**STSW-SPIN3202固件包入门**

简介

用于STEVAL-SPIN3202评估板的STSW-SPIN3202固件包允许使用六步（梯形）控制算法控制三相永磁体（PMSM）或无刷直流（BLDC）电机。

STEVAL-SPIN3202电路板采用STSPIN32F0A系统级封装，带有三相电机控制器和STM32F031微控制器，可实现单分流拓扑结构以及无传感器或霍尔效应传感器控制。

**1缩略语和缩写**

**Table 1: Acronyms and abbreviations**

|  |  |
| --- | --- |
| 6Step Lib | The six-step driving library |
| API | Application programming interface |
| BEMF | Back electromagnetic force |
| BLDC | Brushless direct current |
| CMSIS | Cortex® microcontroller software interface standard |
| HF\_TIMx | High frequency timer used to generate the PWM input of the gate drivers高频定时器用于产生栅极驱动器的PWM输入 |
| IDE | Integrated development environment |
| LF\_TIMx | Low frequency timer used for the six-step commutation  用于六步换向的低频定时器 |
| PMSM | Permanent magnet synchronous motor |
| SIP | System-in-package |

**2 STSW-SPIN3202固件包概述**

STSW-SPIN3202固件包包括：用于STEVAL-SPIN3202评估板的驱动程序，用于六步驱动库的STPIN32F0A SIP和嵌入式STM32中间件以及串行通信用户界面样本电机控制应用项目文件

固件包是为STEVAL-SPIN3202板设计的，但它可以适用于安装有STSPIN32F0A SIP或类似设备的其他板。

六步驱动库（6Step Lib）是STM32六步固件库的定制，用于嵌入STM32F031微控制器的STSPIN32F0A集成电路。

6Step Lib支持四种不同的控制模式：

1。无传感器电压模式

2.无传感器电流模式

3.具有霍尔效应传感器反馈的电压模式

4.具有霍尔效应传感器反馈的电流模式

在无传感器模式下，通过检测确定转子位置BEMF过零。

固件库和示例使用C编程语言编写，使用STM32Cube HAL嵌入式抽象层软件或对STM32F031资源的优化访问。支持的IDE工具是IAR Workbench。

要使用此库，您应具备C，三相电机驱动器和电源逆变器硬件的基本知识。

驱动程序抽象低级硬件信息，因此中间件组件和应用程序可以通过一整套API完全管理STSPIN32F0A，这些API以与硬件无关的方式向电机驱动程序发送命令。该套件包括STEVAL-SPIN3202板的应用示例，用于驱动低压三相BLDC / PMSM电机和多个电机控制参数文件，可直接与相应的电机一起使用或作为类似电机的模板。

**2.1包装内容**

解压缩STSW-SPIN3202软件包后，其内容将被安排到主stm32\_cube文件夹下的不同文件夹中。

**2.1.1 Drivers**

Drivers文件夹包含以下源代码：

STSPIN32F0：STSPIN32F0代码位于Components文件夹中; STSPIN32F0代码使用HAL API。

STEVAL-SPIN3202

STM32CubeHAL：用于外设驱动程序的STM32Cube HAL嵌入式抽象层。

CMSIS驱动程序：ARM®Cortex™-M处理器系列的独立于供应商的硬件抽象层。

stspin32f0.h是关于带嵌入式MCU（STSPIN32F0A）的3相控制器电路的HW资源映射的接口文件。它包括嵌入式MCU和嵌入式栅极驱动器之间的高频定时器映射。通常不应更改此文件。

STEVAL-SPIN3202.h是有关电路板的HW资源映射的接口文件，该电路板安装带有嵌入式MCU的三相控制器电路（STEVAL-SPIN3202）。它包括与为电路板选择的驱动方法相关的低频定时器映射（无传感器或霍尔效应传感器，电压或电流模式）。该文件根据在STEVAL-SPIN3202板上执行的任何修改或使用与STSPIN32F0A不同的板进行更新。

**2.1.2Middlewares**

Middlewares文件夹包含：

6StepLib的源代码（电机控制算法的核心）：6Step\_Lib.c和6Step\_Lib.h

串行通信用户界面（基于UART）：UART\_UI.c和UART\_UI.h

**2.1.3 Projects**

Projects文件夹包含演示板功能和IDE项目文件的项目特定源代码。

STSW-SPIN3202固件包包含一个演示层，用于进一步开发代码，位于Projects \ Multi \ Examples \ MotionControl \ STEVAL-SPIN3202文件夹中。

Src子文件夹包含：

main\_32F0.c：FW初始化的主文件（外设，MC\_6Step和UART通信）。它包含定时器，ADC，GPIO，UART等的所有MCU初始化以及6Step库的入口点。通过包含头文件6Step\_Lib.h和6Step init调用MC\_SixStep\_INIT（），用户级别与电机库链接，并且所有API函数都可用。

 stspin32f0\_hal\_msp.c：用于MCU配置的标准ST Cube HAL文件，其中还包含HAL回调（即ADC回调）。

stspin32f0\_it.c：用于MCU中断请求和处理功能的ST Cube HAL文件。它定义了所有中断处理程序，并包含UART通信的起点。

Inc子文件夹包含：

main\_32F0.h：包含头文件6Step\_Lib.h。

* stm32f0xx\_hal\_conf.h：ST Cube HAL配置，包括要在HAL驱动程序中使用的模块列表。

stspin32f0\_it.h：包含中断处理程序的标头。

MC\_SixStep\_param\_32F0.h：包含用6Step Lib驱动电机的电机控制参数。

一个MC\_SixStep\_param\_ <MotorIdentifier> .h：一个电机模型示例的电机控制参数文件。

**2.1.4 Utilities**

该文件夹包含python 2.7.13脚本，用于测试和调整电机控制固件。

SpeedRecording.py脚本使用固件速度循环的粒度生成速度与时间数据，并实时与固件交互，就像通过串行通信终端一样。当电机停止时，测试结束并创建相应的图形。

BemfRecording.py脚本生成BEMF和步骤位置与时间数据。还可以实时地与固件实时交互，就好像它是通过串行通信终端一样。当电机停止时，测试结束并创建相应的图形。

