

采用正弦驱动的带传感器的三相 BLDC 电机控制

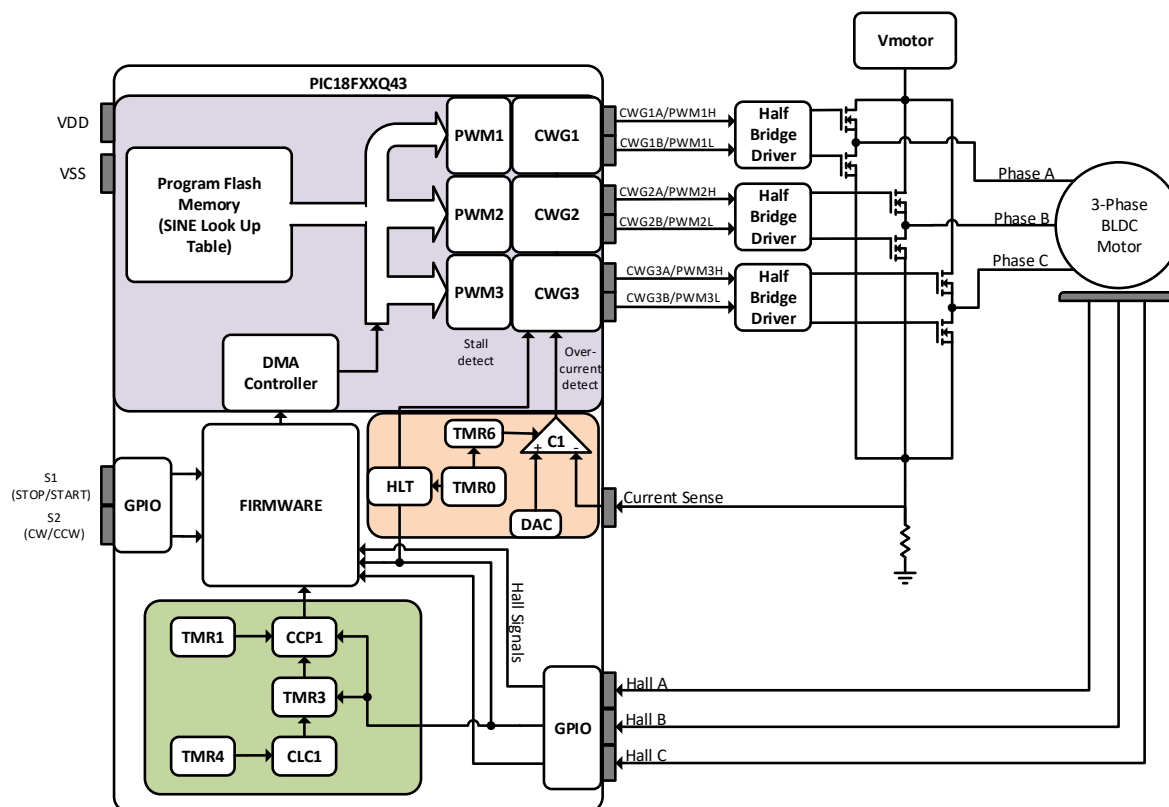
简介

作者: Aldrin Abacan, Microchip Technology Inc.

在工业应用中，正弦电流驱动一直以来都是驱动 **BLDC** 电机最常用的方法之一。与六步换相（梯形驱动）相比，正弦电流驱动的效率更高、转矩纹波和噪声更低，因此成为低速、低噪声电机控制系统用户的首选。此外，在实际应用中挑选合适的电机类型和电机控制器时还需要考虑维护和实现成本。

本应用笔记将介绍如何使用低成本 8 位单片机基于带传感器的 3 相 BLDC 电机设置实现正弦电流驱动。具体方法是使用 Microchip 独立于内核的外设（Core Independent Peripheral, CIP）以及很少需要 CPU 处理的正弦电流驱动功能，减少所需的复杂固件数学运算。本应用笔记还将重点介绍如何使用直接存储器访问（Direct Memory Access, DMA）外设，在未使用的 CPU 周期内将数据从正弦查找表（Look-up Table, LUT）移动到 PWM 外设。通过实现这些功能，CPU 可获得额外的带宽来处理其他系统进程。图 1 给出了使用 PIC18-Q43 系列器件实现正弦电流驱动的基本框图。

图 1. 三相 BLDC 电机控制框图



目录

简介.....	1
1. 概述.....	3
2. 正弦驱动——带传感器的 BLDC 电机的原理.....	4
2.1. BLDC 带传感器操作.....	4
2.2. 正弦电流驱动.....	4
3. 角位置检测器.....	7
4. 驱动信号发生器.....	9
5. 故障检测控制.....	12
5.1. 过流保护.....	12
5.2. 停转检测.....	12
6. 固件实现.....	13
6.1. 按钮控件.....	14
6.2. 电机启动.....	14
6.3. 中断源.....	14
7. 正弦驱动三相 BLDC 性能.....	18
7.1. 三相 BLDC 的 CWG 驱动信号.....	18
7.2. 相电压和线路电流.....	18
8. 结论.....	20
9. 附录 A 原理图.....	21
10. 附录 B: MPLAB®代码配置 (MCC) 外设配置.....	22
11. 附录 C 源代码列表.....	25
Microchip 网站.....	26
产品变更通知服务.....	26
客户支持.....	26
Microchip 器件代码保护功能.....	26
法律声明.....	26
商标.....	27
质量管理体系.....	27
全球销售及服务网点.....	28

1. 概述

在本应用中，基于 PIC18-Q43 实现的正弦电流驱动可分为三个功能：[驱动信号发生器](#)、[角位置检测器](#)和[故障检测控制](#)。驱动信号发生器为实现正弦电流驱动提供所需的信号。此阶段产生的驱动信号连接到外部 MOSFET 驱动器，用于直接驱动 BLDC 电机。驱动信号用于对电源进行门控以提供调制电压。角位置检测器根据电机角位置提供周期性中断。这些中断用于触发驱动信号施加的电压发生变化。故障检测控制器监视过流和停转等事件。检测到故障后，控制器将禁止驱动信号发生器和所有中断，从而使电机停止运转。下面列出了用于确保成功执行这些功能的外设。

- 驱动信号发生器
 - 直接存储器访问（DMA）控制器
 - 互补波形发生器（Complementary Waveform Generator, CWG）
 - 16 位脉宽调制（Pulse Width Modulation, PWM）
- 角位置检测器
 - TMR1/3/4
 - 可配置逻辑单元（Configurable Logic Cell, CLC）
 - 捕捉/比较/PWM（CCP）
- 故障检测
 - 数模转换器（Digital-to-Analog Converter, DAC）
 - 比较器（CMP）
 - TMR0/2/6

图 1-1. 控制图

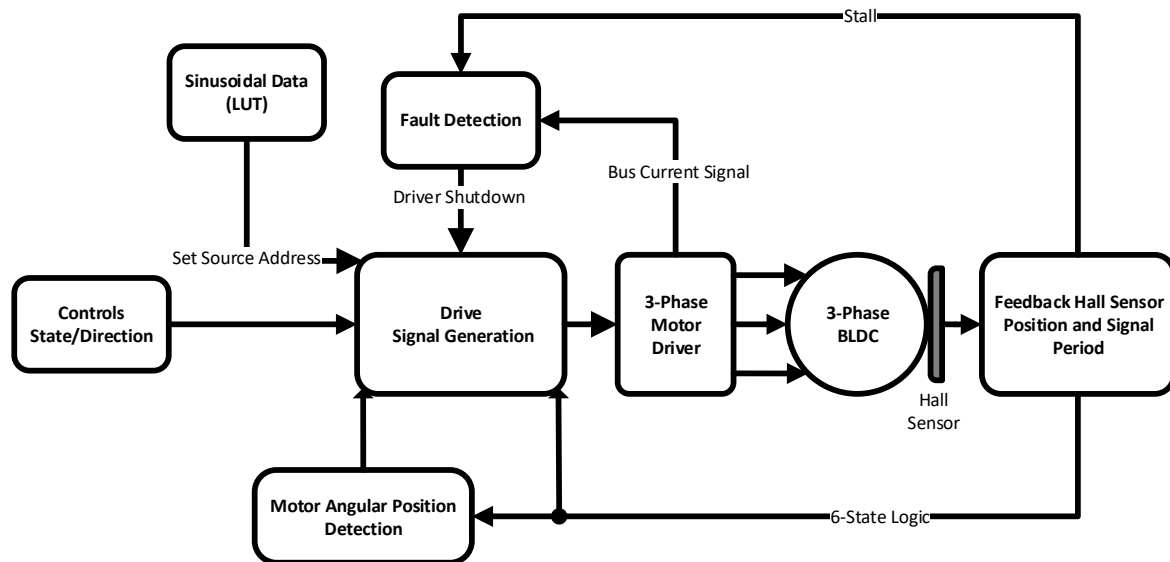


图 1-1 所示为三相 BLDC 电机操作的功能框图。输入控制信号可用于停止/启动命令和方向控制。这些输入控制信号决定电机何时旋转以及驱动信号发生器将遵循的换相顺序。正弦数据根据转子位置提供在可编程闪存（[Programmable Flash Memory, PFM](#)）上存储的 PWM 占空比值，该值随后将装入驱动信号发生器。本应用中使用的电机是 [DMB0224C10002](#)，其中内置一个霍尔效应传感器，用于反馈电机位置。

2. 正弦驱动——带传感器的 BLDC 电机的原理

本应用笔记重点介绍如何使用 8 位 PIC[®] MCU 在 3 相 BLDC 上实现正弦电流驱动操作。霍尔效应传感器用作反馈来识别转子位置，BLDC 驱动算法需要使用转子位置。有关 BLDC 电机控制基础的更详细说明，请参见 [AN885:《无刷直流 \(BLDC\) 电机基础》](#)。

