

利用 MKV10x 实现三相 BLDC 无传感器控制

作者: Petr Staszko

1 简介

本应用笔记说明如何使用 Freescale 32 位 Kinetis MKV10x 器件实现无传感器三相无刷直流(BLDC)电机控制应用(该应用的说明参见 *DRM144: 三相 BLDC 无传感器电机控制应用设计参考手册*)。所提供的应用软件使用了孤立算法软件和硬件的概念。通过这种软件方法,应用很容易移植到其他器件或平台上。应用软件分为以下几部分:

- BLDC 电机控制算法处理输入变量以输出变量和标志。
- MKV10x 硬件和微处理器充当硬件外设模块与 BLDC 电机控制软件算法之间的桥梁。

本应用笔记描述 Kinetis MKV10x 器件的具体实现,而不讨论代码中的 BLDC 电机控制算法部分。

目录

1	简介	1
1.1	MKV10x 器件的优点和特性	2
2	应用原理	3
2.1	I/O 值处理	4
2.2	应用输入值	5
2.3	应用输出值和标志	6
2.4	互补双极开关 PWM	6
3	应用详情	7
3.1	MKV10x 器件配置	7
3.2	应用中断服务例程	10
3.3	FlexTimer 寄存器更新	11
4	用户接口	13
4.1	上/下按钮控制	13
4.2	使用 FreeMASTER 远程控制	14
5	硬件	15
5.1	TWR-MC-LV3PH	15
5.2	TWR-KV10Z32	18
5.3	电机 LINIX 45ZWN24-40	19
5.4	微控制器存储器使用	20
6	缩略语	20
7	参考文献	21
8	修订历史	21

本应用笔记的目的是在 MKV10x 器件上实现无传感器 BLDC 电机控制算法。

本应用笔记使用 MKV10x 器件的外设模块，提供 BLDC 电机控制设计的一个实例，以使用户轻松了解 BLDC 的控制方法。该应用是裸机型。

本应用笔记包括 MKV10x 器件的概述、系统设计概念和硬件实施。

硬件基于 Freescale Tower 快速原型开发系统，包含下列模块：

- TWR-Elevator
- TWR-KV10Z32
- TWR-MC-LV3PH

1.1 MKV10x 器件的优点和特性

Kinetis KV1x 子系列是超低成本的 ARM[®] Cortex[®]-M0+ MCU 产品组合，其外设模块专门用于电机控制应用。典型应用领域包括 BLDC 无传感器、PMSM 无传感器 FOC（低动态）、ACIM V/Hz 和 FOC（低动态）电机控制应用。

MKV10x 器件的主要特性如下：

- 75 MHz ARM Cortex-M0+
- 硬件除法外设模块
- 平方根外设模块
- 4 通道 eDMA
- 16 或 32 KB Flash 存储器
- 8 KB SRAM
- IIC、SPI、2 x UART 通信端口
- 2 x 8 通道 16 位 ADC 模块
- 12 位 DAC 模块
- 2 x 模拟比较器（带 6 位 DAC）
- 1 x 6 通道 FlexTimer 模块
- 2 x 2 通道 FlexTimer（具有正交解码特性）
- 可编程延迟模块（PDB）
- 16/32 位 CRC
- 多达 35 个 GPIO
- 1.71–3.6 V，-40 至 105 °C