StartStopTest.py可靠性测试，用于检查电机是否可以在命令时启动和停止。

**2.2 STSW-SPIN3202架构**

电机控制应用示例使用以下软件层：

演示层：包括演示基于中间件服务层，低级抽象层（驱动程序）和基于板的功能的基本外设使用应用程序的评估板功能的软件。

中间件层：包括用户界面和6Step Lib。 6Step Lib提供了一个API来控制三相BLDC / PMSM电机。用户界面使用串行通信通过终端发送命令以控制3相BLDC / PMSM电机。 6Step Lib和用户界面通过调用各自的API进行交互。

驱动程序层：包括STM32Cube HAL子层和板支持包（BSP）子层。

STM32Cube HAL子层提供低级驱动程序和硬件接口方法，以与上层（应用程序，库和堆栈）进行交互。它提供通用的，多实例和面向功能的API，通过提供现成的流程帮助减少用户应用程序开发时间。

例如，它提供用于通信外设（I²C，UART等）初始化和配置的API，基于轮询，中断或DMA进程的数据传输管理以及通信错误管理。

HAL驱动程序API分为两类：

1。通用API为所有STM32系列提供通用的通用功能

2.具有特定系列或部件号的特殊功能的扩展API BSP子层提供一组相对于硬件板中的硬件组件。它基于模块化架构，因此可以通过简单地实现低级例程将其移植到任何硬件。

BSP子层由以下部分组成：

1。组件驱动程序：对于板上的外部设备（不包括STM32），组件驱动程序为BSP驱动程序的外部组件提供特定的API，并可以移植到任何其他板上。

2. BSP驱动程序：将组件驱动程序链接到特定的板，并提供一组易于使用的API。

**2.3 6Step Lib功能**

6Step Lib源代码包含在www.st.com上的STSW-SPIN3202包中。

主库具有：

六步算法：

无传感器模式：监测非通电相上的反电动势（BEMF）电压并用于触发换向事件基于霍尔的传感器反馈：数字霍尔效应传感器是用于导出转子的位置和速度

电压模式：驱动电压直接通过功率级施加的PWM占空比设置

电流模式：此电流控制环路限制电机绕组中的电流基于速度环控制PI控制器或可选PID控制器。

速度斜坡，有时称为参考设定值斜坡。

通过串行通信的用户界面。

开始/停止按钮和故障LED。

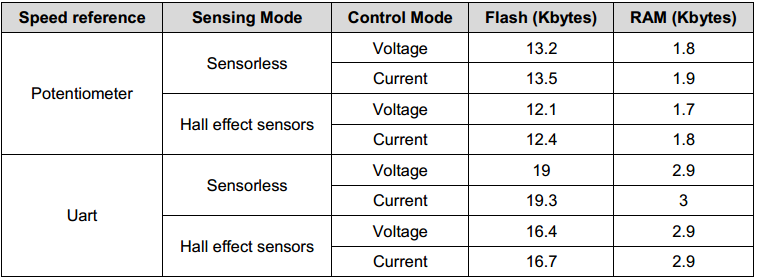
使用电位计的可选速度参考控制。

可选的ADC测量（相电流，直流母线电压，芯片组内部温度）。

故障处理和保护（过流，速度反馈错误等）

**2.4 STSW-SPIN3202 Flash和RAM要求：**

Flash和RAM要求取决于为编译选择的功能和选项。下表中显示的数字是通过优化级别的高特权速度（使用IAR IDE）获得的。



**2.5系统设置指南**

**2.5.1硬件设置**

要将STSW-SPIN3202固件示例与预定义的电机配置一起使用，需要具备以下功能：

一个STEVAL-SPIN3202板

以下电机之一：奔跑BR2804-1700kVRSPro57BL54FaulhaberMinimotor 2036U024B

8V至45 V之间的直流电源（例如，Bull-Running BR2804为12 V）。

带有mini-B连接器的USB电缆。

可以使用现有的MC\_SixStep\_param \_ \* .h文件之一作为模板来驱动不同的电机，以输入新的配置参数。

**2.5.2加载预编译的固件**

硬​​件设置准备就绪后，您需要从主板上的“\ Projects \ Multi \ Examples \ MotionControl \ STEVAL-SPIN3202 \ Binaries”文件夹中加载一个固件二进制文件。

**2.5.3构建和加载自定义固件**

您可以使用以下IDE之一自定义固件：

ARMKeil微控制器开发套件（MDK-ARM）工具链的μVision（V5.17或更高版本）

ARM的IAR嵌入式工作台（EWARM）工具链（V7.50或更高版本），由IARystems®提供

用于STM32的系统工作台，一个基于Eclipse的GCC工具链，由AC6提供。

**2.5.3.2使用KeilμVisionIDE创建构建**

1通过以下文件打开多项目工作区：\ Projects \ Multi \ Examples \ MotionControl \ STEVAL-SPIN3202 \ MDKARM \ STSPIN32F0 \ Projects.uvmpw

工作区包含两个项目：

a。一个用于UART通信

b。一个用于通过电位计进行速度设定

2从菜单中选择Project→Batch build，然后选择两个项目目标并单击Rebuild。

3要编辑项目选项：

右键单击工作区资源管理器中的项目以将其设置为活动

右键单击项目名称正下方的STSPIN32F0A目标名称，然后选择“项目选项...”

4使用“Target STSPIN32F0A选项”窗口启用或禁用STSW-SPIN3202固件的功能：

打开“C / C ++”选项卡，并在“预处理器符号”组的“定义”列表中列出相关符号

5确保电路板已通电并通过USB电缆连接到PC。

6从菜单中选择Flash→下载以加载二进制文件。

7从菜单中，选择Debug→Start / Stop debug Session以调试项目。

您也可以直接从以下位置之一下载二进制文件：

\ Projects \ Multi \ Examples \ MotionControl \ STEVAL-SPIN3202 \ MDKARM \ STSPIN32F0 \ Objects \Project\_Uart.bin

\ Projects \ Multi \ Examples \ MotionControl \ STEVAL- SPIN3202 \ MDKARM \ STSPIN32F0 \Objects\ Project\_Potentiometer.bin

