无感吸尘器应用手册

三相电机控制 MCU
FU6572L

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 硬件实物及原理图 5
2.1.1 电源电路 7
2.1.2 芯片电路 7
2.1.3 反电动势检测电路 8
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 软件原理 **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 确认芯片内部相关数据配置 15
4.3 确认硬件参数 15
4.4 保护参数设置 16
4.5 启动参数配置 17
4.6 硬件驱动电路检测 19
4.7 调试电流环 20
4.8 增加功率环 21
4.9 增加 PWM 等功能 23

4.10 可靠性测试 24
4.10.1 功能可靠性 24
4.10.2 保护可靠性 24
4.10.3 启动稳定性 24
5 功能介绍 25
5.1 启动调试 25
5.1.1 Omega 启动 25
5.1.2 启动常见问题&解决方式 27
5.2 保护介绍 27
5.2.1 过流保护 27
5.2.2 电压保护 28
5.2.3 缺相保护 28
5.2.4 堵转保护 29
5.2.5 过温保护 30
5.2.6 超速保护 30
5.2.7 偏置电压保护 31
5.2.8 其他保护 31
6 其他常见功能调试 31
6.1 限速功能 31
7 方案调试难点&解决方法 32
8 修改记录 33
9 版权说明 34

1 概述

本应用手册详细介绍了如何使用峰岹科技的 FU6572L 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/硬件模块	名称	章节	备注
软件	FU-AM-FU6572-B-025-SW-V1.0.00-20221228	全部	调试需在该工程软件上进行
硬件	FU6572L-XCQ-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 硬件实物及原理图

