

## 使用指南

# 电动自行车驱动器固件使用指南

## 简介

此文档的目的在于让使用者能够快速熟悉基于N32G435CBL7 电动自行车电机驱动器的固件程序，了解电机驱动器软件固件的基本功能、使用说明及注意事项，以使用户快速开发调试。

## 目录

<b>1</b>	<b>固件开发说明.....</b>	<b>1</b>
1.1	FOC 软件架框.....	1
1.2	硬件驱动库.....	2
1.3	APP 模块 .....	4
1.4	FOC 模块 .....	8
1.5	LIB 模块 .....	11
1.6	有感霍尔介绍.....	12
<b>2</b>	<b>电机调试参数.....</b>	<b>12</b>
2.1	电机驱动板相关的参数.....	12
<b>3</b>	<b>整机调试示例.....</b>	<b>13</b>
3.1	配置电机参数.....	14
3.2	配置驱动板参数.....	14
3.3	配置运行模式.....	15
3.4	启动电机调试.....	16
<b>4</b>	<b>历史版本.....</b>	<b>17</b>
<b>5</b>	<b>声 明.....</b>	<b>18</b>

## 1 固件开发说明

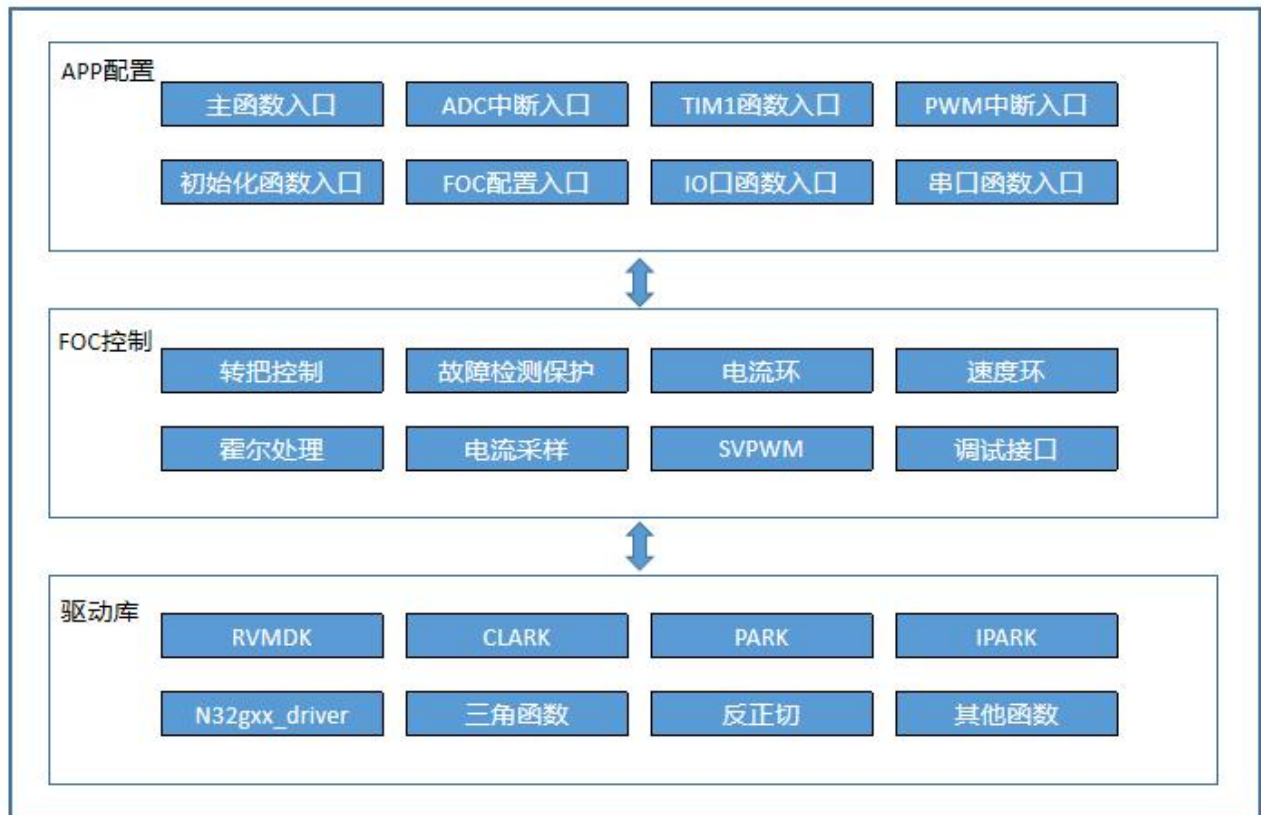
基于国民技术通用 MCU 芯片 N32G43x 系列进行电动车自行车驱动器开发。本电机控制固件采用的控制芯片型号为 N32G435CBL7，支持单电机 FOC 矢量控制应用开发。根据具体型号不同，有如下注意事项。

