

# 基于FRDM-KE04Z和Tower Board的 BLDC电机正弦波控制

作者: Liu Zhen

## 1 简介

本应用手册介绍了基于FRDM\_KE04Z控制板的带霍尔传感器三相无刷直流电机的正弦波控制。该设计充分利用了飞思卡尔KE04Z的外设资源,在电机控制上可以实现良好的控制效果。

本应用是基于霍尔传感器检测转子位置而设计的转速单闭环控制系统。该应用可以作为飞思卡尔最新Kinetis微控制器和三相低压电机驱动板的带霍尔BLDC电机控制的一个范例。这里也介绍了一种使用Kinetis实现BLDC电机正弦波的控制技术。

电机控制理论、系统架构、硬件设计、软件实现都在本应用中得到了一一介绍。此外还包含了可调试工具FreeMASTER的使用以及系统的搭建和演示。

## Contents

1	简介	1
2	KE04 系列的优势和特性	2
3	BLDC 电机控制理论	2
4	系统设计理念	4
5	软件设计	15
6	Demo 的搭建和使用	23
7	图形结果	27
8	参考文献	29

## 2 KE04系列的优势和特性

系列中可用的片上模块包括下列功能：

- Cortex™-M0+内核
- 在电压范围2.7–5.5 V内高达24MHz的CPU主频
- 8 KB FLASH, 1 KB RAM
- 芯片尺寸小，共有24个引脚，价格低
- 具有12通道12位数据的AD转换器
- 模拟比较器
- 2个通道的PIT模块，8个通道的FlexTimer模块
- 1个8位数据的SPI模块，1个SCI/UART模块以及1个I2C模块
- 高达8个KBI中断引脚

## 3 BLDC电机控制理论

无刷直流电机（BLDC电机）是由具有类似于三相感应电机的定子和安装了永磁体的转子组成的。电机上没有机械换相器，换相是在程序中实现的，因此也被称作电子换相电机。转子上的永磁体通常是由永磁材料制成的。BLDC电机的典型横截面如图1所示。三相定子绕组嵌入在沟槽中(分布绕组)或者绕到磁极（集中绕组）上。因为气隙磁场是由永磁体产生的，磁场大小是恒定的。

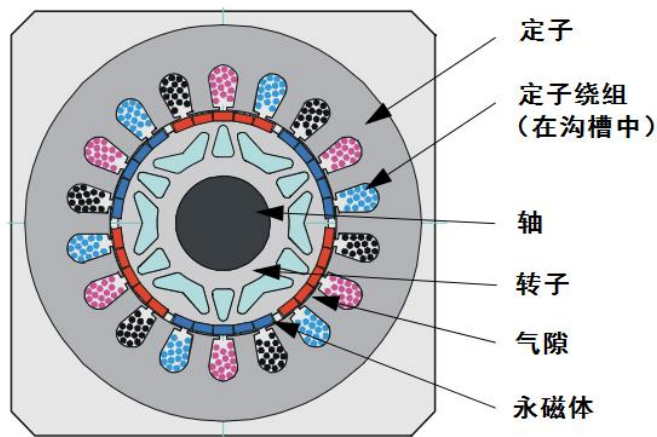


图1 BLDC电机横截面

在本应用中霍尔传感器检测转子位置，在绕组中通以如图2所示的波形电压来驱动BLDC电机。该波形是通过在空间矢量调制技术（SVPWM）中插入零矢量 $O_{000}$ 而获得的，如图3所示。

该控制技术是以三相逆变器为控制对象的，三相定子绕组上施加如图2所示电压，旋转磁场是圆形的，转矩脉动较方波小。

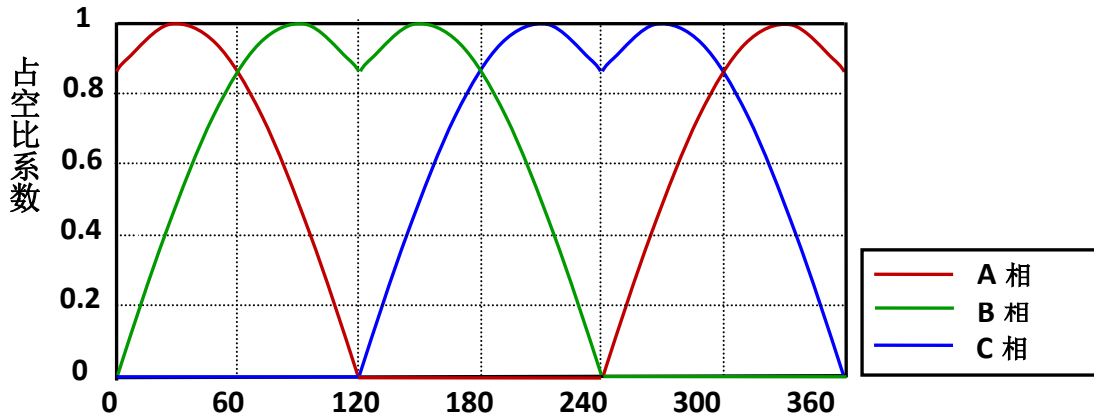


图2 三相BLDC电机定子绕组电压波形

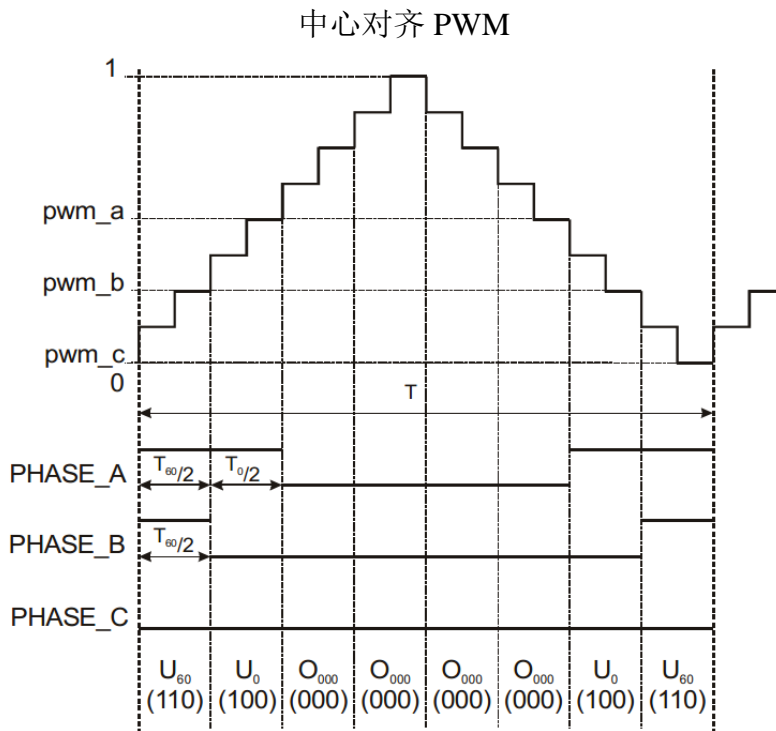


图3 加入了零矢量O000的中心对齐SVPWM波形

图4为本应用设计中电机驱动功率电路，六个功率开关管工作在互补模式下。

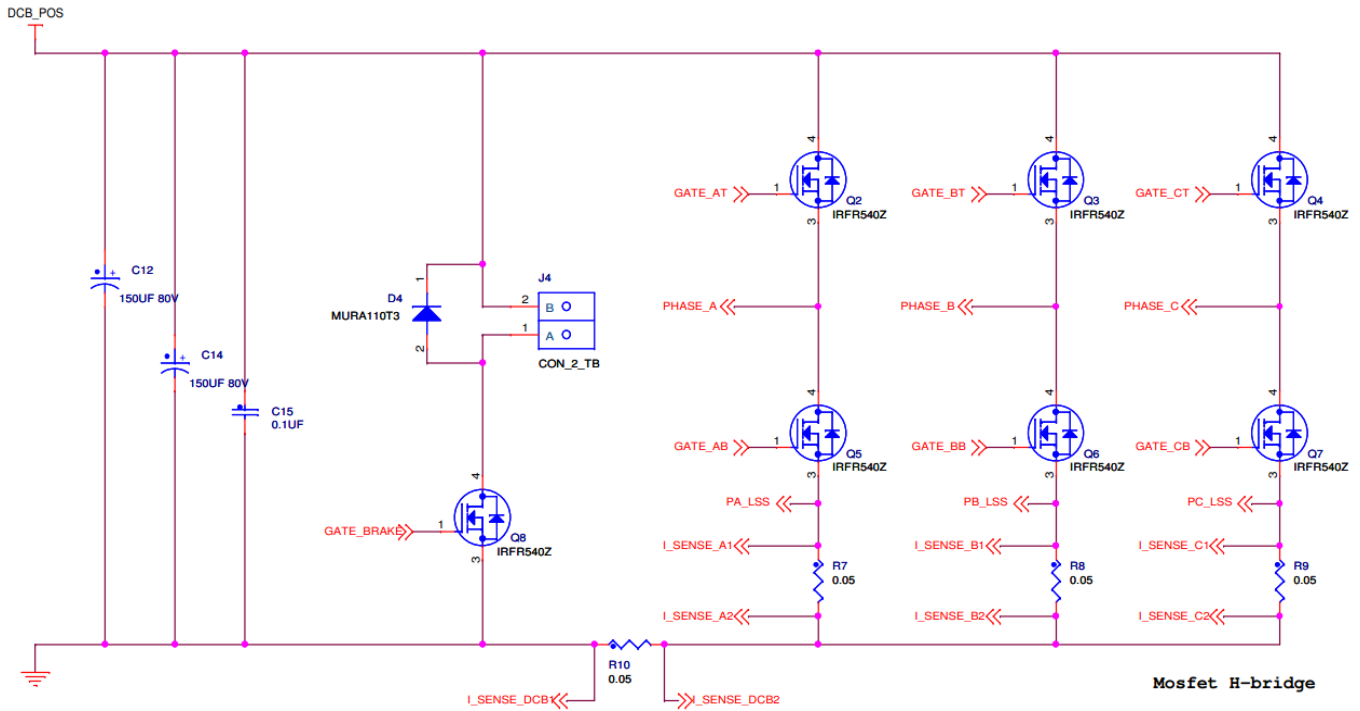


图 4 三相功率电路

本应用手册中将不再详细介绍系统硬件，更多关于硬件的信息请参考应用手册AN4796。

## 4 系统设计理念

### 4.1 系统说明

本应用设计满足以下控制性能：

- 相电流波形为正弦波，转矩脉动小，噪音小
- 相比较于通用的SPWM控制技术，本应用中直流母线电压利用率提高了15.47%
- 相对于七段式的SVPWM技术，三相功率电路的开关损耗减小了1/3
- 控制技术包含：
  - 转速单闭环控制
  - 过压、欠压、过流等故障保护
  - 转子不需要对齐，可从任何位置平稳启动
  - 开环方波启动
  - 支持正向、反向两个方向旋转
  - 转速范围：300 rpm –3000 rpm

