UM2152
User manual

Getting started with the STSPIN32F0 FOC firmware example STSW-SPIN3201

Introduction

This document presents how to use the STSW-SPIN3201 firmware example. In association with the evaluation board STEVAL-SPIN3201, this firmware allows to control a 3-phase permanent magnet motor with field-oriented control algorithms in sensorless operations. The STEVAL-SPIN3201 board is designed around the STSPIN32F0, a 3-phase controller with an integrated STM32, and implements 3-shunt resistors as current reading topology.

This document does not detail the FOC and sensorless algorithms, the MC API, and the Motor Control Workbench.

Contents

1 **Acronyms and abbreviations 3**

2 **STSW-SPIN3201 FW presentation 4**

2.1 Overview 4

2.2 Package content 5

3 **System setup guide 6**

3.1 Hardware setup 6

3.2 Loading the pre-compiled firmware 6

3.3 Building and loading firmware with different parameter configuration 7

3.3.1 Parameter configuration with the STM32 MC workbench 8

3.3.2 Generation of library with new parameters configuration 9

3.3.3 Compile and download the firmware 12

3.3.4 Control the motor 12

4 **Revision history 15**

1 Acronyms and abbreviations

Table 1. Acronyms and abbreviations

Acronym	Description
API	Application programming interface
BLDC	Brushless direct current
CMSIS	Cortex® Microcontroller Software Interface Standard
FOC	Field-oriented control
FW	Firmware
GUI	Graphical user interface
IDE	Integrated development environment
MC	Motor control
MCU	Microcontroller unit
PMSM	Permanent magnet synchronous motor
RTOS	Real time operating system
SDK	Software development kit

2 STSW-SPIN3201 FW presentation

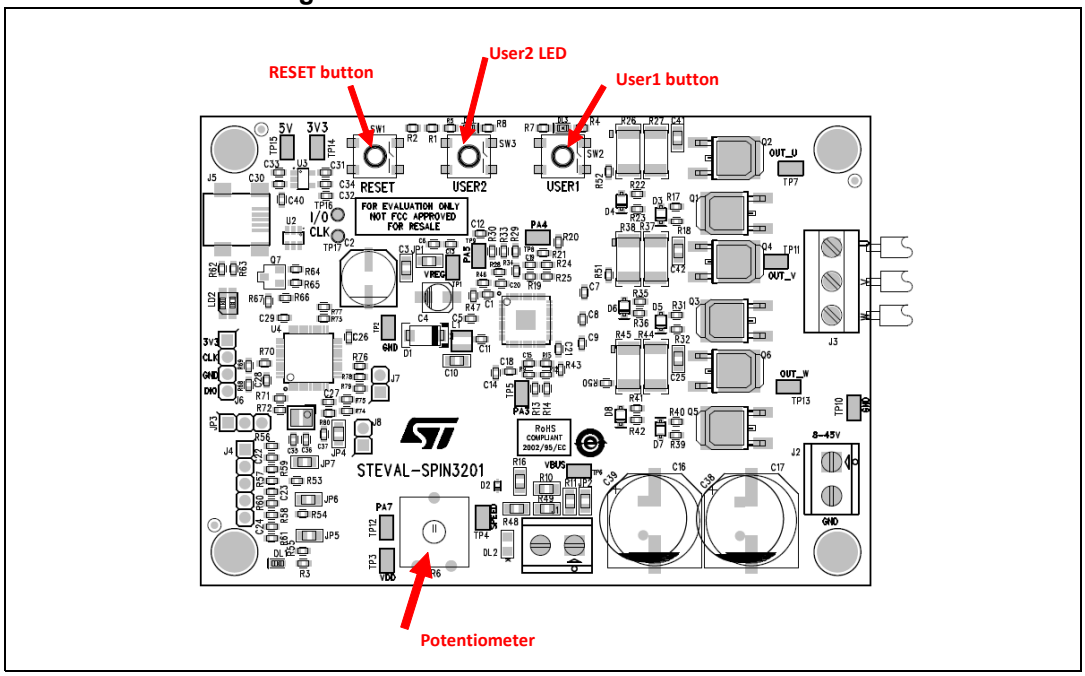
2.1 Overview

The STEVAL-SPIN3201 board is a three-phase brushless DC motor driver board based on the STSPIN32F0 and STD140N6F7 MOSFETs.

The device provides an affordable and easy-to-use solution for the implementation of low voltage motor driving applications such as fans, drones and power tools.

The board is designed for sensored or sensorless vector control - FOC algorithms with 3-shunt sensing.

Figure 1. User interface STEVAL-SPIN3201



In association with the evaluation board STEVAL-SPIN3201, the STSW-SPIN3201 firmware example allows to control a 3-phase permanent magnet motor with field-oriented control algorithms in sensorless operations. The STSW-SPIN3201 firmware is a customization of the STM32 PMSM FOC library, a part of the STM32 MCSDK (Motor Control software development kit) firmware (X-CUBE-MCSDK and X-CUBE-MCSDK-FUL).

