

无刷直流 (BLDC) 电机的无传感器梯形控制

Bilal Akin and Manish Bhardwaj

摘要

这份应用报告提出了使用 TMS320F2803x 微控制器控制无刷直流电机的解决方案。TMS320F280x 器件是 C2000™ 系列微控制器的部件，它通过减少系统组件实现用于三相电机的智能控制器的成本有效设计，并且提高了效率。使用这些器件，有可能实现更加精确的控制算法。下面给出了一个完整的解决方案：可在本文档中找到控制结构、电源硬件拓扑、控制硬件和对能量转换效率的评价。

这份应用报告涵盖了以下内容：

- 梯形 BLDC 电机控制原理的理论背景
- 基于模块化软件块的递增构建级
- 试验结果

内容

1	简介	2
2	BLDC 电机	3
3	BLDC 电机控制	4
4	系统拓扑	6
5	32 位 C2000 控制器针对数字电机控制 (DMC) 的优势	10
6	TI 文献和数字电机控制 (DMC) 库	11
7	硬件配置 (HVD MC R1.1 套件)	15
8	递增系统构建用于无传感器 BLDC 项目	18

图片列表

1	具有一个永磁对极转子的三相同步电机.....	3
2	针对一个 BLDC 电机的速度和电流控制环路配置	5
3	两相打开运行和转矩纹波内的电气波形.....	6
4	正弦波电机内的转矩纹波被控制为一个 BLDC.....	6
5	三相反相器	6
6	依据 PWM 占空比的分流电阻器压降（软斩波）	7
7	HDMC 套件上的反电动势感测.....	8
8	定子端子电气模型	9
9	典型电流和反电动势波形正弦波 BLDC 电机驱动	9
10	一个 3 相 BLDC 驱动实现	13
11	BLDC 电机霍尔传感器控制的总体方框图	13
12	软件流	14
13	使用交流电源生成直流总线电源	16
14	使用外部直流电源为反相器生成直流总线	17
15	PWM 输出：PWM 1（黄色），PWM 2（粉色）和 PWM 5（绿色），PWM 6（蓝色）	20
16	1 级 - 递增系统构建方框图	21

C2000, Code Composer Studio are trademarks of Texas Instruments.
All other trademarks are the property of their respective owners.

17	霍尔效应传感器的输出, 霍尔 A, B 和 C.....	23
18	PWMDAC 输出 BemfA, BemfB 和 BemfC (Vdcbus = 160V)	23
19	(a) mod6 计数器 (b) 脉冲输出, dlog.prescalar = 3	24
20	(a) mod6 计数器, (b) BemfA, (c) BemfB 和 (d)BemfC (dlog.prescalar = 25 而 Vdcbus = 160V)	24
21	2 级 - 递增系统构建方框图	25
22	(a) Neutral, (b) DebugBemf, (c) BemfA 和 (d)BemfB (VdcBus = 160V).....	27
23	3 级 - 递增系统构建方框图	28
24	(a) mod6 计数器, (b) BemfA, (c) BemfB 和 (d) BemfC (dlog.prescalar = 25 而 Vdcbus = 160V)	30
25	PWMDAC 输出 a) BemfA, (b) BemfB, (c) BemfC 和 (d) mod6 计数器 (Vdcbus = 160V).....	30
26	4 级 - 递增系统构建方框图	31
27	(a) mod6 Counter, (b)BemfA, (c) BemfB (c)BemfC (dlog.presalar = 25 而 Vdcbus = 160V).....	33
28	PWMDAC 输出 a) BemfA, (b) BemfB, (c)BemfC 和 (d) mod6 计数器 (Vdcbus = 160V)	33
29	5 级 - 递增系统构建方框图	34
30	(a) mod6 计数器, (b) BemfA, (c) BemfB (c)BemfC (无负载, 速度 0.25pu 时, Vdcbus = 160V)	35
31	PWMDAC 输出 a) BemfA, (b) BemfB, (c)BemfC 和 (d) mod6 计数器 (无负载, 速度 0.25pu 时, Vdcbus = 160V)	36
32	6 级 - 递增系统构建方框图	37

图表列表

1	BLDC 和 PMSM 电机间的比较	3
2	观察窗口变量	18
3	每个递增系统构建中的已测试模块	18

1 简介

经济吃紧和政府制定的新标准对电气系统提出了越来越严格的要求。新一代设备必须具有更高的性能参数, 诸如更加的效率和减少的电磁干扰。必须具有较高的系统灵活性以使市场改进更加方便, 并且减少开发时间。所有这些改进必须与减少系统成本同时实现。

无刷电机技术使实现这些技术规范变为可能。这样的电机将高可靠性与高效率组合在一起, 并且相对于有刷电机, 无刷电机的成本更低。本文档对无刷直流 (BLDC) 电机的使用进行了说明。虽然可将无刷特点应用于几种类型的电机 (交流同步电机、步进电机、开关磁阻电机和交流感应电机), BLDC 电机在传统上被定义为一个具有梯形反电动势波形形状的永磁同步电机。具有梯形反电动势和 (120 电角宽度) 矩形定子电流的永磁同步机器被广泛使用, 这是因为它们首先提供了以下优势, 假定电机具有纯粹梯形反电动势并且定子相位换相过程准确, 此电机产生的机械转矩是恒定的。其次, 无刷直流驱动器显示出极高的机械功率密度。

2 BLDC 电机

BLDC 电机是一个在转子（移动部分）上具有永磁体，而在定子上（固定部分）上具有绕组的交流同步电机。永磁体生成转子磁通，而通电的定子绕组产生磁极。转子（与一个条状磁体等效）被通电定子相位吸引。通过使用适当的序列来为定子相位供电，在定子上创建并保持了一个旋转磁场。转子的这个跟随定子上磁极的操作，是同步永磁电机内使用的基本操作。必须控制转子和旋转磁场间的导程以产生转矩，而且这个同步表示对转子位置的了解。

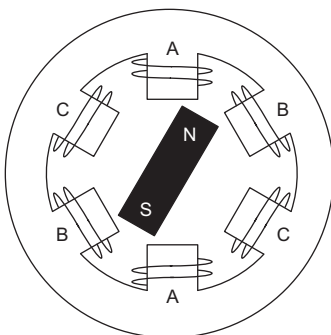


图 1. 具有一个永磁对极转子的三相同步电机

在定子端，三相电机是最常见的。这些电机在精确控制和控制定子电流所需的电力电子器件的数量之间提供一个良好的折衷。对于转子，大量的磁极通常针对同一电平的电流生成更大的转矩。另一方面，通过添加更多的磁体，会达到一个点，由于磁体之间所需的空間，在这个点上，转矩不再增加。制造成本也会随着磁极的数量的增加而增加。因此，磁极的数量是成本、转矩和体积之间的一个折衷因素。

可使用很多方法对永磁同步电机进行分类，但是应关注其中的两个依赖于反电动势系统配置的分类方法：无刷直流 (BLDC) 电机和永磁同步电机 (PMSM)。这个术语定义了同步电机反电动势的形状。BLDC 和 PMSM 电机在转子上都具有永磁体，但是它们的磁通分布和反电动势系统配置是不同的。为了获得同步电机的最佳性能，识别电机的类型十分重要，以按照下面部分描述的那样采用最合适的控制类型

表 1. BLDC 和 PMSM 电机间的比较

BLDC 和 PMSM 电机间的比较	
BLDC	PMSM
同步机器	同步机器
直流供电	正弦电流供电
梯形反电动势	正弦反电动势
定子磁通位置每 60°换相	持续定子磁通位置变化
同一时间只有两个相位打开 (ON)	有可能在同一时间打开三个相位
换相时的转矩纹波	换相时无转矩纹波
可闻范围内的低阶电流谐波	由正弦激发造成的更少谐波
由谐波含量造成的更高内核损耗	更低的内核损耗
更少的开关损耗	同一开关频率上的更高开关损耗
控制算法相对简单	控制算法是数学密集型

- 两个电机类型都是同步机器。它们之间的唯一区别是感应电压的形状不同，造成这一差异的原因是两个不同的定子线圈的接线方式。反电动势在 BLDC 电机中是梯形的，而在 PMSM 电机情况下为正弦曲线。
- BLDC 机器应该由正弦电流驱动，而 PMSM 由直流驱动，但是为了获得更佳的性能，PMSM 电机应该由正弦电流激发，BLDC 机器应该由直流激发。
- 很难用模拟技术实现几个电流传感器和正弦相位电流所要求的正弦电机的控制结构（硬件和软件）。因

此，出于成本和简单性的考虑，很多电机（类似于梯形的正弦曲线）由直流驱动（低分辨率位置传感器和单个低成本电流传感器），从而在效率和动态运行方式之间达到平衡。

- **C2000 数字信号处理器 (DSP)** 控制器处理的数字技术使得为每个电机类型选择适当的控制技术成为可能：处理能力被用来从机器获得最佳性能，并且减少系统成本。可能的选项是使用无传感器技术来减少传感器成本，或者甚至免除对它们的需要，并且复杂算法可有助于简化机械驱动链设计，从而降低了系统成本。

3 BLDC 电机控制

BLDC 电机的有效转矩和速度控制的关键是基于相对简单地转矩和反电动势等式，这与直流电机的那些等式相类似。反电动势振幅可表示为：

$$E = 2NlrBw$$

而转矩项为：

$$T = \left(\frac{1}{2} i^2 \frac{dL}{d\theta} \right) - \left(\frac{1}{2} B^2 \frac{dR}{d\theta} \right) + \left(\frac{4N}{\pi} Brli \right)$$

在这里， N 是每相位绕组匝数的数量， l 是转子的长度， r 是转子的内部半径， B 是转子磁体磁通密度， w 是电机的角速率， i 是相位电流， L 是相位电感， θ 是转子位置， R 是相位电阻。

转矩表达式中的头两项是寄生磁阻转矩分量。第三项产生互转矩，这是在 BLDC 电机情况下使用的转矩产出机制。总之，反电动势与电机速度直接成比例，而转矩产量大体上与相位电流直接成正比。这些因素导致图 2 中显示的 BLDC 电机速度控制系统配置。

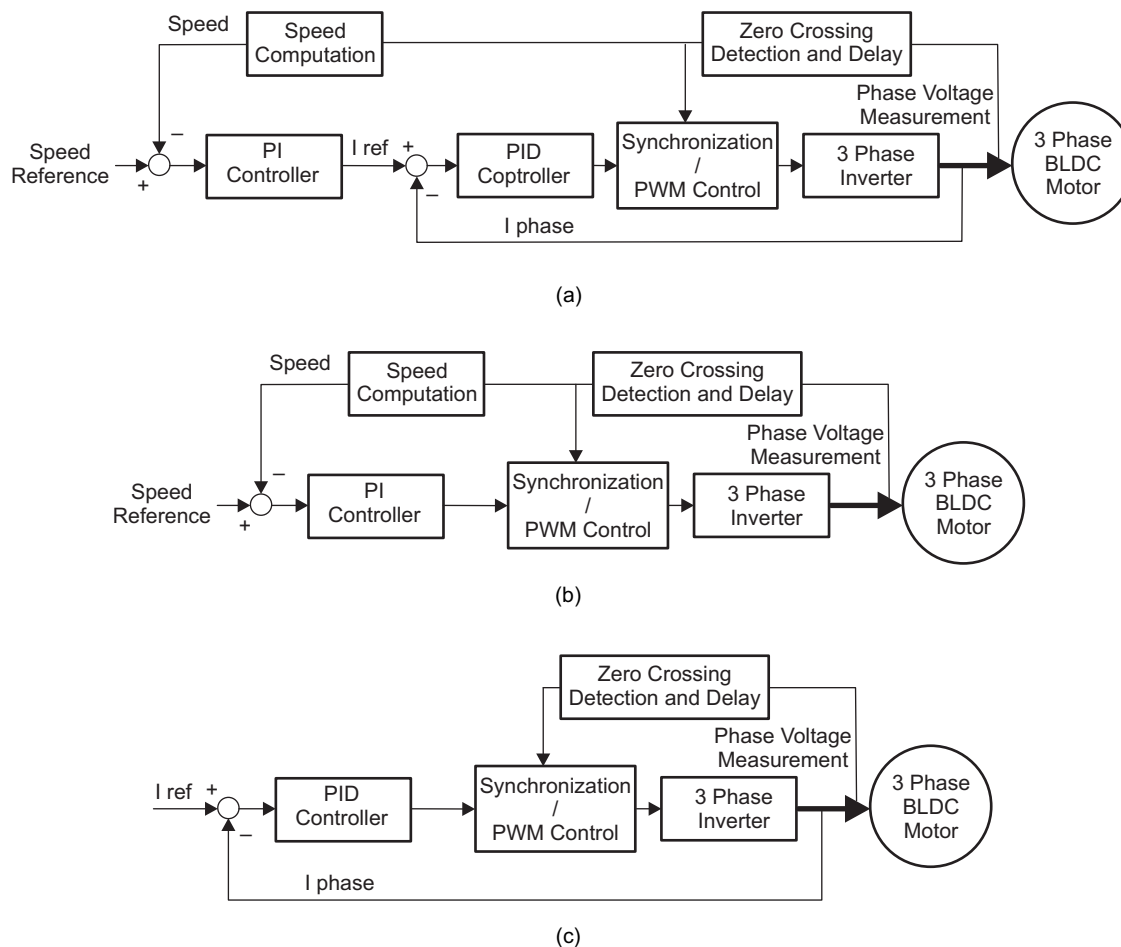


图 2. 针对一个 BLDC 电机的速度和电流控制环路配置

BLDC 电机的特点是两相位打开运行以控制反相器。在这个控制系统配置中，转矩产量遵循的原则是每次电流应该只流入三相中的两个，并且在反电动势过零区域中不应该产生转矩。图 3 描述了两相位打开运行中 BLDC 电机内的电气波形。

这个控制结构具有几个优势：

- 一次只需要控制一个电流
- 只需要一个电流传感器（或者如下一个部分详述的那样，只针对速度环路的话，无需电流传感器）
- 电流传感器的定位可将低成本传感器用作一个分路

BLDC 电机的原理始终是为相位对加电，这可产生最高的转矩。为了优化这个效果，反电动势形状为梯形。理论上，直流与梯形反电动势的组合使得有可能产生一个恒定转矩。实际上，不应在一个电机相位中即时建立电流；因此，转矩纹波出现在每次 60° 相位换相时。

