峰岹科技

FU68xx-无霍尔-FOC

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	Jamie.Xu	John.Luo

1. 概述

本应用笔记介绍了如何使用 FU68XX MCU 中的 FOC 控制直流无刷电机（BLDC）。

本文介绍了如何使用 FU6831 应用于硬件 FU6831-DDGJ_V1.0 电机控制开发板。

FU68XX 系列是一款集成 **8051 内核**和**电机控制引擎（ME）**的电机驱动专用芯片，8051 内核处理常规事务，ME 处理电机实时事务，双核协同工作实现各种高性能电机控制。其中 8051 内核大部分指令周期为 1T 或 2T，芯片内部集成有**高速运算放大器**、**比较器**、Pre-driver（FU6811 除外）、**高速 ADC**、**高速乘/除法器**、CRC、SPI、I2C、UART、**多种 TIMER**、**PWM** 等功能，**内置高压 LDO**，适用于 BLDC/PMSM 电机的方波、SVPWM/SPWM、FOC 驱动控制。

FU68XX 系列现共分为三款芯片：FU6811，FU6831 和 FU6818。

- FU6811 为 Gate Driver 输出；
- FU6831 为 3P3N Pre-driver 输出；
- FU6818 为 6N Pre-driver 输出。

FU68XX 芯片特性：

- 高速 8051 内核，多为 1T 或 2T 指令周期；
- 16Kx8bit Flash ROM、带 CRC 校验功能、支持程序自烧录和代码保护功能；
- 256x8bit IRAM，4Kx8bit XRAM；
- 电机运算专用协处理器；
- Gate Driver 输出；
- 单周期 16*16 位乘法器，32/32 位除法器（16 个时钟周期）；
- 两线制 ICE 在线仿真功能；
- 4 级优先级中断、16 个中断源；
- 32 个 GPIO；
- 4 个通用带 CAPTURE 可编程计数器、1 个加强型高级计数器、1 个带 BLDC 电机专用计数器；
- 内置 RTC 计数器；
- I2C/SPI/UART 接口；
- 8 通道 12 位 ADC，支持突发模式采样；
- 内置 VREF 参考；
- 内置 1/2 VDD 或 1/2VREF 参考输出；
- 内建 3 个独立运算放大器；
- 内建 4 路模拟比较器；
- 内置 24MHz±2%精准时钟。

2. 磁场定向控制原理篇

2.1. 磁场定向控制（FOC）原理

磁场定向控制 (Field Oriented Control, 简称 FOC)又称矢量控制，是通过控制变频器输出电压的幅值和频率控制三相直流无刷电机的一种变频驱动控制方法。

FOC 实质是运用坐标变换将三相静止坐标系下的电机相电流转换到相对于转子磁极轴线静止的旋转坐标系上，通过控制旋转坐标系下的矢量大小和方向达到控制电机目的。由于定子上的电压量、电流量、电动势等都是交流量，并都以同步转速在空间上不断旋转，控制算法难以实现控制。通过坐标变换之后，旋转同步矢量转换成静止矢量，电压量和电流量均变为直流量。再根据转矩公式，找出转矩与旋转坐标系上的被控制量之间关系，实时计算和控制转矩所需的直流给定量，从而间接控制电机达到其性能。由于各直流量是虚构的，在物理上并没有实际意义，因而还需通过逆变换变为实际的交流给定值。结合矢量控制框图，系统的控制过程分析如下：

1、 测量电机三相定子电流，可得到I_a和I_b。将三相电流通过 Clark 变换至两相电流I_α和I_β，其是相互正交的时变电流值。

2、 按照控制环上一次迭代计算出的电机角度，通过 Park 变换得到旋转坐标系下相互正交的电流I_d和I_q。在稳态条件下，I_d和I_q是常量。

3、 I_d的参考值控制转子磁通，I_q的参考值控制电机的转矩输出，其各自的实际值与参考值进行比较得到电流环 PI 控制器的输入。调节 PI 控制器的参数，得到V_d和V_q,即要施加到电机上的电压矢量。

4、 输入V_α、V_β、I_α和I_β，转子位置估算算法估算出新的电机角度和转速。新的电机角度可告知 FOC 算法下一个电压矢量在何处。计算出的电机转速与给定值进行比较得到误差，误差经过速度环 PI 调节器输出I_q的参考值。

5、 通过使用新的电机角度，V_d和V_q经过 Park 逆变换逆变到两相静止坐标系上。该计算将产生下一个正交电压值V_α、V_β。再采用 SVPWM 算法判定其合成的电压矢量位于哪个扇区，计算出三相各桥臂开关管的导通时间。最后经过三相逆变器驱动模块输出电机所需的三相电压。

图 2-1-1 显示了坐标变换、PI 调节器、转子位置估算以及产生 PWM 的整个过程。

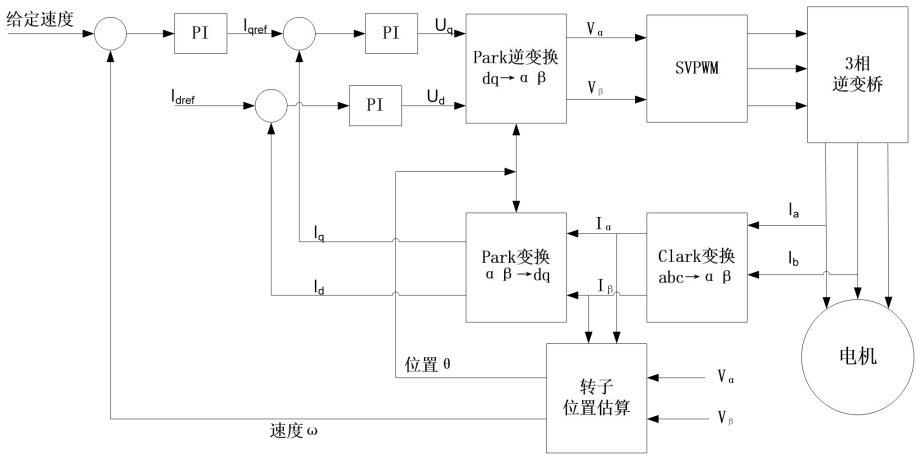


图 2-1-1　矢量控制方框图

2.1.1. 坐标变换

1)　Clark 变换

为方便计算，Clark 坐标变换将三相定子坐标系变换到两相静止坐标系中（见图 2-1-2）。具体的变换公式如式（2-1-1）所示。

$$\begin{aligned} I_a + I_b + I_c &= 0 \\ I_\alpha &= I_a \\ I_\beta &= (I_a + 2I_b)/\sqrt{3} \end{aligned} \qquad (2\text{-}1\text{-}1)$$

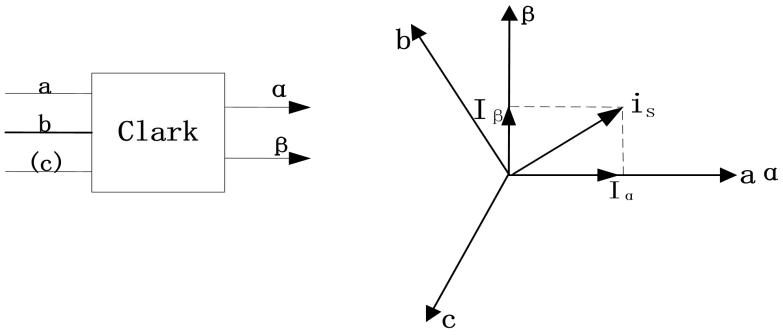


图 2-1-2　Clark 变换

注：Clark 变换采用恒幅值的变换方式。

2)　Park 变换

Park 变换将两相静止坐标系变换到两相旋转 d-p 坐标系。d-p 坐标系跟随转子以相同的电角度旋转，也称转子磁场定向坐标系。其对应的变换如图 2-1-3 所示，变换公式如式（2-1-2）所示，θ 表示转子角度。

$$I_d = I_\alpha \cos\theta + I_\beta \sin\theta$$
$$I_q = -I_a \sin\theta + I_\beta \cos\theta \qquad (2\text{-}1\text{-}2)$$

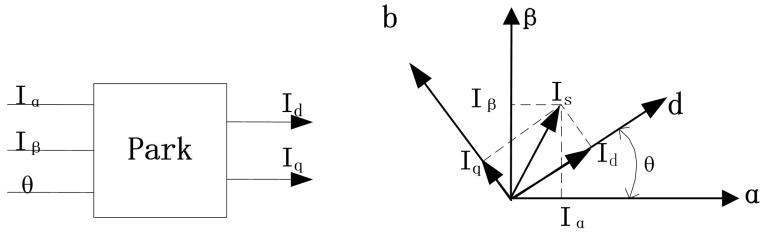


图 2-1-3　Park 变换

3)　Park 逆变换

d、q 轴电流环经过 PI 迭代后，可获得旋转 d-q 坐标系的电压矢量的两个分量。这时需要经过逆变换将其重新转换为三相电机电压。首先将从两相旋转坐标系变换到两相静止坐标系上，该变换使用了 Park 逆变换，如式（2-1-3）所示。

$$U_\alpha = U_d \cos\theta - U_q \sin\theta$$
$$U_\beta = U_d \sin\theta + U_q \cos\theta \qquad (2\text{-}1\text{-}3)$$

2.1.2. PI 调节器

整个控制过程中，使用三个 PI 环分别控制相互影响的转子转速、磁通、转矩三个变量。因考虑 PI 控制器在输入偏差较大的情况下，容易出现饱和现象，因此在控制器输出处进行了限幅处理。

$$u(kT) = u(kT-T) + Kp[e(kT) - e(kT-T)] + Ki * e(kT)$$
$$\text{if}(u(kT) > \text{Outmax})$$
$$u(kT) = \text{Outmax} \qquad (2\text{-}1\text{-}4)$$
$$\text{else if}(u(kT) < \text{Outmin})$$
$$u(kT) = \text{Outmin}$$

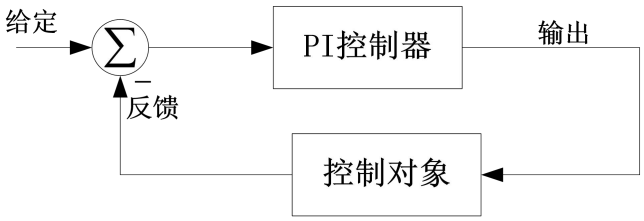


图 2-1-4　PI 调节器

2.1.3. 转子位置估算

1）电机模型

通过使用一个直流电机模型来估算电机的位置，该电机模型由绕组电阻、绕组电感和反电动势来表示，如图 2-1-5 所示。

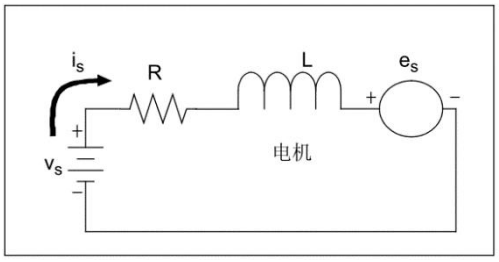


图 2-1-5　电机模型

无传感器控制技术通过估算电机的位置来计算 FOC 所需的换相角度，从而实现 FOC 算法。通常情况下，电机的位置和速度可根据测量电机的电流和电压估算出。在估算 BLDC 位置的算法中，滑膜观测器(SMO)是一种简单且性能较好的算法。在不影响电机控制性能的前提下，将 BLDC 的数学模型进行简化，用绕组电阻、绕组电感和反电动势来表示，如式(2-1-5)所示。

$$V_s = Ri_s + L\frac{d}{dt}i_s + e_s \qquad (2\text{-}1\text{-}5)$$

其中，i_s 为电机电流矢量，e_s 为反电动势矢量，V_s 为输入电压矢量，L 绕组电感，R 绕组电阻。

在数字域中，该方程式可表示为：

$$i_s(n+1) = (1 - T_s * \frac{R}{L})i_s(n) + \frac{Ts}{L}(V_s(n) - e_s(n)) \qquad (2\text{-}1\text{-}6)$$

该公式是电机的一个数字化模型，称为电流观测器。位置和速度估算器是基于电流观测器而构建的。该数学化模型对硬件使用了软件表示方式，为了使测量电流和估算电流相匹配，数字化电机模型需要使用闭环控制来进行校正，如图所示。

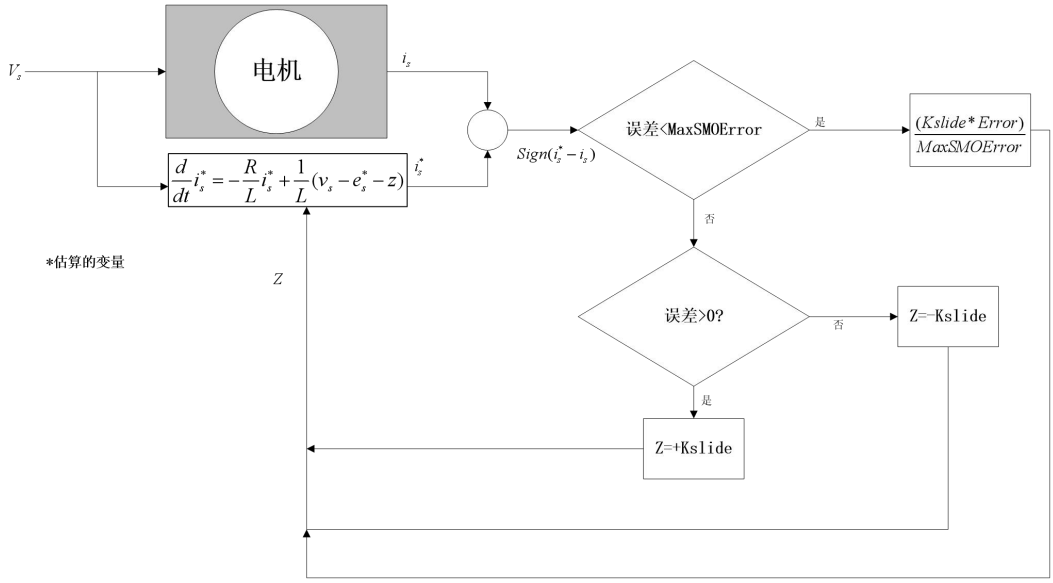


图 2-1-6　电流观测器框图

灰色部分为实际的电机系统，另一个是通过软件方式模拟的数字化电机系统，两个系统中使用相同的输入电压(V_s),在假设数字化模型中的反电动势(e_s^*)与电机的反电动势(e_s)相同的情况下，使用模型中的估算电流(i_s^*)来匹配测量电流(i_s)。

对数字化模型进行补偿后，电机的输入电压和电流与数字化电机模型一致，接着通过对校正因子(Z)滤波估算反电动势(e_s^*)。每个控制周期对反电动势的估算值(e_s^*)进行更新，并反馈给数字化电机模型。e_α和e_β值是e_s的矢量分量，可用于估算θ^*。根据反电动势矢量分量计算出的反余切计算 theta 的公式，如式(2-1-7)所示。

$$\theta = arc\tan(e_\alpha, e_\beta) \qquad (2\text{-}1\text{-}7)$$

由于在计算 theta 期间使用了滤波函数，所以在使用计算得到的角度给电机绕组通电前需对相位进行补偿。其补偿量取决于 theta 的变化速率或电机的速度。Theta 补偿由以下两步组成：

