

文档编号: AN2058

上海东软载波微电子有限公司

用户手册

ES_MC_SDK

修订历史

版本	修订日期	修改概要
V1.0	2024-06-04	初版

地址：中国上海市古美路 1515 号凤凰园 12 号楼 3 层

E-mail: support@essemi.com

电话: +86-21-60910333

传真: +86-21-60914991

网址: <http://www.essemi.com/>

版权所有©

上海东软载波微电子有限公司

本资料内容为上海东软载波微电子有限公司在现有数据资料基础上慎重且力求准确无误编制而成，本资料中所记载的实例以正确的使用方法和标准操作为前提，使用方在应用该等实例时请充分考虑外部诸条件，上海东软载波微电子有限公司不承担或确认该等实例在使用方的适用性、适当性或完整性，上海东软载波微电子有限公司亦不对使用方因使用本资料所有内容而可能或已经带来的风险或后果承担任何法律责任。基于使本资料的内容更加完善等原因，上海东软载波微电子有限公司保留未经预告的修改权。使用方如需获得最新的产品信息，请随时用上述联系方式与上海东软载波微电子有限公司联系。

目录

内容目录

第 1 章	电机控制平台概述	5
1.1	控制框图.....	5
1.2	层次结构.....	5
1.3	时序逻辑.....	6
1.4	数据处理.....	6
第 2 章	ES_MC_SDK 软件平台	8
2.1	信号输出.....	8
2.1.1	函数列表.....	8
2.1.2	参数说明.....	8
2.1.3	信号输出.....	8
2.2	信号输入.....	9
2.2.1	函数列表.....	9
2.2.2	参数说明.....	9
2.2.3	信号输入.....	10
2.3	FOC 数学运算.....	10
2.3.1	函数列表.....	10
2.3.2	参数说明.....	11
2.3.3	FOC 数学运算.....	11
2.4	PID 控制器.....	11
2.4.1	函数列表.....	11
2.4.2	参数说明.....	12
2.4.3	PID 控制器.....	12
2.5	位置观测器.....	13
2.5.1	函数列表.....	13
2.5.2	参数说明.....	13
2.5.3	位置观测器.....	13
2.6	空间矢量脉宽调制.....	14
2.6.1	函数列表.....	14
2.6.2	参数说明.....	14
2.6.3	空间矢量脉宽调制.....	15
2.7	速度位置反馈.....	15
2.7.1	函数列表.....	15
2.7.2	参数说明.....	15
2.8	电流处理.....	16
2.8.1	函数列表.....	16
2.8.2	参数说明.....	16
2.8.3	电流处理.....	17
2.9	电压处理.....	17
2.9.1	函数列表.....	17
2.9.2	参数说明.....	17
2.9.3	电压处理.....	17

2.10	温度处理.....	18
2.10.1	函数列表.....	18
2.10.2	参数说明.....	18
2.10.3	温度处理.....	18
2.11	功率管理.....	18
2.11.1	函数列表.....	18
2.11.2	参数说明.....	18
2.12	错误管理.....	19
第3章	ES_MC_SDK 软件调试.....	20
3.1	配置电机参数.....	20
3.2	配置 PI 参数.....	21
3.3	电流闭环调试.....	21
3.4	位置闭环调试.....	23
3.5	速度闭环调试.....	24
第4章	ES_MC_SDK 硬件平台.....	25
4.1	低压硬件平台.....	25
4.2	高压硬件平台.....	26

1.3 时序逻辑

ES_MC_SDK 软件控制时序分为以下两部分：

1. ADC 中断（最高级）：

高级定时器 AD16C6T1 的 CH4 通道计数触发 ADC 中断，在 ADC 中断中用于处理程序响应最高级的功能模块，如电流采样、FOC 数学运算、过流保护等。

2. System Ticker 中断（次高级）：

System Ticker 中断用于处理有固定运行时序要求且没有特殊运行频率要求的功能模块，如电机状态机、堵转保护、缺相保护、温度保护、过欠压保护等。

1.4 数据处理

本系统控制流程中的计算全部使用定点数进行计算。定点数和实际值转换关系为：

$$X_{fix} = (X * Q15) / X_{max}$$

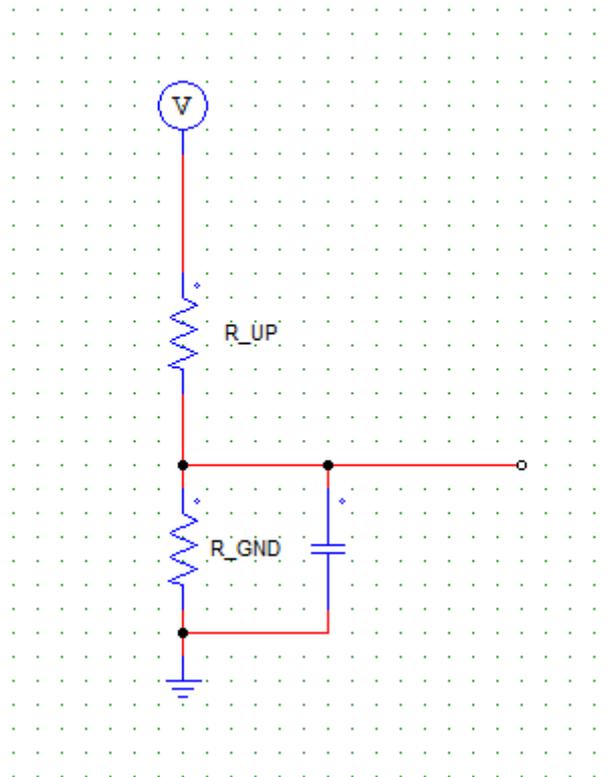
其中 X_{fix} 表示系统中的定点数数值， X 表示实际值， $Q15$ 表示 215， X_{max} 表示该物理量相对应的硬件可支持最大值。

