UM1946
User manual

Getting started with the X-CUBE-SPN7, 3-phase DC motor driver software expansion for STM32Cube

Introduction

This document describes how to get started with the X-CUBE-SPN7 software expansion for STM32Cube.

X-CUBE-SPN7 provides the complete STM32 middleware to build motor control applications (e.g. BLDC/PMSM motors). It is easily ported across different MCU families thanks to STM32Cube. This package contains a user interface layer enabling the transmission of real time data to a PC via the terminal.

The software provides implementation examples for STM32 Nucleo platforms equipped with the X-NUCLEO-IHM07M1 expansion board, featuring motor control applications with a 6-step algorithm.

The software is based on STM32Cube technology and expands the range of STM32Cube-based packages.

Contents

1 What is STM32Cube?

1.1 STM32Cube overview

The STMCube™ initiative was designed by STMicroelectronics to help developers by reducing development effort, time and cost. STM32Cube covers the STM32 portfolio.

STM32Cube Version 1.x includes:

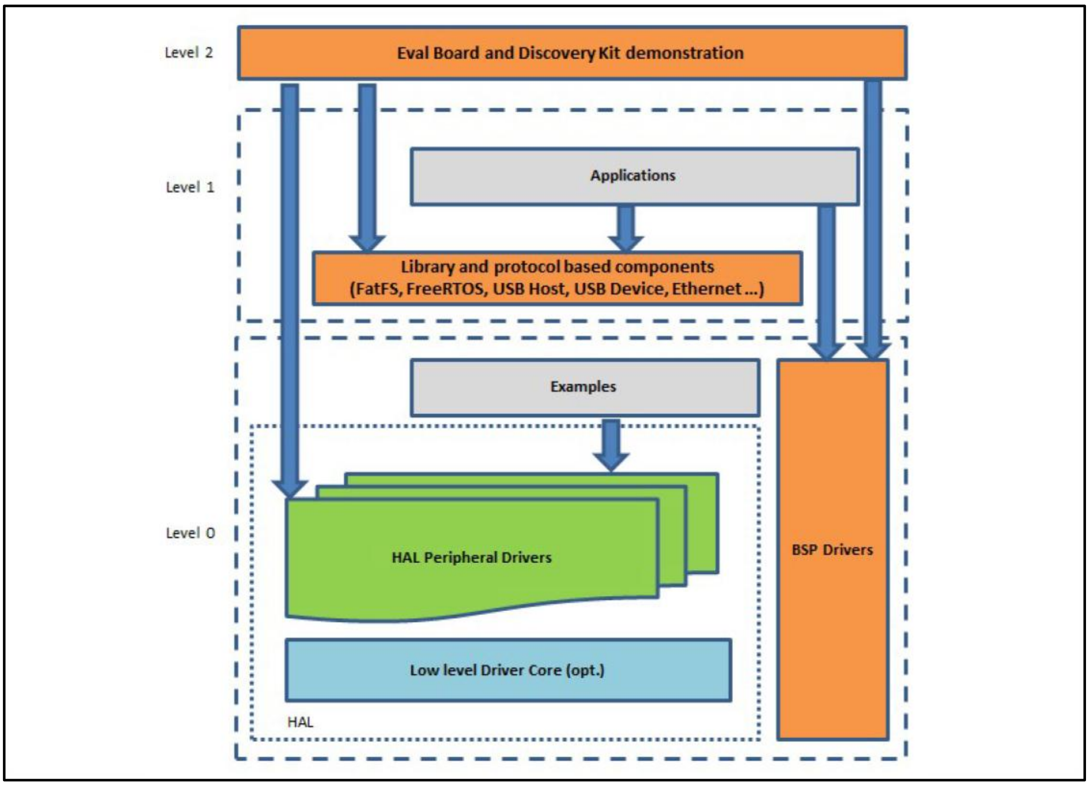
- The STM32CubeMX, a graphical software configuration tool that allows the generation of C initialization code using graphical wizards.
- A comprehensive embedded software platform for each series (such as STM32CubeF4 for the STM32F4 series)
  - The STM32Cube HAL, an STM32 abstraction layer embedded 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

Information about STM32Cube is available on www.st.com at:
http://www.st.com/stm32cube

1.2 STM32Cube architecture

The STM32Cube firmware solution is built around three independent levels that can easily interact with each other as shown below:

Figure 1: Firmware architecture


Level 2
Eval Board and Discovery Kit demonstration
Level 1
Applications
Library and protocol based components
(FatFS, FreeRTOS, USB Host, USB Device, Ethernet...)
Examples
Level 0
HAL Peripheral Drivers
BSP Drivers
Low level Driver Core (opt.)
HAL

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.) and 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 BSP driver external components and can be ported to any other board.
  - BSP driver: permits linking the component driver to a specific board and provides a set of user friendly APIs. The API naming convention is BSP_FUNCT_Action(): e.g. BSP_LED_Init(), BSP_LED_On().

It is based on modular architecture allowing it to be easily ported to any hardware by simply implementing the low level routines.

