

1 简介

本应用笔记介绍了基于 NXP 的 S08PB16 处理器的带霍尔传感器的三相 BLDC 电机(无刷直流电机)控制。

S08PB16 芯片集成了 12 位 ADC, 模拟比较器(ACMP), 运算放大器(OPAMP), 故障检测及关断(FDS)和 FTM 等重要的特性用于简化设计和降低系统的成本。BLDC 电机因其高效率、高可靠性、高功率密度的特性广泛应用于工业控制领域。

S08PB16 芯片为三相直流电机控制的应用进行了优化设计, 特别适合一些具有严格成本控制的低端应用。例如, 冷却风扇和水泵等。

本应用文档主要介绍了带霍尔传感器的 BLDC 电机六步换相控制的原理以及硬件和软件的实现, 其中详细的描述了外设和驱动的配置。

此应用是基于 NXP 提供的免费的 CodeWarrior 11.1 IDE 进行开发的(请安装补丁: [CodeWarrior MCU 11.1 Service Pack for S08PB and S08PLS](#))。

2 S08PB16 的特点和优势

MC9S08PB16 是基于 S08P 内核的高集成、低功耗和低引脚数量的 8 位微控制器。

有以下特点:

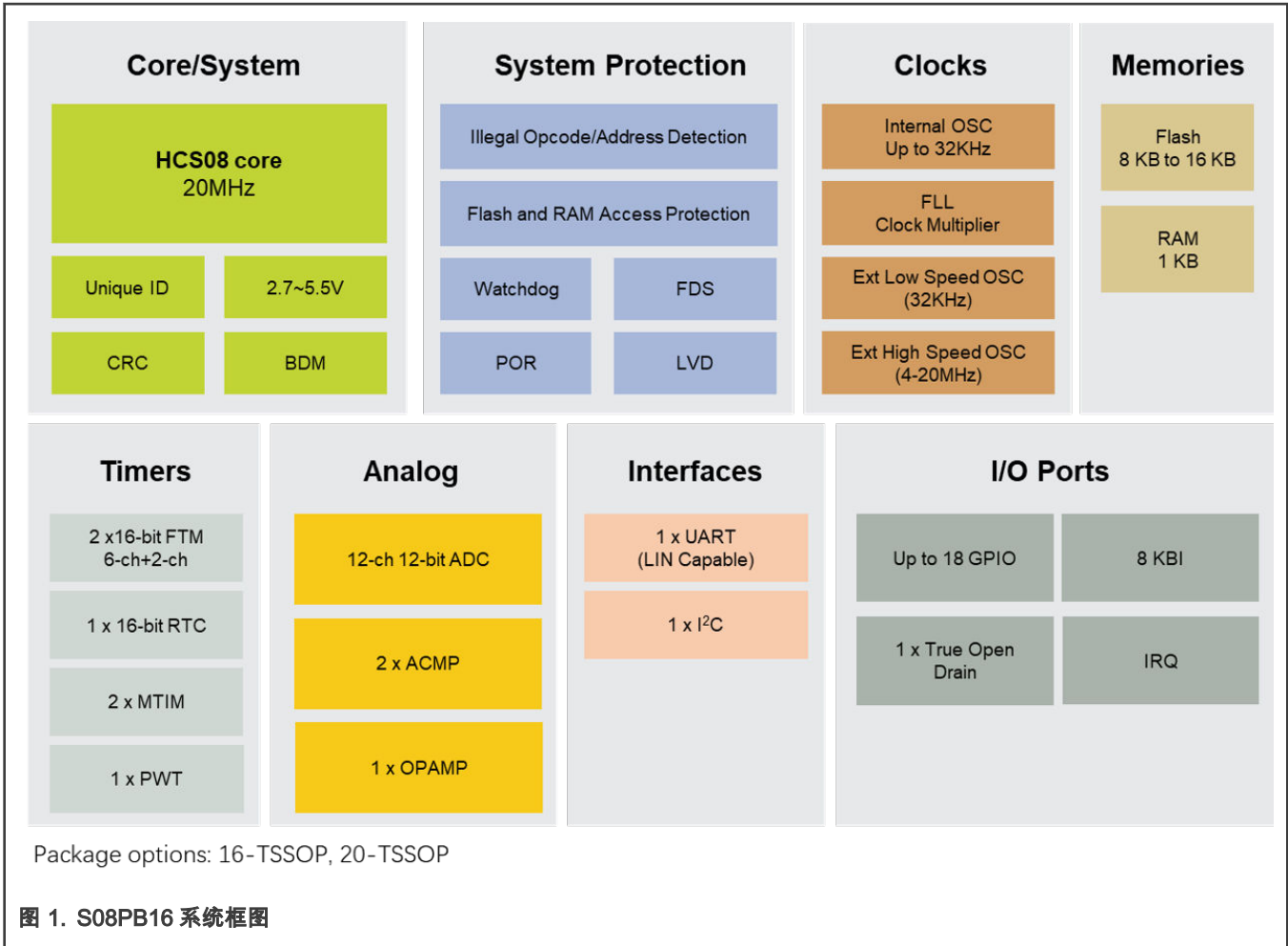
- 最大 CPU 频率可达 20 MHz
- 可扩展的内存, 最高可达 16 KB Flash 和 1 KB RAM
- 在 2.7 V ~ 5.5 V 电压范围内, 支持 Flash 编程/擦除/读取操作的全部功能
- 有两种封装: 20 引脚和 16 引脚
- V 系列的工作环境温度范围在 -40 °C to 105 °C, M 系列的工作环境温度范围在 -40 °C to 125 °C

S08PB16 的系统框图如 [图 1](#) 所示:

目录

1	简介.....	1
2	S08PB16 的特点和优势.....	1
3	BLDC 电机控制原理.....	2
4	硬件和软件的实现.....	6
4.1	系统硬件设计.....	6
4.2	系统软件设计.....	7
5	外围设备配置.....	8
5.1	FTM2.....	8
5.2	ACMP1 and FDS.....	9
5.3	ADC.....	10
5.4	MTIM1.....	10
5.5	IPC.....	10
5.6	KBI.....	11
5.7	PORTA.....	11
6	软件实现.....	11
6.1	主函数流程图.....	11
6.2	MTIM1 中断.....	12
6.3	FTM2 中断.....	17
7	应用指南.....	20
8	参考文献.....	23





S08PB16 用于无刷直流电机控制，有以下优势：

- 有多路 PWM 信号输出。
- 有丰富的计时器和通信接口。
- 内部集成了丰富的模拟 IP，包括 12 通道 12 位精度的 ADC，两个 ACMP 和内部 OPAMP。这些模块的内连可以很方便地用于过流故障保护。S08PB16 的内部运放（OPAMP）节约了外部的运放电路，可以节约 BOM 成本。
- 为了电机控制而优化的增强型的故障保护模块（FDS），可以保证系统的高可靠性。

3 BLDC 电机控制原理

无刷直流电机（BLDC 电机）是一种旋转电机，其定子类似传统感应电动机的三相定子，转子上有安装在表面的永磁体。转子上没有电刷，可在特定的转子位置进行换相。定子通常由硅钢片制成。BLDC 电机的典型截面图如图 2 所示。定子绕组嵌入沟槽中（分布式绕组）。由于气隙磁场是由永磁体产生的，转子磁场的大小是恒定的。

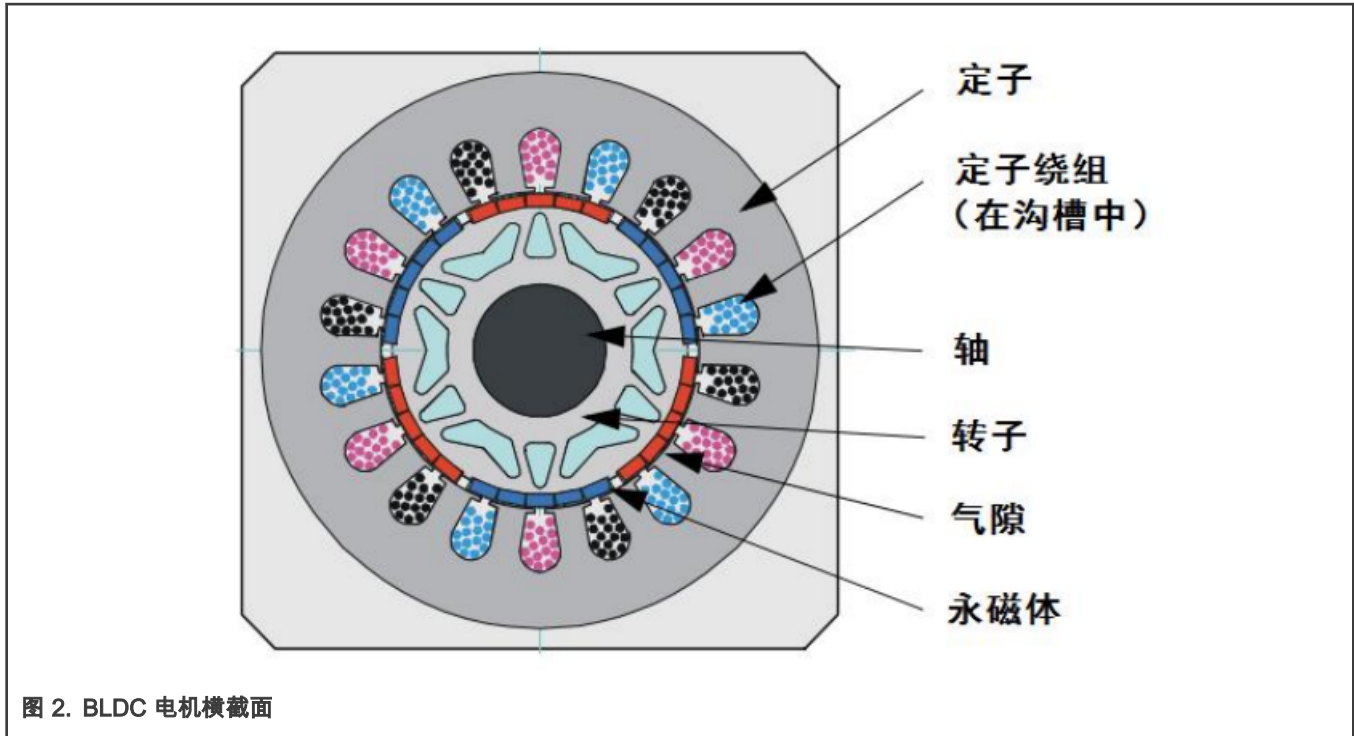


图 2. BLDC 电机横截面

永磁体的磁化强度和放在转子上的位置都需要精确固定，这样反电势（由于转子的转动在定子绕组上感应的电压）的形状才会是梯形的。图 4 显示了矩形形状的直流电压，可以用于创建一个低扭矩脉动的旋转磁场。

