

# 无感油烟机应用手册

## 三相电机控制 MCU FU6812S2

峰昭科技(深圳)股份有限公司

## 目录

<b>1 概述</b> .....	<b>4</b>
<b>2 硬件原理与参数配置</b> .....	<b>5</b>
2.1 硬件图.....	5
2.1.1 电源部分.....	7
2.1.2 芯片主体.....	8
2.1.3 反电动势检测电路.....	9
2.1.4 功率驱动部分.....	10
2.1.5 串口通信接口.....	11
<b>3 软件原理</b> .....	<b>12</b>
3.1 电机状态机流程图.....	12
3.2 程序流程图.....	14
3.3 程序说明.....	14
3.3.1 Main 函数:.....	14
3.3.2 1ms 定时中断.....	14
3.3.3 FOC 中断.....	15
3.3.4 CMP3 中断.....	15
3.3.5 串口中断.....	15
<b>4 调试步骤</b> .....	<b>16</b>
4.1 配置电机参数.....	16
4.1.1 电机参数.....	16
4.1.2 电机参数测量方法.....	16
4.1.3 对应程序.....	17
4.2 确认芯片内部相关数据配置.....	17
4.3 确认硬件参数.....	17
4.4 保护参数设置.....	18
4.5 启动参数配置.....	19
4.6 硬件驱动电路检测.....	21
4.7 调试电流环.....	21
4.8 增加功率环.....	22
4.9 串口调速功能.....	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
<b>5 功能介绍 .....</b>	<b>25</b>
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.....	26
5.2 保护介绍 .....	26
5.2.1 过流保护 .....	26
5.2.2 电压保护 .....	27
5.2.3 缺相保护 .....	27
5.2.4 堵转保护 .....	28
5.2.5 偏置电压保护.....	28
5.2.6 其他保护 .....	28
<b>6 其他常见功能调试.....</b>	<b>29</b>
6.1 限功率功能 .....	29
<b>7 方案调试难点&amp;解决方法 .....</b>	<b>30</b>
<b>8 修改记录 .....</b>	<b>31</b>

## 1 概述

本应用手册详细介绍了如何使用峰昭科技的 FU6812S2 芯片，在油烟机专用 DEMO 板子上，对直流无刷油烟机电机进行无感 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-FU6812-B-082-SW-V1.0.00-20221214	全部	调试需在该工程软件上进行
硬件	FU-AM-FU6812-B-082-HW-V1.0.00-20210601	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

