

SPD1148 低压吸尘器调试指南使用说明书

版本 A/0 - 2022 年 10 月

前言

本使用说明书面向客户、市场以及研发人员，提供有关 SPD1148 低压吸尘器调试指南。

文档编号	RC-109-00002	
产品名称	SPD1148 低压吸尘器调试指南	
编制	罗敏	2022-08-07
审核	罗敏	2022-10-07
批准	李涛	2022-10-07

本文件包含保留信息并归旋智科技（深圳）有限公司所有。
未经旋智科技（深圳）有限公司授权，本文件不得拷贝，传播或对外泄漏。

目录

1	SPD1148 低压吸尘器方案介绍	6
2	方案介绍	7
2.1	程序文件结构	7
2.2	电路参数设定	8
2.3	电机参数设定	8
2.4	控制参数设定	9
2.5	启动控制参数设定	10
2.6	FG 输出设定	10
2.7	PWM 信号作为指令输入设定	10
2.8	保护设定	11
2.8.1	短路保护设定	11
2.8.2	母线过压、欠压保护设定	11
2.8.3	堵转保护设定	12
2.8.4	断线保护设定	12
2.8.5	过温保护设定	12
2.8.6	SPD1148 其他自带保护设定	13
2.8.6.1	SPD1148 自带过压欠压保护实现关机保护（掉电保护设定）	13
2.9	保护自恢复设定	14
2.10	MBD 及其他设定	15
2.11	硬件设计简单介绍	19
3	主要调试方法	21
3.1.1	芯片工作测试	21
3.1.2	PWM 输出测试	24
3.1.3	电流采样测试	24
3.1.4	电机启动测试	26
3.1.5	闭环调试	27
3.1.6	ADC 引脚更换	28
3.1.6.1	电流采样引脚更换	28
3.1.6.2	电压、NTC 采样引脚更换	29
3.1.6.3	反电势采样引脚更换	29
3.1.6.4	母线电流采样引脚	30
3.1.6.4.1	单端采样	30
3.1.6.4.2	差分采样	31

4	串口调试指令	31
5	SEGGER RTT 串口调试	32
5.1	设置方法.....	33
5.1.1	代码配置.....	33
5.1.2	工具配置.....	33
5.1.3	使用.....	33
6	接口函数	35
7	附录	35
7.1	程序框架及中断流程.....	35
7.2	功率计算及母线电流采样.....	36
7.2.1	母线电流差分采样.....	36
7.2.2	母线电流差分采样.....	36
7.2.3	母线电流单端采样.....	37
7.2.3.1	VDD33 补偿.....	37
7.2.3.2	等效采样电阻补偿.....	39
7.2.3.3	等效采样电阻温度补偿.....	39
7.3	启动逻辑.....	40
7.3.1	顺风逻辑.....	40
7.3.2	静止逻辑.....	41
8	MODEL BASE DESIGN 介绍	42
8.1	总体框架.....	42
8.2	各个模块介绍.....	42
8.2.1	电路模型.....	42
8.2.2	PMSM 模型.....	43
8.2.3	状态机模型.....	43
8.2.4	FOC 控制系统模型.....	44
8.2.5	参数设置.....	44
8.3	调试.....	45
8.3.1	电压开环调试.....	45
8.3.2	速度/功率闭环调试.....	47
8.4	代码生成.....	48

表格列表

表 2-1: 各文件夹下程序功能分类.....	7
表 2-2: 系统基本参数表.....	8
表 2-3: PWM 及采样参数表.....	8
表 2-4: 电机电气参数表.....	8
表 2-5: 控制方式参数表.....	9
表 2-6: 电机控制环路参数表.....	9
表 2-7: 电机开环控制参数表.....	10
表 2-8: FG 功能参数表.....	10
表 2-9: PWM 输入信号参数表.....	10
表 2-10: 短路保护参数表格.....	11
表 2-11: 母线过压、欠压保护设定参数表.....	11
表 2-12: 堵转保护设定表.....	12
表 2-13: 断线保护设定参数.....	12
表 2-14: 过温保护设定参数表 I.....	12
表 2-15: 过温保护参数设定表 II.....	12
表 2-16: 过温保护参数设定表 III.....	12
表 2-17: 掉电保护参数表.....	13
表 2-18: 掉电保护阈值表.....	14
表 2-19: 保护回复配置表.....	14
表 2-20: MBD 参数列表.....	15
表 3-1: 开环启动相关配置参数表.....	27
表 3-2: 电流环带宽参数配置表.....	27
表 3-3: 速度环 PI 参数配置表.....	27
表 3-4: 功率环参数配置表.....	28
表 3-5: 反电势采样配置参数表.....	29
表 3-6: 母线电流采样方式配置参数表.....	30
表 3-7: 母线电流单端采样参样配置表.....	31
表 4-1: 串口调试指令表.....	31
表 4-2: 串口调试指令档位配置表.....	32
表 7-1: 预定义接口函数表.....	35
表 8-1: 预定义定时任务表.....	35
表 8-2: 预定义定时任务及功能表.....	35
表 8-3: 母线采样电阻补偿系数参数表.....	39
表 8-4: 常用材料电阻率和电阻温度系数表.....	40
表 8-5: 采样电阻温度补偿系数参数表.....	40
表 8-6: 顺风启动检查参数表.....	41
表 8-7: 顺风启动判断阈值表.....	41

版本历史

版本	日期	作者	变更
A/0	2022 年 8 月 7 日		首次发布。

1 SPD1148 低压吸尘器方案介绍

该低压吸尘器方案是旋智针对用锂电池供电的吸尘器这一应用专门优化定制的，其中主要定制的内容如下：

- 1) 控制芯片采用 SPD1148
- 2) 采样方式为单电阻
- 3) 控制方式为恒功率控制
- 4) 带有限流、过流、母线过压、母线欠压、堵转、缺相、过温等保护
- 5) 最大支持开关频率 30K，最大转速可达到 20W 电转速，适用于单对极、双对极的高速吸尘器。
- 6) 增加弱磁控制，保证最大真空度
- 7) 待机功耗小于 10uA

2 方案介绍

2.1 程序文件结构

图 2-1: 程序文件结构图

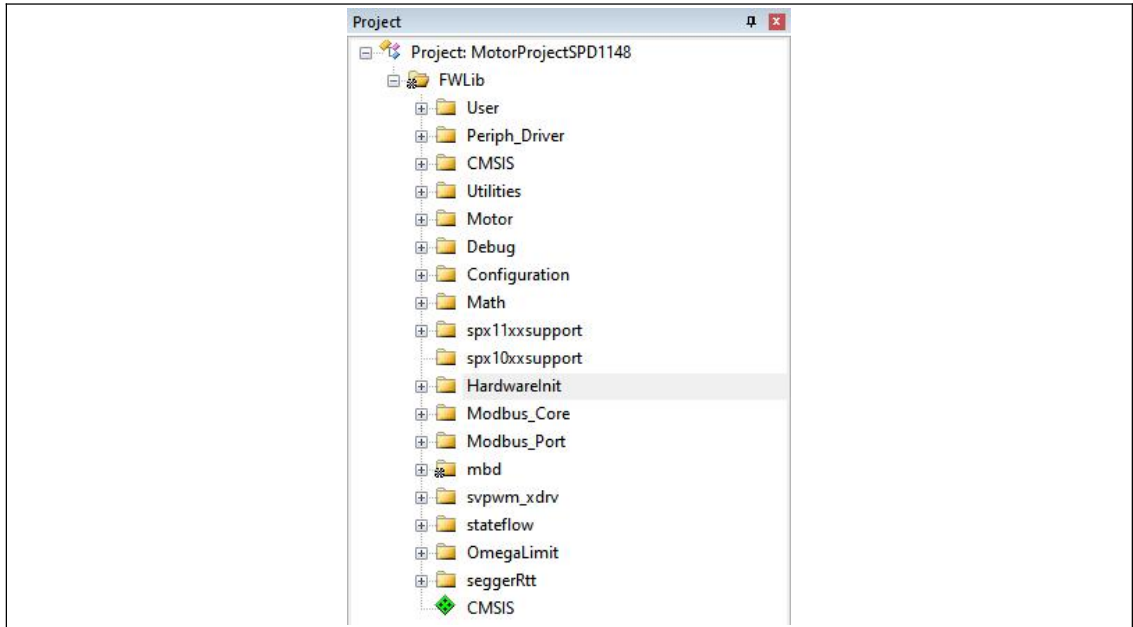


表 2-1: 各文件夹下程序功能分类

文件夹	简单说明
User	用户程序，包括 main 及主中断
Periph_Driver	外设驱动
CMSIS	ARM 专用驱动
Utilities	小的实用程序
Motor	电机控制相关程序
Debug	用于 debug 的程序
Configuration	电机配置文件夹
Math	数学函数
Modbus_Core/Modbus_Port	Modbus 相关 code
Mdb	通过 MBD 方式生成的 FOC 控制代码
Stateflow	通过 MBD 方式生成的状态机控制代码
seggerRtt	seggerRtt 相关代码
HardwareInit	硬件配置相关
Spx11xxsupport	芯片功率部分配置相关
OmegaLimit	为过速保护预留代码

2.2 电路参数设定

1) 以下参数的设定在 motor_sys_config_basic.h 中，是系统的一些基本参数

表 2-2: 系统基本参数表

参数	解释	单位
F32_MOTOR_SYS_VDCBUS_VOLT	系统额定母线电压	V
F32_MOTOR_SYS_SHUNT_RESISTOR_OHM	采样电阻大小	Ω
F32_SHUNT_RES_ADJ	采样电阻校准系数	
I32_MOTOR_SYS_PGA_GAIN	片上运放放大倍数	无
MOTOR_SYS_PWM_SWITCHING_FREQ_HZ	采样信号震荡时间，一般情况下大于 1us，小于 3us	Hz
BEMF_MEASURE_ENABLE	是否启用反电势采样电路	
BEMF_MEASURE_UPPER_RES_OHM	反电势采样电路上端电阻	Ω
BEMF_MEASURE_LOWER_RES_OHM	反电势采样电路下端电阻	Ω
BEMF_MEASURE_CAPACITOR_F	反电势采样电路滤波电容	F

2) 以下参数的设定在 motor_sys_config_basic.h 中，主要会影响采样电路的工作

表 2-3: PWM 及采样参数表

参数	解释	单位
CIRCUIT_DEAD_TIME_S	死区时间	s
CIRCUIT_PWM_DELAY_TIME_S	PWM 开关延迟时间，于功率器件相关	s
CIRCUIT_SAMPLE_TIME_S	采样时间，大于 ADC 的采样时间	s
CIRCUIT_SAMPLE_VIBRATE_TIME_S	采样信号震荡时间，一般情况下大于 1us，小于 3us	s
IC_SYS_FREQ_HZ	MCU 工作频率	Hz

2.3 电机参数设定

以下参数的设定在 motor_sys_config_basic.h 中，是电机本身的一些参数，需要设定的值如下：

表 2-4: 电机电气参数表

参数	解释	单位
F32_MOTOR_POLEPAIRS	电机极对数	无
F32_MOTOR_REAL_PHASE_R_OHM	电机相电阻	Ω
F32_MOTOR_REAL_PHASE_LD_H	电机相电感最小值	H
F32_MOTOR_REAL_PHASE_LQ_H	电机相电感最大值	H
F32_MOTOR_REAL_PHASE_L_H	测量反电势的频率	Hz
MOTOR_BEMF_VL2L_VOLT	测量反电势的峰峰值	V

MOTOR_BEMF_FREQ_HZ	电机相电阻	Ω
F32_MOTOR_INERTIA	电机的转动惯量	$\text{Kg}\cdot\text{m}^2/\text{N}\cdot\text{m}\cdot\text{s}^2$

2.4 控制参数设定

1) 以下参数的设定在 motor_sys_config_basic.h 中，用于选择控制方式，

表 2-5: 控制方式参数表

参数	解释	单位
UART_MODE	UART 控制模式，一般使用模式 3 (1/2/3)	无
EXTERNAL_SWITCH_ENABLE	是否使外部 VSP 控制 (0/1)	无
DIR_CMD_ENABLE	是否使用方向控制 (0/1)	无
ON_OFF_CMD_ENABLE	是否使用开关控制 (0/1)	无

2) 以下参数的设定在 motor_sys_config_basic.h 中，用于选择不同的控制模式

表 2-6: 电机控制环路参数表

参数	解释	单位
SPEED_LOOP_ENABLE	是否使用转速闭环控制 (0/1)	无
POWER_LOOP_ENABLE	是否使用功率闭环控制 (0/1)	无
DUTY_LOOP_ENABLE	是否使用占空比闭环控制 (0/1)	无
TORQUE_LOOP_ENABLE	是否使用电流闭环控制 (0/1)	无
VOLTAGE_LOOP_ENABLE	是否使用电压闭环控制 (0/1)	无
MOTOR_MAX_SPEED	速度闭环时候的最大转速	Erpm
MOTOR_MIN_SPEED	速度闭环时候的最小转速	Erpm
MOTOR_MAX_POWER	功率闭环时候的最大功率	W
MOTOR_MIN_POWER	功率闭环时候的最小功率	W
MOTOR_MAX_DUTY	占空比闭环时候的最大占空比，暂时无效，暂时无效	%
MOTOR_MIN_DUTY	占空比闭环时候的最小占空比	%
MOTOR_MAX_CURRENT	闭环时候的最大电流，这个值对功率、速度、占空比和电流闭环也都有效。	A
MOTOR_MIN_CURRENT	电流闭环时候的最小电流	A
MOTOR_MAX_VQ	电压闭环时候的最大电压，暂时无效	
MOTOR_MIN_VQ	电压闭环时候的最小电压，暂时无效	
POWER_LIMIT_ENABLE	是否使能功率限制，对速度、占空比及电流环有效 (0/1)，暂时无效	
POWER_LIMIT_POWER_W	功率限定值	W

2.5 启动控制参数设定

以下参数的设定在 motor_sys_config_basic.h 中，用于开环控制的设定

表 2-7: 电机开环控制参数表

参数	解释	单位
MOTOR_OPENLOOP_IMAX_A	开环最大电流	A
MOTOR_OPENLOOP_ENTER_RAD	开环进入频率	Rad/S
MOTOR_OPENLOOP_LEAVE_RAD	开环退出频率	Rad/S
MOTOR_PSI_ADJ_STARTUP_COEF	低速阶段反电势系数调整系数	
MOTOR_CURRENT_LIMIT_STARTUP_A	低速阶段 Q 轴电流限制值	A
MOTOR_SPEED_LIMIT_STARTUP_RAD	低速阶段频率阈值	Rad/S

2.6 FG 输出设定

以下参数的设定在 motor_sys_config_basic.h 中，用于使能及控制 FG 输出信号

表 2-8: FG 功能参数表

参数	解释	单位
FG_OUTPUT_ENABLE	是否使能 FG 输出 (0/1)	无
FG_PULSE_PER_CYCLE	每个电转速周期输出脉冲数	无

2.7 PWM 信号作为指令输入设定

以下参数的设定在 motor_sys_config_basic.h 中，用于使能 PWM 输入信号

表 2-9: PWM 输入信号参数表

参数	解释	单位
PWM_PPM_INPUT_ENABLE	是否使能 PWM 输入 (0/1)	无
PWM_COUNTER_PER_CYCLE	PWM 信号静默检测周期，如长时间无 PWM 输入信号，则根据高低电平来判断 PWM 输入占空比	无

2.8 保护设定

2.8.1 短路保护设定

短路保护，主要是防止上下桥直通或者电机本身短路。

在 motor_sys_config_basic.h 中

表 2-10: 短路保护参数表格

参数	解释	单位
OVER_CURRENT_ENABLE	是否打开短路保护，打开：1，关闭：0	
F32_MOTOR_SYS_PHASE_OVERCURRENT_A	短路保护阈值	A

2.8.2 母线过压、欠压保护设定

在吸尘器应用里头，采用了两方面的过压、欠压保护设定，一个是通过 ADC 采集母线电压来做电压保护，另外一部分是用 SPD1148 本身集成的电压保护来做；两部分是同时作用的，所以都需要配置。

1) 通过 ADC 采集的电压保护

过压保护和欠压保护都是通过设计了一个滞环来控制。

在 motor_sys_config_basic.h 中，通过更改下面的值来配置这两个保护。

需要注意的是因为电机运行时候，功率较大，容易造成电压跌落，所以欠压保护分成了停止时候（F32_MOTOR_SYS_UNDER_VOLTAGE_V）和运行时候（F32_MOTOR_SYS_OVER_VOLTAGE_V_RUN）两个保护阈值，一般情况下，运行时候电压保护值较高一些。

表 2-11: 母线过压、欠压保护设定参数表

参数	解释	单位
OVER_VOLTAGE_ENABLE	是否打开母线过压保护，打开：1，关闭：0	
F32_MOTOR_SYS_OVER_VOLTAGE_RELEASE_V	母线过压触发值	V
F32_MOTOR_SYS_UNDER_VOLTAGE_V	母线过压清除值	V
UNDER_VOLTAGE_ENABLE	是否打开母线欠压保护，打开：1，关闭：0	
F32_MOTOR_SYS_UNDER_VOLTAGE_V	电机停止时（启动开机时候），母线欠压触发值	V
F32_MOTOR_SYS_OVER_VOLTAGE_V_RUN	电机运行时，母线欠压触发值	V
F32_MOTOR_SYS_UNDER_VOLTAGE_RELEASE_V	母线欠压清除值	V

2) SPD1148 集成的电压保护

请参考 SPD1148 的保护设定

2.8.3 堵转保护设定

堵转保护主要用于防止电机因为被异常负载卡住后造成长时间转不起来的现象。

在 motor_sys_config_basic.h 中，通过以下值来启用堵转保护

表 2-12: 堵转保护设定表

参数	解释
STALL_DETECTION_EANBLE	是否打开堵转保护，打开：1，关闭：0

2.8.4 断线保护设定

断线保护用于防止电机线断掉一根或者两根之后出现转动异常的情况

在 motor_sys_config_basic.h 中，通过以下值来启用断线保护

表 2-13: 断线保护设定参数

参数	解释
MISSING_PHASE_DETECTION_ENABLE	是否打开堵转保护，打开：1，关闭：0

需要注意的是，断线的保护有时候报堵转保护。

2.8.5 过温保护设定

温度保护的启动可以使用下面参数来打开

表 2-14: 过温保护设定参数表 I

参数	解释
OVER_TEMPERATURE_ENABLE	是否打开温度保护，打开：1，关闭：0

该方案的过温保护采取两级保护：

- 1) SOC 本身的问题保护，使用 SOC 内部的 TSE 来实现保护，需要设定以下两个值

表 2-15: 过温保护参数设定表 II

参数	解释
INNER_TSE_OT_TRIGGER_THRESHOLD_DEG	外部 NTC 温度保护触发的 ADC 阈值
INNER_TSE_OT_RELEASE_THRESHOLD_DEG	外部 NTC 温度保护清除的 ADC 阈值

