

# 03x 和 07x 新平台低压 Demo 板配套程序说明

版本: v1.0.0



@ 2024, 版权归凌鸥创芯所有

机密文件，未经许可不可扩散

## 目录

1. 文档版本.....	4
2. 新平台概述.....	5
2.1 主要特性.....	5
2.2 工程文件.....	5
2.3 软件介绍.....	5
2.4 硬件介绍.....	6
3. 电机启动一般调试步骤.....	7
3.1 程序配置.....	7
3.1.1 电机参数配置.....	7
3.1.2 芯片参数匹配.....	7
3.1.3 GPIO 配置 .....	9
3.1.4 硬件过流配置.....	10
3.1.5 ADC 采样配置 .....	11
3.1.6 双电阻采样通道配置与修改.....	13
3.2 程序基本调试.....	15
3.2.1 MCPWM 通路测试.....	15
3.2.2 强拖启动.....	16
3.2.3 电流环.....	17
3.2.4 速度环.....	17
4. LKScope 简介 .....	19
4.1.1 电机启停、转速调节.....	19
4.1.2 保护标志，如过流、过压等保护 .....	20
4.1.3 系统状态机查看 .....	21
4.1.4 占空比与采样.....	22
4.1.5 观测器的输出.....	22
4.1.6 速度反馈与 PI 调节 .....	23
4.1.7 电流反馈与 PI 调节 .....	23
4.1.8 算法状态变量.....	23

5. 故障保护与恢复.....	25
6. 参考资料.....	26

# 1. 文档版本

本文档是针对 LKS32MC03x 和 LKS32MC07x 系列芯片无感 FOC 无代码开发平台的调试说明手册，主要包含

用户可以参考该手册了解相关参数的具体配置细节和调试开发流程，实际的软件若有迭代更新，以实际的软件为主。

文件说明	03x 和 07x 新平台低压 Demo 板配套程序说明
日期	2024.10.15
适用 MCU	LKS32MC07x、LKS32MC03x
控制模式	电流/速度/功率环路
采样模式	双电阻
有感/无感	无感
调试工具	LKSscope1.38 版本
编制	WangZhuangzhuang

表 1-1：信息说明

## 2. 新平台概述

### 2.1 主要特性

新算法软件平台是凌鸥提供的针对无感 foc 控制开发的平台，结合相关编译工具(keil)和调试工具(lksscope)及其工程例程，可以很好上手开发电机控制相关项目，可以基本适配以下功能：

- 顺逆风启动
- SLVC 启动
- 初始位置检测
- 电流环、速度环、功率环控制
- 单、双、三电阻拓扑结构采样
- 硬/软件过流、过压保护机制

### 2.2 工程文件

新平台算法提供的软件包中，针对 03 和 07 系列，提供了无感驱、6N 和 3P3N 基本的 Demo 例程和 LKSScope 配置工程，方便后续开发和使用。同时该系列的其它 MCU 也可参考，进行快速电机启动。Demo 例程如图 2.1 所示：

1_LKS32MC037	2024/10/10 15:55	文件夹
2_LKS32MC034DO	2024/10/14 15:16	文件夹
3_LKS32MC034E	2024/10/12 14:53	文件夹
4_LKS32MC071	2024/9/27 9:30	文件夹
5_LKS32MC074D0	2024/10/11 9:18	文件夹
6_LKS32MC077E	2024/10/12 11:26	文件夹

图 2.1 LKSMC3203x 和 07x Demo 例程

### 2.3 软件介绍

调试软件由 2 个部分组成：**KEIL 工程+lksscope**

- 下位机 KEIL 工程：程序编写，算法控制-无感 FOC 算法/硬件初始化/保护判断等
- 上位机调试软件 lksscope：用于相关变量的在线调试工具

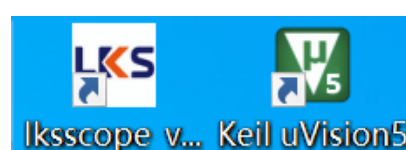


图 2.2 软件工具

## 2.4 硬件介绍

凌鸥针对不同的 MCU 系统都提供了对应的 Demo 板和功率应用板，并提供了原理图和 PCB 布线图。

有关详细信息，请参见凌鸥官方网站 <https://www.lksmcu.com/index.php/DemoBoard/>

### 3. 电机启动一般调试步骤

#### 3.1 程序配置

##### 3.1.1 电机参数配置

电机参数主要保护电阻、电感和极对数。在 MC\_Parameter\_M0.h 内配置。

```
59 #define U_MOTOR_PP_M0 (5.0) /* 电机极对数 */
60 #define U_MOTOR_RS_M0 (0.45) /* 单位: Ω 电机相电阻 */
61 #define U_MOTOR_LD_M0 (234.4485)/(8000.0) /* 单位: uH 电机d轴电感 */
62 #define U_MOTOR_LQ_M0 (343.893)/(10000.0) /* 单位: uH 电机q轴电感 */
```

图 3.1 Demo 电机参数

新平台 VF 启动和顺逆风对磁链精度要求很高（来源李同华李工新平台讲解），测试方法见 FOC 参数调试说明 PDF。

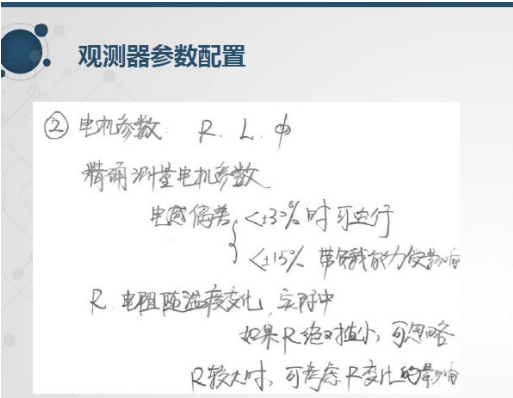


图 3.2 电机参数磁链参数影响

```
/* 电机磁链常数 计算公式: Vpp/2/sqrt(3)/(2*PI)/f, 其中Vpp为电压峰值, f为电频率 3.68V 25Hz
此参数仅影响顺逆风启动的速度检测, 角度估算不使用些参数 */
#define U_MOTOR_FLUX_CONST (0.00687364598)
```

图 3.3 Demo 电机磁链参数设置

反电势常数:

$$k_e = \frac{V_{pp}}{2\sqrt{3} * f}$$

磁链常数:

$$\varphi = \frac{k_e}{2 * \pi i}$$

##### 3.1.2 芯片参数匹配

以 LKS071CBT8 Dem 板+底板预驱 LKS560 为例, LKS560 下管低有效, 程序中 (hardware\_config.h)选择 LKS32MC071 芯片, 预驱极性设置为 P\_HIGH\_\_N\_LOW, 如图 3.4

所示。

```
42 #define P_HIGH__N_HIGH 0 /* 预驱动极性设置 上管高电平有效，下管高电平有效 */
43 #define P_HIGH__N_LOW 1 /* 预驱动极性设置 上管高电平有效，下管低电平有效 */
44
45 #define CHIP_PART_NUMBER LKS32MC071 /* 芯片型号选择，选择不正确将影响芯片模块的初始化 */
46
47 #if ((CHIP_PART_NUMBER == LKS32MC074D) || (CHIP_PART_NUMBER == LKS32MC076D) || (CHIP_PART_NUMBER == LKS32MC077D))
48 #define MCPWM_SWAP_FUNCTION 1
49 #define PRE_DRIVER_POLARITY P_HIGH__N_HIGH /* 预驱动极性设置 上管高电平有效，下管高电平有效 */
50 #else
51 #define PRE_DRIVER_POLARITY P_HIGH__N_LOW /* 预驱动极性设置 上管高电平有效，下管低电平有效 */
52 #endif
53
54 /* 应用NUM 继续定义 */
```