本固件支持评估电动自行车的电机控制算法基于有感霍尔 FOC 应用，支持的控制模式有转矩控制和速度控制。

### 1.1 FOC 软件架框

FOC 软件框架如下图所示。

图 1-1 基本软件框架



整个软件框架包含3 个部分，驱动库、APP配置、FOC控制，采用MDK 5.0及以上版本进行开发。对应的源码工程文件如下图所示。

### 驱动库:

主要是通用 MCU 相关的硬件驱动代码，完成寄存器级的配置封装，由标准 SDK 提供。

### APP:

由主函数、初始化配置函数、中断函数、IO口函数、串口通信函数等部分构成，提供 FOC 运行的系统环境和基本配置。

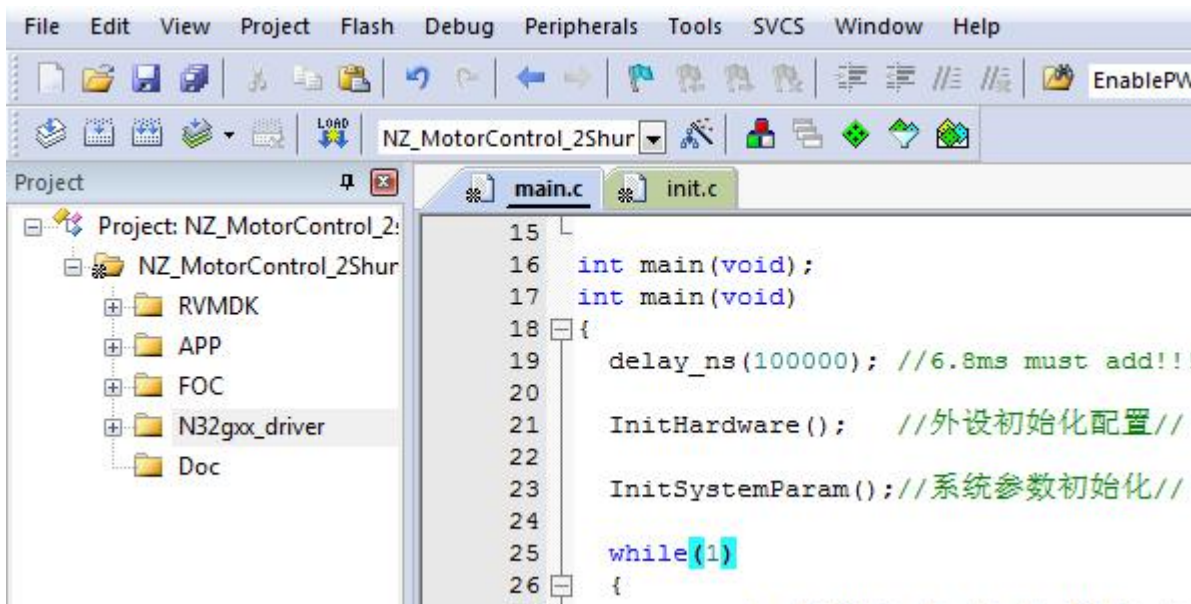
### FOC:

提供和 FOC 控制相关的功能模块，给客户提供了配置，优化，改进的接口，客户可以其于此电机固件框架，增加相应的功能模块，并快速迭代开发验证和量产评估。

### LIB:

提供相应核心的函数，功能模块库文件和相应的 MCU 平台绑定，用户可以直接调用。

图 1-2 软件固件工程示例



## 1.2 驱动库

驱动库是基于国民技术股份有限公司公开发行的相应芯片的开发套件Pack 包，该Pack 包集成了对应 MCU 的型号，以支持在 MDK 中选择对应的 MCU 型号，并支持在线调试时，提供图形化的外设操作界面。