BLDC 电机的控制需要一个三相逆变电路，三相桥由六个功率开关器件（Q1-Q6）组成，通过六步换相控制算法来驱动各个开关元器件，如 图 3 所示。

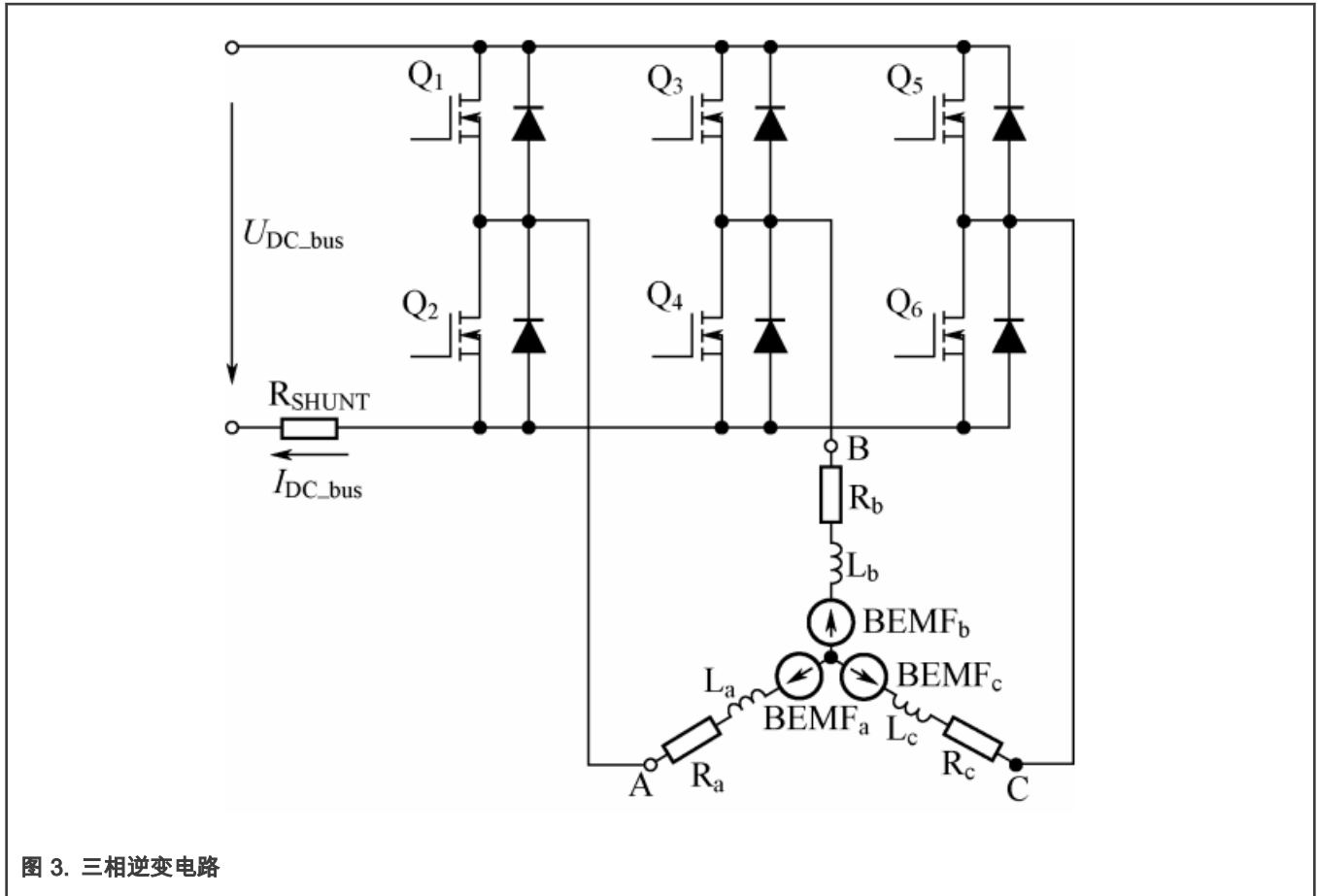
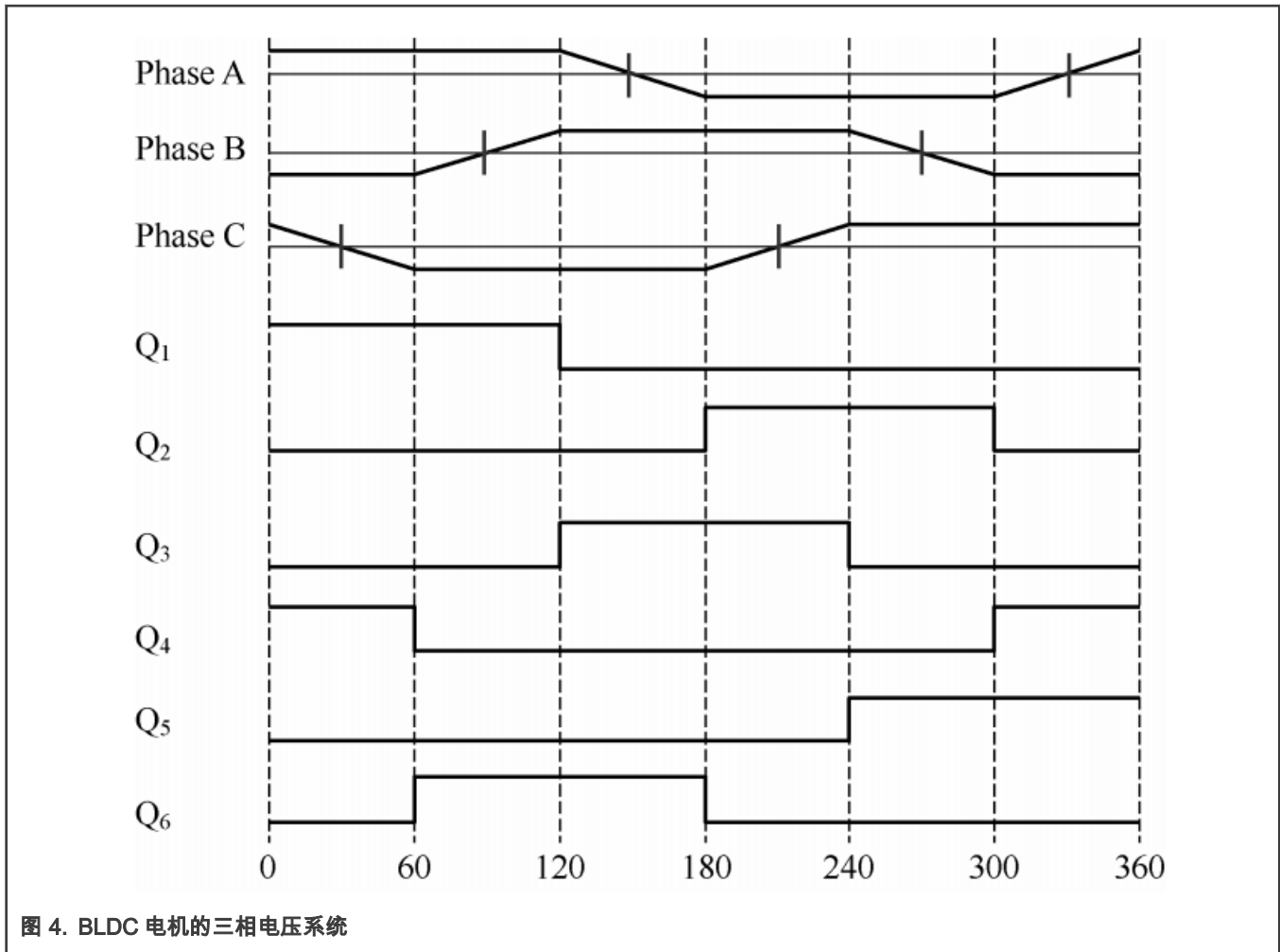
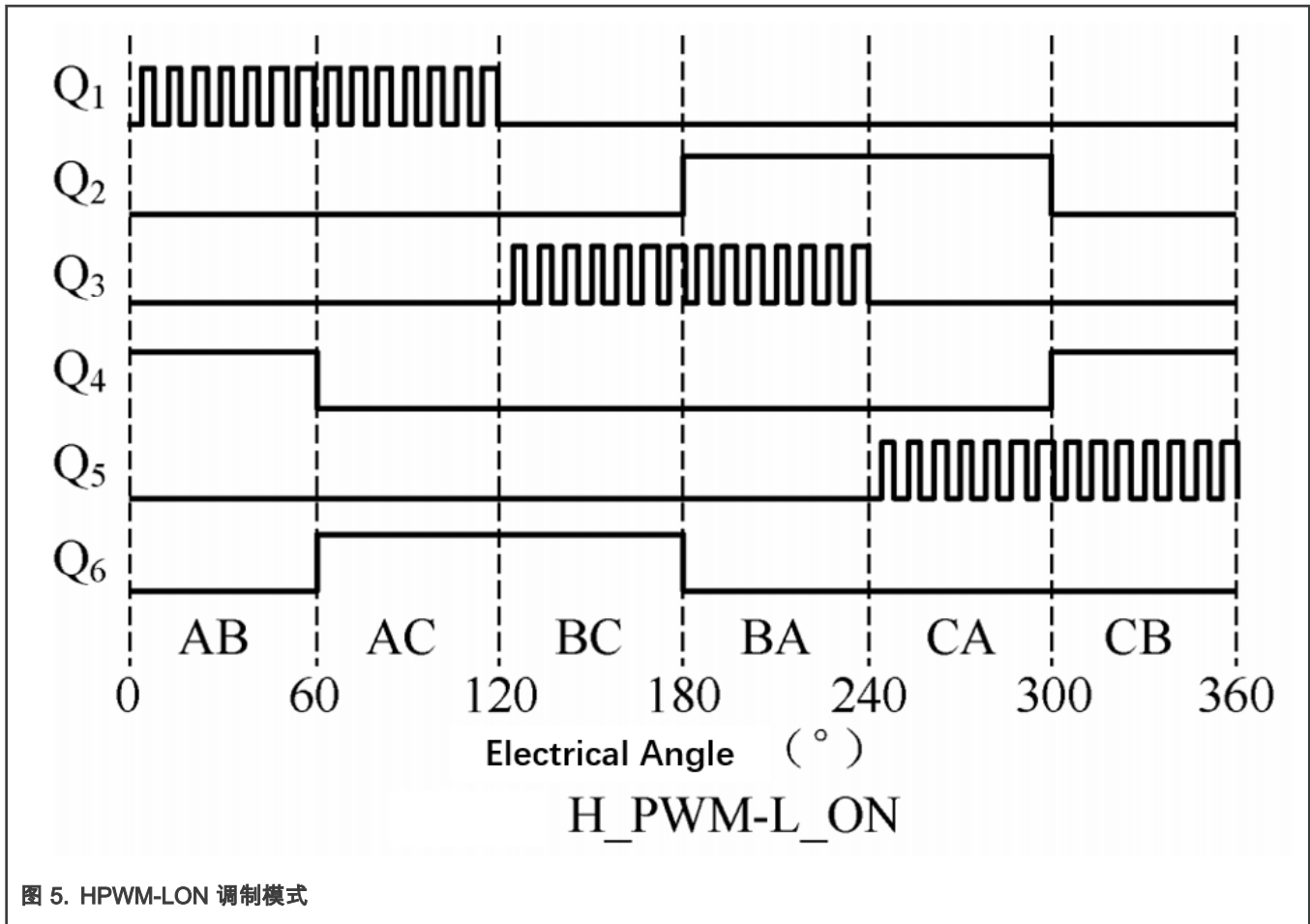


图 3. 三相逆变电路



六步换相控制可使用很多 PWM 调制方式。考虑到 S08PB16 FTM 模块中没有死区插入的功能，所以采用 HPWM_LON 模式调制，如 图 5 所示。



4 硬件和软件的实现

4.1 系统硬件设计

本次应用的硬件主要包括以下部分:

- [S08PB16-EVK](#)

S08PB16-EVK 是用于 NXP S08PB 和 S08PLS 5V 8 位 MCU 的经济有效的开发套件。它虽然很小，但功能强大，有丰富的集成外设，可以用来评估 S08PB 和 S08PLS MCUs 的所有特性。S08PB16-EVK 支持板载 OSBDM，无需外部调试工具，便可进行程序下载和调试。

- [FRDM-MC-LVBLDC](#)

