南京凌鸥创芯电子有限公司

LKS_08xDemo_FOC_V3.9 程序说明

第一章　绪论

本文档是针对 LKS08 系列芯片的 FOC 程序的调试说明，硬件是基于 LKS081/083/088 Demo 板，完整的原理图参考附录 1 LKS081/083/088 原理图. 下面从调试工具、硬件设计、软件说明、调试说明四个方面做详细说明。

可以使用 JScope 来辅助调试，详细的设置和使用方法参照《Jscop HSS 及 RTT 模式使用》。

第二章 硬件参数配置

2.1 电机参数测量

电机参数的计算可以参照《电机参数生成表》，把测得的相电感（uH）、相电阻（Ω）、反电势峰峰值(V)、电频率(Hz)填到《电机参数生成表》里，然后把生成的电机参数填到 MC_Parameter.h 对应位置即可。一个示例应用如表 2-1

表 2-1 电机参数生成表

极对数	P	5	U_MOTOR_PP
电阻/Ω	RS	2.160667	U_MOTOR_RS
电感/uH	LD/LQ	2701.6667	U_MOTOR_LD
磁链常数	Φ	0.01420692	U_MOTOR_FLUX_CONST
电阻测试数据	R1	4.33	线电阻，Ω
	R2	4.316	线电阻，Ω
	R3	4.318	线电阻，Ω
电感测试数据	L1	5.487	线电感，mH
	L2	5.153	线电感，mH
	L3	5.57	线电感，mH

峰峰值	Vpp	5.6	反电势，V
频率	F	18.11	反电势，Hz

电阻和电感可以采用数字电桥测试，测试电阻时要选用 100Hz，测试电感选用 1KHz。

对于无法测试的应用，可以让客户提供电机极对数。大部分应用，电机极对数可以通过开环电频率和机械转速计算，电机极对数 N = 60F/S，其中 F 为电频率，S 为机械转速。

磁链常数用示波器测试，用电压探头接电机两相，转动转子，捕捉反电势波形，测出峰峰值和电频率。对应某些不方便测试的情况，也可以开环把电机转起来，然后断电捕捉自由减速时的反电势波形，同样的测出峰峰值和电频率。反电势峰峰值和电频率测试结果如图 2-1 所示，Vpp = 5.4V，F = 16.78Hz。磁链常数计算公式为ϕ=Vpp/2/(2π)/sqrt(3)/F

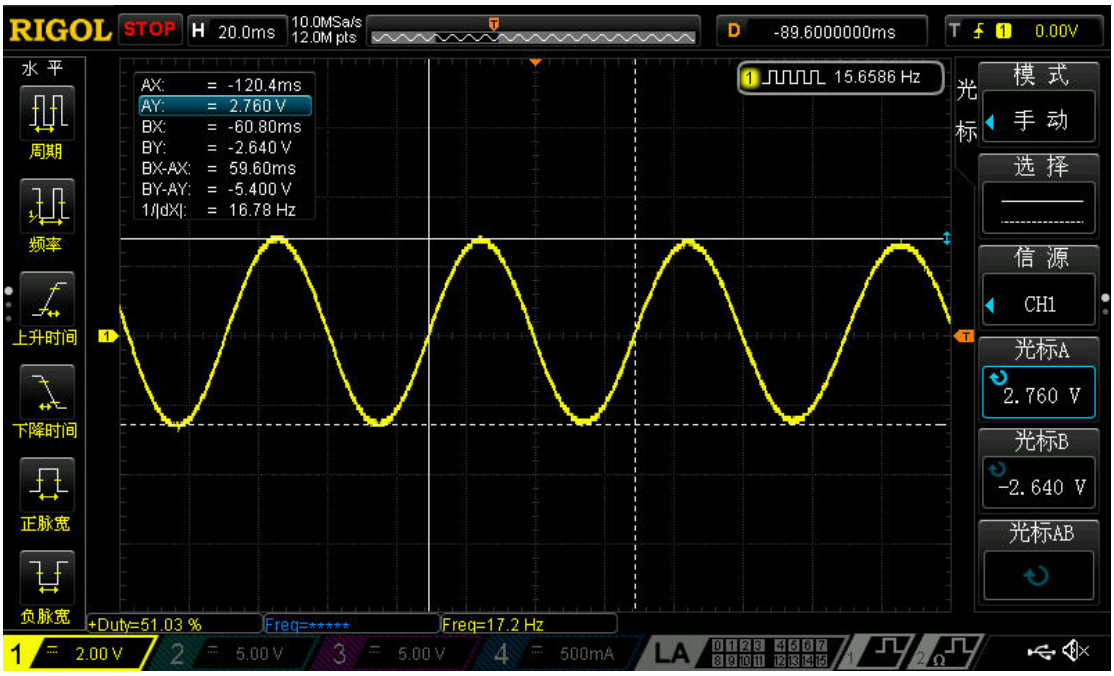


图 2-1 电机反电势波形

● 2.2 电压采样

母线电压采样电阻分压的方式处理，原理如图 2-2，母线电压的分压比为 R76/(R74+R75+R76)。母线电压采样通道对应的是 ADC_CHANNEL_12.

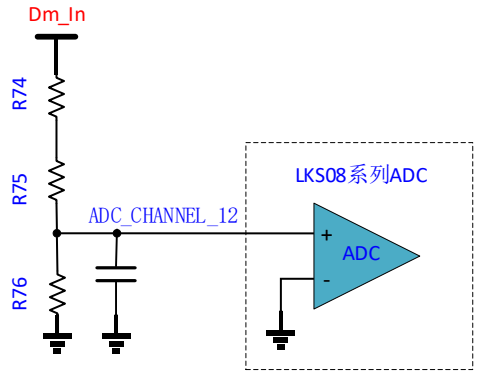


图 2-2 母线电压采样

● 2.3 电流采样

关于运放的详细使用请参考《LKS08x 运放应用笔记_V1.1》。由于芯片支持差分采样，故电流采样拓扑电流非常简单。Demo 板支持单电阻、双电阻、三电阻采样，采样电阻均为 0.005Ω。08 系列芯片内置了 4 组运放反馈电阻，OPA0&OPA1 的是 200K/10.2K、190K/20.2K、180K/30.2K、170K/40.2K，OPA2&OPA3 的是 200K/10.4K、190K/20.4K、180K/30.4K、170K/40.4K。在设计的时候尽量采用 OPA0&OPA1 或者 OPA2&OPA3 作为 AB 相的电流采样运放，这样可以进一步减小误差。

忽略内置运放电阻的误差，采用 200K/10.4K 这一组，设计的相线最大采样电流值为 3.6/0.005/(200/(10.4+20*2)) = 181.44A

母线最大采样电流值为 3.6/0.005/(200/(10.4+1)) = 41.04A

在实际项目中要注意合理设置最大采样电流值，一般按照 3 倍过载来设计，对于某些应用可以适当降低最大采样电流值，以提高电流采样精度。

电流的原理如图 2-3 所示，其中“Rshunt”为采样电阻。硬件上的实现见图 2-4。

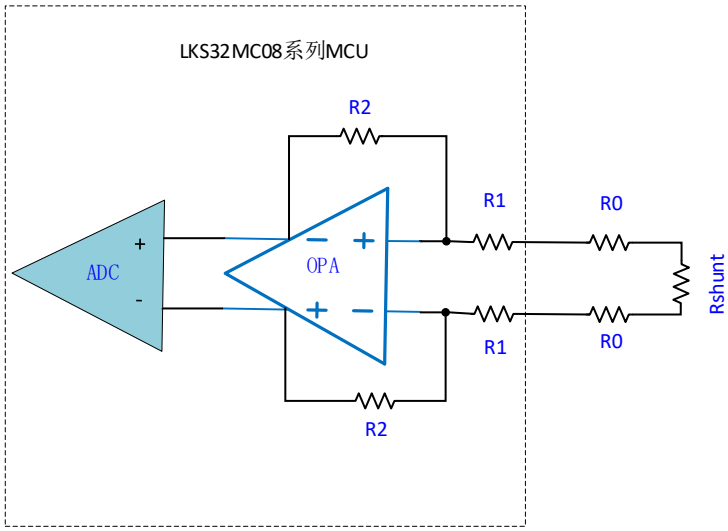


图 2-3 电流采样原理图

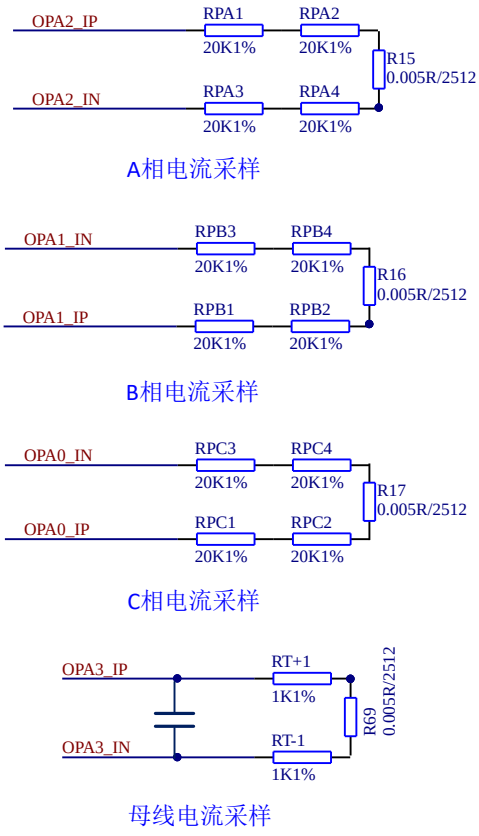


图 2-4 电流采样拓扑

电流的 ADC 采样通道对应关系为：

Ia ----------ADC_CHANNEL_2

Ib ----------ADC_CHANNEL_1

Ic ----------ADC_CHANNEL_0

IBus---------ADC_CHANNEL_3

● 2.4 硬件过流保护

硬件过流保护采用比较器和 DAC 处理,原理如图 2-5 所示。R69 为母线采样电阻，OC 接 CMP1_IP0，在程序中配置 CMP1 正端接 CMP1_IP0，负端接 DAC。

