

低压压缩机调试手册

三相电机控制 MCU FU6861Q2

峰昭科技(深圳)股份有限公司

目录

1 概述	4
2 硬件原理与参数配置	5
2.1 硬件原理	5
2.1.1 电源部分	6
2.1.2 芯片主体	6
2.1.3 功率驱动部分.....	7
2.1.4 运放配置电路.....	7
2.1.5 母线电压采样.....	8
2.1.6 可选电路一：温度检测电路.....	8
2.1.7 可选电路二：可调电阻的速度控制电路	8
2.1.8 可选电路三：外接电压的速度控制电路	9
2.1.9 未使用电路一：HALL 检测电路	9
2.1.10 未使用电路二：顺逆风检测电路.....	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	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 外部中断	12
4 调试步骤	13
4.1 配置电机参数	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 确认芯片内部相关数据配置	14
4.3 确认硬件参数	15
4.4 保护参数设置	15
4.5 启动说明	16
4.6 低速观测器启动模式.....	18

4.7 OMEGA 启动模式.....	20
5 保护介绍	21
5.1 保护介绍	21
5.1.1 过流保护	21
5.1.2 电压保护	21
5.1.3 缺相保护	22
5.1.4 堵转保护	22
5.1.5 其他保护	23
6 方案调试难点&解决方法	24
7 修改记录	25

1 概述

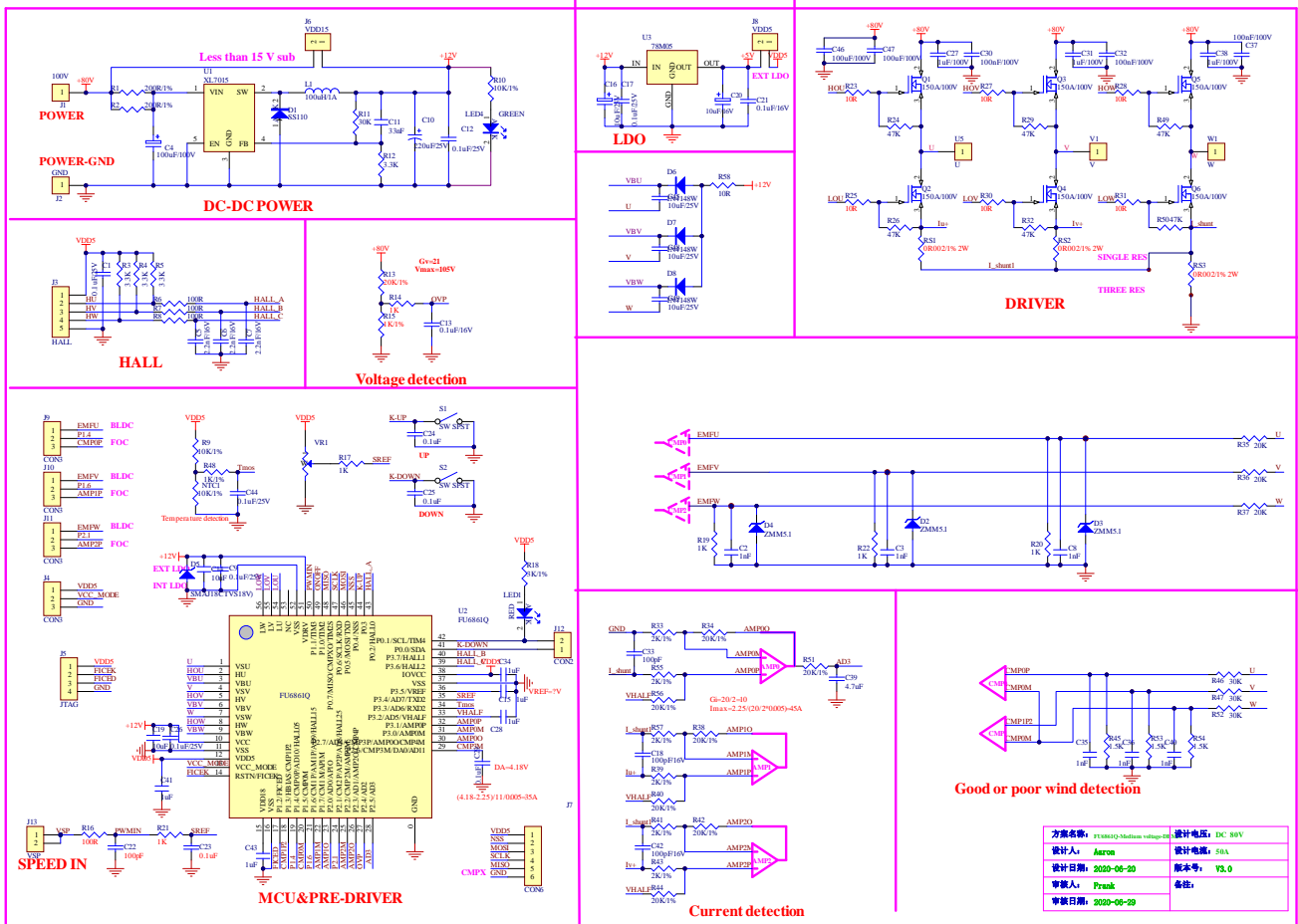
本调试手册详细介绍了如何使用峰昭科技的 FU6861Q2 芯片，对低压压缩机电机实现无霍尔的 FOC 驱动控制。阅读手册时，第二章硬件原理跟第三章软件原理可以大致先浏览一遍，重点放在第四章调试步骤。