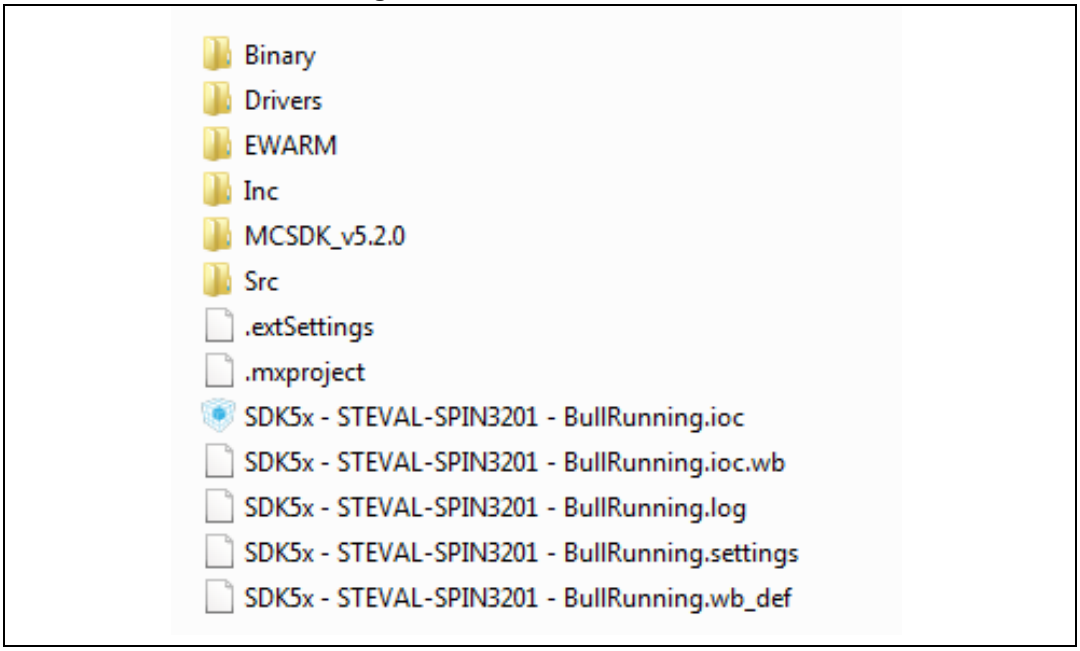
In this customization, the target speed is imposed through potentiometer and the motor is started/stopped using the USER1 button of the board. It is also possible to use the STM32 Motor Control Workbench (GUI) to monitor and change in real time some of the variables of the algorithm and to initialize a new project according to the application needs. This GUI is not included in the STSW-SPIN3201 package but it is distributed with the STM32 MCSDK.

2.2 Package content

The STSW-SPIN3201 package is provided in the form of a zip file. Once unzipped the working folder and the project file are available:

- SDK5x - STEVAL-SPIN3201 - BullRunning.stmcx: this is the initial configuration of parameters in the STM32 Motor Control Workbench. Loading this file it is possible to modify the parameters for the application;
- SDK5x - STEVAL-SPIN3201 - BullRunning: it is the main folder of the firmware. This directory, as shown in *Figure 2*, is organized in subfolders which contain;

Figure 2. Main folder structure



- Binary: a compiled version of the firmware as configured by the initial SDK5x - STEVAL-SPIN3201 - BullRunning.stmcx. In this configuration the STEVAL-SPIN3201 board is associated to the Bull-Running BR2804-1700 kV motor;
- Drivers: the HAL driver library and the CMSIS files from ARM®;
- EWARM: this is the working folder when using IAR IDE;
- Inc: application includes files;
- MCSDK_v5.2.0: the files for FOC library (revision 5.2);
- Src: application source files.

The SDK5x - STEVAL-SPIN3201 - BullRunning.ioc is a project file used by the STM32CubeMX to generate the code.

Note: ***IMPORTANT: do not use the STM32CubeMX project file to change configurations related to the STM32 PMSM FOC library (timers, DMA, interrupts, etc.). All the changes must be done through the STM32 Motor Control Workbench only.***

