

简介

作者：Gheorghe Turcan、Grig Barbulescu 和 Teodor Lina, Microchip Technology Inc.

驱动无刷直流（Brushless DC, BLDC）电机或永磁同步电机（Permanent Magnet Synchronous Motor, PMSM）的标准化方法有两种：梯形驱动（或方波换向）与正弦驱动。

梯形驱动相对简单，即便采用 8 位单片机（MCU）也能轻松实现，因为其本质是由少量反馈源控制状态切换的驱动状态机。在无传感器解决方案中，电机的反电动势（Back Electromotive Force, BEMF）作为转速指标，其基于电机三相的过零检测（Zero-Cross Detection, ZCD）结果每旋转 60° 提供一次转子位置。

若要为旋转中的电机测量 BEMF，必须停止驱动线圈以避免信号失真。当半桥的上桥臂和下桥臂 MOSFET 均驱动为逻辑低电平时，即可满足该条件。

在梯形驱动中，由于一个完整电周期的六个步骤中有两步能够依次每相满足所需的条件，并通过硬件电路和软件算法滤除其他相的干扰，因此可以实现 BEMF 测量。

与梯形驱动相比，正弦驱动具有转矩纹波更低、效率更高、低速转矩更优等优势，但代价是复杂度显著增加。此外，在正弦驱动中，三相始终处于驱动状态。因此，没有时间窗口可用于直接采集 BEMF。

在正弦驱动中，通常采用磁场定向控制（Field Oriented Control, FOC）方法来采集 BEMF，这需要快速精确的单个线圈电流测量、复杂的数学变换，以及动态信号重建与预测。为了实现 FOC，MCU 需具备高于 100 MHz 的时钟频率、快速 ADC，以及只有 16 位和 32 位内核才具备的数学处理能力。

本技术简介中介绍的方法为直接测量 BEMF 创造了必要的条件，同时仍能持续对三相进行正弦驱动，从而无需依赖复杂的采集和控制算法。

该解决方案可轻松在电机每旋转 60° 时获取一次 ZCD，从而将驱动同步机制简化至 8 位 MCU 也能处理的程度，以便实现简单的无传感器正弦电机控制方案。

目录

简介.....	1
1. 概述.....	3
2. 正弦驱动测量窗口插入.....	4
3. BEMF 测量与解释.....	6
4. 使用 BEMF 实现驱动同步.....	7
5. BEMF 采集演示.....	10
6. BEMF 采集演示结果.....	15
7. 结论.....	16
8. 参考资料.....	17
9. 版本历史.....	18
Microchip 信息.....	19
Microchip 网站.....	19
产品变更通知服务.....	19
客户支持.....	19
Microchip 器件代码保护功能.....	19
法律声明.....	19
商标.....	20
质量管理体系.....	20
全球销售及服务网点.....	21

1. 概述

当电感在磁场中移动时，其内部会产生感应电压。无刷电机由三个电感组成，这些电感在环形排列的磁场中旋转，从而在电机的线圈中产生感应电压。BEMF 是电机旋转时必然会产生感应电压。在理想情况下，只要电机转速非零，就会产生 BEMF。但在实际应用中，受噪声和测量分辨率的限制，需要达到最低要求的每分钟转数（Rotation Per Minute, RPM）才能测量到 BEMF，以此通过 ZCD 检测提供转子的位置。

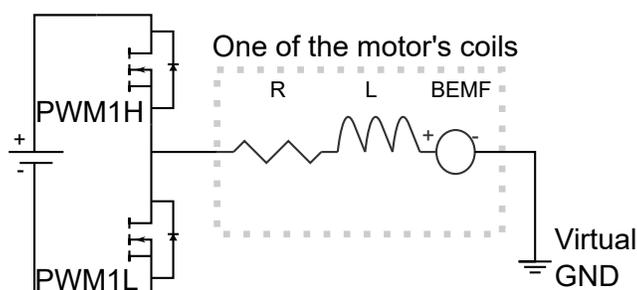
BEMF 电压具有三个主要参数：幅值、频率和相位。

幅值与转子转速成正比且取决于电机具体型号。

频率与转子的转速和极对数成正比。通过使用 BEMF 频率以及表征电机特性的方程，可以计算出电机的实际转速（RPM）。

相位对于定子的旋转磁场与转子之间的同步至关重要。相位和频率用作驱动同步（与电机的相位和速度保持一致）及控制环实现的反馈信号。

图 1-1. 三相无刷电机的其中一相



为确保正常进行 BEMF 测量和 ZCD 检测，必须同时禁止所有三相的驱动信号以避免干扰测量。但这一条件过于苛刻，即使在梯形波驱动中也只能禁止驱动其中一相，其他两相仍处于驱动状态并会产生噪声。ZCD 通过使用模拟和数字滤波器以及多数检测算法来实现。更多信息，可参考[多数检测算法](#)。这些滤波器旨在降低噪声以实现尽可能精确的 ZCD 识别，但是会在实际 ZCD 与测量 ZCD 之间引入延时。

在无传感器正弦驱动中，所有三相持续导通会导致没有不受干扰的时间窗口可用于直接测量电机 BEMF。因此，通常采用的 FOC 方法需测量各线圈电流并通过数学模型重构 BEMF，这对硬件性能要求较高，不适用于 8 位 MCU。对于 8 位 MCU 实现的正弦驱动，主流方法是采用基于锁相环（Phase-Locked Loop, PLL）同步机制的带传感器解决方案。

本技术简介提出了一种新型正弦驱动方法，通过创建一个可确保驱动信号对相位测量点无干扰的直接 BEMF 采集窗口，实现了无需滤波处理且更接近实际电机线圈电流 ZCD 的简单 ZCD 检测方案。

这种新型正弦驱动方法需要脉宽调制（Pulse-Width Modulation, PWM）上升事件对齐和死区控制，具体可通过纯软件或者软硬件结合的方式来实现。因此，本文所提出的方法可用于任何能够生成互补 PWM 信号的 8 位 MCU。但是，使用 Microchip AVR[®] EB 系列中的专用硬件外设可大幅缩短代码长度和中断时间。

2. 正弦驱动测量窗口插入

为消除驱动信号的影响，必须在相应半桥的两个晶体管均驱动为低电平时对 BEMF 进行采样。

如图 2-1 所示，互补 PWM 死区为采样 BEMF 提供了这样的时间窗口。死区旨在防止半桥中发生电流直通。

但是，各相的死区未必对齐。为此，提出的方法在所有六个 PWM 信号上增加了一个死区，并且所有死区对齐，从而创建了一个足够长的时间窗口来测量 BEMF，如图 2-2 所示。