涉及的软/硬件

软/硬件和模块	名称	章节	备注
软件	FU6861Q2 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		全部	调试需在该硬件上进行

2 硬件原理与参数配置

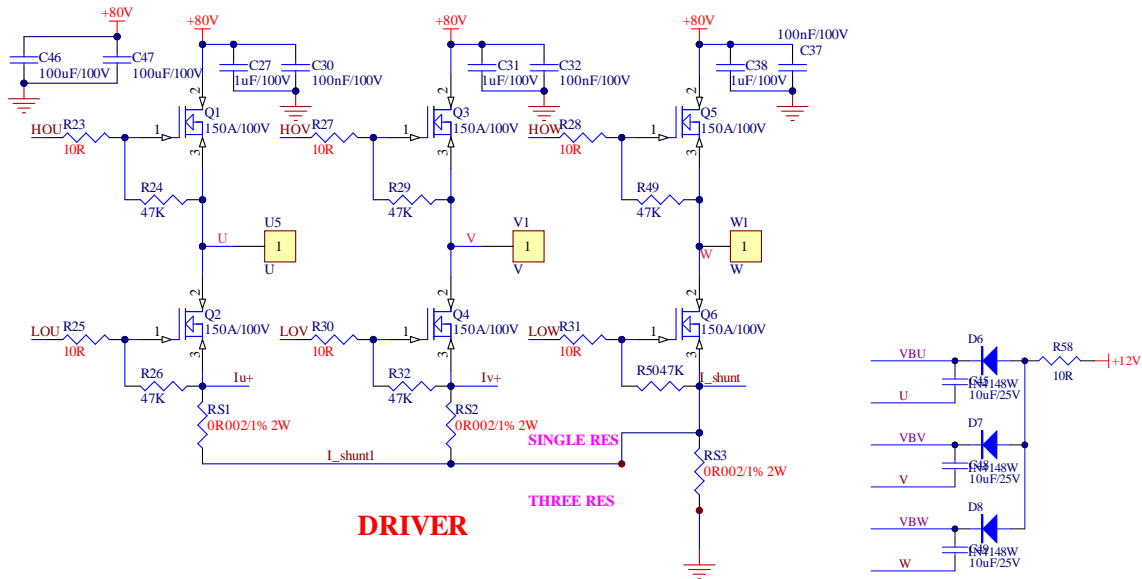
2.1 硬件原理



使用方式: 该板子作为中压 8-80V 的 6861Q2-DEMO 板, 可以作为 12-24V 的压缩机的驱动板, 因为压缩机一般采用单电阻采样的模式, 无需处理 J8/J10/J11 处的接线帽, 直接上电即可使用。

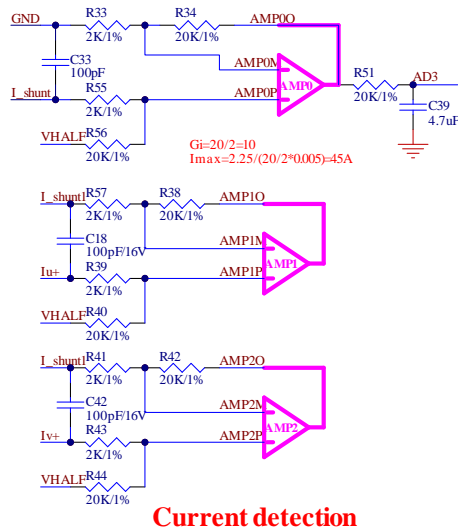
注意事项: 根据具体电机电压和电流大小, 合理配置母线电压比, 运放放大倍数, 采样电阻, 反电势检测电路分压比。

2.1.3 功率驱动部分



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%。

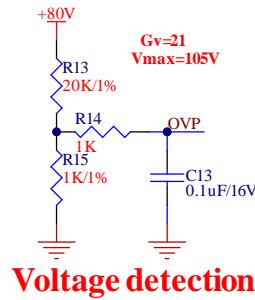
2.1.4 运放配置电路



注意事项:

1. C33、C18、C42 参数不可调整, 精度要求 10%;
2. 采用单电阻采样的方式, 使用“AMP0”的运放电路有效。放大倍数 = $R34/R33 = R56/R55 = 20K/2K = 10$, 放大倍数可以根据需要修改, 需要保证 R34、R56 电阻值相等、R33、R55 电阻值相等;
3. 最大采样电流 = $(VREF/2)/放大倍数/采样电阻值$ 。

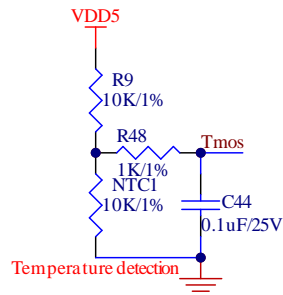
2.1.5 母线电压采样



注意事项:

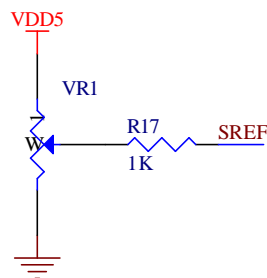
1. R14、C13 参数不可调整;
2. R13、R15 需要用 1%精度电阻;
3. 最大采样电压 = $(R13 + R15)/(R15)*VREF$;
4. 最大采样电压一般选择为 2 倍的最大应用电压, OVP 此处的电压需要低于 $0.8*VREF$ 。

2.1.6 可选电路一: 温度检测电路



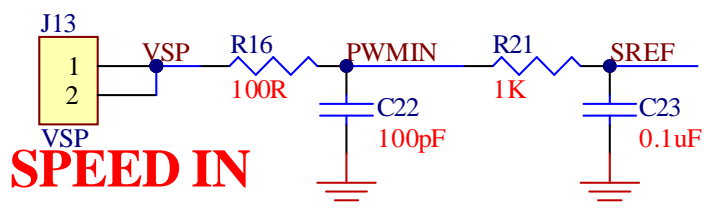
功能说明: NTC1 为热敏电阻, Tmos 端采集对应 AD 值。芯片采集 NTC 的电压, 计算对应的温度值, 进行相关控制。

2.1.7 可选电路二: 可调电阻的速度控制电路



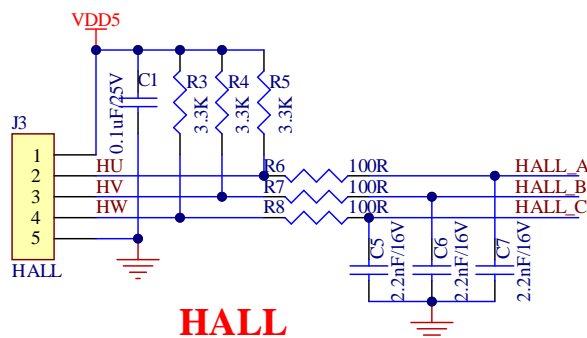
功能说明: 该电路通过可调电阻大小控制 SREF 的电压, 芯片采集电压进行相关控制, 可用于速度控制。

2.1.8 可选电路三：外接电压的速度控制电路



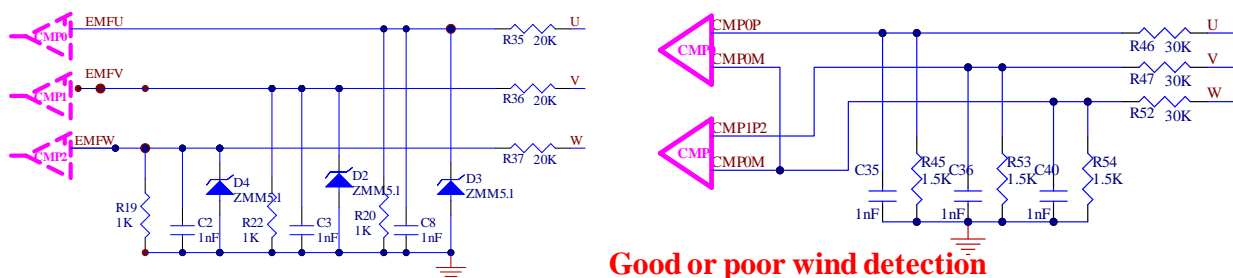
功能说明：该电路可在 VSP 处接入大小不超过基准电压值的电压，控制 SREF 口的电压。芯片采集电压进行相关控制，可用于速度控制。

2.1.9 未使用电路一：HALL 检测电路



功能说明：该电路为 6861Q2DEMO 板自带的电路，用于 HALL 检测，低压压缩机方案未使用该电路。

2.1.10 未使用电路二：顺逆风检测电路



功能说明：该电路为 6861Q2DEMO 板自带的电路，用于顺逆风检测，低压压缩机方案未使用该电路。

则转入下一个状态 mcRun。电机启动过程由 ME 内核实现：

5. 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；
6. mcStop: 停机状态，该状态用于停机操作，当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；
7. 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: 电机状态机一共分为 7 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。