- 2) MOS、PCBA 温度保护，采取板载的 NTC 来保护，NTC 串联一个 4.7kOhm 的电阻分压后输入到 SOC 的 ADC 引脚，通过 ADC 的读数来进行相应保护。

需要设定以下两个值：

表 2-16: 过温保护参数设定表 III

参数	解释
OVER_TEMPERATURE_NTC_TRIGGER_THRESHOLD_DIG	外部 NTC 温度保护触发的 ADC 阈值
OVER_TEMPERATURE_NTC_RELEASE_THRESHOLD_DIG	外部 NTC 温度保护清除的 ADC 阈值

因为 NTC 是通过外部一个 3.3V 的分压电路的电压输入到 SPD1148 中的，所以对应的阈值先要通过查表看对应温度 NTC 的阻值来确认。

具体 ADC 的阈值可以通过以下公式得到

$$\text{ADC_Value} = R_NTC / (4.7k + R_NTC) * 3.3 * 8192$$

在吸尘器应用中，当温度过高，采取的措施是直接停机

2.8.6 SPD1148 其他自带保护设定

SPD1148 作为一颗高集成的 SOC，内部也带有很多的 protection，其中在吸尘器的设计中用到的有过压、欠压和过温保护。

目前用于需要自己修改这部分代码来修改保护阈值。

2.8.6.1 SPD1148 自带过压欠压保护实现关机保护（掉电保护设定）

在吸尘器等锂电池应用中，因为常常存在直接掉电的情况，这时候母线电压会迅速（几 ms 内）掉到 0，这时候要增加一个掉电保护，掉电保护可以通过以下参数来打开。

表 2-17: 掉电保护参数表

参数	解释
SHUTDOWN_PROTECTION_ENABLE	是否打开温度保护，打开：1，关闭：0

SPD1148 内部自带母线电压比较器，可以实现 us 级别的保护。

图 2-2: 掉电保护配置图

```

124
125 void Vbat_ProtectionConfig(void)
126 {
127     ErrorStatus      eErrorState;
128     uint8_t          u8PREDRIDATA;
129     // BOD protection
130     /* Enable T20 signal of PRE-DRIVER when BOD protection event happened */
131     //HV_WriteReg(HV_REG_PDRVZOSELO, PDRVZOSELO_ALL_VBATL_ENABLE | PDRVZOSELO_ALL_VDDGL_ENABLE);
132     HV_WriteReg(HV_REG_PDRVZOSELO, PDRVZOSELO_ALL_VBATL_ENABLE | PDRVZOSELO_ALL_VBATH_ENABLE | PDR
133
134     /* Set VBAT protection too-high and too-low threshold */
135     HV_WriteReg(HV_REG_BODVBATVTH, BODVBATVTH_ALL_VMIN_9dot5_V | BODVBATVTH_ALL_VMAX_34dot2_V);
136
137     /* Set VDDG protection high and low threshold */
138     HV_WriteReg(HV_REG_BODVDDGVTH, BODVDDGVTH_ALL_VMIN_8dot0_V | BODVDDGVTH_ALL_VMAX_14dot2_V);
139
140     // Clear status register before we start to read the status register.
141     HV_WriteReg(HV_REG_GLBSTSCLR0, GLBSTSCLR0_ALL_VBATL_CLEAR | GLBSTSCLR0_ALL_VBATH_CLEAR
142                | GLBSTSCLR0_ALL_VDDGL_CLEAR
143                | GLBSTSCLR0_ALL_VDDGH_CLEAR);

```

用户可以通过设置 `BODVBATVTH_ALL_VMIN_XXdotX_V` 来设定关机保护时候的欠压保护阈值，这个阈值有如下几档可供选择。

表 2-18: 掉电保护阈值表

参数	解释
BODVBATVTH_ALL_VMIN_5dot0_V	欠压保护阈值 5.0V
BODVBATVTH_ALL_VMIN_6dot0_V	欠压保护阈值 6.0V
BODVBATVTH_ALL_VMIN_7dot3_V	欠压保护阈值 7.3V
BODVBATVTH_ALL_VMIN_9dot5_V	欠压保护阈值 9.5V
BODVBATVTH_ALL_VMIN_12dot8_V	欠压保护阈值 12.8V
BODVBATVTH_ALL_VMIN_16dot5_V	欠压保护阈值 16.5V
BODVBATVTH_ALL_VMIN_17dot5_V	欠压保护阈值 17.5V
BODVBATVTH_ALL_VMIN_23dot0_V	欠压保护阈值 23.0V

2.9 保护自恢复设定

若用户希望告警的条件消失后，告警能够自动消失，可以打开告警自恢复功能。

表 2-19: 保护回复配置表

参数	解释
OVER_CURRENT_RECOVERY_ENABLE	是否使能过流保护自恢复，打开：1，关闭：0
OVER_VOLTAGE_RECOVERY_ENABLE	是否使能过压保护自恢复，打开：1，关闭：0
UNDER_VOLTAGE_RECOVERY_ENABLE	是否使能欠压保护自恢复，打开：1，关闭：0
STALL_DETECTION_RECOVERY_ENABLE	是否使能堵转保护自恢复，打开：1，关闭：0
MISSING_PHASE_RECOVERY_ENABLE	是否使能缺相保护自恢复，打开：1，关闭：0
PDRV_VDS_MONITOR_RECOVERY_ENABLE	是否使能 VDS 保护自恢复，打开：1，关闭：0
OVER_SPEED_RECOVERY_ENABLE	是否使能超速保护自恢复，打开：1，关闭：0
OVER_TEMPERATURE_RECOVERY_ENABLE	是否使能过温保护自恢复，打开：1，关闭：0
SHUTDOWN_RECOVERY_ENABLE	是否使能关机保护自恢复，打开：1，关闭：0

2.10 MBD 及其他设定

以上设定，其中有一部分会在以下函数中完成或者传值进入 code 中完成。

```
void Motor_InitParameter(struct Motor_T* p)
```

比如如下设定就通过传值完成。

图 2-3: MBD 模型参数传值方法图

```
p->sSysParams.fDeadT = CIRCUIT_DEAD_TIME_S;
p->sSysParams.fDelayT = CIRCUIT_PWM_DELAY_TIME_S;
p->sSysParams.fPWMFreqHz = MOTOR_SYS_CURRENT_SAMPLING_FREQ_HZ;
p->sSysParams.fSampleT = CIRCUIT_SAMPLE_TIME_S;
p->sSysParams.fSysHz = IC_SYS_FREQ_HZ;
p->sSysParams.fTpwm = 1.0f / p->sSysParams.fPWMFreqHz;
p->sSysParams.fTSys = 1.0f / p->sSysParams.fSysHz;
p->sSysParams.fVibrationT = CIRCUIT_SAMPLE_VIBRATE_TIME_S;
```

还有些设定需要手动完成的，比如如下参数。

下表是 MBD 生成的 Code 中的常用参数及它们的解释：

表 2-20: MBD 参数列表

时间参数	解释	单位
PMSM_FOC_P.ts_calc_MCU	计算周期时间	S
PMSM_FOC_P.ts_calc_obs	观测器计算周期	S
PMSM_FOC_P.ts_calc_spd	速度环/功率环/速度规划计算周期	S
电机参数		
PMSM_FOC_P.Rs	电机电阻	Ω
PMSM_FOC_P.Ld	电机直轴电感	H
PMSM_FOC_P.Lq	电机交轴电感	H
PMSM_FOC_P.psir	电机反电势系数	V/(Rad/s)
PMSM_FOC_P.acctimeS	转速规划，加速到最大转速时间	S
PMSM_FOC_P.dectimeS	转速规划，降低到 0 转速时间	S
弱磁控制参数		
PMSM_FOC_P.FW_ENABLE	是否使能弱磁	
PMSM_FOC_P.flux_wkn_kp	弱磁控制 KP	
PMSM_FOC_P.flux_wkn_ki	弱磁控制 KI	
PMSM_FOC_P.fw_angle_max_pu	弱磁控制最大超前角度	Pu
PMSM_FOC_P.fw_us_max	弱磁控制门限电压	Pu
开环定位参数		
PMSM_FOC_P.idc_pos_time		
PMSM_FOC_P.idc_pos_ref	开环定位电流大小	A
开环参数		
PMSM_FOC_P.opnlp_acc_time	开环加速时间	S
PMSM_FOC_P.openloop_curr_iq_refA	开环 D 轴电流大小	A
PMSM_FOC_P.sw_wr_hi	开环跳出速度	Rad/s

PMSM_FOC_P.sw_wr_lo	开环进入速度	Rad/s
电流环参数		
PMSM_FOC_P.cur_ctrl_kp	电流环 KP	
PMSM_FOC_P.cur_ctrl_ki	电流环 KI	
PMSM_FOC_P.oneOvercur_ctrl_kp		
PMSM_FOC_P.cur_ctrl_out_max	电流环输出最大电压指令	Pu
功率环参数		
PMSM_FOC_P.pwr_kp	功率环 KP	
PMSM_FOC_P.pwr_ki	功率环 KI	
PMSM_FOC_P.IdcLimitA	直流母线电流限制，该功能需要在 MDB 模型中增加	
PMSM_FOC_P.LoopIndex	控制环路选取，1：速度环，2：功率环等外环	
速度环参数		
PMSM_FOC_P.spd_ctrl_kp	速度环 KP	
PMSM_FOC_P.spd_ctrl_ki	速度环 KI	
PMSM_FOC_P.spdPIAbsMaxA	速度环输出电流绝对值最大	A
PMSM_FOC_P.spdPIAbsMinA	速度环输出电流绝对值最小	A
电流指令参数		
PMSM_FOC_P.IdRefMaxA	直轴电流最大指令	A
PMSM_FOC_P.IdRefMinA	直轴电流最小指令	A
PMSM_FOC_P.IqRefMaxA	交轴电流最大指令	A
PMSM_FOC_P.IqRefMinA	交轴电流最大指令	A
PMSM_FOC_P.cmd_LPF_CutoffFreqRadS	直轴电流指令滤波器参数	Rad/s
观测器参数		
PMSM_FOC_P.ObsserverGain	观测器增益	
PMSM_FOC_P.BWRatioMax	观测器增益增大系数	
PMSM_FOC_P.WrSwitchRadS	观测器增益变化频率阈值	Rad/s
PMSM_FOC_P.we_adapt_kp	观测器锁相环 KP	
PMSM_FOC_P.we_adapt_ki	观测器锁相环 KI	
PMSM_FOC_P.ObslDelay	观测器延迟	观测器周期
PMSM_FOC_P.UoutDelay	电压输出延迟	观测器周期
PMSM_FOC_P.psiStartup	低速反电势系数	
PMSM_FOC_P.psiTransFreqRad	低速反电势系数变化速度阈值	Rad/s
PMSM_FOC_P.Wr_Max_Limit	观测器最大速度	Rad/s
PMSM_FOC_P.Wr_Min_Limit	观测器最小速度	Rad/s
PMSM_FOC_P.gParaVoltageCompensateUdRatio	观测器输入电压：直轴增益	
PMSM_FOC_P.gParaVoltageCompensateUqRatio	观测器输入电压：交轴增益	
速度规划、时间等参数		
PMSM_FOC_P.maxfreqHz	速度规划，最大频率	Hz
PMSM_FOC_P.motor_start_mode	电机工作模式，暂时无用	
PMSM_FOC_P.LowSpeedThresholdRadS	低速工作速度阈值	Rad/s
PMSM_FOC_P.LowSpeedIqLimitA	低速工作电流最大值	A
PMSM_FOC_P.spd_lower_lmt	速度环最小指令绝对值	A

PMSM_FOC_P.spd_upper_lmt	速度环最大指令绝对值	A
PMSM_FOC_P.sw_wr_mag_hi	高速模式进入阈值，进入此模式后，直轴上的开环的电流给定将逐渐衰减	Rad/s
PMSM_FOC_P.sw_wr_mag_lo	低速模式进入阈值，进入此模式后，直轴上的开环的电流给定将逐渐增加	Rad/s
PMSM_FOC_P.StopBrakeEnable	是否使能关机刹车，如果使能，则关机后会短下桥刹车	
PMSM_FOC_P.GeneralFaultBrakeEnable	遇到一般告警后是否刹车，如果使能，则遇到告警后会刹车	
PMSM_FOC_P.gParaBootTestS	下桥充电时间	S
PMSM_FOC_P.gParaStartDetctTimeS	启动检测时间	S
PMSM_FOC_P.gParaStopWaitTimeS	停机等待刹车时间，如果使能刹车，则等待这段时间后再刹车	S
PMSM_FOC_P.gParaBrakeTimeS	刹车时长	S
飞车启动参数		
PMSM_FOC_P.gParaFlyStartDetectTimeS	飞车检测时长	S
PMSM_FOC_P.gParaFlyStartEmfPu	进入顺风/逆风启动的反电势阈值	Pu
PMSM_FOC_P.gParaFlyStartSpeedThresholdRad	进入顺风/逆风启动的速度阈值	Rad/s
PMSM_FOC_P.gParaFlyStartZeroCurrentControlTimeS	飞车启动后零电流控制时间	S
PMSM_FOC_P.emf_detect_kp	飞车启动观测器 KP	
PMSM_FOC_P.emf_detect_ki	飞车启动观测器 KI	
PMSM_FOC_P.gParaFlyStartCurrentThresholdA	当没有反电势电路时，飞车启动的电流阈值	A
PMSM_FOC_P.gParaFlyStartCurrentGain	飞车启动电流增益，因为飞车启动只是开很短一段时间下桥，电流很小，可根据下桥开通占空比来设定一个增益值	
死区补偿参数		
PMSM_FOC_P.DT_Comp_Is_filt_fc_rad_HI	死区补偿电流滤波频率	
PMSM_FOC_P.DT_Comp_ThetaLPF_cutoff_Coef	死区补偿电流二次滤波系数（a），滤波频率最终值为 $a \cdot \omega_n$	
PMSM_FOC_P.DT_Comp_DeadTimeLost	死区补偿电流二次滤波系数（a），滤波频率最终值为 $a \cdot \omega_n$	
PMSM_FOC_P.DT_Comp_Coef_LPF_CutoffFreqRadS	死区补偿衰减滤波频率	
PMSM_FOC_P.DT_Comp_AmpLimitLowPu	死区补偿衰减进入占空比，高于此占空比，死区补偿将逐渐减小	Pu
PMSM_FOC_P.DT_Comp_AmpLimitHighPu	死区补偿完全衰减占空比，高于此占空比，死区补偿将完全消失	Pu
PMSM_FOC_P.boolIqloopEnable	电流环使能，如果使能，电流环指令将通过直接给定而不是速度环输出	
PMSM_FOC_P.boolLookToThetaOpenLoop	角度开环使能	
PMSM_FOC_P.boolLookToVoltageOpenLoop	电压开环使能	
PMSM_FOC_U.IdRefA	电流开环时，直轴电流给定值	A
PMSM_FOC_U.IqRefA	电流开环时，交轴电流给定值	A

PMSM_FOC_U.VdRefPu	电压开环时，直轴电压给定值	Pu
PMSM_FOC_U.VdRefPu	电压开环时，交轴电压给定值	Pu
状态机参数		
SpintrolMotorStateFlow_U.CmdInput	状态机输入指令	
SpintrolMotorStateFlow_P.MaxCmd	状态机输出指令最大值	
SpintrolMotorStateFlow_P.MinCmd	状态机输出指令最小值	

2.11 硬件设计简单介绍

1) SPD1148

SPD1148 配置:

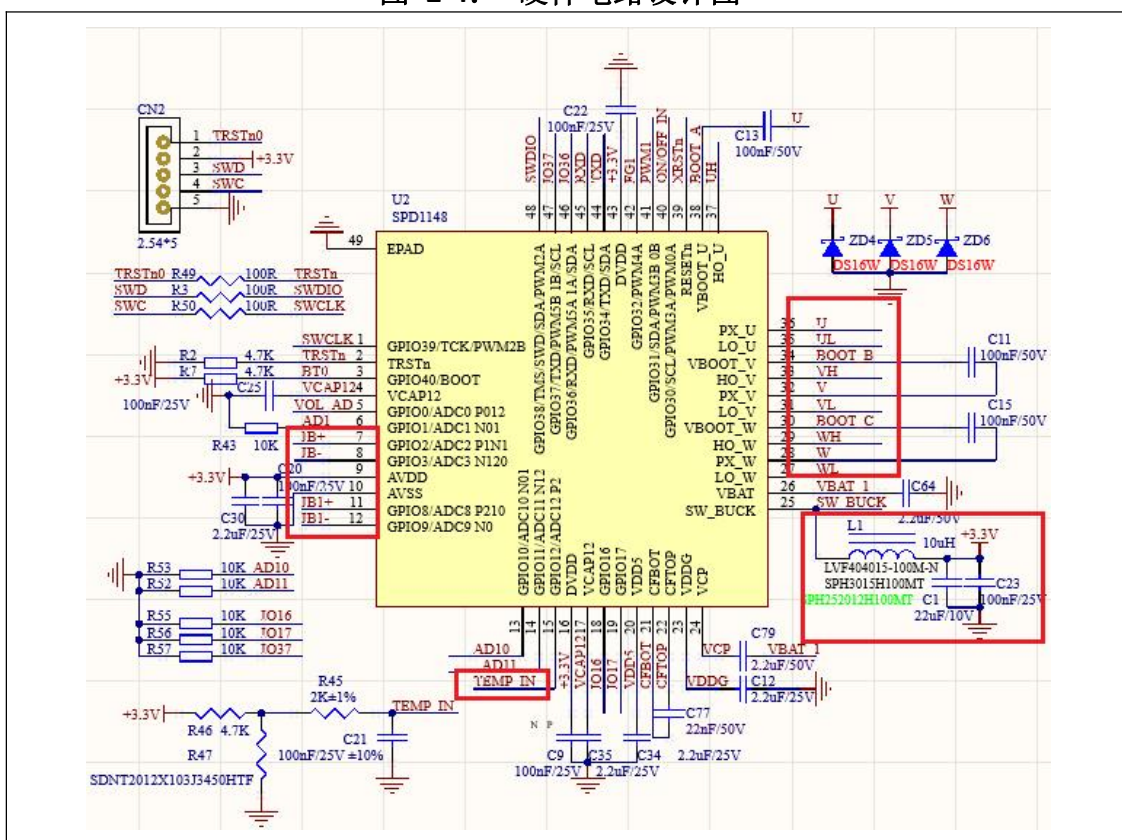
Buck 电路: 用于把 15V 电压转换为 3.3V 供芯片使用

PWM 输出: 控制后级驱动电路

电流采样: 单电阻方式用于采样电机相电流

其他采样电路: 母线电压采样、NTC 采样

图 2-4: 硬件电路设计图



2) 驱动电路

SPD1148 自带最大耐压 40V 的驱动电路，且集成了自举二极管

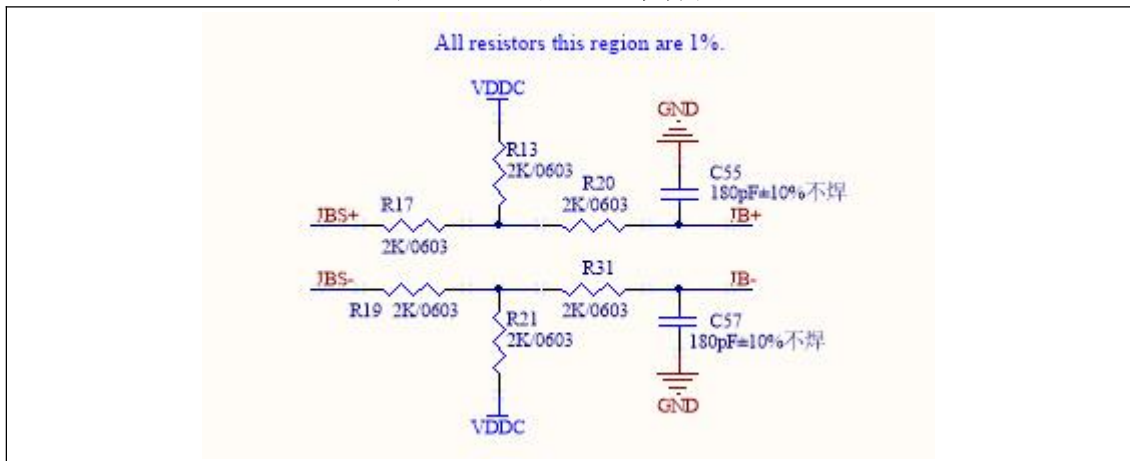
3) 电压采样

SPD1148 芯片内部自带了母线电压检测部分，电路上也集成了外部的母线电压检测。当对电压精度要求较高时候，建议采用外部电路。

4) 电流采样电路

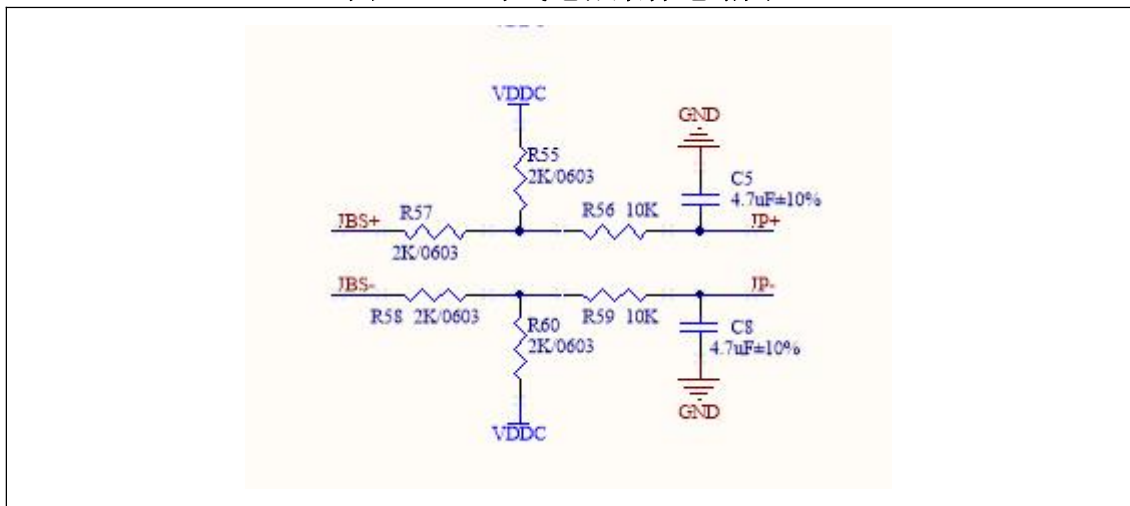
a. 用于电机控制的电流采样部分，采用全差分的电流采样电路，滤波时间常数较小

图 2-5: 相电流采样图



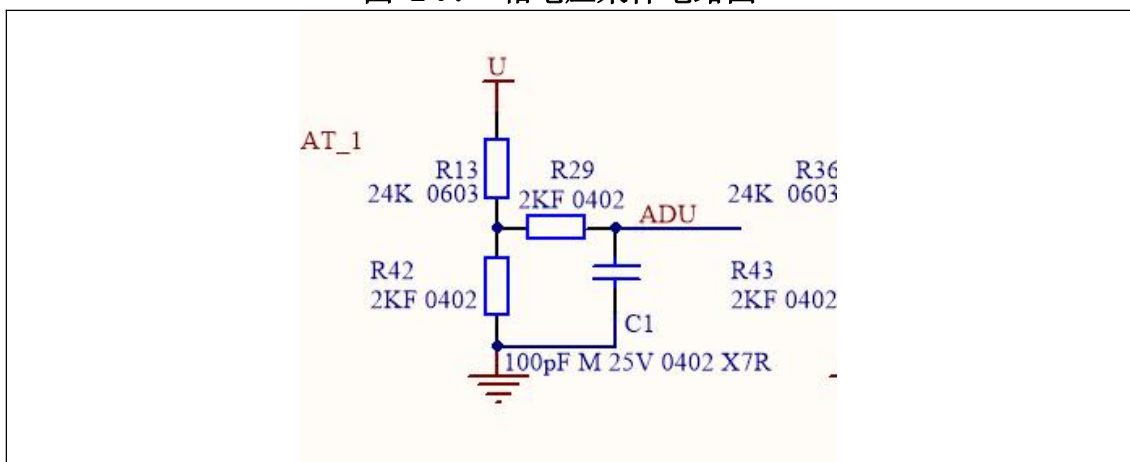
b.用于电流控制部分的采样电路，同样采用全差分电路，但是滤波时间较大

图 2-6: 母线电流采样电路图



5) 反电势采样电路

图 2-7: 相电压采样电路图

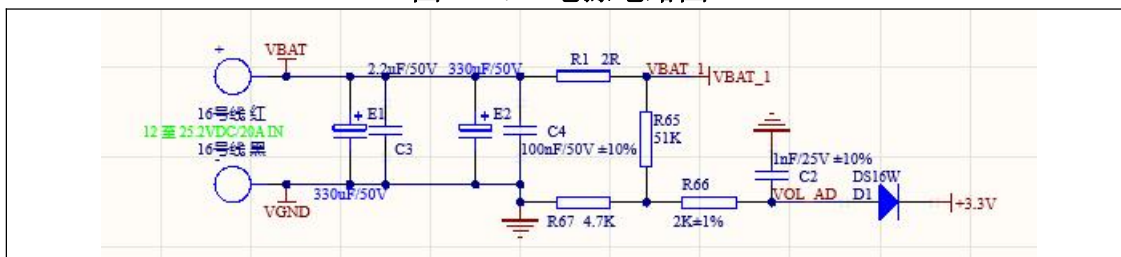


采用电阻分压的方法获取反电势电压

6) 电源电路

输入电压为电池电压，直接接到母线电容上。

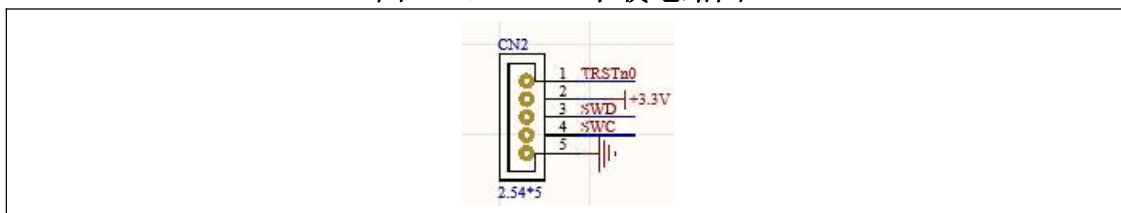
图 2-8: 电源电路图



7) SWD 下载电路

使用 4 线的 SWD 接口，用于程序下载

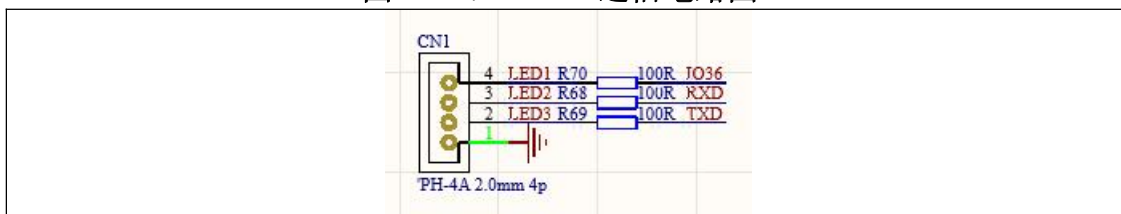
图 2-9: SWD 下载电路图



8) UART 调试电路

UART 通信接口如下所示

图 2-10: UART 通信电路图



3 主要调试方法

3.1.1 芯片工作测试

通电前检查:

首先观察板子本身是否有物理损坏，3.3V 到地，VBAT 到地，VDDG 对地是否短路

PCB 板通电，VBAT 上电压大于 6V，检查以下几点:

- 1) 3.3V 是否正常:

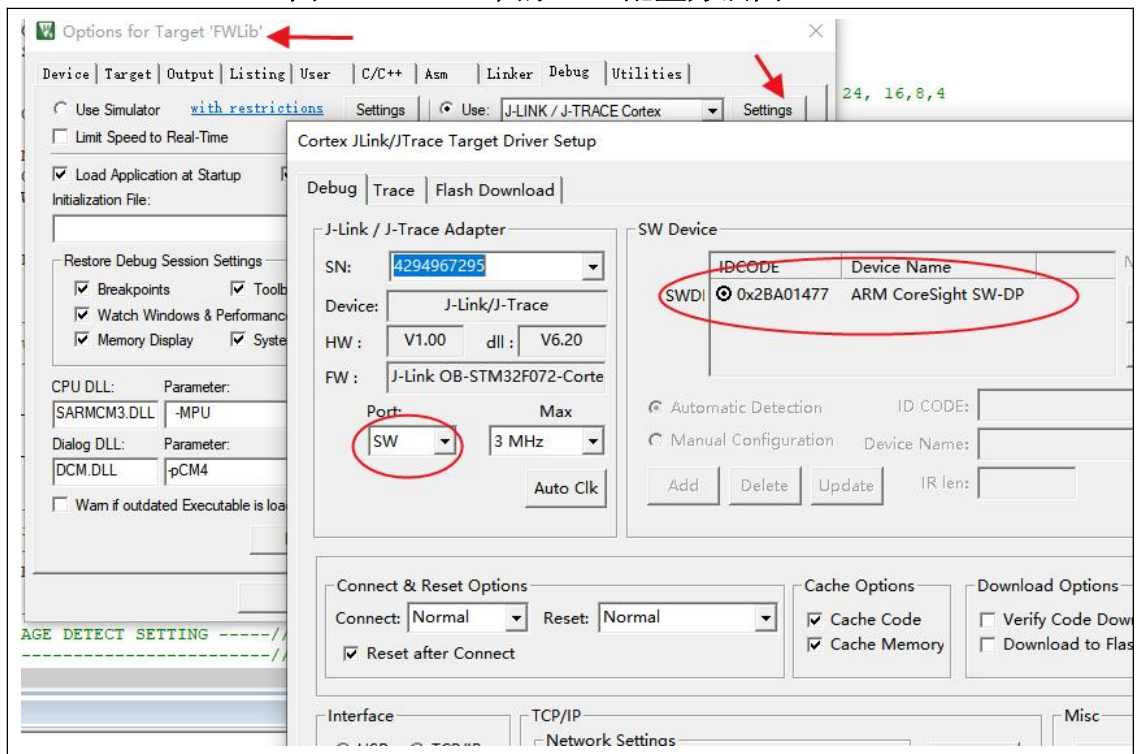
万用表量测 VDD，应该量到 3.3V 左右

2) 1.2V 是否正常：

万用表量测 VDD12，应该量到 3.3V 左右

3) 连接 SWD 接口到 Jlink 等调试器，打开 Keil-->Flash-->Configure Flash Tool，检查是否识别到 SPD1148。

图 3-1: Keil 中的 Jlink 配置方法图



4) Keil 中编译 Code 并下载到芯片。

5) 串口连接检查

该程序默认提供一个波特率 115200bps 的 UART 接口，在程序中 UART_DEBUG_ENABLE 设定为 1 的时候，可以通过这个接口看到一些打印信息。

图 3-2: 串口调试使能开关图

```

250 #define F32_RPM_RAD_S_RATIO (0.10472) // rpm to rad/s ratio
251 #else
252 #define F32_MOTOR_TORQUE_INERTIA_RATIO 300 // arbitrary parameter for torque coefficient / ine
253 #endif
254
255 #define SPEED_GAIN_AUTO 0
256
257 #if SPEED_GAIN_AUTO == 1
258 #define PI_CONTROL_SPEED_DAMPING 3 // range: 2-30, higher means slower response b
259 #define PI_CONTROL_SPEED_FILTER_BW 200 // smo speed filter cutoff bandwidth
260 //
261 #endif
262
263 #define PI_CONTROL_DUTY_I_GAIN ((int32_t){})
264 #define PI_CONTROL_DUTY_P_GAIN ((int32_t){0.8*32767})
265
266
267 #define PI_CONTROL_POWER_I_GAIN ((int32_t){0.001*32767}) // KI of Power loop
268 #define PI_CONTROL_POWER_P_GAIN ((int32_t){0.1*32767}) // Kp of Power loop
269
270 #define PI_CONTROL_VOLTAGE_I_GAIN ((int32_t){0.001*32767}) // KI of Power loop
271 #define PI_CONTROL_VOLTAGE_P_GAIN ((int32_t){0.3*32767}) // Kp of Power loop
272 //-----//
273 //-----//
274 //-----//
275 #define UART_DEBUG_ENABLE
276
277
278 // The following codes are to avoid configuration problem
279 #if ((SPEED_LOOP_ENABLE + DUTY_LOOP_ENABLE + POWER_LOOP_ENABLE + TORQUE_LOOP_ENABLE + VOLTAGE_LOOP_ENABLE)>1) ||
    
```

通过串口调试工具连接（3.3V, GND, RX, TX），电路板上电，在正常情况下，串口调试工具上应该开到一些打印信息。

输入'q'命令，可以看到打印电压电流等信息。

通过输入'q'指令，可以看到当前目前电压，请检查是否和母线电压一致

图 3-3: 串口调试示意图

```

COM16:115200baud - Tera Term VT
File Edit Setup Control Window KanjiCode Help
Stall Counter0 speed 0 3336 392
*****Reboot*****
U zero -303 V zero 0
Motor STOP
BT
CK
ST
Data Ready
current Power is 8.530432W
Vbus Voltage is 303.974634V
Applied Voltage is 26.248970 percent
I is 0.127925A
Mechanical Speed is 515RPM, Electrical Speed is 2060RPM
meager state 1
u32Inner_Temper_AVG = 1345
T-sensor Measure Temperature is 34 oC
NTC value 3067
I U Zero=-303 I_V_Zero=0 I_W_Zero=0
UART_Cmd=q
current Power is 8.868750W
Vbus Voltage is 304.198749V
Applied Voltage is 27.134007 percent
I is 0.125553A
Mechanical Speed is 566RPM, Electrical Speed is 2266RPM
    
```

3.1.2 PWM 输出测试

- 1) 设定系统工作在电压闭环（代码中如果设定电压环状态，则会自动配置为角度开环状态）；
- 2) 关闭除过流告警外的其他告警

图 3-4： 开环测试告警配置图

```

184
185 #define CONTINUOUS_START_TEST          0
186
187 //----- Protection Configuration -----//
188 //----- Protection Configuration -----//
189 //----- Protection Configuration -----//
190 #define OVER_CURRENT_ENABLE            1 // This is to protect
191 #define OVER_VOLTAGE_ENABLE           0
192 #define UNDER_VOLTAGE_ENABLE         0
193 #define STALL_DETECTION_ENABLE        0
194 #define MISSING_PHASE_DETECTION_ENABLE 0
195 #define OVER_TEMPERATURE_ENABLE       0
196 #define INNER_TEMPERATURE_ENABLE      0
197 #define VDDG_UV_PROTECT_ENABLE        0
198 #define VDDG_OV_PROTECT_ENABLE        0
199 #define ADC_UNBALANCE_PROTECTION_ENABLE 0
200
201

```

- 3) 不连接电机，发送电压指令，发送电机启动指令；
- 4) 测量 U、V、W 三相预驱输出，是否带有死区的 PWM 波，然后检查 U、V、W 三相的输出端，是否也是 PWM 波。
- 5) 如果 PWM 输出不正确，检查硬件参数及连接。

3.1.3 电流采样测试

- 1) 关闭堵转告警；
- 2) 设定系统工作在电压闭环（代码中如果设定电压环状态，则会自动配置为角度开环状态）；

```
#define VOLTAGE_LOOP_ENABLE            1
```

- 3) 关闭死区时间补偿

```
#define DEADTIME_COMPENSATE_ENABLE    0
```

- 4) 在 debug.c 中配置采集三相电流

```

p->api32Addr[0] = (int32_t*)&myMotor[0].sCoreSignal.f32IuA;
p->api32Addr[1] = (int32_t*)&myMotor[0].sCoreSignal.f32IvA;
p->api32Addr[2] = (int32_t*)&myMotor[0].sCoreSignal.f32IwA;