图 3.4 驱动配置宏定义

集成预驱的芯片，如 6N 和 3P3N，预驱下管输出高有效，修改芯片型号的同时，预驱极性设置为 P\_HIGH\_\_HIGH。下图 3.5 为 LKS32MC074D，同理其它含预驱芯片。

```
45 #define CHIP_PART_NUMBER LKS32MC074D /* 芯片型号选择，选择不正确将影响芯片模块的初始化 */
46
47 #if ((CHIP_PART_NUMBER == LKS32MC074D) || (CHIP_PART_NUMBER == LKS32MC076D) || (CHIP_PART_NUMBER == LKS32MC077D))
48 #define MCPWM_SWAP_FUNCTION 1
49 #define PRE_DRIVER_POLARITY P_HIGH__N_HIGH /* 预驱动极性设置 上管高电平有效，下管高电平有效 */
50 #else
51 #define PRE_DRIVER_POLARITY P_HIGH__N_LOW /* 预驱动极性设置 上管高电平有效，下管低电平有效 */
52 #endif
```

图 3.5 LKS32MC074D 预驱配置

```
885
886 #ifdef MCPWM_SWAP_FUNCTION
887     MCPWM0_PRT = 0x0000DEAD; /*enter password to unlock write protection */
888     PWM_SWAP = 0x67;
889 #endif
890
```

图 3.6 自带预驱的芯片需要修改 PWM\_SWAP 的值（根据数据手册修改）

同时新平台程序为方便与 LKScope 进行通信，将重要结构体如 STR\_DrvCfgGlobal 和 STR\_DrvCfgPara 在 flash 内使用 \_\_attribute\_\_((at())) 进行了绝对定位（图 3.7），因此需要根据不同芯片 flash 大小进行修改绝对定位位置。128Kflash 地址位置 ADDR\_PARA\_LOCATION 设置为 0x1FE00（图 3.8）；64Kflash 地址位置 ADDR\_PARA\_LOCATION 设置为 0xFE00（图 3.9）；32Kflash 地址位置 ADDR\_PARA\_LOCATION 设置为 0x7E00（图 3.10）。同时 Keil 上（魔法棒按钮）flash 编译位置也需要修改，0x200 为结构体预留 512 字节进行存储（图 3.11）。

```
27
28 const STR_DrvCfgPara tS_DrvCfgPara[MAX_CFG_PARA_ID] __attribute__((at(ADDR_PARA_START_LOCATION))) = {
29     DRV_CFG_PARA_TAB_M(0)
30 };
31
32 // 全局配置参数通过编译器伪指令放置在指定的FLASH空间
33 volatile const STR_DrvCfgGlobal tS_DrvCfgGlobal __attribute__((at(ADDR_PARA_LOCATION))) = DRV_CFG_GLOBAL_PARA_TAB;
34
35
```

图 3.7 \_\_attribute\_\_((at()))flash 分配地址

```
218
219 #define ADDR_PARA_LOCATION (0x1FE00)
220 #define ADDR_PARA_START_LOCATION (ADDR_PARA_LOCATION + sizeof(STR_DrvCfgGlobal))
221
```

图 3.8 128Kflash 预分配地址位置



```

219 // #define ADDR_PARA_LOCATION (0x1FE00)
220 #define ADDR_PARA_LOCATION (0xFE00)
221 #define ADDR_PARA_START_LOCATION (ADDR_PARA_LOCATION + sizeof(STR_DrvCfgGlobal))
222

```

图 3.9 64Kflash 预分配地址位置

```

212
213 #define ADDR_PARA_LOCATION (0x7E00)
214 #define ADDR_PARA_START_LOCATION (ADDR_PARA_LOCATION + sizeof(STR_DrvCfgGlobal))
215

```

图 3.10 03 系列 32Kflash 预分配地址位置

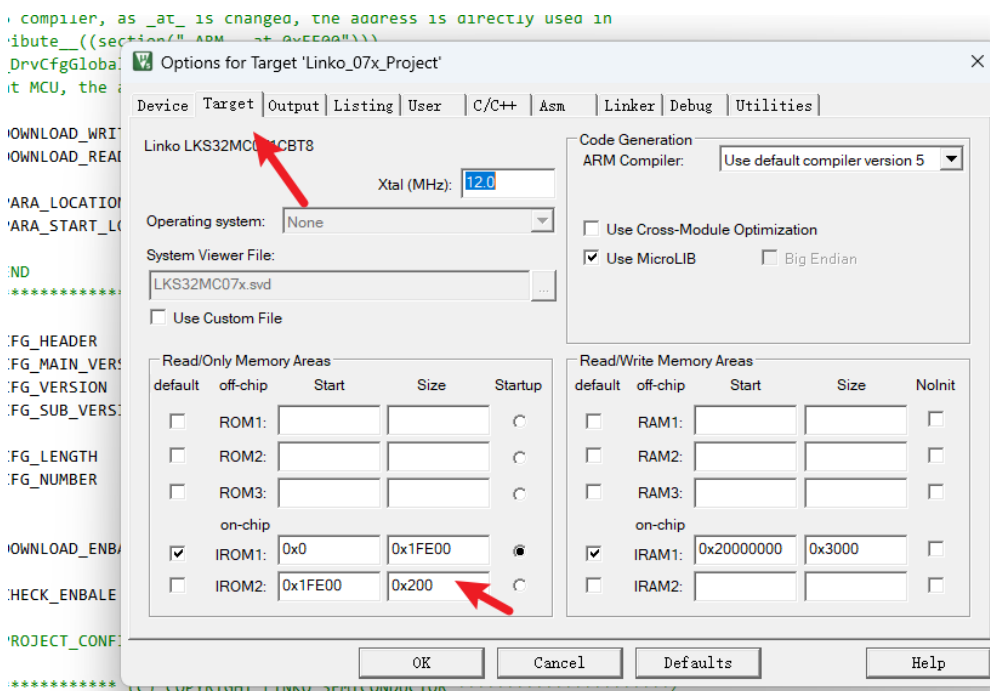


图 3.11 Keil 软件上修改

### 3.1.3 GPIO 配置

根据所使用 MCU 控制 PWM 输出的引脚，配置 GPIO，如 LKS32037 芯片，GPIO0.10~GPIO0.15 控制 PWM 输出，配置方式如图 3.12 所示。

```

GPIO_InitTypeDef GPIO_InitStructure;
GPIO_StructInit(&GPIO_InitStructure);

/* MCPWM P1.4~P1.9 */
GPIO_InitStructure.GPIO_Mode = GPIO_Mode_OUT;
GPIO_InitStructure.GPIO_Pin = GPIO_Pin_10 | GPIO_Pin_11 | GPIO_Pin_12 | GPIO_Pin_13 | GPIO_Pin_14 | GPIO_Pin_15;
GPIO_Init(GPIO0, &GPIO_InitStructure);
GPIO_PinAFConfig(GPIO0, GPIO_PinSource_10, AF3_MCPWM);
GPIO_PinAFConfig(GPIO0, GPIO_PinSource_11, AF3_MCPWM);
GPIO_PinAFConfig(GPIO0, GPIO_PinSource_12, AF3_MCPWM);
GPIO_PinAFConfig(GPIO0, GPIO_PinSource_13, AF3_MCPWM);
GPIO_PinAFConfig(GPIO0, GPIO_PinSource_14, AF3_MCPWM);
GPIO_PinAFConfig(GPIO0, GPIO_PinSource_15, AF3_MCPWM);