— FreeMASTER软件控制界面（电机启动/停止，转速设置）

## 4.2 系统架构和转子位置检测

如下图 5 所示，系统包含以下硬件结构：

- FRDM – KE04Z核心板
- TWR – MC – FRDMKE02Z转接板
- TWR – MC – LV3PH驱动板
- TWR – SER串口转接板
- USB 转串口线 HL-340
- Tower Elevator
- 带霍尔传感器的 BLDC（分布绕组）电机
- 供电电源24V DC，3.75A

FRDM – KE04Z中央控制板和TWR – MC – FRDMKE02Z转接板的配置信息将在第六节中详细介绍。

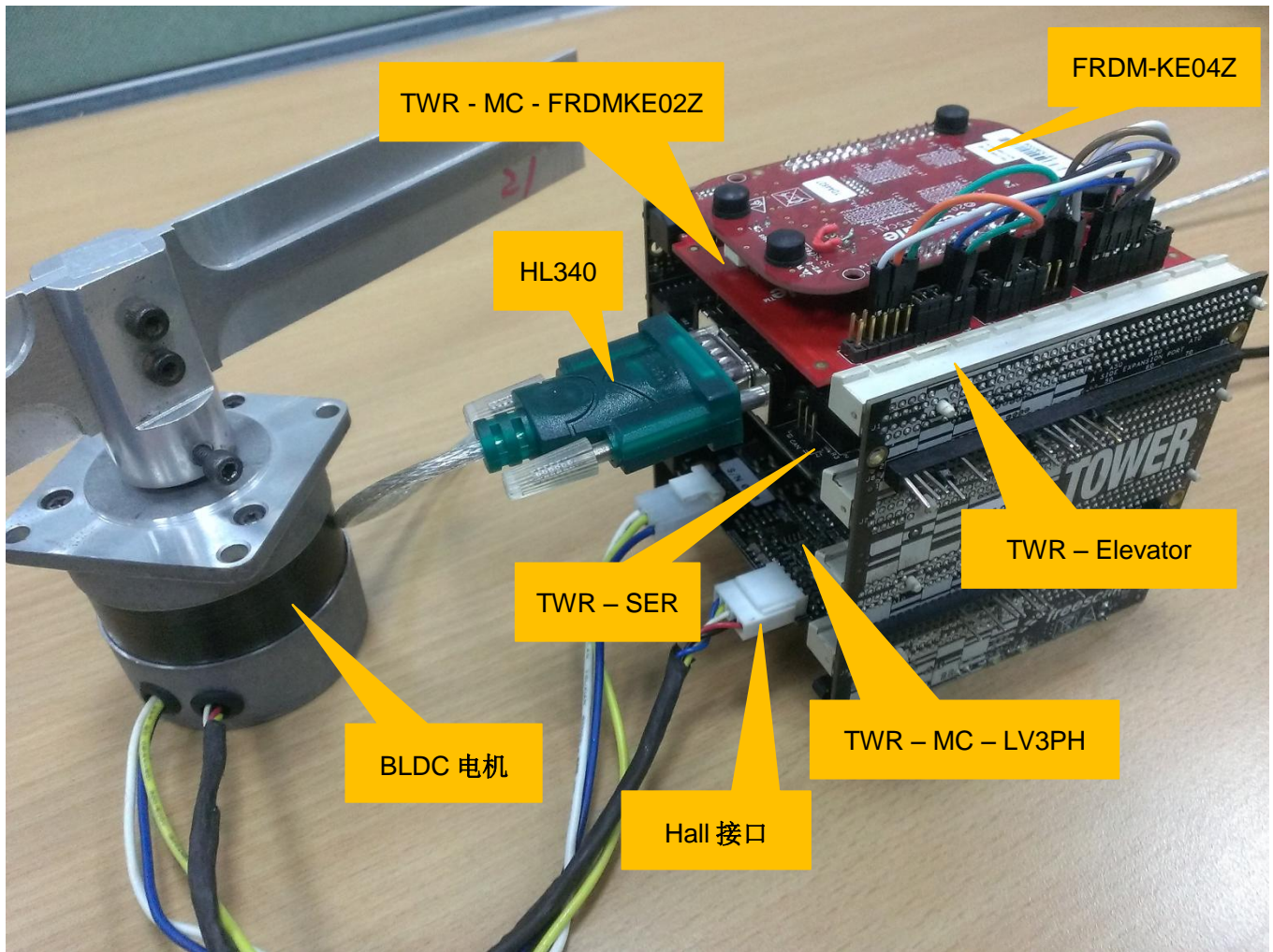


图 5 MKE04Z8VFK4 Demo

系统框图如图 6 所示。系统是转速单闭环控制，根据反馈的霍尔信号和查表来确定输出到三相逆变桥的PWM占空比。电机的期望转速通过FreeMASTER设置，同时检测过压和欠压等故障。

系统的后台程序中，只有FreeMASTER轮询函数的调用。在10 ms的定时中断（RTC\_ISR）内，功能包括状态机的运行，直流母线电压AD转换的轮询，过压、欠压等故障的检测和闭环转速PI调节等。在125us的定时中断（PIT1\_ISR）内，功能包含delta角度值的计算、三相相电压占空比的计算和占空比的寄存器赋值。Hall输入捕捉中断用到FTM0和KBI模块，其功能包括开环下双极性PWM的换相，闭环下delta角度值的清零等。SPI模块用于驱动器MC33937的配置。SCI模块用于FreeMASTER的串口通信。

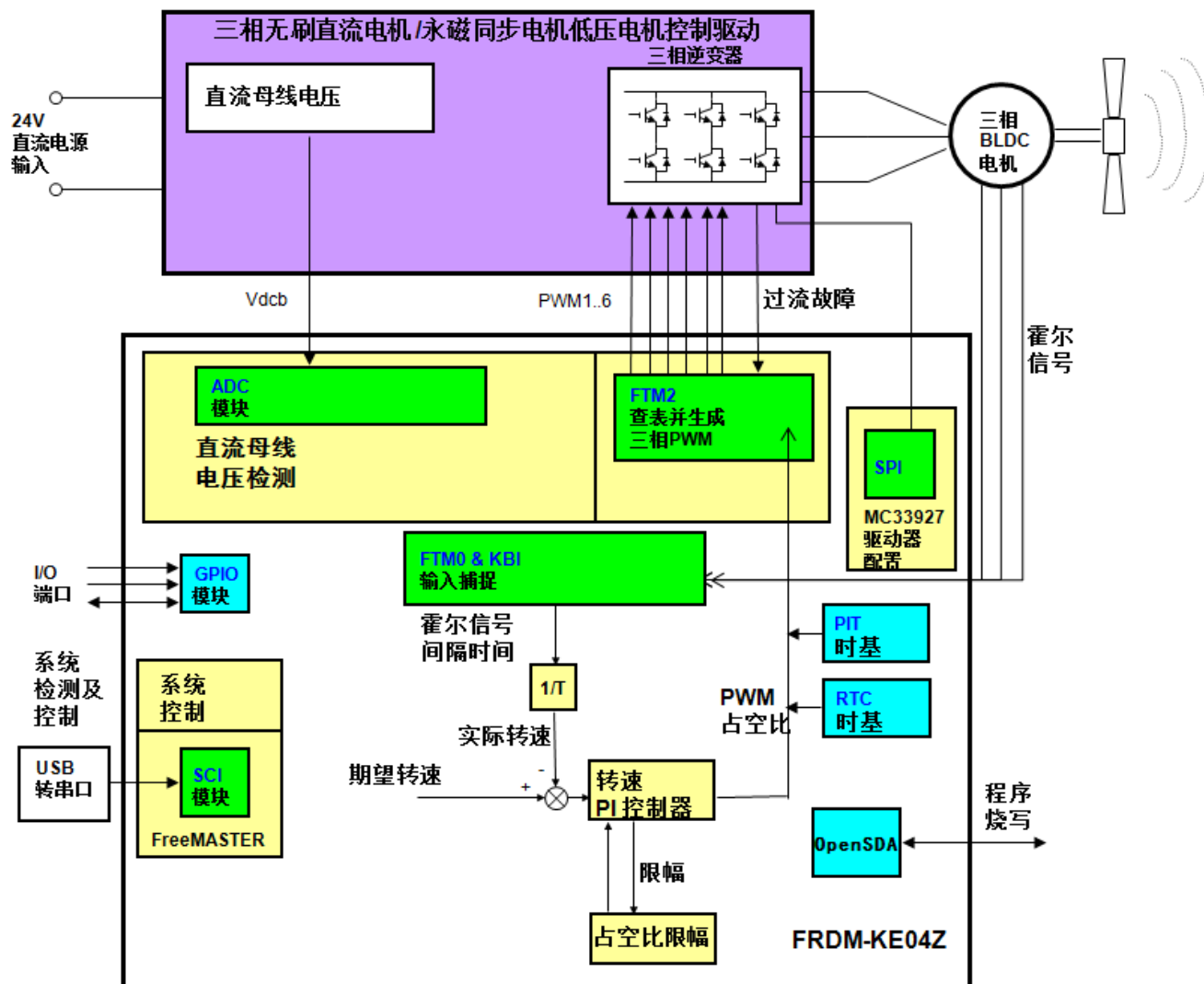


图 6 控制系统架构

## 4.3 BLDC正弦控制

### 4.3.1 霍尔信号角度值

图 7为本应用中霍尔信号相对应的定子磁链角度值。红色圆点为霍尔信号的变化点，蓝色圆点为霍尔传感器的安装位置。



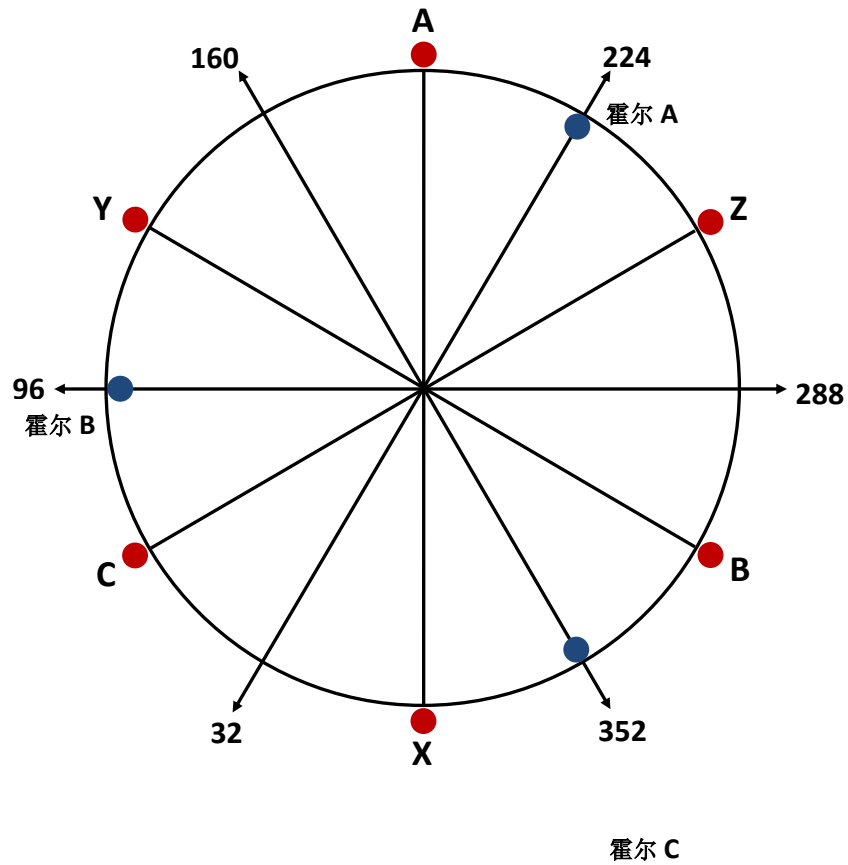


图 7 正弦波控制霍尔信号对应角度关系图

为电机高效运行，励磁磁链和转矩磁链之间的角度需要尽可能的保持在90度。定子磁链角度根据霍尔传感器检测到的转子位置和表 1 中所示的对应角度信息进行调节。霍尔传感器在转子旋转一个电周期产生六个有效信号，每个有效信号对应于定子磁链的一个电角度。更多关于定子磁链角度的说明将会在下节马鞍波形表中介绍。