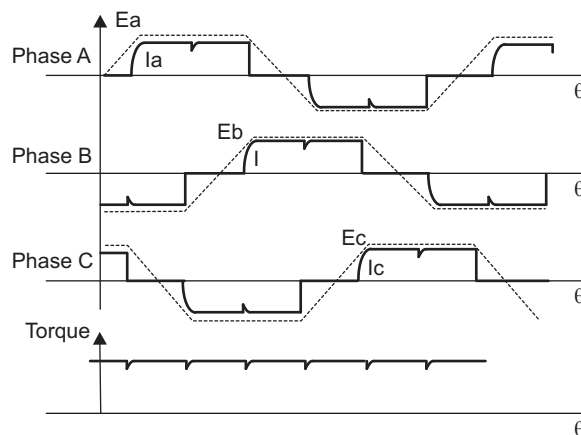


图 3. 两相打开运行和转矩纹波内的电气波形

如果使用的电机具有一个正弦反电动势形状，可采用这个控制，但是产生的转矩为：

- 非恒定但是由一个正弦波的部分组成。这是由于它是一个梯形电流控制策略和一个正弦波反电动势的组合。请牢记，由正弦波策略（三相打开）控制的正弦波反电动势波形电机产生恒定转矩。
- 产生的转矩值变弱。

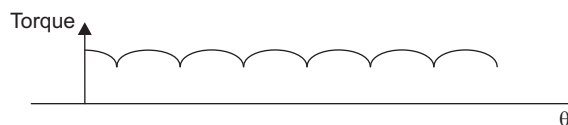


图 4. 正弦波电机内的转矩纹波被控制为一个 BLDC

4 系统拓扑

4.1 三相反相器

BLDC 电机控制由电机相位中生成的直流组成。这个控制被细分为两个独立操作：定子和转子磁通同步，以及电流值的控制。两个操作都通过图 5 中图示的三相反相器实现。

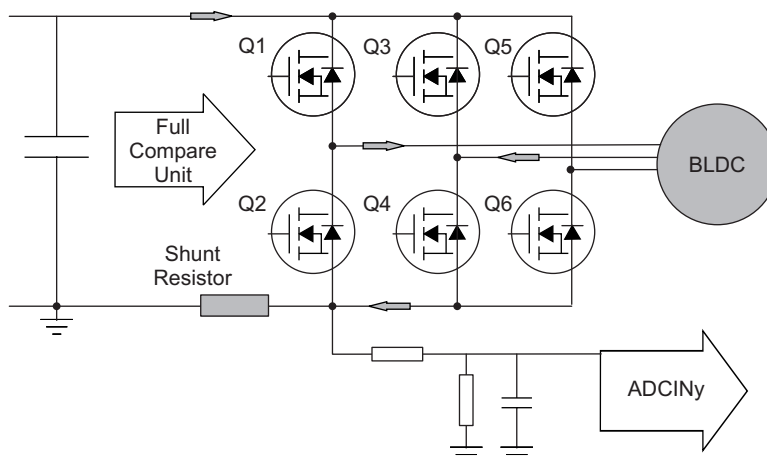


图 5. 三相反相器

磁通同步来自传感器，或无传感器技术提供的位置信息。根据位置，控制器确定必须驱动的晶体管（Q1 至 Q6）中的适当成对晶体管。可通过两种不同模式中的任何一个来将电流调节至固定 60° 基准。

- 脉宽调制 (PWM) 模式:

在一个固定频率上，电源电压由一个取决于电流误差的占空比斩波。因此，可以控制电流和电流的变化速率。两相电源持续时间受到两相换相角的限制。PWM 策略的主要优势是斩波频率是一个固定参数；因此，可闻和电磁噪声的滤除相对简单。

有两种处理驱动电流开关的方式：硬斩波和软斩波。在硬斩波技术中，相位晶体管由同一脉冲信号驱动：两个晶体管同时打开和关闭。这样，由于电力电子电路板只处理三个脉冲信号，它的设计更加简单，而且更加便宜。硬斩波操作的一个缺点是，相对于软斩波方法，它大大增加了电流纹波。

软斩波方法不但实现了对电流和电流变化速率的控制，而且也大大减少了电流纹波。在这个软斩波模式中，低侧晶体管在相位电源期间保持打开，高侧晶体管按照脉冲信号切换。在这个情况下，电力电子电路板必须处理 6 个 PWM 信号。

- 滞后模式:

在滞后类型电流稳压器中，功率晶体管根据电流与基准电流的关系（大于或小于）关闭和打开。误差被直接用来控制功率晶体管的状态。滞后控制器被用来将相位电流限制在预先设定的滞带内。由于电源电压是固定的，结果是开关频率随着电流误差的变化而变化。因此，电流斩波操作并不是一个固定斩波频率 PWM 技术。这个方法更加常见于电机速度和负载变化不大的驱动中，所以开关频率的变化很小。此外，硬斩波和软斩波系统配置都是可能的。由于耐受范围的宽度是一个设计参数，这个模式使得电流控制可以达到要求的精度，但是由于不断变化的开关频率，很难滤除可闻和电磁噪声。

4.2 电流感测

BLDC 控制的一个特点是电机内每次只有一个电流（两相位打开）。因此，无需将一个电流传感器放置在电机的每一个相位中；一个放置在线路反相器输入中的传感器使得每个相位的电流控制成为可能。而且，在接地线路上使用这个传感器，就不需要系统绝缘，并且可以使用一个低成本电阻器。它的值被设定，这样，它在达到电源电路板所允许的最大电流时激活集成过流保护。

每个电流测量导致一个在 PWM 周期的开始载入的全新 PWM 占空比。请注意，关闭期间，不论反相器是在硬斩波中驱动还是在软斩波模式中驱动，分流电阻器不感测这个电流。图 6 图示了软斩波模式中的分路电流，并且显示，在关闭操作中，逐渐减少的电流流经 M2 续流二极管以及保持关闭的 M4（所以关闭期间，在这个斩波模式下，在分路中没有可以观察到的电流 9）。这表示，有必要在 PWM 占空比的中间启动一个电流转换。

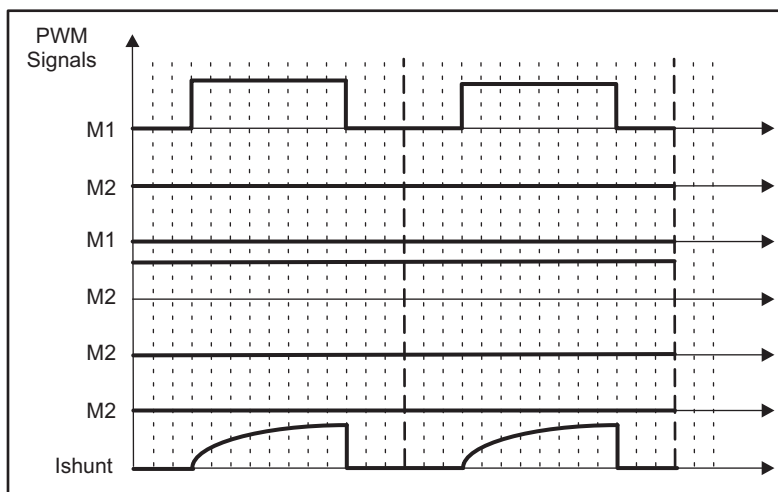


图 6. 依据 PWM 占空比的分流电阻器压降（软斩波）

在硬斩波模式下，关闭期间，M1 或 M4 都不驱动电流，因此，逐渐减少的相位电流从接地经由 M2 和 M3 续流二极管流经分流电阻器，通过电容器返回接地。在这个斩波模式中，由于出现负分路电压压降，分路上的相位电流有可能成指数下降。假定电源电路板和控制电路板都不支持负电压，有必要在打开的中间感测此电流。

实现 BLDC 速度控制需要执行三个控制层。最内层是获得转子位置以准确地换相定子磁通。一旦知道定子位置，必须生成和控制定子磁通的通量。假定定子磁通与流经定子线圈的电流成正比，定子磁通通量的控制与输入电流的控制等效。最外部的控制环路是速度调节环路。

4.3 反电动势过零点计算

电阻分压器电路被指定为这个电压感测电路的最大输出采用全部 ADC 转换范围。滤波电容器应该滤除斩波频率，所以只需要很小的值（在 nF 或甚至更小的范围内）。无传感器算法只基于三个电机端子电压测量值，并且只需要四个 ADC 输入线路。

在传感控制结构中，相位在转子的每次 60° 机械旋转时换相一次。这表示驱动一个 BLDC 电机只需六个换相信号。此外，一个有效控制意味着相位反电动势与相位电源间的同步，这样，反电动势在非导通 60° 磁区期间过零一次。下一段显示如何获得三个反电动势和它们的过零。图 8 图示了电机端子模型，在这里，L 是相位电感，R 是相位电阻，E 是反电动势，Vn 是相对于接地的星形连接电压，而 Vx 是相对于接地的相位电压。使用 ADC 部件并通过图 8 中图示的电阻电桥来测量 Vx 电压。

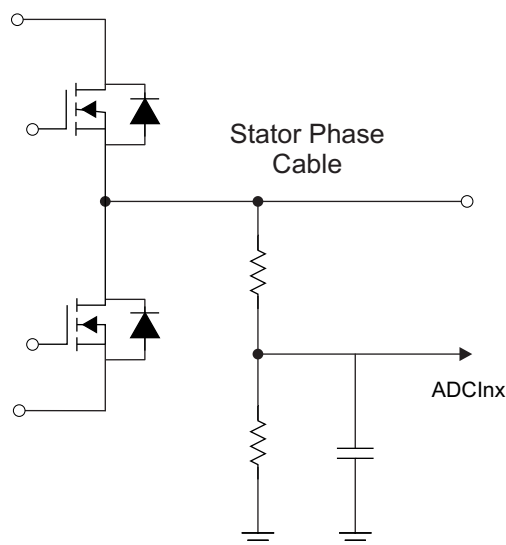


图 7. HDMC 套件上的反电动势感测

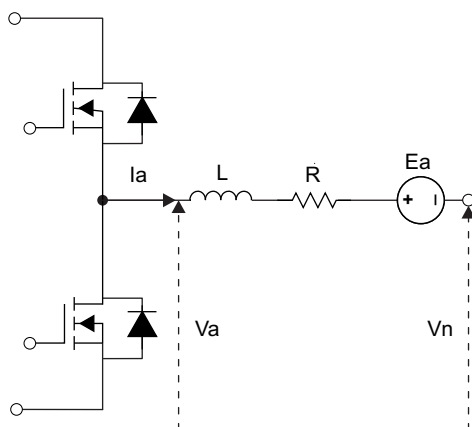


图 8. 定子端子电气模型

假定相位 C 是非导通相，可使用以下等式表达三个端子电压：

$$\begin{aligned} V_a &= R I_a + L \frac{dI_a}{dt} + E_a + V_n \\ V_b &= R I_b + L \frac{dI_b}{dt} + E_b + V_n \\ V_c &= E_c + V_n \end{aligned}$$

由于每次只有两个电流流入定子绕组，两个相位电流时相等且相反的。因此，

$$I_a = -I_b$$

因此，通过将三个端子电压等式相加，您可获得：

$$V_a + V_b + V_c = E_a + E_b + E_c + 3V_n$$

图 9 中显示了 BLDC 电机的即时反电动势波形。从这个图表中，可以很明显地看出，在反电动势过零点上，三个反电动势的和等于零。因此，最后一个等式精简为：

$$V_a + V_b + V_c = 3V_n$$

这个等式在代码中执行以计算中性点电压。在代码中， $3V_n$ 的数量由被称为 *neutral* 的变量表示。

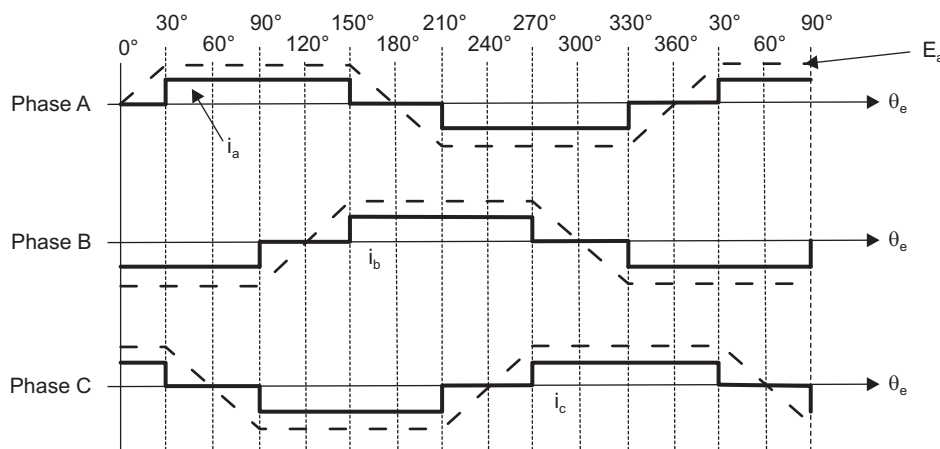


图 9. 典型电流和反电动势波形正弦波 BLDC 电机驱动

对于非导通相（零电流），定子端子电压可被重新表示如下：

$$3E_c = 3V_c - 3V_n$$

在代码中使用这个等式来计算非导通相 C 的反电动势过零点。用相类似的等式来计算其它反电动势电压 E_a 和 E_b 的反电动势过零点。由于反电动势过零点内的影响，有可能只检查反电动势符号变化；这假定反电动势扫描环路周期远远短于机械时间常量。这个函数在三个端子电压采样后计算，通常情况下每 $50\mu s$ 一次。

4.4 换相点上的电气运行方式

在相位换相的瞬间，由于直流电平或寄生电感和电源电路板的电容，有可能出现高 dV/dt 和 dI/dt 毛刺脉冲。这可导致对经计算得出的中性点电压的误读。这可以通过在新的相位换相发生时丢弃反电动势的头几个扫描来克服。在代码中，这由名为‘NOISE_WIN’函数执行。持续时间取决于电源开关、电源电路板设计、相位电感和驱动的直流电流。这个参数是系统相关的，并且被设定为电机低速范围内的一个较大的值。随着速度增加，由于在较高速时也逐渐接近反电动势过零，此软件逐渐降低这个持续时间。