**2.5.4固件定制的定义符号**

表3：符号列表

|  |  |  |  |
| --- | --- | --- | --- |
| Symbol | Description | Default UART | Default Potentiometer |
| USE\_HAL\_DRIVER | Enable the HAL drivers (mandatory) | Enabled | Enabled |
| STM32F031x6 | Enable the specific configuration for the STM32F031x6 microcontroller (mandatory) | Enabled | Enabled |
| UART\_COMM | Enable the UART communication | Enabled | Disabled |
| PID | Enable the PID regulator instead of the standard PI regulator启用PID调节器而不是标准PI调节器 | Disabled | Disabled |
| HALL\_SENSORS | Enable the Hall effect sensor feedback disabling the sensorless driving启用霍尔效应传感器反馈，禁用无传感器驱动 | Disabled | Disabled |
| ALL\_WINDINGS\_ENERGIZATION | Enable current to circulate in all motor phases during the alignment procedure. 在对准过程中，使电流在所有电机相位中循环。 | Disabled | Disabled |
| FIXED\_HALL\_DELAY | Set the delay method when the Hall effect sensor feedback is enabled启用霍尔效应传感器反馈时设置延迟方法 | Disabled | Disabled |
| PWM\_ON\_BEMF\_SENSING | Enabled the BEMF sensing during the ON time of the PWM when sensorless driving is used使用无传感器驱动时，在PWM的ON时间内启用BEMF检测 | Disabled | Disabled |
| MC\_FAN\_4PP\_24V | Select a generic fan motor with hall sensors configuration file. 选择带霍尔传感器配置文件的通用风扇电机。 | Disabled | Disabled |
| MC\_RS\_57BL54 | Select the RS Pro 57BL54 configuration file | Disabled | Disabled |
| MC\_2036\_U\_024\_B\_K312 | Select the Faulhaber Minimotor 2036U024B configuration file | Disabled | Disabled |

|  |  |  |  |
| --- | --- | --- | --- |
| MC\_BR2804\_1700\_KV\_1 | Select the Bull-Running BR2804-1700 kV configuration file | Enabled | Enabled |
| DELTA\_6STEP\_TABLE | Enabled differential strategy in six-step sequence generation在六步序列生成中启用差分策略 | Enabled | Enabled |
| COMPLEMENTARY\_DRIVE | Complementary driving strategy辅助驾驶策略 | Enabled | Enabled |
| VOLTAGE\_MODE | Enable the voltage mode driving instead of the current mode启用电压模式驱动而不是当前模式 | Enabled | Enabled |
| POTENTIOMETER | Enable the control of the target speed through the potentiometer通过电位计启用目标速度控制 | Disabled | Enabled |
| CURRENT\_SENSE\_ADC | Enable the current monitoring through the ADC通过ADC启用电流监控 | Disabled | Disabled |
| VBUS\_SENSE\_ADC | Enable the supply voltage monitoring through the ADC通过ADC启用电源电压监控 | Disabled | Disabled |
| TEMP\_SENSE\_ADC | Enable the MCU temperature monitoring through the ADC通过ADC启用MCU温度监控 | Disabled | Disabled |
| BEMF\_RECORDING | Enable the recording of the BEMF启用BEMF的录制 | Disabled | Disabled |
| SPEED\_SENDING | Enable the monitoring of the speed through UART通过UART启用速度监控 | Disabled | Disabled |
| SPEED\_RAMP | Set the target speed of the motor limiting the acceleration设置限制加速度的电机目标速度 | Enabled | Enabled |

2.5.4.1 USE\_HAL\_DRIVER（必需）

必须定义此符号，以使应用程序，中间件和BSP代码基于HAL驱动程序API。否则，只能直接访问嵌入式STM32F031外设寄存器。

2.5.4.2 STM32F031x6（必需）

必须定义该符号，以使用STSPIN32F0A芯片中嵌入的STM32F031的数据结构，地址映射，位定义和宏。

2.5.4.3 UART\_COMM或POTENTIOMETER（必须至少有一个）

UART\_COMM符号允许将串行端口用作用户界面。该服务基于UART\_serial\_com中间件。

如果未定义UART\_COMM符号，则必须定义POTENTIOMETER以允许通过板载电位计进行速度控制。

除了UART\_COMM符号之外，还可以定义POTENTIOMETER符号。在此配置中，速度可以通过板载电位计控制，也可以通过SETSPD命令使用串行端口控制

2.5.4.4 PID（可选）该符号启用PID调节器，用于速度和转矩控制，而不是PI调节器。

2.5.4.5 HALL\_SENSORS（可选）此符号用于支持霍尔效应传感器（霍尔传感驱动器）并禁用无传感器驱动器。

2.5.4.6 FIXED\_HALL\_DELAY（可选，仅在HALL SENSING驱动器中有效）

此符号控制在大厅状态更改后计算六步换向延迟的方式。

当defined此符号且电机处于RUN状态时，延迟是低频定时器（LF\_TIMx）的恒定时钟周期数;否则，它是步时间的一半。

2.5.4.7 PWM\_ON\_BEMF\_SENSING（可选，仅在电压传感器驱动中有效）

此符号在高频定时器（HF\_TIMx）PWM开启时间内启用BEMF电压检测：

对于低占空比，在PWM关闭时间内检测BEMF电压对于高负载周期，

在PWM ON时间内检测BEMF电压。

该符号仅在电压模式无传感器驱动中有效。

2.5.4.8 MC\_FAN\_4PP\_24V，MC\_RS\_57BL54，MC\_2036\_U\_024\_B\_K312，MC\_BR2804\_1700\_KV\_1（强制和互斥）

您必须列出这些符号或定义自己的符号，以指示固件可以使用的电机控制参数。

四个现有符号中的每一个都具有与其相关联的电机控制参数文件。符号到文件的映射在MC\_SixStep\_param\_32F0.h文件中完成。

2.5.4.9 DELTA\_6STEP\_TABLE（可选，默认）

此符号用于启用六步高频定时器PWM通道配置差分更改：仅编程上一步和新步之间的差异。这样可以减少微控制器的计算量，从而提高电机速度。

2.5.4.10 COMPLEMENTARY\_DRIVE（可选，默认）此符号用于使能HF\_TIMx PWM的互补输出。这也称为同步整流。它通过让续流电流流入下部MOSFET晶体管而不是体二极管来降低传导损耗。

2.5.4.11 VOLTAGE\_MODE（可选，默认）

此符号模式启用电压模式控制：HF\_TIMx PWM占空比由PI稳压器输出直接设置。

未定义此符号时，将启用电流模式控制：HF\_TIMx占空比编程为恒定值，HF\_TIMx PWM在STSPIN32F0A中进行门控当OC\_Comp输入电压高于OC阈值时，通过栅极驱动器控制逻辑。 PI速度调节器输出用于设置OC\_Comp输入电压。

