Fortior Tech
峰岹科技

峰岹科技

FU68xx-有感方波

调试说明文档-V1.0.0

峰岹科技(深圳)有限公司

Fortior Technology(Shenzhen) Co.,Ltd.

深圳市南山区科技中二路软件园 11 栋 2 楼 203 室,518057
Room203,2/F,Building No.11,Keji Central Road 2,Software Park,
High-Tech Industrial Park, Shenzhen,P.R.China
Tel：86-755-26867710
Fax：86-755-26867715

修改记录

版本号：第 1 位-原理 第 2 位-模块 第 3 位-细节

版本号	修改详细内容说明	生效日期	修订者	审核者
V1.0.0	初稿	2020-3-5	Dean.Wang	John.Luo

1. 概述

- 本调试指导针对软件方波程序
  芯片类型: FU6861Q
  软件支持直流无刷电机无 HALL 方波控制
  速度开环和速度闭环控制

- 软件编写的几点原则：
  1、禁止 此文件中的结构体变量不能改动，改动后编译会出错。
  2、禁止 使用”TAB”键禁止缩进。
  3、禁止 在使用没有明确含义的全局变量。
  4、建议 每个功能函数，放在明确的“层”中。

  本调试文档参考 datasheet-MCU-DS-010C_FU6812_61_Datasheet_V1.32(20181115 中文版)。

2. 方波控制工作原理

2.1. BLDC 的六步换相

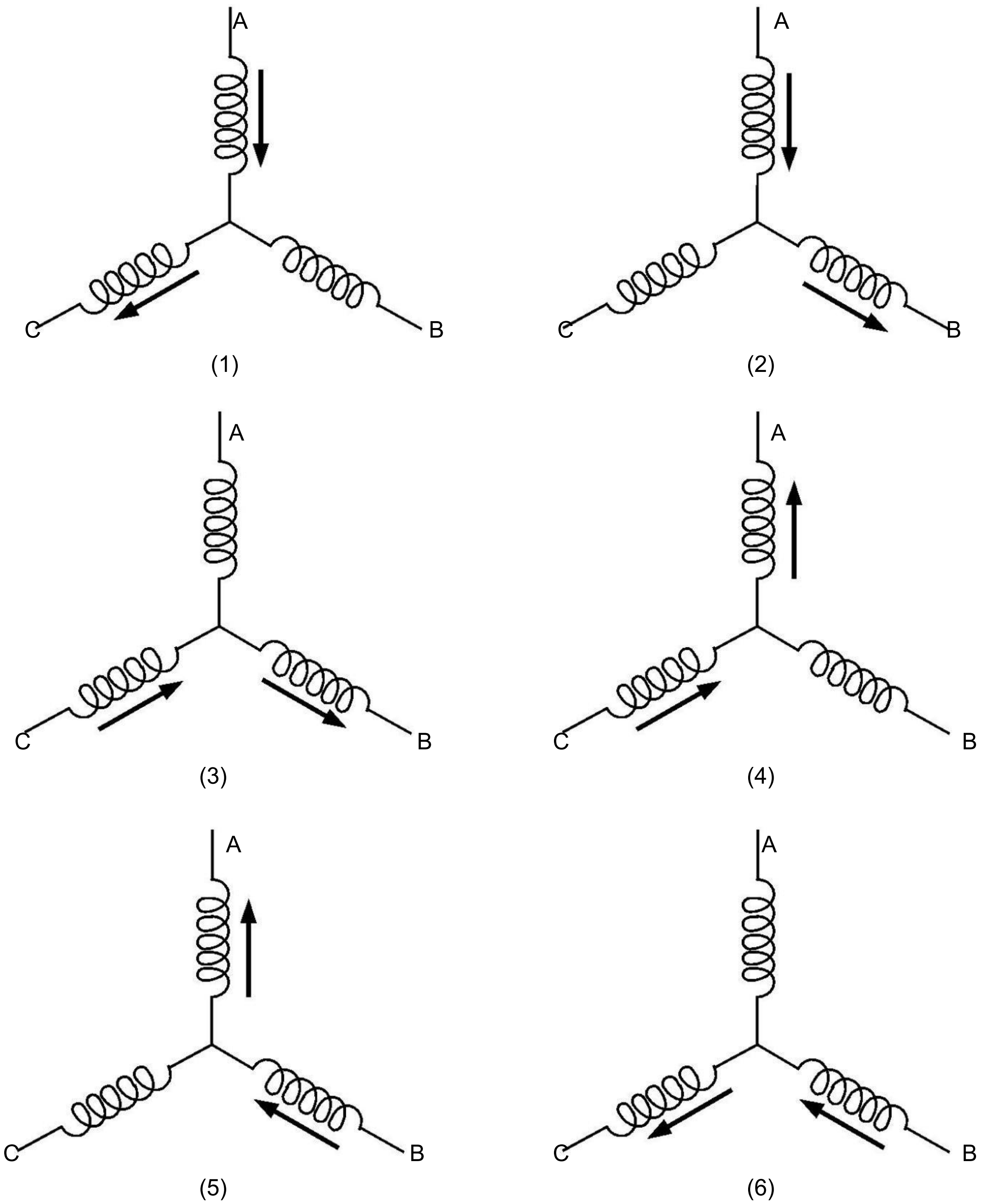


图 2-1-1　通电顺序

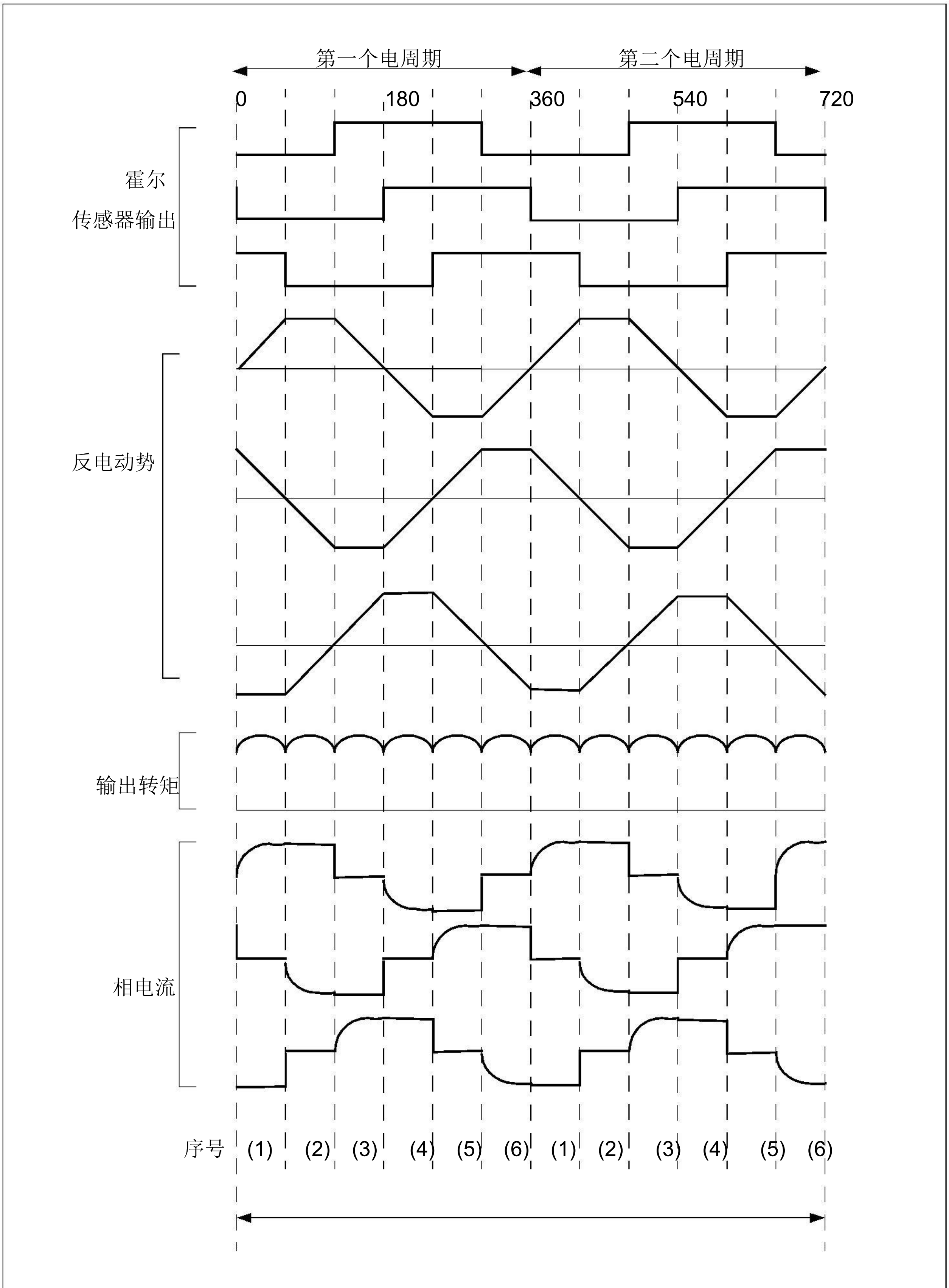


图 2-1-2　霍尔传感器信号、电机反电动势、输出转矩、相电流波形

上面图 2-1-1 是按照霍尔信号给绕组通电的示意图，图 2-1-2 展示了霍尔信号和反电势、输出转矩及相电流的关系。

注意：图 2-1-1 中的序号和图 2-1-2 中的序号是一一对应关系。

由图 2-1-2 中可以得出，在电机转动 60 度后，霍尔信号是会发生改变的，通电的顺序也需要发生相应的变化，所以这样一个电周期就需要六次换相（注意：这里的电周期和机械周期不同，电周期*极对数 = 机械周期）。下面就根据上图中的信息来对应到功率器件的开通和关断。