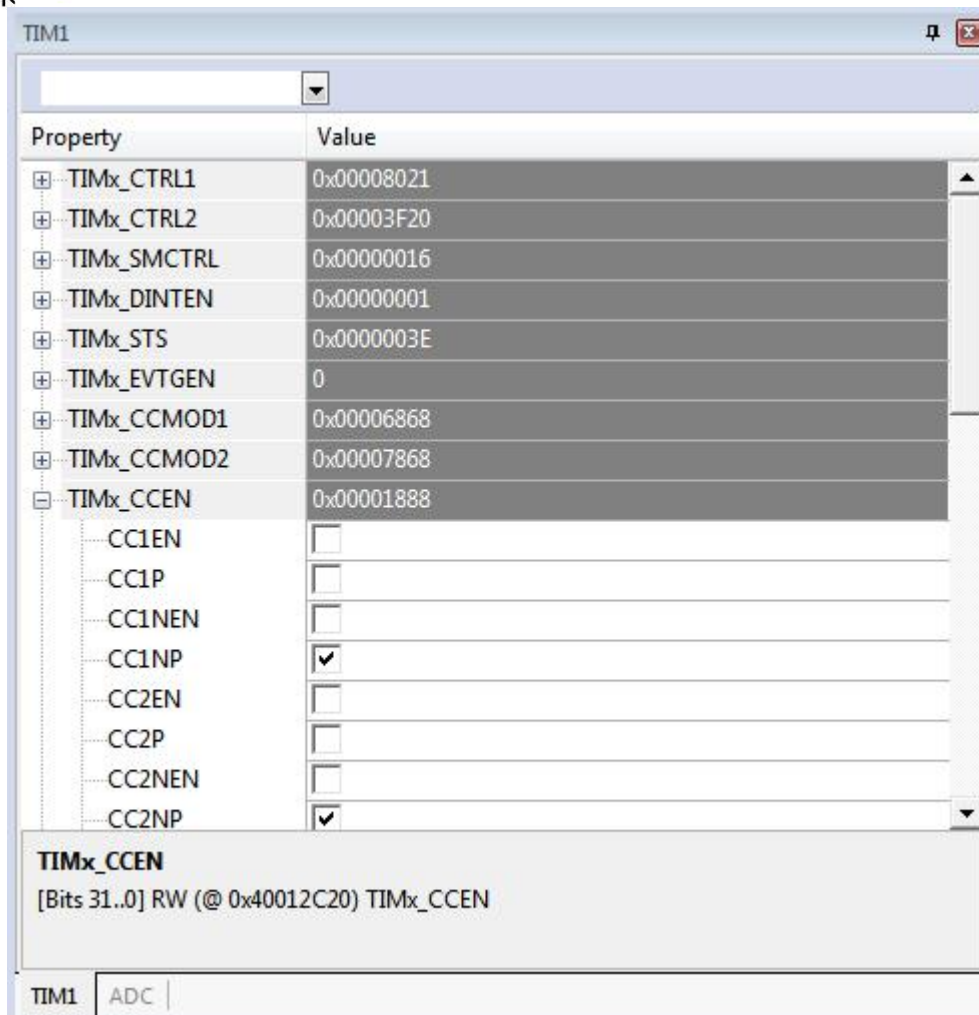


图 1-3 图形化的外设模块操作

驱动库还包括国民技术提供的对应芯片的 SDK，该 SDK 提供了标准外设的驱动库，对电机控制开发必选外设 ADC、TIM、UART。用户可以按需添加其他模块，也可以包含全部驱动库文件。如下图所示引用了所有的驱动库文件。

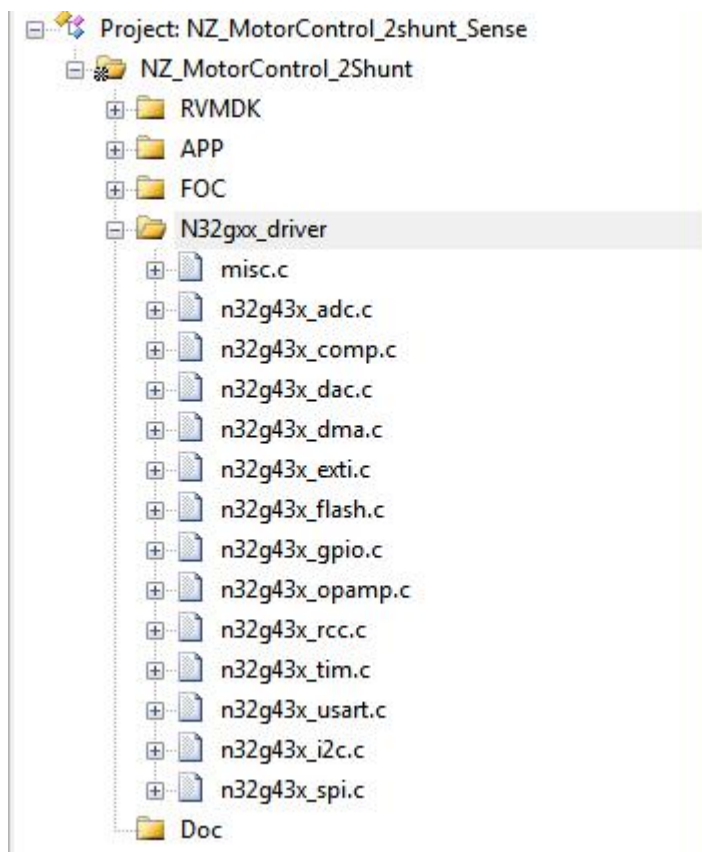


图 1-4 电机控制开发引用的驱动库

## 1.3 APP模块

APP 模块主要实现FOC电机算法执行的固件环境初始化，完成 MCU 系统所需的外设初始化，以及客户需要的逻辑处理功能。本电机控制开发固件主要完成功能有：

- 实现FOC 三电流采样的 ADC 模块和 TIM1 模块配置；
- 实现过流保护作用的配置模块；
- 实现与基本功能之间的数据交互；
- 实现电机驱动的应用逻辑控制；

APP 模块主要包含如下几个 .C 和.H 文件，具体说明描述如表： 1-1 和1-2 所示。

表 1-1 APP 源码文件说明

文件名	描述	包含函数	函数描述
Main.c	主函数入口	int main(void)	主函数入口
		InitHardware(void)	//外设初始化配置//
		InitSystemParam(void)	//系统参数初始化//
init.c	模块初始化函数初始化参数初始化	void RCC_Configuration(void)	//使能时钟配置//
		static void InitIO(void)	//端口初始化//
		static void Init_TIM1(void)	//定时器初始化配置//
		voidInit_NVIC_Configuration(void)	//中断向量表配置//
		static void Init_ADC(void)	//ADC初始化配置//
		void Init_USART1(void)	//串口1初始化配置//
		void EnableMotorPWMOut(void)	//使能电机PWM输出
		void CloseMotorAllMosFet(void)	//关闭电机PWM输出//
		void InitSystemParam(void)	//初始化系统参数//
		void InitHardware(void)	//初始化外设配置//
interrupt.c	中断入口 硬件异常	void HardFault_Handler(void)	//硬件异常中断//
		void TIM1_UP_IRQHandler(void)	//PWM周期中断//
current.c	电流控制	u16 my_abs_diff_u16(u16 a, u16 b)	//求绝对值//
		void MuchCurrentDeal(void)	//过流保护//
		void ProCurrentCtrl(void)	//限流控制//
handle.c	转把控制	void ProctectHandleBar(void)	//转把控制//
Pwm.c	速度控制	void SpeedPwmAdjust(void)	//速度控制//
voltage.c	母线电压	void ProctectVoltage(void)	//欠过压保护//
ebs.c	刹车制动	void ProctectEBS(void)	//电子刹车控制//
motornew.c	其他功能	void SetMotorIbusCurrent(void)	//最大限流值设置//
		void sendata(u8 Data)	//串口发送数据函数//