4.5 换相即时计算

在有效传感控制中，反电动势过零点在相位换相瞬间移动 30° 。所以，在借助六个过零事件运行无传感器 BLDC 电机前，为了实现精确换相点，有必要计算相对于这个 30° 延迟角的时间延迟。这可通过执行一个位置内插函数来实现。在这个软件中，执行如下：让 T 为转子完成之前旋转所需的时间，而 α 为所需的延迟角。用 360° 除以 α ，并将结果乘以 T ，就获得了下一相位对换相前花费的时间长度。在代码中，这个延迟角被固定为 30° 。相应的时间延迟用采样时间周期的数量表示，并被存储在变量 `cmtn_delay` 中。因此，

$$\text{Time delay} = \text{cmtn_delay} \cdot T_s = T(\hat{\alpha} / 360) = v_timer \cdot T_s(\hat{\alpha} / 360) = v_timer \cdot T_s / 12$$

在这里， T_s 是采样时间周期，而 v_timer 是计算转子之前旋转期间采样周期数量的定时器。

上面的等式被进一步简化为：

$$\text{cmtn_delay} = v_timer / 12$$

在代码中执行这个等式来计算相对于 30° 换相延迟角的时间延迟。

5 32 位 C2000 控制器针对数字电机控制 (DMC) 的优势

C2000 系列器件具有执行复杂控制算法所需的计算能力，以及与 DMC 硬件的不同组件对接的正确混合外设，诸如模数转换器 (ADC)，增强型脉宽调制器 (ePWM)，正交编码器脉冲 (QEP)，增强型捕捉 (ECAP) 等器件。这些外设具有符合安全标准的用于执行系统所必须的全部钩子程序，像针对 PWM 和比较器的触发区。与这些外设一起，C2000 软件（库和应用软件）和硬件（应用套件）系统有助于减少开发数字电机控制解决方案所需的时间和精力。DMC 库提供可被重新使用来执行全新的控制策略的可配置块。IQMath 库可实现浮点算法到定点的轻松迁移，从而加速了开发周期。

因此，借助于 C2000 系列器件，可以轻松且快速的执行针对电机控制的复杂控制算法（传感和无传感器）。C2000 器件和高级控制系统配置的使用提供了以下系统改进：

- 通过全部速度范围内的有效控制支持系统成本减少，前提是电源器件电路的尺寸正确
- 通过使用先进的控制算法，有可能减少转矩纹波，从而导致更低的抖动和更长的电机使用寿命。
- 先进的控制算法减少由反相器生成的谐波，因此减少了滤波器成本。
- 无传感器算法的使用免除了对于速度或位置传感器的需要。
- 减少查询表的数量，这减少了所需存储器的数量。
- 平滑接近最优基准配置文件和移动轨迹的实时生成导致更佳的性能
- 使用 ePWM 外设来控制电源开关反相器有可能生成高分辨率 PWM
- 提供单芯片控制系统

对于高级控制，C2000 控制器也可执行：

- 使用诸如神经网络和模糊逻辑的现代智能方法来实现多变量和复杂系统的控制。
- 执行自适应控制。C2000 控制器具有速度功能来同时监视系统并对其进行控制。一个动态控制算法在实时中自适应至系统运行方式的变化。
- 针对无传感器控制算法、自我启动、在线参数估算更新执行参数识别。
- 执行高级转矩纹波和可闻噪声减少
- 提供具有频谱分析的诊断监控。通过观察机械抖动的频率频谱，可在早期阶段预测故障模式。
- 产生锐截止陷波滤波来消除窄带机械谐振。陷波滤波移除了电能，否则，这些电能将激发谐振模式并有可能使得系统不稳定。

6 TI 文献和数字电机控制 (DMC) 库

数字电机控制 (DMC) 库由表示为块的函数组成。这些块被分类为变换和估算器（Clarke, Park, 滑动模式观察器，相位电压计算和解算器，磁通和速度计算器以及估算器），控制（信号生成，PID, BEMF 换相，空间矢量生成）和外设驱动器（针对多个拓扑结构和技术的 PWM 提取，ADC 驱动器和电机传感器接口）。每个块具有模块化软件宏，具有源代码、使用和技术原理的单个文档。要获得宏区块的源代码和解释说明，请从www.ti.com/controlsuite中并选择 HVMotorKit 安装来安装 controlSUITE。

- C:\TI\controlSUITE\libs\app_libs\motor_control\math_blocks\v4.0
- C:\TI\controlSUITE\libs\app_libs\motor_control\drivers\f2803x_v2.0

这些模块使您能够快速构建，或者定制您的系统。此库支持三个电机类型：ACL, BLDC, PMSM，它还包括外设相关（软件驱动程序）和目标相关模块。

DMC 库组件已经由 TI 用来提供系统示例。初始化时，所有 DMC 库变量被定义和互连。运行时，宏函数被按顺序调用。每个系统使用一个递增构建方法建立，这个方法可分多次建立编码的某些部分，这样开发人员可以一次一步验证他们应用的每个部分。这在实时控制应用中十分重要，在这里，很多不同的变量可影响到系统，而且需要调整很多不同的电机参数。

注： 为了实现优化用途，TI DMC 模块被以宏的形式写入。更多细节，请参见《优化数字电机控制 (DMC) 库》(SPRAAK2)。宏在头文件内定义。您可以打开各自的头文件，并在需要时改变宏定义。在宏定义中，如**Example 1**中所示，在每一行的末尾应该有一个反斜杠“\”，这意味着编码在下一行继续。反斜杠后包括诸如“space”或“tab”等不可见字符在内的任一字符将导致编译错误。因此，请确保反斜杠是行的最后一个字符。在代码开发方面，宏几乎与 C 函数完全一样，而用户应该可以很轻松地将宏定义转换为一个 C 函数。

Example 1. 一个典型的 DMC 宏定义

```
#define PARK_MACRO(v) \
\
v.Ds = _IQmpy(v.Alpha,v.Cosine) + _IQmpy(v.Beta,v.Sine); \
v.Qs = _IQmpy(v.Beta,v.Cosine) - _IQmpy(v.Alpha,v.Sine);
```

6.1 系统概览

本文档描述了“C”实时控制框架，此框架用来演示 BLDC 电机的梯形控制。“C”框架被设计用来在基于 TMS320C2803x 的控制器上用 Code Composer Studio™ 软件运行。此框架使用以下模块：⁽¹⁾：

⁽¹⁾ 请参考电机控制文件夹内的 pdf 文档，此文档解释了每个宏的细节和理论背景。

宏名称	说明
BLDCPWM/PWMDAC	PWM 和 PWMDAC 驱动
COM_TRIG	换相触发生成器模块
PID	PID 稳压器
RC	斜坡控制器（转换率限制器）
RC2	斜升和斜降模块
RC3	斜降模块
SPEED_FR	速度测量（基于传感器信号频率）
脉冲	脉冲发生器
MOD6_CNT	Mod 6 计数器

在这个系统中，试验了 BLDC 电机的无传感器梯形控制，并且将探究速度控制器的性能。此 BLDC 电机由一个传统的电压源反相器驱动。TMS320F2803x 控制卡被用来生成三个 PWM 信号。此电机由一个集成电源模块通过使用 BLDC 专用 PWM 技术驱动。直流总线返回电流（ia 和 ib）被测量，并通过模数转换器 (ADC) 被发送给 TMS320x2803x。

HVBLDC_Sensorless 项目具有以下属性：

C 语言框架		
系统名称	程序存储器用量 2803x	数据存储器用量 2803x ⁽¹⁾
HVBLDC_Sensorless	4098 个字 ⁽²⁾	2012 个字

⁽¹⁾ 不包括堆栈尺寸

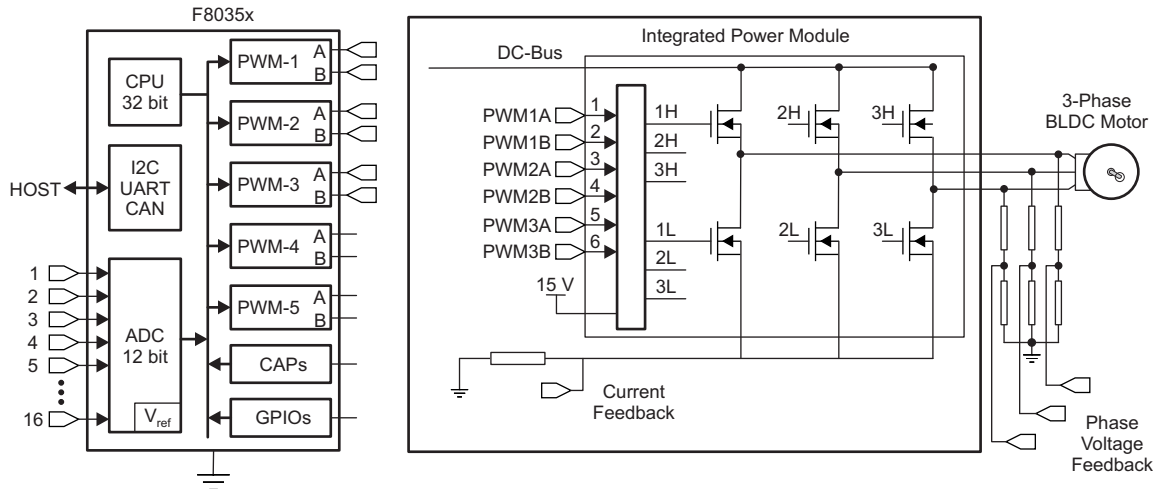
⁽²⁾ 不包括“IQmath”查询表

CPU 利用率- PMSM 无传感器	
总的周期数量	431 ⁽¹⁾
CPU 初始化 @ 60Mhz	14.4%
CPU 初始化 @ 40Mhz	21.6%

⁽¹⁾ 在 20kHz ISR 频率下。不包括调试宏。

系统特性	
开发和仿真	Code Composer Studio v4.1（或更高版本），支持实时调试
目标控制器	TMS320F2803x
PWM 频率	20kHz PWM（缺省值），60kHz PWMDAC
PWM 模式	不与无死区时间对称
中断	CPU 定时器 0 - 执行 40kHz ISR 执行速率
使用的外设	PWM 1, 2, 3 用于电机控制 PWM 6A, 6B, 7A 和 7B 用于 DAC 输出 针对低侧直流总线电流感测的 ADC A2, B1-3 用于反电动势感测