- 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 which render the implementation of user applications unnecessary by providing ready to use processed. For example, for the communication peripherals (I²S, UART, etc.) it provides APIs to initialize and configure the peripheral, manage data transfer based on polling, interrupt or DMA processes, and manage communication errors that may arise during communication. The HAL Drivers APIs are split into two categories: generic APIs, which provide common and generic functions to all the STM32 series, and extension APIs which provide special and customized functions for specific product families or part numbers.
- Basic peripheral usage examples: this layer includes the examples built around the STM32 peripheral only using the HAL and BSP resources.

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

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

Level 2: This level is composed of 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 functionalities.

2 X-CUBE-SPN7 software, expansion for STM32Cube

2.1 Overview

X-CUBE-SPN7 is a software package that expands the functionality provided by STM32Cube.

The key features of the package are:

- A complete firmware package to build motor control applications based on a single driver (L6230) and hardware expansion board (X-NUCLEO-IHM07M1)
- An API function to send any application command to the motor driver
- Easy portability across different MCU families thanks to STM32Cube
- Free user-friendly license terms
- Example implementations available on the X-NUCLEO-IHM07M1 board plugged on top of a NUCLEO-F030R8, NUCLEO-F103RB, NUCLEO-F302R8 or NUCLEO-F401RE board

The software provided is based on the STM32CubeHAL, the 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 L6230. The drivers abstract low-level hardware specifics and allow the middleware components and applications to send several application commands to the L6230 in a hardware-independent manner. It allows complete management of the L6230 by providing a full set of APIs to send any application command to motor driver. The package includes an application example to drive a low voltage three phase BLDC/PMSM motor.

2.2 Architecture

This software is an expansion for STM32Cube and therefore fully complies with the STM32Cube architecture and expands it in order to enable development of applications using the DC motor driver. The previous chapter provides an introduction to the STM32Cube architecture.

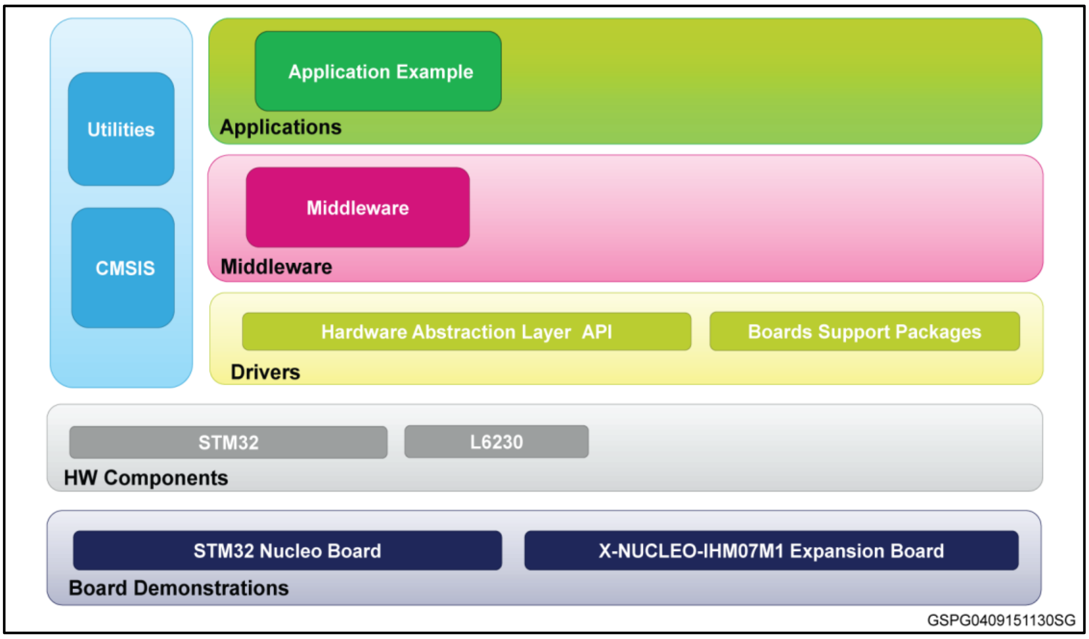
The software is based on the STM32CubeHAL, the hardware abstraction layer for the STM32 microcontroller. The package extends STM32Cube by providing a Board Support Package (BSP) for the DC motor driver expansion board and some middleware components for serial communication with a PC.

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

- The STM32Cube HAL layer: provides a generic, simple, 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 generic architecture and allows the layers that are built upon, such as the middleware layer, to implement their functions without depending on specific hardware configurations for a given Microcontroller Unit (MCU). This structure improves the library code reusability and guarantees easy portability to other devices.
- Board Support Package (BSP) Layer: supports the peripherals on the STM32 Nucleo board apart from the MCU. 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. For MC expansion boards, it provides the code to manage the L6230 driver and the X-NUCLEO board.

The following figure illustrates the software architecture of the package:

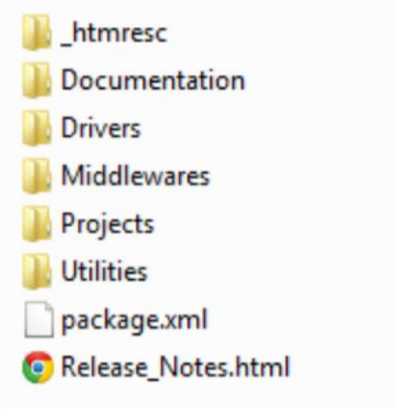
Figure 2: X-CUBE-SPN7 software architecture



2.3 Folders structure

This section provides an overview of the package folder structure.