以电流举例，假设实际电流为 3A，硬件可支持最大电流为 30A，则系统中电流定点数数值为：

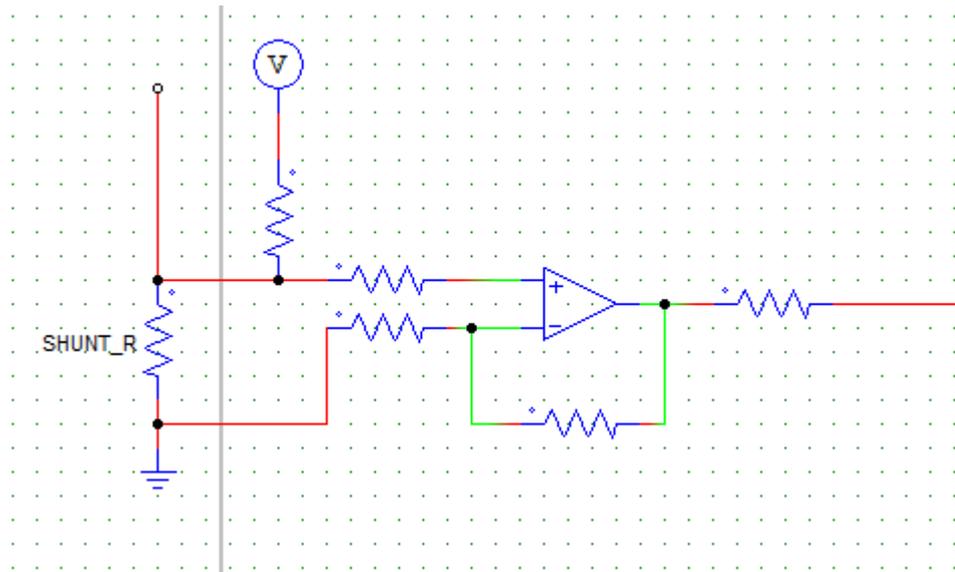
$$I_{fix} = (3A * Q15) / 30A = 3277$$

以电压举例，假设系统中电压定点数数值为 900，硬件可支持最大电压为 900V，则实际电压为：

$$U = (900 * 900V) / Q15 = 24.7V$$



硬件可支持最大电压： $V_{max} = (MCU_VCC * (R_UP + R_GND)) / (R_GND)$



硬件可支持最大电流： $A_{max}=(MCU_VCC-AMP_OFFSET)/(GAIN*SHUNT_R)$

其中，MCU_VCC 表示 MCU 供电电压，R_UP 表示母线电压采样端口与母线电压之间的电阻阻值，R_GND 表示母线电压采样端口与地之间的电阻阻值，AMP_OFFSET 表示运算放大器偏置值，GAIN 表示运算放大器放大倍数，SHUNT_R 表示采样电阻阻值。

第2章 ES_MC_SDK软件平台

2.1 信号输出

2.1.1 函数列表

工程源文件 mid_sys_output_mngr.c 用于管理系统输入/输出功能，包括 GPIO、DAC 和通信等。

软件库向应用层提供常用的硬件信号输出操作接口函数，如下表所示：

函数名	形式参数	功能简述
SysOutput_Component_Init	SysOutput_Struct_t, HWOption_Struct_t	输出组件初始化
SysOutput_SetOutputState	SysOutput_Struct_t, state	输出端口状态设置
SysOutput_GetOutputState	SysOutput_Struct_t	读输出端口状态
SysOutput_SetDAC	SysOutput_Struct_t, DataOut, ChNum	设置 DAC 通道
ReadDACData	SysOutput_Struct_t, ChNum	读 DAC 输出值
SysOutput_ReadDACChNum	SysOutput_Struct_t	取当前 DAC 通道
SysOutput_RegisterUpdate	SysOutput_Struct_t	更新 PWM 占空比
SysOutput_SwitchOnPwm	SysOutput_Struct_t	打开 PWM
SysOutput_SwitchOffPwm	SysOutput_Struct_t	关闭 PWM
SysOutput_OpenLowSideMos	SysOutput_Struct_t	打开下桥 MOS

2.1.2 参数说明

类别	参数	类型	说明
SysOutput_Struct_t	IO_State	SysOutputState_t	端口使能
	OutputPin	uint16_t	端口值
	OutputPolar	uint8_t	端口极性
	u16hTimePhA	uint16_t*	通道比较寄存器
	u16hTimePhB	uint16_t*	通道比较寄存器
	u16hTimePhC	uint16_t*	通道比较寄存器
	u16hTimePhD	uint16_t*	通道比较寄存器
	u16hTimePhE	uint16_t*	通道比较寄存器
	u16hTimePhF	uint16_t*	通道比较寄存器

2.1.3 信号输出

信号输出操作中，用户较常使用的信号输出接口主要有端口输出状态设置，PWM 控制和 DAC 控制。

端口输出状态设置函数 SysOutput_SetOutputState()，输入参数是 SysOutput_Struct_t 和端口输出状态值 0 或 1。

PWM 控制主要有两个：

(1) SysOutput_OpenLowSideMos ()，输入参数是 SysOutput_Struct_t 结构体指针变量，无返回值。给应用层提供一种硬件操作的抽象接口，功能是打开下桥 MOS 管，一般用于系统上电时自举充电阶段；

(2) SysOutput_SwitchOnPwm()，输入参数是 SysOutput_Struct_t 结构体指针变量，无返回

值，同样给应用层提供一种硬件操作的抽象接口，其功能是打开 MCU 管脚的 PWM 脉冲输出。
具体操作由硬件实例化源文件的特定函数实现，示例如下：

```
case SYS_Stage2:
    if (gVarTsk.tEna_flag.bits.POS_EN)
    {

        gVarEsmc.pVarCurrSmp->s16I_Cu_d->Target += (gVarEsmc.pVarCurrSmp->s16IqRef_Add<<2);
        gVarEsmc.pVarCurrSmp->s16I_Cu_q->Target = 0;
        gVarEsmc.pVarTorSpd->pSpdFdbk->s16ElecAngle = 0;

        SysOutput_SwitchOnPwm(gVarEsmc.pVarSysOut);
        if(gVarEsmc.pVarCurrSmp->s16I_Cu_d->Target >= gVarEsmc.pVarCurrSmp->s16IqRef_Start)
        {
            gVarEsmc.pVarCurrSmp->s16I_Cu_d->Target = gVarEsmc.pVarCurrSmp->s16IqRef_Start;
            /*使用PWM触发ADC*/
            SysOutput_ADC_J_ExtStar(gVarEsmc.pVarSysOut);
            gVarTsk.Sys_Stage = SYS_Stage3;
        }
    }
    else
    gVarTsk.Sys_Stage = SYS_StageEnd;
    break;
```

2.2 信号输入

工程源文件 mid_sys_input_mgr.c 用于管理系统输入/输出功能，包括 GPIO、ADC 等。

2.2.1 函数列表

软件库向应用层提供信号输入接口函数以读端口输入状态及相电流、母线电流、温度、电压等模拟量采集，信号输入函数接口，如下表所述：

函数名	形式参数	功能简述
SysInput_GetIOState	InputSignal_Struct_t, IOPin	读端口输入状态
SysInput_GetADCCConv	InputSignal_Struct_t, ChNum	取标准通道 ADC1 当前通道值
SysInput_GetICH1Conv	InputSignal_Struct_t	取 ADC1 插入组第一次 ADC 转换值
SysInput_GetICH2Conv	InputSignal_Struct_t	取 ADC1 插入组第二次 ADC 转换值
SysInput_GetICH3Conv	InputSignal_Struct_t	取 ADC1 插入组第三次 ADC 转换值
SysInput_GetICH4Conv	InputSignal_Struct_t	取 ADC1 插入组第四次 ADC 转换值
SysInput_GetADC2ICH1Conv	InputSignal_Struct_t	取 ADC2 插入组第一次 ADC 转换值
SysInput_GetNCH1Conv	InputSignal_Struct_t	取 ADC1 标准组第一次 ADC 转换值
SysInput_GetNCH2Conv	InputSignal_Struct_t	取 ADC1 标准组第二次 ADC 转换值
SysInput_GetNCH3Conv	InputSignal_Struct_t	取 ADC1 标准组第三次 ADC 转换值
SysInput_GetNCH4Conv	InputSignal_Struct_t	取 ADC1 标准组第四次 ADC 转换值
SysInput_GetOffsetConv	InputSignal_Struct_t	读取相电流和母线电流零点偏置值

2.2.2 参数说明

类别	参数	类型	说明
InputSignal_Struct_t	SW1	uint8_t	电机启动标志（硬件电位器）
	SW2	uint8_t	电机启动标志（软件）
	AD_SPEED	AD_PARM	速度采样值
	AD_VBUS	AD_PARM	母线电压采样值