```

图 3.12 GPIO 配置

### 3.1.4 硬件过流配置

目前 Demo 板上硬件过流主要方式：使用比较器（CMP）产生过流信号，触发 MCPWM 的 Fail 事件，以关断 MCPWM 输出。比较器有两个输入，正端是 Demo 板上 OC 引脚；负端是 DAC 的输出。

进入 HD 层文件夹下的 hardware\_init.c 函数，有 DAC\_init()、CMP\_init()和 MCPWM\_init()三个函数，对其进行配置。

DAC\_init() 函数如图 3.12 所示，这里 wDACCm 由 MC\_Parameter\_M0.h 内的 I\_PH\_HARD\_OC\_FAULT\_CMP\_VOLT\_M0 传递而来，通过修改该值大小，控制过流保护。07x 和 03 系列功率板的过流保护电阻大小为  $0.005\ \Omega$ ，如希望设置最大过流值为 3A，则 I\_PH\_HARD\_OC\_FAULT\_CMP\_VOLT\_M0 可设置为 15mV。

```
328 |
329 |     wDACCmp      = (wDACCmp * 4096)/nDACRef;
330 |     wDACCmp      = sat(wDACCmp,0,4095);
331 |
332 |     //设置DAC的数值
333 |     DAC_OutputValue(DAC_Channel_0, wDACCmp) ;    /* 根据母线采样电阻计算保护值*/
334 |
335 |     DAC_Cmd(DAC_Channel_0, ENABLE);              /*使能DAC时钟*/
336 | }
```

图 3.13 DAC\_init()函数内

CMP\_init()函数如图 3.13 所示，LKSMC3207CBT8 DEMO 板 OC 引脚使用的是 CMP1，其它 DEMO 板要根据原理图上过流引脚设置。

```
5 |
7 |     CMP_InitStruct.CMP1_EN      = ENABLE;        /* 比较器1开关 */
8 |     CMP_InitStruct.CMP1_SELN     = SELN_DAC;      /* CMP1_N 内部参考 */
9 |     CMP_InitStruct.CMP1_SELP     = SELP_CMP_IP0;   /* CMP1_P CMP1_OPA3 */
10 |    CMP_InitStruct.CMP1_IN_EN     = DISABLE;       /* 比较器信号输入使能 */
11 |    CMP_InitStruct.CMP1_IE        = DISABLE;       /* 比较器1信号中断使能 */
```

图 3.14 CMP\_init()函数配置

MCPWM 有 FAIL0 和 FAIL1 两路信号，并列关系。都配置时，谁先发生，都会触发 MCPWM 输出默认信号。MCPWM\_init()内 FAIL 事件配置如图 3.14，FAIL 事件中断默认关闭。

```
MCPWM_InitStructure.FAIL0_INT_EN = DISABLE;          /* FAIL0事件 中断使能或关闭 */
MCPWM_InitStructure.FAIL0_INPUT_EN = ENABLE;         /* FAIL0通道急停功能打开或关闭 */
MCPWM_InitStructure.FAIL0_Signal_Sel = FAIL_SEL_CMP; /* FAIL0事件信号选择，比较器或IO口 */
MCPWM_InitStructure.FAIL0_Polarity = HIGH_LEVEL_ACTIVE; /* FAIL0事件极性选择，高有效 */
```

图 3.15 MCPWM 内 FAIL 事件配置

### 3.1.5 ADC 采样配置

ADC 采样主要采集相电流、反电动势和温度等参数。采样相电流时候需要配置放大倍数、采样电阻大小、母线电压和反电动势电压分压比（图 3.15），在 MC\_Parameter\_M0.h 内。放大倍数参考用户手册和原理图，分压比参考 DEMO 板的原理图。

最大采用电流计算：

$$I_{max} = \frac{ADC_{\text{量程}}}{Gain_{\text{放大倍数}} * R_{\text{采样电阻}}}$$

```
9 //200/(10+20*2) = 4;  
1 //200/(10+0.2) = 19.6078  
2 #define AMPLIFICATION_GAIN_M0 (19.608) /*运放放大倍数*/ // 3.6/(19.608 * 0.005)= 36.72  
3 #define RSHUNT_M0 (0.005) /* 单位：Ω 采样电阻阻值 */  
4 #if 1  
5 #define VOLTAGE_SHUNT_RATIO_M0 (1.0/(20.0 * 2 + 1.0)) /* 母线电压分压比 */  
5 #define BEMF_SHUNT_RATIO_M0 (1.0/(20.0 * 2 + 1.0)) /* 反电动势电压分压比 */  
7 #else
```

图 3.16 放大倍数与分压比配置

### ADC 配置注意事项

#### 注意事项 1:

直接进行 VF 启动时（没加负载），相电流波形有杂波，发现放大器的选择电阻和相电流采样电阻影响很大。记录如下，以供后续参考：

配置采样电阻 0.005 Ω 电阻，放大器外围选用 20k+20k 电阻，放大倍数 200/(10.2+20\*2)=3.98，ADC 差不多 0.16V，量程 3.6v 波形如下。

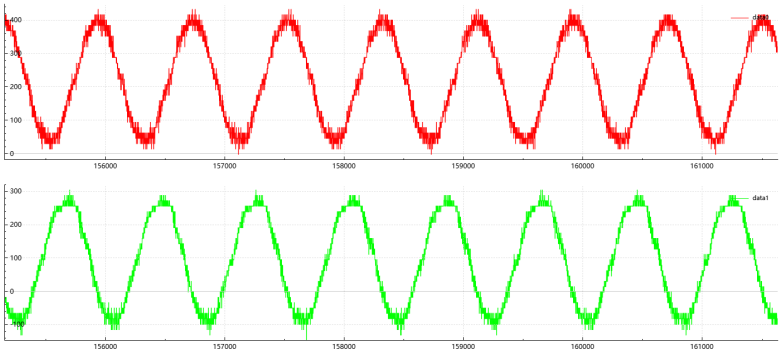


图 3.17 相电流波形

配置采样电阻 0.005 电阻，放大器选用 1k+1k，放大倍数 16.129，ADC 差不多 0.16V，量程 3.6V 即相比于第一个配置提高了四倍，波形如下，相比上一个波形干扰小很多。