若 SYS_AFE_REG1 |= (DAC_RANGE_1V2 << 6)--(设置 ADC 量程为 1.2V)

SYS_AFE_DAC = 512，则设置的过流值为 512/4096/0.005*1.2 = 30A

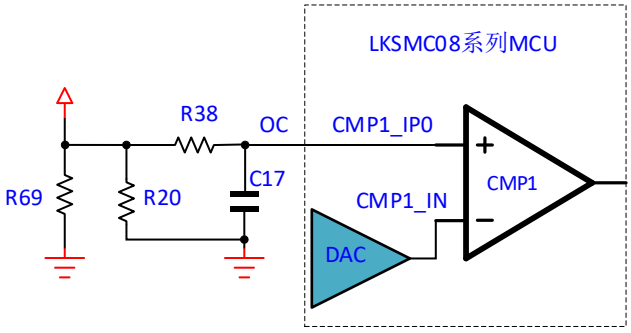


图 2-5 硬件过流原理图

● 2.5 MOS 管驱动方案

081Demo 板采用的是三个 LSK560 去驱动六个 NMOS 管，单个驱动拓扑如图 2-6 所示

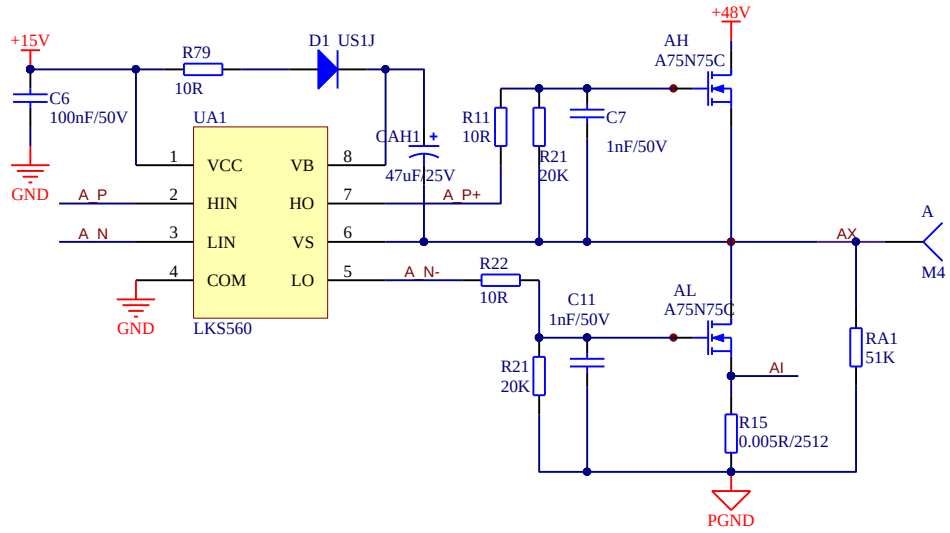


图 2-6 MOS 管驱动电路

MCPWM 的极性配置要参考 LKS560 的控制逻辑时序，由图 2-7 所示，当 HIN 为高时，HO 为高;LIN 为低时，LO 为高。对于 NMOS 管来说，高电平导通，低电平关闭。

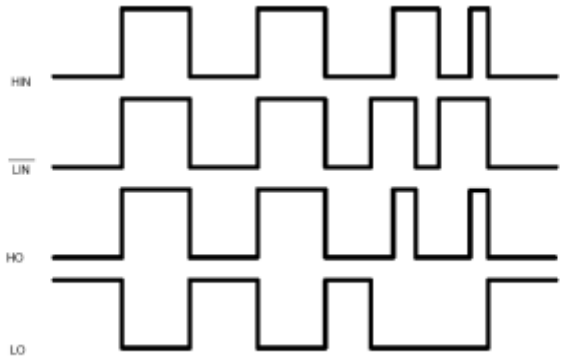


图 2-7 LKS560 的控制逻辑时序图

由图 2-8 可知，当 P 通道为高时，LKS560 的 HO 为高，上桥 NMOS 导通；此时 N 通道为低，LKS560 的 LO 为高，下桥 NMOS 也导通，就会造成 MOS 管直通，因此 N 通道输出要反相。结合硬件图的设置，MCPWM 的极性配置为 P/N 不交换，N 通道反相，P 通道不反相。

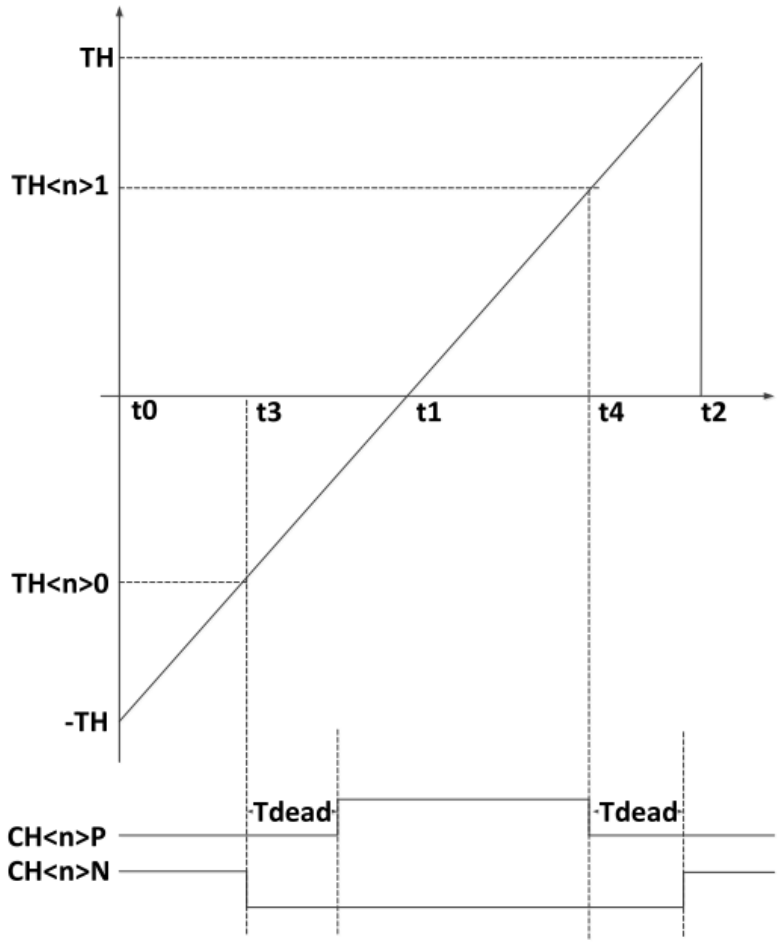


图 2-8 MCPWM 模块中心对齐互补输出模式控制逻辑

2.6 硬件配置程序

详细的配置程序请参见硬件初始化函数“Hardware_init( )”

2.6.1 ADC 设置 & ADC 采样通道

采用通道的设置在 hardware_config.h 里面。

注意：ADC_NormalModeCFG( )中 ADC0_CHN0、ADC0_CHN1 的配置不可以修改. ADC_BUS_VOL_CHANNEL 对应的是母线电压采样通道，如果修改了采样顺序，请读取母线电压值的时候改为对应的结果寄存器。关于 ADC 的详细说明请参考《LKS32MC08x_User_Manual》中 ADC 章节。

本程序采用的 ADC 配置是单段采样，左对齐，MCPWM_T0 触发，第一段采样结束触发中断,详细的配置方式请参考程序。ADC 采样通道如

下：

```
#define  ADC0_CURRETN_A_CHANNEL      (ADC0_CHANNEL_OPA2)   //A相电流通道
#define  ADC0_CURRETN_B_CHANNEL      (ADC0_CHANNEL_OPA1)   //B相电流通道

#define  ADC_BUS_VOLT_CHANNEL        (ADC_CHANNEL_12)      //母线电压采样通道
#define  M0_ADC_BUS_CURR_CH          (ADC0_CHANNEL_OPA3)   //母线电流通道
```

- **2.6.2 DAC & CMP 配置**

具体配置请参考程序中 DAC_Init()和 CMP_Init()函数，设置的方法参考 2.4 章节。

```
void DAC_Init(void)
{
  SYS_AnalogModuleClockCmd(SYS_AnalogModule_DAC, ENABLE);

  SYS_AFE_REG1 |= (DAC_RANGE_1V2 << 6);    /* 设置DAC满量程为1.2V；00:3V| 01:1.2V| 10:4.85V */

  SYS_AFE_DAC = 512;                       /* 1.2*512/4096/0.05 = 3A, 其中0.05为母线采样电阻*/
}

void CMP_Init(void)
{
  CMP_InitTypeDef CMP_InitStruct;

  CMP_StructInit(&CMP_InitStruct);

  CMP_InitStruct.CMP0_EN = DISABLE;          /* 比较器0开关 */
  CMP_InitStruct.CMP0_SELN = SELN_DAC;       /* CMP0_N 内部DAC 输出  */
  CMP_InitStruct.CMP0_SELP = SELP_CMP_IP0;  /* CMP0_P CMP1_IP0母线 */
  CMP_InitStruct.CMP0_InEnable = DISABLE;    /* 比较器信号输入使能 */
  CMP_InitStruct.CMP0_IE = DISABLE;          /* 比较器0信号中断使能 */

  CMP_InitStruct.CMP1_EN = ENABLE;           /* 比较器1开关 */
  CMP_InitStruct.CMP1_SELN = SELN_DAC;       /* CMP1_N 内部DAC 输出  */
  CMP_InitStruct.CMP1_SELP = SELP_CMP_IP0;  /* CMP1_P CMP1_IP0母线 */
  CMP_InitStruct.CMP1_InEnable = ENABLE;     /* 比较器信号输入使能 */

  CMP_InitStruct.CMP1_IE = ENABLE;           /* 比较器1信号中断使能 */
  CMP_InitStruct.CMP_FltCnt = 15;            /* 即滤波宽度=tclk 周期*16*CMP_FltCnt */
  CMP_InitStruct.CMP_CLK_EN = ENABLE;        /* 时钟使能 */

  Comparator_init(&CMP_InitStruct);

}
```

