UM2106
User manual

Getting started with the X-CUBE-SPN11, low voltage three-phase brushless DC motor driver software expansion for STM32Cube

Introduction

The X-CUBE-SPN11 is an expansion software for STM32Cube. The software runs on the STM32 Nucleo providing management of STSPIN230 to control low voltage three-phase brushless DC motors. The expansion is built on STM32Cube software technology to ease portability across different STM32 microcontrollers.

The software comes with a sample implementation driving a low voltage three-phase brushless DC motor, with BEMF sensing. It is preconfigured for a NUCLEO-F401RE board connected to an X-NUCLEO-IHM11M1 expansion board, but can be easily ported across different MCU families, thanks to STM32CUBE.

The package contains a user interface layer enabling real-time transmission of data to a PC through the terminal.

Contents

List of tables

List of figures

1 Acronyms and abbreviations

Table 1: List of acronyms

Acronym	Description
API	application programming interface
BSP	board support package
CMSIS	Cortex® microcontroller software interface standard
HAL	hardware abstraction layer
IDE	integrated development environment
LED	light emitting diode
BLDC motor	brushless DC motor
BEMF	back electromotive force

2 What is STM32Cube?

STMCube™ represents the STMicroelectronics initiative to make developers' lives easier by reducing development effort, time and cost. STM32Cube covers the STM32 portfolio.

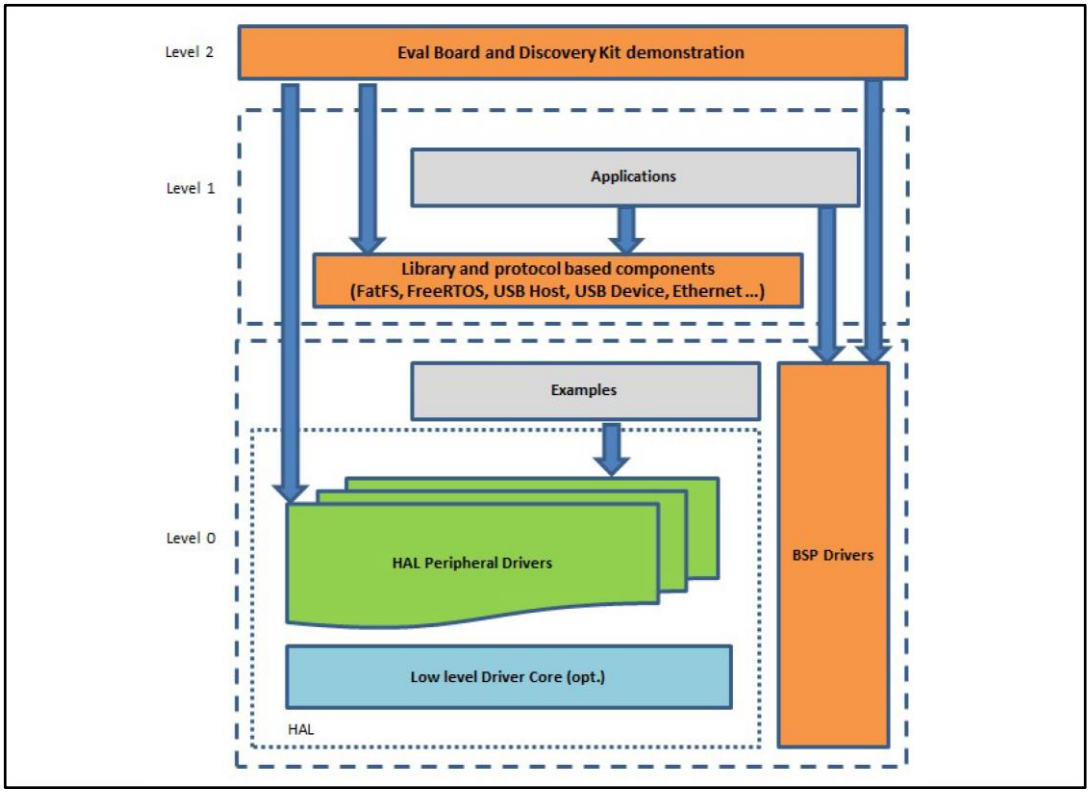
STM32Cube version 1.x includes:

- STM32CubeMX, a graphical software configuration tool that allows the generation of C initialization code using graphical wizards.
- A comprehensive embedded software platform specific to each series (such as the STM32CubeF4 for the STM32F4 series), which includes:
  - the STM32Cube HAL embedded abstraction-layer software, ensuring maximized portability across the STM32 portfolio
  - a consistent set of middleware components such as RTOS, USB, TCP/IP and graphics
  - all embedded software utilities with a full set of examples

2.1 STM32Cube architecture

The STM32Cube firmware solution is built around three independent levels that can easily interact with one another, as described in the diagram below.