1、 通过未补偿的 theta 来计算电机的速度；

2、 对计算得到的速度进行过滤，并计算补偿量，如图 2-1-7 所示。

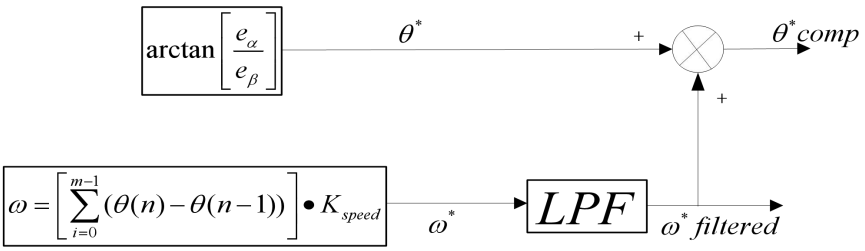


图 2-1-7　速度计算框图

2.2. 电流采样原理

FU68XX 支持双电阻和单电阻采样，用户只需要设置 68 系列 FOC 模块中的 FOC_CR1 寄存器的 CSM 位，CSM 设置 1 为双电阻采样，CSM 设置 0 为单电阻采样。

注：

FU6801 系列支持单电阻采样和双电阻采样；

FU6802、FU6803、FU6804 系列支持单电阻采样、双电阻采样和三电阻采样；

1) **双电阻采样：**

采样时序是在三个下桥臂同时导通时，采集其中两相电流，68 系列是默认采样 U,V 两相电流，并计算出 W 相电流。

推导公式：

$$i_a + i_b + i_c = 0$$

$$i_c = -i_a - i_b$$

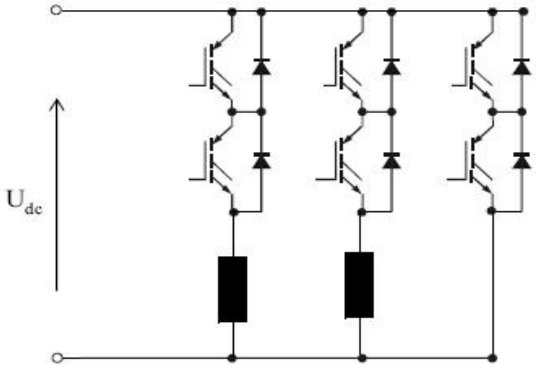


图 2-2-1 双电阻框图

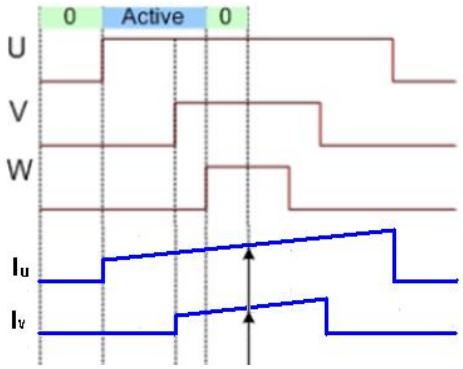


图 2-2-2 双电阻采样时序

注：

1. 双电阻采样需采样完 Ia 后再采 Ib，导致采样时间较长，Ts = Ia(采样时间+转换时间) + Ib(采样时间)。采样时间最小值是 3 clock(AD 模块时钟是 12MHz)，转换时间 14 clock。
2. 双电阻采样启始采样点是基于 PWM 计数的零点开始采样，故 Ts 最小值需乘以 2。
   即： Ts = ((3+14)/12MHz + 3/12MHz)*2 = 3.33us。
3. PWM 最大输出占空比为 Tpwm = T – Ts。(Tpwm 为 PWM 输出占空比，T 为 PWM 周期时间，Ts 为 Ia+Ib 最小采样时间)。
4. 调试过程中，Tpwm 设置不合适会导致 Ia 或 Ib 采样出错，用户可调节 FOC_TRGDLY 的值来设置采样点。

FOC_TRGDLY 设置：假设 MCU 时钟为 24MHz(41.67ns)，FOC_TRGDLY = 5，则延迟 41.67*5 = 208ns；FOC_TRGDLY 为-5，则提前 208ns.

5. 经试验&调试发现，双电阻 Tpwm 最大输出占空比为 93%~95%之间，FOC_TRGDLY 典型值设置为-7。PWM 输出最大占空比和 FOC_TRGDLY 具体值与客户设计的产品板功率器件开关噪音有关。

   公式： $Tpwm = Us = \sqrt{Ud^2 + Uq^2}$

2) 单电阻采样：

采样时序是通过分时采集一个周期内的两次电流(一个或者两个下桥臂导通)，重构出电机三相电流。

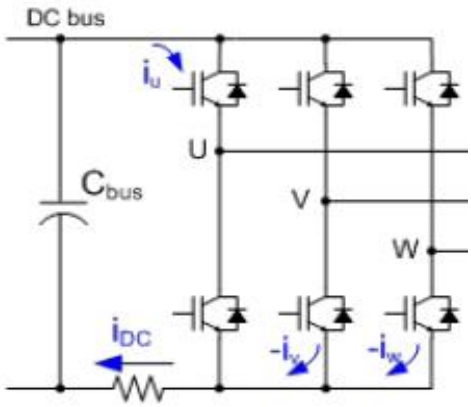


图 2-2-3 单电阻框图

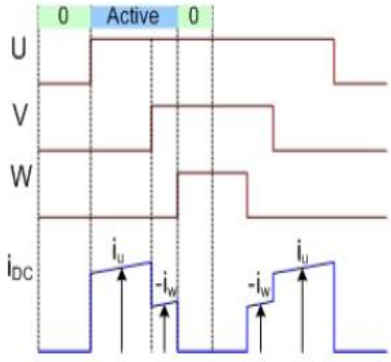


图 2-2-4 单电阻采样时序

注：

1. 单电阻采样需要设置最小采样窗口时间，即设置寄存器 FOC_TSMIN，推荐值 FOC_TSMIN = FOC_DTR + Tsmin (Tsmin：最小采样窗口时间典型值 1us)。FOC_DTR 为死区时间。
2. 单电阻 PWM 输出最大占空比 Tpwm 值可以设置为 100%。

   公式： $Tpwm = Us = \sqrt{Ud^2 + Uq^2}$

2.3. 五段式和七段式 SVPWM 输出原理

SVPWM 是空间矢量脉宽调制(Space Vector Pulse Width Modulation)的简称。其理论基础是平均值等效原理，即在一个开关周期内通过对基本电压矢量加以组合，使其平均值与给定电压矢量相等。在某个时刻，电压矢量旋转到某个区域中，可由组成这个区域的两个相邻的非零矢量和零矢量在时间上的不同组合来得到。两个矢量的作用时间在一个采样周期内分多次施加，从而控制各个电压矢量的作用时间，使电压空间矢量接近按圆轨迹旋转，通过逆变器的不同开关状态所产生的实际磁通去逼近理想磁通圆，并由两者的比较结果来决定逆变器的开关状态，从而形成 PWM 波形。

其中，以第一扇区为例：

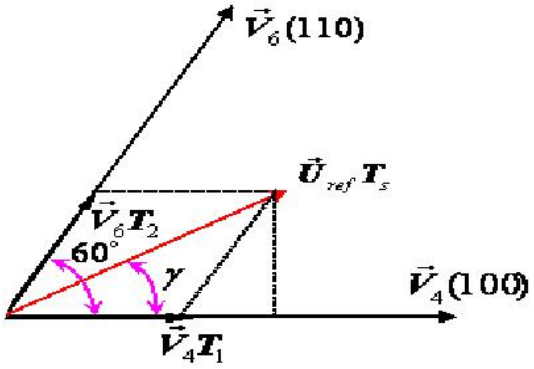


图 2-3-1 第一扇区矢量图

根据图，可得出：

$$\begin{cases}\vec{U}_{ref}T_S=\vec{V}_4T_1+\vec{V}_6T_2\\ T_0=T_s-T_1-T_2\end{cases}$$

$$\Rightarrow\begin{cases}\vec{u}_{ref\alpha}T_S=T_1\vec{V}_4+T_2\vec{V}_6\cos 60^\circ\\ \vec{u}_{ref\beta}T_S=T_2\vec{V}_6\sin 60^\circ\end{cases}\Rightarrow\begin{cases}T_1=\dfrac{1}{2}(\sqrt{3}u_{ref\alpha}-u_{ref\beta})\dfrac{\sqrt{3}T_s}{V_{dc}}\\ T_2=u_{ref\beta}\dfrac{\sqrt{3}T_s}{V_{dc}}\end{cases}$$

$T_0 = T - T_1 - T_2$

2.3.1. 七段式 SVPWM

我们以减少开关次数为目标，将基本矢量作用顺序的分配原则选定为：在每次开关状态转换时，只改变其中一相的 开关状态。并且对零矢量在时间上进行了平均分配，以使产生的 PWM 对称，从而有效地降低 PWM 的谐波分量。

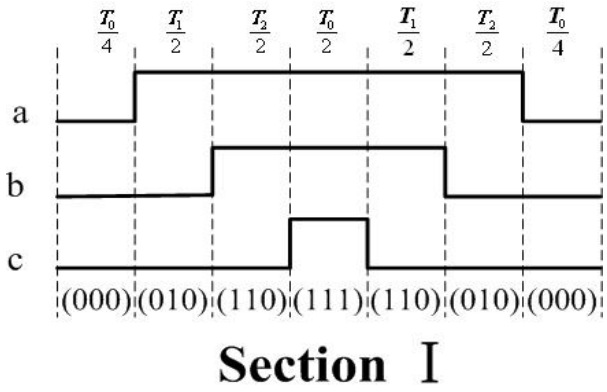


图 2-3-2 Section I 第一扇区七段式三相 PWM 波

优点：

电流波形更接近圆形，谐波干扰少；

缺点：

开关次数多，开关损耗大；

7 段式 PWM 输出最高占空比 93%~95%。

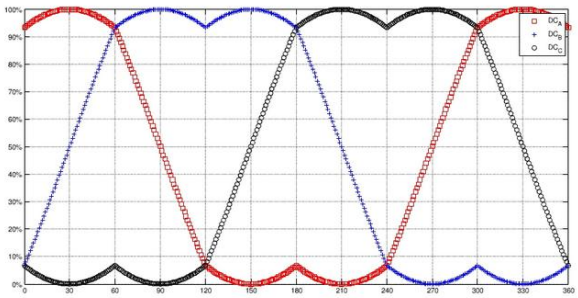


图 1-18，七段式 SVPWM 占空比(m = 1)

图 2-3-3 七段式相电压波形

2.3.2. 五段式 SVPWM

对 7 段而言，发波对称，谐波含量较小，但是每个开关周期有 6 次开关切换，为了进一步减少开关次数，采用一相开关在每个扇区状态维持不变的序列安排，使得每个开关周期只有 4 次开关切换，但是会增大谐波含量。具体序列安排见下表。

1. 在五段式 SVPWM 中，有一相的相电压在一个 PWM 周期内不发生翻转，可以为恒零（占空比输出为 0%），也可以恒 1（占空比为 100%）。FU68XX 系列芯片采用恒零模式。

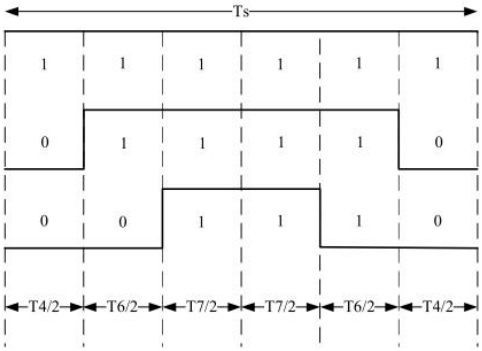


图 2-3-4 Section I 第一扇区五段式三相 PWM 波

2. 五段式 SVPWM 在一个 PWM 周期中也有不同的波形方式，如边缘对齐和中心对齐两种模式，FU68XX 系列芯片采用中心对齐模式。
3. SVPWM 波形以对称方式时，PWM 作用时间分为 5 段，这也是五段式 PWM 命名的来源

五段式 SVPWM

优点：

一个 PWM 周期开关次数减少 1/3，功率器件发热减少；

PWM 最大输出在占空比比七段式高，最大可以达到 97%。

缺点：

电流的谐波干扰更大。

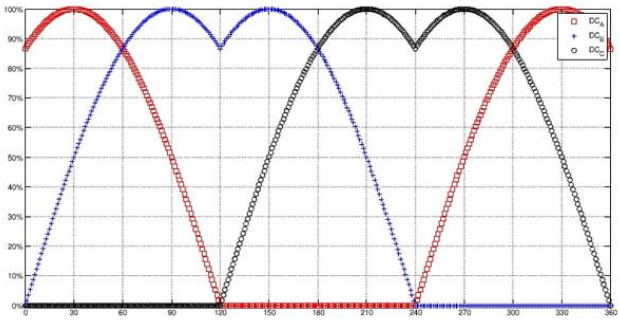


图 1-16，某相低高占空比(m = 1)

图 2-3-5 五段式相电压波形

3. FOC 启动调试篇

3.1. 硬件参数配置

3.1.1. 电机参数测量

不同电机的极对数、电阻、电感、反电动势都不同，因 FOC 计算受这些参数影响很大，调试时需写入调试电机的参数。电机参数测量方法如下：

1、 **极对数 P**：电机设计时需给出的参数；

2、 **相电阻 Rs**：万用表测量电机两相线电阻 RL，相电阻 Rs=RL/2；

3、 **相电感 Ls**：电桥测 1KHz 频率下的两相线电感 LL，相电感 Ls=LL/2；

4、 **反电动势常数 Ke**：示波器的探头接电机的一相，地接电机另外两相中的某一相，转动负载，测出反电动势波形。取中间的一个正弦波，测量其峰峰值 Vpp 和频率 f。计算公式如下：

$$Ke = 1000 * P * \frac{Vpp}{2 * 1.732 * 60 * f}$$

P 为电机极对数。

示例，测量反电动势波形如下：

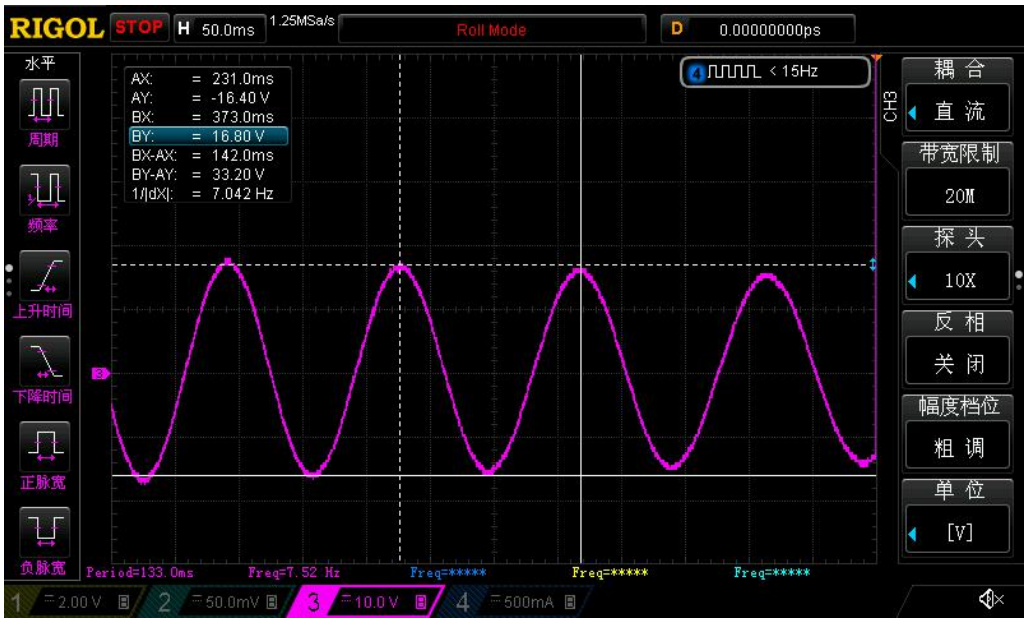


图 3-1-1 反电动势波形

测量峰峰值 Vpp 为 33.2V，频率 f 为 7.042Hz，极对数为 4，则：

反电动势 $Ke = 1000 * 4 * \frac{33.2}{2*1.732*60*7.042}$=90.73

配置文件：customer.h

参数如下：

```
/*电机参数值-------------------------------------------------------------------*/
/*motor Parameter*/
#define     Pole_Pairs                  (4.0)                       // 极对数
#define     RS                          (11.6)                      // 相电阻 , ohm
#define     LD                          (0.022)                     // D轴相电感 , H
#define     LQ                          (0.022)                     // Q轴相电感 , H
#define     Ke                          (90.73)                     // 反电动势常数, V/KRPM
```