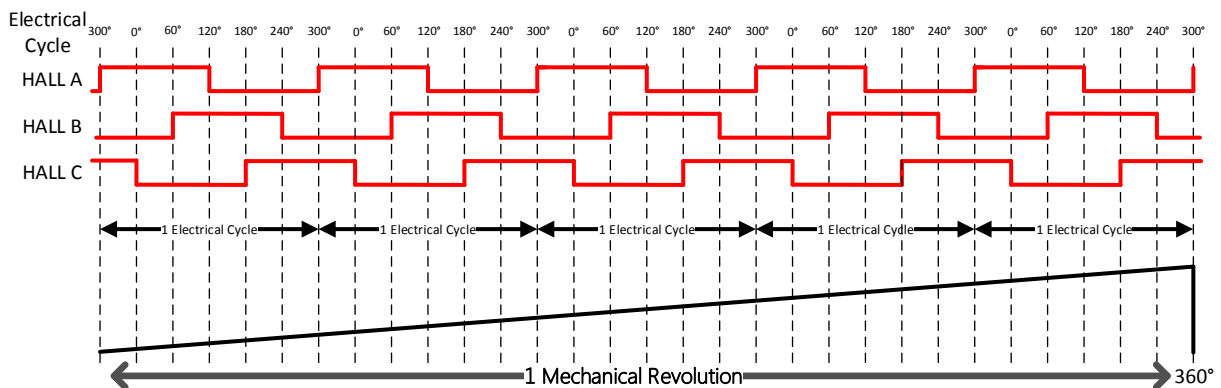
2.1 BLDC 带传感器操作

识别转子位置是 BLDC 电机控制的重要环节。转子位置用于确定适当的电子换相。识别 BLDC 电机转子位置最常用的方法是使用霍尔效应传感器。霍尔效应传感器是一种根据检测到的磁场来产生逻辑电平的感应开关。随着电机旋转，霍尔效应传感器会识别在电机中安装的磁极（正极或负极）的位置信息，并将其发送到控制器。在本应用中，电机内部预装了三个霍尔效应传感器。这些霍尔传感器均匀分布在定子周围（彼此呈 120°），因此转子的输出可以在一个电气周期中产生六种不同的组合，每移动 60°就会发生变化。这些组合可以转化为数字 1 到 6（用 3 个二进制数字表示）。这种传感器布局可确保即使在转子到达下一次换相位置之前，磁体的极性也会发生变化，从而防止转子卡住。

一次完整旋转需要的电气周期数取决于电机的极对数，如图 2-1 所示。由于本应用中使用的是 5 极对电机，因此一次完整的机械旋转共需要 5 个电气周期。此外，霍尔传感器在每个电气周期都会改变状态，因此通过测量每次状态变化的间隔时间可以获得角速度或电机速度。

在本应用中，单片机使用一套定时器对一个霍尔周期或一个电气周期内累积的系统时钟节拍进行计数。一个霍尔周期内的时钟节拍数代表电机的速度，具体来说，电机速度（单位为时钟节拍数）是一个霍尔周期内系统时钟节拍数的 5 倍。该信息可用于将霍尔周期细分为更小的时间间隔以获取施加到驱动器的电压的变化率，这对于正弦电流驱动或仅测量其当前速度至关重要。

图 2-1. 霍尔传感器信号、电气周期和机械旋转之间的关系



2.2 正弦电流驱动

正弦驱动的基本前提是根据转子位置为电机的每个绕组提供正弦变化的电流。这些电流的相位彼此呈 120°（与各自的霍尔传感器相对应）。在 BLDC 电机控制中，使用的驱动信号需要随电机速度和位置而变化的可变电压。该可变电压使用 PWM 技术施加。通过 PWM 模块向 MOSFET 驱动器提供正弦信号，电机的每个绕组上都会产生电流。由于施加的电压逐渐变化，与梯形驱动相比，正弦驱动的转矩纹波在某种程度上更小。

在本应用中，使用依靠空间矢量脉宽调制（Space Vector Pulse-Width Modulation, SVPWM）技术提取的数据来应用正弦电流驱动。相比于定期将单个正弦信号应用于电机绕组，这种技术凭借更高的效率和更复杂的算法而被广泛用于数字实现。

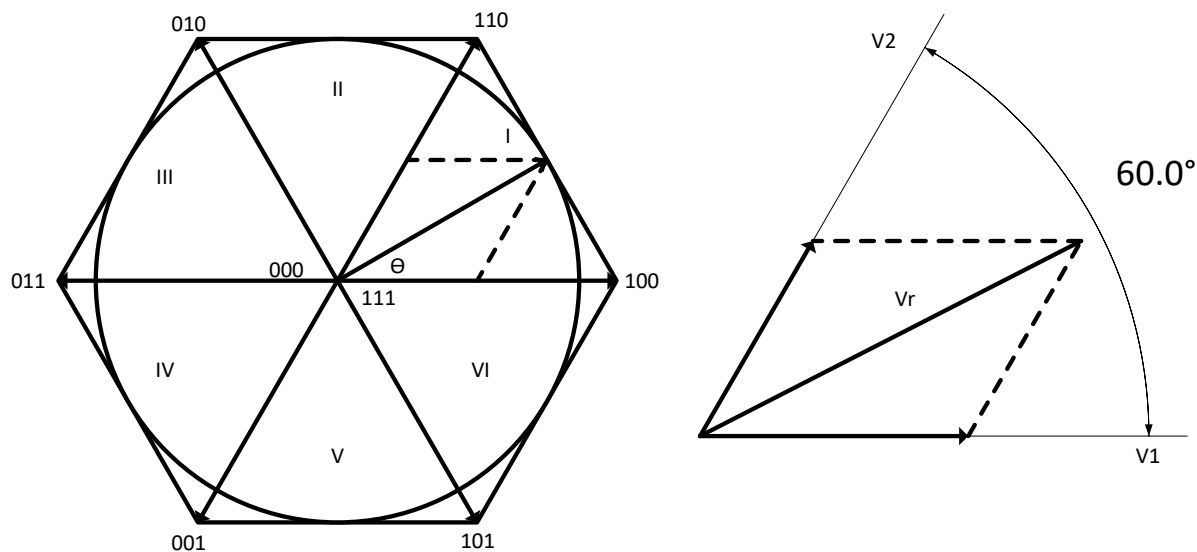
SVPWM 方法是一种基于矢量的方案，用于三相系统，例如电机控制应用。SVPWM 不是根据每个电机端子相对于地的驱动信号产生纯正弦波形，而是在每两个端子之间产生总共 3 个正弦线间电压（差分电压）。空间矢量表示按照图 2-2 所示创建，其中相邻两个单位矢量之间的空间均呈 60.0°，分别对应于电气周期中的每个状态。每个所需的电压均可通过将相邻两个活动矢量的分量和零矢量（由 000 和 111 逻辑状态（位于原点）表示）相加来仿真。最终得到的电压通过操控 PWM 外设的占空比来表示。

表 2-1 给出了每个扇区的 PWM 开关时间公式。确定近似角位置后，便可基于相邻电压空间矢量和零矢量来计算合成矢量幅值， T_0 、 T_1 和 T_2 表示一个周期内的导通时间。 T_1 和 T_2 的值从查找表中获取，该查找表包含 0° 到 60° 之间共 172 个小数正弦值。

表 2-1. 每个传感器的 PWM 开关时间公式

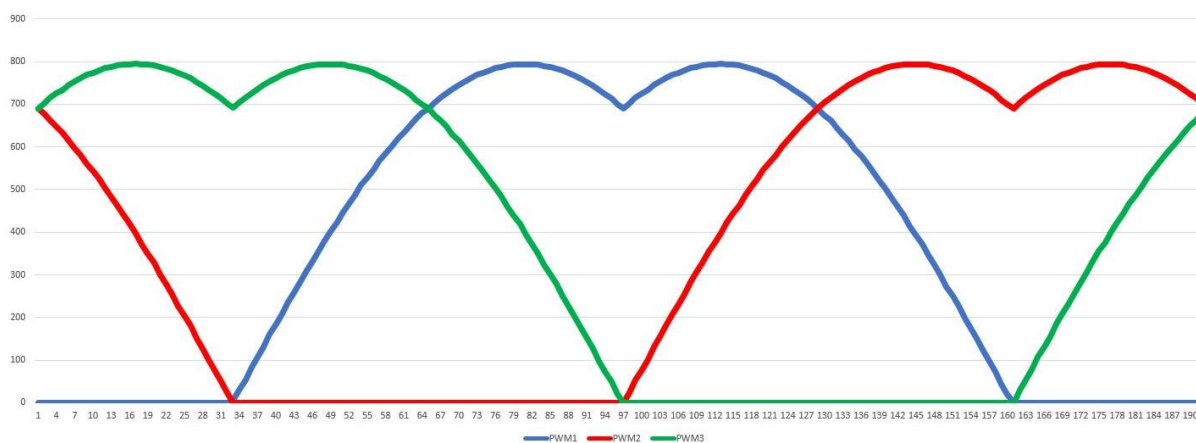
扇区	开关时间公式
1	$PWM1 = T_1 + T_2 + T_0/2$ $PWM2 = T_2 + T_0/2$ $PWM3 = T_0/2$
2	$PWM1 = T_1 + T_0/2$ $PWM2 = T_1 + T_2 + T_0/2$ $PWM3 = T_0/2$
3	$PWM1 = T_0/2$ $PWM2 = T_1 + T_2 + T_0/2$ $PWM3 = T_2 + T_0/2$
4	$PWM1 = T_0/2$ $PWM2 = T_1 + T_0/2$ $PWM3 = T_1 + T_2 + T_0/2$
5	$PWM1 = T_2 + T_0/2$ $PWM2 = T_0/2$ $PWM3 = T_1 + T_2 + T_0/2$
6	$PWM1 = T_1 + T_2 + T_0/2$ $PWM2 = T_0/2$ $PWM3 = T_1 + T_0/2$

图 2-2. 正交平面中的 SVPWM 表示



SVPWM 的近似 PWM 输出（相对于其角位置）如图 2-3 所示。其波形呈马鞍形状，由于 SVPWM 的三次谐波注入所致。三个 PWM 输出彼此呈 120° 。在本应用中，角位置从 360° 缩小到 192° 以便于实现。为了降低开关损耗，三相中的每一相在一个周期内都有三分之一的时间不进行开关。