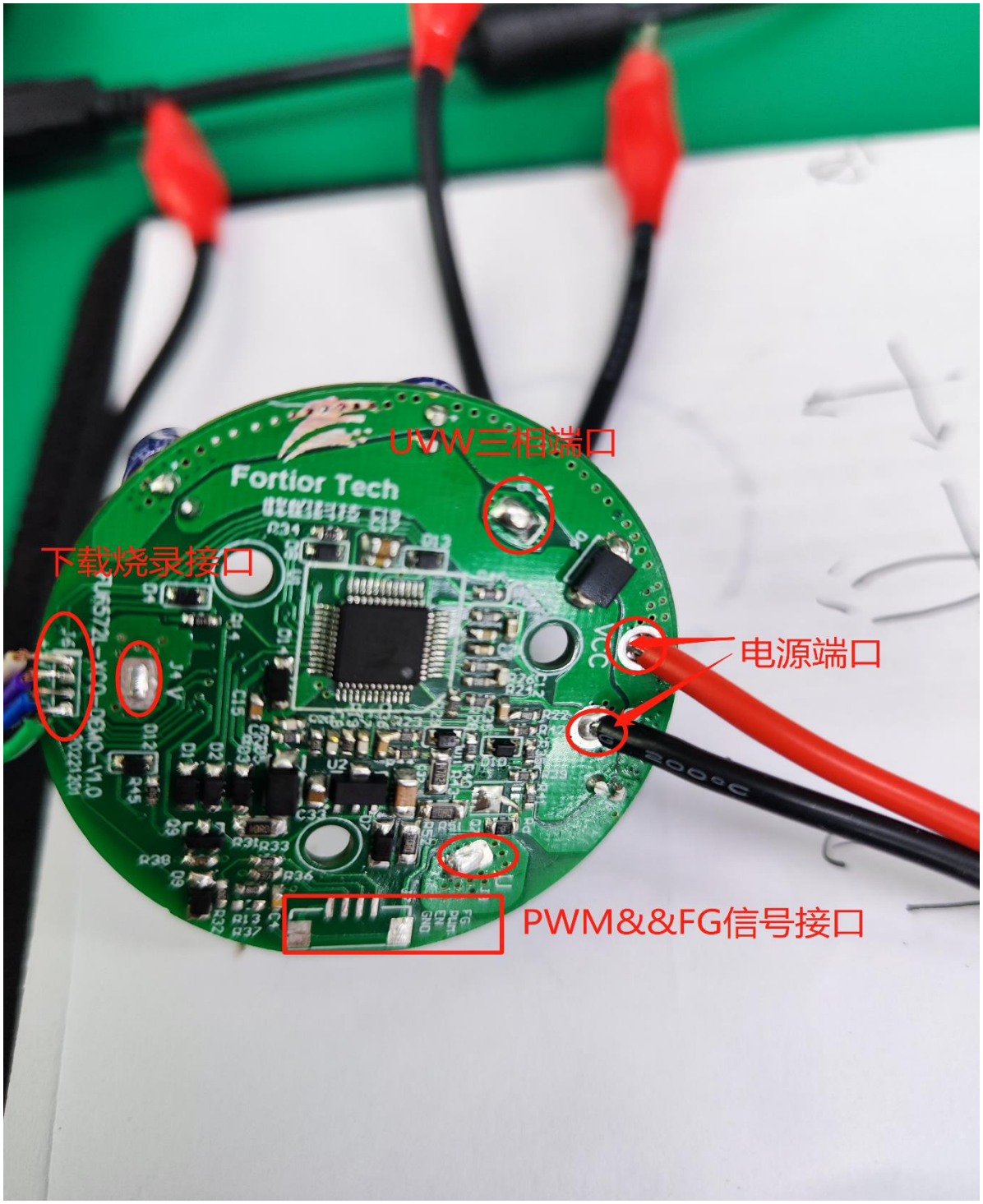


图 2-1 硬件实物图

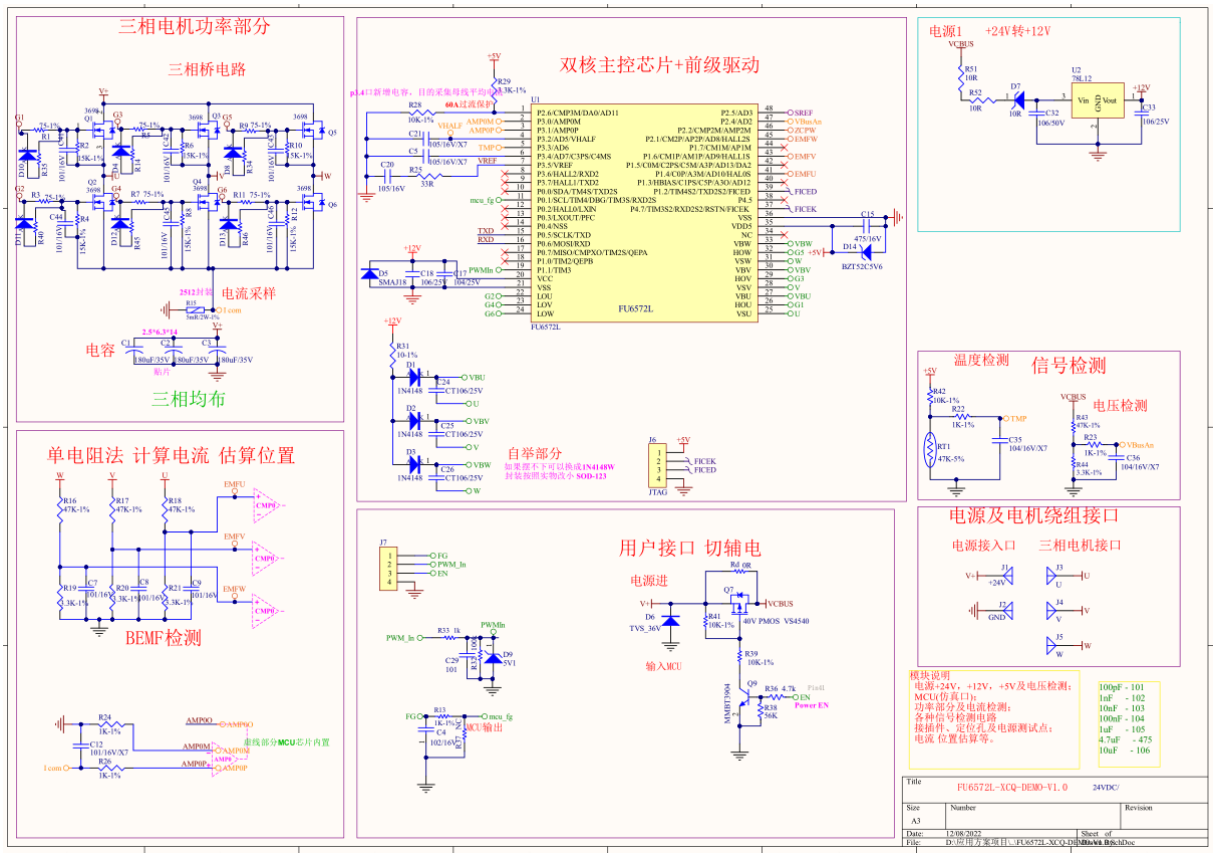


图 2-2 硬件原理图

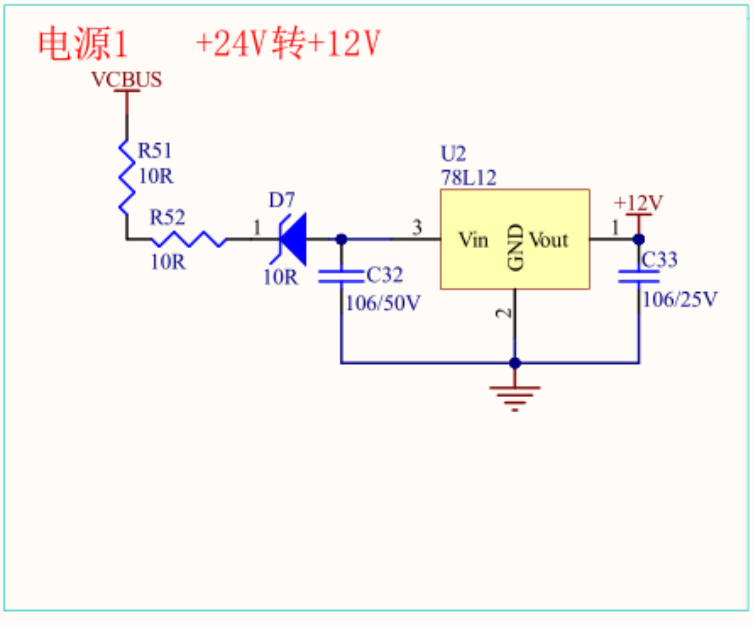
使用方式:

该板子为吸尘器应用方案的专用 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。

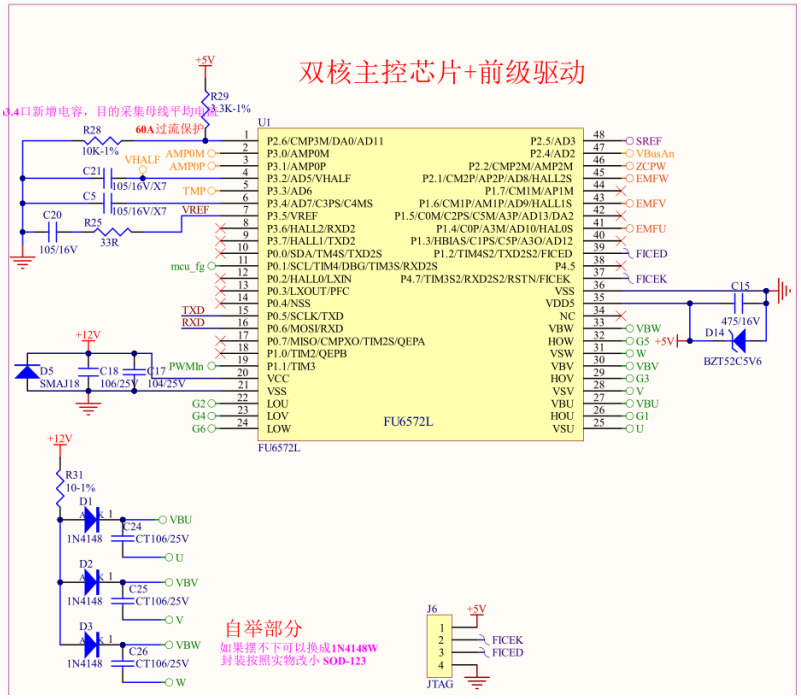
2.1.1 电源电路



使用方式:

直流电源正极接 VCBUS 端，负极接 GND 端。

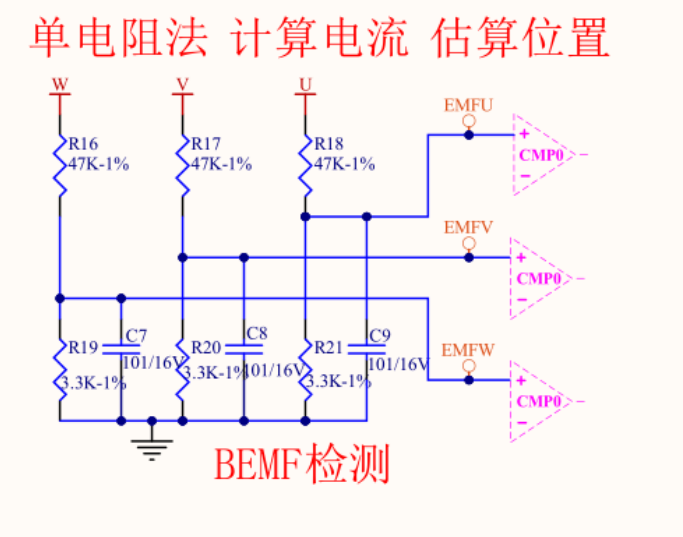
2.1.2 芯片电路



使用方式:

FU6572L 应用于中低压 6-N MOSFET 驱动应用。其中 J6 为烧录线接口。

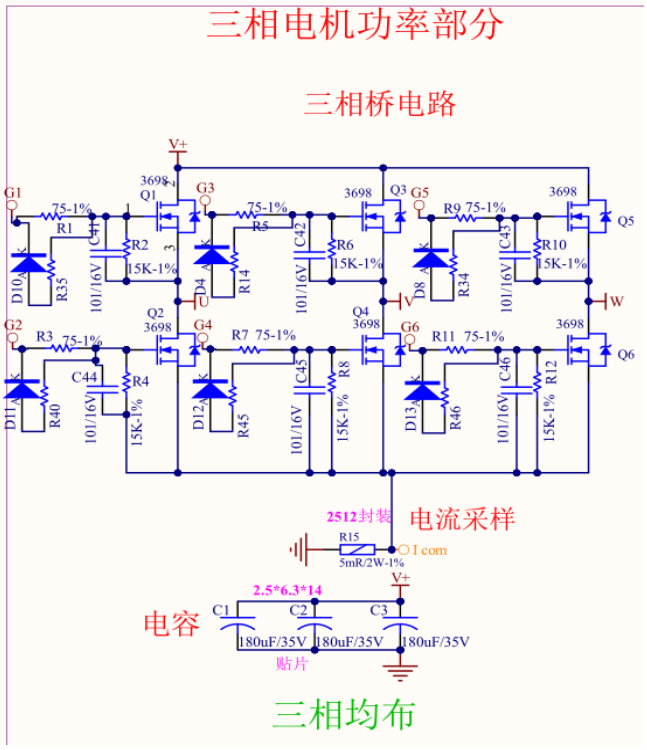
2.1.3 反电动势检测电路



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。

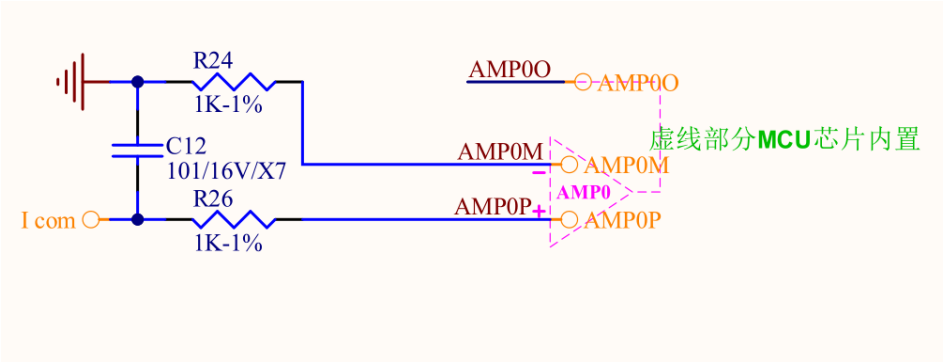
2.1.4 功率驱动电路



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%。

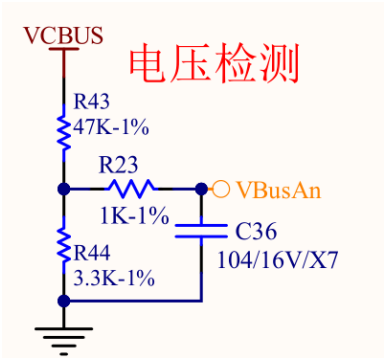
2.1.5 运放配置电路



注意事项:

1. C12 参数不可调整，精度要求 10%；
2. R26、R24 需要用 1%精度电阻；
3. 此处 AD7 用于母线平均电压采样；
4. 放大倍数选择内部放大；
5. 最大采样电流 = (VREF - VHALF)/放大倍数/采样电阻值；
6. 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. R23、C36 参数不可调整；
2. R43、R44 需要用 1%精度电阻；
3. 最大采样电压 = (R43 + R44)/(R44)*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 软件原理

3.1 电机状态机流程图

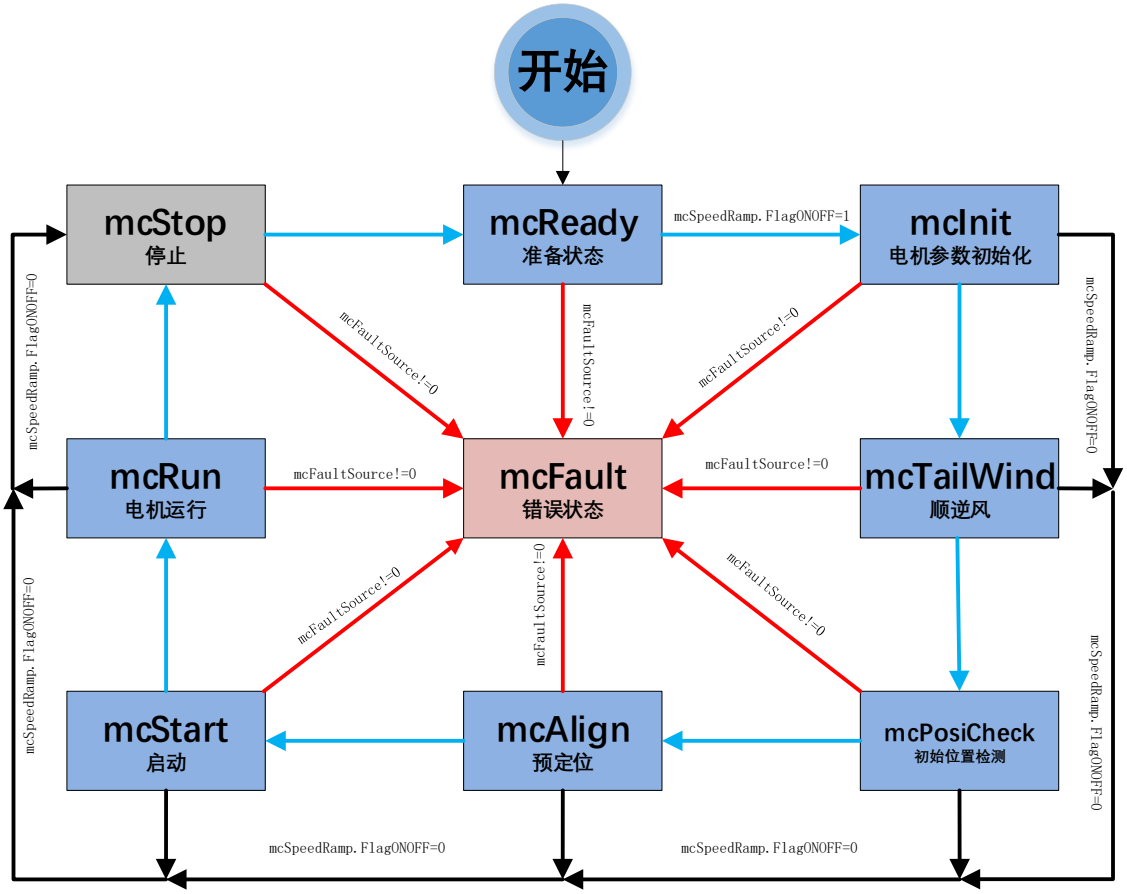


图 3-1 电机状态机流程图

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、mcTailWind、mcPosiCheck、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