图 10和图 11中图示了执行 3 相无传感器 BLDC 控制的总体系统。



A 请注意，直流总线返回电流由 R1.1 中的三相电流相加获得。

图 10. 一个 3 相 BLDC 驱动实现

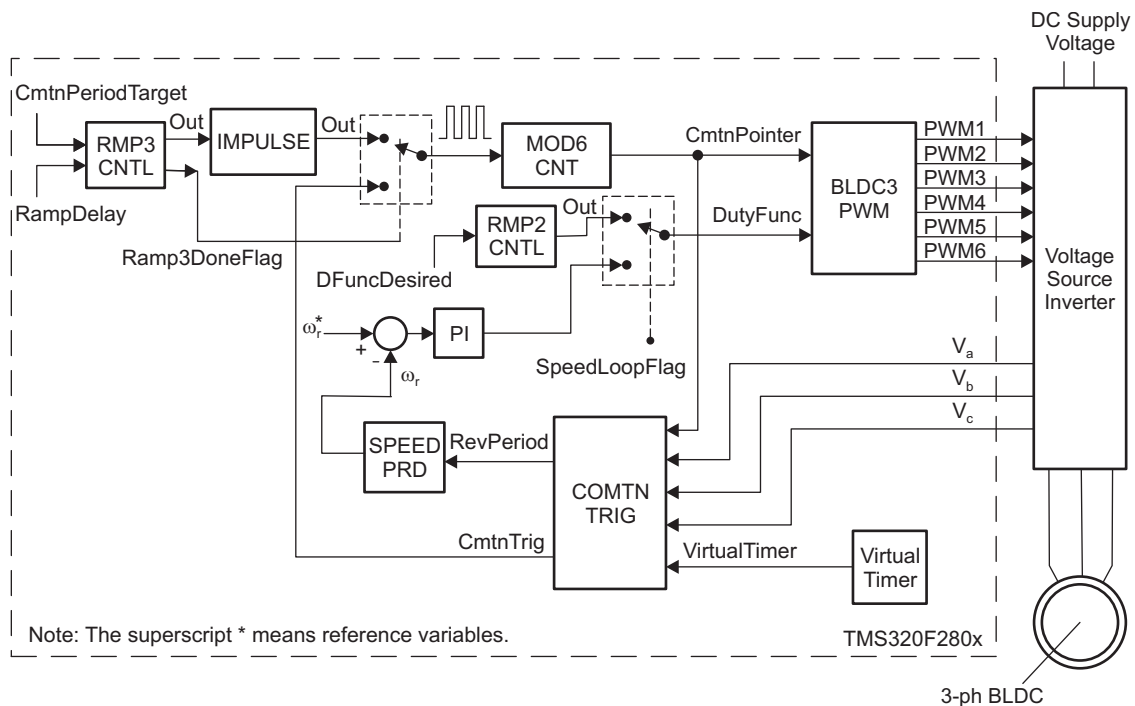


图 11. BLDC 电机霍尔传感器控制的总体方框图

图 12中描述了软件流。

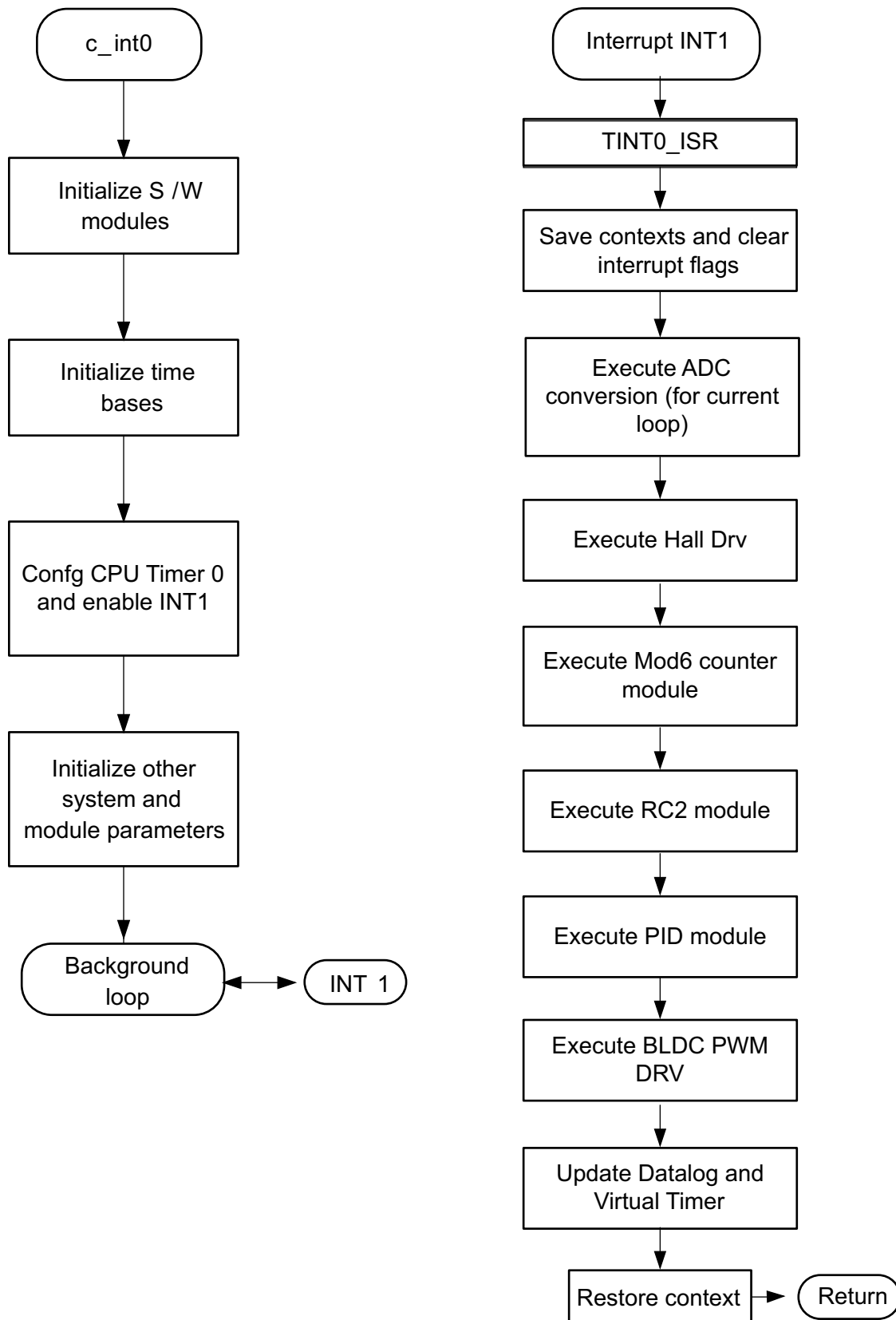


图 12. 软件流

7 硬件配置 (HVDMC R1.1 套件)

要获得套件硬件的概述和设置这个套件的步骤, 请参见《HVMotorCtrl+PFC 如何运行指南》位于: www.ti.com/controlsuite 并选择 HVMMotorKit 安装。

下面列出了某些硬件设置操作指南以便快速参考。

1. 打开 HV 套件的盖子。
2. 安装跳线 [Main]-J3, J4 和 J5, J9 至 3.3V, 5V 和 15V 电源轨以及 JTAG 复位线路。
3. 打开包装取出 DIMM 类型 controlCARD, 并将其放置在 [Main]-J1 的连机器插槽内。使用均匀的压力将卡的两端垂直向下按压, 直到卡子发生吧嗒一声并锁住。要移除卡, 用拇指打开周围的固定卡子即可。
4. 将一条 USB 线缆连接至接头 [M3]-JP1。这禁用了到 C2000 器件的隔离式 JTAG 仿真。[M3]-LD1 应该打开。请确保 [M3]-J5 未被组装。如果所包含的 Code Composer Studio 被安装, 针对板载 JTAG 仿真的驱动程序将被自动安装。如果出现了一个 windows 安装窗口, 请试着从那些已经安装在您的计算机内的驱动程序中来安装驱动程序。仿真驱动程序可在以下链接内找到<http://www.ftdichip.com/Drivers/D2XX.htm>。列出的驱动程序是支持 FT2232 的正确驱动程序。
5. 如果使用了一个第三方 JTAG 仿真器, 将 JTAG 插头连接至 [M3]-J2, 并且需要额外组装的 [M3]-J5 以将板载 JTAG 置于复位状态。
6. 请确保 [M6]-SW1 处于“Off”位置。将 15V 直流电源连接到 [M6]-JP1。
7. 打开 [M6]-SW1。现在, [M6]-LD1 应该接通。请注意, 控制卡 LED 也点亮, 这表示控制卡正在由电路板供电。
8. 请注意, 在您完成首个递增构建步骤后, 电机应该被接至 [M5]-TB3 端子。
9. 请注意, 递增构建期间, 应该只施加直流总线电源 下面讨论了获得直流总线电源的两个选项:
 - 将电源输出设定为零, 并将 [Main]-BS5 和 BS6 分别连接至直流电源和接地, 以使用直流电源。
 - 要使用交流主电源, 使用香蕉插头连接线将 [Main]-BS1 和 BS5 互连。现在, 将交流电源线的一端接至 [Main]-P1。需要将另外一端接至一个自耦变压器的输出。请确保自耦变压器的输出被设定为零, 并通过一个绝缘体接至墙上的电源。

注: 相位电压感测电容器 (C21, 22, 23) 针对交流电机控制进行了优化。使用较低值电容器 ($\leq 2.2\text{nF}$) 来达到峰值转矩。还请注意, 直流总线电流反馈由全部三个相位相加获得, 而不是通过直流总线返回路径上的分流电阻器得到。

图 13 和图 14 显示了这个实验中需要被连接的跳线和接头供您参考。请注意, 被连接到反相器输出的电机电缆颜色的顺序应该显示如下。更多信息请参见器件专用电机数据表。

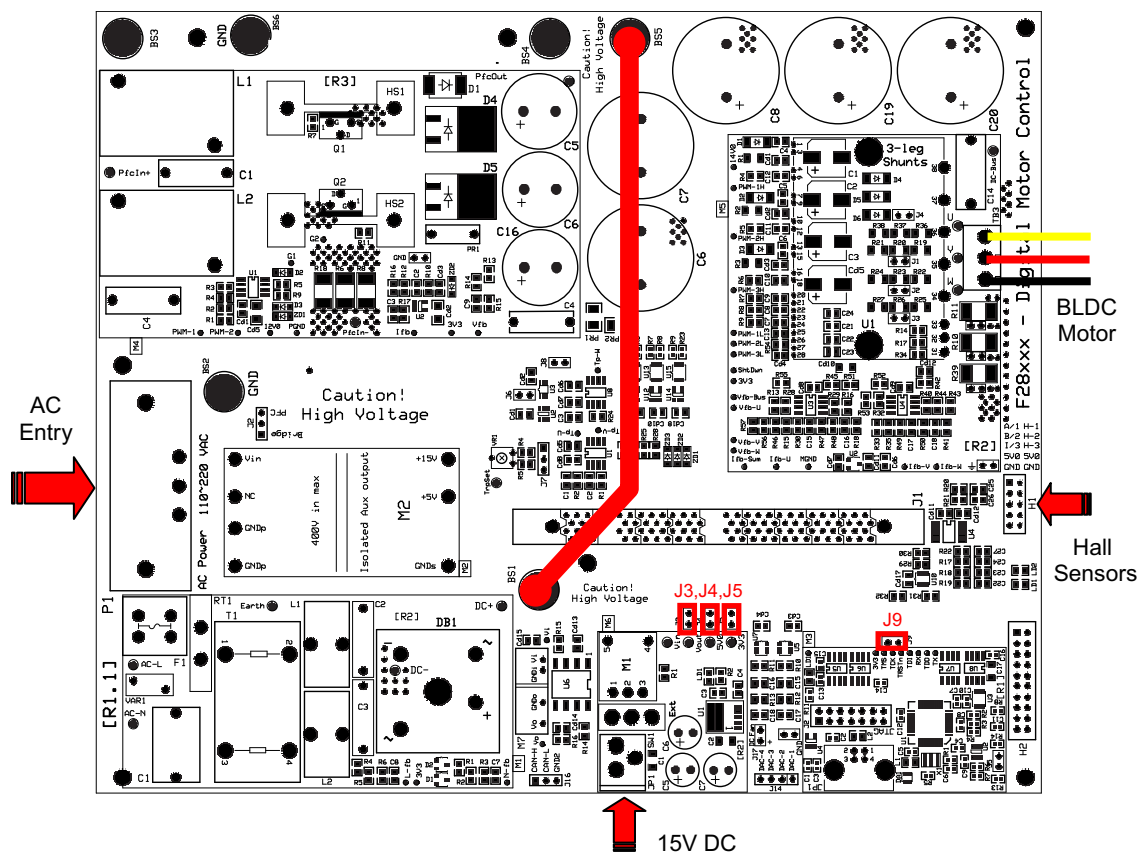


图 13. 使用交流电源生成直流总线电源

CAUTION

在高功率线路电源被关闭或断开后，反相器总线电容器仍长时间的保持充电状态。请小心进行下一步操作！

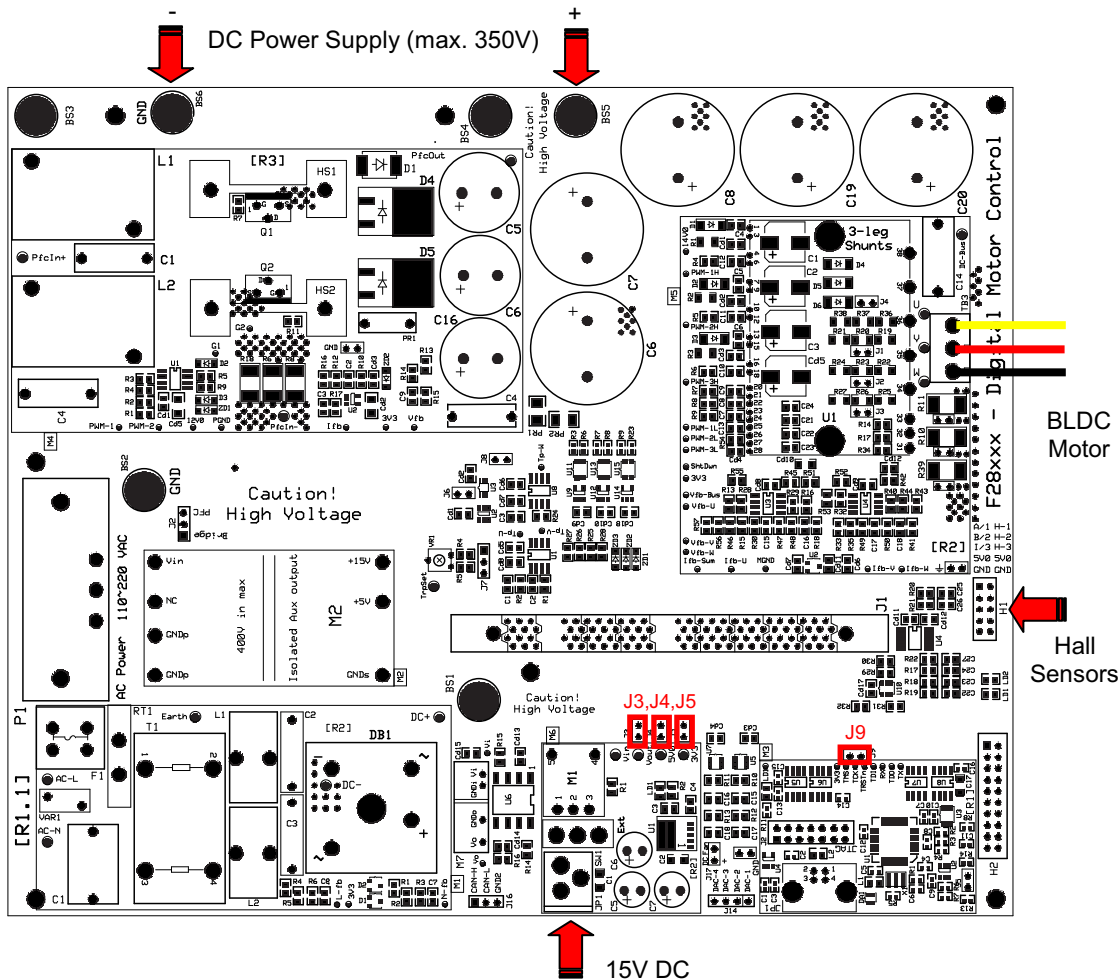


图 14. 使用外部直流电源为反相器生成直流总线

CAUTION

在高功率线路电源被关闭或断开后，反相器总线电容器仍长时间的保持充电状态。请小心进行下一步操作！

7.1 运行 HVBLDC_Sensorless 项目的软件设置指令


更多信息，请参见《HVMotorCtrl+PFC 套件运行指南》中的《针对 HVMotorCtrl+PFC 套件项目的软件设置》部分，此部分可在www.ti.com/controlsuite网页上的 HVMotorKit 安装内找到。

在这个部分中讨论如何安装 Code Composer Studio 以及如何将其设置为运行这个项目。

1. 将 HVBLDC_Sensorless 选中为激活项目。
2. 验证此构建级被设定为 1，然后右键单击项目名称并选择“Rebuild Project”（重建项目）。一旦构建完成，启动一个调试会话来将编码载入到控制器中。
3. 打开一个观察窗口，按照表 2 中显示的那样添加关键变量，并为他们选择合适的 Q 格式。

表 2. 观察窗口变量

变量名称	被视为
EnableFlag	无符号整数
IsrTicker	无符号整数
SpeedRef	Q24
Dlog.prescalar	整数
SpeedLoopFlag	无符号整数
ILoopFlag	无符号整数
CmntnPeriodTarget	无符号整数
DFuncDesired	无符号整数
ClosedFlag	无符号整数
pi_spd.Out	Q24
pi_idc.Out	Q24
Pi_spd.Kp	Q24
speed1.Speed	Q24

- 通过从以下位置导入 Graph1.graphProp 和 Graph2.graphProp 来建立时间曲线图窗
□: www.ti.com/controlsuite -
(development_kits\HVMotorCtrl+PfcKit_v2.0\HVBLDC_sensorless\)M
- 单击  graph tab (曲线图标签页) 左上角上的 Continuous Refresh (持续刷新) 按钮来启用微控制器数据的定期捕捉。

8 递增系统构建用于无传感器 BLDC 项目

此系统被逐步构建，所以最终系统可以安心运转。递增系统构建的 6 个阶段被设计成验证系统中使用的主要软件模块。表 3 总结了每个递增系统构建中的模块测试和使用。

表 3. 每个递增系统构建中的已测试模块⁽¹⁾

软件模块	阶段 1	阶段 2	阶段 3	阶段 4	阶段 5
PWMDAC_MACRO		√√	√	√	√
RC3_MACRO	√√	√√	√	√	√
MOD6_CNT_MACRO	√√	√√	√	√	√
IMPULSE_MACRO	√√	√√	√	√	√
BLDCPWM_MACRO	√√	√√	√	√	√
RC2_MACRO			√√	√	√
HALL3_READ_MACRO			√√	√	√
SPEED_FR_MACRO			√√	√	√
PI_MACRO (IDC)				√√	√
RC_MACRO					√√
PI_MACRO (SPD)					√√