Figure 1: Firmware architecture



Level 0: This level is divided into three sub-layers:

- Board Support Package (BSP): this layer offers a set of APIs relative to the hardware components in the hardware boards (Audio codec, IO expander, Touchscreen, SRAM driver, LCD drivers. etc…); it is based on modular architecture allowing it to be easily

ported on any hardware by just implementing the low level routines. It is composed of two parts:

  - Component: is the driver relative to the external device on the board and not related to the STM32, the component driver provides specific APIs to the external components of the BSP driver, and can be ported on any other board.
  - BSP driver: links the component driver to a specific board and provides a set of easy to use APIs. The API naming convention is BSP_FUNCT_Action(): e.g., BSP_LED_Init(), BSP_LED_On().

- Hardware Abstraction Layer (HAL): this layer provides the low level drivers and the hardware interfacing methods to interact with the upper layers (application, libraries and stacks). It provides generic, multi-instance and function-oriented APIs to help offload user application development time by providing ready to use processes. For example, for the communication peripherals (I²C, UART, etc.) it provides APIs for peripheral initialization and configuration, data transfer management based on polling, interrupt or DMA processes, and communication error management. The HAL Drivers APIs are split in two categories: generic APIs providing common, generic functions to all the STM32 series and extension APIs which provide special, customized functions for a specific family or a specific part number.
- Basic peripheral usage examples: this layer houses the examples built around the STM32 peripherals using the HAL and BSP resources only.

Level 1: This level is divided into two sub-layers:

- Middleware components: set of libraries covering USB Host and Device Libraries, STemWin, FreeRTOS, FatFS, LwIP, and PolarSSL. Horizontal interaction among the components in this layer is performed directly by calling the feature APIs, while vertical interaction with low-level drivers is managed by specific callbacks and static macros implemented in the library system call interface. For example, FatFs implements the disk I/O driver to access a microSD drive or USB Mass Storage Class.
- Examples based on the middleware components: each middleware component comes with one or more examples (or applications) showing how to use it. Integration examples that use several middleware components are provided as well.

Level 2: This level is a single layer with a global, real-time and graphical demonstration based on the middleware service layer, the low level abstraction layer and basic peripheral usage applications for board-based functions.

3 X-CUBE-SPN11 software expansion for STM32Cube

3.1 Overview

The X-CUBE-SPN11 software package expands STM32Cube functionality, and features:

- Sample application to drive a low voltage three phase brushless motor, managing an X-NUCLEO-IHM11M1 board on top of a NUCLEO-F401RE board
- Timer to generate step clock and voltage reference
- Management of parameters like minimum and maximum speed, direction etc.
- GPIO, PWM and IRQ configuration
- API function available to send any application command to the motor driver
- User interface utility based on PC terminal to control the motor
- Speed control through potentiometer
- Motor control by user button
- Easy portability across different MCU families, thanks to STM32Cube
- Free, user-friendly license terms

To use the STSPIN230 driver library, first call its initialization function to:

- set up the required GPIOs to handle the bridges, enable the EN\FAULT and the STBY\RESET pins
- set up the drivers
- load the driver parameters with the values in “MC_SixStep_param.h” (acceleration, deceleration, minimum and maximum speed, starting Duty Cycle, PWM frequency, etc.).

You can easily modify the driver parameters in “MC_SixStep_param.h”. The speed profile is completely handled by the microcontroller depending on the potentiometer setting. The 6-step algorithm is based on the 1-shunt current sensing mode and the sensorless algorithm for BEMF detection.

