AN0064
应用笔记

AT32 电机库使用指南

前言

这份文件描述了如何使用 AT32 电机函数库，文中分别针对环境建立、电机库架构、头文件设定内容、个别函数、以及实际应用范例的程序结构等一一说明。

支持型号列表：

支持型号	AT32F 系列 AT32L 系列

目录

1 电机库算法概述 6

2 环境准备 8

2.1 硬件环境准备 8

2.2 软件环境准备 8

3 电机库文档说明 10

4 电机库函数使用说明 29

4.1 通用电机库函数 29

4.2 矢量控制电机库函数 32

4.3 六步方波控制电机库函数 41

5 电机库应用范例程序结构 43

5.1 状态机 43

5.1.1 状态描述 43

5.1.2 状态机流程说明 44

6 版本历史 45

表目录

表 1. 对应闪存存储空间之 ROM 配置表 8
表 2. 电机库文档总表 11
表 3. 模式宏定义 13
表 4. 控制参数宏定义 14
表 5. 驱动器参数宏定义 19
表 6. 电机参数宏定义 20
表 7. 周边配置相关函数 24
表 8. 电机控制相关中断函数 26
表 9. 电机库枚举/类型列表 26
表 10. 函数 get_fw_id 29
表 11. 函数 mc_param_init 29
表 12. 函数 pid_controller 30
表 13. 函数 current_auto_tuning 30
表 14. 函数 atan2_fixed 30
表 15. 函数 sqrt_fixed 31
表 16. 函数 param_identify 31
表 17. 函数 param_id_process 31
表 18. 函数 current_read_foc_1shunt 32
表 19. 函数 rds_auto_calibration 32
表 20. 函数 enc_speed_get_MTmethod 33
表 21. 函数 position_cmd_ramp 33
表 22. 函数 hall_rotor_angle_get 34
表 23. 函数 hall_delta_theta_calculation 34
表 24. 函数 svpwm_3shunt 34
表 25. 函数 svpwm_2shunt 35
表 26. 函数 svpwm_1shunt 35
表 27. 函数 pwm_shift 36
表 28. 函数 foc_circle_limitation 36
表 29. 函数 foc_vq_limitation 36
表 30. 函数 startup_openloop 37

表 31. 函数 startup_alpha_axis 37
表 32. 函数 flag_status startup_angle_init 38
表 33. 函数 flag_status startup_angle_init2 38
表 34. 函数 foc_sensorless_angle_init 38
表 35. 函数 current_angle_init_3shunt 39
表 36. 函数 current_angle_init_2_1shunt 39
表 37. 函数 obs_pll_execute 40
表 38. 函数 rotor_angle_sensorless 40
表 39. 函数 calc_adc_sample_point 41
表 40. 函数 bldc_sensorless_angle_init 41
表 41. 函数 angle_init_estimation 42
表 42. 函数 change_phase_period_ramp 42
表 43. 文档版本历史 45

图目录

图 1. AT32F413RCT7 之 ROM 配置(AT32IDE) 9
图 2. 电机库控制程序架构图 10
图 3. 电机库文档结构说明图 10
图 4. 编码器 ABZ 信号的关系图 21
图 5. PMSM 反电势、霍尔状态与电气角的关系图(HALL_LEARN_DIR=0) 22
图 6. BLDC 反电势、霍尔状态与 MOS 导通状态的关系图 23
图 7. 状态机流程图 44

1 电机库算法概述

电机库相关算法主要内容如下

目标电机：三相永磁同步电动机（直流无刷电动机）

控制模式：

- FOC 矢量控制
- 120°方波控制

三相 PWM 调制模式：

- SVPWM
- 120°导通 PWM 控制

相电流检测模式：

- 三电阻电流检测
- 双电阻电流检测
- 单电阻电流检测和重构方式

转子位置检测模式：

- 霍尔效应位置传感器
- 光电增量编码器

初始转子位置估测模式：

- 三相电压矢量转子初始角度估测
- 两相电压矢量转子初始角度估测

FOC 弦波驱动的转子位置估测模式：

- 龙伯格估测器(Luenberger observer)反电势估测

120°方波控制的转子位置估测模式：

- 比较器回授反电势零交越点信号
- ADC 检测反电势零交越点

有传感器 FOC 弦波控制模式：

- 电压矢量控制
- 转矩控制 (电流矢量控制)

- 转速控制
- 弱磁控制
- 定位控制
- 回生刹车

无传感器弦波控制模式：

- 开环启动
- 转矩控制 (电流矢量控制)
- 转速控制
- 弱磁控制
- 回生刹车

有传感器 120°方波控制模式：

- 120°方波电压控制
- 转矩控制 (120°方波电流控制)
- 转速控制

无传感器 120°方波控制模式：

- 120°方波电压控制
- 开环启动
- 转矩控制 (120°方波电流控制)
- 转速控制

自动调试功能：

- 霍尔状态自学习(霍尔传感器专用)
- 电机线圈参数自动辨识
- 电流 PI 控制参数自整定

2 环境准备

2.1 硬件环境准备

需要准备硬件项目主要包括 PMSM(BLDC)电机、AT-Link 或第三方调试下载器以及一块电机控制板，若控制板使用 AT-MOTOR-EVB 电机发展板，则相关硬件配置可参考低压电机控制开发板使用手册。

- PMSM(BLDC)电机
- 调试下载器
- 电机控制板

2.2 软件环境准备

1. 用户可以先参照各个 AT32 MCU 之入门使用指南搭建 AT32 开发环境，以及芯片本身的功能配置。
2. 当用户依照使用指南建立一个新的工程后，即可将本电机库之相关头文件与函数库加入新建工程中，并于相关头文件分别输入电机控制型式、电机参数、控制板参数、控制器参数，以及硬件外设规划参数。
3. AT32IDE 的配置需根据各个 AT32 MCU 的闪存存储大小修改 Id 文件，详细参照表 1，例：AT32F413RCT7 的闪存存储大小为 256 K 字节，则其 IROM1 的起始位置为 0x8000000，大小为 0x3F800，其 IROM2 的起始位置为 0x803F800，大小为 0x320，IROM3 的起始位置为 0x803FB20，大小为 0x4E0, AT32F413RCT7 的修改范例如图 1 所示。
4. 用户可根据其应用需求自行撰写控制程序，并在程序中调用电机库函数，经由已优化的电机库函数，快速完成电机控制工程项目。

表 1. 对应闪存存储空间之 ROM 配置表

Flash size	1024K	512K	256K
IROM1(adress)	0x8000000	0x8000000	0x8000000
IROM1(size)	0xFF800	0x7F800	0x3F800
IROM2(adress)	0x80FF800	0x807F800	0x803F800
IROM2(size)	0x320	0x320	0x320
IROM3(adress)	0x80FFB20	0x807FB20	0x803FB20
IROM3(size)	0x4E0	0x4E0	0x4E0

Flash size	128K	64K	32K	16K
IROM1(adress)	0x8000000	0x8000000	0x8000000	0x8000000
IROM1(size)	0x1FC00	0x0FC00	0x07C00	0x03C00
IROM2(adress)	0x801FC00	0x800FC00	0x8007C00	0x8003C00
IROM2(size)	0x320	0x320	0x320	0x320
IROM3(adress)	0x801FF20	0x800FF20	0x8007F20	0x8003F20
IROM3(size)	0xE0	0xE0	0xE0	0xE0

注[1]:若使用 keil v5.33 版本，因 AT32 BSP 源码不支持 V6.15 编译器，请使用 keil complier version 5 版本进行编译。

图 1. AT32F413RCT7 之 ROM 配置(AT32IDE)

AT32F413xC_FLASH.ld

```
/* Specify the memory areas */
MEMORY
{
FLASH (rx)      : ORIGIN = 0x08000000, LENGTH = 0x3F800
MC_DATA1 (r)    : ORIGIN = 0x0803F800, LENGTH = 0x320
MC_DATA2 (r)    : ORIGIN = 0x0803FB20, LENGTH = 0x4E0
RAM (xrw)       : ORIGIN = 0x20000000, LENGTH = 32K
}

/* Define output sections */
SECTIONS
{
  /* The startup code goes first into FLASH */
  .isr_vector :
  {
    . = ALIGN(4);
    KEEP(*(.isr_vector)) /* Startup code */
    . = ALIGN(4);
  } >FLASH

  .mc_data1 :
  {
    . = ALIGN(4);
    . = ORIGIN(MC_DATA1);
    _MC_VectStoreAddr1 = .;
    *(.MC_VectStoreAddr1);
    . = ALIGN(4);
  } >MC_DATA1

  .mc_data2 :
  {
    . = ALIGN(4);
    . = ORIGIN(MC_DATA2);
    _MC_VectStoreAddr2 = .;
    *(.MC_VectStoreAddr2);
    . = ALIGN(4);
  } >MC_DATA2

  /* The program code and other data goes into FLASH */
  .text :
  {
    . = ALIGN(4);
    *(.text)           /* .text sections (code) */
    *(.text*)          /* .text* sections (code) */
    *(.glue_7)         /* glue arm to thumb code */
    *(.glue_7t)        /* glue thumb to arm code */
    *(.eh_frame)
```

3 电机库文档说明

图 2 为一个电机控制工程中电机控制函数与其它 MCU 基础函数(BSP)、UI 通讯程序以及硬件的关系图。其中最低层的硬件外设直接透过 bsp 函数控制，而电机库函数与 UI 函数均建构于 BSP 与 Low level 函数之上，而用户自行撰写的控制程序则植基于电机库函数与 UI 函数之上。因此用户可以很方便地调用电机控制函数控制 MCU 硬件外设，实现电机控制程序。并可同时经由 UI 控制函数与外部个人计算机 UI 软件工具链接，传输实时的电机控制状态或实时改变控制参数与命令。

图 2. 电机库控制程序架构图

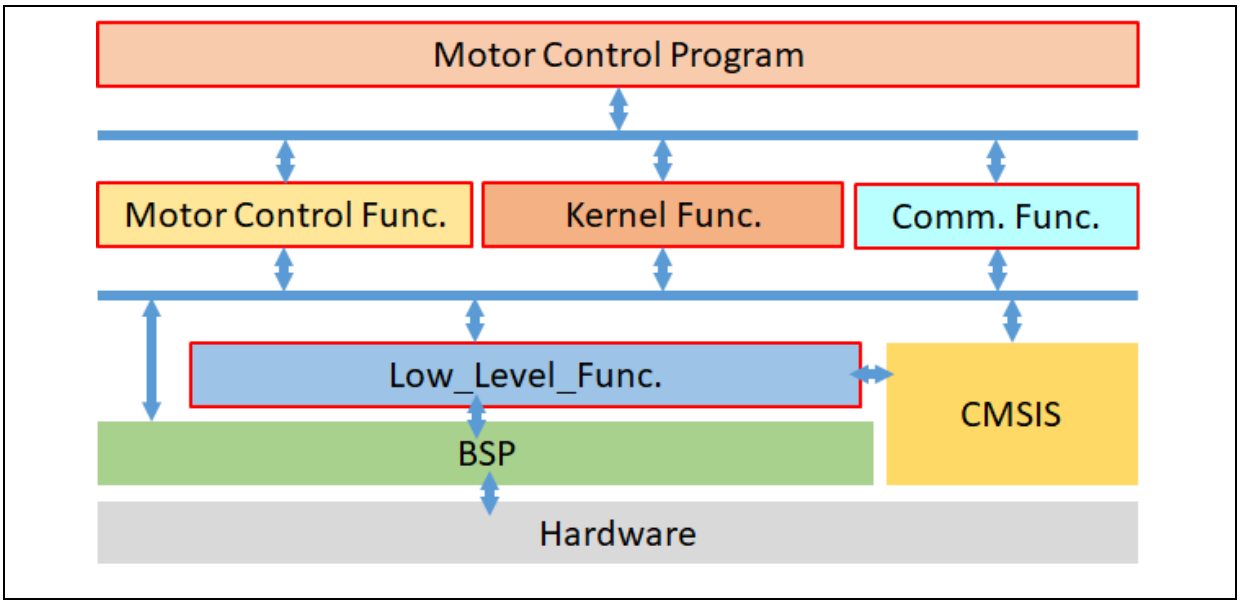


图 3 为电机库文档结构说明图，说明电机库文档中各文件之间的关系，其中 motor_control_drive_param.h 头文件提供用户自行输入电机控制型式、电机参数、控制板参数、控制器参数等，以及 mc_hwio.h 头文件可根据 MCU 外设与控制板连结的接脚对应关系，设定 MCU 外设规划参数。相关的设定参数于 mc_foc_globals.h (mc_bldc_globals.h) 整合后，于 mc_foc_globals.c(mc_bldc_globals.c)中的函数设定变量初值，提供电机库函数使用。而在 MCU 外设规划部分，则由 mc_hwio.c 文件执行相关外设初使化设定。