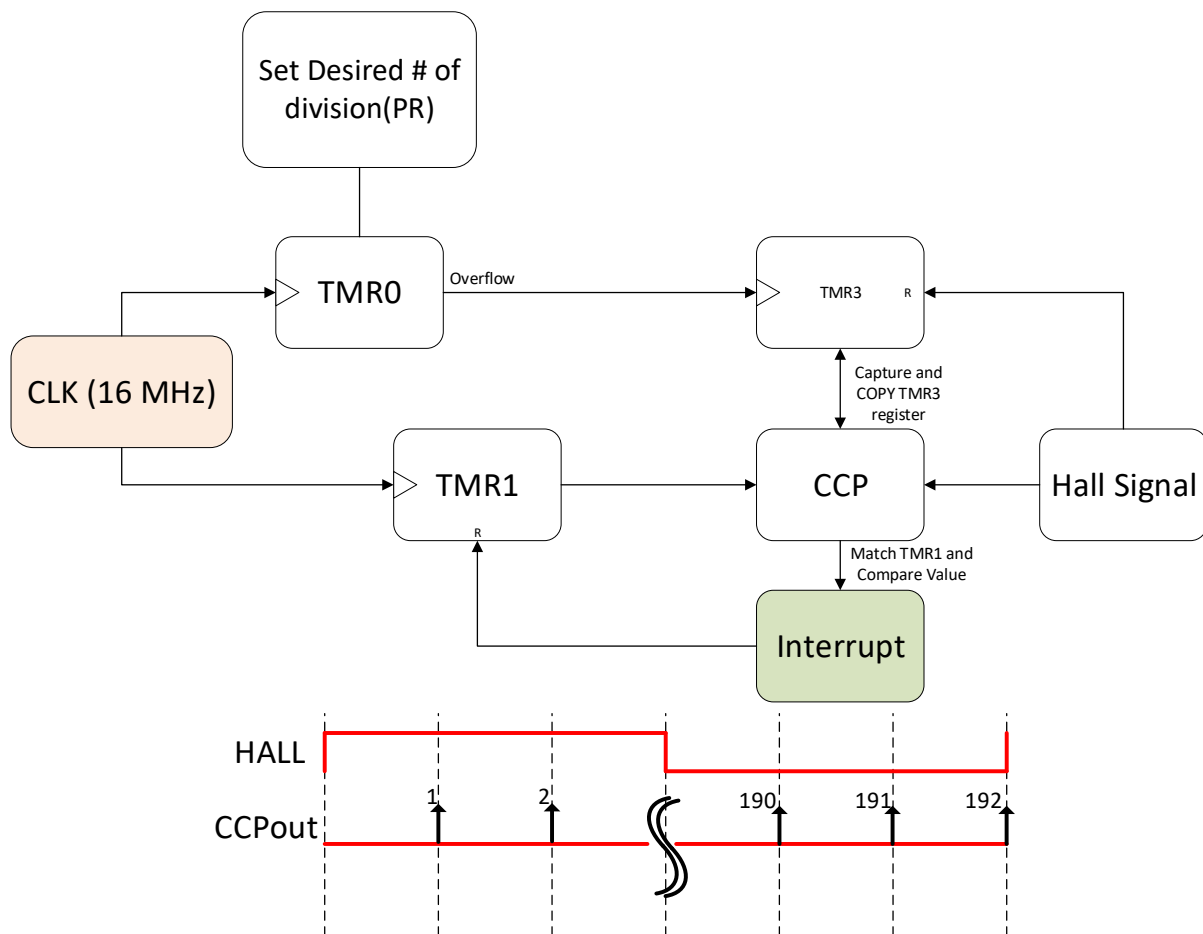
图 2-3. SVPWM 电压波形表示



3. 角位置检测器

在应用正弦电流驱动期间，转子的角位置对于产生与电机同步的波形至关重要。霍尔效应传感器提供的信息有限，因为它在电气周期内只能检测近似的位置。要将正弦电流驱动应用到 BLDC 电机，需要基于转子的准确位置不断改变在 MOSFET 驱动器上施加的电压。在本应用中，设计了电机角位置功能来将霍尔信号转换为代表单位旋转角的均等小信号。将霍尔信号细分为更小的单位后，可在特定时间和位置触发改变施加的电压。为了创建此功能，使用了定时器和 CCP 等多个外设。这种方法的基本思路是，首先对完成一个霍尔周期所需的系统时钟节拍进行计数，然后将其除以下一个周期所需的单位数（假设下一周期采用相同的时钟节拍计数）。这基本上属于硬件除法，因此计算结果中可能会有余数。这些余下的时钟将添加到周期的最后一度/相，尤其是在使用较慢的时钟源时。为了减少此类错误，应使用较快的时钟源。本设计中使用的是 16 MHz 时钟。图 3-1 给出了实现的电机角位置方案框图。

图 3-1. 电机角位置检测方案框图



TMR4 的周期寄存器用于设置每个霍尔信号周期所需的除数。由于声明的数组数为 6，每个数组有 32 个元素，因此 T4PR 中设置的除法数量为 192。

TMR3 没有来自 TMR4 溢出位的直接输入连接。通过使用 CLC 作为缓冲器，可以将 TMR4 输出连接到 TMR3 输入。TMR4 的值将不断递增，直到检测到霍尔信号的参考边沿为止。检测到参考边沿后，TMR3 寄存器中的值便会复制到 CCP 比较值，该比较值也可以使用公式 3-1 计算。例如，200 Hz 霍尔信号需要 80,000 个时钟节拍。由于每次 TMR3 递增需要 192 个时钟节拍，因此当检测到霍尔传感器边沿时，TMR3 计数器将记录的值为 416。该值将复制到 CCP 比较寄存器，随后不断与 TMR1 寄存器值进行比较。TMR1 与 TMR4 共用同一个时钟源，即 16 MHz 时钟。如果发生匹配，将触发 CCP 中断以重载 TMR1 寄存器并向 DMA 发送信号以使下一个 PWM 值的源地址递增。对于 MCC 配置，TMR4 寄存器需要基于时间的周期而不是寄存器值，具体可以使用公式 3-2 来计算。T4TMR 是一个 8 位寄存器，可以

存储 255 个值。每个时钟周期都会使 T4TMR 递增，直到其与 T4PR 发生匹配。使用的时钟源是 64 MHz 时钟的 $F_{osc}/4$ 。

公式 3-1. 比较值计算

$$\text{比较值} = \frac{\text{时钟源}}{\text{霍尔频率} \times \text{所需中断数}}$$

公式 3-2. 基于所需中断数的 TMR4 周期计算

$$\text{TMR4 周期} = \frac{\text{所需CNTR 最终值} + 1}{\text{时钟源}}$$

$$\text{TMR4 周期} = \frac{192 + 1}{16 \text{ MHz}} = 12.062 \mu\text{s}$$

4. 驱动信号发生器

正弦数据分为六个数组，以供 DMA 控制器轻松将其地址指为源地址，如图 4-1 所示。三个绕组需要三个彼此呈 120° 的相同信号，因此可以使用一组正弦数据。

图 4-1. 正弦数据查找表

```
//PWM sinusoidal lookup table for six states
const uint16_t PWM_1[32] = {0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0};
const uint16_t PWM_2[32] = {5, 30, 54, 83, 107, 131, 160, 184, 208, 236, 259, 283, 310, 332, 355, 380, 401, 423, 448, 467, 487, 510, 529, 547, 568, 584, 601, 620, 635, 650, 666, 680};
const uint16_t PWM_3[32] = {690, 704, 715, 725, 736, 745, 753, 762, 769, 774, 780, 785, 788, 791, 794, 794, 794, 794, 793, 790, 787, 784, 779, 773, 767, 759, 751, 744, 733, 723, 713, 699};
const uint16_t PWM_4[32] = {690, 701, 715, 725, 734, 745, 753, 761, 769, 774, 779, 785, 788, 791, 794, 794, 795, 794, 793, 791, 787, 784, 779, 773, 767, 761, 751, 744, 734, 723, 713, 701};
const uint16_t PWM_5[32] = {687, 674, 661, 644, 629, 614, 594, 578, 561, 540, 521, 502, 479, 459, 439, 414, 393, 372, 345, 323, 301, 273, 250, 227, 198, 175, 151, 122, 98, 73, 44, 20};
const uint16_t PWM_6[32] = {0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0};
```

每个数组由 32 个 16 位元素组成。这些数组存储在 MCU 的闪存程序存储器（PFM）中。使用 DMA 控制器时，第一个元素的地址设置为 DMA 源地址。数据从 PFM 传输到 PWM 占空比寄存器，该寄存器设置为 DMA 目标地址。DMA 控制器要求目标地址大小与源地址大小互为倍数关系。DMAxDSZ 寄存器决定目标地址大小，DMAxSSZ 寄存器决定源地址大小。PWM 占空比寄存器是一个 16 位寄存器，因此 DMAxDSZ 寄存器的值等于 2 个字节。DMAxSSZ 的值取决于存储在 PFM 中的数组大小，具体可以使用公式 4-1 来计算。

公式 4-1. DMAxSSZ 寄存器值

$$\text{DMAxSSZ} = \text{元素大小 (字节)} \times \text{元素数量}$$

有两个事件可以访问或修改 DMA 控制器的设置：一个在霍尔状态跳变期间，另一个在电机角位置中断期间。当检测到霍尔状态变化时，固件会根据扇区位置和方向装载 DMA 控制器源地址寄存器。在操作期间更改地址需要在运行时使能和禁止 DMA 通道。为此，需要使用锁定和解锁序列。另一方面，电机角位置中断会触发 DMA 控制器，使 DMA 源地址递增，从而指向数组的下一个元素，直到所有元素都装载到目标或检测到另一个霍尔状态变化为止。

每个 DMA 通道的目标地址都指向用于设置可变电压的 PWM 片寄存器。电机控制应用的开关频率取决于电机的电感。如果电机电流可以快速变化，开关频率应该足够高以避免磁芯饱和。在本应用中，开关频率设置为 40 kHz，以满足为电机操作提供高频率和高分辨率的需求。公式 4-2 给出了 PWM 周期的计算方法（参考所选频率）。本应用中使用的正弦数据是基于 PWM 分辨率和 PWM 周期的占空比值，具体可以使用公式 4-3 来计算。选择中心对齐模式的目的是在每个周期内产生两个线间脉冲。这样可以降低纹波电流，同时还不会增加功率器件的开关损耗。PWM 寄存器为双缓冲寄存器，需要通过外部源或 LD 位来触发。在本应用中，使用 DMA 作为外部装载事件。每次 DMA 通道成功传输消息后，都会发出装载请求。缓冲器每隔一个请求更新一次。

公式 4-2. 中心对齐模式 PWM 中的 PWM 周期寄存器值

$$PWMxPR = \frac{CLK \text{ 源}}{\text{开关频率} \times \text{预分频值} \times 2} = \frac{64 \text{ MHz}}{40 \text{ kHz} \times 1 \times 2} - 1 = 799 = 0x31F$$

公式 4-3. PWM 占空比寄存器值

$$PWMxS1P1 = PWMxPR \times \frac{\text{所需占空比} \%}{100}$$

为了使用正弦电流驱动三相电机，我们使用了 CWG 外设。三相 BLDC 电机的 MOSFET 驱动器由三个半桥组成，每个电机端子一个。每个半桥由两个需要互补 PWM 信号的开关组成。因此，CWG 模块配置为半桥模式，以便每个模块以输入的真值和反相形式生成两个输出信号。CWGxA 输入配置为驱动器的上桥臂开关，CWGxB 输入配置为驱动器的下桥臂开关。图 4-2 给出了每个半桥的 CWG 输出表示。

图 4-2. CWG 输出



由于功率 MOSFET 的导通或关断需要一定的时间，因此有时可能会存在半桥的两个开关同时导通的情况。这会形成一条从电源到地的低电阻路径，可能导致短路。此事件称为直通。在 CWG 中，我们插入了死区延时来防止输出电平跳