3.2 程序流程图

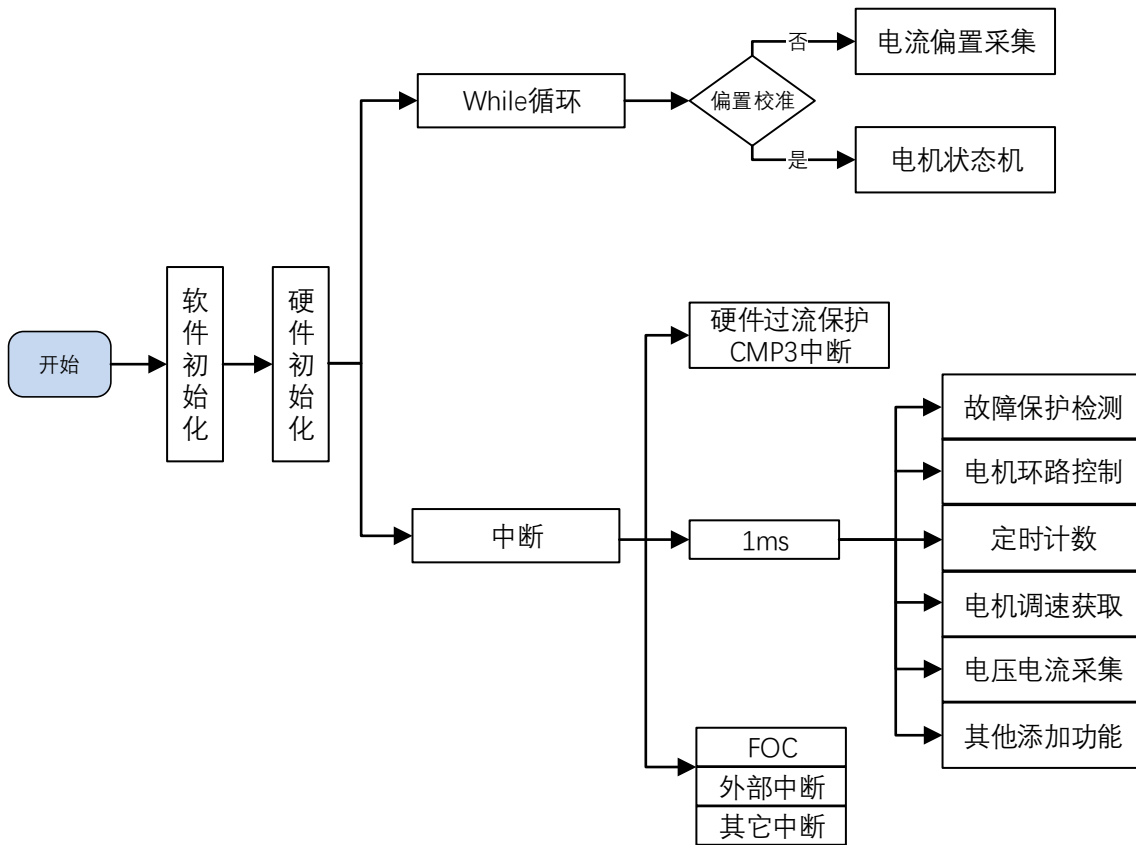


图 3-2 程序执行流程图

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() + 1ms 定时处理 TickCycle_1ms()等。

3.3.2 1ms 定时中断

产生 1ms 标志位，通过该标志位在主函数中调用 TickCycle_1ms()函数，TickCycle_1ms()函数主要包含了调速、故障保护检测、母线电流、母线电压采集等相关处理，调用子函数如下：

```
Speed_response();           // 环路控制函数
Fault_Detection();          // 故障检测
Fault_Recovery();           //故障恢复
StarRampDealwith();         // 电机启动 ATO 爬坡控制
LED_Display();              // LED 故障警报提示
```

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，如转矩补偿计算、低速观测器处理等。

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 5.1.1。](#)

3.3.5 外部中断

外部中断主要是用作睡眠唤醒的中断。

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。补充说明：压缩机无法被人为转动，需使用估算的 KE 参数运行起来后，使用软件复位或断电等方式，得到压缩机内电机的转动波形。选取波形反电动势中间的一个正弦波，测量其峰峰值 KeVpp 和周期 KeT。计算公式如下：