图 2-1. 死区与 PWM 信号图

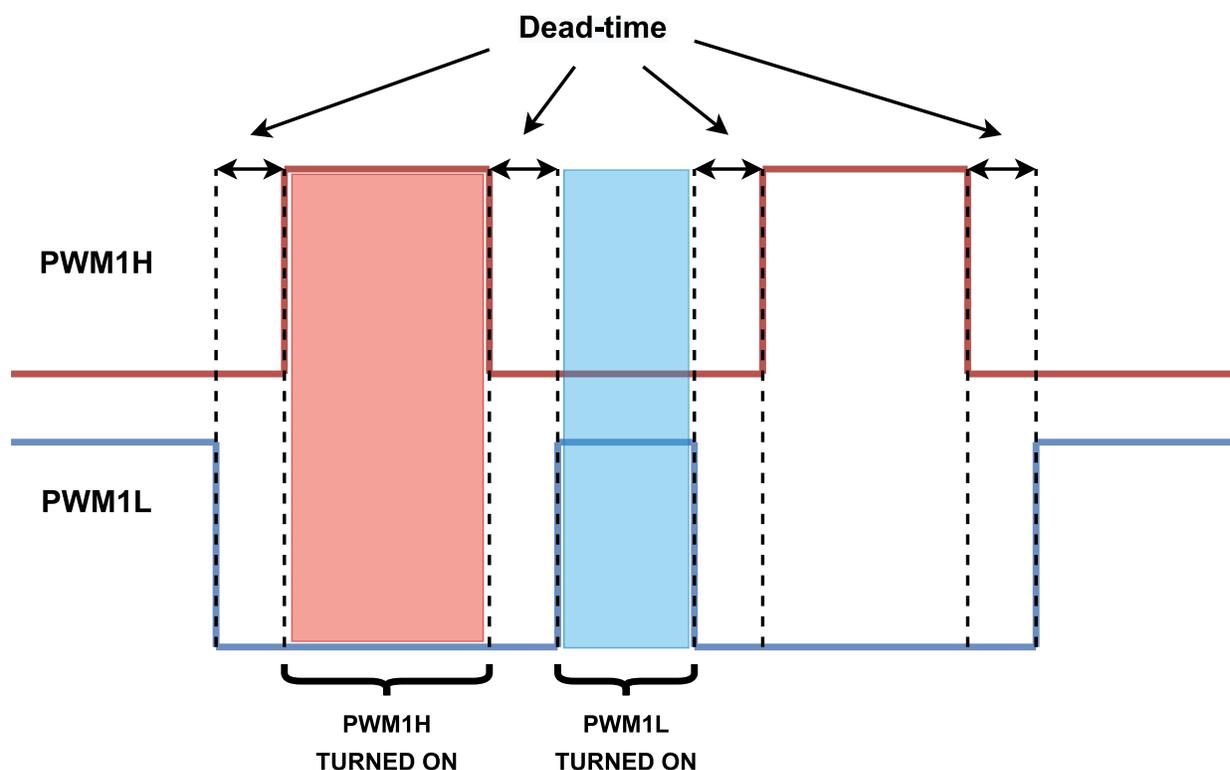
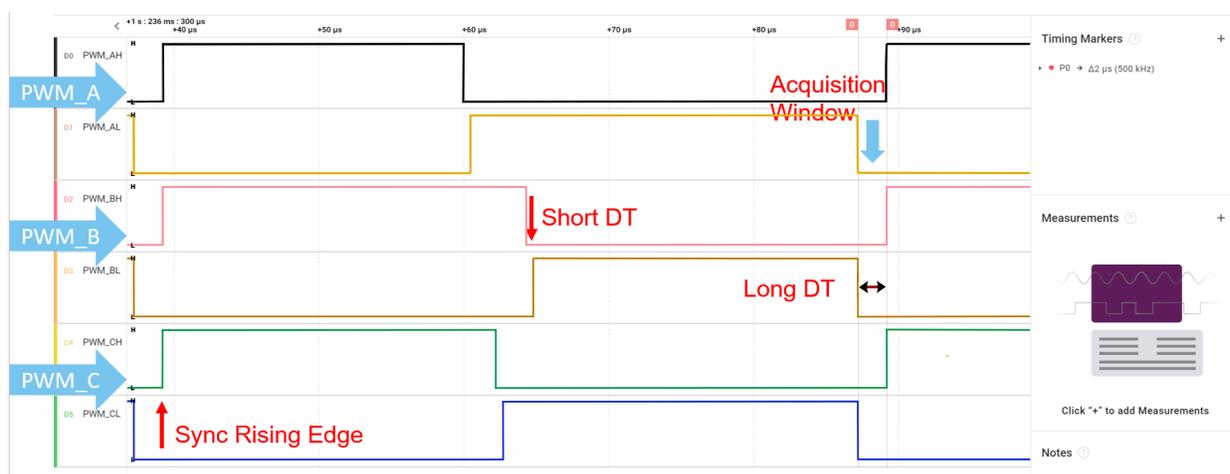


图 2-2. 通过逻辑分析器捕捉波形以可视化较长的死区



与普通的正弦驱动相比，插入 BEMF 采集窗口存在以下两个小缺点：

1. 由于死区延长，PWM 占空比范围按比例减小，这一比例取决于 PWM 频率以及正确评估 BEMF 所需的时间。
2. 由于死区延长，半桥内部二极管温度升高。当 BEMF 为正值时，功率损耗增大。

除上述缺点外，该方法是三相电机（BLDC 或 PMSM）正弦无传感器控制的低成本替代方案，并且可以在 8 位 MCU 上实现，因为其在完整的 PWM 周期内无需大量算力。电机控制通常采用 20 kHz 的 PWM 开关频率，因此一个完整的 PWM 周期为 50 μ s。

3. BEMF 测量与解释

可使用模数转换器（AD converter, ADC）或模拟比较器（Analog Comparator, AC）等多种外设对 BEMF 进行采样并检测过零点。在本方法中，通过模拟比较器对 BEMF 进行采样。所有 PWM 驱动信号必须同步保持逻辑低电平，此时未驱动线圈，经所有传播延时后测得的信号即为电机的 BEMF。

模拟比较器的同相输入连接至电机分相信号，反向输入连接至电机中性点，如图 4-1 所示。中性点既可通过软件重构，也可采用星型电阻网络在硬件层面实现。比较器输出的采集时序需与死区及潜在延时相匹配。比较器输出提供与线圈当前位置对应的 BEMF 位置。利用该信号，可以实现过零点检测并估计转子的位置。图 3-1 显示了模拟比较器输出，图 3-2 显示了过零点的总体特征。

