电子扇调试手册

三相电机控制 MCU
FU6865-Q

峰岹科技(深圳)股份有限公司

目录

目录 2
1 概述 4
2 硬件原理与参数配置 5
2.1 硬件原理图 5
2.1.1 电源电路 6
2.1.2 芯片电路 6
2.1.3 反电动势检测电路 7
2.1.4 功率驱动电路 8
2.1.5 运放配置电路 9
2.1.6 母线电压采样电路 9
3 软件架构 10
3.1 电机状态机流程图 10
3.2 程序流程图 12
3.3 程序说明 12
3.3.1 Main 函数: 12
3.3.2 1ms 定时中断 12
3.3.3 FOC 中断 13
3.3.4 CMP3 中断 13
3.3.5 Timer3 中断 13
4 调试步骤 14
4.1 配置电机参数 14
4.1.1 电机参数 14
4.1.2 电机参数测量方法 14
4.1.3 对应程序 15
4.2 芯片内部参数配置 15
4.3 硬件参数配置 15
4.4 保护参数配置 17
4.5 启动参数配置 17
4.6 硬件驱动电路检测 19
4.7 电流环调试 19
4.8 增加 PWM 功能 20
4.9 其他功能 22

4.10 可靠性测试 22
4.10.1 功能可靠性 22
4.10.2 保护可靠性 22
4.10.3 启动稳定性 22
5 功能介绍 23
5.1 启动调试 23
5.1.1 Omega 启动 23
5.1.2 启动常见问题&解决方式 25
5.2 保护介绍 25
5.2.1 过流保护 25
5.2.2 电压保护 26
5.2.3 缺相保护 26
5.2.4 堵转保护 27
5.2.5 过温保护 28
5.2.6 偏置电压保护 28
5.2.7 其他保护 29
6 其他常见功能调试 29
6.1 限功率功能 29
7 方案调试难点&解决方法 30
8 修改记录 31
9 版权说明 32

1 概述

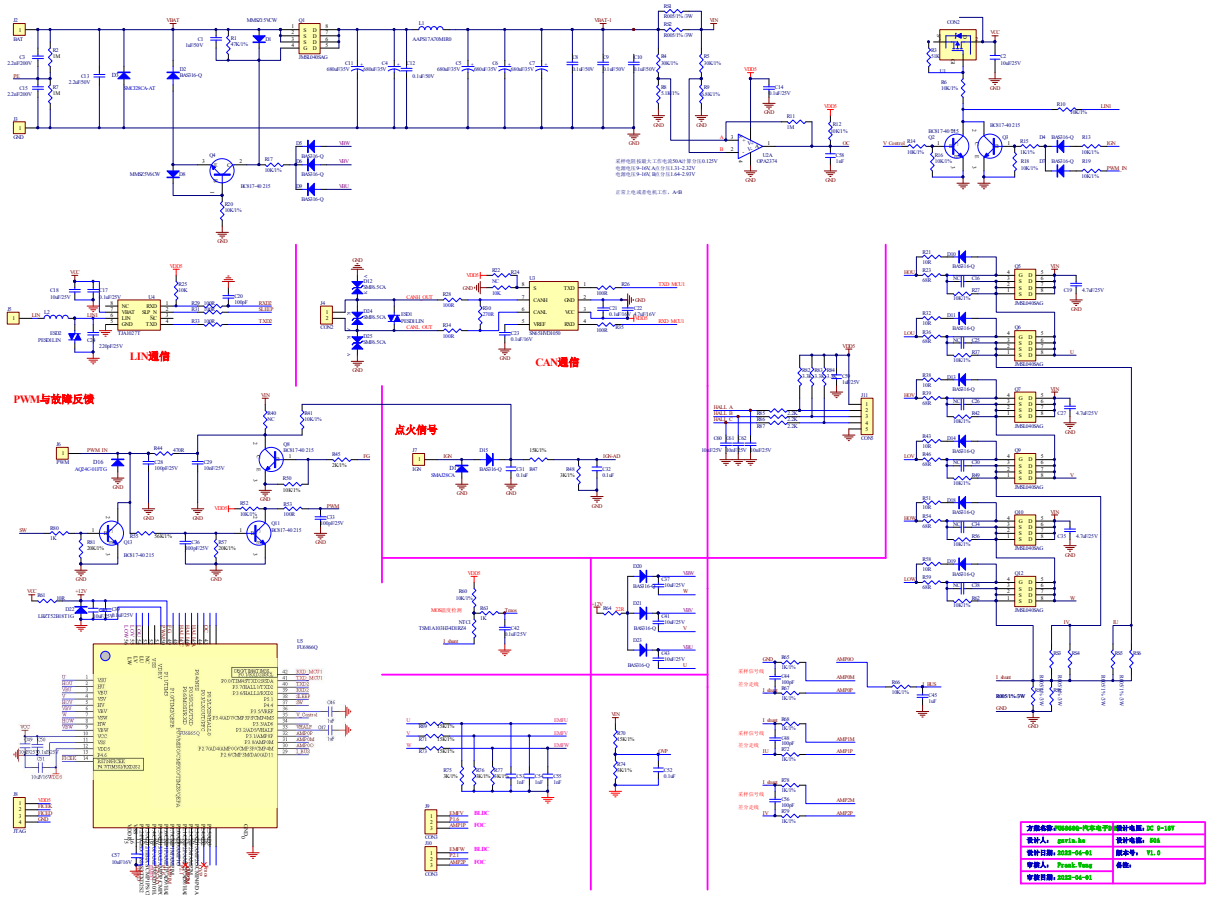
本调试手册详细介绍了如何使用峰岹科技的 FU6861Q2 芯片，在吸尘器专用 DEMO 板子上，对直流无刷吸尘器电机进行无感 FOC 驱动控制。阅读手册时，第二章节硬件原理跟第三章节软件原理可以大致先浏览一遍，重点放在第四章调试步骤。

涉及的软/硬件

软/硬件和模块	名称	章节	备注
软件	FU6865Q 电子扇应用方案标准程序	全部	调试需在该工程软件上进行
硬件	FU6866Q-QCDZ-DEMO_V1.0	全部	调试需在该硬件上进行

2 硬件原理与参数配置

2.1 硬件原理图



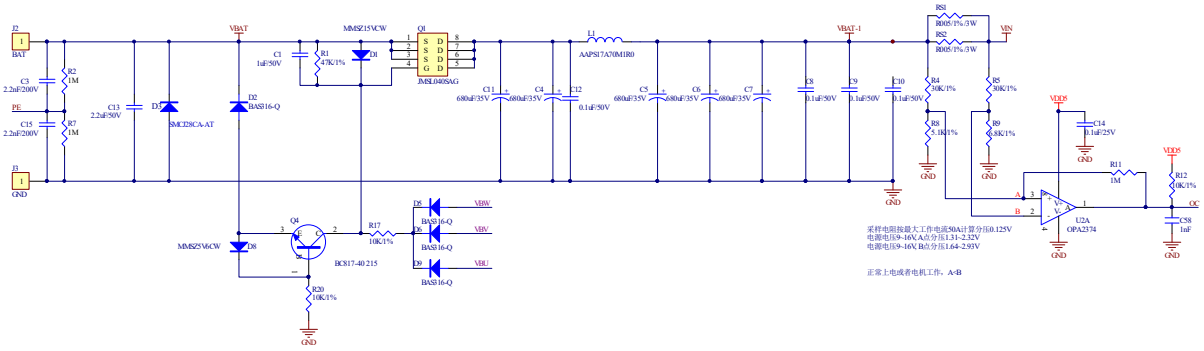
使用方式:

该板子为电子扇应用方案的专用 DEMO 板子，直接上电即可使用。

注意事项:

