

冰箱压缩机应用手册

三相电机控制 MCU FU6862Q

峰昭科技(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 功率驱动电路.....	7
2.1.4 运放配置电路.....	8
2.1.5 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.....	11
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.....	12
3.3.4 CMP3 中断.....	12
3.3.5 Timer3 中断.....	12
3.3.6 FO 中断.....	12
4 调试步骤.....	13
4.1 电机参数配置.....	13
4.1.1 电机参数.....	13
4.1.2 电机参数测量方法.....	13
4.1.3 对应程序.....	14
4.2 芯片内部参数配置.....	14
4.3 硬件参数配置.....	15
4.4 保护参数配置.....	16
4.5 启动参数配置.....	17
4.6 常见问题及解决办法:.....	20
4.7 速度环调试.....	20
4.8 增加 CLOCK 调速功能.....	21
5 功能介绍.....	22

5.1 启动调试	22
5.1.1 Omega 启动	22
5.1.2 启动常见问题&解决方式.....	23
5.2 弱磁控制	23
5.3 保护介绍	23
5.3.1 过流保护	24
5.3.2 电压保护	24
5.3.3 缺相保护	24
5.3.4 堵转保护	25
5.3.5 过温保护	25
5.3.6 过功率保护	26
6 修改记录	27

1 概述

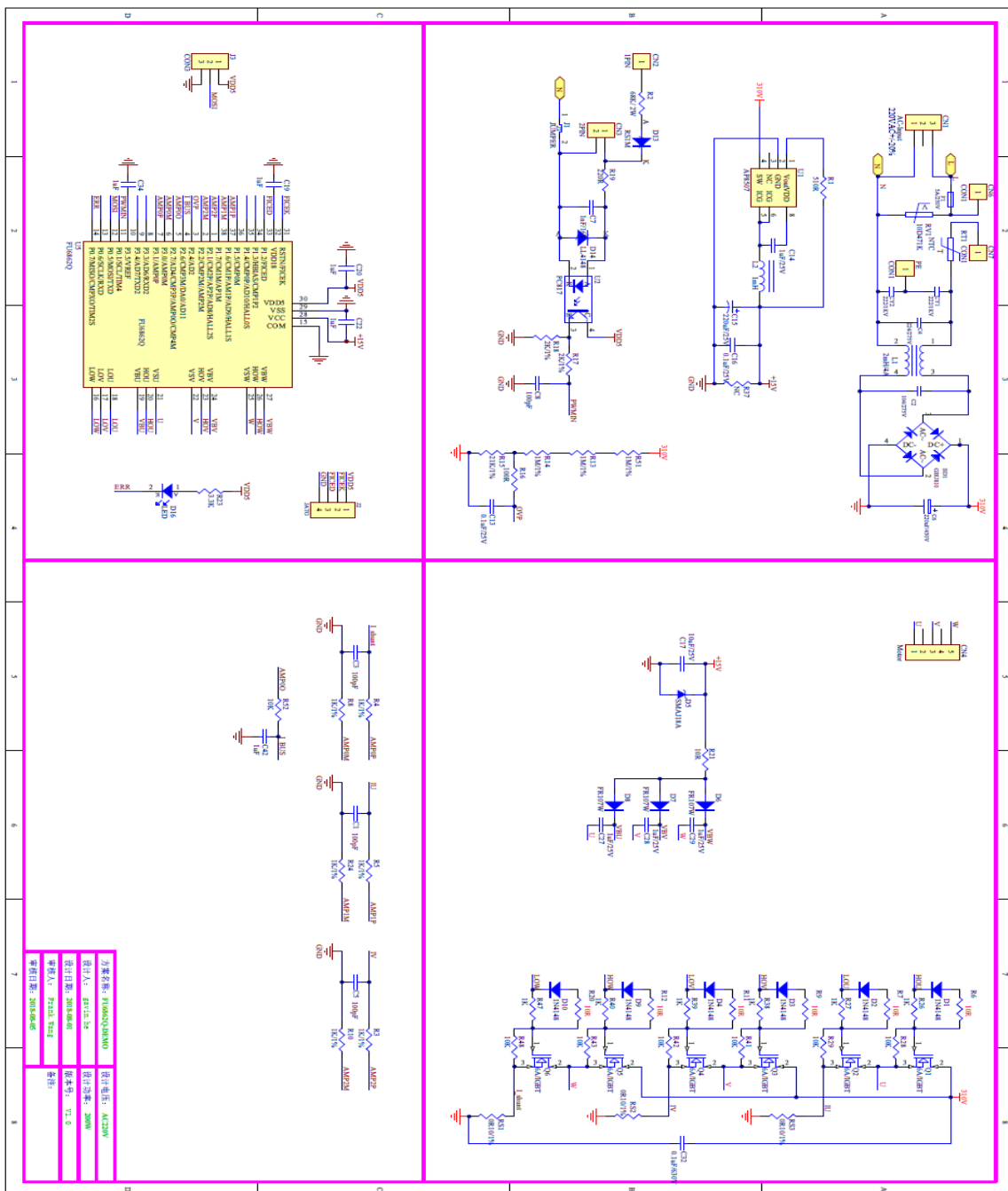
本调试手册详细介绍了如何使用峰昭科技的 FU6862Q 芯片，在冰箱压缩机专用 DEMO 板子上，对直流无刷冰箱压缩机电机进行无感 FOC 驱动控制。阅读手册时，第二章节硬件原理跟第三章节软件原理可以大致先浏览一遍，重点放在第四章调试步骤。

涉及的软/硬件

软/硬件和模块	名称	章节	备注
软件	FU6862Q 冰箱压缩机应用方案标准程序	全部	调试需在该工程软件上进行
硬件	FU6862-BX-DEMO_V2.0	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



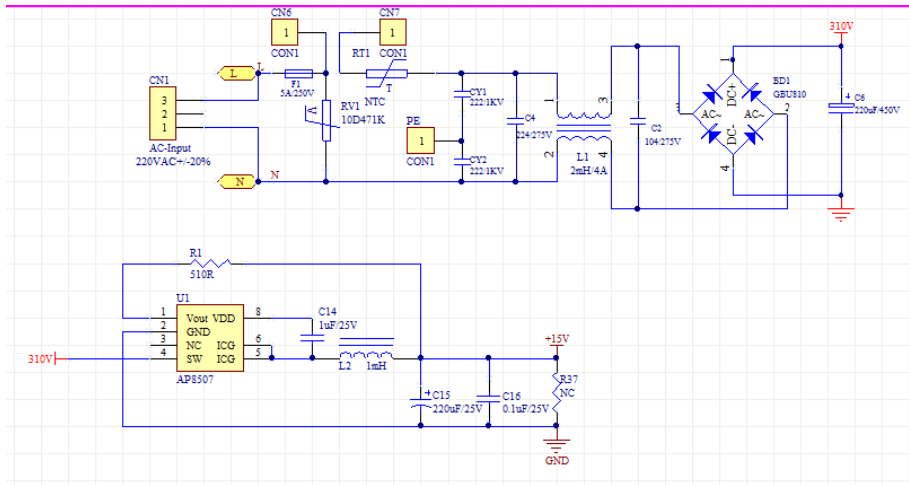
使用方式:

该板子为冰箱压缩机应用方案的专用 DEMO 板子，直接上电即可使用。

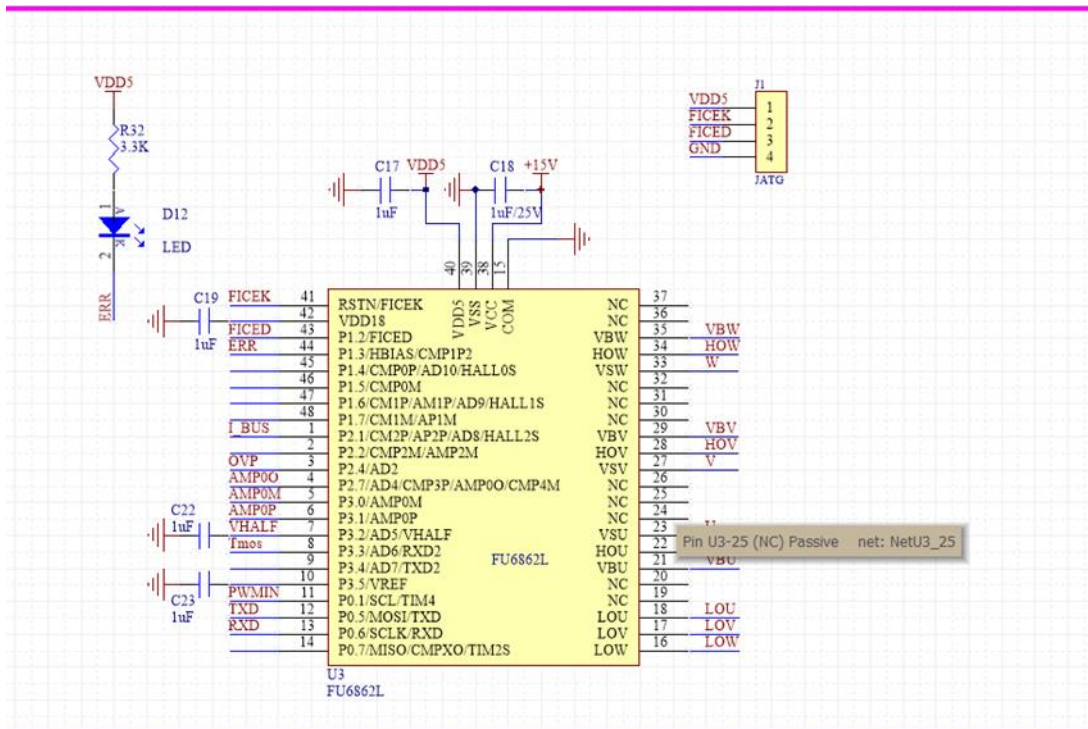
注意事项:

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。

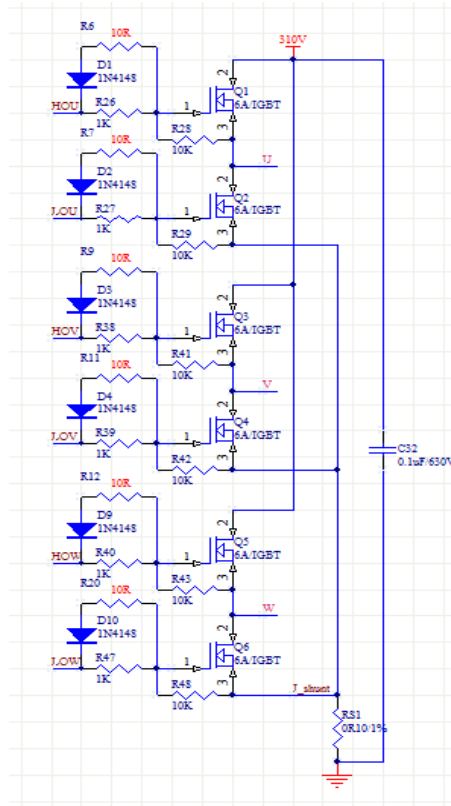
2.1.1 电源电路



2.1.2 芯片电路



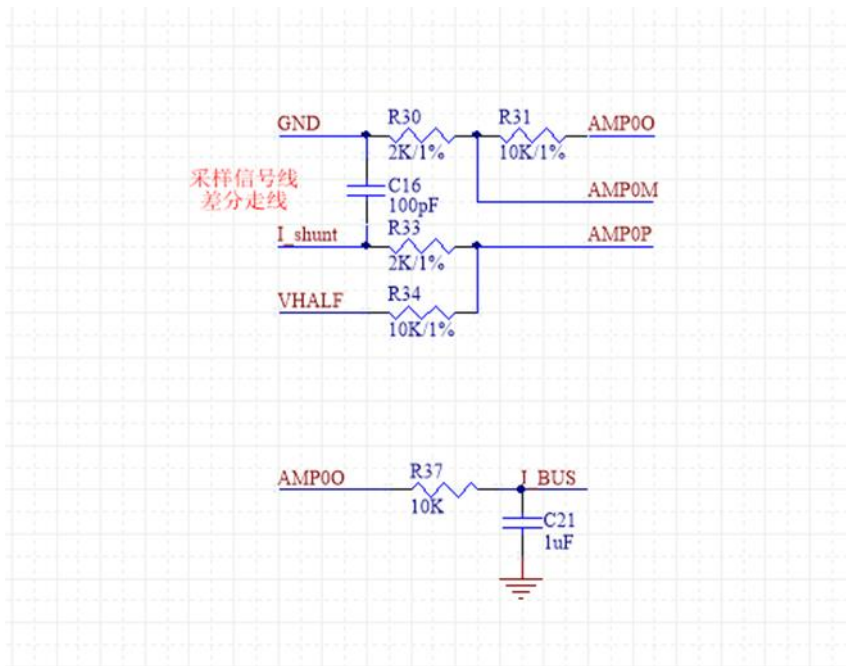
2.1.3 功率驱动电路



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. 驱动电阻 R26、R27、R38、R39、R40、R47 建议 1K，减缓 MOS 打开时的震荡；这样处理是为了保证在电机 UVW 线很长情况下还能采样到正确的相电流，因为 UVW 线很长会导致 MOS 打开时震荡严重，驱动电阻加大可以缩短震荡幅值和震荡时间；
3. 驱动电阻上需并联反接二极管，缩短 MOS 关断时间，驱动电阻过大会导致死区时间变长，为了缩短死区时间，加入反接二极管缩短 MOS 关断时间，进而缩短死区时间。

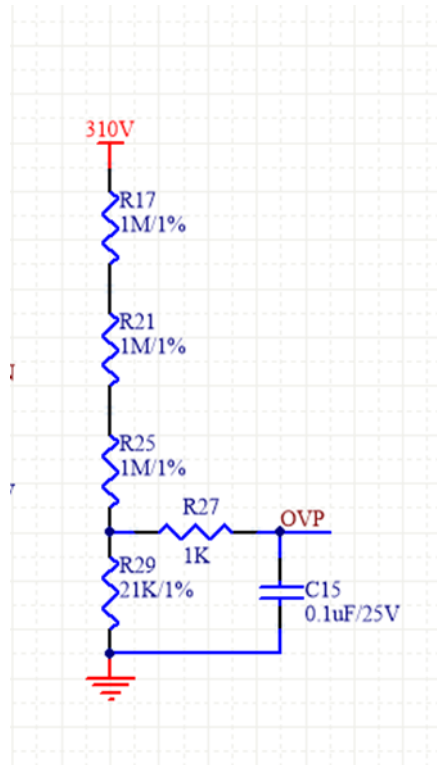
2.1.4 运放配置电路



注意事项:

1. C21 参数不可调整，精度要求 10%；
2. R30、R31、R34、R33、R37 需要用 1%精度电阻；
3. 此处 AD5 用于母线平均电流采样；
4. 放大倍数 = $R31/R30$ ；
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。

2.1.5 母线电压采样电路



注意事项:

1. R27、C15 参数不可调整;
2. R17、R21、R25、R29 需要用 1%精度电阻;
3. 最大采样电压 = $(R17 + R21 + R25 + R29)/(R29) * VREF$;
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

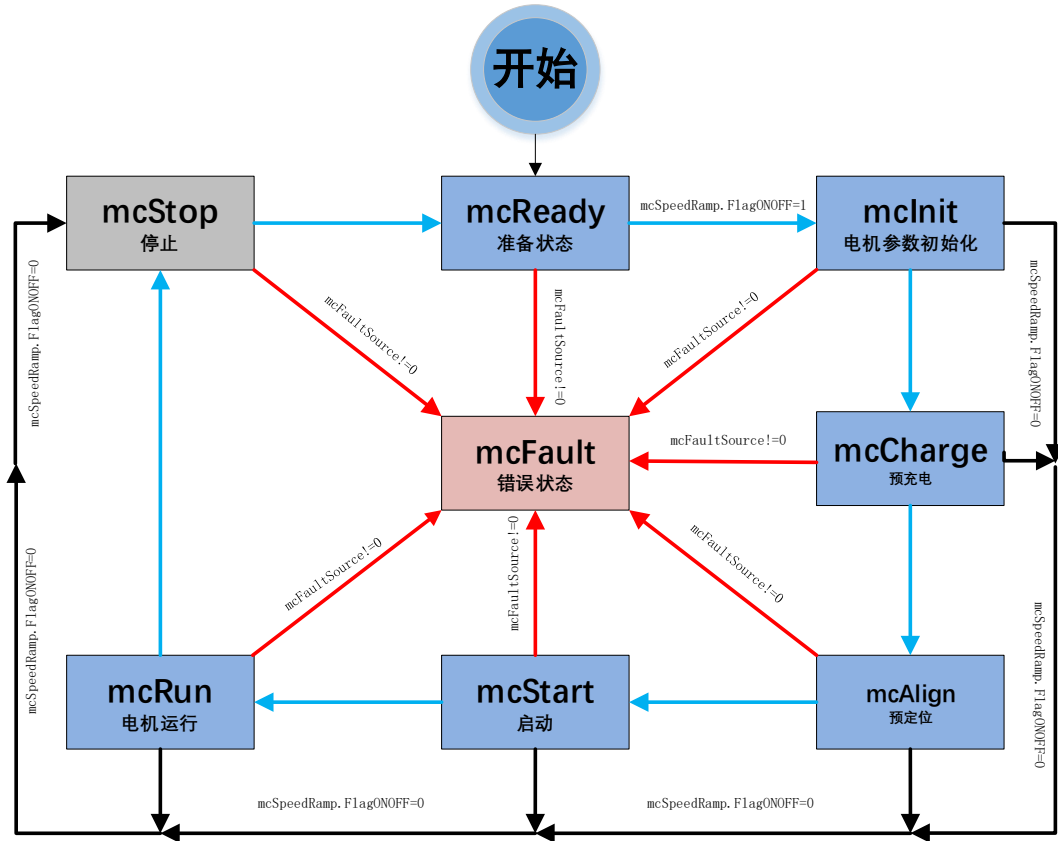


图 3-1 电机状态机流程图

如图所示，电机状态机分为三条路径：

1. 运行: mcReady -> mcInit -> mcCharge -> 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. mcCharge: 预充电状态，给自举电容充电，充电完成跳转到 mcAlign 状态;
4. mcAlign: 预定位状态，定位第一阶段定位电流由 IQ_Align_CURRENT_START 逐渐增加到 IQ_Align_CURRENT_END，同时定位角度由 Align_Angle1 逐渐增加 Align_Angle2; 定位第二阶段

为保持定位角度 Align_Angle2 和定位电流 IQ_Align_CURRENT_END 一段时间；然后角度切入 Align_Angle3，以 MotorRsEstimateID0 运行一段时间后再以 MotorRsEstimateID1 运行一段时间。这部分主要用于做启动前的电阻参数检测。定位结束则跳入下一个状态 mcStart；

5. 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；
6. 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；
7. 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；
8. 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 mclnit 和 mcFault 状态跳转。

3.2 程序流程图

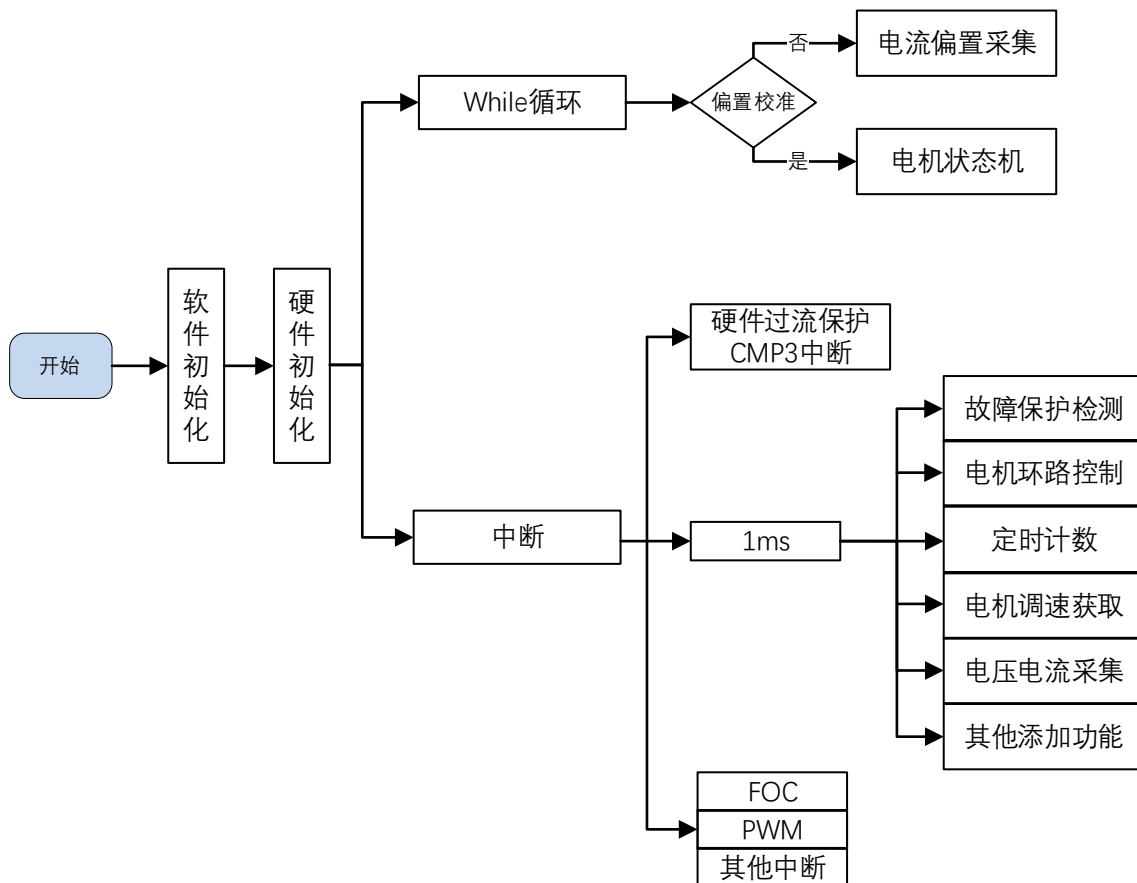


图 3-2 程序执行流程图

3.3 程序说明

3.3.1 Main 函数:

程序初始化->运放偏置电压检测 `GetCurrentOffset()` + 电机运行控制 `MC_Control()` + 1ms 定时部分处理程序 + 看门狗喂狗 `WatchDogRefresh()`。