Figure 3: X-CUBE-SPN7 package folders structure



_htmresc
Documentation
Drivers
Middlewares
Projects
Utilities
package.xml
Release_Notes.html

The following folders are included in the software package:

- **Documentation**: contains a compiled HTML file generated from the source code and documents the software component and API details.
- **Drivers**: contains the HAL drivers, the board specific drivers for each supported board or hardware platform, including the on-board component drivers and the CMSIS layer which is a vendor-independent hardware abstraction layer for the Cortex-M processor series.
- **Middlewares**: contains MC libraries and protocols related to the serial communication of DC motor driver data with a connected PC application.
- **Projects**: this folder contains a sample application used for MC applications, provided for the NUCLEO-L030R8, NUCLEO-F103RB, NUCLEO-F302R8 or NUCLEO-F401RE platforms with three development environments (IAR Embedded Workbench for ARM,

RealView Microcontroller Development Kit (MDK-ARM), AC6 System Workbench for STM32. The binary sub folder also contains the binary file for the P-NUCLEO-IHM001 kit. Please refer to the P-NUCLEO-IHM001 user manual for programming the NUCLEO-F302R8 with the factory firmware kit, if needed.

- **Utilities**: contains a folder for PC_software in which a Windows PC utility is provided. In this case no PC software is included.

2.4 APIs

Detailed technical information about the APIs available to the user can be found in a compiled HTML file located inside the *Documentation* folder of the software package where all the functions and parameters are fully described.

2.5 Sample application description

An example application using the X-NUCLEO-IHM07M1 expansion board with either NUCLEO-L030R8, NUCLEO-F103RB, 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 3ph motor is driven by the L6230 device. The firmware implements a 6-Step MC algorithm. The current measured in single shunt mode is compared with a reference generated by a digital speed loop for current control.

The main functions to manage a motor control application are:

MC Initialization: MC_SixStep_INIT();

Send Start motor command: MC_StartMotor();

Send Stop motor command: MC_StopMotor();

Send change motor speed: MC_SetSpeed(value);

command:

2.6 User Interface utility based on PC-terminal

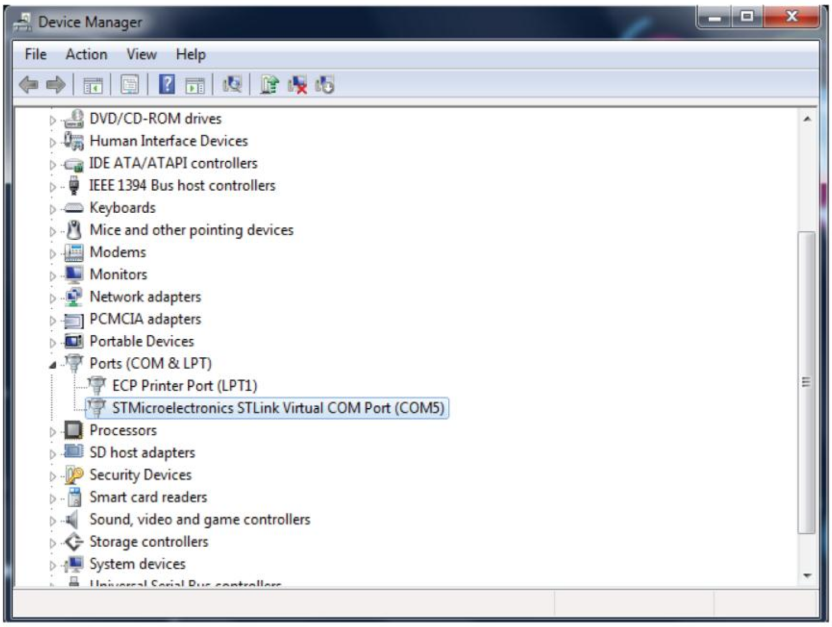
The X-CUBE-SPN7 expansion for STM32Cube implements serial communication between the STM32 Nucleo board UART peripheral and a Windows PC-terminal respectively. This utility is available by recompiling the project in COMM mode and configuring a terminal like hyper-terminal.

This section describes the procedure to configure the terminal to connect STM32 Nucleo board to the PC. Before using this utility, the user must ensure that the necessary drivers, as explained in the chapter dedicated to the system setup, are installed and the expansion board along with STM32 Nucleo is connected to a PC.

Please follow these steps.

- Check the windows *Device Manager* to obtain the ST COM port; COM5 in the example below.