根据具体电机电压和电流大小，合理配置母线电压比，运放放大倍数，采样电阻，反电动势检测电路分压比。

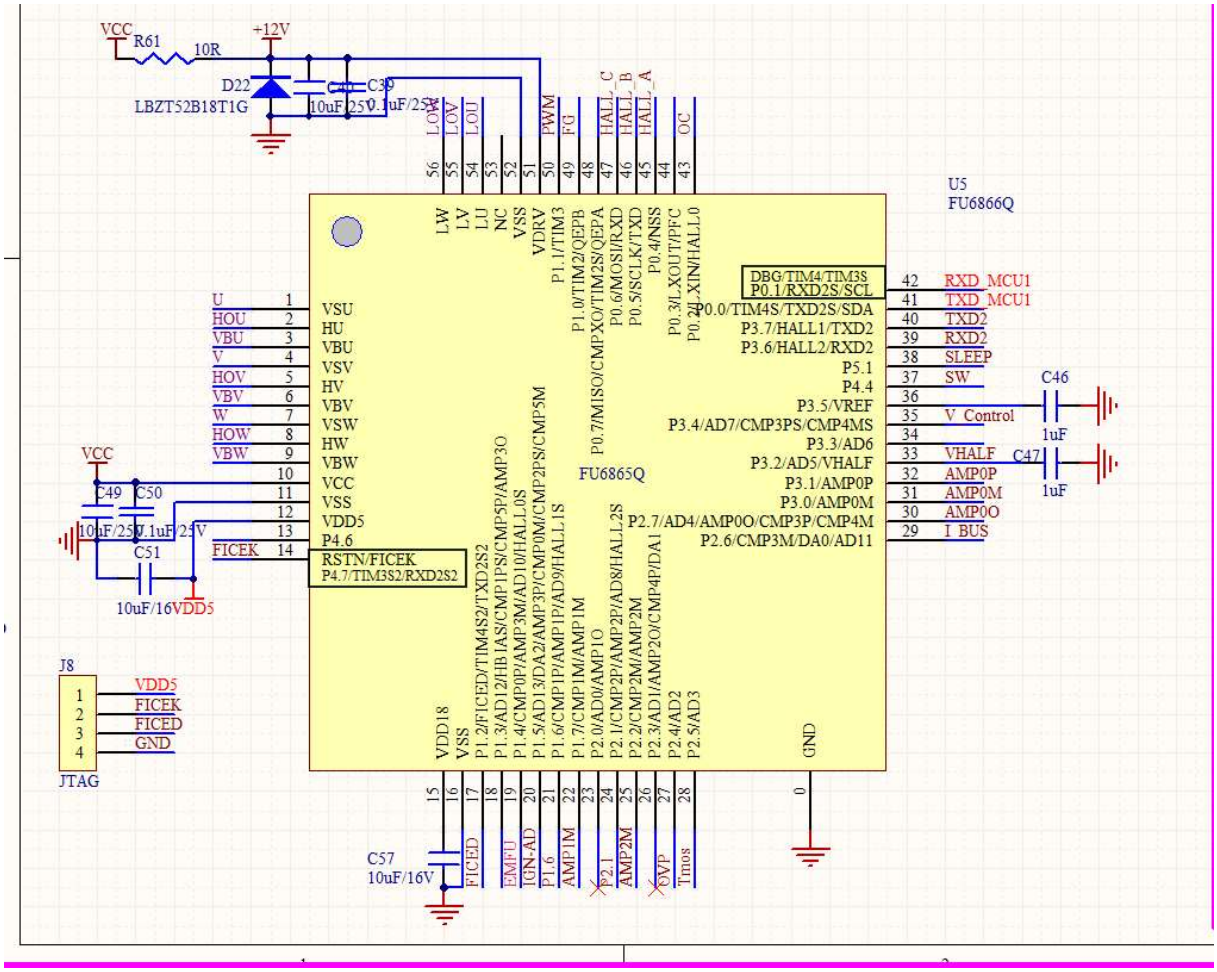
2.1.1 电源电路



使用方式:

直流电源正极接 BAT 端，负极接 GND 端。

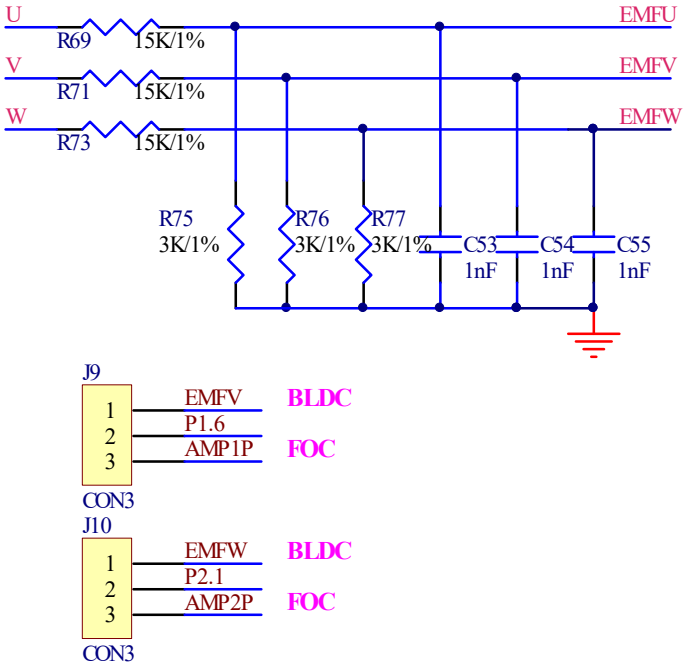
2.1.2 芯片电路



使用方式:

FU6866Q 应用于中低压 6-NMOSFET 驱动应用。其中 J8 为烧录线接口。

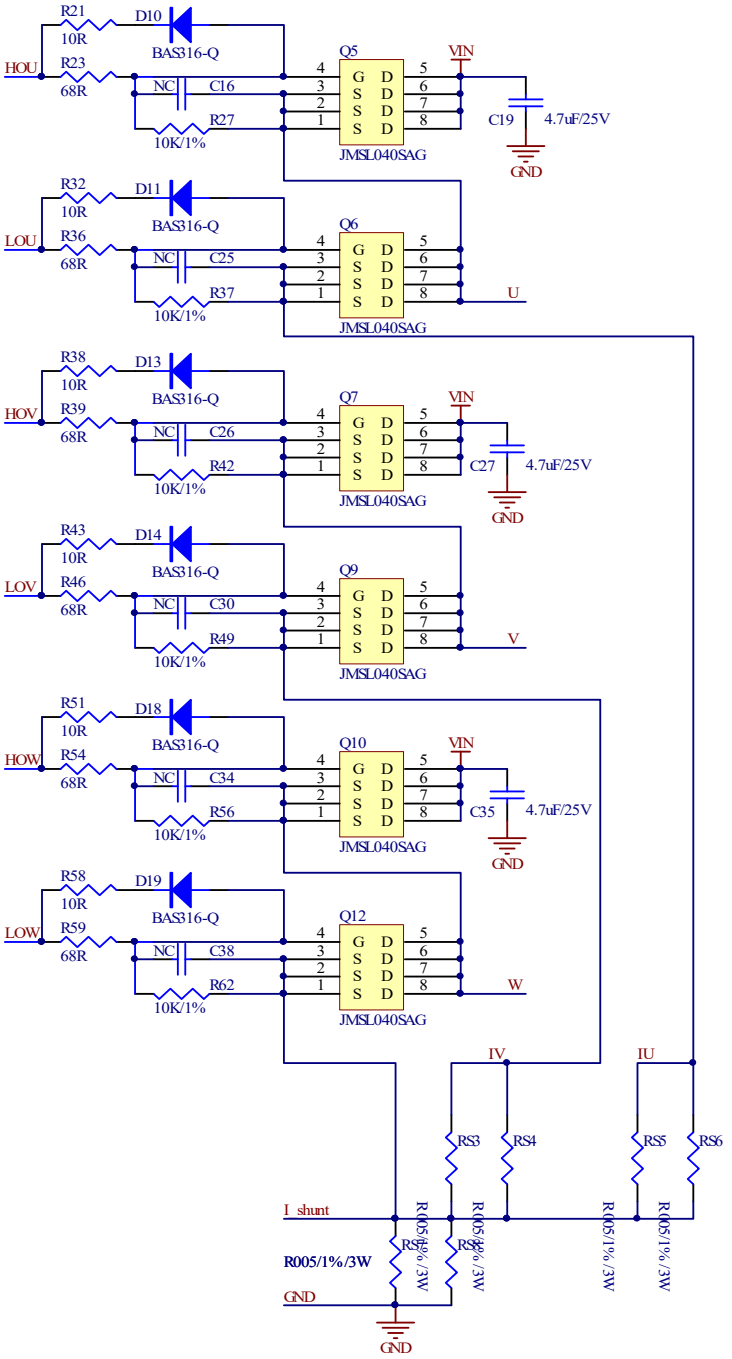
2.1.3 反电动势检测电路



注意事项:

该电路是检测反电动势的。因此，顺逆风的检测方式只能选择反电动势 BEMF 检测。

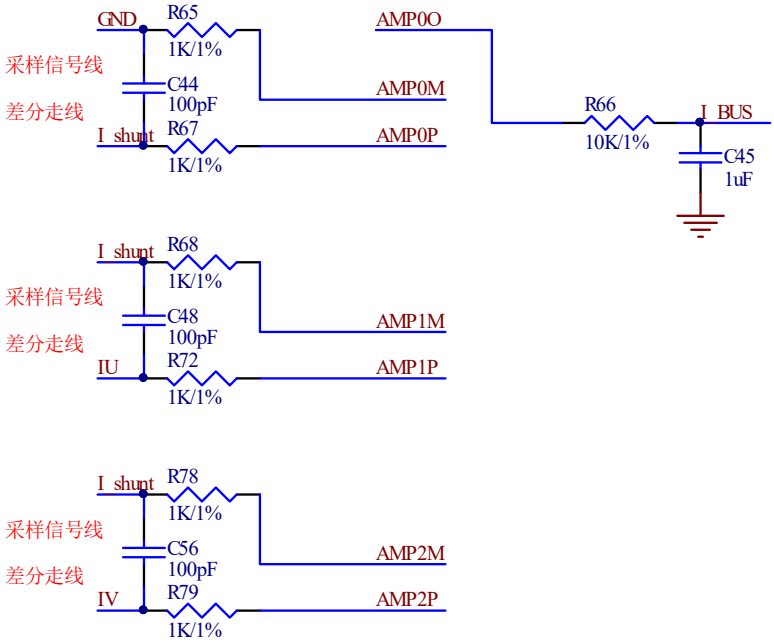
2.1.4 功率驱动电路



注意事项:

最大电流情况下，采样电阻功率不能超过额定功率的 80%。

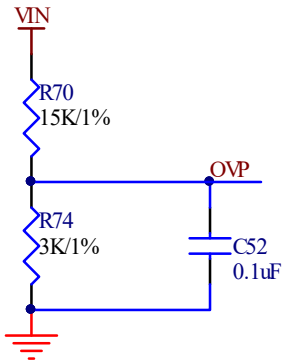
2.1.5 运放配置电路



注意事项:

1. R65、R67、R68、R72、R78、R79 需要用 1%精度电阻；
2. 放大倍数需要软件配置；
3. 最大采样电流 = (VREF - VHALF)/放大倍数/采样电阻值；
4. 最大采样电流一般设置为最大母线电流的 4 倍左右。

2.1.6 母线电压采样电路



注意事项:

1. R252 参数不需要进行调整；
2. R70、R74 需要用 1%精度电阻；
3. 最大采样电压 = (R70 + R74)/(R74)*VREF；

4. 最大采样电压一般选择为 2 倍的最大应用电压，OVP 此处的电压需要低于 0.8*VREF。

3 软件架构

3.1 电机状态机流程图

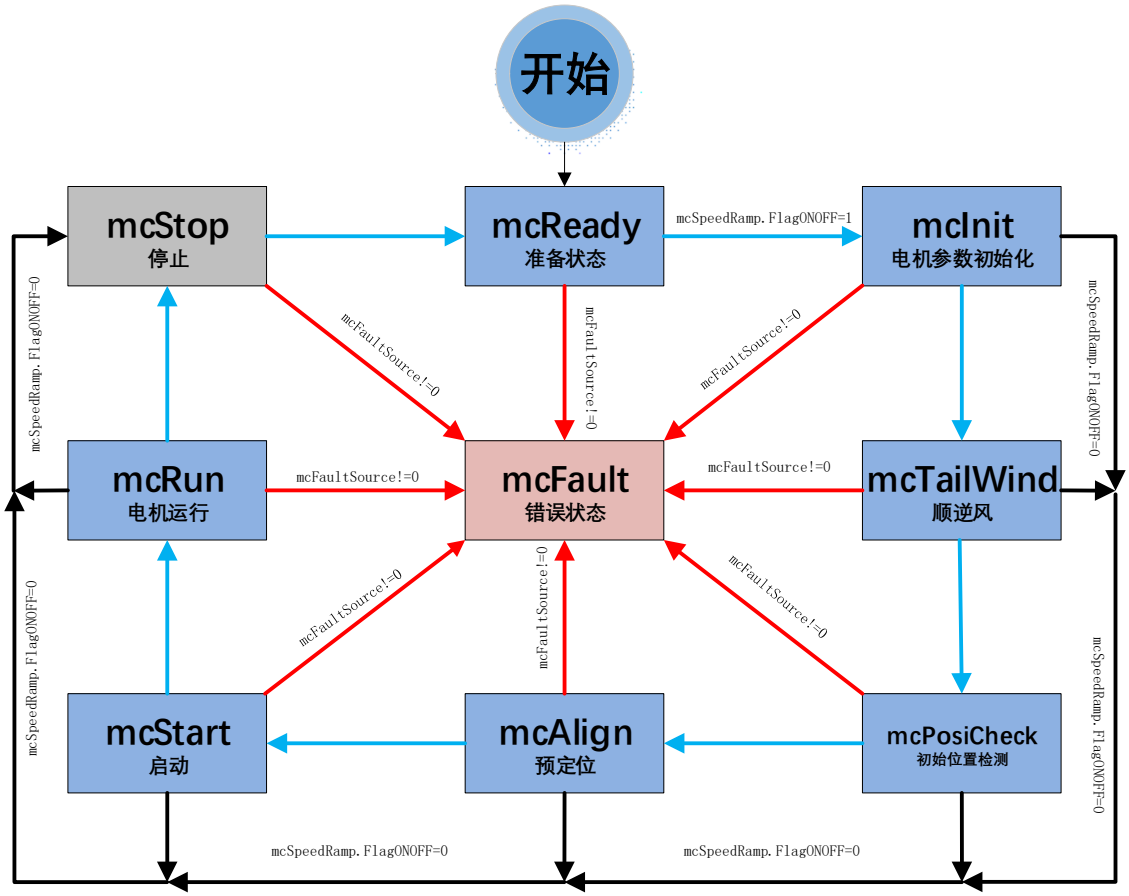


图 3-1 电机状态机流程图

如图所示，电机状态机分为三条路径:

1. 运行: mcReady -> mcInit -> mcTailWind -> mcPosiCheck -> mcAlign -> mcStart -> mcRun；
2. 停机: mcInit、mcCharge、mcAlign、mcStart、mcRun 状态下如果检测到关机信号则会切入到 mcStop 状态进行降速关机；
3. 故障: 所有状态下发生故障均会跳转至 mcFault 状态，在 mcFault 状态将不再进行故障检测，因此不支持多故障并发的同时上报。

说明:

1. mcReady: 准备状态，等待开机命令，如果开机使能则跳转到 mcInit 状态；
2. mcInit: 相关变量和 PI 初始化，关闭电流，母线采样的外部 ADC 触发，然后跳转到下一状态；

3. mcTailWind: 顺逆风检测状态，检测到顺风时，直接切到 mcRun 状态运行；检测到逆风时，先刹车再往下执行(吸尘器没有逆风的情况)；检测到静止时，往下执行；
4. mcPosiCheck: 初始位置检测状态，检测电机的初始位置，再正常启动；
5. mcAlign: 预定位状态, 该状态下控制器输出恒定的电流将电机强行拖动到固定的角度上。定位结束则跳入下一个状态 mcStart；
6. mcStart: 启动状态，该状态主要用于电机的启动代码配置，对相关寄存器代码与变量进行配置之后则转入下一个状态 mcRun。电机启动过程由 ME 内核实现；
7. mcRun: 运行状态，该状态包含: 电机启动阶段,电机运行阶段，电机速度的控制在该状态进行；
8. mcStop: 停机状态，该状态用于停机操作，高速进行刹车降速，速度降低到比较低的转速之后关闭输出，切入到 mcReady 状态，等待新的开机命令；
9. mcFault: 错误状态，当发生保护时，程序会记录错误源并且状态机会跳转到错误状态关机保护，当错误源被清掉时，会切入到 mcReady 状态，等待新的开机命令。

注意事项:

1. 电机状态机一共分为 8 个状态，状态之间只允许固定的状态跳转 例如: mcReady 状态只能向 mcInit 和 mcFault 状态跳转；
2. 特别的，mcTailWind，mcPosiCheck，mcAlign 三个状态都有使能位，当没使能时，直接跳转到下一个状态。例如: mcPosiCheck 没使能，mcAlign 也没使能时，mcTailWind 直接跳转到 mcStart 状态。

3.2 程序流程图

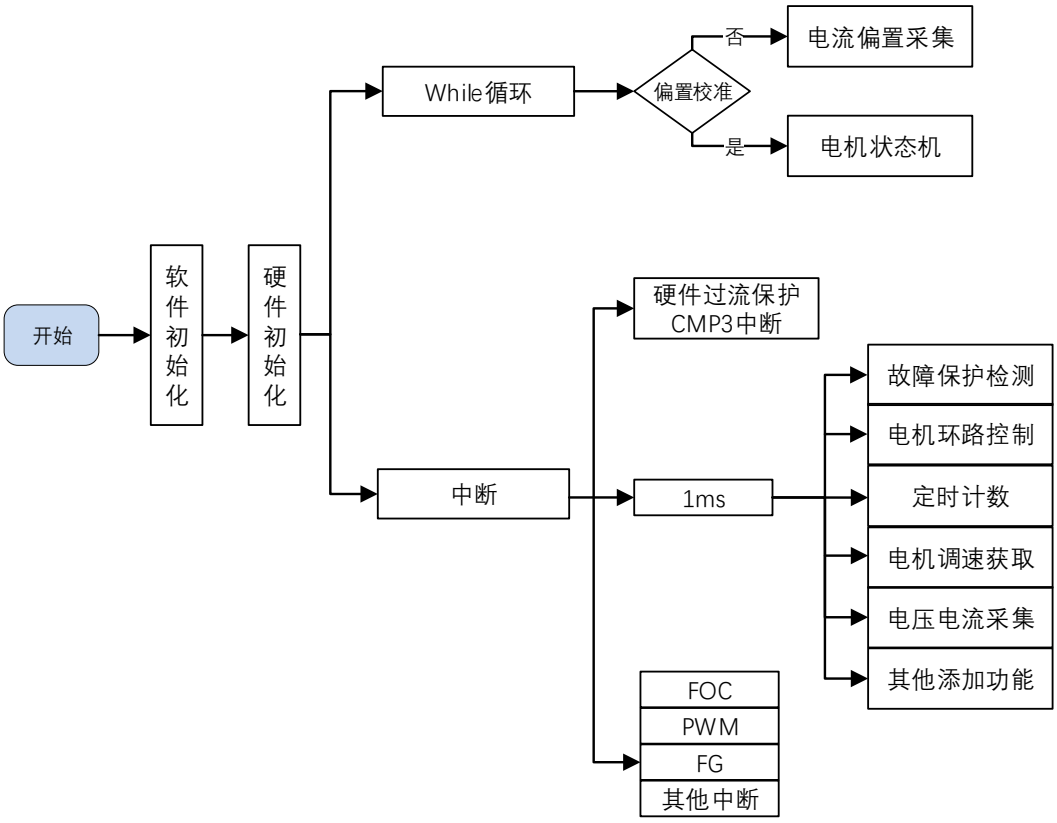


图 3-2 程序执行流程图

3.3 程序说明

3.3.1 Main 函数:

程序初始化 -> 偏置电压检测 GetCurrentOffset() + 电机运行控制 MC_Control()。

3.3.2 1ms 定时中断

程序中调速、故障保护检测、母线电流、母线电压采集等功能都在 1ms 中断中调用，包括以下函数:

```
Speed_response();          // 环路控制函数
PWMInputCapture();         // PWM 调速功能
VSPSample();               // 模拟电压调速功能
KeyScan();                 // 按键调速功能
ONOFF_Starttest();         // 启停测试验证启动可靠性
StarRampDealwith();        // 电机启动 ATO 爬坡控制
Fault_Detection();         // 故障检测
```

3.3.3 FOC 中断

FOC 中断，即载波中断，主要处理一些时序比较快的程序，如调用除法器等。

3.3.4 CMP3 中断

比较器 3 中断主要是处理硬件过流保护，具体原理可参考章节 5.2.1。

3.3.5 Timer3 中断

Timer3 中断主要是 PWM 占空比的获取，通过该中断获取到 PWM 的高电平 TIM3__DR 跟 PWM 的周期值 TIM3__ARR，之后再通过计算算出 PWM 的占空比大小。

