
Getting started with X-CUBE-IPS02 24 V Intelligent Power Switch software expansion for STM32Cube

Introduction

This document describes how to get started with the X-CUBE-IPS02 software.

X-NUCLEO-IPS02A1 provides a low cost, robust and easy-to-use solution for driving grounded loads, i.e. relay replacement, heating elements, solenoid driver, LED driver, light bulb driver, pumps and fans, in STM32 Nucleo projects.

It is easily portable across different MCU families thanks to STM32Cube. This package contains a sample application enabling to activate or de-activate output channels and measure current circulating on the loads connected to the channels.

The software provides implementation examples for STM32 Nucleo platforms equipped with the X-NUCLEO-IPS02A1 expansion board, featuring with VPS2535H Intelligent Power Switch.

The software is based on STM32Cube technology and expands STM32Cube based packages.

Contents

- 1 **What is STM32Cube? 5****
 - 1.1 STM32Cube overview 5
 - 1.2 STM32Cube Architecture 5

- 2 **X-CUBE-IPS02 software expansion for STM32Cube 8****
 - 2.1 Overview 8
 - 2.2 Architecture 8
 - 2.3 Folders structure 9
 - 2.4 APIs 10
 - 2.5 Sample application description 10

- 3 **System setup guide 12****
 - 3.1 Hardware description 12
 - 3.1.1 STM32 Nucleo platform 12
 - 3.2 X-NUCLEO-IPS02A1 expansion board 12
 - 3.3 Software description 16
 - 3.4 Hardware and software setup 16
 - 3.4.1 Hardware setup 16
 - 3.4.2 Software setup 17
 - 3.4.3 System setup guide 17

- Appendix A **Further information 18****
 - A.1 Document reference 18

- Revision history 19**

List of tables

Table 1.	Arduino UNO R3 connector table	14
Table 2.	ST morpho connector table	15
Table 3.	Document revision history	19

List of figures

Figure 1.	Firmware architecture	6
Figure 2.	X-CUBE-IPS02 Architecture	9
Figure 3.	X-CUBE-IPS02 package folders structure	9
Figure 4.	X-CUBE-IPS02 terminal output	10
Figure 5.	STM32 Nucleo Board	12
Figure 6.	Connectors position	13
Figure 7.	Jumpers position.	14
Figure 8.	X-NUCLEO-IPS02A1 expansion board connected to STM32 Nucleo board	16

1 What is STM32Cube?

1.1 STM32Cube overview

STM32Cube™ initiative was originated by STMicroelectronics to ease developers' life by reducing development efforts, time and cost. STM32Cube covers STM32 portfolio.

Version 1.0 of the STM32Cube includes:

- The STM32CubeMX, a graphical software configuration tool that allows the generation of C language initialization code using graphical wizards.
- A comprehensive embedded software platform, delivered per series (such as STM32CubeF4 for STM32F4 series):
 - The STM32Cube HAL, an STM32 abstraction layer embedded software, ensuring maximized portability across STM32 portfolio
 - A consistent set of middleware components such as RTOS, USB, TCP/IP, Graphics
 - All embedded software utilities coming with a full set of examples.

