UM1992
User manual

Getting started with the X-CUBE-SPN8, low-voltage BLDC motor driver software expansion for STM32Cube

Introduction

This document describes how to get started with the X-CUBE-SPN8 software expansion for STM32Cube.

X-CUBE-SPN8 provides the complete STM32 firmware for developers to build motor control applications (e.g. BLDC/PMSM motor). It is highly portable across different MCU families thanks to STM32Cube. This package contains a user interface layer enabling the transmission of real-time data to a PC through the terminal.

The software provides sample implementation projects for STM32 Nucleo platforms equipped with the X-NUCLEO-IHM08M1 expansion board, featuring a motor control application with the 6-Step algorithm.

The software is based on STM32Cube technology and expands the STM32Cube range of package solutions.

Contents

List of tables

List of figures

1 Acronyms and abbreviations

Table 1: List of acronyms and abbreviations

Acronym	Description
MC	Motor Control
6-Step	6-step algorithm for motor control
UI	User Interface serial communication
3ph	Three phase motor
FOC	Field Oriented Control

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.

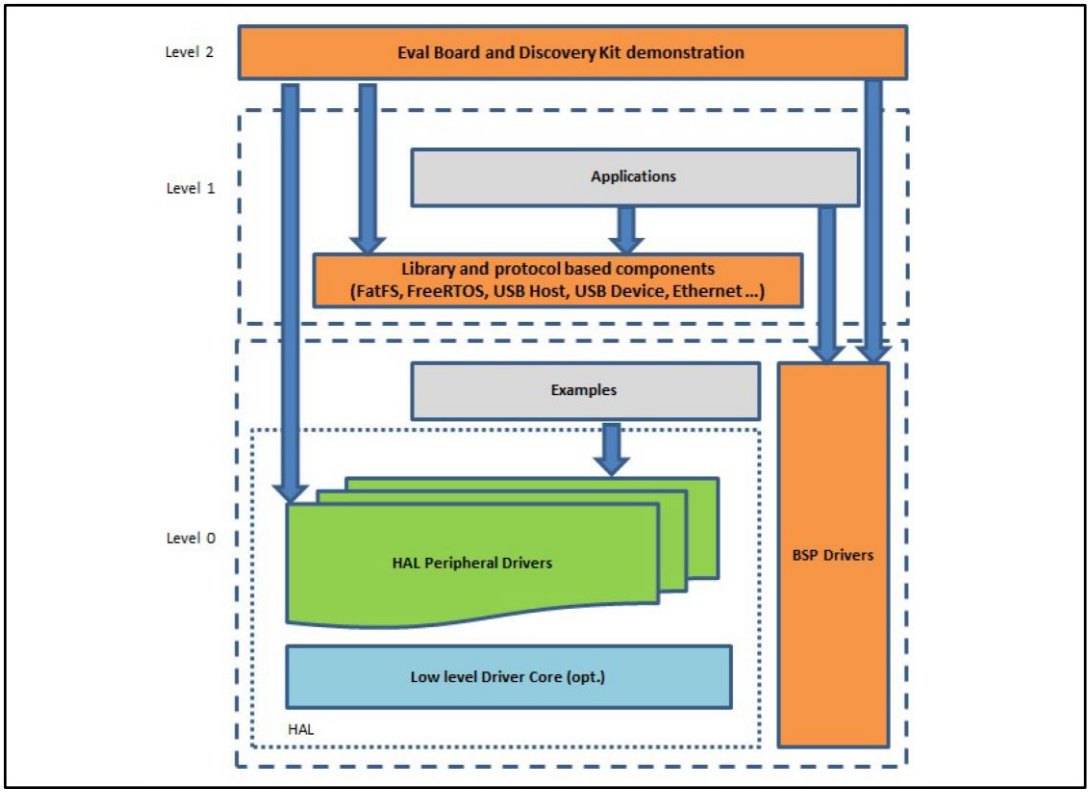
STM32Cube version 1.x includes:

- 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-SPN8 software expansion for STM32Cube

3.1 Overview

X-CUBE-SPN8 is a software package that expands the functionality of STM32Cube.

The key features include:

- complete firmware package to build motor control applications based on discrete solution (STL220N6F7 Power MOSFETs and L6398 gate driver) and hardware expansion board (X-NUCLEO-IHM08M1)
- API function available to send any application command to motor driver.
- easy portability across different MCU families thanks to STM32Cube
- free user-friendly license terms
- sample implementations available on the X-NUCLEO-IHM08M1 plugged onto of a NUCLEO-F302R8 or NUCLEO-F401RE board

This software is based on the STM32CubeHAL hardware abstraction layer for the STM32 microcontroller. The package extends STM32Cube by providing a board support package (BSP) for the STM32 expansion board based on the STL220N6F7 Power MOSFET and L6398 gate driver. The drivers abstract low-level hardware information so the middleware components and applications can fully manage the L6398 through a complete set of APIs which send application commands to the motor driver, in a hardware-independent manner. The package includes an application example to drive one low voltage three-phase BLDC/PMSM motor.