$$Ke = Pole_Pairs * \frac{KeVpp * KeT}{207.846}$$

示例，测量反电动势波形如下：

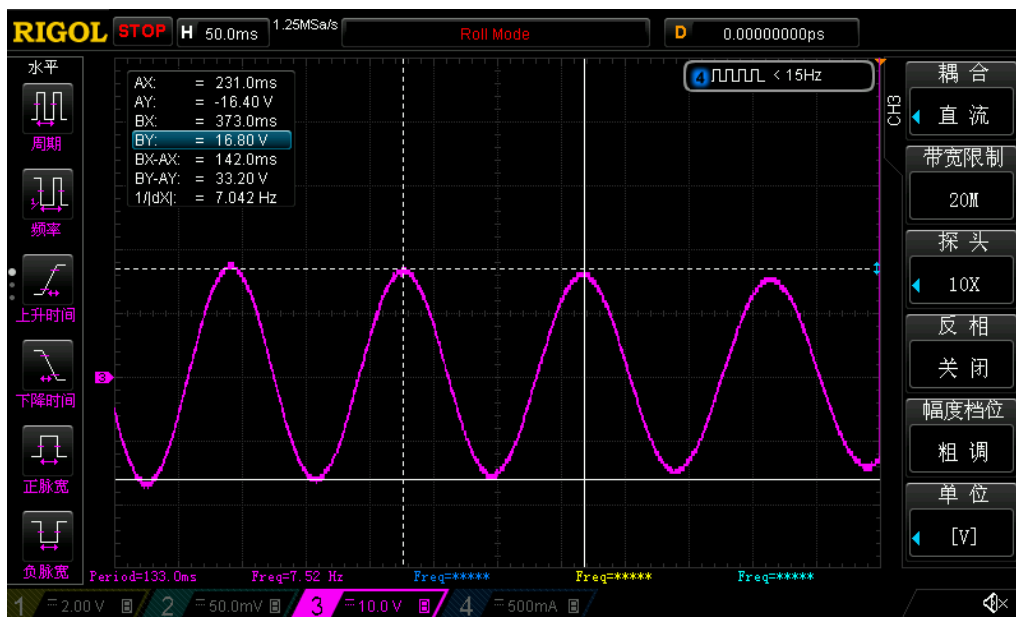


图 4-1 反电动势波形

测量峰峰值 KeVpp 为 33.2V，周期 KeT 为 142ms，极对数 Pole_Pairs 为 4，则：

$$\text{反电动势Ke} = 4 * \frac{33.2 * 142}{207.846} = 90.73$$

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

4.1.3 对应程序

```

/* -----电机参数值----- */
#define Pole_Pairs          (3)                ///< 极对数
#define RS                  (0.21)            ///< 电阻, ohm
#define LD                  (0.000224)        ///< 电感, H
#define LQ                  (0.000224)        ///< 电感, H
#define KeVpp              (12.8)            ///< (V) 反电势电压-峰峰值 */
#define KeF                (131.6)           ///< (Hz) 反电势测试时频率 */
#define Ke                  (Pole_Pairs * KeVpp * 1000 / KeF / 207.84)  ///< 反电势系数 */
#define Psi                 (float)(KeVpp / 2.0 / 1.732 / _2PI / KeF)
    
```

4.2 确认芯片内部相关数据配置

```

/* -----载波频率设置----- */
#define PWM_FREQUENCY        (5.0)
/* -----死区时间设置----- */
#define PWM_DEADTIME        (1.3)
/* -----单电阻最小采样窗口设置----- */
#define MIN_WIND_TIME       (PWM_DEADTIME +1.0)
/* -----正反转设置----- */
#define IRMODE               (CW)
    
```

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。低压压缩机一般默认 16K/5K 都可，使用低速观测器启动则需要 5K 载波；
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，如果电机反转了，则直接修改 IRMODE 配置即可。压缩机应用应当保证电机转向正常，不可反转。

4.3 确认硬件参数

```

/* -----电流基准的电路参数----- */
#define HW_RSHUNT          (0.006) //          ///< (Ω) 采样电阻
#define HW_ADC_REF        (4.5)      ///< (V)  ADC参考电压
#define HW_AMP_GAIN       (10.5)     ///< 运放放大倍数

/* -----母线电压采样分压电路参数----- */
#define RV1                (27.0)    ///< (kΩ) 母线电压分压电阻1
#define RV2                (0.0)     ///< (kΩ) 母线电压分压电阻2
#define RV3                (3.0)     ///< (kΩ) 母线电压分压电阻3
#define VC1                (1.0)     ///< 电压补偿系数 0.9
    
```

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 $0.8 \times V_{REF}$ ，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 = $V_{REF}/HW_RSHUNT/HW_AMP_GAIN$ ；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_AMP_GAIN 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)。

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，需设置小于电机的退磁电流，软件过流为软件触发，保护时间不及硬件过流。

2. 设置过欠压保护以及保护恢复参数，详细设置参考[章节 5.1.2](#)；

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，其中硬件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)。

```

}/* -----
                                     1.过流保护参数配置区域
-----

/* -----硬件过流保护比较值来源----- */
#define Compare_Mode                (Compare_DAC)

/* -----硬件过流保护值----- */

#define OverHardcurrentValue        (20.0)

/* -----软件过流保护使能----- */
#define SoftCurrentProtectEnable    (1)

/* -----软件过流保护值----- */
#define OverSoftCurrentValue        I_Value (15.0)
#define OverSoftCurrentDectTime    (30)
}/* -----
                                     2.过欠压保护参数配置区域
-----

/* -----过欠压保护使能----- */
#define VoltageProtectEnable        (1)

/* -----过欠压保护值----- */
#define Over_Protect_Voltage        (28.0)
#define Over_Recover_Vlotage       (26.0)
#define Under_Protect_Voltage       (9.0)
#define Under_Recover_Vlotage       (10.0)

```

4.5 启动说明

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。

1. 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 过大有可能出现在低速估算器切入硬件估算器的瞬间出现失步。

2. 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。

3. ATO 参数

程序可以使用固定的 ATO，也可以增加。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：

```

9
0 /* -----启动的ATO参数设置----- */
1 #define ATO_BW1 (100.0)
2 #define ATO_BW2 (110.0)
3 #define ATO_BW3 (120.0)
4 #define ATO_BW4 (150.0)
5 #define ATO_BW5 (180.0)//280
6
    
```

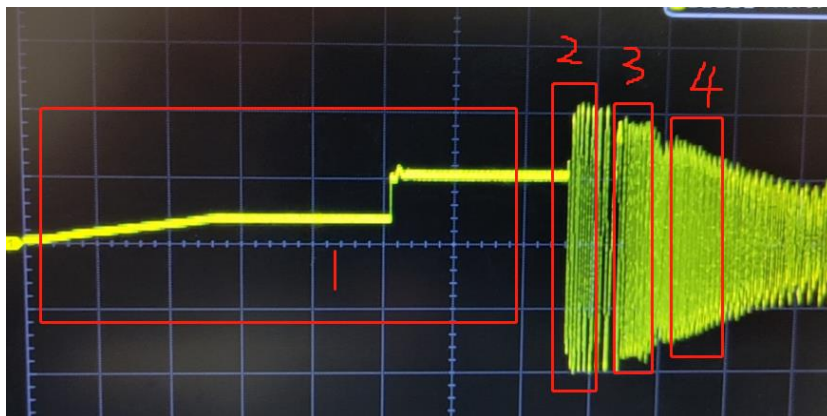
4. 启动电流

```

/* -----启动电流----- */
#define ID_Start_CURRENT I_Value(0.0)
#define IQ_Start_CURRENT I_Value(10.0)
    
```