图 3-1. 通过模拟比较器解释的 BEMF 信号

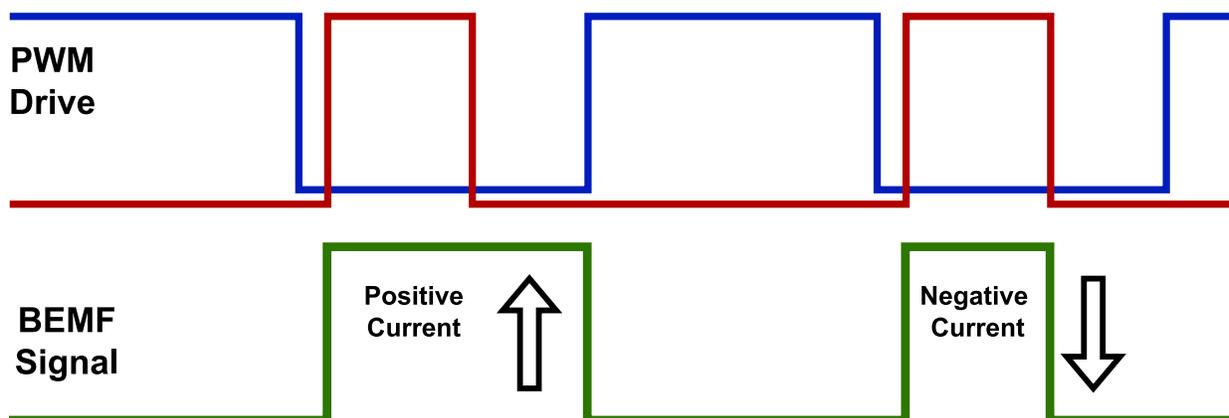
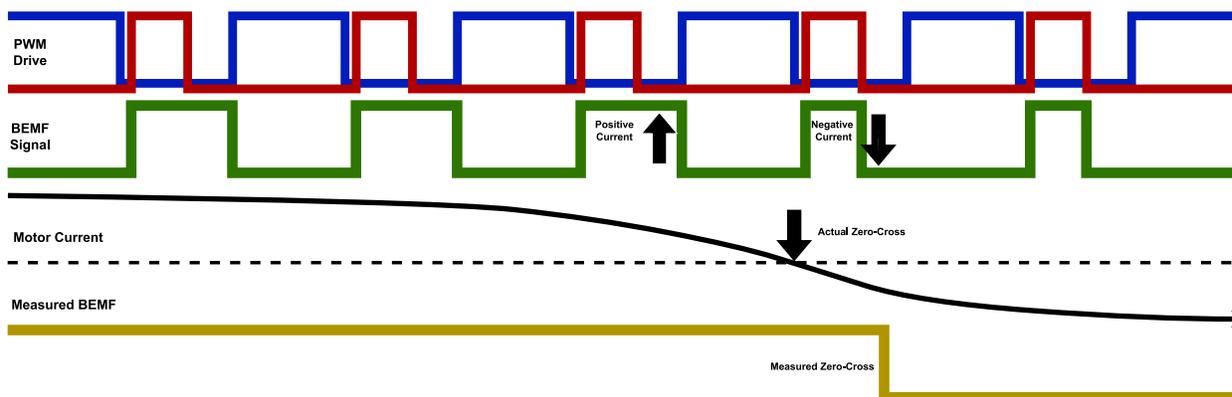


图 3-2. BEMF 过零点

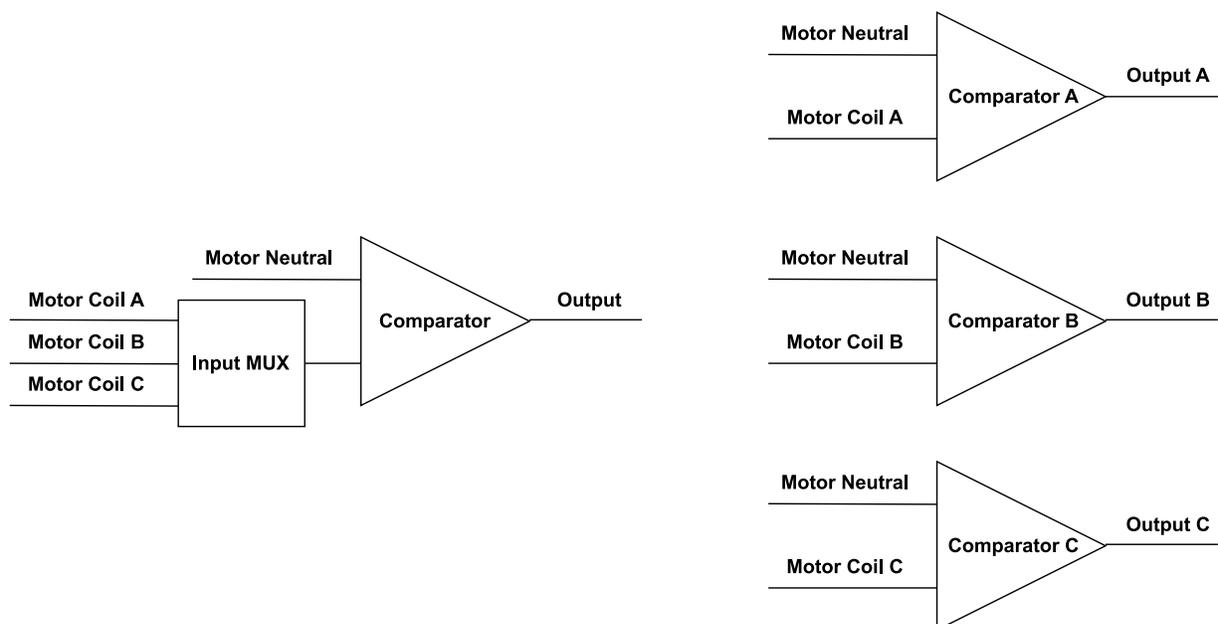


4. 使用 BEMF 实现驱动同步

由于每次只对一相进行 ZCD，无需同时对所有三相进行采样，因此对 BEMF 进行采样所需的元件可以是三个比较器，或者是一个可通过多路开关选择输入的比较器，如图 4-1 所示。

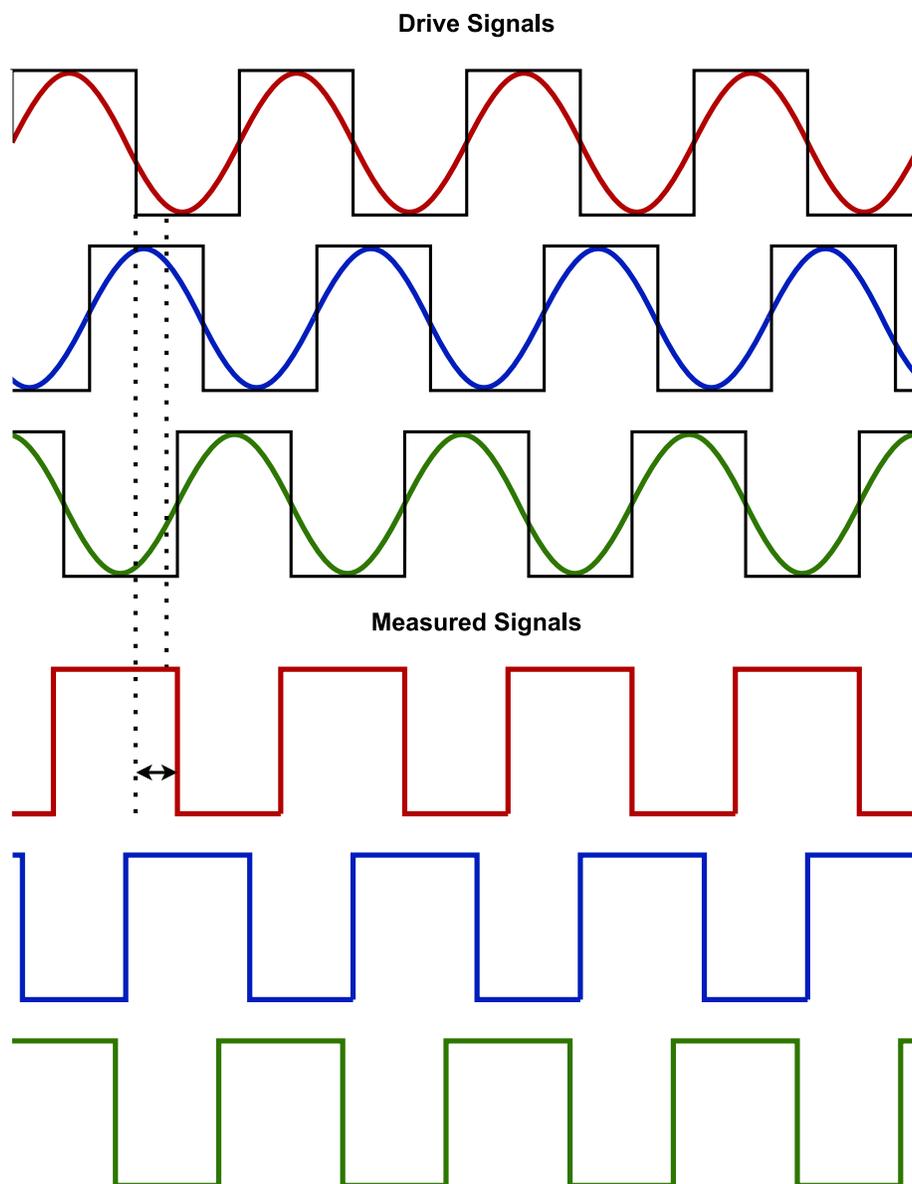
必须考虑比较器延时和多路开关稳定性。为了避免延时，可以使用优先级选择方法：首先测量电机的第一相，直至达到过零点；然后测量第二相，直至达到另一个过零点，以此类推。