图 3. 电机库文档结构说明图

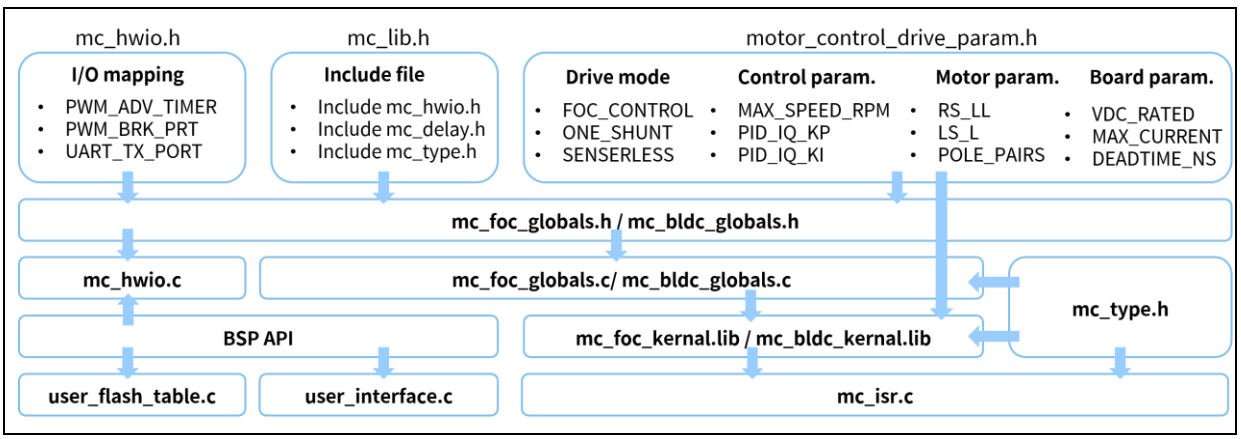


表 2 即为电机库文档说明总表，以下对于不同文档分别进行说明。

表 2. 电机库文档总表

文档名称	描述
FOC/BLDC 共通文档	
mc_lib.h	头文件统一管理
motor_control_drive_param.h	用户定义电机驱动架构模式 (电流采样模式、传感器模式…)、控制相关参数、驱动器相关参数、电机相关参数
mc_hwio.c	硬件外设配置
mc_hwio_v1.h	硬件 IO 接口宏定义配置(电机开发版 AT-MOTOR-EVB V1.x)
mc_hwio_v2.h	硬件 IO 接口宏定义配置(电机开发版 AT-MOTOR-EVB V2.x)
mc_isr.c	相关电机控制中断函数
mc_type.h	全局变量类型定义、枚举定义
mc_delay.c	时间延迟相关函数
mc_delay.h	时间延迟相关函数声明
mc_comm_uart.c	通讯界面相关外设配置
mc_comm_uart.h	通讯 uart 相关函数声明、配置
mc_pid_control.c	PID 控制器相关函数
mc_pid_control.h	PID 控制器相关函数声明
mc_curr_fdbk.c	电流检测相关函数
mc_curr_fdbk.h	电流检测相关函数声明
mc_math.c	滤波器相关函数
mc_math.h	滤波器相关函数声明
mc_hall.c	霍尔传感器相关函数
mc_hall.h	霍尔传感器相关函数声明
FOC 专用文档	
mc_foc_kernal.lib	电机库核心函数(Keil 专用)(适用于芯片有 FPU)
mc_foc_kernal_noFPU.lib	电机库核心函数(Keil 专用)(适用芯片无 FPU)
Libmc_foc_kernal.a	电机库核心函数(AT32IDE 专用)(适用于芯片有 FPU)
Libmc_foc_kernal_noFPU.a	电机库核心函数(AT32IDE 专用)(适用于芯片无 FPU)
mc_foc_kernal.h	电机库核心函数声明
motor_control_foc.c	电机控制相关函数
motor_control_foc.h	电机控制相关函数声明
mc_foc.c	矢量控制相关函数
mc_foc.h	矢量控制相关函数声明
mc_encoder.c	编码器相关函数
mc_encoder.h	编码器相关函数声明
mc_field_weakening.c	弱磁相关函数

mc_field_weakening.h	弱磁相关函数声明
mc_foc_sensorless.c	无传感器相关函数
mc_foc_sensorless.h	无传感器相关函数声明
mc_foc_globals.c	全局变量定义与默认值、全局函数声明
mc_foc_globals.h	全局变量、全局函数声明、宏定义
user_interface_foc.c	通讯界面相关函数
user_interface_foc.h	通讯界面相关函数声明
mc_flash_data_table_foc.c	写入 flash 参数表
mc_flash_data_table_foc.h	写入 flash 参数表的相关配置
BLDC 专用文档	
mc_bldc_kernal.lib	电机库核心函数(Keil 专用)(适用于芯片有 FPU)
mc_bldc_kernal_noFPU.lib	电机库核心函数(Keil 专用)(适用于芯片无 FPU)
Libmc_bldc_kernal.a	电机库核心函数(AT32IDE 专用)(适用于芯片有 FPU)
Libmc_bldc_kernal_noFPU.a	电机库核心函数(AT32IDE 专用)(适用于芯片无 FPU)
mc_bldc_kernal.h	电机库核心函数声明
motor_control_bldc.c	电机控制相关函数
motor_control_bldc.h	电机控制相关函数声明
mc_bldc.c	六步方波相关函数
mc_bldc.h	六步方波控制相关函数声明
mc_bldc_sensorless.c	无传感器相关函数
mc_bldc_sensorless.h	无传感器相关函数声明
mc_bldc_globals.c	全局变量定义与默认值、全局函数定义
mc_bldc_globals.h	全局变量、全局函数声明、宏定义
user_interface_bldc.c	通讯界面相关函数
user_interface_bldc.h	通讯界面相关函数声明
mc_flash_data_table_bldc.c	写入 flash 参数表
mc_flash_data_table_bldc.h	写入 flash 参数表的相关配置

1) motor_control_drive_param.h 文档

- 此文档主要分成四个部份，提供用户自行输入电机控制型式、电机参数、控制板参数、控制器参数等分别说明如下：
- 表 3 为模式宏定义，基于用户的硬件与电机配置，可定义适当的模式。电流采样需根据硬件设置来进行配置，三电阻、双电阻与单电阻择一定义。位置传感器也需由实际的硬件来配置，光电增量编码器、霍尔传感器或无传感器择一配置。弱磁控制则视需求选择使用与否。

表 3. 模式宏定义

宏定义名称	描述
FOC_CONTROL	矢量控制模式
SIX_STEP_CONTROL	六步方波控制模式
THREE_SHUNT	三电阻电流采样
TWO_SHUNT	双电阻电流采样
ONE_SHUNT	单电阻电流采样
U_V_SHUNT	电路使用 U 以及 V shunt 电阻(TWO_SHUNT 专用)
V_W_SHUNT	电路使用 V 以及 W shunt 电阻(TWO_SHUNT 专用)
U_W_SHUNT	电路使用 U 以及 W shunt 电阻(TWO_SHUNT 专用)
AT_MOTOR_EVB_V1	适用于电机板 V1.x 版本
AT_MOTOR_EVB_V2	适用于电机板 V2.x 版本
COMPLEMENT	开启下臂 MOS 管开关与上臂 PWM 互补(详细说明见图 6)
GATE_DRIVER_LOW_SIDE_INVERT	开启下臂 MOS 反向输出(详细说明见图 6)
EMF_COMPENSATE	反电势电压补偿
INCREM_ENCODER	光电增量编码器
REVERSE_ENCODER_COUNT	编码器反向计数(光电增量编码器)
ABZ	AB 信号含零位校正(光电增量编码器)
AB	AB 信号不含零位校正(光电增量编码器)
M_METHOD	M 速度检测方法(光电增量编码器)
MT_METHOD	M/T 速度检测方法(光电增量编码器)
HALL_SENSORS	霍尔传感器(通用)
LOW_SPEED_VOLT_CTRL	低速电压控制(FOC 霍尔传感器)
LOW_SPEED_CONTROL	低速电压控制(六步方波霍尔传感器)
WITHOUT_CURRENT_CTRL	纯电压控制,无电流环
PHASE_ADVANCE	相位提前(六步方波无传感器控制)
FIELD_WEAKENING	弱磁控制
SENSORLESS	无传感器控制的模式(通用)
OPENLOOP_STARTUP	无传感器之开环启动(通用)
ALIGN_AND_GO_STARTUP	无传感器之对齐启动(通用)
INIT_ANGLE_STARTUP	无传感器之初始角度检测启动(通用)

VOLT_SENSE	无传感器矢量控制使用电压检测
CONST_CURRENT_START	固定电流启动(六步方波无传感器控制)
CONST_VOLTAGE_START	固定电压启动(六步方波无传感器控制)
BLDC_SENSORLESS_ADC	六步方波无传感器控制以 ADC 检测反电势的模式
BLDC_SENSORLESS_COMP	六步方波无传感器控制以比较器检测反电势的模式
EMF_CONTINOUS_SAMPLE	反电势过零点侦测连续取样模式(六步方波无传感器比较器模式)
CURRENT_LP_FILTER	获得 d/q 轴电流低通滤波的信号(矢量控制)
INTERNAL_CLOCK_SOURCE	使用 MCU 内部晶振
E_BIKE_SCOOTER	电动机车/脚踏车/滑板车专用模式
DC_CURRENT_LIMIT	母线电流限制(E_BIKE_SCOOTER 专用)
RDS_AUTO_CALIBRATION	MOS 内阻自动校正(E_BIKE_SCOOTER 专用)
MOTOR_PARAM_IDENTIFY	电机线圈参数自动辨识

- 相关电机的控制参数定义如表 4 说明。根据不同的硬件、电机需求与控制特性，可定义相对应的参数。亦可进行调试，如：电流 d、q 轴 PI 控制器、速度 PI 控制器等。

表 4. 控制参数宏定义

宏定义名称	描述
BLDC/FOC 共通定义	
PWM_FREQ	PWM 输出频率 (unit: Hz)
MOTOR_CONTROL_MODE	电机控制模式(转速控制、转矩控制等……详见 type.h 里的 motor_control_mode)
CTRL_SOURCE	命令来源设置(外部来源控制/软件控制)
UI_UART_BAUDRATE	电机应用工具串口波特率。
TUNE_TARGET_CURRENT	调控 PI 参数时的目标电流大小(unit: ampere)
TUNE_CURRENT_TOTAL_PERIOD	调控 PI 参数时的总周期(unit: ms)
TUNE_CURRENT_STEP_PERIOD	调控 PI 参数时步阶周期(unit: ms)
SPEED_LOOP_FREQ	转速环控制频率 (unit: Hz)
MIN_SPEED_RPM	电机最低速度 (unit: rpm)
MAX_SPEED_RPM	电机最高速度 (unit: rpm)
MIN_CONTROL_SPEED	电机转速控制环可控的最低速度(unit: rpm)
ACC_SPD_SLOPE	加速度斜率 (unit: rpm/ms, 当 systick 频率=1kHz)
DEC_SPD_SLOPE	减速度斜率 (unit: rpm/ms, 当 systick 频率=1kHz)
SP_MAX_VOLT	外部命令来源的最大电压(unit: voltage)
SP_THRESHOLD	外部命令来源的最小有效电压(unit: voltage)
SP_RUN_VALUE	外部命令来源模式，开始驱动的最低电压值(unit: voltage)
SP_STOP_VALUE	外部命令来源模式，停止驱动的最大电压值(unit: voltage)
PID_SPD_KP_DIV_DIV	转子速度比例增益除数 (Q16 mode)