3.1.2. 硬件参数配置

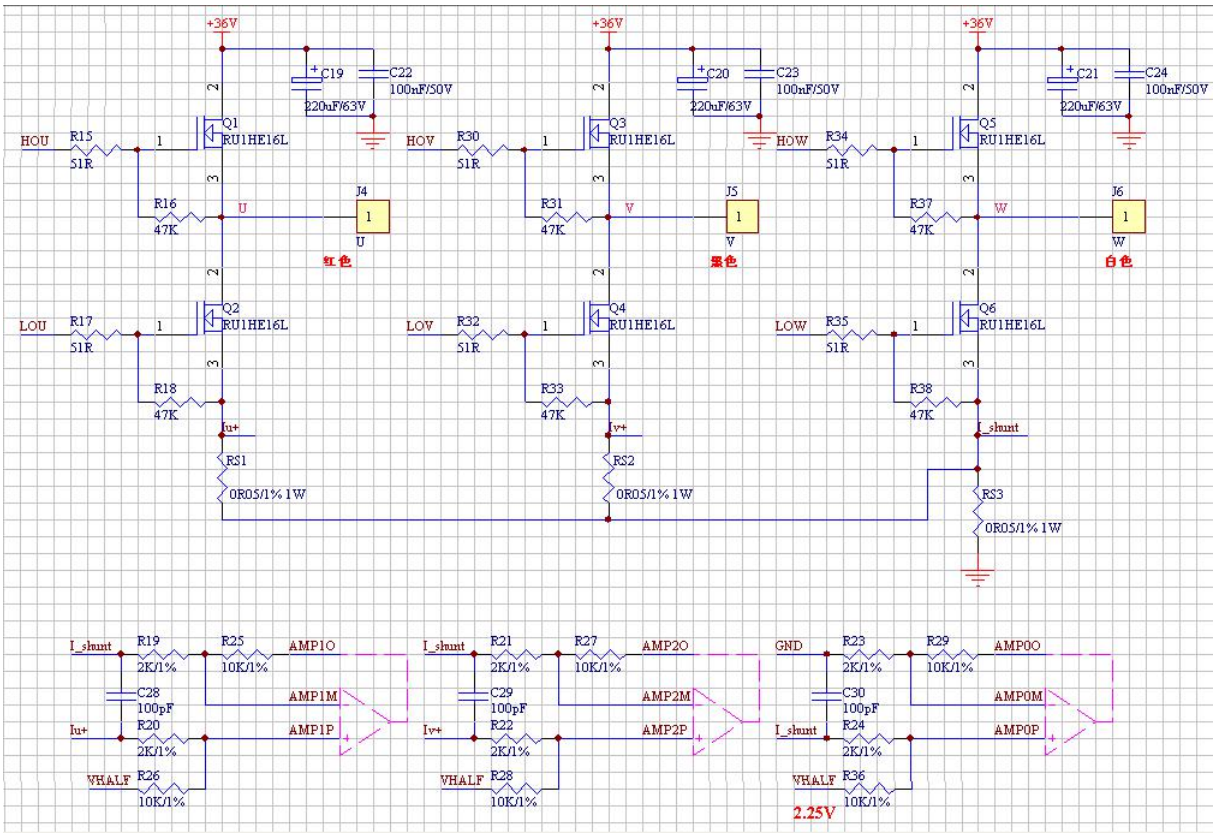


图 3-1-2 驱动原理图

在软件调试过程中，通常要遇到采样电阻选择和运算放大倍数不合适导致电机运行性能不佳，选择一个合适的采样电阻和放大倍数值很关键。这个两个值相辅相成，相互影响，一起确定。

按照理论，采样电阻越大越好，放大倍数越小越好，最好是情况是，采样到的值，MCU 可以直接进行采样，不需要放大。在电路中，采样电阻变为 2 倍，相同电流的情况下，功率变成 4 倍，同时功耗也会成变成 4 倍，价格相应的要增加很多，所以采样电阻不能放的很大，要取一个合适的值。同理，采样电阻很小，运算放大倍数很大，这样干扰也会跟着放大，导致采样不准确。通常我们会按照理论上的最大电流选择一个合适的电阻，阻值一般情况下要低于 0.5 欧姆，封装要在 1206、2512 等中选择。放大倍数一般为 3-15 倍，不建议再增大、减小，且 AMP1 和 AMP2 放大倍数一定要一

样，AMP0 放大倍数可以不同于 AMP1 和 AMP2。当使用双电阻采样的时候，AMP1 和 AMP2 用于采集电机相电流用于角度计算，AMP0 用于采集母线电流用作过流保护。当使用单电阻采样的时候，可以没有 AMP1 和 AMP2 这两路运放，AMP0 既做相电流采样也做母线电流采样，那么在选择放大倍数和采样电阻的时候优先以相电流的采样做考虑然后再计算母线电流，根据母线电流的大小调整 CMP3 比较器的阻值。

注意：

AMP1 采样 U 箱电流，AMP2 采样 V 相电流，不能交换，假如画板的时候不慎交换，只能使用单电阻采样先调试。

如上图，客户要求功率在 80W，额定电压为 36V，最低电压 30V，按照极值计算，最大额定电流为 80/30=2.67A，最大电流为 2.67*1.414=3.77A，采样电阻采用 0.05Ω，最大功率为 0.71W，考虑到高温至少要降额使用 70%，所以最好是选自 1W 或者以上功率的采样电阻，综合价格等因素，这里选择 0.05Ω/1W 的采样电阻。在采样电阻基本确定的情况下，放大倍数一定留有足够的余量，通常的余量为最大电流的 2 倍以上，这里的最大电流为 3.77A，按照 4A 计算，一般情况下，我们要选择可以采样至少 8A 的范围。可采样范围为 2.25±2.25V，2.25/0.05/8=5.6 倍，这里选择 5 倍，所以可采样电流范围为 9A，满足要求。

根据以上硬件参数，需要相应配置 customer.h 中的和硬件相关的参数。另外，以上参数是理论计算参数，在调试过程中可以根据实际测试数据进行调整，比如实测采样电阻发热过大，可以适当降低采样电机的阻值然后同比例放大运放的放大倍数。

3.1.3. 电流采样参数

```
#define   HW_RSHUNT                 (0.05)
```
采样电阻阻值（单位：欧姆）

```
#define   HW_AMPGAIN                (5.0)
```
运放放大倍数，详细描述如下：

电路上运放放大倍数受采样电阻配置影响，改变以下语句可配置成单电阻采样（Single_Resistor）或双电阻采样（Double_Resistor）

```
#define   Shunt_Resistor_Mode       (Double_Resistor)
```

当配置成双电阻采样（Double_Resistor）时，运放放大倍数取决于芯片内部运放 AMP1 AMP2 外围电路配置，如上图，最终放大倍数为 R25/R19=5 倍。

此时芯片内部运放 AMP0 可用于母线电路采样，可配合芯片内部比较器 CMP3 用于硬件过流保

护，AMP0 电路配置如下图。

CMP3 的正端芯片内部与 AMP0 输出连接，负端可外部设置参考电压，当 AMP0 输出电压比外部参考电压高时触发 CMP3 上升沿中断，硬件过流详细配置见第四章 FOC 保护篇。

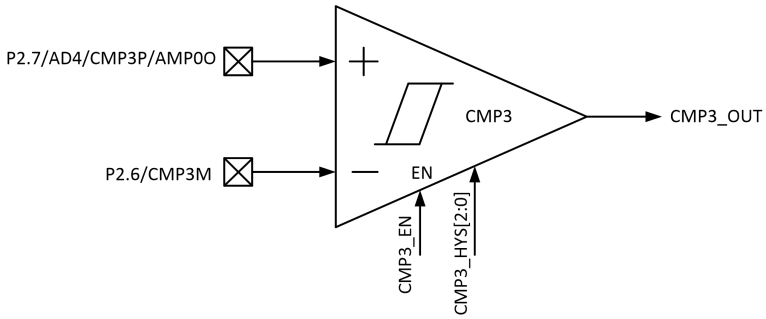


图 3-1-3 比较器电路电路

当配置成单电阻采样（Single_Resistor）时放大倍数取决于 AMP0 外部电路配置，如下图，此时运放放大倍数为 R8/R3=6 倍，使用单电阻采样时无需 AMP1 AMP2 功能。

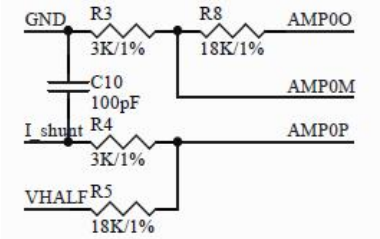


图 3-1-4 母线电流放大电路

```
#define  HW_ADC_REF          (4.5)
```

ADC 基准电压，目前程序上使用芯片内部 VREF = 4.5V 参考基准。

芯片的参考电压可以由芯片引脚 VREF 的供电电压提供也可以由芯片内部自己产生的参考电压提供，通过配置程序中 void ADC_Init(void)函数中 ADC_CFG 寄存器可以选择参考电压来自 IO 口还是芯片内部。

ADCREF = 00：选择 VDD5 作为 ADC 参考电压

ADCREF = 01：选择外部 VREF 作为 ADC 参考电压

ADCREF = 10：选择内部 VREF 作为 ADC 参考电压

ADCREF = 11：选择内部 VREF 作为 ADC 参考电压，同时参考电压输出到 VREF 管脚。

```
SetBit(ADC_CFG, ADCREF1, 1);
SetBit(ADC_CFG, ADCREF0, 1);
```

当选择 ADC 的电参考压为外部 VREF 前，应将 P3.5 配置为模拟形式（P3_AN[5]=1）；

当选择参考电压来自芯片内部的时候，如果是 6801 系列需要将寄存器 VREF_CR 的 VREFEN 使能位为 1，为 ADC 提供参考电压。6801，6802，6803 三代芯片的配置模式存在差异，详细需要参考对应芯片手册的 ADC 和 VREF 参考电压章节，重点关注 ADC_CFG 寄存器的介绍。芯片一共提供 5.0V，4.5V，4.0V，3.0V 几种电压模式供选择，需要通过配置 VREF_VHALF_CR 寄存器配置相关电平。

```
VREFConfig(VREF4_5, Enable);
```

VREF5_0：内部 5V 参考电压
VREF4_5：内部 4.5V 参考电压
VREF4_0：内部 4V 参考电压
VREF3_0：内部 3V 参考电压

注意：

1.对于 FU6812s 芯片参考电压只能选择 VDD5。

2.FU6801 系列 MCU 电压经过 LDO 才会给到 ADC 模块，而 LDO 存在压降，所以选择 5.0V 作为 ADC 参考电压的时候，MCU 的外部电压必须大于 5.3V。FU6802 与 FU6803 系列 MCU 中 ADC 模块电压直接取自芯片 VCC5，所以芯片供电电压只要到达 5V 即可。

3.1.4. 最大采样电压

在 FOC 的 SVPWM 模块中需采集母线电压进行计算，而在高低压应用中，因电源电压与 MCU 的 ADC 最大采样电压不等，需根据实际情况将母线电压用分压方式来处理，原则是芯片电压采集端口(注意母线电压采样在 6802，6803 系列中需要固定为 AD2 端口，不可以变动到其他 AD 口)的电压必须能侦测到母线上面的最大电压。假设电路工作电压是 AC220V，那么电路在过压的情况下面可以到 AC300V，那么母线上面最大的直流电压就是 300*1.414=424V，那么经过分压电路之后电压采样 AD 口的电压不可以超过上面所配置的 HW_ADC_REF 电压。

注意：

对于 6801 系列芯片电压 AD 采样口电压不能超过 HW_ADC_REF*0.7（普通非电压采样口最大采样电压依然为 HW_ADC_REF，电压采样口主要是用于内部 FOC 运算限制了 ADC 采样的最大格式），在 6802 和 6803 系列芯片就没有这样的要求。

那么配置电路中的分压电阻 RV1，RV2，RV3(低压情况下省去 RV2，即 RV2=0)，则对应的缩小倍数和最大采样电阻计算方式如下：

缩小倍数 $RV = \frac{RV1+RV2+RV3}{RV3}$

最大采样电压为 Vsmax=RV*Vs

示例：

```
#define   RV1              (470.0)
#define   RV2              (470.0)
#define   RV3              (6.8)
#define   HW_ADC_REF       (4.5)
```

则最大采样电压为

FU6801：$Vsmax=\frac{470+470+6.8}{6.8} * 4.5 * 0.7 = 438V$

FU6802&FU6803：$Vsmax=\frac{470+470+6.8}{6.8} * 4.5 = 626V$

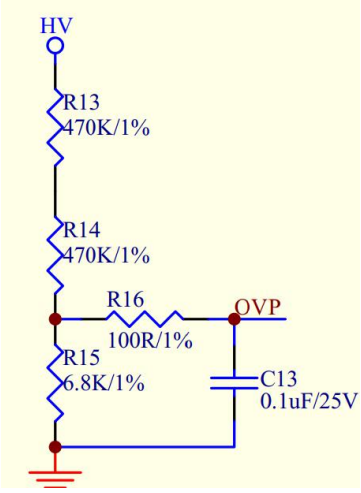


图 3-1-5 母线分压电路

3.1.5. 电流基准计算

根据硬件板上的采样电阻 Rsample，运放放大倍数 Amp，MCU 的 ADC 参考电压 Vs（HW_ADC_REF），可计算电流基准、最大采样电流、最小采样电流。在调节过程中，电机的电流不能超过最大采样电流，也不能小于最小采样电流。若出现超出情况，需根据实际情况重新配置采样电阻和运放放大倍数。采样电阻的功率选择上，采用的原则为 $P = I^2 Rsample < Pmax$。

电流基准 $Ibase = \frac{Vs}{Rsamle*Amp}$，最大采样电流 $Ismax = \frac{Ibase}{2}$，最小采样电流 $Ismin = -\frac{Ibase}{2}$。

示例：

Rsample（HW_RSHUNT）=0.5Ω，

Amp（HW_AMPGAIN）=4，

Vs（HW_ADC_REF）=4.5V，

则 $Ibase = 2.25A, Ismax = 1.125A, Ismin = -1.125A$。

3.1.6. 驱动输出配置

1) 6801 驱动模式

```
#define  DriverMode                    (Driver_6N)
```

高压方案使用 FU6811，FU6811 没有集成 Pre-driver，需要外加 Pre-driver 启动 N 管，驱动模式为 Driver_6N，即外部使用 6 个 NMOS；低压一般使用 FU6831，内部集成 3P3N 的 pre-driver，驱动模式为 Driver_3P3N，即外部上桥使用 3 个 PMOS，下桥使用 3 个 NMOS；中压一般使用 FU6818，内部集成 6N 的 pre-driver，驱动模式为 Driver_6N，即外部使用 6 个 NMOS。