	AD_Temp_MOS	AD_PARM	MOS 温度采样值
	AD_Temp_BAT	AD_PARM	电池温度采样值
	AD_U_Curr	AD_PARM	U 相电流采样值
	AD_V_Curr	AD_PARM	V 相电流采样值
	AD_W_Curr	AD_PARM	W 相电流采样值
	AD_S_Curr	AD_PARM	总电流采样值
	AD_I_PP	AD_PARM	电流峰值
	AD_I_AVG	AD_PARM	电流平均值
	ADC_TEMP[4]	uint16_t	标准通道采样值

2.2.3 信号输入

函数 SysInput_GetIOState() 读端口输入状态，输入参数是 InputSignal_Struct_t 结构体指针变量和 Pin 号，返回值为端口状态。

相电流和母线电压的采样由分别由插入组 ADC1 和 ADC2 完成，其余模拟量采集由标准组 ADC1 或 ADC2 完成。

由于相电流是交流信号，应用电路会将交流信号增加一个直流偏置供 ADC 采。在电机运行前必须获取相电流零点值才能进行 FOC 运算。相电流零点值由函数 SysInput_GetOffsetConv() 完成，SysInput_GetOffsetConv() 的输入参数是 CurrSmp_Struct_t 结构体指针变量，无返回值，在电机启动前调用该函数以获取相电流的 ADC 通道零点偏置值，该函数内部采样 8 次并进行了均值计算。示例如下：

```

case SYS_Stage2:
gVarEsmc.pVarSvpwm->hTimePhA = PWM_CCR;
gVarEsmc.pVarSvpwm->hTimePhB = PWM_CCR;
gVarEsmc.pVarSvpwm->hTimePhC = PWM_CCR;
SysOutput_RegisterUpdate(gVarEsmc.pVarSysOut);
SysOutput_ADC_J_SoftStar(gVarEsmc.pVarSysOut);
SysInput_ADC_check(gVarEsmc.pVarSysInput);
SysInput_GetOffsetConv(gVarEsmc.pVarSysInput);
++gVarTsk.p_sysTimer.wTicks;

if (++gVarTsk.p_sysTimer.wTicks >100)
{
gVarTsk.p_sysTimer.wTicks = 0;
gVarTsk.flag_sysTimer = 0;
gVarTsk.Sys_Stage = SYS_Stage3;
}
break;

```

2.3 FOC 数学运算

2.3.1 函数列表

软件库向应用层提供转子位置和速度等相关的接口函数，如下表所述：

函数名	形式参数	功能简述
Clarke	CurrSmp_Struct_t	CLARKE 坐标计算
Park	CurrSmp_Struct_t, PosFdbk_Struct_t	PARK 坐标计算
VoltQDLimitation	Svpwm_Struct_t	电压限幅处理

Rev_Park	Svpwm_Struct_t, PosFdbk_Struct_t	反 PARK 坐标计算
----------	----------------------------------	-------------

2.3.2 参数说明

类别	参数	类型	说明
CurrSmp_Struct_t	s16I_U_Curr	int16_t *	U 相电流
	s16I_V_Curr	int16_t *	V 相电流
	s16I_W_Curr	int16_t *	W 相电流
	s16I_S_Curr	int16_t *	总电流
	s16IqRef_Add	int16_t	电流增加步长
	s16IqRef_Start	int16_t	启动电流
	s16IqRef_Dec	int16_t	电流减小步长
	s16I_Alpha	int16_t	Ialpha
	s16I_Beta	int16_t	Ibeta
	s16I_OV	int16_t	相电流过流阈值
	s16I_Cu_d	Pid_Struct_t *	d 轴 PID
	s16I_Cu_q	Pid_Struct_t *	q 轴 PID
	sampCur1	int16_t *	单电阻采样点
sampCur2	int16_t *	单电阻采样点	
Svpwm_Struct_t	swVoltDQ16	int16_t	Vd
	swVoltQQ16	int16_t	Vq
	swVoltBetaQ16	int16_t	Vbeta
	swVoltAlphaQ16	int16_t	Valpha
PosFdbk_Struct_t	s32Rotor_Angle_Sin	int16_t	角度正弦值
	s32Rotor_Angle_Cos	int16_t	角度余弦值

2.3.3 FOC数学运算

FOC 数学运算接口函数作为 FOC 控制的核心部件, 在 TSK_FOCMotorControl()任务中进行调用, 示例如下:

```

if (gVarTsk.tEna_flag.bits.PWM_OUT_EN)
{
    /* 采集转子电角度位置 */
    GetRotorElecAngle (gVarEsmc.pVarTorSpd->pSpdFdbk);
    /* CLARKE */
    Clarke (gVarEsmc.pVarCurrSmp);
    /* PARK */
    Park (gVarEsmc.pVarCurrSmp, gVarEsmc.pVarTorSpd->pSpdFdbk);
}

```

2.4 PID控制器

2.4.1 函数列表

软件库向应用层提供 PID 控制器接口。PID 调节计算在 FOC 矢量控制中是必不可少的环节。用户可调用该文件的相关接口函数完成对速度 PID、力矩电流 PID 和磁链电流 PID 的计算。相关的接口函数, 如下表所述:

函数名	形式参数	功能简述
PID_SetKP	Pid_Struct_t, int16_t	设置比例项系数
PID_SetKI	Pid_Struct_t, int16_t	设置积分项系数
PID_GetKP	Pid_Struct_t	获取比例项系数
PID_GetKI	Pid_Struct_t	获取积分项系数
PID_SetIntegralTerm	Pid_Struct_t, int16_t	设置积分值
PID_GetKPDivisor	Pid_Struct_t	获取比例项缩放因子
PID_SetKPDivisorBS	Pid_Struct_t, int16_t	设置比例项缩放因子移位数
PID_GetKIDivisor	Pid_Struct_t	获取积分项缩放因子
PID_SetKIDivisorBS	Pid_Struct_t, int16_t	设置积分项缩放因子移位数
PID_SetLowerIntegralTermLimit	Pid_Struct_t, int16_t	设置积分项累加下限边界值
PID_SetUpperIntegralTermLimit	Pid_Struct_t, int16_t	设置积分项累加上限边界值
PID_SetLowerOutputLimit	Pid_Struct_t, int16_t	设置控制器输出下限边界值
PID_SetUpperOutputLimit	Pid_Struct_t, int16_t	设置控制器输出上限边界值
PID_Regulators	int16_t, int16_t, Pid_Struct_t	PID 运算

2.4.2 参数说明

类别	参数	类型	说明
Pid_Struct_t	Target	int16_t	目标值
	Actual	int16_t	实际值
	s32Kp	int32_t	Kp
	u16KpDiv	uint16_t	Kp*Err Div 值
	u8KpDivBS	uint8_t	Kp*Err Div
	s32Ki	int32_t	Ki
	u16KiDiv	uint16_t	Ki*Err Div 值
	u8KiDivBS	uint8_t	Ki*Err Div
	s32Kd	int32_t	Kd
	u16KdDiv	uint16_t	Kd*Err Div 值
	u8KdDivBS	uint8_t	Kd*Err Div
	s16MinLimitOutput	int16_t	输出限值最小值
	s16MaxLimitOutput	int16_t	输出限值最大值
	s32MinLimitIntegral	int32_t	积分限制最小值
	s32MaxLimitIntegral	int32_t	积分限制最大值
	s32Integral	int32_t	积分值
	s32PreviousError	int32_t	前次差值
s16CurrentError	int16_t	当前差值	
s32PidOutput	int32_t	PID 输出值	