Figure 4: Device manager view on windows

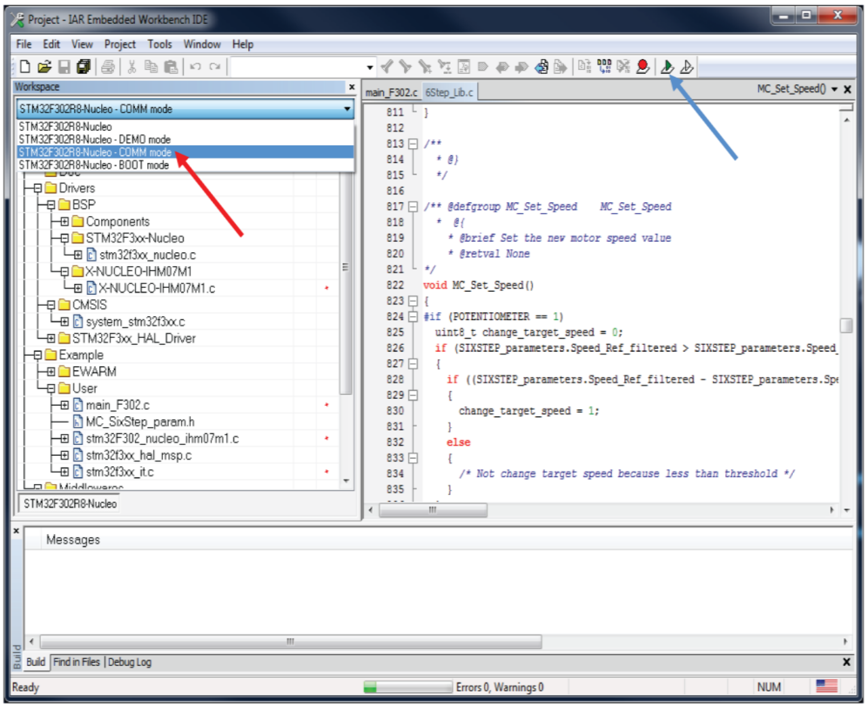


- DVD/CD-ROM drives
- Human Interface Devices
- IDE ATA/ATAPI controllers
- IEEE 1394 Bus host controllers
- Keyboards
- Mice and other pointing devices
- Modems
- Monitors
- Network adapters
- PCMCIA adapters
- Portable Devices
- Ports (COM & LPT)
  - ECP Printer Port (LPT1)
  - STMicroelectronics STLink Virtual COM Port (COM5)
- Processors
- SD host adapters
- Security Devices
- Smart card readers
- Sound, video and game controllers
- Storage controllers
- System devices
- Universal Serial Bus controllers

GSPG0409151155SG

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

Figure 5: IAR Workbench - compiling in COMM mode and flash



STM32F302R8-Nucleo - COMM mode

- STM32F302R8-Nucleo
- STM32F302R8-Nucleo - DEMO mode
- STM32F302R8-Nucleo - COMM mode
- STM32F302R8-Nucleo - BOOT mode

```
Drivers
  BSP
    Components
    STM32F3xx-Nucleo
      stm32f3xx_nucleo.c
    X-NUCLEO-IHM07M1
      X-NUCLEO-IHM07M1.c
  CMSIS
    system_stm32f3xx.c
  STM32F3xx_HAL_Driver
Example
  EWARM
  User
    main_F302.c
    MC_SixStep_param.h
    stm32F302_nucleo_ihm07m1.c
    stm32f3xx_hal_msp.c
    stm32f3xx_it.c
Middlewares
```

STM32F302R8-Nucleo

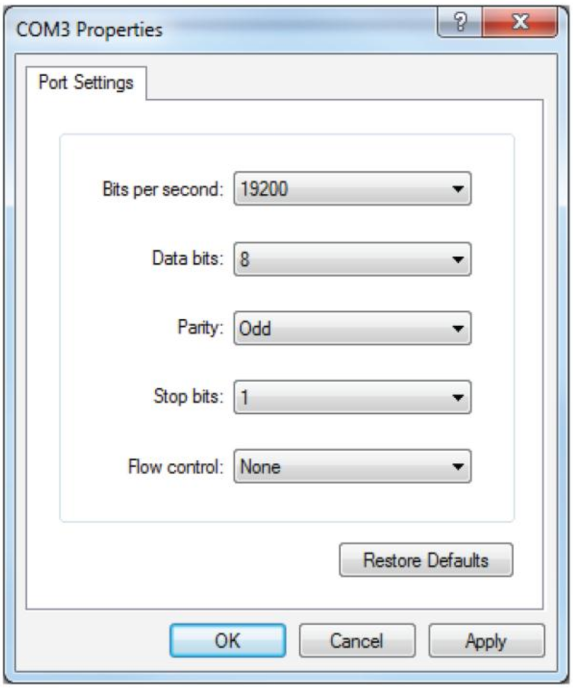
```
}

/**
  * @}
  */

/** @defgroup MC_Set_Speed    MC_Set_Speed
  *  @{
    * @brief Set the new motor speed value
    * @retval None
*/
void MC_Set_Speed()
{
#if (POTENTIOMETER == 1)
  uint8_t change_target_speed = 0;
  if (SIXSTEP_parameters.Speed_Ref_filtered > SIXSTEP_parameters.Speed_
  {
    if ((SIXSTEP_parameters.Speed_Ref_filtered - SIXSTEP_parameters.Spe
    {
      change_target_speed = 1;
    }
    else
    {
      /* Not change target speed because less than threshold */
    }
```

Errors 0, Warnings 0

GSPG0409151230SG

- Launch Hyper-terminal 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 6: PC terminal parameters