2.2. BLDC 中 MOS 的开关顺序

序号	霍尔信号输入			有效 PWM		相电流		
	A	B	C			A	B	C
1	0	0	1	Q1	Q4	DC+	关闭	DC-
2	0	0	0	Q1	Q5	DC+	DC-	关闭
3	1	0	0	Q3	Q5	关闭	DC-	DC+
4	1	1	0	Q3	Q6	DC-	关闭	DC+
5	1	1	1	Q2	Q6	DC-	DC+	关闭
6	0	1	1	Q2	Q4	关闭	DC+	DC-

表 2-2-1：顺时针方向转动电机

序号	霍尔信号输入			有效 PWM		相电流		
	A	B	C			A	B	C
1	0	1	1	Q3	Q5	关闭	DC-	DC+
2	1	1	1	Q1	Q5	DC+	DC-	关闭
3	1	1	0	Q1	Q4	DC+	关闭	DC-
4	1	0	0	Q2	Q4	关闭	DC+	DC-
5	0	0	0	Q2	Q6	DC-	DC+	关闭
6	0	0	1	Q3	Q6	DC-	关闭	DC+

表 2-2-2：逆时针方向转动电机

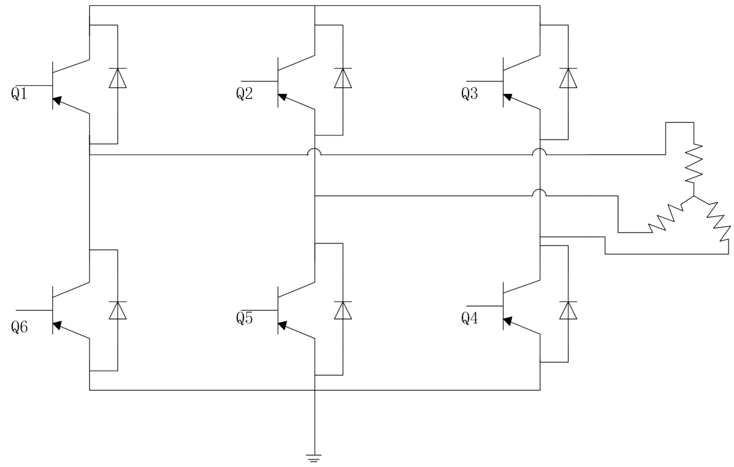


图 2-2-1：功率器件示意图

上面表 2-2-1 和表 2-2-2 中展示的是，在不同转向的时候，不同的霍尔状态对应的不同的 MOS 开关，和在绕组中不同的电流流向。表中的信息和图 2-2-1 中是对应关系，有效 PWM 代表的就是对应的 MOS 在开通状态，其余的 MOS 处于关闭状态。

这里需要注意的是，本文中的霍尔器件为 60 度安装，所以 HALL 信号会出现全零和全一的状态，120 度安装的时候是不会出现的。并且程序中控制一般是按照 120 度安装的电机来控制，当检测到 HALL 出现全零或全一时就会报 HALL 故障，程序中处理 60 度安装的 HALL 是将 HALL B 进行反向然后处理。

2.3. BLDC 电机的无感控制

2.3.1. 反电动势的定义

在无刷电机转动时，每个绕组都会产生反电动势的电压，根据楞次定律，反电势的方向与提供给绕组的主电压相反。反电势主要取决于三个因素：转子角速度、转子磁体产生的磁场和定子绕组的匝数。

公式：反电动势 =（E）∝ N*I*R*B*W

N 为每相绕组的匝数，l 为转子的长度，R 为转子的内径，B 为转子的磁场密度，W 为电机的角速度。

由上文中的公式得知，在电机做定后，电机的绕组和转子是固定的，所以电机的反电势只和角速度有关，角速度的提高，电机的反电势也会相应的提高。

2.3.2. BLDC 电机的无感控制

上面我们已经介绍了通过霍尔传感器控制电机换向的过程和步骤。无刷电机还可以通过反电势来控制电机换向，这就是通常说的无感控制。通过图 2-1-2 可以看出，在 HALL 信号发现变化的时

候，悬空相的反电势也会通过过零点。这就是无感控制中采样过零点控制电机换向的原理。

通过反电势控制电机换向能节省 HALL 的成本和 HALL 安装的成本，但是需要注意的是，在电机未启动的时候，反电势基本为零，这时是采集不到过零点。因此，在电机从静止状态启动的时候就必须先要将电机拖动起来后再用过零点控制。这个拖动一般分为两种，一种是强拖，强拖就是在电机静止的时候通过人为的切换 MOS 的开关来控制内部磁场变化来拖动电机。二是拖过高频注入（IPD）来判断电机的第一步的位置后，再进行拖动。