3.2 Architecture

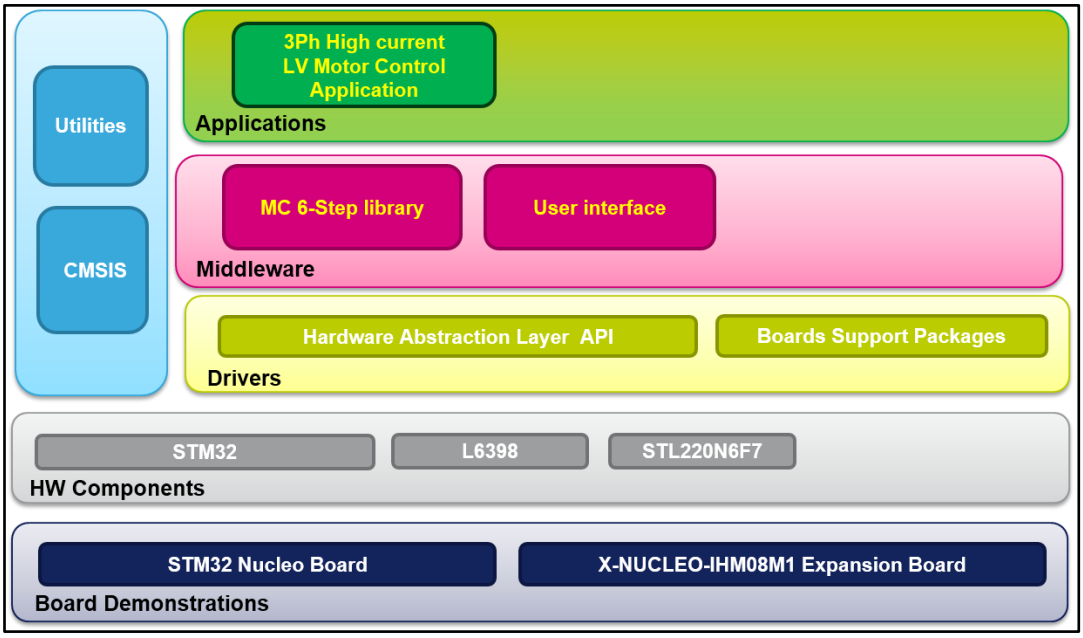
This software is a fully compatible STM32Cube expansion which enables the development of applications using the L6398 gate driver and the STL220N6F7 Power MOSFET. The STM32Cube architecture is detailed elsewhere in this document.

The software is based on the STM32CubeHAL hardware abstraction layer for the STM32 microcontroller, and extends STM32Cube by providing a Board Support Package (BSP) for the low-voltage BLDC motor driver expansion board and some middleware components for serial communication with a PC.

The software layers used by the application software to access and use the motor control expansion board are:

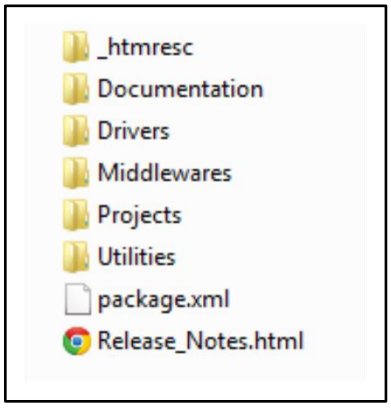
- STM32Cube HAL Layer: The HAL driver layer provides a generic multi instance simple set of APIs (application programming interfaces) to interact with the upper layers (application, libraries and stacks). It is composed of generic and extension APIs. It is directly built around a generic architecture and allows the layers that are built upon, such as the middleware layer, to implement their functionalities without dependencies on the specific hardware configuration for a given Microcontroller Unit (MCU). This structure improves the library code reusability and guarantees an easy portability on other devices.
- Board Support Package (BSP) Layer: The software package needs to support the peripherals on the STM32 Nucleo board apart from the MCU. This software is included in the board support package (BSP). This is a limited set of APIs which provides a programming interface for certain board specific peripherals, e.g. the LED, the user button etc. This interface also helps in identifying the specific board version. In case of MC expansion board, it provides the code to manage the L6398 driver and the X-NUCLEO board.