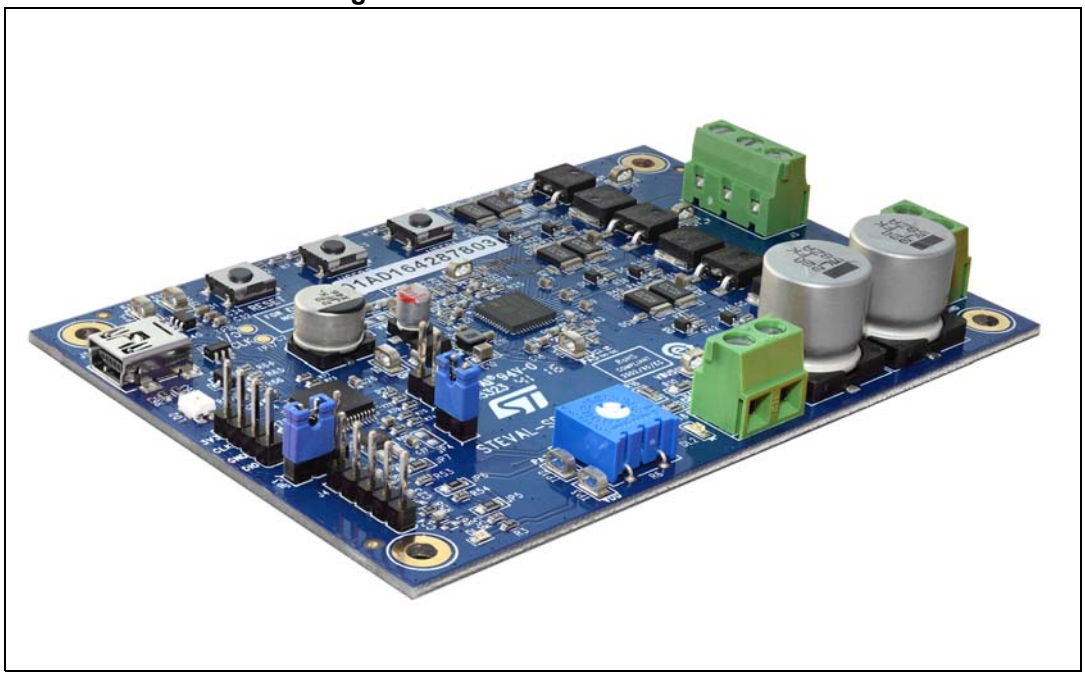
3 System setup guide

3.1 Hardware setup

To use the firmware example in its default configuration, it is required to have:

- One STEVAL-SPIN3201 board;
- One Bull-Running BR2804-1700 kV brushless motor. A different motor can be used varying the respective settings via the MC Workbench (see *Section 3.3.1 on page 8*). In this case the firmware must be rebuilt (see *Section 3.3.2 on page 9*);
- A 12 V DC power supply. In any case the board supports a power range between 8 V and 45 V. If the supply voltage is changed, this new setting must be applied to the FOC library via the MC Workbench (see *Section 3.3.1 on page 8*). In this case the firmware must be rebuilt (see *Section 3.3.2 on page 9*);
- A USB cable with a mini-B connector.

Figure 3. STEVAL-SPIN3201 board



3.2 Loading the pre-compiled firmware

Once the hardware setup is ready, it is necessary to load the board with firmware.

The easiest way to load a new firmware into the device is to copy the binary file into the mass storage interface provided by the ST-LINK, for example to drag and drop it into the disk. The default binary from the SDK5x - STEVAL-SPIN3201 - BullRunning\Binary folder can be loaded using the process described below:

- First time only: install the STLINK V2-1 drivers that can be downloaded from the ST website (STSW-LINK009)
- Connect the evaluation board to a PC with the USB cable

- Supply the evaluation board through the Vin connector (J2) with a DC voltage within the operative range (8 V - 45 V)
- A new drive named “SPIN32F0” should appear in the list of removable storages
- Delete all the files on this drive (for example “DETAILS.TXT” file). This step is not mandatory but it is recommended when the binary size is close to the 32 Kbytes maximum limit of the STSPIN32F0 (this is the case of the example binaries). Otherwise, the firmware loading can fail.
- Copy the binary file into the root of the drive. The red/green LED of the ST-LINK starts blinking. Then refresh your file explorer: if the binary file has disappeared and no error log file has been generated, the binary file has been successfully loaded.
- Reset the board to start the loaded firmware.

3.3 Building and loading firmware with a different parameters configuration

It is possible to develop the firmware via one of the supported different IDEs:

- **µVision** of the ARM Keil Microcontroller Development Kit (MDK-ARM) toolchain;
- **IAR** embedded workbench for the ARM (EWARM) toolchain which is provided by IAR Systems®;
- **System workbench for the STM32**, a GCC toolchain based on Eclipse and provided by AC6;
- **Atollic TrueSTUDIO for STM32**, a development and debugging IDE for STM32 MCU developers.

Note: The firmware must be rebuilt every time the configuration files of the FOC library are changed as, for example, the characteristics of the target motor or the supply voltage of the application is changed.

STSW-SPIN3201 is a customization of the STM32 PMSM FOC library, where the STEVAL-SPIN3201 board is associated to the Bull-Running BR2804-1700 kV motor.

The customization consists of the introduction of some functions that permit the use of the start/stop button in the STEVAL-SPIN3201, the use of the potentiometer and the Fault Acknowledgement:

```
/* USER CODE BEGIN PFP */
/* Private function prototypes -----------------------------
#if defined (ENABLE_START_STOP_BUTTON)
void StartStoButtonGPIOInit (void)
#endif
void FaultLEDInit (void)
void PotentiometerInit (void)
void PotentiometerHandleSpeed (void)
int16_t PotentiometerGetAverageSpeed (int16_t adcconv);
/* USER CODE END PFP */
```

These functions can be found in the main.c file and are included between comment pairs USER CODE BEGIN and USER CODE END. This portion of the file is not overwritten by STM32CubeMX when it generates the code.

Also the external interrupt handler in file stm32f0xx_mc_it.c has been changed.

```
void EXIIO_1_IRQHandler (void)
{
/* USER CODE BEGIN START_STOP_BTN */
LL_EXTI_ClearFlag_0_31 (LL_EXTI_LINE_0) ;
#define CLEAR_ERROR_WIIH_SIART_STOP_BUTTON
MC_AcknowledgeFaultMotor1 () ;
UI_HandleStartStopButton_cb () ;
/* USER CODE END START_STOP_BTN */
}
/* USER CODE BEGIN 1 */
#if defined (ENABIE_START_STOP_BUTTON) &&!defined (CLEAR_ERROR_WITH_START_STOP_BUTTON)
#warning "See following comments regarding start/stop button handling"
/* If a fault occured, the function UI_HandleStartStopButton_cb does not start
 the motor. To prevent this, it is recommended to acknovledge first the fault
 by adding the two following lines in EXTI0_1_IRQHandler:
 #define CLEAR_ERROR_WITH_START_STOP_BUTTON
 MC_AcknovledgeFaultMotor1 () ;
just after line LL_EXTI_ClearFlag_0_31 (LL_EXTI_LINE_0);
*/
#endif
```

Also in this part of the code, the modifications are included between comment pairs USER CODE BEGIN and USER CODE END.