2) 6802，6803 驱动模式

```
#define  High_Level                    (0)
#define  Low_Level                     (1)
#define  UP_H_DOWN_L                   (2)
#define  UP_L_DOWN_H                   (3)
#define  PWM_Level_Mode                (High_Level)
```

中压方案选择 FU6861，内部集成 6N 的 pre-driver FD6287，根据芯片数据手册，输入逻辑写着（HIN，LIN*），所以驱动模式为 UP_H_DOWN_L（上高下低）。高压方案使用 FU6812 和 FU6813，FU6812 和 FU6813 没有集成 Pre-driver，需要外加 Pre-driver 才能驱动 N 管，需要根据使用的 Pre-driver 选择相关的宏定义。选择 High_Level 或 Low_Level 逻辑为若 pre-driver 输入端输入高电平若输出端也输出高电平则为高有效（High_Level）；若 pre-driver 输入端输入低电平若输出端也输出高电平则为低有效（Low_Level），概括为当 pre-driver 输出端为高电平时对应的输入端电平就是相应的有效电平模式。下面我司几种 pre-driver 的电平选择：

芯片型号	驱动模式	芯片型号	驱动模式
FD6636S	Low_Level	FD2606S	High_Level
FD6287T	UP_H_DOWN_L	FD2501S	High_Level
FD6288T	High_Level	FD2103S	UP_H_DOWN_L
FD6288Q	High_Level	FD2203S	UP_H_DOWN_L
FD6187T	High_Level		

3) 有效电平：

```
#define  PWM_Level_Mode                (High_Level)
```

该定义只有在 6801 系列芯片中才会出现，高压方案外部使用 FD2606 时选择高电平有效（High_Level），使用 FD6536 时选择低电平有效（Low_Level），选择 High_Level 或 Low_Level 逻辑为若 pre-driver 输入端输入高电平若输出端也输出高电平则为高有效（High_Level）；若 pre-driver 输入端输入低电平若输出端也输出高电平则为低有效（Low_Level），概括为当 pre-driver 输出端为高电平时对应的输入端电平就是相应的有效电平模式。6818 和 6818 选择高电平有效（High_Level）。

4) SVPWM 模式：

双电阻采样时可配置为 5 段或 7 段 SVPWM，单电阻时固定为 7 段 SVPWM。使用 5 段式 SVPWM 时 MOS 管发热会较 7 段 SVPWM 小，使用 7 段 SVPWM 时电流波形会较 5 段 SVPWM 好且整机干扰较小。

```
#define  SVPWM_5_Segment      (0)                    //5 段 SVPWM
#define  SVPWM_7_Segment      (1)                    //7 段 SVPWM
#define  SVPMW_Mode           (SVPWM_5_Segment)      //SVPWM 模式选择
```

3.2. FOC 的调试步骤

3.2.1. FOC 工作流程图与程序执行流程图

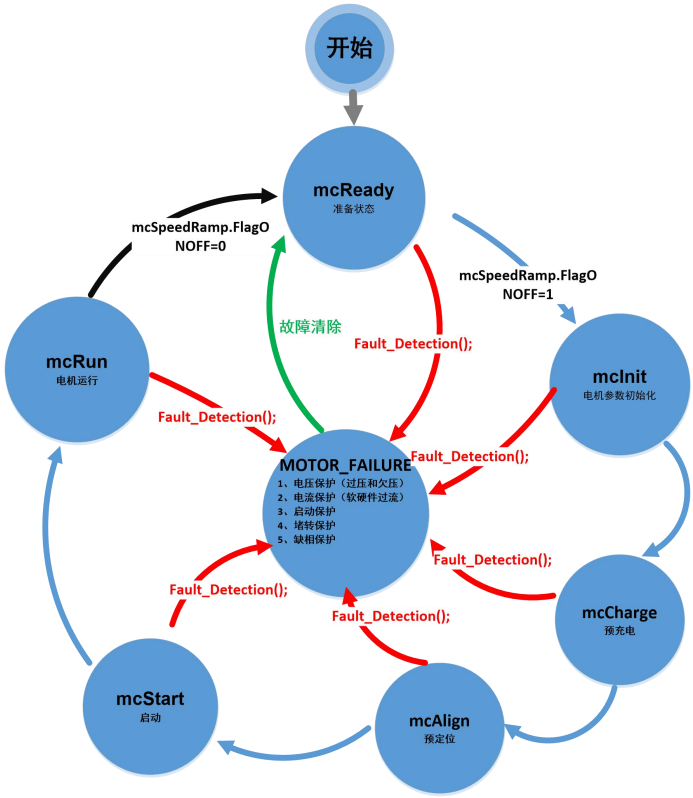


图 3-2-1 FOC 工作流程图

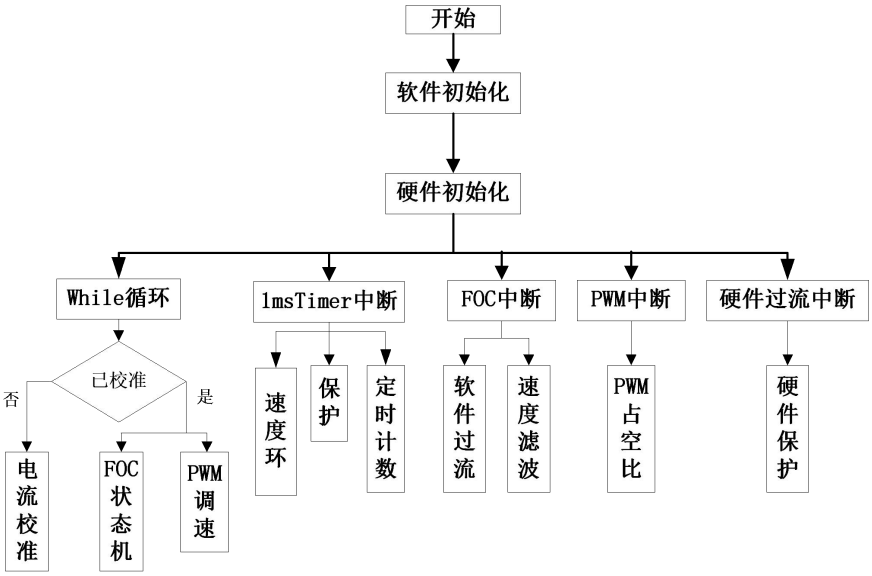


图 3-2-2 程序执行流程图

3.2.2. FOC 初始化

FOC 在运行前需根据对应的电机和硬件原理图配置初始化的一些参数和寄存器，具体如下：

1) 电机参数，速度基准，最大最小转速；
2) 采样电阻、运放放大倍数、基准电流；
3) 母线分压电阻；
4) MCU 的初始化，如时钟，变量初始化，硬件 I/O 口初始化等；
5) 电流采样的校准；
6) 状态初始化，配置 FOC 寄存器，如 PWM 有效电平、估算算法模式、单双电阻模式等，具体可参考程序。

FOC 初始化结束后，一直处于 mcInit 态，此时 MCU 的六路 PWM 波没有输出。当有启动信号时，进入下一个状态，即预充电 mcCharge 状态。

3.2.3. 预充电或者 IPM 验证

预充电采用的方式是 U 相上下桥臂先互补输出，再接着 U、V 相一起互补输出，最后 U、V、W 三相都互补输出。上下桥臂互补输出中实际有效的是下桥臂的电平输出，同时充电时间可调。若在电机常规运行模式下，预充电时间到，主程序进入预定位状态。若电机性能有顺逆风要求时，主程序进入顺逆风判断状态。

为验证硬件板或者 IPM 的可行性，当 U、V、W 三相都互补输出后，可选择不进入下一个状态，即一直处于预定位状态，用示波器观察 MCU 的六路输出是否正常，IPM 的输出是否正常。注意，此情况下不能接电机。

3.2.4. 顺风逆风判断

在风机应用中，当存在经常有外界风力吹动负载反向运行或者负载正转还未停止就要重新启动的情况时，风机启动部分可加入顺风逆风判断功能。6801 里，顺逆风检查可选择两种实现方式，一种是 FOC 估算模式，一种是 BEMF 模式。FOC 估算模式，不需要额外的硬件，利用估算器本身进行角度估算，再根据估算角度来判断。BEMF 模式即为方波无感的反电动势检测，此方法需采用三个比较器，因比较器 2、3 和运放 2、3 引脚共用，所以 BEMF 适用于单电阻的顺逆风检测。

1) FOC 估算模式

此功能加在预定位状态前，当预充电结束后，FOC 估算模式通过将给定电流设置为 0，FOC 采集电流电压信息估算当前转速和角度。程序可根据当前估算的转速来判断电机正反转，也可根据估算得到的角度来判断正反转。前者判断方便，计算出的速度是标幺值，当有些电机低速或者静止时响应不好，其容易将估算转速弄错。后者实现相对麻烦，其通过 FOC 中断来处理角度变化趋势，将

角度分为三个区间，小于-170°，大于 170°，在-10°到 10°。当角度在-170°以下和 170°以上来回摆动，不经过-10°到 10°时，认为静止。在角度-10°到 10°里，进行速度计算，并判断正反转，同时清零抖动次数。当电机正转时，速度是从-180°计数到-10°的，反转时速度是从 180°计数到 10°的，所以速度计算时，计数只计算半个电周期的。此方式计算出的速度单位是 rpm/min。

FOC 估算模式下，需用 SPI 观察估算角度是否正常。如，正转时角度是爬坡递增的变化，反转时角度是下坡递减的变化，静止时，角度可能是不变化的，也可能是上下电平跳变形式。估算角度方向如果不正确，或者响应不够快，需调节以下三个参数：观测器带宽滤波值 ATO_BW_Wind、速度带宽滤波值 SPD_BW_Wind、电流环 Kp, Ki。为加快估算响应，顺逆风判断时电流环 Kp, Ki 会比实际运行时电流环的偏大。估算角度方向正确后，再看计算的转速是否正确。

顺逆风判断后，根据当前转速和方向分别处理。当电机正转且大于一定值时，采用速度跟随，直接将状态切换到运行状态。若风机中用 Omega 启动，可根据正转转速设置不同的 FOC_EKP、FOC_EKI 以提高切入闭环的可靠性。判断反转时，采取刹车方式，根据不同转速设置不同刹车时间。当刹车超过一定次数时，可设置强制启动，此情况下，启动电流要适当增大。当电机静止或正转转速太小时，直接跳入下一个状态。

因 FOC 估算模式在顺逆风判断时输出占空比非常小，在低压应用中，当顺逆风转速太高时，反电动势比较大，易出现电压过冲现象。因此在低压风机应用中，为避免顺逆风转速过高导致电压过冲，建议顺逆风判断前先进行刹车再判断。

2) BEMF 模式

BEMF 模式下，控制器关闭输出，通过比较器将三相反电动势与过零点比较，产生对应的跳变沿中断。程序在比较器中断里根据比较器的输出电平和定时器计数来处理方向和速度。中断具体处理如下：

1、检测三个反电动势与过零点比较的状态；

2、根据反电动势与过零点比较的状态变化，得到正反转信息；

3、根据两个跳变沿的时间间隔进行速度检测，即读取定时器的计数值；

4、根据定时器计数值，算出当前的速度；

5、当检测到电机为正转，且速度比较高时，直接切入闭环运行中。

比较器中断中，根据比较器输出的高低电平来判断当前 BEMF 的状态，并根据状态变化来判断正反转，同时通过定时器的计数值来处理速度。当电机正转速度比较高时，在 U 相 BEMF 上升沿启动。当检测到电机反转时，虽检测时间还没结束，程序直接进入刹车模式。同时关闭比较器的中断。程序根据不同速度刹车不同时间。刹车结束后，重新开始顺逆风判断。当顺逆风检测时间结束时，若当前正向速度比较低或者电机静止时，程序跳入下一个状态。当曾经发生过逆风情况时，可根据逆风的转速，设置启动电流稍大于静止启动时电流以提高启动的可靠性。

由于不同电机反电动势值不同，BEMF 模式采集反电动势时需将反电动势分压处理。分压系数可与母线电压的分压系数相等。其需要用到三个比较器和定时器，因此得对比较器和定时器进行初

始化。因比较器的引脚与运放的引脚存在共用情况，所以其应用场合相对受限，68 系列里不适合用在双电阻 FOC 中。

3) RSD 模式

RSD 模式下，控制器关闭输出，通过比较器将三相反电动势的大小通过比较器进行比较（电路图如下），产生对应的跳变沿中断。程序在比较器中断里根据比较器的输出电平和定时器计数来处理方向和速度，注意次模式只在 FU6802 及其之后系列芯片可以使用。RSD 模式和 BEMF 模式的顺逆风检测方法不同，处理方法相同，RSD 模式可以减少比较器数量并提高顺逆风检测的最低转速。

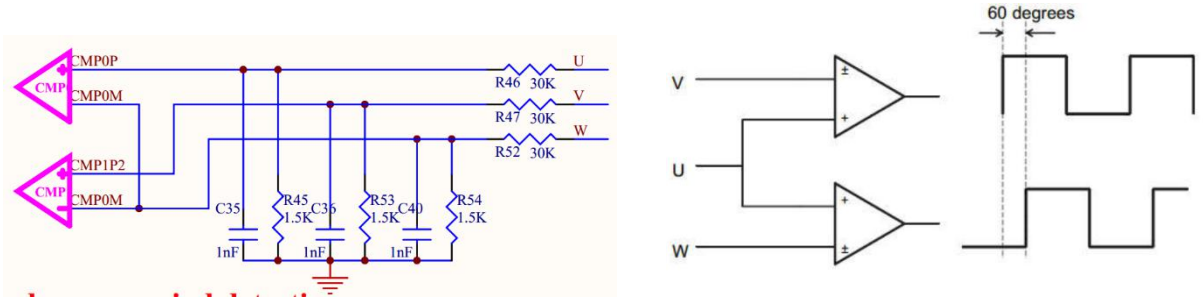


图 3-2-3 RSD 检测电路

中断具体处理如下：

1、 使用比较器 0 和比较器 1 检测三个反电动势两两比较的状态；

2、 UW 相比较器输出高电平时刻超前 UV 相输出高电平时刻 60°，认为电机正转，否则认为电机反转，如上图时序所示；

3、 通过测量两个比较器上升沿的时间间隔进行速度检测，即读取定时器的计数值；

4、 根据定时器计数值，算出当前的速度；

5、 当检测到电机为正转，且速度比较高时，直接切入闭环运行中；

6、 当两个比较电平没有变化超过一定时间认为电机没有运行时间间隔可以得到电机的转速。

3.2.5. 预定位

预定位采样的方式是 D 轴电流给定为 0，设置一定的 Q 轴电流，使电机铁芯处产生磁通量，将转子的位置固定在初始角度位置。当电机常规运行模式下，Q 轴电流为正，幅值比较小；当有逆风功能时，预定位的 Q 轴电流为负，辅助电机快速停下。预定位电流的幅值，电流环的 Kp、Ki，预定位时间都需根据实际情况做调整。