表 1-2 APP 头文件说明

文件名	描述	结构体	函数
userconfig.h	用户配置头文件	无	无

FOC 的应用，主要占用如下外设模块：

TIM1、ADC 构成电机控制单元下表列出了各中断函数的说明。

表 1-3 电机控制软件中 N32G435 各中断函数说明

函数名	功能介绍	注意事项
void TIM1_UP_IRQHandler(void)	TIM1 Update 中断函数	用于电机的电流采样和FOC电机执行位置
void UART_IRQHandler(void)	串口中断函数	当前使用串口1作为通信口

基于N32G435CBL7的 FOC 的控制算法，固定采样电机的 3 路相电流，得益于 MCU 内置的 ADC，采样速度高达 5MHz，使用外置高速运放，三电阻 FOC 的调制深度高，同时具有执行速度快，控制响应性能方面都有优良的表现。

为了能准确采样到流经采样电路  $R_u$ ， $R_v$ ， $R_w$  的电流  $I_u$ ， $I_v$ ， $I_w$ ，需要在下管开通期间进行电流采样。通道PWM 定时器工作于中心对齐模式，因此最佳采样点为PWM 定时器的过零点。在 FOC 电机控制中，通过TIM1 定时器的过零 CC4 的上升沿触发 ADC 开启多通道采样，依次采样并转换完成  $I_u$ ， $I_v$ ， $I_w$ ，采样窗口示意图如下所示。

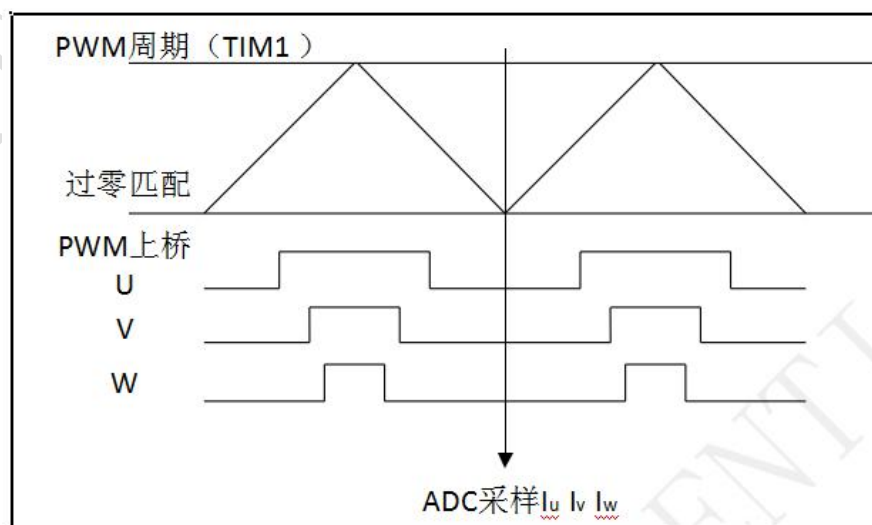


图 1-4 三电阻采样窗口定义



APP 模块必需的外设模块有 ADC、TIM1、UART，在 FOC 中所实现的功能介绍如下：

- ADC 模块

主要完成电流，电压，转速的模块信号采集功能，在每个PWM 载波周期均会完成一次采样，具有注入采样完成中断功能。

- TIM1 模块

主要提供基于高级定时器的提供PWM 载波发生器功能，以及高级定时器具有更新中断功能，通用定时器支持数据上溢或下溢的功能。

- UART 模块

固件支持和PC 工具通信，可以完成多种在线的参数修改，命令控制，各种波形显示等功能。具体配置参数以宏定义的形式给出，如下所示。

```
#define UART_BaudRate          (256000)
```

```
#define UARTx                    UART1
```

配置程序如下：注意软件上位机的配置和固件中的配置需要保持一致。

```
USART_InitType USART_InitStructure;
GPIO_InitType  GPIO_InitStructure;
GPIO_InitStructure.Pin          = GPIO_PIN_6; //PB6  TXD
GPIO_InitStructure.GPIO_Mode    = GPIO_Mode_AF_PP;
GPIO_InitStructure.GPIO_Alternate = GPIO_AF0_USART1;
GPIO_InitStructure.GPIO_Pull    = GPIO_No_Pull;
GPIO_InitPeripheral(GPIOB, &GPIO_InitStructure);
GPIO_InitStructure.Pin          = GPIO_PIN_7; //PB7  RXD
GPIO_InitStructure.GPIO_Mode    = GPIO_Mode_AF_PP;
GPIO_InitStructure.GPIO_Alternate = GPIO_AF0_USART1;
GPIO_InitStructure.GPIO_Pull    = GPIO_Pull_Up;
GPIO_InitPeripheral(GPIOB, &GPIO_InitStructure);
USART_InitStructure.BaudRate     = UART_BaudRate; //256000;
USART_InitStructure.WordLength   = USART_WL_8B;
USART_InitStructure.StopBits     = USART_STPB_1;
USART_InitStructure.Parity       = USART_PE_NO;
USART_InitStructure.HardwareFlowControl = USART_HFCTRL_NONE;
USART_InitStructure.Mode         = USART_MODE_RX|USART_MODE_TX;
USART_Init(USART1, &USART_InitStructure);
USART_ConfigInt(USART1, USART_INT_RXDNE, ENABLE);
USART_Enable(USART1, ENABLE);
```