Figure 2: X-CUBE-SPN8 software architecture



3.3 Folder structure

Figure 3: X-CUBE-SPN8 package folder structure



The following folders are included in the software package:

- Documentation: contains a compiled HTML file generated from the source code, detailing the software components and APIs.
- Drivers: contains the HAL drivers, the board specific drivers for each supported board or hardware platform, including those for the on-board components and the CMSIS vendor-independent hardware abstraction layer for the Cortex-M processor series.
- Middlewares: contains motor control libraries and protocols relating to serial data transmission between the device and a connected PC application.
- Projects: this folder contains a sample motor control application for the NUCLEO-F302R8 or NUCLEO-F401RE platforms under the (IAR Embedded Workbench for ARM, RealView Microcontroller Development Kit (MDK-ARM) and AC6 System Workbench for STM32 development environments.
- Utilities: this folder contains a PC_software folder with Windows PC utilities. In this case, no such utilities are required.

3.4 APIs

Detailed technical information with full API function and parameter descriptions are available in a compiled HTML file in the Documentation folder.

3.5 Sample application description

An sample application using the X-NUCLEO-IHM08M1 expansion board with either NUCLEO-F302R8 or NUCLEO-F401RE boards is provided in the Projects directory. Ready-to-build projects are available for multiple IDEs.

In this application, a low voltage 3-phase motor is driven by the L6398 device. The firmware implements a 6-Step control algorithm and current measurements in single-shunt mode are compared with a reference generated by a digital speed loop for current control.

The main functions involved in managing a motor control application are:

- motor control initialization - MC_SixStep_INIT();
- start motor command - MC_StartMotor();
- stop motor command - MC_StopMotor();
- change motor speed - MC_SetSpeed(value);
- command:

3.6 User Interface utility based on PC-terminal

The X-CUBE-SPN8 expansion for STM32Cube drives serial communication between the STM32 Nucleo board and a Windows PC via a UART peripheral and PC-terminal. A utility is available to recompile the project in COMM mode and configure a terminal with hyper-terminal, for example.

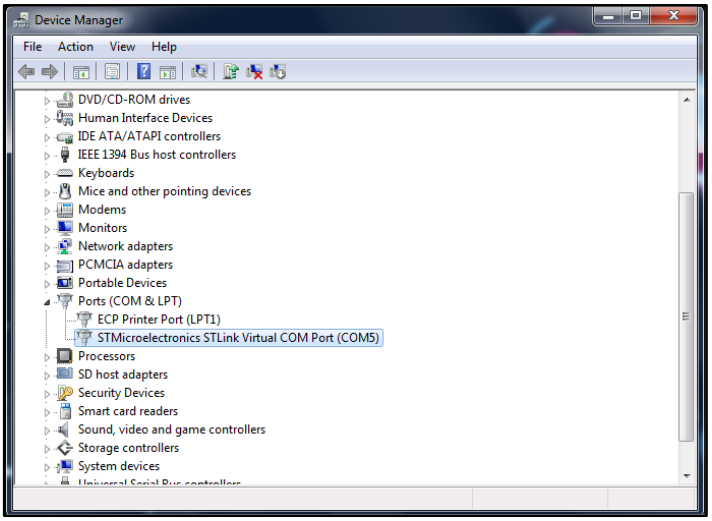
This section describes the procedure to configure the terminal which connects the STM32 Nucleo board to the PC.

Before using this utility, ensure that the necessary drivers are installed and the STM32 Nucleo and expansion board are connected to a PC, as described in the system setup section of this document.

Please follow the steps below.