宏定义名称	描述
PID_SPD_KI_DIV_DIV	转子速度积分增益除数 (Q16 mode)
PID_SPD_KP_DEFUALT	转子速度比例增益 (Q15 mode)
PID_SPD_KI_DEFUALT	转子速度积分增益 (Q15 mode)
HYSTERESIS_LOW_SPEED	低速电压控制时电流控制切换到电压控制的最小速度值 (unit: rpm) (LOW_SPEED_VOLT_CTRL 专用)
HYSTERESIS_HIGH_SPEED	低速电压控制时电压控制切换到电流控制的最大速度值 (unit: rpm) (LOW_SPEED_VOLT_CTRL 专用)
PID_SPD_VOLT_KP_DEFUALT	低速电压控制比例增益 (Q15 mode) (LOW_SPEED_VOLT_CTRL 专用)
PID_SPD_VOLT_KI_DEFUALT	低速电压控制积分增益 (Q15 mode) (LOW_SPEED_VOLT_CTRL 专用)
PID_SPD_VOLT_KP_GAIN_DIV	低速电压控制比例增益除数 (Q16 mode) (LOW_SPEED_VOLT_CTRL 专用)
PID_SPD_VOLT_KI_GAIN_DIV	低速电压控制积分增益除数 (Q16 mode) (LOW_SPEED_VOLT_CTRL 专用)
CURRENT_BANDWIDTH	电流 PI 控制器带宽(电流 PI 参数自整定专用)
FOC 专用定义	
STABLE_SPEED_RPM	电机稳定速度 (unit: rpm) (霍尔传感器专用)
SLICK_SPEED_RPM	电机顺滑速度 (unit: rpm) (霍尔传感器专用)
MIN_POSCTL_SPD	位置环控制最低速度命令 (unit: rpm)
POSITION_LOOP_FREQ	位置环控制频率 (unit: Hz)
MAX_POSITION_ANGLE	电机最大位置 (unit: Degree)
MIN_POSITION_ANGLE	电机最小位置 (unit: Degree)
CMD_TO_VAL_GAP	当转子位置差距目标位置多少位置以内时，采用 PID_POS_KI_DEFUALT_STABLE 积分增益
SMALL_POS_CMD_GAP	当转子位置差距目标位置多少位置以内时，采用位置环控制最低速度命令(MIN_POSCTL_SPD)
ROTOR_LOCK_ANGLE_GAP	当转子位置差距目标位置多少电气角以内时锁住转子位置 (霍尔传感器专用)
PID_POS_KP_DEFUALT	转子位置比例增益 (Q15 mode)
PID_POS_KI_DEFUALT	转子位置积分增益 (Q15 mode)
PID_POS_KI_DEFUALT_STABLE	转子位置积分增益 (Q15 mode) (转子位置接近目标位置)
PID_POS_KD_DEFUALT	转子位置微分增益 (Q15 mode)
PID_POS_KP_GAIN_DIV	转子位置比例增益除数 (Q16 mode)
PID_POS_KI_GAIN_DIV	转子位置积分增益除数 (Q16 mode)
PID_POS_KD_GAIN_DIV	转子位置微分增益除数 (Q16 mode)
PID_ID_KP_DEFUALT	d 轴电流比例增益 (Q15 mode)
PID_ID_KI_DEFUALT	d 轴电流积分增益 (Q15 mode)

宏定义名称	描述
PID_ID_KP_GAIN_DIV	d 轴电流比例增益除数 (Q16 mode)
PID_ID_KI_GAIN_DIV	d 轴电流积分增益除数 (Q16 mode)
PID_IQ_KP_DEFUALT	q 轴电流比例增益 (Q15 mode)
PID_IQ_KI_DEFUALT	q 轴电流积分增益 (Q15 mode)
PID_IQ_KP_GAIN_DIV	q 轴电流比例增益除数 (Q16 mode)
PID_IQ_KI_GAIN_DIV	q 轴电流积分增益除数 (Q16 mode)
OLC_ANGLE_INC	开环控制角度增加量 (每 PWM 输出频率)
OLC_VOLT	开环控制电压输出 (unit: voltage)
ALIGN_VOLT	编码器或 HALL 自学习时的对齐电压 (unit: voltage)
LEARN_OLC_VOLT	HALL 自学习时的开环电压 (unit: voltage)
LEARN_OLC_ANGLE_INC	HALL 自学习时的角度增加量 (每 PWM 输出频率)
LEARN_TIME	HALL 自学习的时间 (unit: ms)
LEARN_ALIGN_TIME	HALL 自学习时的对齐时间 (unit: ms)
FW_MAX_ID_CURR	弱磁控制最大 d 轴电流 (unit: ampere)
FW_KP_GAIN	弱磁控制比例增益 (Q15 mode)
FW_KI_GAIN	弱磁控制积分增益 (Q15 mode)
FW_KP_GAIN_DIV	弱磁控制比例增益除数 (Q16 mode)
FW_KI_GAIN_DIV	弱磁控制积分增益除数 (Q16 mode)
CURR_LP_BANDWIDTH	d/q 轴电流低通滤波器带宽 (unit: Hz)
OBS_SPD_LP_BANDWIDTH	观测器之速度低通滤波器带宽 (unit: Hz)
OBS_GAIN1	观测器增益系数 1 (Q15 mode)
OBS_GAIN2	观测器增益系数 2 (Q15 mode)
PLL_KP_GAIN	PLL 控制比例增益 (Q15 mode)
PLL_KI_GAIN	PLL 控制积分增益 (Q15 mode)
PLL_KP_GAIN_DIV	PLL 控制比例增益除数 (Q16 mode)
PLL_KI_GAIN_DIV	PLL 控制积分增益除数 (Q16 mode)
STARTUP_MAX_SPD	切入闭环前开环最高速度 (unit: RPM)
STARTUP_CURRENT	开环或对齐启动电流 (unit: ampere)
STARTUP_OL_SLOPE	开环启动加速度 (unit: rpm/s)
STARTUP_ALIGN_TIME	启动前转子对齐时间 (unit: ms)
STARTUP_START_TIME	转子对齐后切入闭环时间 (unit: ms)
DETECT_PULSE_WIDTH	转子初始角检测脉波宽度 (unit: us)
REVERSE_MAX_SPEED_RPM	最大反转速度 (unit: rpm) (E_BIKE_SCOOTER 专用)
REVERSE_CURRENT	反转电流命令 (unit: ampere) (E_BIKE_SCOOTER 专用)
BRAKING_CURRENT	刹车电流命令 (unit: ampere) (E_BIKE_SCOOTER 专用)
BLDC 专用定义	

宏定义名称	描述
I_SAMPLE_CHANGE_DUTY	改变电流取样点的 PWM DUTY 值 (unit: PWM 计时器时基)
I_SAMPLE_MIN_DUTY	可执行电流取样的 PWM 最小 DUTY 值 (unit: PWM 计时器时基)
SENSE_HALL_TIMES	设定开环切到闭环控制前的换相次数 (unit: 次) (BLDC 无传感器专用)
REBOOT_PERIOD_MS	设定启动失败时的重启时间(unit: ms) (BLDC 无传感器专用)
SPEED_FILTER_TIMES	速度计算移动平均次数(unit:次)
INIT_SPD_COUNT	初始速度计数值(BLDC 无传感器专用)
START_CURRENT	定电流启动电流值 (unit: ampere) (BLDC 无传感器专用)
START_VOLTAGE	定电压启动电压值(unit: V 伏特)
START_PERIOD	定电流/电压启动持续时间 (unit: usec)
OLC_INIT_SPD	开环控制初始速度(unit: rpm)
OLC_FINAL_SPD	开环控制最终速度(unit: rpm)
OLC_TIMES	从初始速度递增到最终速度的递增次数(unit:次)
OLC_INIT_VOLT	开环控制初始电压(unit: V 伏特)
OLC_VOLT_INC	每一次的递增电压(unit: V/次)
OLC_STARTUP_PERIOD	开环启动切入闭环控制时间(unit: ms) (BLDC 无传感器专用)
LOCK_VOLT	启动前转子对齐电压 (unit: V 伏特) (BLDC 无传感器专用)
LOCK_PERIOD	启动前转子对齐时间 (unit: ms) (BLDC 无传感器专用)
PID_IS_KP_DIV_LOG	母线电流控制比例增益除数 (Q16 mode)
PID_IS_KI_DIV_LOG	母线电流控制积分增益除数 (Q16 mode)
PID_IS_KP_DEFUALT	母线电流控制比例增益 (Q15 mode)
PID_IS_KI_DEFUALT	母线电流控制积分增益 (Q15 mode)
EMF_CHANGE_PERCENT_H	反电势零交越检测时机由低速切换到高速区间的切换点 DUTY 值(unit: PWM 计时器时基) (BLDC 无传感器专用)
EMF_CHANGE_PERCENT_L	反电势零交越检测时机由高速切换到低速区间的切换点 DUTY 值(unit: PWM 计时器时基) (BLDC 无传感器专用)
EMF_LOW_SPD_SAMPLE_POINT	反电势零交越点在低速区间的取样位置 DUTY 值(unit: PWM 计时器时基) (BLDC 无传感器专用)
EMF_HIGH_SPD_SAMPLE_DELAY	反电势零交越点在高速区间的取样位置 DUTY 值(unit: PWM 计时器时基) (BLDC 无传感器专用)
EMF_SAMPLE_INTERVAL	连续取样模式的取样间隔(unit: PWM 计时器时基) (BLDC 无传感器专用)
EMF_LOW_SPD_CONT_SAMPLE_END	连续取样模式的低速取样终点(unit: PWM 计时器时基) (BLDC 无传感器专用)

宏定义名称	描述
EMF_HIGH_SPD_CONT_SAMPLE_DELAY	连续取样模式的高速取样终点(unit: PWM 计时器时基) (BLDC 无传感器专用)
EMF_PHASE_ADV_SPD	相位提前的最大速度(unit: rpm)
EMF_MIN_DELAY	相位提前的最小延迟时间(unit: usec)
EMF_AVOID_NOISE_INIT_PERIOD	初始延后侦测反电势零交越点的延迟时间(unit: ms)
EMF_AVOID_NOISE_TIMES	侦测反电势零交越点避开换相噪声的延迟时间(unit: PWM 计时器时基)
EMF_LOW_SPD_OFFSET_RISING	在低速区间判断为反电势正缘零交越点的位准 (unit: 电压数字转换值) (BLDC 无传感器 ADC 侦测反电势法专用)
EMF_LOW_SPD_OFFSET_FALLING	在低速区间判断为反电势负缘零交越点的位准 (unit: 电压数字转换值) (BLDC 无传感器 ADC 侦测反电势法专用)
EMF_HIGH_SPD_OFFSET_RISING	在高速区间判断为反电势正缘零交越点的位准 (unit: 电压数字转换值) (BLDC 无传感器 ADC 侦测反电势法专用)
EMF_HIGH_SPD_OFFSET_FALLING	在高速区间判断为反电势负缘零交越点的位准 (unit: 电压数字转换值) (BLDC 无传感器 ADC 侦测反电势法专用)
MAX_VOLT_CMD	低速电压控制时的最大电压命令(Q15 mode) (BLDC Hall 传感器专用)
MAX_V_CONTROL_SPD	低速电压控制时切换到电流控制的最大速度值(rpm) (BLDC Hall 传感器专用)
MIN_I_CONTROL_SPD	低速电压控制时切换到电压控制的最小速度值(rpm) (BLDC Hall 传感器专用)