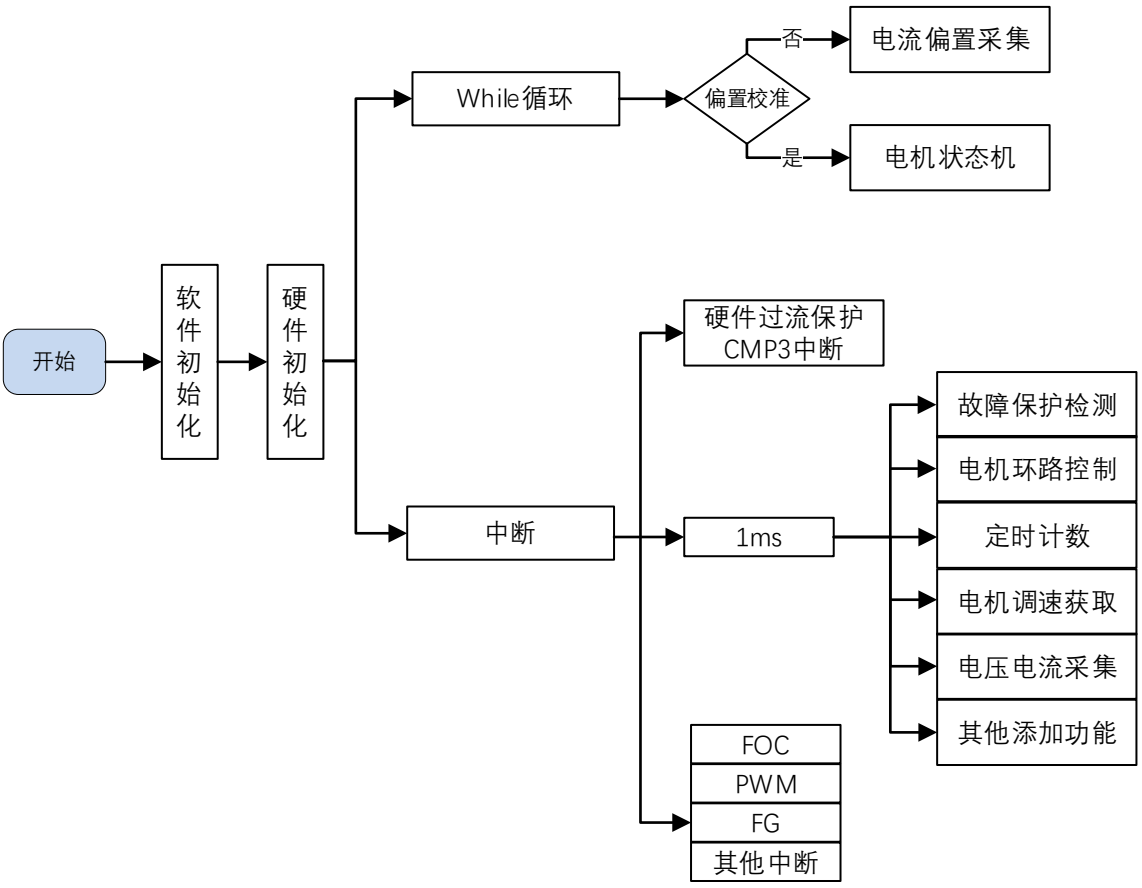


图 3-2 程序执行流程图

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，以及反电动势常数 Ke；
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:

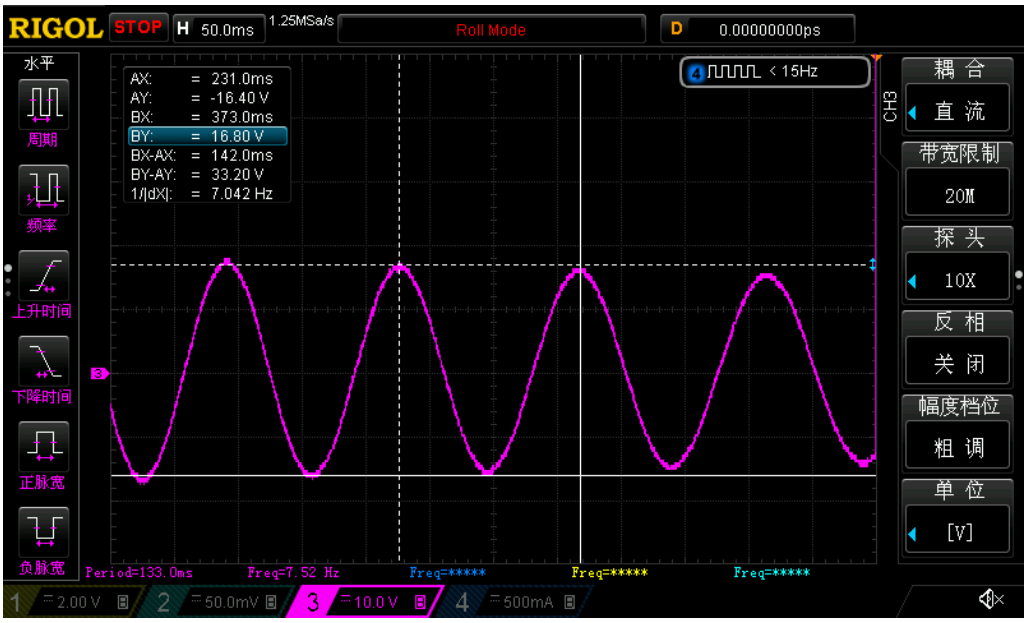


图 4-1 反电动势波形

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

```
Customer.h
/*
                                    2.电机参数
*/
#define R                           (0.6)                   ///<      电机 误差系数
#define Pole_Pairs                  (1.0)                   ///<      电机 极对数
#define RS                          (0.0103*R)              ///< (Ω)  电机 相电阻
#define LD                          (0.00001753*R)          ///< (H)  电机 相电感
#define LQ                          (LD)                    ///< (H)  电机 相电感
#define MOTOR_SPEED_BASE            (160000.0)              ///< (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 确认芯片内部相关数据配置

```
Customer.h
#define PWM_FREQUENCY               (30.0)                  ///< (kHz) 载波频率
#define PWM_DEADTIME                (0.8)                   ///< (us) 死区时间
#define MIN_WIND_TIME               (PWM_DEADTIME + 1.2)    ///< (us) 单电阻最小采样窗口，建
#define DLL_TIME                    (PWM_DEADTIME + 0.3)    ///< (us) 双电阻最小脉宽设置，建
/**
 * 硬件PCBA参数设置根据驱动芯片的类型选择，6865为 HIGH_LEVEL
 * @param (HIGH_LEVEL)   驱动高电平有效
 * @param (LOW_LEVEL)    驱动低电平有效
 * @param (UP_H_DOWN_L)  上桥高电平有效，下桥低电平有效
 * @param (UP_L_DOWN_H)  上桥低电平有效，下桥高电平有效
 */
#define PWM_LEVEL_MODE              (HIGH_LEVEL)    ///< 驱动电平设置
/*
                                    9. 正反转模式
*/
#define FR_MODE                     (CW)            ///< 转向设置
```

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。吸尘器一般转速较高，可先用默认 30K 的调试；
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 确认硬件参数

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，其中，HW_RSHUNT 为采样电阻，HW_AMPGAIN 为放大倍数。

- 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

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)。