- **2.6.3 PGA 配置**

程序中采用的是双电阻采样，使用的运放是 OPA2(Ia)和 OPA1(Ib)，因

此要使能对应的 OPA 时钟。不使用的 OPA 要关闭时钟，以节省功耗，详细的配置参考 PGA_Init()函数。关于运放的详细使用请参考《LKS08x 运放应用笔记_V1.1》。运放倍数的配置在 hardware_config.h 里面，如下所示：

```
#define PGA_GAIN_20          0        /* 反馈电阻200:10 */
#define PGA_GAIN_9P5         1        /* 反馈电阻190:20 */
#define PGA_GAIN_6           2        /* 反馈电阻180:30 */
#define PGA_GAIN_4P25        3        /* 反馈电阻170:40 */

#define  OPA0_GIAN           (PGA_GAIN_20)
#define  OPA1_GIAN           (PGA_GAIN_20 << 2)
#define  OPA2_GIAN           (PGA_GAIN_20 << 4)
#define  OPA3_GIAN           (PGA_GAIN_20 << 6)
```

- **2.6.4 MCPWM 配置**

MCPWM 的极性参考 2.5 节的说明，配置为中心对齐、互补输出模式，具体的实现方式参考程序中 MCPWM_init()函数。MCPWM 也可以使能通过 CMP 产生 FAIL 信号，关闭 PWM 输出。关于 MCPWM 的详细说明请参考《LKS32MC08x_User_Manual》中 MCPWM 章节。

- **2.6.5 Timer0 配置**

Timer0 产生 0.5ms 的定时中断，提供 1ms 计数时基。1ms 是系统状态机处理的周期。

```
TIM_TimerInitTypeDef TIM_InitStruct;

TIM_TimerCmd(TIMER0, ENABLE);                                   /* Timer0 模块使能 */

TIM_TimerStrutInit(&TIM_InitStruct);
TIM_InitStruct.Timer_CH0_WorkMode = TIMER_OPMode_CMP;  /* 设置Timer CH0 为比较模式 */
TIM_InitStruct.Timer_CH0_CapMode = TIMER_CapMode_None;
TIM_InitStruct.Timer_CH0Output = 0;                             /* 计数器回零时，比较模式输出极性控制 */
TIM_InitStruct.Timer_CH1_WorkMode = TIMER_OPMode_CMP;  /* 设置Timer CH1 为比较模式 */
TIM_InitStruct.Timer_CH1_CapMode = TIMER_CapMode_None;
TIM_InitStruct.Timer_CH1Output = 0;                             /* 计数器回零时，比较模式输出极性控制 */
TIM_InitStruct.Timer_TH = 48000;                                /* 设置计数器计数模值 */
TIM_InitStruct.Timer_CMP0 = 24000;                              /* 设置比较模式的CH0比较值 */
TIM_InitStruct.Timer_CMP1 = 500;
TIM_InitStruct.Timer_Filter0 = 0;                       /* 设置捕捉模式或编码器模式下对应通道的数字滤波值 */
TIM_InitStruct.Timer_Filter1 = 0;
TIM_InitStruct.Timer_ClockDiv = TIM_Clk_Div1;           /* 设置Timer模块数据分频系数 */
TIM_InitStruct.Timer_IRQEna = Timer_IRQEna_Zero;        /* 开启Timer模块过0中断 */
TIM_TimerInit(TIMER0, &TIM_InitStruct);
```

- **2.6.6 GPIO 配置**

GPIO 的配置仅提供了 PWM 端口和串口端口的配置，其他的应用请参照格式另外添加。P1.4、P1.5、P1.6、P1.7、P1.8、P1.9 是固定的 PWM 输出口，对于 087D、08E、084D、086 等内置驱动的芯片，还要配置 P3.13/P1.12/P1.15 为输出状态。

```
/* MCPWM P1.4~P1.9 */
GPIO_InitStruct.GPIO_Mode = GPIO_Mode_OUT;
GPIO_InitStruct.GPIO_Pin = GPIO_Pin_4 | GPIO_Pin_5 | GPIO_Pin_6 | GPIO_Pin_7 | GPIO_Pin_8 | GPIO_Pin_9;
GPIO_Init(GPIO1, &GPIO_InitStruct);
GPIO_PinAFConfig(GPIO1, GPIO_PinSource_4, AF3_MCPWM);
GPIO_PinAFConfig(GPIO1, GPIO_PinSource_5, AF3_MCPWM);
GPIO_PinAFConfig(GPIO1, GPIO_PinSource_6, AF3_MCPWM);
GPIO_PinAFConfig(GPIO1, GPIO_PinSource_7, AF3_MCPWM);
GPIO_PinAFConfig(GPIO1, GPIO_PinSource_8, AF3_MCPWM);
GPIO_PinAFConfig(GPIO1, GPIO_PinSource_9, AF3_MCPWM);

/* P0.15-RX0, P1.0-TX0  UART0 */
GPIO_StructInit(&GPIO_InitStruct);
GPIO_InitStruct.GPIO_Mode = GPIO_Mode_IN;
GPIO_InitStruct.GPIO_Pin =  GPIO_Pin_15 ;
GPIO_Init(GPIO0, &GPIO_InitStruct);

/* P0.15-RX0, P1.0-TX0  UART0 */
GPIO_InitStruct.GPIO_Mode = GPIO_Mode_OUT;
GPIO_InitStruct.GPIO_Pin =  GPIO_Pin_0;
GPIO_Init(GPIO1, &GPIO_InitStruct);

/* P0.15-RX0, P1.0-TX0  UART0 */
GPIO_PinAFConfig(GPIO0, GPIO_PinSource_15, AF4_UART);
GPIO_PinAFConfig(GPIO1, GPIO_PinSource_0, AF4_UART);
```

第三章 软件说明

程序主要分为两部分，一部分是状态机的调度，处理周期是 1ms；另外一部分是 FOC 的计算，放在 ADC 采样完成中断里面处理，处理周期随着 PWM 频率变化，本程序中用的是 16K，62.5us。

3.1 状态机的调度

状态机按照电机的运行顺序来调度，处理函数为 Sys_State_Machine()，处理周期为 1ms。状态机的切换顺序可以根据不同负载做相应调整，调度的变量为“struFOC_CtrProc.eSysState”。控制流程图如图 3-1 所示。

状态机启动变量为“struFOC_CtrProc.bMC_RunFlg”，当“struFOC_CtrProc.bMC_RunFlg ！= 0”时，状态机启动。常用的流程为预充电()-->顺逆风检测-->初始位置检测/闭环运行-->预定位-->开环拖动-->闭环运行，故障检测监视每个状态的运行情况，一旦有故障发生，即“stru_Faults.R!= 0”则进入故障状态，当故障解除后，即“stru_Faults.R= 0”时，回到 idle 状态。具体各个状态的功能实现，这里面不再做详细介绍，请参考各个状态的程序。

