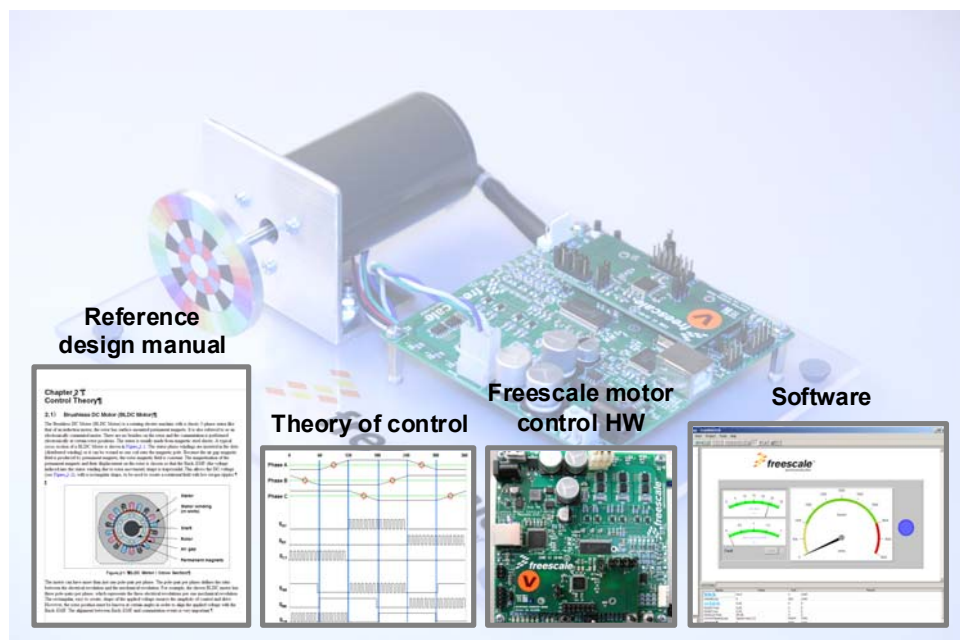


图 1-1. 采用 MC56F8006 的 BLDC 无传感器应用参考设计

客户可用的主要应用组件如下所示：

- 软件 — 基于 56800E 系列数字信号控制器的汇编代码和 C 代码。
- 硬件 — 基于带控制器的 Freescale 通用低压功率平台和子板。
- 文档 — 本文档。

1.2 Freescale 控制器优势和特性

56F8002/56F8006 属于基于 56800E 内核的数字信号控制器 (DSC) 系列。它通过一组可灵活选择的外设将 DSP 功率处理和微控制器功能集成于一个芯片上，构成一个具有极高性价比的解决方案。56F8002/56F8006 成本低、配置灵活、程序代码紧凑，因而适合许多应用领域。

56F8002/56F8006 具有以下特性：

- 内核频率为 32 MHz 时，高达 32 MIPS
- DSP 和 MCU 功能集成于一个统一的架构中。
- 程序 Flash 56F8002: 12 KB ($6\text{ K} \times 16$) 56F8006: 16 KB ($8\text{ K} \times 16$).
- 2 KB ($2\text{ K} \times 16$) 单元的数据 / 程序 RAM。
- 一个带 PWM 输出的 6 通道 PWM 模块、故障输入、带死区时间插入的容错设计、支持中间对齐和边缘对齐模式。
- 两个带单次或持续转换功能的 14 通道 12 位模数转换器 (ADC)、可配置采样时间和转换速度/功率、转换完成标志和中断、可从四种源选择输入时钟、可选异步硬件转换触发器、温度传感器等等。
- 两个增益高达 32 倍的可编程增益放大器 (PGA)
- 三个模拟比较器。
- 一个可编程间隔定时器 (PIT)。
- 一个带 LIN 从机功能的串行通信接口 (SCI)
- 1 个串行外设接口 (SPI)。
- 一个 16 位双路定时器 (2×16 位定时器)。
- 一个可编程延迟块 (PDB)。
- 一个内置集成电路 (I^2C) 端口。
- 一个实时计数器 (RTC)。
- 计算机正确操作 (COP)/ 看门狗。
- 两个片上振荡器 — 1 MHz 和 8 MHz。
- 晶体振荡器。
- 集成式上电复位 (POR) 和低压复位
- 中断 (LVI) 模块。
- JTAG/ 增强的片上仿真 (OnCE™)，用于非干扰性实时调试。

带单极性调制和反电动势多次采样功能的无传感器 BLDC 控制在极大程度上获益于灵活的 PWM 模块、快速 ADC 和双路定时器模块。

PWM 配置灵活，能够实现高效的 3 相电机控制。生成的 PWM 重新加载同步信号可以与其他模块（双路定时器、ADC、PDB）同步。

该应用在软件单次模式下使用 ADC 模块。转换启动以及 ADC 通道的选择和存储均在双路定时器快速中断程序下执行，并且这会在内部与 PWM 重新加载同步信号同步。ADC 用于估算反电动势过零检测（无需任何外部比较器）和检测 BLDC 电机控制所需的其他模拟量。

双路定时器是一个十分灵活的模块，它可以提供所需的全部时间事件相关服务。它具有以下特性：

- 两个 16 位计数器 / 定时器。
- 递增 / 递减计数。
- 计数器可级联。
- 可编程计数模块。
- 最大计数速率等于外设时钟速率 / 2（为外部事件计数时）。
- 最大计数速率等于外设时钟速率 / 1（使用内部时钟时）。
- 一次或重复计数。
- 计数器可预加载。
- 计数器可共用可用的输入引脚。
- 每个计数器均具有一个单独的预分频器。
- 每个计数器均具有捕捉和比较功能。

该应用使用双路定时器的两个通道：

- 第一个通道用于多采样操作和在线过零估算。
- 第二个通道用于完成换向事件。

1.3 前页

1.3.1 参考文献

1. *56F8006 数据手册 - MC56F8006* Freescale Semiconductor, 2008
2. *Design of Brushless Permanent-magnet Motors*, J.R. Hendershot JR and T.J.E. Miller, Magna Physics Publishing and Clarendon Press, 1994
3. *Sensorless Vector and Direct Torque Control*, P. Vas (1998), Oxford University Press, ISBN 0-19-856465-1, New York.
4. *Sensorless BLDC Motor Control Using the MC68HC908MR32*, DRM028/D, Motorola 2003
5. *3-phase BLDC Motor Control with Sensorless Back-EMF ADC Zero Crossing Detection using DSP 56F80x*, AN1913, Freescale 2005
6. *CodeWarrior™ Development Studio for Freescale™ 56800/E Digital Signal Controllers*, Freescale Semiconductor, 2006
7. *3-Phase BLDC Motor Sensorless Control Using MC56F8013*, DRM70, Freescale 2005
8. *Free Master Software Users Manual*, Freescale Semiconductor, 2004

如需现行文档列表, 请访问 www.freescale.com。

1.3.2 首字母缩略词和缩写词

表 1-1 包含本文档中使用的首字母缩略词和缩写词示例。

表 1-1. 首字母缩略词和缩写词

术语	含义
ADC	模数转换器。
EMF	电动势。
BLDC	无刷直流电机。
DC	直流。
DSC	数字信号控制器。
GPIO	通用输入 / 输出。
I/O	计算机系统和外部世界之间的输入 / 输出接口。CPU 读取输入以感测外部信号水平, 并写入到输出以更改外部信号水平。
JTAG	联合测试行为组织的英文首字母缩写。指允许进行片上仿真和编程的接口。
LED	发光二极管。
MC56F80x	专用于电机控制的 Freescale 16 位 DSC 系列。
PI 控制器	比例积分控制器。
PWM	脉宽调制。
RPM	每分钟的转数。
SCI	串行通信接口模块: 支持异步通信的模块。

1.3.3 术语词汇表

表 1-2 介绍本文档中使用的术语词汇。

表 1-2. 术语表

术语	定义
电刷	用于将电能从定子上的非旋转端子传输至转子的部件。
换向器	用于调整直流换向器电机中的直流电流并使换向器电机旋转的机械器件。
占空比	信号通电时间与信号关断时间之比。占空比通常为百分比。
霍尔传感器	用于为每个电气循环（3 相电机）提供六个定义事件（每 60 电角度）的位置传感器。
中断	在循序执行某一程序时，通过执行子程序而产生的临时中止，以响应来自外设器件的信号。
PI 控制器	比例积分控制器。
快速启动	软件算法和驱动程序集，包括用于 DSC/CPU 初始化的图形配置工具。
复位	用于强制器件进入已知状态。
软件	用于控制微控制器工作的指令和数据。

第 2 章 控制理论

2.1 无刷直流电机（BLDC 电机）

无刷直流电机（BLDC 电机）是一种旋转电机，带有与感应电机类似的经典 3 相定子；转子表面贴装永磁体。它也称为电子换向电机。转子上没有电刷，换向在某些转子位置以电子方式完成。定子常用磁铁薄片制成。典型的 BLDC 电机横截面如图 2-1 所示。定子相绕组嵌入定子齿槽（分布绕组）中，或者做成绕在磁极上的一个线圈（集中绕组）。由于气隙磁场由永磁体产生，因此，转子磁场是恒定的。可以选择永磁体的磁化强度及其在转子上的分布，从而使反电动势（定子绕组中由于转子运动而感应的电压）的波形为梯形。这样就可以采用具有矩形波形的直流电压（参见图 2-2）来生成低转矩纹波的旋转磁场。

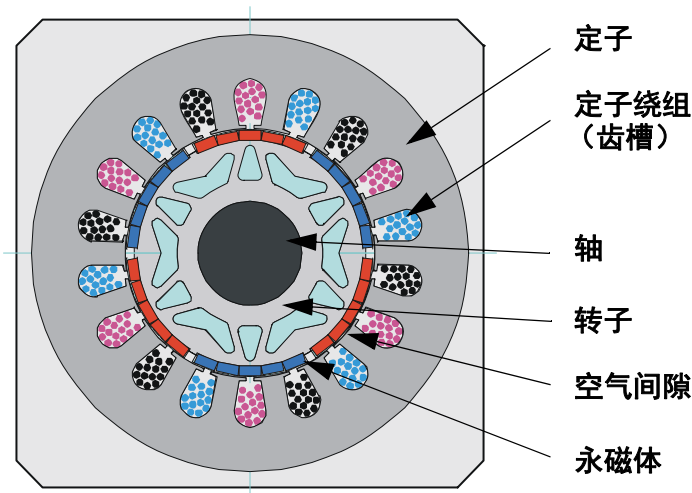


图 2-1. BLDC 电机 / 横截面

电机的每相可以有多对磁极。每相的磁极对数决定电气旋转与机械旋转之间的比率。例如，所示 BLDC 电机的每相具有三对磁极，表示三次电气旋转对应于一次机械旋转。所施加的矩形电压很容易产生，确保控制和驱动也很简单。但是，必须知道转子在特定时刻的角度位置，才能让所施加的电压与反电动势对齐。反电动势与换向动作是否对齐非常重要。