- 关于驱动器的相关参数定义如表 5 所示，如死区时间、电流检测电阻、电流放大增益…等等。

表 5. 驱动器参数宏定义

宏定义名称	描述
VDC_RATED	直流母线电压值
V_SENSE_GAIN	电压回授的比例
ADC_REFERENCE_VOLT	ADC 参考电压(unit: voltage)
ADC_DIGITAL_SCALE_12BITS	ADC 分辨率
SYSTEM_CORE_CLOCK	系统频率速度 (unit: Hz)
TMR_CLK	时钟频率速度 (unit: Hz)
CHANGE_PHASE_TMR_DIV	换相时钟除频(六步方波无感专用)
DEADTIME_CLK_SFT_BITS	死区频率除频位移数
DEADTIME_NS	死区时间 (unit: ns)
MIN_INTERVAL_TIME	PWM 位移时两相信号最小间隔时间 (unit: ns)
MAX_CURRENT	电机控制器输出最大电流 (unit: ampere)
MIN_CURRENT	电机控制器输出最小电流 (unit: ampere)
DC_MAX_CURRENT	母线最大电流 (unit: ampere) (DC_CURRENT_LIMIT 专用)
CURRENT_SPAN_SHIFT	电流标么化所需的位移数
R_SHUNT	Shunt 电阻 (unit: Ω)
OP_GAIN	电流放大增益
CURR_OFFSET_VOLT	零电流偏移量 (unit: voltage)
RDC_SHUNT	母线 Shunt 电阻 (unit: Ω) (DC_CURRENT_LIMIT 专用)
DC_OP_GAIN	母线电流放大增益(DC_CURRENT_LIMIT 专用)
IDC_OFFSET_VOLT	母线零电流偏移量 (unit: voltage) (DC_CURRENT_LIMIT 专用)
EMF_SENSE_GAIN	反电势(EMF)回授的比例
OVER_CURRENT_VREF	过电流临界点 (unit: voltage)
OVER_VOLT_THRESHOLD	过电压临界点 (unit: voltage)
UNDER_VOLT_THRESHOLD	欠电压临界点 (unit: voltage)
V0_V	NTC 的电压与温度关系的近似曲线之参数 V0(注[2])
T0_C	NTC 的电压与温度关系的近似曲线之参数 T0(注[2])
dV_dT	NTC 的电压与温度关系的近似曲线之参数 dV/dT(注[2])
OVER_TEMP_THRESHOLD	过温临界点 (unit: Celsius degrees)
MC_ERROR_MASK	保护检测遮蔽

注[2]: 电压与温度关系之近似曲线方程式 V[V]=V0+dV/dT[V/Celsius](T-T0)[Celsius]*

- 相关电机参数的宏定义如表 6 所示，如极对数、编码器参数、霍尔传感器参数等。

表 6. 电机参数宏定义

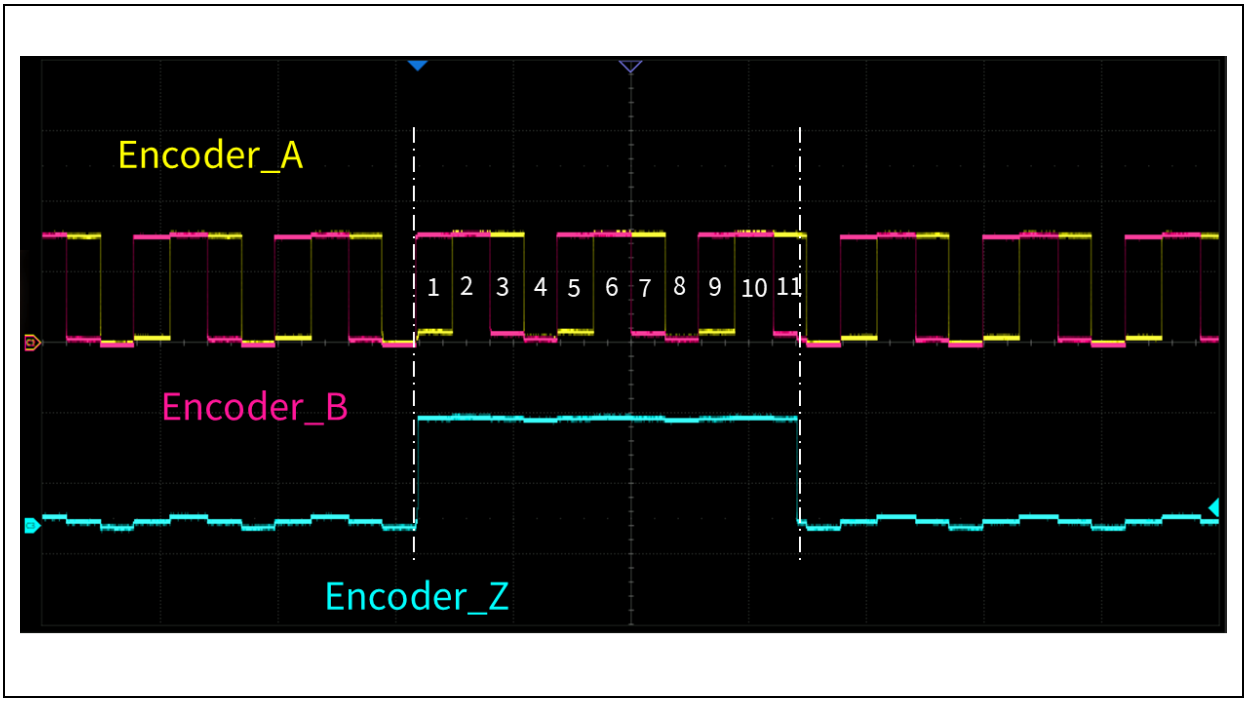
宏定义名称	描述
BLDC/FOC 通用参数	
RS_LL	电机线间电阻值(unit: Ω)
LS_LL	电机线间电感值(unit: H)
POLE_PAIRS	极对数
NOMINAL_CURRENT	电机额定电流 (unit: ampere)
HALL_LEARN_DIR	HALL 自学习后的电机转向
HALL_LEARN_0_STATE	HALL 自学习后的第一个状态(FOC: 电气角 0 度的霍尔状态(注[4])；BLDC:)
HALL_LEARN_1_STATE	HALL 自学习后的第二个状态(FOC: 电气角 60 度的霍尔状态(注[4])；BLDC:)
HALL_LEARN_2_STATE	HALL 自学习后的第三个状态(FOC: 电气角 120 度的霍尔状态(注[4])；BLDC:)
HALL_LEARN_3_STATE	HALL 自学习后的第四个状态(FOC: 电气角 180 度的霍尔状态(注[4])；BLDC:)
HALL_LEARN_4_STATE	HALL 自学习后的第五个状态(FOC: 电气角 240 度的霍尔状态(注[4])；BLDC:)
HALL_LEARN_5_STATE	HALL 自学习后的第六个状态(FOC: 电气角 300 度的霍尔状态(注[4])；BLDC:)
FOC 专用参数	
ENCODER_PPR	编码器每圈脉冲数 (unit: pulse per revolution)
ENC_IDX_COUNT	编码器零位信号宽度 (uint: count) (注[3])
ENC_STALL_TIME	编码器堵转时间 (unit: ms)
BLDC 专用参数	
ANGLE_INIT_DETECT_DUTY	初始角侦测的 PWM DUTY 值 (unit: PWM 计时器时基)
KE	电机 KE 值
OUTPUT_AH_BL_HALL_STATE	输出 A 相上臂 PWM 与 B 相下臂 ON 对应的电机霍尔状态(注[5])
OUTPUT_AH_CL_HALL_STATE	输出 A 相上臂 PWM 与 C 相下臂 ON 对应的电机霍尔状态(注[5])
OUTPUT_BH_CL_HALL_STATE	输出 B 相上臂 PWM 与 C 相下臂 ON 对应的电机霍尔状态(注[5])
OUTPUT_BH_AL_HALL_STATE	输出 B 相上臂 PWM 与 A 相下臂 ON 对应的电机霍尔状态(注[5])
OUTPUT_CH_AL_HALL_STATE	输出 C 相上臂 PWM 与 A 相下臂 ON 对应的电机霍尔状态(注[5])
OUTPUT_CH_BL_HALL_STATE	输出 C 相上臂 PWM 与 B 相下臂 ON 对应的电机霍尔状态(注[5])
HALL_STATE_ONE_NEXT_CW	HALL 状态 1 的下一个状态(正转)
HALL_STATE_TWO_NEXT_CW	HALL 状态 2 的下一个状态(正转)
HALL_STATE_THREE_NEXT_CW	HALL 状态 3 的下一个状态(正转)
HALL_STATE_FOUR_NEXT_CW	HALL 状态 4 的下一个状态(正转)
HALL_STATE_FIVE_NEXT_CW	HALL 状态 5 的下一个状态(正转)

宏定义名称	描述
HALL_STATE_SIX_NEXT_CW	HALL 状态 6 的下一个状态(正转)
HALL_STATE_ONE_NEXT_CCW	HALL 状态 1 的下一个状态(逆转)
HALL_STATE_TWO_NEXT_CCW	HALL 状态 2 的下一个状态(逆转)
HALL_STATE_THREE_NEXT_CCW	HALL 状态 3 的下一个状态(逆转)
HALL_STATE_FOUR_NEXT_CCW	HALL 状态 4 的下一个状态(逆转)
HALL_STATE_FIVE_NEXT_CCW	HALL 状态 5 的下一个状态(逆转)
HALL_STATE_SIX_NEXT_CCW	HALL 状态 6 的下一个状态(逆转)

注[3].此为光电增量编码器具有 ABZ 信号含零位校正模式下适用。

图 4 为 JK42BLS01-X056ED 编码器 ABZ 信号的关系图。Z 信号的上升沿到下降沿的宽度对齐于 AB 信号为 11 个 count 数，则定义 ENC_IDX_COUNT 为 11。(一般光电编码器通常为 1 或 2 个 count)

