

ARFEREY 雅特力

AT32电机库示范讲解 - BLDC 有感控制

1.16 (四) 13:30-15:30 在线培训

主讲：电机应用软件资深工程师 林明赞 博士

2025.01.16 活动议程

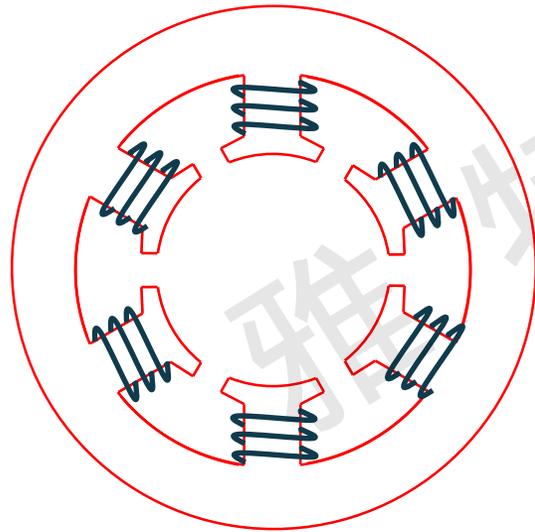
时间	主题	讲师
13:30 -14:30	BLDC控制原理与 AT32 6-step 电机控制函数库应用解说	电机应用软件 资深工程师 林明赞 博士
中场休息		
14:40 -15:30	AT32 电机库快速上手操作解说	电机应用软件 资深工程师 林明赞 博士

直流无刷电机 原理介绍

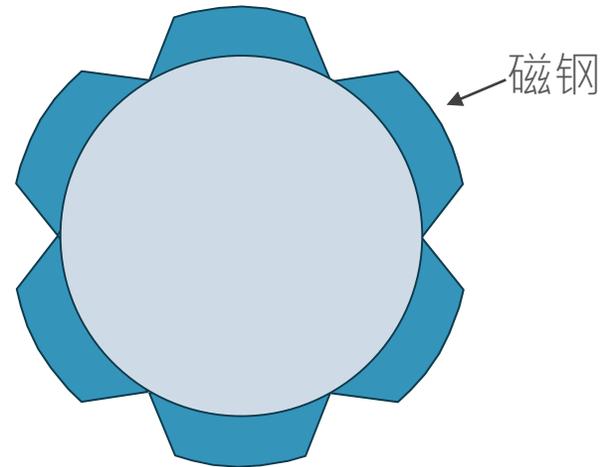


直流无刷电机 (BLDC)

- 严格定义下的直流无刷电机，其反电势为梯形波
- 由特殊充磁的梯形磁钢搭配集中式绕组，使反电势为梯形波



集中式绕组定子

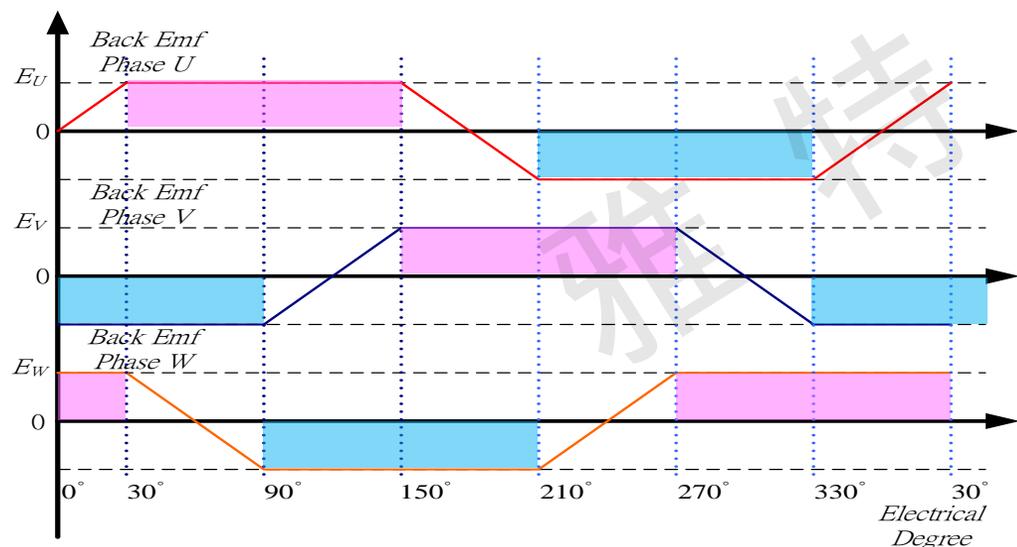


梯形磁钢转子

直流无刷电机(BLDC)/永磁同步电机(PMSM,PMAC)

• 直流无刷电机

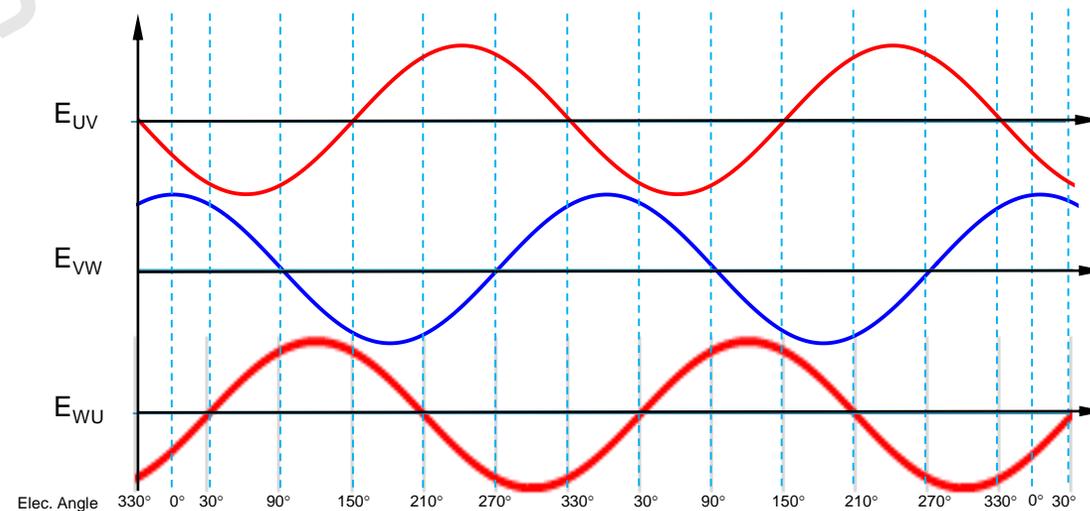
- 每次导通两相直流电流
- 控制简单可靠，扭矩特性类似直流电机
- 步阶式切换易产生转矩与速度抖动噪音较大



BLDC EMF.

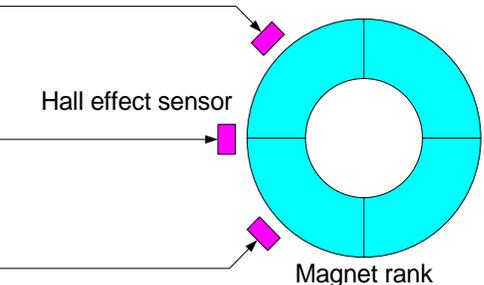
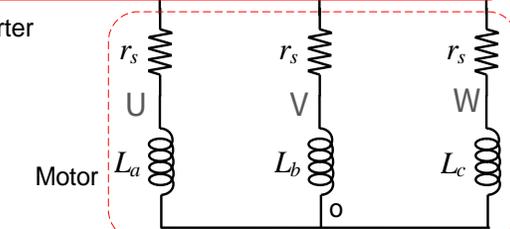
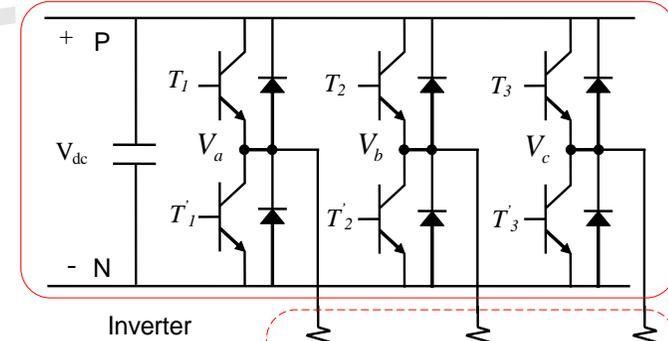
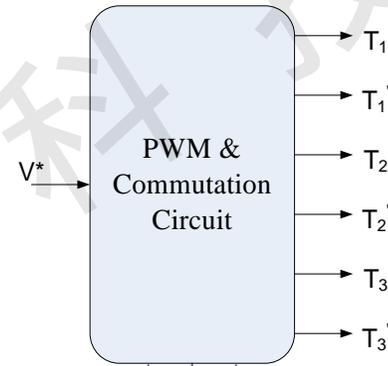
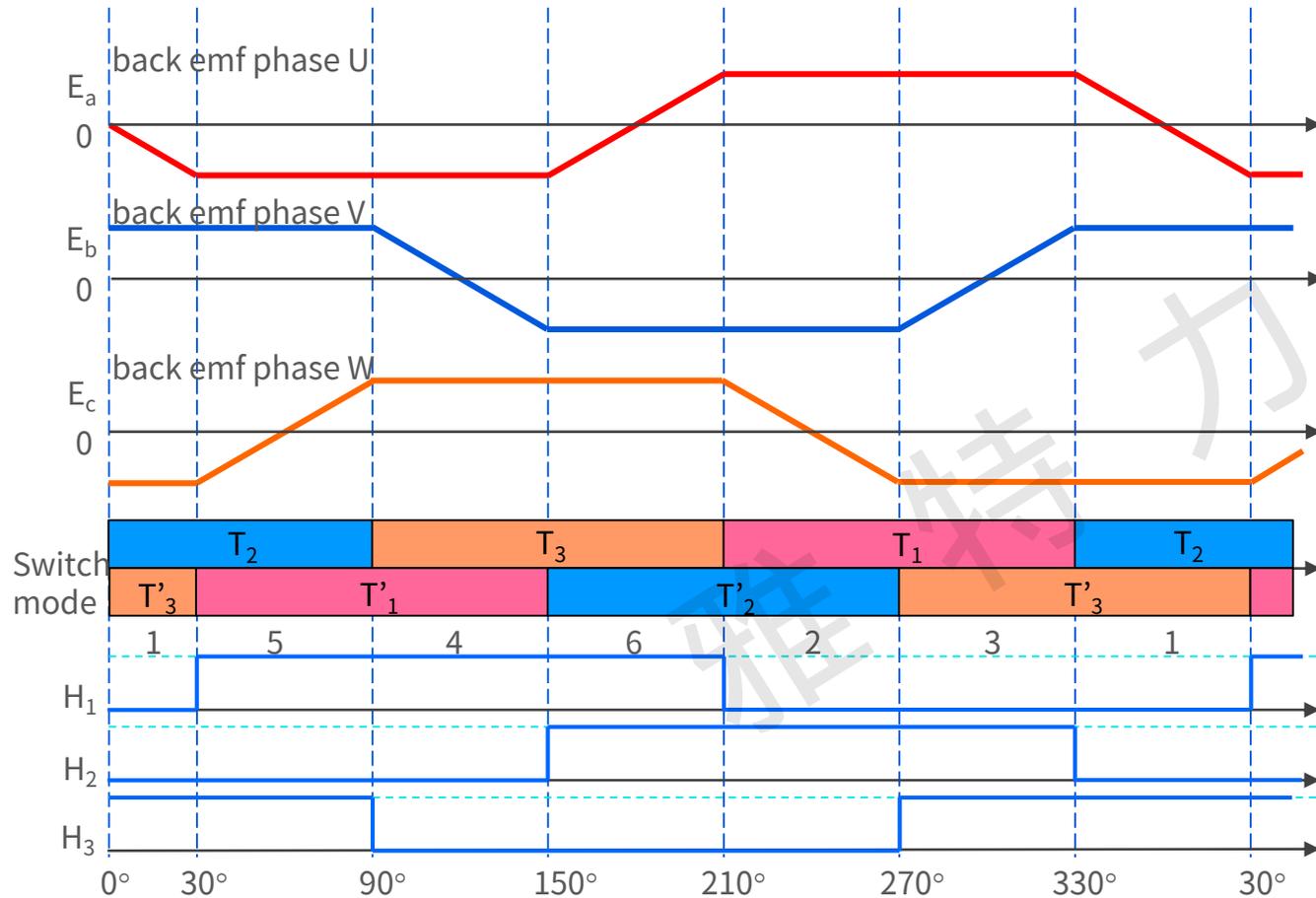
• 永磁同步电机

- 导通三相弦波电流
- 弦波电流旋转磁场，可输出稳定连续扭力，运转噪音小，效率高，响应快
- 控制法则较复杂



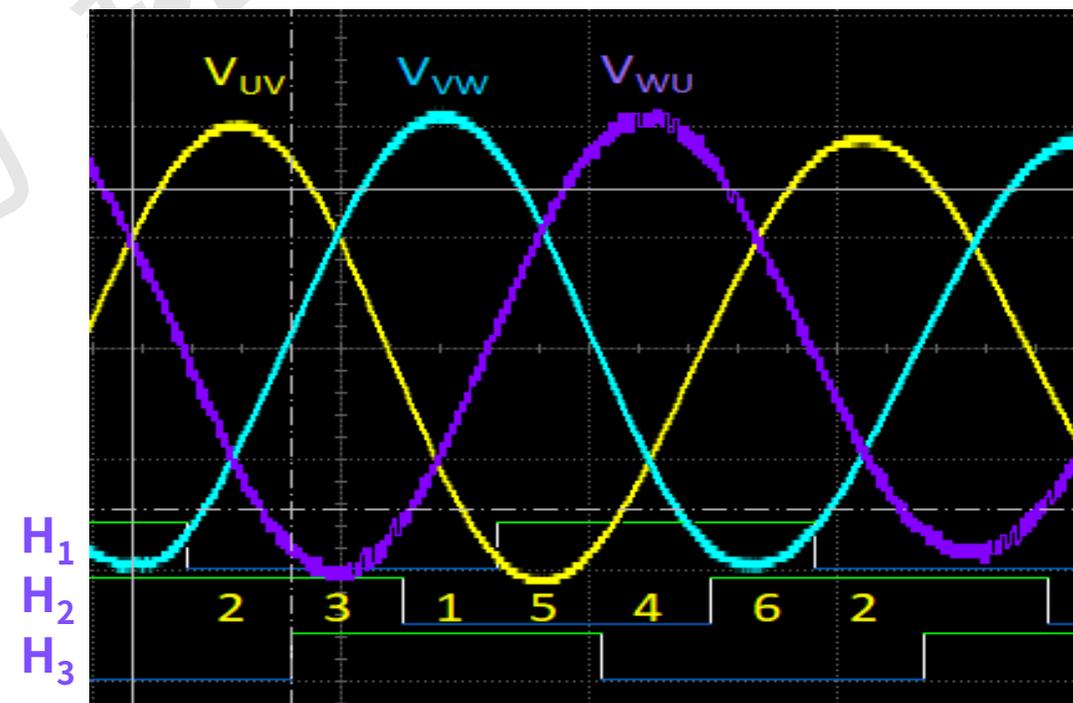
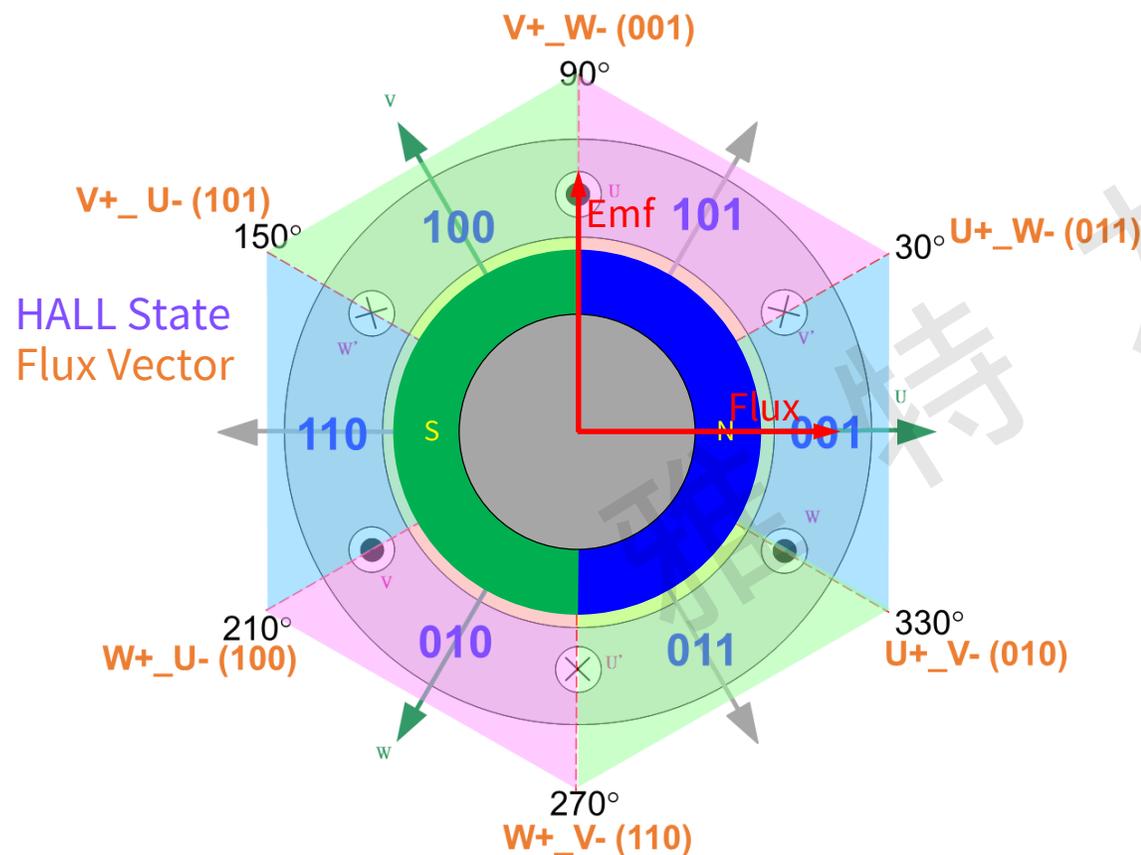
PMSM EMF.