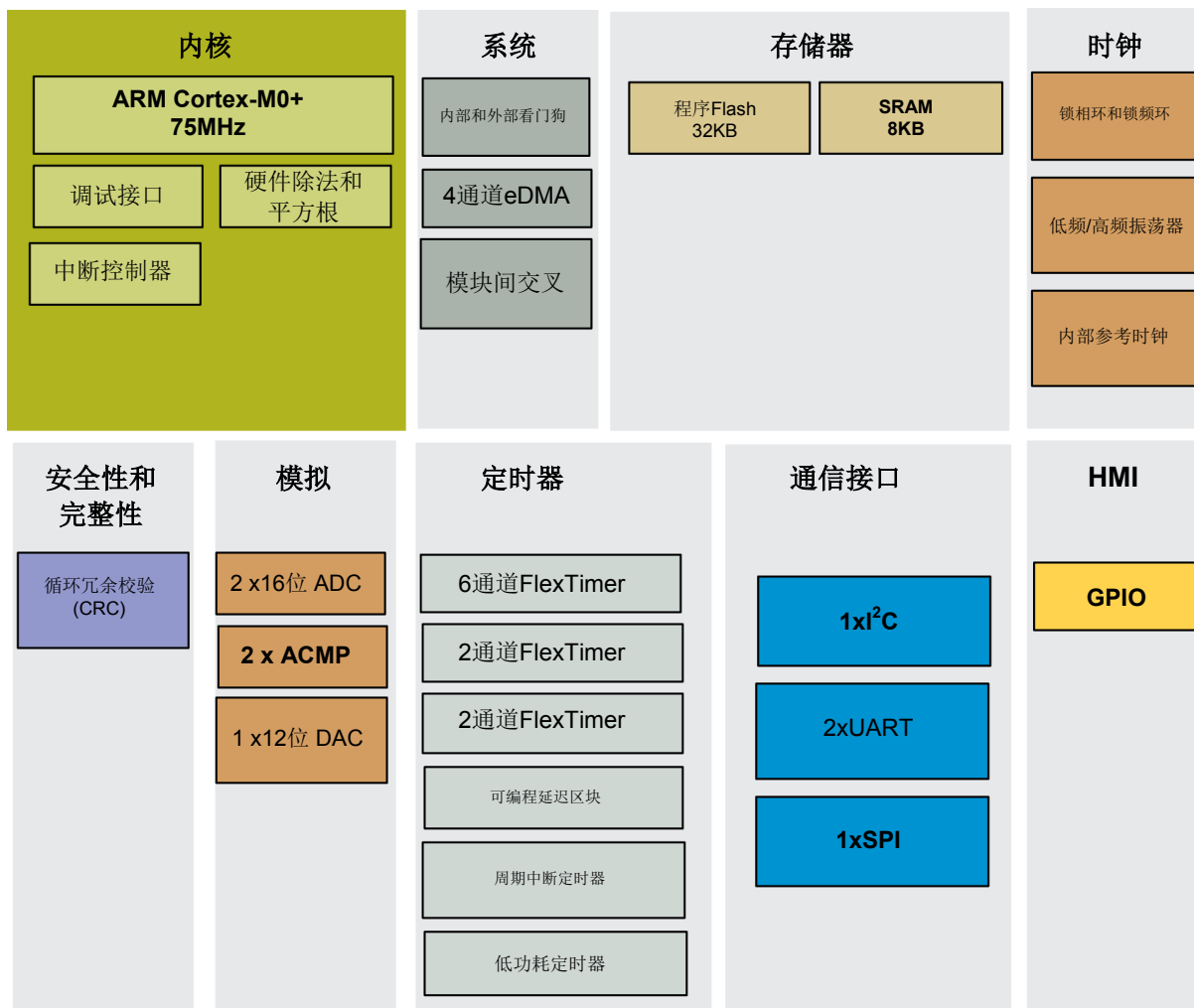


图 1. MKV10x 器件概览

2 应用原理

本应用的一个重要开发理念是软件能够在不同器件或平台之间轻松移植。因此，应用软件分为两个独立的代码部分：

- 硬件相关代码（取决于所用的硬件板和 MCU 器件的外设模块），包括 CPU 和外设模块初始化、I/O 控制驱动程序以及中断服务例程处理。
- 硬件无关代码（纯粹的 BLDC 电机控制应用）

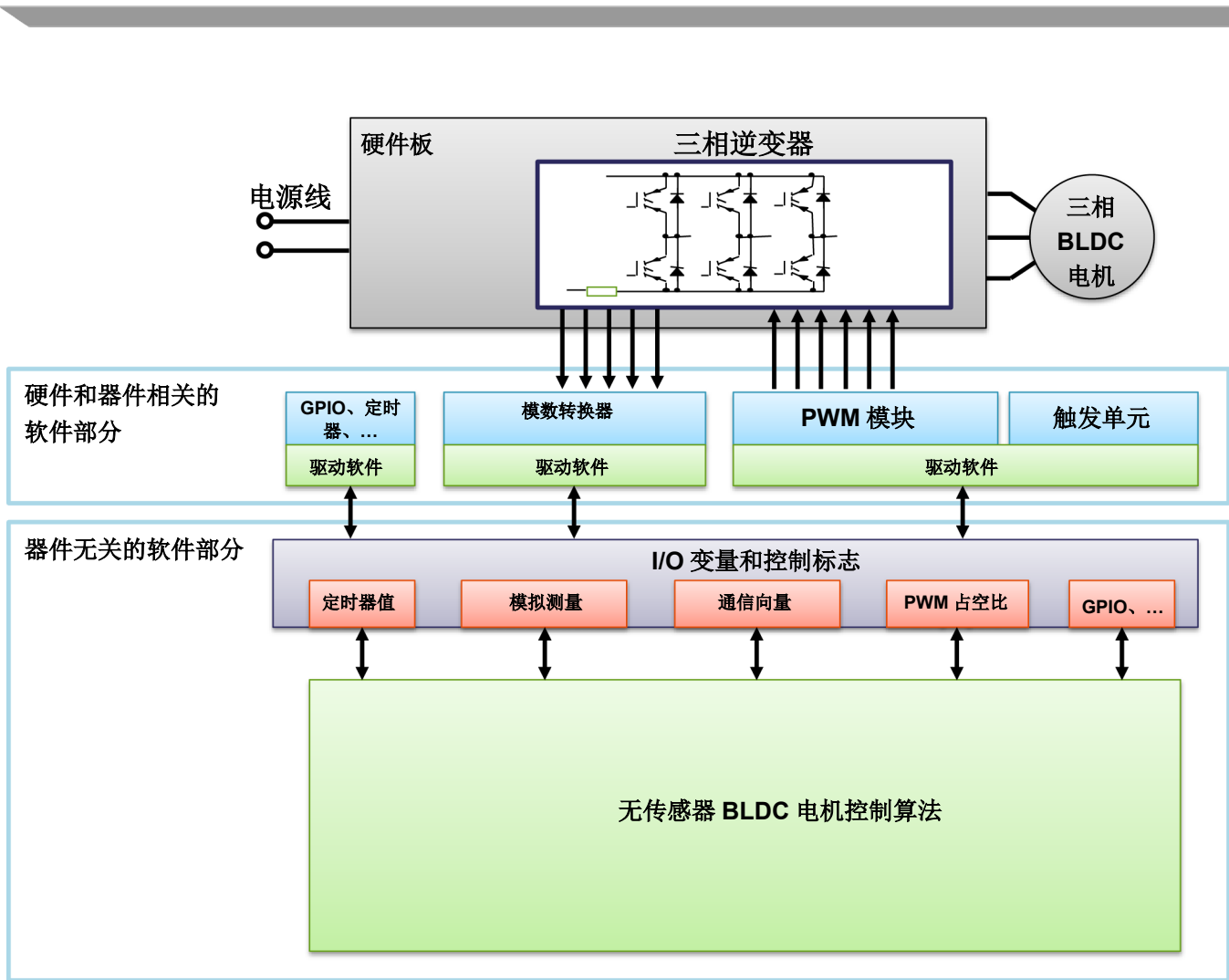


图 2. 系统原理概览

应用的系统原理概览如图 2 所示。输入和输出变量及输出控制标志用作硬件相关代码部分和电机控制算法之间的桥梁。应用的大部分输入和输出变量以小数形式（介于-1 到 1 之间）存储，这些变量与实际值的转换在驱动程序（硬件相关的代码部分）中计算。

2.1 I/O 值处理

本应用使用了将 MCU 相关代码部分与硬件无关 BLDC 电机控制算法相分离的概念。硬件相关代码从外设模块读取输入值（定时器计数值、ADC 结果及其他值），将其缩放为小数形式（若需要）并存储为输入变量。然后，硬件无关的 BLDC 电机控制算法软件部分会处理这些输入变量。类似地，BLDC 算法输出变量在硬件相关代码中进行处理，并更新到相应的外设模块中。如果需要更新输出硬件参数，则在应用控制算法执行过程中，变量 `uw16HwUpdateRequests` 将更新对应的请求标志(request flag)。函数 `UpdateBldcHwModules()` 用于检查请求标志的状态，并执行相应的硬件模块更新。