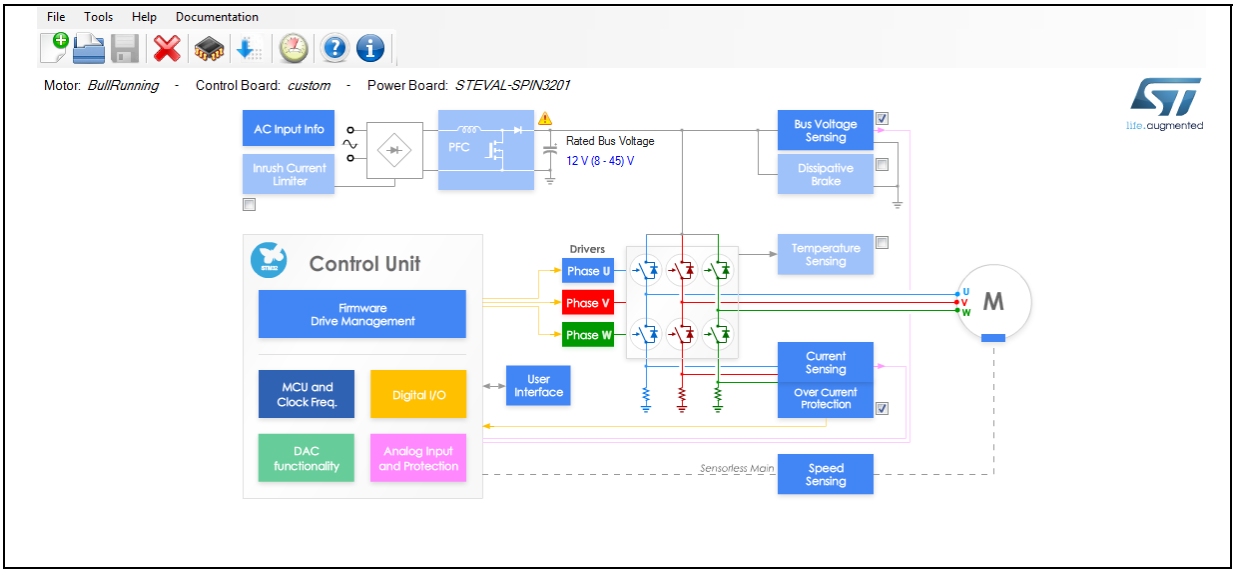
If the user wants to use the features available in STSW-SPIN01 in a different application, a configuration of parameters is needed. The STM32 MC Workbench permits to create a new library with different parameters.

3.3.1 Parameter configuration with the STM32 MC workbench

The STM32 MC Workbench can be used to set the parameters of the FOC library depending on your motor and application.

As long as the STEVAL-SPIN3201 is used, the MC Workbench project file can be used as a starting point: "SDK5x - STEVAL-SPIN3201 - BullRunning.stmcx". When the project is opened by the ST MC Workbench, a configuration window appears (*Figure 4*). Starting from the default parameters, there are 4 main families of parameters to be changed to best fit the application:

Figure 4. ST motor control workbench: configuration window



- Motor Section: If a different motor is used, the respective parameters must be updated. If motor parameters are not correct, it may be useful to change gain parameters. For this purpose, see the documentation of MCSDK available on www.st.com.
- Power Stage Section: They must be changed only if the hardware of the board is modified changing the MOSFETs, adjusting the gain of the sensing circuitry, etc…
- Drive Management Section: They must be used to choose the correct configuration of the algorithm. For example, the start-up configuration of the algorithm, the control mode chosen (Speed or Torque) and also the gain parameters if needed.
- Control Stage Section: Set the mapping of the STM32 peripherals which should not be changed as long as the STEVAL-SPIN3201 board is used.

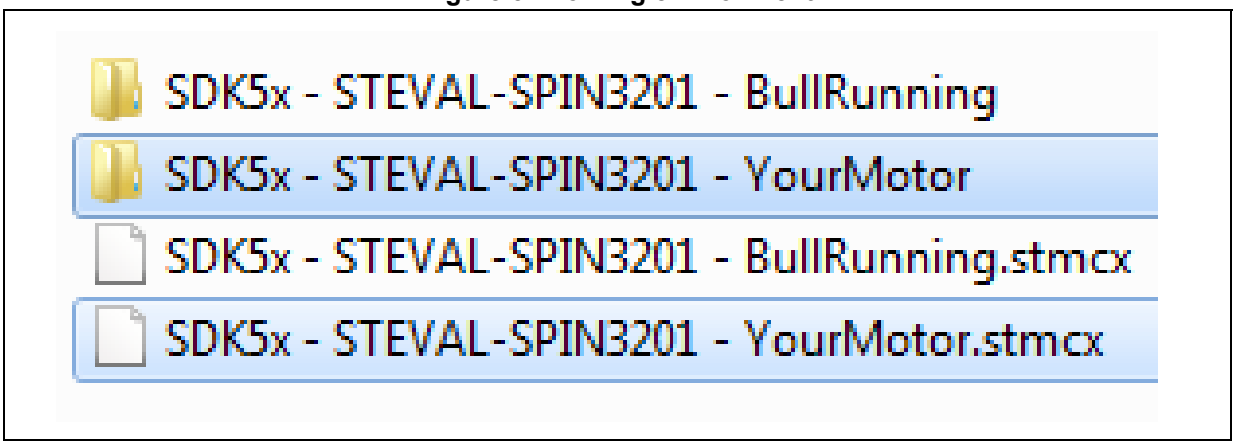
For more details about the use of the STM32 MC Workbench refer to the UM2168 and the documentation that you can find for X-CUBE-MCSDK available on www.st.com.