直流无刷电机120°导通换相 (6-Step)



直流无刷电机工作原理

- 适当时机切换导通相线圈即可驱动无刷电机旋转



霍尔状态与反电势线电压关系图

无刷直流电机数学模型

- 若电机磁钢为表贴型无磁阻转矩，电机线圈电气方程式为

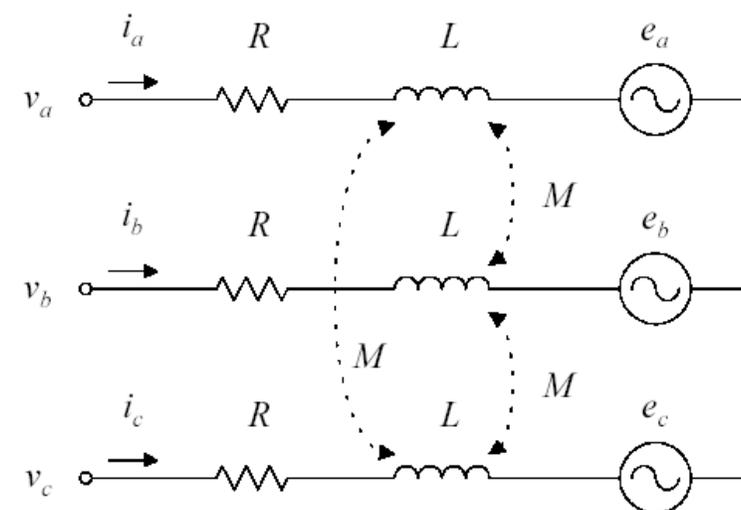
$$\begin{bmatrix} v_a \\ v_b \\ v_c \end{bmatrix} = \begin{bmatrix} R & 0 & 0 \\ 0 & R & 0 \\ 0 & 0 & R \end{bmatrix} \begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix} + p \begin{bmatrix} L & M & M \\ M & L & M \\ M & M & L \end{bmatrix} \begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix} + \begin{bmatrix} \lambda(\theta)\omega_e \\ \lambda\left(\theta - \frac{2\pi}{3}\right)\omega_e \\ \lambda\left(\theta + \frac{2\pi}{3}\right)\omega_e \end{bmatrix}$$

因电机三相电流和为零

$$i_a + i_b + i_c = 0$$

三相线圈互感关系为

$$-Mi_a = Mi_b + Mi_c$$



直流无刷电机线圈等效电路

无刷直流电机数学模型

- 电气方程式可改写为

$$\begin{bmatrix} v_a \\ v_b \\ v_c \end{bmatrix} = \begin{bmatrix} R & 0 & 0 \\ 0 & R & 0 \\ 0 & 0 & R \end{bmatrix} \begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix} + p \begin{bmatrix} L-M & 0 & 0 \\ 0 & L-M & 0 \\ 0 & 0 & L-M \end{bmatrix} \begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix} + \begin{bmatrix} \psi(\theta)\omega_e \\ \psi\left(\theta - \frac{2\pi}{3}\right)\omega_e \\ \psi\left(\theta + \frac{2\pi}{3}\right)\omega_e \end{bmatrix}$$

$$p \begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix} = \begin{bmatrix} \frac{1}{L-M} & 0 & 0 \\ 0 & \frac{1}{L-M} & 0 \\ 0 & 0 & \frac{1}{L-M} \end{bmatrix} \begin{bmatrix} v_a \\ v_b \\ v_c \end{bmatrix} - \begin{bmatrix} R & 0 & 0 \\ 0 & R & 0 \\ 0 & 0 & R \end{bmatrix} \begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix} - \begin{bmatrix} \psi(\theta)\omega_e \\ \psi\left(\theta - \frac{2\pi}{3}\right)\omega_e \\ \psi\left(\theta + \frac{2\pi}{3}\right)\omega_e \end{bmatrix}$$

其中 $\psi(\theta)$ 为随着角度变化的梯形函数， p 为微分运算符

- 采 120° 导通控制模式，每相电流可化简为

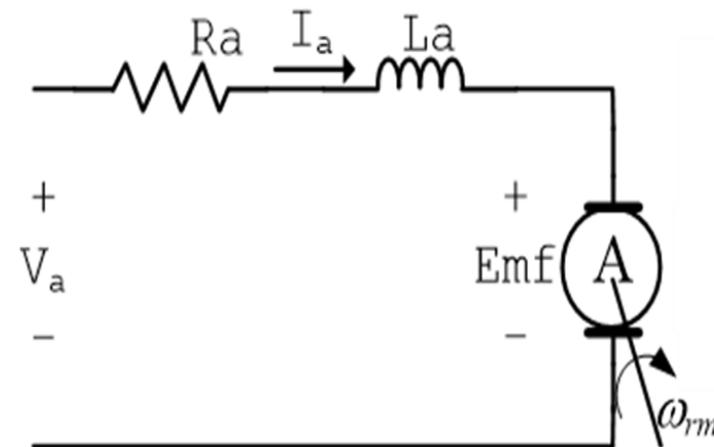
$$pi_n = \frac{1}{L-M} (v_n - Ri_n - \psi_m \omega_e)$$

- 电磁转矩方程式

$$T_e = \frac{Pole}{2} (\psi_m i_a + (-\psi_m) i_b) = Pole(\psi_m i)$$

转轴动态方程式

$$p\omega_r = \frac{1}{J} (T_e - T_l - B\omega_r)$$



直流电机示意图

转矩 / 转速 / 功率关系

- 根据转轴动态方程式，转子加速度由净转矩($T_e - T_l$)决定

$$p\omega_r = \frac{1}{J}(T_e - T_l - B\omega_r)$$

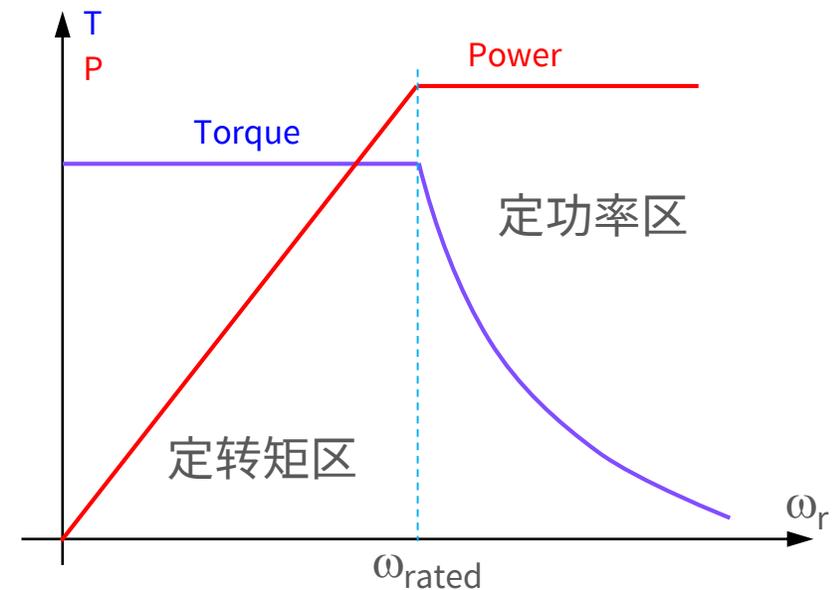
- 故转速可由电磁转矩控制 $T_e = Pole(\psi_m I_s)$

- 而转矩由电机电流 I_s 决定

- 又根据线圈电气方程式，线圈电压控制电流

$$pi_n = \frac{1}{L-M}(v_n - Ri_n - \psi_m \omega_e)$$

- 功率 $P_e = T_e \omega_r = (2 \psi_m \omega_e) I_s = BEmf_{LL} I_s$

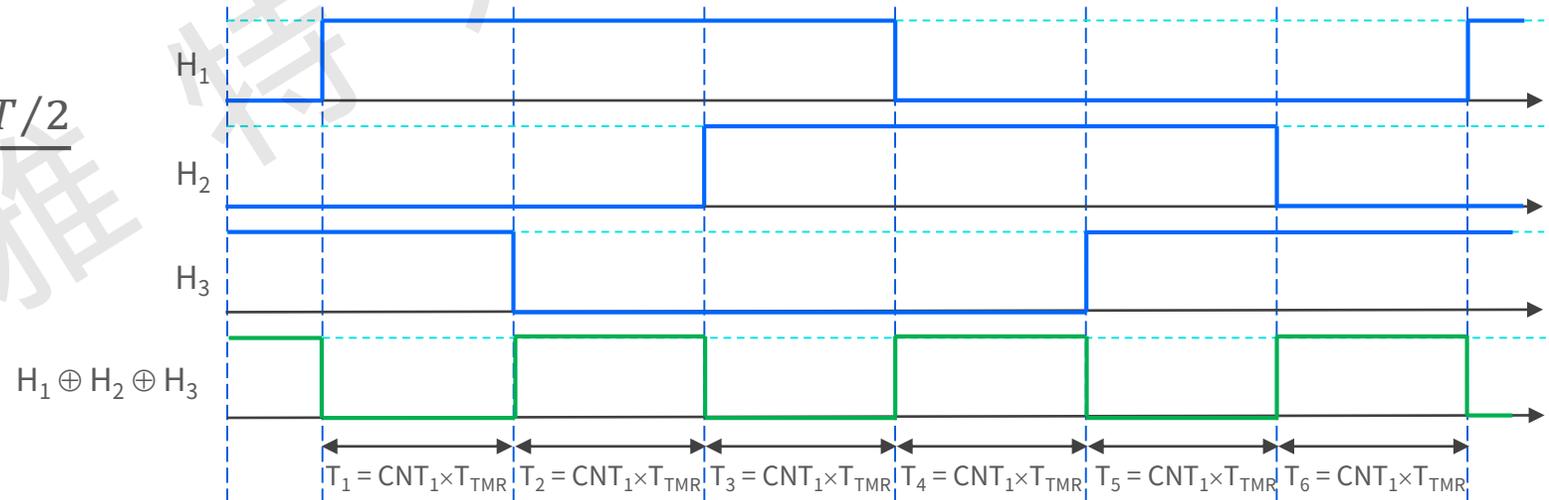


转轴速度估测

- 将三相霍尔信号经MCU定时器的斥或功能倍频
- 根据最低转速的霍尔波宽决定捕获定时器的计数脉波周期 T_{TMR}
- 在每个脉波正负缘触发定时器捕获中断同时获取波宽计数值 CNT_x
- 将6次计数值以移动平均计算均值 $CNT_{AVG} = AVERAGE(CNT_1 \sim CNT_6)$
- 计算转速

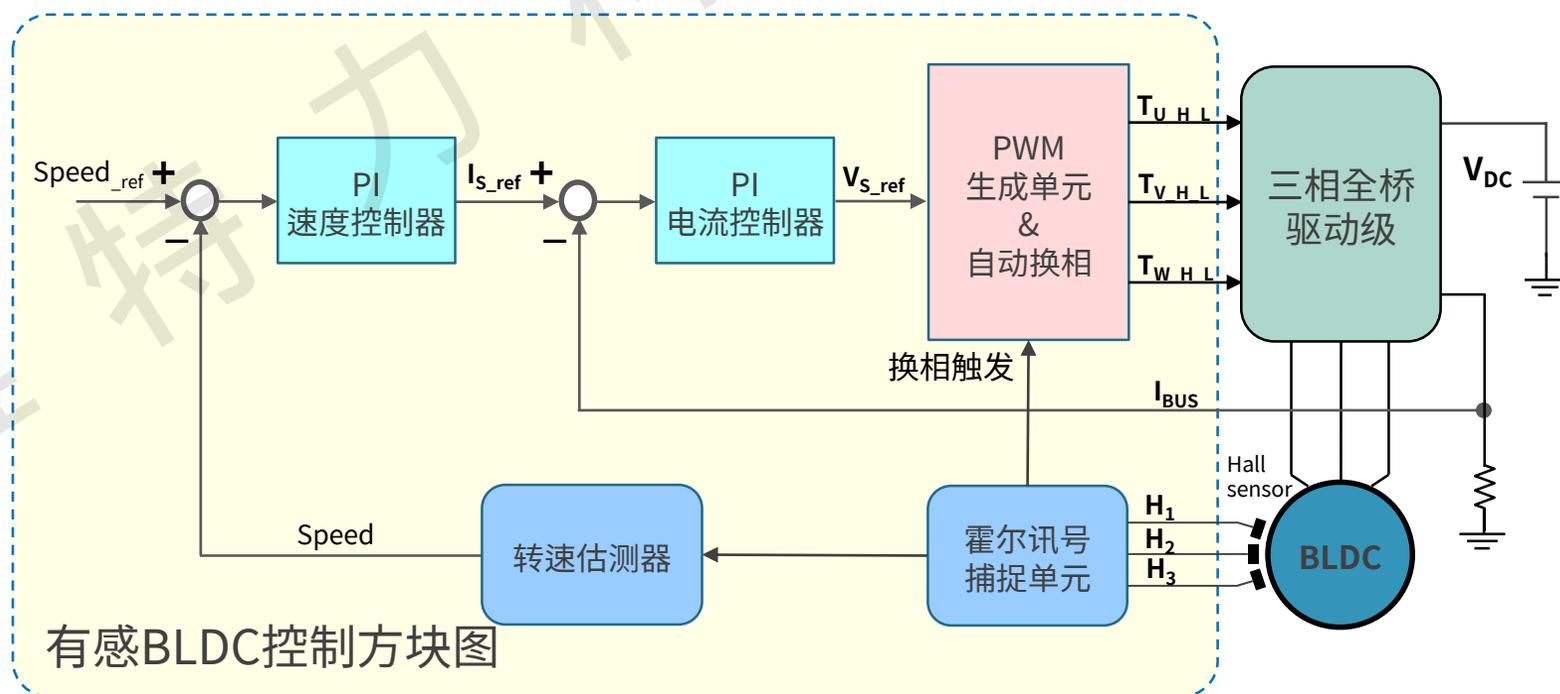
$$SPEED = \frac{MIN_SPD_CNT / 2}{CNT_{AVG}}$$

- 霍尔信号波宽决定速度取样率



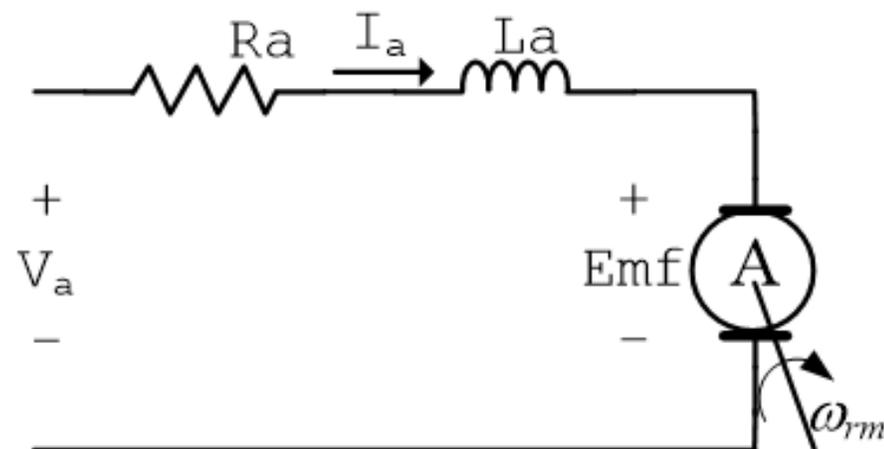
直流无刷电机控制

- 根据霍尔状态决定导通线圈
- 电流控制器决定输出电压，控制电机电流
- 控制转矩改变转速
- 速度控制器依转速误差改变电流命令



单独速度控制器 (无电流环)