4 调试步骤

4.1 配置电机参数

4.1.1 电机参数

1. 电机极对数 Pole_Pairs；
2. 电机的相电阻 RS、相电感 LD、LQ；
3. 电机速度基准，速度基准 MOTOR_SPEED_BASE = 2*电机额定转速。

4.1.2 电机参数测量方法

1. 极对数 Pole_Pairs: 电机设计时需给出的参数；
2. 相电阻 Rs: 万用表或者电桥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；LD = LQ = Ls；
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:

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

其中，P 为电机极对数。

示例，测量反电动势波形如下:

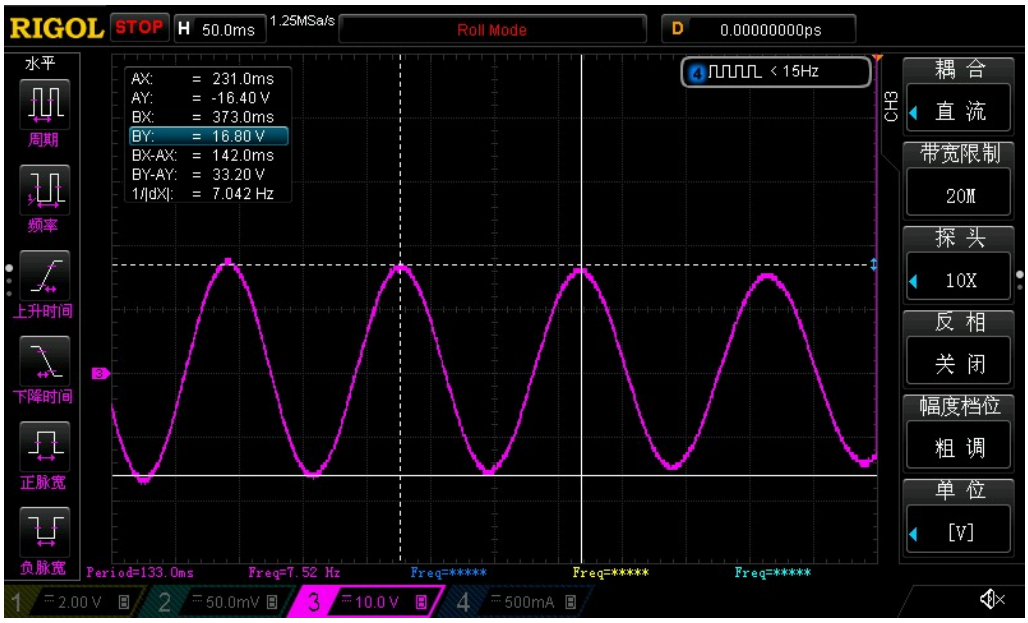


图 4-1 反电动势波形

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

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

5. 速度基准 MOTOR_SPEED_BASE: 速度基准一般设置为电机最大转速的 2 倍左右，该值会影响启动等性能，一般需要提前定好之后，后面不要轻易改动。

4.1.3 对应程序

```
/* ------------------------------------------------------------------------------ */
/*电机参数值*/
#define Pole_Pairs                  (1.0)                                  // 极对数
//#define RS                        (0.50)                                 // 电机相电阻/2, ohm
//#define LD                        (0.000500)                             // D轴电机相电感/2,H
#define RS                          (0.010)                                // 电机相电阻/2, ohm
#define LD                          (0.000010)                             // D轴电机相电感/2,H
#define LQ                          LD                                     // Q轴电机相电感/2,H

#define MOTOR_SPEED_BASE            (16000.0)                             // (RPM) 速度基准
// 若选择AO自适应观测器 则无需填写Ke
#define KeVpp                       (1.832)                                // (V)       反电动势测量的峰峰值
#define KeT                         (68.40)                                // (ms)      反电动势测量的周期
#define Ke                          (Pole_Pairs * KeVpp * KeT / 207.84) // (V/KRPM) 反电动势常数
/*------------------------------------------------------------------------------*/
```

4.2 芯片内部参数配置

```
/* ------------芯片参数值------------------------------------------------------------ */
/*PWM Parameter*/
#define PWM_FREQUENCY               (26.0)                                 // (kHz) 载波频率

/*deadtime Parameter*/
#define PWM_DEADTIME                (1.0)                                  // (us) 死区时间

/*single resistor sample Parameter*/
#define MIN_WIND_TIME               (PWM_DEADTIME + 0.9)                   // (us) 单电阻最小采样窗口，建议值死区时间+0.9us
/*double resistor sample Parameter*/
#define DLL_TIME                    (PWM_DEADTIME + 0.3)                   // (us) 双电阻最小脉宽设置，建议值为死区时间值+0.2us以上
```

注意事项:

1. 载波频率一般需要设置为最大电周期 10 倍左右，载波频率会影响启动，MOS 温升等等，调试之前需要选择好合适的载波频率。默认可以写 16K
2. 死区大小根据实际的 MOS 开关速度设置，保证没有直通风险；
3. 最小采样窗口设置，最小窗口最小需要大于 2 倍的死区，小于载波周期的 1/16，即 1000/16/PWM_FREQUENCY > MIN_WIND_TIME > 2*PWM_DEADTIME；
4. 正反转设置，根据实际接线设置，吸尘器电机反转会有高频噪声，且出风较正转小很多，如果电机反转了，则将该位取反即可。

4.3 硬件参数配置

1. 通过电机的电压范围和功率范围确认母线分压比、采样电阻值、放大倍数。
2. 电阻阻值跟放大倍数选取规则:
   1) 母线分压电阻:
      - 分压比不宜太小: 一般建议最大采集电压为 0.8*VREF，如某电机的最大电压为 30V，ADC 基准 VREF 为 4.5V，此时分压比建议不低于: 30/0.8/4.5 = 8.33；如果分压比大小，如分压比为 5，则 30V 时，经过分压后到 AD 口的电压为 6V，此时溢出了。
      - 分压比不宜太大: 分压比太大的话会导致 AD 采集电压精度不够，如最大电压为 30V，当分压比

为 40 时，经过 AD 口的电压为 30V/40V = 0.75V，28V 时为 0.7V，此时精度比较低，而且 AD 还有 4.5 - 0.75 = 3.75V 的余量。

2) 采样电阻与放大倍数:

最大采集电流 = VREF/HW_RSHUNT/HW_AMPGAIN；这里要注意的是，最大采集电流不是电源上显示的电流(电源上显示的是滤波后的)，而是流经采样电阻的电流。

- 采样电阻不宜太大: 太大的话容易导致采样溢出，或者本身的功率超过范围；2512 封装的采样电阻常见功率为 1W 或者 2W，1206 封装电阻的功率常见位 1/4W，选择时，要注意流经采样电阻的功率 I^2R 不要超过该功率。
- 采样电阻不宜太小，太小的话精度不够
- 放大倍数结合采样电阻调整，先确定了采样电阻，再去调整放大倍数

其中，HW_RSHUNT 为采样电阻，HW_AMPGAIN 为放大倍数。

3. 母线分压比、采样电阻值、放大倍数对应填写到程序中(在 Customer.h 文件)。

```
/* 放大倍数设置 */
/* -----(AMP2x)---- 内部PGA放大2倍-- */
/* -----(AMP4x)---- 内部PGA放大4倍-- */
/* -----(AMP8x)---- 内部PGA放大8倍-- */
/* -----(AMP16x)--- 内部PGA放大16倍- */
/* -----(xxxxxx)--- 外部放大模式填写相应倍数- */
#define HW_AMPGAIN                      (AMP4x)

/* 参考电压设置 */
/* -----(VREF3_0)--- 参考电压设置为3.0V-- */
/* -----(VREF4_0)--- 参考电压设置为4.0V-- */
/* -----(VREF4_5)--- 参考电压设置为4.5V-- */
/* -----(VREF5_0)--- 参考电压设置为5.0V-- */
#define HW_ADC_VREF                     (VREF4_5)                       // (V)  ADC参考电压

/* -----电流采样模式选择----- */
/* -----(Single_Resistor)---- 单电阻电流采样模式- */
/* -----(Double_Resistor)---- 双电阻电流采样模式- */
/* -----(Three_Resistor)----- 三电阻电流采样模式- */
#define Shunt_Resistor_Mode             (Single_Resistor)

/* 基准电压VREF对外输出使能 */
/* -----(Disable)---禁止-- */
/* -----(Enable)----使能-- */
#define VREF_OUT_EN                     (Enable)

/* VHALF偏置电压设置 */
/* -----(VHALF1_8)-----   VHALF电压设置为1/8VREF---- */
/* -----(VHALF1_4)-----   VHALF电压设置为1/4VREF---- */
/* -----(VHALF25_64)---   VHALF电压设置为25/64VREF-- */
/* -----(VHALF1_2)-----   VHALF电压设置为1/2VREF---- */
#define HW_VHALF_SEL                    (VHALF1_2)

/* VHALF输出使能 */
/* -----(Disable)---禁止-- */
/* -----(Enable)----使能-- */
#define VHALF_OUT_EN                    (Enable)

/* 母线电压采样相关硬件参数 */
#define RV1                             (20.0)                          // (kΩ) 母线电压分压电阻1
#define RV2                             (2.0)                           // (kΩ) 母线电压分压电阻2
```