只有这样，电机才能像直流电机一样运行，且运行在最佳工作点。因此，控制和实现的简单使得 BLDC 电机成为低成本、高效率应用的最佳选择。

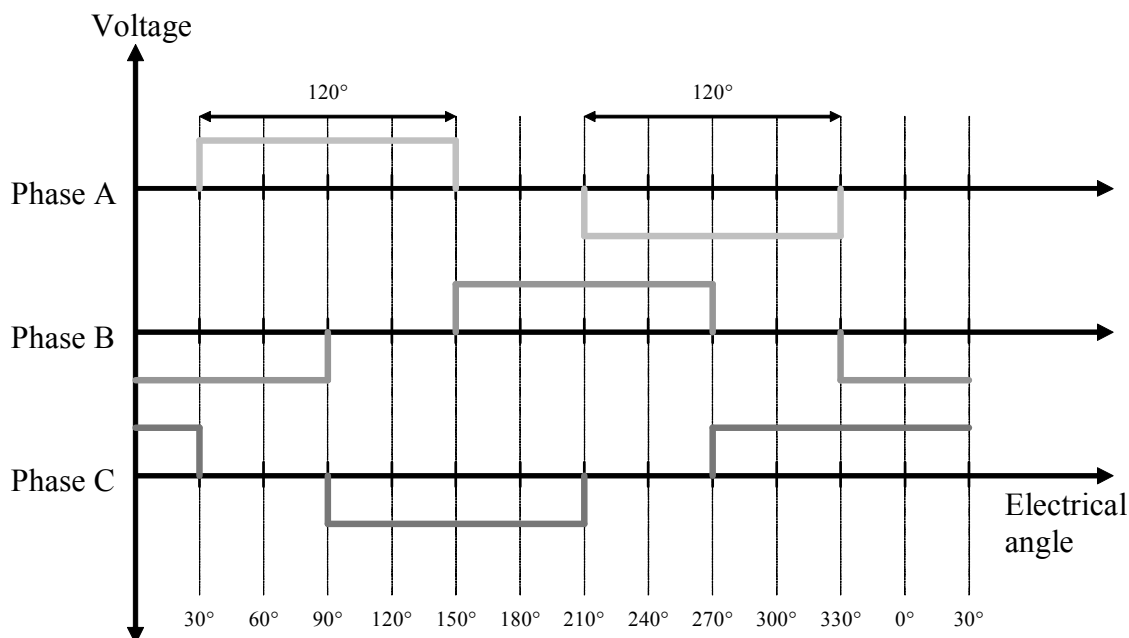


图 2-2. BLDC 电机的 3 相电压系统

BLDC 无传感器控制的主要任务就是位置估计。两种基本技术如下：

- 基于反电动势感应的技术：
 - 速度范围为额定转速的 5-10 % 至 100 %。
 - 反电动势必须足够高。
 - 基于反电动势过零的位置检测方法。
 - 其他高级反电动势估算技术：
 - 系统观测器。
 - 通过多次采样测量不导通相。
- 基于电机电感凸极的技术：
 - 速度范围为静止至标称速度的约 20%。
 - 技术：
 - 简单技术：基于瞬态电流测量。
 - 高级技术：需要强大的内核和精确的电流采样的电流注入方案。

可以将这两种方法相结合，但需要使用复杂的控制算法。在本应用中，采用了基于反电动势电压的检测，该检测是通过非导通相的多次采样实现的。什么是过零点，正确预估过零点的方法等将在下面章节中介绍。

2.2 无传感器控制基础知识和要点

本小节介绍无传感器控制的基础知识。包括功率平台介绍、BLDC 电机模型以及采用反电动势过零检测的原因及条件。

2.2.1 功率平台—电机系统模型

为解释和模拟反电动势传感技术的概念，本文档提供了基于基本电路拓扑的简化数学模型（请参见图 2-3）。3 相 BLDC 电机的电压通常由 IGBT 或 MOSFET 开关设计组成的 3 相功率平台提供。3 相功率平台由 DSC 片上 PWM 模块控制，它可创建所需的开关控制模式。此模型的作用在于了解开关管的导通和关断是如何影响电机特性的。开关角是实际开关事件与理想事件（此时，相位间的反电动势过零）之间的角度差。电机驱动模型由一个 3 相功率平台和一个无刷直流电机组成。系统由电压源 (U_d) 供电。六个半导体开关 ($S_A/B/C$ t/b) 施加矩形电压波形（请参见图 2-2）。半导体开关和二极管则模拟为理想器件。电机中性点电压为直流母线电压的一半。这可简化数学表达式。

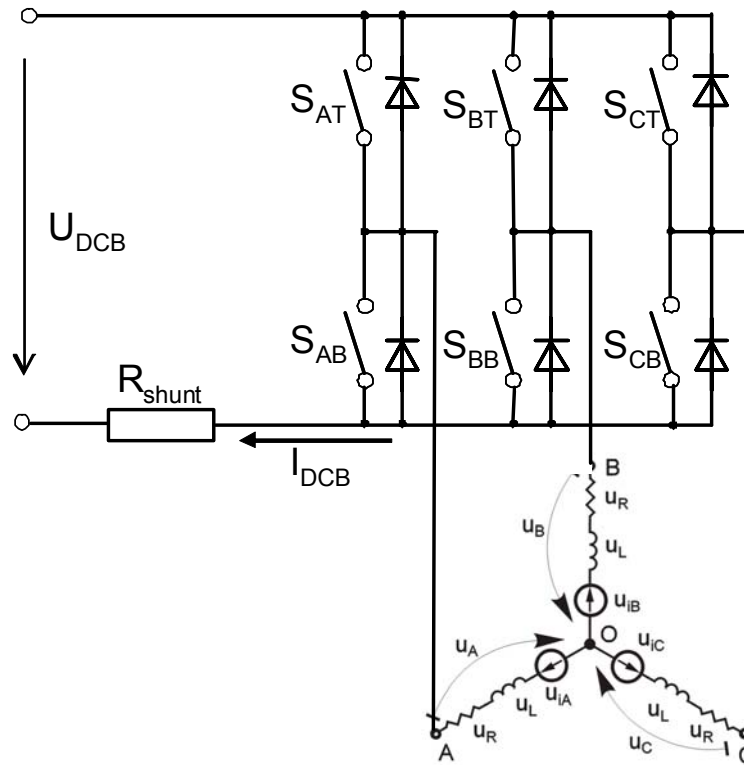


图 2-3. 功率平台和电机拓扑

2.2.2 反电动势过零检测

为了实现最有效的 BLDC 电机控制，导通相和非导通相要配合一致。这根据反电动势电压检测过零点进行换相实现。图 2-4 所示为正确换相的电机相绕组电压波形。从图中可得到正确的换相点应在两个过零点中间。因此，反电动势过零信号可以用作估计正确换相时间点的计算依据。

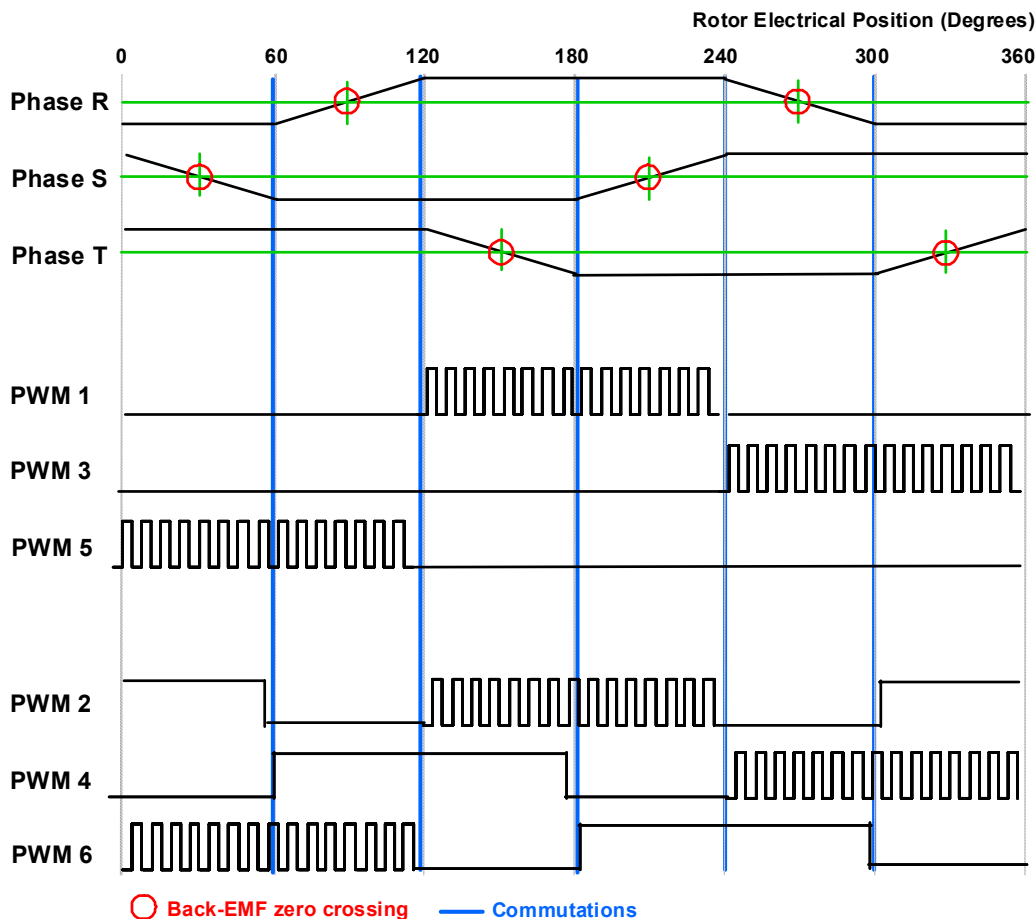


图 2-4. 反电动势过零、单极 PWM 切换、换相

图 2-4 中的相位信号为反电动势电压。它们为图 2-5 中的 U_{iX} 电压。

反电动势检测技术基于任一时刻只有两相通电这一情况（请参见图 2-5）。第三个非导通相用于感测反电动势电压。

不过也有一个问题，就是标准的 BLDC 电机仅有三个相位连接器且电机中性点无法引出。因此，如要获取过零时间点，需要将相电压与 $u_{DCB}/2$ 关联，如下节所述。

我们假定相位 A 和 B 通电，相位 C 未通电。无电流通过此相位。

可以满足以下条件：

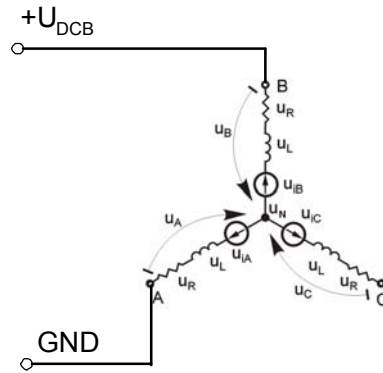


图 2-5. 反电动势感测环境

$$S_{Ab}, S_{Bt} \leftarrow \text{PWMswitching}$$

$$u_N = u_{DCB} - r i - L \frac{di}{dt} - u_{iB}$$

$$u_N = r i + L \frac{di}{dt} - u_{iA}$$

$$u_N = \frac{u_{DCB}}{2} - \frac{u_{iB} + u_{iA}}{2}$$

等式 2-1

电压 u_C 计算如下：

$$u_{iA} + u_{iB} + u_{iC} = 0$$

$$u_N = \frac{u_{DCB}}{2} + \frac{u_C}{2}$$

$$u_C = u_N + u_{iC}$$

等式 2-2

$$u_C = \frac{3}{2}u_{iC} + \frac{u_{DCB}}{2}$$

等式 2-3

电压 u_{iC} 在过零点为零，因此，得到下式：

$$u_C = \frac{u_{DCB}}{2}$$

等式 2-4

2.3 BLDC 驱动状态

为了启动并运行 BLDC 电机，控制算法必须经由以下状态：

- 对齐 (初始位置设置)
- 启动 (强制换相)
- 运行 (无传感器运行，提供反电动势 AD 采样和过零检测)

2.3.1 对齐

如前文所述，BLDC 电机无传感器控制的主要任务就是位置估计。但在启动电机之前，初始位置为未知。对齐的主要目的就是让转子对齐到已知位置。必须在这个已知位置才能以正确的方向启动旋转，并在启动过程中产生最大扭矩。在对齐过程中，所有三个相位均需通电。相位 A 连接至母线电压的正端，相位 B 和 C 连接至母线电压的负端。无论旋转方向如何，所产生的转子位置均会提供相同的启动条件（请参见图 2-6）。对齐时间取决于电机的机械常量，包括负载。时间必须确保在对齐过程结束时转子处于稳态。