⁽¹⁾ 符号√表示这个模块正在被使用，而符号√√表示这个模块正在这个阶段中进行测试。

8.1 1 级递增构建

假定《HVMotorCtrl+PFC 套件运行指南》中描述的负载和构建步骤已经成功完成，这个部分描述了针对“最小”系统检查退出步骤，在这些步骤中，要确认系统中中断、外设和目标独立模块的运行以及一个外设相关模块的运行。

- 打开 HVBLDC_Sensorless-Settings.h 并通过将 BUILDLEVEL 设定为 LEVEL1 (# define BUILDLEVEL LEVEL1 (定义构建级 1 级)) 来选择 1 级递增构建选项。
- 右键单击项目名称并单击 Rebuild Project。
- 一旦构建完成，单击 debug (调试) 按钮，复位 CPU，重新启动，启用实时模式并运行。

4. 在观察窗口中将“EnableFlag”设定为 1。名为“IsrTicker”的变量将持续增加。
5. 通过观察观察窗口中的变量来确认这一点。这确认系统中断运转正常。

在软件中，被调整的关键变量汇总如下：

- RampDelay（格式 Q0）：用于改变斜坡时间
- CmtnPeriodTarget（格式 Q0）：用于改变目标换相间隔

下面给出了关键说明和步骤：

- BLDC 电机的启动和初始加速由 RMP3CNTL 模块控制。这个模块生成一个斜坡功能。RMP3CNTL 模块的这个斜坡特性可在开环配置中使 BLDC 电机从停止开始加速（类似于一个步进电机）。
- RMP3CNTL 模块的输入中的一个，DesiredInput 在电机加速阶段的末尾确定最终速度。这个输入由系统使用系统变量 CmtnPeriodTarget 提供。您可以根据 BLDC 电机的类型，使用合适的值来初始化这个系统变量。RMP3CNTL 模块的第二个输入是 rmp3_dly，也可由用户使用系统变量 RampDelay 初始化。这确定了电机加速的速度。RMP3CNTL 模块的输出为 Out，它提供了一个逐渐减少的可变时间周期。RMP3CNTL 模块的第二个输出是 Ramp3DoneFlag，当此变量被设定为 0x7FFF 时，此输出表示斜坡（或电机加速）阶段的末尾。
- Out 被用来为 IMPULSE 模块提供输入周期。这个模块生成周期脉冲，此周期的脉冲由它的输入周期指定。
- DATALOG 模块被用来观察模块的输出变量。在 1 级递增构建初始化例程中完成这一操作所需的初始化。在初始化期间，DATALOG 输入中的一个被配置为指向 mod1.Counter。因此，Out 信号被显示在 Code Composer Studio 的曲线图中。
- 周期脉冲输出，Out，被施加到 MOD6_CNT 模块的输入 TrigInput 上。这个模块的输出是 Counter，可采用六个可能值 0，1，2，3，4 或 5 中的一个。当一个触发脉冲被施加到这个输入上时，这个输出从一个状态改变为下一个状态。最后，这个计数器被用作针对模块 BLDC_3PWM_DRV 的指针输入，CmtnPointer。指针变量，CmtnPointer 的这六个值，被用来生成驱动 BLDC 电机的功率反相器的六个换相状态。所生成的 PWM 输出的占空比（根据六个换相状态）在电机加速阶段，由输入 DfuncTesting 确定。
- 实时编译、载入和运行此程序。
- 在观察窗口中将“EnableFlag”设定为 1。当 RMP3CNTL 最初斜坡时，周期（Out 的周期）也将逐渐下降。在斜坡周期的末尾（当 Out 等于 DesiredInput 时），周期将变得恒定，而 Ramp3DoneFlag 将设定为 0x7FFF。为 CmtnPeriodTarget 输入一个新的更低的值 (DesiredInput)。然后周期将逐渐减少至新的值。
- 在观察窗口和曲线图窗口中检查 MOD6_CNT 输出变量计数器。这个值在 0 到 5 之间。
- 使用一个观察仪器来检查由外设相关模块 - BLDC_3PWM_DRV 控制的 PWM 输出。奇数 PWM 输出（PWM1，PWM3 和 PWM5）将生成 PWM 脉冲或保持关闭。偶数 PWM 输出（PWM2，PWM4 和 PWM6）将保持打开或关闭。
- 全部六个 PWM 输出的输出状态将是如此，这样，它们在一起生成驱动 BLDC 电机的功率反相器的六个换相状态。
- 在验证这个状态后，使控制器离开实时模式（禁用），复位处理器，然后终止调试会话。

当运行这个构建时，PWM 输出应该如图 15 中显示的那样。

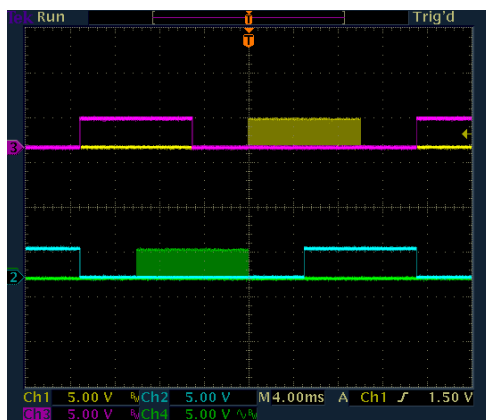


图 15. PWM 输出: PWM 1 (黄色), PWM 2 (粉色) 和 PWM 5 (绿色), PWM 6 (蓝色)

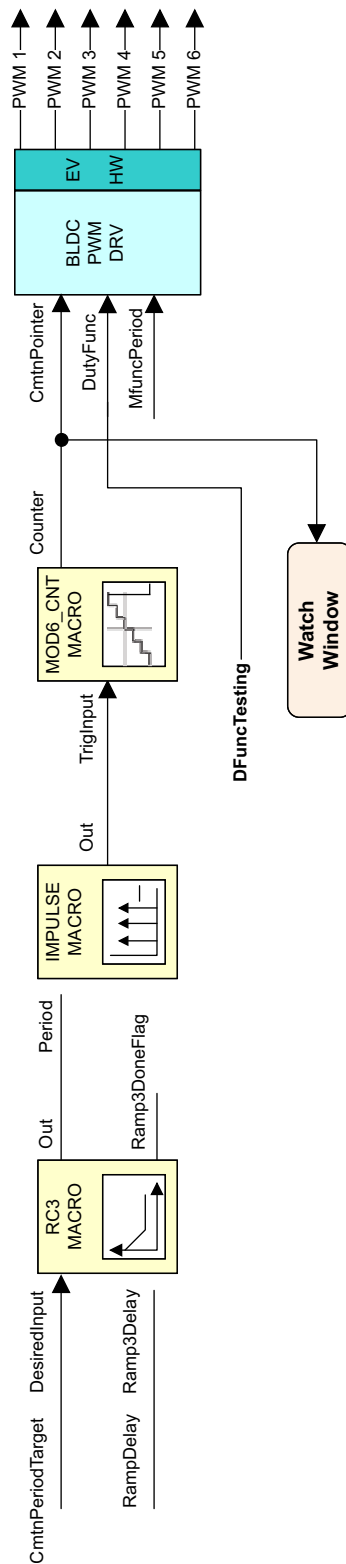


图 16.1 级 - 递增系统构建方框图

1 级描述了针对“最小”系统检查退出步骤，在这些步骤中，要确认系统中断、某些外设和目标独立模块的运行以及一个外设相关模块的运行。

8.2 2 级 - 递增构建

假定构建 1 已成功完成，这个部分验证开关电机运行和电流测量。

1. 打开 HVBLDC_Sensorless-Setting.h 并通过将 BUILDLEVEL 设定为 LEVEL2 (#define BUILDLEVEL LEVEL2 (定义构建级 2 级)) 来选择 1 级递增构建选项，并保存此文件。
2. 右键单击项目名称，单击 Rebuild Project。
3. 一旦此构建完成，单击 debug 按钮，复位 CPU，重新启动，启用实时模式并运行。
4. 在观察窗口中将“EnableFlag”设定为 1。名为“IsrTicker”的变量在观察窗口中递增，以确认中断正常运转。

在软件中，将被调整的关键变量汇总如下：

- RampDelay (格式 Q0)：用来改变斜坡时间
- CmtnPeriodTarget (格式 Q0)：用来改变目标换相间隔

8.3 2A 级 - 开环测试


关键步骤可解释如下：

- 实时模式编译、载入和运行此程序，并增加自耦变压器和直流电源上的电压来获得适当的直流总线电压。现在电机以缺省 DFuncTesting 值运行。
- 如果适当选择了开环换相参数，那么电机将逐渐加速，并且最终在开环换相模式以恒定速度运行。
- 电机的最终速度取决于参数 CmtnPeriodTarget。针对这个变量的值越低，电机的最终速度越高。由于电机反电动势取决于它的速度，为 CmtnPeriodTarget 选择的值也将确定生成的反电动势。
- 启动期间施加到电机上的平均电压取决于参数 DfuncTesting。参数 DfuncTesting 和 CmtnPeriodTarget 应该使生成的反电动势在电机加速阶段的末尾低于施加到电机绕组上的平均电压。这样防止了电机停止或抖动。初始化部分中的缺省 DfuncTesting 和 CmtnPeriodTarget 值在 HVDMC 套件中被选择用于的电机。当测试一个不同的电机时，需要调整这些值以防止可能的抖动，并使电机完全启动。应该观察窗口中相应地调整 DfuncTesting 和 CmtnPeriodTarget 以增加电机速度。电机加速时间取决于 RampDelay，主采样环路的时间周期以及 CmtnPeriodTarget 和 CmtnPeriodSetpt 间的差异。

注：这个步骤并不针对宽速度和转矩范围运行；而是在无负载下以特定的速度关闭环路前测试且校准整个系统。

通过减少总线电压，如下面所描述的那样使系统安全停止，使控制器脱离实时模式并复位。

CAUTION

验证了这些后，减少直流总线电压，使控制器脱离实时模式（禁用），并且将处理器复位 （详细情况，请参见《HVMotorCtrl+PFC 运行指南》）。请注意，每次测试后，出于安全考虑，需要重复此步骤。还请注意，不适当的关断有可能将 PWM 暂停在某些特定状态，在这些状态下，可汲取高电流，因此，在进行这些试验时要小心。

8.4 2B 级 - ADC 验证和偏移校正

- 通过监控直流总线返回电流和全部三个反电动势（可选）来验证 ADC 运行。
- 关闭电压，并且实时编译、载入和运行程序。当直流总线电压为零时，观察窗口中显示的电流 (DCbus_current) 应该为零。如果不为零，通过以下方式调整代码中的偏移值：

```
DCbus_current = _IQ12toIQ(AdcResult.ADCRESULT4) - _IQ(0.5);
```

并根据偏移的符号和数量来改变 IQ15(0.50) 偏移值 (IQ15(0.5087) 或 IQ15(0.4988))。一旦完成这个步骤，打开电源并将输出设定为零。

当运行 2 级时，BLDC 霍尔效应传感器的输出和 PWM DAC 输出应该如图 17 和图 18 中显示的那样。

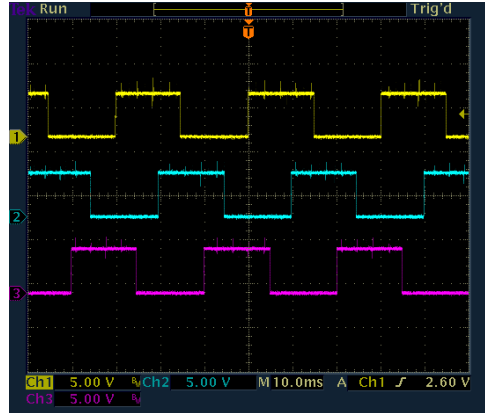


图 17. 霍尔效应传感器的输出，霍尔 A，B 和 C

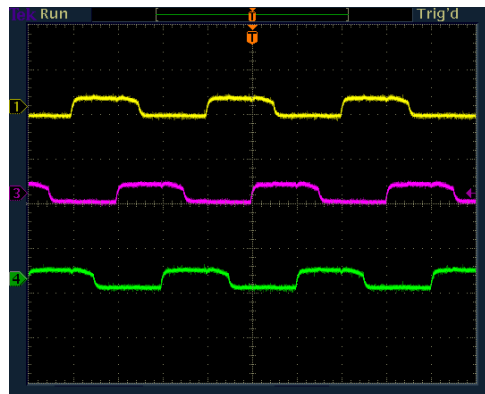


图 18. PWM DAC 输出 BemfA，BemfB 和 BemfC (Vdcbus = 160V)

当运行这个构建时，Code Composer Studio 曲线图内的电流波形应该如图 19 和图 20 中显示的那样。

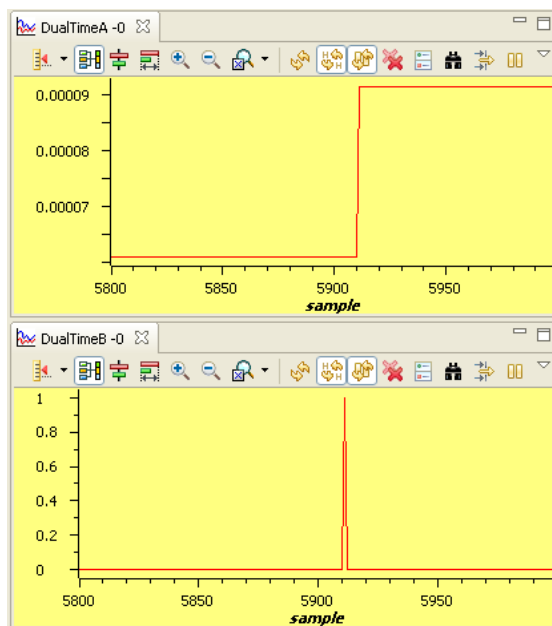


图 19. (a) mod6 计数器 (b) 脉冲输出，dlog.prescalar = 3

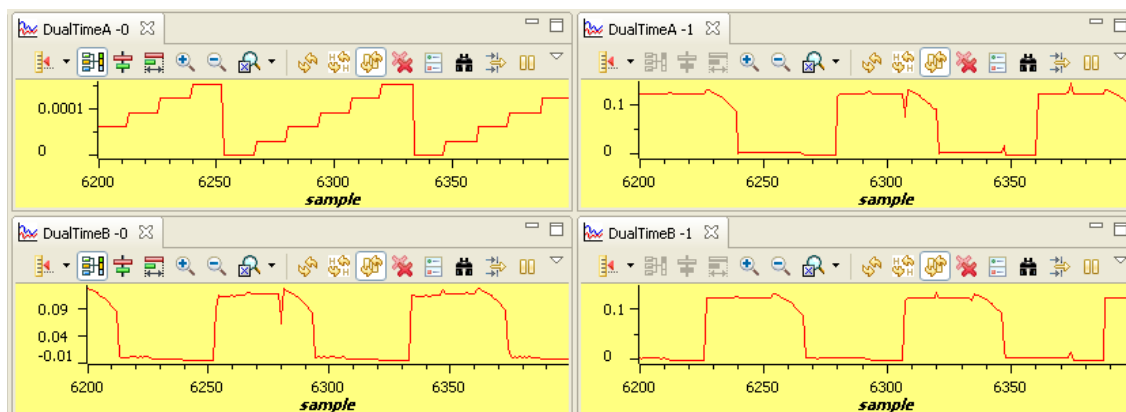


图 20. (a) mod6 计数器, (b) BmfA, (c) BmfB 和 (d) BmfC
(dlog.prescalar = 25 而 Vdcbus = 160V)