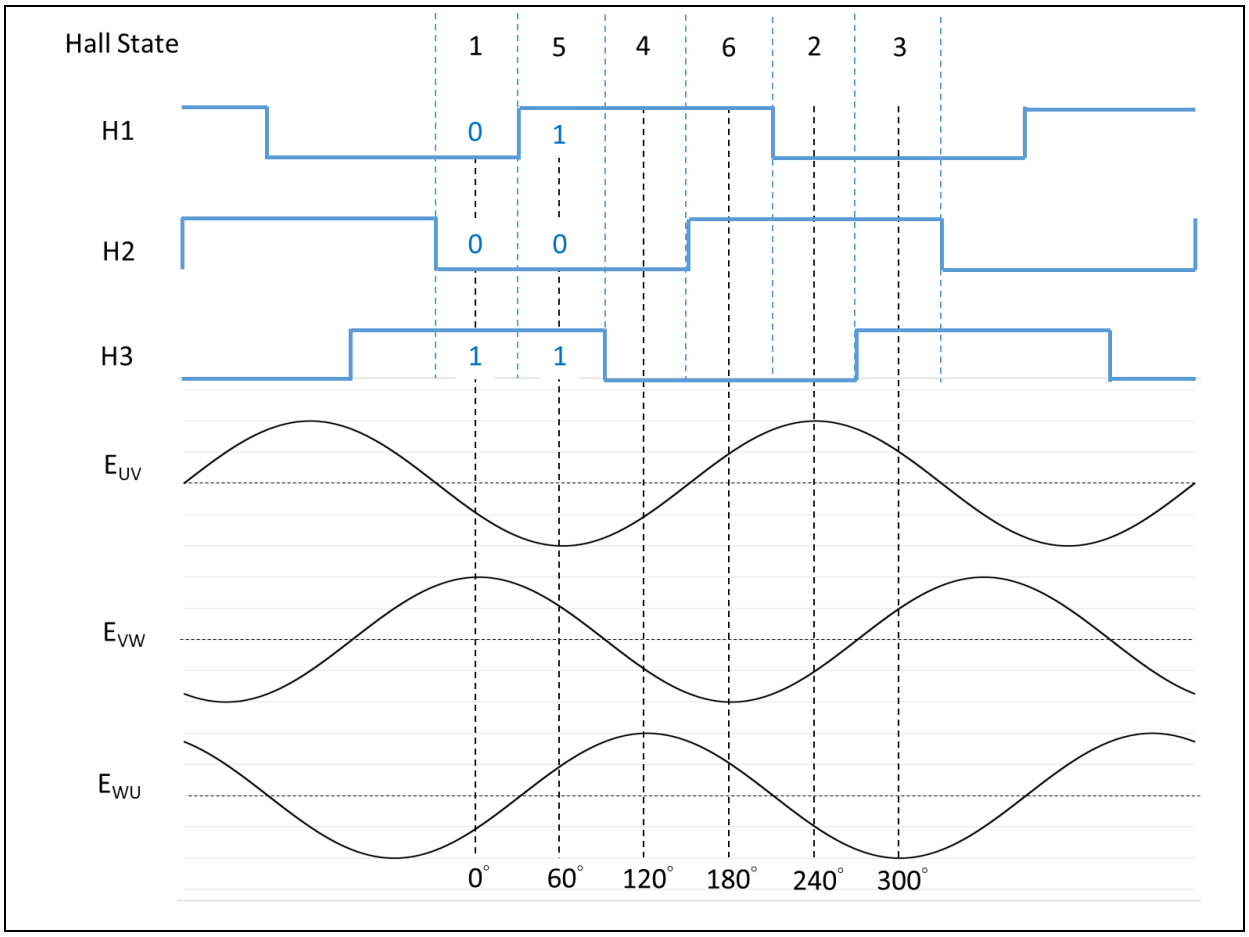
图 4. 编码器 ABZ 信号的关系图



注[4].根据电机反电势与霍尔状态的对应可设置霍尔状态顺序与对应的电气角。

图 5 为 JK42BLS01-X056ED 电机反电势、霍尔状态与电气角在 HALL_LEARN_DIR=0 时的关系图。电机三相的线间反电势需对齐此图，电气角 0 度对应 VW 线间反电势的最大值，电气角 60 度则对应 UV 线间反电势的最小值…以此类推，根据不同霍尔状态值可对应不同的电气角。用户不需自行对应，使用电机库的霍尔自学习功能可自动对应相对的霍尔状态值，只要将学习后的霍尔状态值填入对应的霍尔宏定义即可，霍尔自学习功能使用说明请参照霍尔传感器工程文档(AN0167_AT32_PMSM_FOC_Hall_Deme 或 AN0194_AT32_PMSM_FOC_Hall_Demo(E-Bike-Scooter))

图 5. PMSM 反电势、霍尔状态与电气角的关系图(HALL_LEARN_DIR=0)

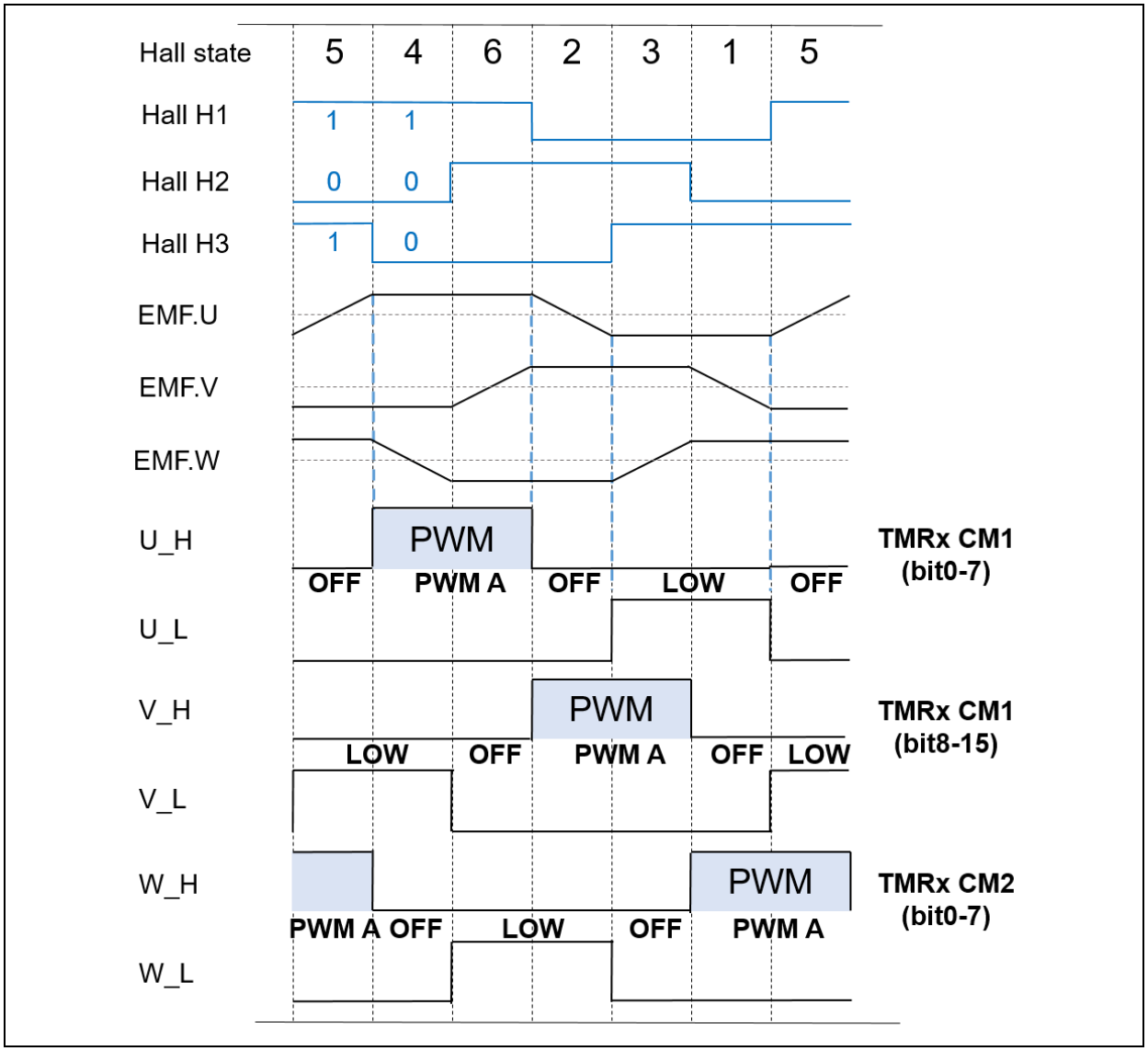


注[5].根据电机反电势与霍尔状态的对应可设置不同的 MOS 上下臂开关状态。

图 6 为 BLDC 反电势、霍尔状态与 MOS 导通状态的关系图。以此图为例，A 相反电势最大且 B 相反电势最小所对应到的霍尔状态为 4，因此于 OUTPUT_AH_BL_HALL_STATE 参数定义为 4，同理，A 相反电势最大且 C 相反电势最小所对应到的霍尔状态为 6，因此于 OUTPUT_AH_BL_HALL_STATE 参数定义为 6，以此类推将对应的 6 个霍尔状态填入定义后即可配合不同电机输出正确的 PWM 波形。

当驱动器使用下臂输入信号无反向型式的闸极驱动器(Gate driver)，且上臂 PWM 切换时下臂不输出互补 PWM 时，所对应的 PWM 输出状态如图例所示。如果所使用的闸极驱动器(Gate driver)下臂输入为反向型式、上臂 PWM 切换时下臂要输出互补 PWM 时，均可透过修改 motor_control_drive_param.h 文档内的宏定义修改成用户想要输出的模式。例如三个六步方波控制的范例工程所采用的模式即为下臂有反向输出且开启互补的模式。

图 6. BLDC 反电势、霍尔状态与 MOS 导通状态的关系图



2) mc_hwio.h 文档

- 此文档主要根据用户的硬件 IO 接口、周边进行宏定义配置。同时也包含 mc_hwio.c 文件的函数声明。

3) mc_hwio.c 文档

- 此文档主要依据用户的硬件周边进行配置。如 TMR、ADC、DMA、GPIO…等等，可根据不同的硬件来进行配置。同时也包含按钮、LED 灯等函数。详细如表 7 所示。

表 7. 周边配置相关函数

函数名称	描述
nvic_config	中断优先级配置
tmr_pwm_init	PWM 输出相关时钟(tmr)、crm clock、GPIO、DMA 配置
gpio_hall_init	霍尔传感器 GPIO 配置(BLDC 专用)
tmr_hall_init	霍尔传感器时钟(tmr)、crm clock 配置(BLDC 专用)
tmr_sensorless_change_phase_init	无传感器换相时钟(tmr)、crm clock 配置(BLDC 无传感器专用)
hall_timer_init	霍尔传感器时钟(tmr)、crm clock、GPIO 配置(FOC 专用)
encoder_time_init	增量型光电编码器时钟(tmr)、crm clock、GPIO、EXINT 配置(FOC 专用)
encoder_capture_timer_init	增量型光电编码器捕获时钟(tmr)、crm clock、GPIO(FOC/MT_METHOD 专用)
adc_ordinary_config	ADC 普通通道相关之 ADC、DMA、GPIO 配置
adc_preempt_config	ADC 抢占通道相关之 ADC、DMA、GPIO 配置
speed_timer_init	速度控制时钟(tmr)、crm clock 配置
uart_init	UART 相关 crm clock、GPIO、UART 配置
button_switch_init	switch 按钮 GPIO、crm clock 配置(BLDC hall 传感器专用)
button_exint_init	按钮中断事件 EXINT 设置
led_config	LED 初始 GPIO、crm clock 配置
led_on	LED 灯亮
led_off	LED 灯灭
led_toggle	LED 翻转(灯亮变灯灭或灯灭变灯亮)
led_init	LED 初始状态设置
led_blink	LED 闪烁
mode_switch_init	指拨开关外设配置
current_offset_tmr_setting	检测电流 offset 相关 tmr 配置(BLDC 专用)
bldc_angle_init_config	检测初始角相关 tmr 配置(BLDC 无传感器专用)
tmr_sensorless_change_phase_init	无传感器换相的 tmr 相关配置(BLDC 无传感器专用)
tmr_read_emf_init	无传感器侦测过零点连续取样模式的 tmr 相关配置(BLDC 无传感器专用)

bldc_sensorless_detectEMF_config	无传感器侦测过零点的 tmr、ADC、DMA 相关配置(BLDC 无传感器专用)
foc_angle_init_config	检测初始角相关 tmr、ADC 配置(FOC 无传感器专用)
motor_parameter_ID_config	电机线圈参数自动辨识相关 tmr、ADC 配置

4) mc_isr.c 文档

- 此文档主要为中断函数，包含 ADC、TMR、SYSTICK 等中断。详细如表 8 所示。

表 8. 电机控制相关中断函数

函数名称	描述
ADVTMR_PWM_CYCLE_IRQ	控制环中断函数 (PWM 更新中断)
ADVTMR_PWM_BRK_IRQ	刹车输入中断函数 (PWM 输出禁止)
ADC_SHUNT_SAMP_READY_IRQ	电流/反电势采样完成中断函数
ENCODER_CAPTURE_IRQ	编码器捕获中断函数(FOC 专用)
EXINT_ENCODER_IDX_IRQ	编码器零位归零中断函数(FOC 专用)
HALL_CAPTURE_IRQ	霍尔信号输入捕获中断函数
SPEED_LOOP_TIMER_IRQ	速度环中断函数(FOC 专用)
SysTick_Handler	系统中断函数 (1ms)、状态机切换状态流程
BUTTON_EXINT_IRQHandler	启动 HALL 自学习中断函数(霍尔传感器专用)、按钮中断函数
CHANGE_PHASE_IRQ	无传感器换相中断函数(BLDC 专用)
READ_EMF_IRQ	无传感器侦测过零点连续取样模式的中断函数(BLDC 专用)