这意味着PI速度调节器是电流回路的输入，实际上在功率MOSFET输入上产生PWM占空比。

2.5.4.12 CURRENT\_SENSE\_ADC，VBUS\_SENSE\_ADC，TEMP\_SENSE\_ADC（可选）

这些符号允许使用ADC来检测功率MOSFET中的电流，功率MOSFET桥的高压侧的VBUS电压或电路温度。

2.5.4.13 BEMF\_RECORDING（可选）

此符号启用固件BEMF记录和发送：最多BEMF\_ARRAY\_SIZE个连续BEMF样本和相应的步骤位置存储，然后使用UART DMA发送到串行端口。

发送到串行端口的帧也被“时间”标记，以便外部跟踪工具（BemfRecording.py python脚本）构建BEMF电压与时间和步长位置对时间的关系图。

2.5.4.14 SPEED\_SENDING（可选）

此符号启用固件速度发送：每次运行速度循环时，滤波后的速度反馈都将发送到串行端口。然后，外部跟踪工具（SpeedRecording.py python脚本）可以构建速度与时间的关系图。

2.5.4.15 SPEED\_RAMP（可选）该符号可通过速度斜坡实现电机加速和减速的固件控制。速度目标以电动机控制参数文件中定义的机械速率ACC线性倾斜。速度目标是PI调节器回路的设定点输入。

**2.5.5加载固件**

将新固件加载到设备上的最简单方法是将二进制文件复制到ST-LINK提供的大容量存储接口中;对磁盘进行简单的拖放操作就足够了。执行此操作的步骤说明如下：

1仅限第一次：安装可从ST网站（STSW-LINK009）下载的STLINK V2-1驱动程序。

2通过USB电缆将评估板连接到PC。

3通过连接器（J2）为评估板提供直流电压，工作电压范围为8 V - 45 V.

4名为SPIN32F0的新驱动器应出现在可移动存储列表中。

5删除此驱动器上的所有文件。

当二进制大小接近STSPIN32F0的32 KB限制时，建议执行此步骤。

6将二进制文件复制到驱动器根目录。

ST-LINK上的红色/绿色LED应开始闪烁。

7刷新文件资源管理器。

如果二进制文件已消失且未生成错误日志文件，则已成功加载二进制文件。

8重置电路板以运行新加载的固件。

**2.6示例应用**

**2.6.1电位计示例**

该示例的硬件设置为：

STEVAL-SPIN3202板正确供电（USB供电，VBUS连接到电源）

电机的三相正确连接到电路板（默认情况下）目标电机为Bull-Running BR2804-1700 kV）

按下USER1按钮启动电机运动。电位计（R6）可用于动态调节1200 RPM至12000 RPM之间的电机速度。

再次按下USER1按钮可停止电机。

如果电机运行时发生错误，“USER2”键的LED亮起，电机停止。按USER1按钮清除故障。

**2.6.2 Uart示例**

该示例的硬件设置为：

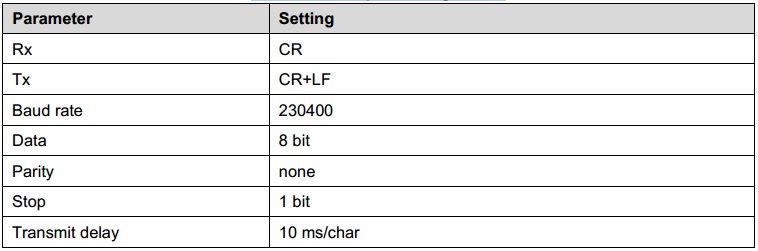
STEVAL-SPIN3202板正确供电（USB供电，VBUS连接到电源）

电机的三相正确连接到电路板（默认情况下） ，目标电机为Bull-Running BR2804-1700 kV）

串口接口可用于控制电机。打开连接到PC的USB上的串行端口，并按照下表进行配置。

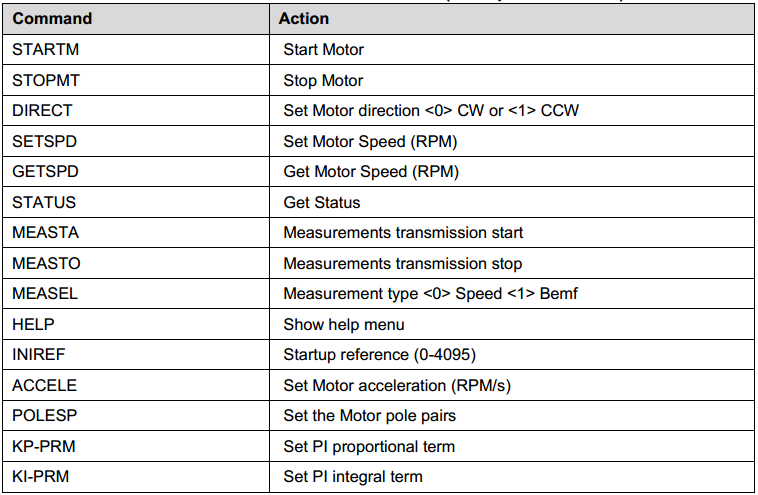
然后选择评估板对应的COM端口;它被标记为STMicroelectronics STLink虚拟COM端口。

表4：串行端口配置



下表列出了可用的命令及其操作。

表5：STSW -SPIN3202用户界面（串口命令）



默认情况下，在Uart示例中，电位计功能被禁用。但是，仍然可以按下“USER1”按钮来启动和停止电机运动。

**3使用STSPIN32F0A实现六步电机控制算法**

**3.1 BLDC理论简介**

无刷三相电机由带三个绕组的固定定子和带内部永磁体的移动转子组成。

在六步驱动中，电循环分为六个换向步骤。对于每个步骤，总线电压施加到电动机的三相绕组中的一个，而接地施加到第二绕组。第三个绕组保持打开状态。除了电动机相绕组改变以产生旋转的定子磁场之外，以相同的方式执行连续的步骤。