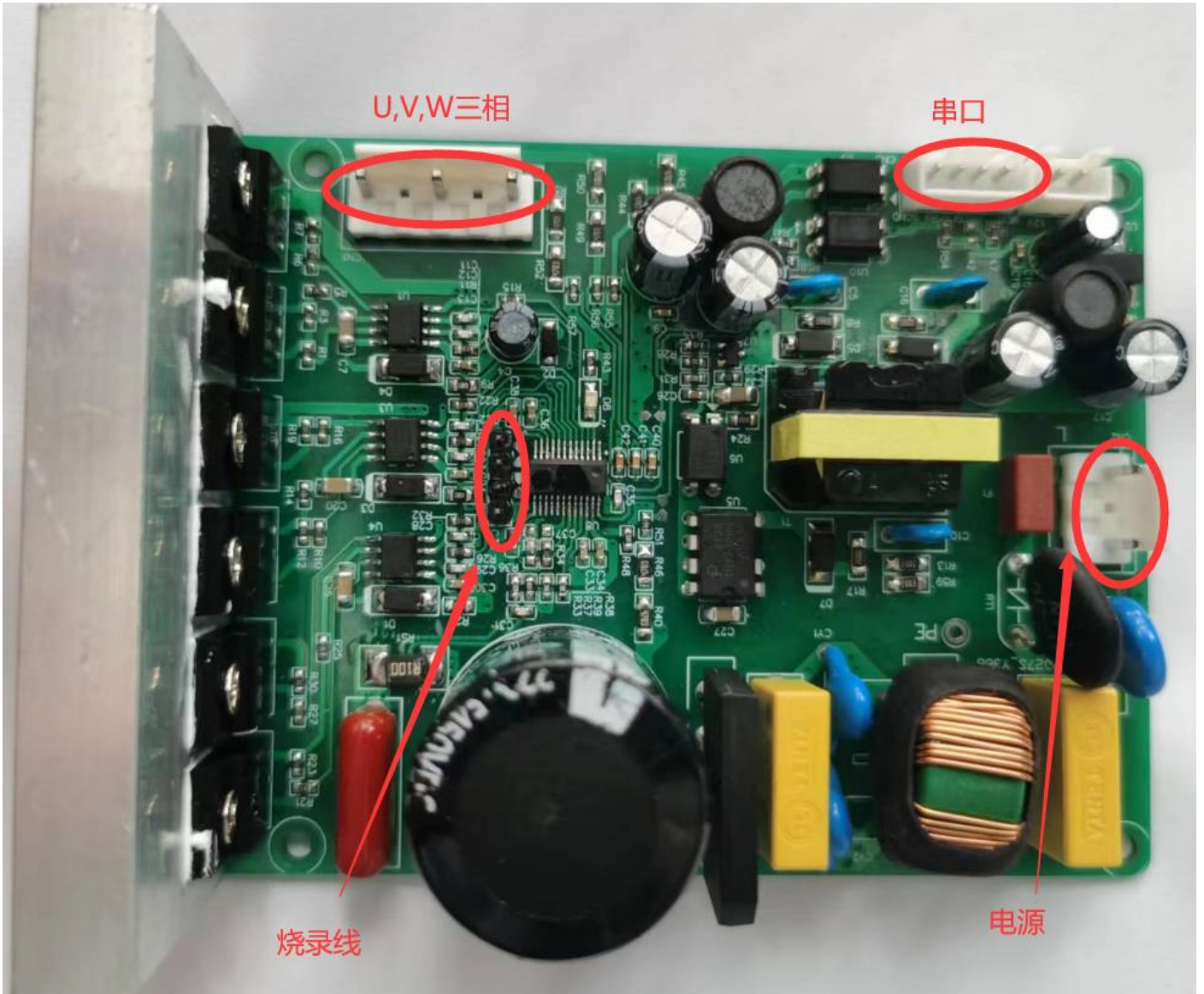


图 2-1 硬件实物图

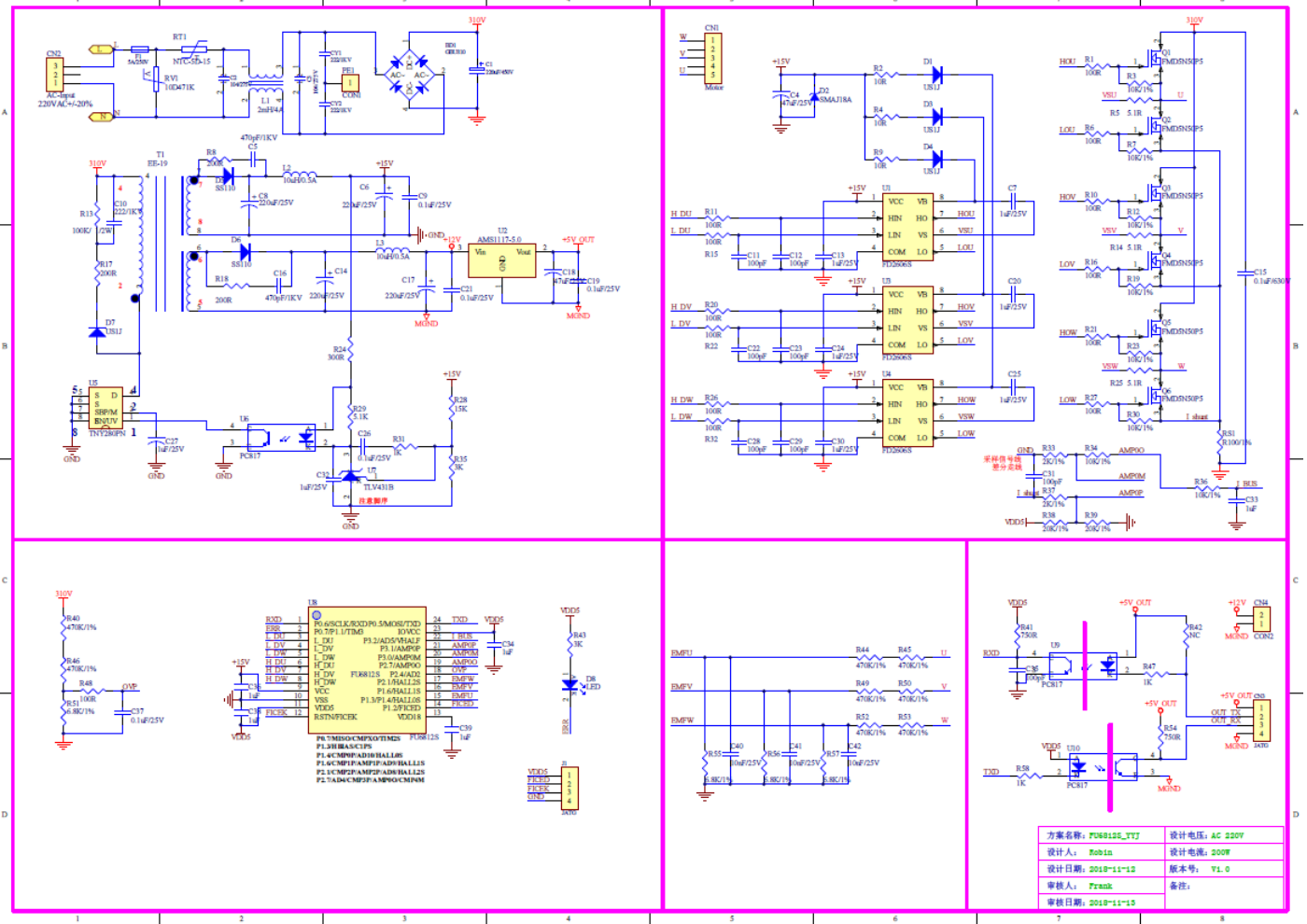
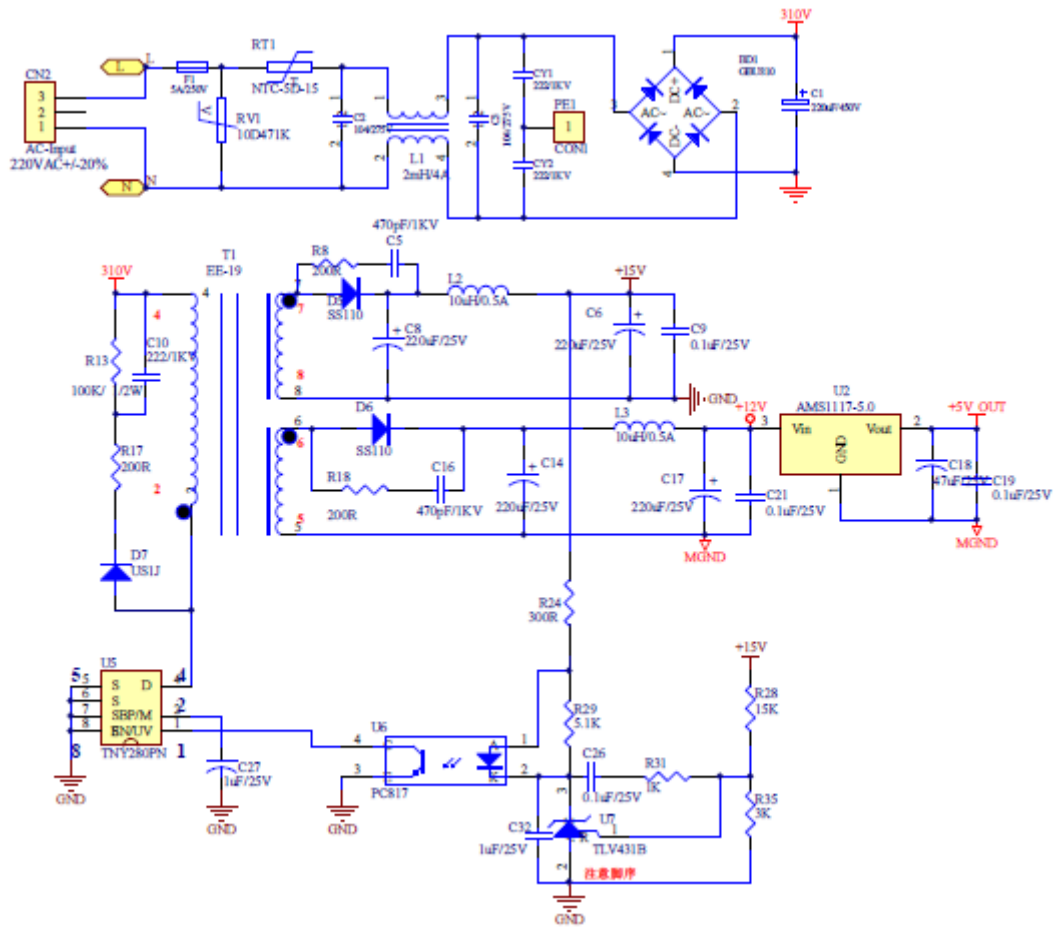


图 2-2 硬件原理图

使用方式: 该板子为油烟机应用方案的专用 DEMO 板子, 直接上电即可使用。

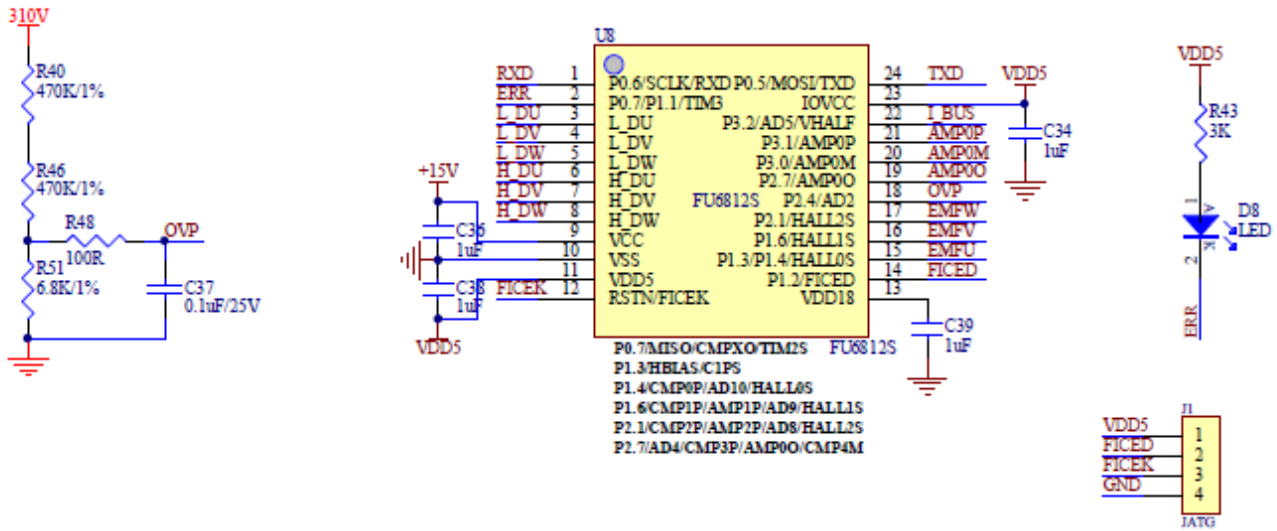
注意事项: 根据具体电机电压和电流大小, 合理配置母线电压比, 运放放大倍数, 采样电阻, 反电动势检测电路分压比。