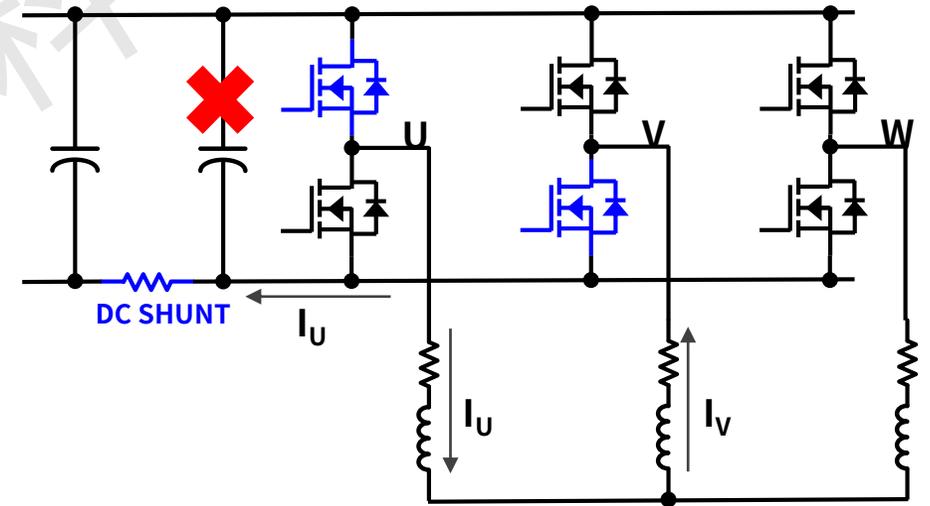
- 由速度控制器直接控制电压
- 优点：
 - 没有电流环控制架构简单，电流采样仅做过流保护
 - 低速遇负载变动时电流可自动补偿电流，转速较稳定
- 缺点：
 - 高速时因电流响应较慢，动态响应较差
 - 高速时转矩变动较大，易产生过流
 - 无法做转矩控制



直流电机示意图

速度环与电流环双环控制

- 由速度控制器决定电机电流命令，电流控制器根据电流误差控制输出电压
- 优点：
 - 电流响应快，动态响应佳，加速快
 - 可执行转矩控制，转矩变动较小
- 缺点：
 - 须获取母线瞬时电流
 - 低速时因速度取样率低，电流命令调整不实时，易产生抖动

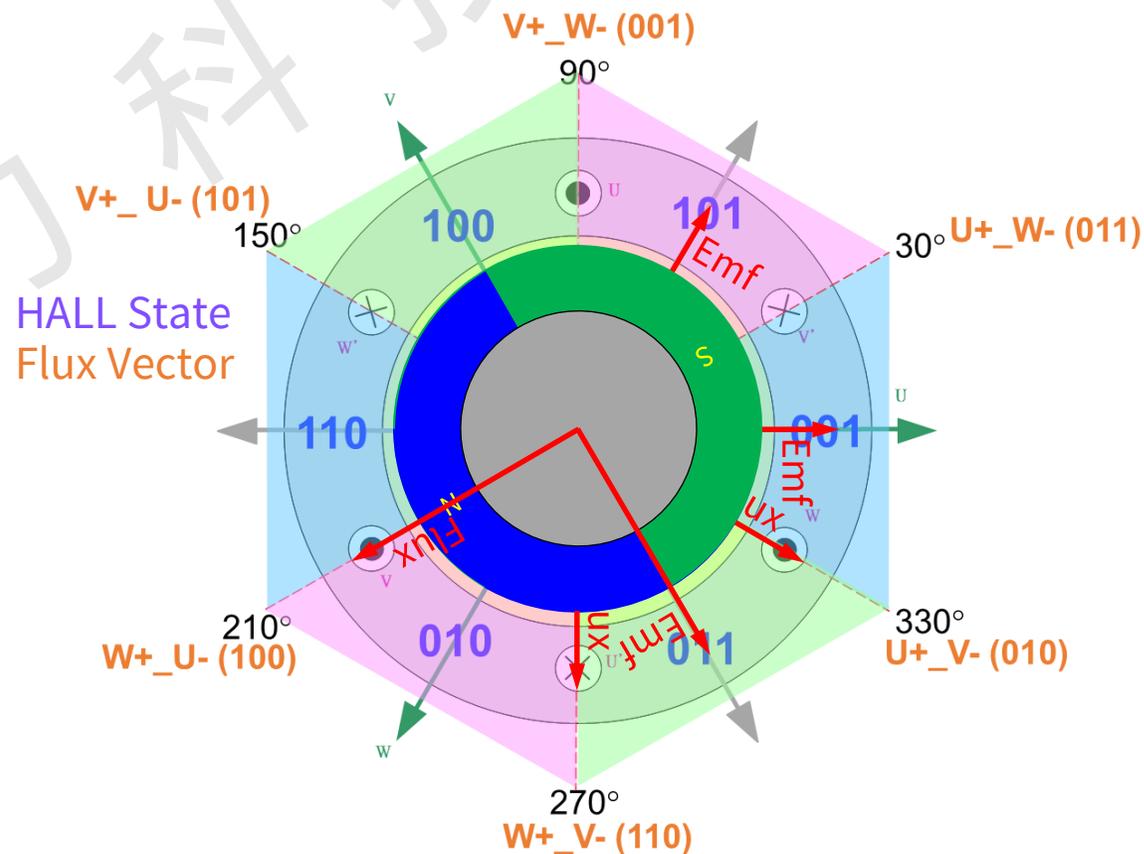


负电压输出

- 极性反相的电压矢量，其对应的霍尔状态也反相，取反相霍尔状态电压矢量，即可输出负电压

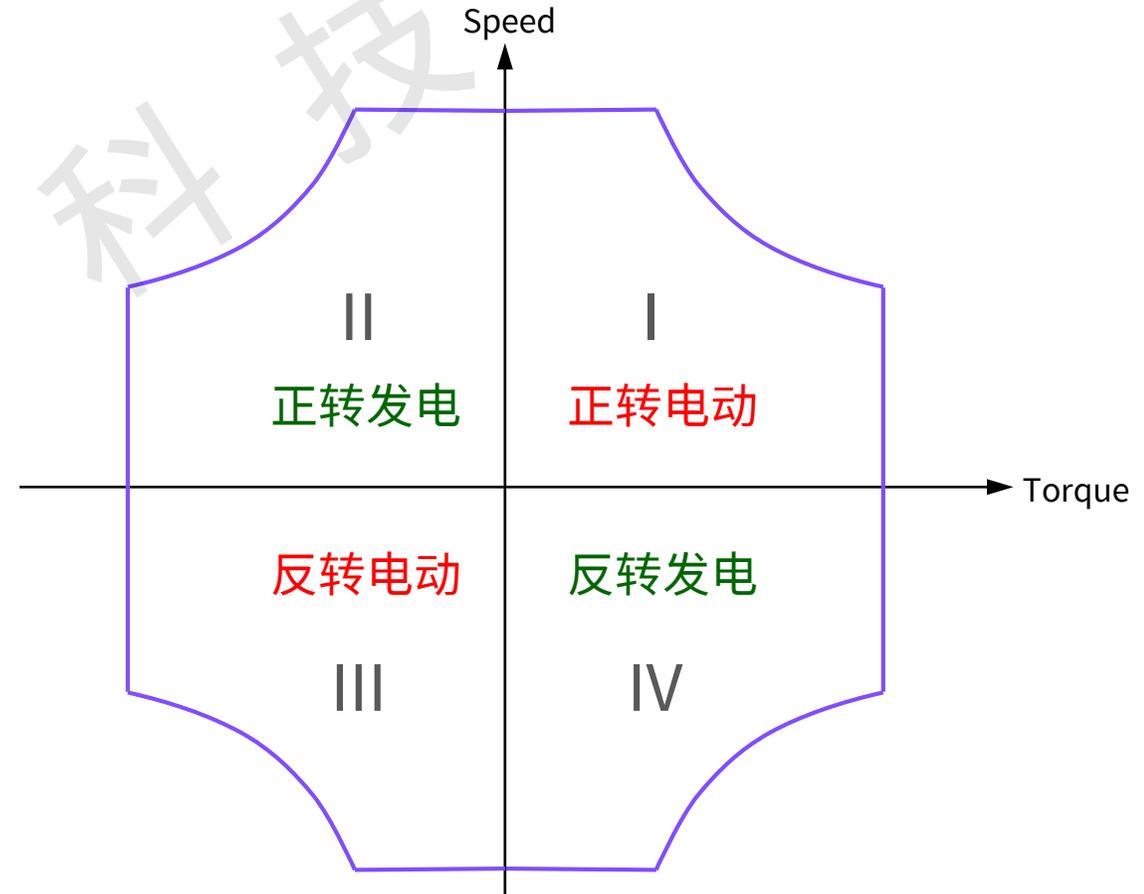
霍尔状态	5 (101)	4 (100)	6 (110)	2 (010)	3 (011)	1 (001)
正电压	V+_U- (101)	W+_U- (100)	W+_V- (110)	U+_V- (010)	U+_W- (011)	V+_W- (001)
负电压	U+_V- (010)	U+_W- (011)	V+_W- (001)	V+_U- (101)	W+_U- (100)	W+_V- (110)

霍尔状态对应电压矢量表



四象限控制

- 第一象限：正转电动模式
 - 电机正转，输出转矩与转速方向相同
 - 电能转换为机械能
- 第二象限：正转再生制动模式
 - 电机正转，但转矩方向与转速相反
 - 机械能转换为电能并回馈至电源
- 第三象限：反转电动模式
 - 电机反转，输出转矩与转速方向相同
 - 电能转换为机械能。
- 第四象限：反转再生制动模式
 - 电机反转，但转矩方向与转速相反
 - 机械能转换为电能并回馈至电源

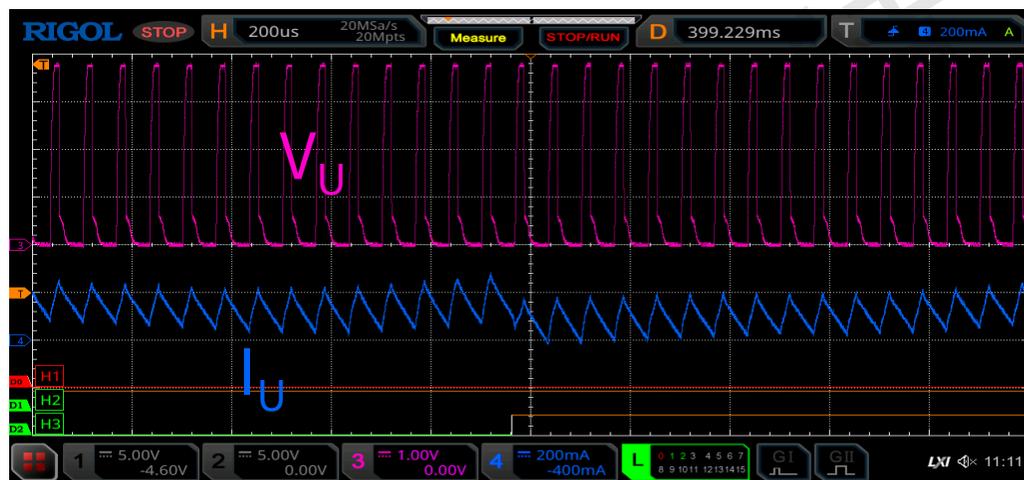
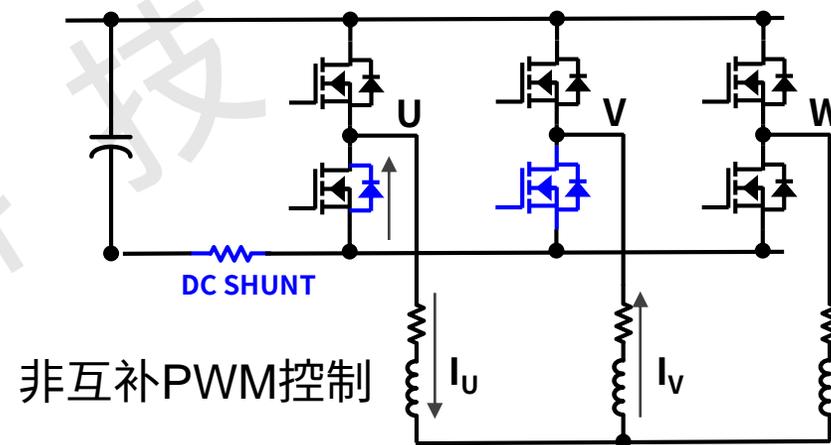


互补PWM控制与非互补PWM控制

- 互补PWM控制：上臂PWM调制时，下臂以互补的PWM信号切换
 - 上臂截止时，下臂导通压降小，电流波形更平滑，转矩波动较小
 - 大电流与回生刹车时导通损失较小
 - 增加下臂切换损失
- 非互补PWM控制：上臂PWM调制时，下臂保持关断
 - 控制简单，实现容易
 - 下臂切换损失小，但下臂二极管导通损失较大
 - 上臂截止时，电流经下臂二极管，电流波形不平滑，转矩波动较大