启动电流：一般 ID_Start_CURRENT 固定设置为 0，低压压缩机中 IQ_Start_CURRENT 一般固定为 10A，启动电流 IQ_Start_CURRENT，不能过小否则启动转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；

5. 启动波形分析



1——预定位阶段，通过预定位控制压缩机启动位置。

2——强拖阶段：低速观测器运行阶段/OMEGA 启动的强制角度输出阶段，此阶段应尽可能短，一般 1-3 个波形内结束（图中故意拉长方便观察），表现为：相电流峰值与启动设定值一致、周期缓慢减小。

3、4——估算器运行到速度环运行，表现为：周期变化明显，电路峰值改变，可能出现电流波形凹陷，电流峰值出现下降/提升，速度向设定值接近。

4.6 低速观测器启动模式

低速观测器使能开关在 LowSpeedObserver.h 的头文件中，使能后启动方式变为低速观测器启动。

```

1 #ifndef __LowSpeedObserver_H_
2 #define __LowSpeedObserver_H_
3
4
5 #define LowSpeedObserverEnable      (0)
6
7 #define LowSpeedObserverLD          (LD)
8 #define LowSpeedObserverLQ          (LQ)
9
10 #define LowSpeedObserverGain1      (1500.0) //1500
11 #define LowSpeedObserverGain2      _Q12(1.8)//1.2 //0.7
12
    
```

注意事项：低速观测器主要目的是为了应对负载高的情况。第一次启动时，先关闭此功能尝试启动，目的是确保硬件电路和基础参数配置正确。

1. 需要修改启动 KPKI 实现相电流峰值快速达到设定值（过大是振荡抖动，影响启动；过小达不到设定电流——变相降低启动能力），具体表现为运行后前两个相电流波形的峰值为设定值
2. 修改预定位时间和 KPKI 预定位波形平稳，预定位波形为上电后的稳定电流大小的波形
3. 确定 RS、IS、母线电压等参数正确
4. 启动电流增大可以提升启动能力
5. 可先关闭低速观测器，空载带动电机；成功后，确保其他地方没问题，进入低速观测器空载启动、带载启动

1) 启动过程

可以简单看成：先通过“低速观测器”进行启动；低速观测器拉起电机，等待反电动势起来（此时 SMO/AO 估算器可以估算出速度）且估算器的速度(SMO/AO 估算器的估算速度值 speedfit)达到要求，切换到 SMO/AO 滑膜速度估算器，等到速度到达设定值，就切入速度环；

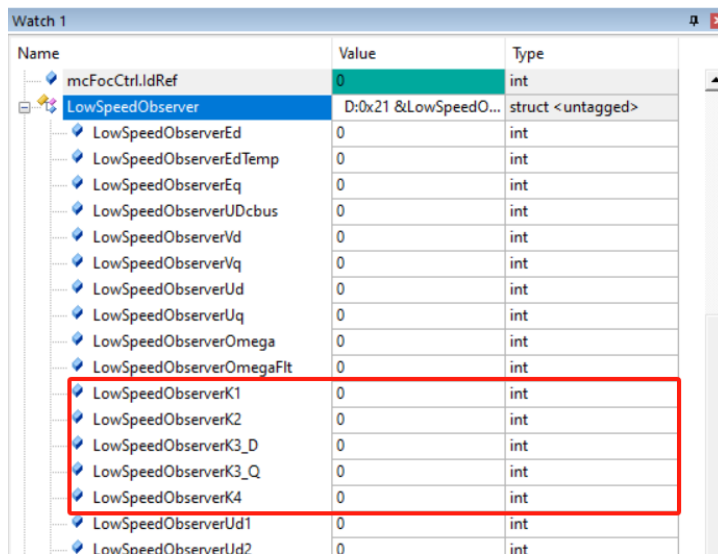
2) 低速观测器启动说明

参数	参数说明
LowSpeedObserverEnable	低速观测器使能标志，设置为1 将使能低速观测器。
LowSpeedObserverGain1	低速观测器增益值1。
LowSpeedObserverGain2	低速观测器增益值2，该增益过小会导致电机启动时在原来位置一直抖动，过大时会导致电机启动反转，取值范围0.0~8.0的Q12 格式。
LowSpeedObserverToAoObserverSpeed_Q15	低速观测器切AO /SMO观测器转速，当转速大于该设定转速时，由低速观测器切到SMO/AO 观测器。
SMOThetaComp	低速观测器和SMO/AO观测器切换时刻角度补偿值，该补偿值不需要改动，SMO/AO 估算器反电动势低通滤波引起的固定角度延迟，而低速观测器没有反电动势滤波环节。

如上表所示，低速观测器在 `LowSpeedObserver.h` 文件进行参数修改。低速观察器启动阶段结束后，对应标志位 `LowSpeedObserverToAoObserverFlag` 会置 1。因此，可通过观察该数值判断问题出现在启动的哪一个过程，如果 `LowSpeedObserverToAoObserverFlag` 为 0 则此时低速观测器启动阶段出现问题，否则，问题出现在其他地方（估算器参数、速度环参数等）。适当增加 `GAIN2`，可以提升启动能力(未出现反转前)。