## 2.1.1 电源部分



使用方式: AC220V 通过 CN2 端口接入。

## 2.1.2 芯片主体

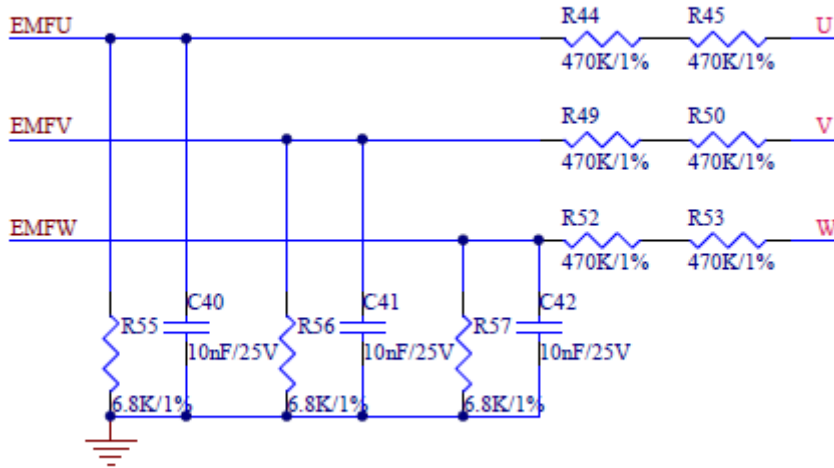


使用方式:

1. FU6812S2 应用于高压 6-NMOSFET 驱动应用。其中 J1 为烧录线接口。
2. R40,R46 和 R51 为母线采样电路，R48 和 C37 为一阶滤波电路，对母线电压进行滤波，OVP 端固定连接 AD2 口，芯片内部电压采集端口已经固化不可更改。
3. R40、R46 和 R51 需要用 1%精度电阻；
4. 最大采样电压 =  $(R40 + R46 + R51)/(R51) * VREF$ ；
5. 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  $0.8 * VREF$ 。