1ms 定时序处理程序中包括速度、电压、电流等信号的采集 + 功率计算 + 调速 PWM 频率计算 `TargetRef_Process()` + 环路控制 `Speed_response()` + 保护功能 `Fault_Detection()` + led 灯闪烁控制 `Led_Display(mcFaultSource)`

3.3.2 1ms 定时中断

1ms 定时器中不做直接的程序处理，只有对标志位 `VariableFlag.g_1msTick` 置一的操作，在主函数中通过这个标志位来识别 1ms 时间的到来，做程序的处理。

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。内部集成低速估算器处理程序 `LowSpeedObserverFunction()` 以及在不开启低速估算器时的电流解耦功能的程序 `CurrentDecoupledControlFunction()`。

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 5.3.1](#)。

3.3.5 Timer3 中断

Timer3 中断主要是 PWM 信号频率的获取，通过该中断获取到 PWM 的周期值 `TIM3_ARR`，之后再通过计算出 PWM 的频率大小。

3.3.6 FO 中断

FO 中断。

4 调试步骤

4.1 电机参数配置

4.1.1 电机参数

1. 电机极对数 Pole_Pairs;
2. 电机的相电阻 RS、DQ 轴电感 LD、LQ(凸极性电机 LD、LQ 都填 D 轴电感值即可), 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 $R_s = RL/2$;
3. DQ 轴电感: 电桥测 1KHz 频率下, 手动转动电机转子, 测出线电感最小值和最大值, LD 为最小值的一半, LQ 为最大值的一半, 一般电机厂会提供 DQ 轴电感值, 程序里的 LD、LQ 都填电机厂提供的 D 轴电感值即可;
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:

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

其中, 1000 为 1000RPM 的反电动势, 2 为峰峰值转换为峰值, 1.732 为线电压转换为相电压, P 为电机极对数。

示例, 测量反电动势波形如下:

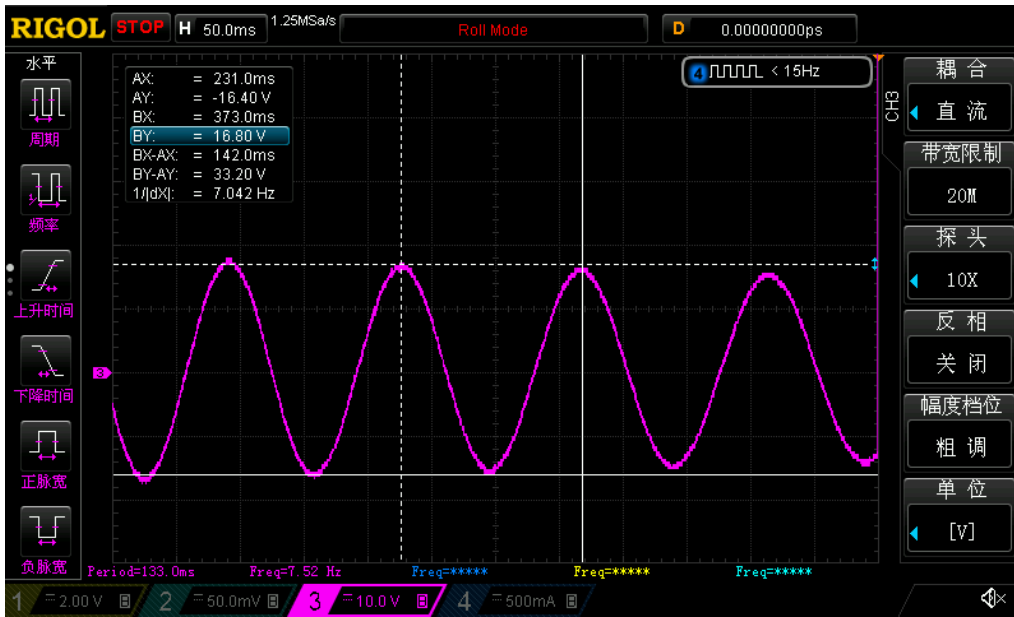


图 4-1 反电动势波形

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

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

- 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

```

/*-----
2.Motor Parameter
-----*/
#define Pole_Pairs      (3.0)          ///< Pole pairs
#define LD              (0.059)       ///< (H) d-Axis inductance
#define LQ              (LD)          ///< (H) q-Axis inductance
#define RS              (6.2)         ///< (Ω) Phase resistance
#define Ke              (45.25)       ///< (V/kRPM) Back EMF constant
#define Psi             (float)(Ke * 207.84 / 1000.0 / 2.0 / 1.732 / Pole_Pairs / _2PI)
#define MOTOR_SPEED_BASE (9000.0)    ///< (RPM) Motor speed base

```

4.2 芯片内部参数配置

```

/*-----
1.Motor PWM Parameter
-----*/
/*----- 硬件设置 -----*/
#define PWM_FREQUENCY   (5.0)         ///< (kHz) Motor PWM frequency
#define PWM_DEADTIME    (1.0)         ///< (us) Deadtime
#define MIN_WIND_TIME   (PWM_DEADTIME+3.0) ///< (us) Minimum sampling window of single-resistance current

```

注意事项:

- 载波频率一般需要设置为最大电周期 10 倍以上，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。考虑 MOS 温升，默认选用 5K 载波频率进行调试；

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 $1000/16/PWM_FREQUENCY > MIN_WIND_TIME > 2*PWM_DEADTIME$ ，为了满足在几米长(测功机测试存在该情况)的电机线上能采样到真实电流，一般采样窗口设置较大，因为长线下 MOS 开关震荡严重；
4. 正反转设置，根据实际接线设置，冰箱压缩机反转会使机械结构损坏，因此电机启动前需要配置正确的方向。

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。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)。

```

#define HW_ADC_VREF                (VREFS_0)                ///< (V) ADC voltage reference

/**
 * @brief Current sampling mode selection
 * @param (Single_Resistor) Single-resistance current sampling mode
 * @param (Double_Resistor) Double-resistance current sampling mode
 * @param (Three_Resistor) Triple-resistance current sampling mode
 */
#define Shunt_Resistor_Mode        (Single_Resistor)

#define HW_RSHUNT                   (0.1)                  ///< ((Ω) Current sampling resistance

/**
 * @brief AMP mode selection
 * @param (AMP_NOMAL) External AMP mode
 * @param (AMP_PGA_DUAL) Internal PGA differential dual input mode (connected with a 1kΩ resistor in the external circuit)
 * @param (AMP_PGA_SINGLE) Internal PGA differential single input mode (connected with a 1kΩ resistor in the external circuit)
 */
#define HW_AMP_MODE                 (AMP_NOMAL)            ///< AMP mode selection

/**
 * @brief AMP amplification gain
 * @param (AMP2x) Available in AMP_PGA_SINGLE mode. The amplification gain is 7/3
 * @param (AMP4x) Available in AMP_PGA_SINGLE mode. The amplification gain is 24/5
 * @param (AMP8x) Available in AMP_PGA_SINGLE mode. The amplification gain is 88/9
 * @param (AMP16x) Available in AMP_PGA_SINGLE mode. The amplification gain is 336/17
 * @param (xxxx.x) Available in AMP_NOMAL mode. The amplification gain is selected based on the external circuit
 */
#define HW_AMPGAIN                  (3.75)                ///< AMP amplification gain selection

#define VREF_OUT_EN                 (Disable)

/**
 * @brief VHALF output enable
 * @param (Disable)
 * @param (Enable)
 */
#define VHALF_OUT_EN                (Disable)

/*-----
 * 4. Bus Voltage Sampling Parameter
 *-----*/
#define RV1                          (3200.0)            ///< (kΩ) kΩ Bus voltage divider resistor1
#define RV2                          (10.0)              ///< (kΩ) Bus voltage divider resistor2
    
```