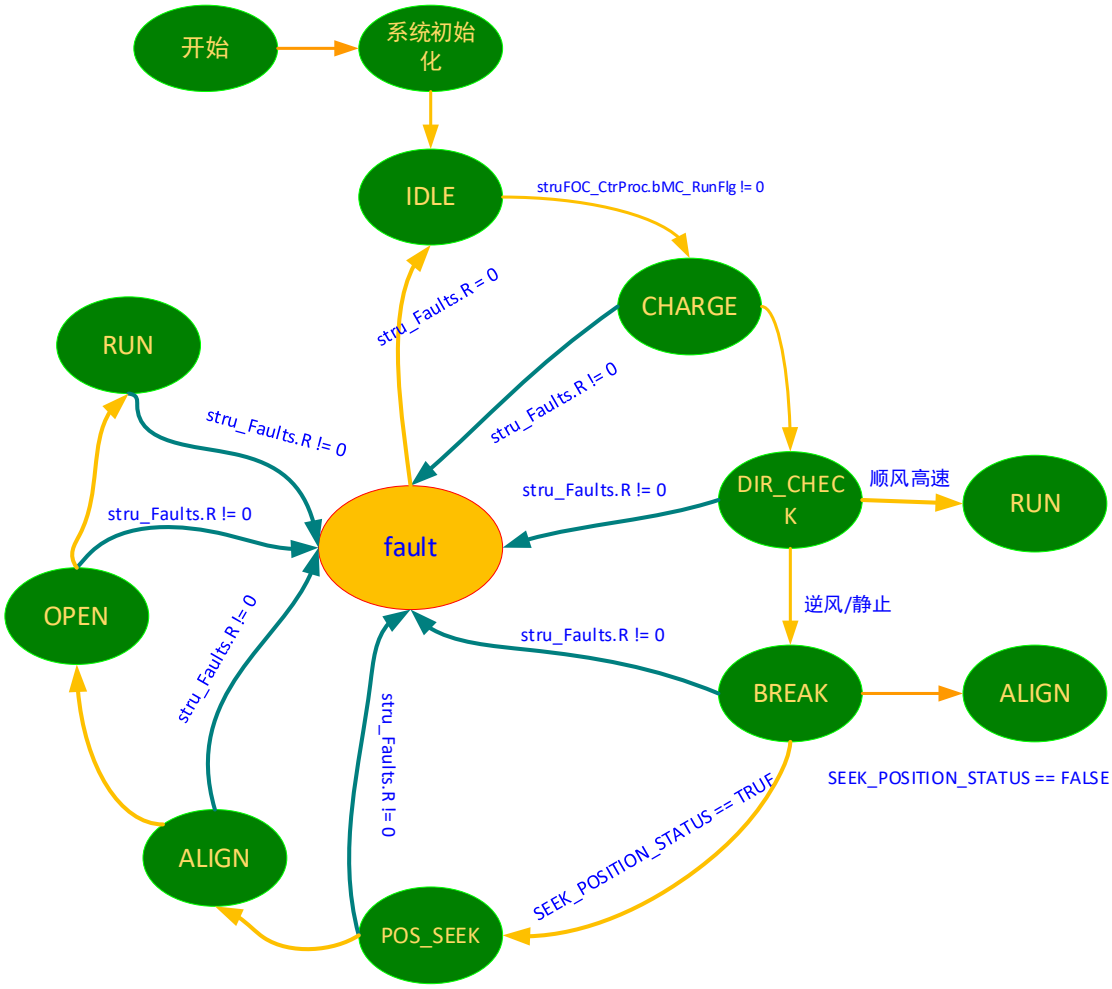


图 3-1 状态机调度逻辑图

3.2 Interrupt 处理

主中断采用的是 ADC 第一段采样完成中断。运行周期是设置的 PWM 频率，主要处理的是 FOC 计算和转子位置估算，处理函数为 FOC_Model()，处理过程的调度通过 struFOC_CtrProc.eSysState 来实现。控制时序图如 3-2 所示。

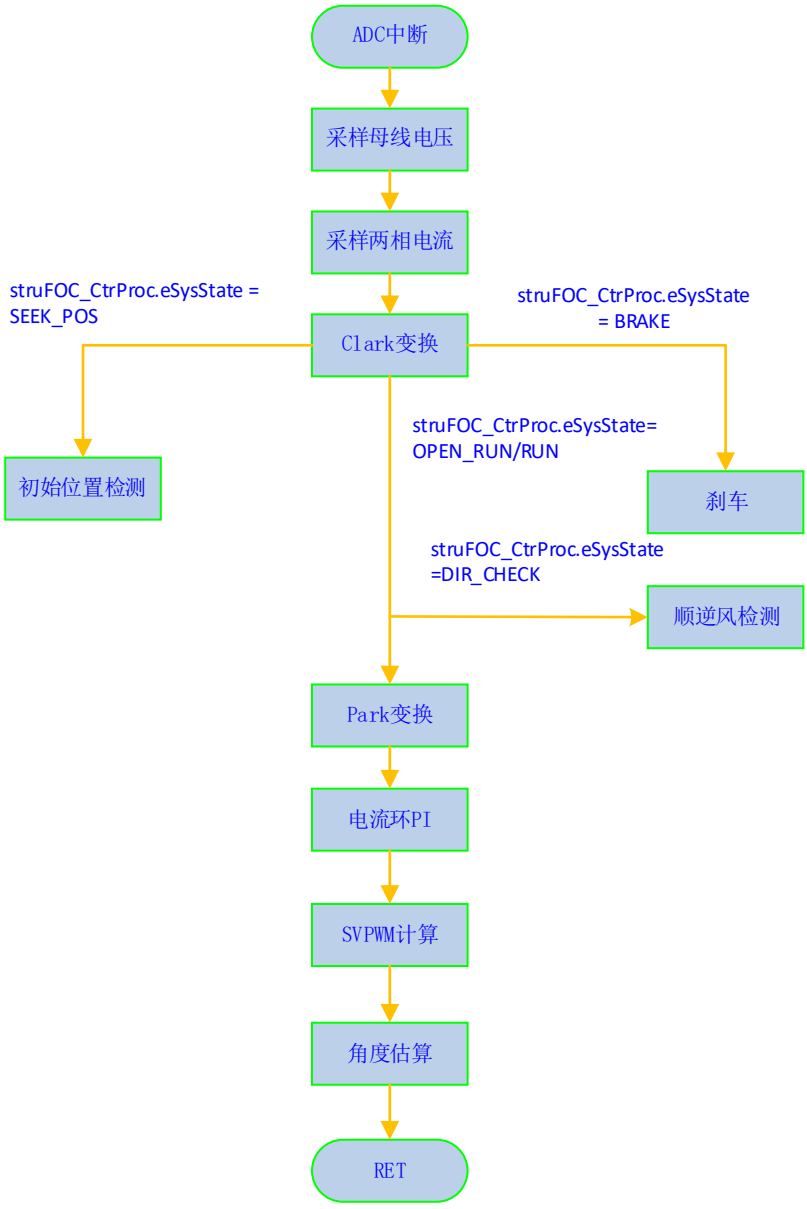


图 3-2 中断服务子程序流程图

3.2 PI 整定说明

电流环和速度环都是采用 PI 调节来保证环路的稳定，因此 PI 参数的整定非常重要，关于 PI 的整定技巧，网上有很多资料可以参考，这里不再对其原理和整定方法做详细说明。

算法库里根据电机参数内置了电流环 PI 参数，因此在测量电机参数时一定要尽量准确，否则可能会导致系统不稳定。

在实际调试中要先整定电流环 PI，然后再整定速度环 PI。PI 整定的参考波形如图 3-3，3-4 所示，图中黑色是给定值，红色是反馈值。整定的总原则是反馈值能很快的跟随给定值，超调小，无静态误差。

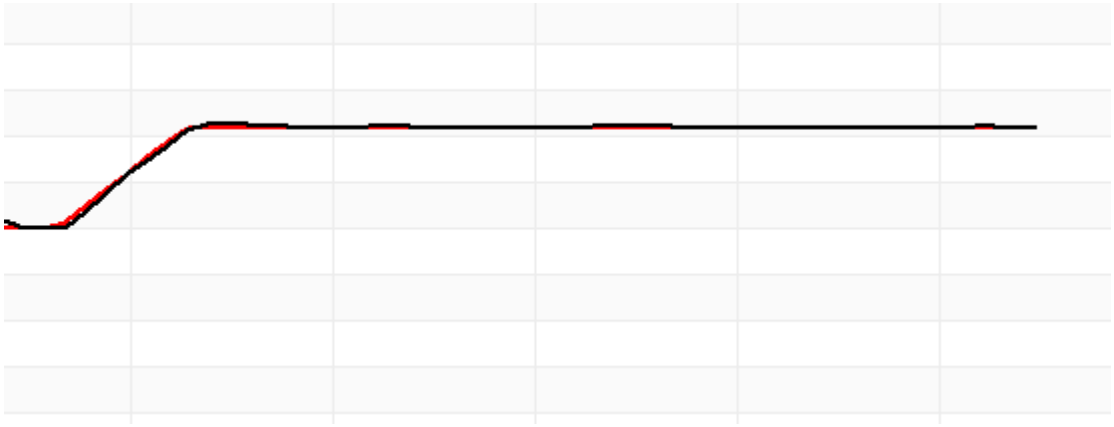


图 3-3　PI 参数正常

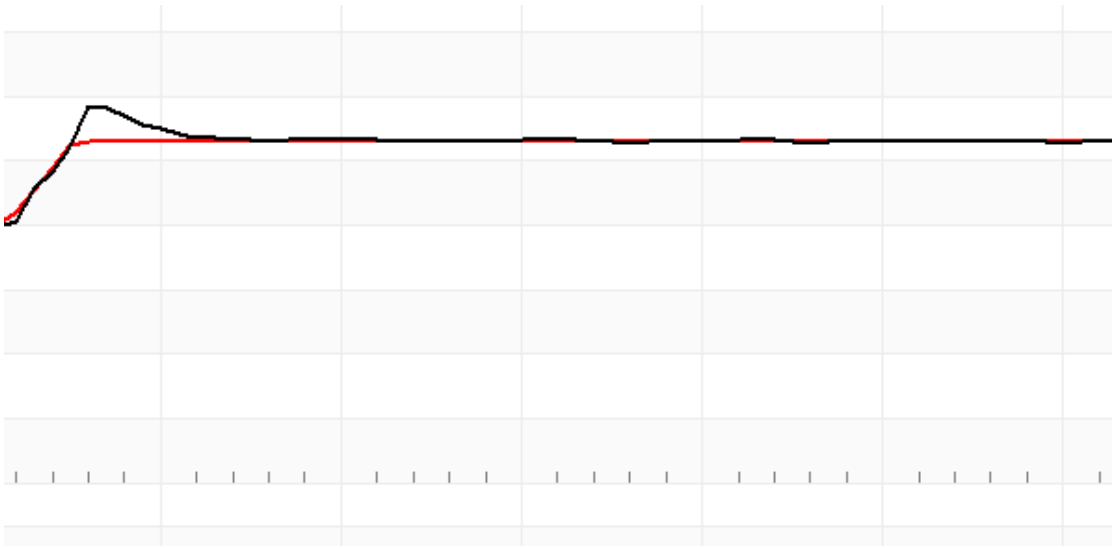


图 3-4 PI 参数异常

3.3 常用保护设计逻辑

电机控制中常用的保护功能有过欠压、过流、堵转、缺相、过温、离水空转、二次启动等。具体的实现过程请参照程序中 fault_detection.c 里面的对应处理函数。

软件过流、硬件过流、过欠压、过温的原理简单，只要设定合理的保护阈值即可。堵转、缺相、离水空转、二次启动相对复杂一点，下面就这 4 中保护的设计做具体说明。

3.3.1 堵转保护

堵转保护的检测原理是电流和转速不匹配。因此，目前有三种检测方法来实现。

1. 转速超出设定的范围，即判定为堵转；

2. 相电流超出设定的范围，即判定为堵转；

3. q 轴电流(Iq)和转速不匹配，Iq 很大，转速却很小，即判定为堵转。

对于前两种方法，参考正常工作的转速和相电流最大值设定即可；第三种方法要根据实际负载确定，请使用《堵转参数测试 -Iq-Uq》表，表中 Iq 为“struFOC_CtrProc.struFOC_CurrLoop.mStatCurrDQ.nAxisQ”，Uq 为“struFOC_CtrProc.struFOC_CurrLoop.mStatVoltDQ.nAxisQ”，Speed 为“struMotorSpeed.wSpeedfbk”。根据表 3-1、3-2 设定合理的堵转检测电流和堵转检测转速.