FRDM-MC-LVBLDC，此平台是针对无刷直流电机控制推出的硬件平台，支持霍尔传感器或者无速度传感器的 BLDC 电机控制。

选用 LINIX 45ZVN24-40 无刷直流电机。

电机控制开发平台的框图和实际的演示平台搭建如 [图 6](#) 和 [图 7](#) 所示。

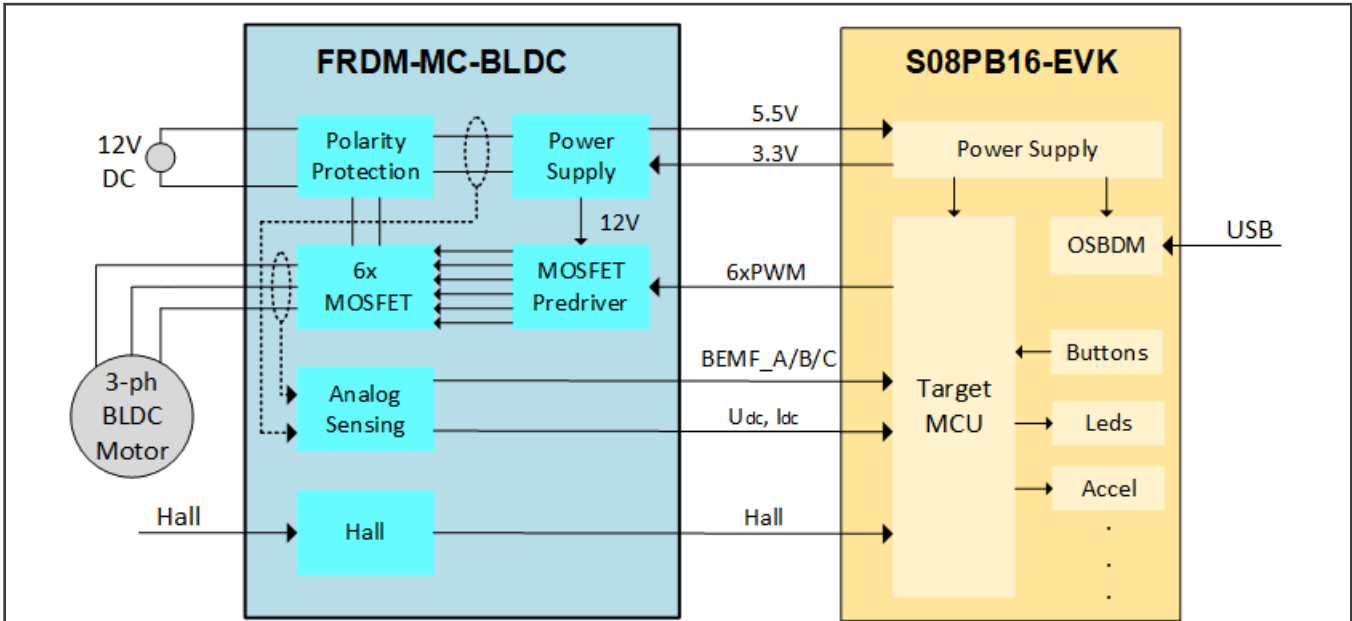


图 6. 电机控制开发平台框图



图 7. 实际的演示平台搭建

4.2 系统软件设计

本次软件和硬件设计，符合以下的设计要求：

- 选择 S08PB16 作为控制器
- 基于霍尔传感器的全速度闭环控制
- 基于软件和硬件实现过压、欠压、过流的故障保护
- 最小转速 300 rpm，最大转速可达 2500 rpm (取决于选用的电机)
- 设置的默认限流值是 4 A
- 支持正反向旋转
- 可从任意位置启动，不需要转子对齐
- SW3 按键控制 demo 模式
- 基于 FreeMASTER 的电机状态的实时监测

系统框图如 图 8 所示。

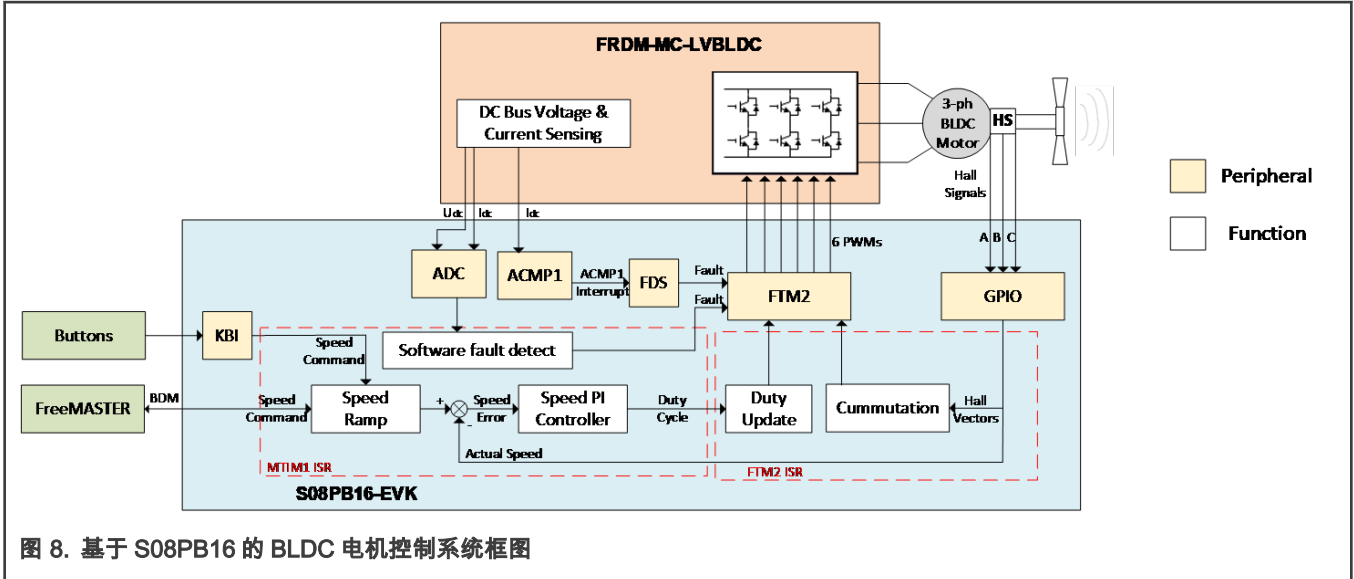


图 8. 基于 S08PB16 的 BLDC 电机控制系统框图

从该框图中，我们可以看到整个控制过程在三个中断中完成。

- FTM2 ISR

霍尔信号采集，换相控制，PWM 占空比更新和实际速度测量均在 FTM2 ISR(10KHz)中完成。电机转子所在的扇区是通过不断查询连接霍尔信号的 GPIO 值确定的。

- MTIM1 ISR

电机状态机的更新，速度 PI 闭环控制，电机软件故障保护等均在 MITM1 ISR(1KHz)中完成。速度 PI 控制器输出 PWM 占空比会在 FTM2 ISR 中更新。电机的软件故障保护系统，是通过 ADC 采样直流母线电流/电压值来实现对过流/过压/欠压故障的检测。

- ACMP1 中断

此方案通过 ACMP1 中断触发系统硬件过流故障，在 ACMP1 中断服务函数中仅处理过流故障相应的标志位。

5 外围设备配置

本节介绍了用于 S08PB16 上的电机控制的外围设备的配置，包括 FTM2，ACMP1 和 FDS，ADC，KBI，MTIM1，IPC，PORTA。

5.1 FTM2

配置 FTM2 输出 6 个 PWM 信号用于驱动 BLDC 电机，使能 FTM2 计数器溢出中断，在中断服务函数中实时读取 HALL 信号以获得转子位置信息，并执行相应的换相操作。

FTM2 配置:

- 系统时钟源
- 运行频率 10 kHz，周期 100 μ s
- 输出边沿对齐且高电平有效的 PWM 脉冲
- 配置 FTM2 的通道 0-5，用于三相驱动 BLDC 电机
- 使能计数器溢出中断

BLDC 电机控制中最重要的是换相。对于 HPWM_LON 调制模式，在任何时候，只有一个 MOSFET 处于 PWM 调制状态。对于 S08PB16，我们可以使用 SYS_SOPT8 和 SYS_SOPT7 方便地控制每个电桥的输出，而不会影响实际的 FTM 时序。以扇区 1 为例(A + B-)：

```

SYS_SOPT8 = SYS_SOPT8_FTM2CH3OCV_MASK;//Q4 active
SYS_SOPT7 = SYS_SOPT7_FTM2CH1OC_MASK | SYS_SOPT7_FTM2CH2OC_MASK | SYS_SOPT7_FTM2CH3OC_MASK |
SYS_SOPT7_FTM2CH4OC_MASK | SYS_SOPT7_FTM2CH5OC_MASK;//Q1 PWM output

```

6 个 PWM 输出也可以被 FDS 模块控制。过流信号作为 FDS 的输入信号，一旦过流信号产生，会触发 FDS 模块，FDS 将会关闭 6 个 PWM 信号并控制这 6 个引脚的输出。产生和控制 PWM 信号的过程如图 9 所示。

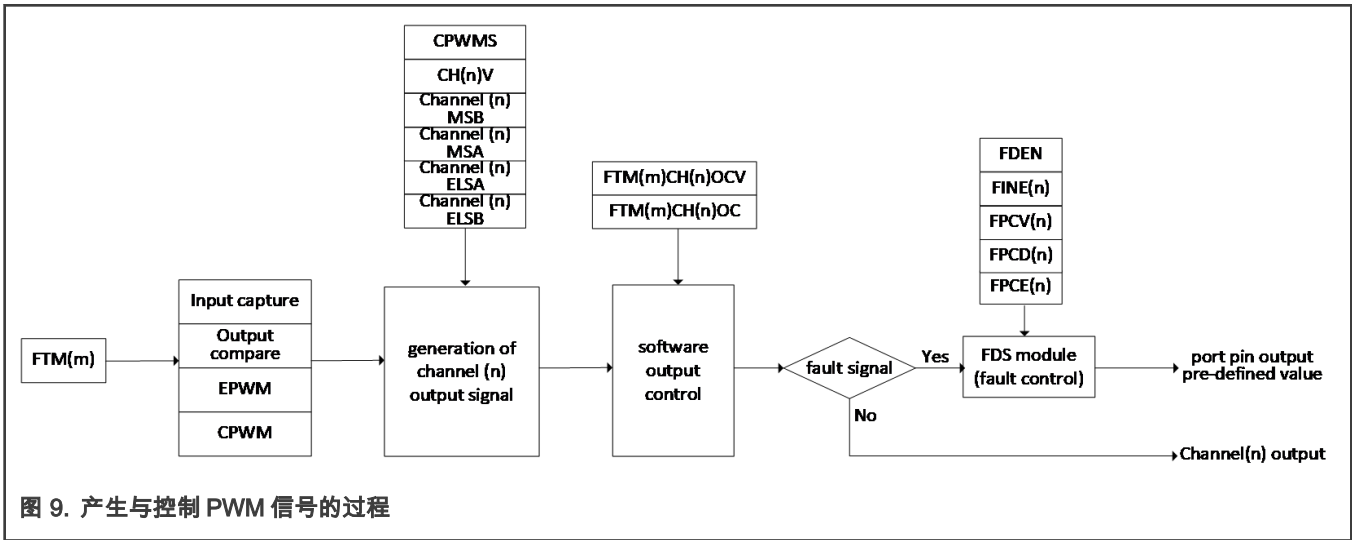


图 9. 产生与控制 PWM 信号的过程

5.2 ACMP1 and FDS

系统硬件过流故障信号是由内部 ACMP1 中断产生的。

外部的运算放大器输出连接到 ACMP1 + 输入，ACMP1 内部 DAC 的输出是 ACMP1- 的输入，ACMP1 可以将 OPAMP 输出信号与内部 DAC 的电压进行比较。同时，我们可以配置 **ACMP1_C1_DACVAL** 来更改 DAC 输出电压以调整限电流值。当 OPAMP 输出大于 DAC 输出值时，ACMP1 中断标志位置位，过流故障信号产生。

FDS 故障输入 2 源 (FIN2) 是 ACMP1 中断输出。一旦出现过流故障信号，FDS 就会关闭 6 个 PWM 通道的输出，并将相应的端口引脚设置为输出 0。有关内部 OPAMP，ACMP 和 FDS 的详细信息，请参考 *How to Use the Interconnection of OPAMP, ACMP1 and FDS for S08PB16* (document [AN12836](#))。

ACMP1 配置：

- ACMP1+ 输入源：PTA3/ACMP1IN0/OPAMP+，PTA3 引脚用于连接片外运放的输出
- ACMP1- 输入源：DAC 输出
- ACMP1 输出使能，判断故障源是否清除；如果故障源移除，则清除 FDS 中断标志位
- ACMP1 中断使能，ACMP1 输出上升边沿触发中断

FDS 配置：

- FDS 输入通 2 使能 (FIN2)，ACMP1 中断输出作为 FDS 输入源
- 配置 FDS FDSOUT0-FDSOUT5 输出通道：使能相对应的 6 个引脚配置，将 6 个引脚的数据方向配置为输出，6 个引脚的输出值设为 0