5) mc_type.h 文档

- 此文档为全局枚举/类型定义。详细如表 9 所示。

表 9. 电机库枚举/类型列表

枚举/类型名称	描述
BLDC/FOC 通用参数	
firmware_id_type	区分不同电机控制模式的编号
motor_control_mode	电机控制模式(开环控制、电压控制、Id 调控模式、Iq 调控模式、转速控制、转矩控制、位置控制模式等)
ctrl_source_type	控制来源(软件控制、外部电路控制)
process_state_type	程序处理的状态
esc_state_type	电机控制流程状态机(详见章节 5.1)
err_code_type	电机错误讯息类别(过压、欠压、过温、过流、编码器错误、霍尔错误、电机线圈参数辨识错误、霍尔自学习错误)
shunt_nbr_type	相电流检测模式(单电阻、双电阻、三电阻)
hall_learn_process_type	霍尔自学习程序状态
curr_offs_type	ADC 电流取样偏移值
current_type	电流相关变量
pid_ctrl_type	PID 控制环相关变量
pid_ctrl_dc_type	PID 控制环相关变量
speed_type	速度相关变量
value_type	变量类型

hall_sensor_type	霍尔传感器相关变量
ramp_cmd_type	斜坡命令相关变量
usart_data_index	UART 伫列相关变量
moving_average_type	移动平均相关变量
usart_config_type	UART 外设配置相关变量
ui_wave_param_type	界面波形显示相关变量
i_auto_tune_type	电流 PI 控制参数自整定相关变量
motor_param_id_type	电机线圈参数自动辨识相关变量
hall_learn_type	霍尔自学习相关变量
FOC 专用参数	
low_spd_ctrl_type	低速电压控制类型
qd_type	d,q 类型
abc_type	a,b,c 类型
alphabeta_type	α,β 类型
i_dc_type	母线电流相关变量
trig_components_type	三角函数类型
voltage_type	电压相关变量
adc_trigger_type	ADC 触发相关变量
pwm_duty_type	PWM 大小相关变量
rotor_angle_type	转子角度相关变量
encoder_type	编码器相关变量
open_loop_type	开环控制相关变量
position_type	位置控制相关变量
angle_type	机械角度相关变量
field_weakening_type	弱磁相关变量
lowpass_filter_type	低通滤波器相关变量
motor_volt_type	电压输出相关变量
motor_emf_type	反电势检测相关变量
state_observer_type	无传感器观测器相关变量
sensorless_startup_type	无传感器启动相关变量
foc_angle_init_type	初始角度检测相关变量
rds_cali_type	MOS 内阻校正相关变量
BLDC 专用参数	
angle_init_type	初始角度检测相关变量(BLDC 无传感器专用)
start_state_type	检测初始角度程序的状态机(BLDC 无传感器专用)
init_current_type	初始角度检测电流相关变量(BLDC 无传感器专用)
angle_init_type	初始角度检测相关变量(BLDC 无传感器专用)

i_bus_type	母线电流相关变量
olc_type	开环控制相关变量
emf_sample_type	反电势取样相关变量(BLDC 无传感器专用)
adc_sample_type	ADC 取样相关变量(BLDC 无传感器专用)

4 电机库函数使用说明

用户根据电机控制方式可选择使用六步方波电机库(mc_bldc_kernal.lib)或是矢量控制电机库(mc_foc_kernal.lib)，两个电机库皆包含有传感器的控制函数以及无传感器的控制函数，注意在使用电机库时皆须调用初始设定相关函数以进行软件的初始设定，以下为电机库相关函数内容以及使用方式的详细说明。

4.1 通用电机库函数

初始设定相关函数(使用此电机库必须的初始函数)

- 下表描述了函数 get_fw_id

表 10. 函数 get_fw_id

项目	描述
函数名	get_fw_id
函数原型	firmware_id_type get_fw_id(void);
功能描述	获取控制模式 ID，控制模式 ID 为控制方式的分类(注[6])
输入参数	无
输出参数	无
返回值	控制模式 ID 值
先决条件	无
被调用函数	无

示例

```
firmware_id = get_fw_id();
```

注[6]. 控制方式分为弦波 FOC 控制、六步方波控制、无传感器、HALL 传感器、Encoder 编码器等分类。

- 下表描述了函数 mc_param_init

表 11. 函数 mc_param_init

项目	描述
函数名	mc_param_init
函数原型	flag_status mc_param_init(uint8_t fw_id);
功能描述	电机库函数的初始参数设定
输入参数	fw_id：控制模式 ID
输出参数	无
返回值	初始设置是否完成标志，SET 为设置完成，RESET 为设置失败
先决条件	执行 get_fw_id 函数后获得 fw_id
被调用函数	无

示例

```
param_initial_rdy = mc_param_init(firmware_id);
```

PID 控制器相关函数

- 下表描述了函数 pid_controller

表 12. 函数 pid_controller

项目	描述
函数名	pid_controller
函数原型	int16_t pid_controller(pid_ctrl_type *pid_handler, int32_t var_err);
功能描述	PID 控制器（比例-积分-微分控制器），由比例单元（Proportional）和积分单元（Integral）和微分单元（Derivative）组成
输入参数 1	pid_handler：指向结构体 pid_ctrl_type 相关参数
输入参数 2	var_err：输入误差值
输出参数	无
返回值	PID 控制器输出数值
先决条件	无
被调用函数	无

示例

```
speed_ramp.cmd_final = pid_controller(&pid_pos, pos_err_temp);
```

- 下表描述了函数 current_auto_tuning

表 13. 函数 current_auto_tuning

项目	描述
函数名	current_auto_tuning
函数原型	void current_auto_tuning(i_auto_tune_type *i_tune_handler);
功能描述	电流 PI 控制参数自整定函数
输入参数	i_tune_handler：指向结构体 i_auto_tune_type 相关参数
输出参数	i_tune_handler：指向结构体 i_auto_tune_type 相关参数
返回值	无
先决条件	无
被调用函数	无

示例

```
current_auto_tuning(&I_auto_tune);
```

数学运算相关函数

- 下表描述了函数 atan2_fixed

表 14. 函数 atan2_fixed

项目	描述
函数名	atan2_fixed
函数原型	int16_t atan2_fixed(int32_t y, int32_t x);
功能描述	定点数反正切函数
输入参数 1	y：y 轴信号
输入参数 2	x：x 轴信号
输出参数	无

项目	描述
返回值	反正切后的信号
先决条件	无
被调用函数	无

示例

```
local_degree = atan2_fixed(local_I.beta, local_I.alpha);
```

- 下表描述了函数 sqrt_fixed

表 15. 函数 sqrt_fixed

项目	描述
函数名	sqrt_fixed
函数原型	void sqrt_fixed(int32_t input, int16_t *out_ptr);
功能描述	定点数平方根函数
输入参数	input：输入参数
输出参数	out_ptr：输出参数
返回值	无
先决条件	无
被调用函数	无

示例

```
local_degree = sqrt_fixed(cal_temp, &cal_temp1);
```

电机线圈参数自动辨识相关函数

- 下表描述了函数 param_identify

表 16. 函数 param_identify

项目	描述
函数名	param_identify
函数原型	void param_identify(motor_param_id_type *mot_param_id_handler);
功能描述	电机线圈参数辨识函数
输入参数	mot_param_id_handler：指向结构体 motor_param_id_type 相关参数
输出参数	mot_param_id_handler：指向结构体 motor_param_id_type 相关参数
返回值	无
先决条件	无
被调用函数 1	tmr_channel_enable
被调用函数 2	tmr_channel_value_set

示例

```
param_identify(&motor_param_ident);
```

- 下表描述了函数 param_id_process

表 17. 函数 param_id_process

项目	描述
函数名	param_id_process

项目	描述
函数原型	void param_id_process(motor_param_id_type *mot_param_id_handler);
功能描述	电机线圈参数辨识流程
输入参数	mot_param_id_handler：指向结构体 motor_param_id_type 相关参数
输出参数	mot_param_id_handler：指向结构体 motor_param_id_type 相关参数
返回值	无
先决条件	无
被调用函数	tmr_channel_value_set

示例

```
param_id_process(&motor_param_ident);
```

4.2 矢量控制电机库函数

电流采样相关函数

- 下表描述了函数 current_read_foc_1shunt

表 18. 函数 current_read_foc_1shunt

项目	描述
函数名	current_read_foc_1shunt
函数原型	void current_read_foc_1shunt(current_type *curr_handler, voltage_type *volt_handler);
功能描述	在矢量控制时读取母线电流值进而重建出三相电流
输入参数 1	curr_handler：指向结构体 current_type 的母线电流偏移量数值
输入参数 2	volt_handler：指向结构体 voltage_type 的电压扇型控制区
输出参数	curr_handler：指向结构体 current_type 的三相电流数值
返回值	无
先决条件	无
被调用函数	adc_preempt_conversion_data_get

示例

```
current_read_foc_1shunt(&current, &volt_cmd);
```

- 下表描述了函数 rds_auto_calibration

表 19. 函数 rds_auto_calibration

项目	描述
函数名	rds_auto_calibration
函数原型	void rds_auto_calibration(rds_cali_type *rds_cali_handler, current_type *curr_handler, voltage_type *volt_handler);
功能描述	在 E_BIKE_SCOOTER 模式时自动校准 MOS 内阻
输入参数 1	rds_cali_handler：指向结构体 rds_cali_type 的 d/q 轴电流滤波数值
输入参数 2	curr_handler：指向结构体 current_type 的母线电流滤波数值
输入参数 3	volt_handler：指向结构体 voltage_type 的 d/q 轴电压数值
输出参数	rds_cali_handler：指向结构体 rds_cali_type 的 MOS 内阻自动校正
返回值	无
先决条件	无

项目	描述
被调用函数	lowpass_filtering

示例

```
rds_auto_calibration(&Rds_Cali, &current, &volt_cmd);
```

编码器相关函数

- 下表描述了函数 enc_speed_get_MTmethod

表 20. 函数 enc_speed_get_MTmethod

项目	描述
函数名	enc_speed_get_MTmethod
函数原型	int32_t enc_speed_get_MTmethod(encoder_type *enc_handler, speed_type *spd_handler);
功能描述	光电增量编码器转换后的电机转子速度(M/T Method)
输入参数 1	enc_handler：指向结构体 encoder_type 相关参数
输入参数 2	spd_handler：指向结构体 speed_type 相关参数
输出参数	无
返回值	电机转子速度
先决条件	无
被调用函数	无

示例

```
rotor_speed_encoder.val_temp = enc_speed_get_MTmethod(&encoder, &rotor_speed_encoder);
```

- 下表描述了函数 position_cmd_ramp

表 21. 函数 position_cmd_ramp

项目	描述
函数名	position_cmd_ramp
函数原型	void position_cmd_ramp(position_type *pos_handler, ramp_cmd_type *cmd_ramp_handler, pid_ctrl_type *pid_handler);
功能描述	位置步阶命令转换成 S 曲线命令
输入参数 1	pos_handler：指向结构体 position_type 相关参数
输入参数 2	cmd_ramp_handler：指向结构体 ramp_cmd_type 相关参数
输入参数 3	pid_handler：指向结构体 pid_ctrl_type 相关参数
输出参数	无
返回值	无
先决条件	无
被调用函数	无