COM3 Properties

Port Settings

Bits per second: 19200

Data bits: 8

Parity: Odd

Stop bits: 1

Flow control: None

GSPG0409151240SG

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

3 System setup guide

3.1 Hardware description

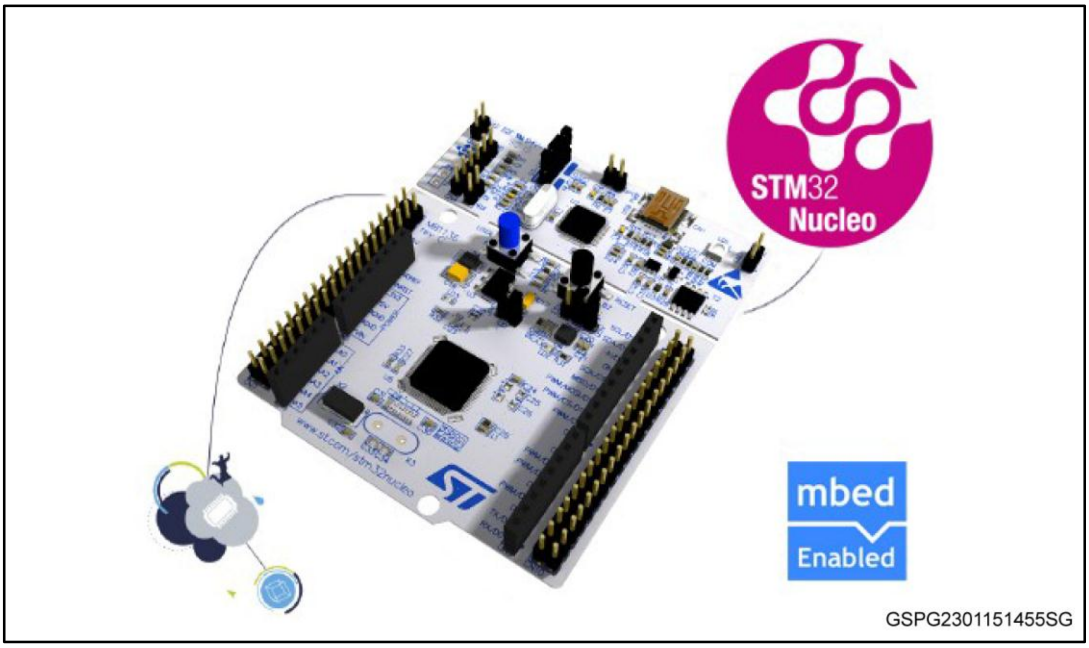
This section describes the hardware components needed for developing a DC motor driver based application.

3.1.1 STM32 Nucleo platform

The STM32 Nucleo boards provide an affordable and flexible way for users to try out new ideas and build prototypes with any STM32 microcontroller lines. The Arduino™ connectivity support and ST morpho headers 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 any separate probe 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.

Information regarding STM32 Nucleo boards is available on www.st.com at http://www.st.com/stm32nucleo

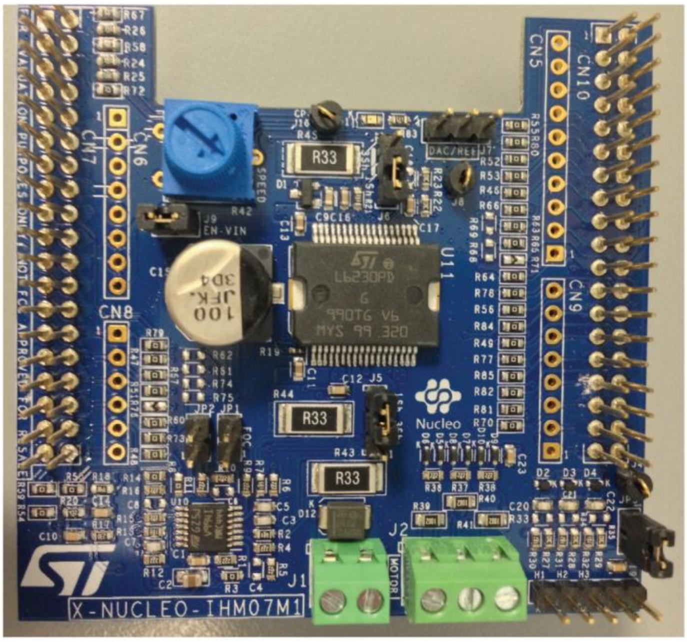
Figure 7: STM32 Nucleo board



3.1.2 X-NUCLEO-IHM07M1 expansion board

The X-NUCLEO-IHM07M1 is a three-phase brushless DC motor driver expansion board based on L6230 for STM32 Nucleo. It provides an affordable and easy-to-use solution for driving three-phase brushless DC motors in your STM32 Nucleo project. The X-NUCLEO-IHM07M1 is compatible with the ST morpho connector and supports the addition of other boards which can be stacked with a single STM32 Nucleo board. The user can also mount the Arduino UNO R3 connector. The driver used on this STM32 Nucleo board is the L6230: a DMOS fully integrated driver for three-phase brushless DC motors assembled in a PowerSO36 package, with overcurrent and thermal protection. This driver is optimized for 6-Step and FOC algorithms thanks to the independent current sensing.