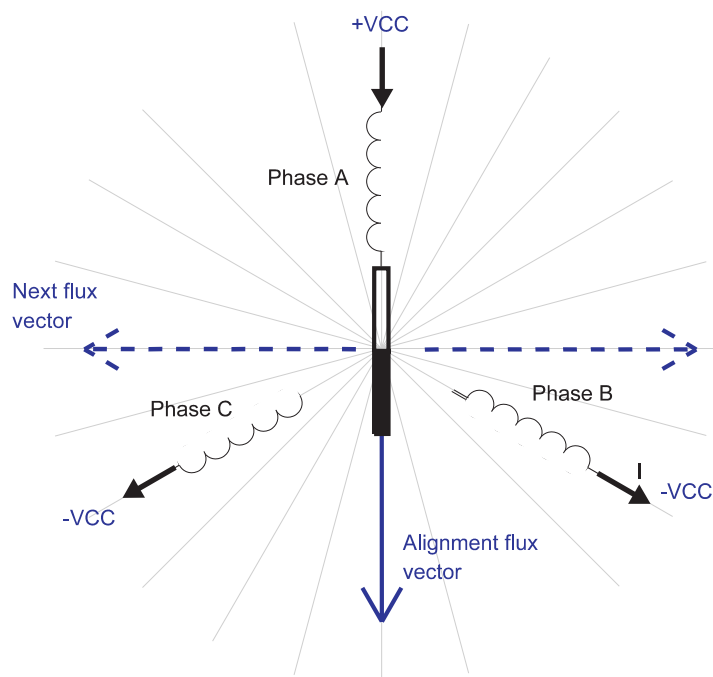


图 2-6. 转子对齐

2.3.2 启动

当转子运动时，可使用反电动势技术。启动流程可确保在转子初始运动之后进行反电动势估计。启动流程在对齐流程完成之后立即启动。应用于电机的第一个磁通向量与对齐磁通向量正交。驱动过程中，磁通向量所作用的时间为恒值。接着应用后五个磁通向量，以产生旋转磁场。用周期定时表来决定启动时间。必须根据负载惯性整定初始周期。

启动整定是反电动势无传感器技术中最困难的任务。整定难度取决于负载类型。对于无源负载，如果初始负载扭矩为 0 Nm 并随电机速度增长，则非常简单。更为复杂的情况是：启动扭矩为非零或者一次旋转中存在可变扭矩，例如压缩机。如果负载惯性可变，则可以更改启动磁通向量数。如果为可变负载惯性，则磁通向量数小于 6 产生的启动最稳健，但是，如果启动时的磁通向量数小于 3，则产生的启动不可靠。在其他情况下，6 个启动磁通向量（一个电周期）似乎最佳。执行完最后一个磁通向量后，可使能反电动势电压检测，电机进入运行（无传感器）模式。

2.3.3 运行

运行无传感器模式将反电动势采集和过零检测结合，以进行换相控制。在此模式下，换相时间根据过零反馈信号计算，如 2.2.2 “反电动势过零检测” 所述。

第 3 章 系统概念

3.1 系统规范

该参考设计用于驱动速度和电流闭环的 3 相无刷直流电机。该应用符合以下性能规范：

- 针对 MC56F8006 （或 56F8002）数字信号控制器。
- 针对 24V 的低压三相电机驱动系统。
- 控制技术包括：
 - 通过反电动势过零检测技术实现无传感器无刷直流电机控制。
 - 通过自动电流调整和限幅实现闭环速度控制。
 - 反电动势多次采样方法—PWM 接通期间采样周期为 6 μ s
 - 对齐、开环启动。
 - 完整的 4 象限运行。
 - 手动接口（运行 / 停止开关、加速 / 减速按钮控件）。
 - FreeMASTER 软件控制接口（电机运行 / 停止、速度 / 当前设置）。
 - FreeMASTER 软件远程监控。

3.2 无传感器驱动概念

系统框图如下所示。下面的框图简单介绍了基于无传感设计的系统的各个模块。

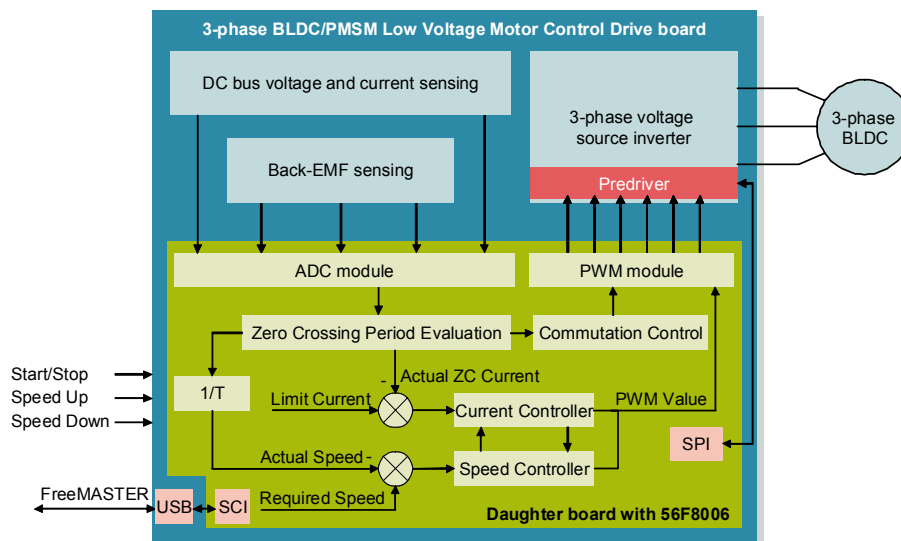


图 3-1. 系统概念

定期扫描用户接口状态，并由 ADC 对直流母线电流、电压和反电动势进行采样。使用 PWM 重新加载信号使 AD 采样与 PWM 同步。每个 PWM 周期内对直流母线电压采样一次。根据占空比值，在预定义周期 (6 μ s) 内对直流母线电流和反电动势采样。过零估算功能将经过处理的反电动势信号与直流母线电压值的一半对比。检测到过零点时，启动进行多项计算。计算所得的下一次换相时间为实际平均转速计算的源值。根据控制信号的状态、启动 / 停止以及所需的 FreeMASTER 速度设定速度命令。随后，通过速度 RAMP 算法对速度命令进行处理。从 RAMP 算法输出获得的实际速度命令与所测得的计算速度进行比较，得到一个速度误差。该速度误差是速度 PI 控制器的输入，从而生成新的 PWM 值。当平均过零电流达到预定义限值时，电流 PI 控制器接管调整任务，直到平均过零电流降至预定义的值以下。该应用可以通过主机 PC 的 FreeMASTER 控制页面控制，或者可以通过电机控制板上的手动接口开关轻松控制。FreeMASTER 通过串行 RS232/USB 协议进行通信。

第 4 章 硬件

4.1 硬件实现

无传感器的 BLDC 驱动运行于 Freescale 3 相 BLDC/PMSM 低压电机控制驱动板（带 MC56F8006 子板和 LINIX 45ZWN24-40 BLDC 电机）上。该应用的硬件系统配置如图 4-1 所示。

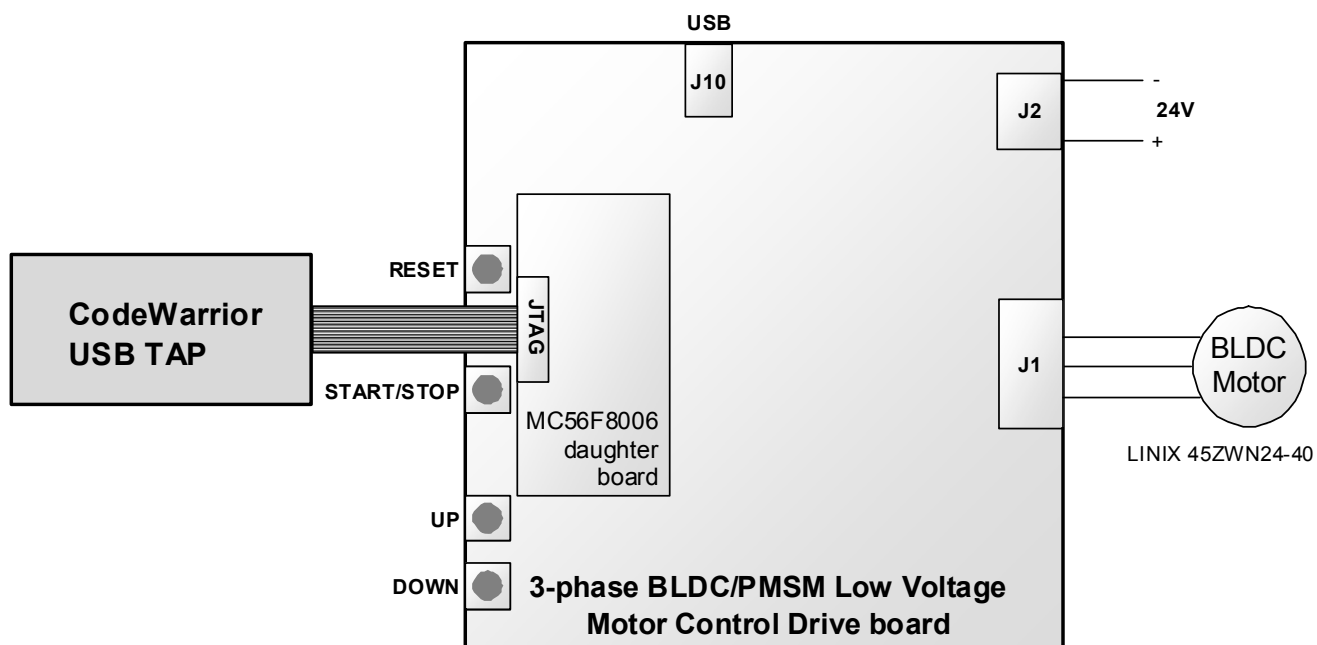


图 4-1. 硬件系统配置

4.2 组件介绍

MC56F8006 无传感器 BLDC demo 基于 3 相 BLDC/PMSM 低压电机控制驱动板模块，该模块可以为测试和开发带多个微控制器和 DSC 子板的低功耗驱动和演示提供广泛的解决方案（请参见图 4-2）。它演示了 MC56F8006 的功能，并提供有助于开发使用 MC56F8006（针对电机控制应用）的应用的硬件工具。详细的介绍，包括 3 相 BLDC/PMSM 低压电机控制驱动板的硬件规格，参见用户手册 LVMCDBLDCPMSMUG。本用户指南包含驱动板原理图、各个功能模块介绍以及材料清单。



图 4-2. 带子板的 3 相 BLDC/PMSM 低压电机控制驱动板

此应用中所使用的电机是标准的 BLDC 电机，其霍尔传感器安装在定子槽中，而不是在控制板上。该电机规格如下：

表 4-1. 电机规格

电机规格：	电机类型：	LINIX 45ZWN24-40 BLDC 电机
	额定母线电压	24 V RMS
	额定转速	4000 RPM
	额定相电流	2.34 A
	额定转矩	0.0924 Nm

第 5 章 软件设计

5.1 简介

本节介绍驱动的软件模块设计。软件介绍包括以下主题：

- 主要软件流程图
- 数据流

5.2 主要软件流程图

主要软件流程图包括复位后进入的主程序（后台循环）和中断事件。主程序包括微控制器和主循环的初始化。

初始化后，主程序进入无限循环，执行包括初始化、停止、对齐、启动、运行和错误状态的应用状态机。状态之间的转变取决于 START/ STOP 开关的状态和驱动状态。主要软件流程图如图 5-1 所示。除后台循环之外，软件还包括以下三种中断：

- 双路定时器通道 0 比较中断

中断优先级：2

执行周期：6 μ s

中断调用的函数：

— IsrAdcStartRead

快速中断例程，执行启动转换并读取 ADC 结果。测量的物理量：三个反电动势电压（通过两次采样求平均值程序）、直流母线电压和直流母线电流。此函数的下一个任务是 AD 通道的切换。第一次和最后一次采样识别。过零检测以及过零点至换相的时间计算。如果检测到过零点，则 *iDcb* 的实际值添加至 *iDcbSum*，*zcToCmt* 时间添加至 *periodSum* 且过零点计数器 *zcCnt* 递增用于下一次均值计算。

— IsrAdcBlock

快速中断例程，在换相之后执行，用于测量 20% 到 80% 母线电压范围内的反电动势值。

- 双路定时器通道 1 比较中断

中断优先级：2

执行周期：事件驱动

中断调用的函数：

— *IsrCommutationTimer*

此函数的主要任务是执行换相。当前扇区值加一，是接下来代码执行的基础。根据扇区，从换相表 *COMM_TABLE_MASK*、*COMM_TABLE_SWCTRL* 中选择相应的 MASK 值和软件控制模式，并将其应用到 PWM 模块。对于扇区值，将从表 *BEMF_CH_TABLE* 中选择新的反电动势测量相，并将其存储在变量 *bemfChannel* 中，用于中断函数 *IsrAdcStartRead* 和 *IsrAdcBlock* 中。中断函数 *IsrCommutationTimer* 也是以所选择的反电动势测量相为基础启动执行的。