```

- 5) 设定采集总数、采集间隔及每行数据

图 3-5: 采样数据配置示意图

```

12 #define _DEBUG_H_
13
14 #include "motor_software_config.h"
15
16 #define DEBUG_BUFFER_RECORD_PERIOD 20 // Timer interval for data collection, e.g, if 2, every 2 PWM
17 // Information Store in RAM(Buffer) -//
18 #define TOTAL_RECDED_LENGTH 8000 // How many data totally will be stored
19 #define NUM_RECORD_VAR 13 // For ecah line, how many datas
20
21 #define DEBUG_WITH_INTERNAL_BUFFER_ENABLE 1
22 #define DEBUG_WITH_PWM_CARRIER_ENABLE 0
23 #define DEBUG_WITH_DAC_ENABLE 0
24
25 #define RECORD_LENGTH (TOTAL_RECDED_LENGTH/NUM_RECORD_VAR)
26
27 typedef enum
28 t

```

6) 连接电机

7) 通过'o'/'p'发送电压指令(注意不要太大, 否则容易导致过流), 送指令 t 让电机转动, 发送'r'指令, 开始记录数据, 记录满采集数据总量后停止, 并打印“Data Ready”, 发送'R'指令, 会把记录到的数据

8) 示波器采集相电流波形, 软件中采集收集到的电流信号, 通过 excel 表格观察数据

图 3-6: 开环运行相电流波形图

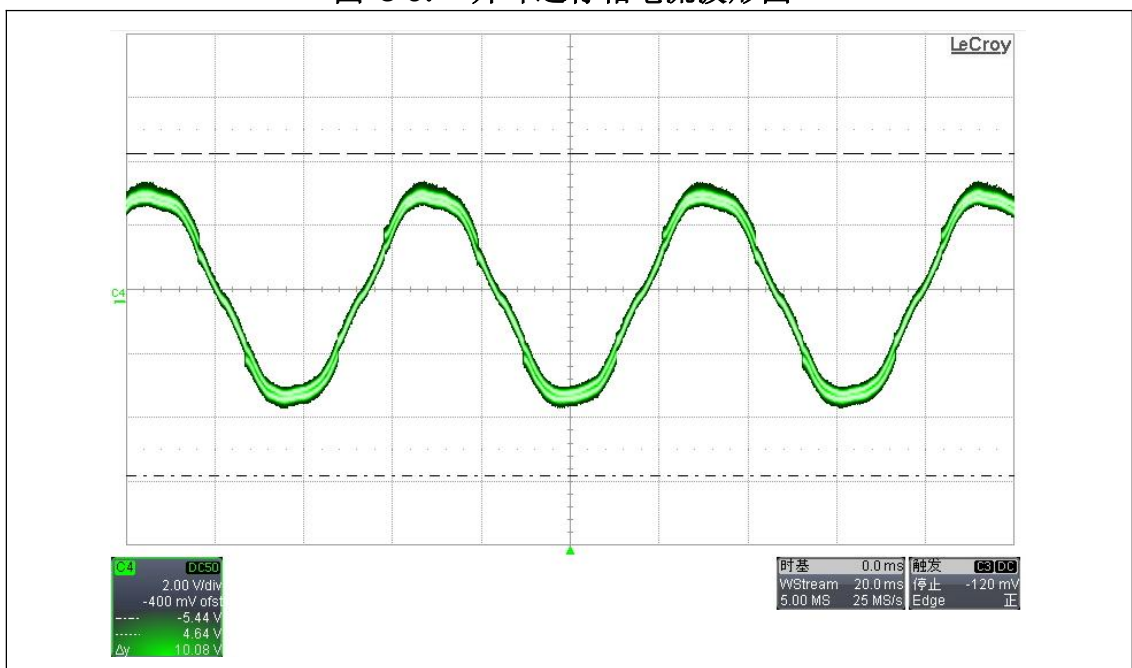
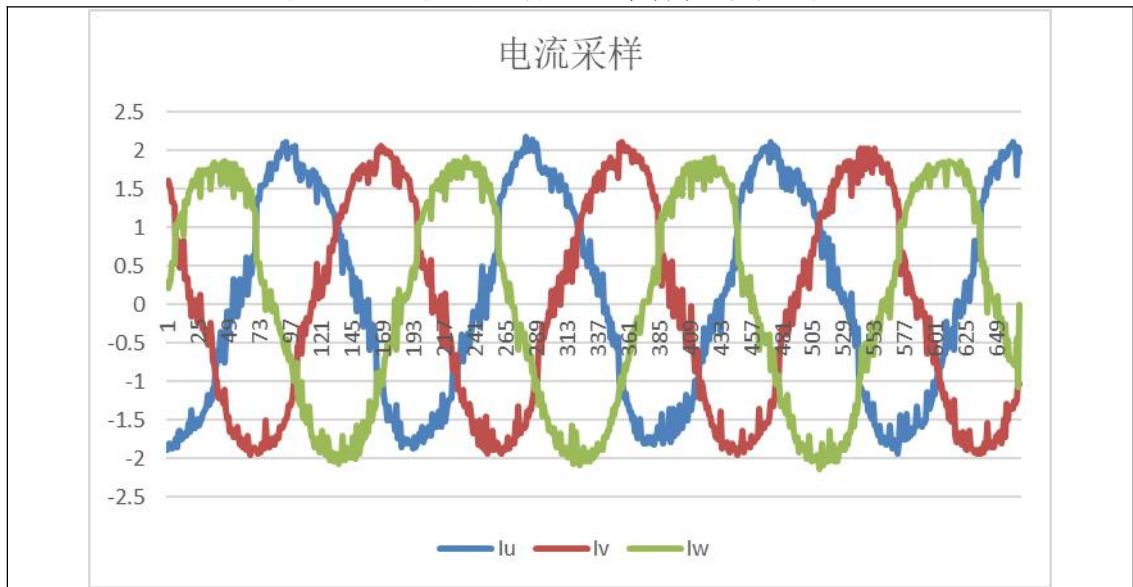


图 3-7: 开环运行电流采样信号示意图



实际电流应该呈稳定的正弦状态，采集到的电流也应该在幅值、频率和相位上跟实际电流相接近。

如果采集回的电流在形状上有较大差距，则判断为单电阻采样配置异常，一方面检查硬件电流，软件上检查单电阻采样参数是否配置正确：

```
#define CIRCUIT_DEAD_TIME_S          1000.0e-9f
// dead time
#define CIRCUIT_PWM_DELAY_TIME_S     50.0e-9f
// PWM on/off delay time
#define CIRCUIT_SAMPLE_TIME_S        300.0e-9f
// Sample time, should be a little higher than ADC sample time, as to give some
margin
#define CIRCUIT_SAMPLE_VIBRATE_TIME_S 1.0e-6f
// signal vibrate time, usually higher than 1us, maybe as high as 3 us.
#define IC_SYS_FREQ_HZ                200.0e6f
// The MCU clock
```

如果采集回的电流在幅值上有较大差距，则判断为采样电阻设置不正确（因为布板已经采样电阻本身的影响，尤其是对阻值很小的采样电阻，阻值往往会存在一定偏差），这时可以调整下面参数来匹配：

```
#define F32_MOTOR_SYS_SHUNT_RESISTOR_OHM
(0.002f * F32_SHUNT_RES_ADJ)
```

3.1.4 电机启动测试

- 1) 按照应用要求，设定电机参数、运行参数、保护参数等；
- 2) 空载启动电机，看是否能够正常启动；

3) 如果不能正常启动,则需要调解以下参数:

表 3-1: 开环启动相关配置参数表

解释	定义	位置	解释
开环电流	MOTOR_OPENLOOP_IMAX_A	motor_sys_config_basic.h	开环时候最大 D 轴电流给定
开环进入频率	MOTOR_OPENLOOP_ENTER_RAD	motor_sys_config_basic.h	从闭环进入开环频率
开环退出频率	MOTOR_OPENLOOP_LEAVE_RAD	motor_sys_config_basic.h	从开环进入闭环频率
启动反电势调整系数	MOTOR_PSI_ADJ_STARTUP_COEF	motor_sys_config_basic.h	W 低速阶段反电势系数调整系数,因为低速阶段死区等非线性因素的影响,观测器并不准确,可以通过调整这一系数来补偿

4) 对于吸尘器等应用,建议在不同电压下,分别测试启动,做到在所有工作电压范围内,都可以满足启动成功率。

5) 堵转测试以保证启动成功率

在启动过程中,可以用一些外力手段,卡住风轮叶片,然后再启动,如果松开后,都能保证启动,那么启动的成功率就有较高的保证。

3.1.5 闭环调试

闭环控制相关参数调试

1) 电流环参数调试

电流环作为内环控制,其控制器的 PI 参数需要设置较大的带宽。带宽越大对电流调节的响应越快,调节时间越短。

表 3-2: 电流环带宽参数配置表

参数	解释	默认值	单位
CURRENT_LOOP_BANDWIDTH_HZ	电流环 PI 带宽	500	Hz

2) 速度环参数

表 3-3: 速度环 PI 参数配置表

参数	解释	默认值	单位
SPEED_LOOP_KP	速度环 PI Kp		
SPEED_LOOP_KI	速度环 PI Ki		

3) 功率环参数调试

功率环主要用于根据功率误差给定速度环指令，其调节参数如下：

K_p 为比例系数，值越大则调节的增益越大，响应时间越小，太大会出现震荡，表现为高频不稳定或者出现报过流等问题。

K_i 为积分系数，值越大则积分越快，表现为功率提升较快，太大了容易出现超调及较低频率不稳定，可以适当减小或者增加 K_p 。

表 3-4: 功率环参数配置表

参数	解释	默认值	单位
POWER_LOOP_KP	功率环 PI 比例系数	无	无
POWER_LOOP_KI	功率环 PI 积分系数	无	无

4) 弱磁环参数调试

表 3-5: 弱磁 PI 参数配置表

参数	解释	默认值	单位
PMSM_FOC_P.flux_wkn_kp	弱磁控制 KP	无	无
PMSM_FOC_P.flux_wkn_ki	弱磁控制 KI	无	无

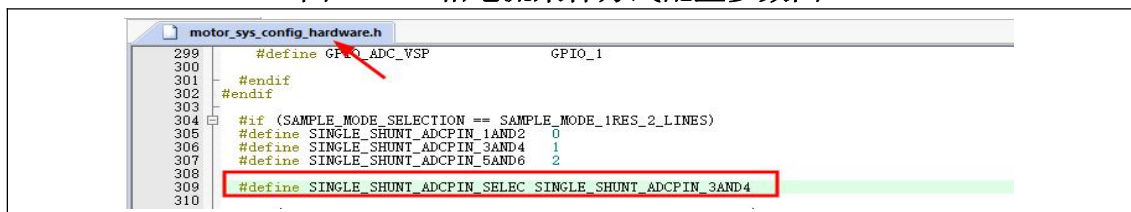
3.1.6 ADC 引脚更换

3.1.6.1 电流采样引脚更换

1) 确认 ADC 引脚对应的 PGA，更改 ADC 引脚配置

电流采样 ADC 的输入引脚已经配置成为可以用宏语言控制，只需要更改相应宏定义，就会更改 PGA 及 ADC 的配置

图 3-8: 相电流采样方式配置参数图



2) 母线电流采样配置

需要根据母线电流输入的引脚进行 PGA 的配置和 ADC 的配置。

图 3-9: 母线电流采样 PGA、ADC 配置图示

```

motor_adc.c
157 ADC->ADCSOCCTL[ADC_SOC_6].all = ADC_SOC_AP
158 ADC_SOC_AN
159 ADC_SOC_BP
160 ADC_SOC_BN
161 ADCSOCCTL_ALL_SHAEN_ENABLE
162 ADCSOCCTL_ALL_SHEEN_ENABLE
163 ((ADC_TRIG10 << ADCSOCCTL_ALL_TRIGSEL_Pos) & ADCSOCCTL_ALL_TRIGSEL_Msk)
164 ((ADC_SAMPLE_TIME_DEFAULT << ADCSOCCTL_ALL_SAMPCNT_Pos) & ADCSOCCTL_ALL_SAMPCNT_Msk)
165 ((ADC_CONVERSION_TIME_DEFAULT << ADCSOCCTL_ALL_CONVNT_Pos) & ADCSOCCTL_ALL_CONVNT_M
166 #endif
167 #endif
168
169 #if (SINGLE_SHUNT_IDC_SENSE_ENABLE == 1)
170 PGA_DiffInit(PGAU, PGAU_CH_P_ADC2, PGAU_CH_N_ADC1, PGA_GAIN_SEL);
171 ADC_EasyInit2(ADC_SOC_IDC, ADCx_PGAOP, ADCx_PGAON, ADC_TRIG0);
172 #endif
173
    
```

3.1.6.2 电压、NTC 采样引脚更换

默认情况下，使用 SPD1148 内部的电压采样，所以无需配置电压采样引脚。

NTC 采样引脚的配置如下：

图 3-10: NTC 采样引脚配置图示

```

motor_sys_config_basic.h motor_parameter.c motor_control.c motor_sys_config_hardware.h
481 #define GPIO_PWM_U_L_SEL GPIO28_PWM5A
482 #define GPIO_PWM_V_L_SEL GPIO26_PWM4A
483 #define GPIO_PWM_W_L_SEL GPIO24_PWM3A
484
485 #define MOTOR_VBUSVOLTAGE_SPD1148_INNER_DETECT 1
486 #define SAMPLE_MODE_SELECTION SAMPLE_MODE_1RES_2_LINES // Which mode selected
487 #define SINGLE_SHUNT_ADCPIN_SELEC SINGLE_SHUNT_ADCPIN_2AND3 // Only Valid when using single shun
488 #define SAMPLEPIN_ADSTOADI3 0 // 0: for SAMPLE_MODE_3RES_6_LINES
489 #define SAMPLE_UV_TWISTED 0
490
491 #define SINGLE_SHUNT_IDC_SENSE_ENABLE 0
492 #define ADC_SOC_IDC ADC_SOC_12
493 #define GPIO_IDC GPIO_1
494 #define GPIO_ADC_NTCPTC GPIO_12
495 #define GPIO_LED GPIO_32
496
497 #if (MOTOR_VBUSVOLTAGE_SPD1148_INNER_DETECT == 1)
498 #define MOTOR_VBUS_VOLTAGE_DETECT_UPPER_RES_OHM 19000.0f //
499 #define MOTOR_VBUS_VOLTAGE_DETECT_LOWER_RES_OHM 1000.0f //
500 #define GPIO_ADC_VDCBUS ADCx_ATEST
501 #else
502 #define MOTOR_VBUS_VOLTAGE_DETECT_UPPER_RES_OHM 100000.0f //
503 #define MOTOR_VBUS_VOLTAGE_DETECT_LOWER_RES_OHM 3000.0f //
504 #define GPIO_ADC_VDCBUS GPIO_1
505 #endif
    
```

3.1.6.3 反电势采样引脚更换

反电势采样电路的引脚如下，但如果要使用反电势采样，记得要提前打开

表 3-5: 反电势采样配置参数表

参数	解释
BEMF_MEASURE_ENABLE	是否启用反电势检测电路
BEMF_MEASURE_UPPER_RES_OHM	反电势采样电路上端电阻
BEMF_MEASURE_LOWER_RES_OHM	反电势采样电路下端电阻
BEMF_MEASURE_CAPACITOR_F	反电势采样电路滤波电容

图 3-11: 反电势采样配置引脚图示

```

motor_sys_config_basic.h  motor_sys_config_hardware.h  motor_adc.c  isr.c  motor_core.c

#define MOTOR_VBUS_VOLTAGE_DETECT_UPPER_RES_OHM 19000.0f //
#define MOTOR_VBUS_VOLTAGE_DETECT_LOWER_RES_OHM 1000.0f //
#define GPIO_ADC_VDCBUS ADCx_ATEST
#else
#define MOTOR_VBUS_VOLTAGE_DETECT_UPPER_RES_OHM 100000.0f //
#define MOTOR_VBUS_VOLTAGE_DETECT_LOWER_RES_OHM 3000.0f //
#define GPIO_ADC_VDCBUS GPIO_1
#endif

#if (BEMF_MEASURE_ENABLE == 1)
#define GPIO_ADC_PHASE_U GPIO_0
#define GPIO_ADC_PHASE_V GPIO_1
#define GPIO_ADC_PHASE_W GPIO_11
#endif
#elif (PARAMETER_SETTING == DC118R0001)

```

3.1.6.4 母线电流采样引脚

表 3-6: 母线电流采样方式配置参数表

参数	解释
IDC_SENSE_SINGLE_END_ENABLE	是否使用单端采样 IDC, 0: 使用差分, 1: 使用单端

3.1.6.4.1 单端采样

如果使用单端采样, 需要在如下文件位置, 配置单端采样的输入引脚, 及 PGA 的放大倍数, 其中放大倍数, 需要和 F32_MOTOR_IDC_PGA_GAIN 一致。

图 3-12: 母线电流单端采样放大倍数配置图示

```

motor_adc.c  isr.c  motor_core.c  motor_core.h  motor_software_config.h  main.c  motor_so

89
90 #if (EXTERNAL_SWITCH_ENABLE == 1)
91 ADC_EasyInit1(ADC_SOC_VSP, GPIO_ADC_VSP, ADC_TRIGO);
92 ADC_SetSampleAndConvertTime(ADC_SOC_VSP, 1000, ADC_DEFAULT_CONVERSION_TIME_NS);
93 #endif
94
95 #if (IDC_SENSE_SINGLE_END_ENABLE == 0)
96 PGA_DifferentialInit(PGA0, PGA0_CH_P_ADC8, PGA0_CH_N_ADC9, PGA_GAIN_SEL);
97 ADC_EasyInit2(ADC_SOC_IDC, ADCx_PGA0P, ADCx_PGA0N, ADC_TRIGO);
98 ADC->ADCSOCCTL[ADC_SOC_IDC].bit.AVGCNT = 3; // 8x avg
99 #else
100 PGA->PGA0CTL.bit.INSELP = PGA0_CH_P_ADC8;
101 PGA->PGA0CTL.bit.GAINP = PGA_SCALE_16X;
102 PGA->PGA0CTL.bit.MODE = PGA0CTL_BIT_MODE_SINGLE_POSITIVE;
103 PGA->PGA0CTL.bit.EN = PGA0CTL_BIT_EN_ENABLE;
104 ADC_EasyInit1(ADC_SOC_IDC, ADCx_PGA0P, ADC_TRIGO);
105 ADC->ADCSOCCTL[ADC_SOC_IDC].bit.AVGCNT = 3; // 8x avg
106 ADC_EasyInit1(ADC_SOC_VDD33, ADCx_VDDA, ADC_TRIGO);
107 ADC->ADCSOCCTL[ADC_SOC_VDD33].bit.AVGCNT = 3;
108 #endif
109