互补PWM控制与非互补PWM控制输出比较

- 非互补PWM控制上臂截止时，下臂二极管导通
- 因二极管导通压降大，导通损失较大
- 电流涟波也较大



互补PWM控制响应波形



非互补PWM控制响应波形

BLDC有感电机库 应用解说



有感 BLDC电机库

■ 驱动方法

- 120°方波控制

■ 相电流检测方法

- 单电阻电流检测

■ 换相时机检测方式

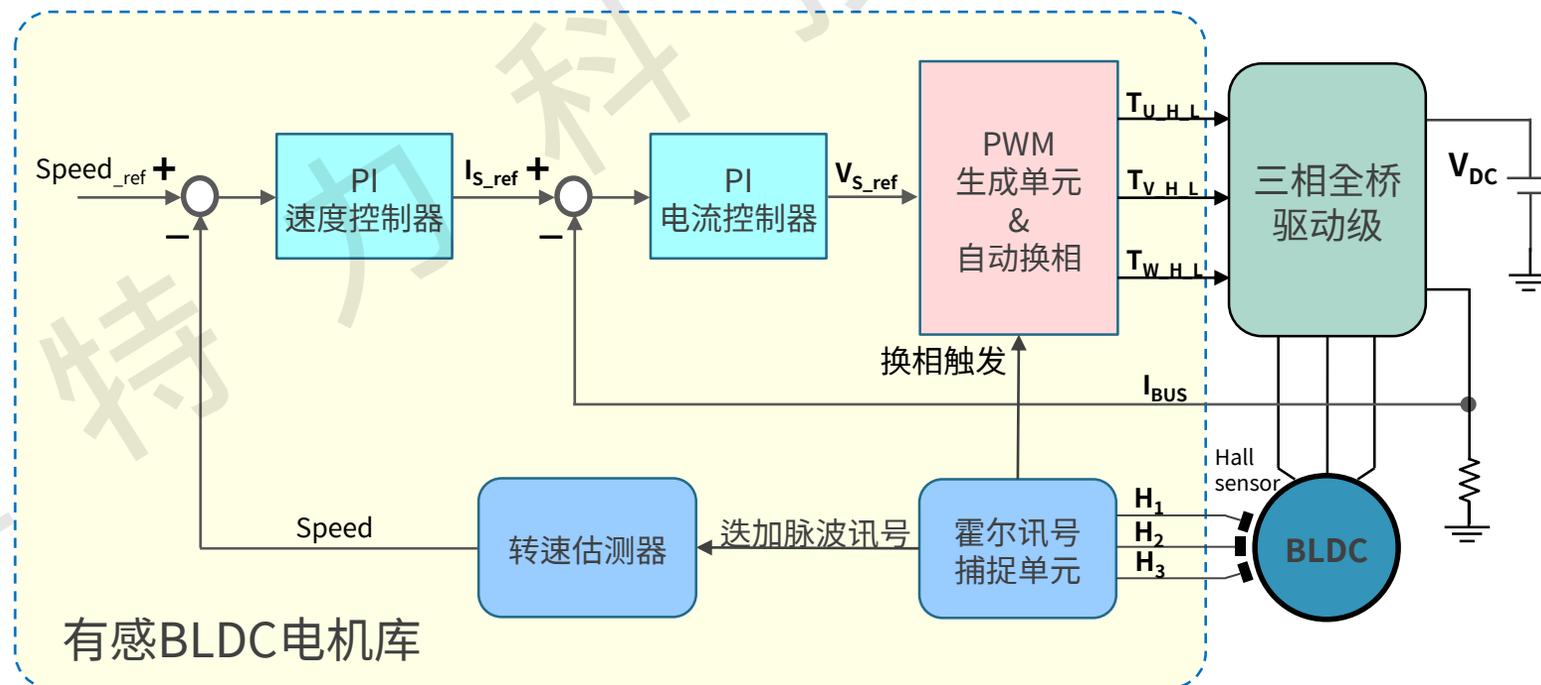
- 霍尔效应位置传感器

■ 转速估测方法

- 以霍尔讯号脉波周期推算

■ 可实现有传感器控制方法

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

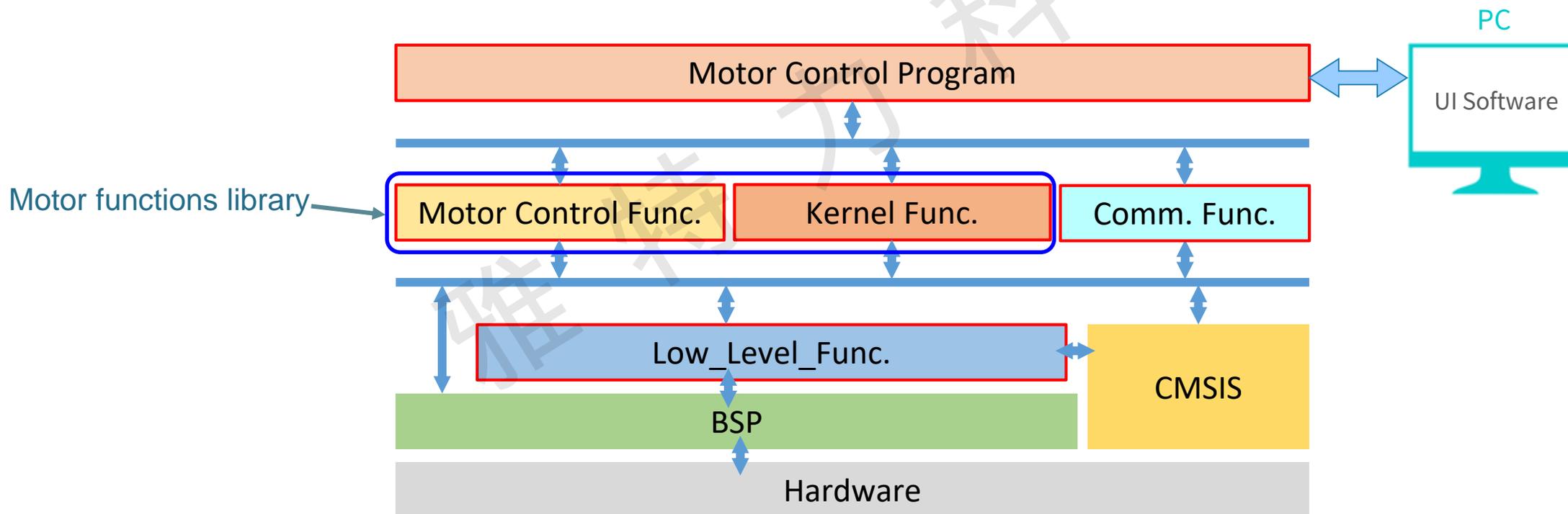


BLDC电机库控制技术与应用方案

应用产品		高速风机	低速风机	电动工具	链锯	滑板车 电动自行车	电摩	扫地机
两相 驱动	120°导通法	●	●	●	●	●	●	●
	提前激磁弱磁控制			●		●	●	●
控制 回路	转矩控制	●	●	●	●	●	●	●
	速度控制	●	●	●	●			●
电流 感测	单电阻电流感测	●	●	●	●	●	●	●
有传 感器	霍尔传感器	●	●	●	●	●	●	●
无传 感器	转子初始角度侦测	●	●	●	●			●
	反电势零交越点信号回授	●	●	●	●	●	●	●
	反电势回授侦测零交越点	●	●	●	●	●	●	●

电机库架构图

- 电机功能库、用户自定义程序和UI程序均基于BSP构建
- 基于电机函数库、用户自定义程序和UI程序设计电机控制程序



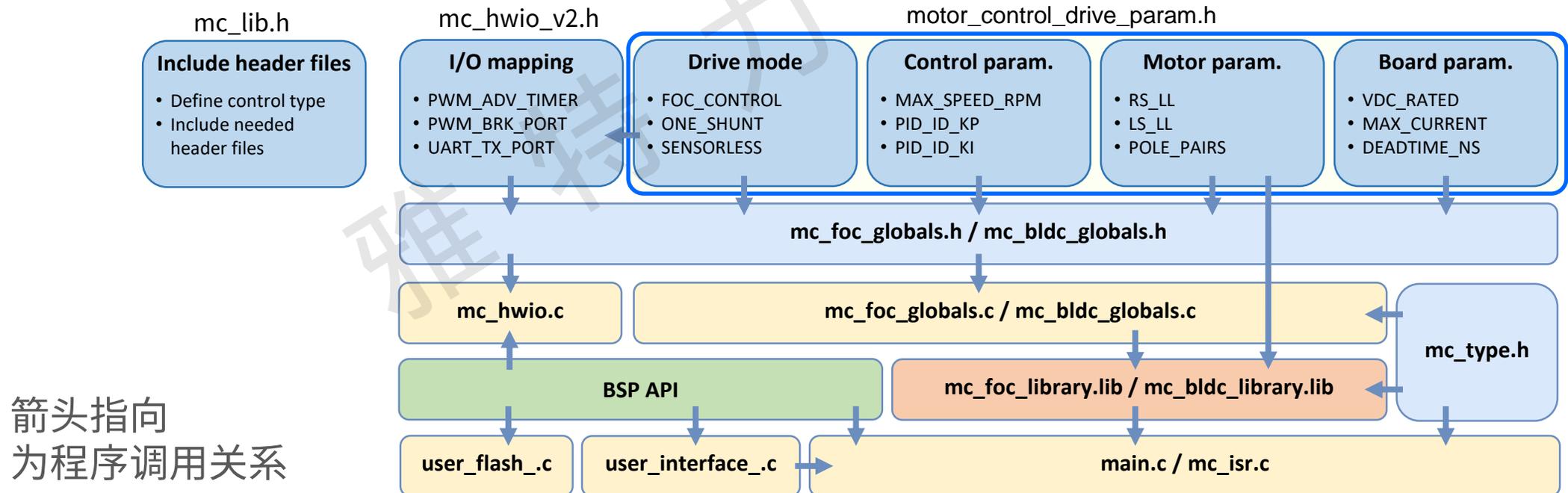
BLDC有感工程专案文件说明

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、电机参数、控制参数、驱动器参数
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	时间延迟相关函数宣告
mc_comm_uart.c	通讯界面相关外设配置
mc_comm_uart.h	通讯uart相关函数宣告、配置
mc_pid_control.c	PID控制器相关函数
mc_pid_control.h	PID控制器相关函数宣告
mc_curr_fdbk.c	电流检测相关函数
mc_curr_fdbk.h	电流检测相关函数宣告
mc_math.c	滤波器相关函数
mc_math.h	滤波器相关函数宣告
mc_hall.c	霍尔传感器相关函数
mc_hall.h	霍尔传感器相关函数宣告

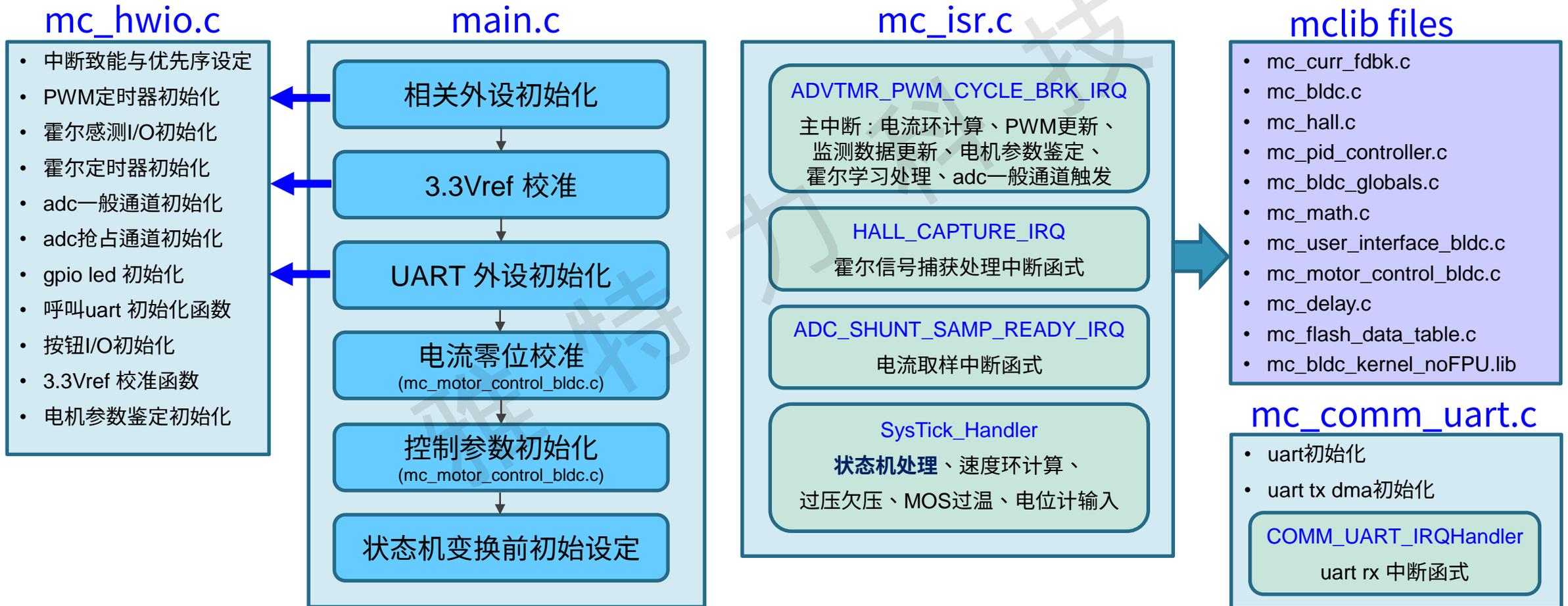
BLDC专用文档	
mc_blcdc_kernal.lib	电机库核心函数(Keil专用)(适用于芯片有FPU)
mc_blcdc_kernal_noFPU.lib	电机库核心函数(Keil专用)(适用于芯片无FPU)
libmc_blcdc_kernal.a	电机库核心函数(AT32IDE专用)(适用于芯片有FPU)
libmc_blcdc_kernal_noFPU.a	电机库核心函数(AT32IDE专用)(适用于芯片无FPU)
mc_blcdc_kernal.h	电机库核心函数宣告
motor_control_blcdc.c	电机控制相关函数
motor_control_blcdc.h	电机控制相关函数宣告
mc_blcdc.c	六步方波相关函数
mc_blcdc.h	六步方波控制相关函数宣告
mc_blcdc_sensorless.c	无传感器相关函数
mc_blcdc_sensorless.h	无传感器相关函数宣告
mc_blcdc_globals.c	全局变量定义与默认值设定、全局函数定义
mc_blcdc_globals.h	全局变量、全局函数宣告、宏定义
user_interface_blcdc.c	通讯界面相关函数
user_interface_blcdc.h	通讯界面相关函数宣告
mc_flash_data_table_blcdc.c	写入flash参数表
mc_flash_data_table_blcdc.h	写入flash参数表的相关配置

电机库文件关系图

- 头文件提供设定MCU外设、控制形式、电机参数、控制板参数、控制器参数
- 相关的设定参数于 mc_xxx_globals.c中的函数设定变数初值
- MCU外设规划，则由mc_hwoio.c文件执行相关外设初使化设定

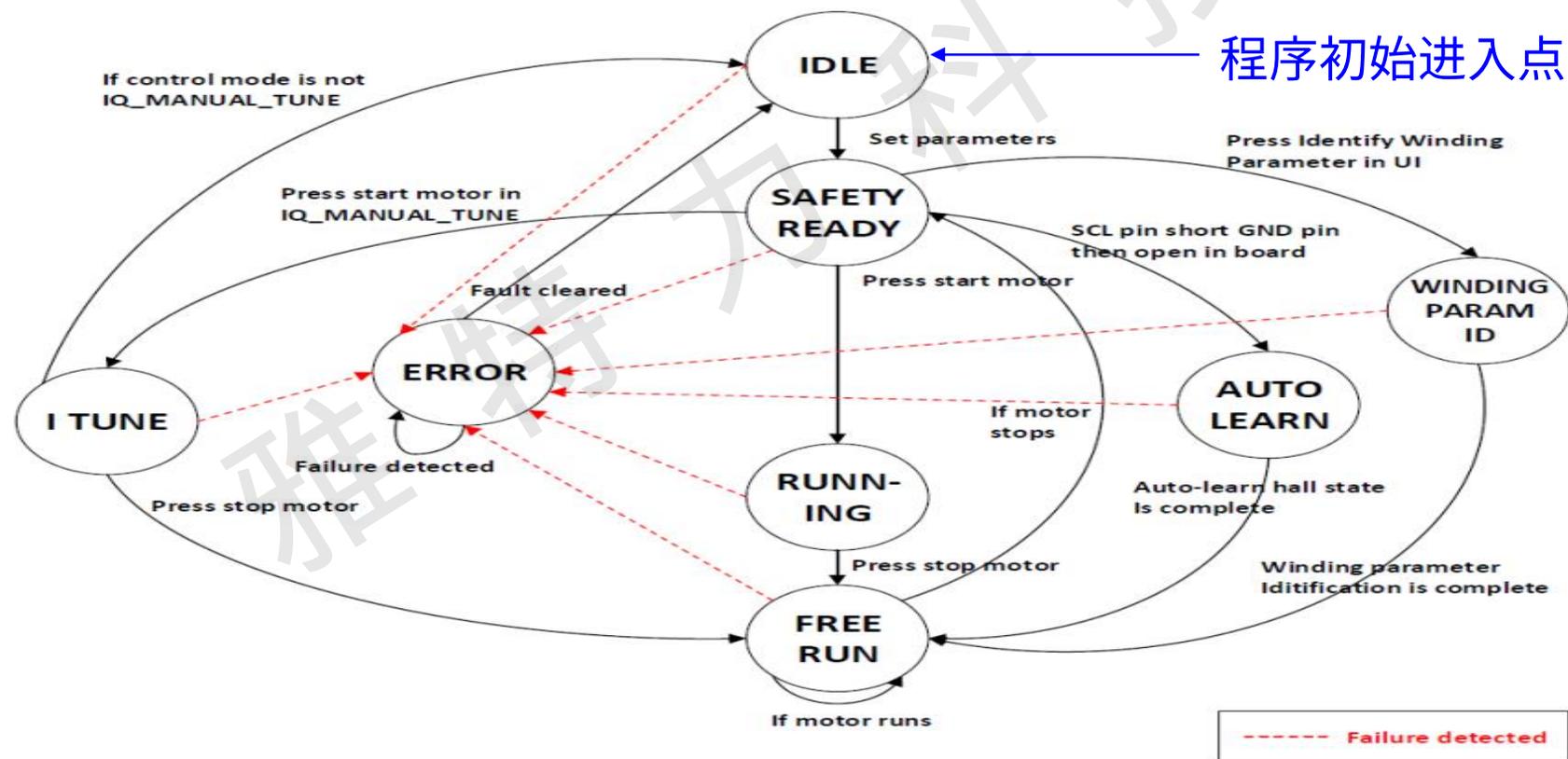


BLDC有感控制程序流程图



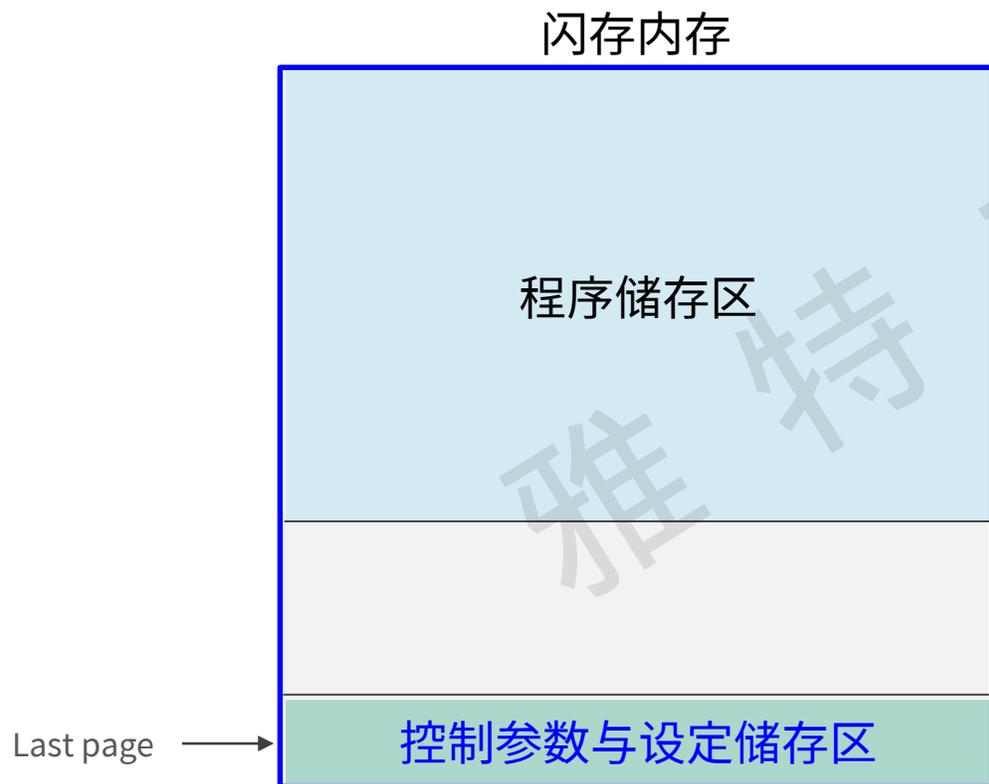
程序状态机流程图

- 程序主要工作于Running运转模式，用户可于UI界面实时调整参数或用外部电位计改变命令



程序闪存配置

- 程序将控制参数与设定储存区放于闪存最后一页



Flash size	4032K	1024K	512K	448K	256K
IROM1(address)	0x8000000	0x8000000	0x8000000	0x8000000	0x8000000
IROM1(size)	0x1FF000	0xFF800	0x7F800	0x6F000	0x3F800
IROM2(address)	0x81FF000	0x80FF800	0x807F800	0x0806F000	0x803F800
IROM2(size)	0x1000	0x800	0x800	0x1000	0x800