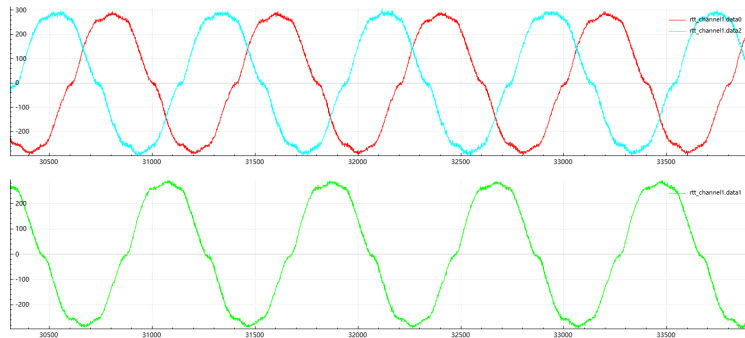


图 3.18 相电流波形

配置采样电阻 0.05 电阻，放大器选用 20k+20K，放大倍数 3.98，，ADC 差不多 0.4，量程 3.6V，即相比于第一个配置提高了⑩倍波形如下，干扰小更多。

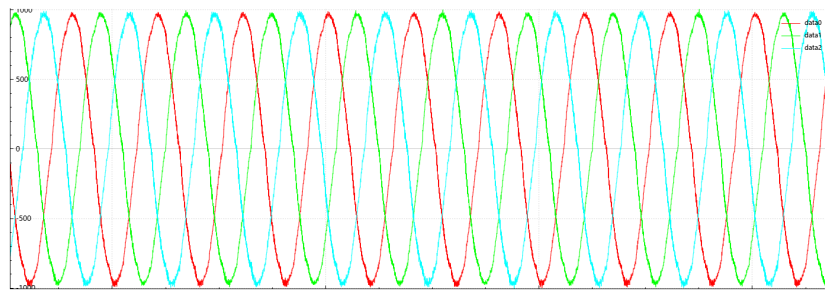


图 3.19 相电流波形

#### 注意事项 2:

（LKS32034DO Demo 板现象，其它芯片未发现）图 3.19 为使用 LKS32034DO Demo 板 ADC 采集运算放大器数据，未接底板，采集的波动值变换幅度大。

此时运放外围放置 2 个 20k 电阻，运放内部放大比例为 200k:10k。

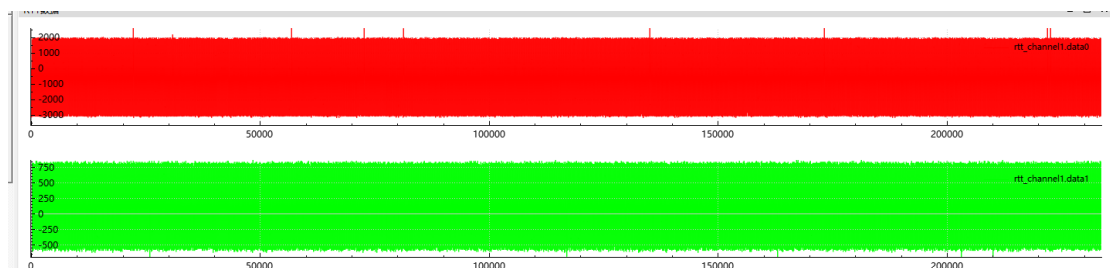


图 3.20 034DO 芯片 ADC 采样波动大

分析步骤：将图 3.20 内①和②处进行短接，波动很小，正常范围，排除②初到芯片引脚以及芯片的问题。



图 3.21 034DO Demo 板

通过将 2N2P 引脚短接（1P1N 短接）波动依然很大。干扰源在排针到②之间。将 20k 电阻更改为 1k，波动很大。换成 100Ω 后，ADC 采样波动正常。

原因：干扰源在排针到②之间，可能为电荷干扰，电阻越大，ADC 采集电压信号，干扰越大。

### 3.1.6 双电阻采样通道配置与修改

LKS3207x 和 LKS3203x 不带域驱，正常默认程序配置就行。以 LKS32034DO 为例介绍双电阻采样通道修改。

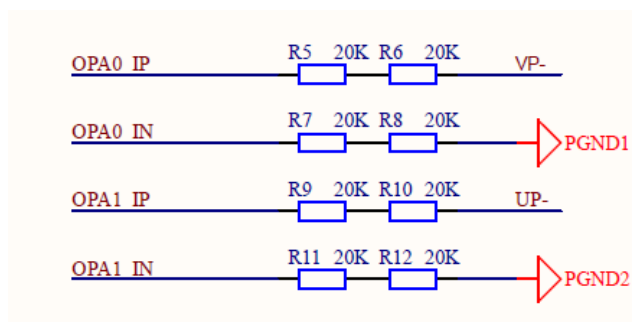


图 3.22 原理图中相电流采样

```
55 void ADC_2Shunt_NormalModeCFG(void)
56 {
57     // ADC_CHN0 = ADC0_CURRETN_A_CHANNEL | (ADC0_CURRETN_B_CHANNEL << 4) | (BEMF_CH_A << 8) | (BEMF_CH_B << 12);
58     // ADC_CHN1 = ADC0_CURRETN_A_CHANNEL | (BEMF_CH_C << 4) | (ADC0_SPIN_CHANNEL << 8);
59     ADC_CHN0 = ADC0_CURRETN_A_CHANNEL | (BEMF_CH_A << 4) | (BEMF_CH_B << 8) | (ADC0_CURRETN_B_CHANNEL << 12);
60     ADC_CHN1 = ADC0_CURRETN_A_CHANNEL | (BEMF_CH_C << 4) | (ADC0_SPIN_CHANNEL << 8);
61     MCPWM_TMR0 = (s16)(100-PWM_PERIOD_M0); // 100-PWM_PERIOD
62 }
63
```

图 3.23 ADC 采样顺序

```
void SVPWM_2ShuntGetPhaseIaIb(PSTR_CurSamplingRes this)
{
    this->pAdcRes->nSampDatU = (s32)(this->pAdcRes->nPhaseUOffset) - this->pAdcRes->nSampCurDat0;
    this->pAdcRes->nSampDatV = (s32)(this->pAdcRes->nPhaseVOffset) - this->pAdcRes->nSampCurDat1;
    this->pAdcRes->nSampDatW = -(this->pAdcRes->nSampDatU + this->pAdcRes->nSampDatV);
}
```

图 3.24 ADC 采集数据赋值给相应相位

首先根据 Demo 板原理图配置程序上 OPA 对应的相位（U，V 和 W）。（图 3.21）原理

图中 OPA1 对应 U 相，OPA0 对应 V 相。（图 3.22）程序中 ADC 第一次采样 OPA1 值，第四次采样 OPA0 值，因此将 ADC\_DAT0 给 U 相，ADC\_DAT3 给 V 相（图 3.23）。

```
297 | *****
298 void MCPWM0_RegUpdate(STR_PubDatMDHD* tS_pMdToHd)
299 |{
300 |    MCPWM_TH00 = tS_pMdToHd->mMcPwm.nuTimePhCN;
301 |    MCPWM_TH01 = tS_pMdToHd->mMcPwm.nuTimePhC;
302 |
303 |    MCPWM_TH10 = tS_pMdToHd->mMcPwm.nuTimePhBN;
304 |    MCPWM_TH11 = tS_pMdToHd->mMcPwm.nuTimePhB;
305 |
306 |    MCPWM_TH20 = tS_pMdToHd->mMcPwm.nuTimePhAN;
307 |    MCPWM_TH21 = tS_pMdToHd->mMcPwm.nuTimePhA;
308 |
309 |    MCPWM_TMR0 = tS_pMdToHd->mMcTmr.nuPoint0;    // 第一个采样点
310 |    MCPWM_TMR1 = tS_pMdToHd->mMcTmr.nuPoint1;    // 第二个采样点
311 |}
```