Motor activity is commanded differently, depending on workspace configuration:

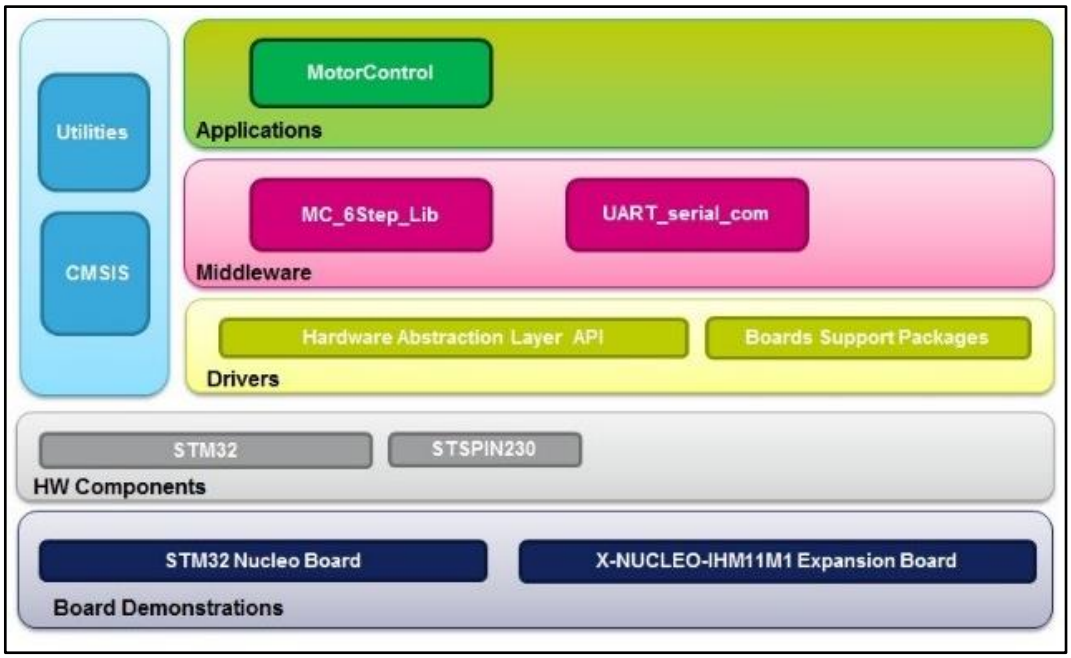
- Configuration mode waits for the blue button event to start the motor and the black button to reset the firmware
- comm mode enables communication via an external PC terminal
- demo mode starts and stop the motor automatically

A motion command can be stopped at any moment by pushing the user button, via the external PC terminal, or automatically. The firmware can be reset by pushing the reset button on the STM32 Nucleo development board.

3.2 Architecture

This fully compliant STM32Cube software expansion enables development of BLDC applications using motor drivers.

Figure 2: X-CUBE-SPN11 architecture



The software is based on the STM32CubeHAL hardware abstraction layer for the STM32 microcontroller and extends STM32Cube with a board support package (BSP) for the DC motor driver expansion board (X-NUCLEO-IHM11M1) and some middleware components for motor control and serial communication with a PC.

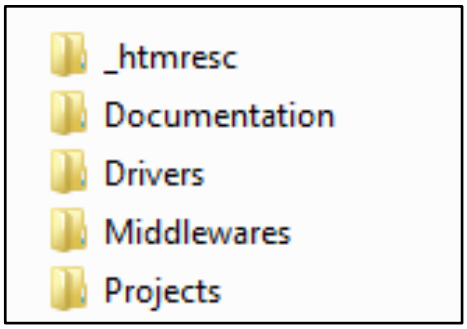
The software layers used by the application software to access and use the three-phase brushless DC motor drivers expansion board are:

STM32Cube HAL layer: provides a generic, multi-instance set of simple APIs (application programming interfaces) to interact with the upper layers (application, libraries and stacks). These generic and extension APIs are based on a common architecture and the layers above them like the middleware layer can function without requiring specific hardware configuration data for a given microcontroller unit (MCU). This structure improves the library code reusability and guarantees easy portability across other devices.

Board support package (BSP) layer: supports the peripherals on the STM32 Nucleo board, except for the MCU. This limited set of APIs provides a programming interface for certain board specific peripherals like the user button, the reset button, etc, and helps in identifying the specific board version. The BSP in the X-CUBE-SPN11 software provides the drivers to manage the STSPIN230 motor driver.