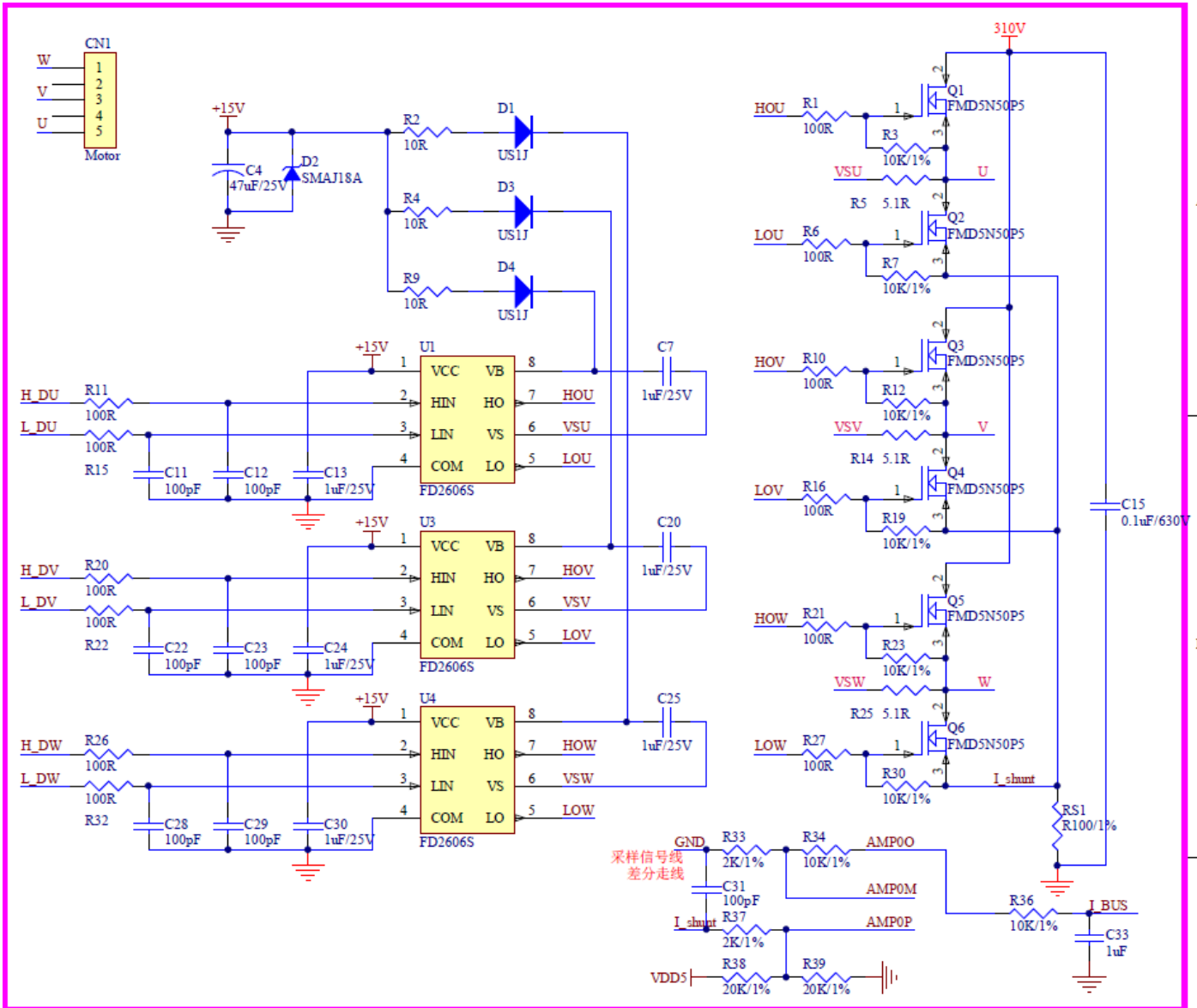


### 2.1.3 反电动势检测电路



注意事项: 该电路是检测反电动势的, 可用于反电动势 BEMF 顺逆风检测功能。

## 2.1.4 功率驱动部分

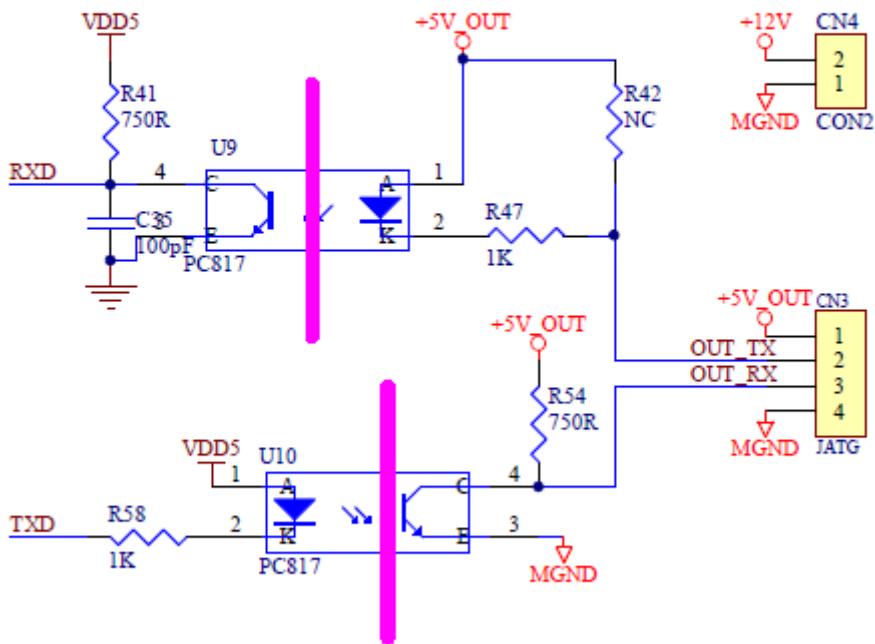


注意事项:

1. 最大电流情况下，采样电阻功率不能超过额定功率的 80%；
2. R33,R34,R37,R38 和 R39 电路部分为运放配置电路，此电路按图中所示配置为外部运放配置，软件中需要配置为外部运放，放大倍数为 5 倍；如果只焊接 R33 和 R37,两个电阻阻值都为 1K 欧，R34,R38 和 R39 不需要焊接，可配置内部运放，放大倍数由软件设置；
3. C19 参数不可调整，精度要求 10%；
4. R33、R34、R37、R38 和 R39 需要用 1%精度电阻；
5. R36 和 C33 电容搭建的一阶 RC 滤波电路主要用于对母线电流进行滤波，I\_BUS 端接 AD 口用于采集母线电流；
6. 放大倍数 =  $R34/R33 = 5$ ；

7. 最大采样电流 =  $(V_{REF} - V_{HALF}) / \text{放大倍数} / \text{采样电阻值}$ ;
8. 最大采样电流一般设置为最大母线电流的 4 倍左右。

### 2.1.5 串口通信接口



注意事项: 串口通信电路需要用光耦进行隔离。

### 3 软件原理

#### 3.1 电机状态机流程图

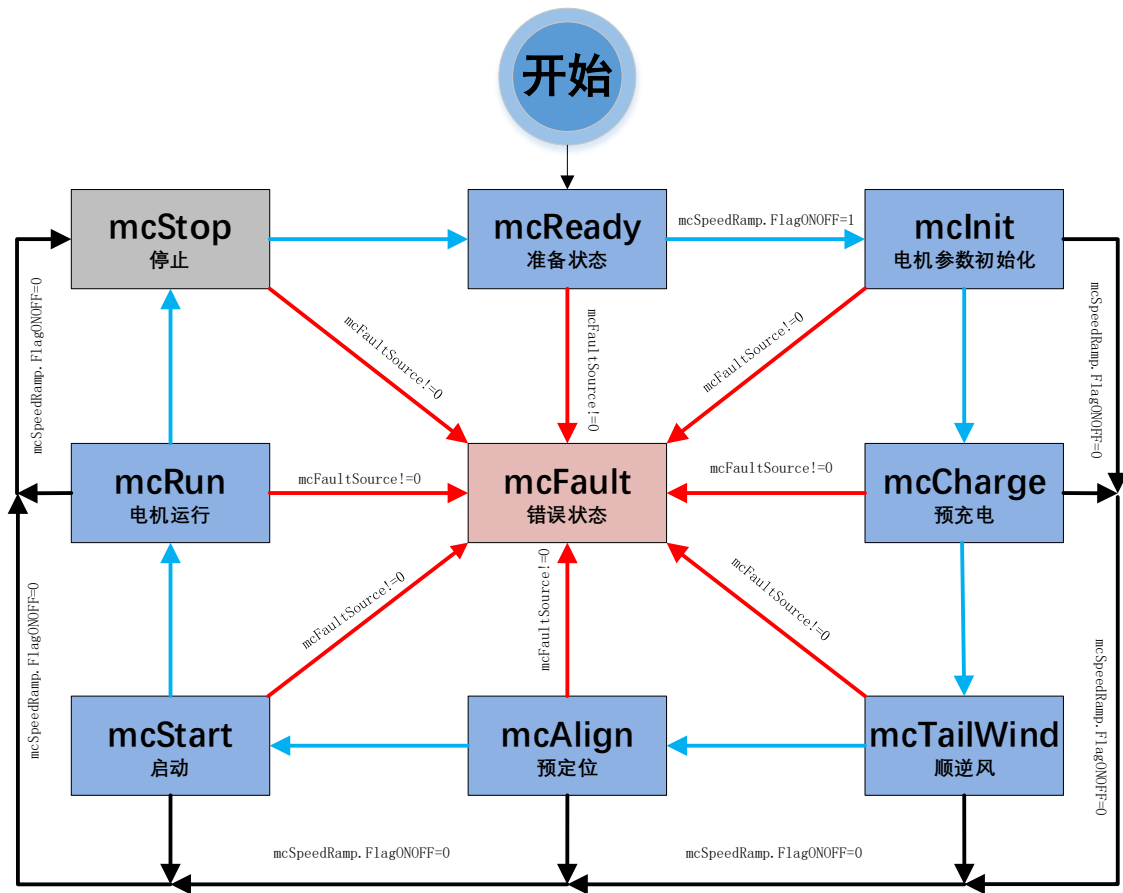


图 3-1 电机状态机流程图

如图所示，电机状态机分为三条路径：

1. 运行: mcReady → mcInit → mcCharge → mcTailWind → mcAlign → mcStart → mcRun;
2. 停机: mcInit、mcCharge、mcTailWind、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. mcCharge: 预充电状态主要分别开 U 相，UV 相和 UVW 相对三相对应的自举电容进行充电；

4. **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(油烟机没有逆风的情况); 检测到静止时, 往下执行;
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**, **mcAlign** 两个状态都有使能位, 当没使能时, 直接跳转到下一个状态。例如: **mcAlign** 没使能时, **mcTailWind** 直接跳转到 **mcStart** 状态。