```
101  * @param (VHALF1_8)          VHALF电压设置为1/8VREF
102  * @param (VHALF1_4)          VHALF电压设置为1/4VREF
103  * @param (VHALF25_64)        VHALF电压设置为25/64VREF
104  * @param (VHALF1_2)          VHALF电压设置为1/2VREF
105  */
106 #define HW_VHALF_SEL                    (VHALF1_2)      ///< 偏置电压设置
107
108 /**
109  * @brief  运放0偏置电压配置（根据实际电路匹配）
110  * @param (Enable)          运放0有接VHALF
111  * @param (Disable)         运放0没接VHALF
112  */
113 #define VHALF_EN                        (Enable)        ///< VHALF输出使能
114
115 /* ------------------------------------------------------------------------------------------
116                                         4.母线电压采样参数
117 ------------------------------------------------------------------------------------------*/
118
119 #define RV1                             (47.0)          ///< (kΩ) 母线电压分压电阻1
120 #define RV2                             (3.3)           ///< (kΩ) 母线电压分压电阻2
121
```

其中，

1) 母线分压比 = (RV1 + RV2)/RV2；

4.4 保护参数设置

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)。

```
Protect.h
16
17  /*保护参数设置-----------------------------------------------------------------------------*/
18
19  /* Faults processing Enable */
20  #define OC_SW_ProtectEn                 (1)                 // 软件过流保护使能:  0,不使
21  #define OV_ProtectEn                    (0)                 // 过欠压保护使能:    0,不使
22  #define LP_ProtectEn                    (1)                 // 缺相保护使能:      0,不使
23  #define OT_ProtectEn                    (0)                 // 过温保护使能:      0,不使
24  #define Stall_ProtectEn                 (0)                 // 堵转失速保护使能:  0,不使
25  #define GetCurrentOffsetEnable          (1)                  // 偏置电压保护, 0,不使能: 1
26  #define OverSpeedProtectEnable          (1)                 ///< 超速保护使能
27
28  /*  保护重启参数设置  */
29  #define OC_RecoveryTimes                (5)                // 重启次数,设定值<255 达到重启
30  #define OC_RecoveryDelayTime            (1000)             // (ms)重启延迟时间
31
32  #define OV_RecoveryTimes                (5)                 // 重启次数,设定值<255 达到重启
33  #define OV_RecoveryDelayTime            (1000)              // (ms)重启延迟时间
34
35  #define LP_RecoveryTimes                (5)                 // 重启次数,设定值<255 达到重启
36  #define LP_RecoveryDelayTime            (1000)              // (ms)重启延迟时间
```

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
181 /* 启动电流 */
182 #define ID_Start_CURRENT              I_Value(0.0)      ///< (A) D轴启动电流
183 #define IQ_Start_CURRENT              I_Value(12.0)     ///< (A) Q轴启动电流
184
185 /* 顺风启动切入闭环电流  */
186 #define ID_RUN_CURRENT                I_Value(0.0)      ///< (A) D轴运行电流
187 #define IQ_RUN_CURRENT                I_Value(15.0)      ///< (A) Q轴运行电流
188
```

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

```
188
189 /* 启动ATO参数 */
190 #define ATO_BW_START                (0.0)           ///< 观测器带宽的滤波值，经典值为1.0-200.0
191 #define ATO_BW_RUN1                 (160.0)
192 #define ATO_BW_RUN2                 (180.0)
193 #define ATO_BW_RUN3                 (220.0)
194 #define ATO_BW_RUN4                 (260.0)
195
```

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

```
206
207 #define MOTOR_SPEED_SMOMIN_RPM          (800.0)          ///< (RPM) SMO运行最小转速影响启动
208
209 #define SPD_BW                          (30.0)          ///< 速度带宽的滤波值，经典值为5.0-40.0
210 #define MOTOR_LOOP_RPM                  S_Value(800.0)  ///< (RPM) 启动后电流环切入外部环路转速阈值
211
```

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

```
200
201
202 /* OMEGA启动参数 */
203 #define MOTOR_OMEGA_RAMP_ACC            (60)             ///< omega启动的增量（每个载波递增值）
204 #define MOTOR_OMEGA_RAMP_MIN            S_Value(50.0)    ///< (RPM) omega启动的最小切换转速
205 #define MOTOR_OMEGA_RAMP_END            S_Value(300.0)   ///< (RPM) omega启动的限制转速
206
207 #define MOTOR_SPEED_SMOMIN_RPM          (800.0)          ///< (RPM) SMO运行最小转速影响启动
```

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；

Customer.h

```
225
226 /* -----启动时候的电流环KPKI设置值----- */
227 #define DKPStart                        _Q12(1.3)                   // 启动时DQ轴KP
228 #define DKIStart                        _Q15(0.02)                  // 启动时DQ轴KI
229 #define QKPStart                        DKPStart
230 #define QKIStart                        DKIStart
231
232 /* ----------电流环参数设置值-------------------------------------------------------------
233 #define DKP                             _Q12(2.6)       ///< 运行DQ轴电流环KP
234 #define DKI                             _Q15(0.4)       ///< 运行DQ轴电流环KI
235 #define QKP                             DKP             ///< 运行DQ轴电流环KP
236 #define QKI                             DKI             ///< 运行DQ轴电流环KI
```

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。

Customer.h

```
/* D轴参数设置 */
#define DOUTMAX                 _Q15(0.99)        ///< D轴电压最大限幅值，单位：输出占空比
#define DOUTMIN                 _Q15(-0.99)       ///< D轴电压最小限幅值，单位：输出占空比
/* Q轴参数设置 */
#define QOUTMAX                 _Q15(0.99)        ///< Q轴电压最大限幅值，单位：输出占空比
#define QOUTMIN                 _Q15(-0.99)       ///< Q轴电压最小限幅值，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

```
/* ------------------------------------------------------------------------------------------
                                        2.预定位参数
------------------------------------------------------------------------------------------*/
/**
 * 预定位模式选择
 * @param (ALIGN_DSIABLE)       禁止
 * @param (ALIGN_NOMAL)         正常预定位
 * @param (ALIGN_TEST)          测试模式，可用于手动测试电机极对数
 */
#define ALIGN_MOME                      (ALIGN_DSIABLE)   ///< 预定位模式选择

#define Align_Angle                     (-30.0)           ///< (°) 预定位角度

#define Align_Time                      (1)               ///< (ms) 预定位时间

/* 预定位的Kp、Ki */
#define DQKP_Alignment                  _Q12(1.0)         ///< 预定位的KP
#define DQKI_Alignment                  _Q15(0.01)        ///< 预定位的KI
#define ID_Align_CURRENT                I_Value(0.0)      ///< (A) D轴定位电流
#define IQ_Align_CURRENT                I_Value(2.8)      ///< (A) Q轴定位电流
```

Customer.h

```
/* ------------------------------------------------------------------------------------------
                                        5. 调速开关模式
------------------------------------------------------------------------------------------*/
/**
 * 闭环方式选择
 * @param (PWMMODE)         PWM调速
 * @param (SREFMODE)        模拟调速
 * @param (NONEMODE)        直接给定值，不调速
 * @param (ONOFFTEST)       启停测试工具
 * @param (KEYMODE)         按键调速

 */
#define SPEED_MODE                      (NONEMODE)          ///< 闭环方式选择
```

将AlignTestMode置为ALIGN_NOMAL，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 调试电流环

1. 将环路选择为电流环；