- PIT 周期中断

中断优先级：0

执行周期：5 ms

中断调用的函数：

— *IsrPitPeriod*

此中断执行 PI 控制器的计算，速度斜坡的计算，速度命令开关的检查，速度和电流的平均滤波和根据 PWM value 值决定采样次数。

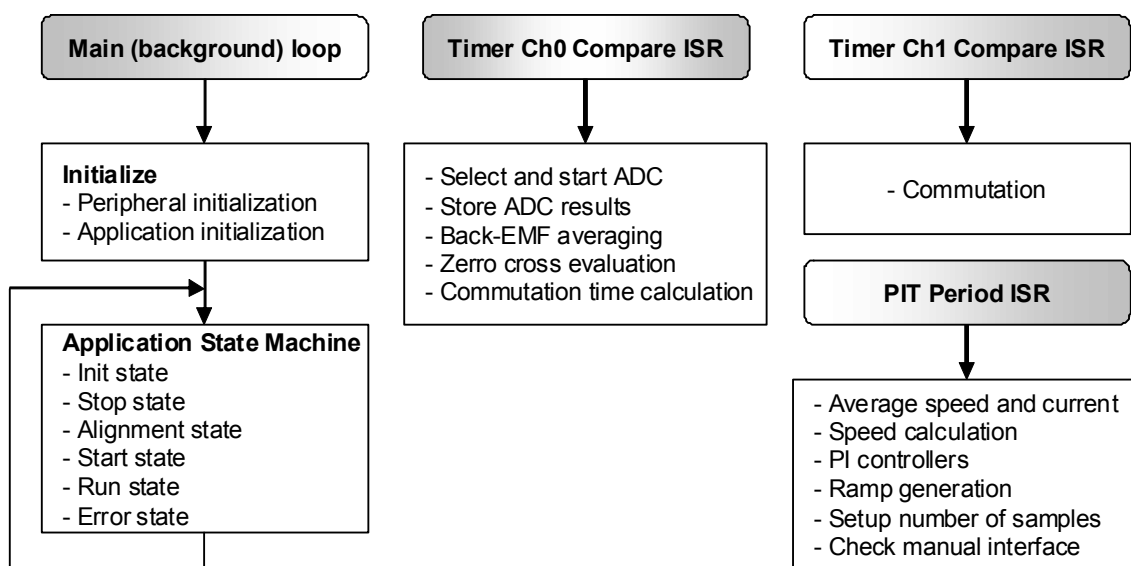


图 5-1. 主要软件流程图

5.3 数据流

图 5-2 为无传感器的无刷直流电机控制算法的数据流程图。*bemf*、*vDcbHalf* 和 *iDcb* 为代码执行过程中的 AD 采样值。过零检测的流程就是将相电压和直流母线电压的一半进行对比。相电压的选取以所在扇区的扇区值为基础。过零点的结果决定换相的时间差，保存在 *timeZcCmt* 中。此外，单个过零至换相时间值添加至 *periodSum*，用于进一步计算平均周期。同样，对于每个过零事件，*iDcb* 值均添加至 *iDcbSum*。实际转速为多个周期的平均值。同样，实际过零电流 (*iDcbZc*) 等于 *iDcbSum* 除以每个 PIT 周期的过零点数的平均值。这两个值为速度和电流 PI 控制器的实际输入值。客户可以根据实际需要调整所需的速度环和电流环的限幅值。速度斜率由 *ramp* 值限制，以免引起过流。速度环和电流环的输出根据转子磁链所在扇区更新 PWM 模块的占空比。

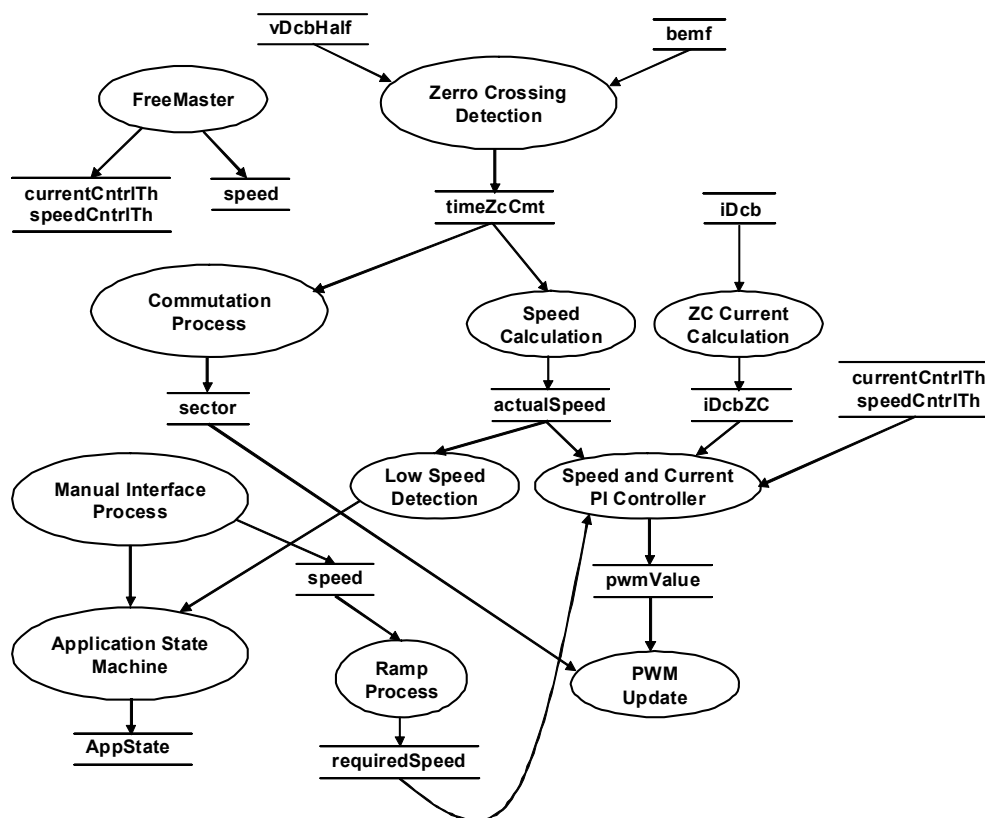


图 5-2. 数据流程图

为使流程图更加清晰，数据流程图中未提及其他变量，例如控制器参数设置、接口类型选择（手动 /FreeMASTER 或仅限 FreeMASTER）、PI 环的中间变量、系数调整等等。应用状态机根据手动接口、低速检测（静止）、FreeMASTER 接口以及主数据流程图中未提及的其他状态变量设置其状态 *AppState*。

5.4 进程描述

无刷直流电机无传感器驱动应用的主要进程如图 5-2 所示。

5.4.1 过零检测进程

当 PWM 导通时，每 6 μ s 调用一次该进程。此进程位于 *IsrAdcStartRead* 中断中。在此进程之前，对选定通道执行模数转换，并对所 AD 转换的数据结果进行基本的软件处理。将前一个反电动势 AD 值和当前的反电动势 AD 值进行平均。另一个重要的过零点估算值为直流母线电压的中点 *vDcbHalf*。实际上，*vDcbHalf* 并非完全等于此值的一半，它可用系数 *dcbHalfCoef* 来整定。随后对比平均 BEMF 和 *vDcbHalf*，并估算转换情况。此外，当发生过零事件时，还会通过 *bemf* 扫描直流母线电流并取其值。过零点处的直流母线电流值稳定性可以接受，因此，适用于控制进程。如果检测到过零点，则会计算新的时间事件 *timeZcCmt*，即从过零点至下一次换相的时间。随后将计算值添加至 *periodSum*，同样，*iDcb* 添加至 *iDcbSum*，过零点计数器 (*zcCnt*) 递增，用于将来的平均值计算。将换相定时器—双路定时器通道 1 中的新换相事件设置为新比较事件。

5.4.2 过零电流和速度计算进程

在多周期的时间内对电流求平均值，获得适合于电流 PI 控制器的输入值，随后，重新计算平均转速。该进程在 *IsrPitPeriod* 中断中每 5 ms 执行一次。

5.4.3 低速检测

该进程将重新算得的实际速度与最低速度比较。如果低于最低转速，则电机停止，然后重复启动。该进程在每 5ms 的 *IsrPitPeriod* 定时中断中执行。

5.4.4 换相进程

该进程更新扇区值。在 PWM 更新进程中，根据新的换相扇区执行 PWM 更新。每个新的换相时刻都会调用此换相进程。因此，执行周期取决于电机速度。

5.4.5 手动接口检测进程

该进程检查连接 START/STOP 开关和 UP/DOWN 按钮的输入引脚的状态。在 *IsrPitPeriod* 中断中每 5 ms 检测 UP/DOWN 按钮是否按下，在应用状态机中每 5 ms 处理一次 START/STOP 按钮。

5.4.6 应用状态机进程

应用状态机 (ASM) 包含初始化、停止、对齐、启动、运行和错误状态（请参见图 5-3）。MCU 复位后，ASM 从初始化状态进入停止状态。用户将 START/STOP 开关转至 START 位置后，ASM 进入对齐状态，此时电机对齐到已知位置。经过一段预定义时间后，ASM 继续进入启动状态。在启动状态下，将会执行六次换相，无需任何反馈。换相之间的时间由启动表定义。启动状态之后，ASM 进入运行状态，此时控制环路关闭。在运行状态下，PI 控制器可接受新速度和电流请求并维持相应的速度和电流。如果检测到转子静止（低速），则 ASM 进入错误状态。如果发生任何故障，则 ASM 进入错误状态，然后从初始化状态进入停止状态。ASM 此时将在后台循环中执行。实际工作状态在 FreeMASTER 控制页面显示。ASM 流程图如下图所示。