在图17中：“BLDC电机控制序列”，红色箭头表示定子磁通矢量旋转，蓝色箭头表示方向。

BLDC电机经历在电机相绕组中引起的梯形反电磁力（BEMF）。由于相电流的形状，六步驱动也称为梯形控制。这种控制方法提供了最大的效率和最小的转矩波动，因为电机本质上是以这种方式驱动的。

Figure 17: BLDC 电机控制顺序

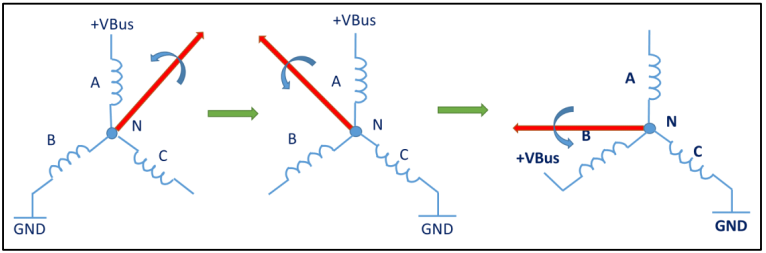
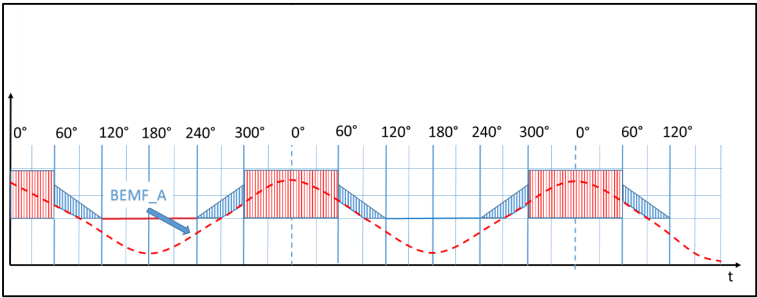


图18：具有BEMF序列和PWM调制的六步相电压



**3.2转子同步**

六步驱动是同步控制驱动，因此如果在转子处于正确位置时执行两个连续步骤之间的换向，则实现最大效率，其中BEMF信号和相电流同步且同相。

可以实施两种方法来执行转子同步：

传感驱动：使用位置传感器确定转子位置。

无传感器驱动：基于BEMF检测。它分析了BEMF信号的过零点，以建立换向点。浮动相的BEMF信号相对于接地点之间的匹配用于产生两个连续步骤之间的换向，以便实现转子同步。

6Step Lib可以使用传感器（霍尔效应传感器）或无传感器驱动器，具体取决于用户项目选项中定义的符号。

**3.3电机驱动器控制**

通过两种不同控制方法生成的PWM序列调制相电压来执行六步驱动：

电流模式（峰值控制）：在这种情况下，需要两个控制回路，一个内部和一个外部。内部回路通过模拟比较器通过硬件产生用于从模拟参考开始的电动机电流调节的PWM序列。它提供了绕组中的转矩控制和固有电流限制。内部环路参考由外部环路产生，该外部环路包含一个数字PI调节器，将速度参考值与电机速度反馈进行比较。

电压模式：在这种情况下，只需要一个环路。 PI调节器直接产生PWM，调节相电压，将速度参考与电机sp的反馈进行比较

**3.3.1电流模式控制**

在图19：“电流模式控制”中，内环包含获取电机电流的所有元件，放大并调节要与参考值进行比较的信号。

比较器的输出连接到嵌入式MCU的ETR定时器功能，并可能连接到栅极驱动器的控制逻辑。

当比较器的输出变为高电平并连接到栅极驱动器逻辑时，所有高侧栅极驱动器输出都被关闭，因此外部高侧电源开关也是如此。高侧开关保持关闭，直到相应驱动输入的下一个上升沿。使用MCU ETR定时器功能可以获得类似的操作模式，但是这在6Step Lib中没有实现，因为嵌入式栅极驱动器逻辑中已经存在这种机制。该机制可以按时调节PWM的持续时间。

运算放大器OP1，OP2，OP3和比较器是STSPIN32F0A的内部器件。

图19：电流模式控制

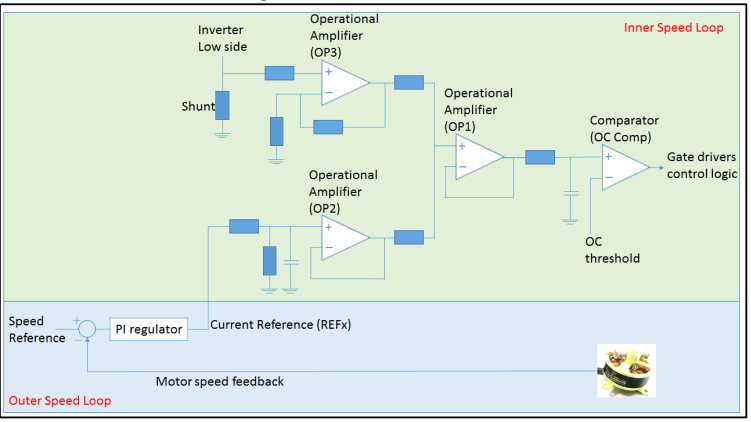
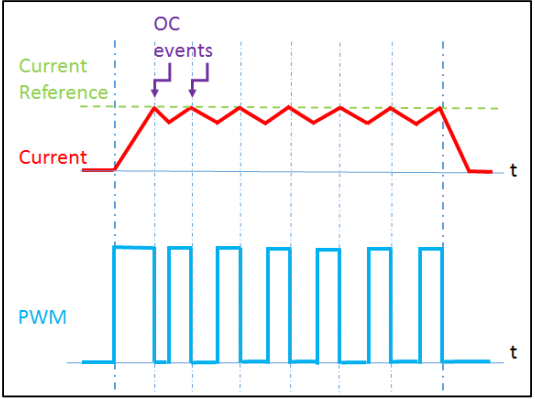