## 1.4 FOC模块

FOC 模块主要实现电机矢量控制整体的框架，矢量控制的架构如下图所示。

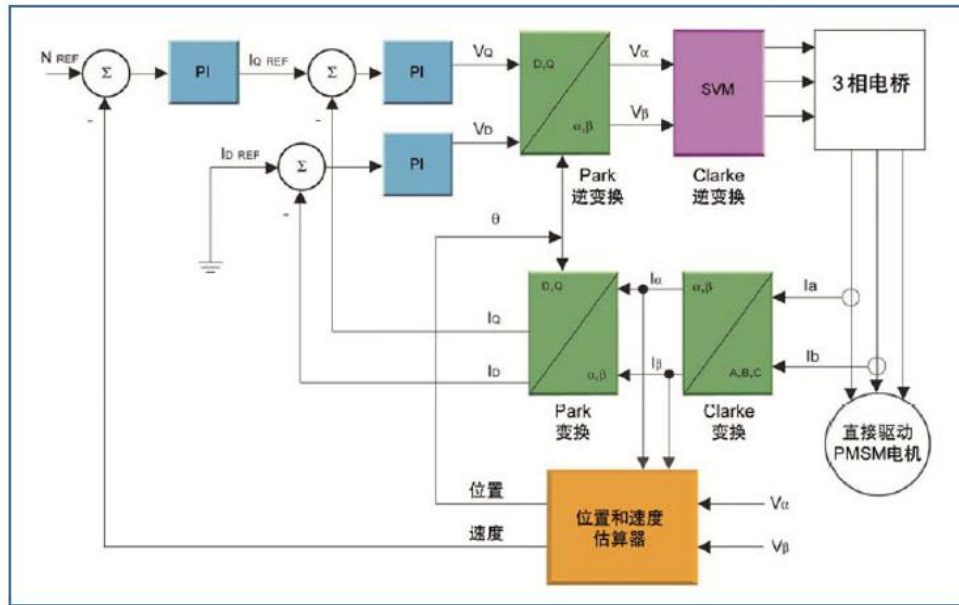


图 1-5 FOC 矢量控制框架：

FOC 控制核心包括以下几个部分：

- 通过测量定子电流矢量，根据磁场定向原理分别对电机的励磁电流和转矩电流控制
- 坐标变换 (Clark, Park, IPark)
- 励磁电流和转矩电流分别采用PI环控制
- PWM 调制模式 (SVPWM) ----- 空间矢量调制
- 转子位置和速度观测 (霍尔传感器)

本固件支持 FOC 控制框架的所有模块，同时提供包含：启停控制策略，故障检测，数据处理以及电机调试参数接口等。内部包含的源文件和头文件说明分别如下。

表 1-4 FOC 内部源文件说明

文件名	描述	包含函数	函数描述
foc.c	Foc矢量控制的数据处理	void SensorHallProcess(void)	//HALL电角度计算//
		void Motor_GetIabc(u16 u16Ia, u16 u16Ib, u16 u16Ic)	//读取三相电流//
		void InitPid(void)	//PID控制参数初始化//
		void PID_Id(void)	//Id闭环控//
		void PID_Iq(void)	//Iq闭环控//

		void Motor_clark(void)	//Clark变换//
		void Motor_park(void)	//Park变换//
		void Motor_ipark(void)	//反Park变换//
		void SVPWM(void)	//SVPWM调制输出//
		void Motor_FocCtrl(void)	//电机FOC控制//

表 1-5 FOC 内部头文件说明

文件名	描述	变量体	描述
focc.h	FOC 层集成头文件	U8Global_Sector;	//电流扇区//
		U16Q15Global_MotorCurrentAngle;	//电机的电角度//
		S16Global_abc_Ia, S16Global_abc_Ib, S16Global_abc_Ic;	//3相静止坐标系下的电流//
		S16Global_static_abc_Ia, S16Global_static_abc_Ib, S16Global_static_abc_Ic;	//3相静止坐标系下的静态电流//
		S16Global_ab_Ia, S16Global_ab_Ib;	//2相静止坐标系下的电流//
		S16Global_ab_Ua, S16Global_ab_Ub;	//2相静止坐标系下的电压//
		S16Global_dq_Uq, S16Global_dq_Ud;	//2相旋转坐标系下的电压//
		S16Global_dq_Id, S16Global_dq_Iq;	//2相旋转坐标系下的电流//
		S16Global_dq_IdRef, S16Global_dq_IqRef;	//2相旋转坐标系下的参考电流//