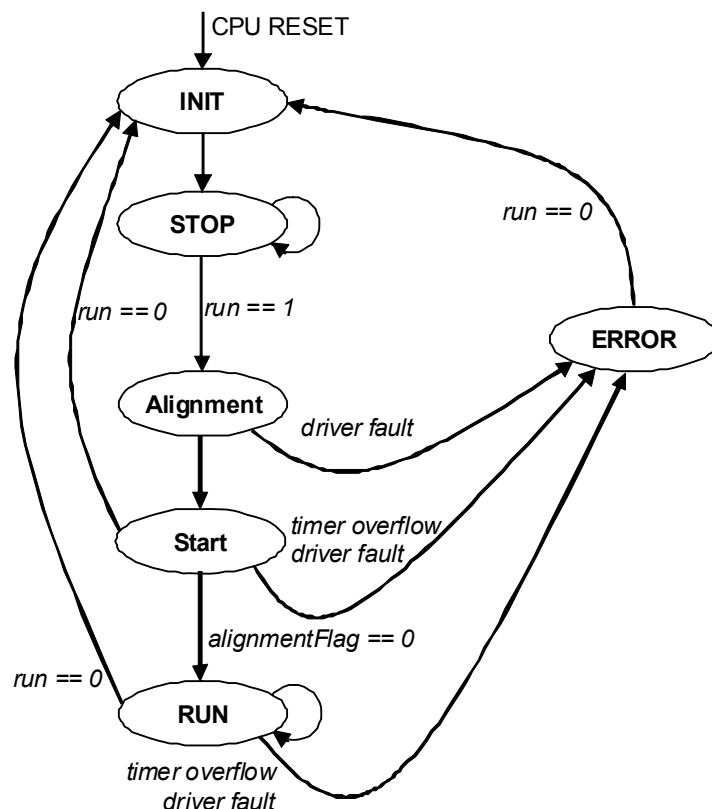


图 5-3. 应用状态机

5.4.7 速度和电流 PI 控制器进程

控制环路的一般原理如图 5-4 所示。

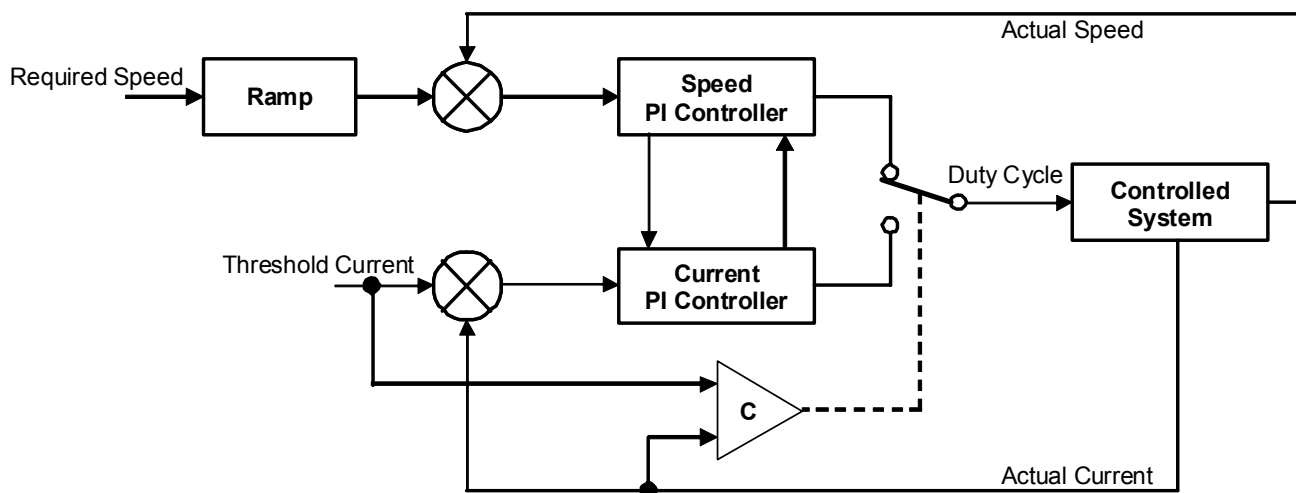


图 5-4. 闭环控制系统

速度和电流闭环控制通过实际电机速度和平均过零电流反馈来体现。该信息与参考设定点对比后，生成误差信号。误差信号的幅度和极性对应于实际和所需速度 / 电流之差。PI 控制器根据速度 / 电流误差生成校正后的电机电压，以补偿此误差。

此应用允许在具有自动电流限制的速度模式下工作。这意味着，在实际平均过零电流值达到阈值 *currentCntlTh* 且从电流 PI 控制器输出中获取占空比值之前，系统处于速度模式。在控制器之间切换时，内部变量 *f32IntegPartK_1* 在两个控制器之间传递。如果驱动处于电流限制模式且实际平均过零电流值达到阈值 *speedCntlTh*，则会开启速度模式。在图 5-4 中，阈值电流表示为单个值。这是为了简化图形表示，在实际软件应用中，它由上述两个变量表示。以经验值设定 PI 控制器的比例和积分常数。每 5 ms 在 *IsrPitPeriod* 中断中执行一次 PI 控制器。对于 Freescale 56F800E 系列数字信号控制器，可以通过通用函数库使用 PI 控制器函数。

5.4.8 RAMP 进程

RAMP 的作用为确保电机平稳运转，防止过流现象发生。RAMP 进程用于获取所需的电机速度。该进程在 *IsrPitPeriod* 中断中每 5 ms 执行一次。对于 Freescale 56F800E 系列数字信号控制器，可以通过通用函数库使用 RAMP 函数。

5.4.9 FreeMASTER 进程

FreeMASTER 进程不是应用软件的一部分。FreeMASTER 通信完全由 RS232 驱动实施，因此，无需其他软件支持即可提供远程控制。FreeMASTER 直接将新值写入具有特定变量的数据 RAM。

5.4.10 PWM 更新进程

该进程根据实际扇区和所需的输出电压在电机上生成正确的电压模式。需要根据实际扇区屏蔽和通过软件控制从换相表中选择正确的 PWM 输出。

5.5 应用变量和常量定标

5.5.1 小数表示

本应用使用小数表示除时间以外的所有实际量。N 位带符号的小数格式用 1. [N-1] 格式（1 个带符号位，N-1 个小数位）。带符号的小数 (SF) 数值范围如下所示：

$$-1.0 \leq SF \leq +1.0 - 2^{-(N-1)}$$

等式 5-1

对于字和长字带符号小数，最小负数可以表示为 -1.0，内部表示为 0x8000 和 0x80000000。对于字，最大正数为 0x7FFF 或 $1.0 - 2^{-15}$ 且对于长字，最大正数为 0x7FFFFFFF 或 $1.0 - 2^{-31}$ 。

5.5.2 模拟量调整

模拟量的最大可测量范围（如电压、电流和频率）取决于硬件。例如，demo 板直流母线电流测量范围为 -4 至 4 A，表示电压范围 0 至 3.3 V，转换后用数字 0 至 2^{15} 表示。因此，测量的结果必须调整到这个范围之内。直流母线电流以带符号的小数表示，范围为 -0.5 至 0.5。在此情况下，完整的带符号小数范围表示 -8 至 8 A 电流。这也用于 FreeMASTER 中，以正确表示实际值，无需复杂的重新换算。

部分头文件定义电流缩放比例及其应用，以实现以下常数计算。这些常数在预编译的时候计算。*REG_THRESHOLD_CURRENT* 将 1.5 A 电流表示为范围 *DCB_CURRENT_SCALE* 中的小数。

```
#define DCB_CURRENT_SCALE      8.0
#define REG_THRESHOLD_CURRENT  FRAC16(1.5/DCB_CURRENT_SCALE)
```

另一个源代码示例为计算直流母线电压的 20% 和 80%。第一步，以小数 20% 和 80% 表示定义的常数。

```
#define V_DCB_LOW          FRAC16(0.2)// low limit of BEMF window - 20%  
#define V_DCB_HI          FRAC16(0.8)// high limit of BEMF window - 80%
```

定义的常数用于计算，方法是使用其中一个内部函数乘以两个 16 位小数值，然后生成 16 位小数结果。

```
vDcbLowLimit=__mult_r(vDcb,V_DCB_LOW);  
vDcbHighLimit=__mult_r(vDcb,V_DCB_HI);
```

5.6 算法实现

5.6.1 ADC 采样机制

非导通相的相电压在 PWM 开关边沿处会受到干扰。因此，所示 BLDC 电机控制应用将反电动势过零检测与 PWM 同步。非导通相相电压稳定时，将在初始反电动势振荡后立即多重触发非导通相相电压的 A/D 转换。此时，以经验值设定第一个采样延迟时间，具体取决于电机和系统参数。在 PWM on 的周期内每隔 6 μ s 的时间多次采样。根据变量 *pwmValue*，采样次数由 *IsrPitperiod* 表决定。采样机制的部分详细信息如图 5-5 所述。ADC 采样在快速中断函数 *IsrAdcStartRead* 中执行。

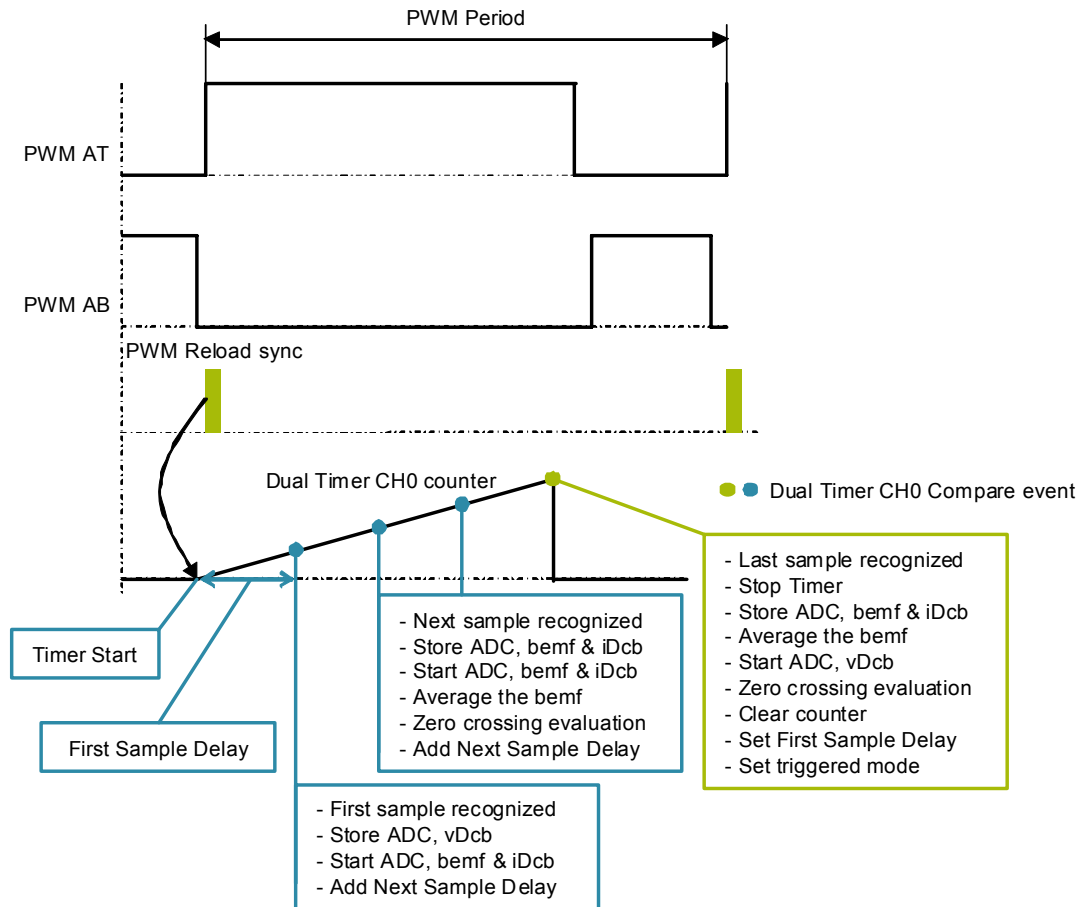


图 5-5. ADC 多重采样机制

采样周期根据 ADC 采样速度、快速中断函数的执行时间以及整个控制系统 CPU 的负荷进行设定。

图 5-6 为实验结果，其中，蓝色轨迹为反电动势，绿色轨迹为双路定时器 CH0 输出（当执行 ADC 采样和处理时，如果发生比较事件，它将切换其状态）。

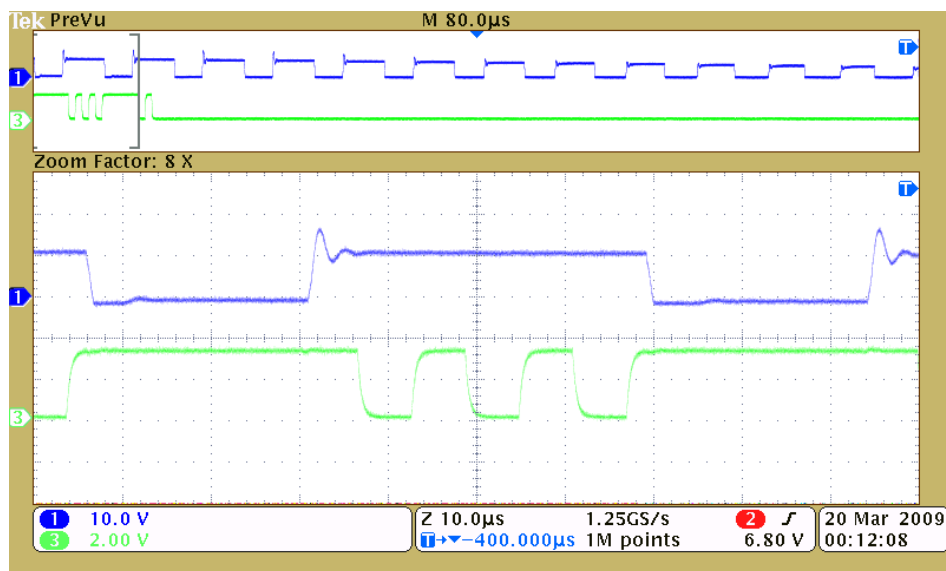


图 5-6. ADC 多重采样机制的实际运作

5.6.2 反电动势过零检测

反电动势过零点的检测是通过 ADC 采样非导通相电压和母线电压来是实现的。

要获得正确的反电动势电压，有两点必须保证：

- 顶部和底部开关（对角）由相同的 PWM 信号驱动。
- 无电流经过用于感应反电动势的非导通相。

第一个条件可以通过将 PWM 模块与 ADC 同步实现。第二个条件通过感应的反电动势电压直接检测。此相与直流母线断开之后，续流二极管中仍有电流经过。导通时间取决于电机的瞬时负载。在某些情况下，导通时间非常长，以致无法检测反电动势电压。由于反向二极管的续流，非导通相的相电压瞬时相反。