下图概要显示了应用 I/O 数据流：

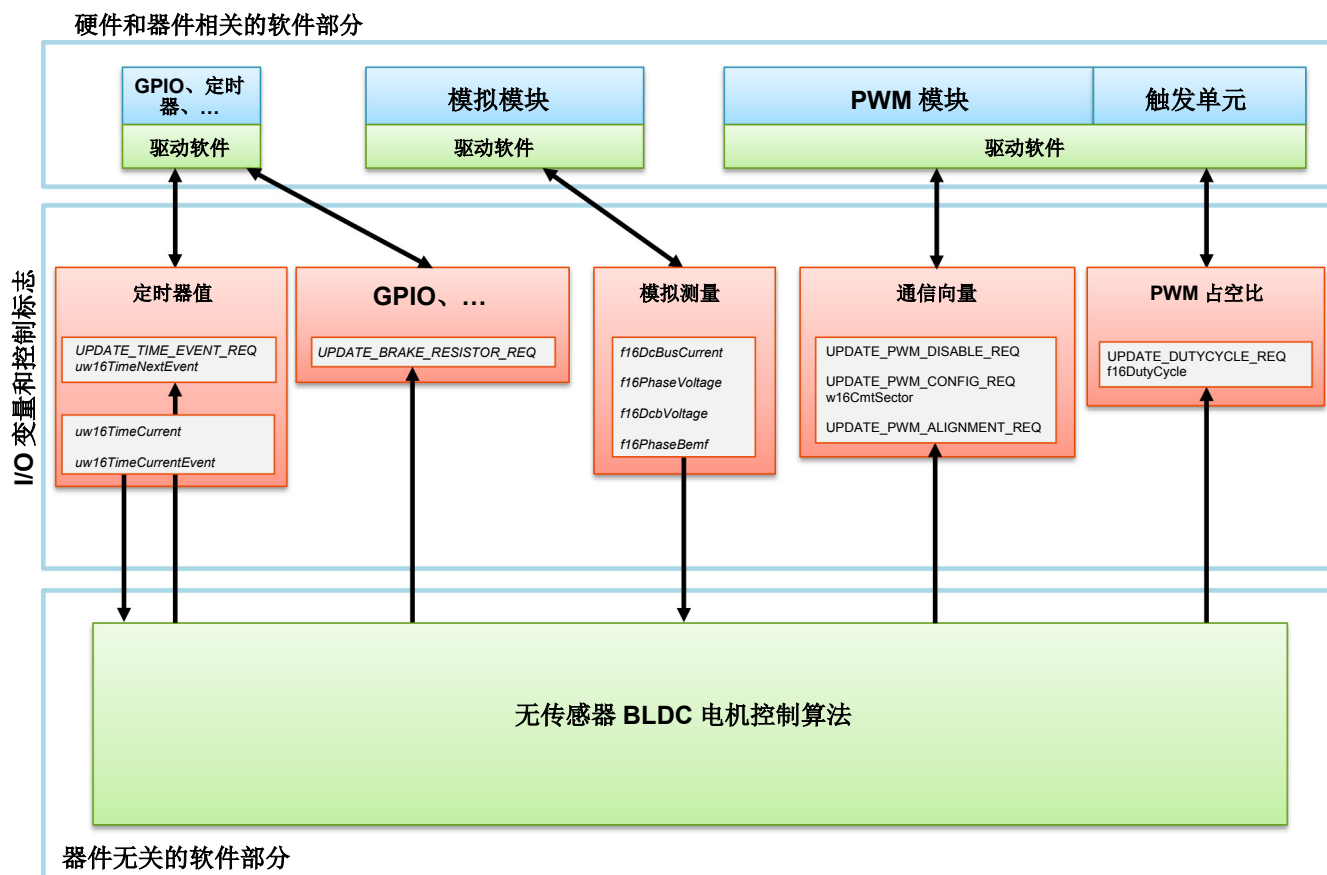


图 3. 应用输入/输出数据流概览

2.2 应用输入值

应用输入变量在相应中断服务例程开始时更新：

- *uw16TimeCurrent* – 实际时间值。如果定时器用于计时目的，则定时器计数器寄存器的实际值存储在此变量中。
- *uw16TimeCurrentEvent* – 上一时间事件(*Time Event*)的时间。如果定时器用于计时目的，则定时器值寄存器存储在此变量中。
- *f16DcBusCurrent* – 直流母线电流值以小数形式存储在该 16 位变量中。-1 表示最大负直流母线电流；+1 表示最大正直流母线电流。
- *f16DcBusCurrentOffset* – 直流母线电流失调值以小数形式存储在该 16 位变量中。该值在应用初始化之后的校准阶段测量。

- *f16DcbVoltage* – 直流母线电压值以小数形式存储在该 16 位变量中。0 表示 0 V，+1 表示最大可测量直流母线电压。
- *f16PhaseVoltage* – 相电压值以小数形式存储在该 16 位变量中。0 表示 0 V，+1 表示最大可测量直流母线电压。
- *f16PhaseBemf* – 计算得到的反电动势电压值以小数形式存储在该 16 位变量中。该值通过下式计算 (*f16DcbVoltage* / 2 被视为电机绕组的中间点)：

$$f16PhaseBemf = f16PhaseVoltage - \frac{f16DcbVoltage}{2}$$

2.3 应用输出值和标志

BLDC 电机控制算法执行之后，可能存在待处理的硬件输出更新请求。为此，需要执行 *UpdateBlcHwModules()* 函数。该函数检查 *uw16HwUpdateRequests* 变量中存储的所有请求标志，并根据硬件配置执行适当的操作：

- *UPDATE_TIME_EVENT_REQ* – 请求设置时间事件的新时间。变量 *uw16TimeNextEvent* 中存储要设置的新时间。如果定时器用于计时目的，则定时器值寄存器更新为 *uw16TimeNextEvent* 的值。
- *UPDATE_PWM_DISABLE_REQ* – 请求禁用所有 PWM 输出。必须关闭功率级输出。
- *UPDATE_DUTYCYCLE_REQ* – 请求更新 PWM 模块产生的占空比。变量 *f16DutyCycle* 中存储小数形式的新占空比值。小数值在范围<-1; +1)中，由适当的 PWM 驱动函数处理并放大，以适应实际的 PWM 模块配置。
- *UPDATE_PWM_CONFIG_REQ* – 请求更新 PWM 输出以设置变量 *w16CmtSector* 中存储的新通信向量。*w16CmtSector* 中存储的是 0 到 5 范围内的整数值。
- *UPDATE_PWM_ALIGNMENT_REQ* – 请求对 PWM 输出应用对齐向量。对齐向量可以利用 PWM 模块配置的特殊情形来应用。
- *UPDATE_BRAKE_RESISTOR_ON_REQ* – 如果该位置 1，则用于控制制动电阻的输出 MOSFET 开启；如果该位清 0，则用于控制制动电阻的输出 MOSFET 关断。

2.4 互补双极开关 PWM

BLDC 电机控制应用采用互补双极开关 PWM。利用互补双极开关，两相由互补 PWM 信号供电（底部 MOSFET 的开关与一相之内的顶部 MOSFET 的开关互补），一相的占空比大于 50%，另一相的占空比为互补值（小于 50%），如图 4 所示。