2.4.3 PID控制器

函数 PID_Regulators()作为 FOC 控制的核心部件之一，在 TSK_FOCMotorControl()任务中进行调用，示例如下：

```

/* 电流环 */
if (gVarTsk.tEna_flag.bits.Iq_LOOP_EN)
{
    gVarEsmc.pVarSvpwm->swVoltQ16 = PID_Regulators(gVarEsmc.pVarCurrSmp->sl6I_Cu_q->Target, gVarEsmc.pVarCurrSmp->sl6I_Cu_q->Actual, gVarEsmc.pVarIqCurrentPid);
}
if (gVarTsk.tEna_flag.bits.Id_LOOP_EN)
{
    gVarEsmc.pVarSvpwm->swVoltDQ16 = PID_Regulators(gVarEsmc.pVarCurrSmp->sl6I_Cu_d->Target, gVarEsmc.pVarCurrSmp->sl6I_Cu_d->Actual, gVarEsmc.pVarIdCurrentPid);
}
/* 逆PARK4SVFWM */
Rev_ParkANDSVFWM(gVarEsmc.pVarSvpwm, gVarEsmc.pVarTorSpd->pSpdFdbk);
SysOutput_RegisterUpdate(gVarEsmc.pVarSysOut);
    
```

2.5 位置观测器

2.5.1 函数列表

FOC 软件库位置观测器使用滑模控制观测器的接口函数，如下表所述：

函数名	形式参数	功能简述
MotorDrv_Smclnit	Smc_Struct_t	滑模参数初始化
MotorDrv_Smc	Smc_Struct_t	滑模算法
MotorDrv_Atan2	int16_t	反正切
MotorDrv_LPF	LPF_struct, int16_t	低通滤波

2.5.2 参数说明

类别	参数	类型	说明
Smc_Struct_t	pEdPID	Pid_Struct_t *	PLL PID
	Ealpha_LPF	LPF_struct	Ealpha 滤波
	Ebeta_LPF	LPF_struct	Ebeta 滤波
	Spd_LPF	LPF_struct	速度滤波
	LPF_Kvalue	int16_t	Ealpha/Ebeta 滤波截止频率
	LPF_SpdKvalue	int16_t	速度截止频率
	s16Zalpha	int16_t	Zalpha
	s16Ualpha	int16_t*	Ualpha
	s16Ealpha	int16_t	Ealpha
	s16lalpha	int16_t*	lalpha
	s32lalpha_Est	int32_t	lalpha 估算值
	s32lalpha_Err	int32_t	lalpha 差值
	s16Zbeta	int16_t	Zbeta
	s16Ubeta	int16_t*	Ubeta
	s16Ebeta	int16_t	Ebeta
	s16lbeta	int16_t*	lbeta
	s32lbeta_Est	int32_t	lbeta 估算值
	s32lbeta_Err	int32_t	lbeta 差值
s16Kslide	int16_t	滑模增益	

2.5.3 位置观测器

函数 MotorDrv_Smc()是滑模观测器控制函数，在 TSK_FOCMotorControl()任务中进行调用，示例如下：

```

/* 位置估算 */
if(gVarTsk.tEna_flag.bits.Omega_EN ) /* 启动位置估算 */
{
    gVarEsmc.pVarTorSpd->pSpdFdbk->s16CurSpeedHz = MotorDrv_Smc(gVarEsmc.pVarSmc);
}

if(gVarTsk.tEna_flag.bits.ElecAngle_updata_EN) /* 位置启用*/
{
    gVarEsmc.pVarTorSpd->pSpdFdbk->s16OmegaFix = gVarEsmc.pVarTorSpd->pSpdFdbk->s16CurSpeedHz;
}
else
{
    gVarEsmc.pVarTorSpd->pSpdFdbk->s16OmegaFix = gVarEsmc.pVarTorSpd->pSpdFdbk->s32OmegaComFix.Word.HWord;
}
    
```

2.6 空间矢量脉宽调制

2.6.1 函数列表

SVPWM 作为 FOC 控制关键环节之一，由三相功率逆变器的 6 个功率管组成的特定开关组合所产生的脉宽调制波，能够使电流波形尽可能接近于理想的正弦波。经过 SVPWM 算法处理，目标电压矢量可转换成用户可控的占空比调节。

软件库向应用层提供空间矢量脉宽调制 SVPWM 的接口函数，如下表所述：

函数名	形式参数	功能简述
Svpwm_Component_Init	Svpwm_Struct_t	SVPWM 组件初始化
Svpwm_Generator	Svpwm_Struct_t	三电阻七段式 SVPWM 生成
ThreeShuntFiveSec_SvpwmGenerator	Svpwm_Struct_t	三电阻五段式 SVPWM 生成
Svpwm_GetCurrentSectorID	Svpwm_Struct_t	获取当前电压矢量所在的扇区区号

2.6.2 参数说明

类别	参数	类型	说明
Svpwm_Struct_t	tMode	PWM_MODE_t	SVPWM 运行模式
	swX	int32_t	SVPWM 反 CLARKE 中间变量
	swY	int32_t	SVPWM 反 CLARKE 中间变量
	swZ	int32_t	SVPWM 反 CLARKE 中间变量
	swVoltDQ16	int16_t	Vd
	swVoltQQ16	int16_t	Vq
	swVoltBetaQ16	int16_t	Vbeta
	swVoltAlphaQ16	int16_t	Valpha
	DMA_Pwm	Pwm_Cmpval_t	SVPWM DMA 传输值
	hTimePhA	uint16_t	U 相占空比
	hTimePhB	uint16_t	V 相占空比
	hTimePhC	uint16_t	W 相占空比
	hTimePhD	uint16_t	匹配占空比
	tDTCompEN	uint8_t	死区补偿使能
	tDTComp	uint16_t	死区补偿值
	tDTIDir	SEW_FLAG_DTI	死区补偿方向
tPWMperiod	uint16_t	PWM 周期	
tPWMComp	uint16_t	移相时间	

	tBus_Volt	uint16_t*	母线电压
	tSector	uint8_t	扇区

2.6.3 空间矢量脉宽调制

使用软件 SVPWM 时，在 TSK_FOCMotorControl()任务中，通过 Rev_ParkANDSVPWM()进行调用，示例如下：

```
/* 逆PARK&SVPWM */
Rev_ParkANDSVPWM(gVarEsmc.pVarSvpwm, gVarEsmc.pVarTorSpd->pSpdFdbk);
SysOutput_RegisterUpdate(gVarEsmc.pVarSysOut);
```

2.7 速度位置反馈

2.7.1 函数列表

电机控制库固件工程中源文件 mid_rotor_position_fdbk.c 负责向应用层提供转子位置和速度信息反馈相关的接口，支持 HALL 传感器，编码器以及无传感器位置反馈方式。