换相后，选择 *IsrAdcBlock* 快速中断函数。该函数由双路定时器 CH0 的比较中断调用。在此函数内，需估算一个主要条件。在 PWM 导通期间，仅采集一次反电动势，并直接通过 20% 和 80% 的直流母线电压测试。反电动势电压降至直流母线电压的 20% 至 80% 时，ADC 多重采样技术中的快速中断函数 *IsrAdcStartRead* 发生变化，并且启动估算过零点。

随后将反电动势电压样本与直流母线电压的一半对比。如果识别出过零点，则计算新周期和换相事件，且在下次换相之前双路定时器 CH0 保持停止（此时启用此定时器的触发功能并选择 *IsrAdcBlock* 函数）。过零检测的部分详细信息如图 5-7 所示。

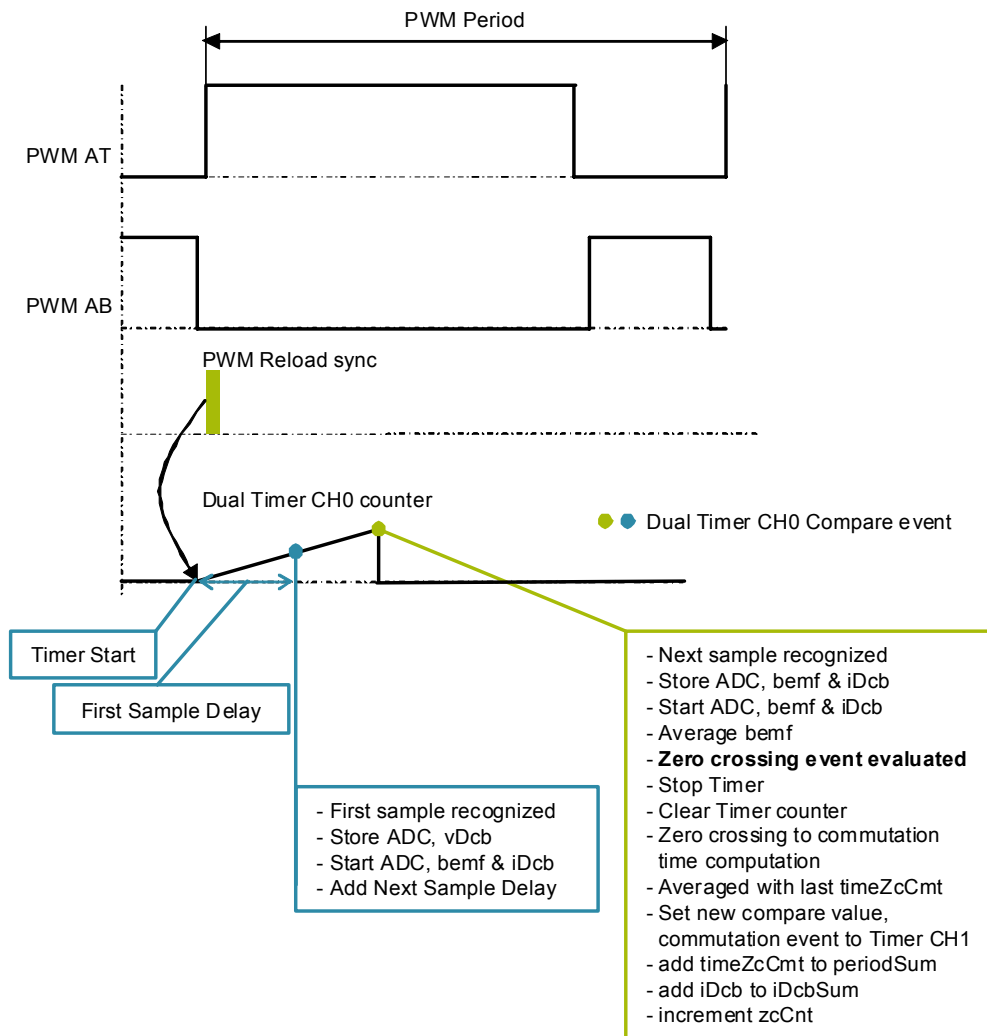


图 5-7. 过零检测

图 5-8 为过零点检测的波形图。其中，蓝色和红色轨迹为反电动势，绿色轨迹为双路定时器 CH0 输出（如果发生比较事件，它将切换其状态）。

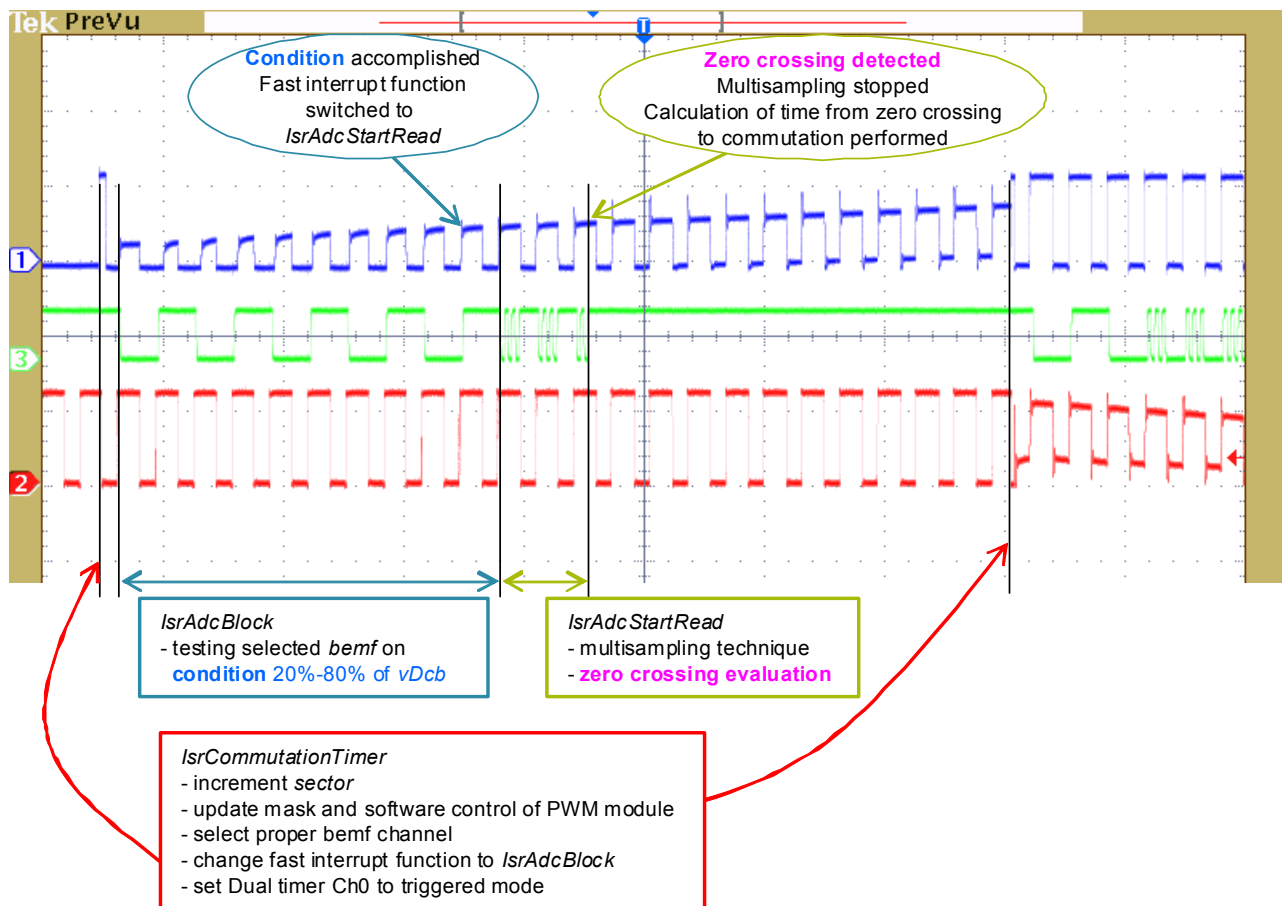


图 5-8. 过零、换相和反电动势测试实用说明

5.7 FreeMASTER 软件

FreeMASTER 软件用于为算法和应用开发提供调试、诊断和演示工具。它也用于不同功率平台和电机的整定应用。几乎所有的应用参数均可通过 FreeMASTER 接口更改。它由一个运行于 PC 上的组件和另一个运行于目标 DSC（通过 RS-232/USB 端口连接）上的器件组成。DSC 中的一个小程序与 FreeMASTER 软件通信，以解析命令、将状态信息返回给 PC 并通过 PC 处理控制信息。将 Microsoft Internet Explorer 用作用户接口，在 PC 上执行 FreeMASTER 软件。

附注

为了正确连接 RS-232/USB，必须先在 PC 上安装 *HC9S08JMxx.inf* 驱动程序，并且必须在 FreeMASTER 项目 / 选项菜单中选择相应的 COM 端口。

5.7.1 FreeMASTER 串行通信驱动程序

该应用包含 FreeMASTER 串行通信驱动程序。FreeMASTER 串行通信驱动程序可完全替代原来的 PC 主机驱动程序。新 FreeMASTER 驱动程序与旧 PC 主机驱动程序提供的通信接口完全兼容，而且具有许多有用的增强功能和优化设置。

新驱动程序的主要优势为统一了所有支持的 Freescale 处理器产品，并新增了多种新功能。新驱动程序的其中一项主要功能为目标端寻址 (TSA) 功能，该功能可让嵌入式应用描述其允许主机访问的内存对象。启用 TSA-Safe 选项后，无法非法或无效访问应用内存。

如果要将新的 FreeMASTER 串行通信驱动程序包括在应用中，用户必须手动将驱动程序文件包括在 CodeWarrior 项目中。此应用中已包含该驱动程序。

FreeMASTER 驱动程序文件位于以下文件夹中：

- *{Project}\support\freemaster\56F8xxx* — 包含平台专用驱动程序 C 源代码和头文件（包含主机头文件 *freemaster.h*）。
- *{Project}\support\freemaster\common* — 包含适合于所有支持平台的驱动程序共享的驱动程序源文件）。

将 FreeMASTER 文件夹中包含的所有 C 文件添加至项目中，以进行编译和链接。请参见项目中的 support group。头文件 *freemaster.h* 声明 FreeMASTER 驱动程序 API 函数的常见数据类型、宏和原型。需要调用任何 FreeMASTER 驱动程序 API 函数时，必须通过 *#include* 指令将其包括在应用中。

FreeMASTER 驱动程序不会初始化或配置其用于通信的 SCI 模块。在通过调用 **FMSTR_Init()** 初始化 FreeMASTER 驱动程序之前，须由用户负责配置通信模块。默认的 SCI 通信波特率设为 9600 Bd。

FreeMASTER 使用轮询驱动通信模式。它无需设置 SCI 中断。通信和协议解码均在应用后台循环中进行。轮询模式需要定期调用主应用中的 **FMSTR_Poll()** 函数。

该驱动程序可通过 *appconfig.h* 头文件配置。最好通过 CodeWarrior 工具栏项目 / 配置工具中提供的快速启动图形配置工具更改此文件。用户必须修改此文件才能配置 FreeMASTER 驱动程序。FreeMASTER 驱动程序 C 源代码文件包括配置文件以及为条件和参数编译而定义的宏。