Figure 8: X-NUCLEO-IHM07M1 - 3ph motor control expansion board



GSPG0409151245SG

Main characteristics:

The information below shows the board specification data and the main parameter set for the X-NUCLEO-IHM07M1 expansion board:

- 3 phase driver board for BLDC/PMSM motors
- Nominal operating voltage range from 8 V to 48 V DC
- 2.8 A output peak current (1.4 A RMS)
- Operating frequency up to 100 kHz
- Non dissipative overcurrent detection and protection
- Cross conduction protection
- Thermal measuring and overheating protection
- Fully compatible with the 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
- PCB type and size:
  - PCB material - FR-4
  - 4-layer layout
  - Copper thickness: 70 µm (external layer), 35 µm (internal layer)
  - Total dimensions of the expansion board: 70 mm x 66 mm

3.2 Software description

The following software components are needed for a suitable development environment to create applications for the STM32 Nucleo equipped with the MC expansion board:

- X-CUBE-SPN7: an expansion for STM32Cube dedicated to MC application development. The X-CUBE-SPN7 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

3.3 Hardware and Software setup

This section describes the hardware and software setup procedures, and the associated system setup.

3.3.1 Hardware setup

The following hardware components are needed:

1. One STM32 Nucleo development platform (suggested order code: NUCLEO-F030R8, NUCLEO-F103RB, NUCLEO-F302R8 or NUCLEO-F401RE)
2. One 3ph motor control expansion board (order code: X-NUCLEO-IHM07M1)
3. One USB type A to Mini-B USB cable to connect the STM32 Nucleo to the PC

Hardware settings:

Table 1: Jumper settings

Jumper	Permitted configurations	Default condition
JP1	Selection for pull-up insertion (BIAS) in current sensing circuit	OPEN
JP2	Selection for Op.Amp gain modification in current sensing circuit	OPEN
JP3	Selection for pull-up enabling in Hall/Encoder detection circuit	CLOSED
J9	Selection to supply the STM32 Nucleo board through the X-NUCLEO-IHM07M1 (1)	OPEN
J5	Selection for single/three SHUNT configuration (single shunt by default)	2-3 CLOSED
J6	Selection for single/three SHUNT configuration (single shunt by default)	2-3 CLOSED
J7	Debug connector for DAC. Available for probe connection	OPEN

Notes:

(1)IT IS RECOMMENDED TO REMOVE THE JUMPER J9 BEFORE POWER-ON ON J1. WITH J9 CLOSED, DO NOT PROVIDE MORE THAN 12 V DC ON THE J1 CONNECTOR TO AVOID DAMAGING THE NUCLEO BOARD. Jumper JP5 (STM32 Nucleo) MUST be connected between pin 2-3 to enable the supply of external power to STM32 Nucleo.

Table 2: Screw terminal table

Screw terminal	Function
J1	Motor power supply input (8 V - 48 V DC)
J2	3-PH motor connector

The X-NUCLEO-IHM07M1 expansion board is based on the ST morpho connector, male pin headers (CN7 and CN10) accessible on both sides of the board. They can be used to connect this power board to the STM32 Nucleo board. All signals and power pins of the MCU are available on the ST morpho connector. For further details, please refer to the document UM1724 (5.12 STMicroelectronics morpho connector) available on website www.st.com