Flash size	128K	64K	32K	16K
IROM1(address)	0x8000000	0x8000000	0x8000000	0x8000000
IROM1(size)	0x1FC00	0x0FC00	0x07C00	0x03C00
IROM2(address)	0x801FC00	0x800FC00	0x8007C00	0x8003C00
IROM2(size)	0x400	0x400	0x400	0x400

对应闪存存储空间 ROM 配置表

The screenshot shows the "Read/Only Memory Areas" and "Read/Write Memory Areas" configuration dialog. The "on-chip" section for "Read/Only Memory Areas" is highlighted with a red box, showing the following configuration:

default	off-chip	Start	Size	Startup
<input type="checkbox"/>	ROM1:			<input type="radio"/>
<input type="checkbox"/>	ROM2:			<input type="radio"/>
<input type="checkbox"/>	ROM3:			<input type="radio"/>
on-chip				
<input checked="" type="checkbox"/>	IROM1:	0x8000000	0xFC00	<input checked="" type="radio"/>
<input checked="" type="checkbox"/>	IROM2:	0x800FC00	0x400	<input type="radio"/>

The "Read/Write Memory Areas" section shows:

default	off-chip	Start	Size	NoInit
<input type="checkbox"/>	RAM1:			<input type="checkbox"/>
<input type="checkbox"/>	RAM2:			<input type="checkbox"/>
<input type="checkbox"/>	RAM3:			<input type="checkbox"/>
on-chip				
<input checked="" type="checkbox"/>	IRAM1:	0x20000000	0x4000	<input type="checkbox"/>
<input type="checkbox"/>	IRAM2:			<input type="checkbox"/>

Buttons: OK, Cancel, Defaults, Help

电机库下载

ARTERY | 雅特力科技

关于雅特力 产品讯息 技术与开发支持 应用方案 新闻中心 联系雅特力

21ic LCES 天猫

应用文件支持

设计资源

电机控制

USB应用

以太网应用

培训中心

电机控制

USB应用

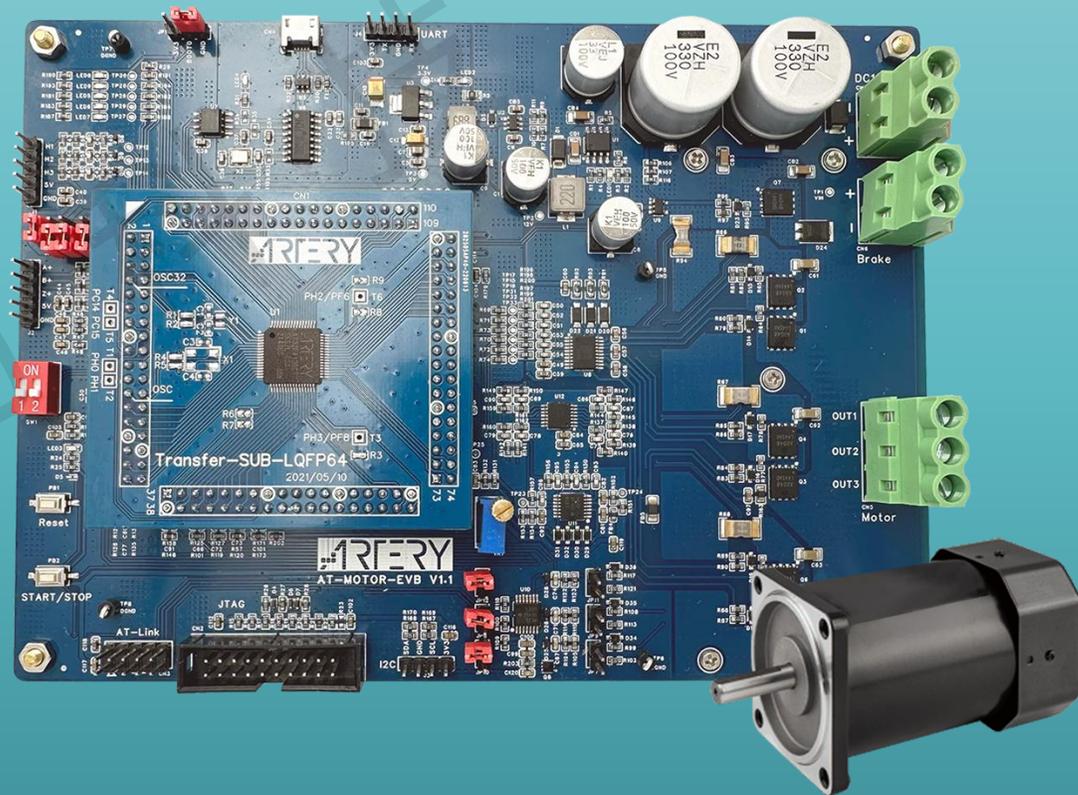
以太网应用

Quick Start Guide(快速入门指南)

简介	中文文档	英文(EN)文档	支持型号	版本	日期
AT32 BLDC Hall 快速入门指南 描述了如何快速使用AT-MOTOR-EVB电机开发板搭配AT32 BLDC电机库调适不同型号带霍尔传感器的电机运转。	AN0213	AN0213	AT32F413 AT32F421	V2.1.0	2024/05/17
AT32 BLDC Sensorless 快速入门指南 描述了如何快速使用AT-MOTOR-EVB电机开发板搭配AT32 BLDC电机库调适不同型号无传感器的电机运转。	AN0214	AN0214	AT32F421 AT32F413	V2.1.0	2024/05/17
AT32 PMSM FOC Incremental Encoder 快速入门指南 描述了如何快速使用AT-MOTOR-EVB电机开发板搭配AT32 PMSM FOC电机库调适不同型号带光电增量编码器的电机运转。	AN0217	/	AT32F403A AT32F413 AT32F415 AT32F421 AT32F423	V2.1.0	2024/05/24
AT32 PMSM FOC Hall 快速入门指南 描述了如何快速使用AT-MOTOR-EVB电机开发板搭配AT32 PMSM FOC电机库调适不同型号带霍尔传感器的电机运转。	AN0218	/	AT32F413 AT32F415 AT32F403A AT32F421 AT32F423	V2.1.0	2024/05/24
AT32 PMSM FOC Hall(E-Bike-Scooter) 快速入门指南 描述了如何快速使用AT-MOTOR-EVB电机开发板搭配AT32 PMSM FOC电机库调适不同型号带霍尔传感器的电机运转并应用在电动自行车、电瓶车、电动滑板车等应用。	AN0219	/	AT32F403A AT32F413 AT32F415 AT32F421 AT32F423	V2.1.0	2024/05/24
AT32 PMSM FOC Sensorless 快速入门指南 描述了如何快速使用AT-MOTOR-EVB电机开发板搭配AT32 PMSM FOC电机库调适不同型号无传感器的电机运转。	AN0220	/	AT32F403A AT32F413 AT32F415 AT32F421 AT32F423	V2.1.0	2024/05/24

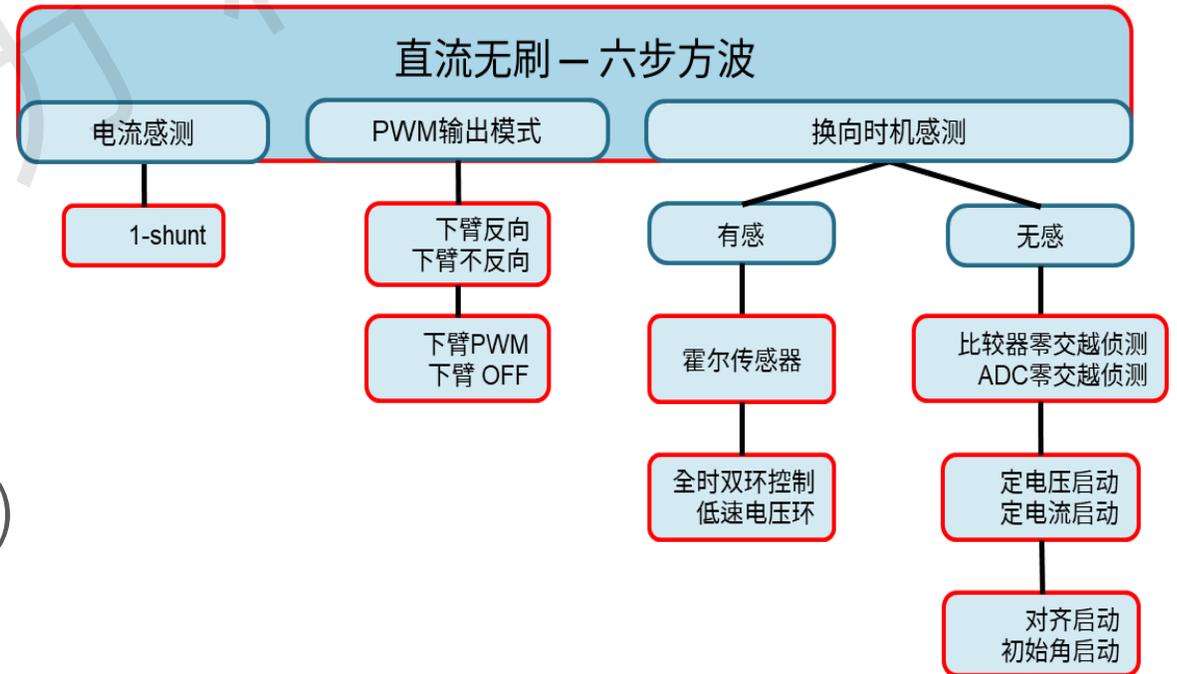
网址：https://www.arterytek.com/cn/support/motor_control.jsp?index=0

AT32 电机库 BLDC 快速上手 操作讲解



BLDC电机控制库支持

- 支援BLDC 6-step控制
- 提供AT32F421、AT32F413、AT32F415、AT32F425、AT32L021应用范例
- 支持免费AT32IDE编译环境
- 只须设定mc_hwio_v2(v1).h 和 motor_control_drive_param.h头文件
- 宏定变量的说明可参阅 AT32 电机库使用指南(AN0064)
- 霍尔专案的说明可参阅(AN0169)
- BLDC HALL快速入门指南(AN0213)

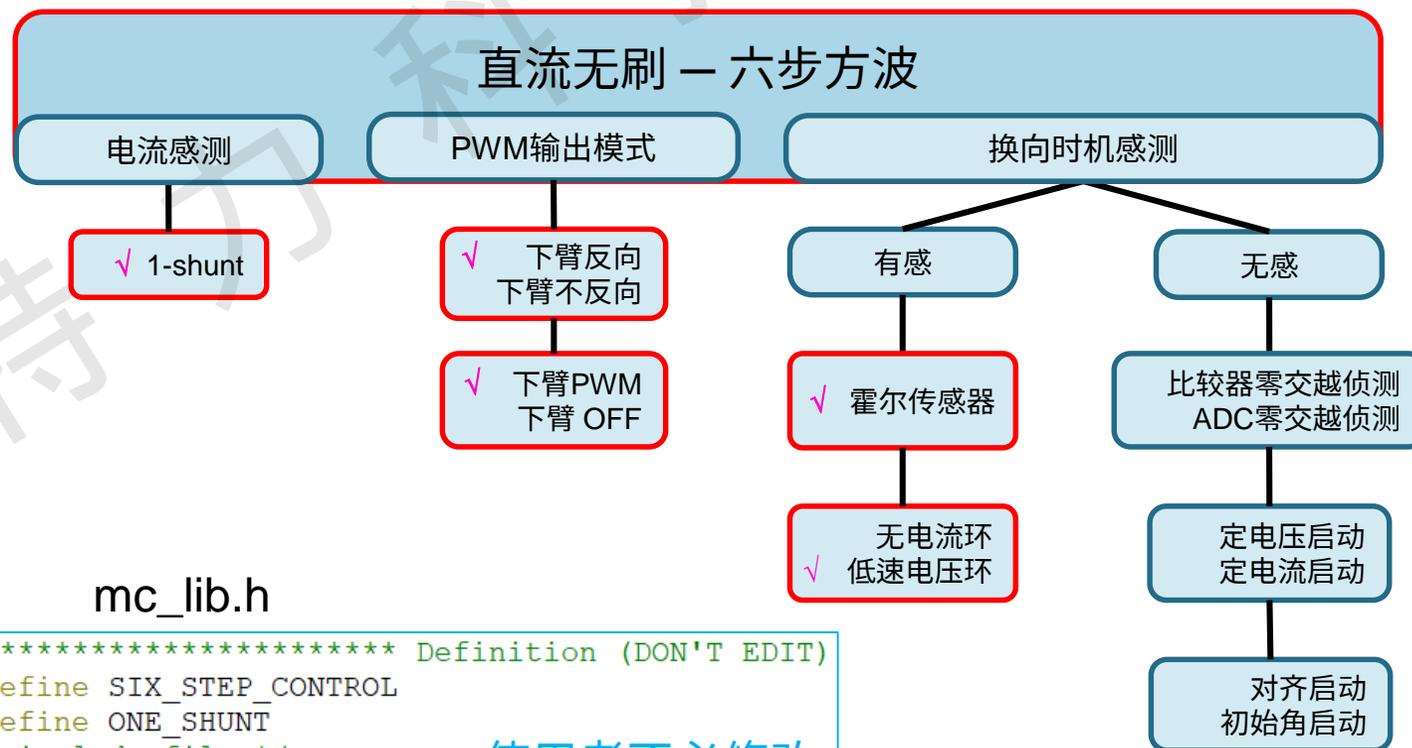


参数头文件 - 电机控制形式 (六步方波)

motor_control_drive_param.h
控制型式

```
/* internal clock or external crystal */  
//#define INTERNAL_CLOCK_SOURCE  
  
/* choose MOTOR_EVB_BOARD version */  
#define AT_MOTOR_EVB_V2  
//#define AT_MOTOR_EVB_V1  
  
/* mosfet low side complement (high side)  
#define COMPLEMENT  
/* gate driver low side inverting logic  
#define GATE_DRIVER_LOW_SIDE_INVERT  
  
/* choose sensor */  
#define HALL_SENSORS  
//#define SENSORLESS  
//#define BLDC_SENSORLESS_ADC  
//#define BLDC_SENSORLESS_COMP  
  
/* choose remove current-loop-control or  
//#define WITHOUT_CURRENT_CTRL  
  
/* choose low speed control or not */  
#define LOW_SPEED_VOLT_CTRL
```