3.3 Folders structure

Figure 3: X-CUBE-SPN11 package folder structure



The software is packaged in the following main folders:

- **Documentation**: with a compiled HTML file generated from the source code that details the software components and APIs.
- **Drivers**:
  - STM32Cube HAL driver files located in the STM32F4xx_HAL_Driver subdirectories. These files are not described here as they derive directly from the STM32Cube framework. Only those files required to run the motor driver examples are present.
  - CMSIS folder with the CMSIS (Cortex® microcontroller software interface standard) files from ARM. These files form a vendor-independent hardware abstraction layer for the Cortex-M processor series. This folder is also unchanged from the STM32Cube framework.
  - BSP folder with code files required to the STSPIN230 driver and the motor control API. The STSPIN230.c file implements all the API functions to manage the motor driver: start/stop PWM signals, enable/disable each inverter leg, etc.
- **Projects**: contains a sample MC application using the NUCLEO-F401RE platform.
- **Middlewares**: contains MC libraries and the serial driver-PC communication protocol. The middleware folder includes the core 6-step sensorless algorithm with BEMF detection, able to drive a generic three-phase brushless DC motor with speed control loop. The respective header file contains the data structure and the list of main APIs for the MC.

3.4 APIs

Detailed function and parameter descriptions for the user-APIs are compiled in an HTML file in the software package Documentation folder.

3.5 Sample application description

A sample application using the X-NUCLEO-IHM11M1 expansion board with NUCLEO-F401RE board is provided in the Projects directory. Ready to be built projects are available for multiple IDEs.

In this application, a low voltage three-phase brushless motor is driven by the STSPIN230 device. The firmware implements a 6-Step sensorless algorithm.

The main functions are:

```
MC_SixStep_INIT();

MC_StartMotor();

MC_StopMotor();

MC_SetSpeed(value);
```

4 System setup guide

4.1 Hardware description

This section describes the hardware components which are required to execute the X-CUBE-SPN11 software and successfully drive a low voltage three-phase brushless motor.

4.1.1 STM32 Nucleo platform

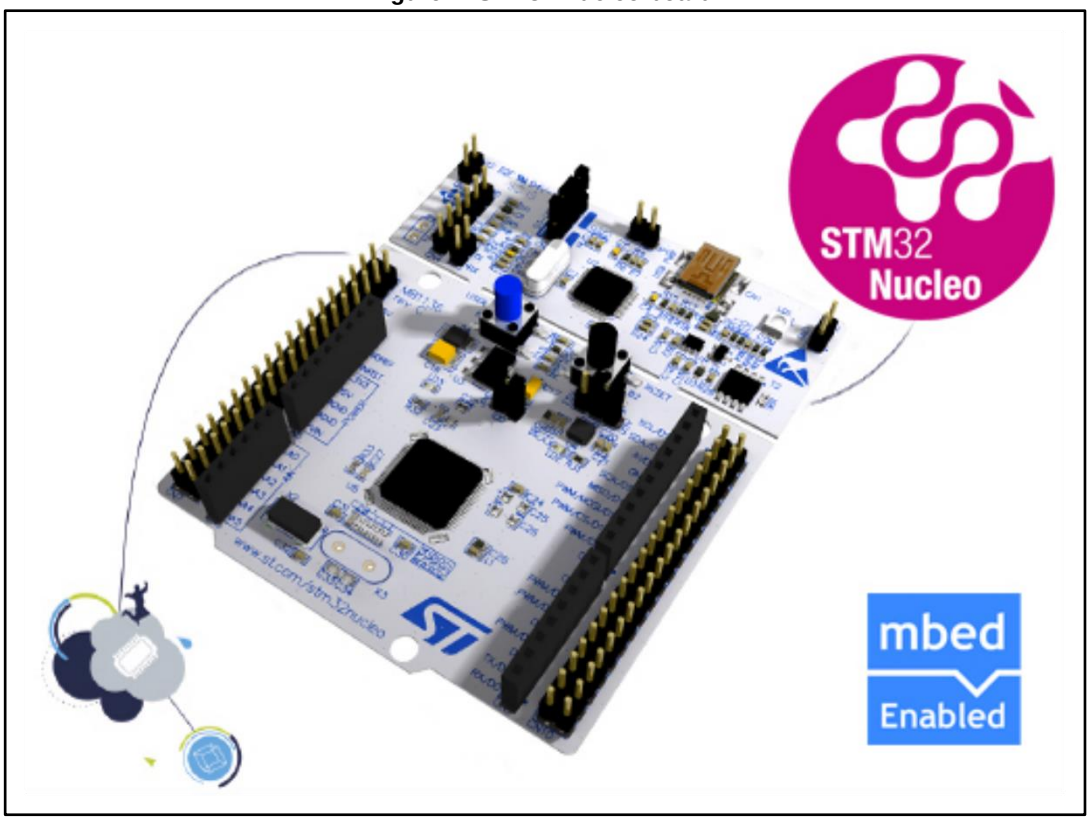
STM32 Nucleo development boards provide an affordable and flexible way for users to test solutions and build prototypes with any STM32 microcontroller line.