### 3.2 程序流程图

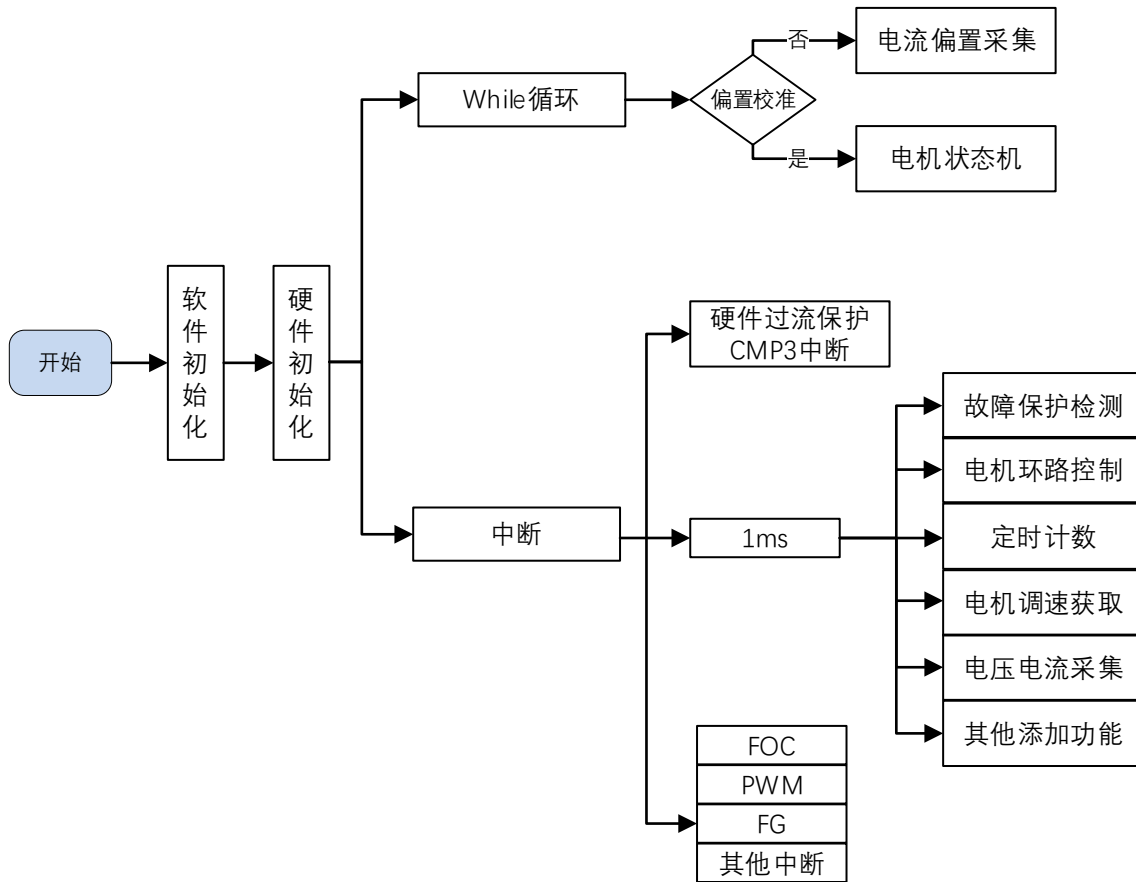


图 3-2 程序执行流程图

### 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
StarRampDealwith();        // 电机启动 ATO 爬坡控制
Fault_Detection();         // 故障检测
TargetSpeed_Collection();  // 获取调速信号，选择不同速度控制方式
  
```

### 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 串口中断

串口中断主要是接收上位机发送速度控制等指令启动电机运行，并将错误源等数据发送给上位机。

## 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:

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

其中，P 为电机极对数。

示例，测量反电动势波形如下：

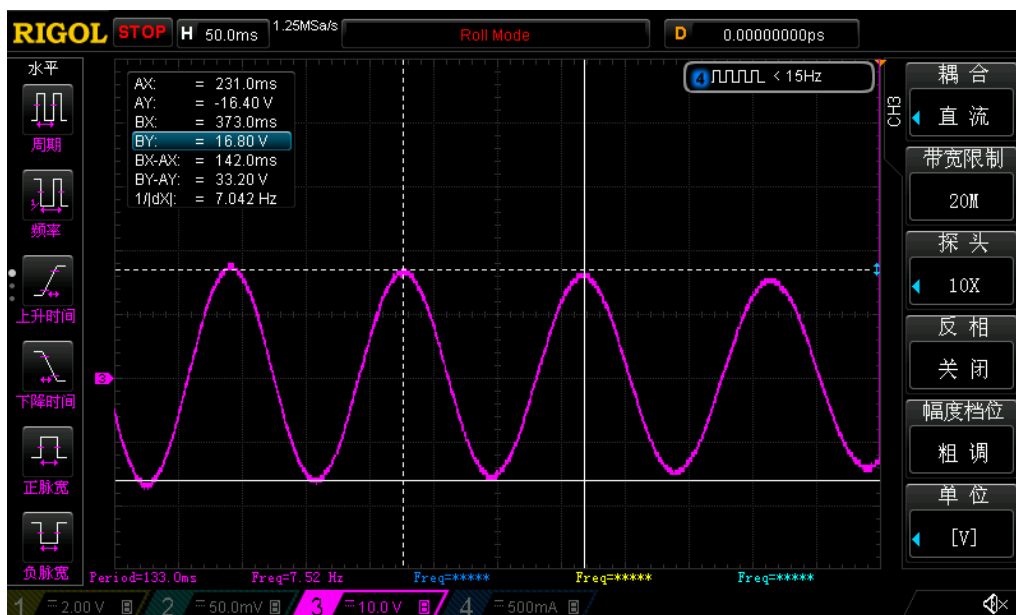


图 4-1 反电动势波形

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  $K_e = 1000 * 4 * \frac{33.2}{2 * 1.732 * 60 * 7.042} = 90.73$  (注: 选择 A0 估算器可以不用测试反电动势常数  $k_e$  值)

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

```

/*-----*/
#define Pole_Pairs          (4.0)           // polar logarithm
#define LD                  (0.0302* 1.0)  // (H) D-axis inductor
#define LQ                  (0.0302* 1.0)  // (H) Q-axis inductor
#define RS                  (17.86*1.0)    // (Ω) Phase resistance

/****AO does not need Ke****/
#define Ke                  (129.23)//(76.84)//(78.0)//(69.1)//(81.78)

#define MOTOR_SPEED_BASE   (4000.0)       // (RPM) Speed Base
/*-----*/
    
```

### 4.2 确认芯片内部相关数据配置

```

/*Chip parameter values-----*/
/*CPU and PWM Parameter*/
#define PWM_FREQUENCY      (16.0)         // (kHz) Carrier frequency
/*deadtime Parameter*/
#define PWM_DEADTIME       (1.0)         // (us) Dead time
/*single resistor sample Parameter*/
#define MIN_WIND_TIME      (2.2)         // (us) Minimum sampling window for single resistor, recommended value: dead time + 0.9us
/*Motor parameters*/
    
```

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。油烟机一般转速不太高, 可先用默认 16K 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  $1000/16/PWM\_FREQUENCY > MIN\_WIND\_TIME > 2 * PWM\_DEADTIME$ 。

### 4.3 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  $0.8 * V_{REF}$ , 如某电机的最大电压为 30V, ADC 基准  $V_{REF}$  为 4.5V, 此时分压比建议不低于:  $30/0.8/4.5 = 8.33$ ; 如果分压比太小, 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  $30V/40 = 0.75V$ , 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)。