变期间出现直通电流。死区延时是在开关切换之间插入的延时，用于防止上桥臂和下桥臂开关同时导通。可以在输入源的上升沿或下降沿插入死区延时。死区计数器在 **CWGxDBR** 和 **CWGxDBF** 寄存器中设置。

5. 故障检测控制

本系统中实现了故障检测控制，用于在检测到可能导致电机驱动故障的事件时停止电机操作。为避免误触发，应在电机启动后等待不到一秒的时间（使用 TMR0）再使能每个检测单元。

5.1 过流保护

如果由于换相失败、负载过大或内部故障等原因而导致电路消耗的电流超过允许的最大电流，则电路可能会出现过热或系统故障。为了防止此类情况，应使用过流检测，具体通过监视三相 MOSFET 驱动器的母线电流来实现。电阻 R_{SHUNT} 分接在电机驱动器的下桥臂。R_{SHUNT} 两端的电压代表流过驱动器的电流，因为该电阻两端的电压与流过该电阻的电流成正比。母线电流可以使用公式 5-1 计算。

公式 5-1. 母线电流计算

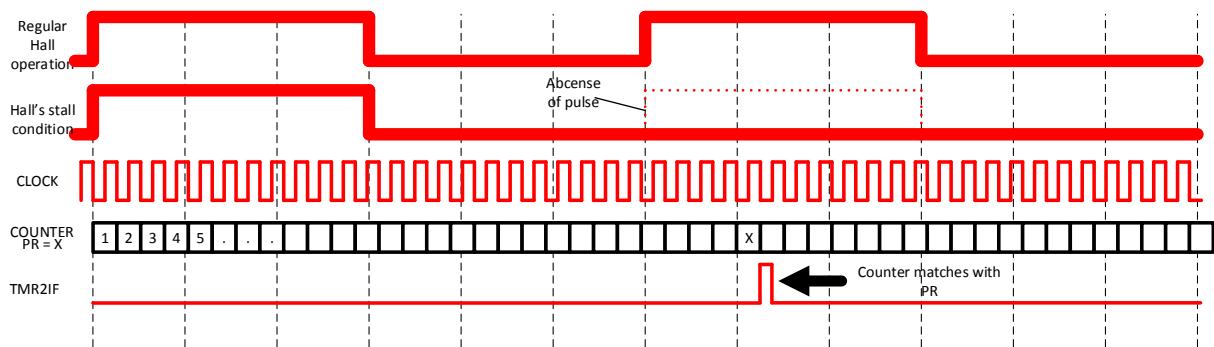
$$I = \frac{V_{DD} - V_{BIAS}}{R_{SHUNT} \times A_V}$$

电流检测电阻的电压会不断与 MCU 提供的参考电压进行比较。在电机运行期间，母线电流由比较器（CMP）输入采样。每 50 ms 会监视一次比较器输出电压，以验证其是否超过 DAC 中设置的参考电压。如果比较器输出为高电平，将禁止 CWG 外设并使电机停止运转。在电机启动期间，电机在达到稳定状态之前会消耗较高的电流。这可能会导致误触发并使电机停止运转。为防止出现这种情况，应在启动后插入 500 ms 的延时，之后再使能故障检测单元。

5.2 停转检测

当电机运转时，电机霍尔传感器会以脉冲形式提供稳定的周期性信号。如果长时间缺失这类脉冲，则表明电机已停止运转或处于停转状态。这种情况通常发生在电机负载转矩大于电机轴转矩时。在停转状态下，电机消耗的电流会超过额定值，这也会导致电机出现过热和故障。

图 5-1. HLT 工作原理



为了检测这种情况，我们还实现了停转检测功能。通过使用 TMR2 外设的硬件限制定时器（Hardware Limit Timer, HLT），可以检测到一段时间内缺失霍尔信号。此定时器被设置为在计满返回脉冲模式下工作。在此模式下，每次检测到下降沿跳变时，都会允许输入信号将 TMR2 寄存器清零。当脉冲缺失或周期时间过长时，TMR2 将不断递增，直至达到其最大值，同时将其溢出位置 1。HLT 随后将触发一个中断以禁止所有 CWG 外设，让电机停止运转。在本应用中，连接到 RC6 输入的霍尔 B 信号用作参考信号。霍尔传感器提供的信号在常规设置中的周期约为 5 ms。如果定时器计数器匹配 PR 值，中断标志将置 1，如图 5-1 所示。在 TMR2 上设置的 PR 值应大于霍尔传感器信号的周期，以便能够识别电机的停转情况。要计算 TMR2 PR 寄存器值，可使用公式 5-2。

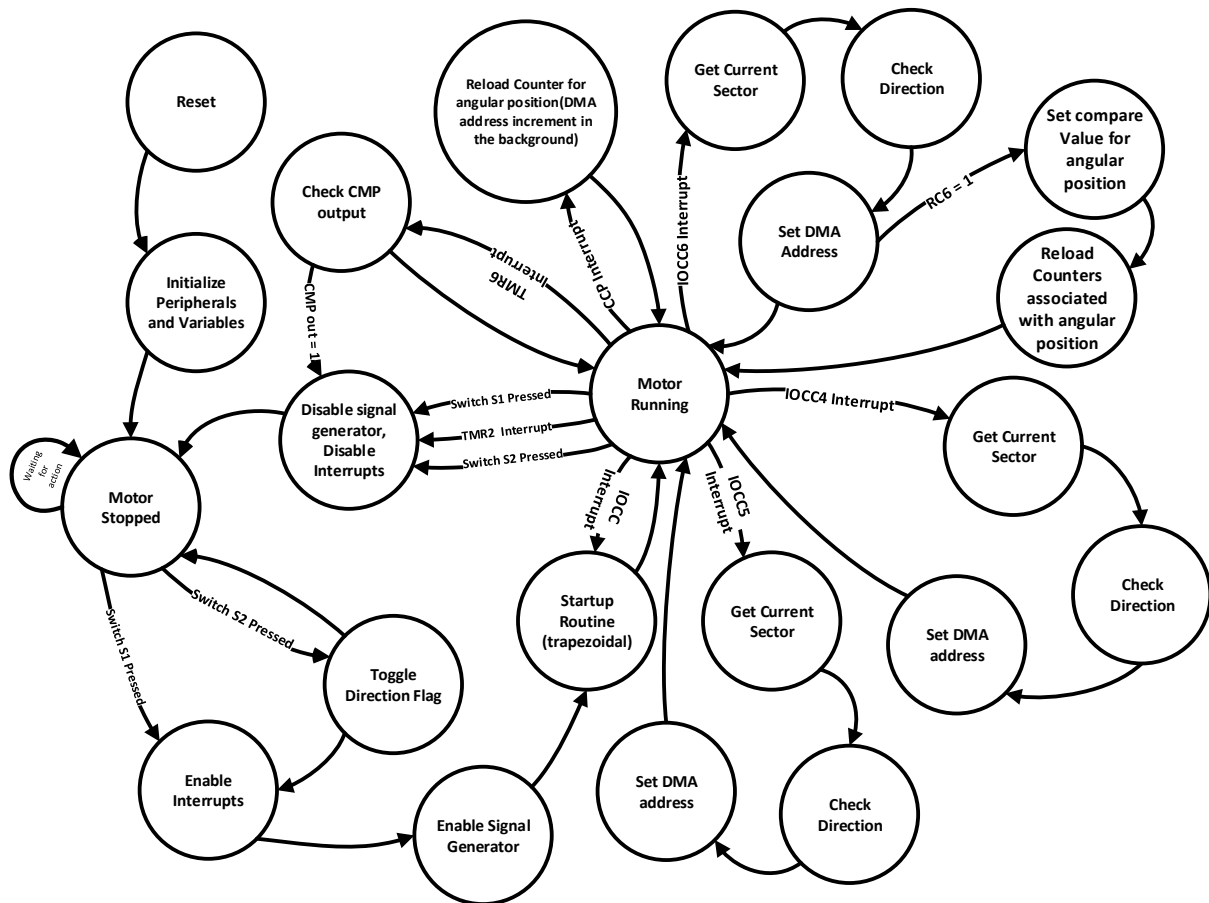
公式 5-2. PR2 计算

$$PR2 = \frac{TMR2\text{时钟源}}{\text{霍尔信号周期} \times \text{预分频值} \times \text{后分频值}}$$

6. 固件实现

图 6-1 中的状态图说明了电机控制系统及其与系统中可用的每项中断服务的交互。固件基本上围绕电机的状态（处于运行状态还是停止状态）来实现。电机的状态通过标志来识别。复位后，对所有外设和变量进行初始化，同时让中断保持禁止状态。电机最初设置为停止状态和 CCW（逆时针）方向。有关完整的源代码，请参见附录 C：源代码列表。

图 6-1. 电机控制状态图



本应用中使用的器件是 PIC18F47Q43。该器件配备 8192 字节的数据存储器、131,072 字节的程序存储器和 2048 字节的 EEPROM。本应用使用 94 字节（1%）的数据存储器空间和 7,190 字节（5%）的程序存储器空间，未使用 EEPROM。

6.1 按钮控件

使用两个按钮作为输入来触发启动/停止操作和 **CCW/CW**（逆时针/顺时针）方向。初始化后，这两个输入按钮会被持续监视，如例 6-1 所示。当按下开关 **S1** 时，将检查电机处于运行状态还是停转状态。如果电机处于停转状态，将调用 `RunMotor()` 子程序，以使能 **CWG**、中断程序和 **CCP**。如果电机处于运行状态，将调用 `StopMotor()` 子程序，以禁止 **CWG**、中断程序和 **CCP**，翻转电机状态标志。

例 6-1. 程序主循环

```
while (1) {
    //等待至检测到操作
    CheckTriggerButton();
    CheckDirectionButton();
}
```

开关 **S2** 用于翻转方向标志。在电机停止状态下按下 **S2** 时，只会将方向标志从 **CCW** 更改为 **CW**，或者从 **CW** 更改为 **CCW**。通过翻转方向标志，可改变在霍尔跳变期间基于正弦数据数组的换相顺序。如果在运行状态下按下 **S2** 按钮，将强制电机在改变换相顺序之前停止运转。当方向标志发生翻转并且识别到位置后，电机将重新开始运转。

6.2 电机启动

电机从停止状态启动后的初始速度为零。这种情况下很难应用正弦电流驱动，因为没有可用的参考速度。在本应用中，使用低速梯形驱动作为启动换相，待电机达到稳定状态后切换为正弦驱动。梯形驱动通过将 **CWG** 外设配置为转向模式来执行。每次检测到霍尔跳变时，**PWM** 信号都会根据程序闪存中存储的 6 步驱动表转向输出引脚，如图 6-2 所示。每个驱动表数组包含六个元素，分别基于霍尔传感器逻辑的六个状态。随着电机开始运转，**PWM** 占空比会逐渐增加，直至达到稳定运转速度并完成机械旋转。