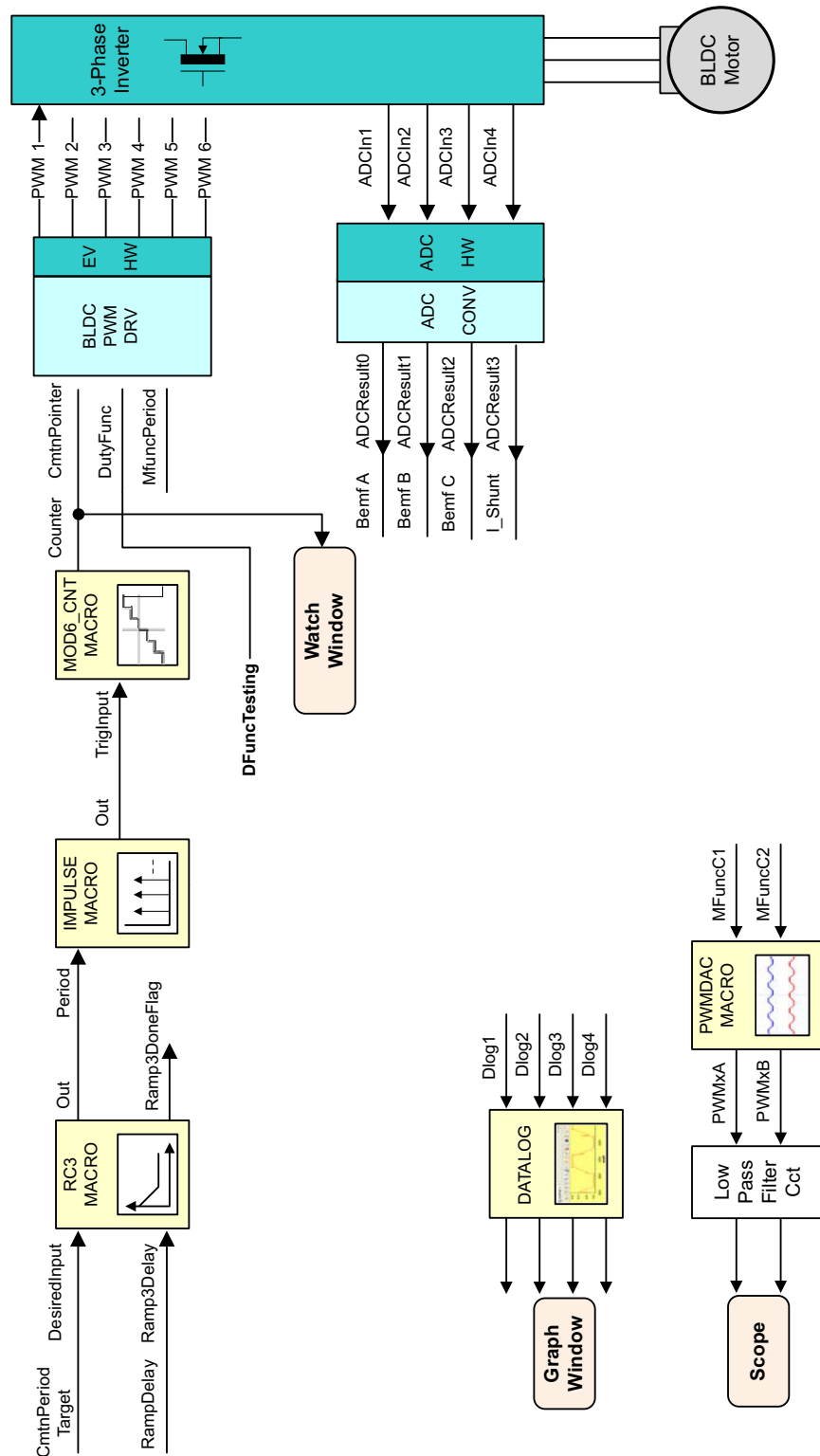


图 21.2 级 - 递增系统构建方框图

2 级验证开环电机运行和电流测量

8.5 3 级递增构建

假定之前的部分已成功完成，整个部分验证外设独立模块 COMTN_TRIG。

1. 打开 HVBLDC_Sensored-Settings.h 并通过将 BUILDLEVEL 设定为 LEVEL3 (# define BUILDLEVEL LEVEL3 (定义构建级 3 级)) 来选择 3 级递增构建选项。
2. 右键单击项目名称并单击 Rebuild Project。
3. 一旦构建完成，单击 debug (调试) 按钮，复位 CPU，重新启动，启用实时模式并运行。
4. 在观察窗口中将“EnableFlag”设定为 1。名为“IsrTicker”的变量在观察窗口中增加，以确认中断正常运转。

在软件中，将被调整的关键变量汇总如下：

- DFuncDesired (格式 Q15)：用来改变每个部件中的 PWM 占空比函数
- CmtnPeriodTarget (格式 Q0)：用来改变目标换相间隔

关键步骤解释如下：

- 实时编译、载入和运行此程序。
- 增加自耦变压器和直流电源上的电压来获得适当的 直流总线电压。现在电机以缺省 DFuncDesired 值运行。
- 从曲线图窗口或观察仪器中观察 COMTN_TRIG 模块输出变量 CmtnTrig, Neutral, ZcTrig 和 DebugBemf。
- 验证 *Neutral* 已经被正确重建并且 *DebugBemf* 显示六个“未通电”反电动势斜坡。过零触发 (ZcTrig) 应该与反电动势斜坡的零点正确对齐。
- 换相触发 (CmtnTrig) 应该出现在 ZcTrig 之后 30°，并且与换相点对齐，（反电动势斜坡的方向变化）。
- 通过减少总线电压，（如下面所描述的那样）使系统安全停止，使控制器脱离实时模式并复位。

当运行这个构建时，Code Composer Studio 曲线图内的电流波形应该如图 22 中显示的那样。



图 22. (a) Neutral, (b) DebugBemf, (c) BemfA 和 (d) BemfB (VdcBus = 160V)

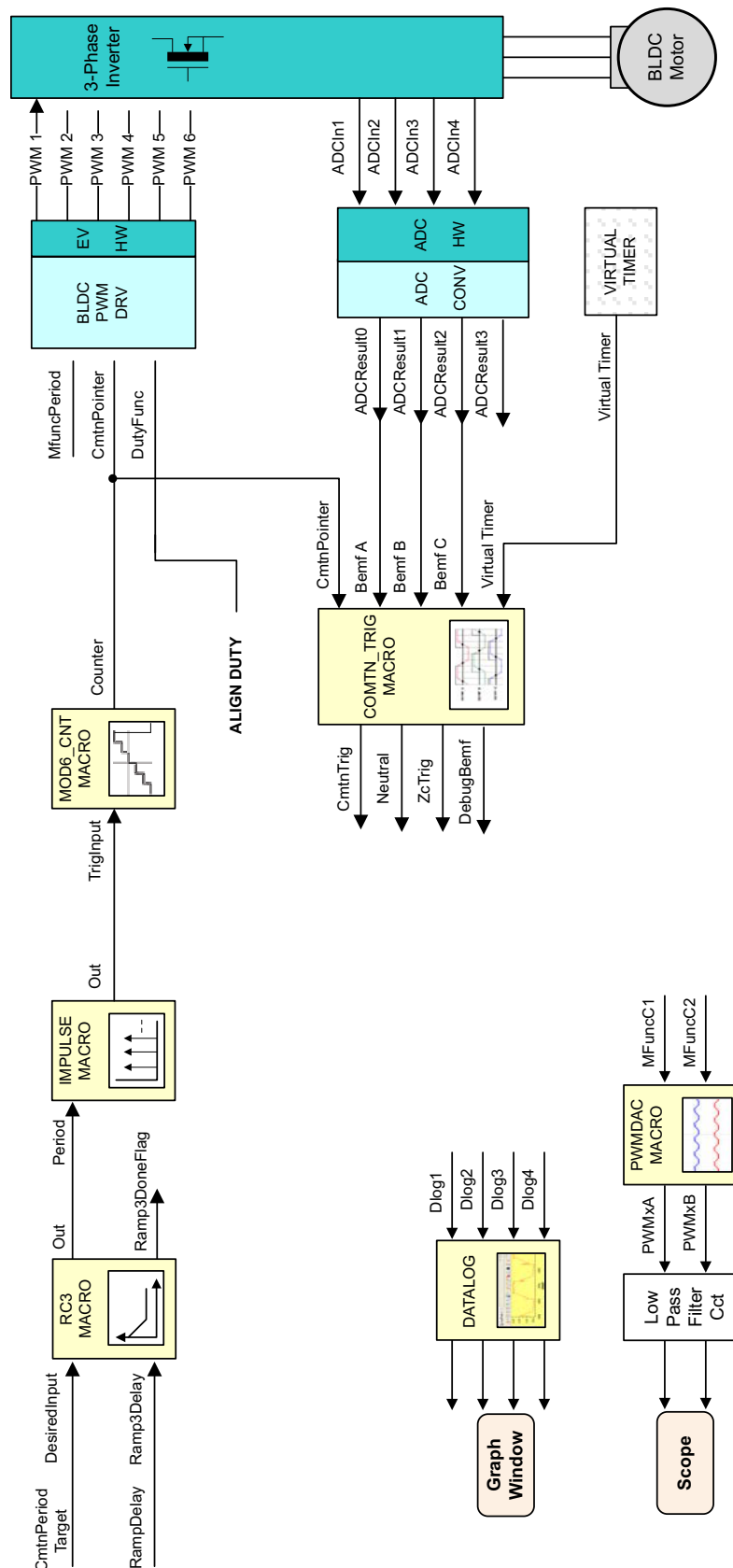


图 23.3 级 - 递增系统构建方框图

3 级验证外设独立模块 COMTN_TRIG。

8.6 4 级递增构建

假定之前部分已成功完成，这个部分验证闭环电机运行，此运行基于经计算得出的反电动势过零和产生的换相触发点。

1. 打开 HVBLDC_Sensorless-Settings.h 并通过将 BUILDLEVEL 设定为 LEVEL4 (# define BUILDLEVEL LEVEL4 (定义构建级 4 级)) 来选择 4 级递增构建选项。
2. 右键单击项目名称，单击 Rebuild Project。
3. 一旦此构建完成，单击 debug 按钮，复位 CPU，重新启动，启用实时模式并运行。
4. 在观察窗口中将“EnableFlag”设定为 1。名为“IsrTicker”的变量在观察窗口中增加，以确认中断正常运转。

在软件中，将被调整的关键变量汇总如下：

- RampDelay (格式 Q0)：用来改变斜升时间
- CmtnPeriodTarget (格式 Q0)：用来改变目标换相间隔
- DFuncDesired (格式 Q15)：用来改变每个部件内的 PWM 占空比函数

关键步骤可解释如下：

- 实时编译、载入和运行此程序，并增加自耦变压器和直流电源上的电压来获得适当的直流总线电压。
- 电机将逐渐加速，并最终切换至闭环换相模式。
- 当 Ramp3DoneFlag 被设定为 0x7FFFFFFF 时（表示电机加速阶段的末尾），开环换相切换至闭环换相。在这个切换发生前，MOD6_CNT 模块由 IMPULSE 模块的输出触发。切换后，MOD6_CNT 模块由 COMTN_TRIG 模块的输出触发。
- 当加速阶段结束时，通过改变 DFuncDesired 来改变电机速度。这个参数被用作到第二斜坡控制模块 - RMP2CNTL 的输入。这个模块的输出为 Out，此输出控制占空比参数 DutyFunc。这改变了发送给电机的功率和速度。到 RMP2CNTL 模块的第二个输入时 Ramp2Delay。这控制了斜坡值变化的速率。这个变量由您在 HVBLDC_Sensorless.c 文件中初始化。
- 通过减少总线电压，使系统安全停止，使控制器脱离实时模式并复位。