示例

```
position_cmd_ramp(&pos, &speed_ramp, &pid_pos);
```

霍尔位置传感器相关函数

- 下表描述了函数 hall_rotor_angle_get

表 22. 函数 hall_rotor_angle_get

项目	描述
函数名	hall_rotor_angle_get
函数原型	int16_t hall_rotor_angle_get(hall_sensor_type *hall_handler, rotor_angle_type *rotor_angle_handler);
功能描述	读取霍尔位置传感器经过计算后的电机转子角度数值
输入参数 1	hall_handler：指向结构体 hall_sensor_type 的转子角度变化量
输入参数 2	rotor_angle_handler：指向结构体 rotor_angle_type 的转子角度
输出参数	无
返回值	电机转子角度
先决条件	无
被调用函数	无

示例

```
rotor_angle_hall.elec_angle_val = hall_rotor_angle_get(&hall, &rotor_angle_hall);
```

- 下表描述了函数 hall_delta_theta_calculation

表 23. 函数 hall_delta_theta_calculation

项目	描述
函数名	hall_delta_theta_calculation
函数原型	int16_t hall_delta_theta_calculation(speed_type *rotor_speed_handler, rotor_angle_type *rotor_angle_handler, hall_sensor_type *hall_handler);
功能描述	获取霍尔位置传感器转子角度变化量
输入参数 1	hall_handler：指向结构体 hall_sensor_type 的霍尔传感器状态值
输入参数 2	rotor_angle_handler：指向结构体 rotor_angle_type 的转子角度
输入参数 3	rotor_speed_handler：指向结构体 speed_type 的转子角度
输出参数	无
返回值	电机转子角度变化量
先决条件	无
被调用函数	无

示例

```
hall.theta_inc = hall_delta_theta_calculation(&rotor_speed_hall, &rotor_angle_hall, &hall);
```

脉冲宽度调制(PWM)相关函数

- 下表描述了函数 svpwm_3shunt

表 24. 函数 svpwm_3shunt

项目	描述
函数名	svpwm_3shunt
函数原型	pwm_duty_type svpwm_3shunt(voltage_type *volt_handler, pwm_duty_type *pwm_duty_handler);
功能描述	空间矢量调制函数(3-shunt 使用)
输入参数 1	volt_handler：指向结构体 voltage_type 相关参数
输入参数 2	pwm_duty_handler：指向结构体 pwm_duty_type 相关参数
输出参数	无

项目	描述
返回值	pwm_duty_type：三相 PWM 的 duty 值
先决条件	无
被调用函数	无

示例

```
pwm_duty = svpwm_3shunt(&volt_cmd, &pwm_duty);
```

- 下表描述了函数 svpwm_2shunt

表 25. 函数 svpwm_2shunt

项目	描述
函数名	svpwm_2shunt
函数原型	pwm_duty_type svpwm_2shunt(voltage_type *volt_handler, pwm_duty_type *pwm_duty_handler);
功能描述	空间矢量调制函数(2-shunt 使用)
输入参数 1	volt_handler：指向结构体 voltage_type 相关参数
输入参数 2	pwm_duty_handler：指向结构体 pwm_duty_type 相关参数
输出参数	无
返回值	pwm_duty_type：三相 PWM 的 duty 值
先决条件	无
被调用函数	无

示例

```
pwm_duty = svpwm_2shunt(&volt_cmd, &pwm_duty);
```

- 下表描述了函数 svpwm_1shunt

表 26. 函数 svpwm_1shunt

项目	描述
函数名	svpwm_1shunt
函数原型	pwm_duty_type svpwm_1shunt(voltage_type *volt_handler, pwm_duty_type *pwm_duty_handler);
功能描述	空间矢量调制函数(1-shunt 使用)
输入参数 1	volt_handler：指向结构体 voltage_type 相关参数
输入参数 2	pwm_duty_handler：指向结构体 pwm_duty_type 相关参数
输出参数	无
返回值	pwm_duty_type：三相 PWM 的 duty 值
先决条件	无
被调用函数	pwm_shift

示例

```
pwm_duty = svpwm_1shunt(&volt_cmd, &pwm_duty);
```

- 下表描述了函数 pwm_shift

表 27. 函数 pwm_shift

项目	描述
函数名	pwm_shift
函数原型	adc_trigger_type pwm_shift(int16_t *dTa, int16_t *dTb, int16_t *dTc, int16_t Ta, int16_t Tb, int16_t Tc, pwm_duty_type *pwm_duty_handler);
功能描述	控制脉冲宽度调制位移函数(1-shunt 使用)
输入参数 1	Ta：三相最大占空比
输入参数 2	Tb：三相第二大占空比
输入参数 3	Tc：三相最小占空比
输入参数 4	pwm_duty_handler：指向结构体 pwm_duty_type 相关参数
输出参数 1	dTa：三相最大占空比位移量
输出参数 2	dTb：三相第二大占空比位移量
输出参数 3	dTc：三相最小占空比位移量
返回值	adc_trigger_type：电流采样时间点
先决条件	无
被调用函数	无

示例

```
adc_trig = pwm_shift(&dTa, &dTb, &dTc, Ta, Tb, Tc, &local_pwm_duty);
```

矢量控制相关函数

- 下表描述了函数 foc_circle_limitation

表 28. 函数 foc_circle_limitation

项目	描述
函数名	foc_circle_limitation
函数原型	void foc_circle_limitation(voltage_type *volt_handler);
功能描述	限制矢量电压合成最大输出
输入参数	volt_handler：指向结构体 voltage_type 相关参数
输出参数	volt_handler：指向结构体 voltage_type 的 Q 轴电压
返回值	无
先决条件	无
被调用函数	arm_sqrt_q15

示例

```
foc_circle_limitation(&volt_cmd);
```

- 下表描述了函数 foc_vq_limitation

表 29. 函数 foc_vq_limitation

项目	描述
函数名	foc_vq_limitation
函数原型	void foc_vq_limitation(voltage_type *volt_handler, pid_ctrl_type *pid_handler);
功能描述	限制矢量电压 vq 最大输出
输入参数	volt_handler：指向结构体 voltage_type 相关参数
输出参数	pid_handler：指向结构体 pid_ctrl_type 相关参数

项目	描述
返回值	无
先决条件	无
被调用函数	arm_sqrt_q15

示例

```
foc_vq_limitation(&volt_cmd, &pid_iq);
```

矢量控制无传感器相关函数

- 下表描述了函数 startup_openloop

表 30. 函数 startup_openloop

项目	描述
函数名	startup_openloop
函数原型	flag_status startup_openloop(sensorless_startup_type *startup_handler, voltage_type *volt_handler);
功能描述	电机开环启动函数
输入参数 1	volt_handler：指向结构体 voltage_type 相关参数
输入参数 2	startup_handler：指向结构体 sensorless_startup_type 相关参数
输出参数	无
返回值	flag_status：开环启动是否完成标志，SET 为完成，RESET 为未完成
先决条件	无
被调用函数	无

示例

```
startup.closeloop_rdy = startup_openloop(&startup, &volt_cmd);
```

- 下表描述了函数 startup_alpha_axis

表 31. 函数 startup_alpha_axis

项目	描述
函数名	startup_alpha_axis
函数原型	flag_status startup_alpha_axis(sensorless_startup_type *startup_handler, voltage_type *volt_handler);
功能描述	电机定位启动函数
输入参数 1	volt_handler：指向结构体 voltage_type 相关参数
输入参数 2	startup_handler：指向结构体 sensorless_startup_type 相关参数
输出参数	无
返回值	flag_status：定位启动是否完成标志，SET 为完成，RESET 为未完成
先决条件	无
被调用函数	无

示例

```
startup.closeloop_rdy = startup_alpha_axis(&startup, &volt_cmd);
```

- 下表描述了函数 flag_status startup_angle_init

表 32. 函数 flag_status startup_angle_init

项目	描述
函数名	flag_status startup_angle_init
函数原型	flag_status startup_angle_init(sensorless_startup_type *startup_handler, voltage_type *volt_handler);
功能描述	电机初始角度检测定位启动函数
输入参数 1	volt_handler：指向结构体 voltage_type 相关参数
输入参数 2	startup_handler：指向结构体 sensorless_startup_type 相关参数
输出参数	无
返回值	flag_status：初始角度检测定位启动是否完成标志，SET 为完成，RESET 为未完成
先决条件	无
被调用函数	无

示例

```
startup.closeloop_rdy = startup_angle_init(&startup, &volt_cmd);
```

- 下表描述了函数 flag_status startup_angle_init2

表 33. 函数 flag_status startup_angle_init2

项目	描述
函数名	flag_status startup_angle_init2
函数原型	flag_status startup_angle_init2(sensorless_startup_type *startup_handler, voltage_type *volt_handler);
功能描述	电机初始角度检测开环启动函数
输入参数 1	volt_handler：指向结构体 voltage_type 相关参数
输入参数 2	startup_handler：指向结构体 sensorless_startup_type 相关参数
输出参数	无
返回值	flag_status：初始角度检测开环启动是否完成标志，SET 为完成，RESET 为未完成
先决条件	无
被调用函数	无

示例

```
startup.closeloop_rdy = startup_angle_init2(&startup, &volt_cmd);
```

- 下表描述了函数 foc_sensorless_angle_init

表 34. 函数 foc_sensorless_angle_init

项目	描述
函数名	foc_sensorless_angle_init
函数原型	void foc_sensorless_angle_init(foc_angle_init_type *angle_detect_handler, current_type *curr_handler);
功能描述	电机初始角度检测函数
输入参数 1	angle_detect_handler：指向结构体 foc_angle_init_type 相关参数
输入参数 2	curr_handler：指向结构体 current_type 相关参数

项目	描述
输出参数	angle_detect_handler：指向结构体 foc_angle_init_type 的电机初始角度
返回值	无
先决条件	无
被调用函数 1	tmr_output_enable
被调用函数 2	tmr_channel_value_set
被调用函数 3	tmr_counter_enable

示例

```
foc_sensorless_angle_init(&angle_detector, &current);
```

- 下表描述了函数 current_angle_init_3shunt

表 35. 函数 current_angle_init_3shunt

项目	描述
函数名	current_angle_init_3shunt
函数原型	void current_angle_init_3shunt(foc_angle_init_type *angle_detect_handler, current_type *curr_handler);
功能描述	3-shunt 电流采样函数(用于电机初始角度检测)
输入参数 1	angle_detect_handler：指向结构体 foc_angle_init_type 相关参数
输入参数 2	curr_handler：指向结构体 current_type 相关参数
输出参数	angle_detect_handler：指向结构体 foc_angle_init_type 的电流值
返回值	无
先决条件	无
被调用函数	adc_preempt_conversion_data_get

示例

```
current_angle_init_3shunt(&angle_detector, &current);
```

- 下表描述了函数 current_angle_init_2_1shunt

表 36. 函数 current_angle_init_2_1shunt

项目	描述
函数名	current_angle_init_2_1shunt
函数原型	void current_angle_init_2_1shunt(foc_angle_init_type *angle_detect_handler, current_type *curr_handler);
功能描述	母线电流采样函数(用于电机初始角度检测)
输入参数 1	angle_detect_handler：指向结构体 foc_angle_init_type 相关参数
输入参数 2	curr_handler：指向结构体 current_type 相关参数
输出参数	angle_detect_handler：指向结构体 foc_angle_init_type 的电流值
返回值	无
先决条件	无
被调用函数	adc_preempt_conversion_data_get

示例

```
current_angle_init_2_1shunt(&angle_detector, &current);
```

- 下表描述了函数 obs_pll_execute

表 37. 函数 obs_pll_execute

项目	描述
函数名	obs_pll_execute
函数原型	int16_t obs_pll_execute(state_observer_type *state_obs_handler, int16_t hBemf_alfa_est, int16_t hBemf_beta_est);
功能描述	正交相锁回路环估测器函数
输入参数 1	state_obs_handler：指向结构体 state_observer_type 相关参数
输入参数 2	hBemf_alfa_est：估测反电动势 α 轴电压
输入参数 3	hBemf_beta_est：估测反电动势 β 轴电压
输出参数	无
返回值	电机转子速度
先决条件	无
被调用函数 1	arm_sin_q15
被调用函数 2	arm_cos_q15
被调用函数 3	pi_controller