表 3-1 正常运行时的测试数据

Iq	Uq	Speed	Speed/Iq	Speed/Uq
420	2000	19500	9.75	46.43
724	2700	26600	9.85	36.74
1016	3340	32400	9.70	31.89
1300	5400	38300	7.09	29.46

表 3-2 堵转时的测试数据

Iq	Uq	Speed	Speed/Iq	Speed/Uq
420	420	1500	3.57	3.57
724	550	300	0.41	0.55
1016	610	1000	0.98	1.64
1450	1450	230	0.16	0.16

- **3.3.2 启动保护**

二次启动的检测方法和堵转检测基本相同，不同的是启动保护是在进入 OPEN/RUN 状态 10s 内检测，检测的灵敏度比堵转检测的高。启动保护恢复的时间也比较短，现在的设定时间是 500ms。

对于二次启动现在还加了在 OPEN 之后，在设定的时间内不能进入 RUN 状态也会判定为启动失败。具体的设定值要根据实际负载来调整

- **3.3.3 缺相检测**

图 3-5、3-6 是 Scope 上捕捉的电流波形，由图可知当 C 相缺相时，电流值不连续，只在某些角度点上有值。因此在半周期内对相线电流进行积分，如果有相电流积分值特别小的情况下判定为缺相；另外当不接电机时，三相电流的积份值都非常小，也判定为缺相。具体的实现在 CurrentAmplitudeCalc()、FaultPhaseCheck()里。

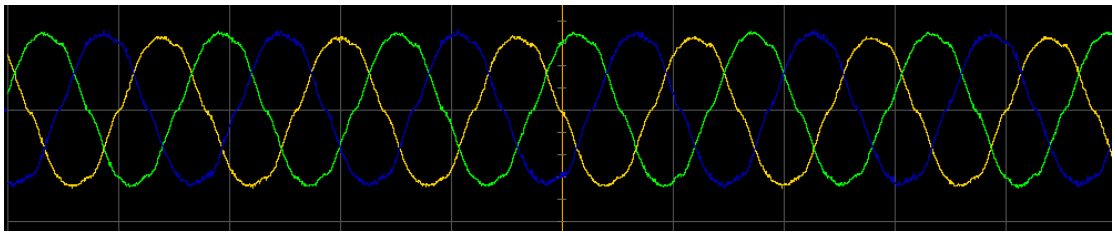


图 3-5 正常运行时的电流波形

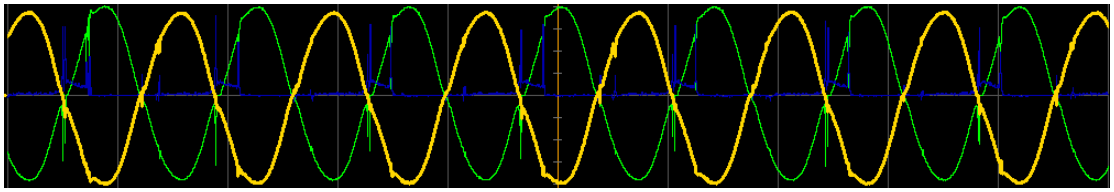


图 3-6 C 相缺相时的电流波形

● 3.3.4 离水空转保护

对于水泵类应用，长时间空转会烧毁机械密封或者填料密封，因此要检测空转的情况。在空转的情况下，转速高而电流小，根据这个特性来用 q 轴电流和转速来检测是否空转，实际的设定阈值要根据实际负载来确定

● 3.4 功率计算说明

功率的计算公式为 P = ABS(Id*Ud) + ABS(Iq*Uq),其中 ABS 为求绝对值函数，功率计算的实现是在函数 PowerCalc()里面。功率的标定请使用《功率拟合数据表》，具体的步骤参照图 3-7

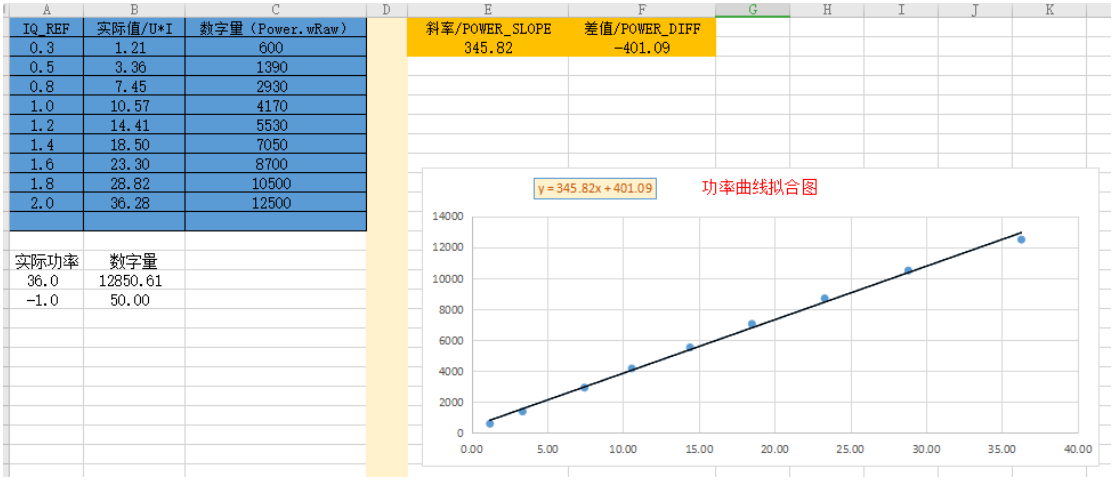


图 3-7 功率曲线拟合图

第一步：记录测试数据，记录不同功率下的实际功率值(电源显示)和计算值(struPower.wPowerValue)，蓝色部分。数据越多，拟合的功率误差就越小。

第二步：运用 excel 自带功能，为了方便，采用线性拟合，生成功率曲线拟合图；

第三步：记录曲线斜率和差值，分别赋给程序中 POWER_SLOPE 和 POWER_DIFF，橙色部分；

在功率环的应用中，采用 POWER_CALC(val)来计算设定功率，其中 val 为实际的功率值。

在功率拟合数据中也提供了实际功率和功率数字量之间的转换，在调试的时候可以作为参考来确定功率环路的调试有没有问题

● 第四章 调试说明

在调试前，请参考“附录 2 MC_Parameter.h 参数说明”，所有的参数均在“MC_Parameter.h”里面。调试之前要测试好电机参数、合理的设计采样电阻和运放倍数，具体的方法参考前面的章节。以下的说明默认电机参数正常，运放参数设计合理。

当调试时，可以先屏蔽故障检测函数，即 FaultCheck()函数，保留硬件过流，当性能调试好后再去匹配故障检测值。

● 4.1 设置额定电流、额定电压、额定转速

额定电流、额定转速、额定电压是整个 FOC 计算系统的基准，要合理这三个值，按照注释，额定电压设置为正常工作电压，对于高压应用，一般要考虑市电波动的影响(一般 15%左右)。额定电流设置为正常工作的相电流的最大值，额定转速设置为最大工作电频率的 2 倍。

```
/*****************************额定参数*********************************/
#define U_RATED_VPN             (24.0)        //单位:V, 电机额定工作电压，设置为正常工作电压

#define U_RATED_CUR             (2.0)         //单位:A, 电机额定工作电流，相电流最大值
#define U_MAX_FREQ              (300.0)       //单位:Hz, 电机额定转速，     电机最高运行频率
*2
```

● 4.2 硬件通路测试

选择 DEBUG_PWM_OUTPUT 为 TEST_ON，设置电源限流 0.1A 左右，测试三相对地的波形。注意示波器的地线不要接，以免误操作，烧坏示波器。

按照例程的设置，三相输出应该是频率为 16K，占空比 25%的方波，如果输出不正确，要检查时程序配置的问题，还是硬件电路的问题，MOS 管或者驱动芯片损坏。

测试信号正常后才能进入下一步调试

● 4.3 初始位置检测

```
/************************初始位置检测参数**************************/
#define SEEK_POSITION_STATUS    (TRUE)    //初始位置检测状态  TRUE 为使能, FALSE 为不使能
#define U_START_ANGLE_COMP    (0)         //单位:度  初始位置检测补偿角度
```

```
#define IPD_PLUS_TIME_SETTING    (500)   /* 脉冲注入时间宽度设置 单位 us */
#define IPD_WAIT_TIME_SETTING    (300)   /* 脉冲空闲等待时间宽度设置 单位 us */

#define IPD_PLUS_TIME_WIDTH      (u32)(IPD_PLUS_TIME_SETTING*(MCU_MCLK/1000000))
                                                    /* 脉冲注入时间宽度设置 单位 clk */
#define IPD_PLUS_WAIT_TIME       (u32)(IPD_WAIT_TIME_SETTING*(MCU_MCLK/1000000))
                                                    /* 脉冲空闲等待时间宽度设置 单位 clk */
```