相关的接口函数，如下表所述：

函数名	形式参数	功能简述
Posfd_GetRotorElecAngle	PosFdbk_Struct_t	获取转子电角度
Posfd_GetRotorMecAngle	PosFdbk_Struct_t	获取转子机械角度
Posfd_GetRotorMechSpeedRPM	PosFdbk_Struct_t	获取机械转速 rpm
Posfd_GetRotorElecSpeedRPM	PosFdbk_Struct_t	获取电转速 rpm
Posfd_GetRotorMechSpeed0_1RPS	PosFdbk_Struct_t	获取机械转速 0.1rpm
Posfd_GetRotorElecSpeed0_1RPS	PosFdbk_Struct_t	获取电转速 0.1rpm
Posfd_GetRotorMechSpeedS16	PosFdbk_Struct_t	获取机械转速 s16 格式
Posfd_GetRotorElecSpeedS16	PosFdbk_Struct_t	获取电转速 s16 格式
Posfd_GetElecAngleIncPerPwm	PosFdbk_Struct_t	获取角速度（每 PWM 周期角度增量）
Posfd_SpeedFaultCheck	PosFdbk_Struct_t	速度反馈错误检查
Posfd_IsMechSpeedReliable	PosFdbk_Struct_t	机械速度可靠性检查
Posfd_GetPolePairsNum	PosFdbk_Struct_t	获取电机极对数
SetPolePairsNum	PosFdbk_Struct_t	设置电机极对数

2.7.2 参数说明

类别	参数	类型	说明
PosFdbk_Struct_t	s16Omega_DPT	int16_t	角速度
	s16ElecAngle	int16_t	电角度
	s16MecAngle	int16_t	机械角度
	s16AverOmega_DPT	int16_t	角速度平均值
	u8RatioMeToEl	uint8_t	电子角度和机械角度比值
	s16EleSpeed_Rpm	int16_t	电转速
	s16MecSpeed_Rpm	int16_t	机械转速
	s16EleSpeed_0_1Rps	int16_t	机械转速 0.1rpm
	s16MecSpeed_0_1Rps	int16_t	电转速 0.1rpm

	u8SpeedErrorNbr	uint8_t	速度错误次数
	u16MaxMecSpeed_Rpm	uint16_t	机械最大转速
	u16MinMecSpeed_Rpm	uint16_t	机械最小转速
	u8MaxSpeedErrors	uint8_t	允许最多错误次数
	u16MeasFreqHz	uint16_t	频率测量值
	u16SpeedFault	uint16_t	转速误差
	s32ElecAngleTemp	int32_t	电角度临时值
	s16OmegaFix	int16_t	16 位角速度
	s32OmegaComFix	LWB_32_16_8	32 位角速度预估值
	s32OmegaUserFix	LWB_32_16_8	32 位角速度给定值
	s32OmegaMinFix	LWB_32_16_8	33 位角速度最小值
	s16ElecAngleConst	int16_t	角度计算常数
	s16CurSpeedHz	int16_t	当前电角速(Hz)
	s32AccFix	int32_t	递增步长
	s32DecFix	int32_t	递减步长
	s16Rotor_Angle_Sin	int16_t	转子角度正弦值
	s16Rotor_Angle_Cos	int16_t	转子角度余弦值

2.8 电流处理

2.8.1 函数列表

工程中源文件 mid_phase_current_smp.c 用于完成相电流采集功能，相电流反馈作为 FOC 控制中最为关键的环节，采集质量直接决定系统最终性能。支持两种典型的电流采集方式：单电阻采样和双电阻采样。此外还提供获取组件内部状态信息的接口函数。

相电流接口函数，如下表所述：

函数名	形式参数	功能简述
Smp_Current_Update	CurrSmp_Struct_t	相电流过流判断

2.8.2 参数说明

类别	参数	类型	说明
CurrSmp_Struct_t	*s16I_U_Curr	int16_t	U 相电流
	*s16I_V_Curr	int16_t	V 相电流
	*s16I_W_Curr	int16_t	W 相电流
	*s16I_S_Curr	int16_t	母线电流
	s16IqRef_Add	int16_t	Iq 电流递增步长
	s16IqRef_Start	int16_t	Iq 启动电流
	s16IqRef_Dec	int16_t	Iq 电流递减步长
	s16I_Alpha	int16_t	$I\alpha$
	s16I_Beta	int16_t	$I\beta$
	s16I_OV	int16_t	过流阈值
	*s16I_Cu_d	Pid_Struct_t	d 轴电流

	*s16I_Cu_q	Pid_Struct_t	q 轴电流
	*smpCur1	int16_t	单电阻第一点
	*smpCur2	int16_t	单电阻第二点

2.8.3 电流处理

相电流接口函数在 TSK_FOCMotorControl()任务中进行调用，示例如下：

```

if(gVarTsk.tEna_flag.bits.cur_updata_EN)
{
    /* 插入通道电流采样 */
    SysInput_GetICH1Conv(gVarEsmc.pVarSysInput);
    SysInput_GetICH2Conv(gVarEsmc.pVarSysInput);
    SysInput_GetICH3Conv(gVarEsmc.pVarSysInput);
    Smp_Current_Update(gVarEsmc.pVarCurrSmp);
}
    
```

2.9 电压处理

2.9.1 函数列表

软件库提供母线电压处理接口函数，如下表所述：

函数名	形式参数	功能简述
BusV_Component_Init	BusV_Struct_t	组件初始化
BusV_AverBusV_Process	BusV_Struct_t	母线电压值滤波
BusV_FaultCheck	BusV_Struct_t	过压欠压判断

2.9.2 参数说明

类别	参数	类型	说明
BusV_Struct_t	u16CurrentConv	uint16_t	当前电压 AD 转换值
	u16Aver_BusV_d	uint16_t	平均电压值(H16)
	u16Aver_BusV_mv	uint16_t	平均电压值(mv)
	s16LPF_Factor	int16_t	滑动平均系数
	u16MaxLimitVoltage	uint16_t	过压阈值
	u16MinLimitVoltage	uint16_t	欠压阈值
	u16BusVFault	uint16_t	电压保护标志

2.9.3 电压处理

母线电压处理接口函数在低频任务中调用，示例如下：

```

TSK_SystemScheduler();
/* 电压处理，包括电压转换，过压和欠压处理 */
BusV_AverBusV_Process(gVarEsmc.pVarBusV, gVarEsmc.pVarSysInput->AD_VBUS.AD_Value);
TempSmp_AverT_Process(gVarEsmc.pVarTSmp, gVarEsmc.pVarSysInput->AD_Temp_MOS.AD_Value);
    
```

2.10 温度处理

2.10.1 函数列表

软件库提供温度采集与处理接口函数，如下表所述：

函数名	形式参数	功能简述
TempSmp_Component_Init	TemperSmp_Struct_t	组件初始化
TempSmp_AverT_Process	TemperSmp_Struct_t	温度 AD 值滤波并过温判断

2.10.2 参数说明

类别	参数	类型	说明
TemperSmp_Struct_t	u16Aver_TempSmp_d	uint16_t	温度采样平均值
	s16LPF_Factor	int16_t	滑动平均系数
	u16MaxLimitTemp	uint16_t	过温阈值
	u16TempSmpFault	uint16_t	过温 Fault 标志
	u16V0	uint16_t	温度曲线系数
	u16T0	uint16_t	温度曲线常数
	u8TempSmpChannel	uint8_t	温度 ADC 通道
	u8TempSmpTime	uint8_t	温度采样时间