当运行这个级时，Code Composer Studio 曲线图内的电流波形应该如图 24 中显示的那样。

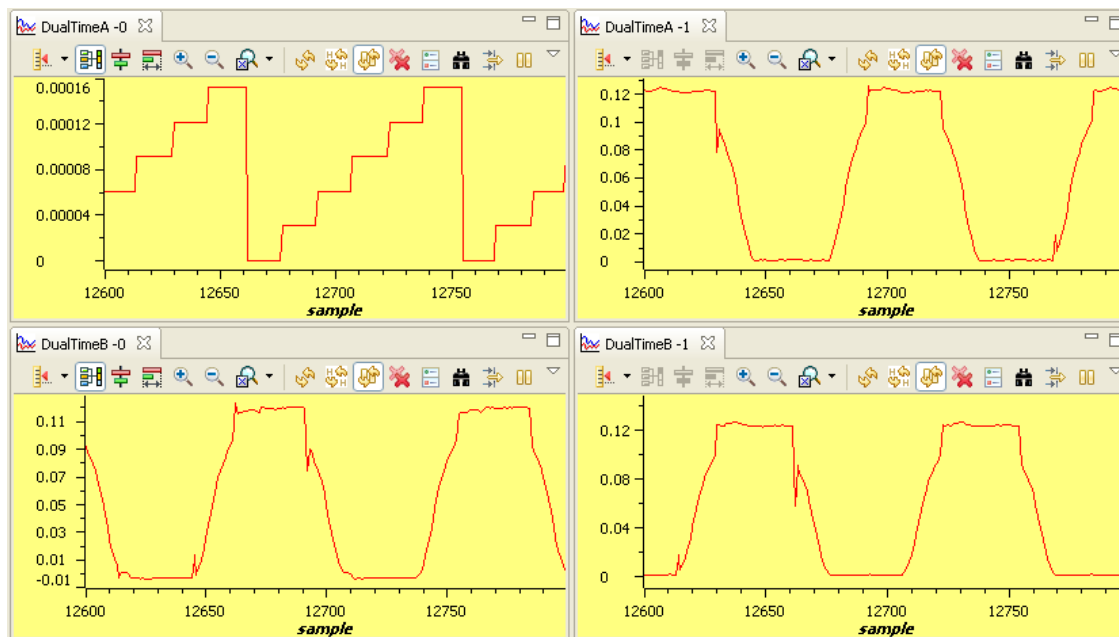


图 24. (a) mod6 计数器, (b) BemfA, (c) BemfB 和 (d) BemfC (dlog.prescalar = 25 而 Vdcbus = 160V)

PWMDAC 输出应该为图 25 中显示的那样。

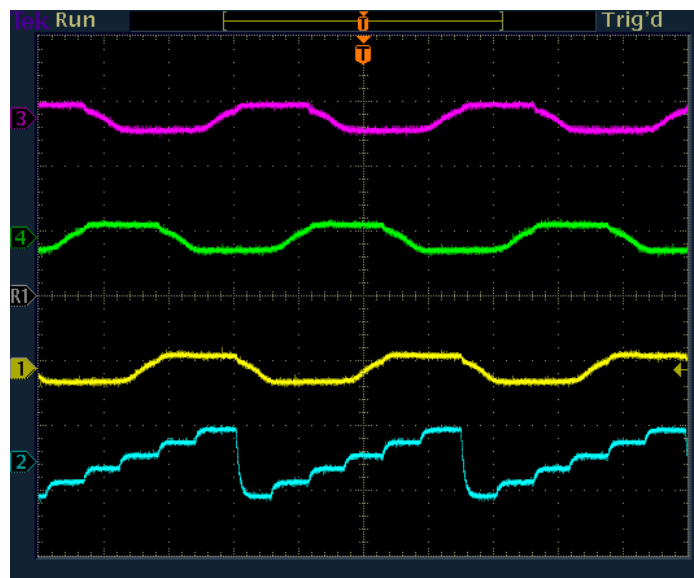


图 25. PWMDAC 输出 a) BemfA, (b) BemfB, (c) BemfC 和 (d) mod6 计数器 (Vdcbus = 160V)

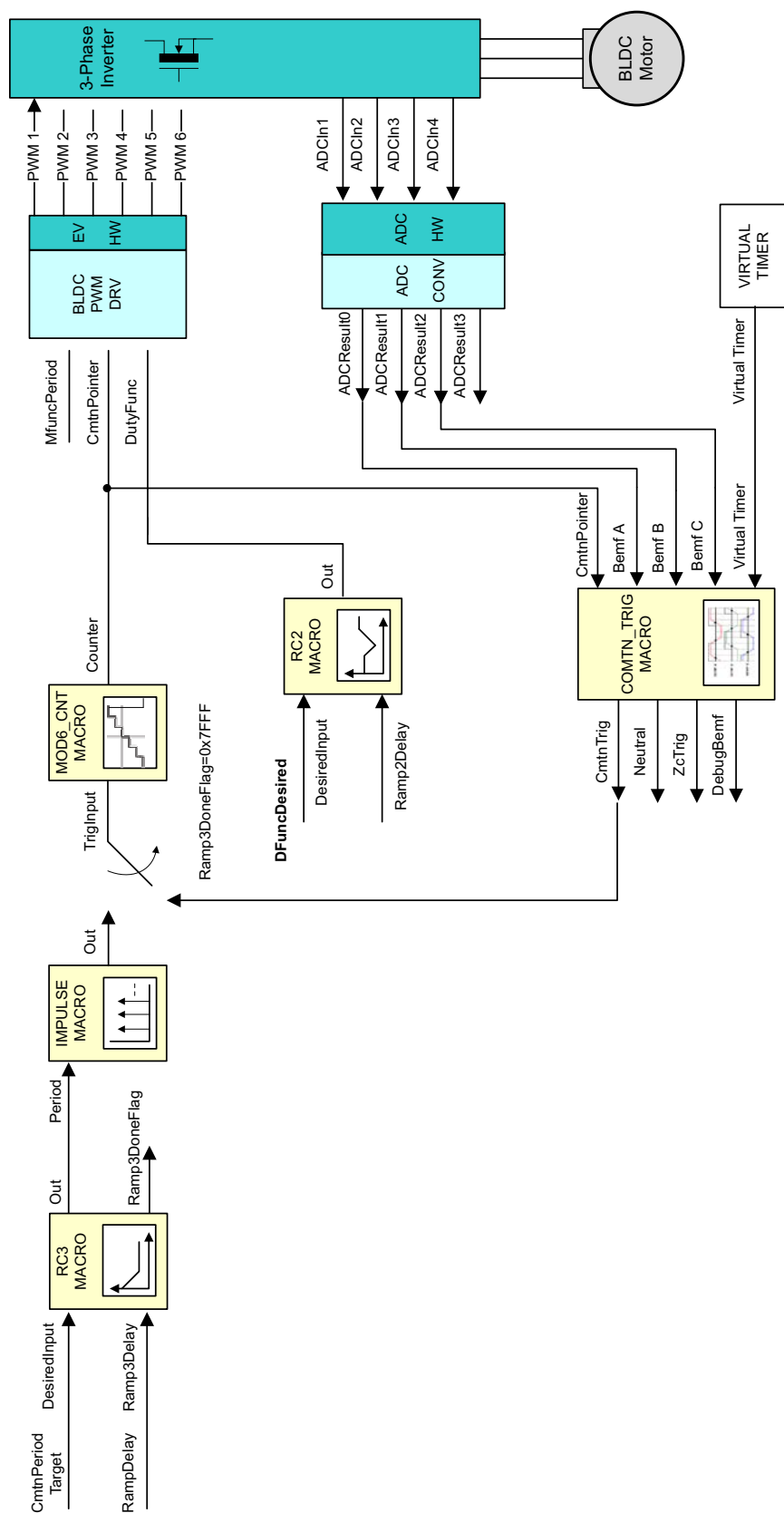


图 26.4 级 - 递增系统构建方框图

4 级验证闭环电机运行，此运行基于计算得出的反电动势过零以及产生的换相触发点。

8.7 5 级 - 递增构建

假定之前的部分已成功完成，这个部分验证闭合电流环路速度 PI 控制器。

1. 打开 HVBLDC_Sensorless-Settings.h 并通过将 BUILDLEVEL 设定为 LEVEL5（# define BUILDLEVEL LEVEL5（定义构建级 5 级））来选择 5 级递增构建选项。
2. 右键单击项目名称并单击 Rebuild Project。
3. 一旦构建完成，单击 debug（调试）按钮，复位 CPU，重新启动，启用实时模式并运行。
4. 在观察窗口中将“EnableFlag”设定为 1。名为“lsrTicker”的变量将持续增加。
5. 通过观察观察窗口中的变量来确认这一点。这确认系统中断运转正常。

在软件中，被调整的关键变量汇总如下：

- DFuncDesired（格式 Q15）：用来改变每个部件中的 PWM 占空比
- CurrentSet（格式 GLOBAL_Q）：用来改变每个部件中的基准直流总线电流
- ILoopFlag（格式 Q0）：用来在固定占空比和受控 Idc 占空比之间切换

关键步骤可解释如下：

- 实时编译、载入和运行程序。
- 增加自耦变压器和直流电源上的电压来获得适当的直流总线电压。
- 电机将逐渐加速并最终切换至闭合环路换相模式。
- 通过改变 DFuncDesired 来增加或电机速度。
- 使用变量 CurrentSet 来为 PI 控制器 - PID_REG3 指定基准电流。一旦 ClosedFlag 在代码中被设定为 1，将 ILoopflag 更改为 1 来激活电流环路 PI 控制器。一旦完成这一操作，PI 控制器开始调节直流总线电流和电机电流。
- 逐渐增加或减少指令电流（CurrentSet 值）来改变转矩指令，并调整 PI 增益。请注意，这个步骤中速度不受控制，一个非零转矩基准持续增加电机速度。因此，在关闭环路后，应该使用制动器、发电机等来加载电机（如果电机足够小的话也可手工加载）。在最开始时，施加一个相对轻的负载，然后逐渐增加负载量。如果施加的负载高于转矩基准，电机不能处理负载，并且在关闭电流环路后立即停止。
- 验证由 SPEED_PR 计算得出的电机速度（pu 和 rpm）。
- 通过减少总线电压，按照构建 1 末尾所描述的那样使系统安全停止，使控制器脱离实时模式并复位。

当运行这个级时，Code Composer Studio 曲线图内的电流波形应该如图 27 中显示的那样。

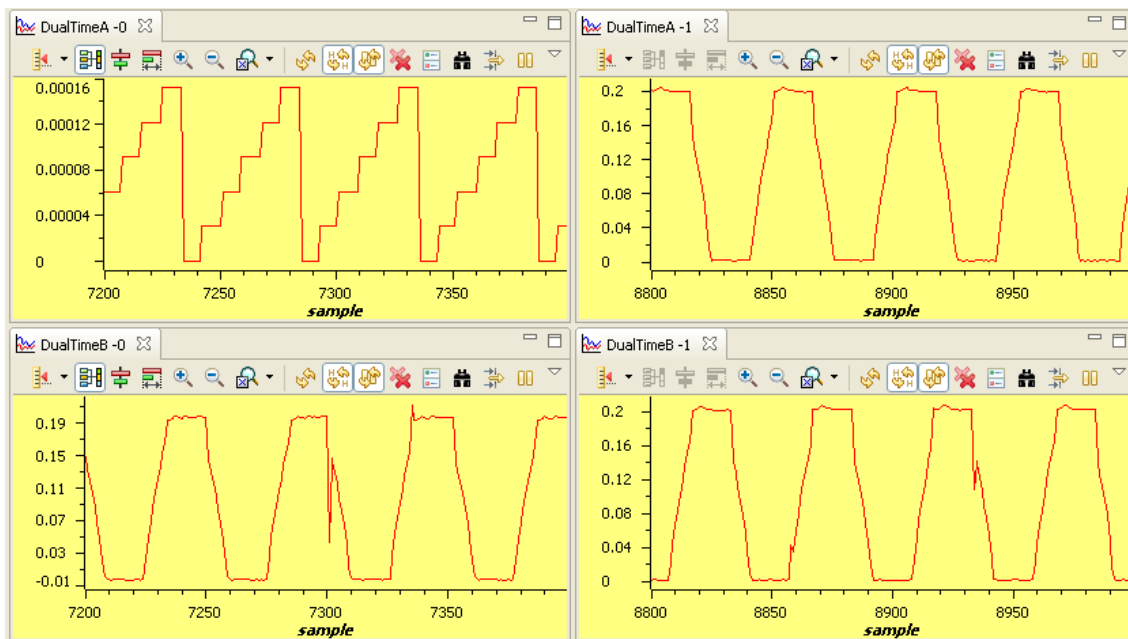


图 27. (a) mod6 Counter, (b)BemfA, (c) BemfB (c)BemfC (dlog.presalar = 25 而 Vdcbus = 160V)

PWMDAC 输出应该如图 28 中显示的那样。

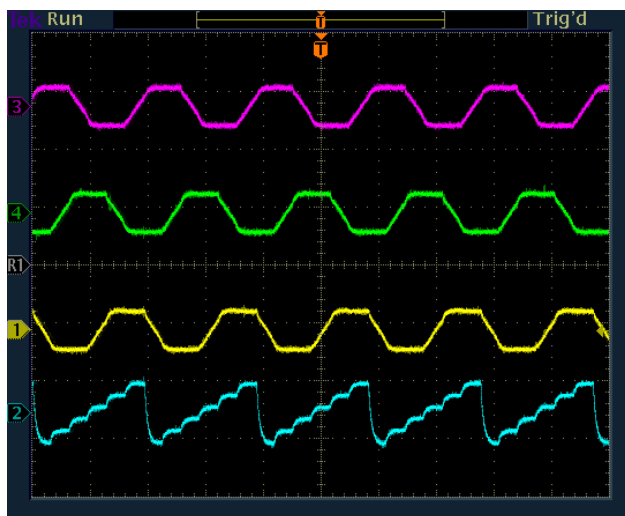


图 28. PWMDAC 输出 a) BemfA, (b) BemfB, (c)BemfC 和 (d) mod6 计数器 (Vdcbus = 160V)