调试时调整脉宽注入的时间宽度和空闲等待时间的大小，去判断检测的精度，一般来说，时间越大，精度越高，同时带来的噪声也越大。

当启动有反偏时，适当加大 U_START_ANGLE_COMP，一般控制在 0~60°之间，U_START_ANGLE_COMP 为实际角度值，即 10 代表 10°。

图 4-1 初始位置检测波形

初始位置检测电流波形如图 4-1 所示，当电流幅值比较小的时候就适当加大检测脉冲宽度。

● 4.4 预定位

```
/*****************************预定位参数*****************************/
#define ALIGN_ANGLE                 (0)       //单位:度 预定位角度
#define U_START_CUR_SET_F           (0.0)     //单位: A 第一段定位电流
#define U_START_CUR_SET_S           (0.8)     //单位: A 第二段定位电流
#define DC_LOCATION_HOLDTIME_TOTAL_LENTH    (1)    //单位: ms 第一段定位时间
#define DC_LOCATION_HOLDTIME_TOTAL_LENTH_STAGE1    (1)    //单位: ms 第二段定位时间
#define DC_ALIGN_TOTAL_LENTH        (DC_LOCATION_HOLDTIME_TOTAL_LENTH +
```

```
                                    DC_LOCATION_HOLDTIME_TOTAL_LENTH_STAGE1)   //定位总时长
#define ALIGN_CURRENT_ACC          (15.0)   //单位: (1/8)A   定位电流加速调整值   初始位置检
                                    //测使能后给到最大值，不能超过 30，否则数据会溢出。
#define ALIGN_CURRENT_DEC          (15.0)     //单位: (1/8)A   定位电流减速调整值   初始位置
                                    //检测使能后给到最大值，不能超过 30，否则数据会溢
出。
```

使能了初始位置检测功能后，预定位状态只是给开环启动提供电流，第一段定位时间和第二段定位时间均设置为 1，定位电流的加速和减速均给到 15.

当不使用初始位置检测功能时，定位时间和定位电流要根据实际负载来设置，负载越重，时间越长，电流越大。第一段和第二段定位时间均为 1s 的相线波形如图 4-2 所示。

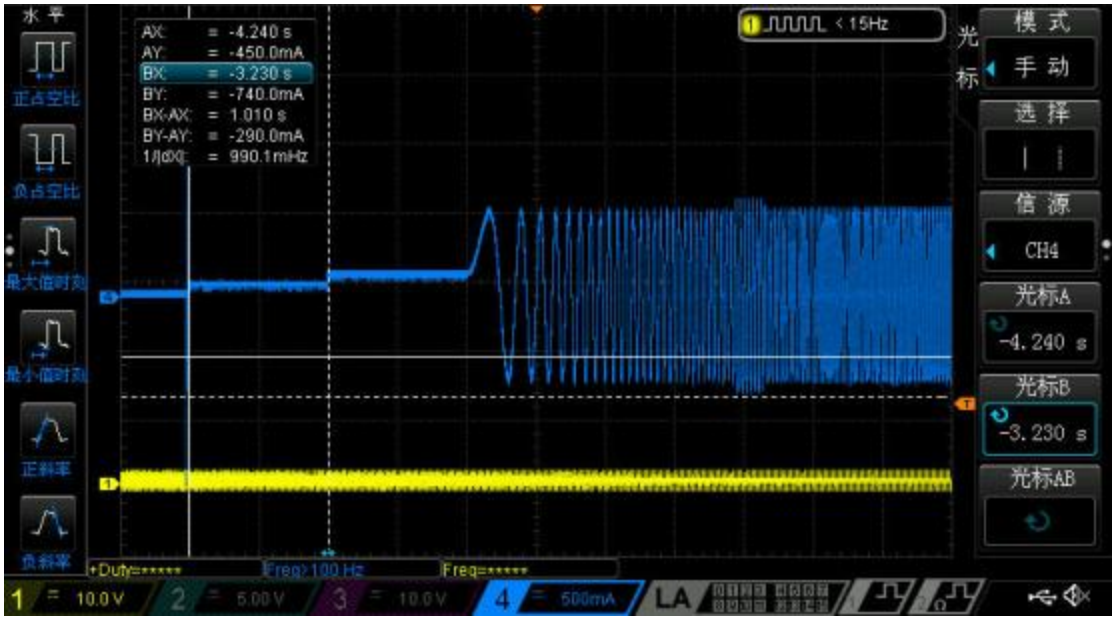


图 4-2 预定位波形

● 4.5 开环启动

```
/****************************开环参数****************************/
#define U_SVC_MIN_FREQ             (20.0)      //单位：Hz  开环拖动最终频率
#define FREQ_ACC                   (1.0)       //单位：(1/128)Hz  开环拖动频率加速调整值
#define FREQ_DEC                   (1.0)       //单位：(1/128)Hz  开环拖动频率减速调整值

#define OPEN_RUN_STATUS            (TRUE)    //开环状态  TRUE =  开环运行, FALSE =  闭环运行
#define OPEN_RUN_DELAY_TIME        (200)     //d、q 轴电流切换调整时间
#define MATCH_TIME                 (5)       //估算和给定电流匹配次数
```

开环启动的调试要结合开环电流来调试，开环电流和第二段定位电流相同，U_SVC_MIN_FREQ 要保证在此频率下，估算角度和给定角度基本一致。在调试的时候设置 OPEN_RUN_STATUS 为 TRUE，通过 FreeMASTER 或者 JScope 检测角度波形，正确的波形如图 4-3 所示，其中黄色为开环给定角度，绿色为估算角度。如果估算角度比较差，就要加大开环频率。

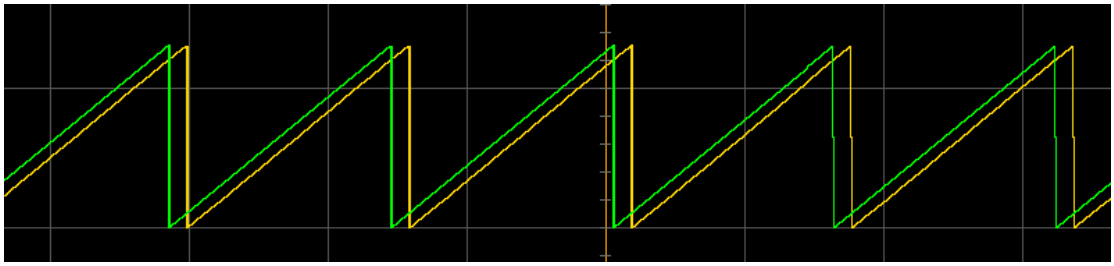


图 4-3 开环估算角度波形

对于重负载，要加大开环电流，减小开环加速度。以启动平滑顺畅为准。

对于轻负载，要加大开环加速度，尽快切到闭环运行。

注意，因为开环给的时 Id，闭环运行时 Id = 0，因此程序里设置了 Id->Iq 电流的转换，这个时候如果电流加减速太快的话，会造成转矩的波动，对于某些应用会引入噪声。因此程序里添加了 d、q 轴电流转换程序，转换的时间通过 OPEN_RUN_DELAY_TIME 来调整。设计的原理时，当满足闭环运行条件时 ，先让 Iq 增加，Id 减小（此时 IdRef 不直接给到 0，大小视电机和负载来定），然后再让 IdRef = 0；Iq 随着实际的给定来运行。一定程度上可以减小电流切换带来的转矩波动。

● 4.6 闭环运行 & 顺逆风检测

```
/*****************************环路选择*****************************/
#define CURRENT_LOOP            (0)       //电流环
#define SPEED_LOOP              (1)       //速度环
#define POWER_LOOP              (2)       //功率环
#define CLOSE_LOOP              (CURRENT_LOOP)    //环路选择
```

电流环是电流控制的核心，在调试的时候要先把电流调试好。当CLOSE_LOOP 选择 CURRENT_LOOP 时即为电流环。调整电流环 Kp、Ki，使电流环稳定，关于 PI 的整定请参考 3.2 章节的说明。

顺逆风检测和闭环运行共用一套 PI 参数，因此不用单独调试，只要设置合理的 SPEED_TRACK_ON_FREQ_THH、EBRK_ON_FREQ_THH 即可，确保在切闭环的时候没有抖动。

调整 IQ_SET 值，使最大运行转速满足需求，然后把这个值赋给速度环或者功率环输出的最大值，即 IQMAX/IQMIN 或者 POWER_IQMAX/POWER_IQMIN。外环的爬坡曲线要和负载适配，重负载爬坡速度慢，即降低相应的 ACC 和 DEC 值，加大外环调整的周期（速度环--SPEED_LOOP_CNTR，功率环--POWER_LOOP_CNTR）；轻负载则可以加大相应的 ACC 和 DEC 值，缩短外环调整的周期。