ACMP1 和 FDS 的内部连接图如图 10 所示。

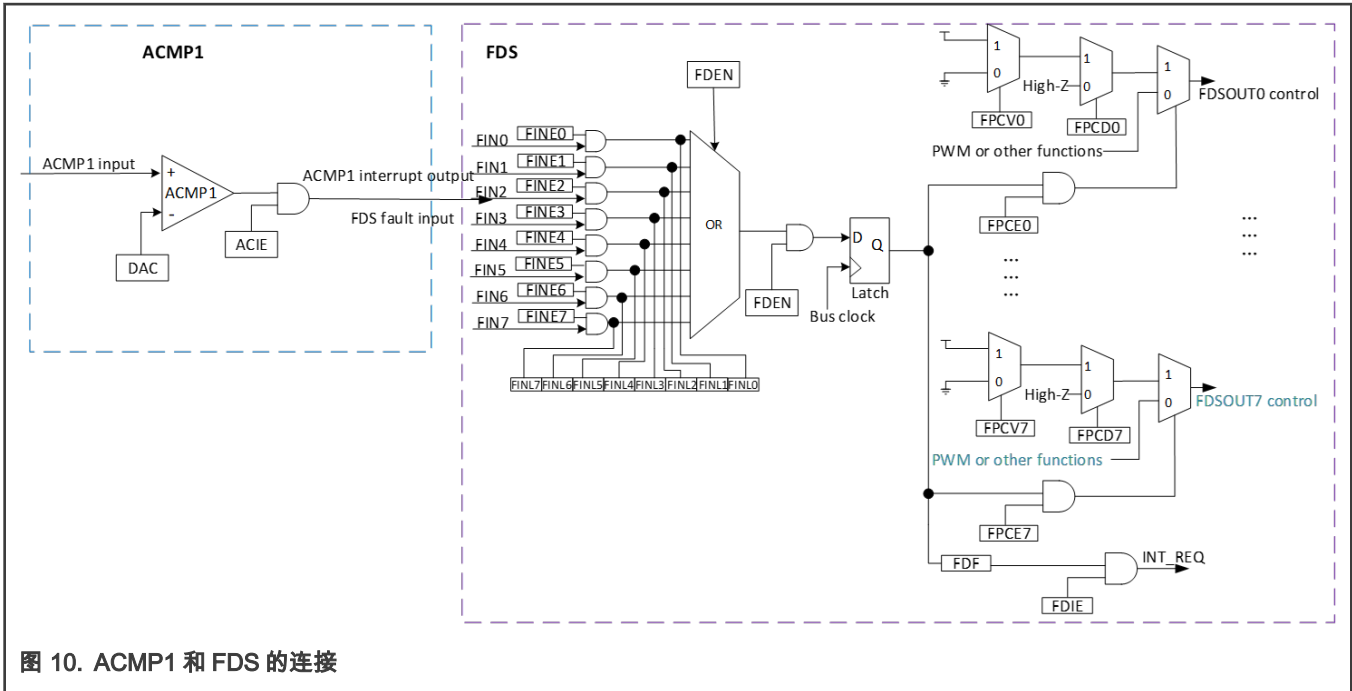


图 10. ACMP1 和 FDS 的连接

- 在 FRDM-MC-LVBLDC 板上已经有一个外部运放，在本文的应用中，不使用 S08PB16 的内部运放采样母线电流。在实际应用中，客户可根据需要选择使用外部运放或者 S08PB16 的内部运放进行采样。

5.3 ADC

ADC 采样直流母线电压和直流母线电流，采样值用于和用户给定的过流值、过压值、欠压值进行比较，实现电机控制系统的软件保护。

ADC 配置：

- 总线时钟，时钟分频值为 1
- 12 位采样精度，长采样时间
- 配置两个采样通道，PTA3/ADP3 通道用于采样直流母线电流，PTB3/ADP7 通道用于采样直流母线电压

5.4 MTIM1

使用 MTIM1 模块产生 1ms 的中断用于实现状态机检测，速度环的控制等。

MTIM1 配置：

- 选用固定频率时钟源
- 没有分频，modulo 的值为 16, MTIM1 时钟频率配置成 1 kHz.
- 使能中断

5.5 IPC

中断优先级控制器 (IPC) 模块用于配 FTM2 和 MTIM1 的中断优先级来实现中断嵌套。在本文 BLDC 电机控制应用的代码中，需要配置 FTM2 中断优先级高于 MTIM1 中断优先级。

MTIM1 的中断优先等级设为 0, FTM2 的中断优先等级设为 3. FTM2 的中断请求可以抢占 MTIM1 的中断服务，而执行 FTM2 中断服务程序时，MTIM1 的中断请求将会被阻止。

IPC 配置：

- 使能中断优先级控制器

- FTM2 中断优先级设为 3, MTIM1 中断优先级设为 0

5.6 KBI

手动按键（板载 SW3）用于控制 demo 模式的打开与关断。根据 S08PB16-EVK 的原理图可知，PTB2/KBI0P6 被配置成 SW3 按键的输入通道。

KBI 配置：

- 只检测下降沿
- KBI 中断关闭
- 使能 PTB2 作为 KBI 引脚

5.7 PORTA

三相霍尔传感器信号连接到 PTA0, PTA1, PTA2。转子所在的扇区是通过查询连接霍尔信号的 GPIO 值确定的。

当电机在工作时，转子的位置可以通过不断地读取这 3 个引脚的值来确定。

PORTA 配置：

- 配置 PTA0, PTA1, PTA2 数据方向为输入，用于读取霍尔值

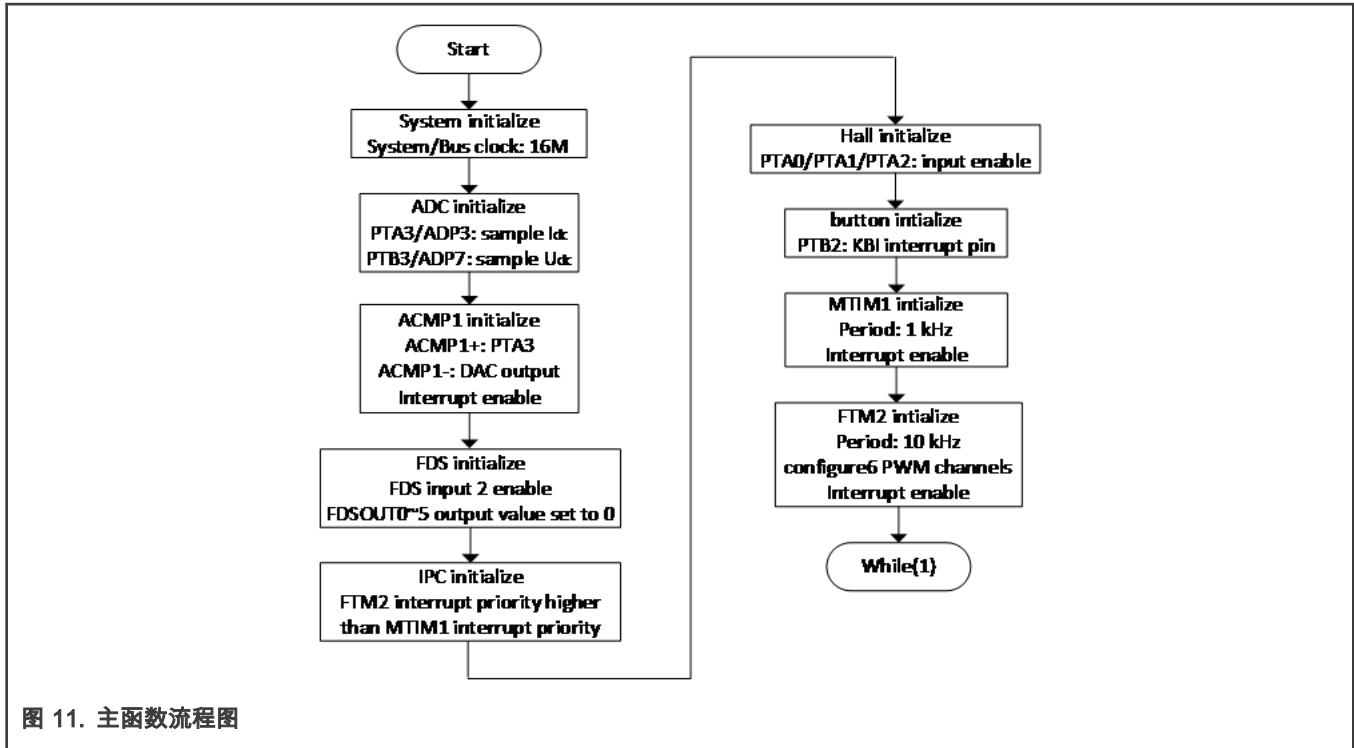
6 软件实现

本节介绍了 BLDC 电机应用的软件设计。对该软件的描述主要包括以下几个部分：

- 主函数流程图
- MTIM1 中断
- FTM2 中断

6.1 主函数流程图

在复位以后，应用程序初始化所有使用的外围设备并进入无限循环，main()函数的流程图如 [图 11](#) 所示。



6.2 MTIM1 中断

MTIM1 中断服务程序详细的执行过程如 图 12 所示。

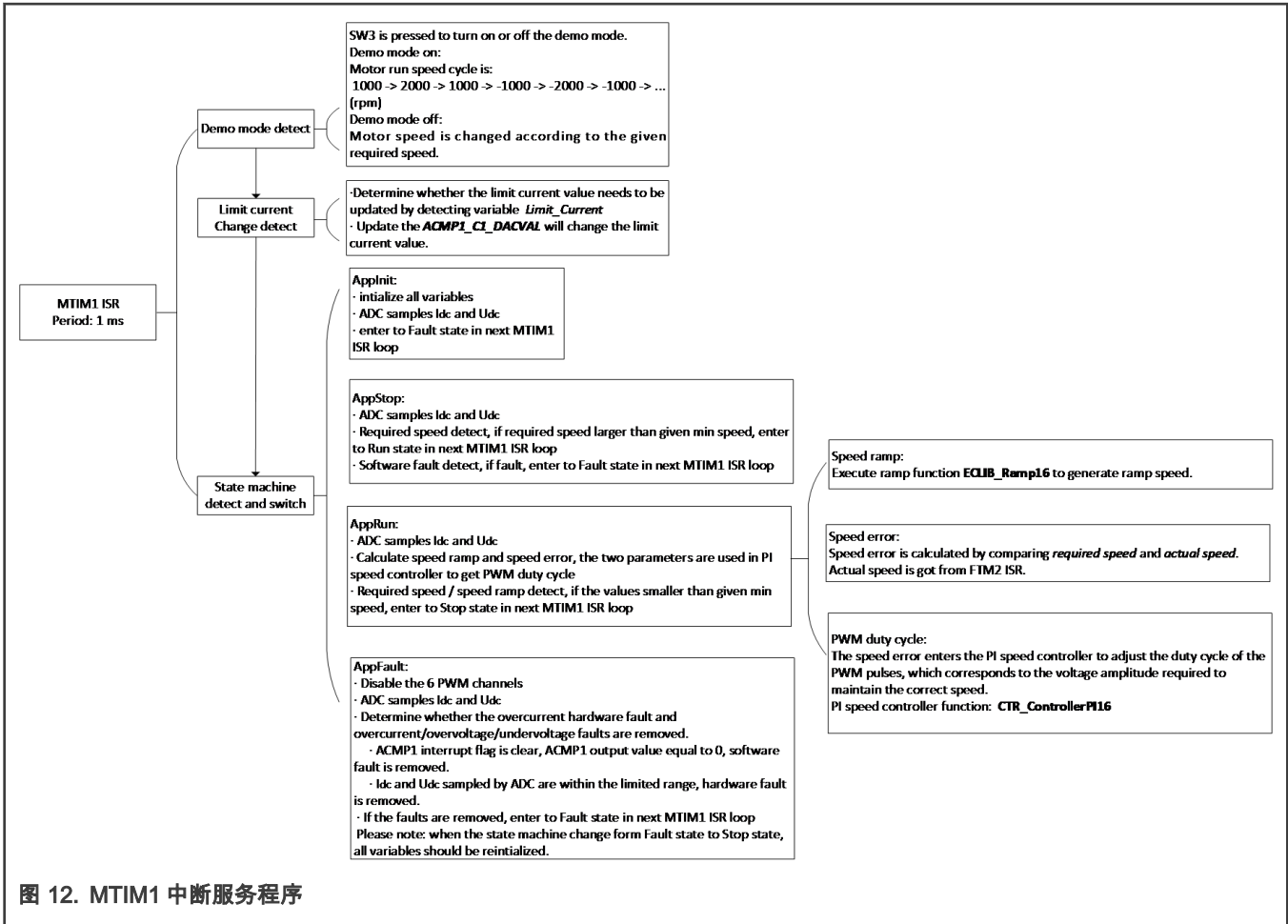
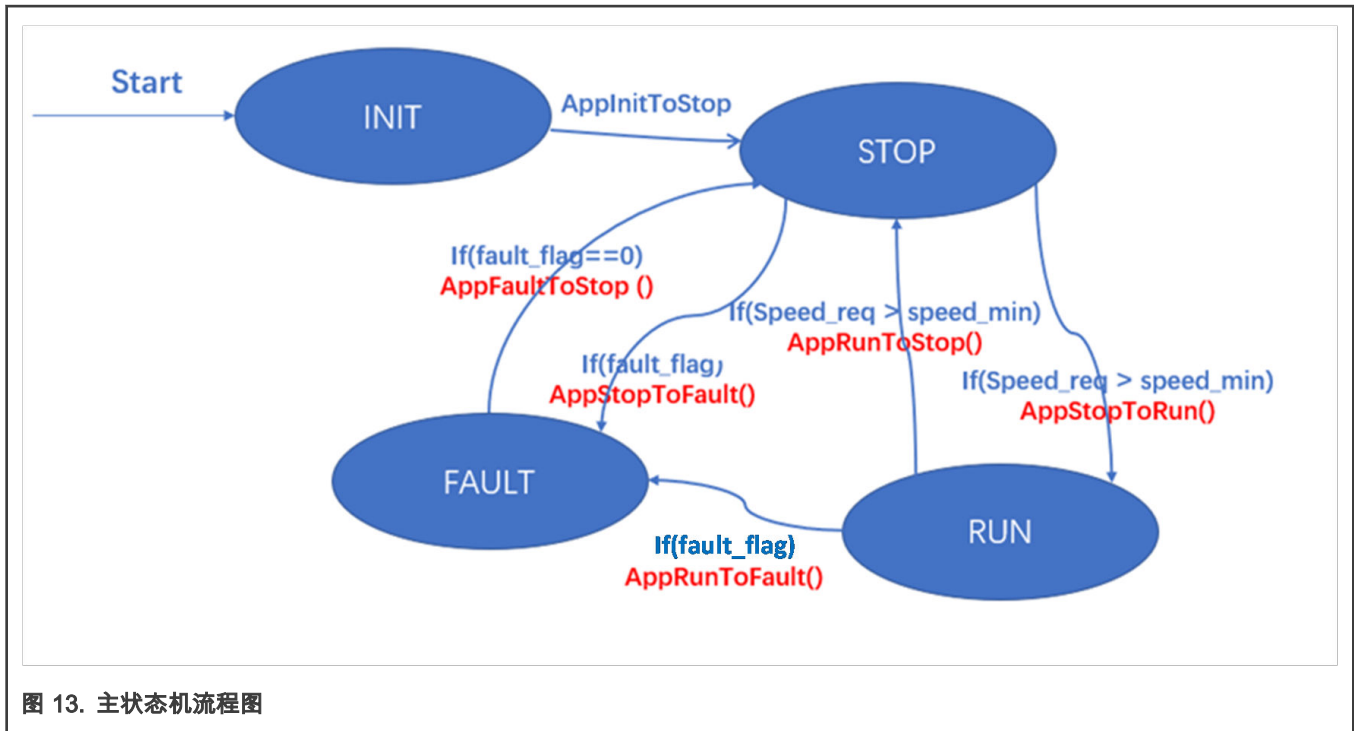