```
Customer.h
246 //-----------------------------------------------------------------------------
247 /**
248  * 闭环方式选择
249  * @param (CURRENT_LOOP_CONTROL)        电流环
250  * @param (SPEED_LOOP_CONTROL)          速度环
251  * @param (POWER_LOOP_CONTROL)          功率环
252  */
253 #define MOTOR_CTRL_MODE                 (CURRENT_LOOP_CONTROL) ///< 闭环方式选择
254
```

2. 调速方式先选为直接给定值，调整给定值 Motor_Min_Current 的大小，以此来控制电流环的电流大小(注意给的是相电流的值，而且因为选的调速方式是直接给定的，程序只认 Motor_Min_Curren，此时 Motor_Max_Current 是没做用的)；

```
Customer.h
270                                     5. 调速开关模式
271 //-----------------------------------------------------------------------------
272 /**
273  * 闭环方式选择
274  * @param (PWMMODE)         PWM调速
275  * @param (SREFMODE)        模拟调速
276  * @param (NONEMODE)        直接给定值，不调速
277  * @param (ONOFFTEST)       启停测试工具
278  * @param (KEYMODE)         按键调速
279
280  */
281 #define SPEED_MODE                      (NONEMODE)          ///< 闭环方式选择
282
```

```
Customer.h
222 /* -----电流控制模式下速度曲线的最大最小值----- */
223 #define Motor_Max_Current               I_Value(20.0)               ///< (A) 运行最大电流
224 #define Motor_Min_Current               I_Value(5.0)                ///< (A) 运行最小电流
225
```

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，达到客户目标功率；
5. 确认电流环情况下最大功率、转速；
6. 记录最大功率下的计算功率值 mcFocCtrl.Powerlpf(该值即是功率环时的功率给定最大参考值)，以及此时设置的相电流 Motor_Min_Curren 大小(该值可作为外环 SOUTMAX 的参考值)。

   注: 母线电流采集的 AD 口要对应上，内部硬件固定 ADC7。具体位置如下图。

```
Interrupt.c
79        SetBit(ADC_CR, ADCBSY);
80
81        /* -----母线电流采样----- */
82        Power_Currt = (ADC7_DR);
83
84         Power_Currt    = Abs_F16(Power_Currt - mcCurOffset.IbusOffset);
85         mcFocCtrl.mcADCCurrentbus  = LPF_3_Function(Power_Currt << 2, mcFocCtrl.mcADCCurrentbus, 20); /*
86
87        if (mcState != mcRun)
88        {
89            mcFocCtrl.mcDcbusFlt    = LPF_3_Function(ADC2_DR, mcFocCtrl.mcDcbusFlt, 50);            /*
90        }
91        else
92        {
93            mcFocCtrl.mcDcbusFlt    = FOC__UDCFLT;
94        }
```

常见问题及解决办法:

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. 记录的 mcFocCtrl.Powerlpf 过大(如可能会超过 32767)或过小(如最大时才几百)；

   解决: 因为该值是通过电流值跟电压值乘积移位后得到的，可以通过修改移位的值来修改该值的大小，让其处于合理范围内(一般最大功率对应的值为一万多两万比较好)。

3. 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。

4. 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 增加功率环

1. 一般吸尘器都是用的功率环，因此将环路选择为功率环；

```
Customer.h
247 /**
248  * 闭环方式选择
249  * @param (CURRENT_LOOP_CONTROL)        电流环
250  * @param (SPEED_LOOP_CONTROL)          速度环
251  * @param (POWER_LOOP_CONTROL)          功率环
252  */
253 #define MOTOR_CTRL_MODE                 (POWER_LOOP_CONTROL) ///< 闭环方式选择
254
```

2. 设置功率曲线的最大值，跟 SOUTMAX 的值，这两个值 4.7 最后一步已经有记录的参考值，其中 SOUTMAX 可能要在得到的基础上再增加一点，以防电压下降时电流还要进一步上升，要有足够的上升空间。Motor_Min_Power 为曲线的最小功率值，可根据客户实际需要设定；

```
Customer.h
217 /* -----功率控制模式下速度曲线的最大最小值----- */
218 #define Motor_Max_Power                 (2800) // 10560==>030  2800==>65W  ///< 运行最大功率
219 #define Motor_Min_Power                 (500)                               ///< 运行最小功率
220
```

```
Customer.h
260
261 #define SOUTMAX                          I_Value(45.0)          ///< (A) 外环电流最大限幅值
262 #define SOUTMIN                          I_Value(0.01)          ///< (A) 外环电流最小限幅值
263
```

3. 通过调整功率环 PI(SKP 和 SKI)和功率环爬坡增量保证功率环稳定，启动响应快且不过冲。

Customer.h

```
255 #define LOOP_TIME                    (2)                ///< (ms) 外环环调
256
257 #define SKP                         _Q12(0.4)           ///< 外环KP
258 #define SKI                         _Q15(0.002)         ///< 外环KI
259 #define SKD                         _Q12(0.2)           ///< 外环KD
260
```

Customer.h

```
249  * @param (CURRENT_LOOP_CONTROL)      电流环
250  * @param (SPEED_LOOP_CONTROL)        速度环
251  * @param (POWER_LOOP_CONTROL)        功率环
252  */
253 #define MOTOR_CTRL_MODE              (POWER_LOOP_CONTROL) ///< 闭环方式选择
254
255 #define LOOP_TIME                     (2)                ///< (ms) 外环环调节周期,默认为 1ms
256
257 #define SKP                          _Q12(0.4)           ///< 外环KP
258 #define SKI                          _Q15(0.002)         ///< 外环KI
259 #define SKD                          _Q12(0.2)           ///< 外环KD
260
261 #define SOUTMAX                      I_Value(45.0)        ///< (A) 外环电流最大限幅值
262 #define SOUTMIN                      I_Value(0.01)        ///< (A) 外环电流最小限幅值
263
264 #define RAMP_INC                     S_Value(60)       ///< (RPS) 每秒爬坡递增量
265 #define RAMP_DEC                     S_Value(60)       ///< (RPS) 每秒爬坡递减量
```

说明:

一般来说此时计算得到的 mcFocCtrl.Powerlpf 跟真实功率是成正比关系，所以我们需要先测量一组 mcFocCtrl.Powerlpf 和真实功率 Power，计算得到功率系数 K。

例如当电机跑 100W 的时候，mcFocCtrl.Powerlpf 此时为 5000，那么功率系数 K = mcFocCtrl.Powerlpf/真实功率值 = 50；那么如果设置目标值为 500*50，即相当于目标值为 500W。

加功率环后常见问题:

1) 电流环相电流稳定，加了功率环之后相电流抖动。

解决: 一般是加了功率环引起的参数，此时主要调整功率环的 PI，或者功率环反馈值 mcFocCtrl.Powerlpf 的滤波系数。因为 mcFocCtrl.Powerlpf 是通过电流跟电压乘积得到的，因此要调整 mcFocCtrl.Powerlpf 的滤波系数，就是调整电压跟电流采集的滤波系数。

Interrupt.c