本固件FOC 矢量控制的流程如下所示。

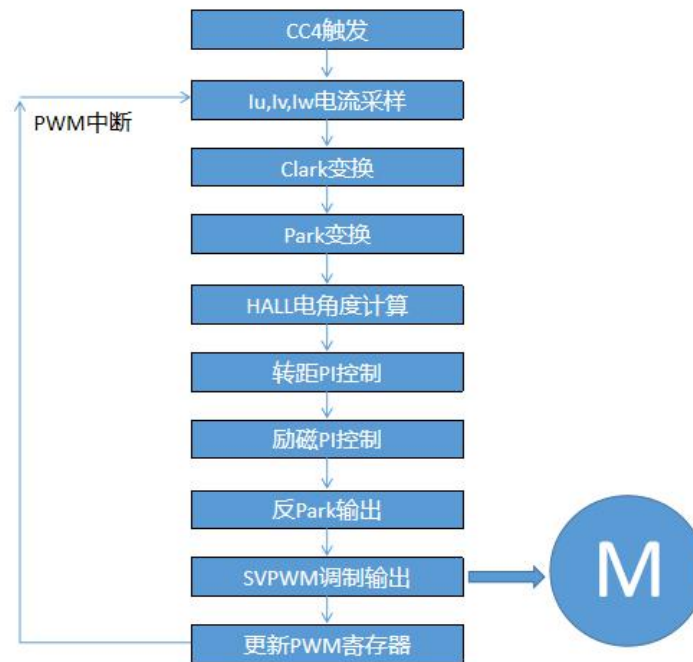


图 1-6 FOC 矢量控制流程图

本固件的算法库在每一个 PWM 周期做一次 FOC实时计算输出 PWM 占空比，并写入 PWM 寄存器中，等待在下一更新事件时刷新控制数据。

PWM 的 CC4 配置为自动触发 ADC 采样信号，当 CC4 上升沿的时候，ADC 按预配置好的注入采样序列对各通道进行采样，ADC 采样注入采样完成后，触发 ADC 采样完成电流采样中断，在PWM中断函数中，软件启动规则通道采样，如下示例代码所示的深色部分代码为启动 ADC 规则通道采样。

```

void TIM1_UP_IRQHandler(void) //PWM周期中断//
{
    GPIOB->PBSC = SW_PB2; //1
    u16 u16Ia, u16Ib, u16Ic;
    static u8 U8_1mscnt, U8AdcIndex;

    MuchCurrentDeal(); //过流//
    //读取AD
    U16Global_ADCResult[Num_Ic] = (ADC->JDAT1); //Ic
    U16Global_ADCResult[Num_Ib] = (ADC->JDAT2); //Ib
    U16Global_ADCResult[Num_Ia] = (ADC->JDAT3); //Ia
    u16Ia = U16Global_ADCResult[Num_Ia];
    u16Ib = U16Global_ADCResult[Num_Ib];
    u16Ic = U16Global_ADCResult[Num_Ic];
}
  
```

```

SensorProcess();//HALL电角度计算//

if(fGlobal_StartRun)
{
    Motor_GetIabc(u16Ia,u16Ib,u16Ic);//相电流计算
    Motor_FocCtrl();//FOC矢量控制//
}

if(ADC->STS&ADC_FLAG_JENDC)//ADC规则组软件启动转换////
{
    ADC->STS = ~(uint32_t)ADC_FLAG_JENDC;//ADC_ClearFlag(ADC, ADC_FLAG_JENDC);//
    ADC->CTRL2 |= CTRL2_EXT_TRIG_SWSTART_SET;
}

```

FOC 整个处理函数集中在ADC 采样中断后完成，首先对ADC 采样到的数据进行整理，重构出完整的 3 相电流信号，然后对电流进行 Clark 变换，得到 Ialfa, Ibeta 两相坐标电流信号，再根据电角度进行Park 变换，得到两相旋转坐标 Id, Iq 电流信号，通过给定的 Iqref, Idref 对 Id, Iq进行PI 偏差控制运算得到两相电压矢量Vd, Vq，进而通过反Park, 反Clark 变换，得到三相互差120° 的三相电压矢量Ua, Ub, Uc，最后由 SVPWM 计算，得到 3 个 PWM 占空比的作用合成矢量控制，实现整个FOC的矢量控制运算。

## 1.5 LIB 模块

LIB 模块主要提供SVPWM、FOC、三角函数、开平方函数等公共函数。一方面这部分功能函数是平台无关的，所需要处理的数据由外部传入。该模块已封装库的形式提供给用户，相应头文件说明哪下。

表 1-6 Lib 对应头文件说明

头文件	变量体
lib.h	无
	无

表 1-7 Lib 功能模块说明

文件名	函数名	函数说明
lib公共函数	s16 IQsin(u16 U16Angle);	正弦查表函数: Angle:Q15 格式角度输入值
	s16 IQCos(u16 U16Angle);	余弦查表函数: Angle:Q15 格式角度输入值

## 1.6 有感霍尔介绍

本固件提供了可选的有感 HALL 模块，对有 HALL 传感器的应用，在初次接线，或不确定 HALL 相序的情况下，需要进行HALL 相序调整。同时选择HALL 作为 FOC 矢量控制电角度计数的来源。

基于HALL 的 FOC 电机控制，为了更稳定的执行，固件算法对读到的 HALL 信号进行了滤波处理预估，所获得的电角度的连贯性更好，电机启动更平顺。相应的代码如下所：



```
if(U32Global_SpeedPeriod<Motor3HallFilterSpeedWord)
{
    U32Global_SpeedPeriod=(U32Global_SpeedPeriod>>3)+(U16Last_SpeedPeriod-(U16Last_SpeedPeriod>>3));
    U16Last_SpeedPeriod=U32Global_SpeedPeriod;
}
else
{
    U16Last_SpeedPeriod=U32Global_SpeedPeriod;
}
```