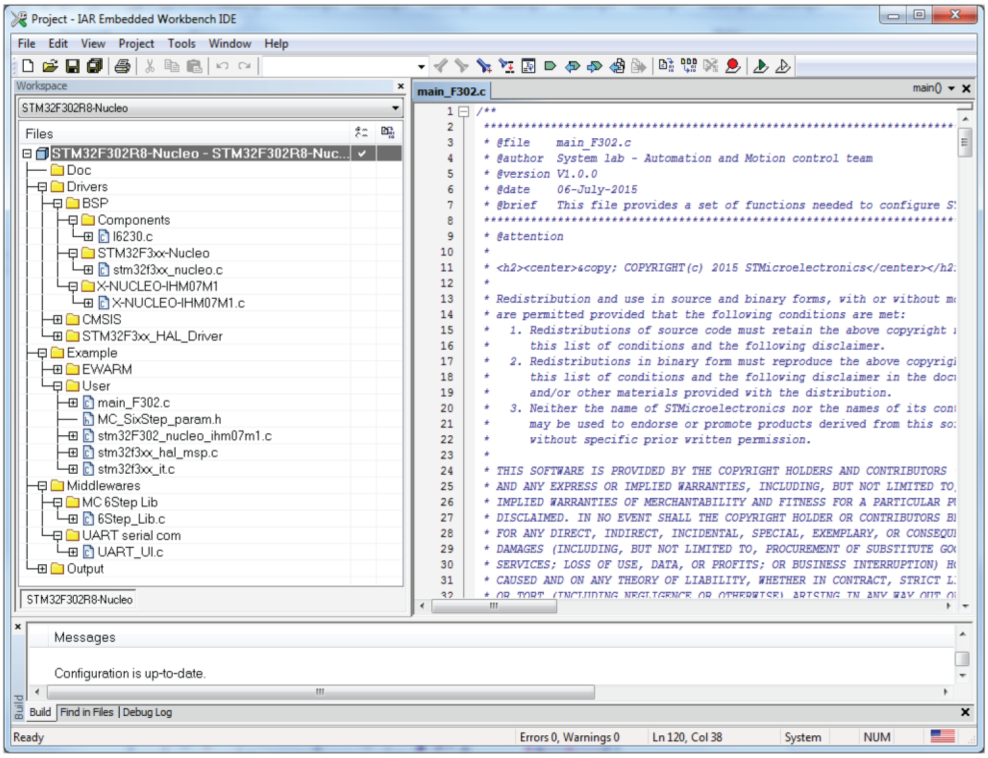
3.3.2 Software setup

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

3.3.3 Development Tool-chains and compilers

Select one of the Integrated Development Environments supported by the STM32Cube expansion software. Please read the system requirements and setup information provided by the selected IDE provider. For instance, the IAR IDE with X-CUBE-SPN7 appears as shown below.

Figure 9: IAR Workspace with X-CUBE-SPN7



```
STM32F302R8-Nucleo
STM32F302R8-Nucleo - STM32F302R8-Nuc... ✓
├ Doc
├ Drivers
│ ├ BSP
│ │ ├ Components
│ │ │ └ l6230.c
│ │ ├ STM32F3xx-Nucleo
│ │ │ └ stm32f3xx_nucleo.c
│ │ └ X-NUCLEO-IHM07M1
│ │   └ X-NUCLEO-IHM07M1.c
│ ├ CMSIS
│ └ STM32F3xx_HAL_Driver
├ Example
│ ├ EWARM
│ └ User
│   ├ main_F302.c
│   ├ MC_SixStep_param.h
│   ├ stm32F302_nucleo_ihm07m1.c
│   ├ stm32f3xx_hal_msp.c
│   └ stm32f3xx_it.c
├ Middlewares
│ ├ MC 6Step Lib
│ │ └ 6Step_Lib.c
│ └ UART serial com
│   └ UART_UI.c
└ Output
STM32F302R8-Nucleo
```

```
/**
  ******************************************************************************
  * @file    main_F302.c
  * @author  System lab - Automation and Motion control team
  * @version V1.0.0
  * @date    06-July-2015
  * @brief   This file provides a set of functions needed to configure S
  ******************************************************************************
  * @attention
  *
  * <h2><center>&copy; COPYRIGHT(c) 2015 STMicroelectronics</center></h2
  *
  * Redistribution and use in source and binary forms, with or without mo
  * are permitted provided that the following conditions are met:
  *   1. Redistributions of source code must retain the above copyright 
  *      this list of conditions and the following disclaimer.
  *   2. Redistributions in binary form must reproduce the above copyrig
  *      this list of conditions and the following disclaimer in the doc
  *      and/or other materials provided with the distribution.
  *   3. Neither the name of STMicroelectronics nor the names of its con
  *      may be used to endorse or promote products derived from this so
  *      without specific prior written permission.
  *
  * THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS 
  * AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO,
  * IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PU
  * DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE
  * FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQU
  * DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GO
  * SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) H
  * CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT L
  * OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT O
```

```
Configuration is up-to-date.
```

GSPG0409151250SG

The main file contains the configuration instructions for each peripheral generated by the ST CubeMX software and the starting point function for the MC library. The stm32fxxx_nucleo_ihm07m1.c/h is the interface file between the library and the Nucleo board; in particular, it contains all the specific functions for the STM32 selected. The header file must be changed according to the modification on the CubeMX file (*.ioc). The L6230.c file implements all the API functions to manage the motor driver; in particular, it is possible to start/stop PWM signals, enable/disable each inverter leg, etc. At the user level, MC_SixStep_param.h contains all the motor driving parameters like the number of pole pairs, target speed, PI parameters for speed regulation, etc.

The middleware folder contains the core of MC structure: the 6-step algorithm. It is a sensorless current control method able to drive a generic three phase BLDC motor with speed control loop. Its header file contains all the data structure and the list of main APIs for the MC.

3.3.4 PC utility

The user interface has following 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
- Hyper-terminal software or equivalent PC-terminal.