● 选择宏定义宣告项目，完成电机控制形式设定



mc_lib.h

```
/* ***** Definition (DON'T EDIT) *****  
#define SIX_STEP_CONTROL  
#define ONE_SHUNT  
/* include file */  
#include <stdlib.h>
```

使用者不必修改

参数头文件 – 电机参数

motor_control_drive_param.h

电机参数 (Motor-related parameter)

```
/****** Motor-related parameter *****/
/* Motor parameters */
#define RS_LL          (1.89)          /* Stator resistance */
#define LS_LL          (0.002387)     /* Stator inductance */
#define POLE_PAIRS    (8/2)
#define KE             (0.003437f)    /* Back EMF constant */
#define NOMINAL_CURRENT (1.7)

/* angle detect duty */
#define ANGLE_INIT_DETECT_DUTY ((int16_t) (0.15*ANGLE_INIT_PERIOD))
#define ANGLE_INIT_I_DIFF      ((int16_t) 500)

/* hall learn table */
#if defined (SENSORLESS)
#define HALL_LEARN_DIR          (0)          /* Polarity, 0 or 1 */
#define HALL_LEARN_0_STATE     (2)          /* BH-CL */
#define HALL_LEARN_1_STATE     (3)          /* BH-AL */
#define HALL_LEARN_2_STATE     (1)          /* CH-AL */
#define HALL_LEARN_3_STATE     (5)          /* CH-BL */
#define HALL_LEARN_4_STATE     (4)          /* AH-BL */
#define HALL_LEARN_5_STATE     (6)          /* AH-CL */
#else
#define HALL_LEARN_DIR          (0)          /* Polarity, 0 or 1 */
#define HALL_LEARN_0_STATE     (1)          /* BH-CL */
#define HALL_LEARN_1_STATE     (5)          /* BH-AL */
#define HALL_LEARN_2_STATE     (4)          /* CH-AL */
#define HALL_LEARN_3_STATE     (6)          /* CH-BL */
#define HALL_LEARN_4_STATE     (2)          /* AH-BL */
#define HALL_LEARN_5_STATE     (3)          /* AH-CL */
#endif
#endif
```

- POLE_PAIRS : 电机极对数
- RS_LL : 绕组线对线电阻值
- LS_LL : 绕组线对线电感值
- NOMINAL_CURRENT : 额定电流
- HALL_LEARN_DIR : 霍尔自学习转向
- HALL_LEARN_X_STATE : 霍尔状态对应的矢量

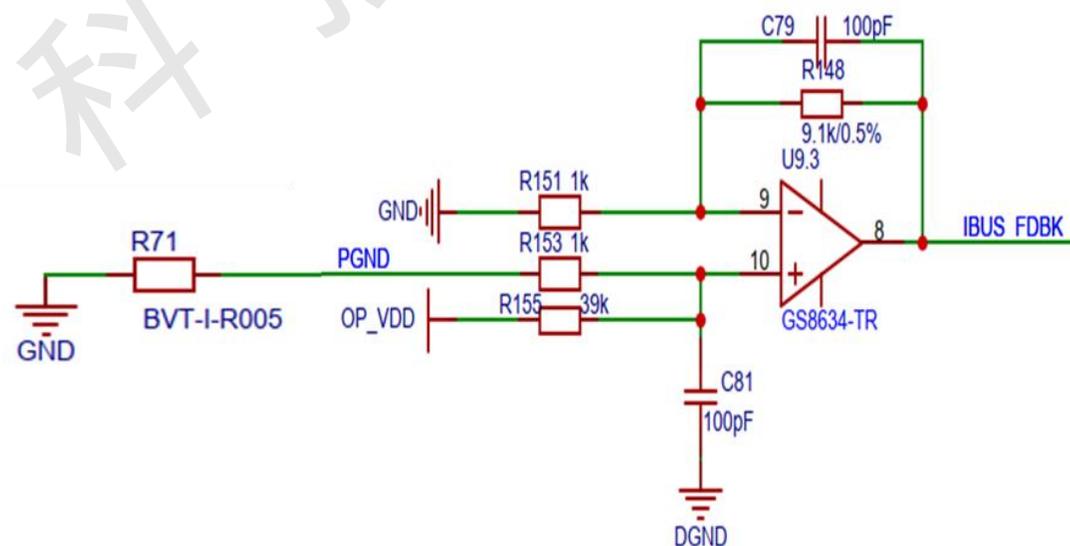
参数头文件 – 驱动器参数

motor_control_drive_param.h

驱动器参数 (Drive-related parameter)

```
123 /* basic */
124 #define VDC_RATED ((double)24.0)
125 #define BAT_LOW_VOLT ((double)12.0)
126 #define V_SENSE_GAIN (10.0f/(3.9f+180.0f+10.0f))
127 #define MAX_CURRENT (35.0f)
128 #define MIN_CURRENT (-MAX_CURRENT)
129 #define CURRENT_SPAN_SHIFT (1)
130 #define ADC_REFERENCE_VOLT (3.30)
131 #define ADC_DIGITAL_SCALE_12BITS (4095)
132 /* Clock */
133 #if defined AT32F413xx
134 #define SYSTEM_CORE_CLOCK (200000000)
135 #elif defined AT32F421xx
136 #define SYSTEM_CORE_CLOCK (120000000)
137 #endif
138 #define TMR_CLK (SYSTEM_CORE_CLOCK)
139 #define CHANGE_PHASE_TMR_DIV (TMR_CLK/2/1000000)
140 /* Dead-time inserted */
141 #define DEADTIME_CLK_SFT_BITS ((tmr_clock_division_type)2)
142 #define DEADTIME_NS ((uint16_t)300)
143 /* Current */
144 #if defined ONE_SHUNT
145 #define R_SHUNT (0.005f)
146 #define OP_GAIN (39.0f/(1.0f+39.0f)*(1.0f+9.1f))
147 #define CURR_OFFSET_VOLT (3.3f*(1.0f/(1.0f+39.0f)*(1.0f+9.1f)))
```

- 根据驱动器电路，设定相关电气参数



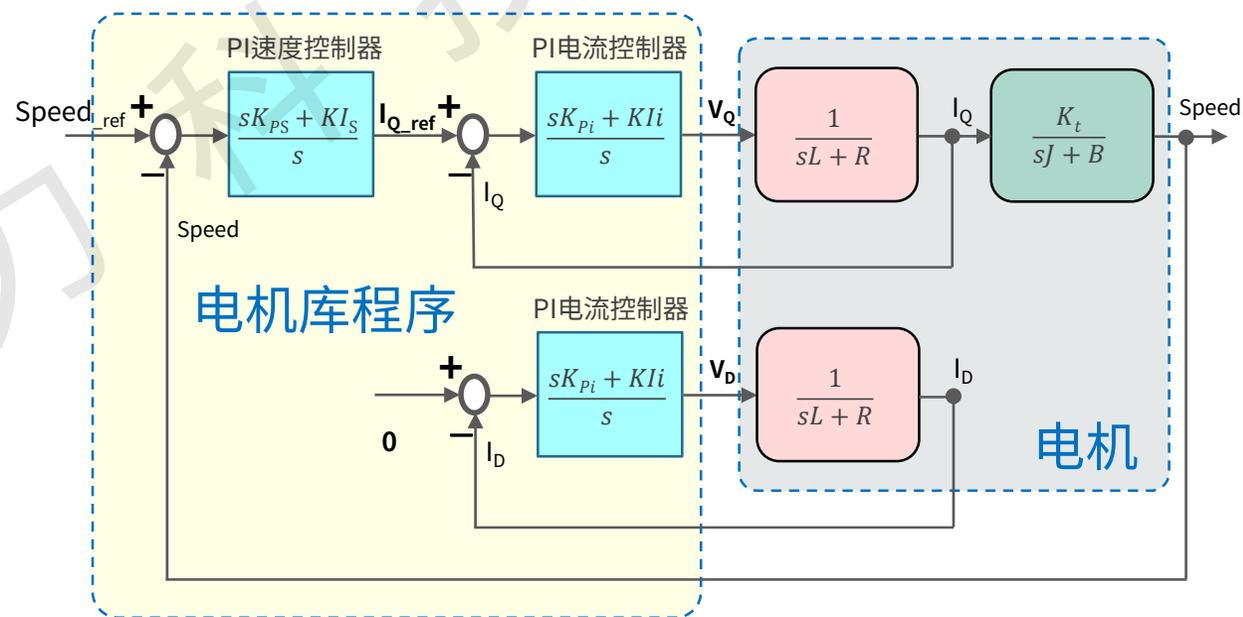
相电流回授
参数设定

母线电流回授电路

参数头文件 – 控制环相关参数

motor_control_drive_param.h
控制参数 (Control-related parameter)

```
#define PWM_FREQ (15000) /*!< Hz */
#define MOTOR_CONTROL_MODE (motor_control_mode)(SPEED_CTRL)
#define CTRL_SOURCE (ctrl_source_type)(CTRL_SOURCE_SOFTWARE)
#define UI_UART_BAUDRATE_ (1500000UL) /*!< bit/s */
/* I-SAMPLE PARAMETER */
#define I_SAMP_MIN_TIME (180) /*!< nsec */
#define I_SAMP_DELAY (1000) /*!< nsec */
/* SPEED */
#define SPEED_FILTER_TIMES (6)
#define SPD_LP_BANDWIDTH (300.0f) /* 2*pi*freq */
#define PWM_IN_FILTER_TIMES (5)
#define SPD_CMD_FILTER_TIMES (5)
/* SPEED PARAMETER */
#define MIN_SPEED_RPM (10)
#define MAX_SPEED_RPM (7200) /*!< rpm */
#define MAX_CCW_SPEED_RPM (7200) /*!< rpm */
#define ACC_SPD_SLOPE (5) /*!< rpm/ms */
#define DEC_SPD_SLOPE (5)
/* pi parameter */
#define PID_IS_KP_DEFAULT 6000
#define PID_IS_KI_DEFAULT 11300
#define PID_IS_KP_DIV 1024
#define PID_IS_KP_DIV_LOG LOG2(PID_IS_KP_DIV)
#define PID_IS_KI_DIV 32768
#define PID_IS_KI_DIV_LOG LOG2(PID_IS_KI_DIV)
```



$$IS_Kp = PID_IS_KP_DEFAULT / PID_IS_KP_DIV$$
$$IS_Ki = PID_IS_KI_DEFAULT / PID_IS_KI_DIV$$

外设定义头文件 - 外设参数定义

外设定义参数 (mc_hwio.h)

```

/* adc reading pin definition */
#define ADC_CONVERTER ADC1
#define ADC_CONVERTER_CRM_CLK CRM_ADC1_PERIPH_CLOCK
#define ADC_CONVERTER_CRM_CLK_DIV CRM_ADC_DIV_8
#define ADC_SHUNT_SAMP_READY_IRQ ADC1_2_IRQHandler
#define ADC_SHUNT_SAMP_READY_IRQn ADC1_2_IRQn
#define ADC_ORDINARY_CH_LEN 6

/* dma1 ch1 for adc ordinary conversion */
#define ADC_ORDINARY_DMA_CRM_CLK CRM_DMA1_PERIPH_CLOCK
#define ADC_ORDINARY_DMA_CHANNEL DMA1_CHANNEL1
#define ADC_ORDINARY_DMA DMA1
#define ADC_ORDINARY_DMA_FLEX DMA_FLEXIBLE_ADC1
#define ADC_ORDINARY_DMA_FLEX_CH FLEX_CHANNEL1

#define CURR_PHASE_A_ADC_CH ADC_CHANNEL_0
#define CURR_PHASE_A_ADC_GPIO_CRM_CLK CRM_GPIOA_PERIPH_CLOCK
#define CURR_PHASE_A_ADC_PORT GPIOA
#define CURR_PHASE_A_ADC_GPIO_PIN GPIO_PINS_0

#define CURR_PHASE_B_ADC_CH ADC_CHANNEL_1
#define CURR_PHASE_B_ADC_GPIO_CRM_CLK CRM_GPIOA_PERIPH_CLOCK
#define CURR_PHASE_B_ADC_PORT GPIOA
#define CURR_PHASE_B_ADC_GPIO_PIN GPIO_PINS_1

#define CURR_PHASE_C_ADC_CH ADC_CHANNEL_2
#define CURR_PHASE_C_ADC_GPIO_CRM_CLK CRM_GPIOA_PERIPH_CLOCK
#define CURR_PHASE_C_ADC_PORT GPIOA
#define CURR_PHASE_C_ADC_GPIO_PIN GPIO_PINS_2
    
```

控制器MCU pin map

Peripheral function	AT32F421C8T6		AT32F413RCT7	
VBAT	1	VBAT	1	VBAT
STATUS_LED1	2	PC13	2	PC13
STATUS_LED2	3	PC14	3	PC14
STATUS_LED3	4	PC15	4	PC15
ENCODER_A+			18	TMR5_CH1(PF4)
ENCODER_B+			19	TMR5_CH2(PF5)
MODE_SW1	35	PF6	47	PF6
MODE_SW2	36	PF7	48	PF7
OSC_IN	5	OSC_IN	5	OSC_IN
OSC_OUT	6	OSC_OUT	6	OSC_OUT
RESET	7	NRST	7	NRST
SPEED_VR(F413)			8	ADC12_IN10(PC0)
			9	ADC12_IN11(PC1)
			10	ADC12_IN12(PC2)
			11	ADC12_IN13(PC3)
VSSA	8	VSSA	12	VSSA
VDDA	9	VDDA	13	VDDA
CURR_FDBK1	10	ADC_IN0(PA0)	14	ADC_IN0(PA0)
CURR_FDBK2	11	ADC_IN1(PA1)	15	ADC_IN1(PA1)
CURR_FDBK3	12	ADC_IN2(PA2)	16	ADC_IN2(PA2)
IBUS_FDBK	13	ADC_IN3(PA3)	17	ADC_IN3(PA3)
BEMF1_LF	14	ADC_IN4(PA4)	20	ADC_IN4(PA4)
BEMF2_LF	15	ADC_IN5(PA5)	21	ADC_IN5(PA5)
BEMF3_LF	16	ADC_IN6(PA6)	22	ADC_IN6(PA6)
VBUS	17	ADC_IN7(PA7)	23	ADC_IN7(PA7)
ENCODER_Z+			24	PC4
			25	ADC_IN15(PC5)

UI程序与电机控制调校

定义头档相关参数

I/O mapping

Board param.

Motor param.

Control param.