图 4-1. 可供选择的模拟比较器配置



根据采样得到的 BEMF 值，电机可能超前或滞后于驱动场。为了减小驱动场与电机之间的误差，可引入控制算法以改善电机换向的同步性能。图 4-2 显示了当电机滞后于驱动场时，所有三个驱动通道与经过滤波后的三个 BEMF 通道的波形。

图 4-2. 电机三相及每相测得的 BEMF



根据从 BEMF ZCD 获得的信息，可能存在以下三种情况：

1. BEMF 信号超前于驱动信号，因此必须降低驱动速度以保持同步。
2. BEMF 信号滞后于驱动信号，因此必须提高驱动速度以保持同步。
3. BEMF 信号与驱动信号在一定范围内保持同相。

图 4-3 和图 4-4 分别展示了第 1 种情况和第 2 种情况下电机的其中一相及其对应的 BEMF 信号。第一张图显示了 BEMF 信号超前于驱动信号的情况。第二张图显示了 BEMF 信号滞后于驱动信号的情况。

图 4-3. BEMF 信号超前于正弦驱动信号

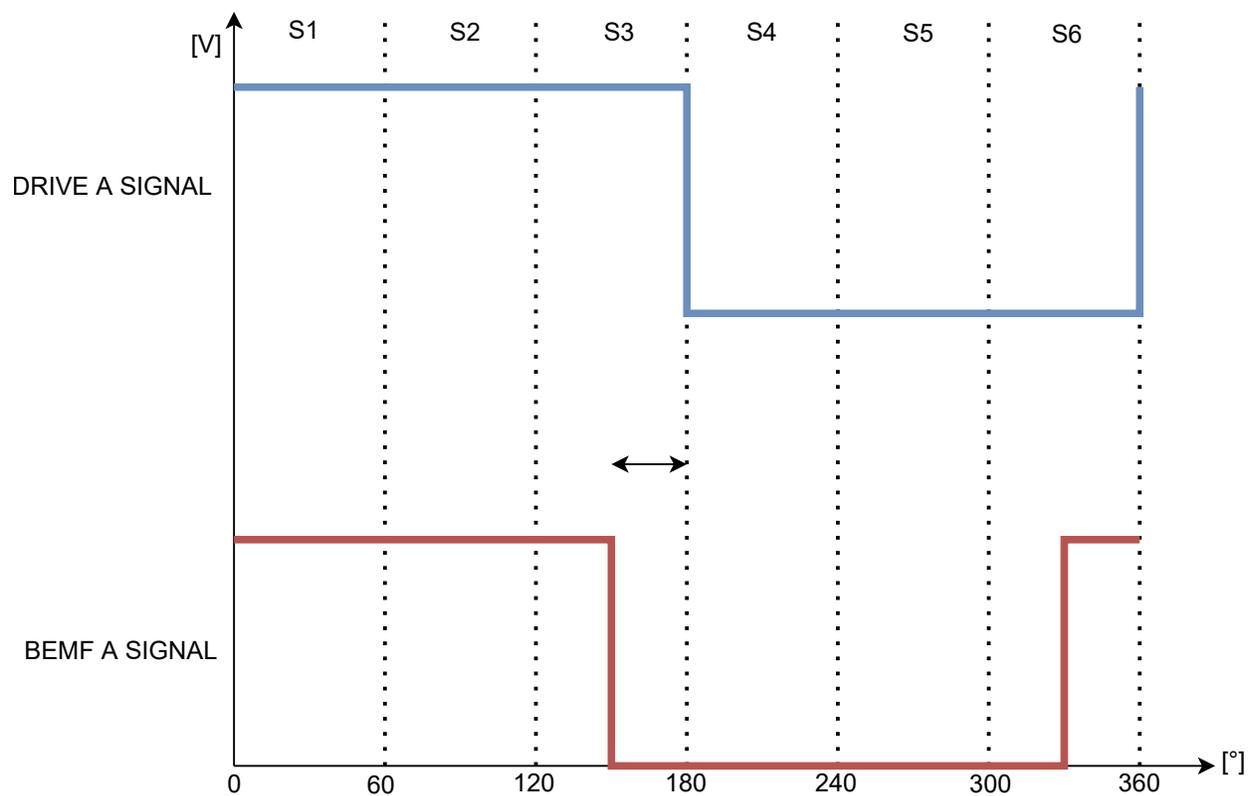
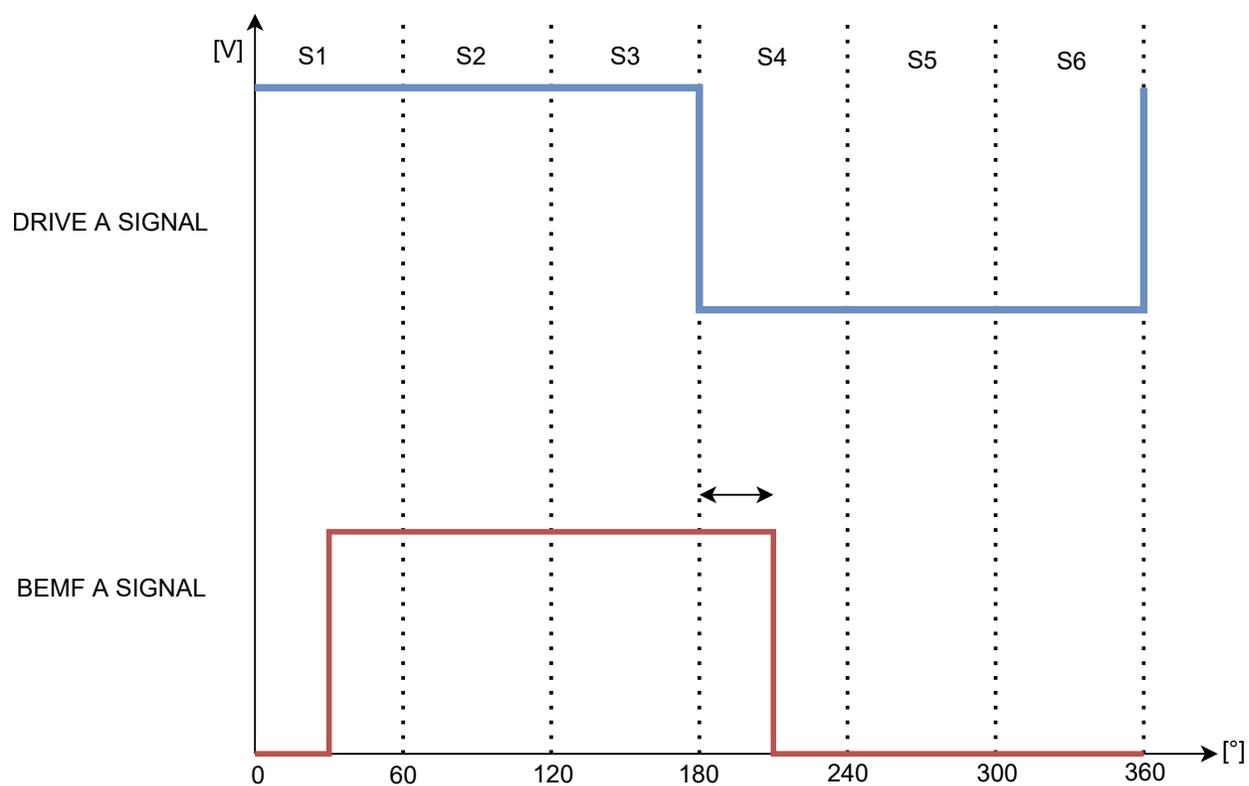