图20：“电流模式控制的栅极驱动器控制逻辑功能”显示正常操作期间的栅极驱动器控制逻辑。红线是以固定电流参考（绿色）调节的电动机电流。蓝色波是由栅极驱动器控制逻辑管理的PWM信号。每当调节的电动机电流达到OC参考阈值时，栅极驱动器控制逻辑关闭内部高压侧开关，从而关闭高压侧功率开关外部输入端的PWM信号。

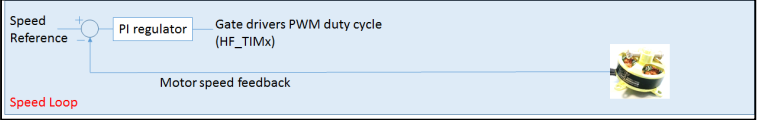
图20：电流模式控制的栅极驱动器控制逻辑功能



**3.3.2电压模式控制**

图21：“电压模式控制”显示电压模式驱动方法。回路通过PI调节器生成输出，PI调节器将速度参考值与实际速度电机反馈进行比较。无栅极驱动器控制逻辑或ETR功能是必需的，因为输出计算PWM信号的新占空比值。

图21：电压模式控制



**3.4无传感器驱动方法**

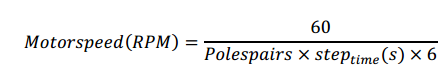
**3.4.1转子速度测量**

每个电机都采用特定的绕组结构;我们将看一下正弦曲线和梯形结构。这些可以通过手动旋转电机和在两个电机相上连接探头来区分。下图显示了在示波器上采集的信号。

转子的旋转磁场在定子绕组中感应出BEMF信号。如果电动机本质上是正弦曲线或梯形，则感应BEMF信号是周期性的，并且它们的频率与电动机转动的频率成比例。比例系数是电机极对数：该数字用于计算电机的机械速度。例如，对于1000Hz（电频率）和10个极对（或20个极点），机械频率为100Hz。

因此，可以通过测量电机的电气频率来计算以RPM为单位的电机速度：

公式1：电机速度（RPM）



其中，步进时间是调节每个步长的低频定时器（LF\_TIMx）的持续时间。它乘以步数并以RPM为单位进行转换。步进时间的新值由过零计算功能管理，并以步骤生成的频率更新（步骤N至步骤N + 1）。

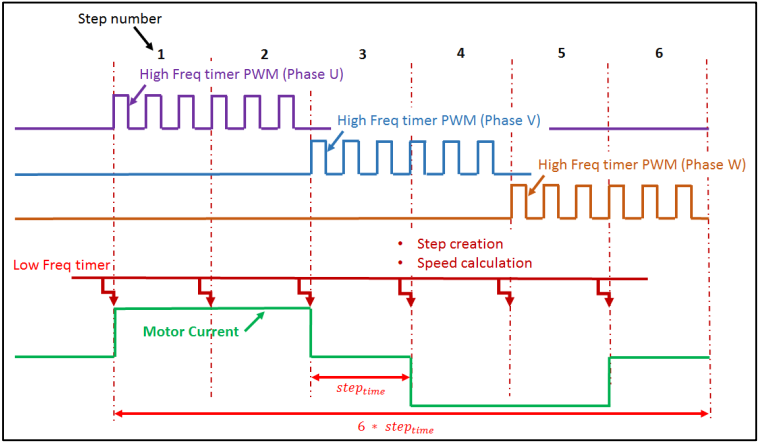
公式2：步长时间



其中，LF\_TIMx持续时间是定时器计数器时钟单位，LF\_TIMx预分频器是系统时钟和定时器计数器时钟之间的分频比。

如图24所示：“高频和低频定时器信号”，每次低频定时器周期结束时，都会创建一个新步骤，并使用相关的高频定时器PWM配置并计算机械速度。

图24：高频和低频定时器信号



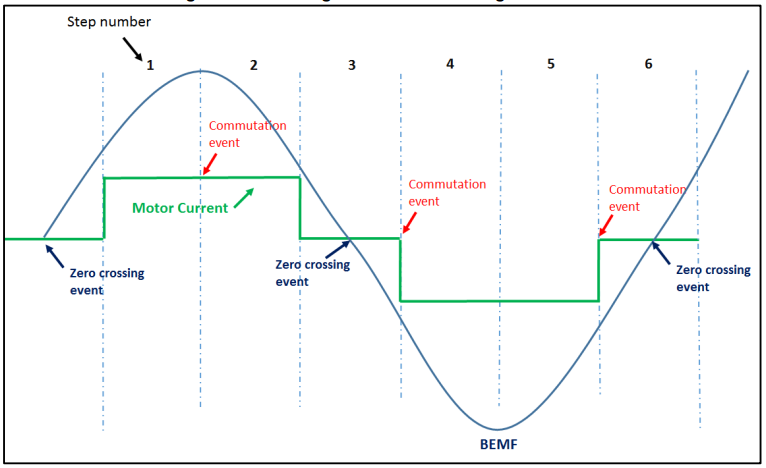
**3.4.2换向，退磁延迟，过零事件**

由于BLDC是同步电动机，驱动器的主要目的是确保转子位置与定子磁场保持同步。可以证明通过使定子的磁场相对于转子磁场保持90度空间提前来获得驱动器的最大效率。

这可以通过在每个步骤的中间保持过零点来实现（参见图25：“BEMF信号和过零检测”）。

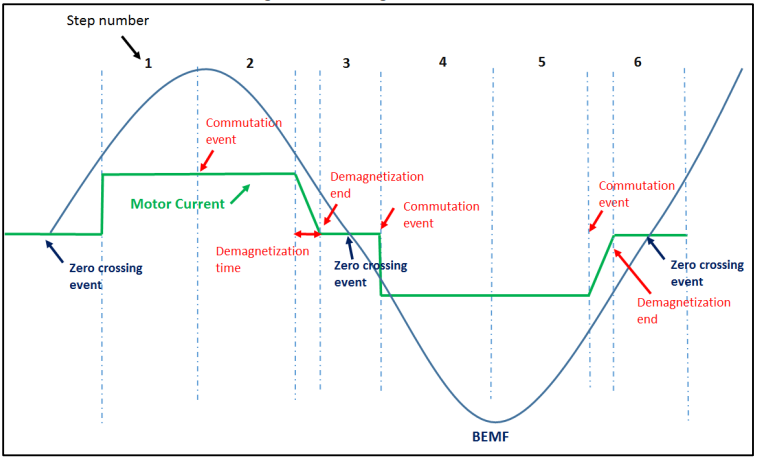
由于每个电动机具有不同的电感值，因此电流可以在每一步换向时延迟达到零。必须处理这种“退磁”事件，并确定过零检测的起点。