Drive mode

- FOC_CONTROL
- ONE_SHUNT
- SENSORLESS

电机控制程序

Motor control project

UI程序调校PID控制参数

a. 在线调整控制参数

b. 实时观察输出响应

c. 控制参数写入Flash

实时监控看响应特性与数据

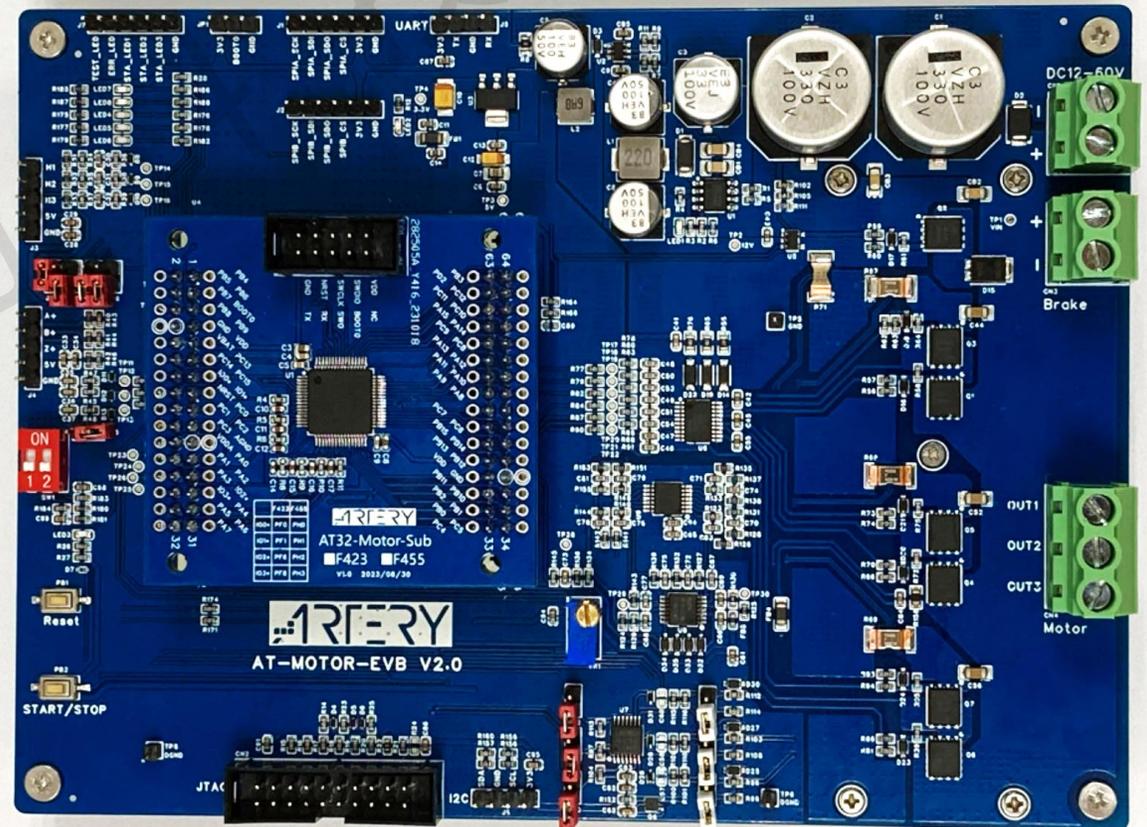
实时监控输出响应

参数与功能

低压电机方案板

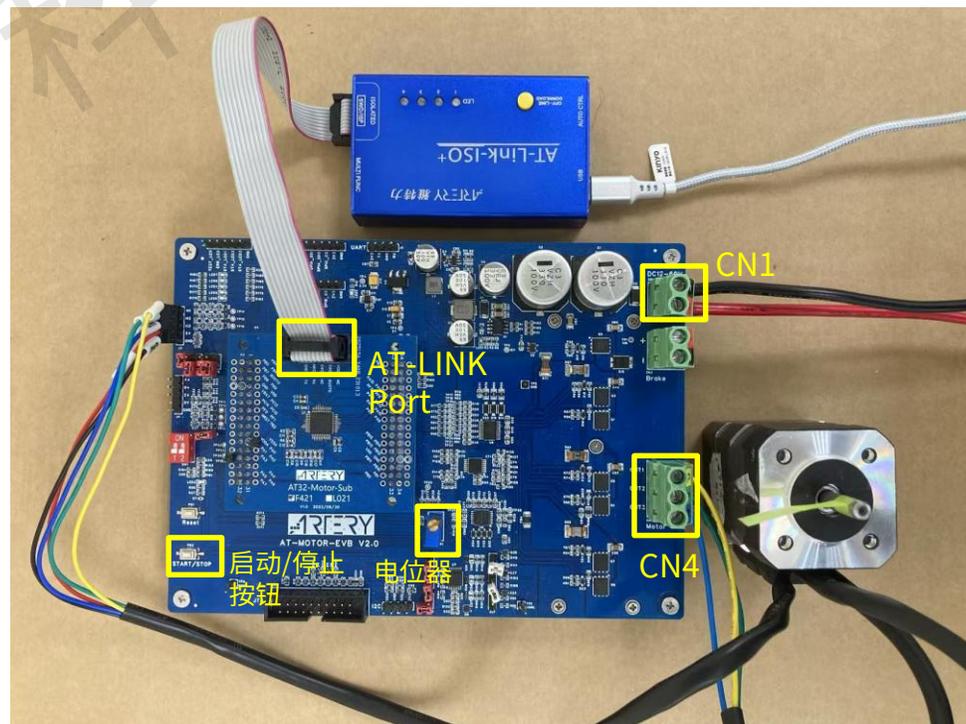
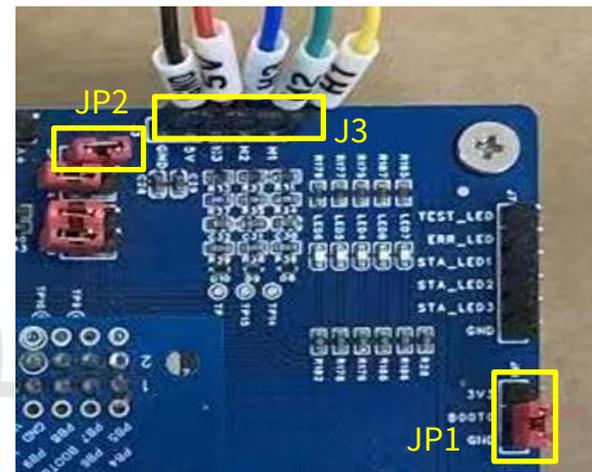
- ★ 泛用型的低压三相电机驱动器
- ★ 输入电压：12V~60V，最大输出电流：30_{APEAK}
- ★ 内建12V与5V降压转换器以及3.3VLDO稳压电路
- ★ 具备一个微控制器转接插座，适用多系列AT32微控制器
- ★ 提供增量编码器界面与霍尔信号界面回授转子位置
- ★ 可应用三电阻、两电阻电流感测，以及单电阻电流感测方式
- ★ 内建相电流与母线电流的过电流比较电路
- ★ 可执行有位置传感器/无位置传感器FOC弦波矢量控制
- ★ 具备三相输出端电压感测回授，以及虚拟中性点比较电路
- ★ 可执行有位置传感器/无位置传感器120°方波BLDC电机控制
- ★ 具备回生刹车电阻界面，可实现快速动态刹车功能

- ★ 输入接口：一个电位器、2个指拨开关与一个按键开关
- ★ 输出指示：4个可控LED指示灯、1个故障LED指示灯
- ★ 具备UART与I²C、SPI通信接口
- ★ 具备AT-Link与20-pin JTAG烧录界面



控制器接线

- 将霍尔线连接到J3 (依线标)
- 检查JP1 Boot0是否接地
- 检查JP2是否短接
- 将AT-LINK连接到小板
- 连接电机线到CN4 (依线色)
- 连接电源供应器电源接线到CN4 (依标示)
- 设定电源供应器电压24V / 3A
- 开启电源供应器输出



电机与开发板接线图

监控界面软件基本操作 - 控制与监看

Artery Motor Monitor V2.1.18

文件 设置 帮助 联机

COM37: ATLink-USART 关闭 绘图

AT32F421_ML_V2.1.3

程序状态

ESC_STATE_SAFETY_READY

no error
 over voltage error
 under voltage error
 over temperature error
 over current error
 encoder error
 hall error
 parameter identify error
 hall learn error
 startup error

错误状态

命令 启动/停止电机

启动电机 停止电机

编码器对齐

错误解除 错误状态解除

写入闪存 写入闪存

BLDC - hall 控制型式 (read)

Speed Control 控制模式

控制源

software control 控制来源

基本 参数调整 自定义功能 监看运行状态 (read)

Power board status

Bus Voltage measured	23.087	volt
Mos temperature measured	20	°C

Speed

Maximum application speed	7200	rpm
Minimum application speed	10	rpm
Speed measured	0	rpm

Diagram parameter setting

Bus Voltage measured

User defined A

保存

选择绘图变量

Target speed

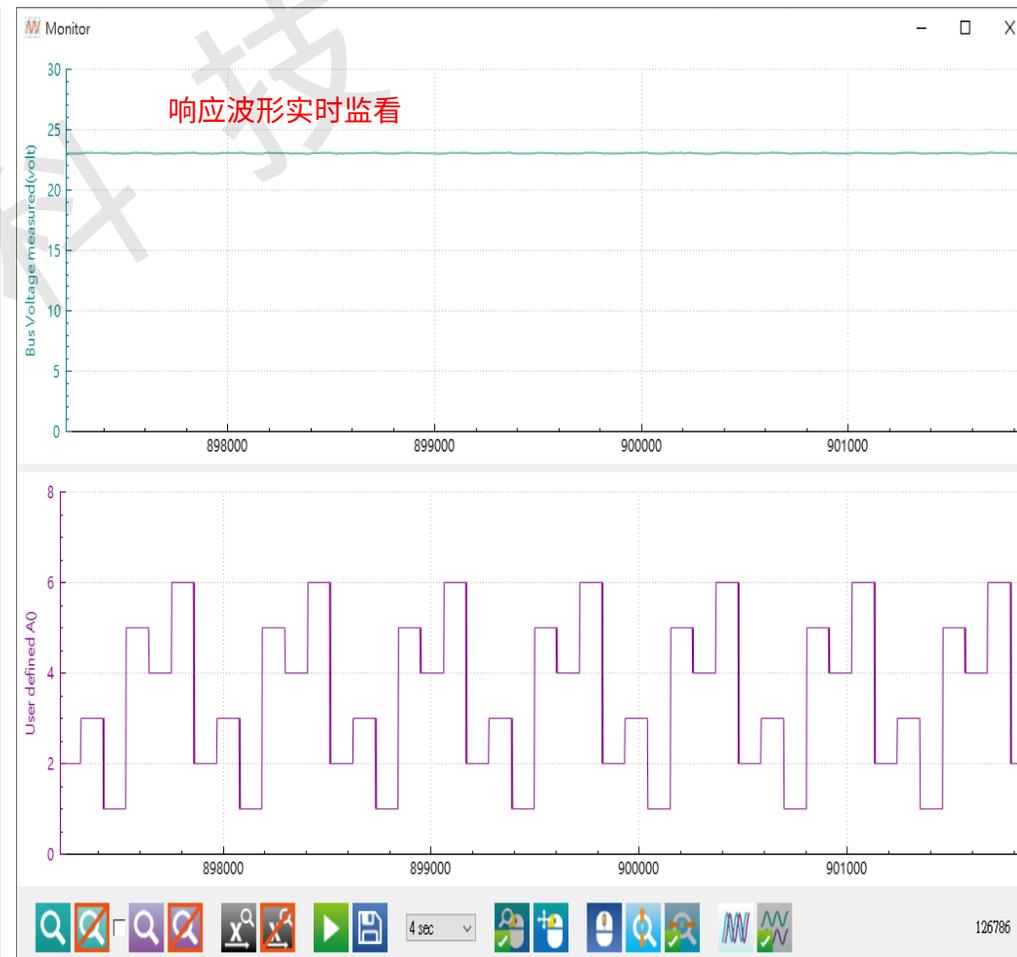
Speed reference	0	rpm
-----------------	---	-----

运行目标

运行日志

1	21:09:47	Load ok
2	21:09:47	Set REG 8 = 5 ok
3	21:09:45	Exec Stop Motor...ok
4	21:06:25	Load ok

传输状态记录



监控界面软件运行操作 - 开环控制

The screenshot shows the Artery Motor Monitor V2.1.18 interface. Key components include:

- Program Status (程序状态):** ESC_STATE_SAFETY_READY.
- Error Status (错误状态):** A list of error types such as 'no error', 'over voltage error', etc., with checkboxes.
- Open Loop Control Parameters (开环控制参数):**

OpenLoop initial voltage	0.728	V
OpenLoop initial speed	100	rpm
OpenLoop increase volt	0.004	V
OpenLoop final speed	350	rpm
OpenLoop times	90	
- Speed Voltage Controller (Without Current control loop):**

Speed Voltage KP	3000
Speed Voltage KI	100
Speed Voltage KP DIV	1024
Speed Voltage KI DIV	1024
- Graph:** A graph showing Voltage (V) vs Speed (rpm). It illustrates a step-wise increase in voltage from an initial value to a final value, with 5 steps labeled 1 to 5. The time between steps is 5ms. The final speed is 350 rpm.
- Command (启动/停止电机):** Buttons for '启动电机' (Start Motor) and '停止电机' (Stop Motor).
- Control Mode (控制模式):** A dropdown menu currently set to 'BLDC - hall 控制型式 (read)'.
- Run Log (运行日志):** A table of system events:

1	20:44:31	Load ok
2	20:44:31	Set REG 8 = 0 ok
3	20:44:28	Exec Stop Motor...ok
4	20:39:58	Set REG 10 = 0 ok



```

mc_isr.c
98 ui_wave_param.user_define_a = (int16_t)hall.state;
99 ui_wave_param.user_define_b = (int16_t)rotor_speed.val;
    
```

使用者可自行将欲观察的变量指定给 user_define_a / user_define_b 便可于上位机中观察

监控界面软件运行操作 - 自整定功能

The screenshot displays the Artery Motor Monitor V2.1.18 software interface. The window title is "Artery Motor Monitor V2.1.18" and it includes a menu bar with "文件", "设置", and "帮助". The main interface is divided into several sections:

- COM37: ATLink-USART**: Shows the connection port and a "关闭" button.
- 程序状态 (ESC_STATE_SAFETY_READY)**: A list of error status indicators, including "no error", "over voltage error", "under voltage error", "over temperature error", "over current error", "encoder error", "hall error", "parameter identify error", "hall learn error", and "startup error".
- 霍尔自学习 (Hall phase sequence self-learning)**: A table showing the learning process for Hall phase sequence.
- 线圈参数识别 (Winding parameter Identification)**: A table showing identified process state and winding parameters.
- 电流环 PI参数自整定 (Auto-tune PI parameter of Current controller)**: A table showing torque control parameters.
- 启动/停止电机 (命令)**: Buttons for "启动电机" and "停止电机".
- 错误解除 (错误状态解除)**: A button to clear error status.
- 写入闪存 (写闪存)**: A button to write to flash memory.
- 控制型式 (read)**: A dropdown menu showing "BLDC - hall".
- 控制模式**: A dropdown menu showing "Speed Control".
- 控制来源**: A dropdown menu showing "software control".
- 运行日志 (传输状态记录)**: A table showing the execution log.

HALL_LEARN_PROCESS_STATE	PROCESS_0_LOCK
HALL_LEARN_DIR	0
HALL_LEARN_0_STATE	1
HALL_LEARN_1_STATE	5
HALL_LEARN_2_STATE	4
HALL_LEARN_3_STATE	6
HALL_LEARN_4_STATE	2
HALL_LEARN_5_STATE	3

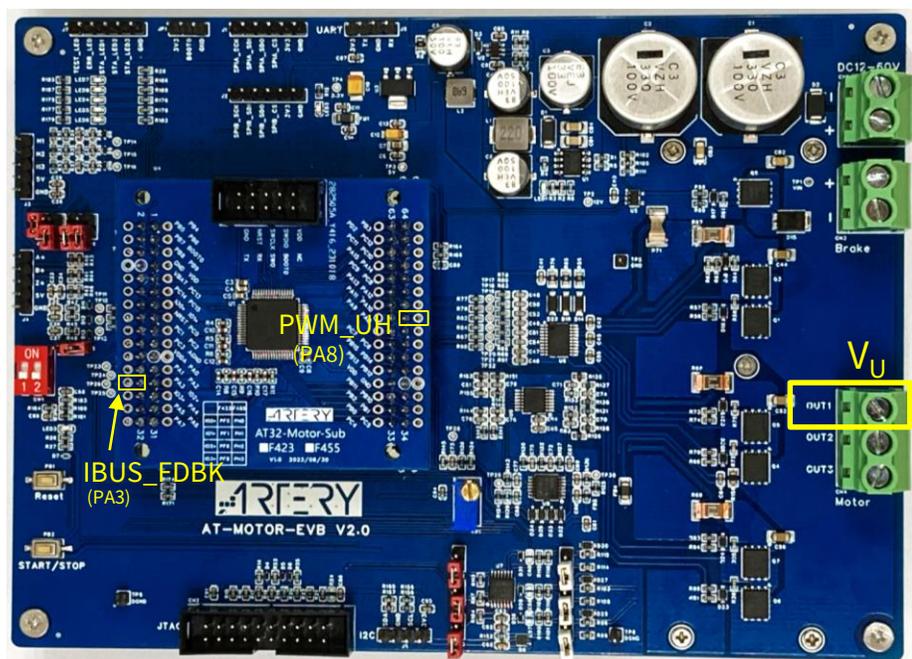
Identified process state	尚未运行	
RS_LL	1.89	Ω
LS_LL	0.002387	H

Parameter	Value	Label
Torque KP	6000	电流环 PI参数自整定
Torque KI	11300	
Torque KP DIV	1024	
Torque KI DIV	32768	