The Arduino™ connectivity support and ST morpho connectors make it easy to expand the functionality of the STM32 Nucleo open development platform with a wide range of specialized expansion boards to choose from.

The STM32 Nucleo board does not require separate probes as it integrates the ST-LINK/V2-1 debugger/programmer.

The STM32 Nucleo board comes with the comprehensive STM32 software HAL library together with various packaged software examples.

Figure 4: STM32 Nucleo board



Information regarding the STM32 Nucleo board is available at *www.st.com/stm32nucleo*

4.1.2 X-NUCLEO-IHM11M1 low voltage three-phase brushless DC motor driver expansion board

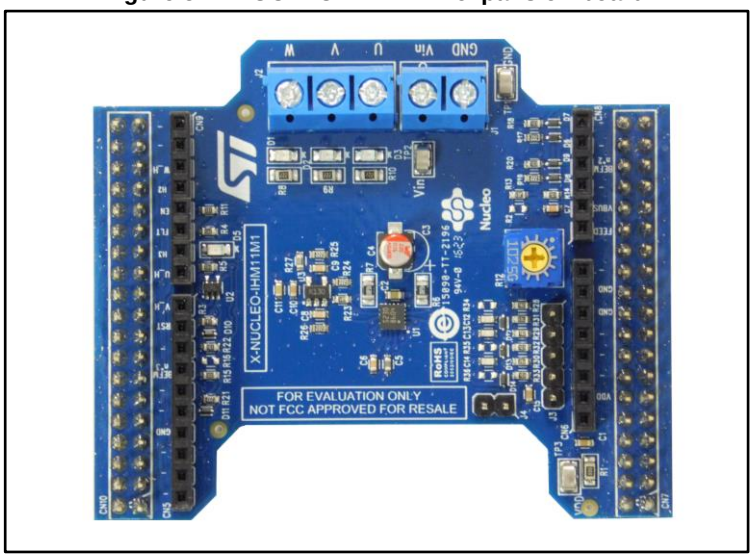
The X-NUCLEO-IHM11M1 is a low voltage, three-phase, brushless DC motor driver expansion board based on the STSPIN230 for STM32 Nucleo.

It provides an affordable and easy-to-use solution for the implementation of portable motor driving applications such as thermal printers, robotics and toys.

The X-NUCLEO-IHM11M1 is compatible with the Arduino UNO R3 connector and most STM32 Nucleo boards.

The board is designed for six-step and FOC algorithms.

Figure 5: X-NUCLEO-IHM11M1 expansion board



Information about the X-NUCLEO-IHM11M1 expansion board is available on www.st.com

4.1.3 Miscellaneous hardware components

To complete the hardware setup, you will need:

- an STM32 Nucleo development board (NUCLEO-F401RE)
- a three-phase low voltage brushless DC motor
- an external DC power supply with two electric cables
- a USB type A to mini-B USB cable to connect the STM32 Nucleo to a PC
- a Windows™ PC (XP, Win 7, Win 8) – Laptop/PC

4.2 Software description

The following software components are needed for a suitable development environment for applications based on the motor driver expansion board:

- ST-LINK/V2-1 USB driver
- ST-LINK/V2-1 firmware upgrade
- X-CUBE-SPN11 expansion for STM32Cube dedicated to STSPIN230 low voltage, 3 phase, brushless motor driver applications development. The X-CUBE-SPN11 firmware and related documentation is available on www.st.com.
- One of the following development tool-chain and compilers:
  - Keil RealView Microcontroller Development Kit (MDK-ARM) toolchain V5.21a
  - IAR Embedded Workbench for ARM (EWARM) toolchain V7.40

– OpenSTM32 System Workbench for STM32 (SW4STM32) V1.3

4.3 Hardware and software setup

This section describes the hardware and software setup procedure for executing the provided examples and to develop new applications based on the motor driver expansion board.