图 3.25 PWM 占空比输出

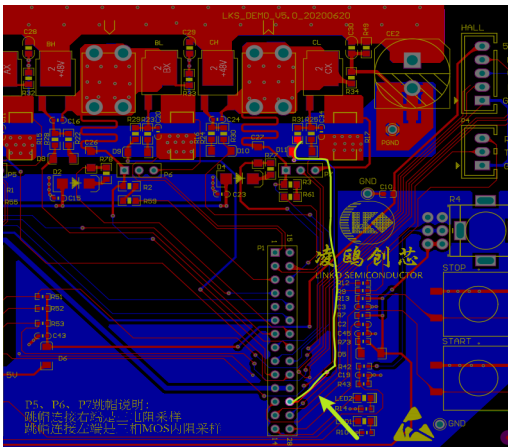


图 3.26 6N 底板 PCB 布线

图 3.24 为占空比的输入，MCPWM\_TH00、MCPWM\_TH01、MCPWM\_TH10、MCPWM\_TH11、MCPWM\_TH21 和 MCPWM\_TH21 等六个值与 nuTimePhAN、nuTimePhA、nuTimePhBN、nuTimePhBN、nuTimePhCN 和 nuTimePhCN 等六个值对应。034DO 中 LO1 对应 MCPWM\_TH00，而在图 3.25 布线中 LO1 对应 W 相也就是 C 相。因此 MCPWM\_TH00 = nuTimePhCN，MCPWM\_TH01 = nuTimePhC，同理修改其它两相。

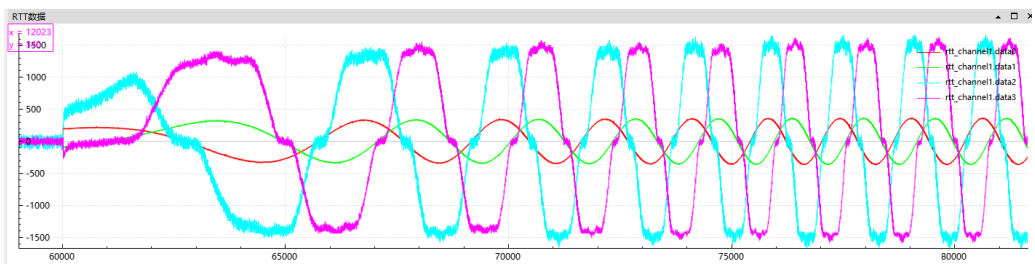


图 3.27 RTT 输出 Ua、Ub、Ia 和 Ib

如果相位配置配置正确，U 相电流（青色）与 U 相电压（红色）相位一样，V 相电流

(紫色) 与 V 相电压 (绿色) 相位一样, 如图 3.26 所示。

## 3.2 程序基本调试

### 3.2.1 MCPWM 通路测试

MCPWM 配置可参考 3.1.2 部分。用户可以通过调用 `DebugPWM_OutputFunction()`;或者将 `DEBUG_PWM_OUTPUT` 宏打开, 来测试硬件 MCPWM 通路, 将 U, V, W 设置成不同的占空比, 比如 50%, 25%, 12.5%。通过示波器, 测量 u, v, 三相正常输出对应的占空比, 验证 MCPWM 通路, 如图 3.28 所示。077E 和 037EDemo 板供电 24V 不接功率板, 测量的互补波形不太好, 接上后输出正常。

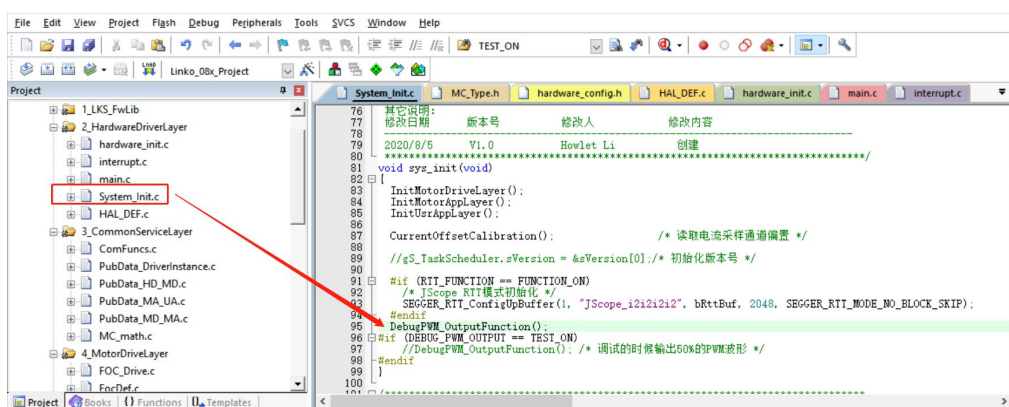


图 3.28 打开 `DebugPWM_OutputFunction()`

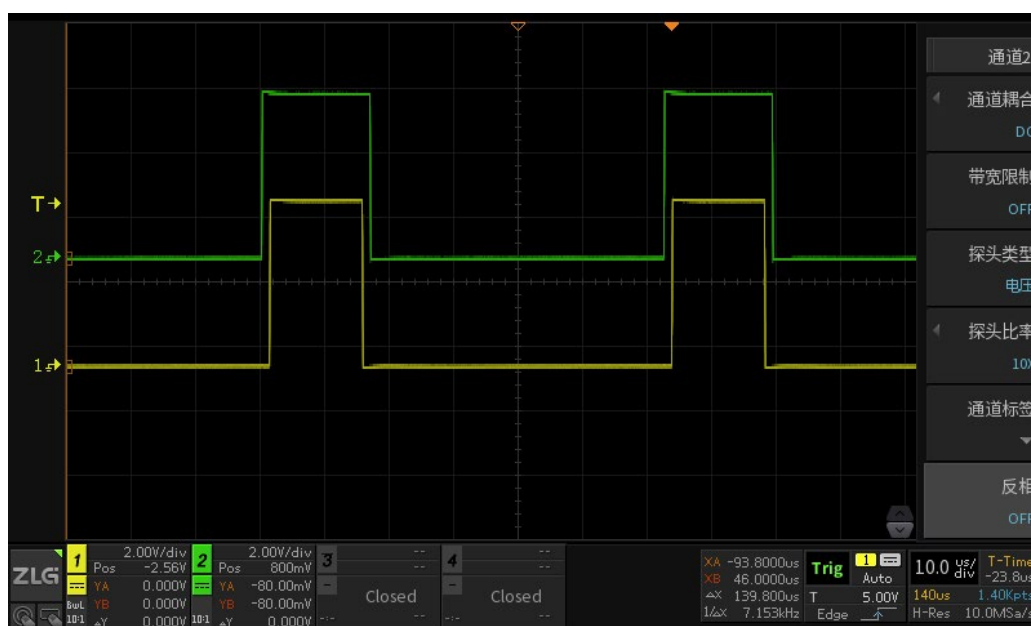


图 3.29 Debug 模式下 U 相占空比波形

下面测试短路保护是否正常。在该情况下, 以前文提到, 过流保护电阻设置为  $0.005\ \Omega$ ,



希望保护电流为 3A，则设置 I\_PH\_HARD\_OC\_FAULT\_CMP\_VOLT\_M0 为 15mV。通过给 OC 引脚外接电源，从 5mV 逐渐增大，当增大到 15mV 左右时，图 3.28 恢复成默认状态，则短路保护设置正确。