表 1 霍尔信号对应转子角度值

顺时针	霍尔信号	110	100	101	001	011	010
	角度	32	96	160	224	288	352
逆时针	霍尔信号	110	010	011	001	101	100
	角度	288	224	160	96	32	352



### 4.3.2 马鞍波形表

如图 2 所示波形，本应用中用到表SinusoidalWaveTable[384]。为了角度计算的方便，波形周期设为384。

图 8 所示的波形中，有三分之一周期的数据为零，这就很容易理解相对于七段式的SVPWM三相逆变器的开关损耗减小了三分之一。

```
static unsigned char SinusoidalWaveTable[384]={
    127,131,135,138,142,145,149,152,155,159,162,165,168,171,174,177,180,183,186,189,192,194,197,200,
    202,205,207,210, 212,214,217,219,221,223,225,227,229,231,232,234,236,237,239,240,242,243,244,245,
    247,248,249,250,250,251,252,253,253,254,254,255,255,255,255,255,255,255,255,255,255,254,254,
    253,253,252,252,251,250,249,248,247,246,245,244,242,241,240,238,237,235,233,232,230,228,226,224,
    222,222,225,226,228,230,232,234,235,237,238,240,241,243,244,245,246,247,248,249,250,251,252,252,
    253,254,254,254,255,255,255,255,255,255,255,255,255,254,254,254,253,252,252,251,250,249,248,
    247,246,245,244,243,241,240,238,237,235,234,232,230,228,227,225,223,220,218,216,214,212,209,207,
    204,202,199,197,194,191,188,185,183,180,177,174,171,167,164,161,158,154,151,148,144,141,137,134,
    130,127,123,119,116,112,108,104,101, 97, 93, 89, 85, 81, 77, 73, 69, 65, 61, 57, 53, 49, 45, 41,
    36, 32, 28, 24, 20, 16, 12, 7, 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, 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, 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, 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, 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, 2, 6, 10, 15, 19, 23, 27, 31,
    35, 39, 44, 48, 52, 56, 60, 64, 68, 72, 76, 80, 84, 88, 92, 96, 99,103,107,111,115,118,121,124
}
```

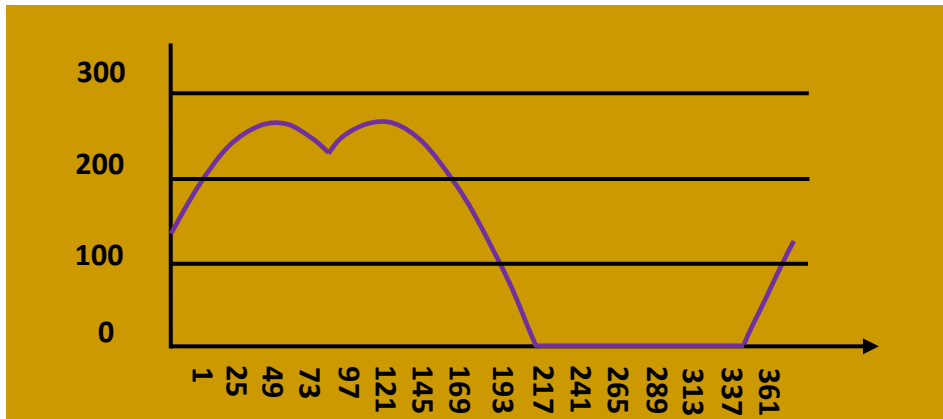


图 8 软件中使用的马鞍波形

图 9 为本应用中霍尔信号和三相相电压关系示意图，考虑到所创建的正弦波形表SinusoidalWaveTable[384]，在顺时针旋转时霍尔信号的六个状态将分别对应于 $32^\circ$ 、 $96^\circ$ 、 $160^\circ$ 、 $224^\circ$ 、 $288^\circ$ 和 $352^\circ$ 角度值，上文表 1 中已介绍到。

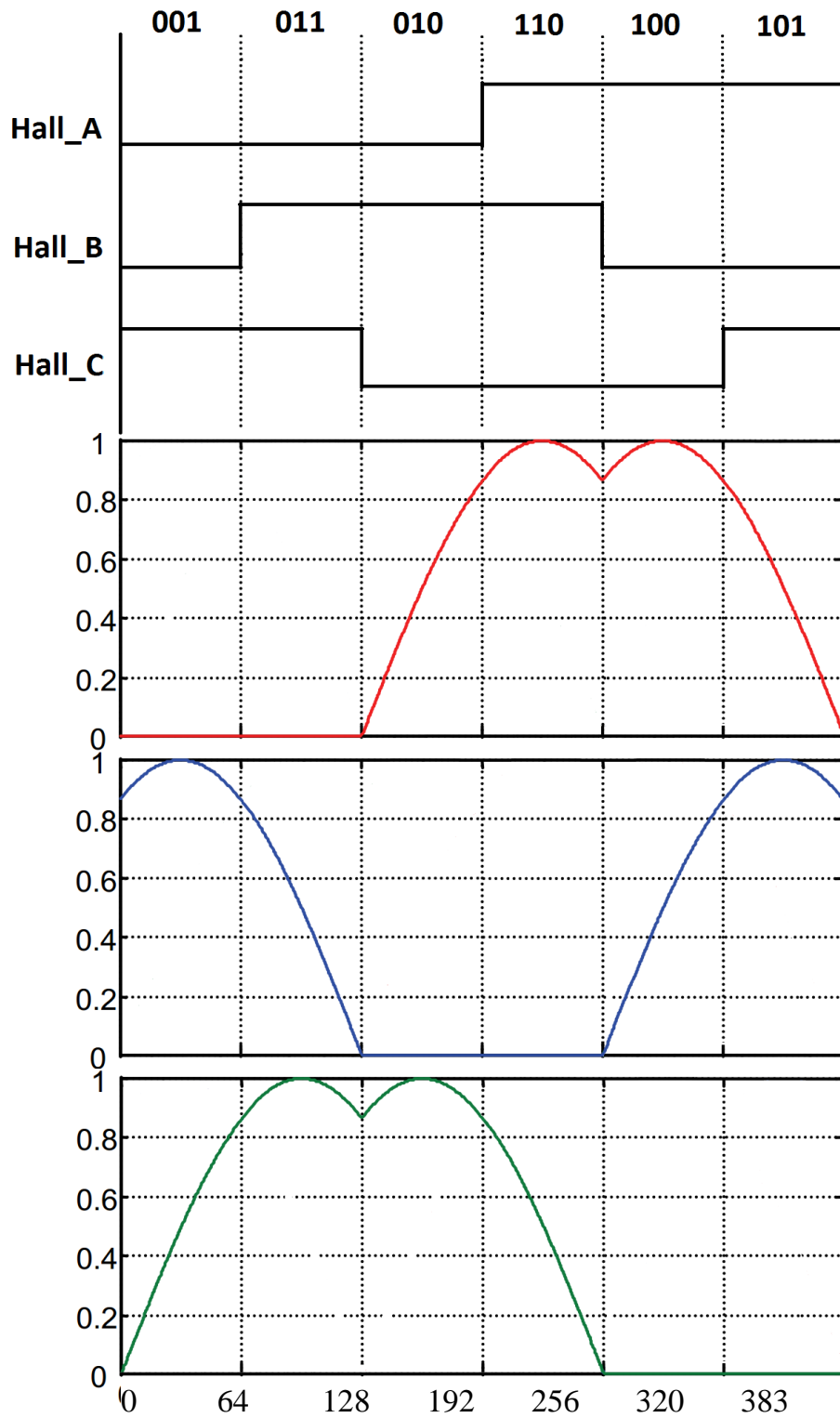


图 9 相电压波形与霍尔信号状态关系图(CW)

### 4.3.3 转速计算

转速PI调节器及下面介绍的delta角度计算都需要电机的实时转速值。在带hall的电机驱动应用中，通常用连续两次霍尔信号之间的时间作为转速计算的参考。Hall传感器在一个电周期内只有6个有效信号，连续六次时间和是一个电周期，电周期和机械周期之间是一个极对数的倍数关系。

在转速稳定的情况下，由于霍尔传感器在定子安装位置上存在一定的误差，连续6次的时间肯定有轻微的不同。为了克服这样的机械误差，在本设计中转速在Hall\_C的中断程序中进行计算，霍尔C的两次中断之间时间间隔为电周期的一半。本应用中所用电机极对数是2，定标转速为5000 rpm，转速计算方法为T法测速。

转速计算公式：

$$speed = \frac{60 \times \frac{f}{128} \times 32768}{(motorPairs \times 6) / 3 \times t \times \max Rpm} = \frac{60 \times \frac{f}{128} \times 32768}{(2 \times 6) / 3 \times t \times 5000}$$

公式 1

这里

$f$  [Hz]: CPU的时钟频率

$t$ : 霍尔C两个连续边沿信号的时间间隔

### 4.3.4 delta角度计算

在正弦波控制模式下，用来控制电机的角度值必须是连续的，如4.3.1所提，通过霍尔信号仅能得到6个定子磁链角度信息，其他定子磁链角度无法直接获得。在本应用中，两个霍尔信号之间的间隔时间内，利用转速可以计算delta角度值， $deltaAngle = speed \times \Delta t$ ，那么定子磁链的角度值可由当前霍尔状态对应的定子磁链角度值加或减delta角度值获得， $mtrAngle \pm = deltaAngle$ ，delta角度值计算原理如图10所示，delta角度值计算公式如公式2所示：

$$deltaAngle = \frac{speed \times \max Rpm \times tableDotsNo \times motorPairs}{32768 \times 60 \times PITfrequency}$$

$$= \frac{speed \times 5000 \times 384 \times 2}{32768 \times 60 \times 8 \times 1000} = \frac{speed}{4096} = speed \gg 12$$