图 12. MTIM1 中断服务程序

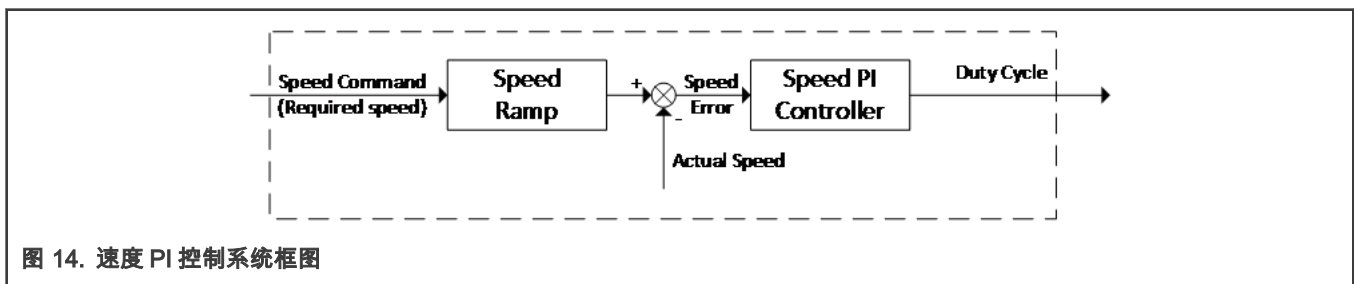
6.2.1 状态机切换

主状态机包括以下几个子状态: init, stop, run, fault。



6.2.2 速度 PI 闭环

根据图 12 可知，速度 PI 控制环在 AppRun 中实现。现在对速度控制过程进行详细的描述，图 14 是速度 PI 控制的系统框图。



1. 速度斜坡

由于整个应用是一个具有大惯量的系统，因此在应用程序中必须改进速度命令，否则系统可能会过载。一种方法是生成斜坡，以通过代码中定义的步长增量使速度斜坡接近速度命令（所需的速率值）。

在本文应用中，ECLIB_Ramp16 需要三个参数来获取实际速度斜坡。

- 预定义增量值(*RAMP_SPEED*)
- 期望转速 (*w16Speed_req*)
- 速度斜坡 (*w16Speed_ramp*)

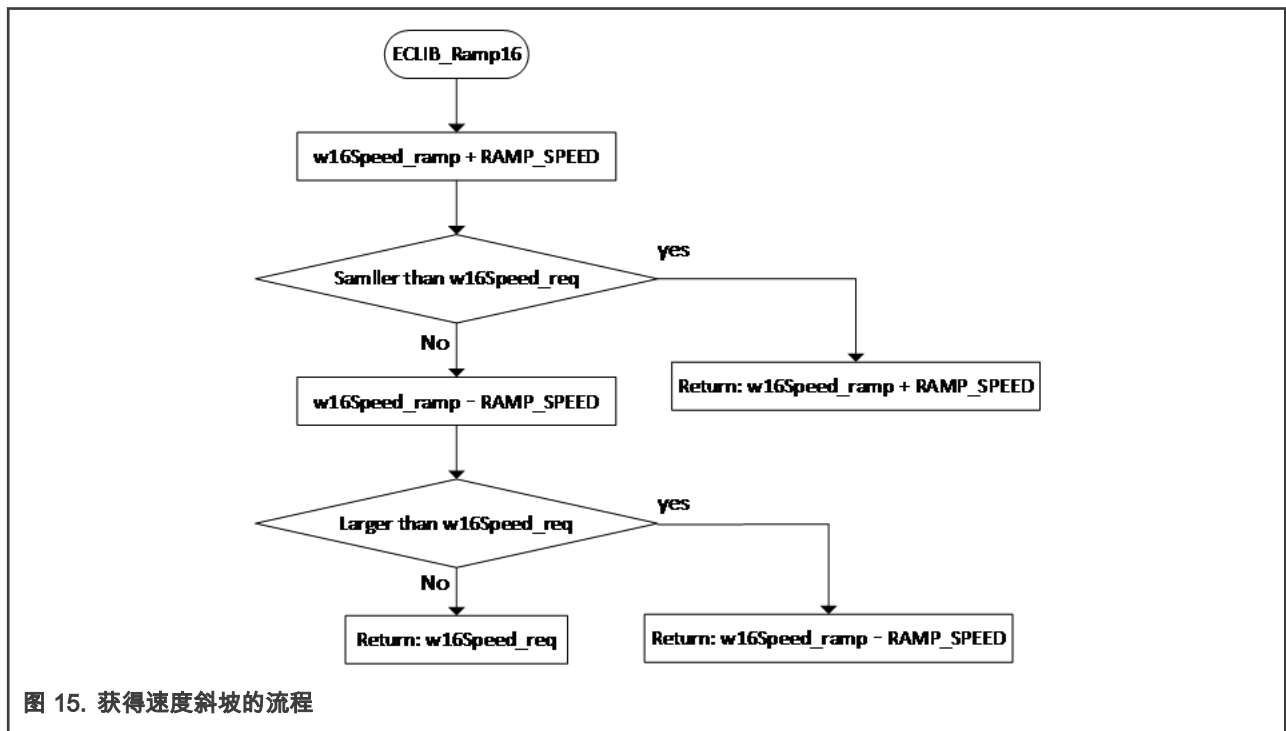
RAMP_SPEED 的定义如下：

```

/* Speed ramp value settings for close loop */
#define SLOW_PERIOD_US    1000.0//us
#define RAMP_SPEED_S      10000.0// (rpm/s)
#define RAMP_SPEED        (RAMP_SPEED_S*SLOW_PERIOD_US*32768.0/1000000.0/N_MAX)
  
```

上面 *RAMP_SPEED* 参数是经过缩放的。关于速度缩放的描述请参考[实际速度测量](#)。

图 15 显示 ECLIB_Ramp16 函数使用定义的步长来递增或递减斜坡速度值，使得斜坡值更接近期望值的过程。



2. PI 控制器

速度 PI 控制算法可以纠正期望速度与实际速度之间的速度误差 ($w16Speed_error = w16Speed_ramp - w16Speed_Act_flt$)。PI 控制器的输出，作为新的施加给电机的电压校正，传递到 PWM 发生器 FTM2。

比例积分 (PI) 算法的时域方程如下所示。

$$u(t) = K_p \cdot e(t) + K_i' \int_0^t e(t) dt$$

式中

- $e(t)$: 时域的输出误差
- $u(t)$: 时域的控制输出
- K_p : 时域的比例系数
- K_i' : 时域的积分系数

离散域方程可以表示为:

$$u(k) = K_p \cdot e(k) + K_i' \cdot T_s \sum_{j=1}^k e(j)$$

式中

$$K_i = K_i' \cdot T_s$$

- T_s : 采样时间
- $e(k)$: 离散域的输出误差

- $u(k)$: 离散域的控制输出
- K_p : 离散域的比例系数
- K_i : 离散域积分系数

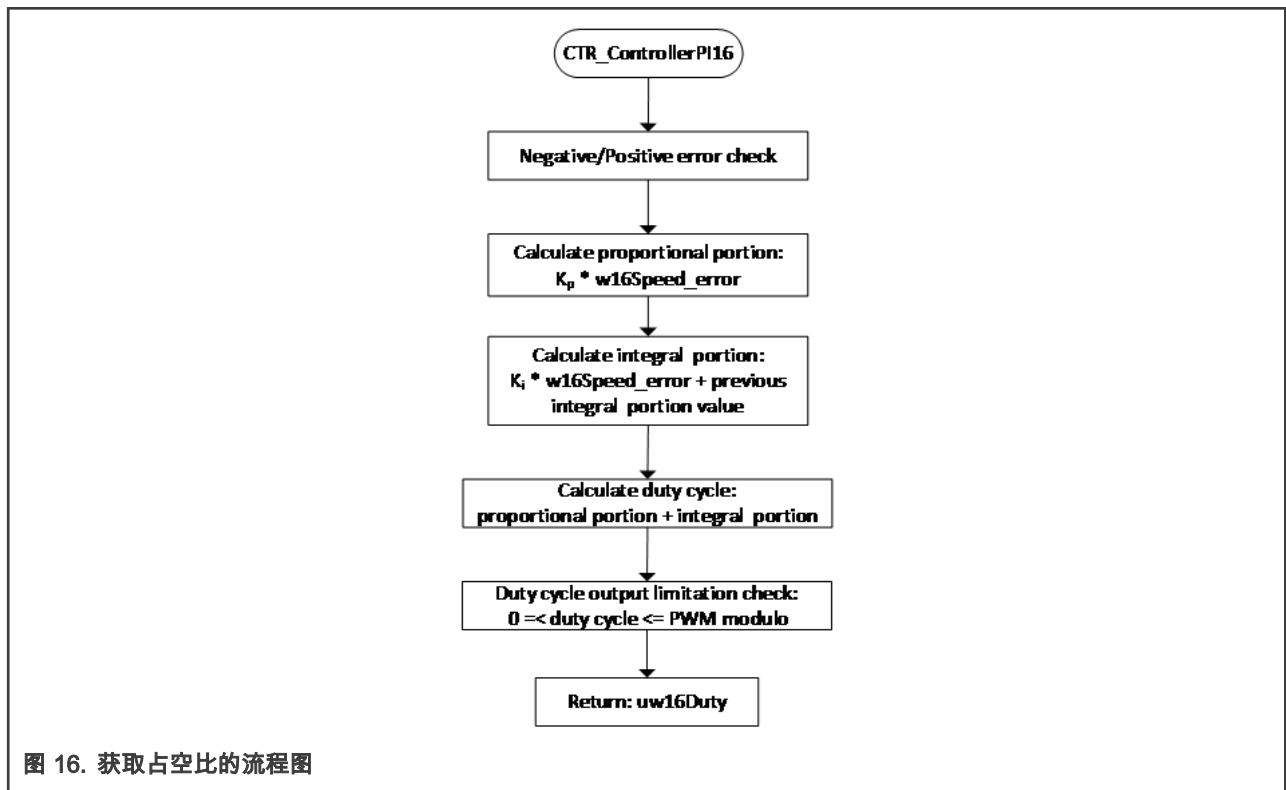
PI 控制器的程序是在 AppRun() 函数中计算的, 该函数每 1 ms 调用一次, 则 PI 控制器的采样时间(T_s)为 1 ms。