其中，

1) 母线分压比 = (RV1 + RV2)/RV2

4.4 保护参数配置

1. 电流保护设置:
   - 硬件过流: 根据功率器件的最大电流值，设置硬件过流保护值，一般硬件过流保护值 OverHardcurrentValue 设置大于母线最大电流值，小于功率器件最大电流值。
   - 软件过流: OverSoftCurrentValue 一般设置比硬件过流小一点即可，软件过流为软件触发，保护时间不及硬件过流。
2. 设置过欠压保护跟保护恢复参数，详细设置参考章节 5.2.2;
3. 关闭上述保护的其他保护，防止启动的时候误触发，后面添加需要的保护再确认，其中过流保护是一定要开的，因此没有使能位;
4. 将参数对应填写到程序中(在 Protect.h 文件)。

```
/* ------------------------------------保护使能设置------------------------------------------------------ */
#define VoltageProtectEnable          (1)                                   // 电压保护, 0,不使能; 1, 使能
#define StallProtectEnable            (0)                                   // 堵转保护, 0,不使能; 1, 使能
#define PhaseLossProtectEnable        (0)                                   // 缺相保护, 0,不使能; 1, 使能
#define GetCurrentOffsetEnable        (0)                                   // 偏置电压保护, 0,不使能; 1, 使能
#define TemperatureProtectEnable      (0)                                   // 温度保护使能
#define OverSpeedProtectEnable        (0)                                   // 超速保护使能

/* ------------------------------------保护参数值-------------------------------------------------------- */

/* -----硬件过流保护方式选择----- */
#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_Mode                  (Compare_DAC)                         // 硬件过流值的来源
#define OverHardcurrentValue          (40.0)                                // (A) DAC模式下的硬件过流值 Imax = VHALF / HW_RSHUNT / HW_AMPGAIN

/* -----软件过流保护----- */
#define OverSoftCurrentValue          I_Value(30.0)                         // (A) 软件过流值
#define OverSoftCurrentTime           (10)                                  // 软件过流检测次数, 初步设定值10
#define OverSoftCurrentClrTime        (1000)                                //(ms)每隔 OverSoftCurrentClrTime 检测次数清零, 初步设定值1000

/* -----偏置电压保护----- */

 #define GetCurrentOffsetValue         _Q14(0.05)                            // (单位:%)偏置电压保护误差范围, 超过该范围保护

/* -----过欠压保护----- */
#define Over_Protect_Voltage          (30.5)                                // (V) 直流电压过压保护值
#define Over_Recover_Vlotage          (29.5)                                // (V) 直流电压过压保护恢复值
#define Under_Protect_Voltage         (12.5)                                // (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
```

4.5 启动参数配置

启动参数都先采用自带的**默认参数**，等启动有问题或者启动不顺的时候再做调整。启动常见的问题即参数调整可以参考章节 5.1。

Customer.h*

```
/* ------------------------------------------------------------------------------------------------------
                                        启动相关参数配置区域
------------------------------------------------------------------------------------------------------ */
/* -----启动电流----- */
#define ID_Start_CURRENT              I_Value(0.0)                          // (A) D轴启动电流
#define IQ_Start_CURRENT              I_Value(12.0)                         // (A) Q轴启动电流

/* -----运行电流----- */
#define ID_RUN_CURRENT                I_Value(0.0)                          // (A) D轴运行电流
#define IQ_RUN_CURRENT                I_Value(8.0)                          // (A) Q轴运行电流
```

1. 启动电流: 一般 ID_Start_CURRENT 固定设置为 0，IQ_Start_CURRENT 根据实际电机设置确认;

注意事项:

IQ_Start_CURRENT，不能过小否则启动力矩太小导致启动失败。

IQ_Start_CURRENT，不能过大否则启动过冲还会引入启动噪声。

2. 切换电流: IQ_RUN_CURRENT 只决定一瞬间的电流。通过实际观测相电流，可通过 IO 口翻转确认在切环瞬间是否存在电流不平滑，可以适当调整 IQ_RUN_CURRENT 解决；
3. 启动 ATO: 由于在较低转速下估算器输出存在误差，此时需要设置 ATO_BW 是(速度带宽滤波值)，以限制 FOC 估算器的最大转速输出；

Customer.h*

```
113 #define IQ_RUN_CURRENT              I_Value(8.0)                    // (A) Q轴运行电流
114
115 /* -----启动的ATO参数设置----- */
116  #define ATO_BW                     (10.0)                        // 观测器带宽的滤波值，经典值为1.0-200.
117  #define ATO_BW_RUN                 (70.0)
118  #define ATO_BW_RUN1                (120.0)
119  #define ATO_BW_RUN2                (140.0)
120  #define ATO_BW_RUN3                (160.0)
121  #define ATO_BW_RUN4                (180.0)
```

注意事项:

对于吸尘器而言，启动的前 3 个 ATO 影响比较明显，需要根据实际情况调整。因为开的是 AO 观测器，所以第一个 ATO_BW 不用太大即可。

4. 速度带宽滤波值 SPD_BW；

Customer.h*

```
122
123 /* -----速度带宽的滤波值----- */
124 #define SPD_BW                      (15.0)                          // 速度带宽的滤波值，经典值为5.0-40.0
125
126 /* -----SMO运行最小转速----- */
127 #define MOTOR_SPEED_SMOMIN_RPM      (1200.0)                        // (RPM)
```

注意事项:

SPD_BW 一般不需要调整。

5. Omega 启动参数设置，影响启动的电流频率，即电机的启动加速度；

Customer.h*

```
128
129 /* -----OMEGA启动参数----- */
130  #define Motor_Omega_Ramp_ACC_Antiwind  (2.0)                    // 顺风时omega启动的增量
131  #define Motor_Omega_Ramp_ACC           (10.0)                   // omega启动的增量
132  #define MOTOR_OMEGA_ACC_MIN            (200.0)                  // (RPM) omega启动的最小切换转速
133  #define MOTOR_OMEGA_ACC_END            (500.0)                  //  RPM) omega启动的限制转速
134
135 /* motor loop control speed value */
136 #define MOTOR_LOOP_RPM              (2000.0)                        // 2000 (RPM) 由mode 0到mode1切换转速，即闭环切换转速
```

注意事项:

1) Motor_Omega_Ramp_ACC 参考值范围 10 ~ 50；
2) MOTOR_OMEGA_ACC_MIN 参考值范围 200 ~ 500；
3) MOTOR_OMEGA_ACC_END 参考值范围 500 ~ 3000；
4) MOTOR_LOOP_RPM 需要大于 MOTOR_OMEGA_ACC_END，参考值范围 2000 ~ 4000。

6. 电流环 PI: 电流环 PI 分启动的电流环 PI 跟运行时的电流环 PI；

```
/* -----------电流环参数设置值------------------------------------------------------------------ */
#define DKP                         _Q12(1.0)                       // 运行DQ轴KP
#define DKI                         _Q15(0.01)                      // 运行DQ轴KI
#define QKP                         _Q12(1.0)                       // 运行DQ轴KP
#define QKI                         _Q15(0.01)                      // 运行DQ轴KI
/* D轴参数设置 */
```

注意事项:

1) 启动的电流环 PI，影响电机的启动；

2) 运行的电流环 PI，影响电流的稳定性，也影响效率；

3) DQKP 建议范围 3.0 ~ 0.1；

4) DQKI 建议范围 0.05 ~ 0.001。

7. DQ 轴最大输出限幅:D 轴影响电机的磁通，Q 轴影响电机的转矩。

```
/* D轴参数设置 */
#define DOUTMAX                 _Q15(0.6)                   // D轴最大限幅值，单位：输出占空比
#define DOUTMIN                 _Q15(-0.6)                  // D轴最小限幅值，单位：输出占空比
/* Q轴参数设置 */
#define QOUTMAX                 _Q15(0.98)                  // Q轴最大限幅值，单位：输出占空比
#define QOUTMIN                 _Q15(-0.98)                 // Q轴最小限幅值，单位：输出占空比
```

注意事项:

1) FOC__UQ 反馈电机已经输出是否饱和；

2) FOC__UD 正得越多表示角度越超前，可以通过增加补偿角(FOC_THECOMP)让电机角度超前，此时能提升最大转速，FOC__UD 是一个正值；

3) 过多的超前角度，会导致关机时候电流过冲，可以通过低压预警关机处理，也可以通过快速欠压保护处理；

4) 过多的超前角度，会导致效率变差，相同功率下，相电流幅值更大，需要合理设置补偿角度。

4.6 硬件驱动电路检测

Customer.h*

```
/* ----------------------------------------------------------------------------------------------------
                                        预定位相关参数配置区域
---------------------------------------------------------------------------------------------------- */
/* -----预定位测试模式，用于验证电机输出正常，或者测试预定位力矩是否足够----- */
#define AlignTestMode                   (0)                                     // 预定位测试模式

/* -----UDQ预定位是否关联电压----- */
/* -----宽电压的时候选择_防止高低电压时候定位力矩太小或太大----- */
#define Align_Associated_Vol_EN         (0)                                     // UDQ预定位是否关联电压

/* -----IQ预定位电流&KPKI设置(默认选择UD预定位)----- */
#define Align_Angle                     (0.0)                                   // (°) 预定位角度
#define Align_Time                      (0)                                 // (ms) 预定位时间，单位：ms

/* -----UDQ预定位参数设定----- */
#define   UDQMAX                  _Q15(0.05)                                    // 最低电压的UDQ定位Duty
#define   UDQMIN                  _Q15(0.08)                                    // 最高电压的UDQ定位Duty
```

Customer.h*

```
/* ----------------------------------------------------------------------------------------------------
                                调速模式&IPM&估算器模式&顺风模式&开环启动方式配置区域
---------------------------------------------------------------------------------------------------- */
/* -----调速模式选择----- */
/* -----(PWMMODE)----- */                                                       // PWM调速
/* -----(SREFMODE)----- */                                                      // 模拟调速
/* -----(NONEMODE)----- */                                                      // 直接给定值，不调速
/* -----(KEYMODE)----- */                                                       // 按键调速模式
/* -----(ONOFFTEST)----- */                                                     // 启停测试模式
#define SPEED_MODE                      (NONEMODE)
```

将AlignTestMode置1，调速模式选择NONEMODE开机，这个时候电机会跑到预定位状态，此时UVW三相会有固定的PWM波形输出，则硬件驱动电路正常。若没有输出，则要查找硬件问题。