公式 2

这里：

电周期为384，delta角度值计算避免了除法，节省了CPU资源。

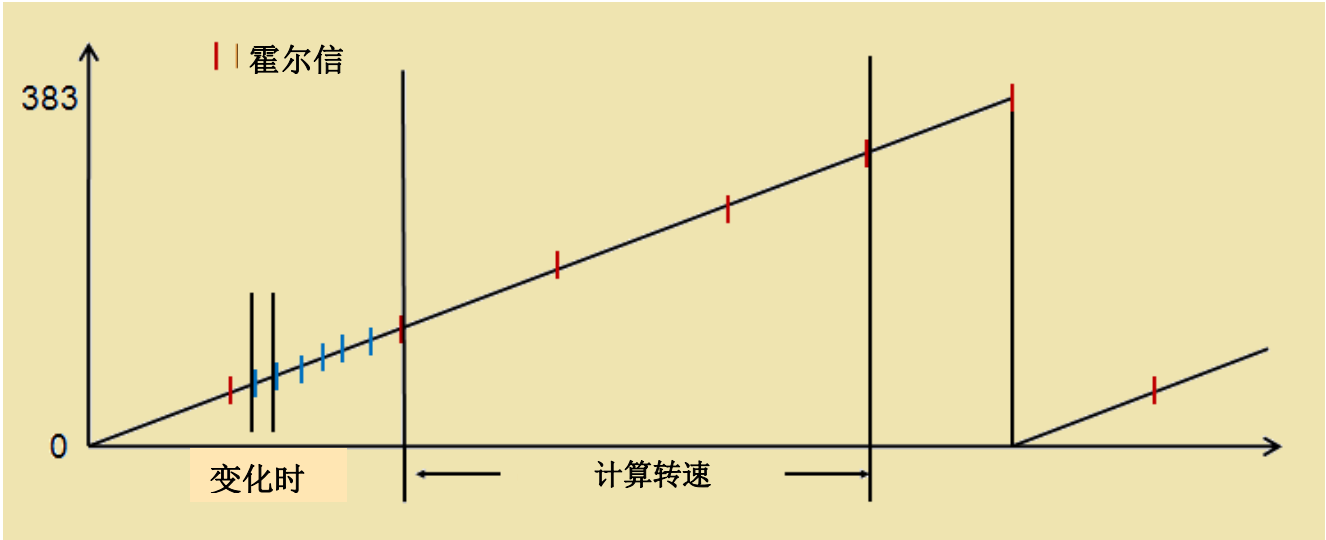


图 10 转子变化角度计算

### 4.3.5 电机启动

在电机启动之前，转子的初始位置通过读取当前霍尔信号值来获得。而此时电机转速为零。因此在电机启动阶段很难直接使用正弦波控制方式启动。

在本应用中，采用六步方波控制技术来启动电机，PWM调制方式为双极性，占空比为20%。图 11 所示。表 2 列出了电机顺时针旋转时的相关寄存器配置信息。

表 2 CW方向

霍尔信号	霍尔传感器			换相	OUTMASK	INVCTRL
	A	B	C			
6	1	1	0	A+, B-	0x0030	0x0002
4	1	0	0	A+, C-	0x000C	0x0004
5	1	0	1	B+, C-	0x0003	0x0004
1	0	0	1	B+, A-	0x0030	0x0001
3	0	1	1	C+, A-	0x000C	0x0001
2	0	1	0	C+, B-	0x0003	0x0002

在电机成功启动且转速近似稳定后由方波切换到正弦波，在本应用中，电机在启动后运行一个机械周期进行切换。在由方波控制切换到正弦波控制时，占空比需要做微调，因为这两种控制在星

形连接下产生的磁链大小不同。在切换点处，正弦波控制模式的占空比应该设为方波控制下占空比的  $2/\sqrt{3}$  倍。

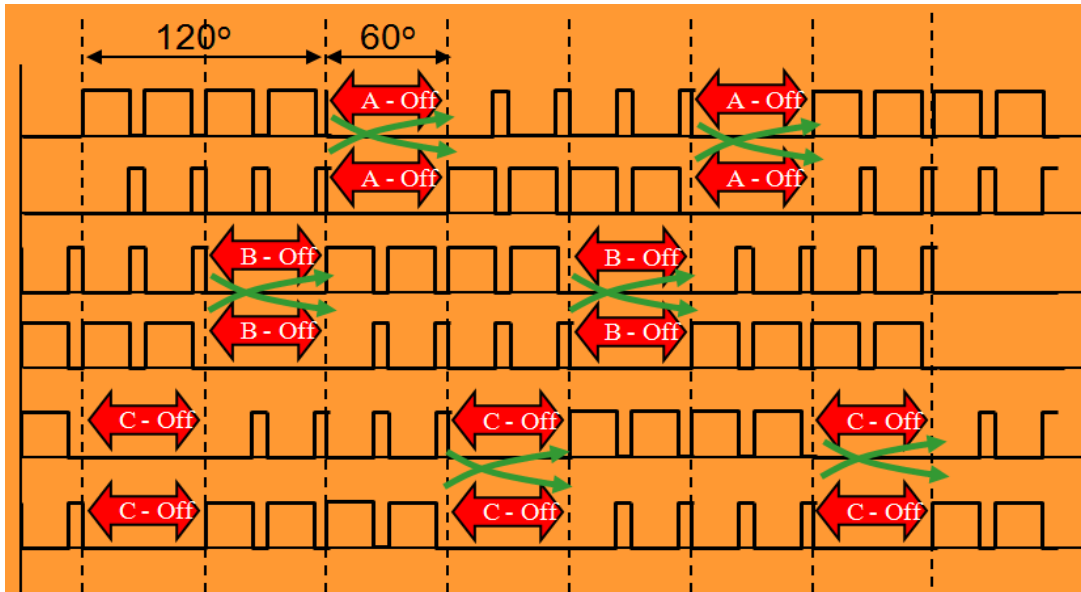


图 11 双极性PWM的六步方波控制技术

#### 4.3.6 超前角计算

如上文所述，霍尔传感器的输出反映转子位置信息的变化，根据霍尔信号生成的相电压与反电动势同相位。由于电机是感性负载，相电流滞后于相电压。因此，转子磁链和定子磁链之间的实际角度并不是90度，转速越高，误差越大。因此需要调整相电压相位来使转子磁链和定子磁链之间尽可能的保持90度，这个调整的角度值称为超前角，见图 12。

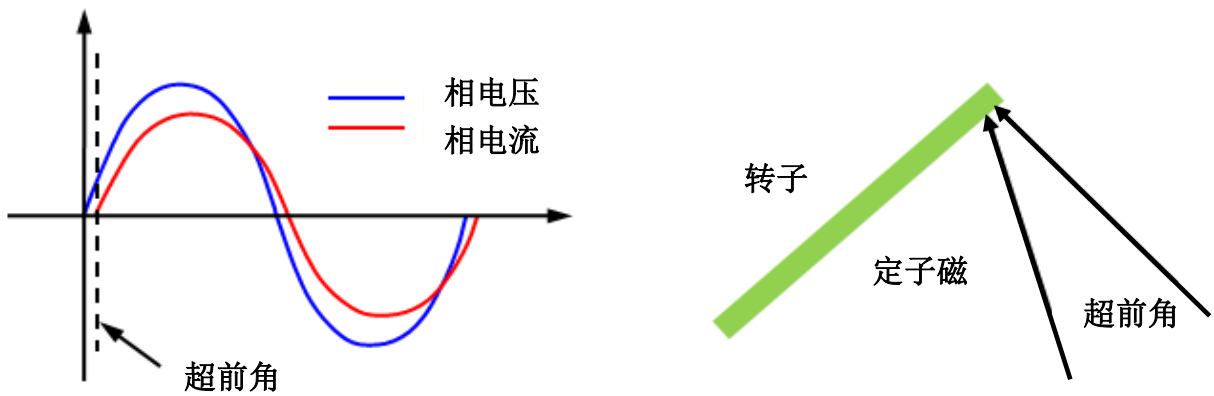


图 12 相电压与相电流之间的超前角

### 4.3.7 电机

本应用中永磁电机的定子绕组是分布式的，电机的规格参数如表 3 所示。

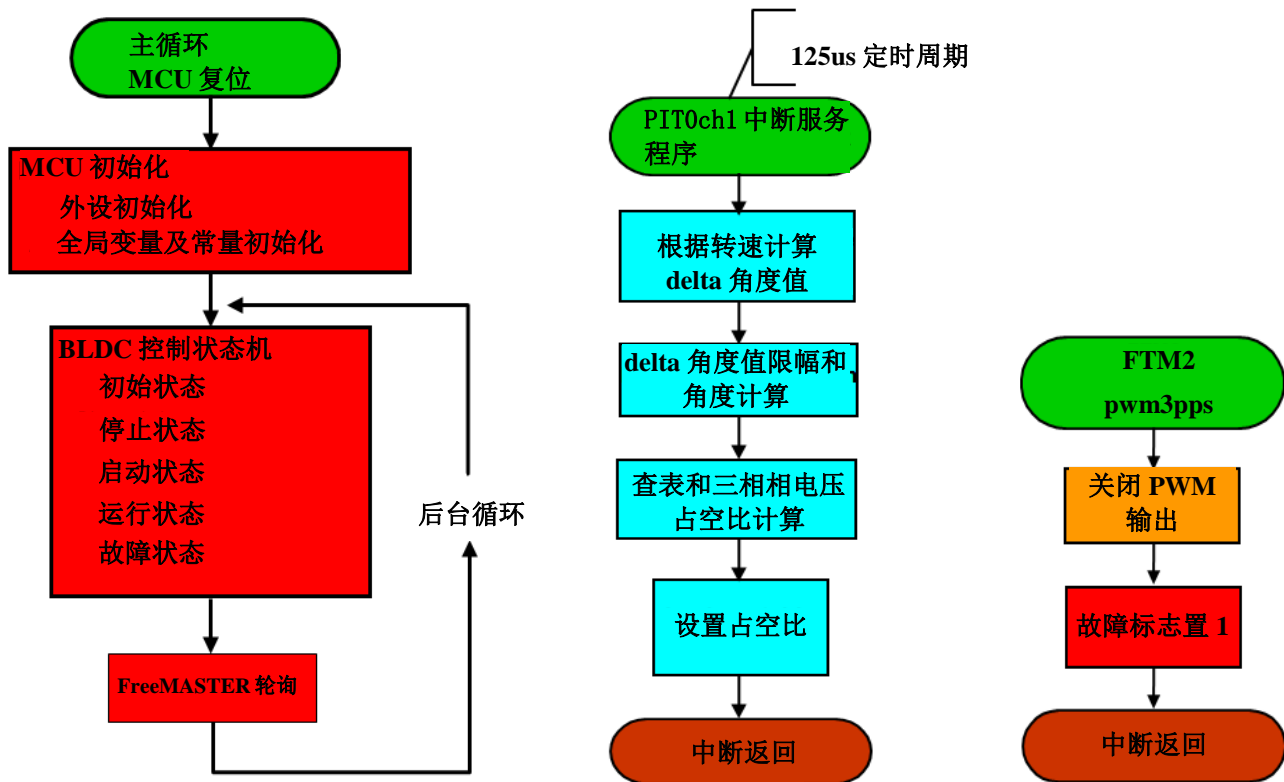
表 3 电机相关参数

特性	符号	最小值	典型值	最大值	单位
参考绕组电压	$V_t$	Jm	—	—	V
转速 @ $V_t$		—	—	—	rpm
转矩常数	$K_t$	—	—	—	Nm/A
电压常数	$K_e$	—	—	—	V/RPM
终端电阻	$R_t$	—	1.46	—	$\Omega$
绕组电感	$L_d$	—	—	—	mH
	$L_q$	—	—	—	mH
持续电流	$I_{CS}$	—	—	—	A
极对数	—	—	2	—	—
额定电流	—	—	—	—	$^{\circ}\text{C}$