2.4. BLDC 工作原理

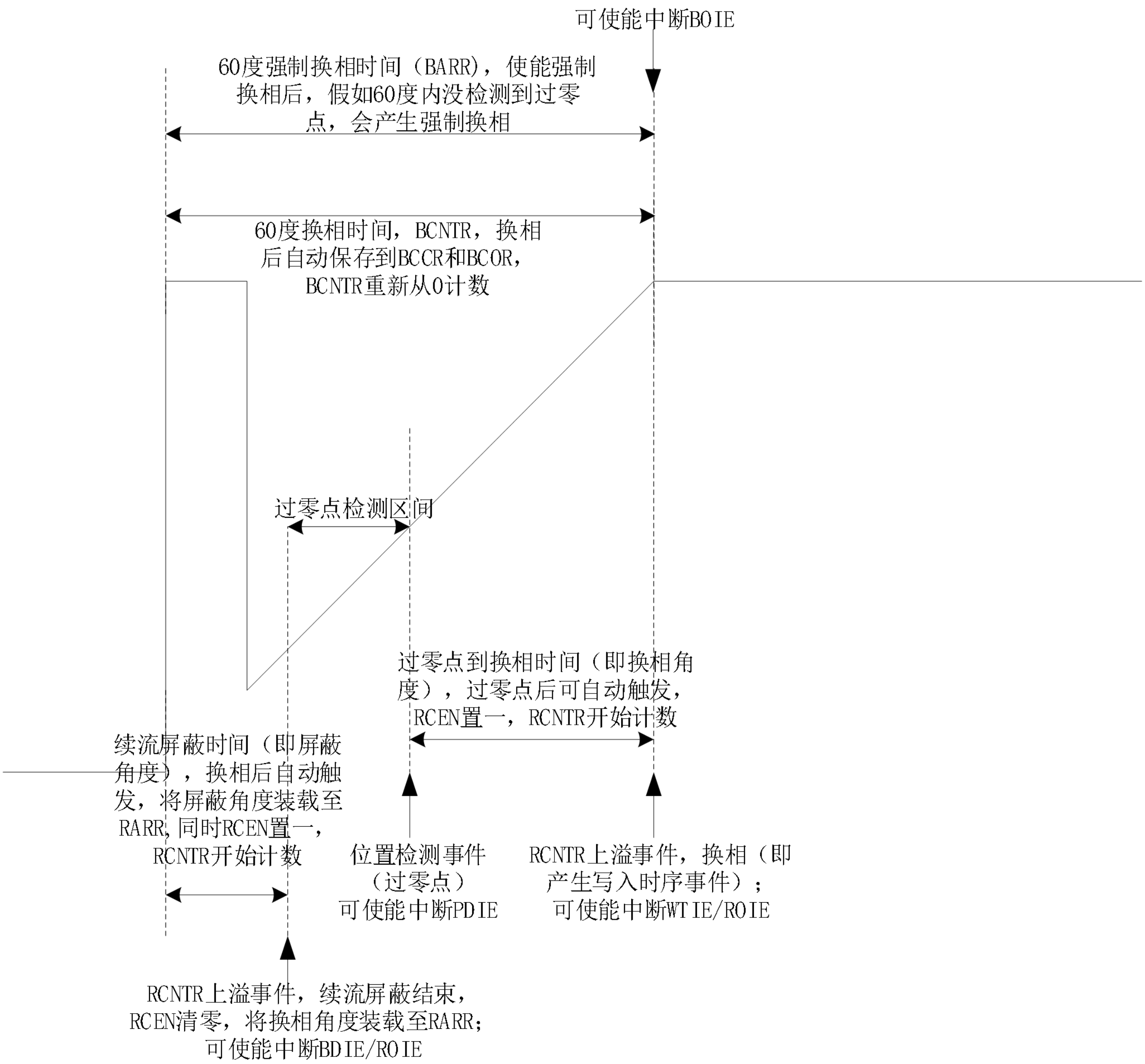


图 2-4-1：BLDC 工作原理简要说明图

方波控制中有续流屏蔽角度和换相角度。如上图为悬空相的 60 度相电压波形，在无感运行的时候采集过零点就是在这个相上采集，此时该相的电感能量需要经过续流二极管释放到电源或者地，

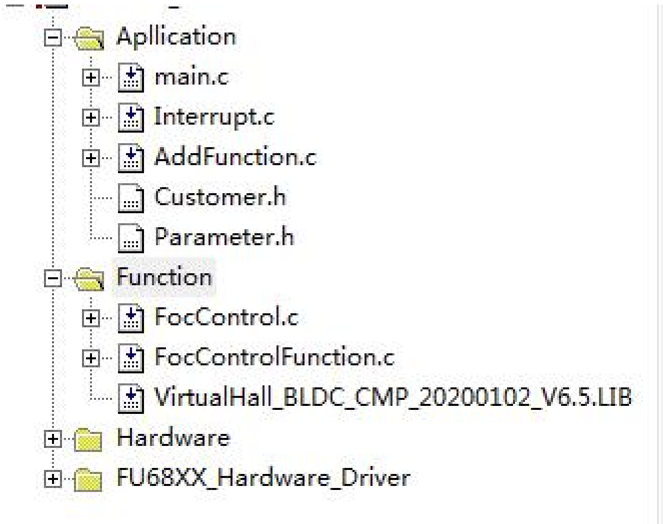
在续流过程中，比较器会受到影响，所以需要屏蔽续流时间内的比较器产生的触发沿信号，以免续流产生的错误信号引起错误换相。理论上过零点会在 30 度的地方出现，所以这个续流屏蔽角度值不能超过 30 度。

上面说到了虚拟过零点一般在 30 度左右出现，但是真正的换相点理论上是在过零点出现后 30 度。所以这里就涉及到换相角度，这个换相角度也是不能超过 30 度的。这个换相角度是调整换相点的，在电机的运行过程中调整这个换相角度是可以调整电机运行的效率。当调整到两个斜边对称的时候就是效率最好的时候。

（此文中介绍的可参考我司 datasheet《MCU-DS-010C_FU6812_61_Datasheet_V1.32 (20181115 中文版)》中 14 章中的 14.2）

3. 程序架构说明

3.1. 程序架构层说明



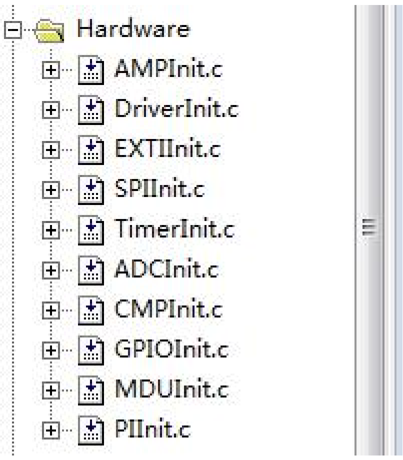
Apllication 为程序中的应用层，Function 为功能层，Hardware 为芯片硬件配置层。

Apllication（应用层）中分主函数、中断程序、调试应用层和系统参数配置层，在调试程序的时候主要调整的为调试应用层。

Function（功能层）主要分为 FocControl.c 和 FocControlFunction.c，FocControl.c 中主要为电机状态机控制，FocControlFunction.c 中主要为功能程序添加，如需添加控制程序，就在此文件中添加。

Hardware 为芯片硬件配置层。

3.2. 芯片硬件配置



在此层中，主要为芯片硬件系统配置，芯片硬件的初始化都在这里完成的。例如（ADC、Timer、

比较器、IO 口等）

3.3. 系统状态机

```
typedef enum
{
        mcReady      = 0,
        mcInit       = 1,
        mcCharge     = 2,
        mcTailWind   = 3,
        mcPosiCheck  = 4,
        mcBLDC       = 5,
        mcAlign      = 6,
        mcStart      = 7,
        mcRun        = 8,
        mcStop       = 9,
        mcFault      = 10,
        mcPllTect    = 11,
        mcBrake      = 12,
}MotStaType;
```

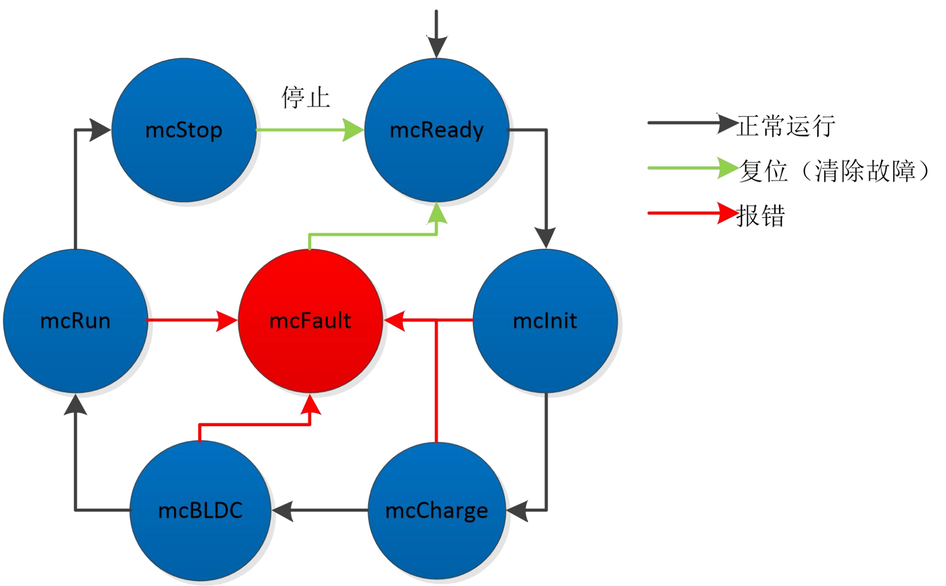
添加状态机工作流程说明：如 FOC 状态机工作流程

- mcReady：　准备参数，将系统运行需要的参数配置到寄存器中。
- mcInit：　　电机参数配置和 PI 参数配置。
- mcCharge：　自举电容充电。
- mcBLDC：　电机启动的程序。
- mcRun：　　电机正常运行的程序。
- mcStop：　电机停机程序。中间包括自由停机和刹车停机。
- mcFault：　电机报错的程序。

注意：

其余状态为无用状态，留做备用的。

程序执行流程：



3.4. 用户文件夹

Parameter.h	系统参数调整
Customer.h	用户参数调整
Interrupt.c	中断文件
Main.c	主函数
FocControl.c	电机控制层
FocControl Function_layer.c	功能层