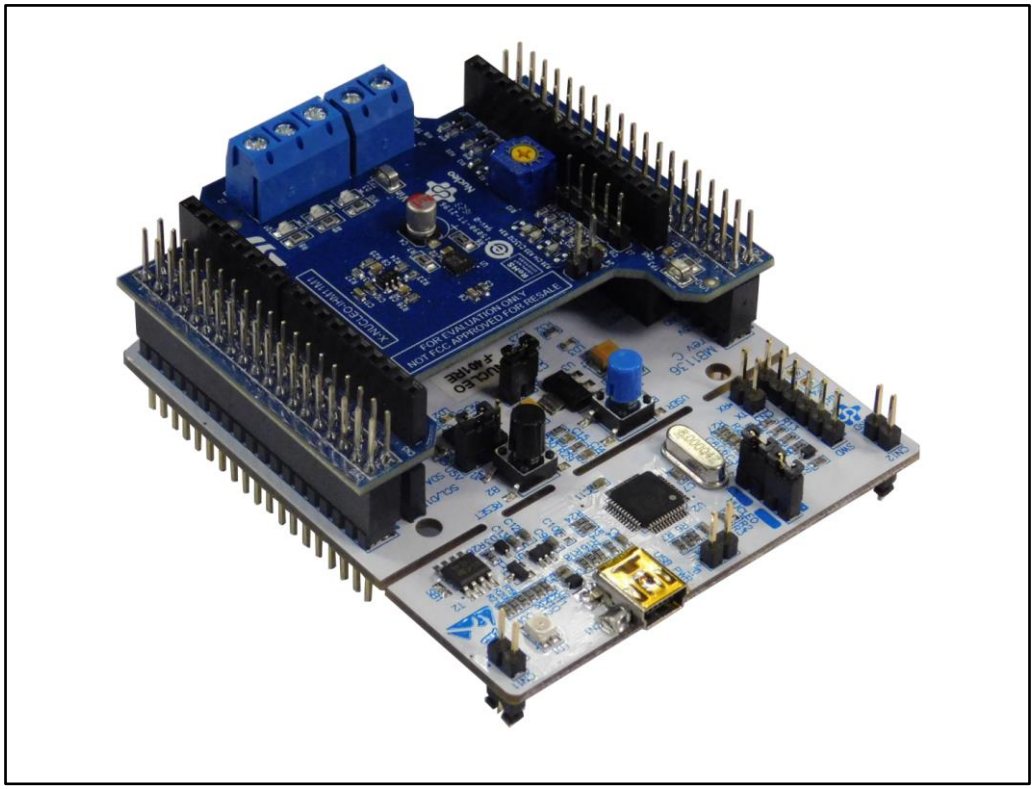
4.3.1 Setup to drive a three-phase brushless motor

1 Configure your NUCLEO-F401RE STM32 Nucleo development board.
JP1 off JP5 (PWR) on UV5 side JP6 (IDD) on

2 A STM32 Nucleo expansion board (X-NUCLEO-IHM11M1) with the following configuration:
Tune R12 potentiometer. J3 and J4 open

3 Plug the X-NUCLEO-IHM11M1 expansion board onto the ST morpho connectors on the STM32 Nucleo board.

Figure 6: STM32 Nucleo board plus X-Nucleo-IHM11M1 expansion board



4 Connect the STM32 Nucleo development board to a PC with the USB cable through USB connector CN1 to power the board

5 Power on the X-NUCLEO-IHM11M1 expansion board by connecting its connectors Vin and Gnd to the DC power supply. The DC supply must be set to deliver the required voltage to the three-phase brushless motor.

6 Connect the three-phase of motor to the X-NUCLEO-IHM11M1 bridge connectors U,V,W.

7 Open your preferred toolchain (MDK-ARM from Keil, EWARM from IAR, or

SW4STM32 from www.openstm32.org)