## 5 软件设计

### 5.1 流程图

软件程序流程图如图 13所示。





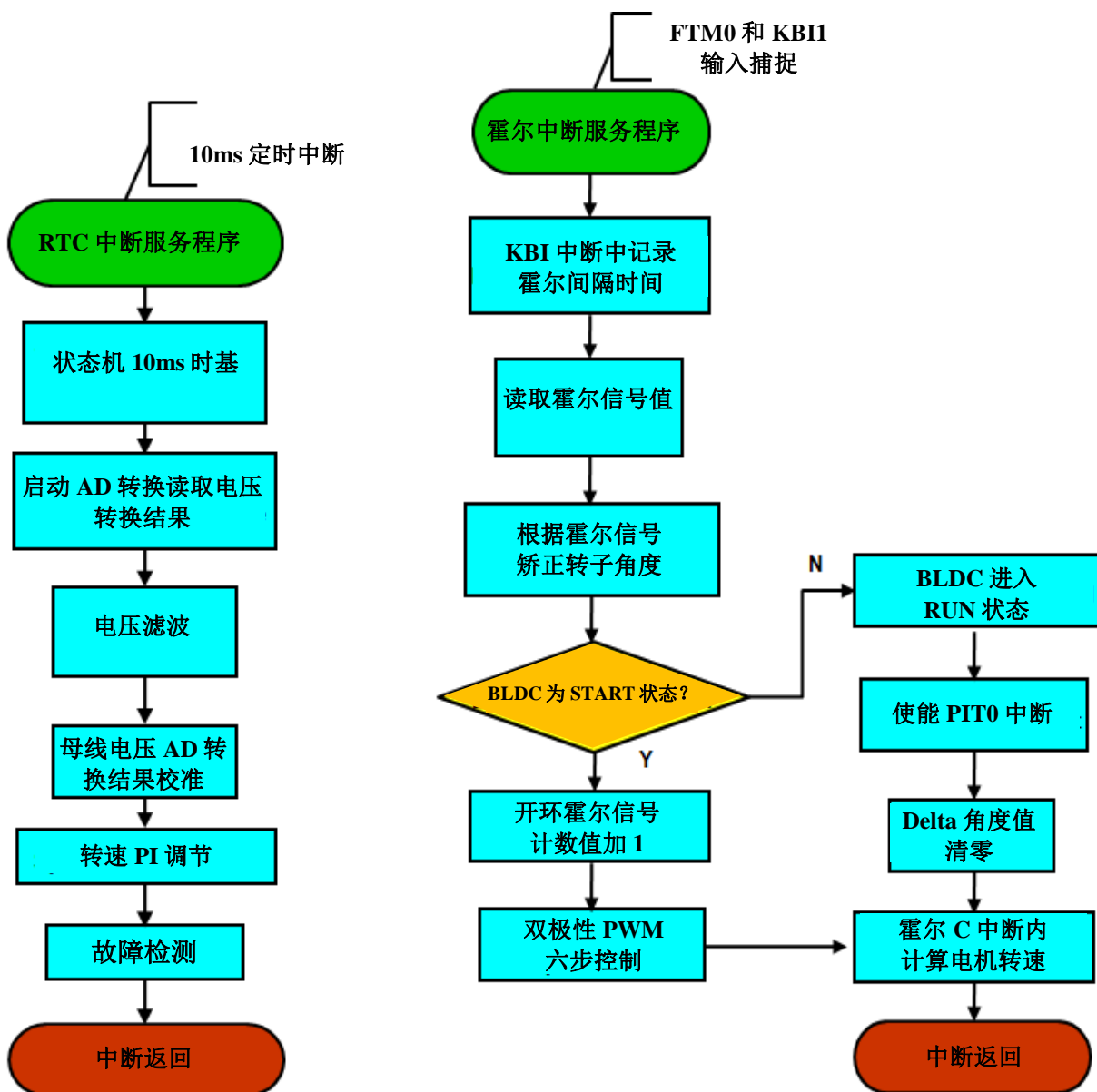


图 13 软件主要程序流程图

## 5.2 状态机

状态机如图 14 所示。本应用中电机的启动和停止命令是由 FreeMASTER 中所设置的转速值为非零和零所决定的，同样电机的期望转速也是由 FreeMASTER 软件设置的。任何时候在按下复位按键或转速设置为零时，电机都将会立刻停转。所有处理程序的执行都是由该状态机所控制的。

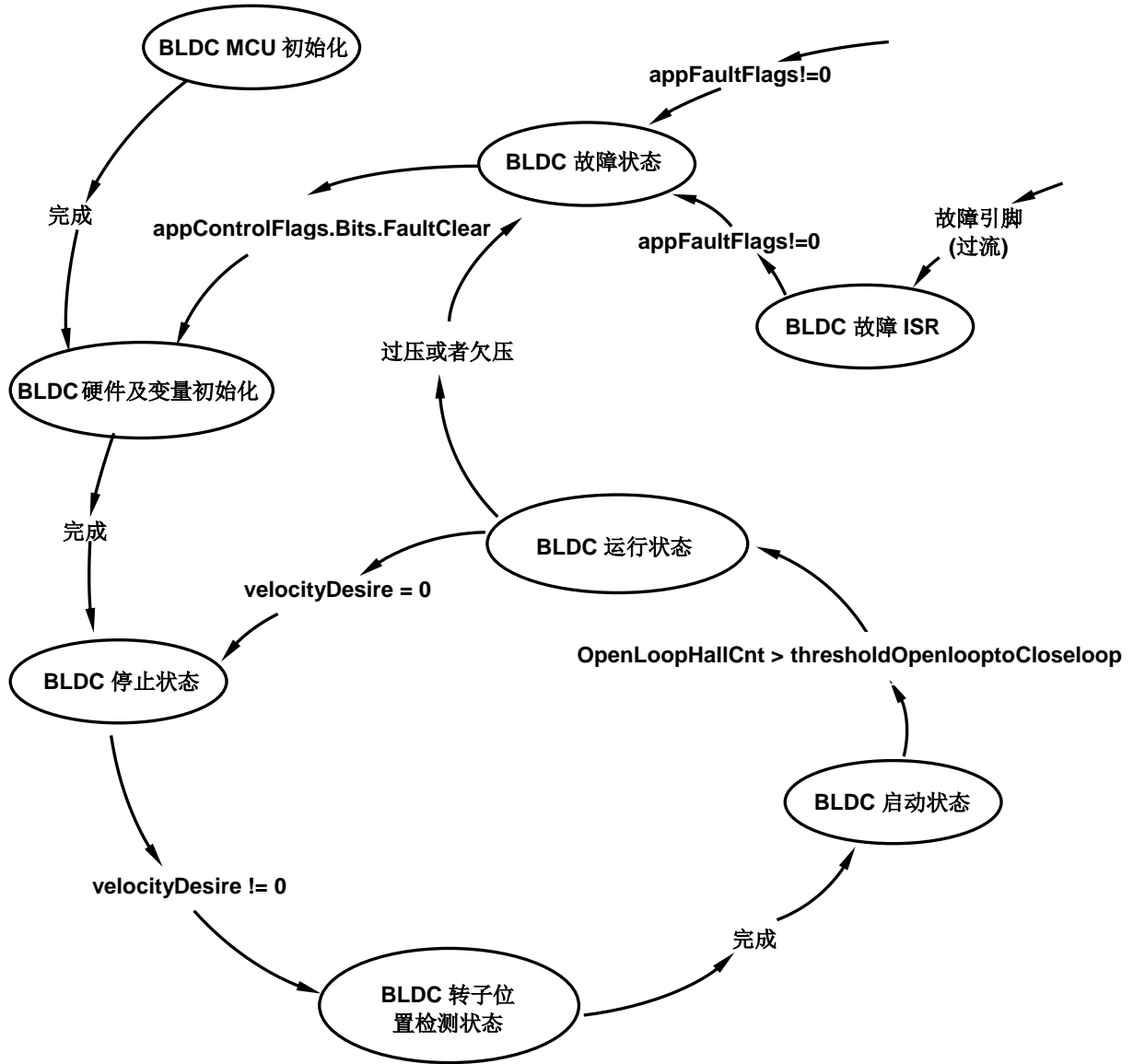


图 14 BLDC电机控制状态框图

### 5.3 霍尔输入捕捉中断服务程序

本应用中，不管是启动状态下的六步方波控制还是运行状态下的正弦波控制，霍尔传感器都起着重要的作用。电机状态从START到RUN的切换时刻取决于变量openLoopHallCnt。在软件中该阈值设为12，代表一个机械周期。六步方波控制技术利用FTM2模块的INVCTRL和OUTMASK寄存器来启动电机，正转代码如下：

```

openLoopHallCnt++;
    PWMState = bldcCommutationTableComp[motorPosition];
    if (PWMState.swap & 1)
    { FTM2->INVCTRL |= FTM_INVCTRL_INV0EN_MASK; }
    else
    { FTM2->INVCTRL &= ~FTM_INVCTRL_INV0EN_MASK; }
    if (PWMState.swap & 2)
    { FTM2->INVCTRL |= FTM_INVCTRL_INV1EN_MASK; }
    else
    { FTM2->INVCTRL &= ~FTM_INVCTRL_INV1EN_MASK; }
    if (PWMState.swap & 4)
    { FTM2->INVCTRL |= FTM_INVCTRL_INV2EN_MASK; }
    else
    { FTM2->INVCTRL &= ~FTM_INVCTRL_INV2EN_MASK; }
    FTM2->OUTMASK = PWMState.mask;           // Update FTM registers
    FTM2->SYNC |= FTM_SYNC_REINIT_MASK;     // Update FTM registers

```

在每个霍尔中断服务程序中，转子的位置需要通过直接读取三个霍尔信号实时更新。下面代码为霍尔C中断服务程序，同理霍尔A和霍尔B中断服务程序类似。

```

hallC = GPIOA->PDIR;
hallC = ((hallC >>3) >> 8) >> 8;//PTC3
hallC = hallC & 1;// status of hallC
bldcIntHallFlags.Byte = (char)( hallA | hallB | hallC);
motorPosition = (bldcIntHallFlags.Byte & 0x07);

```

电机的转速值也是在中断服务程序中计算的，但只在霍尔C的中断程序中计算。FTM0模块用来保存计算转速所需要的时间信息。

```

CalTimePrevious = CalTime;
CalTime = (tU16)FTM0->CNT;
TimeInterval = CalTime - CalTimePrevious; //calculate half
CalculationSpeed();                       electronic cycle time

```

这里需要说明两点，首先由于单片机外设资源和引脚数量的限制，KBI模块也用来捕捉其中一路霍尔（HallC）的输入信号，由于KE04中KBI模块的特性，在HallC中断服务程序中需要KBI1->ES取反才能正确检测到下一个边沿信号。另外，电机状态START切换到RUN时，上面提到的在六步方波启动中使用到的INVCTRL和OUTMASK寄存器清零，保证电机在正弦波控制模式下正确运行。