```
1
2 #define HW_RSHUNT           (0.1)           ///< (Ω) Sampling resistance
3
4 /**
5  * Magnification setting
6  * @param (AMP2x)           Internal PGA magnification 2X
7  * @param (AMP4x)           Internal PGA magnification 4x
8  * @param (AMP8x)           Internal PGA magnification 8x
9  * @param (AMP16x)          Internal PGA magnification 16x
10 * @param (xxxxxx)          External magnification mode fills in the corresponding magnification
11 */
12 #define HW_AMPGAIN           (5.0)           ///< Magnification setting
13
14 /**
15  * Reference voltage setting
16  * @param (VREF5_0)         Reference voltage set to 5.0V
17  */
18 #define HW_ADC_VREF           (VREF5_0)      ///< (V) ADC Reference Voltage
19
20 /**
21  *hardware voltage sample Parameter*/
22 /*Bus voltage sampling voltage divider circuit parameters*/
23 #define RV1                   (470.0)        // (kΩ) Bus voltage divider resistor1
24 #define RV2                   (470.0)        // (kΩ) Bus voltage divider resistor2
25 #define RV3                   (6.8)         // (kΩ) Bus voltage divider resistor3
26 #define VC1                   (1.0)         // Voltage compensation factor
```

其中,

- 1) 母线分压比 =  $(RV1 + RV2 + RV3)/RV3$ ;
- 2) VC1 为电压补偿系数, 只应用在启动时刻, 可先不动。

## 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](#);

- 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；
- 各种保护的使能位在程序 `Protect.h` 文件中，可以选择性配置开启或者关闭。

```

/*Protection Enable*/
#define OverSoftCurrentProtectEnable    (1)           // Software overcurrent protection, 0, disable; 1, enable
#define VoltageProtectEnable           (0)           // Voltage protection, 0, disable; 1, enable
#define StallProtectEnable              (1)           // Stall protection, 0, disable; 1, enable
#define PhaseLossProtectEnable          (1)           // Lost phase protection, 0, disable; 1, enable
#define OverPowerProtectEnable          (0)           // Power protection, 0, disable; 1, enable
#define GetCurrentOffsetEnable          (1)           // Bias voltage protection, 0, disable; 1, enable
    
```

#### 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](#)。

```

)
)
) /***Starting current***/
1 #define ID_Start_CURRENT               I_Value(0.0)    // (A) D-axis starting current
2 #define IQ_Start_CURRENT              I_Value(0.2)    // (A) Q-axis starting current
3
4 /***Operating current***/
5 #define ID_RUN_CURRENT                 I_Value(0.0)    // (A) D-axis operating current
6 #define IQ_RUN_CURRENT                 I_Value(0.2)    // (A) Q-axis operating current
7
8 /* Start ATO parameters */
    
```

- 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。

- 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;
- 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`(速度带宽滤波值), 是以限制 FOC 估算器的最大转速输出;

```

/* Start ATO parameters */
#define ATO_BW                            (2.0)           // Filtered value of the observer bandwidth, classical value 1.0 - 200.0
#define ATO_BW_RUN                        (5.0)
#define ATO_BW_RUN1                       (10.0)
#define ATO_BW_RUN2                       (40.0)
    
```

注意事项: 对于油烟机而言, 启动的前 3 个 ATO 影响比较明显, 需要根据实际情况调整。如果选择的是 AO 观测器, 第一个 `ATO_BW` 不用太大。

- 速度带宽滤波值 `SPD_BW`;

```

#define MOTOR_LOOP_RPM                    (80.0)           // (RPM) Switching speed from mode0 to mode1, i.e. closed loop switching speed
#define SPD_BW                            (10.0)           // Filter values for speed bandwidth, classical values are 5.0 - 40.0
    
```

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：**

```

1 //OMEGA start-up parameters
2 #define Motor_Omega_Ramp_ACC          (5.0)           // omega startup increments
3 #define MOTOR_OMEGA_ACC_MIN          (5.0)           // (RPM) Minimum switching speed for omega startup
4 #define MOTOR_OMEGA_ACC_END          (12.0)          // (RPM) Limiting speed for omega startup
5

```

注意事项：

- 1) Motor\_Omega\_Ramp\_ACC 参考值范围 1 ~ 10
- 2) MOTOR\_OMEGA\_ACC\_MIN 参考值范围 5 ~ 20
- 3) MOTOR\_OMEGA\_ACC\_END 参考值范围 5 ~ 20
- 4) MOTOR\_LOOP\_RPM 需要大于 MOTOR\_OMEGA\_ACC\_END，参考值范围 30 ~ 100

**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:**

```

/*****Runtime parameters*****/
/*Current loop parameter setting value-----*/
#define DQKPstart          _Q12(0.5)           // DQ axis KP at start-up
#define DQKIstart          _Q15(0.002)        // DQ axis KI at start-up

#define DQKP                _Q12(1.0)         // DQ axis KP during operation
#define DQKI                _Q15(0.002)        // DQ axis KI during operation

/* D-axis parameter setting */

```

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-axis parameter setting */
#define DOUTMAX            _Q15(0.4)           // D-axis maximum limit value, unit: output duty cycle
#define DOUTMIN            _Q15(-0.4)         // D-axis minimum limit value, unit: output duty cycle
/* Q-axis parameter setting, default 0.99 can be */
#define QOUTMAX            _Q15(0.96)         // Q-axis maximum limit value, unit: output duty cycle
#define QOUTMIN            _Q15(-0.96)        // Q-axis minimum limit value, unit: output duty cycle

```

注意事项：

- 1) 通过获取 FOC\_\_UQ 的与 QOUTMAX 值比较，判断电机已经输出是否饱和，如果 FOC\_\_UQ 的值很接近 QOUTMAX 表明电机接近饱和；
- 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

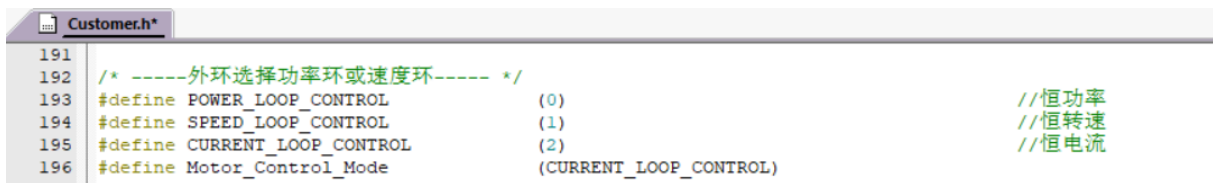
```

92
93 /*Mode selection setting value-----*/
94 /*IPM test mode*/
95 #define IPMtest                (0)                // IPM test or MOS test, MCU output fixed duty cycle
96 #define NormalRun              (1)                // Normal operation by motor status machine
97 #define IPMState                (IPMtest)
98
99
100
101
102 /**
103  * Closed loop method selection
104  * @param (ONOFFTEST)        Start/Stop Test Mode
105  * @param (NONEMODE)        Directly given target value, no speed regulation
106  * @param (UARTMODE)        Serial speed control
107  */
108 #define SPEED_MODE                (NONEMODE)        ///< Closed loop method selection
109
110
    
```

将IPMState置IPMtest，调速模式选择NONEMODE开机，这个时候电机停留在预充电装填，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