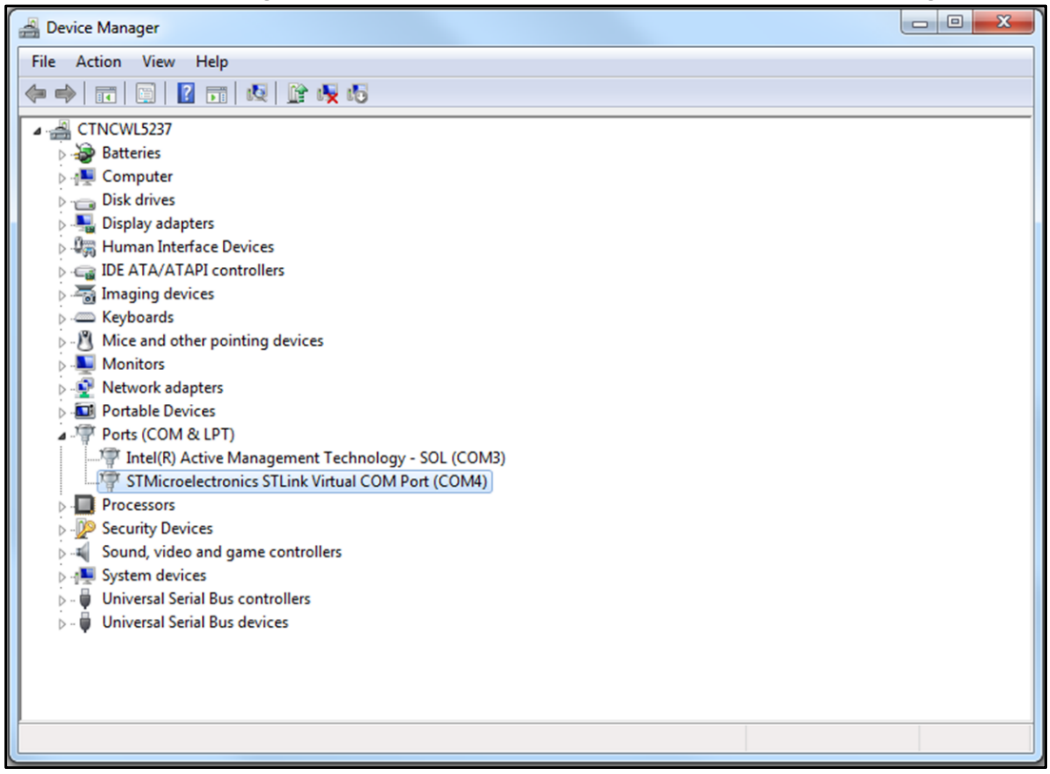
8. Open the software project in \Projects\Multi\Applications\MotorControl\ YourToolChainName\STM32F401RE-Nucleo
9. Adapt the default parameters used by the STSPIN230 to your low voltage three phase brushless DC motor characteristics by modifying the parameters in \Projects\Multi\Applications\MotorControl\Inc\MC_SixStep_param.h. MC_SixStep_param.h contains all the motor driving parameters like the number of pole pairs, target speed, PI parameters for speed regulation, etc.
10. Rebuild all files and load your image \Projects\Multi\Applications\MotorControl\Binary\STM32F401RE-Nucleo\STM32F401RE-Nucleo-YourMode.bin into target memory.
11. Run the sample application.
12. Push the blue button to start and the black button to stop the motor.
13. Push the black button to reset the firmware.
14. You can also use a user interface utility based on PC terminal to control the motor (see *Section 6.3.2: "PC utility "*)

4.3.2 PC utility

The X-CUBE-SPN11 expansion for STM32Cube implements serial communication between the STM32 Nucleo board UART peripheral and a Windows PC-terminal. This utility is available by recompiling the project in COMM mode and configuring a terminal emulation software supporting COM connection. Before using this utility, ensure that the necessary drivers are installed and the expansion board along with STM32 Nucleo board is connected to a PC.