```

表 3-7: 母线电流单端采样参样配置表

参数	解释
IDC_SENSE_SINGLE_END_ENABLE	是否使用单端采样 IDC, 0: 使用差分, 1: 使用单端
F32_MOTOR_IDC_SENSE_HIGH_RES_OHM	单端采样上端电阻, 单位 Ohm
F32_MOTOR_IDC_SENSE_LOW_RES_OHM	单端采样下端电阻, 单位 Ohm
F32_MOTOR_IDC_PGA_GAIN	单端采样 PGA 放大倍数, 注意差分倍数为单端 2 倍
F32_MOTOR_IDC_GAIN_ADJ	单端采样放大调整系数
F32_RES_TEMP_COEF_PER_DEG	单端采样温度调整系数
F32_RES_TEMP_INIT_DEG	单端采样初始温度 (默认 30°)

3.1.6.4.2 差分采样

如果使用差分采样, 则需要配置差分采样的正端和负端, 可以在以下配置中找到

图 3-13: 母线电流差分采样引脚配置图示

```

89
90 #if (EXTERNAL_SWITCH_ENABLE == 1)
91 ADC_EasyInit1(ADC_SOC_VSP , GPIO_ADC_VSP, ADC_TRIG0);
92 ADC_SetSampleAndConvertTime(ADC_SOC_VSP,1000,ADC_DEFAULT_CONVERSION_TIME_NS);
93 #endif
94
95 #if (IDC_SENSE_SINGLE_END_ENABLE == 0)
96 PGA_DifferentialInit(PGA0, PGA0_CH_P_ADC8, PGA0_CH_N_ADC9, PGA_GAIN_SEL);
97 ADC_EasyInit2(ADC_SOC_IDC, ADCx_PGA0P, ADCx_PGA0N, ADC_TRIG0);
98 ADC->ADCSOCCTL[ADC_SOC_IDC].bit.AVGCNT = 3; // 8x avg
99 #else
100 PGA->PGA0CTL.bit.INSELP = PGA0_CH_P_ADC8;
101 PGA->PGA0CTL.bit.GAINP = PGA_SCALE_16X;
102 PGA->PGA0CTL.bit.MODE = PGA0CTL_BIT_MODE_SINGLE_POSITIVE;
103 PGA->PGA0CTL.bit.EN = PGA0CTL_BIT_EN_ENABLE;
104 ADC_EasyInit1(ADC_SOC_IDC, ADCx_PGA0P, ADC_TRIG0);
105 ADC->ADCSOCCTL[ADC_SOC_IDC].bit.AVGCNT = 3; // 8x avg
106 ADC_EasyInit1(ADC_SOC_VDD33, ADCx_VDDA, ADC_TRIG0);
107 ADC->ADCSOCCTL[ADC_SOC_VDD33].bit.AVGCNT = 3;
108 #endif

```

4 串口调试指令

当把 UART_DEBUG_ENABLE 设定为 1 的时候, 会启用串口调试模式, 实现以下特性:

- 1) 可以通过串口输入单字符指令, 进行电机控制, 其中主要指令(区分大小写)如下:

这些指令的相关 code 都写在 motor_debug.c 中。

表 4-1: 串口调试指令表

指令	解释
t	启动电机
s	停止电机
o	按 Gear1 减少指令

p	按 Gear1 增加指令
[按 Gear2 减少指令
]	按 Gear2 增加指令
8	按 Gear3 减少指令
9	按 Gear3 增加指令
q	显示系统状态，如转速、功率、电流等
e	重启系统
f	清除 latch 告警
r	采集数据
R	打印采集数据
c	清除打印数据触发 flag，使得触发模式的打印数据可以再次打印
1	增加数据记录步长
2	减少数据记录步长

其中 GEAR1~3 的设置，用户可以自定义每个 gear 的数值

表 4-2: 串口调试指令档位配置表

定义	解释
CURRENT_GEAR_1	电流指令 1 档
CURRENT_GEAR_2	电流指令 2 档
CURRENT_GEAR_3	电流指令 3 档
SPEED_GEAR_0	速度指令 0 档
SPEED_GEAR_1	速度指令 1 档
SPEED_GEAR_2	速度指令 2 档
SPEED_GEAR_3	速度指令 3 档
POWER_GEAR_1	功率指令 1 档
POWER_GEAR_2	功率指令 2 档
POWER_GEAR_3	功率指令 3 档

用户也可以根据自身需求，增加自定义的串口指令。

2) 可以通过串口输出电机相关状态到上位机

3) 数据采集方法：程序中自带一套数据采集 code，可以把部分变量以主中断的倍数为周期，记录到 RAM 中，然后一次性打印出来。

5 Segger RTT 串口调试

如果在目标系统中，不方便接出串口，可以试用 Jlink 自带的 RttViewer 来模拟出一路串口，这样可以节省出一路串口引脚，缺点是这种连接方式可靠性较低、且断线后需要手动重连。

图 5-1: Segger RTT Viewer 图标



5.1 设置方法

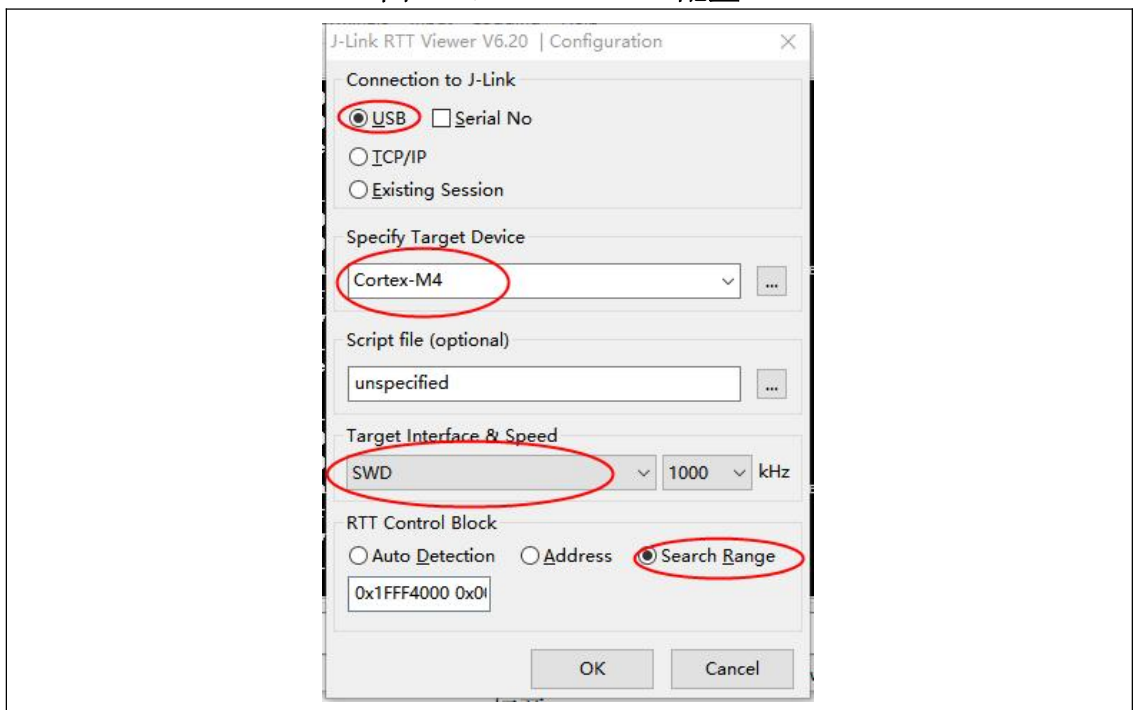
5.1.1 代码配置

旋智提供的吸尘器标准开发包里头，已经包括了 Segger RTT viewer 的配置部分。

5.1.2 工具配置

打开 Segger RTT Viewer，按照如下参数配置，

图 5-2: RTT Viewer 配置

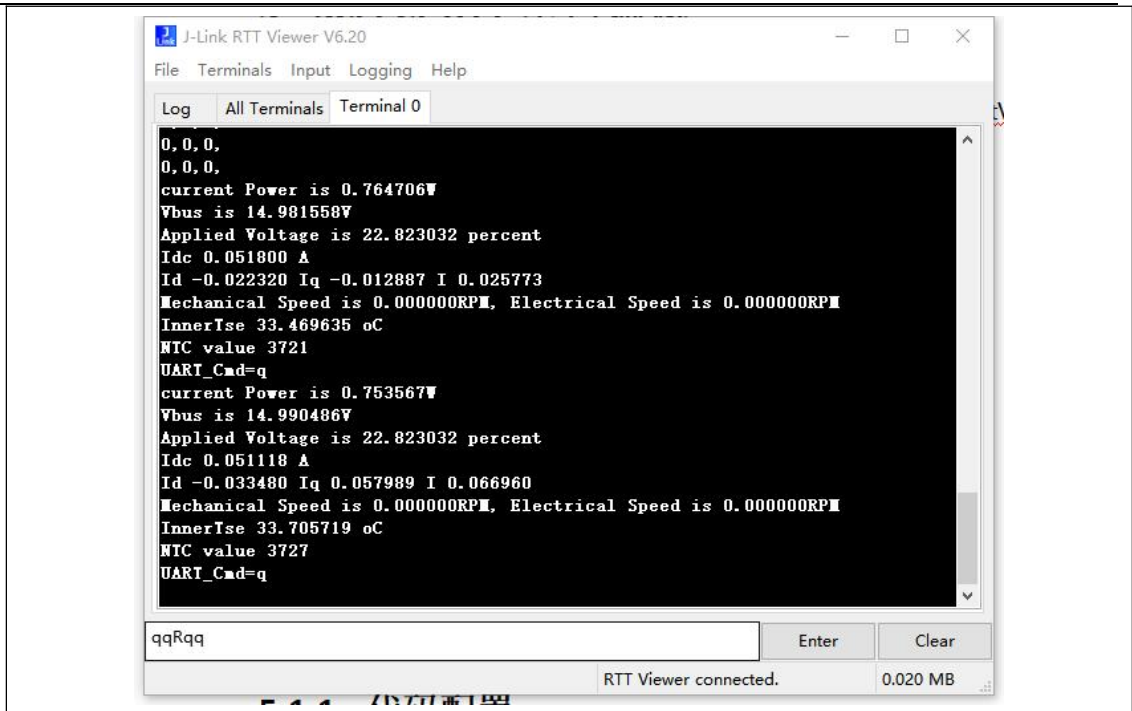


其中 Search Range 部分填写为：0x1FFF4000 0x000100。

5.1.3 使用

连接上之后，就可以按照串口控制的方法进行同样的操作。

图 5-3: RTT Viewer 使用示例图



6 接口函数

系统提供如下接口函数供调用，建议用户只使用下列函数，进行电机控制。

表 7-1: 预定义接口函数表

定义	解释
Motor_Start()	启动电机
Motor_Stop()	停止电机
Motor_GetFaultState()	返回一般错误状态
Motor_GetFatalFaultState()	返回致命错误状态
Motor_GetElectricalSpeedErpm()	获取电机当前电气转速
Motor_GetMechanicalSpeedRpm ()	获取电机当前机械转速
Motor_GetInputPowerW()	获取电机当前运行功率
Motor_GetICTemperature()	获取芯片温度
Motor_SetElectricalSpeedErpm()	设定运行速度
Motor_SetInputPowerW()	设定运行功率

7 附录

7.1 程序框架及中断流程

1) 主中断：即 MAIN_ISR_ROUTINE()，一般和 PWM 同频或者分数关系，在主中断中主要实现以下任务：

- a. 关键信号的采样：相电流、相电压；
- b. 观测器计算
- c. 电流环计算
- d. PWM 发波

2) 1ms 定时中断：TIMER0_IRQHandler()中，主要执行以下任务

- a. 功率环或速度环计算，实现在 Motor_ControlRoutine_1ms()中。
- b. 较慢信号的采样：母线电压、NTC 采样等有较大电容滤波的信号。

3) 异步任务调度：为了用户的控制方便，程序里头自带了以下轮询任务，用户可以把一些可以异步工作的任务放入这些子任务中。

表 8-1: 预定义定时任务表

任务	执行时间
Task_A1(), Task_A2(), Task_A3(),Task_A4()	每个任务 1ms 轮询一次
Task_B1(), Task_B2(), Task_B3(),Task_B4()	每个任务 10ms 轮询一次
Task_C1(), Task_C2(), Task_C3(),Task_C4()	每个任务 1000ms 轮询一次

其中，系统有些自带任务包括

表 8-2: 预定义定时任务及功能表

任务	轮询任务	解释
WorkBench()	Task_A1()	如果使能旋智上位机，该任务负责上位机通

		信
MotorDebugBuffer_DumpData()	Task_A2()	如果使用普通串口通信, 该任务负责打印示波器数据
SlowPrintf()	Task_A2()	如果使用普通串口通信, 该任务负责异步串口信息打印
看门狗喂狗	Task_B1()	负责喂狗, 或者重启系统 (停止喂狗)
Motor_Fault_Print()	Task_C2()	如果使用普通串口通信, 负责打印警告信息

7.2 功率计算及母线电流采样

在吸尘器应用中, 因为使用锂电池供电, 需要较严格的保证产品工作的时长, 所以输入功率的控制是比较重要的功能。

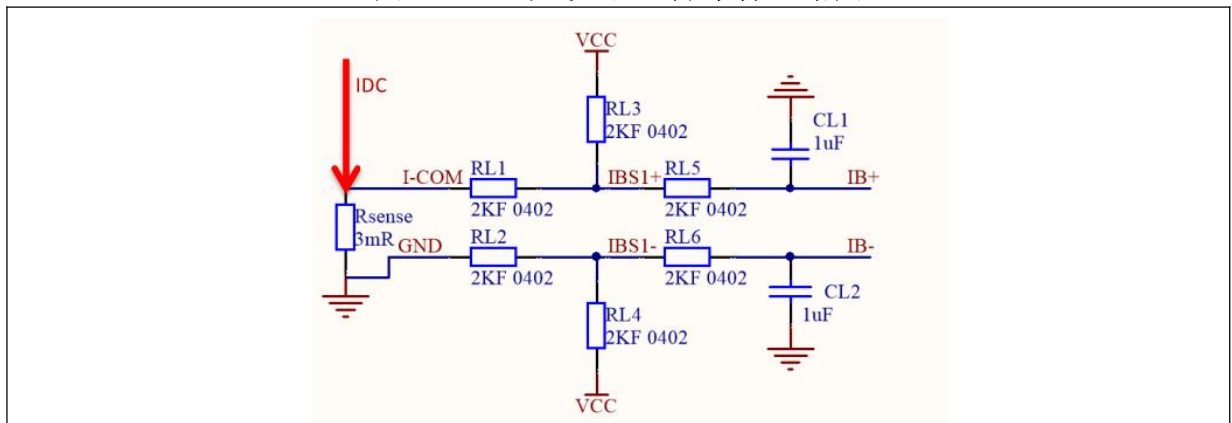
在 SPD1148 吸尘器控制方案中, 采用的是比较直接的 $P = U \times I$ 的方式获取输入功率。

7.2.1 母线电流差分采样

母线电流可以通过电机单电阻采样的上的信号经过低通滤波器后来得到, 电路上可以采用差分或者单端的方式得到该信号。

7.2.2 母线电流差分采样

图 8-1: 母线电流差分采样电路图



如果采用差分电流采样, SPD1148 采样到的母线电流信号为 IB+和 IB-之间的电压信号。
 $(IB+) - (IB-) = [VCC * RL1 / (RL1 + RL3)] - [VCC * RL2 / (RL2 + RL4)] + [(IDC * Rsense) * RL1 / (RL1 + RL3)]$

一般情况下 $[VCC * RL1 / (RL1 + RL3)] - [VCC * RL2 / (RL2 + RL4)]$, 这两项的差值是一个固定值,

则:

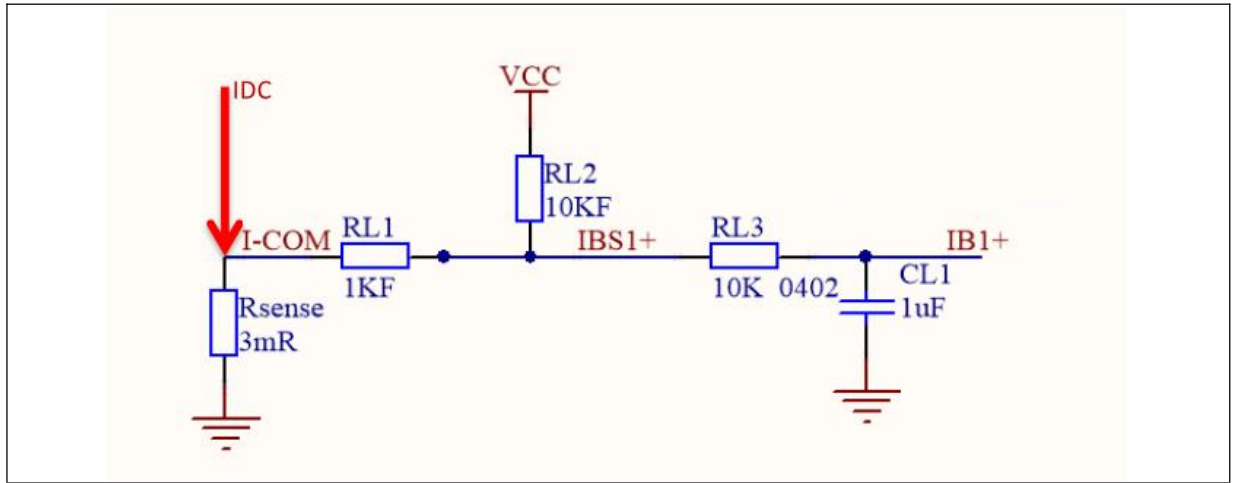
$$(IB+) - (IB-) = [(IDC * Rsense) * RL1 / (RL1 + RL3)],$$