图 4-4. BEMF 信号滞后于正弦驱动信号



5. BEMF 采集演示

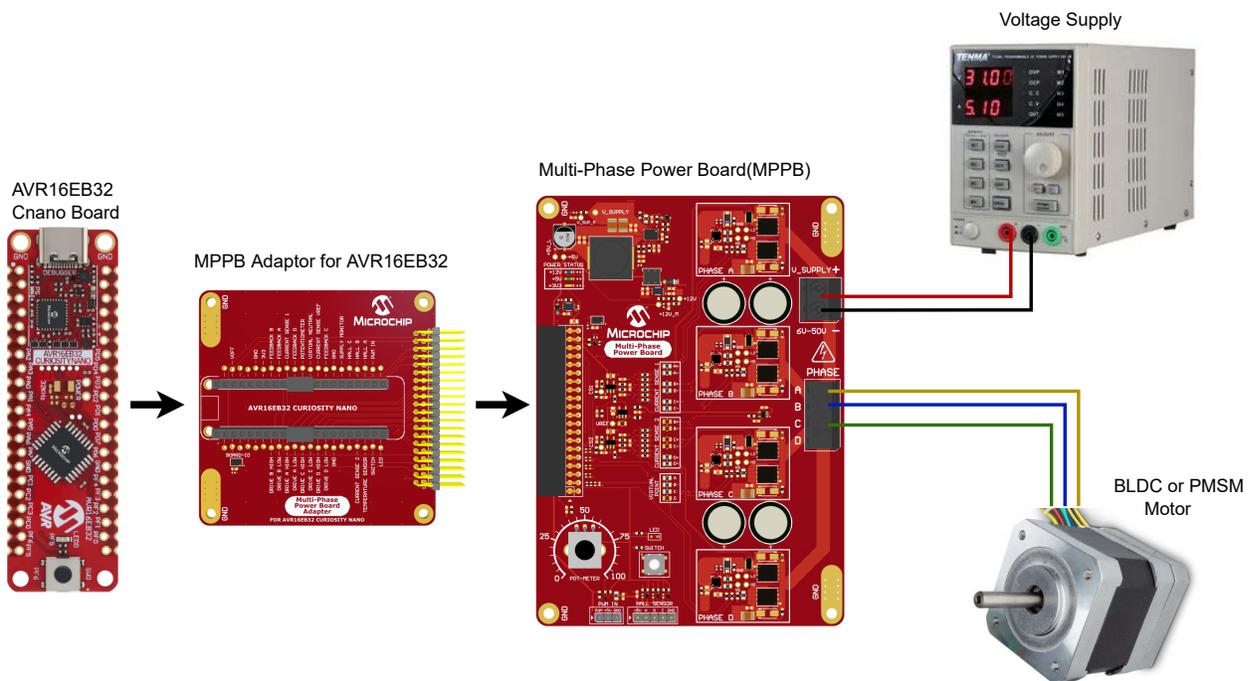
本例展示了如何设置 E 型定时器/计数器（Timer/Counter type E, TCE）和波形扩展（Waveform Extension, WEX）外设以生成六个互补 PWM 信号，从而驱动三相无刷直流电机（BLDC）或采用正弦驱动的永磁同步电机（PMSM）。此外，本例还将模拟比较器（AC）外配置为测量电机旋转时的反电动势（BEMF）。

测量窗口为 PWM 周期内的两个死区之一。将死区延长至 $2\ \mu\text{s}$ 可提供足够大的采集窗口，以便在不受驱动信号影响的情况下检测电流过零点。另一个死区为 $500\ \text{ns}$ 。一个完整的 PWM 周期为 $50\ \mu\text{s}$ （经典的 $20\ \text{kHz}$ MOSFET 开关频率，通常用于电机控制应用）。

在本例中，通过强制换向旋转电机，无需同步或闭环算法。该代码示例重点展示了在死区内实现的新型 BEMF 测量方法。电机以恒定的速度和幅值（运行时保持不变）旋转，因为这并非本应用的主要目标。

下图展示了本应用中使用的硬件设置：

图 5-1. 本应用所需的硬件设置



要使用 MPLAB[®] 代码配置器（MPLAB Code Configurator, MCC）Melody（即 MCC Melody，不支持 MCC Classic）生成该项目，请按照以下步骤操作：

1. 为 AVR16EB32 新建一个 MPLAB X IDE 项目。
2. 从工具栏中打开 MCC（有关安装 MCC 插件的更多信息，请单击[此处](#)）。
3. 在 MCC 内容管理器向导中选择 **MCC Melody**，然后单击 **Finish**（完成）。
4. 单击 **Project Resources**（项目资源），转到 **System**（系统），选择 CLKCTRL，然后禁用预分频器使能按钮。
5. 从 **Device Resource**（器件资源）中转到 **Drivers**（驱动程序）和 **Timer**（定时器），添加 TCE 模块，然后进行以下配置：
 - 模块使能：默认必须使能。如果未使能，请切换按钮（使能后会变成蓝色）
 - 时钟选择：系统时钟（默认情况下，分频值必须为 1——系统时钟）