以上的代码片段为加权滤波函数，具有执行速度快，跟踪准确效果好。实际调试还需根据具体的电机，具体的项目需求，调整加权参数。

## 2 电机调试参数

本固件驱动电机，需要配置MCU 及驱动板的相关参数，有如下参数。

- PWM 载波
- 电流基值
- 电压基值

针对霍尔 FOC 电机控制，增加如下参数

- 电机的极对数
- 霍尔相序
- 霍尔相移

### 2.1 电机驱动板相关的参数

电机驱动板包括所选择的MCU配置，外置运放OP，外置过流比较器等分别介绍如下：

- MCU 配置  
主要配置 MCU 的工作主频，该主频用于指导系统定时器的分配，PWM 载波的分频等功能，该参数配置如下：

```

/*****PWM载波设置*****/
#define MAIN_FREQUENCY      (108000000L) //主频//
#define PWM_Prescaler      (u8) (0)    //PWM时钟预分频//
#define PWM_PERIOD         (u16) (3374) //载波周期//
#define PWM_MAXPERIOD      (u16) (3200) //最大PWM//
#define PWM_MINPERIOD      (u16) (100)  //最小PWM//
#define PWMTimeBase        (u16) (62.5) //PWM周期62.5us//
#define PWMSetCC4MidSamplePiont (u16) (3300) //CC4触发时间设置点//
#define DEADTIME           (u16) (108)  //死区时间//

```

根据具体的MCU 型号配置该参数，该参数需要和系统初始化的主频相同。

1. PWM 载波频率，根据实际电机和项目需求，修改 PWM 载波频率。
2. DEADTIME 死区时间，根据 MOS 所需的死区时间，配置相应的死区时间，以上 108 为死区时间1us，一般 MOS 管死区时间小于1us。

## ● 外置运放OP配置

运放的放大倍数，主要会影响电机驱动板的电流基值，电流基值即为硬件电路电流采样的最大值。电流采样运放的偏置电压Vref，即 0A 电流对应 1.65V，对于 3.3V 系统，电流采集范围即为 3.3V-Vref。再根据硬件电路板上的采样电阻 Rshunt=0.005R，运算放大倍数 Amp，得到电流基值计数公式如下：

$$I_{base} = \frac{3.3V - V_{ref}}{R_{shunt} * Amp}$$

示例：Rshunt = 0.01，Amp=4，Vs=1.65V，则 Ibase = 41.25A。

注：三电阻电流采样运放偏置电压通常为 1.65V。

## ● 外置过流比较器相关

比较器一般用于过流保护，当电流过大时，比较器输出状态改变，通过 MCU 配置，发生过流时对PWM 进行封波停止输出PWM，以实现电机控制中的异常保护和刹车功能。代码如下：

```
TIM_BDTRInitStructure.OssrState      = TIM_OSSR_STATE_ENABLE;
TIM_BDTRInitStructure.OssiState      = TIM_OSSI_STATE_ENABLE;
TIM_BDTRInitStructure.LockLevel      = TIM_LOCK_LEVEL_1;
TIM_BDTRInitStructure.DeadTime       = DEADTIME; // 死区时间 1us//
TIM_BDTRInitStructure.Break          = TIM_BREAK_IN_ENABLE; // 刹车功能开启//
TIM_BDTRInitStructure.BreakPolarity  = TIM_BREAK_POLARITY_LOW;
TIM_BDTRInitStructure.AutomaticOutput = TIM_AUTO_OUTPUT_DISABLE;
TIM_BDTRInitStructure.IomBreakEn     = true; // 过流保护开启//
TIM_ConfigBkdt(TIM1, &TIM_BDTRInitStructure);
```

## 3 整机调试示例

本节举例说明调试有感 FOC 电机所需要主要步骤，整机调试之前，需根据用户需求，仔细阅读 userconfig.h 文件，结合电机和驱动板进行参数的配置，然后调用算法库完成电机驱动控制，userconfig.h中的参数有些已经在前面章节有所介绍，此处集中列出如下：

表 3-1 用户参数说明总表

参数名	参数说明
MAIN_FREQUENCY	//主频//
PWM_Prescaler	//PWM时钟预分频//
PWM_PERIOD	//载波周期//
PWM_MAXPERIOD	//最大PWM//
PWM_MINPERIOD	//最小PWM//
PWMTimeBase	//PWM周期62.5us//
PWMSetCC4MidSamplePiont	//CC4触发时间设置点//
DEADTIME	//死区时间//
ADC_NUMBER	//ADC规则组通道数//
ADREF	//MCU供电电压//
ADCBIT	//ADC 精度//



Speed_KM	//速度//
Motor_N	//极对数//
BATTERY_RATING_VOLTAGE	//母线电压//
BATTERY_LOW_VOLTAGE	//欠压//
BATTERY_LIM_VOLTAGE	//过压//
LOW_VOLTAGE_BACK	//电压回差//
POWER_OFF_Voltage	//掉电电压//
DIVISION_RATIO_AMP	//分压比//
AD_BIT_AMP	//电流比//
CURRENT_AMP	//电流增益//
SENSOR_R	//采样电阻//
AVR_MAX_CURRENT_LIM	//最大限流值//
AVR_MIN_CURRENT_LIM	//最小限流值//
HANDLE_R_DIV	//转把电阻分压比//
HANDLE_LIMIT	//转把超压故障值//
HANDLE_MAX	//转把正常最大值//
HANDLE_HIGH	//转把故障保护值//
HANDLE_BASE	//转把正常启动值//
HANDLE_LOW	//转把低压故障值//
BROCK_TIME	//电机堵转时间//

### 3.1 配置电机参数

根据电机型号，测量电机参数，相应参数填入如下头文件定义中。对有感 HALL，仅需要极对数，可以不需相电阻、相电感、磁链。

图 3-1 确认电机参数示例

```

/*****电机参数*****/
#define Speed_KM      1.0 //速度//
#define Motor_N      23.0//极对数//