4. 调试应用层

4.1. 硬件电路说明

例子：**18V** 电动工具

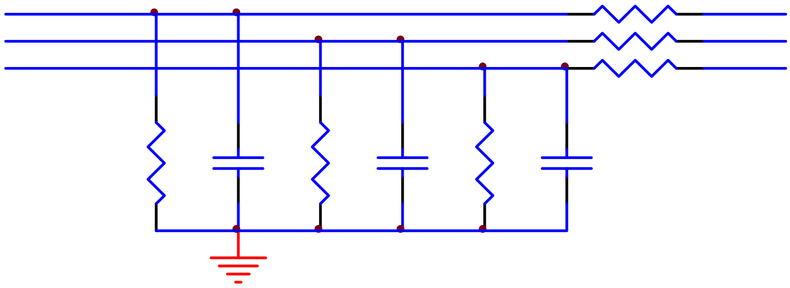


图 4-1-1：BEMF 检测电路

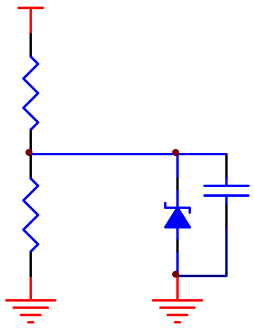


图 4-1-2：母线电压采样电路

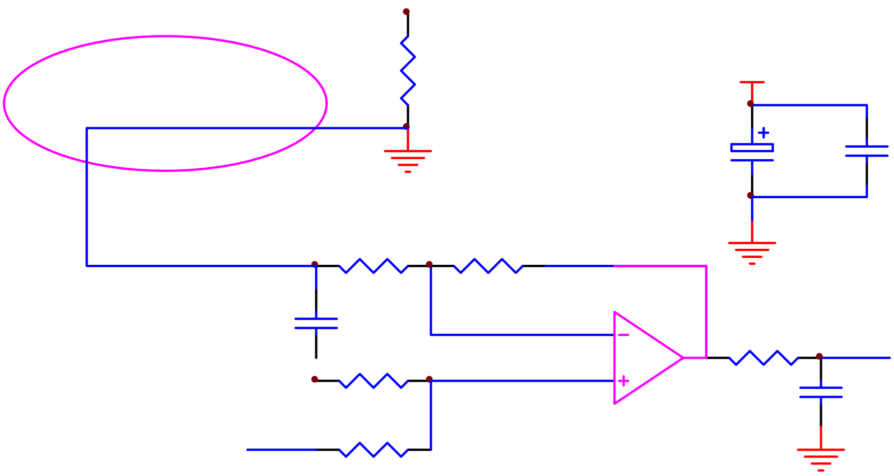


图 4-1-3：母线电流采样电路

4.2. 基本参数调试

4.2.1. 斩波频率选择

```
/*芯片参数值-------------------------------------------------------------------*/
 /*CPU and PWM Parameter*/
 #define MCU_CLOCK                       (24.0)                          // (MHz) 主频
 #define PWM_FREQUENCY                   (30.0)                          // (kHz) 载波频率
 #define PWM_FREQUENCY_BLDC              (16.0)                          // (kHz) 载波频率
 /*deadtime Parameter*/
```

本程序中载波的范围为 8-20K，一般应用为 16K，电动工具一般为 12K 左右，具体可以看 MOS 的发热情况来定。

4.2.2. 电机参数配置

```
#define Pole_Pairs                      (2.0)                           // 极对数
```

电机极对数配置，用于计算电机转速。

4.2.3. 驱动电平配置

```
/*硬件板子参数设置值------------------------------------------------------------*/
/*PWM high or low level Mode*/
/*根据驱动芯片的类型选择*/
 #define All_High_Level                 (0)                        // 驱动高电平有效
 #define All_Low_Level                  (1)                        // 驱动低电平有效
 #define UP_H_DOWN_L                    (2)                        // 上桥臂高电平有效，下桥臂低电平有效
 #define UP_L_DOWN_H                    (3)                        // 上桥臂低电平有效，下桥臂高电平有效
 #define PWM_Level_Mode                 (UP_H_DOWN_L)
```

驱动电平的选择是根据芯片内集成的 Drive 或者芯片外围接的 Drive 决定的，FU6831 内部集成的是 3P3N 的 Drive，驱动电平为上高下低，FU6818 内部集成的为 6N 的 Drive，驱动电平为全高，FU6861 内部集成的为 6N 的驱动，驱动电平为上高下低，如果用 FU6811 或者 FU6812 就需要根据外围接的 Drive 设置驱动电平。

4.2.4. 硬件参数配置（重要）

```
/*hardware current sample Parameter*/
/*电流基准的电路参数*/
 #define HW_RSHUNT                      (0.001)                    // (Ω)   采样电阻
 #define HW_AMPGAIN                     (10.0)                     // 运放放大倍数
 #define HW_ADC_REF                     (4.5)                      // (V)   ADC参考电压
```

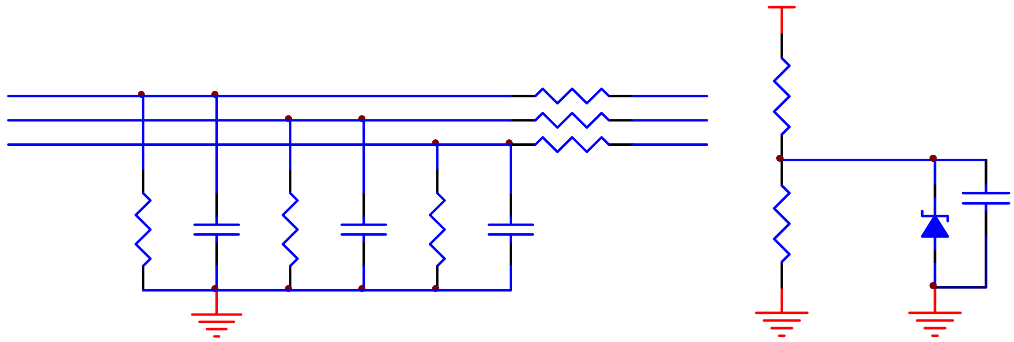
电流采样参数配置，

采样电阻的选择是由系统的最大电流决定的，选择电阻的功率应该能承受电流产生的最大功率，并且需要留下相应的余量。并且放大倍数的选择应该满足，在最大电流的时候在采样电阻上产生的电压乘上放大倍数小于选择的参考电压。

参考电压的选择有 3V、4V、4.5V 和 5V（对应 datasheet 第 25 章）。

```
/*hardware voltage sample Parameter*/
/*母线电压采样分压电路参数*/
 #define RV1                             (0.0)                           // (kΩ) 母线电压分压电阻1
 #define RV2                             (30.0)                          // (kΩ) 母线电压分压电阻2
 #define RV3                             (5.1)                           // (kΩ) 母线电压分压电阻3
 #define RV                              ((RV1 + RV2 + RV3) / RV3)       // 分压比
```

母线分压电阻和反电势分压电阻配置，两个分压电阻必须保持一致。



母线电压分压电阻的选择和反电势分压电阻的选择应该满足在系统最大电压的时候分回来的电压满足小于参考电压，并且最好能满足在正常工作电压的时候分回来的电压在参考电压的 2/3 处。反电势分压处的滤波电容的选择，经验值为 100PF，如果工作条件恶劣或者板子的干扰大的话可以适当增大这个电容，最大 10NF。

4.2.5. IPD 注入 duty 控制

在 IPD 的时候可能会触发硬件过流，在确保硬件没问题后，可以调试这个 duty，并且在开始找位置不准的时候可以调试这个 duty。

```
/***************VirtlHall*********************************/
 #define VirtlHall_Ipd_Duty             (0.65)                   // IPD脉冲Duty, 16K载波时
 #define VirtlHall_cmp_samr             (uint8)(0xCC)                 // IPD脉冲Duty, 16K载波时
```

4.2.6. 启动调试参数

下图中 56 行和 57 行两个参数调试是在需要宽电压启动的时候分段用的 56 行填的 19 就是 19V，57 填的 16 就是 16V，可以根据当前项目需求来填相应的电压。