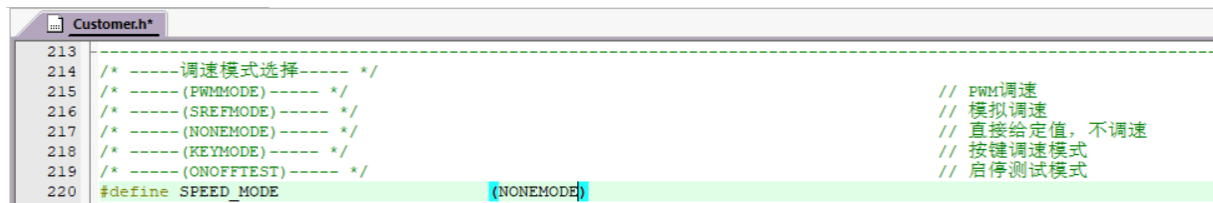
1. 将环路选择为电流环；



```

191
192 /* -----外环选择功率环或速度环----- */
193 #define POWER_LOOP_CONTROL      (0)                //恒功率
194 #define SPEED_LOOP_CONTROL      (1)                //恒转速
195 #define CURRENT_LOOP_CONTROL    (2)                //恒电流
196 #define Motor_Control_Mode      (CURRENT_LOOP_CONTROL)
    
```

2. 调速方式先选为直接给定值，调整给定值电流的大小，以此来控制电流环的电流大小(注意给的是相电流的值)；



```

213
214 /* -----调速模式选择----- */
215 /* -----(PWMMODE)----- */                // PWM调速
216 /* -----(SREFMODE)----- */                // 模拟调速
217 /* -----(NONEMODE)----- */                // 直接给定值, 不调速
218 /* -----(KEYMODE)----- */                // 按键调速模式
219 /* -----(ONOFFTEST)----- */                // 启停测试模式
220 #define SPEED_MODE                (NONEMODE)
    
```

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.PowerIpf(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，要根据实际硬件电路去修改。具体位置如下图。

```

/*****DcBus的采样获取值并滤波*****/
mcFocCtrl.mcDcBusFlt    = ADC2_DR << 3;
mcFocCtrl.mcIbusFlt     = LPFFunction(ADC11_DR << 3, mcFocCtrl.mcIbusFlt, 2);
/* -----环路响应, 如速度环、转矩环、功率环等----- */

```

#### 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. 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. 一般油烟机都是用的速度环，因此将环路选择为速度环；

```

5 |
6 | /*Control method selection*/
7 | #define POWER_LOOP_CONTROL          (0)          //Constant power
8 | #define SPEED_LOOP_CONTROL         (1)          //Constant speed
9 | #define CURRENT_LOOP_CONTROL       (2)          //Constant current
10| #define Motor_Control_Mode         (SPEED_LOOP_CONTROL)

```

2. 设置速度曲线的最大值，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\_Speed 为曲线的最小速度值，可根据客户实际需要设定；调整 SKP 和 SKI 调整速度环控制的响应快慢，保证速度调节快但是不要超调。

```

01 | /*Rotational speed parameter value-----*/
02 | /* motor run speed value */
03 | #define MOTOR_SPEED_MIN_RPM        (500.0)      // (RPM) Minimum operating speed
04 | #define MOTOR_SPEED_MAX_RPM        (1600.0)     // (RPM) Maximum operating speed
05 | #define MOTOR_SPEED_STOP_RPM       (300.0)      // (RPM) Running stop speed
06 | #define MOTOR_POWER_LIMIT          (7000.0)
07 |
228| /*Loop parameter setting value-----*/
229| #define SPEED_LOOP_TIME             (10)         // Loop regulation cycle,Unit: milliseconds
230| #define SKP                         _Q12(1.0)   // Loop KP
231| #define SKI                         _Q15(0.01)  // Loop KI
232|
233| #define SOUTMAX                     I_Value(1.2) // (A) Loop maximum limit value
234| #define SOUTMIN                     I_Value(0.01) // (A) Loop minimum limit value
235|
236| /* -----Climbing increments----- */

```

```

/* -----Climbing increments----- */
#if (Motor_Control_Mode == CURRENT_LOOP_CONTROL)
#define SpeedRampInc          I_Value(0.01) // Current increment
#define SpeedRampDec          I_Value(0.01) // Current reduction
#else
#define SpeedRampInc          (5.0) // Target value climbing acceleration
#define SpeedRampDec          (5.0) // Target value climbing acceleration
#endif
/**

```

说明:

调试过程中，如果加入速度环后速度不稳，电流波形波动也比较大，首先确认是否 PI 参数调试不合适，先确认只有电流环的时候，电流波形是否稳定，如果波形稳定加速度环后不稳定，那就需要优化速度环 PI，如果电流环也不稳定，而且低速时候稳定高速时候不稳定，那可能是油烟机的负载不均，速度高了风阻变大导致速度不稳，这个时候只能尝试优化 PI，可以先将 PI 都给很小，让速度响应变慢，看看电流波形是否变好，否则逐步加大 PI。

## 4.9 串口调速功能

1. 一般油烟机都是上位机发送速度指令，所以调速方式一般选择串口调速: 将调速模式修改为 **UARTMODE**。通过串口调试助手发送数据，测试串口中断能否正常进入，如果能够正常进入判断接收到的数据是否为上位机发送的数据。串口接收正常后，将接收到的数据解析出来，赋值到相应的速度控制变量，实现串口调速功能。

```

/**
 * Closed loop method selection
 * @param (ONOFFTEST)      Start/Stop Test Mode
 * @param (NONEMODE)      Directly given target value, no speed regulation
 * @param (UARTMODE)      Serial speed control
 */
#define SPEED_MODE          (UARTMODE) //< Closed loop method selection

```

2. 接线方式: 完成正常的串口通信，正确的接线是最关键的一步。上位机的 **TX** 需要通过外部电路接到芯片的 **RX**，同样的上位机的 **RX** 和芯片的 **TX** 接一起。

串口通信常见的问题:

- 1) 中断进不去，通常串口中断进不去有两种可能的问题: 一、可能是线没接对，可以用示波器测试上位机到芯片的 **RX** 端时候有信号进入，如果有信号进入表明接线正常，如果没有排查接线问题; 二、串口初始化配置的问题，可能初始化没配置对，可以对照手册检查初始化配置; 另外就是串口中断入口也需要检查一下。
- 2) 串口数据接收到了但是值不对，可能是波特率的配置问题，可以查看上位机的波特率和程序配置的波特率是否一样;
- 3) 数据能够正常接收，但是没办法正常的进行数据解析，有可能是 **CRC** 校验或者奇偶校验出了问题，需要进行排查。

## 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\_Speed 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)。

```
1 |
2 | /**
3 |  * Closed loop method selection
4 |  * @param (ONOFFTEST)      Start/Stop Test Mode
5 |  * @param (NONEMODE)      Directly given target value, no speed regulation
6 |  * @param (UARTMODE)      Serial speed control
7 |  */
8 | #define SPEED_MODE          (ONOFFTEST)          ///< Closed loop method selection
9 |
10 |
11 |
12 | /**
13 |  * Time configuration in test mode
14 |  * @param (StartON_Time)   Opening time
15 |  * @param (StartOFF_Time)  Closing time
16 |  */
17 | #define StartON_Time        (3000)
18 | #define StartOFF_Time       (2000)
```