- 计数器方向：向上
 - 波形生成模式：顶部溢出的单斜率 PWM 模式
 - 请求的周期[s]：0.00005
 - 占空比 0 [%]：0
 - 占空比 1 [%]：0
 - 占空比 2 [%]：0
 - 占空比 3 [%]：98.5
 - 波形输出 n：勾选波形输出 0、1、2、3 的使能列中的框
 - 缩放模式：CMP 值从中间位置（50%占空比）开始缩放
 - 缩放写入寄存器：正常
 - 幅值控制使能：切换按钮（使能后会变成蓝色）
 - 幅值：0.1
 - 产生 ISR：切换按钮（使能后会变成蓝色）
 - 比较 3 中断允许：切换按钮（使能后会变成蓝色）
6. 从 **Device Resource** 中转到 **Drivers** 并添加 WEX 模块，然后进行以下配置：
- 输入矩阵：直接
 - 更新源：TCE（输出信号的更新条件将由 TCE 决定）
- 改写设置：勾选波形输出[0-5]的输出使能列中的所有框
- 死区插入通道 0 使能：切换按钮（使能后会变成蓝色）
 - 死区插入通道 1 使能：切换按钮（使能后会变成蓝色）
 - 死区插入通道 2 使能：切换按钮（使能后会变成蓝色）
 - 请求的下桥臂死区（ μs ）：2
 - 请求的上桥臂死区（ μs ）：0.5
7. 从 **Device Resource** 中转到 **Drivers** 并添加 AC0 模块，然后进行以下配置：
- 使能：切换按钮（使能后会变成蓝色）
 - 同相输入多路开关选择：同相引脚 5
 - 反相输入多路开关选择：反相引脚 1
 - 输出焊盘使能：切换按钮（使能后会变成蓝色）
8. 在 **Pin Grid View**（引脚网格视图）窗口中——检查端口 A 上的 WEX_WO[0-5]引脚是否锁定为输出。当勾选波形输出 n 的使能列中的框时，会锁定引脚。要更改端口，请单击 **Pin Grid View** 中另一个端口的引脚。检查 PA7 引脚是否设置为 AC 输出。检查 PD4 和 PD0 引脚是否设置为 AC 的输入。此外，还需另外两个引脚作为电机其他两相的 AC 的输入。单击 PD5 和 PD6，然后从 Pins -> GPIO（引脚 -> GPIO）中将其设置为输入。最后，必须从 Pins -> GPIO 中将显示采样得到的 BEMF 和采样时刻的引脚设置为输出。这些引脚分别为 PF1、PF2、PF3 和 PF4。
9. 在 **Project Resources** 窗口中，单击 **Generate**（生成）按钮以使 MCC 生成所有指定的驱动程序和配置。

10. 当 MCC Melody 基于上述配置生成项目文件后，按照以下操作改写 main.c 文件的内容：

添加用于驱动电机的宏定义、变量和查找表（Look-Up Table, LUT）：

```

/* BLDC 电机的极对数 */
#define MOTOR_PAIR_POLES 4

/* MOSFET 开关频率 (Hz) */
#define F_SAMPLING 20000.0

/* uint16_t 范围映射: 0 - 359.99 电角度 -> 0 - 65535 */
#define DEGREES_TO_U16(DEG) ((uint16_t)((float)(DEG) * 65536.0 / 360.0 + 0.5))

/* 将转速从 RPM 转换为 LUT 滚动速度 */
#define RPM_TO_U16(RPM) ((uint16_t)(((float)(RPM) * 65536.0 * (float)(MOTOR_PAIR_POLES)) / ((float)(F_SAMPLING) * 60.0) + 0.5))

/* 设置正弦波信号的幅值，从而设置占空比的缩放值，
 * U.Q.1.15 格式，范围从 0 至 1.00。占空比缩放由硬件完成，
 * 使用 TCE 的硬件加速器。 */
#define AMP_TO_U16(X) ((uint16_t)(32768.0*(X) + 0.5))

/* 电机的转速——120 RPM */
#define SPEED RPM_TO_U16(120)

/* 正弦波的幅值——10% */
#define AMPLITUDE AMP_TO_U16(0.1)

#include "mcc_generated_files/system/system.h"

typedef enum
{
    MUX_PHASE_A = (AC_MUXPOS_AINP5_gc | AC_MUXNEG_AINN1_gc),
    MUX_PHASE_B = (AC_MUXPOS_AINP6_gc | AC_MUXNEG_AINN1_gc),
    MUX_PHASE_C = (AC_MUXPOS_AINP3_gc | AC_MUXNEG_AINN1_gc),
} mux_t;

/* 用于生成正弦驱动信号的 LUT */
static const uint16_t sine_lookup_table[] =
{
    16384, 16786, 17187, 17589, 17989, 18389, 18788, 19185, 19580, 19973, 20364,
    20753, 21140, 21523, 21903, 22280, 22653, 23023, 23389, 23750, 24107, 24459, 24807,
    25149, 25486, 25818, 26143, 26463, 26777, 27085, 27386, 27681, 27969, 28250, 28523,
    28790, 29049, 29300, 29543, 29779, 30006, 30226, 30437, 30639, 30833, 31018, 31194,
    31362, 31520, 31670, 31810, 31941, 32062, 32174, 32276, 32369, 32453, 32526, 32590,
    32644, 32689, 32723, 32748, 32763, 32768, 32763, 32748, 32723, 32689, 32644, 32590,
    32526, 32453, 32369, 32276, 32174, 32062, 31941, 31810, 31670, 31520, 31362, 31194,
    31018, 30833, 30639, 30437, 30226, 30006, 29779, 29543, 29300, 29049, 28790, 28523,
    28250, 27969, 27681, 27386, 27085, 26777, 26463, 26143, 25818, 25486, 25149, 24807,
    24459, 24107, 23750, 23389, 23023, 22653, 22280, 21903, 21523, 21140, 20753, 20364,
    19973, 19580, 19185, 18788, 18389, 17989, 17589, 17187, 16786, 16384, 15981, 15580,
    15178, 14778, 14378, 13979, 13582, 13187, 12794, 12403, 12014, 11627, 11244, 10864,
    10487, 10114, 9744, 9378, 9017, 8660, 8308, 7960, 7618, 7281, 6949, 6624, 6304, 5990,
    5682, 5381, 5086, 4798, 4517, 4244, 3977, 3718, 3467, 3224, 2988, 2761, 2541, 2330,
    2128, 1934, 1749, 1573, 1405, 1247, 1097, 957, 826, 705, 593, 491, 398, 314, 241,
    177, 123, 78, 44, 19, 4, 0, 4, 19, 44, 78, 123, 177, 241, 314, 398, 491, 593, 705,
    826, 957, 1097, 1247, 1405, 1573, 1749, 1934, 2128, 2330, 2541, 2761, 2988, 3224,
    3467, 3718, 3977, 4244, 4517, 4798, 5086, 5381, 5682, 5990, 6304, 6624, 6949, 7281,
    7618, 7960, 8308, 8660, 9017, 9378, 9744, 10114, 10487, 10864, 11244, 11627, 12014,
    12403, 12794, 13187, 13582, 13979, 14378, 14778, 15178, 15580, 15981
};

```

添加 Motor_Drive 函数。该函数用于以给定的速度更新驱动信号并生成正弦驱动信号。

```

/* 每 50 μs 调用一次以更新驱动信号的函数 */
void Motor_Drive(void)
{
    /* 在运行时滚动 LUT 的计数器。这些计数器用于创建
     * 电机各相之间的 120 度相移 */
    static uint16_t phase_a = DEGREES_TO_U16(0.0);
    static uint16_t phase_b = DEGREES_TO_U16(120.0);
    static uint16_t phase_c = DEGREES_TO_U16(240.0);
    static const uint16_t speed = SPEED;

    /* 将写入 TCE 的 CMP 通道的值 */
    uint16_t drive_a, drive_b, drive_c;

    /* 更新计数器 */
    phase_a += speed;
    phase_b += speed;
    phase_c += speed;

    /* 从 LUT 中为每个 CMP 通道选择新变量 */
    drive_a = sine_lookup_table[(phase_a >> 8)];
    drive_b = sine_lookup_table[(phase_b >> 8)];
    drive_c = sine_lookup_table[(phase_c >> 8)];

    /* 使用新值更新 CMP 通道中的值 */
    TCE0_CompareChannels012BufferedSet(drive_a, drive_b, drive_c);
}