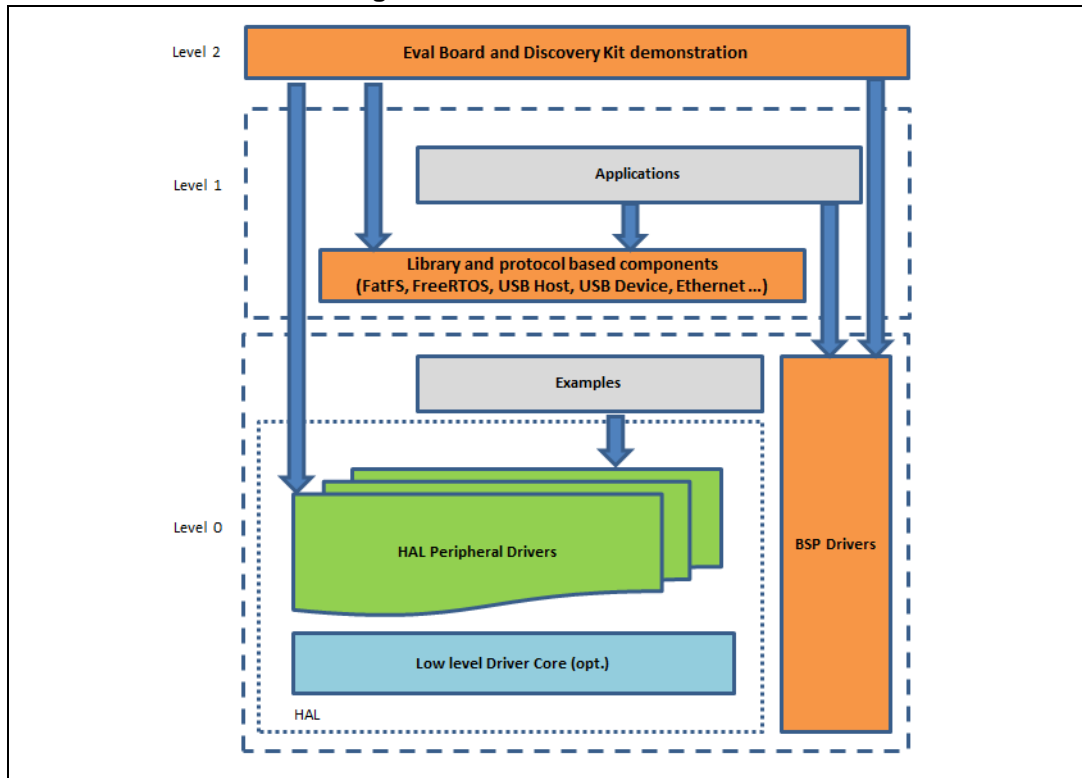
Information about STM32Cube are available on www.st.com at:

<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's as described in [Figure 1](#):

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...) 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 provide specific APIs to the BSP driver external components and could be portable on any other board.
 - **BSP driver:** permits linking the component driver to a specific board and provides a set of friendly used APIs. The APIs naming rule is BSP_FUNCT_Action(): ex. BSP_LED_Init(),BSP_LED_On()

It's based on modular architecture allowing to port it easily on any hardware by just 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 a generic, multi instance and functionalities oriented APIs which permit to offload the user application implementation by providing ready to use process. For example, for the communication peripherals (I²S, UART...) it provides APIs allowing to initialize and configure the peripheral, manage data transfer based on polling, interrupt or DMA process, and manage communication errors that may raise during communication. The HAL Drivers APIs are split in two categories, generic APIs which provides common and generic functions to all the STM32 series and extension APIs which provides specific and customized functions for a specific family or a specific part number.
- **Basic peripheral usage examples:** this layer encloses the examples build over the STM32 peripheral using only the HAL and BSP resources.

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 interactions between the components of this layer is done directly by calling the feature APIs while the vertical interaction with the low level drivers is done through specific callbacks and static macros implemented in the library system call interface. As example, the FatFs implements the disk I/O driver to access microSD drive or the USB Mass Storage Class.
- **Examples based on the Middleware components:** each Middleware component comes with one or more examples (called also Applications) showing how to use it. Integration examples that use several Middleware components are provided as well.

Level 2: this level is composed of a single layer which is global real-time and graphical demonstration based on the Middleware service layer, the low level abstraction layer and the basic peripheral usage applications for board based functionalities.

2 X-CUBE-IPS02 software expansion for STM32Cube

2.1 Overview

X-CUBE-IPS02 is a software package that expands the functionality provided by STM32Cube.

The key features of the package are:

- Complete middleware to build applications using Intelligent Power Switch (VPS2535H)
- Easy portability across different MCU families thanks to STM32Cube
- Sample application to transmit real time current data measurement to a PC
- PC-based application (Microsoft Windows) to print on terminal current in several conditions
- Free user-friendly license terms
- Examples implementation available for X-NUCLEO-IPS02A1 expansion board plugged to a NUCLEO-F401RE or NUCLEO-L053R8 board

This software is used to manage Intelligent Power Switch based on VPS2535H allowing to control the 2 channels output and measure currents on connected load. It is built on top of STM32Cube software technology that ease portability across different STM32 microcontrollers.