有关 FreeMASTER 串行通信驱动程序的详细说明，请参见 *FreeMASTER 串行通信驱动程序用户手册*。

5.7.2 FreeMASTER 记录器

FreeMASTER 软件还包括另一部分，即记录器，它能够以指定的采样率对应用变量采样。样本存储在缓冲器中。采样数据可以图形显示，并且数据可以存储。记录器如同一个带触发器 / 预触发器功能的简单片上示波器。记录器缓冲区大小以及 FreeMASTER 记录器时基可在 *appconfig.h* 配置中定义。

必须从定期中断中调用记录器例程，以进行采样。以下行必须添加至代码中：

```
FMSTR_Recorder(); /* FreeMASTER recorder routine call */
```

在此应用中，FreeMASTER 记录器在 ADC EOS 中断中调用，后者可为记录器函数生成 125 μ s 时基。有关 FreeMASTER 软件的详细说明，请参见 *FreeMASTER 软件用户手册*。

5.7.3 FreeMASTER 控制页面

FreeMASTER 控制页面为无传感器 BLDC 控制应用创建图形用户界面 (GUI)。可以通过单击 *BLDC_control panel.pmp* 文件启动 FreeMASTER 软件窗口。图 5-9 介绍了启动此项目之后的 FreeMASTER 软件控制窗口。

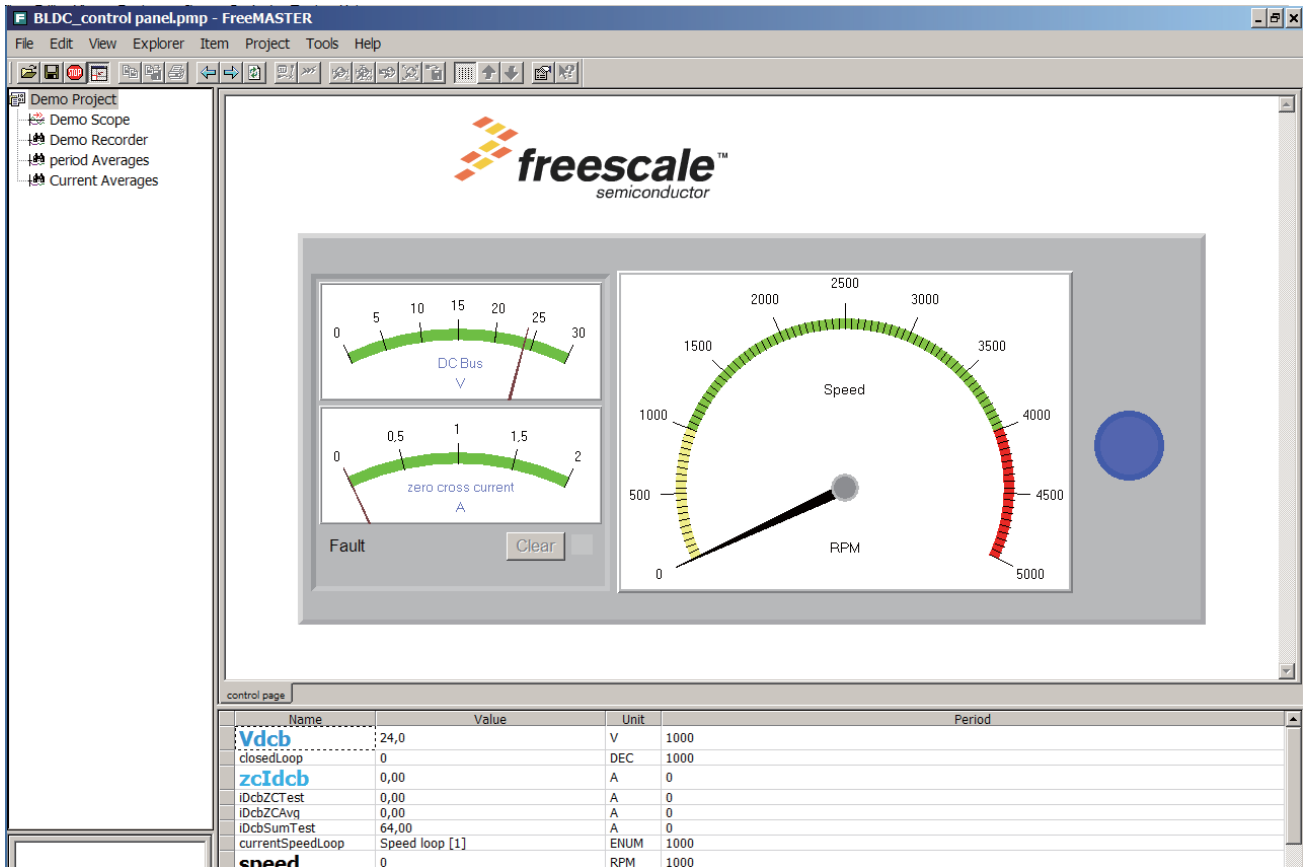


图 5-9. FreeMASTER 控制屏幕

FreeMASTER 软件控制页面。以下所示为支持的主要操作：

- 设置所需的电机速度。
- 复位应用错误状态。

FreeMASTER 软件控制页面显示：

- 直流母线电流和电压。
- 实际电机速度。
- 应用状态。

以下子模块用于介绍软件调试和演示：

- 演示测量。
- 控制调节器，用于调整 PI 控制器参数。

5.8 特定电机的软件参数设置

默认的软件参数是以 LINIX-45ZWN24-40 为对象设置的。如第 6 章“应用设置”所示。如果使用其他电机或硬件需根据实际电机参数值或硬件定标值修改软件参数。软件参数或系数在 *main.h* 中。

第 6 章 应用设置

如先前所述，无传感器 BLDC 电机控制应用针对 MC56F8006 器件。无传感器 BLDC 电机控制驱动包含以下硬件组件：

- 3 相 BLDC/PMSM 低压电机控制驱动板
- MC56F8006 子板
- 3 相低压 BLDC 电机 LINIX 45ZWN24-40



图 6-1. 演示应用

6.1 硬件设置

整个 demo 如图 6-2 所示。



图 6-2. BLDC 无传感器控制演示概述

三相低压电机控制板无需任何设置。要正确运行此 demo，必须检查子板上的跳线配置是否正确。在图 6-3 中，介绍了 MC56F8006 子板的跳线配置。针对无传感器的 BLDC 进行配置 ADC 的输入通道跳线。这意味着需要测量三个反电动势电压和直流母线电流。直流母线电压分压后不需要配置跳线直接连到 MCU 上。最终配置选择取决于 DSC 引脚与 RxD 和 ENC 的信号连接情况。Rx/D 用于使用 FreeMASTER 进行应用控制。

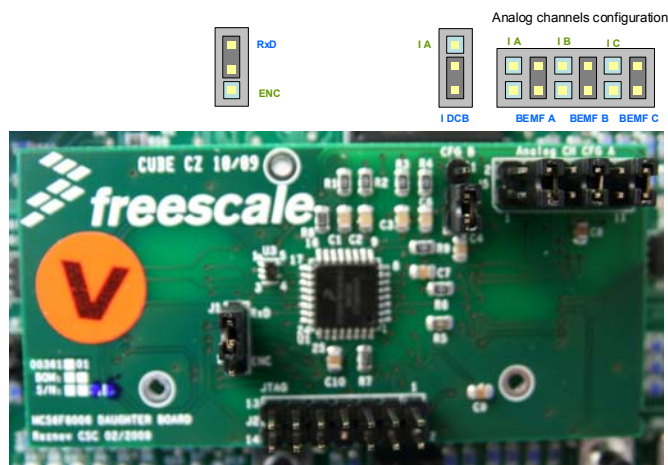


图 6-3. 子板配置

6.2 软件设置

以下步骤简短介绍了基于 MC56F8006 怎样移植和运行代码。无传感器 BLDC 控制软件作为 CodeWarrior development studio 项目运行。必须先安装 CodeWarrior。接着，打开项目文件 *BLDC.mcp*，该文件包含所有需要的源代码、头文件和库（请参见图 6-4）。

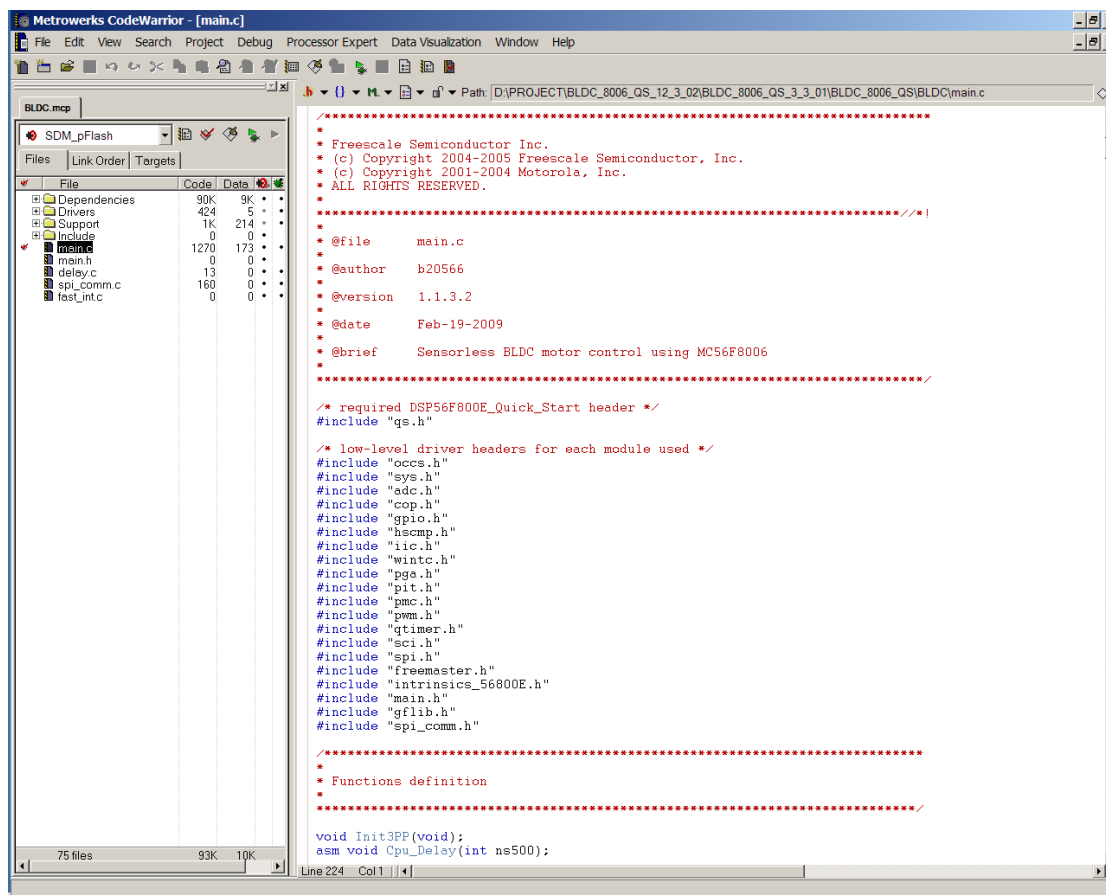


图 6-4. CodeWarrior 屏幕（BLDC 项目已打开）

另一个重要的工具为 CodeWarrior USB TAP，需要将其连接至 DSC 的 JTAG 端口，如图 6-2 所示。随后可以编译项目（菜单 *Project/Make* 或 F7 键）并将其下载到 DSC 中，如图 6-5 所示。编译和下载完成后，可通过单击“运行”图标（图 6-6 中红色高亮显示）启动程序。此时，系统已可以启动工作并且可以通过 FreeMASTER 或手动接口控制。FreeMASTER 设置如 5.7 “FreeMASTER 软件”所述。