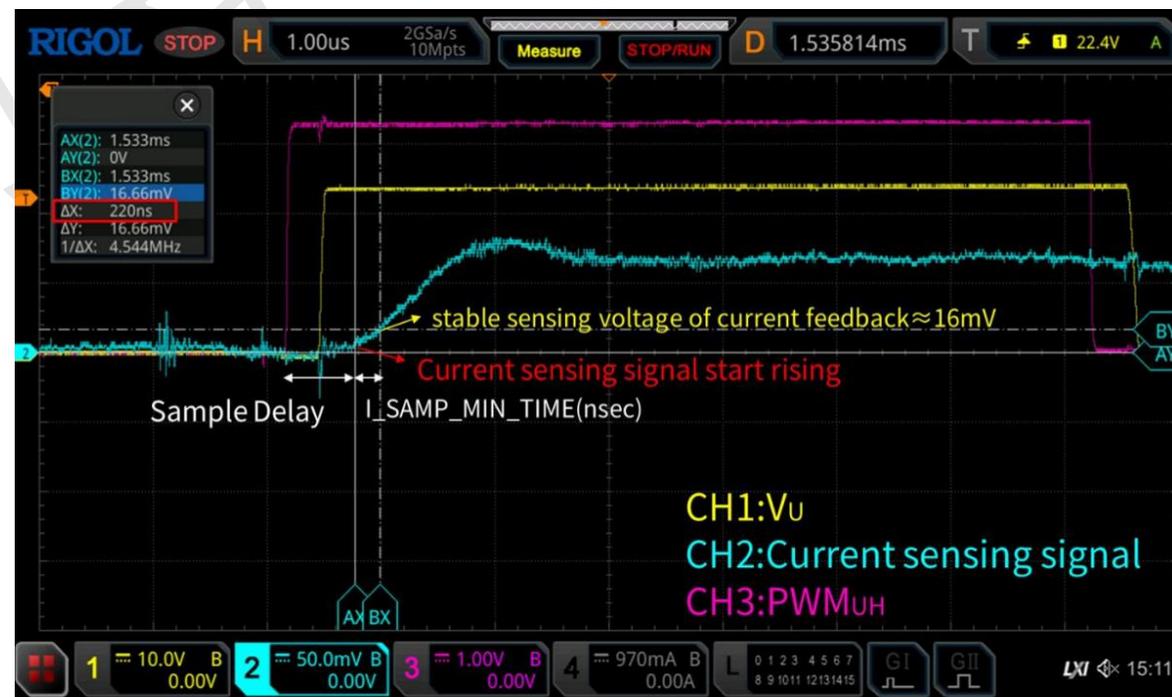
Index	Time	Event
1	21:09:47	Load ok
2	21:09:47	Set REG 8 = 5 ok
3	21:09:45	Exec Stop Motor...ok
4	21:06:25	Load ok

设定电流取样点

- 因预驱与MOSFET作动有时间延迟，故须设定电流取样点
- 量测I_SAMP_DELAY与I_SAMP_MIN_TIME时间



电流取样延迟时间测量点



电流取样延迟时间测量波形

监控界面软件运行操作 - 电流环参数实时调整

The screenshot displays the Artery Motor Monitor V2.1.18 software interface, which is used for monitoring and adjusting motor parameters. The interface is divided into several sections:

- Program Status (程序状态):** Shows the current state as `ESC_STATE_SAFETY_READY`.
- Error Status (错误状态):** A list of error conditions, all of which are currently unchecked, indicating no errors.
- Command (命令):** Includes buttons for starting and stopping the motor, as well as encoder alignment.
- Error Clearance (错误状态解除):** A button to clear any error states.
- Flash Write (写入闪存):** A button to write data to the motor's flash memory.
- Control Mode (控制模式):** Set to `BLDC - hall`.
- Control Mode (控制模式):** Set to `IQ Tune`.
- Unit step config (步阶电流命令参数):** Parameters for step current commands:

Current Tune target current	0.500	A
Current Tune total period	50	ms
Current Tune step period	5	ms
- Current Controller PI Parameters (电流控制器 PI 参数):** Parameters for the current controller:

Torque KP	6000
Torque KI	11300
Torque KP DIV	1024
Torque KI DIV	32768
- Speed Voltage Controller (Speed Voltage Controller(Without Current control loop)):** Parameters for the speed voltage controller:

Speed Voltage KP	3000
Speed Voltage KI	100
Speed Voltage KP DIV	1024
Speed Voltage KI DIV	1024
- Run Log (运行日志):** A table recording system events:

1	19:59:52	Load ok
2	19:59:52	Set REG 8 = 3 ok
3	19:56:11	Load ok
4	19:56:11	Get Board Information ok
- Real-time Monitoring (实时监看):** A graph showing the current loop response waveform. The x-axis represents time (ms) from 916800 to 917100. The y-axis represents Torque(current) reference (Iq)(A) and Torque(current) measured (Iq)(A) from -0.5 to 1.5. The graph shows a step response where the current reference (green line) steps up to 1.0 A, and the measured current (purple line) follows it with a slight overshoot and settling time.

监控界面软件运行操作 - 电流环参数实时调整

The screenshot displays the Artery Motor Monitor V2.1.18 software interface, which is used for monitoring and controlling a motor. The interface is divided into several sections:

- Program Status (程序状态):** Shows the current state as `ESC_STATE_SAFETY_READY`.
- Error Status (错误状态):** A list of error conditions, all of which are currently unchecked, indicating no errors.
- Command (命令):** Buttons for starting and stopping the motor.
- Error Clearance (错误解除):** Buttons for clearing error states and writing to flash.
- Control Mode (控制模式):** A dropdown menu currently set to `BLDC - hall`.
- Unit step config (步阶电流命令参数):** A table for configuring current step parameters:

Parameter	Value	Unit
Current Tune target current	0.500	A
Current Tune total period	50	ms
Current Tune step period	5	ms
- Current Controller PI Parameters (电流控制器 PI 参数):** A table for configuring the current controller:

Parameter	Value
Torque KP	6000
Torque KI	11300
Torque KP DIV	1024
Torque KI DIV	32768
- Speed Voltage Controller (Without Current control loop):** A table for configuring the speed voltage controller:

Parameter	Value
Speed Voltage KP	3000
Speed Voltage KI	100
Speed Voltage KP DIV	1024
Speed Voltage KI DIV	1024
- Run Log (运行日志):** A table recording system events:

Index	Time	Event
1	19:59:52	Load ok
2	19:59:52	Set REG 8 = 3 ok
3	19:56:11	Load ok
4	19:56:11	Get Board Information ok
- Monitor (电流环调适响应波形实时监看):** A real-time graph showing the current response. The x-axis represents time (around 91600-91700) and the y-axis represents current (Torque(current) reference (i(q)(A)) and Torque(current) measured (i(q)(A))). The graph shows a step change in current reference, with the measured current following it closely.

监控界面软件运行操作 - 速度环参数实时调整

Artery Motor Monitor V2.1.18

文件 设置 帮助

COM37: ATLink-USART 关闭 绘图

AT32F421_ML_V2.1.3

程序状态

ESC_STATE_SAFETY_READY

no error
 over voltage error
 under voltage error
 over temperature error
 over current error
 encoder error
 hall error
 parameter identify error
 hall learn error
 startup error

错误状态

启动/停止电机

启动电机 停止电机

编码器对齐

错误解除 错误状态解除
写入闪存 写入闪存

BLDC - hall 控制型式 (read)
Speed Control 控制模式
控制源
software control 控制模式

基本 参数调整 自整定功能

Speed control(With current control loop)

Speed KP	1000
Speed KI	20
Speed KP DIV	4096
Speed KI DIV	32768
Speed acceleration	5 rpm/ms
Speed deceleration	5 rpm/ms

有电流环速度控制 PI参数

Speed Voltage Controller(Without Current control loop)

Speed Voltage KP	3000
Speed Voltage KI	100
Speed Voltage KP DIV	1024
Speed Voltage KI DIV	1024

无电流环速度控制 PI参数

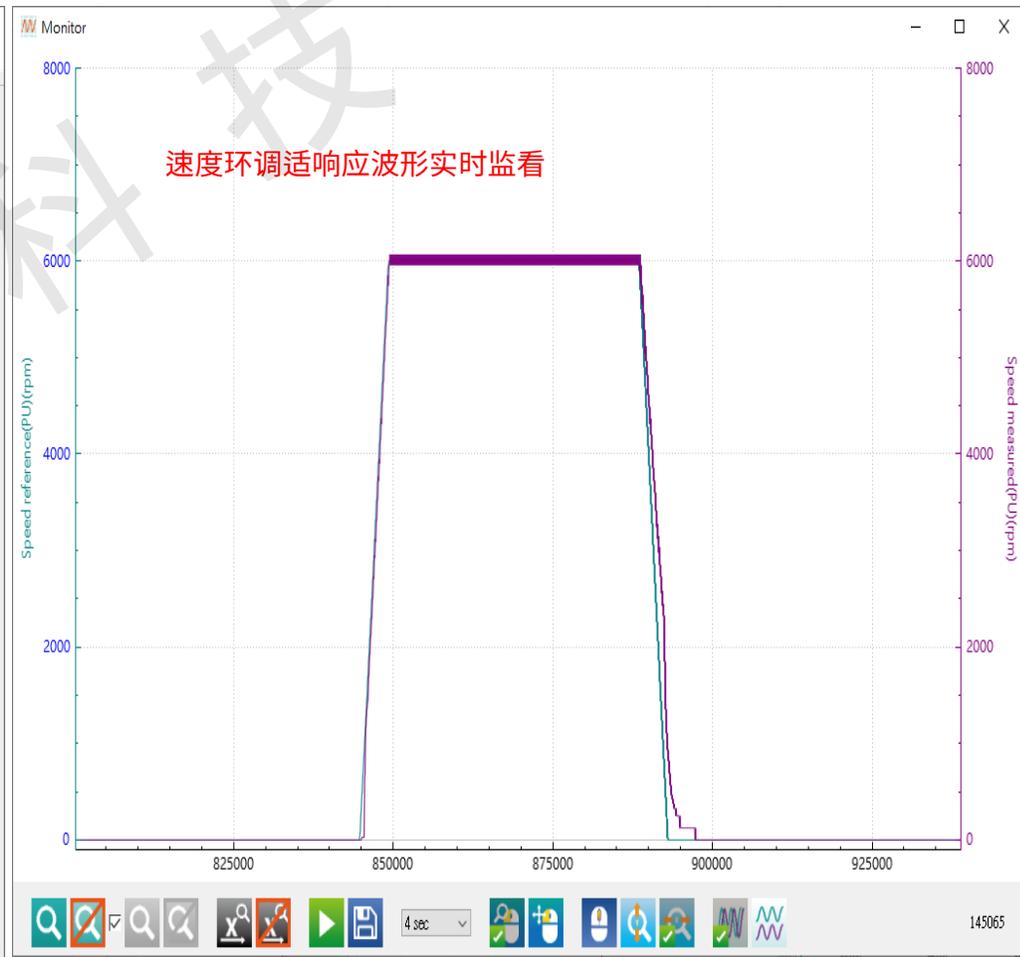
Target speed

Speed reference 0 rpm 运行目标

运行日志

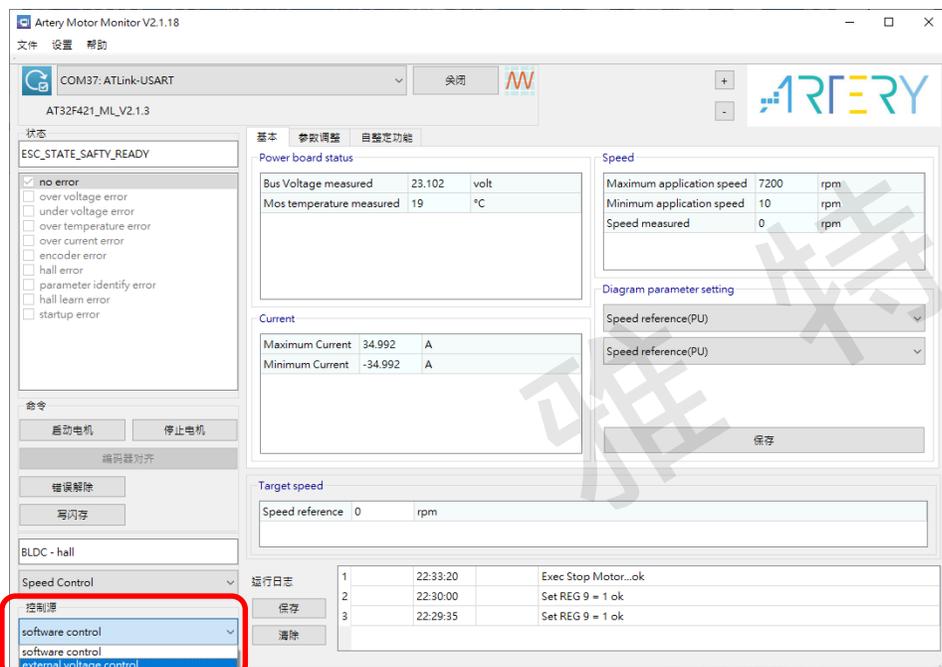
1	20:10:09	Load ok
2	20:10:09	Set REG 8 = 5 ok
3	20:10:05	Load ok
4	20:10:05	Set REG 8 = 4 ok

传输状态记录



外部命令控制

- 在控制源选择外部电压控制，SW1-1可设定转向
- 或在motor_control_drive_param.h头文件中定义'CTRL_SOURCE'



外部控制源



提问时间

10分钟

Q & A

ARTERY 联系我们



重庆办公室

重庆市九龙坡区科城路60号康田西锦荟1栋10F
+86-23-6868 8899

苏州办公室

苏州市工业园区通园路699号港华大厦1501
+86-0512-6835 6375

上海办公室

上海市长宁区仙霞路317号B栋512室
+86-0512-6835 6375

深圳办公室

深圳市福田区天安数码城创新科技广场二期西座603室
+86-0755-8390 0669 分机62568

台湾办公室

新竹市金山八街1号五楼
+886-3-577 8788



中文官网



微信公众号



English Website

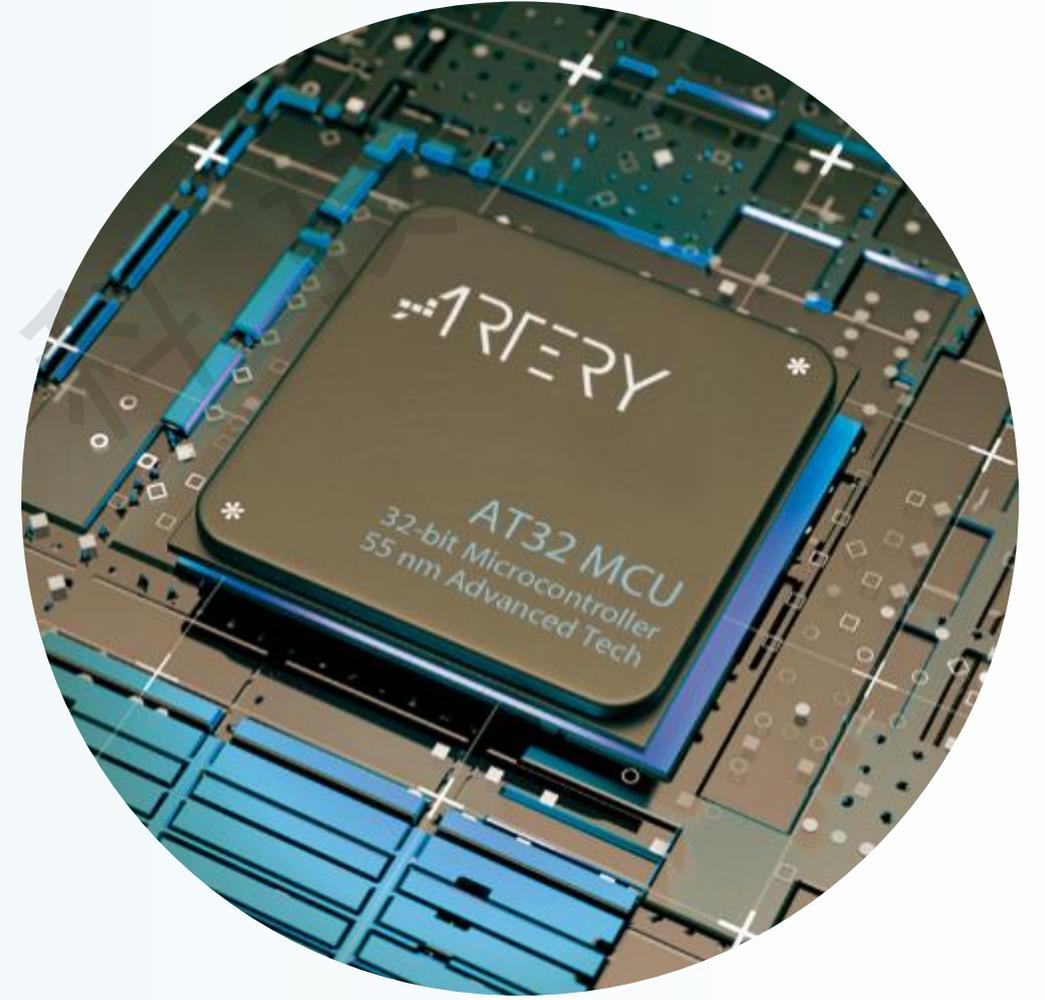


Linkedin

销售咨询 sales_artery@arterytek.com

技术服务 support_artery@arterytek.com

Thank you!



雅特力