```

添加 Mux_Set 函数。该函数用于将 BEMF 监视器从一相切换到另一相，因为本例仅使用一个模拟比较器。

```

/* 运行时切换多路开关以监视所有三相的函数 */
void Mux_Set(uint8_t mode)
{
    uint8_t temp;
    temp = AC0.MUXCTRL;
    temp &= ~(AC_MUXPOS_gm | AC_MUXNEG_gm);
    temp |= mode;
    AC0.MUXCTRL = temp;
}

```

添加 BEMF_Read 函数。该函数用于对 BEMF 信号进行采样并将其显示在通用输入/输出（General Purpose Input/Output, GPIO）引脚上，还用于调用 Mux_Set 和 Motor_Drive 函数。

```

/* 在延长的死区内调用以读取 BEMF 状态的函数 */
void BEMF_Read(void)
{
    bool bemf_state;
    static mux_t mux = MUX_PHASE_A;

    /* 通过翻转 IO 标记 BEMF 采样点 */
    IO_PF4_SetHigh();
    bemf_state = ((AC0.STATUS & AC_CMPSTATE_bm) != 0);
    IO_PF4_SetLow();

    /* 将 AC0 多路开关从一相切换到另一相 */
    switch(mux)
    {
        case MUX_PHASE_A: if(bemf_state) {IO_PF1_SetHigh();} else {IO_PF1_SetLow();}
                          mux = MUX_PHASE_B; break;
        case MUX_PHASE_B: if(bemf_state) {IO_PF2_SetHigh();} else {IO_PF2_SetLow();}
                          mux = MUX_PHASE_C; break;
        case MUX_PHASE_C: if(bemf_state) {IO_PF3_SetHigh();} else {IO_PF3_SetLow();}
                          mux = MUX_PHASE_A; break;
        default: mux = MUX_PHASE_A; break;
    }

    /* 更新模拟比较器多路开关以监视电机的另一相 */
    Mux_Set(mux);

    /* 更新驱动信号以使电机保持旋转 */
    Motor_Drive();
}

```

编辑 main.c 文件。将 BEMF_Read 函数注册为 TCE CMP3 通道的回调函数，使能 TCE 硬件缩放加速器并设置所需的幅值。

```

int main(void)
{
    SYSTEM_Initialize();

    /* 将 BEMF 采样函数注册为回调函数 */
    TCE0_Compare3CallbackRegister(BEMF_Read);

    /* 初始化后使能硬件缩放加速器，以避免在
     * 通过 TCE 的 CMP3 通道读取 BEMF 时发生时序混乱 */
    TCE0_ScaleEnable(true);

    TCE0_AmplitudeSet(AMPLITUDE);

    while(1)
    {
    }
}

```

- 现在即可从 MPLAB X IDE 中编译并运行该项目。在运行时，当电机旋转时，使用 AC 测量电机各相的 BEMF，并通过一些输出 GPIO 引脚进行显示。可以观察到，测得的 BEMF 遵循电机各相的正弦驱动信号。

6. BEMF 采集演示结果

下面给出了逻辑分析器捕捉到的一些波形，以展示死区内 BEMF 的采样时刻，以及采样得到的 BEMF 如何遵循电机各相的驱动信号：

图 6-1. 死区内 BEMF 的采样点

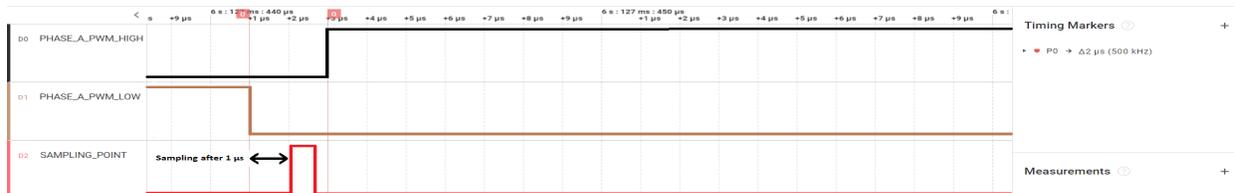
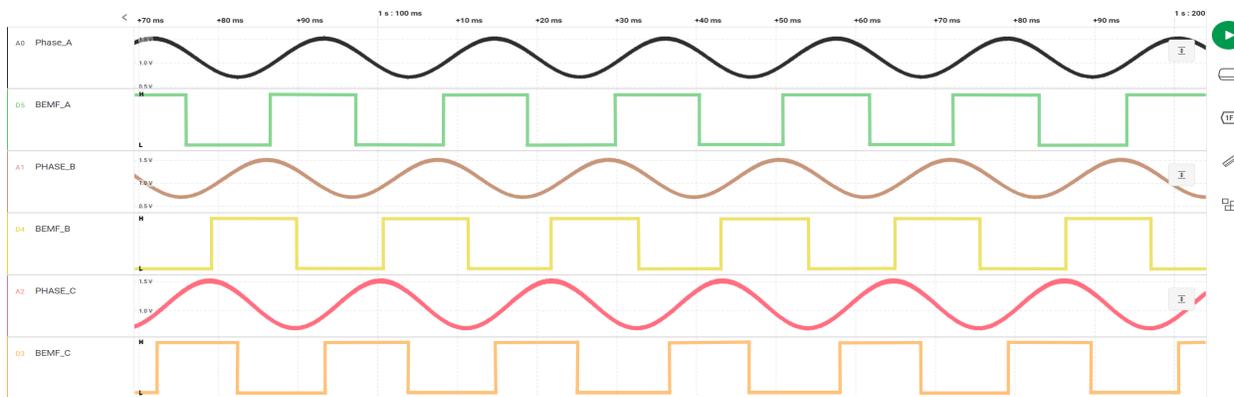


图 6-2. 对电机的其中一相采样得到的 BEMF。模拟比较器输出与采样得到的 BEMF



图 6-3. 电机三相及其采样得到的 BEMF 信号

在以下捕捉到的波形中，将相信号经过硬件低通滤波器处理后可观察到正弦驱动波形。



如需本部分所述功能的使用 MCC Melody 生成的 AVR16EB32 代码示例，可单击此处：



Click to view code examples on MPLAB DISCOVER

7. 结论

本技术简介提供了一种方法来测量以正弦无传感器模式驱动的 BLDC 或 PMSM 电机各相的 BEMF。通过延长用于晶体管开关的死区之一，可创建足够大的采集窗口以测得干净的 BEMF 信号。该方法使用 8 位 MCU 即可实现，不必具备 16 位和 32 位内核的所有功能。

使用该方法时，检测到 BEMF 的过零点信号后即可每 60° 获取一次转子的位置信息，并更新驱动信号以与电机保持同步。BEMF 与驱动电压同步时可实现最高效率，因为此时线圈两端的压降最小。