4.7 电流环调试

1. 将环路选择为电流环；

```
/* -----------外环环路参数设置-------------------------------------------------------------- */
/* -----闭环方式选择----- */
/* -----(CURRENT_LOOP_CONTROL)---电流环 */
/* -----(SPEED_LOOP_CONTROL)-----速度环 */
/* -----(POWER_LOOP_CONTROL)-----功率环 */
/* -----(UQ_LOOP_CONTROL)--------UQ环   */
#define MOTOR_CTRL_MODE                (SPEED_LOOP_CONTROL)
```

2. 调速方式选择

```
/* -----------------调速与开关机控制 ----------------------------------------------------*/

/* -----调速模式选择----- */
/* -----(PWMMODE)-----PWM调速- */
/* -----(SREFMODE)----模拟调速- */
/* -----(NONEMODE)----直接给定值，不调速- */
/* -----(ONOFFTEST)---启停测试工具- */
#define SPEED_MODE                      (SREFMODE)
```

3. 烧录程序，上电启动电机，当电机启动不起来时(目前一般都能起来)，通过调整以下启动参数:
   - 启动电流: IQ_Start_CURRENT，电流不够时电机起不来，可以慢慢增加，也不要一次性给太大。
   - 启动到切环路的电流: IQ_RUN_CURRENT，给到稍微比启动电流小点即可
   - 影响启动频率的 ATO 和 Omega 的参数等等
4. 当上电，电机能跑后，加大电流环给定值，达到客户目标；
5. 确认电流环情况下最大功率、转速。

注: 母线电流采集的 AD 口要对应上，要根据实际硬件电路去修改。具体位置如下图。

Interrupt.c*

```
SetBit(ADC_CR, ADCBSY);                                                    //
/* -----母线电流采样----- */
Power_Currt = (ADC3_DR << 3);          根据实际情况修改AD口

if (Power_Currt > mcCurOffset.Iw_busOffset)
{
    Power_Currt   = Power_Currt - mcCurOffset.Iw_busOffset;
}
else
{
    Power_Currt   = 0;
}

/* -----母线电流滤波----- */
mcFocCtrl.mcADCCurrentbus  = LPFFunction(Power_Currt, mcFocCtrl.mcADCCurrentbus, 32);
```

常见问题即解决办法:

1) 加大电流给定，还是达不到客户要的最大功率值；

解决: 电流波形正弦的情况下，通过观测 FOC__UQ 是否饱和，如果饱和，且 FOC__UD 值比较大的话，通过调整补偿角 FOC_THECOMP(正负都调整看看)确认是否能达到客户需求。

2) 运行过程中，触发过流保护；

解决: 看相电流波形是否异常，看是否是设定值比较小正常触发了过流保护。如果没异常的情况下，查看硬件布线等是否有问题。

3) 相电流波形有抖动。

解决: 调整电流环 PI(即 DQKP，DQKI)的值，电流环 PI 和电流采样对于电流波形的稳定性影响比较大。

4.8 增加 PWM 功能

1. 一般电子扇 PWM 调速，调 PWM 的步骤为:
   1) 将调速模式修改为 PWM 调速。先根据客户给的曲线，调整最小跟最大功率值，以及对应的开关机 PWM 占空比，和最小最大占空比；

```
Customer.h*
213                                  调速模式&IPM&估算器模式&顺风模式&开环启动方式配置区域
214 ------------------------------------------------------------------------------------------------------------------------
215 /* -----调速模式选择----- */
216 /* -----(PWMMODE)----- */                                          // PWM调速
217 /* -----(SREFMODE)----- */                                         // 模拟调速
218 /* -----(NONEMODE)----- */                                         // 直接给定值，不调速
219 /* -----(KEYMODE)----- */                                          // 按键调速模式
220 /* -----(ONOFFTEST)----- */                                        // 启停测试模式
221 #define SPEED_MODE                  (PWMMODE)
```

2) 根据客户给的 PWM 曲线，调整开关机、最大最小占空比；

```
Customer.h*
57 /* ------------------------------------------------------------------------------------------------------------------------
58                                   PWM调速模式下相关参数配置区域
59 ------------------------------------------------------------------------------------------------------------------------
60 /* -----电机开机、关机的设置----- */
61 /* -----motor ON/OFF value----- */
62 #define OFFPWMDuty                  _Q15(0.09)                    // 关机PWM占空比，小于该占空比关机
63 #define OFFPWMDutyHigh              _Q15(1.0)                     // 关机PWM占空比，大于该占空比关机
64 #define ONPWMDuty                   _Q15(0.15)                    // 开机PWM占空比，大于该占空比时开机
65 #define MINPWMDuty                  _Q15(0.1)                     // 速度曲线上最小PWM占空比
66 #define MAXPWMDuty                  _Q15(0.85)                    // 速度曲线上最大PWM占空比
```

3) 根据客户给的最小最大转速。调整最大和最小转速：

```
248
249 /* 电机运行时最大最小转速 */
250
251 #define MOTOR_SPEED_MIN_RPM         S_Value(100.0)                // (RPM) 运行最小转速
252 #define MOTOR_SPEED_MAX_RPM         S_Value(6000.0)               // (RPM) 运行最大转速
253 #define MOTOR_SPEED_STOP_RPM        S_Value(500.0)                // (RPM) 运行最小转速
254
```

得到的曲线最低点为(MINPWMDuty，MOTOR_SPEED_MIN_RPM)，最高点为(MAXPWMDuty，MOTOR_SPEED_MAX_RPM)。

4) 确认客户是正 PWM 调速还是负 PWM 调速，正 PWM 调速: 转速随着占空比增大而增大；负 PWM 调速: 转速随 PWM 增大而减小。

```
Customer.h*  AddFunction.c  Interrupt.c
179 /* -----正PWMduty or 负PWMduty Choose----- */
180 /* -----  (PosiPWMDUTY)  ----- */                                 // 正PWMDuty
181 /* -----  (NegaPWMDUTY)  ----- */                                 // 负PWMduty
182 #define PWMDUTY_Choose              (PosiPWMDUTY)
183
184 /* -----PWM开关机滤波设置----- */
185 #define MotorOnFilterTime           (20)                          // 开机滤波，单位：ms
186 #define MotorOffFilterTime          (20)                          // 关机滤波，单位：ms
187
```

注意事项:

1) 根据 PWM 频率，在 Timer3 初始化的时候，选择合理的 Timer 分频；
2) 开关机占空比，要留有一定的滞回区间，如 10%开机，8%关机。留 2%的滞回区间。开机跟关机占空比如果一样的话，会导致时开时关；
3) 当 PWM 占空比获取不对时，看进入芯片引脚的 PWM 信号是否已经失真，有些如果滤波电容太大的话，会导致 PWM 信号失真；
4) PWM 信号有干扰的，尝试打开捕获 TIM 口的滤波功能，或者调整 PWM 硬件滤波电容，尽量靠近芯片引脚。

4.9 其他功能

1. 有其他功能，如 FG 输出等功能时，则对应添加即可；
2. 添加保护功能，根据客户需求使能缺相保护、堵转保护、过温保护，超速保护等。所有其他程序中还没添加的保护，则要额外再添加。具体保护介绍参考章节 5.2。

4.10 可靠性测试

4.10.1 功能可靠性

全部功能添加完成后，要再按照客户需求表重新测试确保没异常状态发生。

4.10.2 保护可靠性

保护添加之后，要验证保护都可以正常触发，且在电机运行时不会误触发。例如: 如果堵转保护的参数设置不合理，可能会导致电机在正常运行时也会误报堵转保护；或者是电机发生堵转后，不会触发堵转保护。

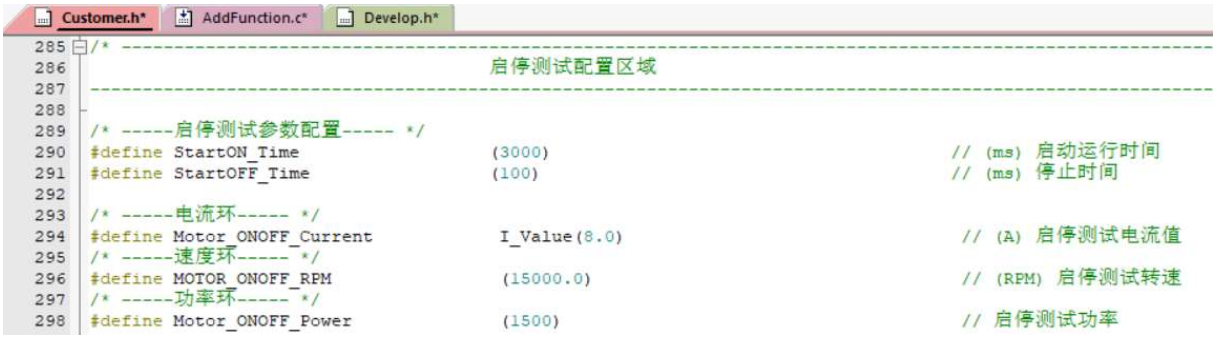
4.10.3 启动稳定性

在功能都基本调试完成之后，要做启动的可靠性测试，可先手动测试，手动测试没问题后，再进行老化测试。老化测试步骤:

1. 将 ONOFFTEST 打开；
2. 根据实际情况配置运行时间 StartON_Time 和停止时间 StartOFF_Time；
3. 调整 Motor_ONOFF_Power 的值可以修改启停的功率大小；
4. 先用工具堵住电机上电，看是否能正常触发堵转保护，且保护后电机不会重启，即验证了启停时如果触发保护电机不会二次重启；
5. 再次上电进行老化测试即可。最后根据电机是否处于停止状态判断启动是否有异常，启动失败后，电机会一直停机不再重启。一般测试 3000 次以上没问题则认为启动可靠(时间允许的情况下越多越好)。