## 5.4 PIT – delta角度估算中断服务程序

本应用中转子的delta角度在PITch1的125us定时中断中计算和更新的。根据4.2.3节的介绍，转子角度和电机的转速成比例关系。变量**mtrAngle**用来表示磁链的实时角度。当顺时针旋转时，代码如下：

```

deltaAngle = F32Add(deltaAngle, F32Abs(velocityAct));
if (deltaAngle >= DELTANGLE_MAX)    //limit deltaAngle range into
    {
        deltaAngle = DELTANGLE_MAX;
    }
mtrAngle = HallTableCW[motorPosition];
mtrAngle += (tU16)((deltaAngle) >> 12);
mtrAngle += (tU16)advanceAngle;
if (mtrAngle >= 384)
    {
        mtrAngle -= 384;
    }

```

当逆时针旋转时，代码如下：

```

deltaAngle = F32Add(deltaAngle, F32Abs(velocityAct));
if (deltaAngle >= DELTANGLE_MAX)    //limit deltaAngle range into
    {
        deltaAngle = DELTANGLE_MAX;
    }
mtrAngle = HallTableCCW[motorPosition];
mtrAngle -= (tU16)((deltaAngle) >> 12);
mtrAngle += (tU16)advanceAngle;
if (mtrAngle >= 384)
    {
        mtrAngle = 384;
    }

```

在计算得到磁链角度**mtrAngle**之后，A、B、C三相占空比计算代码如下：

```

dutyCycleU16A = (Frac16)(((Frac16)dutyCycleU16 *
(Frac16)SinusoidalWaveTable[mtrAngle]) >> 8);
if (mtrAngle < 128)
    {
        dutyCycleU16B = (Frac16)(((Frac16)dutyCycleU16 *
(Frac16)SinusoidalWaveTable[mtrAngle + 256]) >> 8);
    }
else
    {

```

```

        dutyCycleU16B = (Frac16) (((Frac16) dutyCycleU16 *
(Frac16) SinusoidalWaveTable[mtrAngle - 128]) >> 8);
    }
    if (mtrAngle >= 256)
    {
        dutyCycleU16C = (Frac16) (((Frac16) dutyCycleU16 *
(Frac16) SinusoidalWaveTable[mtrAngle -256]) >> 8);
    }
    else
    {
        dutyCycleU16C = (Frac16) (((Frac16) dutyCycleU16 *
(Frac16) SinusoidalWaveTable[mtrAngle + 128]) >> 8);
    }

```

## 5.5 配置

### 5.5.1 MOSFET驱动器配置

在使用MC33937前，该预驱动器通过SPI模块进行配置。该预驱动器的配置需如下两个文件：

- H文件spi\_comm.h：包含了MC33937驱动器所定义的配置和状态常数。
- C文件spi\_comm.c：包含了SPI通信函数以及进行MC33937驱动器配置的调用函数。

### 5.5.2 PWM生成以及定时器

KE04Z64VQH2微控制器的FTM模块拥有两个子模块。FTM2六路通道产生PWM信号驱动MC33937，FTM0两路通道配置为输入捕捉功能用来捕捉Hall信号。PWM信号通过预驱动器功率放大后驱动三相逆变桥的MOSFET。FTM2模块初始化配置如下：

#### FTM2

- 系统时钟源  
`FTM2->SC |= FTM_SC_CLKS(1);`
- 产生频率为16kHz的PWM信号
- 周期寄存器Modulo设为1500，PWM的分辨率为0.067%  
`FTM2->MOD = PWM_MODULO; (#define PWM_MODULO 1500)`
- PWM采用组合并互补模式，插入 $1\mu\text{s}$ 的死区时间  
`FTM2->COMBINE = FTM_COMBINE_FAULTENO_MASK  
| FTM_COMBINE_SYNCENO_MASK | FTM_COMBINE_DTENO_MASK  
| FTM_COMBINE_COMP0_MASK | FTM_COMBINE_COMBINE0_MASK`

```

        | FTM_COMBINE_FAULTEN1_MASK
        | FTM_COMBINE_SYNCEN1_MASK | FTM_COMBINE_DTEN1_MASK
        | FTM_COMBINE_COMP1_MASK | FTM_COMBINE_COMBINE1_MASK
        | FTM_COMBINE_FAULTEN2_MASK
    | FTM_COMBINE_SYNCEN2_MASK | FTM_COMBINE_DTEN2_MASK
        | FTM_COMBINE_COMP2_MASK | FTM_COMBINE_COMBINE2_MASK;

```

```

FTM2->DEADTIME = FTM_PWM_DEAD_TIME;
( #define FTM_PWM_DEAD_TIME 24 )

```

- 控制上桥臂MOSFETs的PWM信号极性为low
- 控制下桥臂MOSFETs的PWM信号极性为high

```

FTM2->POL = FTM2POL_INIT ;
( #define FTM2POL_INIT FTM_POL_POL0_MASK | FTM_POL_POL2_MASK |
    FTM_POL_POL4_MASK )

```

## FTM2 Fault

- 故障输入引脚1上的高电平信号表明故障发生
- 检测到故障信号后，MCU端PWM信号被设置为高电平
- 检测到故障信号后，MCU端PWM信号被设置为低电平
- 禁止故障输入滤波器

```

FTM2->FLTCTRL |= FTM_FLTCTRL_FAULT1EN_MASK;
FTM2->MODE |= FTM_MODE_FAULTM(2) | FTM_MODE_FAULTIE_MASK;

```

BLDC电机控制中使用RTC模块和一个PIT模块（PIT\_Channel1）来产生定时的周期中断，其中的一个用来作为转速闭环调用的时基，另一个用来进行定时计算转子角度值。

## RTC

- 运行频率为 187.5 kHz

```

RTC->SC |= RTC_SC_RTCLKS(3) | RTC_SC_RTCPS(1);

```

- 产生 10 ms 定时中断

```

RTC->MOD = RTC_MOD_MOD(0x753-0x01);

```

## PIT

- 24 MHz 系统时钟频率

```

PIT->MCR = 0x00;

```

- 向上计数直到周期值并重新加载

```
PIT->CHANNEL[1].TCTRL = 0x03;
```

- 周期为 0xBB8 的3ms定时中断

```
PIT->CHANNEL[1].LDVAL = (0xBB8-0x01);
```

### 5.5.3 霍尔信号捕捉

本应用中，FTM0的两个通道和KBI1模块的5号引脚用来捕捉霍尔传感器的输出信号，上述两个外设模块配置如下：

#### FTM0

- 系统时钟128分频

```
FTM0->SC |= FTM_SC_CLKS(1) | FTM_SC_PS(7);
```

- 上升沿和下降沿输入捕捉使能

```
FTM0->CONTROLS[0].CnSC |= FTM_CnSC_ELSA_MASK |
                          FTM_CnSC_ELSB_MASK;
FTM0->CONTROLS[1].CnSC |= FTM_CnSC_ELSA_MASK |
                          FTM_CnSC_ELSB_MASK;
```

- FTM0的两路通道配置为PTB2 和 PTB3

```
SIM->PINSEL |= SIM_PINSEL_FTM0PS0_MASK | SIM_PINSEL_FTM0PS1_MASK;
```

#### KBI1

- 中断不使能

```
KBI1->SC &= ~KBI_SC_KBIE_MASK;
```

- 上拉使能

```
PORT->PUCEL |= PORT_PUEL_PTCPE3_MASK;
```

- 中断使能

```
KBI1->PE |= 0x20;
```

- 中断标志位清除

```
KBI1->SC |= KBI_SC_KBACK_MASK;
```

### 5.5.4 AD 转换

软件中AD转换是在RTC10ms中断服务程序中采用轮询方式实现的，用于直流母线电压的AD转换，初始化配置如下：

- 单次转换

```
ADC->SC1 &= ~ADCADC_SC1_ADCO_MASK;
```



- 转换结果12位，AD转换时钟3 MHz  
`ADC->SC3 = ADC_SC3_MODE(2) | ADC_SC3_ADIV(3);`
- 软件触发  
`ADC->SC2 &= ~ADC_SC2_ADTRG_MASK;`
- 使能AD10  
`ADC->APCTL1 = 0x4000;`

## 6 Demo的搭建和使用

如下操作，demo就可以搭建起来使用了。

### 6.1 硬件安装

系统的硬件如图6所示，使用之前需完成如下操作：

1. 供电电源插头插入到低压电机功率板接口J1。
2. 用USB 2.0数据线连接个人PC和KE04Z中央控制板上的接口J6。
3. 确认KE04Z中央控制板上的跳线和电容电阻如下操作。
  - 为了调试方便J9, J10, J5必须短接。
  - 去掉控制板上以下电阻：R15, R16, R19, R31, R37, R40, R43, R46, R51, R52, R53, R66, R67, R70, R71 以及 R75。
  - 添加电阻 R4, R17, R18, R39, R42, R45, R47, R50, R61, R64, R69, R72 以及 R73。
  - 将电阻 R6, R7 以及 R3 换成零欧姆阻值。
  - 去掉电容 C5, C6 以及 C2 并添加电容 C3(2200pF)。

控制板的详细修改信息见表 4。

表 4 KE04Z控制板的硬件修改

引脚名称	功能复用	转接板引脚	塔式板两侧引脚	三相低压电机控制板信号	备注
PTA0	SWD_CLK				
PTA1	IO	J27_1	B62	DRV_OC	删除 R51, 添加 R50
PTA2	UART0_RX_2ND	J26_13	A43	ELE_RXD(TWR_SER)	添加 R45, 添加 R47
PTA3	UART0_TX_2ND	J27_9	A44	ELE_TXD(TWR_SER)	添加 R42, 添加 R39
PTA4	SWD_DIO				
PTA5	RESET	J9_6	A63	RESET	
PTA6	SPI0_SCK_2ND	J25_6	B48	SCLK	更改 R7 为 0 Ω, 删除 C6
PTA7	SPI0_MOSI_2ND	J25_18	B45	MOSI	更改 R6 为 0 Ω, 删除 C5
PTB0	SPI0_SS_2ND	J26_5	B47	SS	删除 R53
PTB1	SPI0_MISO_2ND	J26_3	B44	MISO	删除 R52
PTB2	FTM0_CH0_2ND	J26_15	A34	ENC_PHASE_A	删除 R66 和 R67
PTB3	FTM0_CH1_2ND	J25_20	A33	ENC_PHASE_B	删除 R75 和 R43
PTB4	FTM2_CH4	J26_7	B40	PWM_CT	删除 R15
PTB5	FTM2_CH5	J26_9	B39	PWM_CB	删除 R16, 添加 R17
PTB6	IO		B22		删除 R31, 添加 R73
PTB7	IO	J27_5	B52	DRV_EN	删除 R71, 添加 R72
PTC0	FTM2_CH0	J2_4	A40	PWM_AT	删除 R70, 添加 R69
PTC1	FTM2_CH1	J2_3	A39	PWM_AB	删除 R19, 添加 R18
PTC2	ADC0_SE10	J25_10	B30	V_SENSE_DCB	添加 R4 和 C3
PTC3	KBI1_IN5	J25_8	B34	END_INDEX	更改 R3 为 0 Ω, 删除 C2
PTC4	FTM2_CH2	J1_8	A38	PWM_BT	添加 R61, 删除 R40
PTC5	FTM2_CH3	J26_11	A37	PWM_BB	添加 R64, 删除 R46