PI 控制器的一个输入是速度误差 ($w16Speed_error$), 另一个输入是指向 PI 控制器参数的结构体的指针 $sSpeedPiParams$ (K_p , K_i , ...)。所有这些参数均用于 PI 控制器函数 CTR_ControllerPI16。

CTR_ControllerPI16 函数的输出为 $uw16Duty$, 在 FTM2 ISR 中被用于更新 PWM 占空比。

在本应用中, K_p 设置为 10, K_i 设置为 1。如果更改速度标定或电动机时, 都必须重新配置这些参数。

图 16 显示了通过使用 CTR_ControllerPI16 函数获取占空比的 PI 控制过程。



6.2.3 限值计算

直流母线电流检测电路在 FRDM-DC-LVBLDC 板上, 如 图 17 所示:

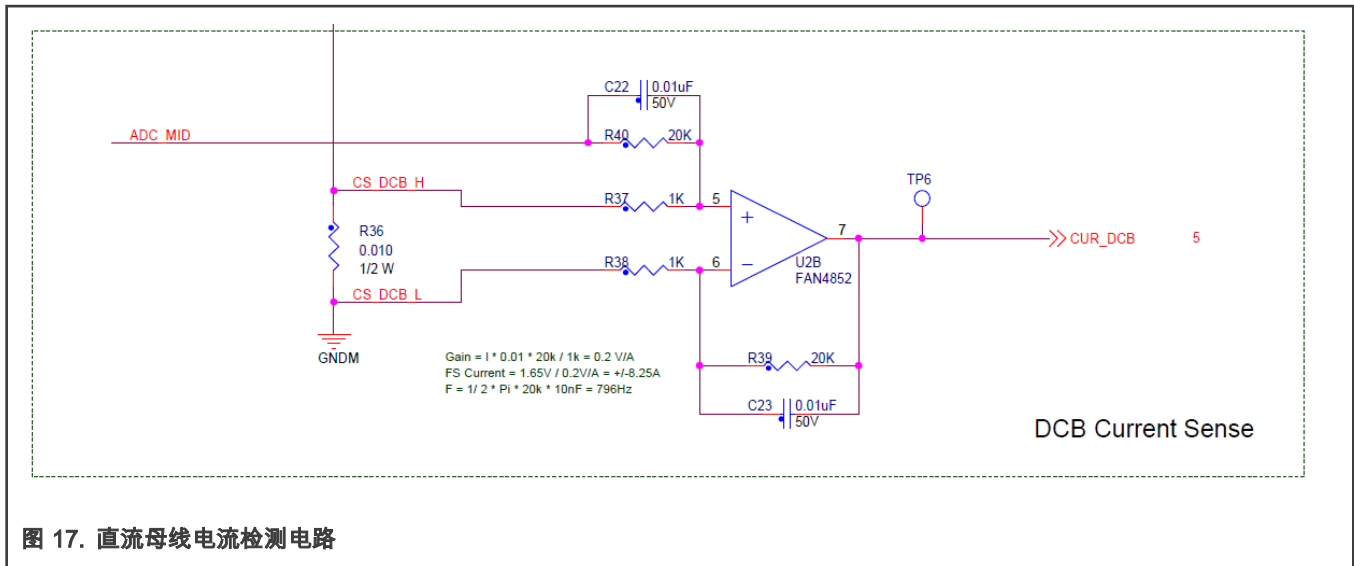


图 17. 直流母线电流检测电路

如 图 17 所示，上面电路的参考电压是 1.65 V，增益是 0.2 V/A，其输出连接到 S08PB16 ACMP1 正端的输入。如果将限流值设为 i ，则满足下面方程：

$$\text{ACMP1 正端输入电压} = 1.65 + 0.2 \times i$$

根据 5.2 章中 ACMP1 的配置可知，DAC 输出， $(\text{ACMP1_C1_DACVAL} + 1) / 64 \times V_{in}$ ，被配置为 ACMP1-的输入，ACMP+的输入应该小于 ACMP-的输入，可以得到下面方程：

$$(\text{ACMP1_C1_DACVAL} + 1) / 64 \times V_{in} > 1.65 + 0.2 \times i$$

式中

- V_{in} ：DAC 的参考电压，本应用中选 VDDA (5V)作为参考电压
- i ：期望的限流值

因此，可以通过根据上述公式更改 ACMP1_C1_DACVAL 的值来更改电流限值。

6.3 FTM2 中断

FTM2 中断服务程序的详细过程如 图 18 所示。

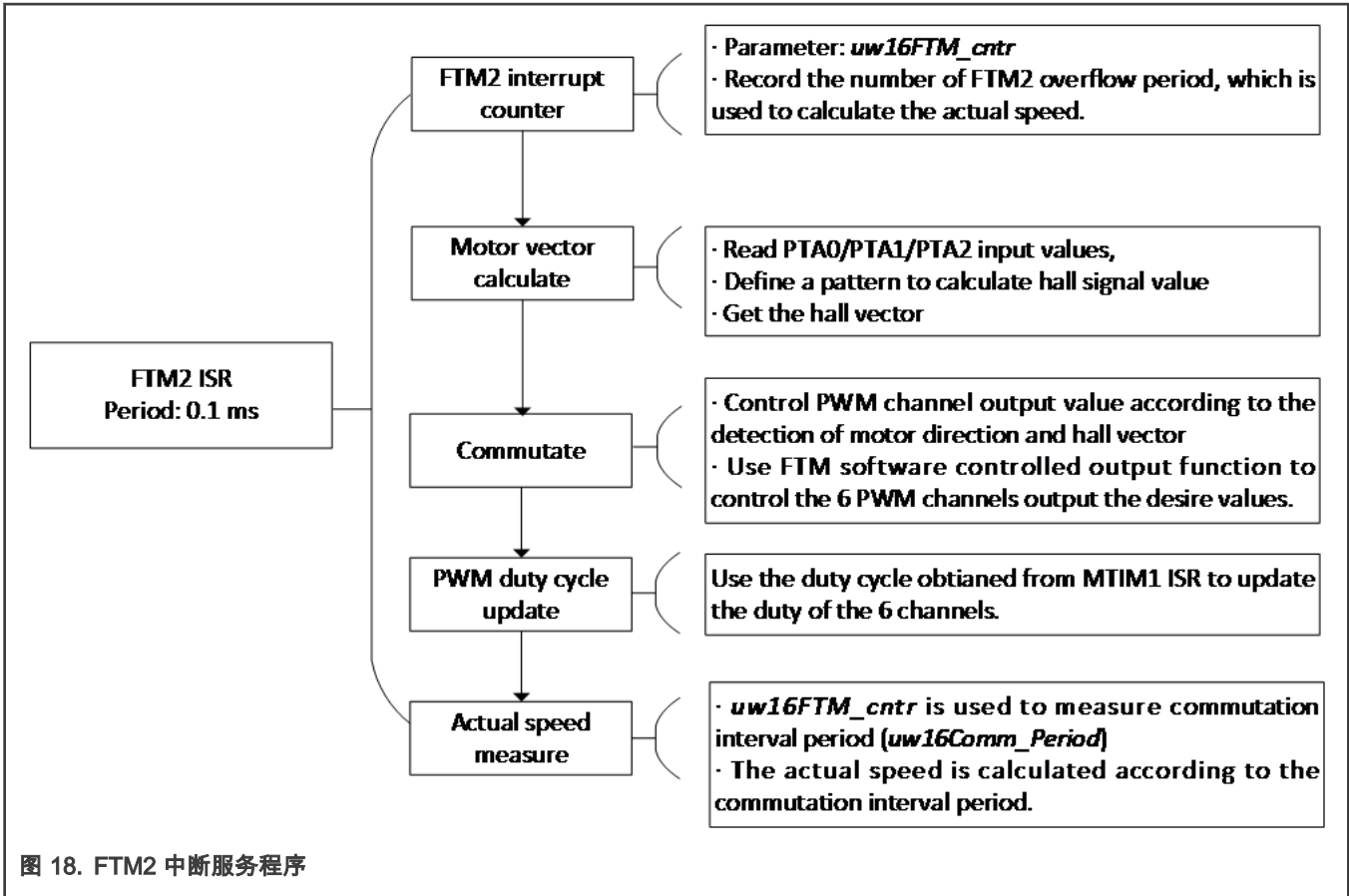


图 18. FTM2 中断服务程序

如 图 18 所示，在 FTM2 ISR 中实现了换相和实际速度测量。以下内容将对其进行详细描述。

6.3.1 换相

正确的换相表是针对客户特定的 BLDC 电机进行应用移植的关键。

在此应用中，当电动机顺时针和逆时针旋转时，霍尔值和三相电压序列之间的对应关系由人工测量确定。客户可以通过电压对齐的方式获得霍尔信号定义换相表，或参考电动机制造商提供的换相表。

本章详细介绍了如何实现换相，本应用换相表的定义如 图 19 所示。

注意

不同电机的霍尔根据安装位置不同及三相相序不同，都会导致不同扇区对应的霍尔信号发生变化，实际使用时可以用开环 align 的方式先确定各个转子扇区对应的霍尔信号。

Hall pattern definition				Vector definition	Commutation definition Clockwise direction				Commutation definition Counterclockwise direction			
PTA2_Hall3	PTA0_Hall2	PTA0_Hall1	HALL_signal	Vector	Phase A	Phase B	Phase C		Phase A	Phase B	Phase C	
1	1	0	6	1	+	-	NC	A+B-	-	+	NC	B+A-
0	1	0	2	2	+	NC	-	A+C-	-	NC	+	C+A-
0	1	1	3	3	NC	+	-	B+C-	NC	-	+	C+B-
0	0	1	1	4	-	+	NC	B+A-	-	+	NC	A+B-
1	0	1	5	5	-	NC	+	C+A-	+	NC	-	A+C-
1	0	0	4	6	NC	-	+	C+B-	NC	+	-	B+C-

图 19. 换相表定义

在我们使用的 BLDC 电机中，霍尔传感器以 120 度电角度安装，它们用于检测转子磁通量。除 111 和 000 外，将所有三个传感器的输出组合在一起将得到六种状态。每 60 度电角度重复一次换相。我们定义了霍尔信号计算模式以获取 Vector，然后根据 Vector 值执行换相过程。换相步骤如下：

- 霍尔信号的计算：

$$\text{Hall_signal} = \text{PTA2_Hall3} \times 0x04 + \text{PTA2_Hall3} \times 0x02 + \text{PTA2_Hall1} \times 0x01$$

- Vector 定义：

Vector 的值可以通过定义的数组得到： $\text{vector_Table}[\text{HALL_signal}] = \{0,4,2,3,6,5,1,0\}$

- 换相：

通过根据 Vector 值来更改 A，B，C 相电压的通电方向来实现换相，请参考图 19。

换相可以产生使旋转磁场。定子磁通和转子磁通之间的角度应保持尽可能接近 90°，以获得最大的转矩。为了使 BLDC 电机正常运行，必须使定子磁通和转子磁通之间的角度保持接近 90°，实际角度在 60°至 120°之间变化。

通过六步控制，我们总共可以获得六个可能的定子磁通矢量。定子磁通矢量必须在转子位于某一位置时改变。通常通过霍尔效应传感器的 6 种状态来感测转子位置。六个状态中的每一个对应于某个定子磁通矢量。参照图 19 的换相顺序，在图 20 中画出了六个霍尔状态以及相应的定子磁通矢量。

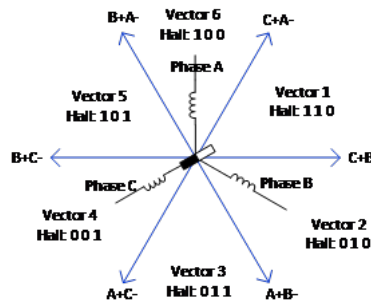


图 20. 六步控制中定子磁通向量

转子的减小定子磁通和转子磁通之间的夹角的方向转动。

如图 21(a)所示，当转子在 Vector 1 时，执行 A + B-换相。转子将从 Vector 1 运行到 Vector 2，减小定子和转子磁通之间的角度，以实现顺时针旋转。

如图 21(b)所示，当转子在 Vector 1 时，执行 B + A-换相。转子将从 Vector 1 运行到 Vector 6，减小定子和转子磁通之间的角度，以实现逆时针旋转。