Customer.h*

```
231 //------------------------------------------------------------------------------------------------------------------
232 /* -----调速模式选择----- */
233 /* -----(PWMMODE)----- */                                    // PWM调速
234 /* -----(SREFMODE)----- */                                   // 模拟调速
235 /* -----(NONEMODE)----- */                                   // 直接给定值，不调速
236 /* -----(KEYMODE)----- */                                    // 按键调速模式
237 /* -----(ONOFFTEST)----- */                                  // 启停测试模式
238 #define SPEED_MODE                  (ONOFFTEST)
```



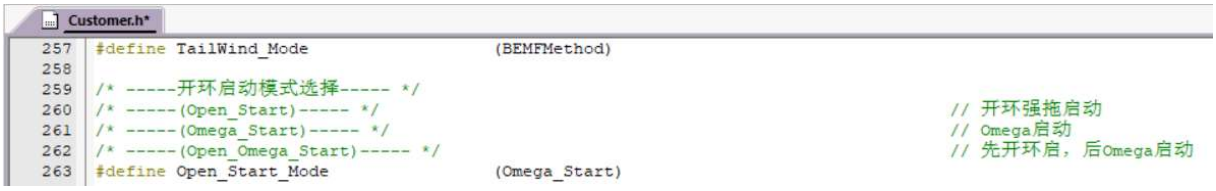
5 功能介绍

目前拿到初始版本程序，配置好电机参数，硬件参数后，给开机信号时，电机基本都能正常启动。若不能正常启动，则在排除是硬件问题的前提下，再调整启动参数。

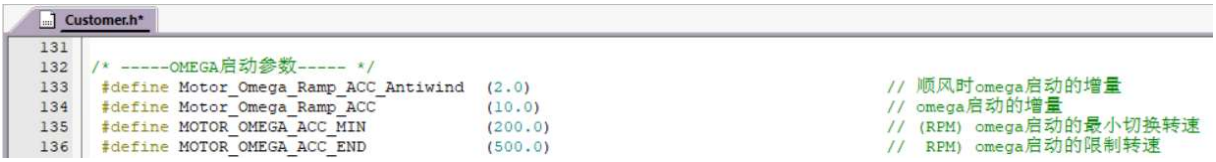
5.1 启动调试

5.1.1 Omega 启动

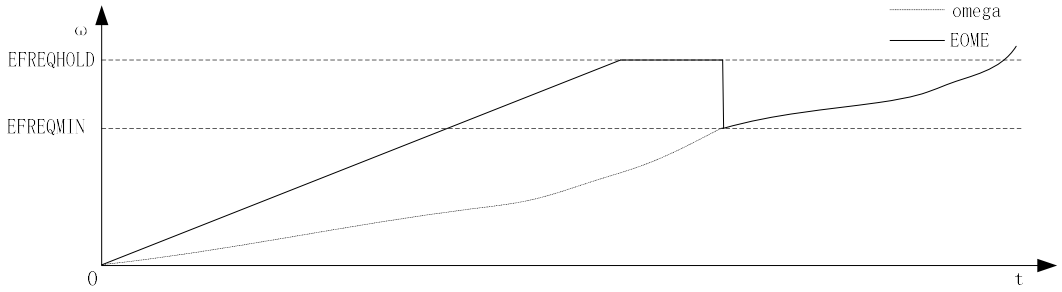
吸尘器选择 Omega 启动，程序对应默认即是该启动方式。



当估算器的估算速度 OMEGA 小于用户设定的最小值 FOC_EFREQMIN(对应 MOTOR_OMEGA_ACC_MIN 参数)，强制速度从 0 开始，每个运算周期与速度增量 FOC_EFREQACC(Motor_Omega_Ramp_ACC 参数)相加，同时根据 FOC_EFREQHOLD(MOTOR_OMEGA_ACC_END 参数)进行最大值限幅，输出强制速度作为最终速度 EOME 供角度计算模块算出估算器角度 ETHETA；当估算器的估算速度 OMEGA 大于等于 EFREQMIN 时，输出估算速度 OMEGA 作为最终速度 EOME。



启动过程如下图所示:


ω
EFREQHOLD
EFREQMIN
0
t
omega
EOME

5.1.2 启动常见问题&解决方式

常见问题	解决方法
电机动一下后静止，且一直有电流输出	1.启动电流太小电机输出无法切换到下一拍换相，增大 IQ_Start_CURRENT； 2.估算器输出速度太小无法换相到下一拍，若排除 A 后此时依次加大 ATO_BW、ATO_BW_RUN、ATO_BW_RUN1、ATO_BW_RUN2； 3.排除 A、B 后，检查硬件电路运放 AMPO 部分是否有问题导致电流采样不准，估算器估算不正常； 4.也有可能是 omega 加速度的频率太高导致，可以减小 Motor_Omega_Ramp_ACC。
电机转一下后停下且一直抖动	1.此种情况一般为 ATO_BW 值太大，导致估算器输出转速较高，启动时失步，此时依次减小 ATO_BW、ATO_BW_RUN、ATO_BW_RUN1、ATO_BW_RUN2； 2. Omega 启动参数也会有影响。
启动正序旋转一定角度后卡顿一下再正常旋转	1.此时可估计从启动到出现卡顿的时间，再设置对应时间的 ATO_BW 值，例如：启动 1s 后电机卡顿一下然后继续正常运行，1s 时间对应的值为 ATO_BW_RUN1、ATO_BW_RUN2，此时该 ATO_BW 值较小限制了电机加速，相应加大该值即可； 2.omega 加速度太小时也会造成卡顿的情况，可以加大 Motor_Omega_Ramp_ACC。
电机启动反转后正转时持续抖动	1.电机启动反转一下后正转所需要的时间较长，此时 ATO_BW 已加到比较大的值，因此减小相应时间的 ATO_BW 值即可； 2.也有可能是 omega 加速度的频率太高导致，可以修改 Motor_Omega_Ramp_ACC。

5.2 保护介绍

每个项目，不同电机，不同板子的保护值都会有所不同，各种保护的保护值都要根据实际项目去匹配。当发现保护，特别是堵转保护或缺陷保护触发不了，或者正常运行时，误触发保护时，说明是保护设定值不合理导致的，此时要调整保护的设定值。

5.2.1 过流保护

1. 硬件过流保护；

芯片通过比较器 3 做硬件过流保护，检测方法：母线电流流经采样电阻，在采样电阻上形成一个电压,这个电压经过运算放大器放大送入比较器的正向输入端。比较器的负向输入端会被设置一个参考电压,这个参考电压可选择 DAC 产生或者由外部分压得到(目前都是用的 DAC 产生)。当母线电流增大到一定数值之后，就会导致比较器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 发生中断并自动关闭 MOE(可选择自动或者不自动关闭 MOE，目前默认都是自动关闭 MOE)，从而完成过流保护。硬件过流保护只需要修改保护值 OverHardcurrentValue 的大小即可。

```
Protect.h
25
26  /* -----硬件过流保护方式选择----- */
27  #define HardwareCurrent_Protect        (Hardware_CMP_Protect)                  // 硬件过流保护实现方式
28
29  /* -----硬件过流保护比较值来源----- */
30  #define Compare_Mode                   (Compare_DAC)                           // 硬件过流值的来源
31  #define OverHardcurrentValue           (40.0)                                  // (A) DAC模式下的硬件过流值
```

2. 软件过流保护。

程序通过获取三相最大电流值，当最大电流值超过设定的软件过流保护值 OverSoftCurrentValue 时，则计一次；在 OverSoftCurrentClrTime 时间内，计数超过 OverSoftCurrentTime 时，则触发保护。

```
Protect.h
31  #define OverHardcurrentValue           (40.0)                                  // (A) DAC模式下的硬件过流值 Imax = VHALF / HW_R
32
33  /* -----软件过流保护----- */
34  #define OverSoftCurrentValue           I_Value(30.0)                           // (A) 软件过流值
35  #define OverSoftCurrentTime            (10)                                    // 软件过流检测次数，初步设定值10
36  #define OverSoftCurrentClrTime         (1000)                                  //(ms)每隔 OverSoftCurrentClrTime 检测次数清零，
```

5.2.2 电压保护

程序通过AD2口检测电压，当检测到的电压超过设定值时，则报过压保护；此时当电压重新低于过压恢复值时，清除过压保护故障。当电压低于设定的欠压值时，则报欠压保护。此时当电压重新高于欠压恢复值时，清除欠压保护故障。

```
Protect.h
40  #define GetCurrentOffsetValue          _Q14(0.05)                              // (单位:%)偏置电压保护误差范围，
41
42  /* -----过欠压保护----- */
43  #define Over_Protect_Voltage           (30.5)                                  // (V) 直流电压过压保护值
44  #define Over_Recover_Vlotage           (29.5)                                  // (V) 直流电压过压保护恢复值
45  #define Under_Protect_Voltage          (12.5)                                  // (V) 直流电压欠压保护值
46  #define Under_Recover_Vlotage          (13.5)                                  // (V) 直流电压欠压保护恢复值
```

5.2.3 缺相保护

电机发生缺相时，三相电流是不对称的。因此可以通过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来实现缺相保护。

具体程序实现方法：若检测到其中一相的最大电流大于另一相最大电流的PhaseLossTimes倍，且该相最大电流大于设定的PhaseLossCurrentValue值，则判定为缺相。

```
Protect.h
61
62  /* -----缺相保护----- */
63  #define PhaseLossCurrentValue          I_Value(1.8)                            // (A)   缺相电流值
64  #define PhaseLossTimes                 (3)                                     // 缺相时电流倍数
```