图25：BEMF信号和过零检测



此延迟是步进时间的百分比（参见图26：“退磁时间”）。当退磁延迟时间之后浮动相中的电流达到零值时，可以测量BEMF信号。

图26：退磁时间



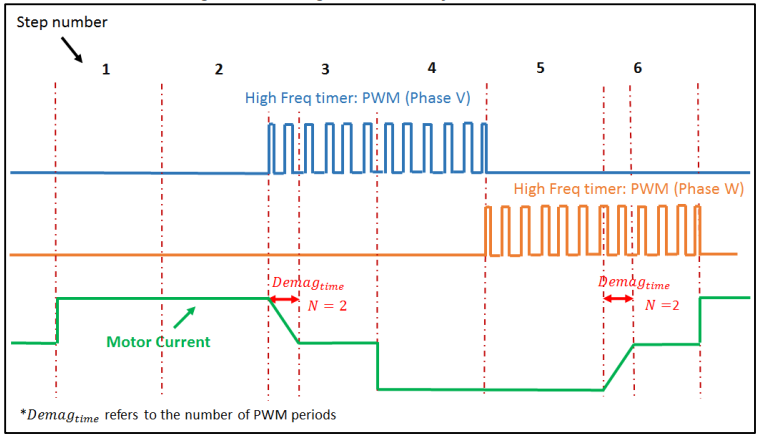
该方法由固件管理，并基于滤波后的速度反馈（与步进时间成反比）。此外，在预定义的速度阈值之上，应用最小预定义的退磁延迟，允许以高速度减少微控制器计算。

退磁延迟时间被计算为多个高频PWM周期

对于中心对齐定时器计数器，PWM周期是定时器周期的两倍。

在STSW-SPIN3202固件包中，高频定时器在无传感器驱动器中居中对齐。下图显示了此配置中N = 2的退磁延迟。

图27：N = 2时的退磁延迟时间



**3.4.3过零检测**

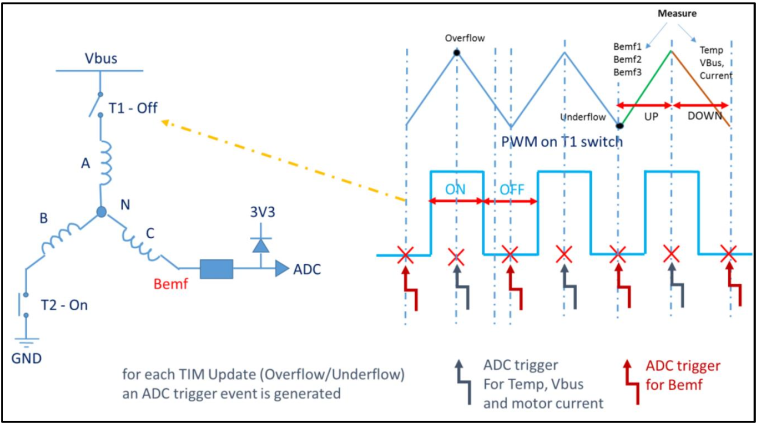
为了使BLDC电机与驱动器同步，BEMF信号等于电机中性点电压的瞬间被识别为过零点。根据为调制高端器件而产生的高频PWM信号，有两种不同的场景来测量BEMF值：

可以在PWM信号的开启和关闭时间内检测过零点。

6Step Lib支持两种BEMF测量模式。但是，默认情况下，STSW-SPIN3202固件包中仅启用基于T-off模式的BEMF测量。要在T-on和T-off时间内启用测量，必须在项目选项中定义PWM\_ON\_BEMF\_SENSING符号。

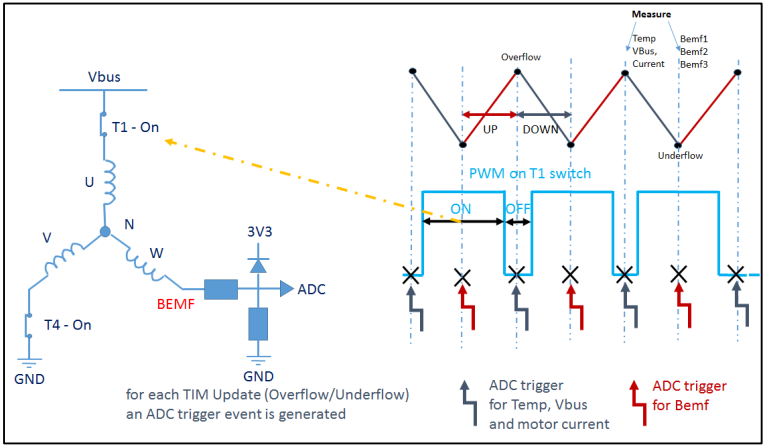
在两种模式中，通过将BEMF转换值与固定阈值进行比较来执行过零点的检测。图28：“PWM关闭期间的BEMF测量和ADC采样”显示T1开关T-OFF期间的BEMF测量，当W相浮动且ADC采样方法用于测量BEMF信号，VBUS，温度，电机当前等

图28：PWM关断时间和ADC采样期间的BEMF测量



下图显示了T1开关T-ON期间的BEMF测量，可用于大PWM占空比。

图29：PWM开启时的BEMF测量和ADC采样



**3.4.4启动程序**

要对BLDC电机执行无传感器控制，BEMF必须达到可读的值才能读取。由于BEMF振幅随转子速度增加，因此可以高于特定阈值的速度读取。因此，必须首先强制电动机旋转而不需要BEMF反馈。因此，实现异步启动策略非常重要，该策略可使电机达到所需的速度。