● 4.7 故障参数匹配

当电机性能调试完成，满足需求后，打开 FaultCheck()函数，按照 3.3 的说明去匹配故障检测参数，并测试故障检测的灵敏度。

● 4.8 外围功能添加

添加外围功能，如 PWM 调速、遥控等功能，建议为不同的外围功能编写独立的处理函数，避免交叉耦合，方便查找问题。

● 4.9 综合测试

在所有功能添加完成后，一定要进行多次综合测试，程序中设计的功能及故障检测都要测试，测试结果一定要和设计值相同。为了方便对比测试结果，建议使用《调试参数转换表》。

在调试过程中，只要把 User2App、App2Core、Core2App 的参数填到对应位置，就可以实现数字量和实际物理量的转换。这三组参数可以在 debug 的时候在 watch 窗口查看。示例应用如下表

User2App		
物理量	转换系数	移位系数
电流	1000	0
电压	100	0
频率	100	0
角度	10	0
转速	10	0
功率	100	0
转矩	1000	0
母线电压	100	0

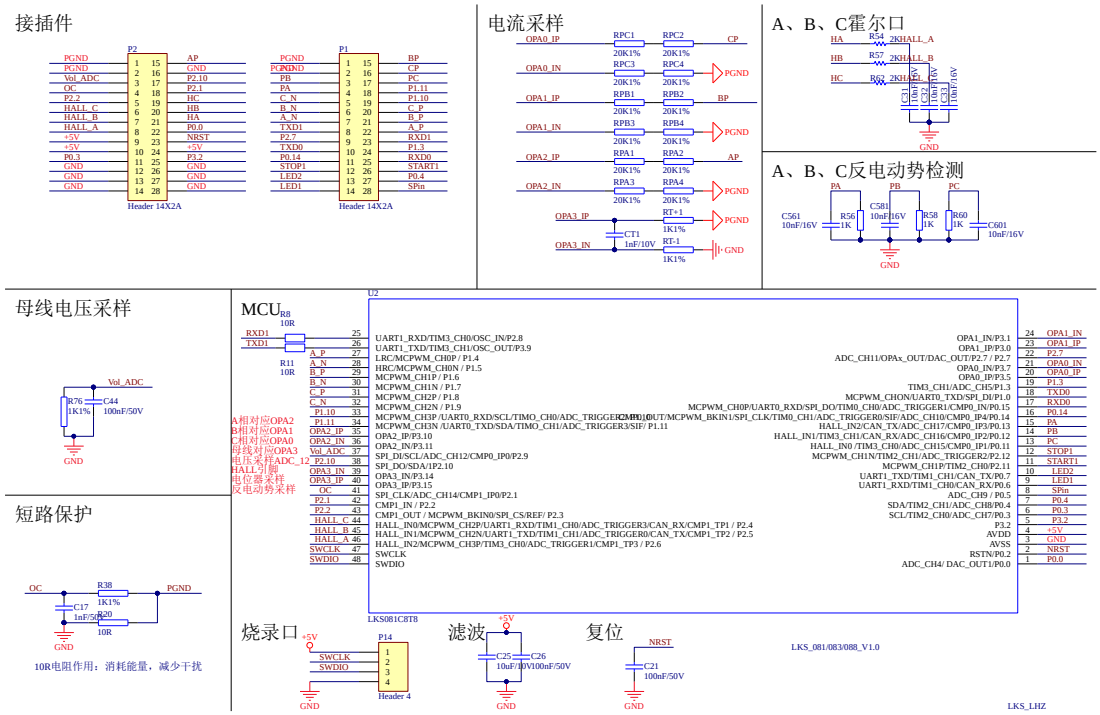
App2Core		
物理量	转换系数	移位系数
电流	5931	13
电压	24215	13
频率	13421	11
角度	9320	9
转速	0	0
功率	0	0
转矩	0	0
母线电压	13981	13

Core2App		
物理量	转换系数	移位系数
电流	11313	13
电压	2771	13
频率	1250	13
角度	450	13
转速	0	0
功率	0	0
转矩	0	0
母线电压	4800	13

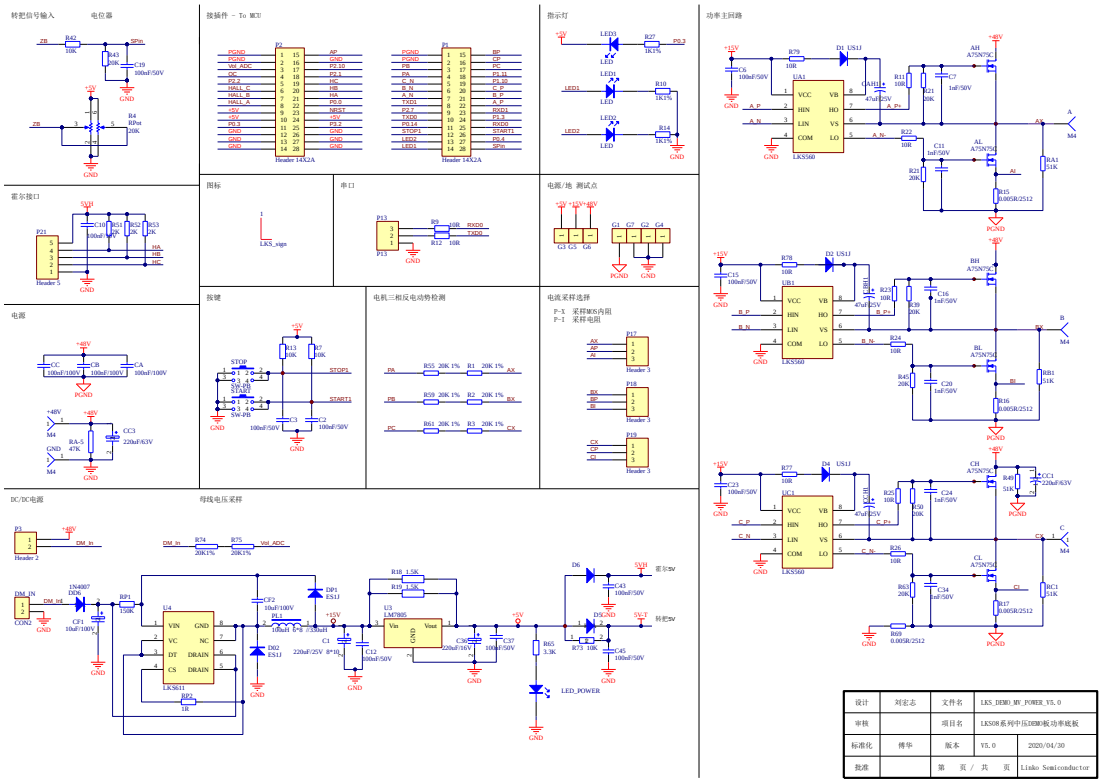
	Core	User	
电流	mCurrentVector.tFedb	Imax	移位系数
	204	0.28172	8192
电压	mVoltageVector.tRef	Vq	
	2350	7.95	8192
母线电压	aBusVolAvg	Vbus	
	2468	24.68	
频率	motorSpeed.fbk	Freq	
	68000	103.76	8192
角度	motorPolePosition.uFoc		
	222222000	18.63	8192

	User	To	Core	移位系数
I	2		1448.00	8192
V	10		2955.93	8192
F	30		19659.67	2048
Theta	60		10921.88	512
Vbus	24		4096.00	8192