```
79        SetBit(ADC_CR, ADCBSY);
80
81        /* -----母线电流采样----- */
82        Power_Currt = (ADC7_DR);
83
84         Power_Currt    = Abs_F16(Power_Currt - mcCurOffset.IbusOffset);
85         mcFocCtrl.mcADCCurrentbus  = LPF_3_Function(Power_Currt << 2, mcFocCtrl.mcADCCurrentbus, 20); /* -----母
86
```

Interrupt.c

```
88         {
89             mcFocCtrl.mcDcbusFlt    = LPF_3_Function(ADC2_DR, mcFocCtrl.mcDcbusFlt, 50);
90         }
```

4.9 增加 PWM 等功能

1. 一般吸尘器都是 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
271 //------------------------------------------------------------------------------
272 /**
273  * 闭环方式选择
274  * @param (PWMMODE)        PWM调速
275  * @param (SREFMODE)       模拟调速
276  * @param (NONEMODE)       直接给定值，不调速
277  * @param (ONOFFTEST)      启停测试工具
278  * @param (KEYMODE)        按键调速
279  *
280  */
281 #define SPEED_MODE                      (PWMMODE)           ///< 闭环方式选择
282
```

   2) 根据客户给的 PWM 曲线，调整开关机、最大最小占空比；

```
Customer.h
332 /**
333  * PWM调速 PWM极性选择
334  * @param (PosiPWMDUTY)       正逻辑
335  * @param (NegaPWMDUTY)       反逻辑
336  */
337 #define PWMDUTY_POLARITY              (PosiPWMDUTY)        ///< PWM调速 PWM极性选择
338
339 /* 开关机Duty设置 */
340 #define OFFPWMDuty                    _Q15(0.04)                ///< 关机PWM占空比，小于该占空比关机
341 #define OFFPWMDutyHigh                _Q15(1.0)                 ///< 关机PWM占空比，大于该占空比关机
342 #define ONPWMDuty                     _Q15(0.065)                ///< 开机PWM占空比，大于该占空比时开机
343 #define MINPWMDuty                    _Q15(0.10)                ///< 速度曲线上最小PWM占空比
344 #define MAXPWMDuty                    _Q15(0.99)                ///< 速度曲线上最大PWM占空比
```

   3) 根据客户给的最小最大功率值，调整最大最小功率；

```
Customer.h
218 #define Motor_Max_Power                 (2800) // 10560==>030   2800==>65W   ///< 运行最大功率
219 #define Motor_Min_Power                 (500)                                ///< 运行最小功率
```

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

   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
332 /**
333  * PWM调速 PWM极性选择
334  * @param (PosiPWMDUTY)       正逻辑
335  * @param (NegaPWMDUTY)       反逻辑
336  */
337 #define PWMDUTY_POLARITY              (PosiPWMDUTY)        ///< PWM调速 PWM极性选择
338
```

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。

2. 有其他功能，如 FG 输出等功能时，则对应添加即可；
3. 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

```
/**
 * 闭环方式选择
 * @param (PWMMODE)       PWM调速
 * @param (SREFMODE)      模拟调速
 * @param (NONEMODE)      直接给定值，不调速
 * @param (ONOFFTEST)     启停测试工具
 * @param (KEYMODE)       按键调速

 */
#define SPEED_MODE                      (ONOFFTEST)          ///< 闭环方式选择
```

```
Customer.h
345
346 /* ----------------------------------------------------------------------------------------------------
347                                         11. 启停测试参数
348 ----------------------------------------------------------------------------------------------------*/
349 /* -----启停测试参数配置----- */
350 #define ONOFFTEST_REF                S_Value(15000)        ///< 测试给定，注意闭环方式不同给定值需要更改
351 #define ONOFFTEST_ON_TIME            (2000)               ///< (ms) 启动运行时间
352 #define ONOFFTEST_OFF_TIME           (4000)               ///< (ms) 停止时间
353
```

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

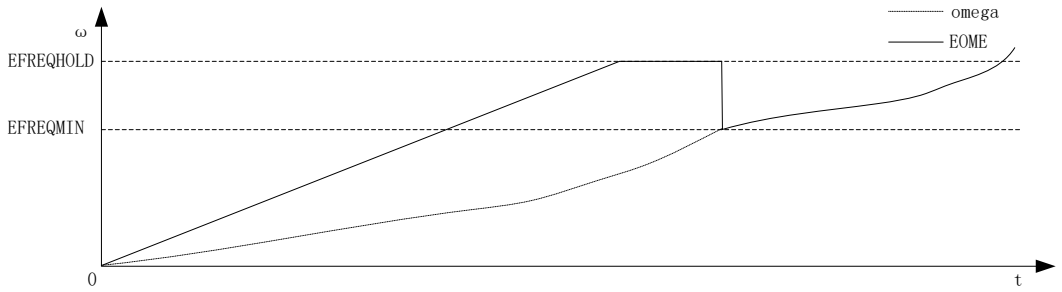
吸尘器选择 Omega 启动，程序对应默认即是该启动方式。

```
Customer.h
174 /**
175  * 开环启动模式选择
176  * @param (Open_Start)          开环强拖启动
177  * @param (Omega_Start)         Omega启动
178  */
179 #define Open_Start_Mode                 (Omega_Start)     ///< 开环启动模式选择
180
```

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_RAMP_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_RAMP_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。

```
Customer.h
201
202 /* OMEGA启动参数 */
203 #define MOTOR_OMEGA_RAMP_ACC         (60)                  ///< omega启动的增量（每个载波递增值）
204 #define MOTOR_OMEGA_RAMP_MIN         S_Value(50.0)        ///< (RPM) omega启动的最小切换转速
205 #define MOTOR_OMEGA_RAMP_END         S_Value(300.0)        ///< (RPM) omega启动的限制转速
206
207 #define MOTOR_SPEED_SMOMIN_RPM       (800.0)               ///< (RPM) SMO运行最小转速影响启动
208
209 #define SPD_BW                       (30.0)               ///< 速度带宽的滤波值，经典值为5.0-40.0
210 #define MOTOR_LOOP_RPM               S_Value(800.0)       ///< (RPM) 启动后电流环切入外部环路转速阈值
211
```

启动过程如下图所示:


omega
EOME
ω
EFREQHOLD
EFREQMIN
0
t

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相保护触发不了，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 HWOCValue 的大小即可。

```
Protect.h
49 #define HW_OC_MODE                    (HW_OC_CMP_MODE)          ///< 硬件过流保护实现方式
50
51 /* 硬件过流保护比较值来源 */
52 #define COMPARE_DAC_MODE              (0)                       ///< DAC设置硬件过流值
53 #define COMPARE_HW_MODE               (1)                       ///< 硬件设置硬件过流值
54 #define COMPARE_MODE                  (COMPARE_DAC_MODE)        ///< 硬件过流值的来源
55 #define HWOCValue                     (45.0)                    ///< (A) DAC模式下的硬件过流值
56
```

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 SW_OC_CurrentVal 时，则计一次；在 SW_OC_DectTime 时间内，计数超过 SW_OC_DectTime 时，则触发保护。