4. 确认转接板(TWR - MC - FRDMKE02Z)上插针J31, J32, J25, J26 以及 J27 的连接如下操作。
  - J31, J32 (引脚2和3短接) 代表从TowerBoard上获得3.3V和5V供电电压。
  - 去掉板上 J25 的跳冒1和跳冒2。
  - 将J26的3号引脚连到J27的14号引脚, 将J26的5号引脚连到J27的10号引脚, 将J26的13号引脚连到J27的16号引脚, 将J27的9号引脚连到J26的2号引脚, 将J25的6号引脚

连到J27的8号引脚，将J25的8号引脚连到J26的14号引脚，将J25的10号引脚连到J25的17号引脚，将J25的18号引脚连到J27的12号引脚。

详细的修改信息见表 5。

表 5 转接板跳线配置

J26_13	<--->	J27_16
J27_9	<--->	J26_2
J25_6	<--->	J27_8
J25_18	<--->	J27_12
J26_5	<--->	J27_10
J26_3	<--->	J27_14
J25_10	<--->	J25_17
J25_8	<--->	J26_14
J26_11	<--->	J26_12

图 15为KE024Z的转接板(FDRM2TWRMC-KE)，图 16为KE04Z中央控制板(FRDM-KE04Z)，图 17为低压电机驱动板(TWR-MC-LV3PH)。

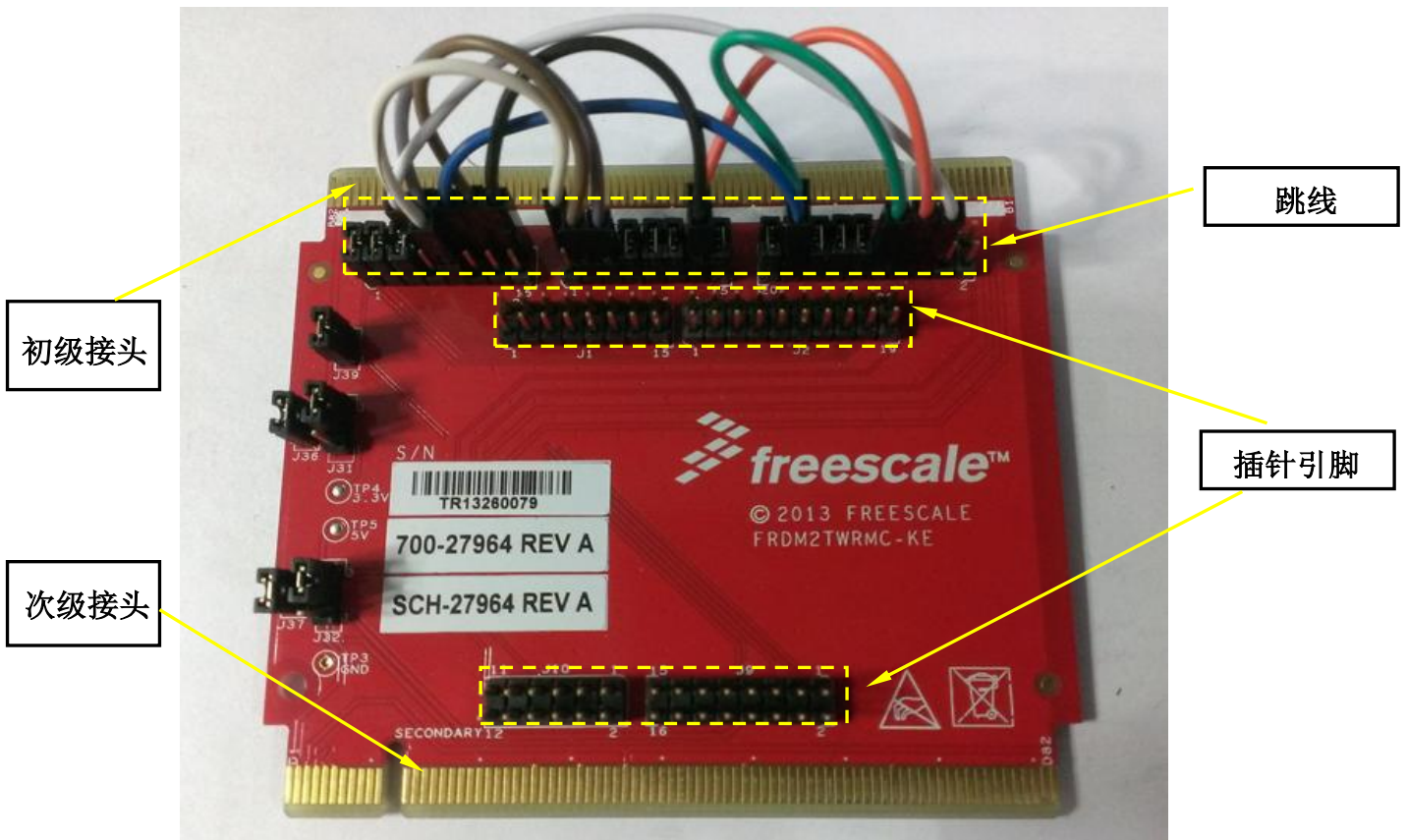


图 15 TWR - MC - FRDMKE02Z转接板

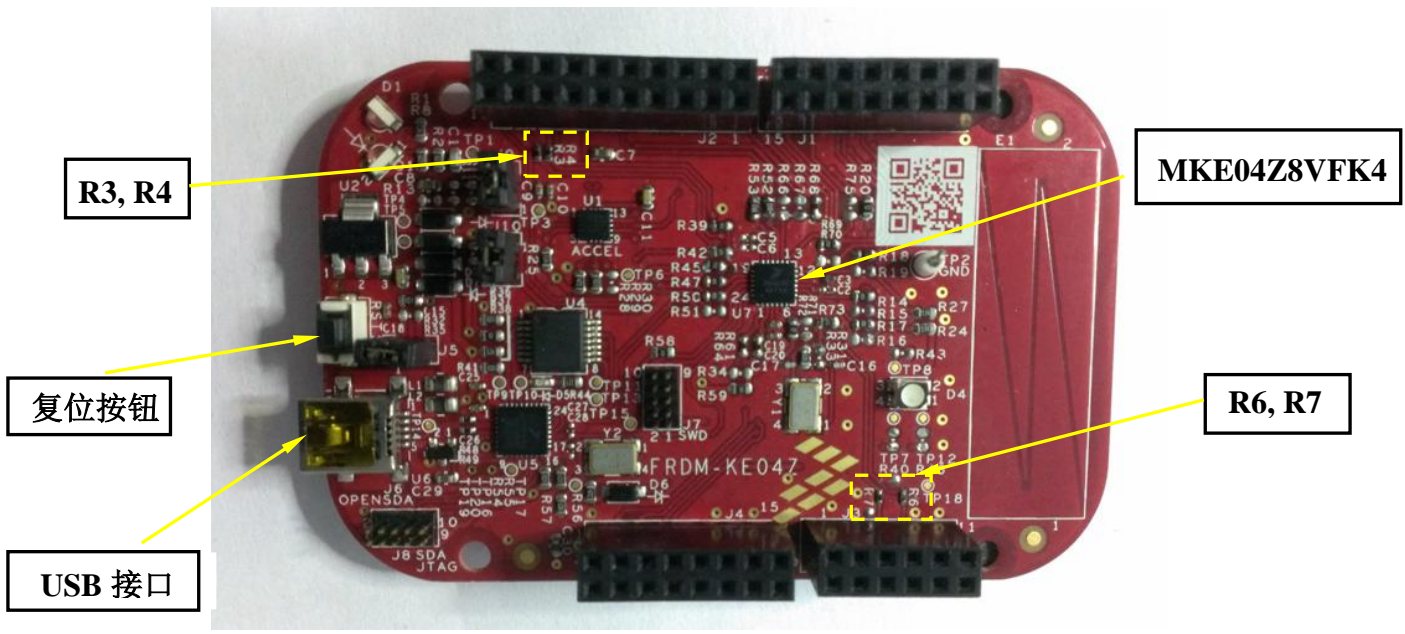


图 16 KE04Z 中央控制板 (FRDM-KE04Z)

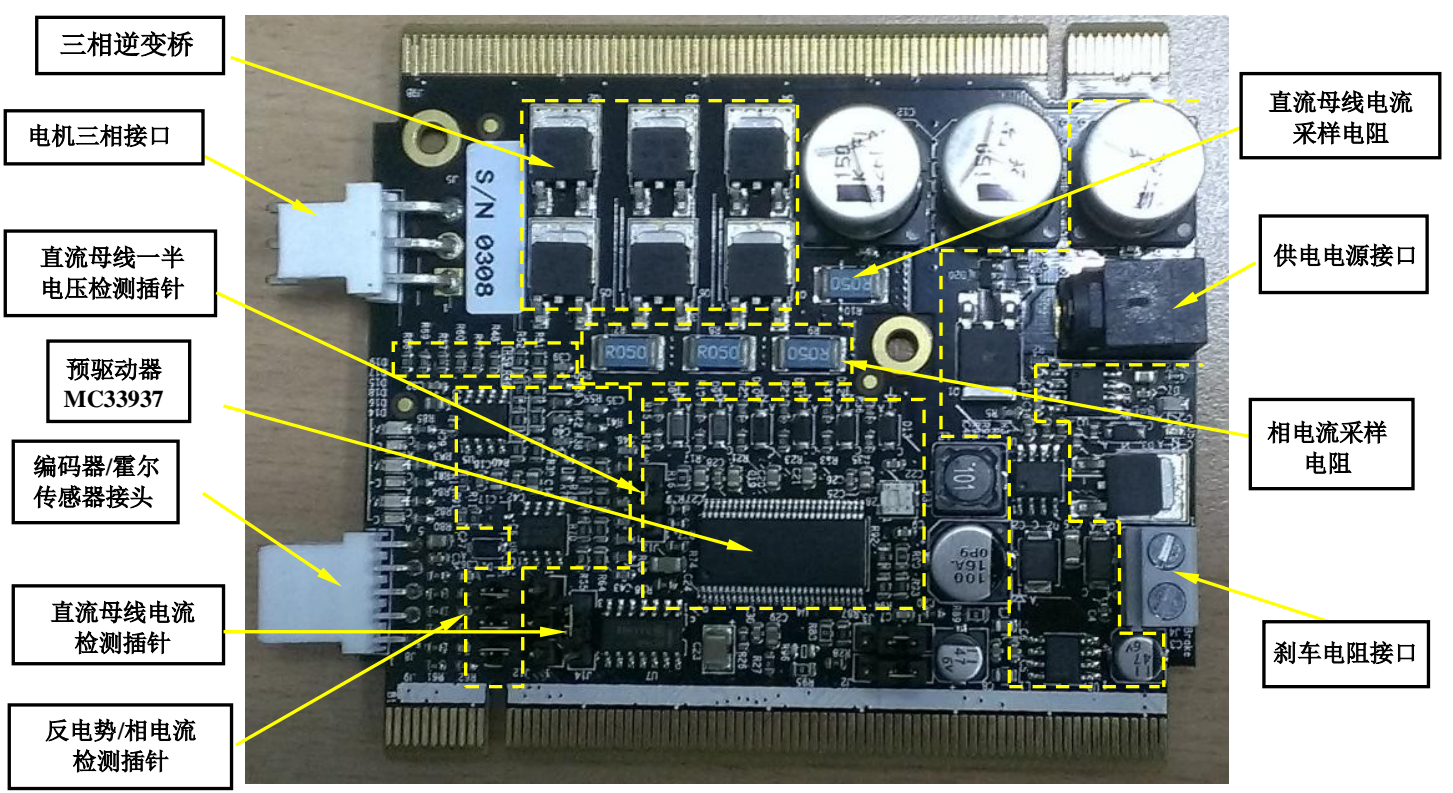


图 17 低压电机驱动板



## 6.2 软件安装

本应用的软件开发环境为IAR Embedded Workbench for ARM v6.6。在第一次使用FreeMASTER软件之前需要安装USB转SCI驱动。在MS word文件“Installation USB/SCI Bridge manual”中描述了驱动的安装步骤。在成功安装驱动之后，选择一个连接到USB端口的串口，这样FreeMASTER调试工具就可以使用了。