59 行和 60 行为在欠压值到低电压阈值时启动的 KR 和 KF 值。下图示例就是欠压值到 16V 时启动的 KR 和 KF 值。

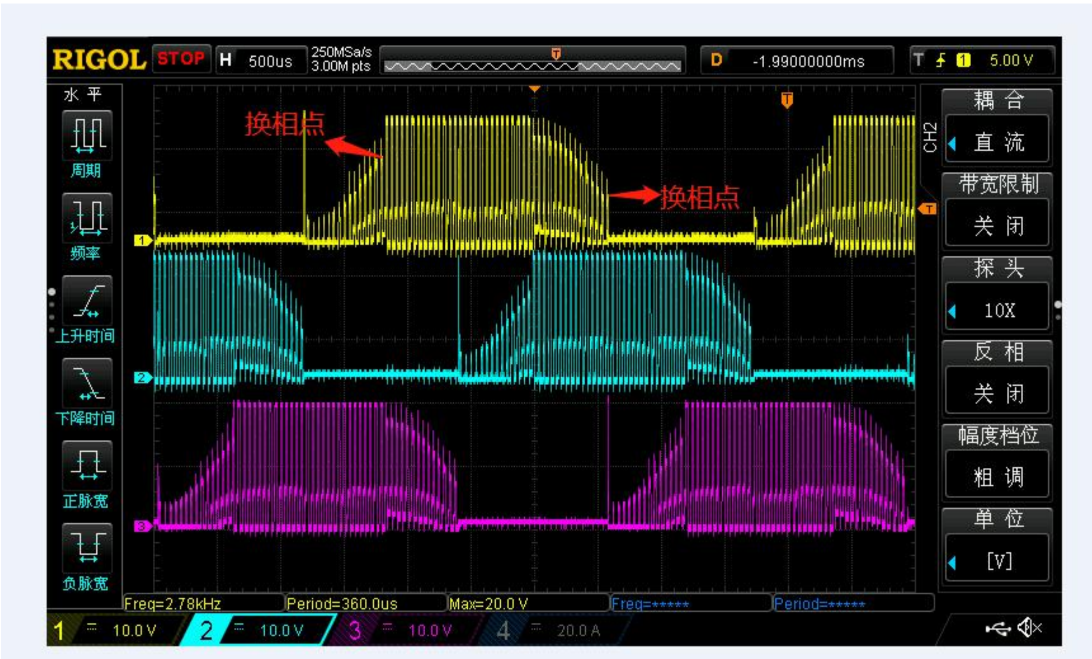
62 和 63 行为在低电压阈值到高电压阈值时启动的 KR 和 KF 值。

65 和 66 行为在高电压阈值到过压值时启动的 KR 和 KF 值。

下面 68-69 行、70-71 行和 74-75 行为对应电压在运行时的 KR 和 KF 值。

```c
#define VirtlHall_High_mcDcbus          _Q15(19.0/HW_BOARD_VOLT_MAX)          // KF和KR电压分段，高电压段阈值
#define VirtlHall_Low_mcDcbus           _Q15(16.0/HW_BOARD_VOLT_MAX)          // KF和KR电压分段，低电压段阈值

#define VirtlHall_KF_Open_MinLowDC      _Q15(0.45)                            // Open段，且母线电压为欠压值到低电压段阈值时的
#define VirtlHall_KR_Open_MinLowDC      _Q15(0.55)                            // Open段，且母线电压为欠压值到低电压段阈值时的

#define VirtlHall_KF_Open_LowHighDC     _Q15(0.45)                            // Open段，且母线电压为低电压段阈值到高电压段阈
#define VirtlHall_KR_Open_LowHighDC     _Q15(0.55)                            // Open段，且母线电压为低电压段阈值到高电压段阈

#define VirtlHall_KF_Open_HighMaxDC     _Q15(0.45)                            // Open段，且母线电压为高压阈值到过压值时的KF值
#define VirtlHall_KR_Open_HighMaxDC     _Q15(0.55)                            // Open段，且母线电压为高压阈值到过压值时的KR值

#define VirtlHall_KF_Run_MinLowDC       _Q15(0.40)                            // Run段，且母线电压为欠压值到低电压段阈值时的
#define VirtlHall_KR_Run_MinLowDC       _Q15(0.65)                            // Run段，且母线电压为欠压值到低电压段阈值时的

#define VirtlHall_KF_Run_LowHighDC      _Q15(0.40)                            // Run段，且母线电压为低电压段阈值到高电压段阈
#define VirtlHall_KR_Run_LowHighDC      _Q15(0.65)                            // Run段，且母线电压为低电压段阈值到高电压段阈

#define VirtlHall_KF_Run_HighMaxDC      _Q15(0.40)                            // Run段，且母线电压为高压阈值到过压值时的KF值
#define VirtlHall_KR_Run_HighMaxDC      _Q15(0.65)                            // Run段，且母线电压为高压阈值到过压值时的KR值
```

调试 KR 和 KF 就是调整的换相点，将启动时候的波形尽量调试到对称。KR 上向上斜边的换相点，KF 为向下斜边的换相点。KR 的范围为 0.5-0.95，KF 的范围为 0.05-0.5，这两个点调整后，想应的换向点就会上移或者下移。在启动的时候如遇到抖动或者卡顿，在启动力矩和启动切换速度设置好后，就调整这两个值，将换相点移动，将启动时候的相电压波形的斜边调整好，启动就会相应的变好。



4.2.7. 启动时速度分段和 duty 分段

转速分段：

MOTOR_SPEED_BLDC2FOC_RPM0 乘以极对数的值必须大于 915

MOTOR_SPEED_FOC2BLDC_RPM 的数值应位于 MOTOR_SPEED_BLDC2FOC_RPM1 与 MOTOR_SPEED_BLDC2FOC_RPM2 之间。

```
//Pole_Pairs*MOTOR_SPEED_BLDC2FOC_RPM0/1/2必须大于915
 #define MOTOR_SPEED_BLDC2FOC_RPM0       (500.0)                                      // 低速无感方波大力矩切高速无感BLDC的速度，单位
 #define MOTOR_SPEED_BLDC2FOC_RPM1       (1000.0)                                      // 低速无感方波大力矩切高速无感BLDC的速度，单
 #define MOTOR_SPEED_BLDC2FOC_RPM2       (1500.0)                                     // 低速无感方波大力矩切高速无感BLDC的速度，单位

 #define MOTOR_SPEED_FOC2BLDC_RPM        (1200.0)                                     // 高速无感BLDC切低速无感方波大力矩的速度，单位
```

Duty 分段：

```
#define Min_BLDC_Duty                  (0.085)                                  // VSP起始最小Duty
#define Max_BLDC_VH_Duty_0             (0.15)                                  // 速度在0到(MOTOR_SPEED_BLDC2FOC_RPM/3)区间的
#define Max_BLDC_VH_Duty_1             (0.20)                                  // 速度在(MOTOR_SPEED_BLDC2FOC_RPM/3)到(2*MOTOR
#define Max_BLDC_VH_Duty_2             (0.25)                                  // 速度在(2*MOTOR_SPEED_BLDC2FOC_RPM/3)到MOTOR_
#define Max_BLDC_Duty                  (1.00)                                  // VSP终点最大Duty
```

在启动阶段，程序中是将 duty 限制了的，大致分了三段，第一段就是 Max_BLDC_VH_Duty_0，在这个 duty 下电机的转速必须达到 MOTOR_SPEED_BLDC2FOC_RPM0，要不就不会切入到第二段。MOTOR_SPEED_BLDC2FOC_RPM 为启动阶段切入到正常运行阶段的转速。当在启动的时候发现电机在一个低速的情况下维持，外部调速信号给上去了转速还是不上去的话，可能就是 duty 限制将 duty 限制了，转速又没有 达到切换转速，这个时候就需要提高限制 duty 或者降低切换转速了。

注意：启动的限制力矩不建议给到太大。

4.2.8. 加速度参数配置

```
#define Motor_Duty_Count_High          (int16)(2)                    //Duty增减响应的载波间隔个数
#define Motor_Duty_Count_Low           (int16)(4)                    //Duty增减响应的载波间隔个数
```

Interrupt.c:

```
          mcSpeedRamp.DelayPeriod++;
          if(mcSpeedRamp.DelayPeriod >= Motor_Duty_Count)
          {
            mcSpeedRamp.DelayPeriod = 0;

            if(McStaSet.SetFlag.BLDCSetFlag == 4)
            {
                if(mcSpeedRamp.BLDC_Value > (DRV_DR + 1))
                {
                  DRV_DR = DRV_DR + 1;
                }
                else if(mcSpeedRamp.BLDC_Value  < (DRV_DR - 1))
                {
                  DRV_DR = DRV_DR - 1;
                }
                else
                {
                  DRV_DR = mcSpeedRamp.BLDC_Value;
                }
             }
```

电机的加速度在这两个地方配置，在上面的配置文件中配置的是增加 duty 的载波间隔数，下面这个图中是在载波中断中加 duty 的步进。上面间隔个数越大加速度越慢，下面每次 DRV_DR 加减越小加速度越慢。下面中断中步进给越大加速度就越大。