3.3.2 Generation of library with new parameters configuration

To generate a library with new parameters maintaining the customized code of STSW-SPN01, follow the procedure:

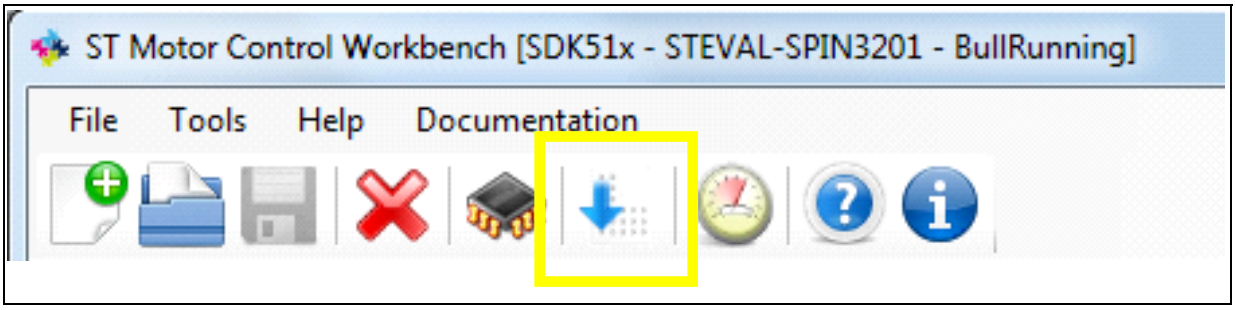
- Open ST Motor Control Workbench and load the project "SDK5x - STEVAL-SPIN3201 - BullRunning.stmcx" contained in the zip file.
- Change the parameters as needed by your application (as shown in *Section 3.3.1*) and save the new project with a different name (e.g. "SDK5x - STEVAL-SPIN3201 - YourMotor.stmcx"). This is the configuration file of parameters used in the STM32 Motor Control Workbench. When the user changes the parameters of the application, this file changes accordingly.
- Create a new folder with the same name of the project file (in the example, the new folder is "SDK5x - STEVAL-SPIN3201 - YourMotor"). You have created the new working environment for your application.

Figure 5. Working environment



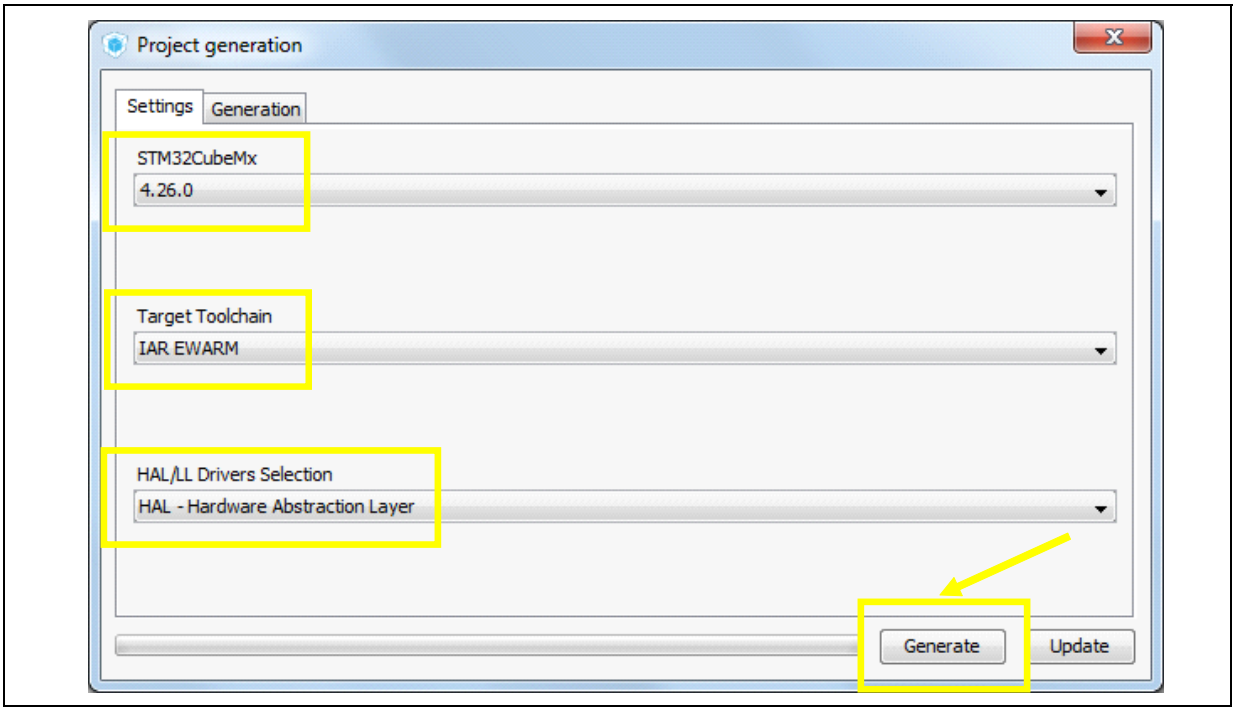
- Copy the "Inc" and "Src" folders from the original directory "SDK5x - STEVAL-SPIN3201 - BullRunning" into the new folder "SDK5x - STEVAL-SPIN3201 - YourMotor".The application main folder "SDK5x - STEVAL-SPIN3201 - YourMotor", has two subfolders.
- Click on generation button to generate the new code as shown in *Figure 6*.