则 IDC 可以恢复为:

$$IDC = [(IB+) - (IB-)] * (RL1 + RL3) / RL1 / Rsense$$

7.2.3 母线电流单端采样

图 8-2: 母线电流单端采样电路图



母线电流单端采样的电路如图，SPD1148 采样到的信号为 IB1+（因为 $R_{sense} \ll RL1$ or $RL2$ ，所以不考虑 R_{sense} 对分压电路的影响）：

$$(IB1+) = [VCC * RL1 / (RL1 + RL2)] + [(IDC * R_{sense}) * RL2 / (RL1 + RL2)]$$

，则可以恢复 IDC 为

$$IDC = \{(IB1+) - [VCC * RL1 / (RL1 + RL2)]\} * (RL1 + RL2) / RL2 / R_{sense}$$

其中这部分 $[VCC * RL1 / (RL1 + RL2)]$ 是一个基本固定的值，一般通过上电后的初始校准来消掉

母线电流的单端采样相比差分采样，可以有效的抑制分压电阻因为温度带来的零点飘动的影响。

但是单端采样会因为：

- 1) VDD33 的飘动；
- 2) Layout 上引入的铜皮会改变等效采样电阻的大小；
- 3) 采样电阻温度特性的改变；

所以需要针对这三方面做特别补偿。

7.2.3.1 VDD33 补偿

如之前所述，采集到的信号中包含跟 $V_{dd}(3.3V)$ 相关的一个信号 $[VCC * RL1 / (RL1 + RL2)]$ ，在差分采样中，因为两路差分信号含有基本相同的部分，所以这部分抵消了， V_{dd} 带来的影响会被抵消掉。但在单端采样中， V_{dd} 的影响则会完全反馈到采集到的信号中 $[VCC * RL1 / (RL1 + RL2)]$ ，所以需要抵消掉 V_{dd} 变化带来的影响。

SPD1148 内部可以采集 V_{dd} 电源上的电压，通过这个信号来补偿 V_{dd} 的影响，具体流程如下：

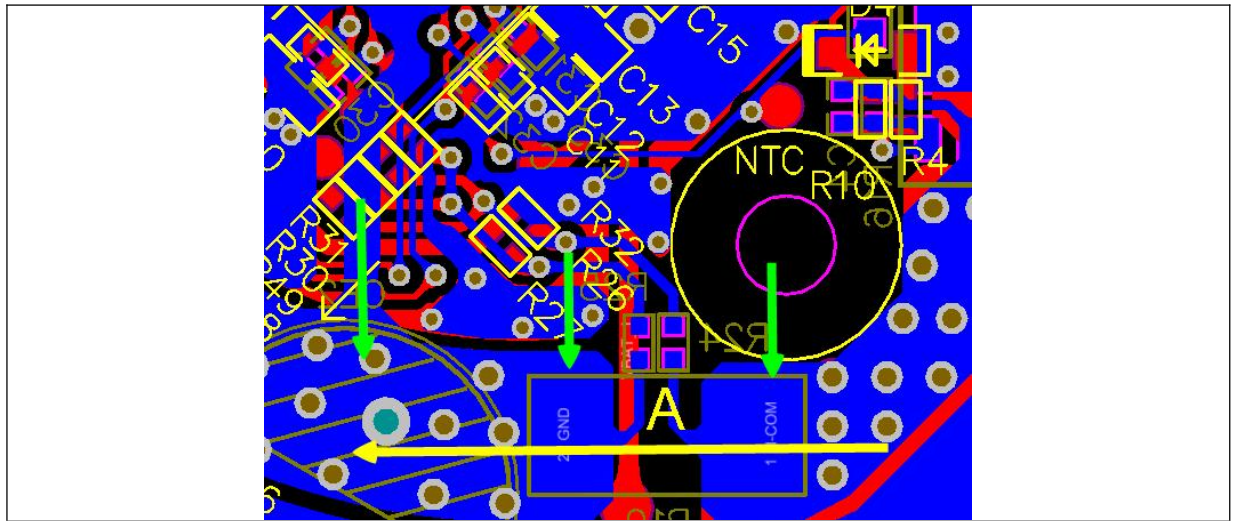
- 1) 上电时采集 V_{dd33} 的值 $V_{dd33init}$ ， $[V_{dd33init} * RL1 / (RL1 + RL2)]$
- 2) 运行过程中不断更新 V_{dd33} 的实时值 $V_{dd33Realttime}$

通过如下公式进行补偿

$$IDC = IDC - [(Vdd33Realtime - Vdd33Init) * RL1 / (RL1 + RL2)]$$

7.2.3.2 等效采样电阻补偿

图 8-3: PCB 上母线采样电阻等效图



如上图的 PCB 所示，2-->3 之间是采样电阻 R_{sense} 的两端，黄色的箭头表示母线电流的通路。采样电阻两端的电压 $V_{rsense} = R_{sense} * I_{dc}$ 。但实际 SPD1148 能够采样到的是 1-->3 之间的电压， $V_{senseReal} = (R_{sense} + R_{delta}) * I_{dc}$ ，其中 R_{delta} 是 1-->2 之间的 PCB 铜皮等效的电阻。

在 demo 代码里里头，这部分使用一个系数 **RatioR** 来做补偿

$$IDC = \{ (IB1+) - [VCC * RL1 / (RL1 + RL2)] \} * (RL1 + RL2) / RL1 / R_{sense} * \mathbf{RatioR}$$

其中 $\mathbf{RatioR} = R_{sense} / (R_{delta} + R_{sense})$ 。

因为 **RatioR** 如果需要直接通过仪器来测量是比较难的，所以这里推荐用户通过微调的方式来调整这个值，在 `motor_sys_config_basic.h` 中，提供了以下参数供用户做调整：

表 8-3: 母线采样电阻补偿系数参数表

7.2.3.3

参数	解释
F32_MOTOR_IDC_GAIN_ADJ	采样电阻补偿系数， RatioR

等效采样电阻温度补偿

采样电阻一般使用锰铜等材料，其温度系数变化较小，但因为在之前的讨论中，PCB 中铜皮的一部分被作为采样电阻来对待了，但是 PCB 本身材料接近纯铜，其温度系数较高，所以对等效的采样电阻的温度系数带来明显的改变，因为锰铜的温度系数远小于铜的温度系数，所以这里可以忽略。

表 8-4: 常用材料电阻率和电阻温度系数表

常用材料电阻率和电阻温度系数		
材料	电阻率 ρ (20℃) $\Omega \cdot m$	平均电阻温度系数 α (0~100℃) 1/℃
银	1.62×10^{-8}	3.5×10^{-3}
铜	1.75×10^{-8}	4.1×10^{-3}
铝	2.85×10^{-8}	4.2×10^{-3}
黄铜 (铜锌合金)	$(2 \sim 6) \times 10^{-8}$	2.0×10^{-3}
铁 (铸铁)	5×10^{-7}	1.0×10^{-3}
钨	5.48×10^{-8}	5.2×10^{-3}
铂	2.66×10^{-8}	2.47×10^{-3}
钢	1.3×10^{-7}	5.77×10^{-3}
汞	4.8×10^{-8}	5.7×10^{-4}
康铜	4.4×10^{-7}	5.0×10^{-6}
锰铜	4.2×10^{-7}	5.0×10^{-6}
镍铬合金	1.08×10^{-6}	1.3×10^{-6}
铁铬铝合金	1.2×10^{-6}	8.0×10^{-5}
炭	1.0×10^{-5}	-5.0×10^{-4}
硬橡胶	1×10^{16}	

一般情况下，这个等效的温度系数 **TempCoef** 会等于，

$$\text{TempCoef} = 4.1e-3 * (1.0f - \text{RatioR})$$

但这里，也推荐通过微调的方式来得到比较准确的值。

在 `motor_sys_config_basic.h` 中，提供了以下参数供用户做调整：

表 8-5: 采样电阻温度补偿系数参数表

参数	解释
F32_RES_TEMP_COEF_PER_DEG	采样电阻补偿系数，TempCoef

7.3 启动逻辑

对于吸尘器应用，因为转向一定，只存在静止启动和顺风启动两种情况。

7.3.1 顺风逻辑

顺风启动，首先要判断是否在顺风状态，一般来说，如果电机在旋转过程中，其三相线端会有较明显的反电势产生，这个反电势的大是转速和反电势系数的乘积 ($Bemf = f * \text{FluxCoef}$)。所以判断电机是否在旋转，就可以根据这个反电势的大小来确定。

同时根据电路上是否有反电势采样电路，又可以分为两种不同的方法。

1) 有反电势采样电路

如果有反电势采样电路，则可以直接采样到反电势电压的大小，同时观测器本身也可以通过采集到的反电势进行闭环控制。所以只要在不打开实际 PWM 发波的情况下，采集反电势，同时让观测器本身工作一小段时间，观测器就可以稳定。这样通过反电势的大小和观测器观察到的电机速度，就可以可靠的判断电机运行状态，进行顺风或者静止启动。

在 motor_sys_config.h 中，提供了以下参数供用户做调整：

表 8-6： 顺风启动检查参数表

参数	解释
STATUS_CHECK_CONTINUE_THRES_BEMF_V	判断电机是否旋转的反电势阈值，单位 V
STATUS_CHECK_TIME_MS	状态判断的时长，单位 ms

$$E_{amp} = \sqrt{E_{\alpha}^2 + E_{\beta}^2} > \text{STATUS_CHECK_CONTINUE_THRES_BEMF_V}$$

如果反电势的有效值大于这个阈值，则认为电机处于旋转的状态。

STATUS_CHECK_CONTINUE_THRES_BEMF_V：这个值如果过大，这样可能会出现很多比较高速的情况，仍旧进入了静止启动，造成电机振动；如果过小，则会在很低速时候，仍旧进入顺风启动，容易造成失步，一般情况下，这个阈值要接近开环最大转速时候的反电势峰值。

2) 没有反电势采样电路

如果没有反电势采样电路，则需要通过打开一段时间 PWM，通过反电势在电机线圈中产生有效电流，同时观测器工作一小段时间。如果有足够大的电流，则观测器可以稳定的工作。最后可以根据电流的大小和观测器观察到的速度，判断电机的运行状态，进行顺风或者禁止启动。

在 motor_sys_config.h 中，提供了以下参数供用户做调整：

表 8-7： 顺风启动判断阈值表

参数	解释
STATUS_CHECK_CONTINUE_THRES_A	判断电机是否旋转的电流阈值，单位 A
STATUS_CHECK_TIME_MS	状态判断的时长，单位 ms

STATUS_CHECK_CONTINUE_THRES_A：这个值如果过大，这样可能会出现很多比较高速的情况，仍旧进入了静止启动，造成电机振动；如果过小，则会在很低速时候，仍旧进入顺风启动，容易造成失步。

目前主要的调试方法是不断微调，刚开始可以设定为额定电流，后续不断降低这个阈值，直到找到一个比较满意的状态。

7.3.2 静止逻辑

如果在之前的顺风状态判断中，认为电机并没有高速运行，则进入静止启动，采用开环转闭环的方式。

具体开环切入闭环逻辑可参考下图。

1) 开环定位（电流定位）：

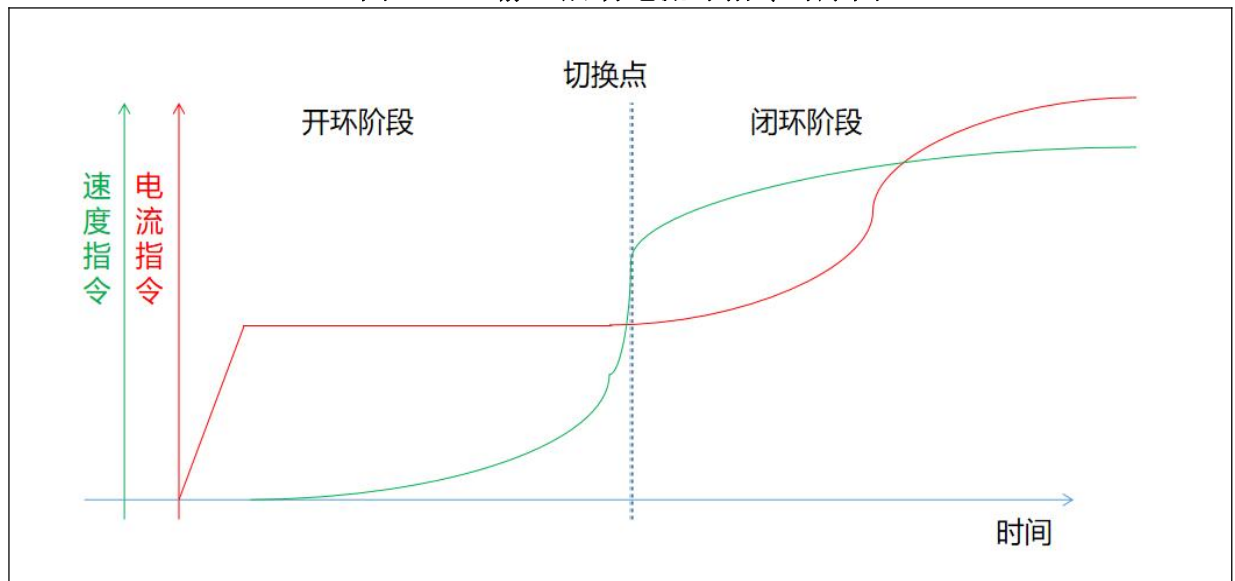
在 motor_control.c 中，如果开环定位时间 $\text{PMSM_FOC_P.idc_pos_time} > 0$ ，则启动时会按照电角度 $\text{PMSM_FOC_P.idc_pos_angle_Pu}$ ，在直轴上施加 $\text{PMSM_FOC_P.idc_pos_refA}$ 进行定位；

2) 开环阶段（电流开环）：

电流指令，较快的从 0 的达到设定的开环最大电流 **MOTOR_OPENLOOP_IMAX_A**

速度指令，从 0 线性增加到 **MOTOR_OPENLOOP_LEAVE_RAD**，其时长为 $\text{PMSM_FOC_P.opnlp_acc_time}$ ；

图 8-4: 静止启动电流环指令时序图



3) 闭环阶段

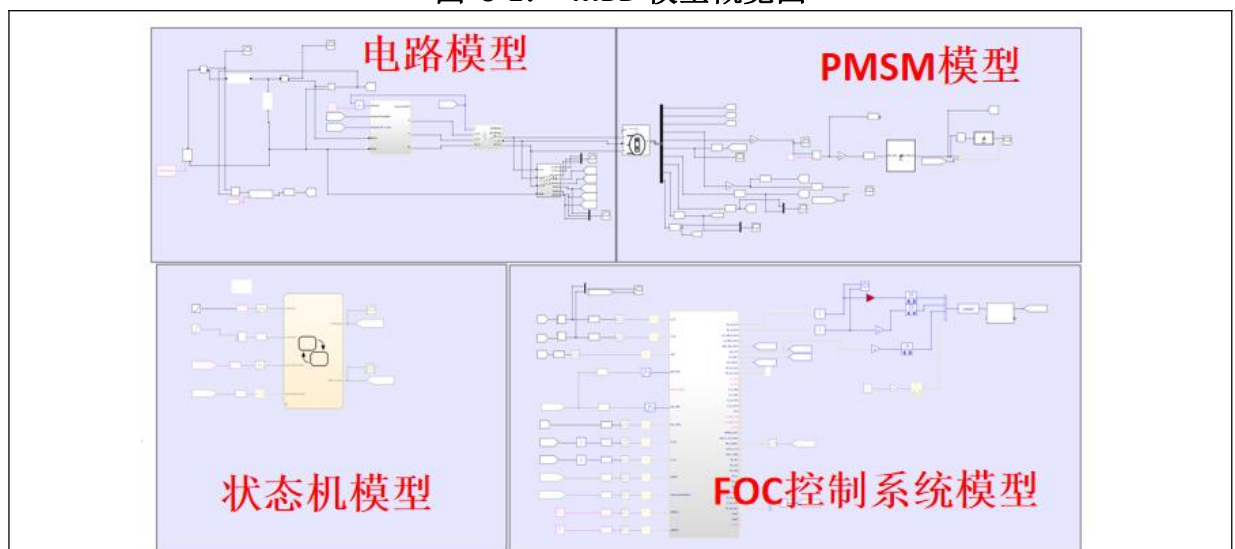
当达到开环最大转速后，系统根据环路设定（速度闭环、功率闭环、电流闭环等），分别进入相应闭环控制。

8 Model Base Design 介绍

使用的模型为“LVModel.slx”