3.2.2 强拖启动

新平台中，没有 VF 和 IF 启动宏定义，是根据设定频率、电流闭环频率和开环拖动最终频率三者的关系决定。如电流环路设置为速度环，当设定频率<电流闭环频率（ CURRENTLOOP\_CLOSE\_FREQ\_M0 ） < 开 环 拖 动 最 终 频 率（OPEN\_ANGLE\_TAG\_FREQ\_M0），为 VF 启动；当电流闭环频率< 设定频率<开环拖动最终频率，进入 IF 启动；当电流闭环频率<开环拖动最终频率< 设定频率，进入 VF 启动，程序如图 3.29 所示。

VF 启动时，Ud 和 Uq 参数根据 ID\_START\_M0 和 IQ\_START\_M0 自动计算，根据启动情况用户自行调整这两个启动电流值。

```
118
119 /*-----开环参数-----*/
120 #define OPEN_ANGLE_TAG_FREQ_M0 (30.0) /* 单位：Hz 开环拖动最终频率 */
121 #define FREQ_ACC_M0 (20.0) /* 单位：Hz/s 开环拖动频率加速调整值 */
122 #define FREQ_DEC_M0 (20.0) /* 单位：Hz/s 开环拖动频率减速调整值 */
123
124 #define MATCH_TIME_M0 (5) /* 估算和给定电流匹配次数，当前未用，预留 */
125
126 #define MIN_RUN_FREQ_M0 (30.0) /* 单位：Hz 最低速度 */
127 #define CLOSE2OPEN_FREQ_M0 (2.0) /* 单位：Hz 切换为开环拖动频率 */
128 #define CURRENTLOOP_CLOSE_FREQ_M0 (20.0) /* 单位：Hz 电流闭环频率 */
129
```

图 3.30 开环参数设置

在 VF 启动下，判断 ADC 采样的相电流是否准确，如果采样正确，可以将电流闭环频率设置为 0，直接 IF 强拖启动。

使用 LKScope 的 RTT 数据输出 ADC 采样电流，与电流钳读取的电流值进行比较。

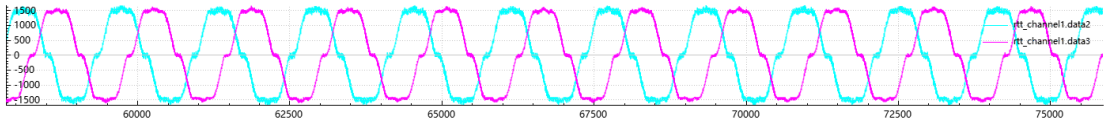


图 3.31 LKScope 读取相电流





图 3.32 示波器读取 U 相电流

通过图 3.30U 相（青色）电流值为： $1550/32767 \times I_{max} = 1550/32767 \times 36.72 = 1.73A$ 。图 3.31 中示波器读取值为 1.74A，近似相等，证明 ADC 和运算放大器配置没问题。I<sub>max</sub> 为最大采样值。

### 3.2.3 电流环

将环路设置为电流环（图 3.32），如果参数 P\_CURRENT\_KP\_M0（图 3.33）设置为 0，则采用自动计算电流环，不要手动调试；如果设置其它值，则以设置的值作为 K<sub>p</sub>。电流环需要带载调试。

```

107  /*-----环路选择-----*/
108  /*-----环路选择-----*/
109
110  #define CLOSE_LOOP_M0          (CURRENT_LOOP) // 环路选择
111

```

图 3.33 环路选择

```

163  /*-----电流环参数-----*/
164  /*-----电流环参数-----*/
165  #define IQ_SET_M0              (1.0)      /* 单位：A IqRef, Iq给定值 */
166
167  #define VQMAX_M0              (0.8)      /* Q轴最大输出限制*/
168  #define VQMIN_M0              (-0.8)     /* Q轴最小输出限制*/
169
170  #define VDMAX_M0              (0.8)      /* D轴最大输出限制*/
171  #define VDMIN_M0              (-0.8)     /* D轴最小输出限制*/
172
173  #define P_CURRENT_KP_M0       (0)        /* 电流环Kp, 实际运用的Kp会根据这个值和电机参数计算出最终的Kp */
174  /* 0 --- 自动计算参数 非0 --- 设定值*/
175  #define P_CURRENT_KI_M0       (1500)     /* 电流环Ki, 实际运用的Kp会根据这个值和电机参数计算出最终的Ki */
176
177  #define P_CURRENT_KP_RUN_M0   (1500)     /* 运行时的电流环Kp*/ /* 当前未用, 可用于动态调整PI参数 */
178  #define P_CURRENT_KI_RUN_M0   (780)      /* 运行时的电流环Ki*/ /* 当前未用, 可用于动态调整PI参数 */
179
180  #define AUTO_FW_LIM_M0        (0.00)     /* 自动弱磁D轴电流限制 A*/
181
182  #define D_CURRENT_REF_FIL_TIME_M0 (2.0)  /* 单位：ms D轴电流指令滤波时间常数 */
183  #define Q_CURRENT_REF_FIL_TIME_M0 (2.0)  /* 单位：ms Q轴电流指令滤波时间常数 */
184

```

图 3.34 电流环参数

### 3.2.4 速度环

速度环 KP 和 KI 的计算用户可在文件中使能自动计算功能如下，关闭 PI 参数自动，可在参数配置中手动设定调节。

```

1475  #if (DRV0_CLOSE_LOOP == SPEED_LOOP)
1476  {
1477      s32 t_wTmpKp;
1478      s32 t_wTmpKi;
1479      pObj->m_pMotorSpeed->mSpeedPI.KP = pDrvCfgPara->mS_FBSpdLoop.PASRKp; //速度环 Kp P_ASr_KP_M0
1480      pObj->m_pMotorSpeed->mSpeedPI.KI = pDrvCfgPara->mS_FBSpdLoop.PASRKi; //速度环 Ki P_ASr_KI_M0
1481      pObj->m_pMotorSpeed->mSpeedPI.wIntegral = 0;
1482      pObj->m_pMotorSpeed->mSpeedPI.wUpperLimitOutput = App2CoreCurTrans(pAppToCore, pDrvCfgPara->mS_FBSpdLoop.m_wIqMax); //速度环输出最大
1483      pObj->m_pMotorSpeed->mSpeedPI.wLowerLimitOutput = App2CoreCurTrans(pAppToCore, pDrvCfgPara->mS_FBSpdLoop.m_wIqMin); //速度环输出最小
1484      pObj->m_pMotorSpeed->mSpeedPI.wInError = 0;
1485
1486      #if 0
1487      t_nTmp = pDrvCfgPara->mS_FBSlvcCfg0.m_wSpeedFilTime + 25;
1488      //自动调节速度环PI功能
1489      SpeedPIAutoTuning(pObj->m_nMACTrIProcID,
1490                      5000,
1491                      t_nTmp,
1492                      &t_wTmpKp,
1493                      &t_wTmpKi);
1494
1495      pObj->m_pMotorSpeed->mSpeedPI.KP = sat(t_wTmpKp, 0, 32767); // PI参数限幅
1496      pObj->m_pMotorSpeed->mSpeedPI.KI = sat(t_wTmpKi, 0, 32767);
1497      #endif
1498