示例

```
hRotor_Speed = obs_pll_execute(state_obs_handler, state_obs_handler->hBemf_alpha_est,
state_obs_handler->hBemf_beta_est);
```

- 下表描述了函数 rotor_angle_sensorless

表 38. 函数 rotor_angle_sensorless

项目	描述
函数名	rotor_angle_sensorless
函数原型	int16_t rotor_angle_sensorless(state_observer_type *state_obs_handler, motor_volt_type *motor_volt_handler);
功能描述	电机转子角度估测器
输入参数 1	state_obs_handler：指向结构体 state_observer_type 相关参数
输入参数 2	motor_volt_handler：指向结构体 motor_volt_type 相关参数
输出参数	无
返回值	电机转子角度
先决条件	执行 motor_volt_calc 或 motor_volt_read 函数后获得驱动器输出 α、β 轴电压
被调用函数	obs_pll_execute

示例

```
state_observer.elec_angle = rotor_angle_sensorless(&state_observer, &motor_voltage);
```

4.3 六步方波控制电机库函数

六步方波控制相关函数

- 下表描述了函数 calc_adc_sample_point

表 39. 函数 calc_adc_sample_point

项目	描述
函数名	calc_adc_sample_point
函数原型	void calc_adc_sample_point (adc_sample_type *adc_sample,int16_t pwm_duty);
功能描述	ADC 取样时机点计算函数(包含电流取样点以及反电势(注[7])的取样点)
输入参数 1	adc_sample：指向结构体 adc_sample_type 相关参数
输入参数 2	pwm_duty：当前输出 PWM 的占空比大小
输出参数	adc_sample：指向结构体 adc_sample_type 的取样点 adc_sample_point
返回值	无
先决条件	无
被调用函数	无

示例

```
adc_sample_point_set(&adc_sample, pwm_comp_value);
```

注[7]：反电势的取样点只在无传感器(Sensorless)时使用，有检测器时只计算电流取样点

六步方波控制无传感器相关函数

- 下表描述了函数 bldc_sensorless_angle_init

表 40. 函数 bldc_sensorless_angle_init

项目	描述
函数名	bldc_sensorless_angle_init
函数原型	void bldc_sensorless_angle_init(angle_init_type *angle_init);
功能描述	母线电流采样函数(用于电机初始角度检测)
输入参数 1	angle_init：指向结构体 angle_init_type 相关参数
输出参数	angle_init：指向结构体 angle_init_type 的电流
返回值	无
先决条件	无
被调用函数 1	tmr_channel_value_set
被调用函数 2	disable_mosfet
被调用函数 3	tmr_output_enable
被调用函数 4	tmr_counter_enable

示例

```
bldc_sensorless_angle_init(PWM_ADVANCE_TIMER,&angle_init,angle_init_step);
```

- 下表描述了函数 angle_init_estimation

表 41. 函数 angle_init_estimation

项目	描述
函数名	angle_init_estimation
函数原型	uint8_t angle_init_estimation(angle_init_type *angle_init);
功能描述	起始角度估测函数
输入参数 1	angle_init：指向结构体 angle_init_type 相关参数
输入参数 2	无
输出参数	angle_init：指向结构体 angle_init_type 相关参数
返回值	检测起始角度所对应的六步方波控制相位(范围为 1-6)
先决条件	bldc_sensorless_angle_init
被调用函数	无

示例

```
hall.state = angle_init_estimation(&angle_init);
```

- 下表描述了函数 change_phase_period_ramp

表 42. 函数 change_phase_period_ramp

项目	描述
函数名	change_phase_period_ramp
函数原型	void change_phase_period_ramp(adc_sample_type *adc_sample);
功能描述	侦测过零点后延迟换相周期梯度递增/递减
输入参数 1	adc_sample：指向结构体 adc_sample_type 相关参数
输入参数 2	无
输出参数	adc_sample：指向结构体 adc_sample_type 的換相週期命令
返回值	无
先决条件	无
被调用函数	无

示例

```
change_phase_period_ramp(adc_sample);
```

5 电机库应用范例程序结构

5.1 状态机

电机库范例的状态流程如图 7 中所示的状态机流程图，它由以下状态组成：Idle、Safety ready、Angle init、Starting、Running、Free run、I_tune、Enc_align 以及 Error。

5.1.1 状态描述

Idle*

此为状态机的初始状态，此状态下马达为静止状态，遇警示状况解除后也会回到此状态。

Safety ready*

于初始状态(Idle)时确认所有参数已被设置、已取得电流 offset 的值，确认电机可以安全被启动的状态。

Angle init

无传感器控制的电机启动前检测初始角度的状态，取得初始角度后会立即跳转至 Starting 状态，此为无传感器之专有状态，若不需要检测马达初始角度则可略过这个状态。

Starting

无传感器控制时检测到起始角度后设置运转模式参数、外设配置等的状态，同时可在此状态设置切换到闭环控制的条件以增加系统稳定度。

Running*

运转模式，在此模式下马达为运转状态。用户可于 UI 界面实时调整参数(如目标速度、目标电流等)、或下命令停止马达。

Free run

相当于停止模式，在此模式下驱动器将停止输出，马达将由原速度慢慢降为 0，在马达完全停止前将会在这个状态，马达完全停止时则会回到 safety ready 模式。

I_tune*

此为调整电流 PID 控制器参数的模式，在此模式下可通过 UI 界面调整目标电流、电流环之 KP、KI 值，将会产生一步阶电流，可经由观察电流响应于此状态调整至适合参数。

Enc_align

此为编码器的零位校正状态，用户可于 UI 界面启动零位校正功能，通常在驱动器重启的情况下，需启动一次零位校正。若无启动校正，则启动电机前会自动开启零位校正。若电机无编码器则会略过这个状态。

Winding param Id

此为电机线圈参数辨识状态，用户可于 UI 界面启动自动参数辨识功能，通常在调适新电机时可做参数辨识，辨识后的线圈参数可用于无传感器矢量控制以及电流 PI 控制参数自整定。

Auto learn

此为自学习状态，目前电机库支持霍尔状态自学习功能，六步方波以及矢量控制皆可适用。

Error*

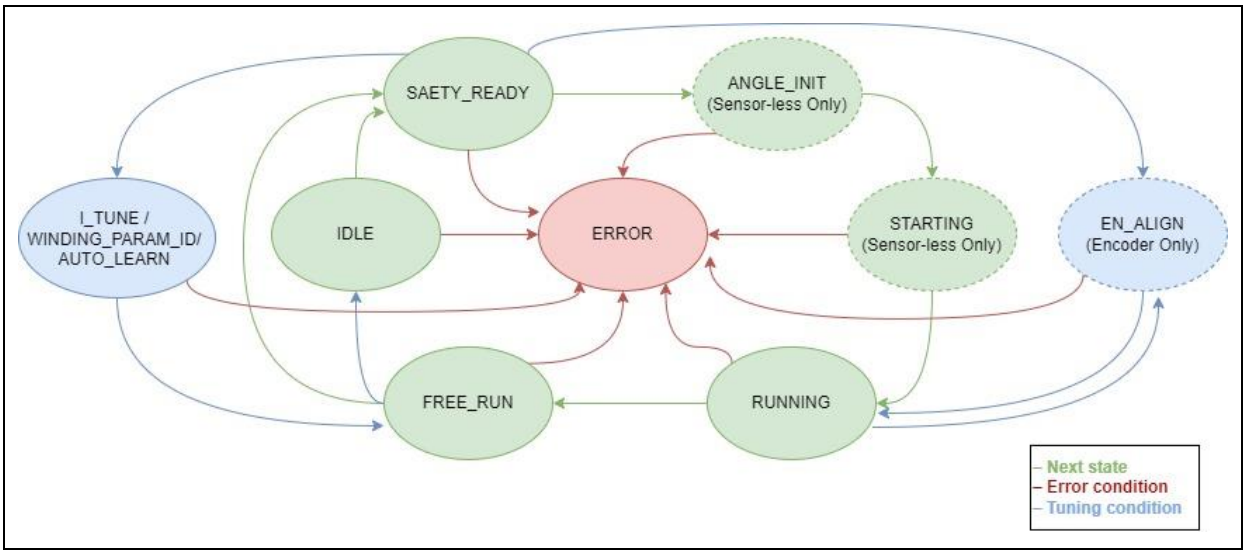
当有错误发生时将会跳至这个状态。

注[8]:上述标记星号(*)的状态会连续执行直到发生事件(用户操作或故障发生)

5.1.2 状态机流程说明

本电机库的状态机流程如图 7 所示，绿色为正常情况下一般的状态机切换流程，蓝色为特殊情况下的流程切换，其中若需调整电流的 PID 控制参数可切换至 I_TUNE 状态，若需电机线圈参数辨识功能则可切换至 WINDING_PARAM_ID 状态，若有霍尔状态自学习的需求可切换至 AUTO_LEARN 状态，若有使用到编码器则需于启动前切换至 ENC_ALIGN 状态做编码器的校准，红色则为错误报警的状态切换，程序运行中若有碰到错误(如过电压、过电流等)则会跳转至错误状态。

图 7. 状态机流程图



6 版本历史

表 43. 文档版本历史

日期	版本	变更
2022.11.18	2.0.0	初始版本
2022.11.18	2.0.1	修正文档内描述名词
2022.11.24	2.0.2	修正文档内描述名词
2022.12.01	2.0.3	更新代码定义以及描述
2022.12.10	2.0.4	电机库代码更新
2022.12.23	2.0.5	修改电机库函数格式
2022.12.29	2.0.6	文字错误修正
2023.03.02	2.0.7	增加keil V5.33编译环境限制、新增软件环境之闪存配置、统一函数描述格式、更新代码定义以及描述
2023.04.20	2.0.8	增加定位控制宏定义、中断函数、控制函数说明以及描述
2023.10.05	2.1.0	更新电机库控制程序架构、电机库函数以及电机库文档说明
2024.02.27	2.1.1	增加电机线圈参数自动辨识、电流PI控制参数自整定以及霍尔状态自学习的宏定义、中断函数、控制函数、状态机说明以及描述

重要通知 - 请仔细阅读

买方自行负责对本文所述雅特力产品和服务的选择和使用，雅特力概不承担与选择或使用本文所述雅特力产品和服务相关的任何责任。

无论之前是否有过任何形式的表示，本文档不以任何方式对任何知识产权进行任何明示或默示的授权或许可。如果本文档任何部分涉及任何第三方产品或服务，不应被视为雅特力授权使用此类第三方产品或服务，或许可其中的任何知识产权，或者被视为涉及以任何方式使用任何此类第三方产品或服务或其中任何知识产权的保证。

除非在雅特力的销售条款中另有说明，否则，雅特力对雅特力产品的使用和 / 或销售不做任何明示或默示的保证，包括但不限于有关适销性、适合特定用途（及其依据任何司法管辖区的法律的对应情况），或侵犯任何专利、版权或其他知识产权的默示保证。

雅特力产品并非设计或专门用于下列用途的产品：（A）对安全性有特别要求的应用，例如：生命支持、主动植入设备或对产品功能安全有要求的系统；（B）航空应用；（C）航天应用或航天环境； （D）武器，且/或（E）其他可能导致人身伤害、死亡及财产损害的应用。如果采购商擅自将其用于前述应用，即使采购商向雅特力发出了书面通知，风险及法律责任仍将由采购商单独承担，且采购商应独力负责在前述应用中满足所有法律和法规要求。

经销的雅特力产品如有不同于本文档中提出的声明和 / 或技术特点的规定，将立即导致雅特力针对本文所述雅特力产品或服务授予的任何保证失效，并且不应以任何形式造成或扩大雅特力的任何责任。

© 2024 雅特力科技 保留所有权利