图 6-2. 梯形查找表

```
const uint8_t fwdDriveTable_CWG1 [7] = {0x00, 0x02, 0x00, 0x02, 0x10, 0x00, 0x10};
const uint8_t fwdDriveTable_CWG2 [7] = {0x00, 0x00, 0x10, 0x10, 0x02, 0x02, 0x00};
const uint8_t fwdDriveTable_CWG3 [7] = {0x00, 0x10, 0x02, 0x00, 0x00, 0x10, 0x02};

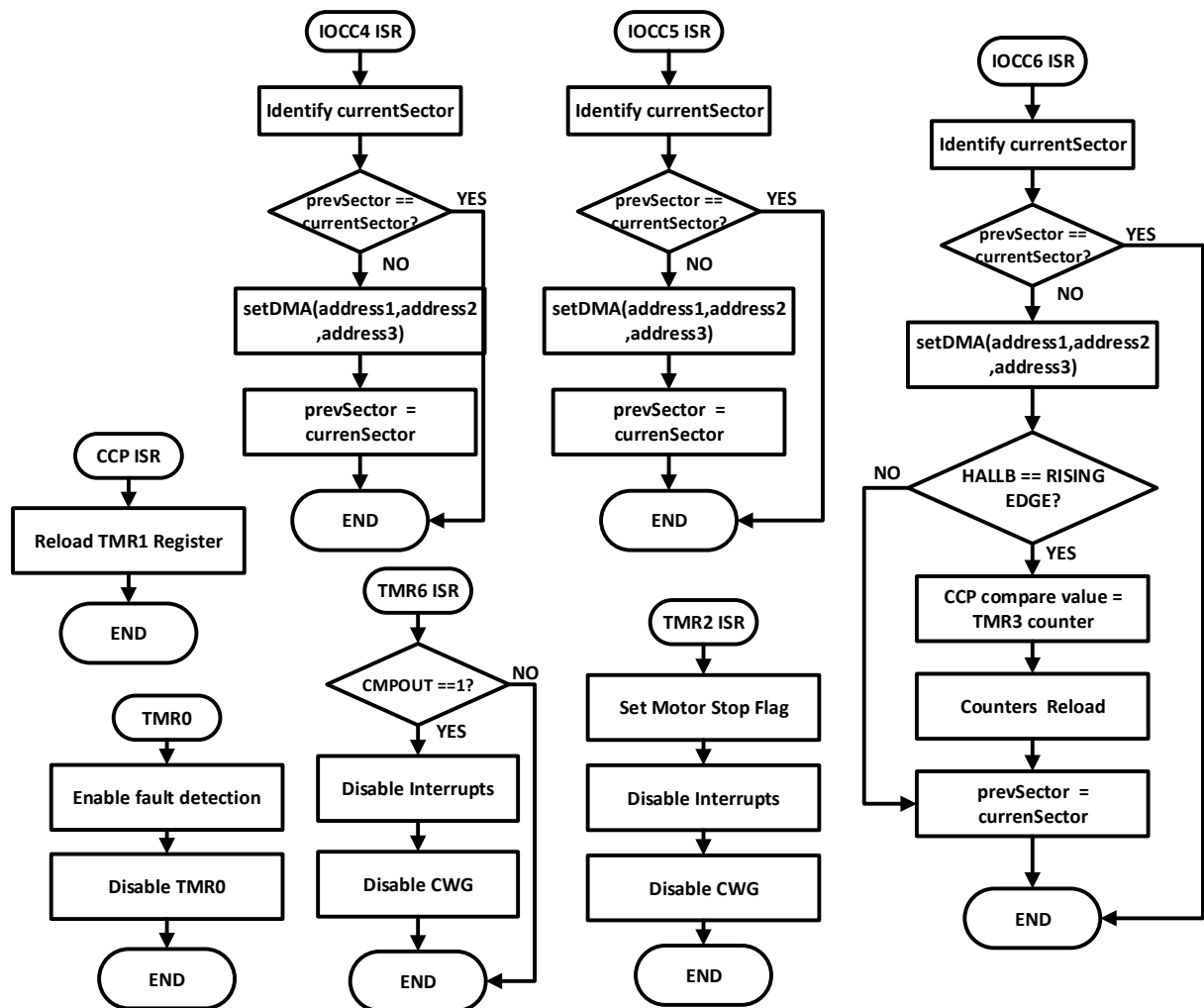
const uint8_t revDriveTable_CWG1 [7] = {0x00, 0x10, 0x00, 0x10, 0x02, 0x00, 0x02};
const uint8_t revDriveTable_CWG2 [7] = {0x00, 0x00, 0x02, 0x02, 0x10, 0x10, 0x00};
const uint8_t revDriveTable_CWG3 [7] = {0x00, 0x02, 0x10, 0x00, 0x00, 0x02, 0x10};
```

完成机械旋转时，**CWG** 外设将变为半桥模式，以准备切换到正弦驱动。用于角度转换的定时器全部重载，并且 **IOC** **ISR** 上的函数也将替换为相应的正弦函数。

6.3 中断源

初始化后，主循环中只对输入按钮进行监视。其他进程和函数由 **CIP** 执行，或由中断资源触发。当满足某个条件时，中断标志会置 1，同时将执行其对应的中断服务程序。图 6-3 说明了如何执行这些程序。

图 6-3. ISR 流程图



固件中使用的所有外设均使用 MPLAB® 代码配置器 (MPLAB Code Configurator, MCC) 进行配置。有关如何配置所有外设的详细步骤, 请参见附录 B: MPLAB 代码配置器 (MCC) 外设置。

6.3.1 霍尔 ISR

每个霍尔传感器均连接到 MCU 上已允许电平变化中断 (Interrupt-on-Change, IOC) 的数字引脚。如果检测到霍尔跳变 (下降沿或上升沿), 将调用霍尔 ISR。霍尔状态以提供当前扇区位置的 3 位逻辑状态表示。在启动期间, 梯形函数设置为霍尔 ISR, 其中将执行六步换相。当完成机械旋转时, 霍尔 ISR 将替换为正弦函数。由于所有霍尔引脚均位于 PORTC 寄存器中, 因此将使用 (uint8_t) ((PORTC >> 4) & 0x07)。DMA 源地址将基于 currentSector0 设置。使用 SetDMA() 子程序时, 每个 DMA 通道的源地址都指向一组不同的正弦数据, 具体取决于检测到的位置。每个 PWM 模块都配置为单独传输, 共使用三个 DMA 通道。DMASELECT 寄存器应在配置单独通道之前设置。图 6-4 给出了霍尔 ISR 中用于配置 DMA 通道的 CheckSector() 子程序。

图 6-4. 检查分区间子程序

```

void CheckSector(void) {
    switch (currentSector) {
        case SECTOR1:
            if (FLAGS.direction == CCW) {
                SetDMA(&PWM_1, &PWM_5, &PWM_3);
            } else if (FLAGS.direction == CW) {
                SetDMA(&PWM_3, &PWM_5, &PWM_1);
            }
            break;
        case SECTOR2:
            if (FLAGS.direction == CCW) {
                SetDMA(&PWM_2, &PWM_6, &PWM_4);
            }
            else if (FLAGS.direction == CW) {
                SetDMA(&PWM_2, &PWM_4, &PWM_6);
            }
            break;
        case SECTOR3:
            if (FLAGS.direction == CCW) {
                SetDMA(&PWM_3, &PWM_1, &PWM_5);
            } else if (FLAGS.direction == CW) {
                SetDMA(&PWM_1, &PWM_3, &PWM_5);
            }
            break;
        case SECTOR4:
            if (FLAGS.direction == CCW) {
                SetDMA(&PWM_4, &PWM_2, &PWM_6);
            } else if (FLAGS.direction == CW) {
                SetDMA(&PWM_6, &PWM_2, &PWM_4);
            }
            break;
        case SECTOR5:
            if (FLAGS.direction == CCW) {
                SetDMA(&PWM_5, &PWM_3, &PWM_1);
            } else if (FLAGS.direction == CW) {
                SetDMA(&PWM_5, &PWM_1, &PWM_3);
            }
            break;
        case SECTOR6:
            if (FLAGS.direction == CCW) {
                SetDMA(&PWM_6, &PWM_4, &PWM_2);
            } else if (FLAGS.direction == CW) {
                SetDMA(&PWM_4, &PWM_6, &PWM_2);
            }
            break;
    }
}

```

霍尔 B 还用于识别电机角位置的参考速度。每次检测到霍尔 B 正边沿跳变时，TMR3 周期的值都会复制到 CCP 比较值并重载电机角位置中使用的所有定时器。

6.3.2 CCP ISR

每次 Timer1 寄存器与比较值寄存器发生匹配时，CCP ISR 标志均会置 1。这会触发 DMA 控制器使下一个要装入 PWM 占空比寄存器的数组元素的源地址递增。当 DMA 在后台执行此任务时，CCP ISR 将重载 Timer1 寄存器。

6.3.3 HLT TMR 2 ISR

如果检测到一段时间内缺失霍尔脉冲并且 PR2 匹配，则 TMR2 标志将置 1。电机随后切换为停转状态，通过调用 `stopMotor()` 函数并禁止中断来触发电机停止运转。

6.3.4 TMR6 ISR

在电机运行期间，使用比较器（CMP）外设持续监视电流。如果 V- 大于 V+（通过 DAC 寄存器设置），则比较器的输出设置为数字高电平。每 50 ms，TMR6 ISR 便检查一次比较器输出。如果 CMP 输出置 1，将调用 `stopMotor()` 函数，从而禁止所有中断和 CWG 外设，让电机停止运转。

6.3.5 TMR0 ISR

在电机启动期间，绕组为闭合电路，会导致系统消耗较高的电流。这种现象称为浪涌电流，可能会导致过流检测器误触发。通过使用 TMR0，可在使能故障检测功能之前插入不到一秒的延时。当 Timer0 标志置 1 时，将使能比较器和 TMR2 外设并允许相应的中断。还将禁止 TMR0，使其在电机运行期间仅执行一次，并且仅在另一次启动期间使用。