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 5.3.2](#);

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)。

```

6 /* Faults processing Enable */
7 #define OC_SW_ProtectEn            (1)                  ///< Soft over current protection. 0 is disable, 1 is enable
8 #define OV_ProtectEn              (1)                  ///< Bus voltage protection. 0 is disable, 1 is enable.
9 #define LP_ProtectEn              (1)                  ///< Phase loss protection. 0 is disable, 1 is enable
10 #define OP_ProtectEn              (1)                  ///< Over power protection. 0 is disable, 1 is enable
11 #define OT_ProtectEn              (0)                  ///< Temperature protection. 0 is disable, 1 is enable
12 #define Stall_ProtectEn           (1)                  ///< Lock protection. 0 is disable, 1 is enable
13 #define PWSL_ProtectEn            (1)                  ///< Lock protection. 0 is disable, 1 is enable
14
    
```

图 4-2 保护功能使能


```

26 /* Protection recover parameters */
27 #define OC_RecoveryTimes          (5)                ///< Recover times. When the set value < 255,
28 #define OC_RecoveryDelayTime     (300)             ///< (s) Delay time of recover
29
30 #define OV_RecoveryTimes          (5)                ///< Recover times. When the set value < 255,
31 #define OV_RecoveryDelayTime     (300)             ///< (s) Delay time of recover
32
33 #define LP_RecoveryTimes          (5)                ///< Recover times. When the set value < 255,
34 #define LP_RecoveryDelayTime     (300)             ///< (s) Delay time of recover
35
36 #define OP_RecoveryTimes          (5)                ///< Recover times. When the set value < 255,
37 #define OP_RecoveryDelayTimes    (300)             ///< (s) Delay time of recover
38
39 #define OT_RecoveryTimes          (5)                ///< Recover times. When the set value < 255,
40 #define OT_RecoveryDelayTime     (300)             ///< (s) Delay time of recover
41
42 #define Stall_RecoveryTimes       (5)                ///< Recover times. When the set value < 255,
43 #define Stall_RecoveryDelayTime   (300)             ///< (s) Delay time of recover
44
45 #define PWMLOSS_RecoveryEn        (1)                ///< Recover times. When the set value < 255,
    
```

图 4-3 保护恢复次数和间隔时间设置

```

58 #define COMPARE_MODE               (COMPARE_DAC_MODE)    ///< Comparison value source of hardware overcurrent protection
59 #define HNOCVValue                 (4.5)                ///< (A) Overcurrent threshold in DAC compare mode. Imax = VHALF / H
60
61 /* -----Software overcurrent protection----- */
62 #define SW_OC_CurrentVal           I_Value(3.0)         ///< (A) Software overcurrent threshold
63 #define SW_OC_DectTime            (30)                 ///< (ms) Software overcurrent detection times
64
65 /* -----Overvoltage/Undervoltage protection----- */
66 #define OVER_VOLTAGE_DECTIME       (300)               ///< (ms) (V) Bus voltage overvoltage detection times
67 #define UNDER_VOLTAGE_DECTIME     (300)               ///< (ms) (V) Bus voltage undervoltage detection times
68
69 #define OVER_VOLTAGE_PROTECT       UDC_Value(380)     ///< (V) Bus voltage overvoltage threshold
70 #define UNDER_VOLTAGE_PROTECT     UDC_Value(200)     ///< (V) Bus voltage undervoltage threshold
71
72 #define OVER_VOLTAGE_RECOVER       UDC_Value(365)     ///< (V) Bus voltage overvoltage recover value
73 #define UNDER_VOLTAGE_RECOVER     UDC_Value(220)     ///< (V) Bus voltage undervoltage recover value
74
75 /* -----Lock protection----- */
76 #define STALL_SPEED_MAX            S_Value(6000)       ///< (RPM) Maximum lock protection speed
77 #define STALL_SPEED_MIN           S_Value(900)        ///< (RPM) Minimum lock protection speed
78
79 #define EsThresholdValueL          (70.0)             ///< BEMF thresholdL
80 #define EsThresholdValueH          (1000.0)           ///< BEMF thresholdH
81
82 #define EsThresholdSpeed           S_Value(1200)      ///< (RPM) (RPM) Lock protection detection speed. This value is work
83
84 /* -----Phase loss protection----- */
85 #define LP_NoLoadCurrentValue      I_Value(0.1)        ///< (A) Phase loss protection current value
86 #define LP_DectDealyTIME           (500)              ///< (ms) Delay time of detection
87 #define LP_DectCycleTIME           (50)               ///< Current multiple of Phase loss protection
88
89 /* -----Overspeed protection----- */
90 #define OVER_POWER_VALUE           P_Value(270.0)     ///< (W) Overpower protection threshold
91 #define OVER_POWER_DECTIME         (200)              ///< (ms) Overpower protection detection times
92
93 /* -----PWM loss protection----- */
94 #define PWM_LOSS_DECTIME           (60000)            ///< (ms) PWM loss protection detection times
95
96 #define TempProtValue              90                  ///< Unit: 1 degree centigrade
97 #define TempWarningOn              88                 ///< Unit: 1 degree centigrade
98 #define TempWarningOff             80                 ///< Unit: 1 degree centigrade
99 #define TempRecoverValue           80                 ///< Unit: 1 degree centigrade
    
```

图 4-4 保护功能参数

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](#)。

冰箱压缩机一般要求一定的背压(即吸气口和排气口之间的压差)启动能力，不同压缩机厂要求的背压启动能力不一样，需要在启动测试仪上进行调试，将吸气压力和排气压力调整到客户要求的压力进行调试。

1. 预定位电流: ID_Align_CURRENT、IQ_Align_CURRENT_START 一般固定设置为 0，IQ_Align_CURRENT_END 一般设置不超过 2A，具体大小要根据定位效果调整，有的冰箱压缩机在 2A 的定位电流下可能抖动过大，抖动严重时可能导致撞缸；为了减小定位抖动，定位第一阶段定位电流由 IQ_Align_CURRENT_START 逐渐增加到 IQ_Align_CURRENT_END，定位角度由 Align_Angle1 逐渐增

加 Align_Angle2: 定位第二阶段为保持定位角度 Align_Angle2 和定位电流 IQ_Align_CURRENT_END 一段时间; 定位角度变化是为了防止定位角度和转子位置相差接近 180 度时拉不动转子, 提高定位可靠性;

2. 在第二次定位完成之后会定位到 0 度, 后续的部分会完成电机的电阻参数识别;

```

43  /*
44  -----
45  3. Motor Alignment Parameter
46  ----- */
47  #define AlignmentRampTime      (500.0)          ///< (ms) Alignment, Iq
48  #define AlignmentHoldTime1     (500.0)          ///< (ms) Alignment current hold time1->to detect phase loss
49  #define AlignmentHoldTime2     (500.0)          ///< (ms) Alignment current hold time2->to detect phase loss
50  #define AlignmentHoldTime3     (1000.0)         ///< (ms) Alignment current hold time3->to detect Rs
51  #define AlignmentHoldTime4     (1000.0)         ///< (ms) Alignment current hold time4->to detect Rs
52  #define Align_Time             (AlignmentRampTime + AlignmentHoldTime1 + AlignmentHoldTime2 + AlignmentHoldTime3 + AlignmentHoldTime4)
53  #define Align_Angle1          (-90.0)          ///< (°) first Alignment electrical angle
54  #define Align_Angle2          (-30.0)          ///< (°) second Alignment electrical angle
55  #define Align_Angle3          (0.0)           ///< (°) last Alignment electrical angle
56
57  #define DKP_Alignment          _Q12(2.0)       ///< D-axis current loop Kp during alignment
58  #define DKI_Alignment          _Q15(0.05)      ///< D-axis current loop Kp during alignment
59  #define QKP_Alignment          _Q12(2.0)       ///< Q-axis current loop Kp during alignment
60  #define QKI_Alignment          _Q15(0.05)      ///< Q-axis current loop Kp during alignment
61
62  #define ID_Align_CURRENT       I_Value(0.0)    ///< (A) D-axis Align current
63  #define IQ_Align_CURRENT_Start I_Value(0.0)    ///< (A) D-axis initial Align current
64  #define IQ_Align_CURRENT_End   I_Value(2.0)    ///< (A) D-axis hold Align current
    
```