注意事项：

有些方案在缺相时，由于缺的那一相会有毛刺的存在，可能会导致采集的最大电流值跟另外两相差不多，这时候通过上述方法可能检测不出来。解决方法: 可以通过积分的方式，在一定时间内通过去比较电流累计值的大小去判断缺相。

5.2.4 堵转保护

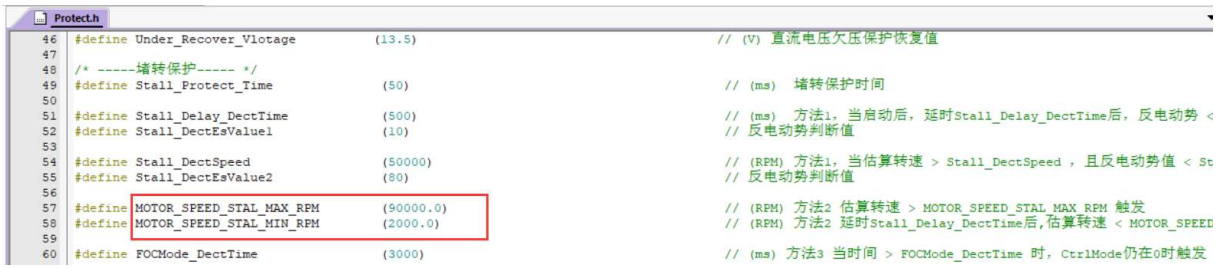
堵转保护有三种方法检测:

1. 通过检测估算器计算出来的FOC_ESQU(估算器计算的反电动势的平方)判断，正常情况下，电机转速越高，FOC_ESQU会越大。在电机发生堵转时，电机失步的情况下，估算转速会很高，但是FOC_ESQU会很小，因此可以通过改方式判断；

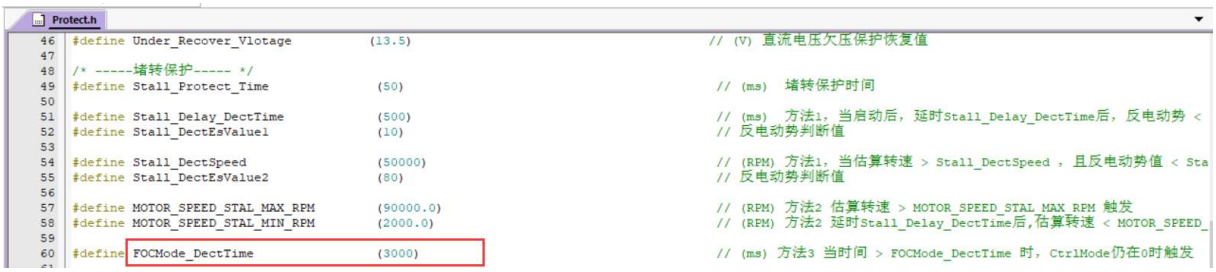
具体程序实现方法: 当开机延时Stall_Delay_DectTime后，判断FOC_ESQU的值还是小于设定值Stall_DectEsValue1；或者当估算转速高于设定值Stall_DectSpeed，但是FOC_ESQU的值小于设定值Stall_DectEsValue2时，则判定为堵转。



2. 通过检测估算转速，当估算转速超过设定转速MOTOR_SPEED_STAL_MAX_RPM，或者低于设定转速MOTOR_SPEED_STAL_MIN_RPM，则认为发生了堵转；

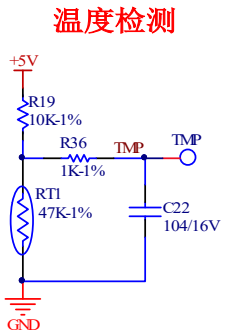


3. 电机启动的时候，程序会在判断估算转速大于MOTOR_LOOP_RPM后，将Mode状态从0置为1，从固定电流启动，进入正常的环路。此时可以通过该Mode去判断是否发生了堵转。若在开机经过FOCMode_DectTime时间后, Mode仍然处于0的状态，则认为电机启动失败，即发生了堵转。



5.2.5 过温保护

过温保护常用的电路图如下所示，分压电阻通常会用一个NTC电阻，该电阻随着温度的上升，阻值逐渐下降。在每个温度都会有对应一个阻值。TD连接到芯片的一个AD端口。程序通过检测该AD口的电压，当该电压小于设定温度下的电压时，则表明NTC电阻温度超过了设定值，触发保护。




```
Protect.h
64  #define PhaseLossTimes              (3)                          // 缺相时电流倍数
65
66  /* -----NTC过温保护----- */
67  #define TemperatureProtectTime      (100)                        // (ms)温度保护检测时间
68  #define OVER_Temperature            Tempera_Value(1.0)           // 过温保护阈值，根据NTC曲线设定，10K上拉电阻，80℃
69  #define UNDER_Temperature           Tempera_Value(2.23)          // 过温保护恢复阈值，根据NTC曲线设定，10K上拉电阻，70℃
```

其中，

OVER_Temperature为保护设定值，1.0为NTC电阻在80℃下的阻值1Ω，UNDER_Temperature为恢复值，2.23为NTC电阻在70℃下的阻值2.23Ω。

注意：

如果上拉电阻不是10K的话，上拉电压不是5V时，此时要去修改定义公式。

```
Protect.h  Develop.h*
89  #define   UDQ_K                     ((float)(UDQMAX-UDQMIN)/(float)(UDQMAX_Volt_VALUE-UDQMIN_Volt_VALUE))
90
91  /* -----过温保护值设置----- */
92  #define   Tempera_Value(NTC_Value)      _Q15((5.0*NTC_Value/(10.0+NTC_Value))/HW_ADC_REF)
93
```

其中，

5.0为分压前的电压，电路图中为5V，该值要根据实际电路设置；10.0为上拉电阻，该值要根据实际电路设置。

5.2.6 偏置电压保护

电机开始之前，会先采集偏置电压，有接 VHALF 时，偏置电压采集值理论上为 2048，左移 3 位后为 16383 左右；没接 VHALF 时，理论值为 0；当采集的值±超过理论值的百分比 GetCurrentOffsetValue 时，这认为偏置电压异常。其中，0.05 代表 5%。

```
Protect.h*
37
38  /* -----偏置电压保护----- */
39  #define GetCurrentOffsetValue          _Q14(0.05)                    // (单位:100%)偏置电压保护误差范围，超过该范围保护
40
```

5.2.7 其他保护

根据客户需求自行添加其他保护。

6 其他常见功能调试

6.1 限功率功能

使用恒转速控制时，需要对母线电流进行限制，

```
#define LIMITIBUSCURRENT                I_Value(20.0)                    // (A) 母线限制电流值
```

目前电流限制通过采集到 AMP0O 上经过 RC 滤波的母线电流值对输出电压进行限制。

7 方案调试难点&解决方法

恒功率调试	
常见问题	**解决方法**
启动一直有异常	启动一直调试不好，软件问题排除后，可以查看硬件采样布线等是否有问题。
顺风启动有异常，一直检测不准	查看硬件反电动势检测电路部分的地线是否布置合理，一般检测不准大都是地线等不合理有干扰导致。
电机速度响应较慢	1.调试外环的 SKP,SKI； 2.调节时间 SPEED_LOOP_TIME； 3.如果只是加减速比较慢，就调节加减速的增减量。
转速或者功率达不到客户要求	1.电流波形正弦的情况下，通过观测 FOC__UQ 是否饱和 2.FOC_UQ 饱和，且 FOC__UD 值比较大的话，通过调整补偿角 FOC_THECOMP(正负都调整看看)确认是否能达到客户需求； 3.以上还是达不到要求时，可考虑开启过调制。一般情况下，不建议开过调制，如果实在达不到最大功率，确认是电机问题时，可让客户直接修改电机。
电机运行到高转速后容易出现大电流的情况	1.调整补偿角 FOC_THECOMP； 2.挪一下采样点，即修改采样点延时时间 FOC_TRGDLY。
电流波形存在正弦度失真	1.看采样偏置基准是否正常； 2.修改电流环的 PI，即 DQKP,DQKI； 3.修改采样点延时时间 FOC_TRGDLY； 4.修改载波频率(注意修改后会影响启动跟运行)。
注意事项：修改参数后，基本都会影响启动和运行性能，解决好问题后要重新测试确认。	

8 修改记录

版本号：第 1 位-原理 第 2 位-模块 第 3 和 4 位-细节

版本号	修改详细内容说明	生效日期	修订者	审核者
V1.0.00	初稿	2022-07-24	李季	

9 版权说明

版权所有©峰岹科技（深圳）股份有限公司（以下简称：峰岹科技）。

为改进设计和/或性能，峰岹科技保留对本文档所描述或包含的产品（包括电路、标准元件和/或软件）进行更改的权利。本文档中包含的信息供峰岹科技的客户进行一般性使用。峰岹科技的客户应确保采取适当行动，以使其对峰岹科技产品的使用不侵犯任何专利。峰岹科技尊重第三方的有效专利权，不侵犯或协助他人侵犯该等权利。
本文档版权归峰岹科技所有，未经峰岹科技明确书面许可，任何单位及个人不得以任何形式或方式（如电子、机械、磁性、光学、化学、手工操作或其他任何方式），对本文档任何内容进行复制、传播、抄录、存储于检索系统或翻译为任何语种，亦不得更改或删除本内容副本中的任何版权或其他声明信息。

峰岹科技（深圳）股份有限公司
深圳市南山区科技中二路深圳软件园二期 11 栋 2 楼 203
邮编：518057
电话：0755-26867710
传真：0755-26867715
网址：www.fortiortech.com

峰岹科技（深圳）股份有限公司

ISO9001 质量管控文件

FU6865

电子扇调试手册

文件版本管制

版本	修订者	主要修改内容	生效日期	审核/批准
V1.0.00	李季	初稿	2022/07/24	

会签单位	签核人员	签核时间
应用部		
品质部		