与本文档提出的方法相比，FOC 算法仍然更优，因为获取转子位置信息的频率更高（每个 PWM 周期获取一次）。这种使用死区测量 BEMF 的新方法适用于控制复杂度要求不高且动态负载变化幅度不大的低端应用。

风扇和泵等应用可采用该测量方法进行 BEMF 检测并实现正弦无传感器驱动。从噪声和效率方面来看，这种新方法实现的驱动控制仍优于经典的梯形驱动。

8. 参考资料

- [“AVR16EB14/20/28/32 Preliminary Data Sheet” \(DS40002522\)](#)。Microchip Technology Inc., 2023 年
- [“AVR16EB32 Curiosity Nano引脚图”](#) Microchip Technology Inc., 2023 年
- [Getting Started with Timer/Counter Type E \(TCE\) and WEX](#) Microchip Technology Inc., 2023 年
- [多项功率板 \(MPPB\) 页面](#)Microchip Technology Inc., 2024 年
- [AVR-EB Curiosity Nano至MPPB适配器页面](#)Microchip Technology Inc., 2024 年

9. 版本历史

文档版本	日期	备注
A	2024 年 2 月	文档初始版本

Microchip 信息

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 (Frequently Asked Questions, FAQ)、技术支持请求、在线讨论组以及 Microchip 设计合作伙伴计划成员名单
- **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
- 嵌入式解决方案工程师 (Embedded Solutions Engineer, 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。本文档附有销售办事处的联系方式。

也可通过以下网站获得技术支持：www.microchip.com/support

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：在按照操作规范正常使用的情况下，Microchip 系列产品非常安全。
- Microchip 重视并积极保护其知识产权。任何试图破坏 Microchip 产品代码保护功能的行为均可视为违反了《数字器件千年版权法案 (Digital Millennium Copyright Act)》并予以严禁。
- 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中。Microchip 承诺将不断改进产品的代码保护功能。

法律声明

本出版物中提供的信息仅仅是为方便您使用 Microchip 产品或使用这些产品来进行设计、测试以及与应用相集成。以任何其他方式使用这些信息，都将违反相关条款。器件应用信息仅为您提供便利，它们可能由更新之信息所替代。确保应用符合技术规范，是您自身应负的责任。如需更多支持，请联系您当地的 Microchip 销售办事处，或访问 www.microchip.com/en-us/support/design-help/client-support-services。

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、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、ClockWorks、The Embedded Control Solutions Company、EtherSynch、Flashtec、Hyper Speed Control、HyperLight Load、Libero、motorBench、mTouch、Powermite 3、Precision Edge、ProASIC、ProASIC Plus、ProASIC Plus 徽标、Quiet-Wire、SmartFusion、SyncWorld、TimeCesium、TimeHub、TimePictra、TimeProvider 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、Clockstudio、CodeGuard、CryptoAuthentication、CryptoAutomotive、CryptoCompanion、CryptoController、dsPICDEM、dsPICDEM.net、Dynamic Average Matching、DAM、ECAN、Espresso T1S、EtherGREEN、EyeOpen、GridTime、IdealBridge、IGaT、In-Circuit Serial Programming、ICSP、INICnet、Intelligent Paralleling、IntelliMOS、Inter-Chip Connectivity、JitterBlocker、Knob-on-Display、MarginLink、maxCrypto、maxView、memBrain、Mindi、MiWi、MPASM、MPF、MPLAB Certified 徽标、MPLIB、MPLINK、mSiC、MultiTRAK、NetDetach、Omniscient Code Generation、PICDEM、PICDEM.net、PICKit、PICKtail、Power MOS IV、Power MOS 7、PowerSmart、PureSilicon、QMatrix、REAL ICE、Ripple Blocker、RTAX、RTG4、SAM-ICE、Serial Quad I/O、simpleMAP、SimpliPHY、SmartBuffer、SmartHLS、SMART-I.S.、storClad、SQI、SuperSwitcher、SuperSwitcher II、Switchtec、SynchroPHY、Total Endurance、Trusted Time、TSHARC、Turing、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 和 Symmcom 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。

© 2025, Microchip Technology Incorporated 及其子公司版权所有。

ISBN: 979-8-3371-1222-0

质量管理体系

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

全球销售及服务网点

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	澳大利亚 Australia - Sydney 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 India - Bangalore 电话: 91-80-3090-4444 印度 India - New Delhi 电话: 91-11-4160-8631 印度 India - Pune 电话: 91-20-4121-0141 日本 Japan - Osaka 电话: 81-6-6152-7160 日本 Japan - Tokyo 电话: 81-3-6880-3770 韩国 Korea - Daegu 电话: 82-53-744-4301 韩国 Korea - Seoul 电话: 82-2-554-7200 马来西亚 Malaysia - Kuala Lumpur 电话: 60-3-7651-7906 马来西亚 Malaysia - Penang 电话: 60-4-227-8870 菲律宾 Philippines - Manila 电话: 63-2-634-9065 新加坡 Singapore 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 Thailand - Bangkok 电话: 66-2-694-1351 越南 Vietnam - Ho Chi Minh 电话: 84-28-5448-2100	奥地利 Austria - Wels 电话: 43-7242-2244-39 传真: 43-7242-2244-393 丹麦 Denmark - Copenhagen 电话: 45-4485-5910 传真: 45-4485-2829 芬兰 Finland - Espoo 电话: 358-9-4520-820 法国 France - Paris 电话: 33-1-69-53-63-20 传真: 33-1-69-30-90-79 德国 Germany - Garching 电话: 49-8931-9700 德国 Germany - Haan 电话: 49-2129-3766400 德国 Germany - Heilbronn 电话: 49-7131-72400 德国 Germany - Karlsruhe 电话: 49-721-625370 德国 Germany - Munich 电话: 49-89-627-144-0 传真: 49-89-627-144-44 德国 Germany - Rosenheim 电话: 49-8031-354-560 以色列 Israel - Ra'anana 电话: 972-9-744-7705 意大利 Italy - Milan 电话: 39-0331-742611 传真: 39-0331-466781 意大利 Italy - Padova 电话: 39-049-7625286 荷兰 Netherlands - Drunen 电话: 31-416-690399 传真: 31-416-690340 挪威 Norway - Trondheim 电话: 47-72884388 波兰 Poland - Warsaw 电话: 48-22-3325737 罗马尼亚 Romania - Bucharest 电话: 40-21-407-87-50 西班牙 Spain - Madrid 电话: 34-91-708-08-90 传真: 34-91-708-08-91 瑞典 Sweden - Gothenberg 电话: 46-31-704-60-40 瑞典 Sweden - Stockholm 电话: 46-8-5090-4654 英国 UK - Wokingham 电话: 44-118-921-5800 传真: 44-118-921-5820
亚特兰大 德卢斯, 佐治亚州 电话: 678-957-9614 传真: 678-957-1455 奥斯汀 Austin, TX 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 休斯敦 Houston, TX 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 Raleigh, NC 电话: 919-844-7510 纽约 New York, NY 电话: 631-435-6000 圣何塞 San Jose, CA 电话: 408-735-9110 电话: 408-436-4270 加拿大多伦多 Toronto 电话: 905-695-1980 传真: 905-695-2078			