3) 启动失败:

`LowSpeedObserverToAoObserverFlag` 为 0 则为低速观测器问题。



Name	Value	Type
mcFocCtrl.IdRef	0	int
LowSpeedObserver	D:0x21 &LowSpeedO...	struct <untagged>
LowSpeedObserverEd	0	int
LowSpeedObserverEdTemp	0	int
LowSpeedObserverEq	0	int
LowSpeedObserverUDcbus	0	int
LowSpeedObserverVd	0	int
LowSpeedObserverVq	0	int
LowSpeedObserverUd	0	int
LowSpeedObserverUq	0	int
LowSpeedObserverOmega	0	int
LowSpeedObserverOmegaFit	0	int
LowSpeedObserverK1	0	int
LowSpeedObserverK2	0	int
LowSpeedObserverK3_D	0	int
LowSpeedObserverK3_Q	0	int
LowSpeedObserverK4	0	int
LowSpeedObserverUd1	0	int
LowSpeedObserverUd2	0	int

可通过仿真查看对应低速观测器参数结果，需要保证 `K1-K4` 的值为正值，才可以使用低速观测器。一般为负值说明：电机参数不合理、运放倍数不合理、母线电压分压比过小等。

- 可能伴随电机抖动、电机转速拉满、如图 4-1、图 4-2 所示的启动波形缩小后类似矩形等；
- 电机反转，此时速度失控，电流波形类似正常，但无法切入 SMO(改小 `RS`)。

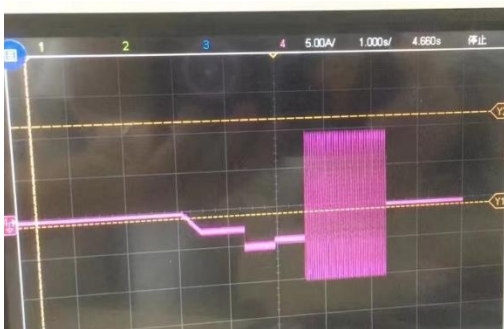


图 4-1 启动失败波形

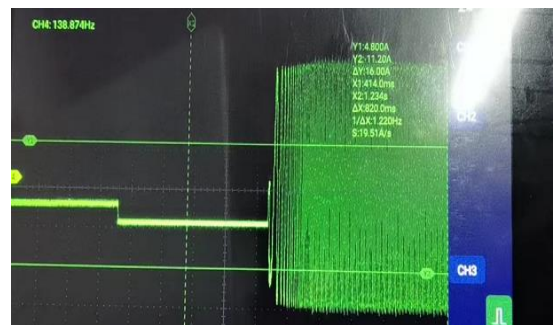


图 4-2 速度失控

- 电机参数不合理或实际采样电阻设置过小，预定位不准，压强差过大等原因导致；
- 4) 运行过程
- 低速出现抖动，可能为低速下压缩机电机本身重量不一致导致撞缸，可加入转矩补偿，一般情况下低压压缩机的转速不应低于 2000RPM；
 - 运行后相电流波形规律波动，调节 KPKI；
 - 切换速度环时，出现电流波形凹陷。



原因分析: 切换速度环前速度值过大；

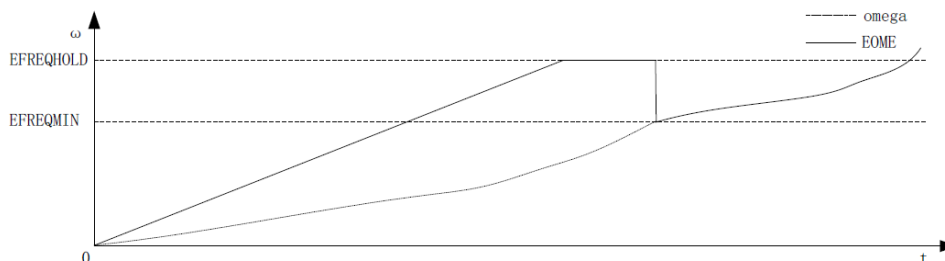
解决: 修改速度环 KPKI、降低切换速度环时达到的速度、提升切换速度环后运行的速度、降低开环的运行的最大电流。

4.7 OMEGA 启动模式

在 LowSpeedObserver.h 的头文件中，不使能低速观测器启动，则启动方式为 OMEGA 启动。

```
/* -----OMEGA启动参数----- */
#define Motor_Omega_Ramp_ACC          (25.0)
#define MOTOR_OMEGA_ACC_MIN           (55.0)
#define MOTOR_OMEGA_ACC_END           (50.0)
-----
```

当估算器的估算速度 ω 小于用户设定的最小值 ω_{min} (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 $\Delta\omega$ (Motor_Omega_Ramp_ACC 参数) 相加，同时根据 ω_{hold} (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 ω_{ome} 供角度计算模块算出估算器角度 θ ；当估算器的估算速度 ω 大于等于 ω_{min} 时，输出估算速度 ω 作为最终速度 ω_{ome} 。启动过程如下：



5 保护介绍

5.1 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，各种保护的设置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，说明是保护设置值不合理导致的，此时要调整保护的设置值。