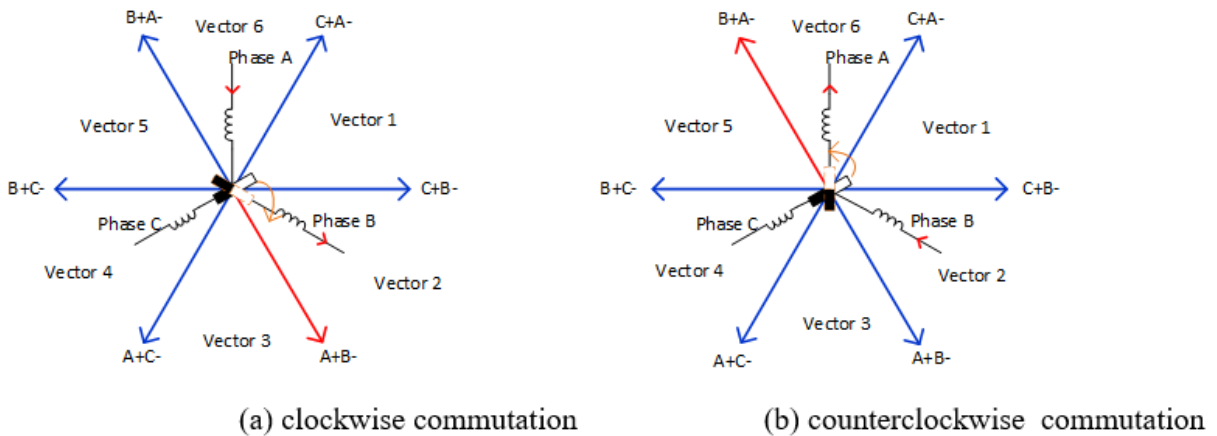
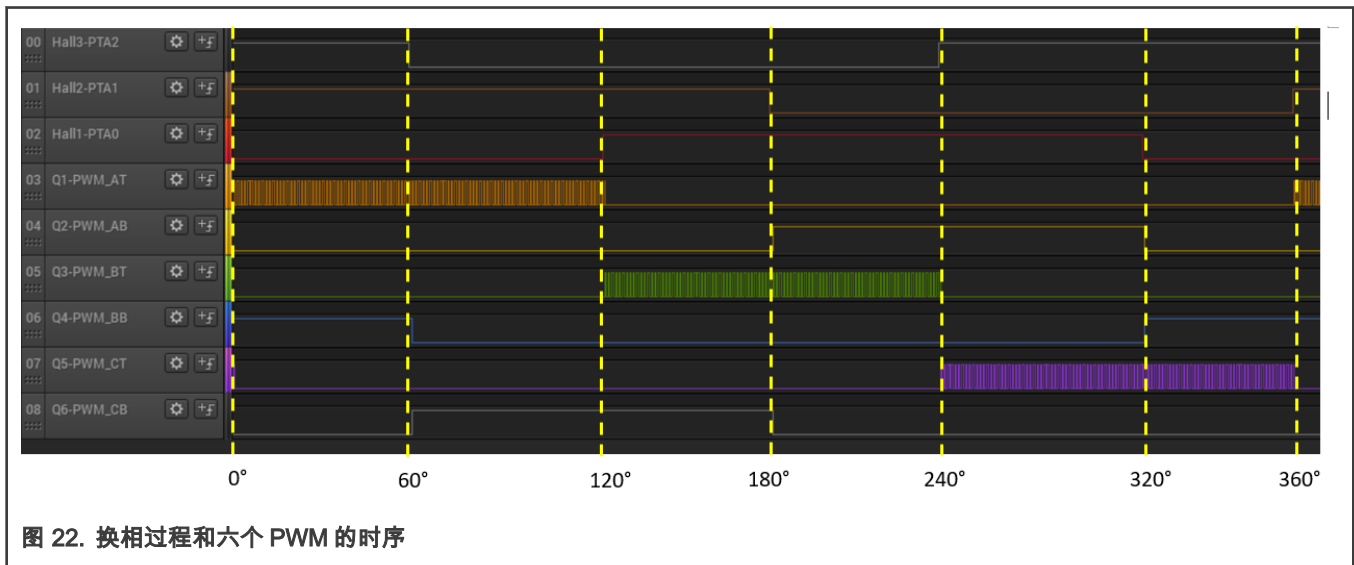


图 21. 当转子位置在 Vector 1 时的一次换相过程

三相电压由 6 个 PWM 调制可得，调制方式选择 HPWM_LON 模式。如果电动机的旋转方向为顺时针，电源顺序为 (A + B-) ->(A+ C-) ->(B + C-) ->(B + A-) ->(C + A-) ->(C + B-)，在一个电角度周期内，换相过程和六个 PWM 的时序如 图 22 所示：



6.3.2 实际速度测量

本次应用程序中的所有速度常量都进行了缩放。将所有速度常量除以预定义的最大值 (N_MAX = 100000 rpm)，然后将[-1,1) 内的带符号的小数转换为 Q1.15 格式的定点 16 位数字 (× 32768)。

每 60°电角度的换相周期，*uw16Comm_Period*，和经过缩放的常量 *NUMERATOR_FOR_SPEED* 可以用来计算实际速度值，该缩放的常量 *NUMERATOR_FOR_SPEED* 可以将时间值转换为速度值。实际速度 *w16Speed_Act_fit* 可以通过对 *w16Speed_Act* 滤波后得到，*w16Speed_Act* 可以使用以下公式计算：

$$w16Speed_Act = \frac{NUMERATOR_FOR_SPEED}{uw16Comm_Period}$$

式中 *NUMERATOR_FOR_SPEED* 通过预定义为：

$$NUMERATOR_FOR_SPEED = ((PWM_FREQUENCY_KHZ * 1000.0 * 60.0/6.0/PP)/N_MAX) * 32768.0$$

式中

- PWM_FREQUENCY_KHZ: 10 (kHz)
- N_MAX: 100000 (rpm)
- PP: 2 (pole pairs)

7 应用指南

本次 BLDC 电机控制应用可使用 FreeMASTER 与 PC 一起控制，或使用 EVK 板上的按钮 SW3 来控制速度。

请严格按照以下步骤搭建硬件和软件，然后启动此 BLDC 电机控制的演示。

1. 根据 图 23 中红色标记的位置安装跳线。

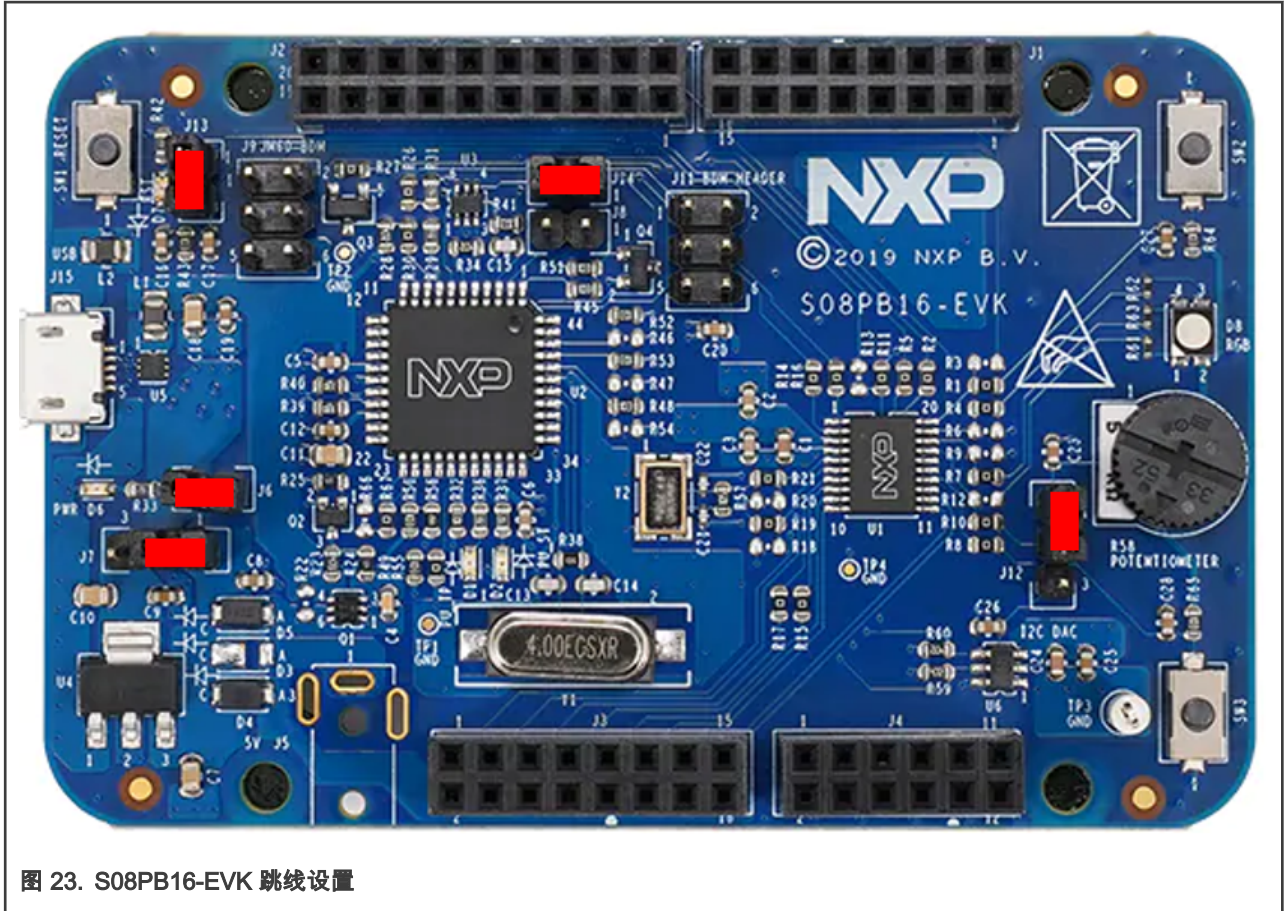


图 23. S08PB16-EVK 跳线设置

2. 修改 S08PB16-EVK 开发板。PTA0/PTA1/PTA2/PTA3/PTB3 在 S08PB16-EVK 开发板上默认连接到外围设备。但是现在，这些引脚需要连接到 FRDM-LVBLDC 开发板上的电机控制信号。需要将 R1/R4/R7/R10/R11 处的 0 Ω 电阻移除，并在 R3/R6/R9/R12/R13 处焊接 0 Ω 电阻。请参考图 24 进行电路修改。