The software comes with examples of implementation of such drivers, running on X-NUCLEO-IPS02A1 plugged to a NUCLEO-F401RE or NUCLEO-L053R8 board

2.2 Architecture

This software is an expansion for STM32Cube, as such it fully complies with the architecture of STM32Cube and expands it in order to enable development of applications using Intelligent Power Switch (24 V) based on VPS2535H chip. Please see the previous chapter for an introduction about 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 Intelligent Power Switch expansion board and some middleware components.

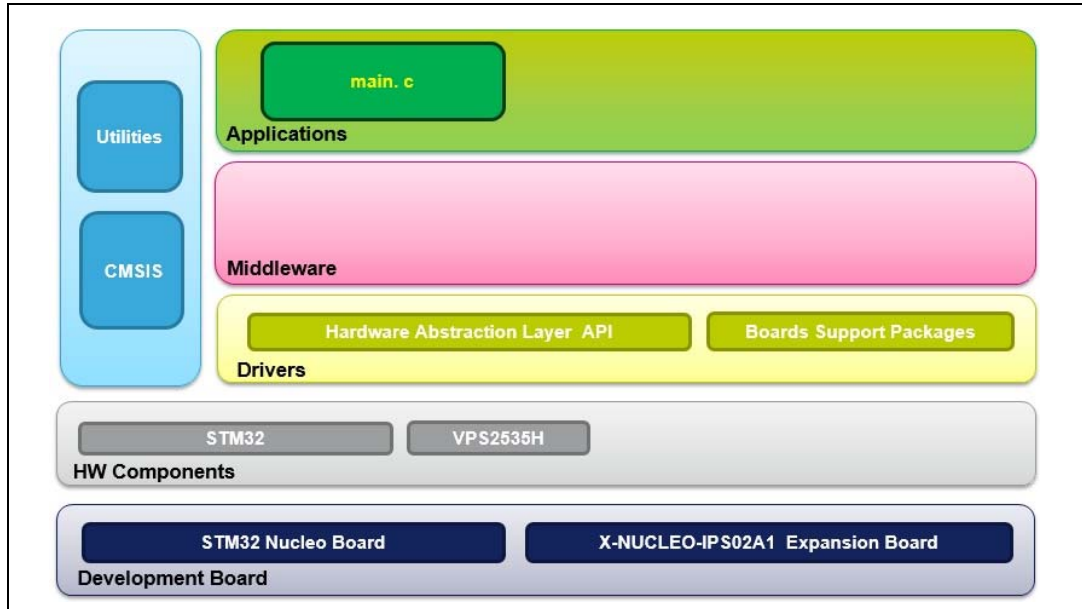
The software layers used by the application software to access and use the X-NUCLEO-IPS02A1 expansion board are the following:

- 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 Intelligent Power Switch expansion board, it provides the programming interface for a generic IPS and the specific VPS2535H chip. It provides support for initializing the switch and reading current measurement.

Figure 2 outlines the software architecture of the package:

Figure 2. X-CUBE-IPS02 Architecture

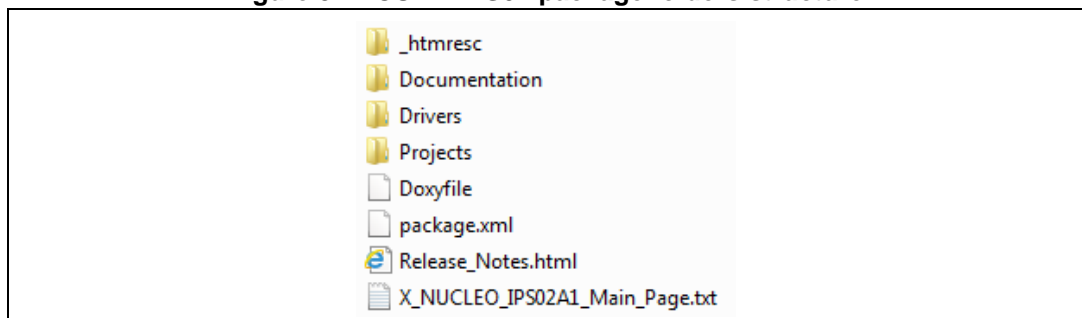


2.3 Folders structure

This section provides an overview of the package folders structure.