2.10.3 温度处理

温度处理接口函数在低频任务中调用，示例如下：

```

TSK_SystemScheduler();
/* 电压处理，包括电压转换，过压和欠压处理 */
BusV_AverBusV_Process(gVarEsmc.pVarBusV, gVarEsmc.pVarSysInput->AD_VBUS.AD_Value);
/* 温度处理，包括温度保护 */
TempSmp_AverT_Process(gVarEsmc.pVarTSmp, gVarEsmc.pVarSysInput->AD_Temp_MOS.AD_Value);
    
```

2.11 功率管理

2.11.1 函数列表

软件库提供功率管理函数接口，如下表所述：

函数名	形式参数	功能简述
PowerMeas_Component_Init	PowerMeas_Struct_t	组件初始化
PowerMeas_CalcMotorPower	PowerMeas_Struct_t, BusV_Struct_t, CurrSmp_Struct_t	功率计算

2.11.2 参数说明

类别	参数	类型	说明
PowerMeas_Struct_t	s16PowerMeasBuffer[PM_BUFFER_LENTH]	int16_t	功率 buffer
	u16NextIndex	uint16_t	下次索引
	u16LastIndex	uint16_t	末次索引
	s16AverPower_W	int16_t	平均功率
	s16OverPower_H	int16_t	低功率阈值

	s16OverPower_L	int16_t	高功率阈值
	s32ConvFactor	int32_t	功率转换因子
	s16OverPower_CNT	int16_t	过功率次数

2. 12 错误管理

工程中源文件 `mid_state_faults_mgr.c` 负责对系统的错误状态进行管理，涉及到过温、过压、欠压以及速度错误等。该组件中结合对状态机的流程控制来达到对系统错误的管理，其中 `SYS_FAULT_OVER` 和 `SYS_FAULT_ON` 两个状态分别表示系统错误结束和错误正在发生。当系统检测到错误正在发生时进入 `SYS_FAULT_ON` 状态进行相应处理；当检测到错误条件失效将立即转入 `SYS_FAULT_OVER` 状态执行操作。

软件库提供错误管理函数接口，如下表所述：

函数名	形式参数	功能简述
<code>StateFault_Component_Init</code>	<code>StateFault_Struct_t</code>	组件初始化
<code>StateFault_FaultsProcess</code>	<code>StateFault_Struct_t, ErrType, BitMsk</code>	错误标志处理
<code>StateFault_GetSystemState</code>	<code>StateFault_Struct_t</code>	读系统错误状态
<code>StateFault_GetFaultsType</code>	<code>StateFault_Struct_t</code>	取错误类型
<code>StateFault_FaultsDisappearClear</code>	<code>StateFault_Struct_t</code>	清错误标志位

第3章 ES_MC_SDK软件调试

本章介绍无传感器的调试方法，采用无位置算法，解算出电机的实际转子角度和电机转速。

为了方便用户尽快掌握固件特性，及时发现每个环节存在的问题，以提高开发效率。在使用该固件库时，可按照以下步骤依次展开调试。

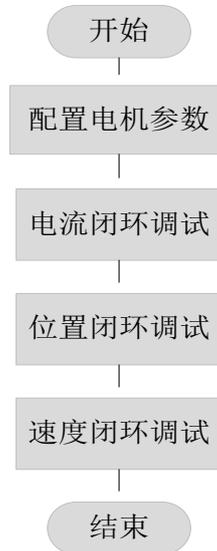


图 4-1 SDK 调试步骤框图

3.1 配置电机参数

首先使用此电机驱动需要根据手中需要驱动的电机属性进行必要的硬件参数添加，系统参数在头文件“esmc_paramconfig.h”中，电机参数在头文件“esmc_parammotor.h”中，硬件参数在头文件“esmc_hwconfig.h”中，其中需用户自行填写变更的参数，如下表所述：

需变更的硬件参数		
宏定义	注释	配置建议
MOTOR_PAIRS	极对数 (Poles)	TBD
MOTOR_RS	相电阻 (Ohm)	TBD
MOTOR_LS	相电感 (H)	TBD
MOTOR_MAX_RPM	最大机械转速(RPM)	TBD
DC_VOLTAGE	直流电压(V)	TBD
BUS_VOLTAGE_V	母线电压(V)	TBD
MCU_VOLTAGE_V	MCU 供电电压(V)	5V
AMP_OFFSET	电压偏置(V)	运放输入偏置电压，一般设计为 2.5V
UP_RESISTOR_OHM	母线电压电源端分压电阻 (Ω)	TBD
GND_RESISTOR_OHM	母线电压地端分压电阻 (Ω)	TBD
R_SHUNT_OHM	采样电阻阻值(mΩ)	TBD
OP_GAIN	放大倍数	根据运放电路的电阻值计算放大倍数
VBUS_RATIO	分压比	((UP_RESISTOR_OHM+GND_RESISTOR_OHM)

		/GND_RESISTOR_OHM)
VOLTAGE_MAX	硬件可支持最大电压(V)	填写好交流电压与分压电阻后自动换算
CURRENT_MAX	硬件可支持最大电流(A)	建议手动填写，以 ADC 运放输入电压偏置 (AMP_OFFSET)2.5V 为例： $\text{最大电流值} = \frac{2.5V}{\frac{OP_GAIN}{R_SHUNT_OHM}} (A)$
IQREF_START	启动电流(A)	根据启动负载情况，负载越大，启动电流越大，不能超过电机的最大运行电流。
IQREF_LIMIT	限制电流(A)	电机的最大运行电流
MOTOR_ACC_FIX	加速度(Hz/s)	填写 STARTRPM 与 ACCTIME 后自动换算
MOTOR_DEC_FIX	减速度(Hz/s)	填写 STARTRPM 与 ACCTIME 后自动换算
KLPF_BASE	截止频率(Hz)	一般设计为电机最大转速 RPM 对应的电频率 Hz
OVERPOWER	过功率(W)	要求的电机最大电压乘以最大电流
SMC_KSLIDE	滑模系数	需要在电机调试时手动调整
OVER_CURRENT	软件过流值(A)	软件过流保护设定值
MAX_VBUS_V	最大电压(V)：直流电压 +20%	软件过压保护设定值 默认直流电压+20%
MIN_VBUS_V	最小电压(V)：直流电压 -20%	软件欠压保护设定值 默认直流电压-20%
MOTOR_MAX_SPEED_HZ	最大电转速(round/s)	MOTOR_MAX_RPM*MOTOR_PAIRS/60
STARTRPM	启动机械转速(RPM)	TBD
ACCTIME	启动加速时间(ms)	TBD

3.2 配置PI参数

配置完成上述硬件参数后，还需配置软件 PI 参数，用户可在“esmc_State.c”中的 MC_Components_Init 中对下述表中的 PI 参数进行配置：

需变更的软件 PI 参数		
参数	注释	配置建议
s32IdKp	d 轴电流环 Kp	建议通过调试获得
s32IdKi	d 轴电流环 Ki	建议通过调试获得
s32IqKp	q 轴电流环 Kp	建议通过调试获得
s32IqKi	q 轴电流环 Ki	建议通过调试获得
s32EdKp	Ed 环 Kp	建议通过调试获得
s32EdKi	Ed 环 Ki	建议通过调试获得
s32SpdKp	速度环 Kp	建议通过调试获得
s32SpdKi	速度环 Ki	建议通过调试获得

3.3 电流闭环调试

首先是电流闭环，位置和速度保持开环，电机运行在拖动阶段 TSK_SysStart，表现为电机在设定的启动电流下保持按照给定的速度（这个速度程序中宏定义为最小速度 SPD_MIN_FIX，默认