Figure 6. Generation button



A generation window appears as shown in *Figure 7*.

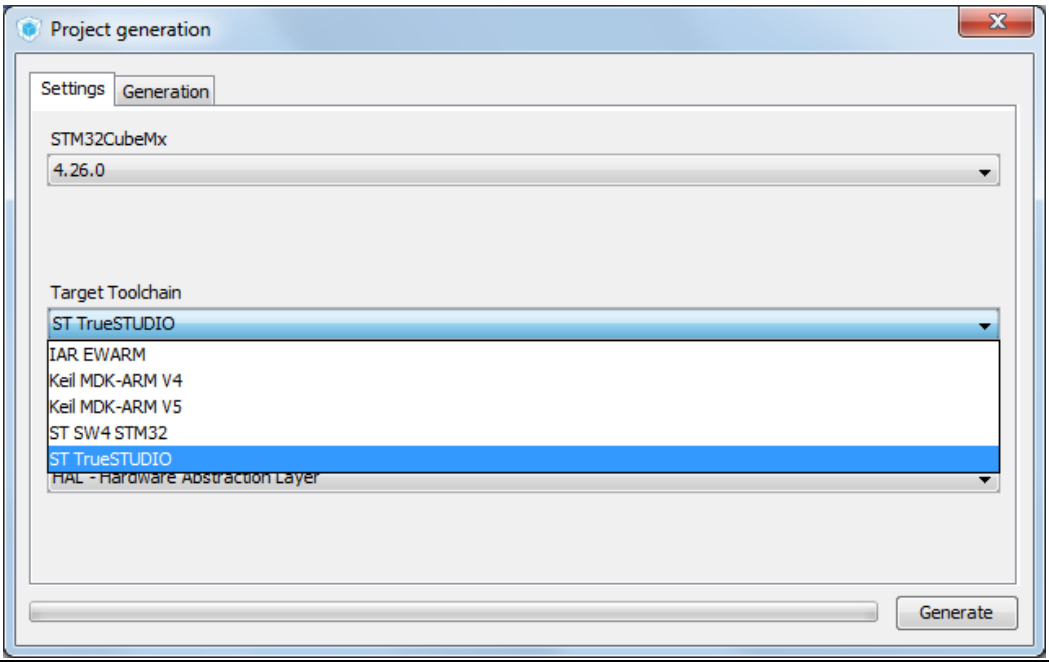
Figure 7. Generation window



In the generation window choose:

- STM32CubeMx should be the latest version;
- Choose your favorite Target Toolchain in the Target Toolchain menu (*Figure 8*);

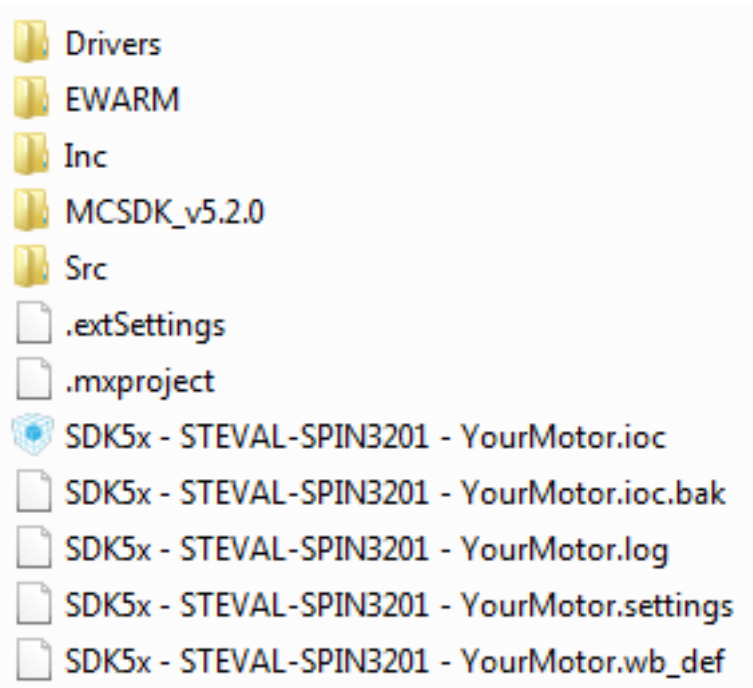
Figure 8. Target toolchain menu



- Choose HAL-Hardware Abstraction Layer in Driver Selection;
- Click on Generate.