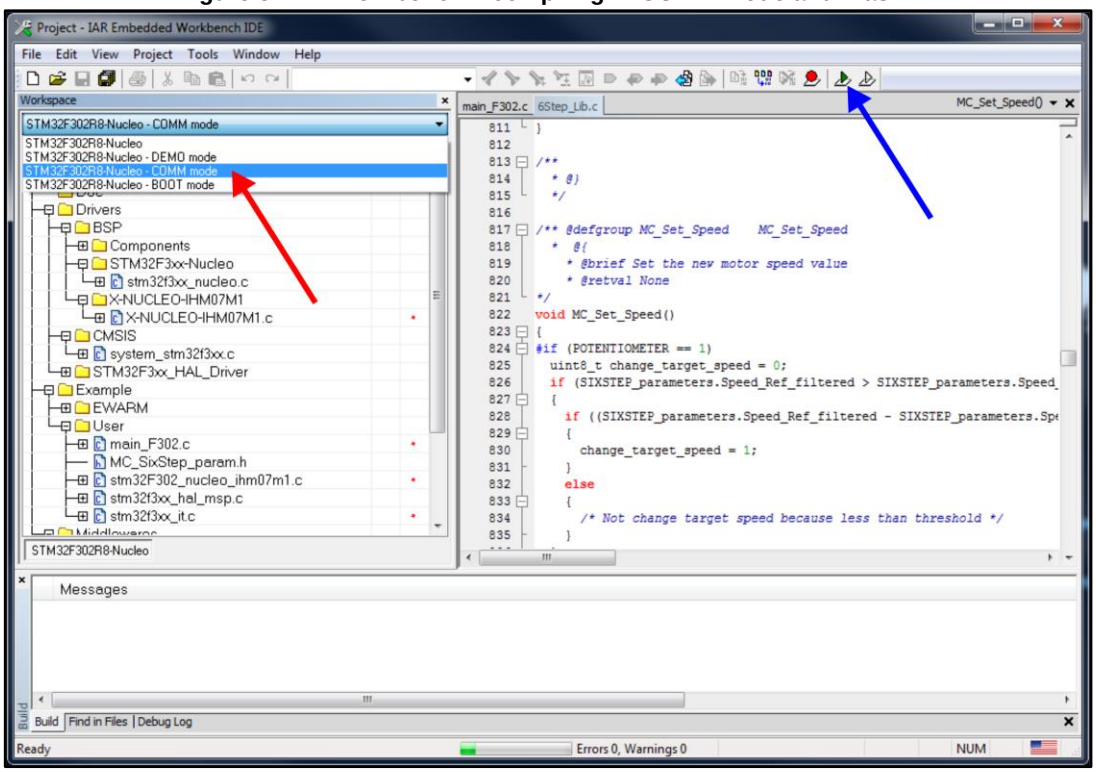
- Check Microsoft Windows Device Manager to determine the ST COM; e.g., COM5 in the figure below.

Figure 4: Device Manager view in Microsoft Windows



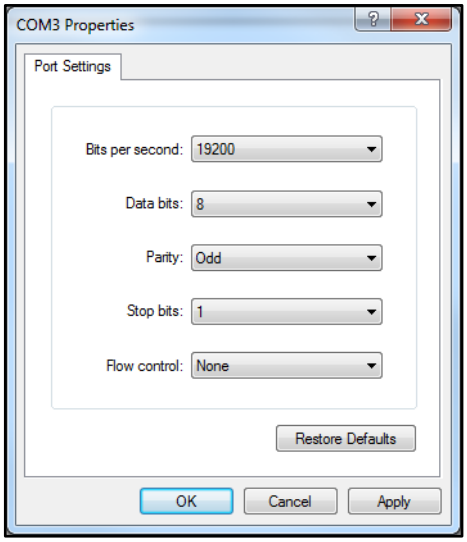
- Open an IDE like IAR Workbench 7.20 and open the EWARM project. In the project configuration tool selector, choose to compile in COMM mode (red arrow in figure below) and upload the firmware (blue arrow) to the STM32 Nucleo board.

Figure 5: IAR Workbench – compiling in COMM mode and Flash



- Launch Hyper-terminal on the PC side and check if the COM Device number for current expansion board is correct and fill in the parameter fields as shown in the figure below.

Figure 6: PC terminal parameters



- Once connection is established, the STM32 expansion board shows a list of available commands.

4 System setup guide

4.1 Hardware description

This section describes the hardware components needed to develop a low-voltage BLDC motor control application.