设置为 50Hz) 开环运行。操作如下:

1. 设定好启动电流大小 (能够拖动当前负载就行)。
2. 注释掉拖动阶段程序 TSK_SysStart 中的阶段三切换语句, 保证程序不会进入到位置闭环。

```
void TSK_SysStart(void)
{
    PosFdbk_Struct_t InitPos;
    if (gVarTsk.Sys_State == SYS_START)
    {
        switch (gVarTsk.Sys_Stage)
        {
            case SYS_Stage1:
                /* Status Init */
                gVarTsk.tEna_flag.bits.Charge_EN = 0;
                gVarTsk.tEna_flag.bits.POS_EN = 0U;
                gVarTsk.tEna_flag.bits.Id_LOOP_EN = 1;
                gVarTsk.tEna_flag.bits.Iq_LOOP_EN = 1;
                gVarTsk.tEna_flag.bits.cur_updata_EN = 1; /*电流采样置位*/
                gVarTsk.tEna_flag.bits.PWM_OUT_EN = 1; /*PWM输出置位*/
                gVarTsk.tEna_flag.bits.Omega_EN = 1; /*启动位置估算, 但不使用*/
                gVarTsk.InitAngle = 0;
                gVarTsk.p_sysTimer.wTicks = 0;
                gVarTsk.Sys_Stage = SYS_Stage2;

            case SYS_Stage2:
                if(gVarTsk.InitAngle <32767)
                {
                    gVarTsk.InitAngle += 20;
                    InitPos.sl6ElecAngle = gVarTsk.InitAngle;
                    GetRotorElecAngle(&InitPos);
                    gVarEsmc.pVarTorSpd->pSpdFdbk->s32OmegaComFix.LONG = DataLimit(gVarEsmc.pVarTorSpd->pSp

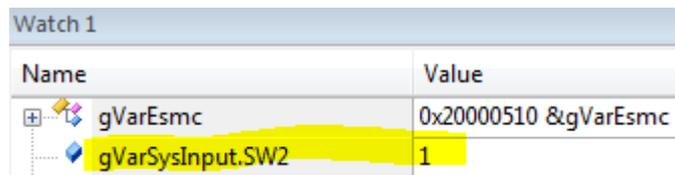
                    gVarEsmc.pVarCurrSmp->s16I_Cu_q->Target = (gVarEsmc.pVarCurrSmp->s16I
                    gVarEsmc.pVarCurrSmp->s16I_Cu_d->Target = gVarEsmc.pVarCurrSmp->s16I

                }

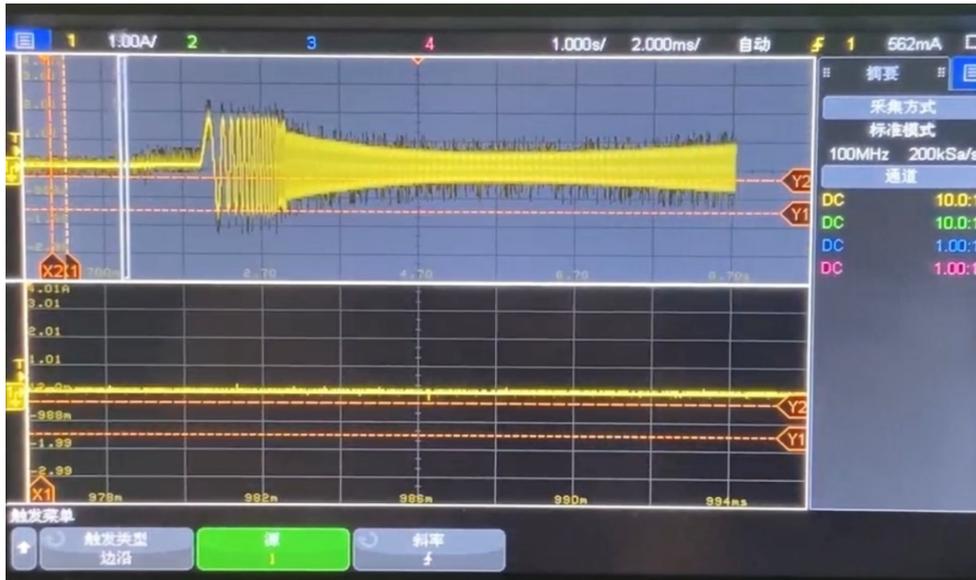
                /*电机开环拖动, 根据电机自行调整时间*/
                if(++gVarTsk.p_sysTimer.wTicks >= 5500)
                {
                    gVarTsk.p_sysTimer.wTicks = 0;
                    gVarEsmc.pVarSmc->s32Ialpha_Est = (*gVarEsmc.pVarSmc->s16Ialpha);
                    gVarEsmc.pVarSmc->pEdPID->s32Integral = gVarEsmc.pVarTorSpd->pSpdFdb
                    gVarEsmc.pVarSmc->Spd_LPF.LPFOutput_Last = gVarEsmc.pVarTorSpd->pSpd
                    //gVarTsk.Sys_Stage = SYS_Stage3;
                }
                break;

            case SYS_Stage3:
                gVarEsmc.pVarTorSpd->pSpdFdbk->s32OmegaComFix.LONG = DataLimit(gVarEsmc.
```

3. Debug 模式运行电机, live watch 中将 gVarSysInput.SW2 设置为 0, 启动并运行电机。



4. 调整 live watch 中 gVarPITest 中的电流环参数, 使电机稳定运行, 电流可从振荡逐渐收敛, 如下图所示。



3.4 位置闭环调试

在电流环能够支持电机开环运行的条件下，开始调整位置 PI 以及滑模系数。

1. 首先将启动电流放的尽量小，因为这个阶段调试位置闭环一般会运行在当前给定电流和负载所能支持的最大速度，电流过大会使电机运行超出额定速度很多，并震荡。
2. 去除拖动阶段程序 TSK_SysStart 中的阶段三切换语句注释，将阶段四切换的语句注释掉。以免进入速度闭环控制。

```

/*电机开环拖动，根据电机自行调整时间*/
if(++gVarTsk.p_sysTimer.wTicks >= 5500)
{
    gVarTsk.p_sysTimer.wTicks = 0;
    gVarEsmc.pVarSmc->s32Ialpha_Est = (*gVarEsmc.pVarSmc->s16Ialpha);
    gVarEsmc.pVarSmc->pEdPID->s32Integral = gVarEsmc.pVarTorSpd->pSpdFdbk->s32OmegaComFix.LONG;
    gVarEsmc.pVarSmc->Spd_LPF.LPFoutput_Last = gVarEsmc.pVarTorSpd->pSpdFdbk->s32OmegaComFix.Word.HWord;
    gVarTsk.Sys_Stage = SYS_Stage3;
}
break;

case SYS_Stage3:
gVarEsmc.pVarTorSpd->pSpdFdbk->s32OmegaComFix.LONG = DataLimit(gVarEsmc.pVarTorSpd->pSpdFdbk->s32OmegaMin
    gVarEsmc.pVarTorSpd->pSpdFdbk->s32AccFix, gVarEsmc.pVarTorSpd->pSpdFdbk->s32DecFix);
gVarTsk.tEna_flag.bits.ElecAngle_update_EN = 1; /*位置启用*/
if (++gVarTsk.p_sysTimer.wTicks >= c500ms)
{
    gVarEsmc.pVarSmc->pEdPID->s32Integral = gVarEsmc.pVarTorSpd->pSpdFdbk->s32OmegaComFix.LONG;
    gVarEsmc.pVarSmc->Spd_LPF.LPFoutput_Last = gVarEsmc.pVarTorSpd->pSpdFdbk->s32OmegaComFix.Word.HWord;
    gVarTsk.p_sysTimer.wTicks = 0U;
    gVarTsk.Sys_Stage = SYS_Stage4;
}
break;

case SYS_Stage4:
gVarEsmc.pVarTorSpd->pSpdFdbk->s32OmegaComFix.LONG = DataLimit(gVarEsmc.pVarTorSpd->pSpdFdbk->s32OmegaMin
    gVarEsmc.pVarTorSpd->pSpdFdbk->s32AccFix, gVarEsmc.pVarTorSpd->pSpdFdbk->s32DecFix);

if (++gVarTsk.p_sysTimer.wTicks >= c100ms)
{
    gVarTsk.p_sysTimer.wTicks = 0U;
    gVarEsmc.pVarSpeedPid->s32Integral = (gVarEsmc.pVarCurrSmp->s16IqRef_Start) << 16U;
    //gVarTsk.Sys_Stage = SYS_StageEnd;
}
}
    
```