The STM32 Motor Control Workbench generates, in the application main folder, the new STM32CubeMx Project file and the new code for your application. The application main folder "SDK5x - STEVAL-SPIN3201 - YourMotor", has these subfolders:

Figure 9. After the code generation



The STM32 MC Workbench has updated the original code contained in the directories "Inc" and "Src" with the application parameters and now, in the folder "SDK5x - STEVAL-SPIN3201 - YourMotor", the new configuration is ready to be used by your favorite IDE.

Note: The directory EWARM is generated when choosing IAR EWARM in the Target Toolchain menu.

3.3.3 Compile and download the firmware

After the generation of the application code, the library is ready to be compiled and downloaded. To build and load STSW-SPIN3201 with the new configuration of parameters, follow the steps:

- Supply the board correctly and connect it to the PC through the USB cable;
- Connect the motor;
- Run your favorite IDE;
- Load the project with your favorite IDE from the releted directory created by the STM32 MC Workbench (for example "EWARM\Project.eww" if you are using IAR embedded workbench for the ARM toolchain);
- Compile and download the code;
- Start the execution of the code.

3.3.4 Control the motor

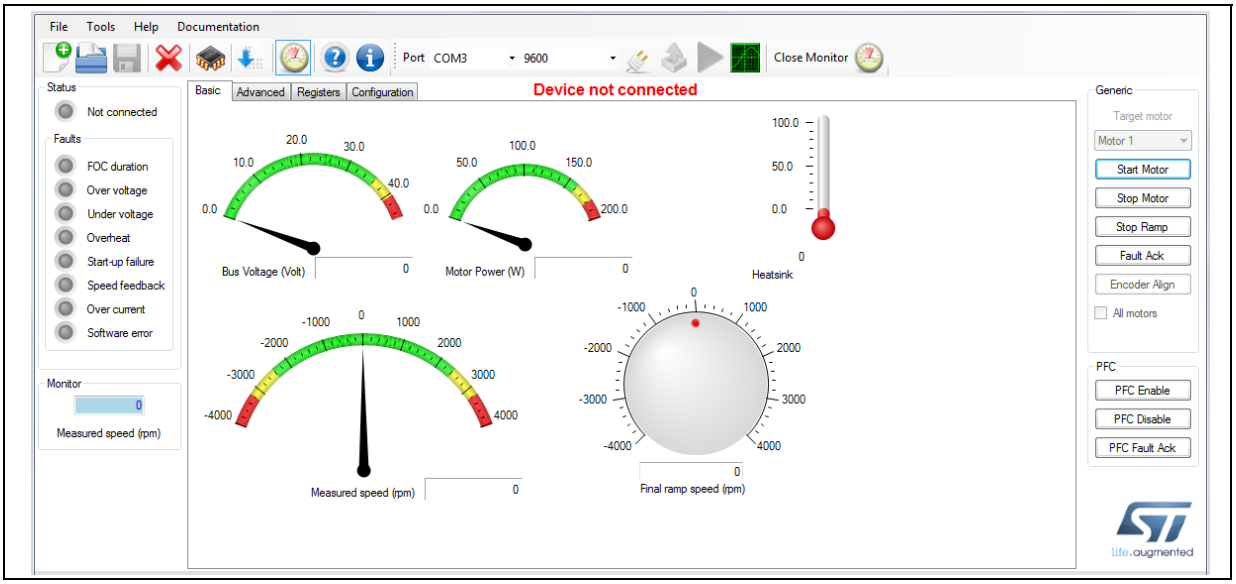
The user can now control the motor using the features available on STSW-SPIN3201. They can push the start/stop button ("USER1" button) to make the motor spin. They can also use potentiometer (R6) to dynamically adjust motor speed. If an error occurs while the motor is running, the LED of the "USER2" key is turned on and the motor stops. Pushing the "USER1" button again clears the failure.

Alternatively the user can also control the motor through the STM32 MC Workbench:

- Open STM32 MC Workbench;
- Click on the "Open Monitor" button:

to open the ‘monitor and spin control GUI’ (*Figure 10*);

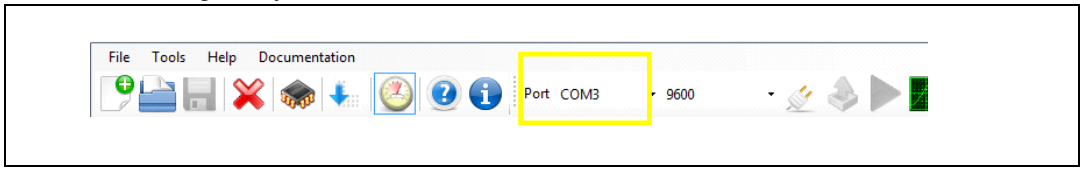
Figure 10. Monitor and spin control GUI



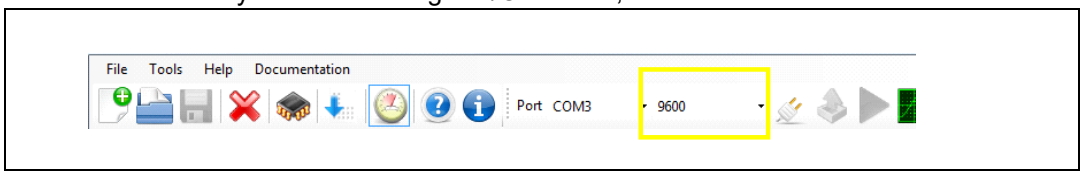
- Connect the board by clicking the connection button:

to make this operation work;