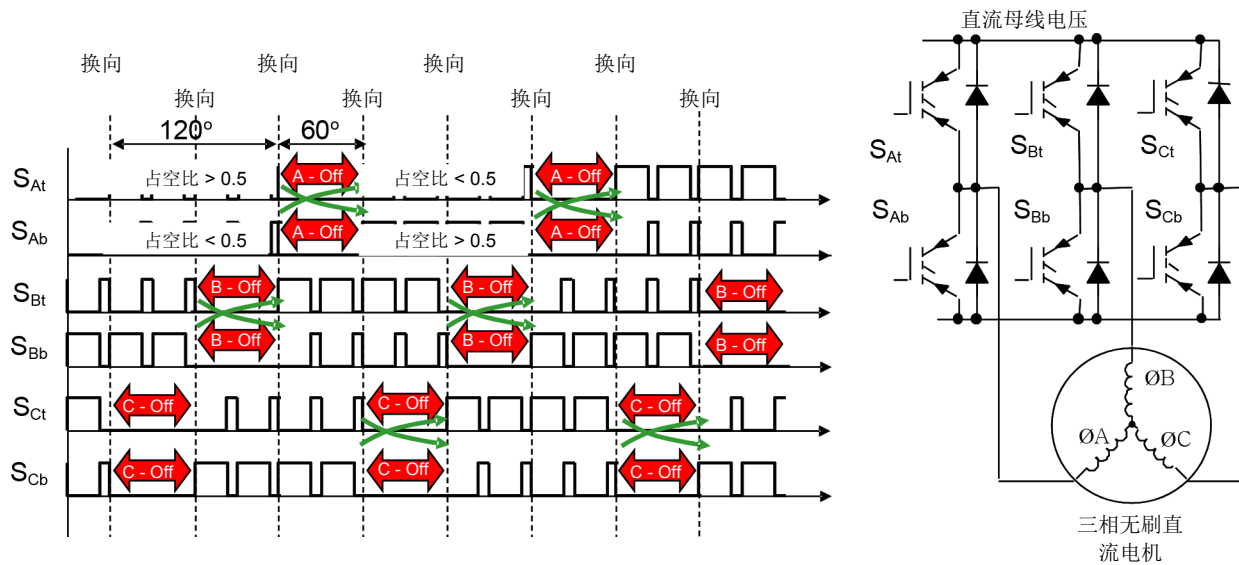


图 4. 互补双极 PWM 开关

互补双极开关 PWM 技术的优点是可以在所有四个工作象限中控制电机。双极 PWM 开关方案要求顶部和底部开关 PWM 信号在换向事件时交换。另一个要求是在互补顶部和底部信号中插入死区。MKV10x 器件的 FlexTimer 能够有效满足所有这些要求，相关说明参见 [FlexTimer 寄存器更新](#) 部分。

3 应用详情

本部分说明 MKV10x 器件外设模块配置、中断服务例程配置以及利用 FlexTimer 特性来支持 BLDC 电机换向。

3.1 MKV10x 器件配置

应用采用以下外设和外部模块来实现无传感器 BLDC 电机控制：

- 时钟分配模块 (MCG、SIM)
- 模数转换器(ADC0)
- FlexTimer 模块 (FTM0、FTM1、FTM2)
- 可编程延迟模块(PDB0)
- 串行外设接口(SPI0)
- 通用异步收发器(UART1)
- MC33927 外部 MOSFET 驱动器 (通过 SPI0)

3.1.1 时钟分配模块 (MCG、SIM)

MKV10Zx 器件利用两个模块来配置和分配外设模块的时钟——MCG 和 SIM 模块。MCG(多用途时钟发生器)模块为 MCU 提供多个时钟源选择。SIM 模块 (系统集成模块) 提供系统控制和芯

片配置。模块配置说明如下：

MCG:

- 利用慢速内部参考时钟(32 768 Hz)作为时钟源
- 利用 FLL 从输入时钟产生 72 MHz 频率（FLL 系数为 2197）

SIM:

- 使能所有外设模块的时钟
- 系统时钟（内核时钟）分频比设为 1 => 系统时钟设置为 72 MHz
- 总线/Flash 时钟分频比设为 3 => 总线时钟设置为 24 MHz

3.1.2 模数转换器(ADC0)

ADC0 模块用于直流母线电压、直流母线电流和相电压测量，其配置如下：

- 输入时钟设置为总线时钟 / 1 = 24 MHz
- 高速模式(24 MHz)
- 短采样（采样时间为 6 ADCK 周期）
- 单端输入转换模式
- 12 位分辨率
- PDB 硬件触发
- 转换完成 ISR，优先级 1（高优先级）

3.1.3 FlexTimer 模块（FTM0、FTM1、FTM2）

FlexTimer 模块用于输出 PWM 生成(FTM0)，通信控制、速度测量和应用计时(FTM1)，以及慢速控制循环(FTM2)。这些模块具有如下配置：

FTM0 – 输出 PWM 生成：

- FTM 处于增强模式
- 输入时钟设置为系统时钟 / 1 = 72 MHz
- 输出 PWM 频率设置为 20 kHz
- 使用(n)和(n+1)组合通道特性配置中心对齐的互补 PWM
- 互补 PWM 生成
- 死区设置为 875 ns
- 利用 *LDOK* 特性更新占空比
- 利用软件同步(*software synchronization*)特性（换向）更新输出屏蔽、逆变（交换）和软件控制寄存器
- 换向时复位 FTM 计数器
- 使能 PDB 触发

FTM1 – 通信控制:

- FTM 处于增强模式
- 自由运行定时器
- 输入时钟设置为系统时钟 / 128 = 562 500 Hz
- 输出比较模式
- 输出比较时执行 ISR，优先级 1（高优先级）

FTM2 – 慢速控制循环:

- FTM 处于增强模式
- 自由运行定时器
- 输入时钟设置为系统时钟 / 2 = 36 MHz
- 输出比较模式
- 输出比较时执行 ISR，优先级 3（低优先级）

3.1.4 可编程延迟区块(PDB0)

PDB0 模块用于同步 PWM 周期内的 ADC0 测量时间。该模块配置如下:

- 输入时钟设置为总线时钟 / 1 = 24 MHz
- 选择 FTM0 外部触发器(EXTRIG)作为源 PDB0 触发器
- 寄存器更新选择双缓存模式
- 预触发 0 点设置为 PWM 脉冲的中点
- 预触发 1 设置背靠背模式（第二 ADC 测量）
- 使能序列错误 ISR，优先级 1（高优先级）