Figure 3 outlines the architecture of the package.

Figure 3. X-CUBE-IPS02 package folders structure



The following folders are included in the software package:

- Documentation: this folder contains a compiled HTML file generated from the source code and documenting in details the software components and APIs.
- Drivers: this folder contains the HAL drivers, the board specific drivers for each supported board or hardware platform, including the on-board components ones and

the CMSIS layer which is a vendor-independent hardware abstraction layer for the Cortex-M processor series.

- Projects: this folder contains a sample application used to access X-CUBE-IPS02A1 for the NUCLEO-L053R8 and NUCLEO-F401RE platforms with three development environments (IAR Embedded Workbench for ARM, Real/View Microcontroller Development Kit (MDK-ARM), and System Workbench for STM32 (SW4STM32)).

2.4 APIs

Detailed technical information about the APIs available to the user can be found in a compiled HTML file, named is **X_CUBE_IPS.chm**, and 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-IPS02A1 expansion board with either NUCLEO-F401RE or NUCLEO-L053R8 boards is provided in the “Projects” directory. Ready to be built projects are available for multiple IDEs.

This example application provides a basic code to show how to use the X-NUCLEO-IPS02A1 expansion board for STM32 Nucleo.

The example performs current measurements on output Channel 1 (Ch1) and Channel 2 (Ch2), continuously, in the following conditions:

1. Ch1 OFF, Ch2 OFF
2. Ch1 ON, Ch2 OFF
3. Ch1 OFF, Ch2 ON
4. Ch1 ON, Ch2 ON

for each configuration the Current and for each channel are displayed over an opened console (use Hyperterminal or whatever, set 9600 as bauds, 8-bit data, no parity).

Figure 4. X-CUBE-IPS02 terminal output

```
COM40:9600baud - Tera Term VT
File Edit Setup Control Window Help
*****
***** START TEST PROCEDURE *****
*****
This demo performs current measurements on Ch1 and Ch2
in the following conditions:
1) Ch1 OFF, Ch2 OFF
2) Ch1 ON, Ch2 OFF
3) Ch1 OFF, Ch2 ON
4) Ch1 ON, Ch2 ON
Note:
RAW is the value measured on sense pin by ADC
ACTUAL is the Current value calculated using raw value and multiplied factor
s
### READ CURRENT STAND BY ###
IM1 = 0, IM2 = 0, FR_STBY = 0
Current Actual Value Ch1 <Ch1 = OFF, Ch2 = OFF> =>0.026354
Current Actual Value Ch2 <Ch1 = OFF, Ch2 = OFF> =>0.079062
### READ CURRENT - CH1 = ON, CH2 = OFF, ST_BY = ON ###
IM1 = 0, IM2 = 1, FR_STBY = 1
Current Actual Value Ch1 <Ch1 = ON, Ch2 = OFF> =>0.685208
Current Actual Value Ch2 <Ch1 = ON, Ch2 = OFF> =>0.079062
### READ CURRENT - CH1 = ON, CH2 = OFF, ST_BY = ON ###
IM1 = 0, IM2 = 0, FR_STBY = 1
Current Actual Value Ch1 <Ch1 = OFF, Ch2 = ON> =>0.026354
Current Actual Value Ch2 <Ch1 = OFF, Ch2 = ON>
```

In the output the term RAW means the analog value coming out the sense pin and going to STM32 ADC, while ACTUAL is the right current value upon factor correction of transfer function and resistor divider.

Note: The IAR project is available into the folders:

\Projects\STM32F4xx-Nucleo\Examples\EWARM\Project.ewp

Or

\Projects\STM32L0xx-Nucleo\Examples\EWARM\Project.ewp

The IAR project is provided with an embedded console activable by setting the macro `DEBUG_ON` in the `main.c` file. Viceversa, when the macro `DEBUG_ON` is disabled, `printf` functions over UART are implemented in the file `uart_support.c` and `uart_support.h`.