```

图 3.35 速度环 PI 参数自动计算使能

## 4. LKscope 简介

lkscope 调试的变量说明如下，图中的可查看的数值变量已经在工程中配置完成，用户可以根据自己的需要进行，每个数值串口与右下角底栏的相对应，对于不需要获取的变量，用户可以取消选中，以提高数据的获取速率。变量基本上都指向 gS\_PhyObjInstanceM 这个结构体，用户可自行搜索。

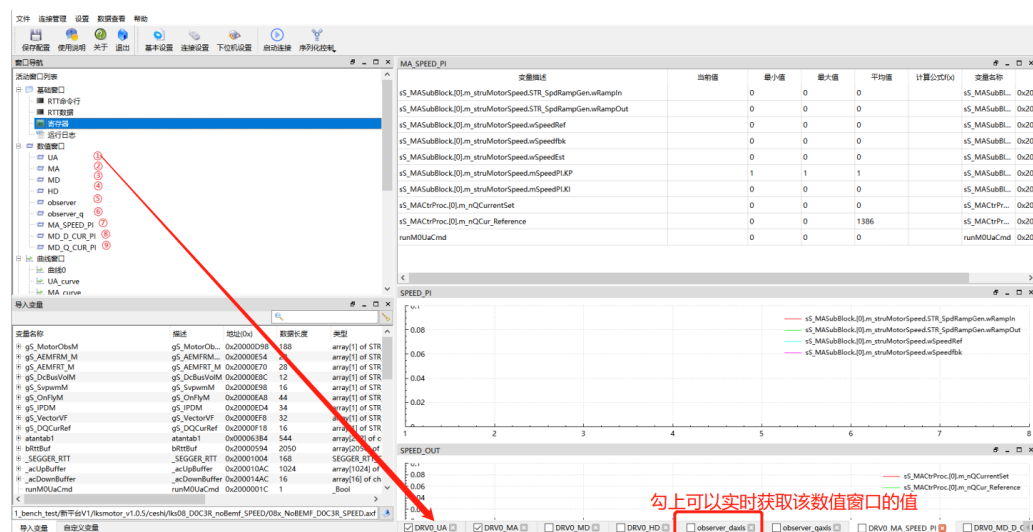


图 4.1 调试界面

根据测试例程文件中的 lkscope 配置工程，将分别对①-⑨中几个关键的调试变量进行简单的介绍，没有介绍的变量，用户可以根据自己的芯片系列以及具体需要，在调试中进行熟悉，具体也可以向凌鸥官方咨询。下面是对软件中相关数值窗口中的一些常用变量的介绍，文档是对一些程序中的基本变量进行介绍，具体实际的 lkscope 中配置的变量以实际 lkscope 中为主，如果有相关不熟悉的变量，可以与凌鸥官方进行沟通。

### 4.1.1 电机启停、转速调节

①: UA 窗口;UserApp 层，用户可以更改电机运行命令，以及相关目标电流，速度值，功率值等，例如用户可以实时更改运行命令变量和目标速度在值进行电机启停和转速的调节

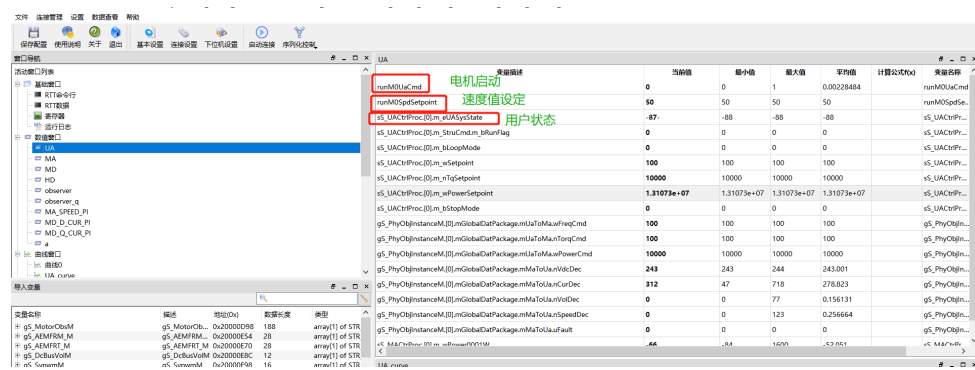


图 4.2 控制运行标志

UA 标签页的相关变量含义

UA	
变量名称	变量描述
runM0UaCmd	启停命令
runM0SpdSetpoint	速度设定值
sS_UACtrlProc.[0].m_eUASysState	UA层运行状态机
sS_UACtrlProc.[0].m_StruCnd.m_bRunFlag	用户控制命令字
sS_UACtrlProc.[0].m_bLoopMode	当前环路
sS_UACtrlProc.[0].m_wSetpoint	速度设置点 (0.01hz)
sS_UACtrlProc.[0].m_nTqSetpoint	力矩设置点 0.001A
sS_UACtrlProc.[0].m_wPowerSetpoint	功率设置点 0.01w
sS_UACtrlProc.[0].m_bStopMode	停止模式
gS_PhyObjInstanceM.[0].mGlobalDatPackage.mUaToMa.wFreqCmd	电频率设定值
gS_PhyObjInstanceM.[0].mGlobalDatPackage.mUaToMa.nTorqCmd	力矩值设定值
gS_PhyObjInstanceM.[0].mGlobalDatPackage.mUaToMa.wPowerC...	功率值设定值
gS_PhyObjInstanceM.[0].mGlobalDatPackage.mMaToUa.nVdcDec	母线电压值(0.1)
gS_PhyObjInstanceM.[0].mGlobalDatPackage.mMaToUa.nCurDec	电流峰值
gS_PhyObjInstanceM.[0].mGlobalDatPackage.mMaToUa.nVolDec	电压值 $\sqrt{ud^2 + uq^2}$
gS_PhyObjInstanceM.[0].mGlobalDatPackage.mMaToUa.nSpeedDec	当前电频率
gS_PhyObjInstanceM.[0].mGlobalDatPackage.mMaToUa.uFault	当前运行故障
sS_MACtrlProc.[0].m_wPower0001W	当前功率值

图 4.3 UA 层变量

#### 4.1.2 保护标志，如过流、过压等保护

②：MA 窗口;MotorDrive 层：查看电机运行过程中的故障异常种类，都会实时更新到该故障标志当中，2 个应用层的错误保护标志是一致的，UA 层是 MA 错误标志的总集。