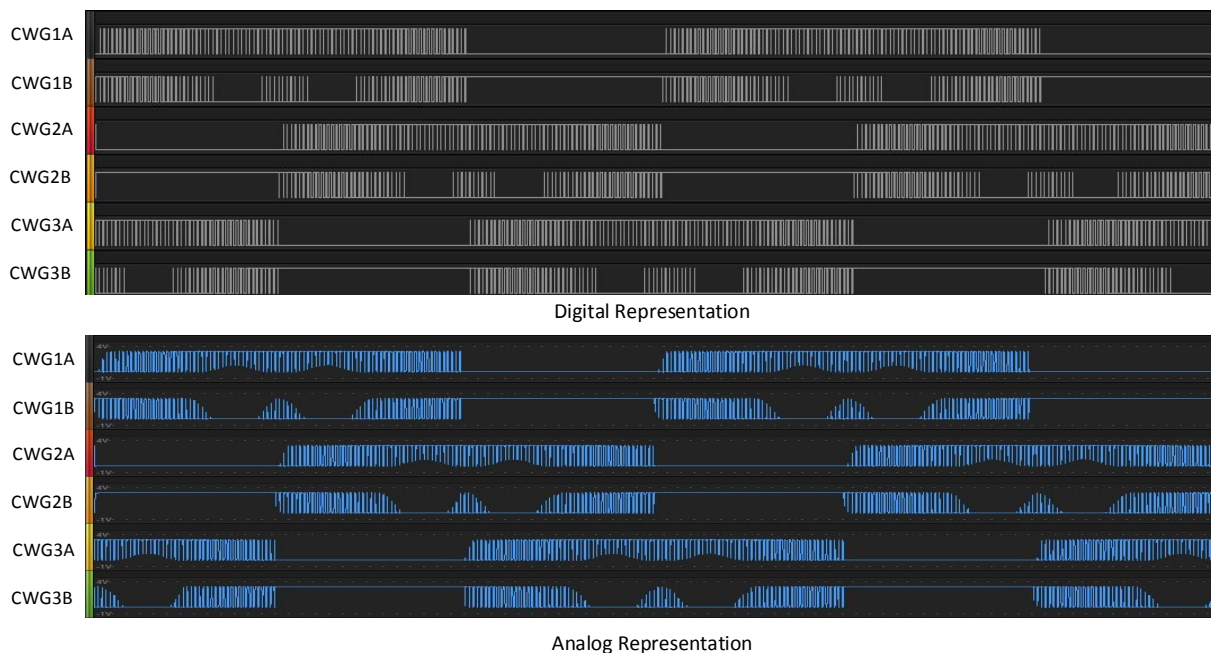
7. 正弦驱动三相 BLDC 性能

为了展示本应用的可行性，我们将捕捉驱动电压信号以及电机电流和电压。通过观察比较正弦驱动与梯形驱动的输出电压和电流。

7.1 三相 BLDC 的 CWG 驱动信号

图 7-1 给出了运行期间由 CWG 模块提供给功率级的驱动信号。

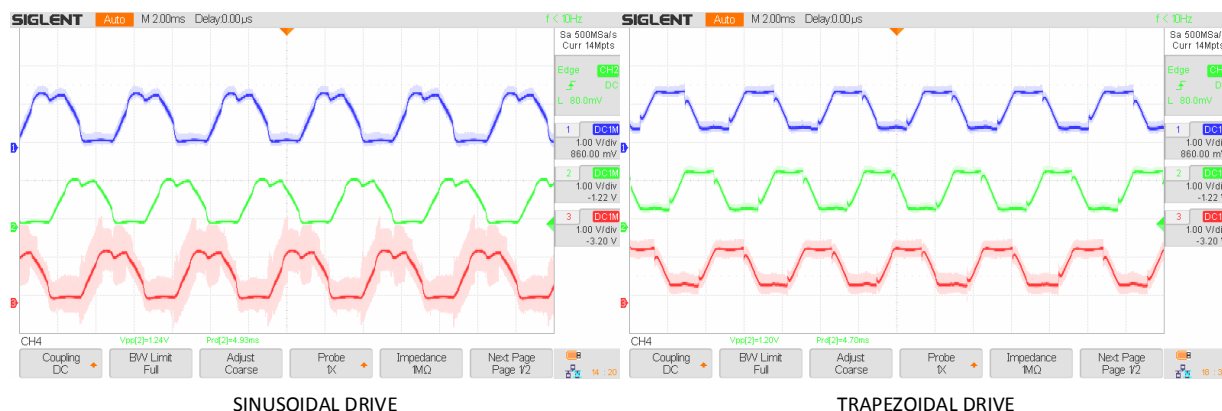
图 7-1. CWG 驱动信号



7.2 相电压和线路电流

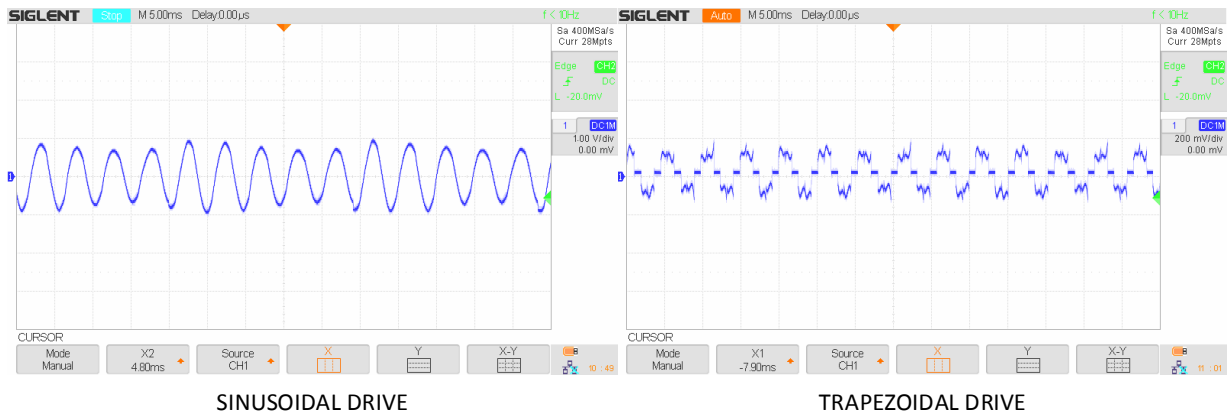
图 7-2 给出了正弦驱动和梯形驱动下三相 BLDC 各绕组的相电压。三个示波器探头分别连接到电机各个绕组的分压器。通过观察可以发现各个驱动相电压之间的差异。正弦驱动产生的波形呈马鞍形状（其中开关电压平滑变化），而梯形驱动产生的波形为梯形。

图 7-2. 相电压



使用电流探头获取梯形电流驱动和正弦电流驱动的相电流，如图 7-3 所示。两种驱动均以 2400 RPM 速度运行。在梯形驱动中可以观察到纹波。相比之下，正弦驱动提供的线路电流较为平滑。梯形驱动中的纹波较为明显。正弦驱动电流波形类似于正弦波，其中纹波和噪声均有所降低。

图 7-3. 相电流

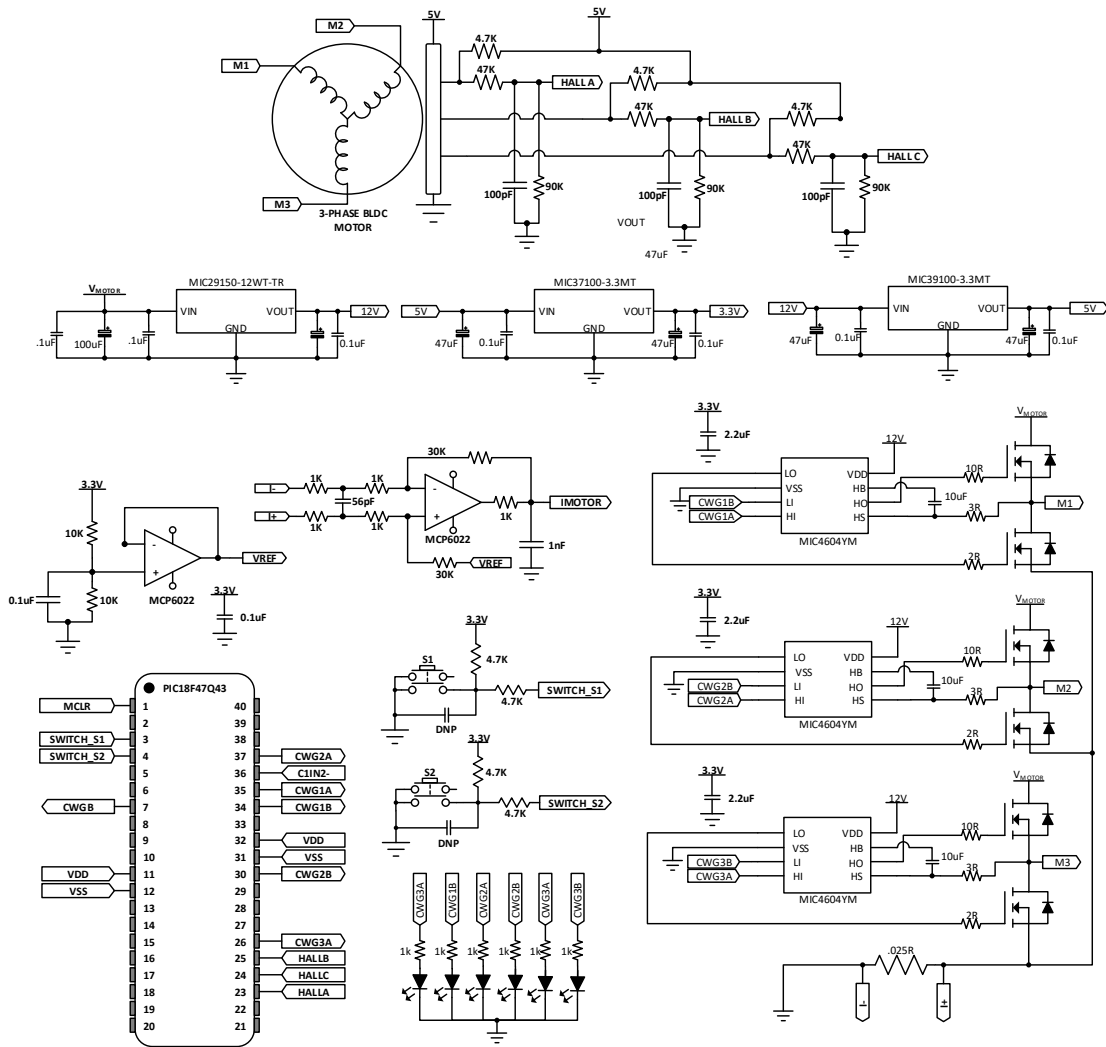


8. 结论

使用 DMA 直接将查找表（LUT）中的数据传输到 PWM，可以减少正弦驱动 BLDC 电机控制中的实时计算。这种方法虽然由于无法修改 LUT 中的数据而缺乏速度控制，但非常适合需要非速度变量的应用和低噪声应用。通过使用 CIP DMA 控制器可以减少函数和指令，以便 CPU 能够执行其他任务。只需使用低成本的 8 位单片机 PIC18-Q43，便可成功执行正弦电流驱动 BLDC 电机操作。此外，实现的故障检测可在检测到可能导致操作失败的事件时自动停止操作。

9.