## 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

油烟机选择 Omega 启动，程序对应默认即是该启动方式。

```

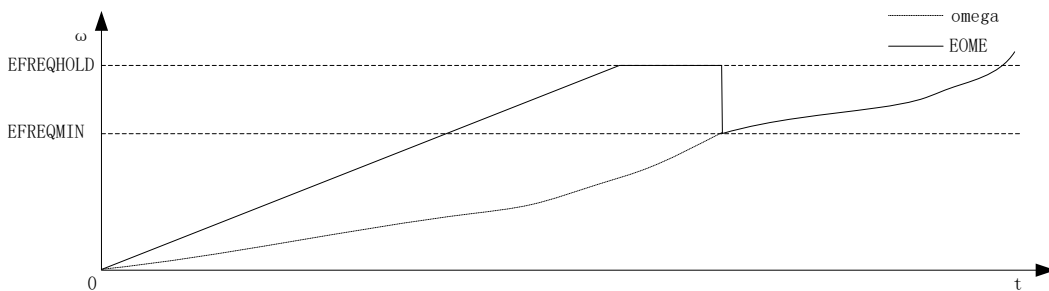
Customer.h*
257 #define TailWind_Mode                (BEMFMethod)
258
259 /* -----开环启动模式选择----- */
260 /* -----(Open_Start)----- */                // 开环强拖启动
261 /* -----(Omega_Start)----- */                // Omega启动
262 /* -----(Open_Omega_Start)----- */            // 先开环启, 后Omega启动
263 #define Open_Start_Mode                (Omega_Start)
    
```

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。

```

//OMEGA start-up parameters
#define Motor_Omega_Ramp_ACC                (5.0)                // omega startup increments
#define MOTOR_OMEGA_ACC_MIN                (5.0)                // (RPM) Minimum switching speed for omega startup
#define MOTOR_OMEGA_ACC_END                (12.0)               // (RPM) Limiting speed for omega startup
    
```

启动过程如下图所示:



### 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 AMP0 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. 启动过早的切入估算器，导致电机速度还没起来就切入估算器从而启动失败，可以适当调整滑膜最小运行速度 MOTOR_SPEED_SMOMIN_RPM； 3. Omega 启动参数中 MOTOR_OMEGA_ACC_MIN 的值影响切入从强制给角度到切入估算器的时机，可以调整此值调整估算器切入时机。
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，各种保护的设定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 OverHardcurrentValue 的大小即可。

```

7 #define Hardware_FO_Protect      (1)           // Hardware FO overcurrent protection enable, suitable for IPM with FO protectic
8 #define Hardware_CMP_Protect    (2)           // Hardware CMP comparison overcurrent protection enable for MOS tube applicatic
9 #define Hardware_FO_CMP_Protect (3)           // Hardware CMP comparison and FO overcurrent protection are both enabled
0 #define Hardware_Protect_Disable (4)         // Hardware overcurrent protection disabled for testing
1 #define HardwareCurrent_Protect (Hardware_CMP_Protect) // Hardware overcurrent protection implementation method
2
3 /*Hardware Overcurrent Protection Comparison Value Source*/
4 #define Compare_DAC              (0)           // DAC setting hardware overcurrent value
5 #define Compare_Hardware         (1)           // Hardware setting hardware overcurrent value
6 #define Compare_Mode             (Compare_DAC) // Source of hardware overcurrent values
7 #define OverHardcurrentValue     (2.0)        // Hardware overcurrent value in DAC mode,Unit: A
8

```

## 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。

```

/*Software overcurrent protection parameter setting*/
#define OverSoftCurrent_DectTime (5)           // (ms)Software overcurrent detection time
#define OverSoftCurrentValue     I_Value(2.0) // (A) Software overcurrent value

```

## 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。

```

/*Over-undervoltage protection parameter setting*/
#define Over_Protect_Voltage      (396)        // (V) DC voltage overvoltage protection value 280*1.414
#define Over_Recover_Vlotage      (368)        // (V) DC voltage overvoltage protection recovery value 260*1.414
#define Under_Protect_Voltage     (230)        // (V) DC voltage undervoltage protection value 162*1.414
#define Under_Recover_Vlotage     (250)        // (V) DC voltage undervoltage protection recovery value 178*1.414

```

## 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。

```

/*Out-of-phase protection parameter setting*/
#define PhaseLoss_DectTIME        (50)         // (ms) Peak detection cycle time, to cover at least one full electrical cycle
#define PhaseLossCurrentValue     I_Value(0.10) // (A) Out-of-phase current value
#define PhaseLossTimes            (2.0)

```

### 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。

```

44  /*Stall protection parameter setting*/
45  #define Stall_Protect_Time           (50)                // (ms) Stall protection time
46
47  #define Stall_Delay_DectTime         (1000)             // (ms) Delayed detection of counter-
48  #define Stall_DectEsValue1          (1500)             // Counter-electromotive force judge
49
50  #define Stall_DectSpeed               (1200)            // (RPM) Stall detection speed, confi
51  #define Stall_DectEsValue2          (3000)             // Counter-electromotive force judgment
52
    
```

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;

```

#define MOTOR_SPEED_STAL_MAX_RPM      (2000.0)           // (RPM) Maximum speed for blocking protection
#define MOTOR_SPEED_STAL_MIN_RPM      (300.0)            // (RPM) Minimum speed for blocking protection
    
```

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。

```

#define FOCMode_DectTime                (3000)           // (ms) Determine the duration of being in mode0
#define StallCurrentValue               I_Value(4.0)     // (A) Stall protection overcurrent value
    
```

## 5.2.5 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%。

```

37  /* -----偏置电压保护----- */
38
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
    
```

## 5.2.6 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，当负载变大时，在相同的转速下，功率会逐渐增大，超过板上能够承受的最大功率从而损坏硬件板子，因此需要进行功率限制处理。

限功率功能目前有 3 种方式：

1. 通过对目标值限制实现，当检测功率超过保护阈值之后，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，检测到功率超过限制值，那么程序会切入功率闭环从而达到限功率功能，当取消堵风口由于负载恢复此时转速下转速会超过目标转速值，此时切回速度闭环，从而实现了堵风口限功率功能。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，由 PI1 运算单元做速度闭环控制，输出给到 IQREF，由 PI2 运算单元做功率闭环控制，输出给到 FOC\_QMAX 用来限制电流环的输出，代码如下：

```
SpeedPICtrl.ExtEK = MCtrl.ActualValue - mcFocCtrl.SpeedFlt;
SpeedPICtrl.ExtOut = HW_One_PI(SpeedPICtrl.ExtEK);           //速度环调节
FOC_IQREF = SpeedPICtrl.ExtOut;
FOC_QMAX = HW_One_PI2(MCtrl.PowerLimitValue - mcFocCtrl.PowerIpf); //限制功率
```

目前程序已经添加了限功率功能，可直接使用。

## 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。
在堵风口后，快速放开，这时候电流会突然变得很大导致硬件过流	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(注意修改后会影响到启动跟运行)。
注意事项：修改参数后，基本都会影响到启动和运行性能，解决好问题后要重新测试确认。	

## 8 修改记录

版本号	修改详细内容说明	生效日期	修订者
V1.0	初稿	2022/06/24	蔡坤旺
V1.1	修改部分格式问题	2023/08/02	李佳妮

## 版权说明

版权所有©峰昭科技（深圳）股份有限公司（以下简称：峰昭科技）。

为改进设计和/或性能，峰昭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昭科技的客户进行一般性使用。峰昭科技的客户应确保采取适当行动，以使其对峰昭科技产品的使用不侵犯任何专利。峰昭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。

本文档版权归峰昭科技所有，未经峰昭科技明确书面许可，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。

峰昭科技（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](http://www.fortiortech.com)

本文件所载内容

峰昭科技（深圳）股份有限公司版权所有，保留一切权力。