3.2.6. 启动调试

3.2.6.1. 启动模式介绍

电机的启动常见有三种模式，一种是爬坡启动，一种是采用位置估算算法的 Omega 启动，第三种是先爬坡启动后再用 Omega 启动。爬坡启动适用于大负载、慢启动应用场合；Omega 启动适用于启动快的应用场合，切换平滑；第三种启动方式适用于有初始位置检测的场合。

1) 爬坡启动：

其原理是给定启动电流(IQREF 为非零)，用爬坡方式将电机的电频率慢慢累加，坐标变换的角度信息由爬坡强制角度提供，拖动电机转动。待拖动电机速度达到一定时启动完成，角度信息逐渐切入到由位置估算算法提供。

此方式中爬坡强制角度由角度 THETA，速度 RTHESTEP，加速度 RTHEACC，爬坡计数器 RTHECNT 组成。对应的爬坡公式为：

速度 RTHESTEP(32bit) = 速度 RTHESTEP(32bit) + 加速度 RTHEACC

角度 THETA(16bit) = 角度 THETA(16bit) + 速度 RTHESTEP(高 16bit)

爬坡模块每个载波周期进行一次运算，爬坡计数器加一，当计数值达到 RTHECNT，RFAE 硬件清零，爬坡结束。爬坡的过程中，估算器也在估算角度，当爬坡结束后切换到估算模式，由于估算角度和爬坡角度存在偏差，因此需要平滑切换。可通过调节角度切换修正值 FOC_THECOR 的值减小切换抖动，实现平滑切换。

具体的调试参数如下表：

	参数调试		
	相关参数	参数范围	具体作用
爬坡启动	启动电流	小于运行最大电流	① 启动电流的大小由实际负载来决定，风机调试时，可在安全条件下用手抓住负载，感觉到有稳定力矩输出即可。 ② 启动电流增大，启动力矩会增大，过大容易启动过冲；过小启动力矩不够，启动容易失败或不能启动。
	爬坡启动增量 MOTOR_OPEN_ACC	1-200	① 爬坡启动增量与速度基准有关，影响启动频率增加的快慢。电机速度基准越高，其值就越大。 ② 增量增大，启动爬坡变快。增量过大，实际频率没法跟上给定的频率，容易失步；增量过小，启动太慢，易出现停顿现象。
	爬坡启动初始速度 MOTOR_OPEN_ACC_MIN		① 初始速度影响启动速度的快慢，默认初始速度为 0。 ② 在要求启动响应非常快的场合，可设置初始速度为非零速度。
	爬坡启动次数 MOTOR_OPEN_ACC_CNT	1-255	① 启动爬坡次数，寄存器最大值为 255。 ② 启动次数增大，启动时间变长。在一些应用中，启动时间长，可设置计数器计数完后重新赋值，赋值次数由实际应用决定。
	运行时观测器带宽滤波值 ATO_BW		① 观测器带宽影响估算器的估算速度响应。 ② 在爬坡启动时，估算器与爬坡启动相独立，只需设置一个让电机能正常运行的带宽滤波值即可。
	SMO 的最小速度		① SMO 的最小速度影响估算器的估算， ② 其值设置与速度基准相关，应低于电机正常运行的最小速度，但也不能过小。

2) Omega 启动：

其原理是，给定启动电流(IQREF 为非零)，位置估算算法估算当前速度，当估算速度小于用户设定的最小切换速度，估算器输出强制角度，将电机拖动。强制速度从 0 开始，每个运算周期与速度增量 EFREQACC 相加，同时根据启动限制转速进行最大值限幅，当估算电机速度大于或等于启动的最小切换转速，启动结束，角度信息由估算算法计算得出。

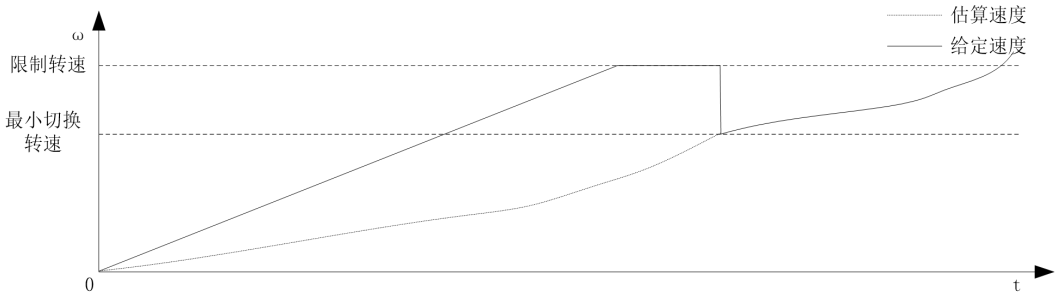


图 3-2-4 Omega 启动下的速度曲线

具体的调试参数如下表：

	参数调试		
	参数	**参数范围**	**具体作用**
Omega 启动	启动电流	小于运行最大电流	① 启动电流的大小由实际负载来决定，风机调试时，可在安全条件下用手抓住负载，感觉到有稳定力矩输出即可。 ② 启动电流增大，启动力矩会增大，过大容易启动过冲或者噪音；过小启动力矩不够，启动容易失败或不能启动。
	启动增量 Motor_Omega_Ramp_ACC	1-200	① 启动增量与速度基准有关，影响启动频率增加的快慢。电机速度基准越高，其值就越大。 ② 增量增大，启动输出频率增大。增量过大，电机实际频率没法跟上给定的频率，容易失步；增量过小，启动太慢，易出现停顿现象。
	启动的最小切换速度（rpm/min） MOTOR_OMEGA_ACC_MIN		① 启动最小切换速度与速度基准有关，电机速度基准越高，其值就越大。 ② 其值小于启动限制转速。 ③ 过小，估算器容易估算错误，电机还没拖动起来启动切换结束，易导致启动失败；过大，容易导致启动无法切入闭环或速度超调。
	启动的限制速度（rpm/min） MOTOR_OMEGA_ACC_END		① 启动限制速度与速度基准有关，电机速度基准越高，其值就越大。 ② 其值应小于电机运行最小速度，大于最小切换转速。 ③ 过小，估算器容易估算错误，电机还没拖动起来启动切换结束，易导致启动失败；过大，容易导致启动无法切入闭环或速度超调。
	启动死区电压补偿系数	0.5-2.0	① 启动死区电压补偿系数影响启动时的输出，与母线分压倍数、启动电流有关。

			② 建议从 1.0 往两边调试，主要范围在 0.8-1.75。当启动电流改变时，其值可能会发生变化。
	运行时观测器 带宽滤波值 ATO_BW		① 观测器带宽影响的是估算器里估算速度的 PI 环。 ② 为防止估算速度在低速时波动很大，一般将带宽滤波值从小到大增加，加速时间与实际的启动时间相关，启动时间要求越短，加速时隔就越短。调试时，应根据启动电流波形和实际转动的效果来修改。 ③ AT0_BW 过大，易导致反偏启动无法拉回正转，造成失步； 过小，易导致正向启动有停顿感。
	SMO 的最小速度		① SMO 的最小速度影响估算器的估算， ② 其值设置与速度基准相关，应低于电机正常运行的最小速度，但也不能过小，比启动限制转速要大。
	电流环 Kp、Ki		① 具体参数范围可参考运行电流环的 PI 系数描述。 ② 调试时可先给大，观察电流波形不好，再改小。

3) 先爬坡启动后 Omega 启动：

先爬坡启动后用 Omega 启动的方式是结合两者优点，先将电机拖动，电机有了一定转速后，再用 Omega 启动，确保可靠性。主要应用于有初始位置检测的启动。其参数设置可参考前面两种启动的具体调试参数。

这三种方式中，启动时间最长的是开环启动，最短的是 Omega 启动。电机启动参数配置后，电机启动进入运行 mcRun 状态。在电机运行状态下，可根据性能需求加入速度闭环控制或其他闭环控制。

3.2.6.2. 启动调试步骤

电机启动是电机调试中的一个重要内容，其可靠性对于整个方案至关重要。对于一个无感 FOC 方案，调试启动的步骤如下：

1、 测量电机的电机参数；

2、 设计合理的电流基准、最大采样电压、硬件过流保护值；

3、 验证硬件，包括电源电压、运放的基准电压、MOS 管输出、硬件过流保护；

4、 软件中填入合理的调试参数，如电机参数、采样电阻、放大倍数、母线分压电阻、速度基准、启动电流、有效电平等；

5、 电机静止，关闭其他保护，只保留硬件过流、软件过流两个保护。不加入顺逆风，在额定电压

下纯电流环运行情况下调试启动相关参数，确保每个点都能启动；

6、 加入顺逆风判断，验证顺逆风判断顺风，逆风，静止三种情况准确，再调试顺风和逆风的启动；若应用中有初始位置检测，应在调试静止启动前验证好初始位置检测是否正确。

3.2.6.3. 启动失败常见原因和分析

当我们调试电机时，会出现各种无法正常启动的情况，如电机一动不动，或者动一下就停了，或者运行一下就声音或者电流异常，或启动触发启动保护等。导致出现以上启动失败的原因很多，如硬件有问题、电机参数不合理、软件参数不合理等。不同现象可能是不同原因，以下是启动失败的常见原因分析：

1、 硬件是否正常，电源电压、运放的基准电压、mos 管输出、有无虚焊、有无触发硬件过流保护

2、 电机参数是否合理，如电机参数、采样电阻、放大倍数、母线分压电阻、速度基准等，可以查看对应的估算器的系数，FOC_EK1、FOC_EK2、FOC_EK3 是否溢出。

3、 软件参数是否合理，有效电平、驱动模式、单双电阻等。

4、 保护参数设置不合理，仿真模式下，看程序是否处于保护状态，查看是什么保护，分析产生原因并解决；

5、 仿真模式下，查看程序是否有启动指令，是否跳入状态机中对应状态。

6、 启动相关参数不合理。

7、 启动切入外环参数不合理。

8、 顺逆风判断速度和方向不正确。

针对不同的启动失败现象，可能原因不同，当出现启动时电机一动不动时，主要分析原因 1、2、3、4；当出现运行一下就停了或者波形异常，主要分析原因 1、2、3、4、6、7；当程序触发保护，主要分析原因 1、2、3、4、8；当静止启动没有问题，加入顺逆风后出现问题，则主要分析 7、8；当静止启动效果不好，但不触发保护，则主要分析 1、2、6、7、8。

启动切入闭环时，低速时速度容易过冲。

在给定档位为低速，由启动切入闭环时，速度容易出现过冲现象的问题，主要解决方法如下：

1、 电流环切入外环时，可将 IQREF 电流降低，外环 PI_UK 值赋初值为降低后的 IQREF，爬坡函数的实际值赋值为当前的电机实际速度。

2、 电流环切入外环时的切换速度适当降低。

3、 修改外环的 Kp、Ki，一般 Ki 是减小；适当时候可以降低外环的输出限幅。

4. FOC 运行调试篇

4.1. 电流环 PI 参数调节

	电流 PI 环参数调试			
	电流环参数	参数范围	参数增大效果	参数减小效果
运行调试	比例系数 Kp (格式：Q12)	4.0-0.001	①电流响应增快，电机负载波动时转矩响应增快。 ②过大时，会产生相电流波形震荡，低速时相电流波形正弦度失真，转矩输出不稳定。	①电流响应变慢，电机负载波动时转矩响应变慢，电流更平稳。 ②过小时，电机输出转矩受限，调速时转矩输出反应迟钝。
	积分系数 Ki (格式：Q12)	0.5-0.001	①对于给定目标电流跟随更快。 ②过大时，会产生相电流波形正弦度失真，无法消除电流静态误差。	①对于给定目标电流跟随变慢，可减小电流静态误差，电流波形正弦度更好。 ②过小时，对于给定输入目标电流反应迟钝，负载变化时输出转矩响应变慢。

4.2. 速度环 PI 参数调节

	速度 PI 环参数调试			
	速度环参数	参数范围	参数增大效果	参数减小效果
运行调试	调节时间 Ts (单位：ms)	1-1000	①对给定速度目标值响应变慢，电机负载波动时速度跟随变慢。 ②过大时，速度调节延迟严重，对外部速度目标值改变反应迟钝。	①对给定速度目标值响应变快，电机负载波动时速度响应变快。 ②过小时，外部速度目标值调节较快时容易导致速度过冲严重。
	比例系数 Kp (格式：Q12)	4.0-0.001	①速度目标值变化响应增快，电机负载波动时速度响应增快。 ②过大时，会产生速度超调严重。	①对速度目标值变化响应变慢，电机负载波动时速度响应变慢，速度更平稳。 ②过小时，对外部速度目标值改变反应迟钝。
	积分系数 Ki (格式：Q12)	0.5-0.001	①对于给定输入速度目标值跟随更快。 ②过大时，无法消除速度静态误差，容易出现速度震荡；速度目标值给定相同时，在不同负载条件下最终稳定速度可能各不相同。	①可减小或消除速度静态误差，电流波形周期波动更小。 ②过小时，对于给定速度目标值反应变慢，负载变化时速度响应变慢。

4.3. 恒转速调试及问题

<table>
<tr><td rowspan="5">控制模式调试</td><td colspan="2">恒转速调试</td></tr>
<tr><td>调试步骤：</td><td>①在无速度环（即利用纯电流环）调速时，在需求的调速范围内，需满足如下要求：
a.最大和最小速度均可达到（最好能留有一定余量，以达到最佳的调速响应性能）；
b.电流大小、波形和电机输出效率均满足要求；
c.电流增大和减小的整个过程中，电机速度增大和减小时始终电流稳定且过渡平滑；
d.电机实测速度与程序计算的速度反馈值一一对应准确；

②设置速度环输入和输出参数：
a.设置外部调速信号与速度环给定的转速目标值对应关系；
b.设置速度环输出最大和最小电流输出值；

③速度环调试初始经典参数设置：
调节时间 Ts=50(ms)，SKp=_Q12(1.0)，SKi=_Q12(0.01)。

④参照运行调试中的速度或电流 PI 环参数调试进行详细调试参数。</td></tr>
<tr><td>常见问题</td><td>解决方法</td></tr>
<tr><td>①速度环动态响应不足</td><td>a.参照运行调试中的电流或速度 PI 环参数调试；
b.修改速度滤波值 SPD_BW；
c.修改速度基准 MOTOR_SPEED_BASE；
d.修改载波频率 PWM_FREQUENCY；
e.无感 FOC 控制时，修改 ATO_BW；</td></tr>
<tr><td>②电机实测速度和程序计算速度不一致</td><td>a.修改速度滤波值 SPD_BW；
b.修改速度基准 MOTOR_SPEED_BASE；
c. 无感 FOC 控制时，修改 ATO_BW；</td></tr>
<tr><td>③速度周期震荡</td><td>a.参照运行调试中的电流和速度 PI 环参数调试；
b.修改速度滤波值 SPD_BW；
c.修改速度基准 MOTOR_SPEED_BASE；
d.修改载波频率 PWM_FREQUENCY；
e.若速度震荡频率为 50-100Hz 左右，请增大母线电容容值；
f.无感 FOC 控制时，修改 ATO_BW；</td></tr>
</table>

4.4. 恒功率调试及问题

<table>
<tr><td rowspan="6">控制模式调试</td><td colspan="2">恒转矩调试</td></tr>
<tr><td>调试步骤：</td><td>①在恒转矩（纯电流环，即 Q 轴电流环）调试前，需满足如下要求：
a.电流大小、波形和电机输出效率均满足要求；
b.最大和最小电流均可达到（最好能留有一定余量，以达到最佳的调节性能）；
c.电流增大和减小的整个过程中，电机速度增大和减小时始终电流稳定且过渡平滑；
d.电机实测转矩与程序计算的 FOC_IQ 反馈值一一对应准确；