4.1.1 STM32 Nucleo platform

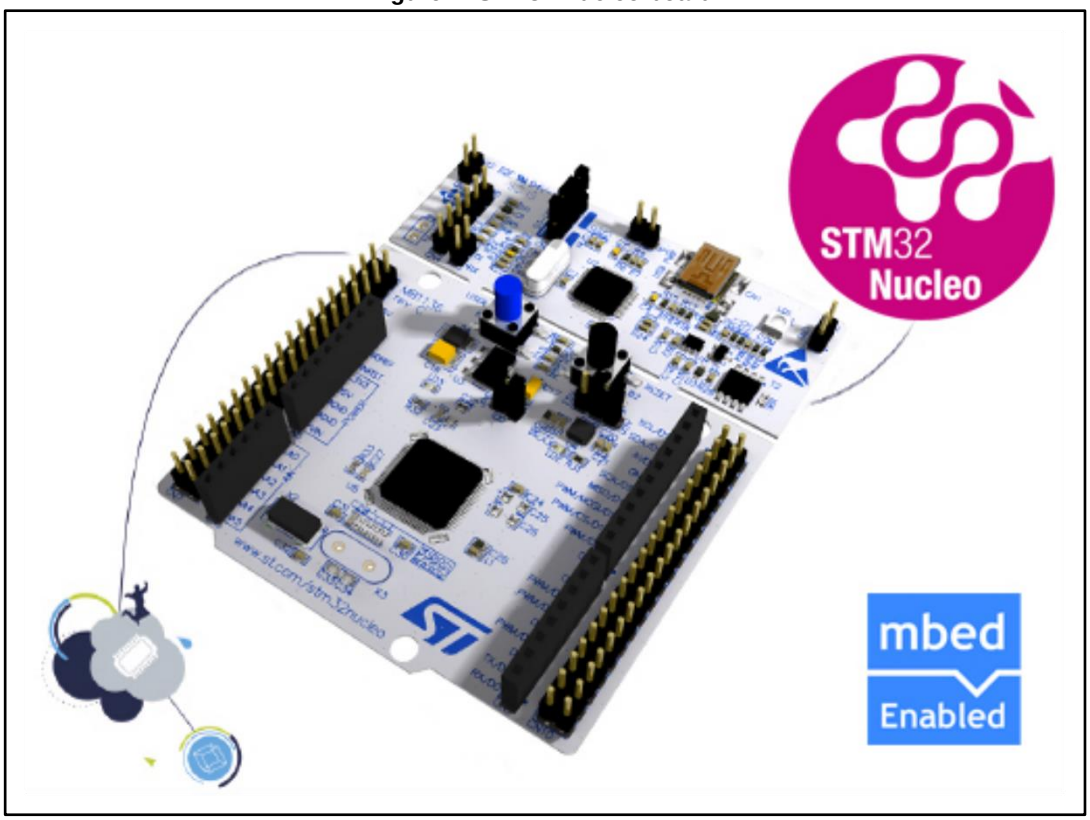
STM32 Nucleo development boards provide an affordable and flexible way for users to test solutions and build prototypes with any STM32 microcontroller line.

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 7: STM32 Nucleo board



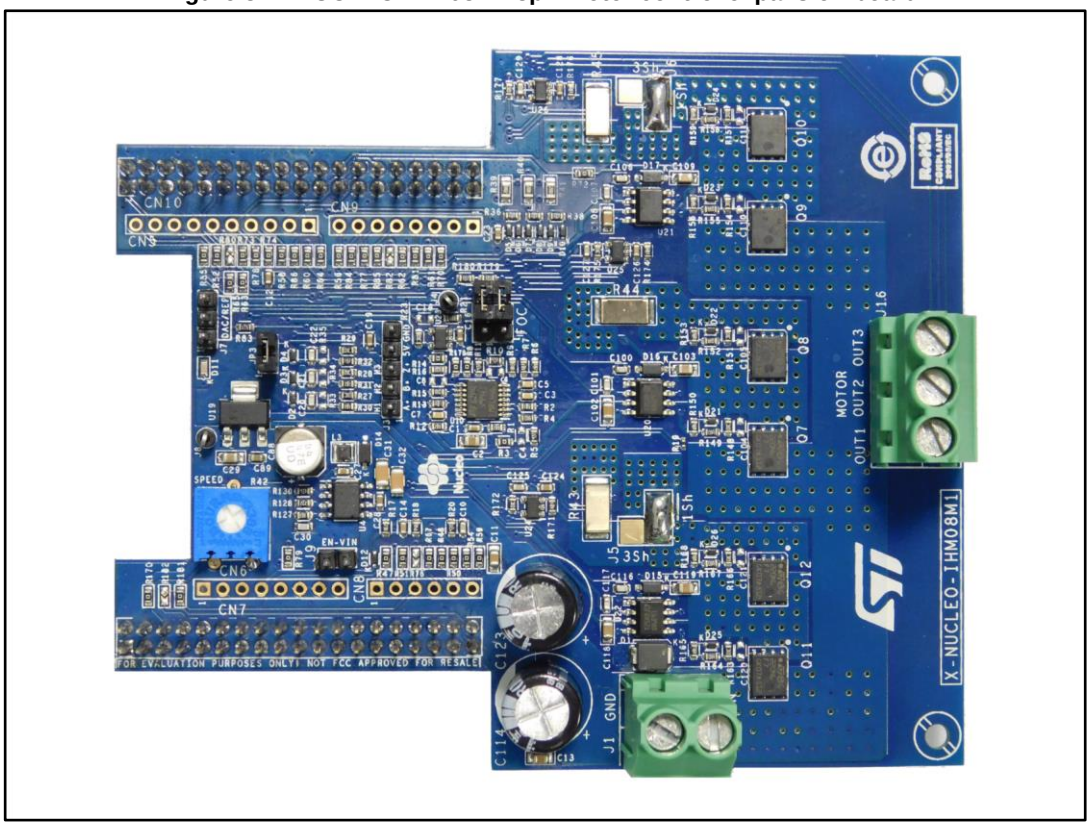
Information regarding the STM32 Nucleo board is available at *www.st.com/stm32nucleo*