8.1 总体框架

图 9-1: MBD 模型概览图

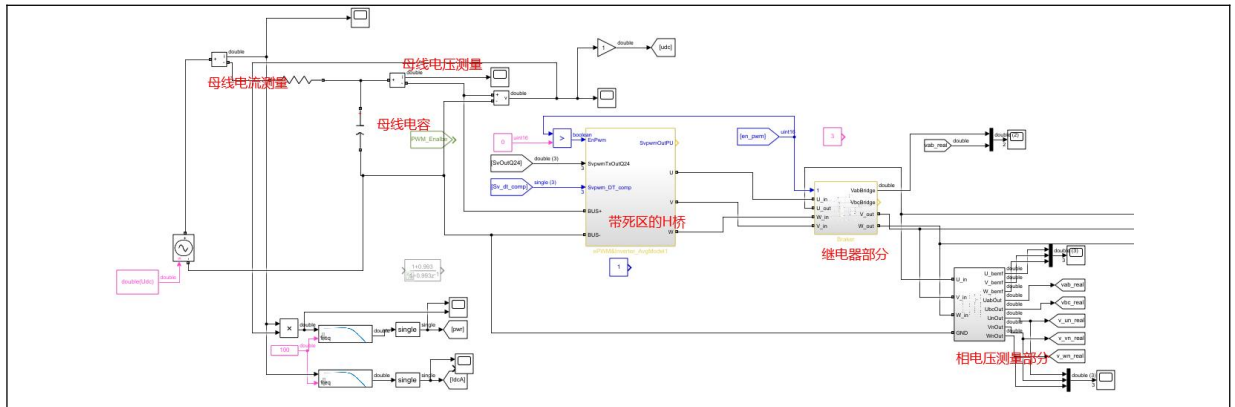


8.2 各个模块介绍

8.2.1 电路模型

电路模型如下：

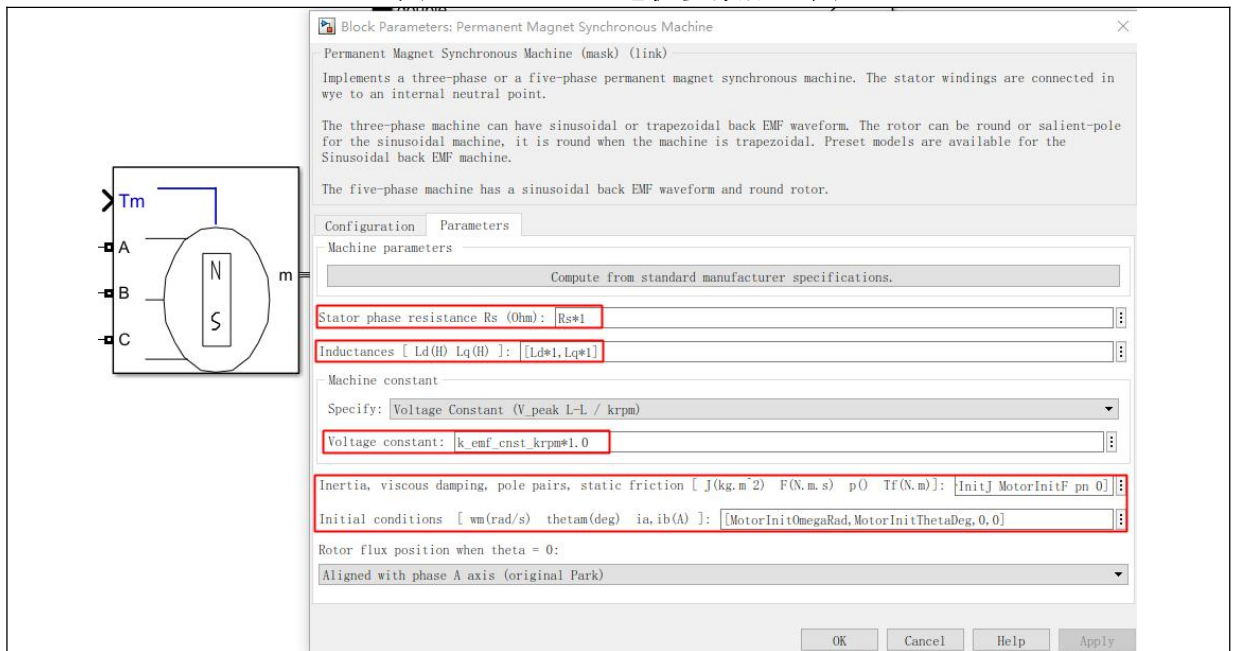
图 9-2: MBD 模型电路图



8.2.2 PMSM 模型

电机模型部分使用了 Matlab 自带的 PMSM 模型，用户可以按照实际电机的特点，填入电机参数、转动惯量 J 、摩擦系数及初始角度速度等信息。

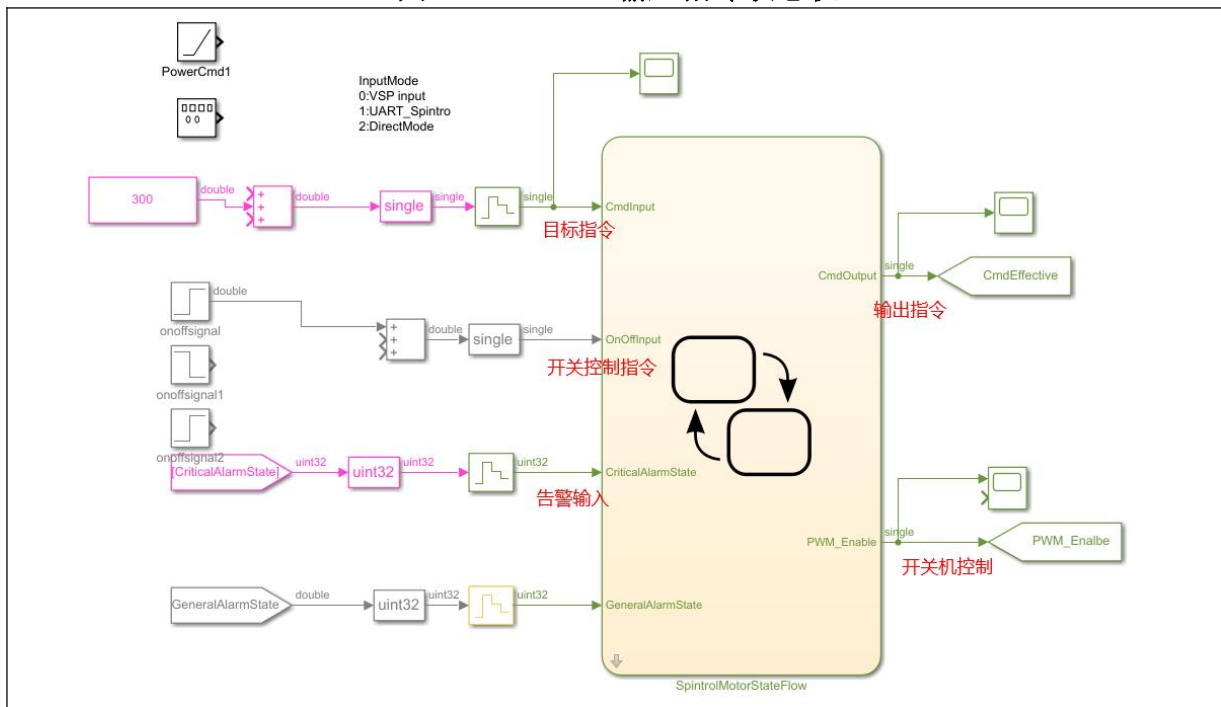
图 9-3: MBD 电机参数配置图



8.2.3 状态机模型

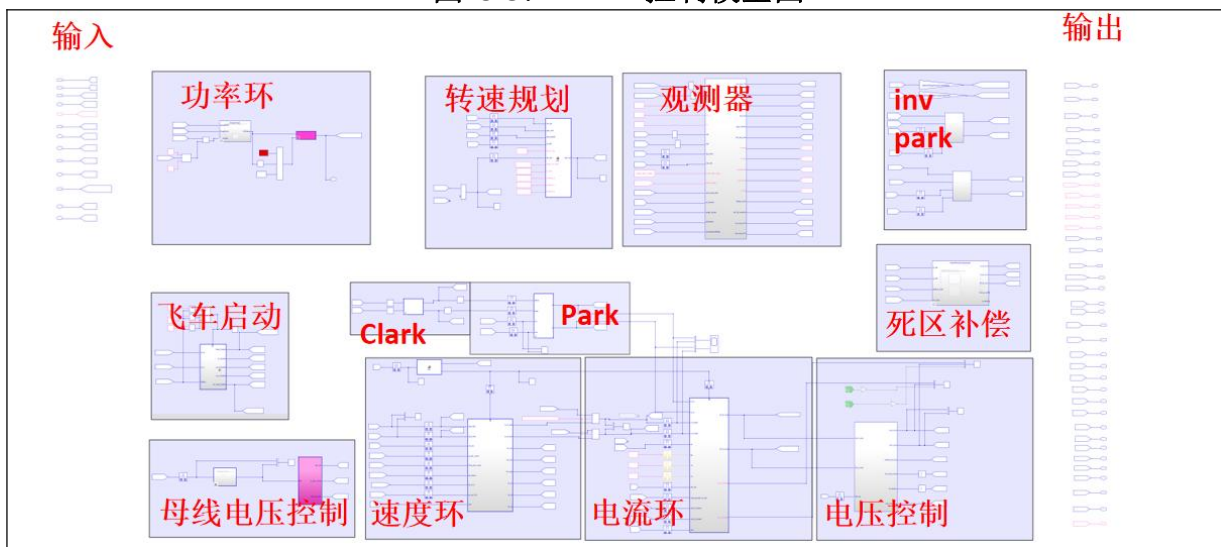
状态机主要设计用来做指令限幅以及开关机控制。

图 9-4: MBD 输入指令状态机



8.2.4 FOC 控制系统模型

图 9-5: MBD 控制模型图



8.2.5 参数设置

一些常用的参数，已经收集到和模型统一目录下的 psms_config.m 中。

其中包括：

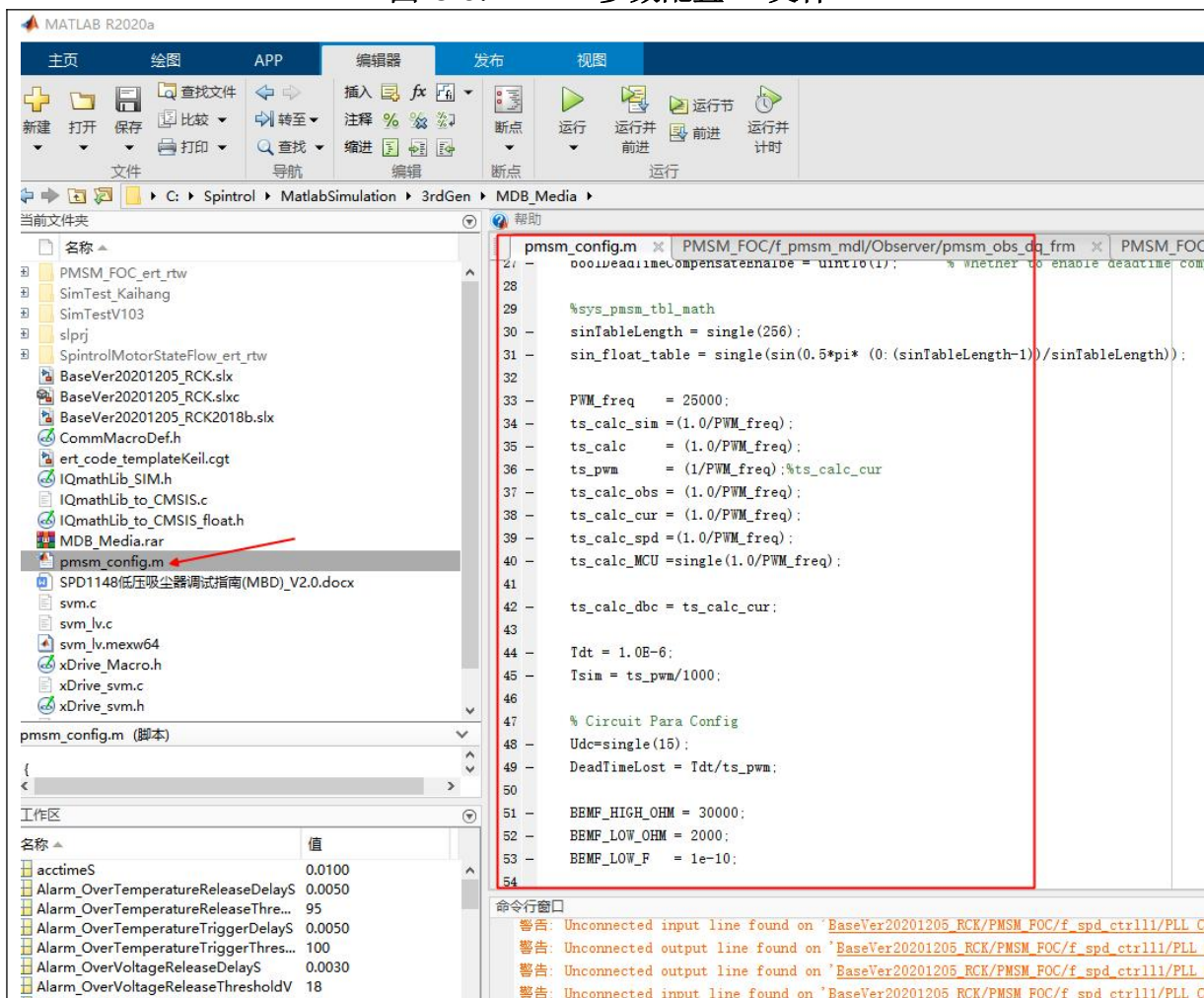
电路参数，如 PWM 频率、死区时间、母线电压等；

电机参数：电阻、电感、惯量、初速度等；

控制参数：观测器设置、电流环、转速环设置、弱磁控制等；

其他相关参数。

图 9-6: MBD 参数配置 M 文件

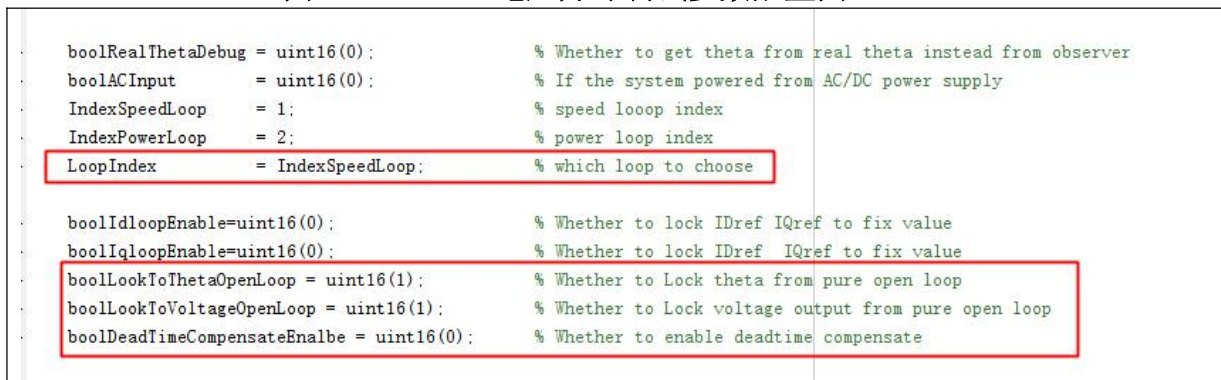


8.3 调试

8.3.1 电压开环调试

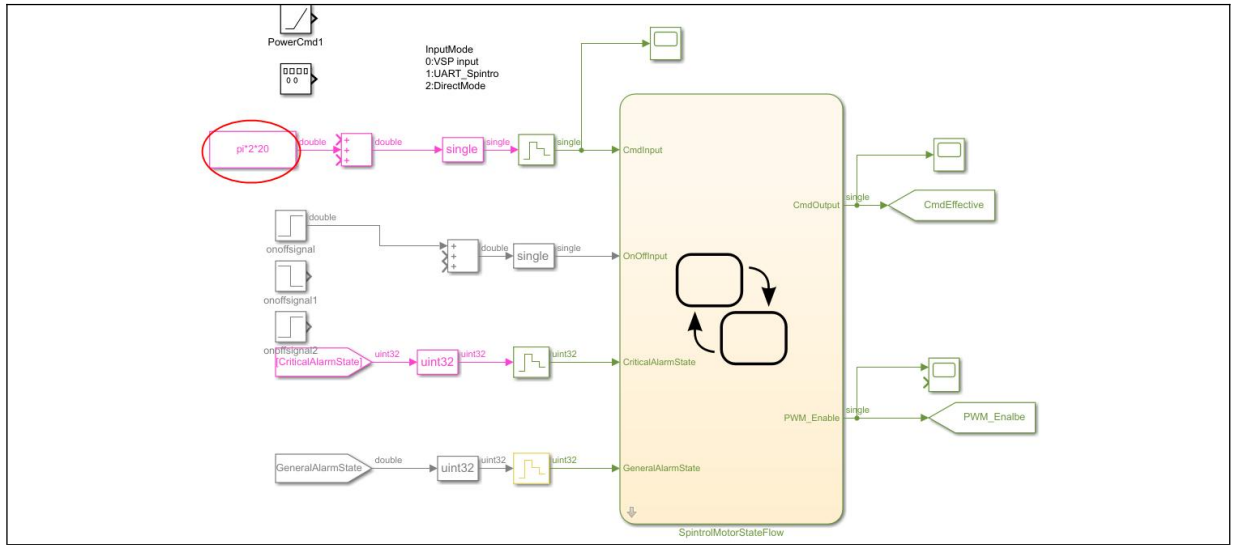
如下设置，可以让系统以不带死区控制的电压开环运行，

图 9-7: MBD 电压开环调试参数配置图



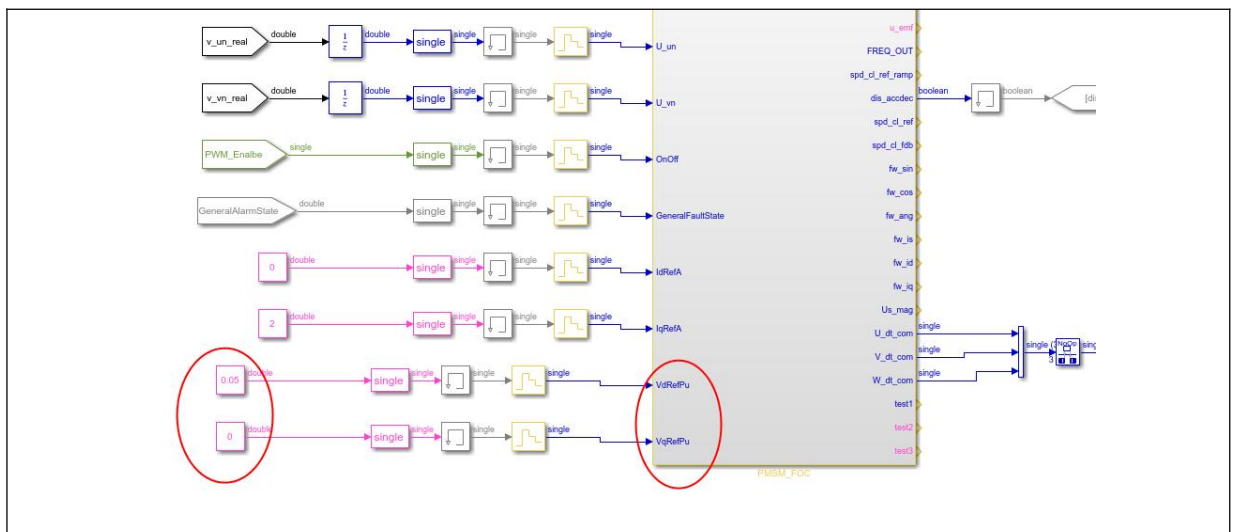
速度指令由如下方式输入：

图 9-8：速度指令输入图



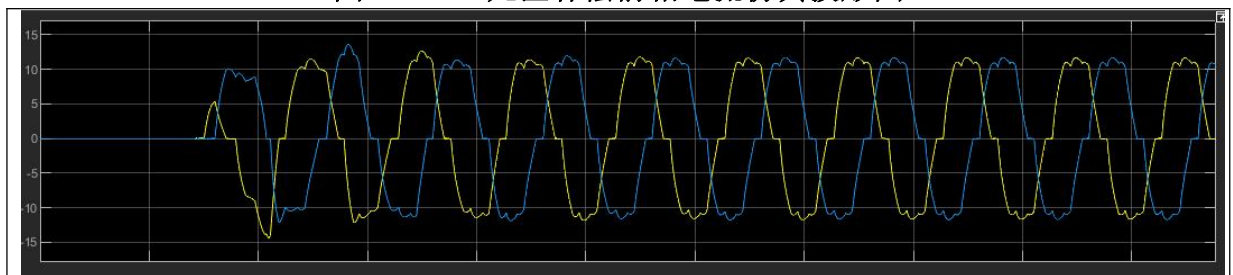
电压指令可以由如下方式输入：

图 9-9：电压指令输入图



运行系统可以看到相电流波形如下（不带死区）：

图 9-10：死区补偿前相电流仿真波形图

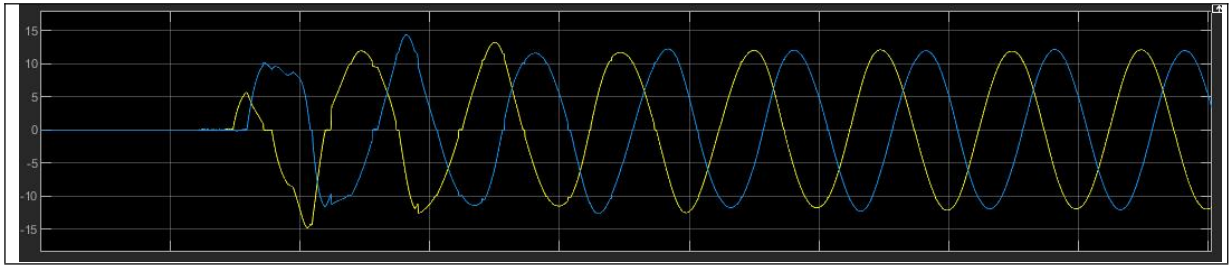


此后打开死区补偿：

```
boolDeadTimeCompensateEnalbe = uint16(1);
```

此时相电流波形如下，由此可以验证死区补偿的有效性：

图 9-11： 死区补偿后相电流仿真波形图



8.3.2 速度/功率闭环调试

关闭角度开环，关闭角度开环，选择速度闭环或者功率闭环：

仿真过程中，可以通过如下示波器观察观测器的收敛过程和最终角度差别。

图 9-12： 配置速度环电机运行参数图

```

boolRealThetaDebug = uint16(1);           % Whether to get theta from real theta instead from observer
boolACInput         = uint16(0);           % If the system powered from AC/DC power supply