注释

PDB 时钟为总线时钟(24 MHz)，FTM 时钟为系统时钟(72 MHz)。

3.1.5 串行外设接口(SPI0)

SPI0 是一个四线接口，用于 MC33927/33937 三相前置驱动器配置和状态读取。该模块配置如下:

- 波特率设置为 1.28 MHz
- 下降沿 SCLK 极性
- 主机 SPI 模式

3.1.6 通用异步收发器(UART1)

UART1 用于 MCU 板与 PC 之间的 FreeMASTER 通信。该模块配置如下:

- 波特率设置为 9600 bps
- 使能接收器和发送器
- 其他设置为默认值

3.1.7 MC33927 外部 MOSFET 驱动器（通过 SPI0）

MC33927/MC33937 三相 MOSFET 驱动器用于输出控制。SPI0 通信通道用于配置和读取驱动器的状态。该驱动器配置如下：

- 使能去饱和和保护。
- 禁用中断（使用池模式）
- 死区设置为 0（使用 FlexTimer 上的死区）

3.2 应用中断服务例程

应用使用四个中断服务，其中三个属于 BLDC 电机控制算法，一个用于触发错误清除。

- ADC0 ISR – 快速控制循环 ISR，每个 PWM 周期(50 us)执行一次
- FTM1 ISR – 时间事件 ISR，根据当前应用需求执行
- FTM2 ISR – 慢速控制循环 ISR，每 1 ms 执行一次
- PDB0 ISR – 触发错误故障清除

可使用以下函数（在 `arm_cm0.h` 文件中定义）来使能、禁用各 ISR 或设置其优先级：

- `disable_irq(IRQ_number)`
- `enable_irq(IRQ_number)`
- `set_irq_priority(IRQ_number, priority_level)`

ISR 编号和优先级汇总参见表 1：

表 1. 中断服务列表

源	模块功能	应用功能	向量编号	ISR 名称	级别
ADC0	转换完成	快速控制循环	VECTOR_031	<code>adc0_isr()</code>	1
FTM1	输出比较	时间事件	VECTOR_034	<code>ftm1_isr()</code>	1
FTM2	输出比较	慢速控制循环	VECTOR_035	<code>ftm2_isr()</code>	3
PDB0	触发错误	触发错误	VECTOR_045	<code>pdb0_isr()</code>	1

默认情况下，所有中断向量都设置为空 ISR 函数 `default_isr()`。要替换默认 ISR，请使用 `hwconfig.h` 文件中的以下代码：

```
#undef VECTOR_031
#define VECTOR_031 adc0_isr
```

```
#undef VECTOR_034
#define VECTOR_034 ftm1_isr

#undef VECTOR_035
#define VECTOR_035 ftm2_isr

#undef VECTOR_045
#define VECTOR_045 pdb0_isr
```

3.2.1 ADC0 转换完成 ISR

ADC0 模块用于在每个 PWM 周期执行三个模拟测量（直流母线电压、直流母线电流和断开的电机相位上的相电压）。KV10x 器件的每个 ADC 模块只有两个通道选择和结果寄存器，因此对于 PDB 的一个硬件触发信号，只能按顺序完成两个测量，然后就会产生 ADC0 转换完成中断。第三个测量在此 ISR 内配置、开始并完成。ADC 配置为软件触发模式，转换由写入通道配置寄存器开始，算法等待转换结束，然后 ADC0 重新配置为硬件触发模式。ADC 转换结果处理为左对齐值。然后执行快速控制循环功能。写入相应的 SC1n（通道选择）寄存器时，或读取相应的 Rn 寄存器（结果寄存器）时，中断请求转换完成标志(COCO)清 0。

3.2.2 FTM1 输出比较 ISR

FTM1 ISR 用于计时目的和开环 BLDC 启动期间的换向定时。FTM1 使用 562 500 Hz 输入时钟，周期由 16 位正数定义，最大值为 58.25 ms（最大换向周期）。中断标志需要通过写入 0 来清除。

3.2.3 FTM2 输出比较 ISR

FTM2 ISR 用于慢速控制循环功能执行和 MC33927 状态读取。此 ISR 每 1 ms 执行一次。FTM2 使用 36 MHz 输入时钟，每完成一次 ISR，通道值寄存器便递增 36 000。中断标志需要通过写入 0 来清除。

3.2.4 PDB0 触发错误 ISR

当 PDB0 无法触发 ADC 模块时，便会产生触发错误中断。通常，如果在产生 PDB 触发信号的同时 ADC 转换仍在进行，就会引起错误，因而无法启动下一 ADC 转换（两个 PDB 触发信号之间的间隔太短）。如果触发错误标志置 1，PDB 模块就会停止工作。要清除错误标志，PDB0 模块需要禁用，然后写入 0 便可清除错误标志，随后可使能 PDB0 模块。

3.3 FlexTimer 寄存器更新

增强模式下的 FlexTimer 为双缓存寄存器更新提供了多个选项。BLDC 电机控制应用中有两个算法控制循环——输出电压控制循环和换向控制循环。这两个控制循环彼此独立，KV10x 器件的 FlexTimer 为输出电压控制循环和换向控制循环的双缓存寄存器的独立更新提供了配置选项。

3.3.1 输出电压控制

在每 1 ms 执行一次的慢速控制循环中，计算并施加输出电压（PWM 占空比）以实现所需的电机转速。所有相位上的 PWM 占空比需要同时应用，以免功率级输出线路发生干扰或危险。为此，FlexTimer 模块提供双缓存机制，在 MOD（模数）、CNTIN（计数器初始值）和 CnV（通道 n 值）寄存器加载其写入缓存值的同时，PWMLOAD 寄存器中的 LDOK 位设为 1，并开始新的 PWM 周期。

下面是使用 LDOK 机制更新 PWM 输出占空比的代码示例：

```
FTM0->CONTROLS[0].CnV = FirstEdge;
FTM0->CONTROLS[1].CnV = SecondEdge;
FTM0->CONTROLS[2].CnV = FirstEdge;
FTM0->CONTROLS[3].CnV = SecondEdge;
FTM0->CONTROLS[4].CnV = FirstEdge;
FTM0->CONTROLS[5].CnV = SecondEdge;
FTM0->PWMLOAD |= FTM_PWMLOAD_LDOK_MASK;
```

3.3.2 换向控制

换向事件的时间取决于转子的实际位置，因此换向周期取决于电机实际转速。换向期间，PWM 输出配置需要同步更新。KV10x 器件的 FlexTimer 支持输出控制寄存器同步更新，此外还支持 FTM 计数器同时初始化。