②设置速度环输入和输出参数：
a.设置外部转矩调节信号与转矩环给定的转矩目标值 FOC_IQREF 对应关系；
b.设置转矩环输出最大 QOUTMAX 和最小 QOUTMIN 输出值；

③转矩环调试初始经典参数设置：
QKp=_Q12(1.0)，QKi=_Q12(0.01)。

④参照运行调试中的电流 PI 环参数调试进行详细调试参数。</td></tr>
<tr><td>常见问题</td><td>解决方法</td></tr>
<tr><td>①转矩环动态响应不足</td><td>a.参照运行调试中的电流 PI 环参数调试；
b.修改载波频率 PWM_FREQUENCY；
c. 修 改 DOUTMAX 、 DOUTMIN 、 QOUTMAX 和 QOUTMIN；
d.无感 FOC 控制时，修改 ATO_BW；</td></tr>
<tr><td>②转矩周期震荡</td><td>a.参照运行调试中的电流 PI 环参数调试；
b.修改载波频率 PWM_FREQUENCY；
c. 修 改 DOUTMAX 、 DOUTMIN 、 QOUTMAX 和 QOUTMIN；
d.无感 FOC 控制时，修改 ATO_BW；
e.若速度震荡频率为 50-100Hz 左右，请增大母线电容容值；</td></tr>
<tr><td>③电流波形存在正弦度失真</td><td>a.修改 DKP、DKI、QKP、QKI；
b.修改载波频率 PWM_FREQUENCY；
c. 修 改 DOUTMAX 、 DOUTMIN 、 QOUTMAX 和 QOUTMIN；
d.无感 FOC 控制时，修改 ATO_BW；
e.观察电流采样基准是否正常；</td></tr>
<tr><td>④电流波形存在正弦度缺口</td><td>a.修改 FOC_TRGDLY；
b.修改载波频率 PWM_FREQUENCY；
c. 修 改 DOUTMAX 、 DOUTMIN 、 QOUTMAX 和 QOUTMIN；</td></tr>
</table>

<table>
<tr><td rowspan="5">控制模式调试</td><td colspan="2">恒功率调试</td></tr>
<tr><td>调试步骤：</td><td>①在恒功率调试前，需满足如下要求：
a.按照以上恒转矩调试步骤，将恒转矩环调试完成；
b.电流大小、波形和电机输出效率均满足要求；
c.最大和最小功率均可达到（最好能留有一定余量，以达到最佳的调节性能）；
d.功率增大和减小的整个过程中，电机速度增大和减小时始终电流稳定且过渡平滑；
e.电机实测功率与程序计算的功率反馈值一一对应准确；

②设置功率环输入和输出参数：
a.设置外部功率调节信号与功率环给定的功率目标值对应关系；
b.设置功率环输出最大功率和最小功率输出值；

③功率环调试初始经典参数设置：
调节时间 Tp=5(ms)，PKp=_Q12(1.0)，PKi=_Q12(0.01)。

④可类似参照运行调试中的速度 PI 环参数调试进行详细调试参数。</td></tr>
<tr><td>常见问题</td><td>解决方法</td></tr>
<tr><td>①功率环动态响应不足</td><td>a.类似参照运行调试中的速度 PI 环参数调试；
b.修改功率环输出最大功率和最小功率输出值；
c.修改 DOUTMAX、DOUTMIN、QOUTMAX 和 QOUTMIN；
d.无感 FOC 控制时，修改 ATO_BW；</td></tr>
<tr><td>②功率周期震荡</td><td>a.类似参照运行调试中的速度 PI 环参数调试；
b.若速度震荡频率为 50-100Hz 左右，请增大母线电容容值；
c.检查电压采样和电流采样值是否会存在周期性波动；</td></tr>
<tr><td>②功率与外部实测功率对应不准</td><td>a.修改电压采样或者电流采样值的系数，使之和实际值对应正确；
b.修改电流采样的母线电流 RC 滤波值，使之与实际电流值对应正确；</td></tr>
</table>

4.5. 恒 UQ 调试及问题

控制模式调试	恒 UQ 调试	
	调试步骤：	①在恒 UQ 调试前，需满足如下要求： a.按照以上恒转矩调试步骤，将恒转矩环调试完成； b.电流大小、波形和电机输出效率均满足要求； c.最大和最小设定 UQ 均可达到（最好能留有一定余量，以达到最佳的调节性能）； d.UQ 增大和减小的整个过程中，电机速度增大和减小时始终电流稳定且过渡平滑； ②设置 UQ 环输入和输出参数： a.设置外部 UQ 调节信号与 UQ 环给定的 UQ 目标值对应关系； b.设置 UQ 环输出最大 Q 轴电流和最小 Q 轴电流输出值； ③UQ 环调试初始经典参数设置： 调节时间 TU=2(ms)，UKp=_Q12(1.0)，UKi=_Q12(0.01)。 ④可类似参照运行调试中的速度 PI 环参数调试进行详细调试参数。
	常见问题	解决方法
	①UQ 环动态响应不足	a.类似参照运行调试中的速度 PI 环参数调试； b.修改 UQ 环输出最大 Q 轴电流和最小 Q 轴电流； c.修改 DOUTMAX、DOUTMIN、QOUTMAX 和 QOUTMIN； d.无感 FOC 控制时修改 ATO_BW；
	②UQ 周期震荡	a.类似参照运行调试中的速度 PI 环参数调试； b.若速度震荡频率为 50-100Hz 左右，请增大母线电容容值； c.修改载波频率 PWM_FREQUENCY； d.修改 DOUTMAX、DOUTMIN、QOUTMAX 和 QOUTMIN；

问题&优化	常见问题	解决方法
	①效率偏低	a.参照电流 PI 环参数调试，使电流波形更加稳定； b.修改 DOUTMAX、DOUTMIN、QOUTMAX 和 QOUTMIN； c.修改载波频率 PWM_FREQUENCY； d.无感 FOC 控制时，修改 ATO_BW； e.无感 FOC 控制时，按比例缩小电机参数 RS、LD、LQ 和 Ke; f.无感 FOC 双电阻控制时，修改为五段式 SVPWM 斩波；
	②最高速偏低	a.参照电流 PI 环参数调试，使高速时电流波形更加稳定； b.修改 DOUTMAX、DOUTMIN、QOUTMAX 和 QOUTMIN； c.无感 FOC 控制时，修改 ATO_BW； d.无感 FOC 控制时，按比例缩小电机参数 RS、LD、LQ 和 Ke; e.无感 FOC 双电阻控制时，修改为五段式 SVPWM 斩波； f.母线电压为低压时，注意电源供电线上的压降损耗；

5. FOC 保护篇

5.1. 过流保护

5.1.1. 硬件过流

我司硬件过流方法主要有两种：比较器，与外部中断。

采用比较器的硬件过流检测方法： 母线电流（I_bus）流经采样电阻（RNF），在采样电阻上形成一个电压 V_Ishunt, 这个电压经过运算 AMP0 放大(放大倍数为 A)送入 CMP3 的正向输入端。CMP3 的负向输入端会被设置一个参考电压（V_ref）,这个参考电压一般由两颗分压电阻对 5V 进行分压得到，在 6802 和 6803 系列芯片中也可以通过配置该口 DA 输出相应的电压来代替电阻的分压，DA 的详细配置方法参考芯片 datasheet 中 DAC 相关章节。当母线电流增大到一定数值之后，就会导致 CMP3 的正向输入端的电压高于负向输入端电压，这个时候就会触发 MCU 的比较器中断，MCU 发生中断并立即进入过流保护程序。从而完成过流保护。

过流保护值为：I_bus=Vref/ A / RNF;

硬件过流值由比较器的负向输入端电压，母线电流放大器倍数，以及母线采样电阻共同决定。

改变这三个参数的任意一个都可以改变过流保护电流值。

比较器硬件过流检测方法的电流过流值计算如下：以下图为例，母线电流采样电阻为 0.1Ω（下图未画出来），运算放大器放大倍数为 10k/2k = 5 倍,VHALF 为 4.5V，比较器负向输入端的参考电压为：1K/(1K+51K)*5V =4.9V 则在此工程中，硬件过流值为：(4.9-1/2*4.5)/5/0.1=5.3A。

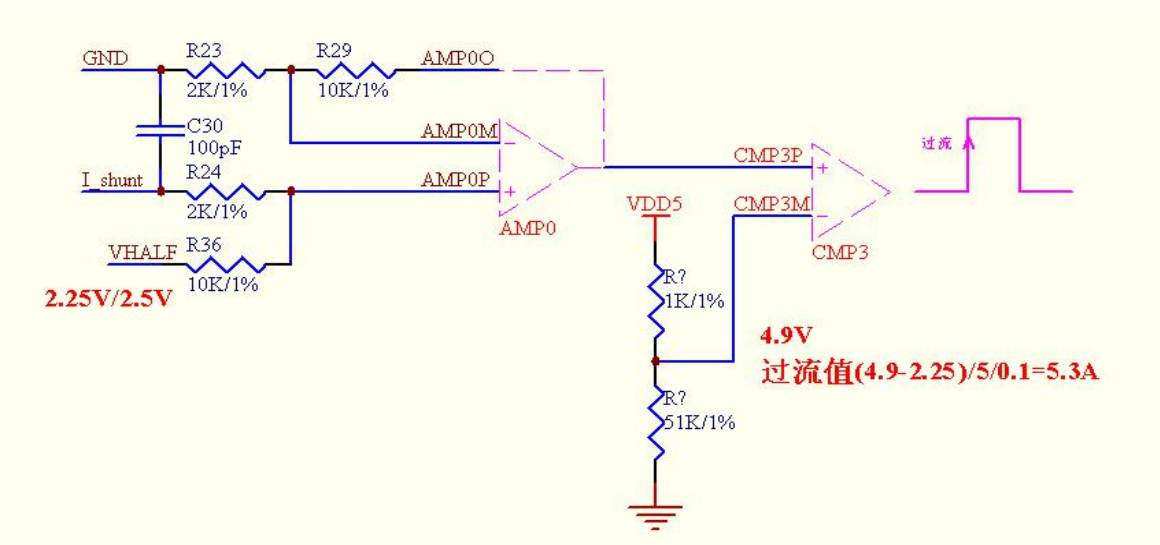


图 5-1-1：硬件过流参考图纸

注意：如果是双电阻采样，电路上面可能没有偏置电阻 R36，那么运放倍数为 10k/2k+1 = 6 倍，此时母线上面没有偏置电压 VHALF，硬件过流值则为：4.9/6/0.1=8.1A。

采用外部中断过流检测方法：

这种方法适用于带过流保护功能的 HVIC 或者 IPM 的电路，当发生过流时，HVIC 或 IPM 会有一个 IO 电平发生改变，以此来触发 MCU 外部中断，来达到过流保护的目的，此时的过流值由 HVIC 或者 IPM 电路部分设置

验证方法：

过流保护在所有保护中的优先级应该位于最高，过流保护能够保护控制器不被损坏，因此在程序调试的最开始就应该验证过流保护是否生效。

不接电机，对 MCU 进行仿真在图 1 中位置打断点

```
52  void CMP_ISR(void) interrupt 7
53  {
54    if(GetBit(CMP_SR, CMP3INTR))
55    {
56      FaultProcess();                    // 关闭输出
57      mcFaultSource=FaultHardOVCurrent;  // 硬件过流保护
58      mcState = mcFault;                 // 状态为mcFault
59      CLR(CMP_SR, CMP3INTR);
60    }
61    #if (BEMF_DETECT_ENABLE)
62    {
63      //通过BEMF做顺风启动功能
64      BEMFDetectFunc();
65    }
66    #endif
67  }
```

运行程序，用 5V 串接一个 10K 的电阻，触碰 AMP0O 管脚，观察程序是否能够停在上图中的位置。如果程序能够运行到此位置，则可以确定硬件过流能够生效。

注意：在任何情况下，硬件过流的中断优先级都应该保证为最高。

5.1.2. 软件过流

软件过流程序运行在 FOC 中断中运行，固定间隔时间检测相电流是否大于设定的过流值。如果在一定时间内连续检测到三次相电流大于设定值，则进行软件过流保护。

具体设定电流值在 customer.h 中 OverSoftCurrentValue 宏定义中更改。

```
#define    OverSoftCurrentValue   I_Value(9.5)
```

5.2. 堵转保护

堵转保护主要作用是在电机运行过程中，电机遇到堵转问题时，能够使电机停止运行从而保护电机不被损坏。

堵转保护检测函数在 1ms 中断函数里被调用，变量 mcFaultDect.segment 由 0 开始每进入一次 1ms 中断加 1，等加到 5 时再清零，不断循环，只有在变量 mcFaultDect.segment 等于 3 时才会调用堵转保护检测函数，即堵转保护检测函数 5ms 被调用一次。

我司无感 FOC 的堵转保护有三种堵转检测方法：

1) 通过检测估算器计算出来的 FOC_ESQU，电机转速越高，FOC_ESQU 越大，若 FOC_ESQU 小于正常运行时候的最低值时候，则认为发生了堵转。根据这个特性，设计堵转保护函数。

估算的速度超过最大堵转速度或者反电动势小于特定的值，且超过一定时间，认定出现堵转问题，进入堵转保护程序，将错误源设定为堵转错误，执行保护程序。FaultProcess();调试堵转保护的时候可以通过手动调整 40 和 Motor_Stall_Max_Speed 这两个比较值。最大堵转速度 Motor_Stall_Max_Speed 是根据最大转速来确定的，一般取最大转速的 1.3 倍。

FOC_ESQU 阈值设置如下：

在仿真下，以工程中要求的最低转速运行电机。查看当前 mcFaultDect.mcEsValue 的值，并记录(假设为A)，将Fault_Stall(&mcFaultDect);函数中的(h_Fault->mcEsValue< NUM) 判断条件中的 NUM 设置为 A 的一半，下面程序设定为 40。

对于在规定最低转速下运行的时候 mcFaultDect.mcEsValue 小于 15 的电机，此种方法可能会导致电机在低速运行时候误触发堵转保护。

```
if((h_Fault->mcEsValue < 40)||(FOC_EOME > Motor_Stall_Max_Speed))
{
    h_Fault->mcStallDelaDectEs++;
    if(h_Fault->mcStallDelaDectEs >= 10)
    {
        h_Fault->mcStallDelaDectEs=0;
        mcFaultSource=FaultStall;
        h_Fault->StallFlag   =   1;
        FaultProcess();
    }
}
else
{
    if(h_Fault->mcStallDelaDectEs>0)
    {
        h_Fault->mcStallDelaDectEs--;
    }
}
```

2) 通过估算器估算出来的电机转速为依据判断是否发生堵转

由于无感 FOC 是用过电流来进行角度与速度的计算，所以在电机发生堵转的时候，估算器计算出的速度可能会很低也可能会很高，这个时候程序进行速度判断，若在一定时间内，电机转速持续低于设定的最低堵转保护转速，或者高于设定的最高堵转保护转速，则进行堵转保护程序。