4.1.2 X-NUCLEO-IHM08M1 expansion board

The X-NUCLEO-IHM08M1 is a three-phase brushless DC motor driver expansion board for STM32 Nucleo, based on the STL220N6F7 Power MOSFET. It provides an affordable and

easy-to-use solution for driving three-phase brushless DC motors in your STM32 Nucleo project. The X-NUCLEO-IHM08M1 is compatible with the ST morpho connector and supports the addition of additional boards plugged to a single STM32 Nucleo board. The user can also mount the Arduino UNO R3 connector. The IC driver used on this expansion board is the L6398: a single-chip, half-bridge gate driver for the N-channel Power MOSFET. The combination of the L6398 and the STL220N6F7 Power MOSFET provides a high current power platform for BLDC motors and the digital management by the STM32 Nucleo board allows for 6-step or FOC control algorithms which can be selected via the firmware.

Figure 8: X-NUCLEO-IHM08M1 3ph motor control expansion board



Main characteristics:

- 3-phase driver board for BLDC/PMSM motors
- nominal operating voltage range from 8 V to 48 V DC
- 15 A_{RMS} output current
- operating frequency selectable via firmware
- overcurrent detection and protection (30 A_{PEAK})
- thermal measurement and overheating protection
- fully compatible with ST 6-Step or ST FOC control algorithms
- full support for sensorless and sensor mode
- 3-shunt and 1-shunt configurable jumpers for motor current sensing
- Hall/Encoder motor sensor connector and circuit
- debug connector for DAC, GPIOs, etc.
- potentiometer available for speed regulation
- fully populated board conception with test points
- user LED
- compatible with STM32 Nucleo boards
- equipped with ST morpho connectors

- RoHS compliant

For 6-step control (X-CUBE-SPN8 FW) keep capacitor C5 mounted; in case of poor motor current regulation during startup, reduce its value.

For FOC control (STSW-STM32100 FW) remove capacitors C3, C5 and C7.

4.2 Software description

You need the following software components to create a suitable development environment for building applications for the STM32 Nucleo equipped with the motor control expansion board:

- **X-CUBE-SPN8** expansion for STM32Cube dedicated to MC applications development. The X-CUBE-SPN8 firmware and related documentation is available on www.st.com.
- **Development tool-chain and Compiler**: the STM32Cube expansion software supports the three following environments:
  - IAR Embedded Workbench for ARM® (EWARM) toolchain + ST-LINK
  - RealView Microcontroller Development Kit (MDK-ARM) toolchain + ST-LINK
  - AC6 System Workbench for STM32 + ST-LINK

4.3 Hardware and software setup

This section describes the hardware and software setup procedures. It also describes the system setup needed for the above.

4.3.1 Hardware setup

The following hardware components are needed:

1. One STM32 Nucleo Development platform (suggested order code: NUCLEO-F302R8 or NUCLEO-F401RE)
2. One 3-phase motor control expansion board (order code: X-NUCLEO-IHM08M1)
3. One USB type A to Mini-B USB cable to connect the STM32 Nucleo to the PC

For X-NUCLEO-IHM08M1 hardware settings, please refer to the relevant user manual available at www.st.com.

4.3.2 Software setup

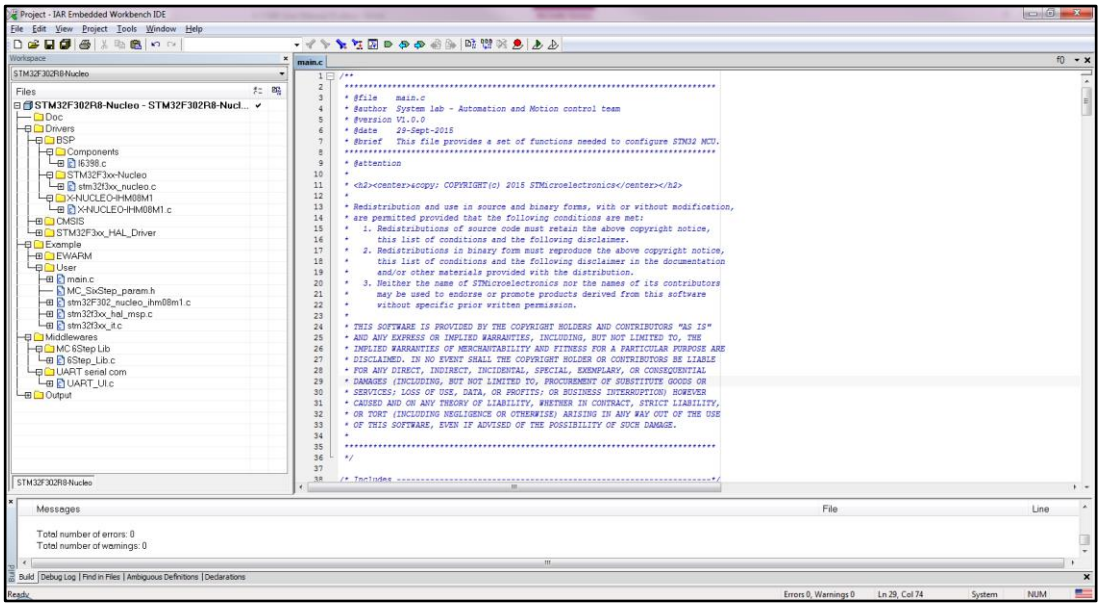
This section lists the minimum requirements for the developer to setup the SDK, run the sample testing scenario and customize applications.