启动电机需要两个不同的步骤，并通过BEMF检测开始闭环控制：

1.对准阶段

2.斜坡生成阶段

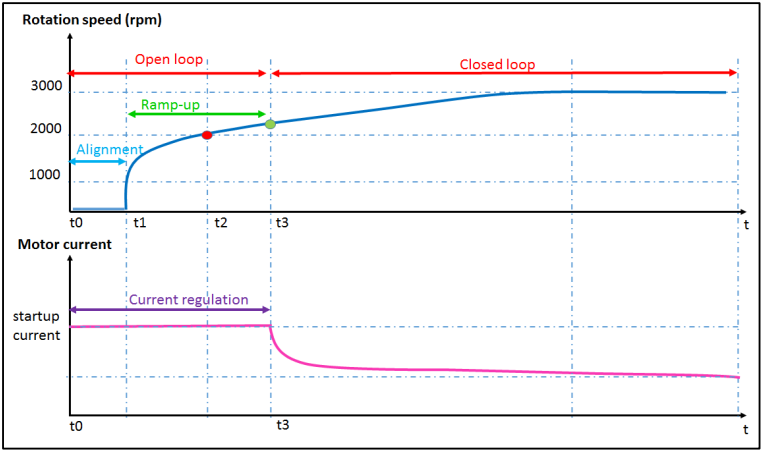
**3.4.4.1对准阶段**

在对准阶段，转子被强制保持在具有特定开关配置的指定位置。保持固定电流（由固件管理）以将转子移动到更近的极对位置。通过选择特定步骤数来执行对齐。

**3.4.4.2斜坡生成阶段**

为了执行良好的BEMF控制，电机以速度斜坡曲线启动。在此期间，保持固定和受控的电流以运行和加速电动机。控制系统监控电机速度并检查该值是否达到指定阈值（由固件管理，图30中的红色圆圈：“对齐和加速生成”）。在肯定检查结果的情况下，过零事件计数器启动，并且在特定值（由固件管理，图30中的绿色圆圈：“对齐和加速生成”）中，确定闭环操作的起始点。如果过零事件的数量达到阈值（可在固件中设置），则达到此条件。

图30：对齐和加速生成



**4 STSPIN32F0六步固件库入门**

**4.1硬件资源映射**

MCU和6Step库之间的stspin32f0.h和STEVAL-SPIN3202.h文件接口。

如果高频定时器（PWM）是TIM1，则stpin32f0.h文件将TIM1 MCU数据结构映射到main\_32F0.c中的演示代码使用的通用名称BSP\_SIP\_HF\_TIMx.HF定时器初始化函数用于附加通用定时器句柄HF\_TIMx 6Step Lib到TIM1 MCU的数据结构：

HF\_TIMx.Instance = BSP\_SIP\_HF\_TIMx;

如果更改计时器，则必须使用正确的外围设备更新此文件。下表列出了无传感器电压模式驱动的主要外围设备。

表6：无传感器coltage模式驱动的主要外围设备映射

|  |  |  |
| --- | --- | --- |
| HF\_TIMx | BSP\_SIP\_HF\_TIMx | TIM1 |
| LF\_TIMx | BSP\_BOARD\_LF\_TIMx | TIM3 |
| ADCx | BSP\_SIP\_ADCx | ADC1 |
| huart | BSP\_SIP\_UART | USART1 |

映射的另一个例子是ADC。 ADC使用的GPIO取决于电路板设计。因此，ADC的GPIO映射位于STEVAL-SPIN3202.h文件中，而ADC外设映射位于stspin32f0.h文件中，因为它仅依赖于集成电路。

STEVAL-SPIN3202板采用无传感器电压模式驱动，ADC用于检测：

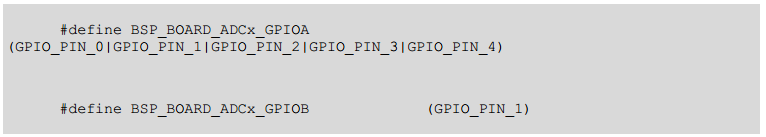
•PA0，PA1和PA2 BEMF

•PA3电位器电压调节速度

PA4电流检测分流电阻器中的电压

•PB1 VBUS功率MOSFET的供电电压

因此在STEVAL-SPIN3202.h中，定义是：



如果出于电路板设计原因，最好使用PA5进行速度感应，则上述定义将如下所示进行修改：



**4.2电机控制参数**

下表说明了所有电机控制参数及其说明。某些常量由编译器预处理器计算，无法修改。

表7：电机控制基本参数

|  |  |  |  |
| --- | --- | --- | --- |
| NUM\_POLE\_PAIRS | Number of pole pairs of BLDC motor | 16bit unsigned | Yes |
| DIRECTION | Set the motor direction: (0) for CW or (1) for CCW. | 0 or 1 | Yes |
| TARGET\_SPEED\_OPEN\_LOOP | Target speed in open loop phase (ramp up) | 32bit unsigned, RPM | Yes |
| TARGET\_SPEED | Target speed on closed loop phase when the potentiometer is disabled | 32bit unsigned, RPM | Yes |

表8：电机控制高级电压模式参数

|  |  |  |  |
| --- | --- | --- | --- |
| STARTUP\_DUTY\_CYCLE | Tenths of percentage of high frequency gate driving PWM on time. In example, for a percentage of 20%, the STARTUP\_DUTY\_CYCLE is 200. 高频栅极驱动PWM导通时间的百分之十。例如，对于20％的百分比，STARTUP\_DUTY\_CYCLE为200。 | 16bit unsigned, between 1 and 1000 | Yes |
| KP\_GAIN | Kp parameter for PI(D) regulator. | 16bit unsigned | Yes |
| KI\_GAIN | Ki parameter for PI(D) regulator. | 16bit unsigned | Yes |
| KD\_GAIN | Kd parameter for PID regulator. | 16bit unsigned | Yes |
| K\_GAIN\_SCALING | Kp, Ki, (Kd) scaling for PI(D) regulator. | 8bit unsigned, typically 14 | Yes |
| LOWER\_OUT\_LIMIT | Low Out value of PI regulator. | 16bit signed | Yes |
| UPPER\_OUT\_LIMIT | High Out value of PI regulator. | 16bit signed | No |
| MAX\_POT\_SPEED | Maximum Speed regulated by potentiometer. 最大速度由电位器调节。 | 32bit unsigned, RPM | Yes |
| MIN\_POT\_SPEED | Minimum Speed regulated by potentiometer. 最小速度由电位器调节。 | 32bit unsigned, RPM | Yes |