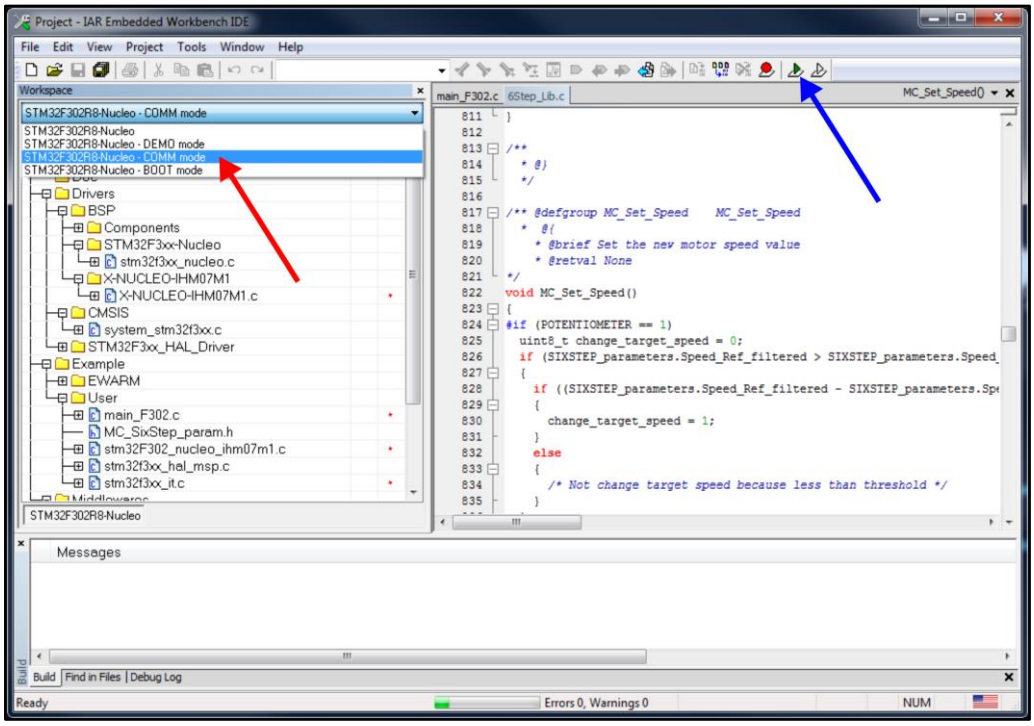
1. Ensure you have the minimum requirements
   - PC with Intel or AMD/INTEL processor running one of following Microsoft operating systems:
     - Windows XP SP3
     - Windows Vista
     - Windows 7
   - At least 128 MBs of RAM
   - 1 x USB port
   - Terminal emulation software (e.g. Hyper-terminal, PuTTY, Tera Term...).
2. Find the ST COM port number in Device Manager (COM4 in the example below).

Figure 7: Finding the STlink COM Port number in Windows™ Device Manager



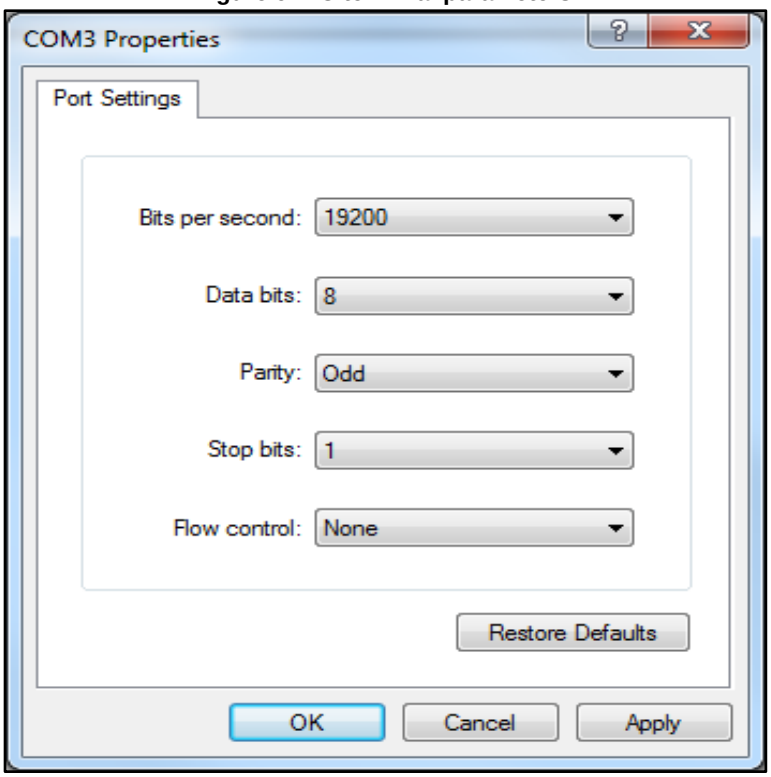
3 Open an IDE like IAR Workbench and open the EWARM project. On the project configuration tool selector, select COMM mode and upload the firmware to the STM32 Nucleo board.

Figure 8: IAR Workbench - compiling in COMM mode and uploading firmware



4 Launch the terminal emulation software on the PC and check that the COM Device number for the current STM32 expansion board is correct and set the parameters as shown below.

Figure 9: PC terminal parameters



Once connection is established, a list of commands is shown.

5 Revision history

Table 2: Document revision history

Date	Version	Changes
25-Aug-2016	1	Initial release.
20-Apr-2017	2	Update: *Section 5.2: "Architecture"*

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.