一般来说最低堵转保护速度按照实际运行的最低速度的 50%设置（不是固定，可以按照具体的应用进行灵活调整）

最高堵转保护转速则需要将电机堵转（先去掉堵转保护），此时电机将会高频振荡，但是不会转动。查看此时的相电流波形，应当能看到频率较高的正弦波形。根据正弦波形的频率，推导出此时电流对应的电机转速（假设为 RPM_A）。则可将最高堵转保护转速设置为 RPM_A * 90%.

```
if((mcFocCtrl.mcSpeedFlt<Motor_Stall_Min_Speed)||((mcFocCtrl.mcSpeedFlt >
_Q15(1300.0/MOTOR_SPEED_BASE))&&(h_Fault->mcEsValue < 120)))
{
    h_Fault->mcStallDeSpeed++;
    if(h_Fault->mcStallDeSpeed>=8)
    {
        h_Fault->mcStallDeSpeed=0;
        mcFaultSource=FaultStall;
        h_Fault->StallFlag =2;
        FaultProcess();
    }
}
else
{
    h_Fault->mcStallDeSpeed=0;
}
```

3) 根据相电流的大小判断是否堵转

电机发生堵转时，电机的相电流会大于正常运行时的电流，如果判断到任意一相的相电流大于所设定的堵转保护电流，则程序执行相应的堵转保护程序。堵转保护电流设置一般会比软件过流值略小。

```
if((h_Fault->Abs_ia>=StallCurrentValue1)||(h_Fault->Abs_ib>=StallCurrentValue1)||
(h_Fault->Abs_ic>=StallCurrentValue1))
{
    h_Fault->mcStallDeCurrent++;
    if(h_Fault->mcStallDeCurrent>=6)
    {
        h_Fault->mcStallDeCurrent=0;
        mcFaultSource=FaultStall;
        h_Fault->StallFlag =3;
        FaultProcess();
    }
}
```

```
else if((h_Fault->Abs_ia<=(StallCurrentValue1-I_Value(0.05)))&&
(h_Fault->Abs_ia<=(StallCurrentValue1-I_Value(0.05))))
{
    h_Fault->mcStallDeCurrent=0;
}
```

5.3. 缺相保护

缺相保护主要作用是在电机运行过程中出现一相缺失或者接触不良的情况时，启动缺相保护程序，停止电机运行。

缺相保护检测函数在 1ms 中断函数里被调用，变量 mcFaultDect.segment 由 0 开始每进入一次 1ms 中断加 1，等加到 5 时再清零，不断循环，只有在变量 mcFaultDect.segment 等于 4 时才会调用缺相保护检测函数，即缺相保护检测函数 5ms 被调用一次。

电机发生缺相时，三相电流是不对称的。因此在程序中检测一定时间内的三相电流的最大值，判断三相电流的最大值是否有不对称的情况，如果是则判定为电机缺相，MCU 执行相应保护程序。

具体程序实现方法：判断三相电流中，其中一相最大电流值是否大于其他任意一相电流的最大值，且大于缺相保护电流值，则认定出现缺相问题，执行缺相保护，将错误源置为缺相错误，电机状态置为错误状态，并执行保护程序 FaultProcess()。缺相电流值 PhaseLossCurrentValue 直接取 0.2A 就 OK 了。函数输入是错误变量（FaultVarible）结构体的地址，输出为错误类型，这里为缺相错误。

```
if(mcState == mcRun)
{
    h_Fault->LphaseDelayCount++;
    if(h_Fault->LphaseDelayCount>=2)
    {
        h_Fault->LphaseDelayCount=0;
        h_Fault->Lphasecnt++;
        if(h_Fault->Lphasecnt>50)
        {
            h_Fault->Lphasecnt=0;
                if(((h_Fault->Max_ia>(h_Fault->Max_ib*3))
            ||(h_Fault->Max_ia>(h_Fault->Max_ic*3)))
            &&(h_Fault->Max_ia>PhaseLossCurrentValue))
            {
                h_Fault->AOpencnt++;
            }
            else
            {
                h_Fault->AOpencnt = 0;
            }
            if(((h_Fault->Max_ib>(h_Fault->Max_ia*3))
            ||(h_Fault->Max_ib>(h_Fault->Max_ic*3)))
```

```
                &&(h_Fault->Max_ib>PhaseLossCurrentValue))
                {
                    h_Fault->BOpencnt++;
                }
                else
                {
                    h_Fault->BOpencnt = 0;
                }
                if(((h_Fault->Max_ic>(h_Fault->Max_ia*3))
                ||(h_Fault->Max_ic>(h_Fault->Max_ib*3)))
                &&(h_Fault->Max_ic>PhaseLossCurrentValue))
                {
                    h_Fault->COpencnt++;
                }
                else
                {
                    h_Fault->COpencnt = 0;
                }

                h_Fault->Max_ia = 0;
                h_Fault->Max_ib = 0;
                h_Fault->Max_ic = 0;
                if(h_Fault->AOpencnt > 1|| h_Fault->BOpencnt > 1 || h_Fault->COpencnt > 1)
                {
                    mcFaultSource=FaultLossPhase;
                    mcState = mcFault;
                    FaultProcess();
                }
            }
        }
    }
```

5.4. 过欠压保护

过欠压保护主要作用是在电路板上电的时候，供电电压过高或者过低，能够使电机停止运行从而保护板子上面的器件不被损坏。

一般情况下，过欠压保护只需要设置合适的电压，并开启了电压保护功能，就能正常生效。

程序中设置过/欠压保护为直流电压值，该值一般计算为交流电压值的 1.414 倍，例如需要设置 AC 260V 保护时 ，则在程序上设置保护电压值为 260*1.414 = 367.64V，过/欠压保护恢复值亦如此；电机运行时进入过/欠压保护后，若电压回落于过/欠压保护恢复值范围内则电机自动重启；高压供电时过/欠压保护恢复电压值需与所设置的过/欠压保护值差 DC 20V 或以上，设置参数如下：

```
#define   Over_Protect_Voltage        (370.0)        //过压保护直流电压值，单位：V
#define   Under_Protect_Voltage       (100.0)        //欠压保护直流电压值，单位：V
#define   Over_Recover_Vlotage        (350.0)        //过压保护恢复值，单位：V
```

```
#define  Under_Recover_Vlotage      (120.0)          //欠压保护恢复值，单位：V
#define  VoltageProtectEnable       (1)              //电压保护使能
```

如果电压保护不能正常工作：

1、仿真查看与母线电压对应的 ADC 是否采集正确。

假设当前使用的是 ADC4 采集母线电压，母线分压比为 1/11 ，ADC 基准为 4.5V。运行程序（电机可以不启动），如果当前 ADC4_DR 的值为 2000

则实际电压应该为 2000/4096 * 4.5 * 11 = 24.16V 。如果发现测量值与实际值差别较大，则需要检查母线分压电阻是否正确，ADC 通道是否对应。

2、检查软件中设置的母线电压分压电阻是否按照实际值来填写。

函数详解：

电压保护检测函数在 1ms 中断函数里被调用，变量 mcFaultDect.segment 由 0 开始每进入一次 1ms 中断加 1，等加到 5 时再清零，不断循环，只有在变量 mcFaultDect.segment 等于 1 时才会调用电压保护检测函数，即电压保护检测函数 5ms 被调用一次。进入电压保护函数后，如果母线电压 AdcSampleValue.ADCDcbus 大于过压保护阈值 OVER_PROTECT_VALUE 时，变量 mcFaultDect. mcOverVoltDetecFaultCount 加 1，否则如果变量 mcFaultDect. mcOverVoltDetecFaultCount 大于 0 则减 1，当变量 mcFaultDect. mcOverVoltDetecFaultCount 达到 20 次（时间 0.1s）时，mcFaultDect. mcOverVoltDetecFaultCount 清零，并进行保护判断和恢复处理。因为母线电压越高，母线电压采样值 AdcSampleValue.ADCDcbus 越大，如果母线电压采样值 AdcSampleValue.ADCDcbus 大于过压保护阀值 OVER_PROTECT_VALUE 时，电机运行状态机进入 mcFault 状态，程序判断为过压保护，电机停机。同理，电压越低，电压采样值 AdcSampleValue.ADCDcbus 越小，如果电压采样值 AdcSampleValue.ADCDcbus 小于欠压保护阀值 UNDER_PROTECT_VALUE 时，电机运行状态机进 mcFault 状态，程序判断为欠压保护，电机停机。

电压过压保护状态下，当检测到母线电压采样值 AdcSampleValue.ADCDcbus 低于过压保护恢复阀值 OVER_RECOVER_VALUE 时，变量 mcFaultDect. mcVoltRecoverCount 开始计数加 1，当连续计数达到 200 次（时间为 1.0s）时，故障保护解除，电机运行状态机进入 mcReady 状态，电机重新启动。

电压欠压保护状态下，当检测到母线电压采样值 AdcSampleValue.ADCDcbus 高于欠压保护恢复阈值 UNDER_RECOVER_VALUE 时，变量 mcFaultDect. mcVoltRecoverCount 开始计数加 1，否则如果变量 mcFaultDect.VoltRecoverCnt 大于 0，则置零；当连续计数达到 200 次(时间为 1.0s)时，故障保护解除，电机运行状态机进入 mcReady 状态，电机重新启动。

5.5. 启动保护

启动保护主要作用是在电机运行过程中，电机遇到启动问题时，能够使电机停止运行然后重新启动，保证启动的正常。

启动保护检测函数在 1ms 中断函数里被调用，变量 mcFaultDect.segment 由 0 开始每进入一次 1ms 中断加 1，等加到 5 时再清零，不断循环，只有在变量 mcFaultDect.segment 等于 2 时才会调用启动保护检测函数，即启动保护检测函数 5ms 被调用一次。

启动判断方法有三种，当电机启动失败时，可能会触发三种方法的任何一种，从而实现保护。

1) Method1 在 5s 内，如果估算器的滤波后的速度大于电机最大速度或者反电动势小于一定值的时候，认定启动失败，将错误源设定为启动错误，电机状态设定为错误状态，并执行保护程序 FaultProcess()，最大运行速度 Motor_Max_Speed 由额定转速确定，对应的反电动势（此处为 150）由最大运行速度下反电动势值的一半确定；

```
if(Time.mcRunStateCount<5000)
{
    if((mcFocCtrl.mcSpeedFlt > Motor_Max_Speed)&&(h_Fault->mcEsValue<150))
    {
        mcFaultSource=FaultStart;
        mcState = mcFault;
        h_Fault->StartFlag = 1;
        FaultProcess();
    }
}
```

2) Method2 在 1.5s 到 6s 内，如果反电动势低于一定值且持续一定时间，认定启动失败，将错误源设定为启动错误，电机状态设定为错误状态，并执行保护程序 FaultProcess();

```
if((Time.mcRunStateCount>1500)&&(Time.mcRunStateCount<=6000))
{
    if(h_Fault->mcEsValue <50)
    {
        h_Fault->mcStartCnt++;
        if(h_Fault->mcStartCnt>=60)
        {
            mcFaultSource=FaultStart;
            mcState = mcFault;
            h_Fault->StartFlag = 2;
            FaultProcess();
        }
    }
    else
    {
        h_Fault->mcStartCnt=0;
    }
}
```

3) Method3 电机处于 mcFocCtrl.CtrlMode == 0 且持续一定时间，认定启动失败，将错误源设定为启动错误，电机状态设定为错误状态，并执行保护程序 FaultProcess()；

```
if(mcFocCtrl.CtrlMode==0)
{
    h_Fault->mcStartFocmode++;
    if(h_Fault->mcStartFocmode>=400)
    {
        h_Fault->mcStartFocmode=0;
        mcFaultSource=FaultStart;
        mcState = mcFault;
        h_Fault->StartFlag = 3;
        FaultProcess();
    }
}
```

启动保护恢复

电机处于错误状态，错误源为启动错误，并且二次启动次数小于重新启动次数时，可执行电机再次启动，将错误源置为没有错误，电机状态设定为停止。

```
if((mcFaultSource==FaultStart)&&(mcState ==mcFault)
&&(mcProtectTime.SecondStartTimes<StartProtectRestartTimes))
{
    mcProtectTime.SecondStartTimes++;
    mcFaultSource=FaultNoSource;
    mcState = mcWait;
}
```

5.6. 保护自动恢复启动

以过欠压保护恢复启动为例：

找到 void Fault_OverUnderVoltage(FaultVarible *h_Fault) 函数（位于 AddFunction.c）此函数用于过欠压保护。运行于定时器 4 或定时器 5 的 1ms 中断服务函数中。过欠压保护恢复程序见下图：

```
if((mcState == mcFault) &&((mcFaultSource==FaultUnderVoltage)||(mcFaultSource==FaultOverVoltage)))
{
  if((mcDcbusFlt< OVER_RECOVER_VALUE)&&(mcDcbusFlt> UNDER_RECOVER_VALUE))
  {
    h_Fault->mcVoltRecoverCount++;
    if(h_Fault->mcVoltRecoverCount>200)//连续检测1s，若正常则恢复
    {
      mcState = mcwait;
      mcFaultSource=FaultNoSource;
      h_Fault->mcVoltRecoverCount = 0;
    }
  }
  else
  {
    h_Fault->mcVoltRecoverCount = 0;
  }
}
```

当系统发生过欠压的时候，程序将会进行相应保护。系统状态就将被软件设置为 mcFault 状态，并将错误状态设置为过欠压状态。此种情况下，如果检测到电压恢复到系统正常运行的电压，则电机将会重新启动。如果不需要恢复功能，则将此段程序屏蔽即可。、

其他保护恢复与过欠压保护恢复类似。

6. 附录 1：FOC 模块

FU68XX 系列中 FOC 模块包含角度模块，PI 控制器，坐标转换模块，输出模块；可以采用内部角度估算模块实现无感 FOC 控制；也可以联合 MCU 处理 HALL 信号实现有 HALL FOC 控制。FOC 模块内部包含电流闭环，用户通过给定 ID、IQ 的参考值，就可以输出六路 PWM 驱动电机，同时 ADC 自动采集电流作电流闭环。这里只介绍无感 FOC 的控制。

无感 FOC 的控制框架和对应变量说明如图 6-1 和表 6-1 所示。

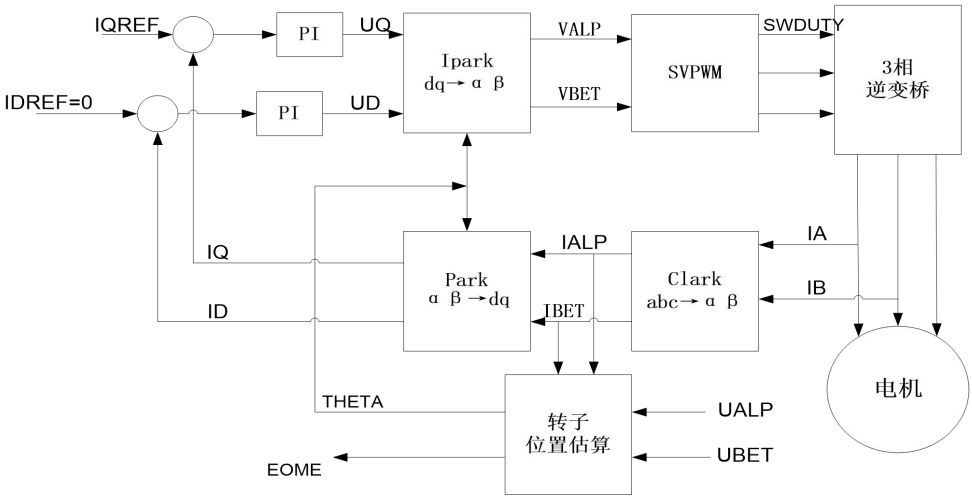


图 6-1 无感 FOC 框架

	寄存器名称	寄存器含义	取值范围
1	FOC_IDREF	用户给定的电流 ID 参考值	（-32768,32767）
2	FOC_IQREF	用户给定的电流 IQ 参考值	（-32768,32767）
3	FOC_ID	PARK 变换算出的 ID	（-32768,32767）
4	FOC_IQ	PARK 变换算出的 IQ	（-32768,32767）
5	FOC_IA	相电流 IA	（-32768,32767）
6	FOC_IB	相电流 IB	（-32768,32767）
7	FOC_IBET	Beta 轴电流	（-32768,32767）
8	FOC_UD	D 轴电压	（-32768,32767）
9	FOC_UQ	Q 轴电压	（-32768,32767）
10	FOC_VALP	IPARK 变换算出的 VALPHA	（-32768,32767）
11	FOC_VBET	IPARK 变换算出的 VBETA	（-32768,32767）
12	FOC_UALP	通过电压计算模块算出的 UALPHA	（-32768,32767）
13	FOC_UBET	通过电压计算模块算出的 VBETA	（-32768,32767）