注意：当加速度太快的时候会造成在启动的时候电流会有过冲的情况。

4.2.9. 换相角度和续流屏蔽角度配置

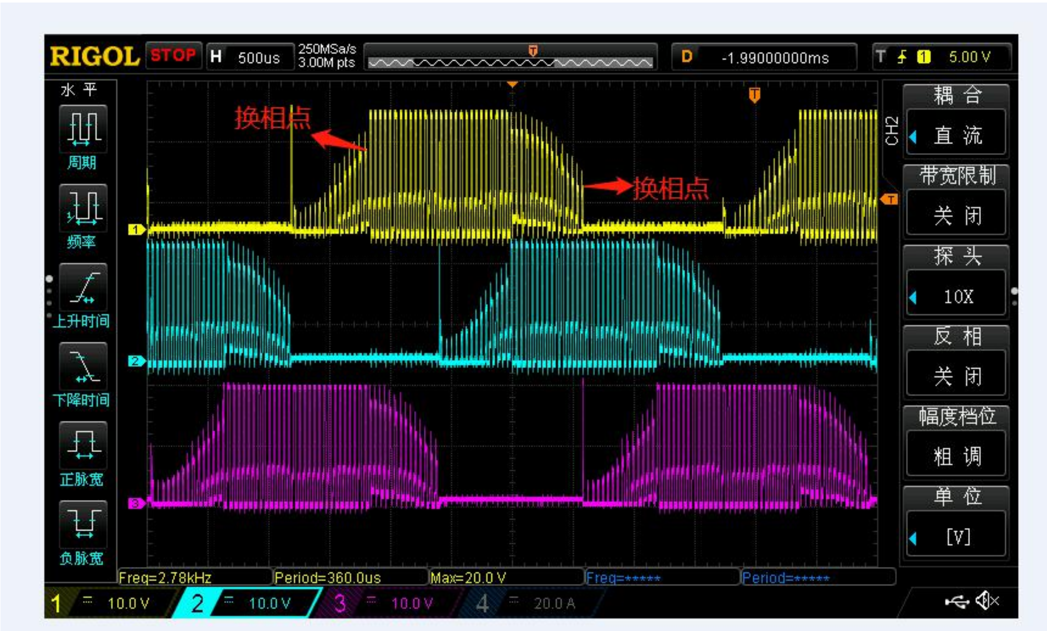
根据 duty 不同，在程序中将换相角度和续流屏蔽角度分成了两段。

```
#define Max_BLDC_High_Duty          (0.50)          // 超前角和屏蔽续流角高速区间生效Duty阈值
#define Max_BLDC_Low_Duty           (0.45)          // 超前角和屏蔽续流角低速区间生效Duty阈值
```

AddFunction.c:

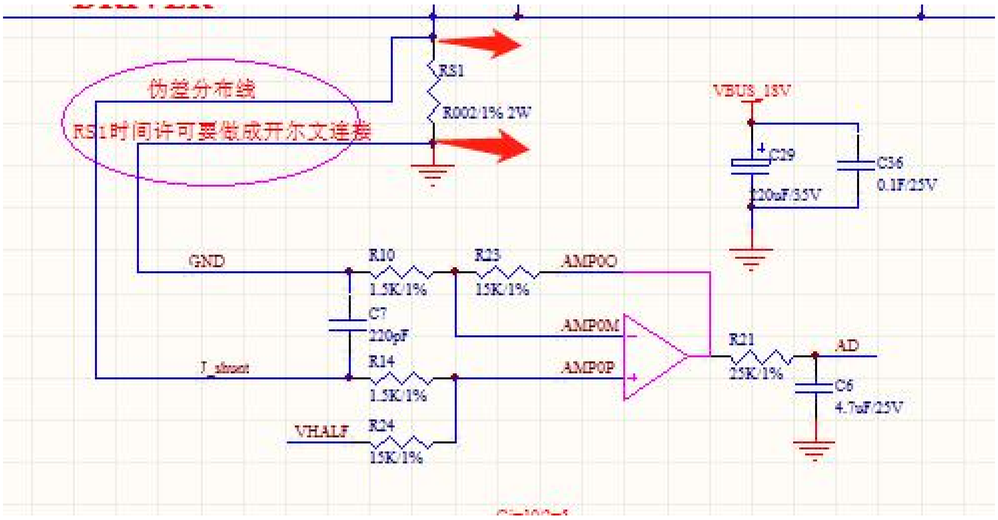
```
void StarRampDealwith(void)
{
  uint8 ANGLE_MTemp,ANGLE_DTemp;

    if((McStaSet.SetFlag.BLDCSetFlag == 4)&&(mcState == mcRun))
    {
      if(DRV_DR > HIGH_BLDC_Duty)
      {
        /*-------------------------------------------------------------------------------------------
        屏蔽续流时间（角度）设置
        -------------------------------------------------------------------------------------------*/
        ANGLE_MTemp = (uint8)(TIM1_CR1&0x7F);   续流角度

            if(ANGLE_MASK_R(38) > ANGLE_MTemp)  { TIM1_CR1 = (TIM1_CR1 &(0x80))|(ANGLE_MTemp+1);}
        else if(ANGLE_MASK_R(38) < ANGLE_MTemp)  { TIM1_CR1 = (TIM1_CR1 &(0x80))|(ANGLE_MTemp-1);}


        /*-------------------------------------------------------------------------------------------
        检测到过零点后的延迟换相时间（角度）设置
        -------------------------------------------------------------------------------------------*/
        ANGLE_DTemp = (uint8)(TIM1_CR2&0x7F);   换相角度

            if(ANGLE_DELAY_R(5) > ANGLE_DTemp)  { TIM1_CR2 = (TIM1_CR2 &(0x80))|(ANGLE_DTemp+1);}
        else if(ANGLE_DELAY_R(5) < ANGLE_DTemp)  { TIM1_CR2 = (TIM1_CR2 &(0x80))|(ANGLE_DTemp-1);}

      }
```

在这里可以直接填写需要的换相角度和续流屏蔽角度。调试这个可以改进控制的效率，将两个斜边调试的差不多对称的时候，控制的效率差不多到了最好。



4.2.10. 电流限制相关参数

```
#define Limt_KP                   _Q12(7.0)                // 外环KP
#define Limt_KI                   _Q12(0.05)               // 外环KI

#define PI_Limt_Duty_OutMAX       (1.00)
#define PI_Limt_Duty_OutMin       (0.20)

#define LimtCurt_K                (float)(123.0)           // 硬件母线电流RC滤波系数, R=10K C=1.0uF      6
#define I_LimtValue_Up            I_LimtValue1(40.5)       // 母线电流限流区间触发值 60
#define I_LimtValue_Out           I_LimtValue1(40.0)       // 母线电流限流区间稳定值 59
#define I_LimtValue_Dw            I_LimtValue1(39.0)       // 母线电流限流区间退出值 50
```



在这个程序中电流限制最关键的参数为 LimtCurrt_K,这个参数需要实际测试才能得到。测试的方法为在采样电阻两端焊接两根线，然后给板子供上电，仿真暂停，读取 Power_Currt1 这个参数值，为 0 的话，再从这两根线用另外一个电源灌入 5A 电流，然后再读取 Power_Currt1 这个的值。注意这两个过程中不能复位，在读 0 的时候是暂停不是复位。读取出来的值除以 5，就是 LimtCurrt_K 这个参数，硬件上运放后面的滤波参数不同这个值就不同，需要测试的。

4.2.11. VSP 调速开关机 duty

这里对应了 VSP 调速的开机电平、最小 duty 对应电平和最大 duty 对应电平。

```
/*电机开机、关机的设置-------------------------------------------------------------*/
 /* motor ON/OFF value */
 #define OFF_Duty                    _Q15(0.40/HW_ADC_REF)               // 关机PWM占空比，小于该占空比关机
 #define ON_Duty                     _Q15(0.40/HW_ADC_REF)               // 开机PWM占空比，大于该占空比时开机
 #define MIN_Duty                    _Q15(0.40/HW_ADC_REF)               // 速度曲线上最小PWM占空比
 #define MAX_Duty                    _Q15(4.2/HW_ADC_REF)               // 速度曲线上最大PWM占空比 4.2
```

4.3. 保护功能介绍

4.3.1. 硬件过流

硬件过流是 MCU 比较器和运放输出的比较，设置好后。过流后硬件直接关闭 PWM 输出。可以直接在下面填写需要的硬件过流电流。这个电流的选择通常是根据硬件 MOS 能承受的最大电流来选择的。

```
/*保护参数值-----------------------------------------------------------------*/
 /*硬件过流保护*/
 #define Hardware_FO_Protect          (1)                      // 硬件FO过流保护使能，适用于IPM有FO保护的场合
 #define Hardware_CMP_Protect         (2)                      // 硬件CMP比较过流保护使能，适用于MOS管应用场合
 #define Hardware_FO_CMP_Protect      (3)                      // 硬件CMP比较和FO过流保护都使能
 #define Hardware_Protect_Disable     (4)                      // 硬件过流保护禁止，用于测试
 #define HardwareCurrent_Protect      (Hardware_CMP_Protect)   // 硬件过流保护实现方式

 /*硬件过流保护比较值来源*/
 #define Compare_DAC                  (0)                      // DAC设置硬件过流值
 #define Compare_Hardware             (1)                      // 硬件设置硬件过流值
 #define Compare_Mode                 (Compare_DAC)            // 硬件过流值的来源

 /*硬件过流保护*/
 #define OverHardcurrentValue         (120.0)                  // (A) DAC模式下的硬件过流值
```