- Remember to use the correct Port number: you can find the correct COM number in the Device Manager of your PC;

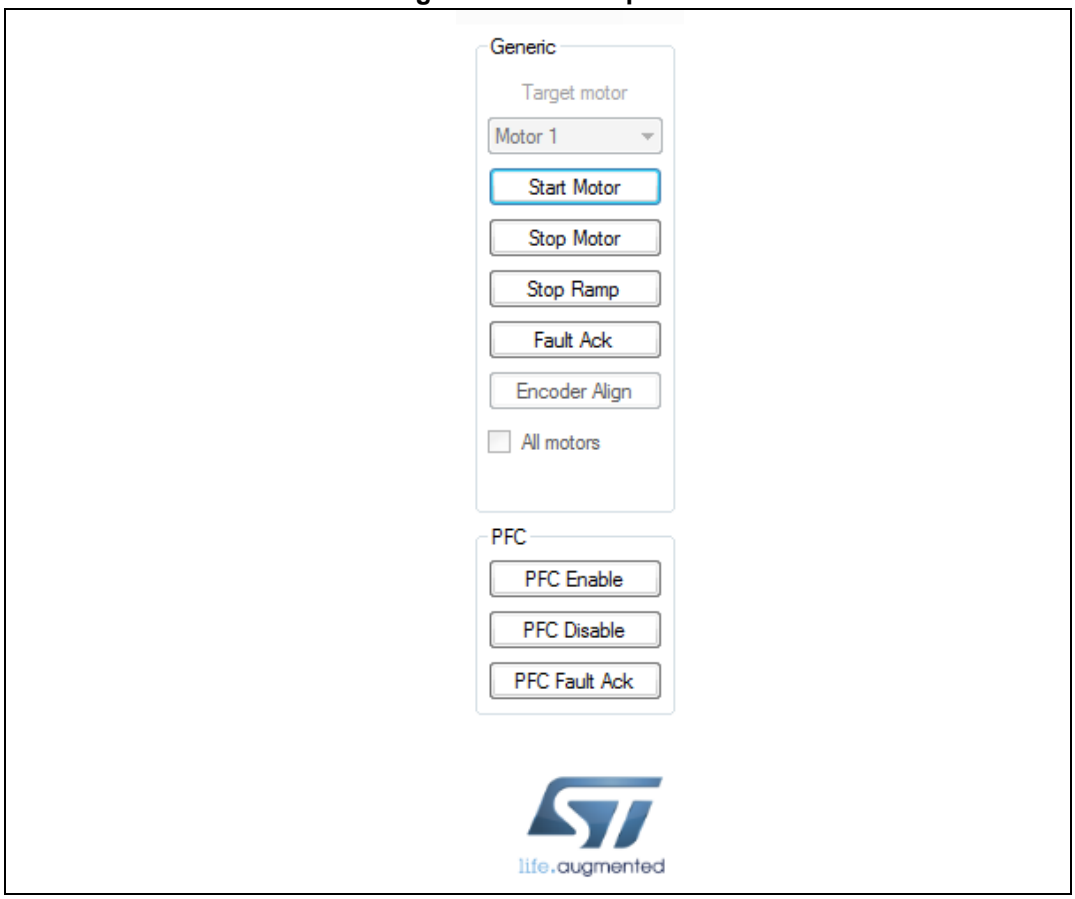


- Use Baudrate 9600 if you are using the original setting of this parameter. It is recommended to use the minimum value of baudrate, because a higher value could reflect on the functionality of the environment. If you have changed this parameter using the STM32 MC Workbench, report in the ‘Monitor and Spin control GUI’ the same value of baudrate that you set in the Digital I/O window;



- Once the board is connected, use the control panel on the right of the control GUI:

Figure 11. Control panel



- Using the STM32 MC Workbench it is also possible to monitor the speed and change the variables of the algorithm in real time. The "Advanced" and "Registers" tabs can be used to dynamically adjust the parameters of the FOC algorithm.

If an error occurs, the respective LED on the left side is turned on. No operation is allowed until the failure is cleared using the "Fault Ack" button.

For details of the use of the STM32 Motor Control Workbench v5, see also UM2374 and UM2380 available on www.st.com.

4 Revision history

Table 2. Document revision history

Date	Revision	Changes
05-Dec-2016	1	Initial release.
22-Oct-2018	2	Update to Revision 5.2.0 of STM32 Motor Control SDK

IMPORTANT NOTICE – PLEASE READ CAREFULLY

STMicroelectronics NV and its subsidiaries (“ST”) reserve the right to make changes, corrections, enhancements, modifications, and improvements to ST products and/or to this document at any time without notice. Purchasers should obtain the latest relevant information on ST products before placing orders. ST products are sold pursuant to ST’s terms and conditions of sale in place at the time of order acknowledgement.

Purchasers are solely responsible for the choice, selection, and use of ST products and ST assumes no liability for application assistance or the design of Purchasers’ products.

No license, express or implied, to any intellectual property right is granted by ST herein.

Resale of ST products with provisions different from the information set forth herein shall void any warranty granted by ST for such product.

ST and the ST logo are trademarks of ST. All other product or service names are the property of their respective owners.

Information in this document supersedes and replaces information previously supplied in any prior versions of this document.

© 2018 STMicroelectronics – All rights reserved