```

### 3.2 配置驱动板参数

- ☐ 确认电机驱动板的直流母线电压电流
- ☐ 根据 MCU 的主频，设置电机的载波频率
- ☐ 运放OPA放大倍数，采样电阻确认电流基值
- ☐ 调速把分压比，启动值，安全值，故障基值
- ☐ 电机驱动运行安全堵转时间基值

具体包括如下相应的参数：

```

/*****电压设置区*****/
#define BATTERY_RATING_VOLTAGE 48.0//母线电压//
#define BATTERY_LOW_VOLTAGE 41.5//欠压//
#define BATTERY_LIM_VOLTAGE 68.0//过压//
#define LOW_VOLTAGE_BACK 2.0 //电压回差//
#define POWER_OFF_Voltage 30.0//掉电电压//
#define DIVISION_RATIO_AMP 31.0//分压比//

/*****电流设置区*****/
#define AD_BIT_AMP ADCBIT/ADREF//电流比//
#define CURRENT_AMP 11.0 //电流增益//
#define SENSOR_R 0.0025//采样电阻//
#define AVR_MAX_CURRENT_LIM 21.5 //最大限流值//
#define AVR_MIN_CURRENT_LIM 10.0 //最小限流值//
#define AVR_MAX_CURRENT_LIM_WORD (u16) (AVR_MAX_CURRENT_LIM*SENSOR_R*CURRENT_AMP* AD_BIT_AMP)
#define BROCK_MIN_CURRENT_LIM_WORD (u16) (AVR_MIN_CURRENT_LIM*SENSOR_R*CURRENT_AMP* AD_BIT_AMP)

/*****PWM载波设置*****/
#define MAIN_FREQUENCY (108000000) //主频//
#define PWM_Prescaler (u8) (0) //PWM时钟预分频//
#define PWM_PERIOD (u16) (3374) //载波周期//
#define PWM_MAXPERIOD (u16) (3200) //最大PWM//
#define PWM_MINPERIOD (u16) (100) //最小PWM//
#define PWMTimeBase (u16) (62.5) //PWM周期62.5us//
#define PWMSetCC4MidSamplePiont (u16) (3300) //CC4触发时间设置点//
#define DEADTIME (u16) (108) //死区时间//

/*****转把设置区*****/
#define HANDLE_R_DIV (47.0/(47.0+22.0))//转把电阻分压比//
#define HANDLE_LIMIT (u16) (3.9 * ADCBIT *HANDLE_R_DIV/ ADREF)//转把超压故障值//
#define HANDLE_MAX (u16) (3.3 * ADCBIT *HANDLE_R_DIV/ ADREF)//转把正常最大值//
#define HANDLE_HIGH (u16) (1.2 * ADCBIT *HANDLE_R_DIV/ ADREF)//转把故障保护值//
#define HANDLE_BASE (u16) (1.2 * ADCBIT *HANDLE_R_DIV/ ADREF)//转把正常启动值//
#define HANDLE_LOW (u16) (0.2 * ADCBIT *HANDLE_R_DIV/ ADREF)//转把低压故障值//

/*****堵转时间*****/
#define BROCK_TIME 100//电机堵转时间//

```

图 3-2 确认驱动板参数示例

### 3.3 配置运行模式

配置电机工作在速度环模式或是转距环模式。

图 3-3 电机运行模式选择示例

```

/*****电机工作模式*****/
#define SPEED_MODE //速度模式//
//#define TORQUE_MODE //转距模式//

```

### 3.4 启动电机调试

采用 MDK5.0 环境进行编译，进入调试模式，打开 Watch 窗口，进入观测各项参数是否正常，转动调速把启动电机，观察电机转动相应的参数，实现电机转速控制。

Watch 1		
Name	Value	Type
U8HallCompeState	0x00	uchar
S16Global_dq_Uq	0x03F0	short
S16Global_dq_Uq_Shadow	0x03FF	short
fGlobal_MotorRunDirection	0x0000	ushort
S16Global_dq_Ud	0x00A9	short
fGobal_BatteryOverlowLim	0x00	uchar
fGlobal_MotorBrock	0x00	uchar
fGlobal_SpeedOk	0x00	uchar
U32Global_SpeedPeriod	0x000019AC	uint
S16Global_static_abc_Ia	0x07E2	short
S16Global_static_abc_Ib	0x07D5	short
S16Global_static_abc_Ic	0x07DC	short
S16Global_dq_IdRef	0x0000	short
S16Global_dq_IqRef	0x0000	short
S16Global_dq_Id	0xFFD9	short
S16Global_dq_Iq	0x0000	short
< Enter expression >		

图 3-4 通过调试窗口观测电机运行

#### 4 历史版本

版本	日期	备注
V1.0	2021-05-26	创建文档

NATIONS CONFIDENTIAL

## 5 声 明

国民技术股份有限公司（以下简称国民技术）保有在不事先通知而修改这份文档的权利。国民技术认为提供的信息是准确可信的。尽管这样，国民技术对文档中可能出现的错误不承担任何责任。在购买前请联系国民技术获取该器件说明的最新版本。对于使用该器件引起的专利纠纷及第三方侵权国民技术不承担任何责任。另外，国民技术的产品不建议应用于生命相关的设备和系统，在使用该器件中因为设备或系统运转失灵而导致的损失国民技术不承担任何责任。国民技术对本手册拥有版权等知识产权，受法律保护。未经国民技术许可，任何单位及个人不得以任何方式或理由对本手册进行使用、复制、修改、抄录、传播等。