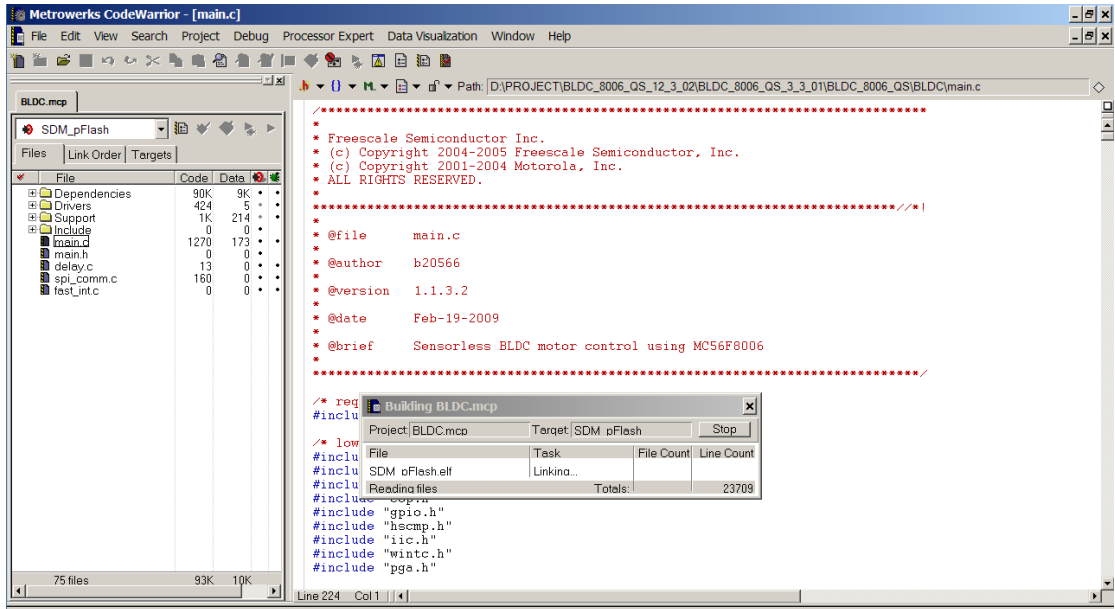


图 6-5. 创建进程中的项目

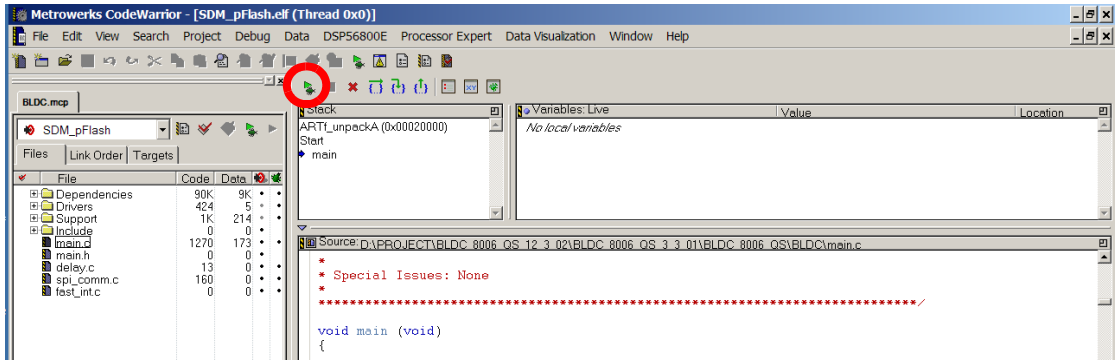


图 6-6. 创建进程后的项目

用户可能会遇到某些可调应用变量，例如 PI 控制器设置 (*gainSpeed*、*integralSpeed*、*gainCurrent*、*integralCurrent*)、控制器阈值 (*currentCntrlTh*、*speedCntrlTh*)、提前换相系数 (*commCoef*) 以及 1/2 直流母线电压系数 (*dcbHalfCoef*)。这些变量的默认值位于 *main.h* 文件中，在初始化过程中定义常数并将其复制到相应的变量中。可以使用 FreeMASTER 工具在线调整这些变量。如果电机类型、设计平台不一样或客户有其他需求，要根据实际修改代码。对于不同的电机，修改的内容主要是换相表 *COMM_TABLE_MASK*、*COMM_TABLE_SWCTRL* 和 *BEMF_CH_TABLE*。

第 7 章 结果和测量

本节介绍无传感器 BLDC 控制应用的基本测量。首先，图 7-1 所示为发送至用户输出端的三个反电动势电压波形和一个过零信号（绿色轨迹）。

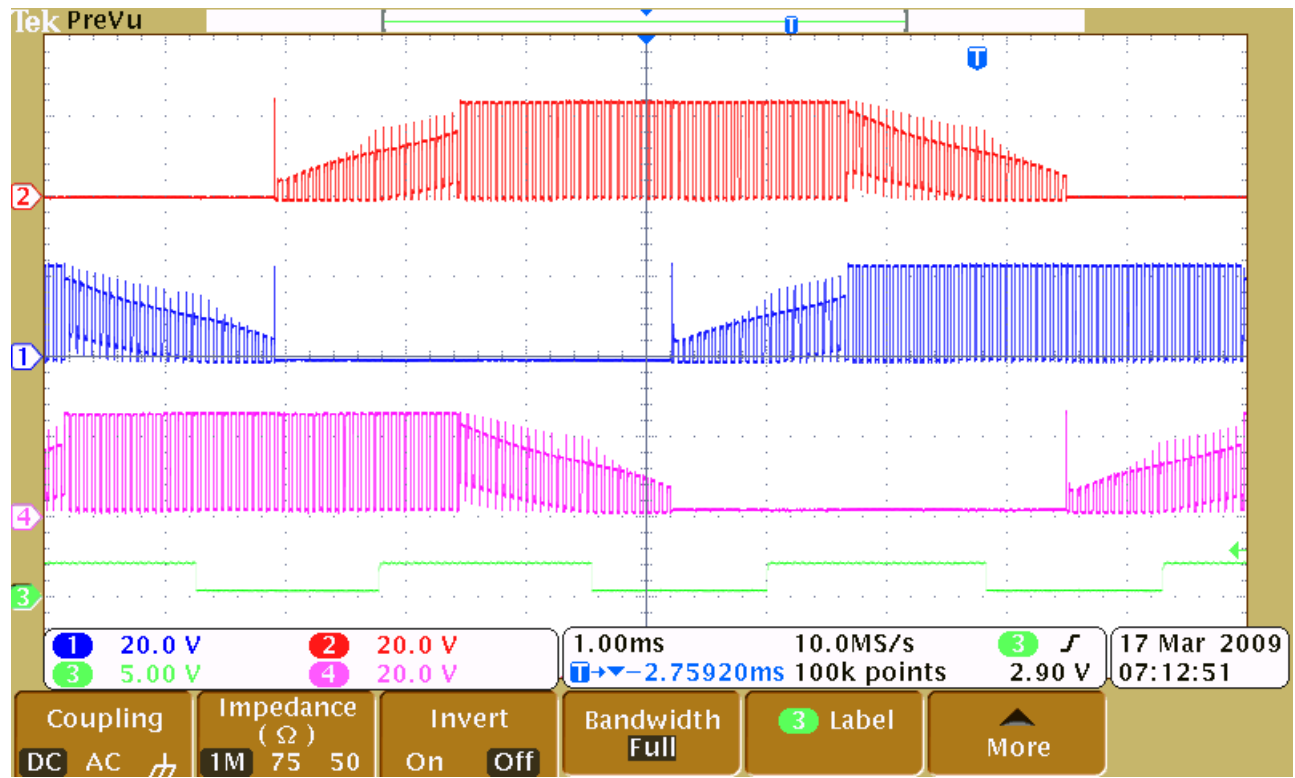


图 7-1. 无传感器 BLDC 电机实际波形

接着，图 7-2 通过 FreeMASTER 捕捉，并显示所需速度和实际速度的轨迹。

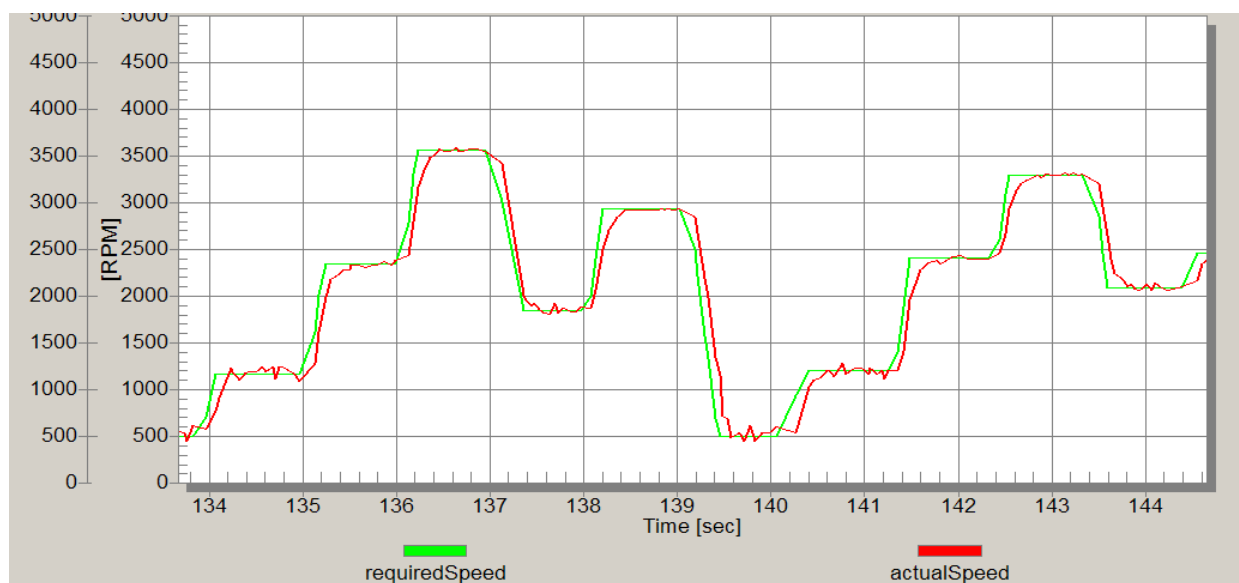


图 7-2. 所需速度和实际速度轨迹

7.1 结论

此 DRM 介绍了 MC56F8006 在 BLDC 电机驱动中的应用。MC56F8006 的特性和计算能力完全符合无传感器 BLDC 电机控制的要求。测量结果显示，所开发控件在宽速度范围及建议的正确算法下能够提供稳定的结果。因此，可通过多重采样实现准确的位置估算，无需其他系统中常用的每个 PWM 周期单个样本的复杂内插方法。

How to Reach Us:**Home Page:**freescale.com**Web Support:**freescale.com/support

本文档中的信息仅供系统和软件实施方使用 Freescale 产品。本文并未明示或者暗示授予利用本文档信息进行设计或者加工集成电路的版权许可。Freescale 保留对此处任何产品进行更改的权利，恕不另行通知。

Freescale 对其产品在任何特定用途方面的适用性不做任何担保、表示或保证，也不承担因为应用程序或者使用产品或电路所产生的任何责任，明确拒绝承担包括但不限于后果性的或附带性的损害在内的所有责任。Freescale 的数据表和 / 或规格中所提供的“典型”参数在不同应用中可能并且确实不同，实际性能会随时间而有所变化。所有运行参数，包括“经典值”在内，必须经由客户的技术专家对每个客户的应用程序进行验证。Freescale 未转让与其专利权及其他权利相关的许可。Freescale 销售产品时遵循以下网址中包含的标准销售条款和条件：

freescale.com/SalesTermsandConditions.

Freescale™ and the Freescale logo are trademarks of Freescale Semiconductor, Inc. The ARM POWERED logo is a registered trademark of ARM Limited. ARM is a registered trademark of ARM Limited. Java and all other Java-based marks are trademarks or registered trademarks of Sun Microsystems, Inc. in the U.S. and other countries. The PowerPC name is a trademark of IBM Corp. and is used under license. The described product contains a PowerPC processor core. The PowerPC name is a trademark of IBM Corp. and used under license. The described product is a PowerPC microprocessor. The PowerPC name is a trademark of IBM Corp. and is used under license. The described product is a PowerPC microprocessor core. The PowerPC name is a trademark of IBM Corp. and is used under license. All other product or service names are the property of their respective owners.

© Freescale Semiconductor, Inc. 2009. All rights reserved.

Document Number: DRM108
Rev. 0, 04/2009