IndexSpeedLoop     = 1;                    % speed loop index
IndexPowerLoop     = 2;                    % power loop index
LoopIndex          = IndexSpeedLoop;      % which loop to choose

boolIdloopEnable   = uint16(0);           % Whether to lock Idref IQref to fix value
boolIqloopEnable   = uint16(0);           % Whether to lock Idref IQref to fix value
boolLookToThetaOpenLoop = uint16(0);     % Whether to Lock theta from pure open loop
boolLookToVoltageOpenLoop = uint16(0);   % Whether to Lock voltage output from pure open loop
boolDeadTimeCompensateEnable = uint16(1); % Whether to enable deadtime compensate
    
```

图 9-13： MBD 查看实际角度和模型角度偏差 I

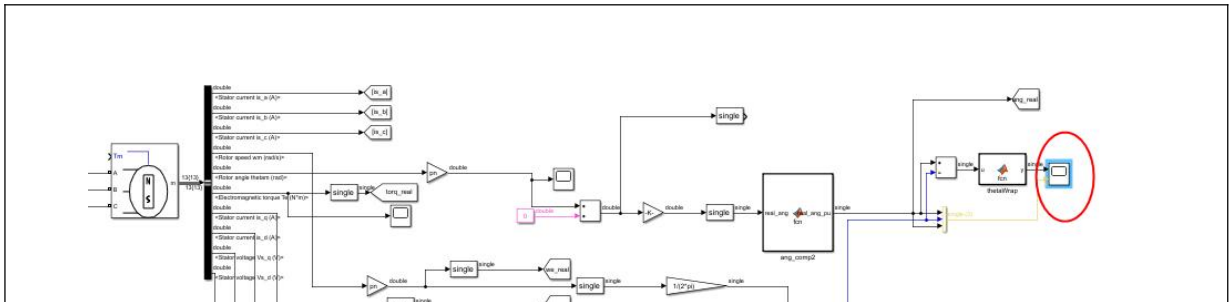
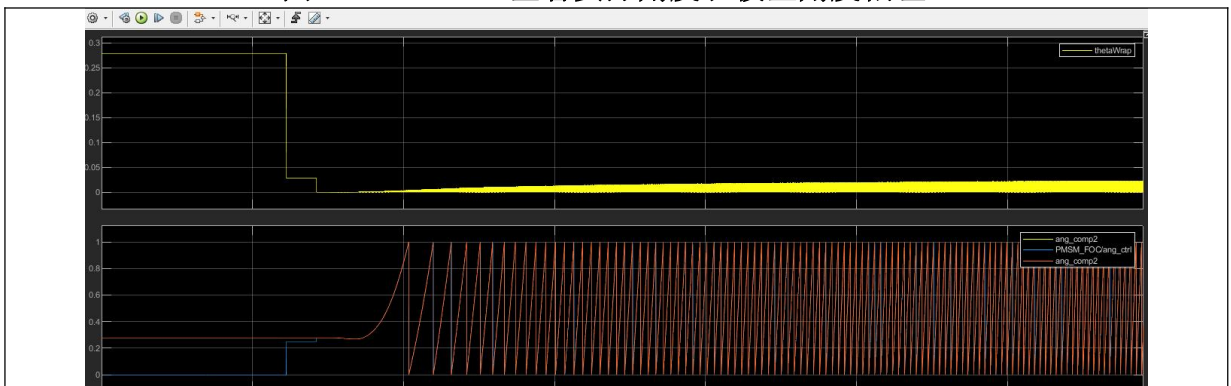


图 9-14： MBD 查看实际角度和模型角度偏差 II

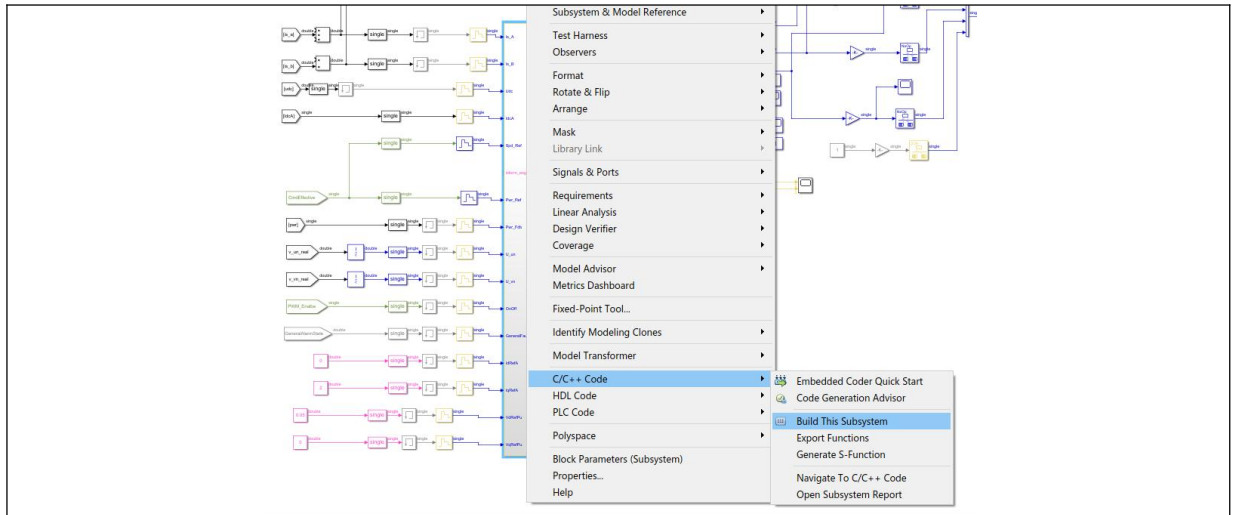


8.4 代码生成

对于已经验证好的模块，可以按照下述步骤

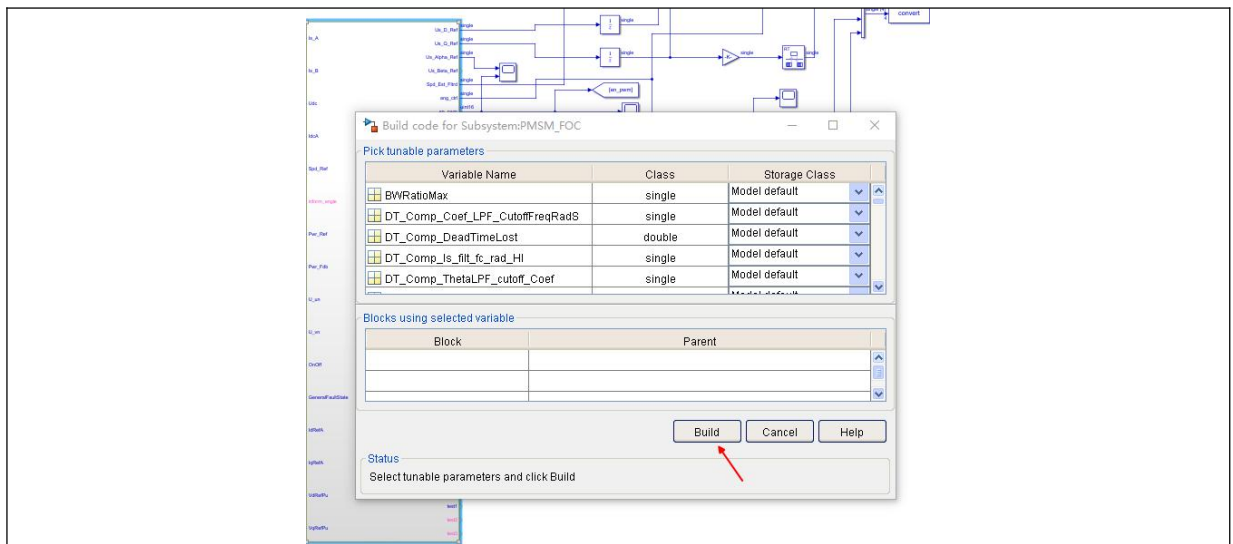
- 1) 模块上按右键，找到 C/C++ Code --> Build This Subsystem

图 9-15: MBD 生成代码操作 I



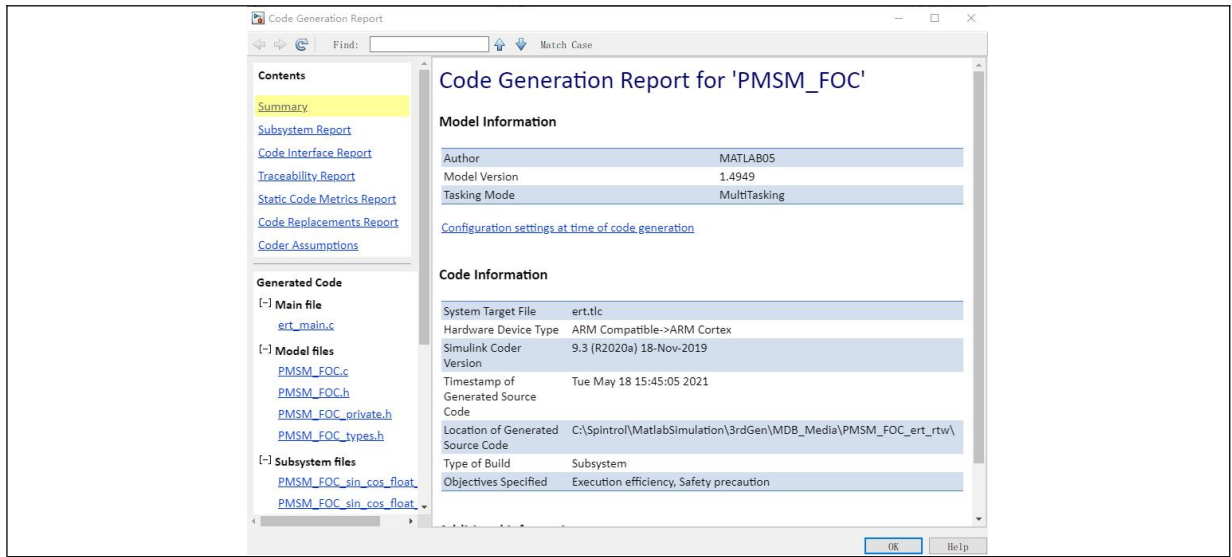
- 2) 点击“Build”

图 9-16: MBD 生成代码操作 II



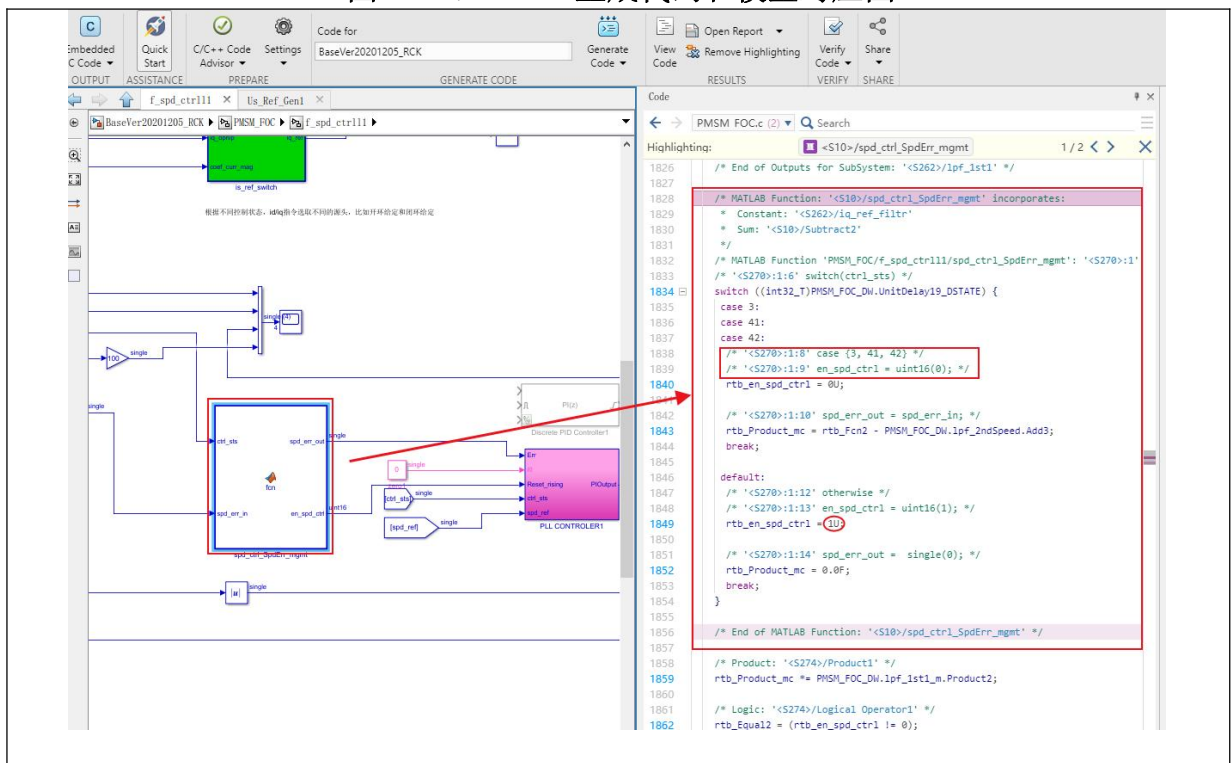
3) 结束后，将生成代码生成报告

图 9-17: MBD 生成代码报告图



FOC 控制部分的代码会自动生成在根目录的\PMSM_FOC_ert_rtw 中，如果选择生成状态机的代码，则会生成在\SpintronicMotorStateFlow_ert_rtw 中，点击相应子模块还可以找到对应代码

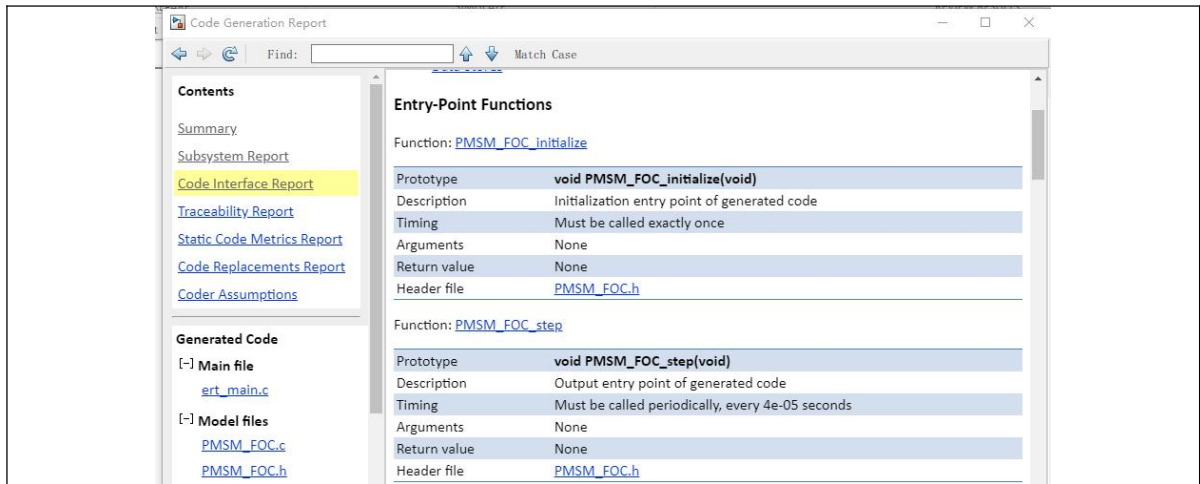
图 9-18: MBD 生成代码和模型对应图



代码报告的 Code Interface 中包含如何调用 MBD 生成代码的信息

a. 接口函数

图 9-19: MBD 生成代码接口函数图



b. 接口变量

图 9-20: MBD 生成代码接口变量表