图 4-5 预定位参数

3. 启动电流: 一般 ID_Start_CURRENT 固定设置为 0, IQ_Start_CURRENT 一般固定为 2A

```

/* -----Startup current----- */
#define ID_Start_CURRENT      I_Value(0.0)      ///< (A) d-Axis startup current
#define IQ_Start_CURRENT      I_Value(2.0)      ///< (A) q-Axis startup current
    
```

图 4-6 启动电流

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, 不能过大否则启动转速过冲过大, 必须小于退磁电流否则导致电机退磁, 启动电流过大也容易导致启动过程撞缸;

4. ATO, 程序使用固定的 ATO。ATO 是估算器后级的锁相环的响应 PI 参数, 填入的数值的物理意义是 Hz, 通过频率计算出带宽的 PI 参数;

```

197 /* -----ATO bandwidth----- */
198 #define ATO_BW              (150.0)           ///< (Hz) ATO bandwidth, the recommended value is 1.0 ~ 2
199
200 /* -----Speed filter bandwidth----- */
201 #define SPD_BW              (15.0)            ///< (Hz) Speed filter bandwidth, the recommended value is
202
    
```

图 4-7 估算器带宽滤波

注意事项:

对于冰箱整机而言启动一般分为两种情况, 即背压启动和平衡压启动。

背压启动: 即排气和吸气有一定的压差, 此时电机启动的时候负载比较大, 成功启动之后随着吸气压力的很快降低到 0 负载会很快缩小, 功率达到稳定。这里启动的难点在于启动的前几个周期。如果 ATO_BW 过大有可能会出现在低速估算器切入硬件估算器的瞬间出现失步。

平衡压启动: 即压缩机的吸气口与排气口的气压值是相等的情况。此时启动的前几个周期的负载是比较小的。启动成功之后随着吸气压力的降低和排气压力的升高, 负载会逐渐增大, 当吸气压力接近于 0 的时候负

载才会逐渐减小直到稳定。并且平衡时的气压越大最大负载也会越大，因此要在有限的电流条件下完成启动就需要估算器的响应要尽可能地快，保证驱动角度的跟随性好。如果 ATO_BW 过小就会导致不少电流没有做有用功，出现还没有到达平衡点就因为负载过大报功率保护或者压机堵转的情况。

5. 速度带宽滤波值 SPD_BW:

```
/* -----Speed filter bandwidth----- */
#define SPD_BW (15.0) ///< (Hz) S1
```

注意事项:

SPD_BW 一般不需要调整。

6. 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:

```
0  /* -----Startup Parameter----- */
1  4. Startup Parameter
2  /*
3  @brief Startup mode selection
4  @param (Open_Start) Open loop Startup
5  @param (Omega_Start) Omega startup
6  @param (Open_Omega_Start) Open loop Startup switch to Omega startup
7  */
8  #define Open_Start_Mode (Omega_Start)
9
10 /* -----OMEGA startup parameter----- */
11 #define MOTOR_OMEGA_RAMP_ACC (5) ///< Forced speed increment step in standstillstate
12 #define MOTOR_OMEGA_RAMP_MIN S_Value(30.0) ///< (RPM) The minimum speed to switch form forced speed to estimated speed
13 #define MOTOR_OMEGA_RAMP_END S_Value(30.0) ///< (RPM) The maximum speed of forced speed
14
```

图 4-8 Omega 启动参数

注意事项:

- 1) Motor_Omega_Ramp_ACC 参考值范围 100 ~ 500;
- 2) MOTOR_OMEGA_ACC_MIN 参考值范围 100 ~ 500;
- 3) MOTOR_OMEGA_ACC_END 参考值范围 200 ~ 800;
- 4) MOTOR_LOOP_RPM 需要大于 MOTOR_OMEGA_ACC_END，参考值范围 800 ~ 1200。

7. 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:

```
206 /* -----Current loop coefficient at startup state----- */
207 #define DKPStart _Q12(2.5) ///< d-Axis KP of current loop in startup state
208 #define DKISStart _Q15(0.02) ///< d-Axis KI of current loop in startup state
209
210 #define QKPStart _Q12(2.5) ///< q-Axis KP of current loop in startup state
211 #define QKISStart _Q15(0.02) ///< q-Axis KI of current loop in startup state
212
213 /* -----Current loop coefficient after controller switch to model----- */
214 #define DKP _Q12(2.5) ///< d-Axis KP of current loop after controller switch to model
215 #define DKI _Q15(0.05) ///< d-Axis KI of current loop after controller switch to model
216 #define QKP _Q12(2.5) ///< q-Axis KP of current loop after controller switch to model
217 #define QKI _Q15(0.05) ///< q-Axis KI of current loop after controller switch to model
218
```

图 4-9 电流环参数

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 1.0 ~ 5.0;
- 4) DQKI 建议范围 0.05 ~ 0.5。

8. 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。

```

219 /* -----d-Axis voltage reference limit value----- */
220 #define DOUTMAX          _Q15(0.99)          ///< d-Axis voltage reference maximum value. The recommended value is _Q15(0.99)
221 #define DOUTMIN          _Q15(-0.99)         ///< d-Axis voltage reference minimum value. The recommended value is _Q15(-0.99)
222
223 /* -----q-Axis voltage reference limit value----- */
224 #define QOUTMAX          _Q15(0.99)          ///< q-Axis voltage reference maximum value. The recommended value is _Q15(0.99)
225 #define QOUTMIN          _Q15(-0.99)         ///< q-Axis voltage reference minimum value. The recommended value is _Q15(-0.99)
226
    
```

图 4-10 DQ 轴电流环输出限幅值

注意事项:

- 1) FOC__UQ 反馈电机输出是否饱和;
- 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。

4.6 常见问题及解决办法:

1. 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。

2. 相电流波形大小波过大(由冰箱压缩机结构决定, 对于两对极的冰箱压缩机, 一个机械周期有两个幅值不同的电流波形; 对于三对极的冰箱压缩机, 一个机械周期有三个幅值不同的电流波形);

解决: 调整电流环 PI(即 DQKP, DQKI)的值, 电流环 PI 对电流波形的稳定性影响比较大, 电流环带宽越小, 大小波越明显; 电流环带宽越大, 大小波越不明显。