```
Protect.h
56
57 /* 软件过流保护参数设置 */
58 #define SW_OC_CurrentVal              I_Value(40.0)             ///< (A)软件过流值
59 #define SW_OC_DectTime                (30)                      ///< (ms)软件过流检测时间
60
```

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
65 /* 直流母线电压保护参数设置值 */
66
67 #define OVER_VOLTAGE_DECTTIME         (100)                     ///< (ms) 直流母线电压过压检测
68 #define UNDER_VOLTAGE_DECTTIME        (100)                     ///< (ms) 直流母线电压欠压检测
69
70 #define OVER_VOLTAGE_PROTECT          UDC_Value(23.5)           ///< (V) 直流母线电压过压保护
71 #define UNDER_VOLTAGE_PROTECT         UDC_Value(9.0)            ///< (V) 直流母线电压欠压保护
72
73 #define OVER_VOLTAGE_RECOVER          UDC_Value(22.5)           ///< (V) 直流母线电压过压保护
74 #define UNDER_VOLTAGE_RECOVER         UDC_Value(8.0)            ///< (V) 直流母线电压欠压保护
75
```

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
85 /* 缺相保护参数设置值 */
86 #define LP_NoLoadCurrentValue         I_Value(0.1)              ///< (A) 相电流峰值小于该电流值则
87 #define LP_DectDealyTIME              (500)                     ///< (ms) 启动延迟检测时间，用于
88 #define LP_DectCycleTIME              (50)                      ///< (ms) 峰值检测周期时间,需满足
89
```

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。

MotorProtect.c

```
141
142         fault.Stall.EsValue = mcFocCtrl.EMFsquare;
143
144         if (fault.Stall.DectDealyCnt < 500)     /* Delay for a period of time to test */
145         {
146             fault.Stall.DectDealyCnt++;
147         }
148         else
149         {
150             /* ****** 1 ****** */
151             if ((fault.Stall.EsValue < EsThresholdValueL))
152             {
153                 fault.Stall.EsDectCnt++;
154
155                 if (fault.Stall.EsDectCnt >= 3000)
156                 {
157                     fault.Stall.EsDectCnt = 0;
158                     mcFaultSource         = FaultStall;
159                     fault.Stall.Type      = 11;
160                 }
161             }
```

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；

Protect.h

```
75
76  /* 堵转保护参数设置值 */
77  #define STALL_SPEED_MAX                 S_Value(90000)
78  #define STALL_SPEED_MIN                 S_Value(2000)
79
80  #define EsThresholdValueL               (50.0)
81  #define EsThresholdValueH               (100.0)
82  #define EsThresholdSpeed                S_Value(60000)                  ///< (RPM) 电机转速
83
```

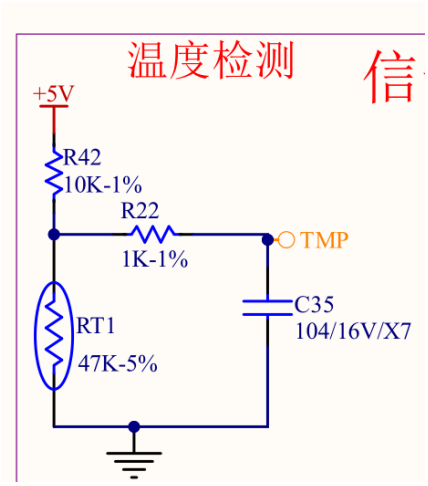
3. 电机启动的时候，程序会在判断估算转速大于MOTOR_LOOP_RPM后，将Mode状态从0置为1，从固定电流启动，进入正常的环路。此时可以通过该Mode去判断是否发生了堵转。若在开机经过FOCMode_DectTime时间后，Mode仍然处于0的状态，则认为电机启动失败，即发生了堵转。

MotorProtect.c

```
201             }
202         }
203
204
205         if (mcFocCtrl.CtrlMode == 0)
206         {
207             fault.Stall.ModeODectCnt++;
208
209             if (fault.Stall.ModeODectCnt >= 3000)
210             {
211                 fault.Stall.ModeODectCnt = 0;
212                 mcFaultSource            = FaultStall;
213
214                 fault.Stall.Type         = 31;
215             }
216         }
217         else
218         {
219             fault.Stall.ModeODectCnt = 0;
220         }
221     }
222 }
```

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
89
90  /* -----NTC过温保护----- */
91  #define TemperatureProtectTime          (1000)                          // (ms)温度保护检测时间
92  #define OVER_Temperature                Tempera_Value(1.0)              // 过温保护阈值，根据NTC曲线设
93  #define UNDER_Temperature               Tempera_Value(2.23)             // 过温保护恢复阈值，根据NTC曲
94
```

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。

```
Parameter.h
71  /* -----过温保护值设置----- */
72  #define    Tempera_Value(NTC_Value)       _Q15((5.0*NTC_Value/(10.0+NTC_Value))/HW_ADC_REF)
73  /*电流基准的电路参数*/
```

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 超速保护

通过检测电机的转速，当电机的转速在连续OVER_SpeedDetectTime时间内，都超过设定转速MOTOR_SPEED_OVER_RPM时，则触发保护。

```
Protect.h
96  /* -----堵入风口超速保护----- */
97  #define  MOTOR_SPEED_OVER_RPM          (110000)                    ///< (RPM) 超速保护速度
98  #define  OVER_SpeedDetectTime          (3000)                      ///< (ms) 超速保护检测时间
99
```

5.2.7 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
60
61  /* -----偏置电压保护----- */
62  #define GetCurrentOffsetValue          _Q14(0.05)                  // (单位:100%)偏置电压保护误
63
64
```

5.2.8 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 限速功能

使用恒功率控制时，当吸尘器入风口堵住时，负载变小电机会运行至较高转速会损坏轴承，而且电机散热较差导致电机损坏，故需要使用限速功能对转速进行限定。

限速功能目前有 3 种方式:

1. 通过对目标值限制实现，当检测速度超过保护阈值之后，在爬坡函数中对目标值限制从而达到限速，此方法容易发生震荡故不做详细说明；
2. 通过切换不同闭环实现，当堵风口超速的时候，检测到转速超过限制值，那么程序会切入速度闭环从而达到限速功能，当取消堵风口由于负载恢复此时转速下功率会超过目标功率值，此时切回功率闭环，从而实现了堵风口限速的功能。此方法需要调节速度环 PI 和功率环 PI 以及 PI 响应周期，而且切环过程容易出现震荡，故不做详细说明；
3. 双 PI 的方式限制转速，硬件 PI 实现功率闭环，硬件 PI 实现转速限制硬件 PI 输出限制 FOC_QMAX,代码如下:

```
#if (OverSpeedLimitEnable)//限速
  {

      FOC_QMAX = HW_One_PI2(Motor_Limit_Speed-mcFocCtrl.SpeedFlt);

  }
  #endif
```

目前程序已经添加了限速功能，可直接使用。

Protect.h

```
99
100 /* -----堵入风口限速功能----- */
101 #define  OverSpeedLimitEnable          (0)                    ///< 限速功能，0，不使能，1使能
102 #define  MOTOR_SPEED_LIMIT_RPM         (60000.0)              ///< (RPM) 限速保护最大速度
103
```

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。
在堵住吸风口后，快速放开，这时候电流会突然变很大导致硬件过流	一般是由于内环电流环没响应过来导致，可以加大电流环的 PI，即 DQKP，跟 DQKI。
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	2022-12-28	刘建华	

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