4.3.2. 软件过流

AD3 采样运放输出通过硬件 RC 滤波后的电压，做过流保护。软件过流的值是根据客户的需求填写的，一般的项目可能只需要一个软件过流，像电动工具等可能需要二级或者三级软件过流，这个可以自己根据项目本身的需求配置，使能和失能二级和三级保护。

```
/*软件过流保护*/
#define  OverSoftCurrentProtect     (1)                  // 一级软件过流保护：1--使能 0--禁止
#define  OverSoftCurrent            I_LimtValue1(20.0)   // (A) 一级软件过流值
#define  OverSoftCurrentTime        (15000)              // (ms)一级软件过流检测时间 1000

#define  OverSoftCurrentProtect1    (1)                  // 二级软件过流保护：1--使能 0--禁止
#define  OverSoftCurrent1           I_LimtValue1(34.0)   // (A) 二级软件过流值
#define  OverSoftCurrentTime1       (500)                // (ms)二级软件过流检测时间 200

#define  OverSoftCurrentProtect2    (1)                  // 三级软件过流保护：1--使能 0--禁止
#define  OverSoftCurrent2           I_LimtValue1(45.0)   // (A) 三级软件过流值
#define  OverSoftCurrentTime2       (150)                // (ms)三级软件过流检测时间  50
```

```
//软件过流保护参数设置
#define  IBUS_CALC_EN      (0)      //单位-  失能，建议关闭，不影响过流保护
#define  OVERCURRENT_EN    (0)      //单位-  过流保护
#define  OVERCURRENT       (60)     //单位A  过流值
#define  OCNMS             (4)      //单位ms 过流监测时间
```

4.3.3. 过欠压保护

最上面是过欠压的使能位，下面分别为保护电压和恢复电压，直接在括号内填写需要的保护电压和恢复电压就好了，单位为 V，软件检测 500ms 电压超过保护电压值就会保护。

```
/*过欠压保护*/
#define VoltageProtect              (1)          // 电压保护：1--使能  0--禁止
#define Over_Protect_Voltage        (21.5)       // (V) 直流电压过压保护值
#define Over_Recover_Vlotage        (21.0)       // (V) 直流电压过压保护恢复值
#define Under_Protect_Voltage       (13.0)       // (V) 直流电压欠压保护值
#define Under_Recover_Vlotage       (13.5)       // (V) 直流电压欠压保护恢复值
```

4.3.4. 过温保护

不同的硬件在做过温保护的时候需要重新配置 Tempera_Value(NTC_Value)这个公式。这里的过温保护分 MOS 管的过温保护和电池包的过温保护，需要就使能，不需要就失能就好。

```
/*Mos温度保护*/
#define  Tempera_Value(NTC_Value)          _Q15((5.0*NTC_Value/(10.0+NTC_Value))/4.5)

#define  Mos_TemperatureProtectEnable       (1)                        // MOS温度保护使能: 1--使能  0--禁止
#define  OVER_Mos_Temperature              Tempera_Value(0.766)        // MOS过温保护阈值, 根据NTC曲线设定  105度保护
#define  UNDER_Mos_Temperature             Tempera_Value(0.97)         // MOS低温恢复阈值, 根据NTC曲线设定  100度恢复
#define  Over_Mos_TemperRecoverTime        (1000)                      // MOS过温恢复时间

/*Bat温度保护*/
#define  Bat_TemperatureProtectEnable       (0)                        // 电池包温度保护使能: 1--使能  0--禁止
#define  OVER_Bat_Temperature              Tempera_Value(1.117)        // 电池包过温保护阈值, 根据NTC曲线设定  95度保
#define  UNDER_Bat_Temperature             Tempera_Value(1.275)        // 电池包低温恢复阈值, 根据NTC曲线设定  90度恢
#define  Over_Bat_TemperRecoverTime        (1000)                      // 电池包过温恢复时间
```

4.3.5. 堵转保护和缺相保护

堵转保护的时间可以在这里配置。需要失能缺相保护的时候，需要将 Enable 变为 Disable。并将下面的数值填为 0.堵转保护分为两种，一种是超过 200MS 没有发生换相，还有一种为失速保护，就是下图中对应的 211 行，如果检测到实时的转速超过了这个转速，就判断为失速保护，这个速度值的设置应该设置为电机最高转速的 2-3 倍。

缺相保护是在电机开始启动的时候，通过 IPD 检测的，这个功能需要可以配置使能，不需要的情况下就失能，并且要将下图中 215 行对应的那个值填为 0.特别需要注意的是，在拿到程序刚开始的时候就报缺相保护，并且保证相线都是连接好的情况下，就将这个功能失能，并填为 0 后，再试着启动，如没有问题了，并且需要缺相保护，就请联系原厂的工程师处理。

```
207
208 /*堵转保护*/
209 #define StallProtect               (Enable)             // 堵转保护: Enable--使能  Disable--禁止
210 #define MOTOR_STALL_COUNT          (200.0)              // 堵转检测时间, 单位毫秒: ms  最大65535/PWM_FF
211 #define MOTOR_StallSPEED_RPM       (40000.0)            //堵转失步速度, 单位RPM
212
213 /*缺相保护*/
214 #define PhaseLossProtect           (Enable)             // 缺相保护: Enable--使能  Disable--禁止
215 #define PhaseLossDetectValue       (uint16)(5000)       // ADCX_DR<<3 缺相电压检测判断阈值
216
```

5. 无 HALL 模式调试步骤

无 hall 算法，采用比较器采样直接采样反电势过零点算法。一般只需要配置分压电阻即可。中性点在 MCU 内部合成。针对凸极电机编写了高频注入检测电机初始位置软件。

5.1. 设置板子参数

电压采样、电流采样参数配置，具体需要注意的地方见上文中的 3.1.4。需要特别注意的是母线分压电阻必须和反电势的分压电阻保持一致。

5.2. 设置电机参数

电机极对数、额定转速。设置电机的极对数的原因就是为了提供给程序中计算电机的实时转速，在程序中 TIM1_BCCR 寄存器为每次 60 度换相的计数值，再根据 TIM1 的分频，可以计算出 60 度换相的时间，这样再通过极对数就可以计算出电机的实时转速。计算的方法为，如 TIM1_BCCR 中计数值为 X，TIM1 分频下来后计数频率为 6M，则当前 60 度所需要的时间就为 X/6M，前面介绍了一个电周期有六次换相（每次换相为 60 度），则一个电周期的时间就为 X/1M，有电机极对数后，就可以计算出一个机械周期所需要的时间就为（X*P）/1M，然后电机的转速为一分钟内转过的圈数，所以还需要用 60 除以（X*P）/1M，这个就为电机当前的实时转速。

5.3. 调整续流屏蔽角度和换相角度

（1）一般高速，大电流应用，续流屏蔽角度需要增加，换相角度需要减小。

（2）需要使用 4 通道示波器观测 a、U 相输出 b、U 相电流 c、换相屏蔽角度 d、换相角度来调整续流屏蔽角度和换相角度。

6. 用户接口

- 电机参数

  1、 电机极对数：必须填写正确；

  2、 基本转速：必须填写正确，一般按照最高转速的 1.5 倍填写。

- 控制板参数

  1、 母线电压采样

  需要按照硬件参数填写 Customer.h 中相应参数。

  2、 电流采样

  需要按照硬件参数填写 Customer.h 中相应参数。

- 故障情形

故障代码	故障编号	故障解释	注释	排除方法
FaultNoSource	0	无故障		
FaultSoftOVCurrent	1	一级软件过流	过流	确认硬件没问题的情况下，调整软件过流值
FaultSoftOVCurrent1	2	二级软件过流	过流	如上
FaultSoftOVCurrent2	3	三级软件过流	过流	如上
FaultHardOVCurrent	4	硬件过流	MOS 直通	1、在确认硬件没有直通，没有换相错误下更改 CMP3 的比较值或者运放的放大倍数，提高硬件过流阈值。
FaultUnderVoltage	5	欠压	欠压	看硬件参数程序中配置是否正确，可以仿真看下程序采样回来的电压是否为正确值。再调整欠压值
FaultOverVoltage	6	过压	过压	看硬件参数程序中配置是否正确，可以仿真看下程序采样回来的电压是否为正确值。再调整过压值
FaultMosOverTemperature	7	MOS 过温	MOS 过温	看硬件参数配置是否正确，然后检查下 NTC 的系数是否正确。最后再调整保护值
FaultBatOverTemperature	8	Bat 过温保护	Bat 过温保护	看硬件参数配置是否正确，然后检查下 NTC 的系数是否正确。最后再调整保护值
FaultLossPhase	9	缺相保护	缺相保护	看是否 MOS 都焊接好，线都接好，然后最后关闭程序中的缺相