● 附录 1 LKS081/083/088 原理图



MCU 控制部分原理图



驱动部分原理图

● 附录 2 MC_Parameter.h 参数说明

```
/*******************************硬件参数*******************************/
#define ADC_SUPPLY_VOLTAGE    (3.6)   //单位: V    ADC 基准电压，3.6 或者 2.4,大部分应用选择
3.6
#define AMPLIFICATION_GAIN     (17.543859649)       //运放放大倍数
#define RSHUNT                  (0.005)              //单位: Ω   采样电阻阻值
#define VOLTAGE_SHUNT_RATIO    (1.0/(20.0*2+1.0))   //母线电压分压比
#define BEMF_SHUNT_RATIO       (1.0/(20.0*2+1.0))    //反电势电压分压比

/*****************************额定参数*********************************/
#define U_RATED_VPN             (24.0)       //单位:V, 电机额定工作电压，设置为正常工作电压
#define U_RATED_CUR             (2.0)        //单位:A, 电机额定工作电流，相电流最大值
#define U_MAX_FREQ              (300.0)      //单位:Hz, 电机额定转速，    电机最高运行频率
*2

/*****************************电机参数*********************************/
#define U_MOTOR_PP              (5.0)        //电机极对数
#define U_MOTOR_RS              (2.16)       //单位: Ω 电机相电阻
```

```
#define U_MOTOR_LD              (2701.67)   //单位: uH  电机 d 轴电感
#define U_MOTOR_LQ              (2701.67)   //单位: uH  电机 q 轴电感
#define U_MOTOR_FLUX_CONST   (0.014206892)  //电机磁链常数  计算公式：
                                    //Vpp/2/sqrt(3)/(2*PI)/f，其中 Vpp 为电压峰峰值，f 为电频率

/*****************************ADC 校准函数*****************************/
#define CALIB_SAMPLES           (512)     //ADC 偏置校准次数，不用修改，如果修改，校准程
                                          //序里需要相应修改
#define OFFSET_ERROR            (3500)    //ADC 偏置误差阈值，不用修改

/*****************************Charge Parameter*************************/
#define CHARGE_TIME             (2)       //每相预充电时间，根据实际硬件参数修改

/*****************************顺逆风检测参数*************************/
#define SPEED_TRACK_DELAYTIME      (500)    //单位: ms  顺逆风检测时间
#define SPEED_TRACK_ON_FREQ_THH    (10)     //单位: Hz  顺风切闭环频率
#define EBRK_ON_FREQ_THH           (20)     //单位: Hz  逆风刹车频率

#define IPM_OFF_CUR1               (120)    //电机停止检测电流阈值，不用修改
#define IPM_OFF_CUR2               (12)     //电机停止检测电流阈值，不用修改

#define STOP_TIME          (10)     //单位：ms  电机停止检测滤波时间，根据实际负载修改
#define STOP_DELAY_TIME    (1000) //单位: ms  电机停止后延迟时间，根据实际负载修改。修改
依
                                    //据：电机在判定为停止后还在转动就加大延迟时间
#define CW                 (0)      //电机转向：顺时针
#define CCW                (1)      //电机转向：逆时针

/*************************初始位置检测参数*************************/
#define SEEK_POSITION_STATUS    (FALSE)   //初始位置检测状态  TRUE 为使能, FALSE 为不使能
#define U_START_ANGLE_COMP      (0)       //单位:度  初始位置检测补偿角度
#define IPD_PLUS_TIME_SETTING    (500)   /* 脉冲注入时间宽度设置  单位 us  */
#define IPD_WAIT_TIME_SETTING    (300)   /* 脉冲空闲等待时间宽度设置  单位 us  */
#define IPD_PLUS_TIME_WIDTH      (u32)(IPD_PLUS_TIME_SETTING*(MCU_MCLK/1000000))
                                          /* 脉冲注入时间宽度设置  单位 clk  */
#define IPD_PLUS_WAIT_TIME       (u32)(IPD_WAIT_TIME_SETTING*(MCU_MCLK/1000000))
                                          /* 脉冲空闲等待时间宽度设置  单位 clk  */
```

```
/*****************************预定位参数*****************************/
#define ALIGN_ANGLE                 (0)       //单位:度  预定位角度
#define U_START_CUR_SET_F           (0.0)     //单位: A  第一段定位电流
#define U_START_CUR_SET_S           (0.8)     //单位: A  第二段定位电流
#define DC_LOCATION_HOLDTIME_TOTAL_LENTH    (1)   //单位: ms  第一段定位时间
#define DC_LOCATION_HOLDTIME_TOTAL_LENTH_STAGE1   (1)    //单位: ms  第二段定位时间
#define DC_ALIGN_TOTAL_LENTH        (DC_LOCATION_HOLDTIME_TOTAL_LENTH +
                                    DC_LOCATION_HOLDTIME_TOTAL_LENTH_STAGE1)    //定位总时长

#define ALIGN_CURRENT_ACC           (15.0)  //单位: (1/8)A   定位电流加速调整值   初始位置检
                                    //测使能后给到最大值，不能超过 30，否则数据会溢出。
#define ALIGN_CURRENT_DEC           (15.0)     //单位: (1/8)A   定位电流减速调整值   初始位置
                                    //检测使能后给到最大值，不能超过 30，否则数据会溢
出。
/*****************************开环参数*****************************/
#define U_SVC_MIN_FREQ              (20.0)      //单位：Hz  开环拖动最终频率
#define FREQ_ACC                    (1.0)       //单位：(1/128)Hz  开环拖动频率加速调整值
#define FREQ_DEC                    (1.0)       //单位：(1/128)Hz  开环拖动频率减速调整值

#define OPEN_RUN_STATUS             (TRUE)    //开环状态  TRUE =  开环运行, FALSE =  闭环运行
#define OPEN_RUN_DELAY_TIME         (200)     //d、q 轴电流切换调整时间
#define MATCH_TIME                  (5)       //估算和给定电流匹配次数

/*****************************环路选择*****************************/
#define CURRENT_LOOP                (0)       //电流环
#define SPEED_LOOP                  (1)       //速度环
#define POWER_LOOP                  (2)       //功率环
#define CLOSE_LOOP                  (CURRENT_LOOP)    //环路选择

/*****************************电流环参数*****************************/
#define IQ_SET                      (0.3)    //单位：A IqRef，Iq 给定值
#define P_ACR_KP                    (6000)   //电流环 Kp，实际运用的 Kp 和电机参数有关
#define P_ACR_KI                    (500)    //电流环 Ki，实际运用的 Ki 和电机参数有关
#define VDMAX                       (6000)   //电流环输出最大值，不用修改
#define VDMIN                       (-6000)  //电流环输出最小值，不用修改
```

```
#define VQMAX                          (6000)   //电流环输出最大值，不用修改
#define VQMIN                          (-6000)  //电流环输出最小值，不用修改

#define PLL_KP_GAIN                    (20)     //PLL_Kp 估算器 Kp
#define PLL_KI_GAIN                    (20)     //PLL_Ki 估算器 Ki

#define TORQUE_MODE_CURRENT_CHANGE_ACC    (0.01)  //单位：A 电流加速调整值
#define TORQUE_MODE_CURRENT_CHANGE_DEC    (0.01)  //单位：A 电流减速调整值
/*****************************速度环参数************************/
#define POWER_LIMIT_STATUS          (FALSE)  //限功率状态，TRUE = 使能，FALSE = 不使能
#define POWER_LIMIT_VALUE           (10.0)  //单位：W   限制功率的大小
#define POWER_LIMIT_TIME            (5)     //单位：速度环周期，   限功率计算周期
#define POWER_LIMIT_SPEED           (10)  //单位：Hz   限功率转速给定，根据实际应用来设
置
#define SPEED_SET                      (50)     //单位：Hz   速度给定值
#define SPEED_LOOP_CNTR                (0)      //单位：ms   速度环路计算周期
#define STATE04_WAITE_TIME             (100)    //单位: ms 速度变量初始化时间
#define P_ASR_KP                       (8000)  //速度环 Kp
#define P_ASR_KI                       (1000)   //速度环 Ki
#define IQMAX                          (1.0)     //单位:A, 速度环输出最大值
#define IQMIN                          (-1.0)    //单位:A, 速度环输出最小值

#define SPEED_RUN_ACC                  (2)     //单位 (1/128)Hz 速度加速调整值
#define SPEED_RUN_DEC                  (2)    //单位 (1/128)Hz 速度减速调整值

/*****************************功率环参数************************/
#define SPPED_LIMIT_STATUS             (FALSE)  // 限转速状态，TRUE = 使能，FALSE = 不使能
#define SPEED_LIMIT_VALUE              (100.0)  //单位：Hz   限制转速的大小
#define SPEED_LIMIT_TIME               (5)     //单位：ms   功率环周期，  限转速计算周期
#define SPEED_LIMIT_POWER_VALUE        (10)    //单位：W   限转速功率给定
#define POWER_SET                   (20)          //单位：W   功率给定值
#define POWER_LOOP_CNTR             (1)           //单位：ms   功率环路计算周期
#define POWER_KP                    (6000)        //功率环 Kp
#define POWER_KI                    (600)         //功率环 Ki
#define POWER_IQMAX                 (2.0)         //单位:A, 功率环输出最大值
#define POWER_IQMIN                 (-2.0)        //单位:A, 功率环输出最小值
#define POWER_RUN_ACC               (0.1)        //单位 w 功率加速调整值
```

```
#define POWER_RUN_DEC            (0.1)       //单位 w 功率减速调整值

/**********************FaultDetection*************************/
//过流检测参数
#define I_PH_OVERCURRENT_FAULT       (2.5)       //单位：A 软件过流检测设定值

//过欠压检测参数
#define U_DCB_OVERVOLTAGE_FAULT        (28)       //单位：V 过压检测设定值
#define U_DCB_OVERVOLTAGE_RECOVER     (26)       //单位：V 过压恢复设定值
#define U_DCB_UNDERVOLTAGE_FAULT      (18)       //单位：V 欠压检测设定值
#define U_DCB_UNDERVOLTAGE_RECOVER    (20)       //单位：V 欠压恢复设定值

//离水空转参数
#define I_PH_EMPTY_FAULT              (0.3)       //单位：A 空转检测电流设定值
#define SPEED_EMPTY_FAULT             (50.0)      //单位：Hz 空转检测转速设定值

//温度检测参数
#define TEMP_FAULT                    (150)       //过温检测设定值
#define TEMP_RECOVER                  (170)       //过温恢复设定值
#define TEMP_BREAK                    (4000)      //NTC 开路设定值


//堵转检测参数
#define SPEED_STALL_MAX_FAULT         (150.0)     //单位：Hz 堵转检测转速最大值
#define SPEED_STALL_MIN_FAULT         (10.0)      //单位：Hz 堵转检测转速最小值

#define I_PH_STALL_FAULT              (2.5)       //单位：A 堵转检测电流设定值

#define SPEED_STALL_FAULT             (20.0)      //单位：Hz 堵转检测转速设定值
#define IQ_STALL_FAULT                (0.2)       //单位：A 堵转检测电流设定值

//二次启动检测参数
#define START_TIME_FAULT              (200)     //单位：5ms 开环之后 1s 内还不进入闭环就重
                                                //启，1s 这个时间根据实际应用调整

//缺相检测参数
#define I_PHASE_LOSS_FAULT            (2000)    //数字量，根据实际计算修改
```

```
//故障恢复时间
#define VOLT_FAULT_RECOVER_TIME        (2000)   //单位：ms   过欠压恢复时间
#define CURRENT_FAULT_RECOVER_TIME     (2000)   //单位：ms   过流恢复时间
#define STALL_FAULT_RECOVER_TIME       (2000)   //单位：ms   堵转恢复时间
#define PHASELOSS_FAULT_RECOVER_TIME   (2000)   //单位：ms   缺相恢复时间
#define TEMP_FAULT_RECOVER_TIME        (2000)   //单位：ms   过温恢复时间
#define START_FAULT_RECOVER_TIME        (500)   //单位：ms   二次启动恢复时间
#define EMPTY_FAULT_RECOVER_TIME       (2000)   //单位：ms   离水空转恢复时间
```