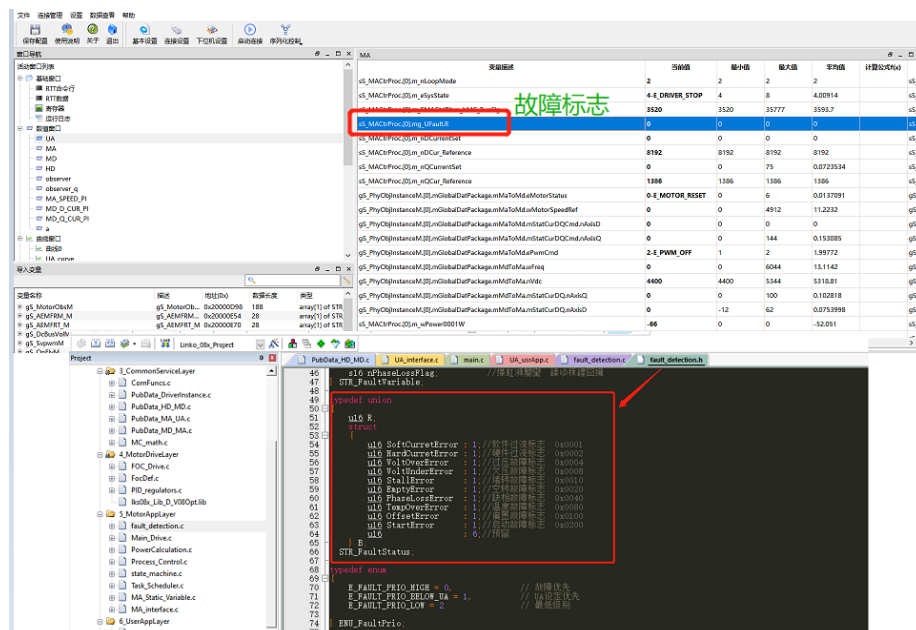


图 4.4 MA 层错误标志

## MA 层相关变量含义

s_MACtrProc[0].m_nLoopMode	当前运行环路
s_MACtrProc[0].m_eSysState	电机当前状态
s_MACtrProc[0].m_SMACtrBit.m_bMC_RunFlg	电机运行控制字域
s_MACtrProc[0].mg_UFault.R	电机当前故障状态
s_MACtrProc[0].m_nDCurrentSet	Id设定值
s_MACtrProc[0].m_nDCur_Reference	Id参考值
s_MACtrProc[0].m_nQCurrentSet	Iq设定值
s_MACtrProc[0].m_nQCur_Reference	Iq参考值
gS_PhyObjInstanceM[0].mGlobalDatPackage.mMaToMd.eMotorStatus	电机当前状态 (Focdrive.c)
gS_PhyObjInstanceM[0].mGlobalDatPackage.mMaToMd.wMotorSpeedRef	速度参考值
gS_PhyObjInstanceM[0].mGlobalDatPackage.mMaToMd.mStatCurDQCmd.n...	Q轴给定值
gS_PhyObjInstanceM[0].mGlobalDatPackage.mMaToMd.mStatCurDQCmd.n...	D轴给定值
gS_PhyObjInstanceM[0].mGlobalDatPackage.mMaToMd.ePwmCmd	PWM使能命令
gS_PhyObjInstanceM[0].mGlobalDatPackage.mMdToMa.wFreq	电频率值
gS_PhyObjInstanceM[0].mGlobalDatPackage.mMdToMa.nVdc	母线电压采集值
gS_PhyObjInstanceM[0].mGlobalDatPackage.mMdToMa.mStatCurDQ.nAxisQ	Q轴反馈值
gS_PhyObjInstanceM[0].mGlobalDatPackage.mMdToMa.mStatCurDQ.nAxisD	D轴反馈值
s_MACtrProc[0].m_wPower0001W	功率值
s_MASubBlock[0].m_struMotorSpeed.wSpeedRef	速度参考值
s_MASubBlock[0].m_struMotorSpeed.wSpeedSet	速度设定值
s_MASubBlock[0].m_struMotorSpeed.STR_SpdRampGen.wRampln	速度斜坡输入

图 4.5 MA 层参数

### 4.1.3 系统状态机查看

状态机有 2 个，分别为①：UA 窗口：用户应用层(UA)-低频任务 1ms 周期，一般只需关注这个即可；②：MA 窗口：电机应用层(MA)-高频任务 MCPWM 周期，想深入了解控制算法的可以查看这个层，不建议修改里面的代码；2 个状态机的状态是保持一致的







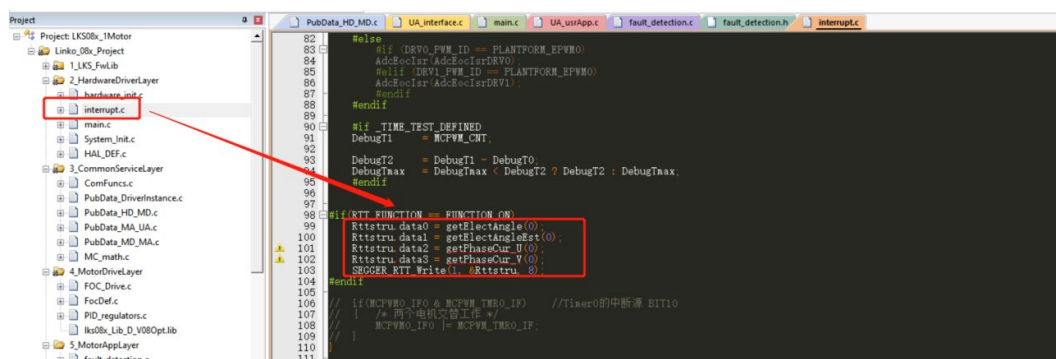
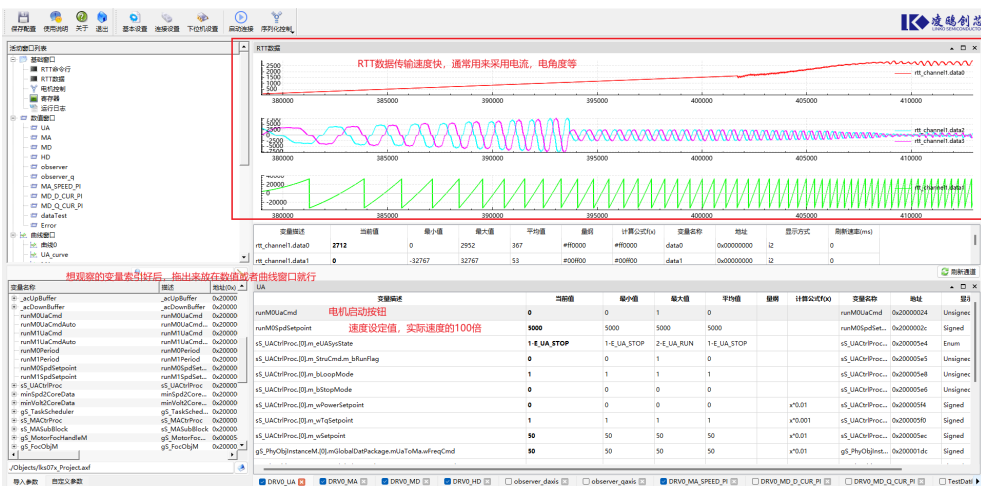


图 4.12 配置 RTT 传输数据



图 4.13 RTT 数据显示

### LKSscope 使用如图





## 5. 故障保护与恢复

见算法新平台使用手册 V1.2.3.PDF

## 6. 参考资料

### 数据手册：

LKSMC03x\_DS\_v2.51

LKSMC03x\_6N\_v2.72

LKSMC03x\_3P3N\_v2.51

LKSMC07x\_DS\_v1.11

LKSMC07x\_6N\_v1.19

LKSMC07x\_3P3N\_v1.09

### 用户手册：

LKSMC03x\_UM\_v1.55

LKSMC07x\_UM\_v1.19