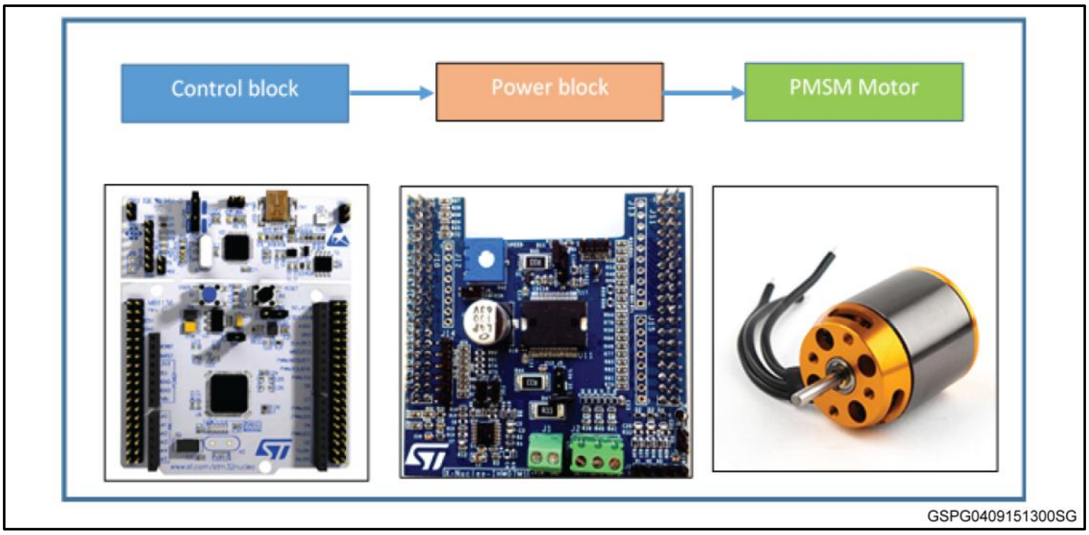
3.3.5 System setup guide

A generic motor control system can be generally configured in the arrangement of three main blocks (see *Figure 10: "System architecture"*):

- **Control block** - its main task is to accept user commands and drive a motor. The X-NUCLEO-IHM07M1 is based on STM32 Nucleo board that provides all the digital signals to implement proper motor driving control.
- **Power block** - it is based on the 3-phase inverter topology. The core of the power block embedded on board is the L6230 driver which contains all the necessary active power and analog components to perform low voltage PMSM motor control.
- **Motor** - the X-NUCLEO-IHM07M1 is able to properly drive a low voltage BLDC/PMSM motor.

This section describes how to setup different hardware parts before writing and executing an application on the STM32 Nucleo board with the DC motor driver expansion board.

Figure 10: System architecture

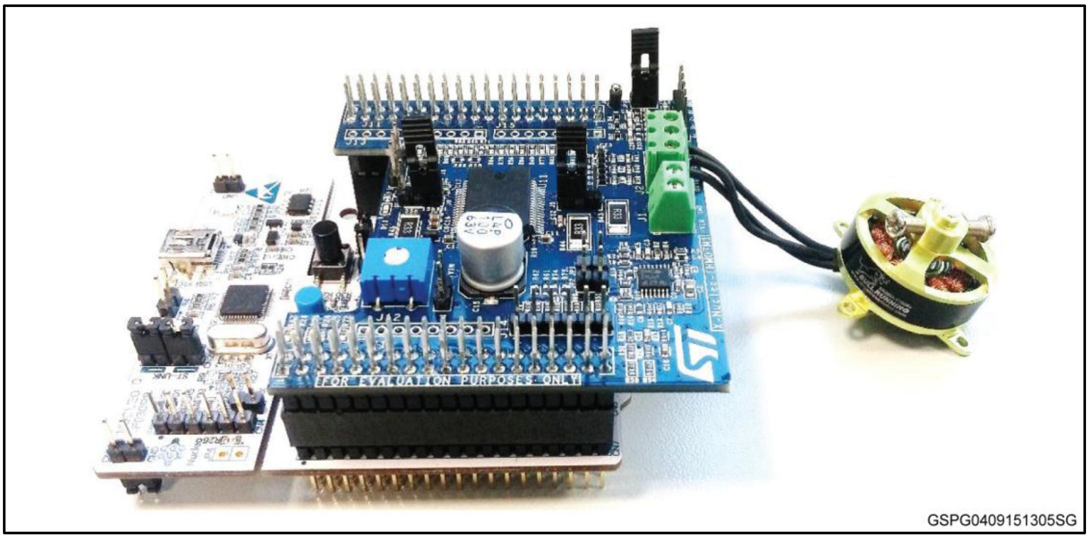


3.3.6 STM32 Nucleo and DC motor driver expansion boards setup

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 STSW-LINK008 or STSW-LINK009 on www.st.com (for Microsoft Windows OS). For regular operation, the X-NUCLEO (power board) must be plugged onto an STM32 Nucleo board (Control block) through the ST morpho connector as shown in *Figure 11: "X-NUCLEO-IHM07M1 expansion board connected to STM32 Nucleo Board and low voltage BLDC motor"*.

Information regarding the X-NUCLEO-IHM07M1 expansion board is available on www.st.com at http://www.st.com/x-nucleo.

Figure 11: X-NUCLEO-IHM07M1 expansion board connected to STM32 Nucleo Board and low voltage BLDC motor



The interconnection between the STM32 Nucleo and the X-NUCLEO-IHM07M1 is designed for full-compatibility with a broad range of STM32 Nucleo boards without modifying any solder bridges. The assembled system is ready to operate when connected with a BLDC/PMSM motor. For correct use, please adopt the hardware and software settings in this document.

4 Acronyms and abbreviations

Table 3: Acronyms

Acronym	Description
MC	Motor Control
6-Step	6-step algorithm for motor control
X-NUCLEO	X-NUCLEO-IHM07M1
UI	User Interface serial communication
3ph	Three phase motor

5 Revision history

Table 4: Document revision history

Date	Revision	Changes
18-Sep-2015	1	Initial release.

IMPORTANT NOTICE – PLEASE READ CAREFULLY