4.3.2.1 Development tool-chains and compilers

Please select one of the Integrated Development Environments supported by the STM32Cube expansion software and follow the system requirements and setup information

supplied by the selected IDE provider. For instance, the IAR IDE with X-CUBE-SPN8 appears as shown in the figure below.

Figure 9: IAR Workspace with X-CUBE-SPN8



The main file contains the configuration instructions for each peripherals generated by ST CubeMX software and the starting point function for MC library. The stm32fxxx_nucleo_IHM08m1.c/h is the interface file between the library and the Nucleo board with all the specific functions for the selected STM32. The header file must be changed according to the modification on CubeMX file (*.ioc). The L6398.c file implements all the API functions to manage the motor driver, with start/stop PWM signals, enable/disable each inverter leg, etc. At the user level, MC_SixStep_param.h contains all the motor driving parameters, including the number of pole pairs, target speed, PI parameters for speed regulation, etc.

The Middlewares folder contains the core software motor control component: the 6-Step sensorless current control algorithm, able to drive a generic three-phase BLDC motor with speed control loop. The header file contains all the data structures and lists the main motor control APIs.

4.3.2.2 PC Utility

The UI interface has following minimum requirements:

- PC running one of following Microsoft operating systems:
  - Windows XP SP3
  - Windows Vista
  - Windows 7
- At least 128 MB of RAM
- 1 X USB port
- Hyper-terminal software or equivalent PC-terminal.

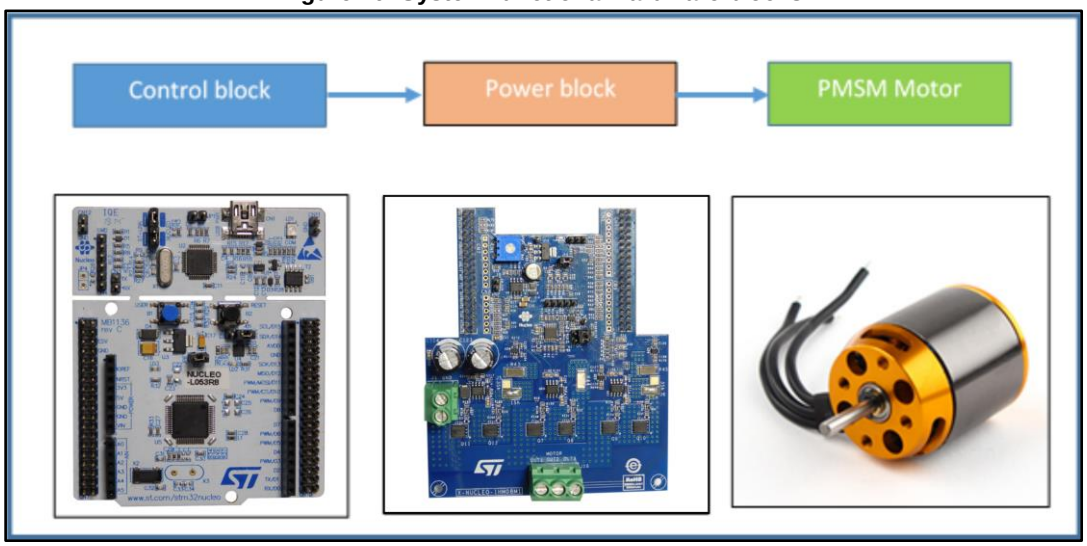
4.3.3 System setup guide

A generic motor control system can be broken down into the arrangement of three main functional blocks (see *Figure 10: "System functional hardware blocks"*):

- The control block accepts user commands to drive a motor. The X-NUCLEO-IHM08M1 is based on the STM32 Nucleo board, which provides all the digital signals for effective motor control.
- The power block is based on the 3-phase inverter topology. The core of the power block is the embedded L6398 driver, which contains all the necessary active power and analog components to perform low voltage PMSM motor control.
- The motor is a low voltage BLDC/PMSM motor.