图 9-1. 原理图 A



10. 附录 B: MPLAB®代码配置 (MCC) 外设配置

本章介绍如何通过 MPLAB®代码配置器 (MCC) 对本应用笔记中使用的外设进行初始化和配置。MCC 是 MPLAB® X IDE 的插件工具，提供了一个图形环境来执行外设配置。MCC 以 C 代码生成驱动程序，用于初始化外设并提供可在固件上调用的函数。有关如何在 MPLAB X IDE 中安装和设置 MCC 的更多信息，请参见《MPLAB®代码配置器 v3.xx 用户指南》(DS40001829B_CN)。

以下步骤提供了本应用笔记中使用的每个 PIC18-Q43 外设的 MCC 设置。

1. 在系统模块中，选择频率为 64 MHz 的 HFINTOSC 作为时钟源。
2. 对于电机角位置，配置 TMR4、CLC1、TMR3 CCP1 和 TMR1。将 TMR4 的时钟源设置为 $F_{OSC}/4$ 。要在霍尔周期内提供 192 个中断，应根据公式 3-2 将 TMR4 周期设置为 12.0625 μs 。
3. 使用 CLC1 模块，将 TMR4 输入重定向到 TMR3 输入。在 CLC1 的 AND-OR 模式下，选择 TMR4 输出作为两个 OR 门的输入。该配置用作向 TMR3 时钟提供输入的缓冲逻辑。
4. 将 CLC1 的输出用作 TMR3 的输入。每次 TMR4 输出置 1 时，TMR3 便会递增 1。其目的是对霍尔周期内的 TMR4 周期进行计数。每次检测到霍尔周期时，TMR3 计数器寄存器的值都会复制到 CCP 的比较寄存器，然后重载初始值。
5. 将 CCP1 模块设置为比较模式。选择 TMR1 模块作为定时器，依此不断地对比较寄存器值进行比较。为脉冲清除定时器选择比较工作模式。允许 CCP 中断以提供 ISR，从而能够在每次发生匹配时重载 TMR1。
6. 选择 $F_{OSC}/4$ 作为 TMR1 时钟源，以使用 TMR4 执行硬件除法。
7. 驱动信号发生器要求配置 DMA 控制器、PWM 和 CWG 模块。对于 PWM 模块，每个桥使用单片。选择 HFINTOSC 作为每个 PWM 模块的时钟源（没有预分频值）。选择中心对齐模式作为工作模式。将请求的频率设置为 40 kHz，以便为占空比提供 10 位分辨率。在寄存器的选项卡中，将 PWMxLDS 设置为相应的 DMA 通道，以便在每次 DMA 成功发送一个元素时将占空比值自动重载到 PWM 输出。
8. 禁止 CWG 模块，因为电机在停止条件下初始化。将 PWMxOUT1 寄存器设置为工作输入源。选择转向模式作为输出模式，因为电机在梯形驱动下启动，但如果电机达到稳定运转速度，则在固件执行期间切换为半桥模式。将下降和上升事件的死区计数设置为 63-64 个计数（使用 HFINTOSC 时钟源时）。
9. DMA 控制器通道最初处于禁止状态，之后将通过固件手动使能。将 DMAxCON1 寄存器中的 DMODE 和 SMODE 位设置为“递增”。选择闪存程序存储器作为 DMAxCON1 寄存器中的源地址。对于 DMA 源地址，输入相应 PWM 片寄存器的地址单元。在 MPLAB® X IDE 中，可通过单击 **Window > Target Memory Views > SFR**（窗口 > 目标存储器视图 > SFR）轻松识别寄存器的地址。随后将弹出 SFR 选项卡，显示寄存器及其地址的列表。将 DMAxDSZ 寄存器设置为 2，因为 PWM 占空比寄存器的大小为 2 字节。对于源地址大小，将 DMAxSSZ 寄存器设置为 0x40。选择 CCP1 作为 DMAxSIRQ 寄存器中的中断请求源。
10. 对于故障检测功能，使用 TMR0 在使能故障检测之前插入延时。将时钟预分频比设置为 1:32，后分频比设置为 1:10。将时钟源设置为 LFINTOSC。在使能故障检测功能之前，将请求的周期设置为所需延序列，即 500 ms。将 TMR0 初始化为禁止状态，因为设置的时间将从电机运转后开始。
11. 将 DAC 输出值设置为 2.5V。对于正参考电压和负参考电压，分别选择 V_{DD} 和 V_{SS} 。在 DACOUT1 寄存器中禁止 DAC 输出。CMP 模块的正输入使用 DACOUT1 作为参考电压。选择 CIN2-作为 CMP 模块的负输入。选择反相输出极性，以在负输入大于 DACOUT 值时提供高电平输出。使用 TMR6，每 50 ms 检查一次比较器输出。如果 CMP 输出状态为高电平，则会触发电机停止运转。要应用此方案，应允许 TMR6 中断并选择 LFINTOSC 作为时钟源。将 TMR6 的预分频值更改为 1:64，并将定时器周期设置为 50 ms。
12. 对于停转检测，使用 TMR2 的 HLT 模式。选择 T2CKIPPS 引脚作为外部复位源，以使用霍尔 B 信号检测停转条件。选择“resets at rising TMR2_ers”（在 TMR2_ers 上升时复位）作为启动/复位选项，以便在 TMR2 标志置 1 前每次检测到霍尔信号时复位定时器。TMR2 周期指示一段特定的时间，在这段时间内，如果未检测到霍尔信号，则将触发中断。允许 TMR2 中断。
13. 按图 10-1 所示配置外设的输入和输出引脚。

图 10-1. 引脚管理器和引脚模块配置

Pin Module

Easy Setup

Registers

Selected Package : PDIP40

Pin Name	Module	Function	Custom Name	Start High	Analog	Output	WPU	OD	IOC
RA1	Pin Module	GPIO	SWITCH_S1	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	none
RA2	Pin Module	GPIO	SWITCH_S2	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	none
RA5	CWG3	CWG3B		<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	none
RB1	CWG1	CWG1B		<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	none
RB2	CWG1	CWG1A		<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	none
RB3	CMP1	C1IN2-		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	none
RB4	CWG2	CWG2A		<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	none
RB6	PWM3	PWM3ERS		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	none
RC2	PWM2	PWM2ERS		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	none
RC2	PWM3	PWMIN0		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	none
RC2	PWM1	PWM1ERS		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	none
RC2	PWM2	PWMIN0		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	none
RC2	PWM1	PWMIN0		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	none
RC3	CCP1	CCP1		<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	none
RC4	Pin Module	GPIO	HALLA	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	any
RC5	Pin Module	GPIO	HALLC	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	any
RC6	Pin Module	GPIO	HALLB	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	any
RC6	TMR2	T2IN	HALLB	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	any
RC7	CWG3	CWG3A		<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	none
RD7	CWG2	CWG2B		<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

Pin Manager: Grid View

Package:	PDIP40	Pin No:	2	3	4	5	6	7	14	13	33	34	35	36	37	38	39	40	15	16	17	18	23	24	25	26	19	20	21	22	27	28	29	30	8	9	10	1
			Port A								Port B								Port C								Port D								Port E			
Module	Function	Direction	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3
CCP1	CCP1	output																																				
CLC1	CLC1	output																																				
CLCx	-	-																																				
CMP1	C1IN0+	input																																				
	C1IN1+	input																																				
	C1INx-	input																																				
	C1OUT	output																																				
CWG1	CWG1	input																																				
	CWG1A	output																																				
	CWG1B	output																																				
	CWG1C	output																																				
CWG2	CWG1D	output																																				
	CWG2	input																																				
	CWG2A	output																																				
	CWG2B	output																																				
CWG3	CWG2C	output																																				
	CWG2D	output																																				
	CWG3	input																																				
	CWG3A	output																																				
Pin Module	CWG3B	output																																				
	CWG3C	output																																				
	CWG3D	output																																				
Pin Module	GPIO	input																																				
	GPIO	output																																				
	RESET	MCLR	input																																			
TMR0	T0CKI	input																																				
	TMR0	output																																				
TMR1	T1CKI	input																																				
	T1G	input																																				
TMR2	T2IN	input																																				
	T3CKI	input																																				
TMR3	T3G	input																																				
	T4IN	input																																				

在所需的外设全部设置完毕后，单击资源管理选项卡中的“Generate code”（生成代码）。随后将生成外设驱动程序和 `main.c`，其中将基于您的配置完成所有外设的初始化。

11. 附录 C 源代码列表

最新软件版本可从 Microchip 网站 (www.microchip.com) 下载。用户将会找到本应用笔记电子版随附的源代码。

Microchip 网站

Microchip 网站 (www.microchip.com/) 为客户提供在线支持。客户可通过该网站方便地获取文件和信息。我们的网站提供以下内容：

- **产品支持**——数据手册和勘误表、应用笔记和示例程序、设计资源、用户指南以及硬件支持文档、最新的软件版本以及归档软件
- **一般技术支持**——常见问题解答 (FAQ)、技术支持请求、在线讨论组以及 Microchip 设计伙伴计划成员名单
- **Microchip 业务**——产品选型和订购指南、最新 Microchip 新闻稿、研讨会和活动安排表、Microchip 销售办事处、代理商以及工厂代表列表

产品变更通知服务

Microchip 的产品变更通知服务有助于客户了解 Microchip 产品的最新信息。注册客户可在他们感兴趣的某个产品系列或开发工具发生变更、更新、发布新版本或勘误表时，收到电子邮件通知。

欲注册，请访问 www.microchip.com/pcn，然后按照注册说明进行操作。

客户支持

Microchip 产品的用户可通过以下渠道获得帮助：

- 代理商或代表
- 当地销售办事处
- 应用工程师 (ESE)
- 技术支持

客户应联系其代理商、代表或 ESE 寻求支持。当地销售办事处也可为客户提供帮助。本文档后附有销售办事处的联系方式。

也可通过 www.microchip.com/support 获得网上技术支持。

Microchip 器件代码保护功能

请注意以下有关 Microchip 产品代码保护功能的要点：

- Microchip 的产品均达到 Microchip 数据手册中所述的技术规范。
- Microchip 确信：在正常使用且符合工作规范的情况下，Microchip 系列产品非常安全。
- Microchip 注重并积极保护其知识产权。严禁任何试图破坏 Microchip 产品代码保护功能的行为，这种行为可能会违反《数字千年版权法案》(Digital Millennium Copyright Act)。
- Microchip 或任何其他半导体厂商均无法保证其代码的安全性。代码保护并不意味着我们保证产品是“牢不可破”的。代码保护功能处于持续发展。Microchip 承诺将不断改进产品的代码保护功能。