下列 FlexTimer 输出寄存器配置为在应用中同步更新：

- 软件输出控制寄存器(SWOCTRL) – 该寄存器使能通道(n)输出的软件控制，并定义通道(n)的输出值。
- 输出屏蔽寄存器(OUTMASK) – 该寄存器禁用或使能通道(n)输出引脚上的输出 PWM 生成。
- 逆变控制寄存器(INVCTRL) – 该寄存器交换一个 PWM 对中的顶部和底部通道。

在本应用软件中，仅 OUTMASK 和 INVCTRL 寄存器用于执行换向。SWOCTRL 寄存器适用于各类 PWM 开关（例如单极 PWM 型）。

应用软件使用数组变量 *bldcCommutationTableComp[8]*，它指向一个表，该表包含施加于 FlexTimer 输出的不同向量及对应的 OUTMASK 和 INVCTRL 寄存器预定义值。详情参见表 2：

表 2. 换向表

OUTMASK	INVCTRL	输出	A 相	B 相	C 相
0x30	0x02	向量 0	正常	反相	关
0x0C	0x04	向量 1	正常	关	反相
0x03	0x04	向量 2	关	正常	反相
0x30	0x01	向量 3	反相	正常	关
0x0C	0x01	向量 4	反相	关	正常
0x03	0x02	向量 5	关	反相	正常
0x00	0x06	对齐向量	正常	反相	反相
0x3F	0x00	PWM 关	关	关	关

下面是 FlexTimer 输出配置和同步的代码示例：

```
FTM0->INVCTRL = bldcCommutationTableComp[sector].swap;
FTM0->OUTMASK = bldcCommutationTableComp[sector].mask;
FTM0->SYNC |= FTM_SYNC_SWSYNC_MASK;
```

为执行换向，FlexTimer 输出控制寄存器（OUTMASK、INVCTRL 和 SWOCTRL）分别加载对应的值，然后将 SYNC 寄存器中的 SWSYNC 位置 1。在使用当前 FlexTimer 配置的应用中，OUTMASK、INVCTRL 和 SWOCTRL 寄存器将同时起作用，FlexTimer 计数器将加载初始化值 (CNTIN)。使用这种同步时，CNTIN、MOD 和 CnV 寄存器不受影响，因为它们是利用 PWMLOAD 寄存器中的 LDOK 位，从写入缓存中加载。

4 用户接口

该应用可利用两个接口来控制：

- TWR-KV10Z32 板上的上/下按钮
- 使用 PC 上运行的 FreeMASTER 进行远程控制，PC 通过 mini-USB 端口连接到 TWR-KV10Z32

4.1 上/下按钮控制

TWR-MC-LV3PH 接通电源之后，电机即可运行。

- 按向上按钮(SW1)，转速提高 500 rpm。电机如果不在旋转，则开始顺时针旋转；如果实际旋转方向为逆时针，则降低转速。
- 按向下按钮(SW2)，转速降低 500 rpm。电机如果不在旋转，则开始逆时针旋转；如果实际旋转方向为顺时针，则降低转速。
- 继续按按钮时，转速在-5000 到 5000 rpm 的限值范围内提高或降低。

- 若同时按住两个按钮 2 秒以上，则开启演示模式（若已开启，则关闭演示模式）。

4.2 使用 FreeMASTER 远程控制

通过 USB 接口利用 FreeMASTER 软件可提供远程操作。为了正确操作 FreeMASTER，请执行如下步骤：

1. 打开 FreeMASTER，转到“项目->选项->通信” (Project->Options->Comm)，将“通信接口” (communication via) 设置为“直接 RS232” (Direct RS232)。
2. 选择 TWR-KV10Z32 板所连接的虚拟 COM 端口；参见“系统属性->设备管理器->端口”，查找“OpenSDA - CDC 串行端口 (<http://www.pemicro.com/opensda>) (COMxx)”。通信速度设置为 9600 bps。
3. 下一步是点击通信按钮（红色 STOP 按钮）。然后，右下角应出现 RS232;COMxx;9600，表示通信已建立。
4. 若没有，点击(STOP)停止通信，拔除再插上 FreeMASTER USB 线缆。然后点击 (START)通信按钮。

启动应用并完成所有必要的设置之后，点击主窗口中的“应用控制” (App Control) 页面（电机控制应用调整工具），如图 5 所示。在该视图中，用于应用状态、转速、PI 控制器和斜坡设置的变量显示在下方窗口；最重要的变量和设置以图形方式显示。利用“应用开关” (App switch) 按钮或在变量列表窗口中选择“Application Switch”变量的状态，可以开启或关闭应用。既可以输入所需的转速（范围是-5000 到 5000），也可以点击转速表来设置所需的转速。直流母线电流限值也可以通过点击安培表来设置。如果检测到故障，必须向“Fault”变量输入 0 值来手动清除故障，然后可以再次开启应用。



图 5. FreeMASTER 项目页面

5 硬件

本应用需要下列硬件模块才能正常工作：

- TWR-KV10Z32
- TWR-MC-LV3PH
- TWR-Elevator

5.1 TWR-MC-LV3PH

三相低压电机控制板(TWR - MC - LV3PH)是一个外围塔式系统模块，支持不同的 Tower 开发平台。它提供相电压和电流反馈信号，以便通过各种算法来控制三相 PMSM 和 BLDC 电机。

MC33937 前置驱动器为控制板提供多种高级保护（过流、过压、过温及其他保护）。

TWRMCLV3PH 模块具有如下特性：

- 12-24 V DC 电源电压输入，最高可扩展至 50 V
- 输出电流高达 8 A
- 电源反接保护电路
- 三相桥式逆变器（6 个 MOSFET）
- 带过流和过压保护的三相 MOSFET 栅极驱动器
- 三相和直流母线电流检测分流器
- 直流母线电压检测
- 三相反电动势电压检测电路
- 低压板载电源
- 编码器/霍尔传感器检测电路
- 电机电源和信号连接器
- 用户 LED、上电 LED、6 个 PWM LED 二极管