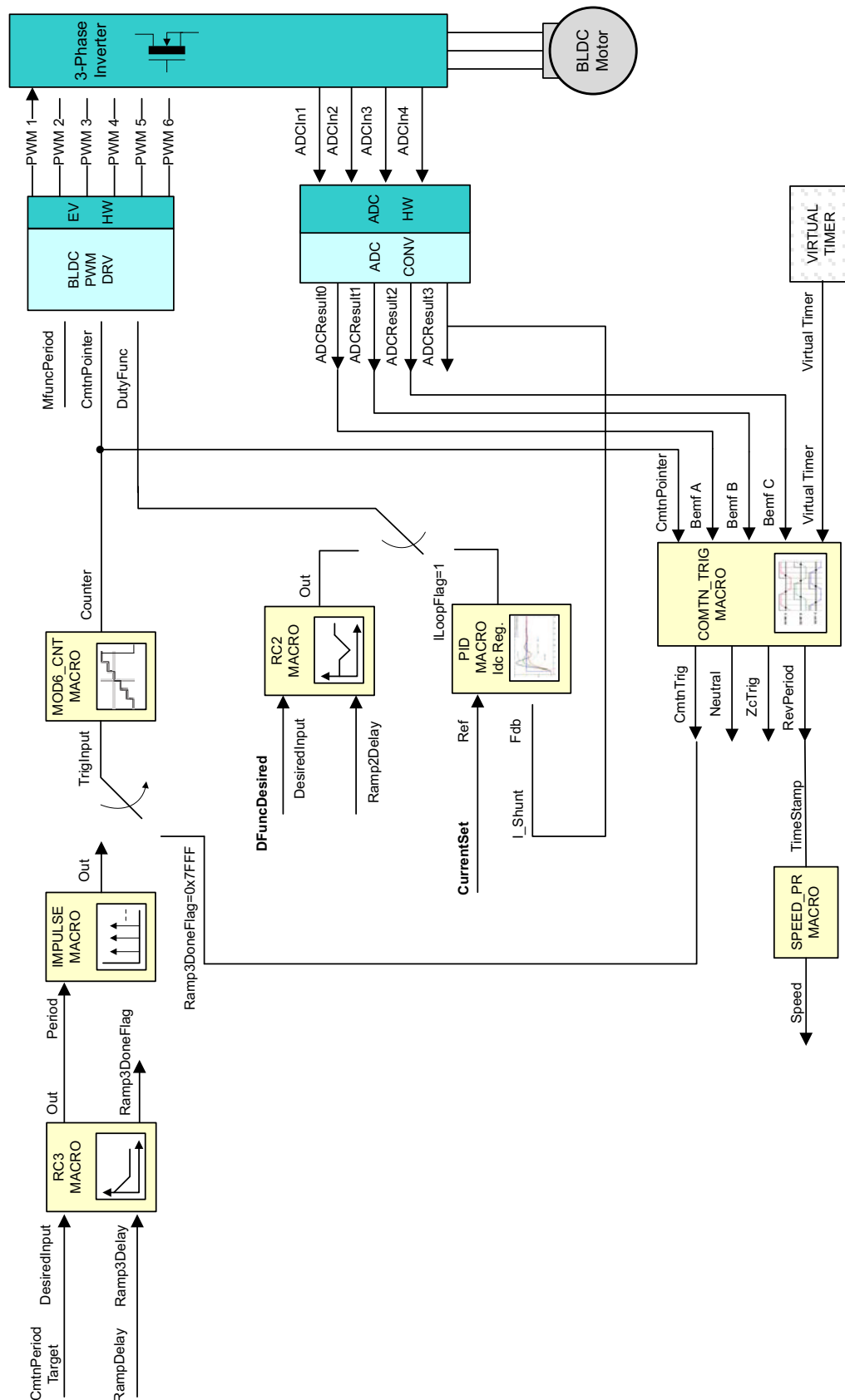


图 29.5 级 - 递增系统构建方框图

5 级验证闭合速度环路和电流 PI 控制器。

8.8 6 级递增构建

假定之前的部分已成功完成，这个部分验证闭合环路和速度 PI 控制器。

1. 打开 HVBLDC_Sensorless-Settings.h 并通过将 BUILDLEVEL 设定为 LEVEL6 (# define BUILDLEVEL LEVEL6 (定义构建级 6 级)) 来选择 6 级递增构建选项。
2. 右键单击项目名称并单击 Rebuild Project。
3. 一旦构建完成，单击 debug (调试) 按钮，复位 CPU，重新启动，启用实时模式并运行。
4. 在观察窗口中将“EnableFlag”设定为 1。名为“lsrTicker”的变量将持续增加。
5. 通过观察观察窗口中的变量来确认这一点。这确认系统中断运转正常。

在软件中，被调整的关键变量汇总如下：

- SpeedRef (格式 GLOBAL_Q)：用于改变每个部件内的基准直流总线电流

关键步骤可解释如下：

- 实时编译、载入和运行程序。
- 电机将逐渐加速并最终切换至闭合环路换相模式。
- 使用变量 CurrentSet 来为 PI 控制 - PID_REG3 指定基准电流。SpeedLoopFlag 在 PI 基准从零速斜升为 SpeedRef 时自动激活。一旦这一操作完成，PI 控制器开始调节电机速度。
- 逐渐增加指令速度 (SpeedRef 值) 来增加电机速度。
- 如果需要的话，调整速度 PI 增益来获得满意的速度响应。
- 通过减少总线电压，(如构建 1 末尾所描述的那样) 使系统安全停止，使控制器脱离实时模式并复位。

当运行这一级时，Code Composer Studio 曲线图内的电流波形应该如图 30 中显示的那样。



图 30. (a) mod6 计数器, (b) BmfA, (c) BmfB (c)BmfC (无负载, 速度 0.25pu 时, Vdcbus = 160V)

PWMDAC 输出应该如图 31 中显示的那样。

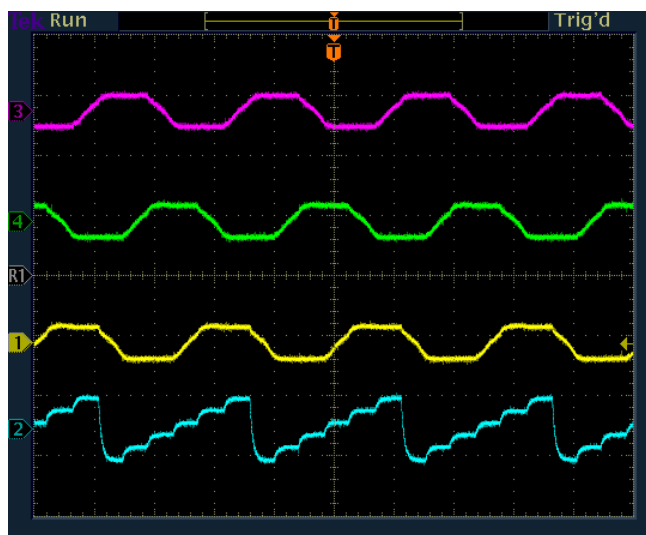


图 31. PWMDAC 输出 a) BemfA, (b) BemfB, (c)BemfC 和 (d) mod6 计数器（无负载，速度 0.25pu 时，Vdcbus = 160V）

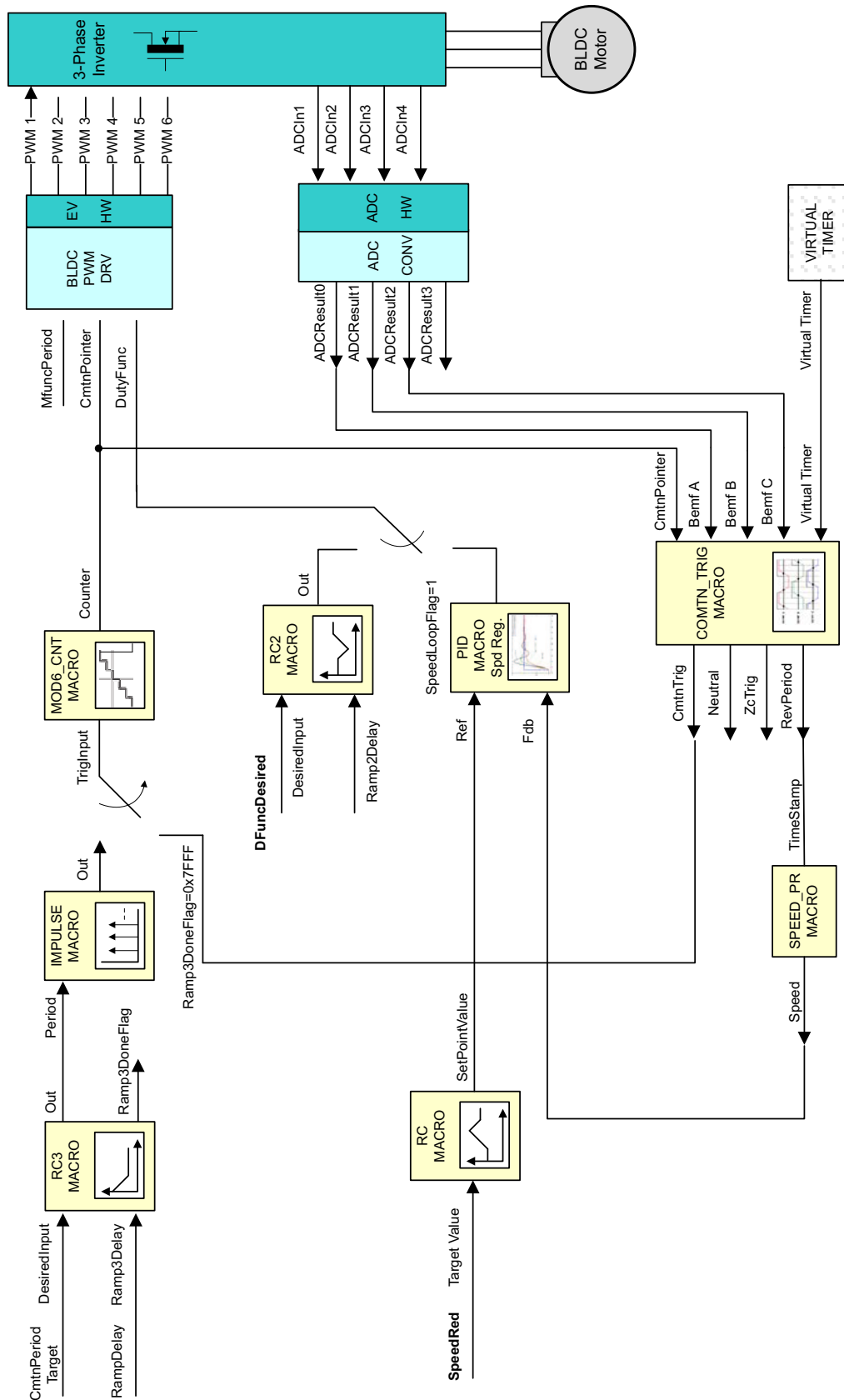


图 32.6 级 - 递增系统构建方框图

6 级验证闭合速度环路和速度 PI 控制器。

重要声明

德州仪器(TI) 及其下属子公司有权根据 JESD46 最新标准, 对所提供的产品和服务进行更正、修改、增强、改进或其它更改, 并有权根据 JESD48 最新标准中止提供任何产品和服务。客户在下订单前应获取最新的相关信息, 并验证这些信息是否完整且是最新的。所有产品的销售都遵循在订单确认时所提供的TI 销售条款与条件。

TI 保证其所销售的组件的性能符合产品销售时 TI 半导体产品销售条件与条款的适用规范。仅在 TI 保证的范围内, 且 TI 认为 有必要时才会使用测试或其它质量控制技术。除非适用法律做出了硬性规定, 否则没有必要对每种组件的所有参数进行测试。

TI 对应用帮助或客户产品设计不承担任何义务。客户应对其使用 TI 组件的产品和应用自行负责。为尽量减小与客户产品和应用相关的风险, 客户应提供充分的设计与操作安全措施。

TI 不对任何 TI 专利权、版权、屏蔽作品权或其它与使用了 TI 组件或服务的组合设备、机器或流程相关的 TI 知识产权中授予 的直接或隐含权限作出任何保证或解释。TI 所发布的与第三方产品或服务有关的信息, 不能构成从 TI 获得使用这些产品或服务 的许可、授权、或认可。使用此类信息可能需要获得第三方的专利权或其它知识产权方面的许可, 或是 TI 的专利权或其它 知识产权方面的许可。

对于 TI 的产品手册或数据表中 TI 信息的重要部分, 仅在没有对内容进行任何篡改且带有相关授权、条件、限制和声明的情况 下才允许进行复制。TI 对此类篡改过的文件不承担任何责任或义务。复制第三方的信息可能需要服从额外的限制条件。

在转售 TI 组件或服务时, 如果对该组件或服务参数的陈述与 TI 标明的参数相比存在差异或虚假成分, 则会失去相关 TI 组件 或服务的所有明示或暗示授权, 且这是不正当的、欺诈性商业行为。TI 对任何此类虚假陈述均不承担任何责任或义务。

客户认可并同意, 尽管任何应用相关信息或支持仍可能由 TI 提供, 但他们将独力负责满足与其产品及其应用中使用的 TI 产品 相关的所有法律、法规和安全相关要求。客户声明并同意, 他们具备制定与实施安全措施所需的全部专业技术和知识, 可预见 故障的危险后果、监测故障及其后果、降低有可能造成人身伤害的故障的发生机率并采取适当的补救措施。客户将全额赔偿因 在此类安全关键应用中使用任何 TI 组件而对 TI 及其代理造成的任何损失。

在某些场合中, 为了推进安全相关应用有可能对 TI 组件进行特别的促销。TI 的目标是利用此类组件帮助客户设计和创立其特 有的可满足适用的功能安全性标准 and 要求的终端产品解决方案。尽管如此, 此类组件仍然服从这些条款。

TI 组件未获得用于 FDA Class III (或类似的生命攸关医疗设备) 的授权许可, 除非各方授权官员已经达成了专门管控此类使 用的特别协议。

只有那些 TI 特别注明属于军用等级或“增强型塑料”的 TI 组件才是设计或专门用于军事/航空应用或环境的。购买者认可并同 意, 对并非指定面向军事或航空航天用途的 TI 组件进行军事或航空航天方面的应用, 其风险由客户单独承担, 并且由客户独 力负责满足与此类使用相关的所有法律和法规要求。

TI 已明确指定符合 ISO/TS16949 要求的产品, 这些产品主要用于汽车。在任何情况下, 因使用非指定产品而无法达到 ISO/TS16949 要 求, TI 不承担任何责任。

	产品		应用
数字音频	www.ti.com.cn/audio	通信与电信	www.ti.com.cn/telecom
放大器和线性器件	www.ti.com.cn/amplifiers	计算机及周边	www.ti.com.cn/computer
数据转换器	www.ti.com.cn/dataconverters	消费电子	www.ti.com.cn/consumer-apps
DLP® 产品	www.dlp.com	能源	www.ti.com.cn/energy
DSP - 数字信号处理器	www.ti.com.cn/dsp	工业应用	www.ti.com.cn/industrial
时钟和计时器	www.ti.com.cn/clockandtimers	医疗电子	www.ti.com.cn/medical
接口	www.ti.com.cn/interface	安防应用	www.ti.com.cn/security
逻辑	www.ti.com.cn/logic	汽车电子	www.ti.com.cn/automotive
电源管理	www.ti.com.cn/power	视频和影像	www.ti.com.cn/video
微控制器 (MCU)	www.ti.com.cn/microcontrollers		
RFID 系统	www.ti.com.cn/rfidsys		
OMAP应用处理器	www.ti.com.cn/omap		
无线连通性	www.ti.com.cn/wirelessconnectivity	德州仪器在线技术支持社区	www.deyisupport.com

邮寄地址: 上海市浦东新区世纪大道 1568 号, 中建大厦 32 楼 邮政编码: 200122
Copyright © 2013 德州仪器 半导体技术 (上海) 有限公司