4.7 速度环调试

1. 一般冰箱压缩机都是用的速度环, 转速范围一般为 1200RPM ~ 4500RPM;
2. 设置速度环输出的最大值, 即 SOUTMAX_RUN 的值, 这个值根据最大负载和电机退磁电流而定, 冰箱压缩机最大负载情况为在 43 摄氏度环温下, 冰箱门全开, 等系统平衡后启动, 该限幅值需要保证压缩机能够成功启动;
3. 不同的压机厂家要求不同的压机的升降速的加速度。

```

228 /* -----S. Loop Control Parameter----- */
229
230 #define LOOP_TIME        (1)                ///< (ms) Outer loop control period
231
232 /* -----Outer loop coefficient----- */
233 #define SKP              _Q12(2.0)          ///< KP of Outer loop
234 #define SKI              _Q15(0.01)         ///< KI of Outer loop
235
236 /* -----Limit value of output----- */
237 #define SOUTMAX          I_Value(2.3)       ///< (A) Outer loop output maximum value
238 #define SOUTMIN          I_Value(0.0)       ///< (A) Outer loop output minimum value
239
240 /* -----Maximum and minimum value of speed regulation curve in current control mode----- */
241 #define MOTOR_SPEED_MAX_RPM S_Value(4500.0) ///< (RPM) Maximum operating speed
242 #define MOTOR_SPEED_MIN_RPM S_Value(1200.0) ///< (RPM) Minimum operating speed
243
244 /* -----Ramping step value----- */
245 #define RAMP_INC_START    SAcc_Value(180.0) ///< (RPS) Increment step value in current control mode
246 #define RAMP_DEC_START    SAcc_Value(180.0) ///< (RPS) Decrement step value in current control mode
247
248 #define RAMP_INC_RUN      SAcc_Value(60.0)  ///< (RPS) Increment step value in speed control mode
249 #define RAMP_DEC_RUN      SAcc_Value(60.0)  ///< (RPS) Decrement step value in speed control mode
250
    
```

图 4-11 速度环运行参数

4.8 增加 CLOCK 调速功能

1. 一般冰箱压缩机都是 CLOCK 调速，步骤为:

- 1) 将调速模式修改为 PWMMODE 调速。一般冰箱设定的转速为调频信号频率的 30 倍。例如，输入 Clock 信号 40Hz，即为设定转速 $40 \times 30 = 1200\text{RPM}$ 。要求输入频率小于等于 30Hz 关机，大于等于 40Hz 开机。具体的开机关机频率阈值可以根据客户的要求制定，但一定是在 30 ~ 40Hz 之间。

```

255 白 /**
256  * Speed Control mode select
257  * @param (PWMMODE)      PWM fre mode
258  * @param (NONEMODE)     Set a instant speed
259  * @param (ONOFFTEST)    On/Off Test
260  */
261  #define REF_MODE          (PWMMODE)
262
263  /* -----On/Off setting----- */
264  #define OFFPWMFreq        (3500)          ///< (0.01Hz) Off PWM frequency. When the input frequency is
265  #define OFFPWMFreqHigh   (18000)         ///< (0.01Hz) Off PWM high frequency. When the input frequen
266  #define ONPWMFreq        (3600)          ///< (0.01Hz) On PWM frequency. When the input frequency is :
267  #define MINPWMFreq       (4000)         ///< (0.01Hz) Minimum frequency in speed regulation curve
268  #define MAXPWMFreq       (15000)        ///< (0.01Hz) Maximum frequency in speed regulation curve
269  #define SpeedFreqK       _Q15(30.0 / 100)/MOTOR_SPEED_BASE
    
```

图 4-12 PWMMODE 调速参数

输入频率 Input Frequency fn[Hz]	电机转速 Motor Speed n[RPM]	备注 Remarks
$fn < 30$	-	停机/不启动 Stop/Not Start
$fn = 30 \sim 40$	$n = N_{min}$	-
$fn = 40 \sim 150$	$n = 30 \times fn$	-
$fn = 150 \sim 200$	$n = N_{max}$	-
$fn > 200$	$n = 0$	停机 Stop

其中 N_{min} 为 1200RPM， N_{max} 为 4500RPM。

注意事项:

- 1) PWM 的输入频率是比较低的，捕获定时器要设置为最大分频比；
 - 2) 输入的频率低于 3Hz 时，捕获定时器不能够捕获得到，计数会溢出，此时需要通过 1ms 中断程序计时，算出 3Hz 以下的频率；
 - 3) 开关机频率，要留有一定的滞回区间，如 35Hz 关机，36Hz 关机。预留至少 1Hz 的滞回区间；
 - 4) 为了保证 PWM 频率的精度，频率计算保留小数点后两位，按照 100 倍取值；
 - 5) PWM 的频率也有可能会有短暂的波动，因此要设置 1s 钟的判断滤波。
2. 压机因为机械结构特点，需要在机体里面加入润滑油保证机械性能，并且是通过电机转动将壳底的润滑油喷射到各个机械结构件。因此在压缩机启动之后会先维持一段时间的上油转速然后再切回正常的控制转速。上油转速由压缩机厂家给出。

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

冰箱压缩机选择 Omega 启动，程序对应默认即是该启动方式。

```

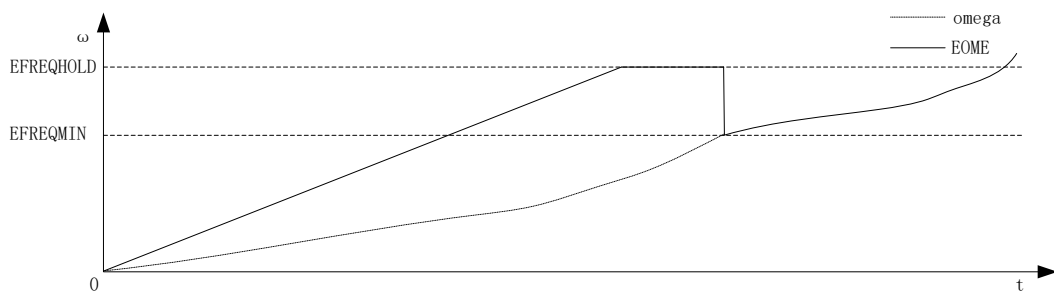
3  /**
4   * @brief Startup mode selection
5   * @param (Open_Start)      Open loop Startup
6   * @param (Omega_Start)    Omega startup
7   * @param (Open_Omega_Start) Open loop Startup switch to Omega startup
8   */
9  #define Open_Start_Mode      (Omega_Start)
    
```

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。

```

179 #define Open_Start_Mode      (Omega_Start)
180
181 /* -----OMEGA startup parameter----- */
182 #define MOTOR_OMEGA_RAMP_ACC      (5)          ///< Forced speed increment step in standstillstate
183 #define MOTOR_OMEGA_RAMP_MIN      S_Value(30.0)  ///< (RPM) The minimum speed to switch form forced speed to es
184 #define MOTOR_OMEGA_RAMP_END      S_Value(30.0)  ///< (RPM) The maximum speed of forced speed
185
    
```

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. 低速估算器的 LowSpeedObserverGain2 增益过小； 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。 5. ATO_BW 的值过大，导致切入硬件估算器的时候自激振荡；
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； 2. Omega 启动参数也会有影响。 3. 低速估算器的 LowSpeedObserverGain2 增益过小；
启动正序旋转一定角度后卡顿一下再正常旋转	1. 此时可能是 omega 启动的 MOTOR_OMEGA_RAMP_MIN 和 MOTOR_OMEGA_RAMP_END 过大，导致前面强拖的效果不好； 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 弱磁控制

冰箱压缩机一般需要进行弱磁扩速，当 DQ 轴合成电压矢量幅值达到设定值 MotorFiledWeakenUs 时进入弱磁控制状态，弱磁角度通过电压闭环得到，电压环的 PI 参数通过 MotorFiledWeakenKp 和 MotorFiledWeakenKi 进行修改。

```

0
1  /*Filed weaken parameters*/
2  #define MotorFiledWeakenEn          (Enable)
3  #define MotorFiledWeakenUs          _Q15(0.9)                // The threshold voltage duty
4
5  #define MotorFiledWeakenKp          _Q12(0.5)                // Kp of filed weaken loop
6  #define MotorFiledWeakenKi          _Q15(0.005)              // Ki of filed weaken loop
7
    
```