BLDC 应用的跳线设置如表 3 所示：

表 3. TWR-MC-LV3PH 跳线设置

跳线	位置
J2	1-2
J3	1-2
J10	2-3
J11	2-3
J12	2-3
J13	2-3
J14	1-2

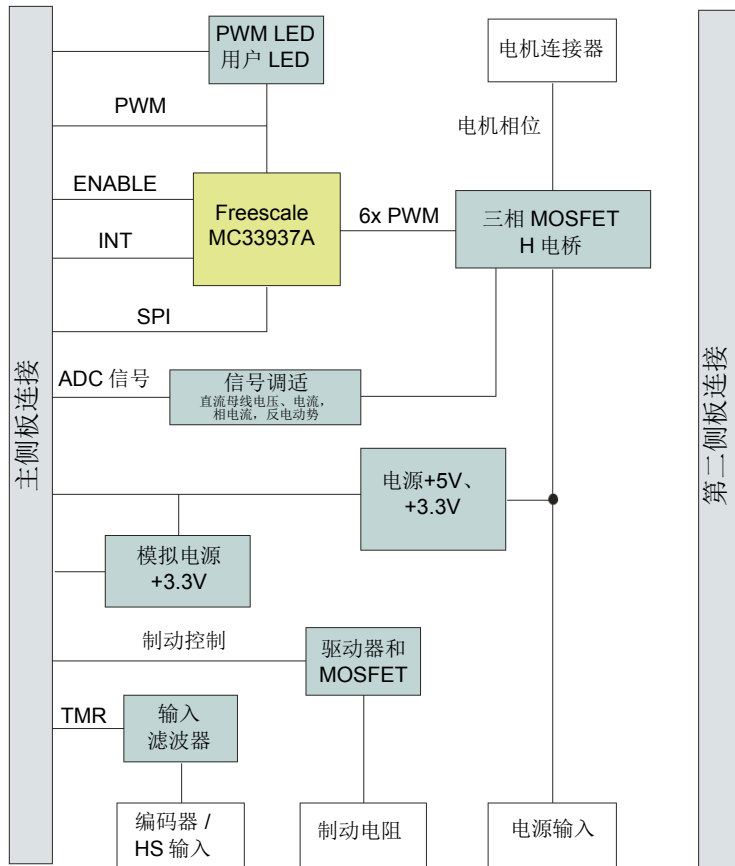


图 6. TWR-MC-LV3PH 框图

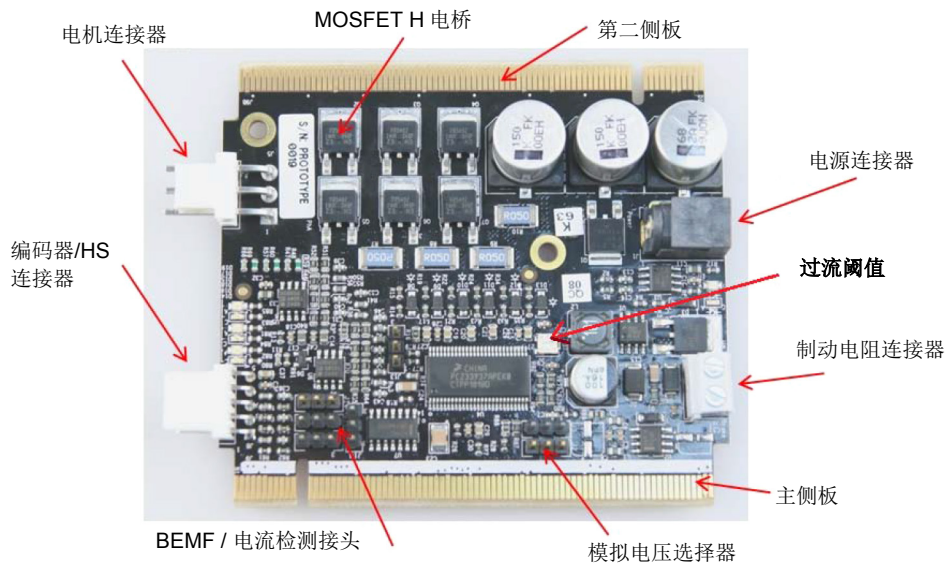


图 7. TWR - MC - LV3PH 图片

5.2 TWR-KV10Z32

TWR-KV10Z32 微控制器模块是 Freescale 塔式系统的一部分，该塔式系统是一个模块式开发平台，通过可重新配置的硬件来支持快速原型开发和工具重用。

- 3.3 V USB 电源稳压器
- 数字加速度计
- 8 个用户 LED
- 2 个用户按钮、1 个复位按钮
- 4 个热敏电阻
- 开放式 SDA 接口
- 电机控制连接器
- Tower 信号兼容 TWR-MC-LV3PH 功率级板
- 10 MHz 晶振

BLDC 应用的跳线设置如表 4 所示：

表 4. TWR-KV10Z32 跳线设置

跳线	位置	跳线	位置	跳线	位置
J1	2-3	J10	1-2	J21	3-4
J2	短路	J11	开路	J22	3-4
J3	2-3	J12	开路	J25	开路
J4	短路	J13	开路	J26	短路
J5	短路	J14	开路	J27	短路
J7	1-2	J18	2-3	J28	短路
J8	1-2	J19	2-3	J29	1-2
J9	1-2	J20	2-3	-	-