5.1.1 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。

```

/* -----硬件过流保护比较值来源----- */
#define Compare_Mode          (Compare_DAC)          ///< 硬件过流值的来源

/* -----硬件过流保护值----- */
#define OverHardcurrentValue   (20.0)                ///< (A) DAC模式下的硬件过流值
    
```

2. 软件过流保护

程序通过获取三相电流值，当相电流值超过设定的软件过流保护值 OverSoftCurrentValue 时，则 1 毫秒计一次；计数值达到 OverSoftCurrentDectTime 后触发保护。

```

/* -----软件过流保护使能----- */
#define SoftCurrentProtectEnable      (1)            ///< 电压保护，0,不使能；1, 使能

/* -----软件过流保护值----- */
#define OverSoftCurrentValue          I_Value(15.0)  ///< (A) 软件过流值
#define OverSoftCurrentDectTime       (30)          ///< (ms) 软件过流检测时间
    
```

5.1.2 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。


```

-----
2. 过欠压保护参数配置区域
-----
/* ----过欠压保护使能---- */
#define VoltageProtectEnable      (1)                ///< 电压保护, 0,不使能; 1, 使能

/* ----过欠压保护值---- */
#define Over_Protect_Voltage      (28.0)            ///< (V) 直流电压过压保护值
#define Over_Recover_Voltage      (26.0)            ///< (V) 直流电压过压保护恢复值
#define Under_Protect_Voltage      (9.0)             ///< (V) 直流电压欠压保护值
#define Under_Recover_Voltage      (10.0)           ///< (V) 直流电压欠压保护恢复值
    
```

5.1.3 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2倍，且该相最大电流大于设定的PhaseLossCurrentValue值，则判定为缺相。

```

-----
/* ----缺相保护使能---- */
#define PhaseLossProtectEnable    (1)                ///< 缺相保护, 0,不使能; 1, 使能

/* ----缺相保护参数---- */
#define PhaseLossCurrentValue1    I_Value(0.1)      ///< (A) 缺相电流值, 运行时缺相保护值。(0.10)
#define PhaseLossCurrentValue2    I_Value(0.5)      ///< (A) 缺两项/三相的相电流值判断值, 启动时缺相保护值。
    
```

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。

5.1.4 堵转保护

堵转保护有3种方法检测:

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，因此可以通过该方式判断。

具体程序实现方法:

1. 方法一，当反电动势mcFocCtrl.esValue < 设定值，或估算转速 > 设定值且mcFocCtrl.esValue < 设定值时，触发堵转保护，延时StartRecoverTime时间重启;

```

/* ----当反电动势太小 或当 转速太大但反电动势却很小时进入保护---- */
if ( (mcFocCtrl.esValue < StallProtect1EsValue)
    || ((FOC_EOME > StallProtect2tSpeedValue)
        && (mcFocCtrl.esValue < StallProtect2tEsValue)))
{
    if (++mcFaultDect.stallDectEs >= 500)
    {
        mcFaultSource = FaultStall;
        FaultProcess();
    }
}
else
{
    if ( mcFaultDect.stallDectEs > 0)
    { mcFaultDect.stallDectEs --; }
}
    
```

2. 方法二，当mcFocCtrl.esValue < 设定值，或估算转速 > 设定值且mcFocCtrl.esValue的值 < 设定值时，触发堵转保护;

```
/* -----判断速度低于堵转最小值或者超过堵转最大值----- */  
if ((mcFocCtrl.speedFlt < Stall Protec3MinSpeed)  
    || (mcFocCtrl.speedFlt > Stall Protec3MaxSpeed))  
{  
    if (++mcFaultDect.stallDectSpeed >= 500) // 判断满足条件达到设置次数, 保护  
    {  
        mcFaultSource = FaultStall;  
        FaultProcess();  
    }  
}  
else  
{  
    if (mcFaultDect.stallDectSpeed > 0)  
    { mcFaultDect.stallDectSpeed --; }  
}
```

3. 方法三, 长时间没有切入速度环, 触发堵转保护

```
/* ----- */  
if (mcFocCtrl.ctrlMode == 0)  
{  
    mcFaultDect.Mode0DectCnt++;  
  
    if (mcFaultDect.Mode0DectCnt >= 3000)  
    {  
        mcFaultDect.Mode0DectCnt = 0;  
        mcFaultSource = FaultStall;  
    }  
}
```

5.1.5 其他保护

根据客户需求自行添加其他保护。

6 方案调试难点&解决方法

电压环调试	
常见问题	解决方法
启动一直有异常	调大低速观测器增益值，电机参数值重新测试矫正，调整切环参数
电机速度响应较慢	1. 调试外环的 SKP, SKI; 2. 调节时间 LOOP_TIME; 3. 如果只是加减速比较慢，就调节加减速的增减量。
转速或者功率达不到客户要求	1. 电流波形正弦的情况下，通过观测 FOC_UQ 是否饱和； 2. FOC_UQ 饱和，且 FOC_UD 值比较大的话，通过调整补偿角 FOC_THECOMP（正负都调整看看）确认是否能达到客户需求； 3. 以上还是达不到要求时，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. 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，基本都会影响启动和运行性能，解决好问题后要重新测试确认。	

7 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	2022/11/28	汤伟
V1.1	修改原理图，添加部分硬件说明，添加和删减部分调试说明，保护说明	2023/11/09	洪填宝

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