				保护试试。如果通过关闭缺相保护就好了，请联系原厂工程师帮忙解决。
FaultStall	10	堵转保护	堵转保护	可能就是在运行过程中电机发生了堵转，或者失速的情况，如果为堵转，请清理堵住的杂物，如为失速保护，请检查运行过程中的电机相电压波形，看是哪里出错了。

程序中报错的变量为 mcFaultSource，可以在仿真的时候看这个参数的值，然后确定报错的错误为什么。

附录 1：电机效率优化

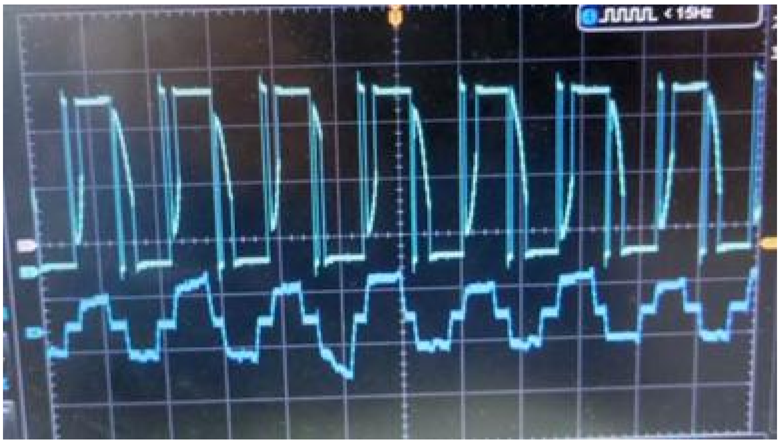
电机的效率指的是机械能和输入电能的比。电能会转化为机械能和电机发热两大块。如果电机发热严重，一般可以初步判断效率低。

一般控制需要电流相位和反电势相位重合电机效率最高。

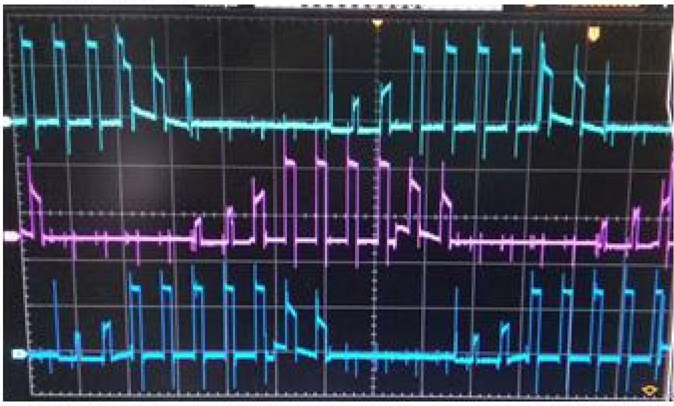
电流相位可以使用电流钳测量线电、电压相位使用电压探头测量线电压。

如图上面是电压波形，下图是电流波形。此时软件设置了超前 20 度换相。电流没有尖峰。

无 HALL 模式下调制换相角优化效率。



附录 2：PWM 调试模式



互补调制

优点：a、斩波续流经过 MOS 管不经过续流二极管，MOS 发热减小。B、减速停机会给母线充电停机快。

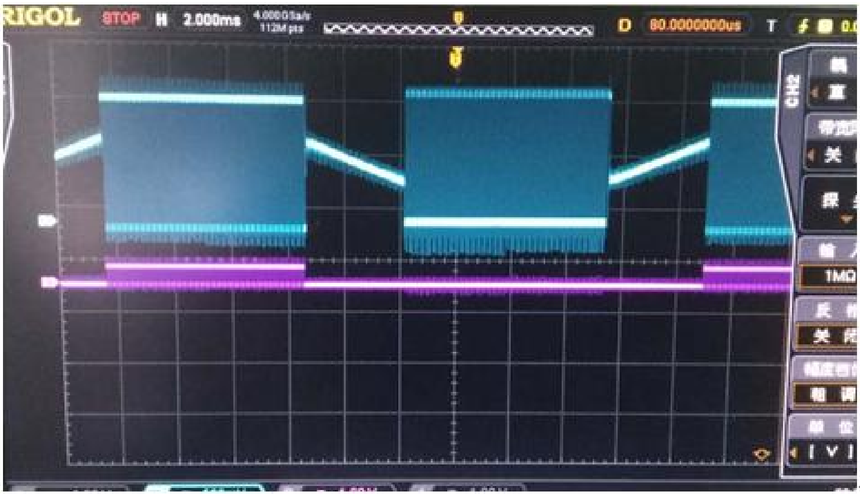
缺点：减速时母线电压会上升。

交替调制

上图：上下桥交替调制，MOS 发热均匀。斩波时的续流方向对无 HALL 检测有好处。

PWM上桥调制

上图：常规调制方法。

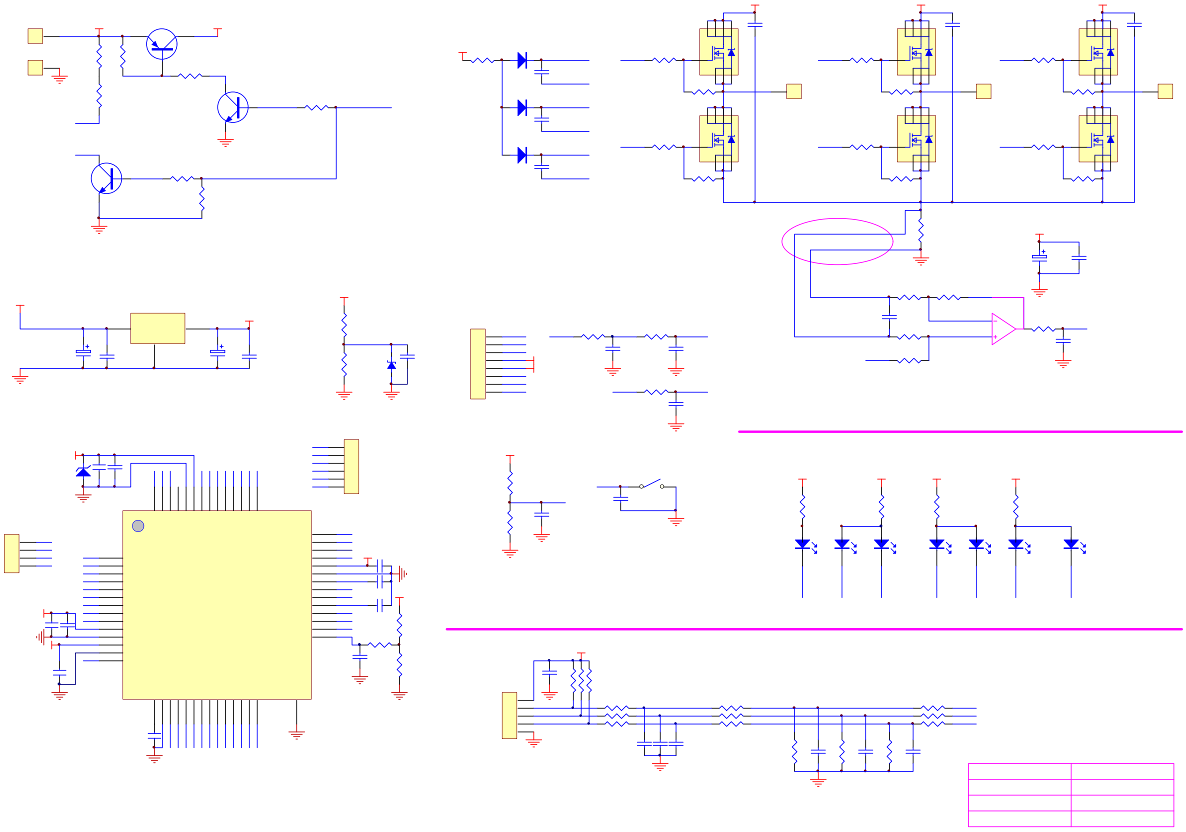


上下桥同时调制

上图：上下桥调试。

不同的调制方式可以通过配置 TIM1_DBR1-6 来实现。具体参考 datasheet 中 14.3.9.

附录 3：原理图



版权说明

峰岹科技（深圳）有限公司

地址：深圳市南山区科技中二路深圳软件园2期（11栋）203

电话：0755-86181158

传真：0755-26867715

网址：www.fortiortech.com