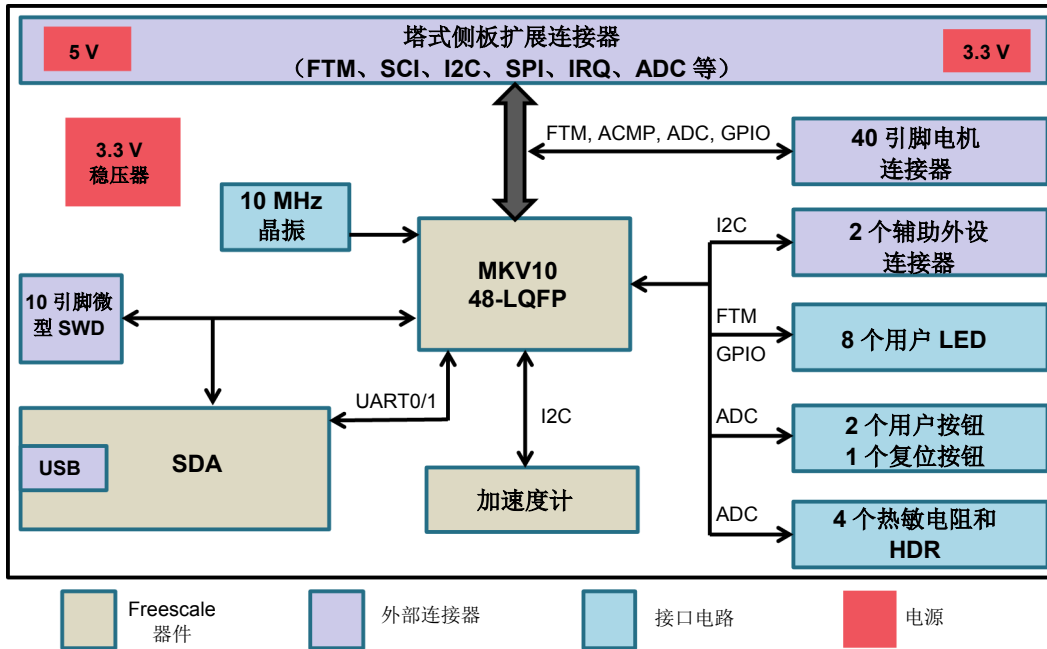


图 8. TWR-KV10Z32 框图

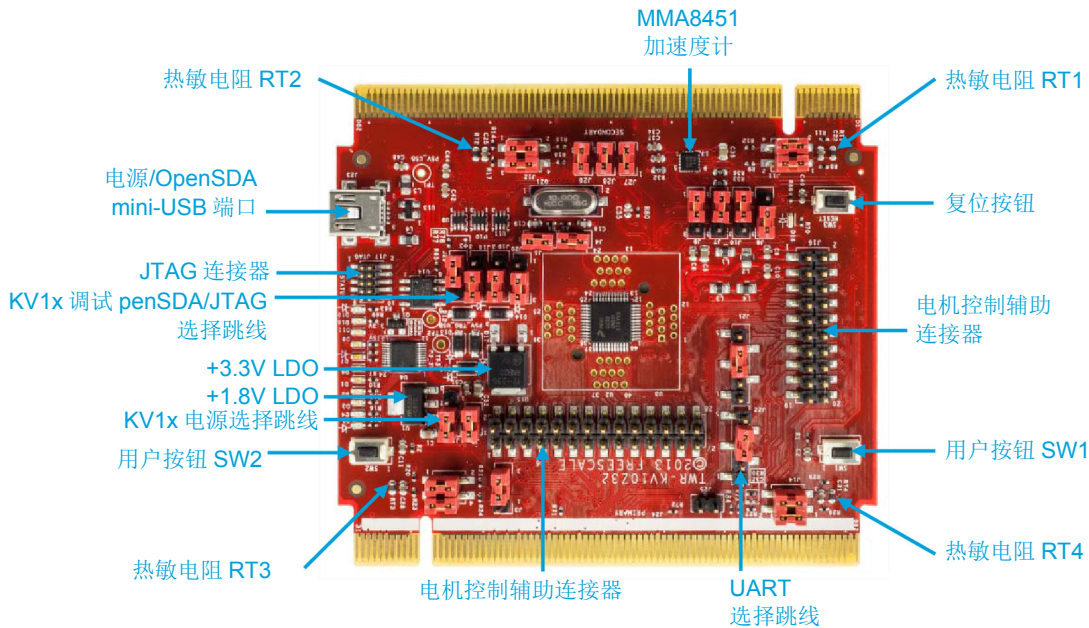


图 9. TWR-KV10Z32 图片

5.3 电机 LINIX 45ZWN24-40

BLDC 无传感器应用采用表 5 所示的电机。该应用也可以使用其他电机，只需定义并更改电机相关的参数即可。

表 5. 电机参数

特性	符号	值	单位
额定电压	Vt	24	V
Vt 时的额定转速	-	4000	RPM
额定扭矩	T	0.0924	Nm
额定功率	P	40	W
连续电流	Ics	2.34	A
极对数	PP	2	-

5.4 微控制器存储器使用

表 6 给出了无传感器 BLDC 电机控制应用（包括 FreeMASTER 工具和演示代码）所用的存储器资源。

表 6. 存储器使用

代码类型	Flash [字节]	RAM [字节]
应用	6212	239
演示代码	836	20
FreeMASTER	2528	2241
总计	9576	2500

6 缩略语

表 7. 缩略语

术语	意义
ACIM	交流感应电机
ADC	模数转换器
BEMF	反电动势
BLDC	无刷直流电机
CCW	逆时针方向
CMP	比较器
CPU	中央处理单元
CW	顺时针方向
DAC	数模转换器
DC	直流
DMA	直接存储器访问模块
DRM	设计参考手册

术语	意义
FOC	磁场定向控制
FTM	FlexTimer 模块
GPIO	通用输入/输出
I/O	计算机系统与外界之间的输入/输出接口：CPU 读取输入以检测外部信号的电平，写入输出以改变外部信号的电平。
I²C	内部集成电路接口
ISR	中断服务例程
KV10x	Freescale 32 位 ARM 子系列微控制器
MCU	微控制器
PDB	可编程延迟区块模块
PWM	脉冲宽度调制
RPM	每分钟转数
UART	通用异步收发器
SPI	串行外设接口模块

7 参考文献

下列参考文献可在 freescale.com 上找到：

1. *K10 子系列参考手册*, KV10P48M75RM, Freescale Semiconductor, Inc., 2013
2. *利用 MC9S08AW60 实现三相 BLDC 电机无传感器控制*, DRM086, Freescale Semiconductor, Inc., 2005
3. *利用 MC9S08MP16 实现三相 BLDC 电机无传感器控制*, DRM117, Freescale Semiconductor, Inc., 2009

8 修订历史

表 8. 修订历史

版本号	日期	重要变更
0	02/2014	初始版

联系我们：

主页：
freescale.com

Web 支持：
freescale.com/support

本文档中的信息仅是为了让系统和软件实施者能够使用 Freescale 产品而提供。本文档并未授予任何明示或默示的许可权以根据本文档中的信息来设计或制造任何集成电路。

Freescale 保留更改本文档所述任何产品的权利，恕不另行通知。Freescale 不保证其产品适合任何特定用途，不承担任何因为应用或使用任何产品或电路而引起的责任，明确否认任何及所有责任，包括但不限于附带或间接损害赔偿。Freescale 数据手册和/或技术规格中可能会提供“典型值”参数，这些参数因应用而异，实际性能可能会随时间而改变。所有工作参数，包括“典型值”，都必须由客户的技术专家针对各种具体应用进行验证。Freescale 并未让与其专利权下的许可权或其它权利。Freescale 依据标准销售条款和条件销售产品，该等条款和条件详见如下地址：freescale.com/SalesTermsandConditions。

Freescale、Freescale 徽标和 Kinetis 是 Freescale Semiconductor, Inc.的商标，已在美国专利商标局注册。Tower 是 Freescale Semiconductor, Inc.的商标。所有其它产品或服务名称属于其各自所有者。ARM 和 Cortex 是 ARM Limited（或其下属公司）在欧盟和/或其他地方的注册商标。保留所有权利。

© 2014 Freescale Semiconductor, Inc.

文档编号：AN4862
修订版 0，2014 年 2 月