法律声明

提供本文档的中文版本仅为了便于理解。请勿忽视文档中包含的英文部分，因为其中提供了有关 Microchip 产品性能和使用情况的有用信息。Microchip Technology Inc. 及其分公司和相关公司、各级主管与员工及事务代理机构对译文中可能存在的任何差错不承担任何责任。建议参考 Microchip Technology Inc. 的英文原版文档。

本出版物及其提供的信息仅适用于 Microchip 产品，包括设计、测试以及将 Microchip 产品集成到您的应用中。以其他方式使用这些信息都将被视为违反条款。本出版物中的器件应用信息仅为您提供便利，将来可能会发生更新。如需额外的支持，请联系当地的 Microchip 销售办事处，或访问 <https://www.microchip.com/en-us/support/design-help/client-supportservices>。

Microchip “按原样”提供这些信息。Microchip 对这些信息不作任何明示或暗示、书面或口头、法定或其他形式的声明或担保，包括但不限于针对非侵权性、适销性和特定用途的适用性的暗示担保，或针对其使用情况、质量或性能的担保。

在任何情况下，对于因这些信息或使用这些信息而产生的任何间接的、特殊的、惩罚性的、偶然的或间接的损失、损害或任何类型的开销，Microchip 概不承担任何责任，即使 Microchip 已被告知可能发生损害或损害可以预见。在法律允许的最大范围内，对于因这些信息或使用这些信息而产生的所有索赔，Microchip 在任何情况下所承担的全部责任均不超出您为获得这些信息向 Microchip 直接支付的金额（如有）。如果将 Microchip 器件用于生命维持和/或生命安全应用，一切风险由买方自负。买方同意在由此引发任何一切损害、索赔、诉讼或费用时，会维护和保障 Microchip 免于承担法律责任。除非另外声明，在 Microchip 知识产权保护下，不得暗或以其他方式转让任何许可证。

商标

Microchip 的名称和徽标组合、Microchip 徽标、Adaptec、AnyRate、AVR、AVR 徽标、AVR Freaks、BesTime、BitCloud、CryptoMemory、CryptoRF、dsPIC、flexPWR、HELDO、IGLOO、JukeBlox、KeeLoq、Kleer、LANCheck、LinkMD、maXStylus、maXTouch、MediaLB、megaAVR、Microsemi、Microsemi 徽标、MOST、MOST 徽标、MPLAB、OptoLyzer、PIC、picoPower、PICSTART、PIC32 徽标、PolarFire、Prochip Designer、QTouch、SAM-BA、SenGenuity、SpyNIC、SST、SST 徽标、SuperFlash、Symmetricom、SyncServer、Tachyon、TimeSource、tinyAVR、UNI/O、Vectron 及 XMEGA 均为 Microchip Technology Incorporated 在美国和其他国家或地区的注册商标。

AgileSwitch、APT、ClockWorks、The Embedded Control Solutions Company、EtherSynch、Flashtec、Hyper Speed Control、HyperLight Load、IntelliMOS、Libero、motorBench、mTouch、Powermite 3、Precision Edge、ProASIC、ProASIC Plus、ProASIC Plus 徽标、Quiet-Wire、SmartFusion、SyncWorld、Temux、TimeCesium、TimeHub、TimePictra、TimeProvider、TrueTime、WinPath 和 ZL 均为 Microchip Technology Incorporated 在美国的注册商标。

Adjacent Key Suppression、AKS、Analog-for-the-Digital Age、Any Capacitor、AnyIn、AnyOut、Augmented Switching、BlueSky、BodyCom、CodeGuard、CryptoAuthentication、CryptoAutomotive、CryptoCompanion、CryptoController、dsPICDEM、dsPICDEM.net、Dynamic Average Matching、DAM、ECAN、Espresso T1S、EtherGREEN、GridTime、IdealBridge、In-Circuit Serial Programming、ICSP、INICnet、Intelligent Paralleling、Inter-Chip Connectivity、JitterBlocker、Knob-on-Display、maxCrypto、maxView、memBrain、Mindi、MiWi、MPASM、MPF、MPLAB Certified 徽标、MPLIB、MPLINK、MultiTRAK、NetDetach、NVM Express、NVMe、Omniscient Code Generation、PICDEM、PICDEM.net、PICKit、PICtail、PowerSmart、PureSilicon、QMatrix、REAL ICE、Ripple Blocker、RTAX、RTG4、SAM-ICE、Serial Quad I/O、simpleMAP、SimpliPHY、SmartBuffer、SmartHLS、SMART-I.S.、storClad、SQL、SuperSwitcher、SuperSwitcher II、Switchtec、SynchroPHY、Total Endurance、TSHARC、USBCheck、VariSense、VectorBlox、VeriPHY、ViewSpan、WiperLock、XpressConnect 和 ZENA 均为 Microchip Technology Incorporated 在美国和其他国家或地区的商标。

SQTP 为 Microchip Technology Incorporated 在美国的服务标记。

Adaptec 徽标、Frequency on Demand、Silicon Storage Technology、Symmcom 和 Trusted Time 均为 Microchip Technology Inc. 在除美国外的国家或地区的注册商标。

GestIC 为 Microchip Technology Inc. 的子公司 Microchip Technology Germany II GmbH & Co. KG 在除美国外的国家或地区的注册商标。

在此提及的所有其他商标均为各持有公司所有。

© 2022, Microchip Technology Incorporated 及其子公司版权所有。

ISBN: 978-1-5224-9057-9

质量管理体系

有关 Microchip 的质量管理体系的信息，请访问 www.microchip.com/quality。

全球销售及服务中心

美洲	亚太地区	亚太地区	欧洲
公司总部 2355 West Chandler Blvd. Chandler, AZ 85224-6199 电话: 480-792-7200 传真: 480-792-7277 技术支持: www.microchip.com/support 网址: www.microchip.com 亚特兰大 德卢斯, 佐治亚州 电话: 678-957-9614 传真: 678-957-1455 奥斯汀, 德克萨斯州 电话: 512-257-3370 波士顿 韦斯特伯鲁, 马萨诸塞州 电话: 774-760-0087 传真: 774-760-0088 芝加哥 艾塔斯卡, 伊利诺伊州 电话: 630-285-0071 传真: 630-285-0075 达拉斯 阿迪森, 德克萨斯州 电话: 972-818-7423 传真: 972-818-2924 底特律 诺维, 密歇根州 电话: 248-848-4000 休斯顿, 德克萨斯州 电话: 281-894-5983 印第安纳波利斯 诺布尔斯维尔, 印第安纳州 电话: 317-773-8323 传真: 317-773-5453 电话: 317-536-2380 洛杉矶 米慎维荷, 加利福尼亚州 电话: 949-462-9523 传真: 949-462-9608 电话: 951-273-7800 罗利, 北卡罗来纳州 电话: 919-844-7510 纽约, 纽约州 电话: 631-435-6000 圣何塞, 加利福尼亚州 电话: 408-735-9110 电话: 408-436-4270 加拿大 - 多伦多 电话: 905-695-1980 传真: 905-695-2078	澳大利亚 - 悉尼 电话: 61-2-9868-6733 中国 - 北京 电话: 86-10-8569-7000 中国 - 成都 电话: 86-28-8665-5511 中国 - 重庆 电话: 86-23-8980-9588 中国 - 东莞 电话: 86-769-8702-9880 中国 - 广州 电话: 86-20-8755-8029 中国 - 杭州 电话: 86-571-8792-8115 中国 - 香港特别行政区 电话: 852-2943-5100 中国 - 南京 电话: 86-25-8473-2460 中国 - 青岛 电话: 86-532-8502-7355 中国 - 上海 电话: 86-21-3326-8000 中国 - 沈阳 电话: 86-24-2334-2829 中国 - 深圳 电话: 86-755-8864-2200 中国 - 苏州 电话: 86-186-6233-1526 中国 - 武汉 电话: 86-27-5980-5300 中国 - 西安 电话: 86-29-8833-7252 中国 - 厦门 电话: 86-592-2388138 中国 - 珠海 电话: 86-756-3210040	印度 - 班加罗尔 电话: 91-80-3090-4444 印度 - 新德里 电话: 91-11-4160-8631 印度 - 浦那 电话: 91-20-4121-0141 日本 - 大阪 电话: 81-6-6152-7160 日本 - 东京 电话: 81-3-6880-3770 韩国 - 大邱 电话: 82-53-744-4301 韩国 - 首尔 电话: 82-2-554-7200 马来西亚 - 吉隆坡 电话: 60-3-7651-7906 马来西亚 - 槟榔屿 电话: 60-4-227-8870 菲律宾 - 马尼拉 电话: 63-2-634-9065 新加坡 电话: 65-6334-8870 台湾地区 - 新竹 电话: 886-3-577-8366 台湾地区 - 高雄 电话: 886-7-213-7830 台湾地区 - 台北 电话: 886-2-2508-8600 泰国 - 曼谷 电话: 66-2-694-1351 越南 - 胡志明市 电话: 84-28-5448-2100	奥地利 - 韦尔斯 电话: 43-7242-2244-39 传真: 43-7242-2244-393 丹麦 - 哥本哈根 电话: 45-4485-5910 传真: 45-4485-2829 芬兰 - 埃斯波 电话: 358-9-4520-820 法国 - 巴黎 电话: 33-1-69-53-63-20 传真: 33-1-69-30-90-79 德国 - 加兴 电话: 49-8931-9700 德国 - 哈恩 电话: 49-2129-3766400 德国 - 海布隆 电话: 49-7131-72400 德国 - 卡尔斯鲁厄 电话: 49-721-625370 德国 - 慕尼黑 电话: 49-89-627-144-0 传真: 49-89-627-144-44 德国 - 罗森海姆 电话: 49-8031-354-560 以色列 - 若那那市 电话: 972-9-744-7705 意大利 - 米兰 电话: 39-0331-742611 传真: 39-0331-466781 意大利 - 帕多瓦 电话: 39-049-7625286 荷兰 - 德卢内市 电话: 31-416-690399 传真: 31-416-690340 挪威 - 特隆赫姆 电话: 47-72884388 波兰 - 华沙 电话: 48-22-3325737 罗马尼亚 - 布加勒斯特 电话: 40-21-407-87-50 西班牙 - 马德里 电话: 34-91-708-08-90 传真: 34-91-708-08-91 瑞典 - 哥德堡 电话: 46-31-704-60-40 瑞典 - 斯德哥尔摩 电话: 46-8-5090-4654 英国 - 沃金厄姆 电话: 44-118-921-5800 传真: 44-118-921-5820