This section describes how to set up different hardware components before writing and executing any applications on the STM32 Nucleo board with the low-voltage BLDC motor driver expansion board.

Figure 10: System functional hardware blocks



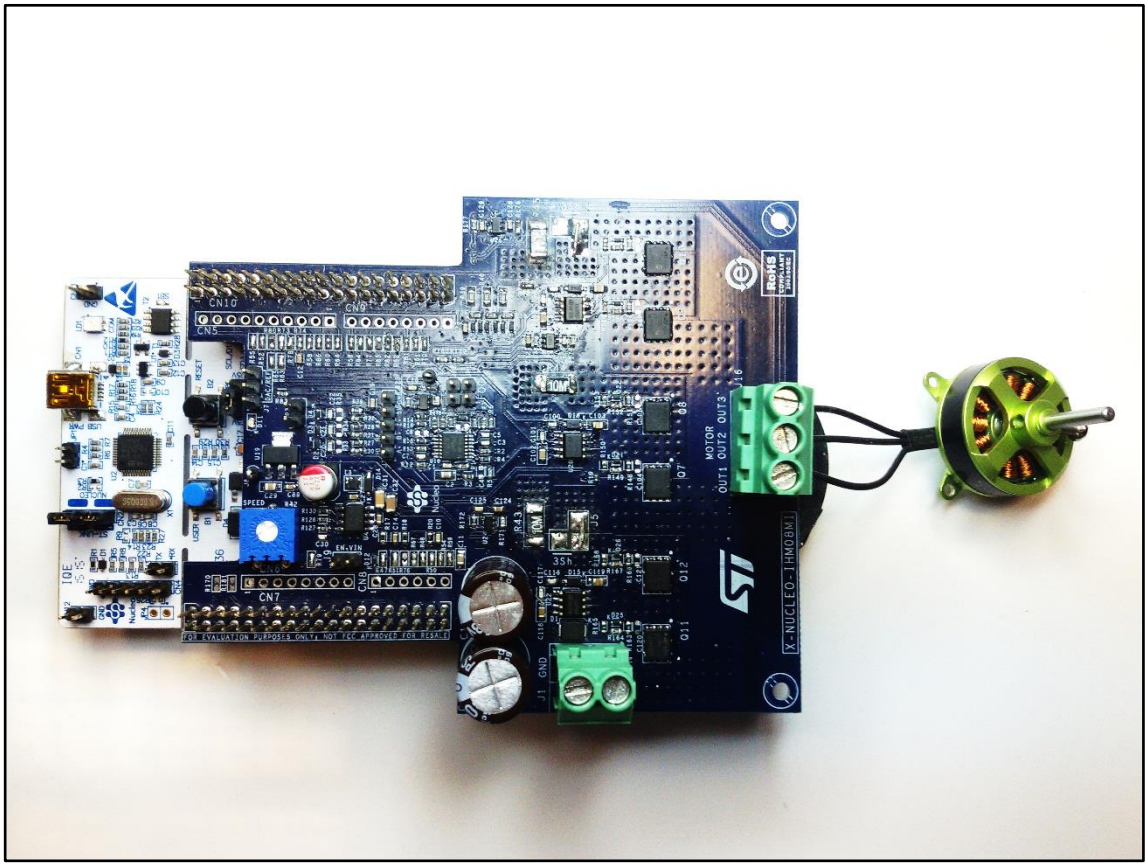
4.3.3.1 STM32 Nucleo and low-voltage BLDC motor driver expansion board setup

The STM32 Nucleo board integrates the ST-LINK/V2-1 debugger/programmer. The developer can download the relevant version of the ST-LINK/V2-1 USB driver by searching “ST-SW-LINK008” or “STSW-LINK009” at www.st.com (depending on your Microsoft Windows operating system).

The X-NUCLEO-IHM08M1 “Power block” must be plugged on top of an STM32 Nucleo board “Control block” through the ST morpho connector, as shown in *Figure 11: "X-NUCLEO-IHM08M1 connected to STM32 Nucleo board and low voltage BLDC motor"*.

Information regarding the X-NUCLEO-IHM08M1 expansion board is available on www.st.com at *http://www.st.com/x-nucleo*.

Figure 11: X-NUCLEO-IHM08M1 connected to STM32 Nucleo board and low voltage BLDC motor



The interconnection between the STM32 Nucleo and the X-NUCLEO-IHM08M1 is fully compatible with a wide range of STM32 Nucleo boards without any solder bridge modifications.

5 Revision history

Table 2: Document revision history

Date	Version	Changes
26-Nov-2015	1	Initial release.
06-Jun-2017	2	In *Section 4.1.2: "X-NUCLEO-IHM08M1 expansion board"*: added suggestions for FOC settings (C3, C5 and C7 capacitors).

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.