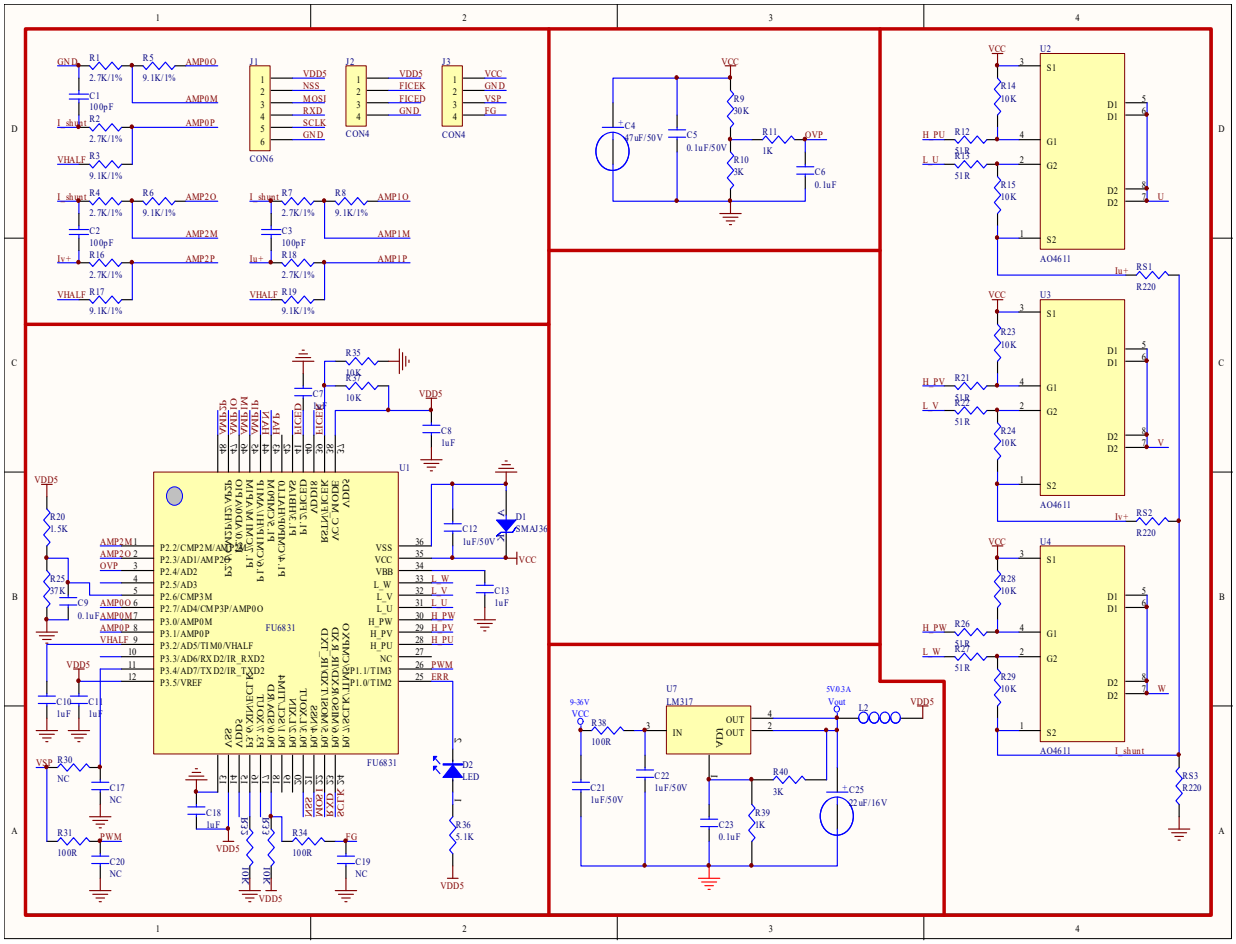
14	FOC_UDCFLT	滤波过后的母线电压	（0,32767）
15	FOC_THETA	软件写：强拉角度； 软件读：当前 FOC 工作的角度；	（-32768,32767）
16	FOC_ETHETA	估算器算出的角度（补偿 FOC_THECOMP 前的角度）	（-32768,32767）
17	FOC_THECOMP	角度补偿值：估算器估算出角度后加上补偿值作为估算器最终的输出	（-32768,32767）
18	FOC_DKP	D 轴的 PI 控制器的 KP 系数	（0,32767） Q 格式可配
19	FOC_DKI	D 轴的 PI 控制器的 KI 系数	（0,32767） Q 格式可配
20	FOC_DMAX	D 轴的 PI 控制器的输出 UD 的上限值	（-32768,32767）
21	FOC_DMIN	D 轴的 PI 控制器的输出 UD 的下限值	（-32768,32767）
22	FOC_QKP	Q 轴的 PI 控制器的 KP 系数	（0,32767） Q 格式可配
23	FOC_QKI	Q 轴的 PI 控制器的 KI 系数	（0,32767） Q 格式可配
24	FOC_QMAX	Q 轴的 PI 控制器的输出 UQ 的上限值	（-32768,32767）
25	FOC_QMIN	Q 轴的 PI 控制器的输出 UQ 的下限值	（-32768,32767）
26	FOC_ARR	计数器重载值，决定载波周期、运算周期	（0,65535）
27	FOC_COMR	计数器的比较匹配值	（0,65535）
28	FOC_SWDUTY	软件写 PWM 占空比	（0,65535）
29	FOC_DTR	Deadtime（死区时间）	（0,65535）
30	FOC_TSMIN	单电阻采样模式下，给 ADC 采样预留的最小窗口	（0,65535）
31	FOC_TRGDLY	双电阻采样模式：电流采集时机	（-32768,32767）
32	FOC_RTHECNT	爬坡次数= RTHECNT*256	（0,255）
33	FOC_RTHESTEP	软件写：初始速度 软件读：当前速度	（-32768,32767）
34	FOC_RTHEACC	爬坡模块的加速度	（-32768,32767）
35	FOC_THECOR	角度切换修正值	（0,32767）
36	FOC_EFREQACC	估算器强制角度模式的 OMEGA 增量	（0,65535）
37	FOC_EFREQMIN	OMEGA 最小值：估算器强制角度模式使能时，估算 OMEGA 小于该值时，强制角度模式生效	（-32768,32768）
38	FOC_EFREQHOLD	OMEGA 保持值：当 OMEGA 增加到等于该值时，就保持为这个值	（-32768,32768）
39	FOC_EK1	估算器估算电流的第一个系数	（0,32767）
40	FOC_EK2	估算器估算电流的第二个系数	（0,32767）

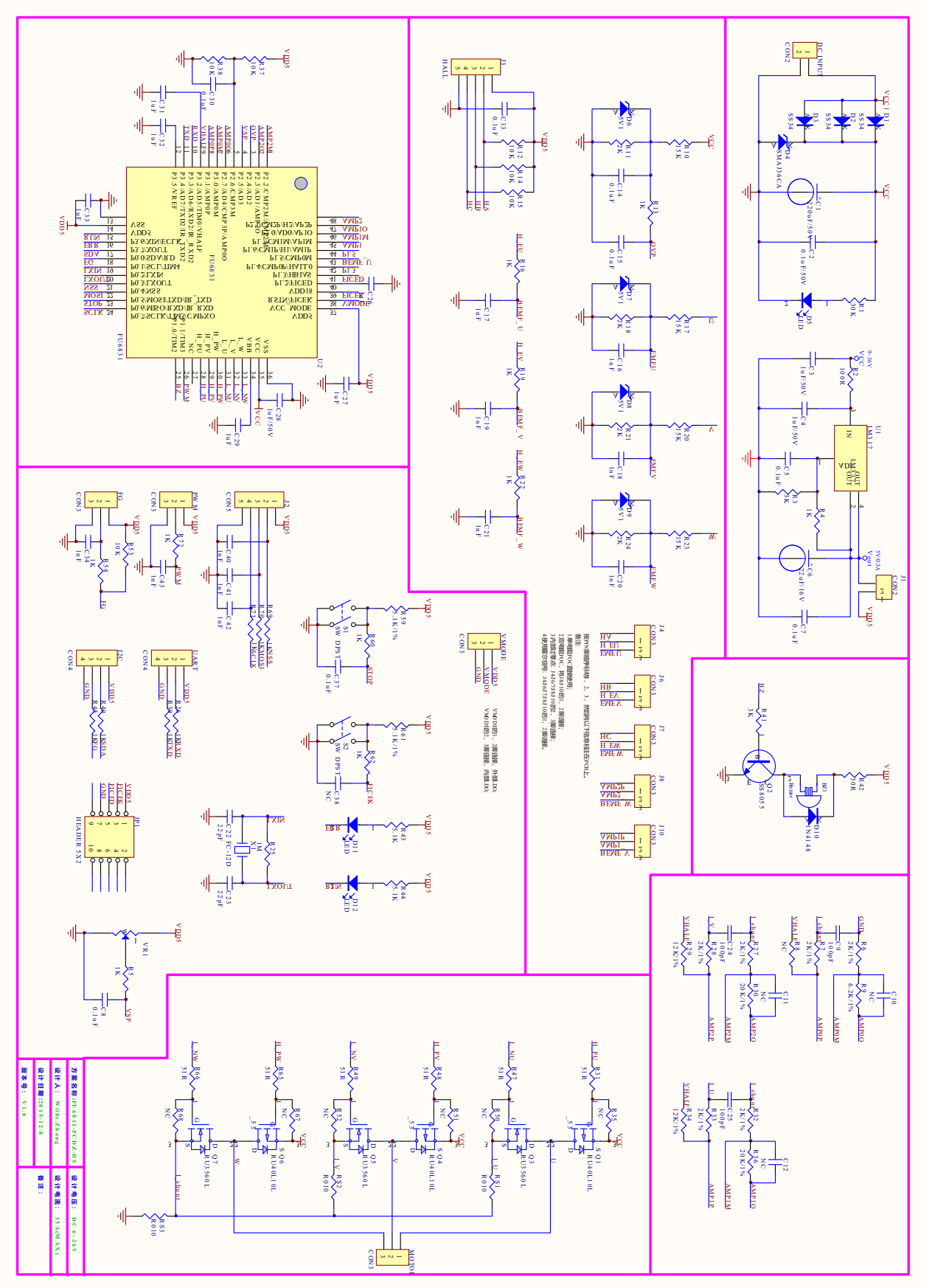
41	FOC_EK3	估算器估算电流的第三个系数	（0,32767）
42	FOC_EK4	估算器估算电流的第四个系数	（0,32767）
43	FOC_FBASE	估算器里由速度 OMEGA 算出角度增量 DELTA THETA 的系数	（0,32767）
44	FOC_EKP	估算器里的 PI 控制器的 KP 系数	（0,32767） Q 格式可配
45	FOC_EKI	估算器里的 PI 控制器的 KI 系数	（0,32767） Q 格式可配
46	FOC_KSLIDE /FOC_PLLKP	当 FOC_CR1 的 ESEL=0（滑模模式）时，为估算器里的 KSLIDE 系数； 当 FOC_CR1 的 ESEL=1（PLL 模式）时，为 PLL 的 PI 控制器的 KP 系数	（0,32767）
47	FOC_EKLPFMIN / FOC_PLLKI	当 FOC_CR1 的 ESEL=0（滑模模式）时，为估算器里反电动势低通滤波系数的最小值；当估算器算出的低通滤波系数小于最小值，系数等于最小值； 当 FOC_CR1 的 ESEL=1（PLL 模式）时，为 PLL 的 PI 控制器的 KI 系数	（0,32767）
48	FOC_EBMFK	估算器里算反电动势低通滤波器系数 EKLPF 的系数	（-32768,32767）
49	FOC_OMEKLPF	估算器里速度计算的低通滤波系数	（0,32767）
50	FOC_EOME	估算器估算的速度 OMEGA	（-32768,32767）
51	FOC_POWKLPF	功率计算的低通滤波系数	（0,32767）
52	FOC_POW	功率	（-32768,32767）

表 6-1　FOC 寄存器

可根据性能需求加入速度闭环控制或其他闭环控制。

7. 附录 2：原理图


GND
R1
2.7K/1%
R5
9.1K/1%
AMP0O
C1
100pF
AMP0M
I_shunt
R2
2.7K/1%
AMP0P
VHALF
R3
9.1K/1%
J1
CON6
VDD5
NSS
MOSI
RXD
SCLK
GND
J2
CON4
VDD5
FICEK
FICED
GND
J3
CON4
VCC
GND
VSP
FG
I_shunt
R4
2.7K/1%
R6
9.1K/1%
AMP2O
C2
100pF
AMP2M
Iv+
R16
2.7K/1%
AMP2P
VHALF
R17
9.1K/1%
I_shunt
R7
2.7K/1%
R8
9.1K/1%
AMP1O
C3
100pF
AMP1M
Iu+
R18
2.7K/1%
AMP1P
VHALF
R19
9.1K/1%
VCC
C4
47uF/50V
C5
0.1uF/50V
R9
30K
R10
3K
R11
1K
OVP
C6
0.1uF
U1
FU6831
VDD5
R35
10K
R34
10K
C7
C8
1uF
D1
SMAJ36
VCC
C12
1uF/50V
C13
1uF
R20
1.5K
R25
37K
C9
0.1uF
AMP2M 1
AMP2O 2
OVP 3
4
5
AMP0O 6
AMP0M 7
AMP0P 8
VHALF 9
10
11
12
VSS
VCC
VBB
L_W
L_V
L_U
H_PW
H_PV
H_PU
NC
PWM
ERR
C10
1uF
C11
1uF
VSP
R30
NC
C17
NC
R31
100R
PWM
C20
NC
C18
1uF
R33
10K
R32
10K
R34
100R
FG
C19
NC
Q2
LED
R36
5.1K
U7
LM317
9-36V
VCC
R38
100R
IN
OUT
VD1
OUT
5V/0.3A
Vout
L2
VDD5
C21
1uF/50V
C22
1uF/50V
R40
3K
R39
1K
C23
0.1uF
C25
22uF/16V
U2
AO4611
H_PU
R12
51R
L_U
R13
51R
R14
10K
R15
10K
D1
D1
G1
G2
D2
D2
S1
S2
U
Iu+
RS1
R220
U3
AO4611
H_PV
R21
51R
L_V
R22
51R
R23
10K
R24
10K
V
Iv+
RS2
R220
U4
AO4611
H_PW
R26
51R
L_W
R27
51R
R28
10K
R29
10K
W
I_shunt
RS3
R220



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