3. Debug 模式运行电机，live watch 中将 gVarSysInput.SW2 设置为 0，启动并运行电机。
4. 调整 live watch 中 gVarPITest 中的电流环和位置环参数，使电机稳定运行。

5. 调整 live watch 中滑模系数，使电机稳定运行。

注意滑模系数在默认值基础上修改，不宜过大也不宜过小，过小会降低电机运行效率，过大会导致电机电流波动明显。

3.5 速度闭环调试

1. 去除上述注释，电机直接三环运行，调整三环 PI，使电机在当前负载下稳定运行。
2. 带载测试，如果电机在带载情况下，无法正常启动或者难以从开环状态下切入闭环应适当增大启动电流 IQREF_START 值。
3. 另外可以根据不同速度使用不同组 PI。

第4章 ES_MC_SDK硬件平台

4.1 低压硬件平台

硬件平台由电源电路、功率驱动电路、位置检测电路、电流采样电路、通信电路以及保护电路等模块构成，其结构框图如下图所示。

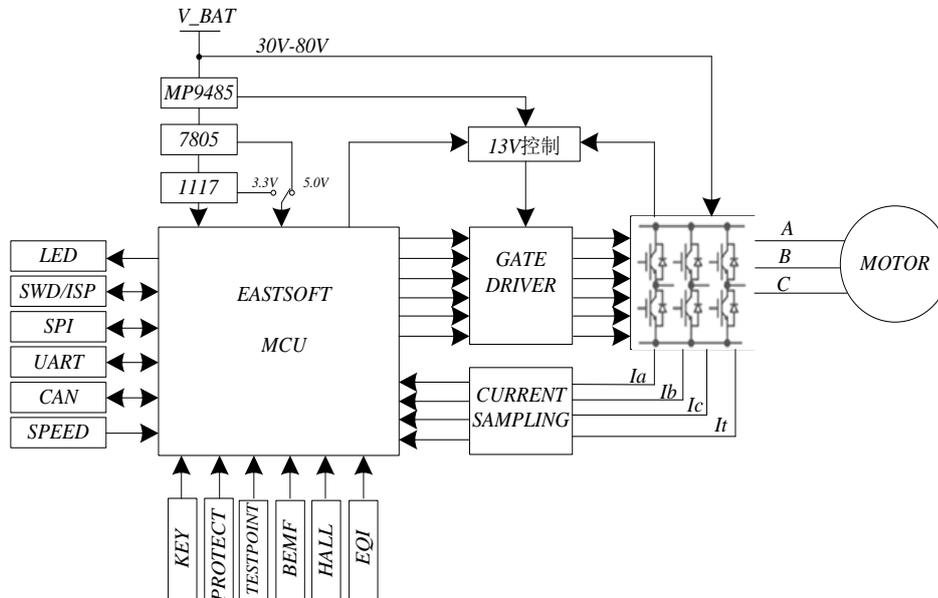


图 3-1 低压硬件平台电路结构框图

ESD-MOTO_LVD 低压驱动开发板有如下特点：

- 支持 HALL/ENCODER 传感器 FOC 控制
- 支持无传感器 BLDC/FOC 控制方式
- 支持无传感器转子位置观测矢量控制方式
- 软硬件过流/过欠压/相间短路/过温保护
- 支持 SPI/UART/RS485/CAN 通信
- 支持 SWD 调试协议
- 正反转/启动/刹车开关
- 调速旋钮、按键和 LED 指示
- 额定值为 100V/192A 的三相逆变桥
- 电压输入范围 30V~80V
- MCU 供电电压 5V/3.3V

4.2 高压硬件平台

硬件平台由电源电路、功率驱动电路、位置检测电路、电流采样电路、通信电路以及保护电路等模块构成，其结构框图如下图所示。

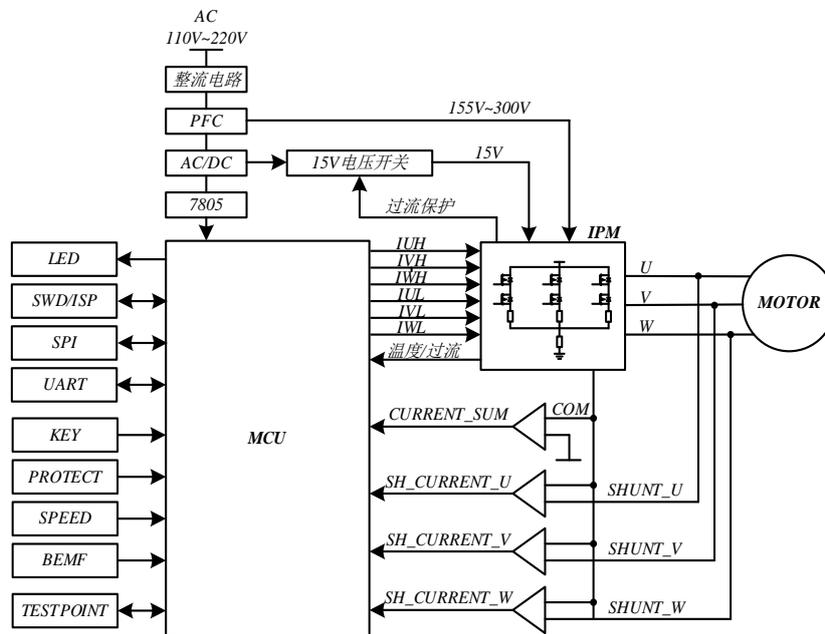


图 3-2 高压硬件平台电路结构框图

ESD-MOTO_HVD 高压驱动开发板有如下特点：

- 支持 HALL 传感器
- 支持无传感器 BLDC/FOC 控制方式
- 支持无传感器转子位置观测矢量控制方式
- 软硬件过流/过欠压/相间短路/过温保护
- 支持 SPI/UART 通信
- 支持 SWD 调试协议
- 调速旋钮、按键和 LED 指示
- 额定值为 20A 的 IPM 三相逆变桥模块
- 电压输入范围为 AC 110V~220V
- MCU 供电电压 5V