## 7 图形结果

### 7.1 FreeMASTER图形界面

图 18为FreeMASTER调试工具的GUI（图形用户界面）。使能UART模块采用串口通信就可以实现FreeMASTER的远程控制。主机通过一个USB线连接到微控制器。用户电脑的USB端口作为一个虚拟串口。USB转UART桥接器完成信号由USB形式到UART形式的转换，反之亦然。在 *project -> Options -> Comm -> Communication* 选项中选择“Direct RS232”作为通信方式，波特率设为9600 bps。在 *project -> Options -> MAP Files* 下的“Defulat symbol file”选项中选择工程生成的可执行文件；在“File format”选项中选择“Binary ELF with DWARF1 or DWARF2 dbg format”。

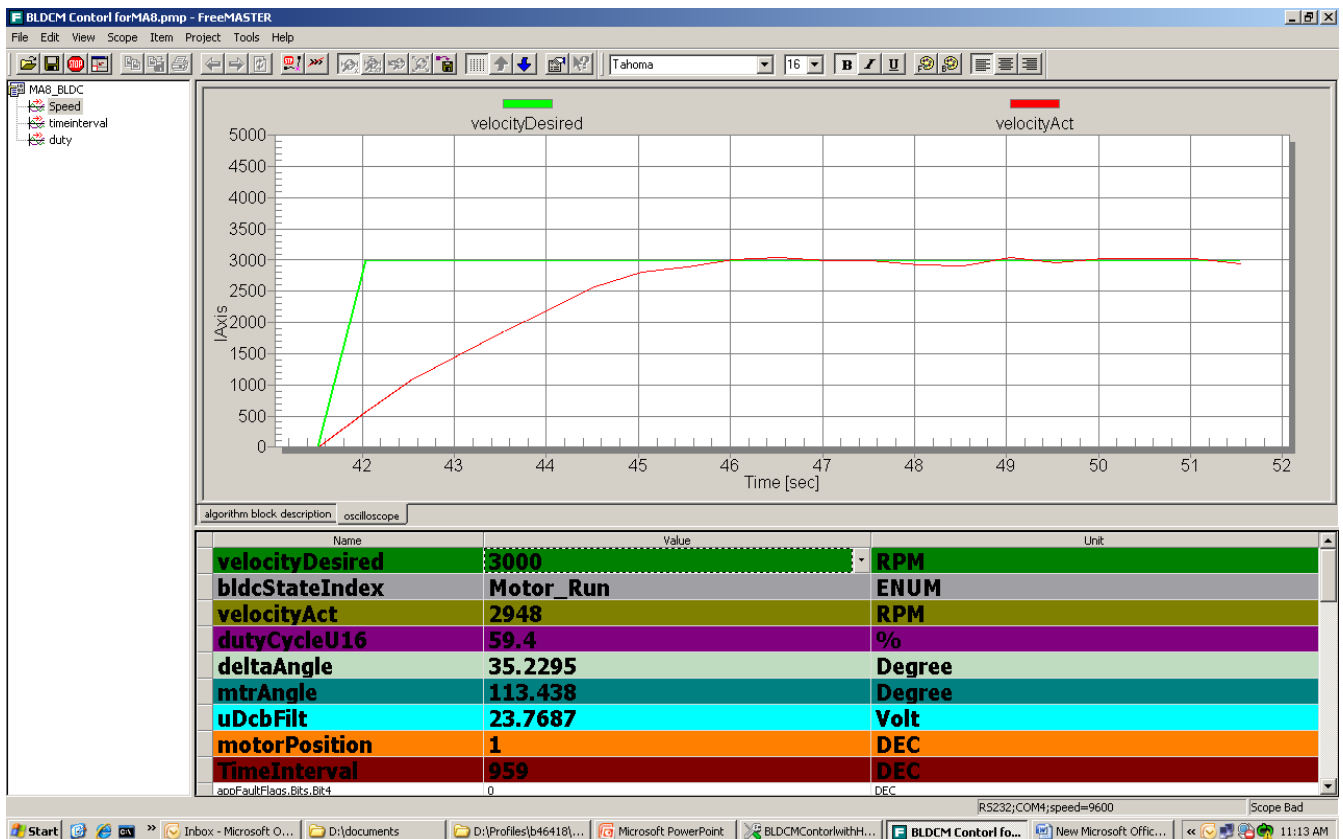
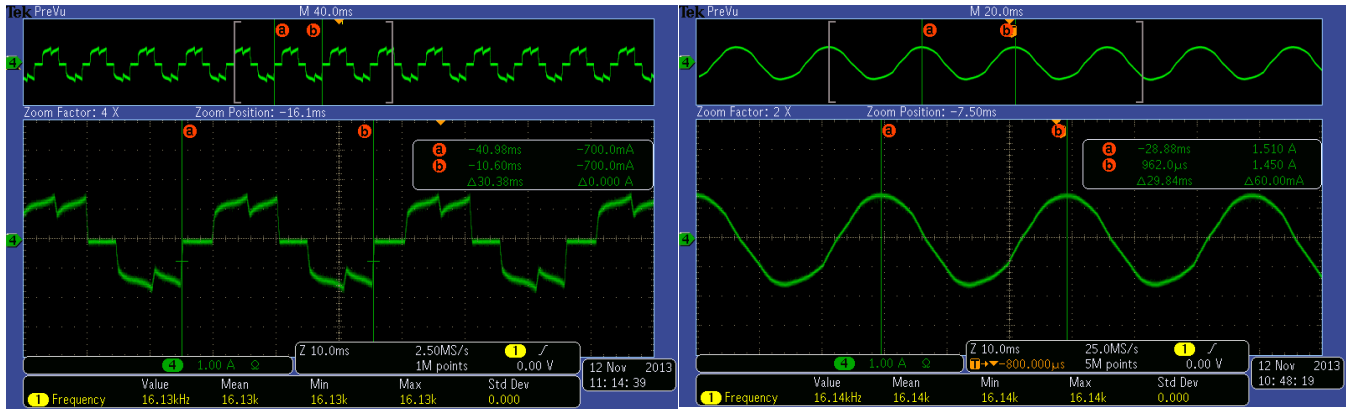


图 18 FreeMASTER 调试界面

## 7.2 相电流和相电压

图 19 为相电流波形，左图为六步方波控制下相电流波形，右图为正弦波控制下的相电流波形。很明显，正弦波控制模式下，相电流波形是正弦的，转矩波动小。

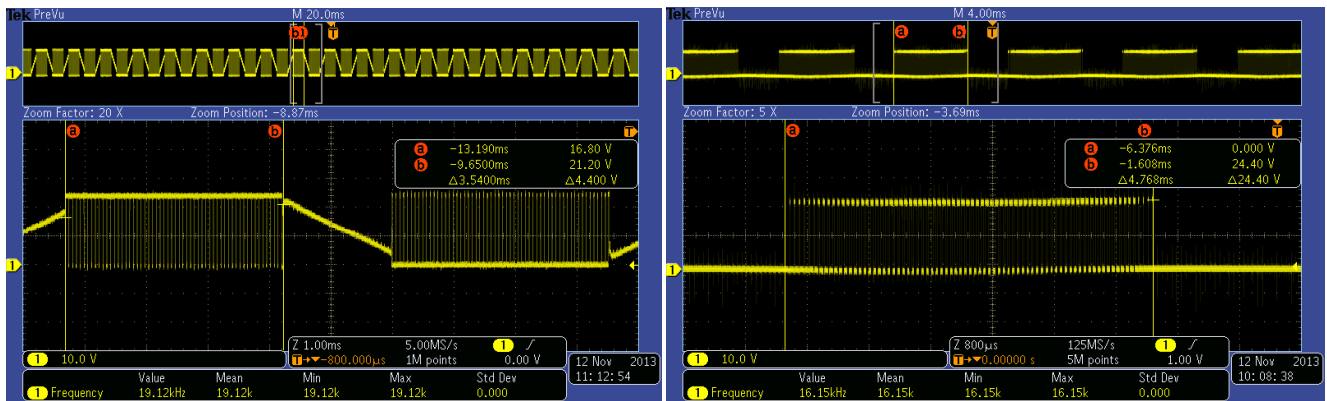


相电流（六步方波控制）

相电流（正弦波控制）

图 19 相电流

图 20 为相电压波形。左图为双极性六步方波控制的相电压波形，右图为正弦波控制的相电压波形。正弦波控制的相电压波形如图 9 所示。



相电压（双极性六步方波控制）

相电压（正弦波控制）

图 20 相电压

## 8 参考文献

以下参考文献均可从飞思卡尔官网freescale.com获得。

- *KE04 Sub-Family Reference Manual* (MKE04P24M48SF0RM)
- *KE04 Series Data Sheet* (MKE04P80M48SF0)
- *TWRMCLV3PHUG: TWR-MC-LV3PH User's Guide* (TWRMCLV3PHUG)
- *TWR-MC-LV3PH Schematics*
- *TWR-SER-SCH Schematics*



**How to Reach Us:****Home Page:**

[www.freescale.com](http://www.freescale.com)

**Web Support:**

[www.freescale.com/support](http://www.freescale.com/support)

本文档中的信息仅供系统和软件实施方使用 Freescale 产品。未包含基于本文档信息设计或加工任何集成电路的任何明示或暗示的版权许可授权。

Freescale 保留对此处任何产品进行更改的权利，恕不另行通知。Freescale 对其产品在任何特定用途方面的适用性不做任何担保、表示或保证，也不承担因为应用或使用产品或电路所产生的任何责任，明确拒绝承担包括但不限于因果性或附带损害在内的所有责任。

Freescale 数据表和/或规格中所提供的“典型”参数在不同应用中可能并且确实不同，实际性能会随时间而有所变化。所有操作参数，包括“典型值”在内，在每个客户应用中必须经由客户的技术专家进行验证。Freescale 未转让与其专利权及其他权利相关的许可。

Freescale 销售产品时遵循以下网址中包含的标准销售条款和条件：

[freescale.com/SalesTermsandConditions](http://freescale.com/SalesTermsandConditions)

Freescale, Kinetis, and the Freescale logo are trademarks of Freescale Semiconductor, Inc., Reg. U.S. Pat. & Tm. Off. All other product or service names are the property of their respective owners.

© 2014 Freescale Semiconductor, Inc.