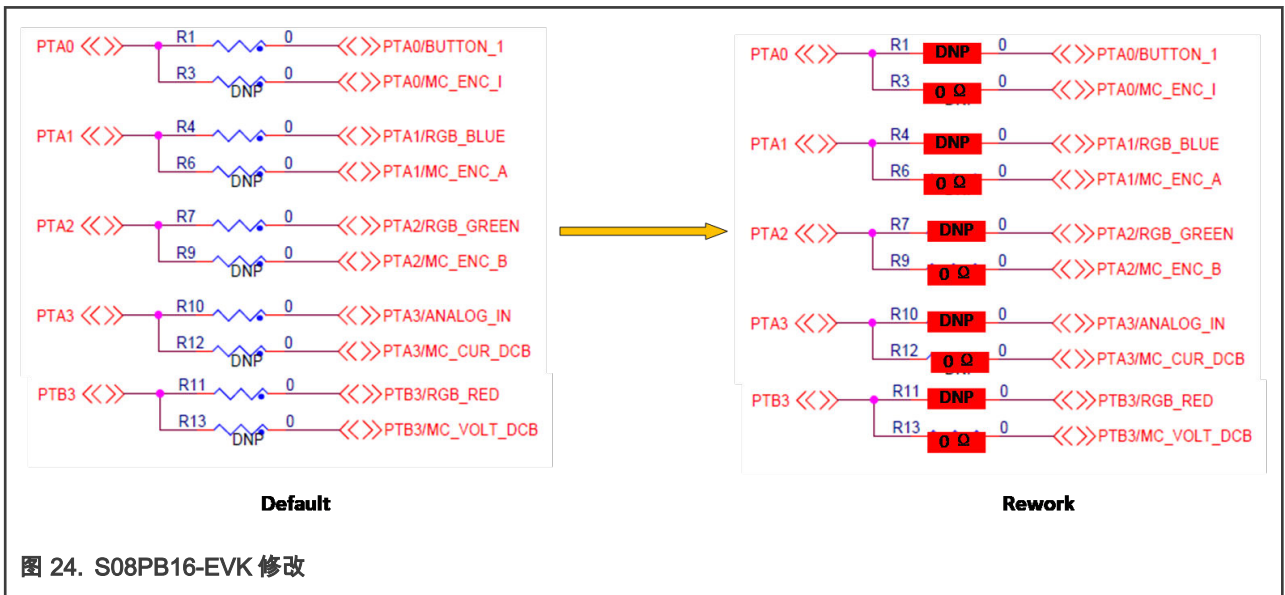
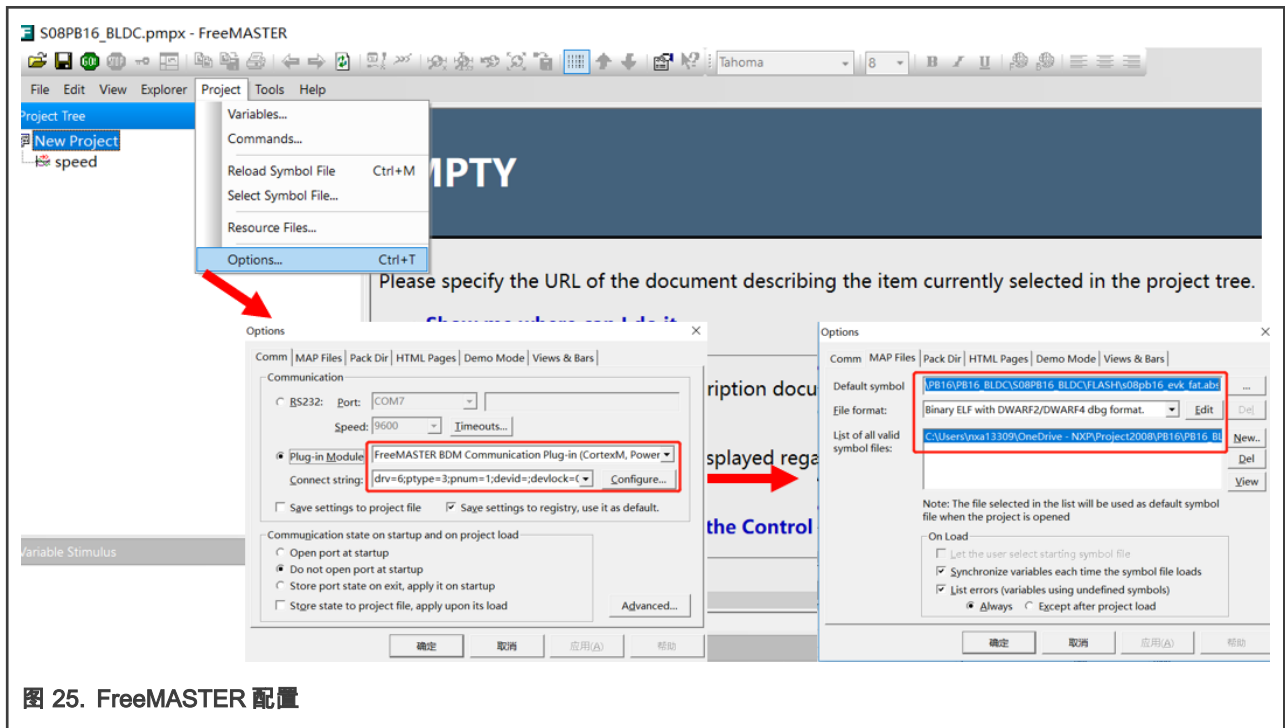


图 24. S08PB16-EVK 修改

3. 通过 Arduino 接口连接 S08PB16-EVK 开发板和 FRDM-MC-LVBLDC 开发板。
4. 依次将电机的三相输出和霍尔信号连接到 FRDM-MC-LVBLDC 开发板上。

- 将 USB 线连接到 S08PB16-EVK 板载 OSBDM 端口。
- 使用 CodeWarrior (11.1 版或更高版本) 打开工程，然后编译并将固件下载到芯片。
- 用 FreeMASTER (3.0 及以上版本) 打开 S08PB16_BLDC.pmpx，点击 project > options > Comm > Communications > Plug-in Module，选择 FreeMASTER BDM 作为通信方式，点击 project > Options > MAP Files，请选择 .abs 文件作为 Default symbol file，File format 为 Binary ELF with DWARF2 or DWARF4 dbg format。具体配置如 图 25 所示。



- 向 FRDM-MC-LVBLDC 板上提供 12 V 直流电压，然后单击 FreeMASTER 上的 GO 按钮，将速度命令值 (rpm) 设置写入 Speed_req 文本框，或者拖动 Speed Required 滑块以启动电动机。
- FreeMASTER 控制页面如 图 26 所示，可用于速度控制，demo 模式控制，限流值的调整以及速度/直流母线电流/直流母线电压/状态机/变量的实时监测。

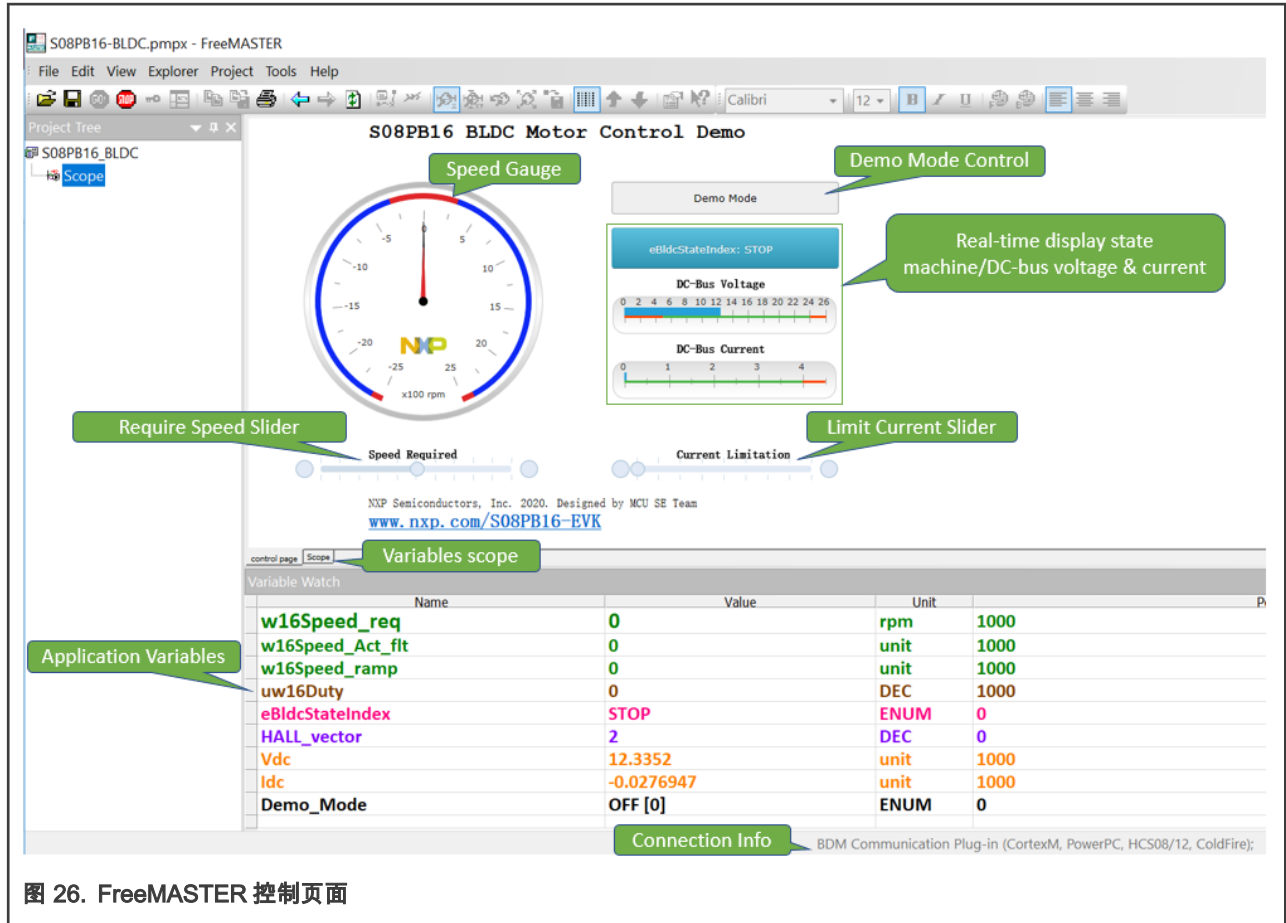


图 26. FreeMASTER 控制页面

8 参考文献

以下手册可从 <https://www.nxp.com> 获取。

1. *MC9S08PB16 Reference Manual* (document [MC9S08PB16RM](#))
2. *How to use the interconnection of OPAMP, ACMP1 and FDS for S08PB16* (document [AN12836](#))
3. *BLDC Motor Control with Hall Sensors Driven by DSC* (document [AN4413](#))
4. *BLDC Motor Control with Hall Effect Sensors Using MQX on Kinetis* (document [AN4376](#))

How To Reach Us

Home Page:

nxp.com

Web Support:

nxp.com/support

Information in this document is provided solely to enable system and software implementers to use NXP products. There are no express or implied copyright licenses granted hereunder to design or fabricate any integrated circuits based on the information in this document. NXP reserves the right to make changes without further notice to any products herein.

NXP makes no warranty, representation, or guarantee regarding the suitability of its products for any particular purpose, nor does NXP assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. "Typical" parameters that may be provided in NXP data sheets and/or specifications can and do vary in different applications, and actual performance may vary over time. All operating parameters, including "typicals," must be validated for each customer application by customer's technical experts. NXP does not convey any license under its patent rights nor the rights of others. NXP sells products pursuant to standard terms and conditions of sale, which can be found at the following address: nxp.com/SalesTermsandConditions.

While NXP has implemented advanced security features, all products may be subject to unidentified vulnerabilities. Customers are responsible for the design and operation of their applications and products to reduce the effect of these vulnerabilities on customer's applications and products, and NXP accepts no liability for any vulnerability that is discovered. Customers should implement appropriate design and operating safeguards to minimize the risks associated with their applications and products.

NXP, the NXP logo, NXP SECURE CONNECTIONS FOR A SMARTER WORLD, COOLFLUX, EMBRACE, GREENCHIP, HITAG, I2C BUS, ICODE, JCOP, LIFE VIBES, MIFARE, MIFARE CLASSIC, MIFARE DESFire, MIFARE PLUS, MIFARE FLEX, MANTIS, MIFARE ULTRALIGHT, MIFARE4MOBILE, MIGLO, NTAG, ROADLINK, SMARTLX, SMARTMX, STARPLUG, TOPFET, TRENCHMOS, UCODE, Freescale, the Freescale logo, Altivec, C-5, CodeTEST, CodeWarrior, ColdFire, ColdFire+, C-Ware, the Energy Efficient Solutions logo, Kinetis, Layerscape, MagniV, mobileGT, PEG, PowerQUICC, Processor Expert, QorIQ, QorIQ Converge, Ready Play, SafeAssure, the SafeAssure logo, StarCore, Symphony, VortiQa, Vybrid, Airfast, BeeKit, BeeStack, CoreNet, Flexis, MXC, Platform in a Package, QUICC Engine, SMARTMOS, Tower, TurboLink, UMEMS, EdgeScale, EdgeLock, eIQ, and Immersive3D are trademarks of NXP B.V. All other product or service names are the property of their respective owners. AMBA, Arm, Arm7, Arm7TDMI, Arm9, Arm11, Artisan, big.LITTLE, Cordio, CoreLink, CoreSight, Cortex, DesignStart, DynamIQ, Jazelle, Keil, Mali, Mbed, Mbed Enabled, NEON, POP, RealView, SecurCore, Socrates, Thumb, TrustZone, ULINK, ULINK2, ULINK-ME, ULINK-PLUS, ULINKpro, μ Vision, Versatile are trademarks or registered trademarks of Arm Limited (or its subsidiaries) in the US and/or elsewhere. The related technology may be protected by any or all of patents, copyrights, designs and trade secrets. All rights reserved. Oracle and Java are registered trademarks of Oracle and/or its affiliates. The Power Architecture and Power.org word marks and the Power and Power.org logos and related marks are trademarks and service marks licensed by Power.org.

© NXP B.V. 2020.

All rights reserved.

For more information, please visit: <http://www.nxp.com>

For sales office addresses, please send an email to: salesaddresses@nxp.com

Date of release: 09/2020
Document identifier: AN12957