5.3 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。

5.3.1 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。

```

9  /* -----hardware overcurrent protection----- */
10 /*
11  * @brief Overcurrent protection mode
12  * @param (HW_OC_FO_MODE)      IFM or HVIC FO single mode
13  * @param (HW_OC_CMP_MODE)     CMP mode
14  * @param (HW_OC_FOCMP_MODE)   IFM or HVIC FO single mode and CMP mode
15  * @param (HW_OC_DISABLE)     No hardware overcurrent protection
16  */
17 #define HW_OC_MODE                (HW_OC_FOCMP_MODE)
18
19 #define COMPARE_MODE                (COMPARE_DAC_MODE)          ///< Comparison value source of hardware overcur
20 #define HWOCValue                  (4.5)                       ///< (A) Overcurrent threshold in DAC compare mo
    
```

2. 软件过流保护。

程序通过计算检测到的 alpha 轴电流和 beta 轴电流的幅值的平方和开方得到相电流的幅值。如果相电流的幅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。

```

60 /* -----Software overcurrent protection----- */
61 #define SW_OC_CurrentVal          I_Value(3.0)                ///< (A) Software overcurrent threshold
62 #define SW_OC_DectTime            (30)                       ///< (ms)Software overcurrent detection times
    
```

5.3.2 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持续OVER_VOLTAGE_DECTTIME(ms)时，则报过压保护；欠压保护同理。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。

```

65 /* -----Overvoltage/Undervoltage protection----- */
66 #define OVER_VOLTAGE_DECTTIME    (300)                       ///< (ms) (V) Bus voltage overvoltage detection times
67 #define UNDER_VOLTAGE_DECTTIME  (300)                       ///< (ms) (V) Bus voltage overvoltage detection times
68
69 #define OVER_VOLTAGE_PROTECT     UDC_Value(380)              ///< (V) Bus voltage overvoltage threshold
70 #define UNDER_VOLTAGE_PROTECT   UDC_Value(200)              ///< (V) Bus voltage undervoltage threshold
71
72 #define OVER_VOLTAGE_RECOVER     UDC_Value(365)              ///< (V) Bus voltage overvoltage recover value
73 #define UNDER_VOLTAGE_RECOVER   UDC_Value(220)              ///< (V) Bus voltage undervoltage recover value
74
    
```

5.3.3 缺相保护

通过积分的方式，在一定时间内通过比较电流累计值的大小去判断缺相。

```

14 /* -----Phase loss protection----- */
15 #define LP_NoLoadCurrentValue    I_Value(0.1)                ///< (A) Phase loss protection current value
16 #define LP_DectDealyTIME         (500)                       ///< (ms) Delay time of detection
17 #define LP_DectCycleTIME         (50)                        ///< Current multiple of Phase loss protection
    
```


5.3.4 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: 在当开机延时500ms后
 - 1) 判断反电动势的值持续3000ms小于设定值EsThresholdValueL;
 - 2) 当转速大于EsThresholdSpeed的同事反电动势的幅值小于EsThresholdValueH持续3000ms;
2. 通过检测估算转速, 当估算转速超过设定转速 STALL_SPEED_MAX, 或者低于设定转速 STALL_SPEED_MIN, 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3000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。

```

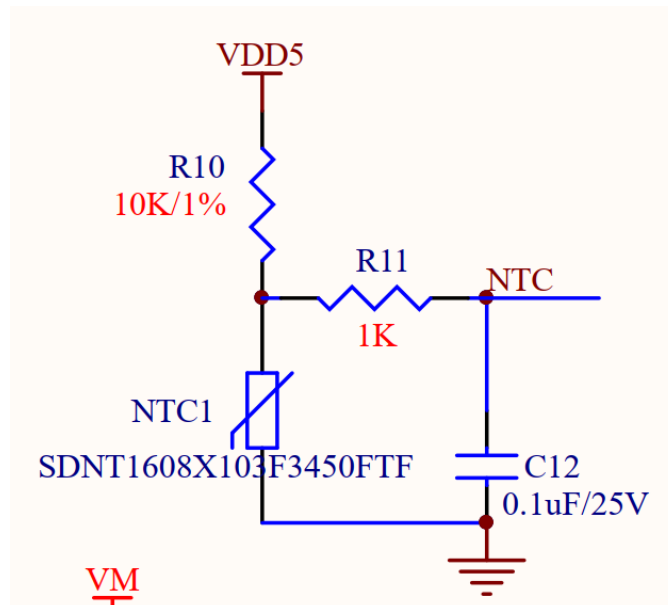
#define STALL_SPEED_MAX          S_Value(6000)          ///< (RPM) Maximum lock protection speed
#define STALL_SPEED_MIN         S_Value(900)           ///< (RPM) Minimum lock protection speed

#define EsThresholdValueL       (70.0)                ///< BEMF thresholdL
#define EsThresholdValueH       (1000.0)              ///< BEMF thresholdH

#define EsThresholdSpeed        S_Value(1200)         ///< (RPM) (RPM) Lock protection detection speed. This value is w
    
```

5.3.5 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。



```

96 #define TempProtValue          90          ///< Unit: 1 degree centigrade
97 #define TempWarningOn        88          ///< Unit: 1 degree centigrade
98 #define TempWarningOff       80          ///< Unit: 1 degree centigrade
99 #define TempRecoverValue      80          ///< Unit: 1 degree centigrade

```

其中，

- TempProtValue为保护设定值；
- TempWarningOn为警告设定值；
- TempWarningOff 为警告关闭设定值；
- TempRecoverValue 为保护关闭设定值；

5.3.6 过功率保护

在制冷器具启动阶段，由于负载较大易出现功率过高而由于功率保护停机的情况，为了避免过功率停机只能设置较大过功率保护值，对变频器及压缩机的损害较大，鉴于此，对于新开发产品的程序均采用降频处理。具体操作如下：

1. 功率保护检出时间: $\leq 10s$;
2. 降频处理转速范围: 2000rpm ~ 4500rpm, ($< 2000rpm$ 过功率直接停机);
3. 在当前转速点运行时，若功率超过功率限值时，转速降 5Hz(300rpm)，若功率值仍然超过当前转速所对应的功率限值，转速再降 5Hz(300rpm)，依此类推，降频到的最低转速为 2000rpm;
4. 降频后的持续检测运行功率，持续功率不超标运行 5min 后，尝试按降频策略逐级升频回到目标转速。

降频功率保护整定值的选取一般为最大功率的 1.3 倍

由于负载或压缩机本身故障引起功率超过过功率保护整定值 Pov 时，变频器停止输出。该保护检出时间为 $\leq 10s$ 。

功率保护停机后，等待 5min 后尝试启动，若启动失败按照堵转或大力矩启动策略处理。

停机功率保护整定值的选取一般为最大功率的 1.5 倍。

注: 具体数值需要参考电机匹配参数表。

```

89 /* -----Overspeed protection----- */
90 #define OVER_POWER_VALUE      P_Value(270.0)  ///< (W) Overpower protection threshold
91 #define OVER_POWER_DECTIME    (200)          ///< (ms) Overpower protection detection times

```

6 修改记录

版本号	修改详细内容说明	生效日期	修订者
V1.0.00	初稿	2022/12/16	衡龙腾
V1.0	修订格式问题	2023/07/06	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

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。