The basic operation done by the demo code, which can be used in the customer application are:

1. In order to use the singleton instance of the X_NUCLEO by calling class method `\Instance()\``:

```
// IPS expansion board singleton instance
static X_NUCLEO_IPS02A1 *ips_expansion_board =
X_NUCLEO_IPS02A1::Instance();
```

2. Switch-on or Switch-off loads output (Channel 1 or Channel 2) by setting or clearing associated digital input:

```
ips_expansion_board.vps2535h.Fr_Stby = 1; // set Fr_Stby pin
ips_expansion_board.vps2535h.In_1 = 1; // switch-on Channel 1
ips_expansion_board.vps2535h.In_2 = 0; // switch-off Channel 2
```

3. Read Current circulating on Channel 1 or Channel 2 and print on the Terminal

```
Multisense_Signal= ips_expansion_board.GetCurrent(CHANNEL_1);
printf("Current Ch1 = %2.3fA \n\r", Multisense_Signal);
Multisense_Signal= ips_expansion_board.GetCurrent(CHANNEL_2);
printf("Current Ch2 = %2.3fA \n\r", Multisense_Signal);
```

3 System setup guide

3.1 Hardware description

This section describes the hardware components needed for developing an IPS based application.

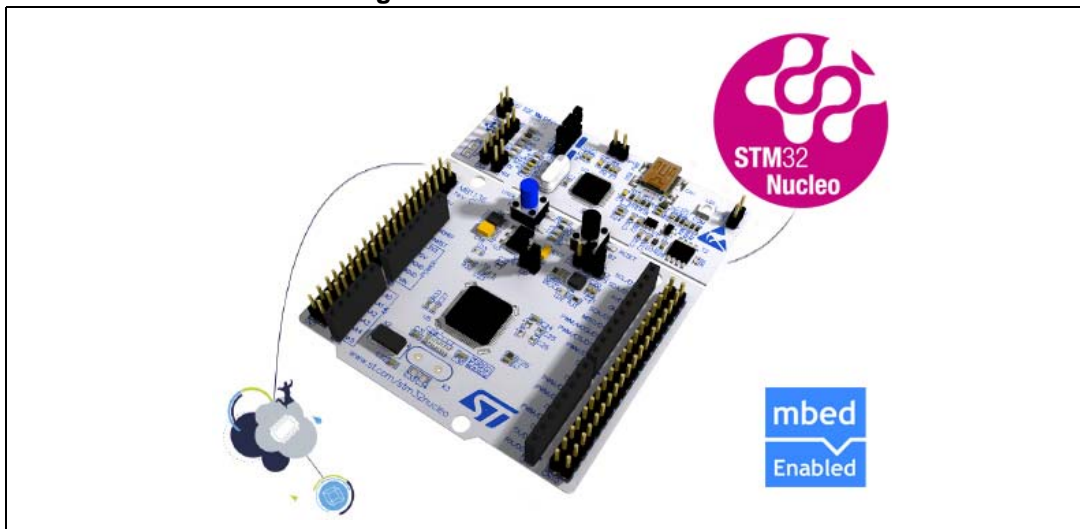
The following sub-sections describe the individual components.

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 choice of specialized expansion boards. 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 STM32 comprehensive software HAL library together with various packaged software examples.

Information about the STM32 Nucleo boards is available on www.st.com at <http://www.st.com/stm32nucleo>.

Figure 5. STM32 Nucleo Board



3.2 X-NUCLEO-IPS02A1 expansion board

The X-NUCLEO-IPS02A1 is a 24 V Intelligent Power Switch expansion board based on the VPS2535H, a double channel High Side Driver made using STMicroelectronics® VIPower technology.

X-NUCLEO-IPS02A1 provides a low cost, robust and easy-to-use solution for driving grounded loads, i.e. relay replacement, heating elements, solenoid driver, LED driver, light bulb driver, pumps and fans, in STM32 Nucleo projects.

The X-NUCLEO-IPS02A1 is compatible with the Arduino™ UNO R3 connector, and supports the addition of other expansion boards with a single STM32 Nucleo board. The user can also solder the ST morpho connectors.

In the boards some test points to allow the user to check and measure both analog and digital signals. Two red LEDs shown the output status; a green LED shows if the board is supplied properly.

Two X-NUCLEO-IPS02A1 boards can be plugged to drive up to 4 loads with a single STM32 Nucleo board

[Figure 6](#) and [Figure 7](#) show the position of the connectors and the jumpers respectively present in the board.

Figure 6. Connectors position

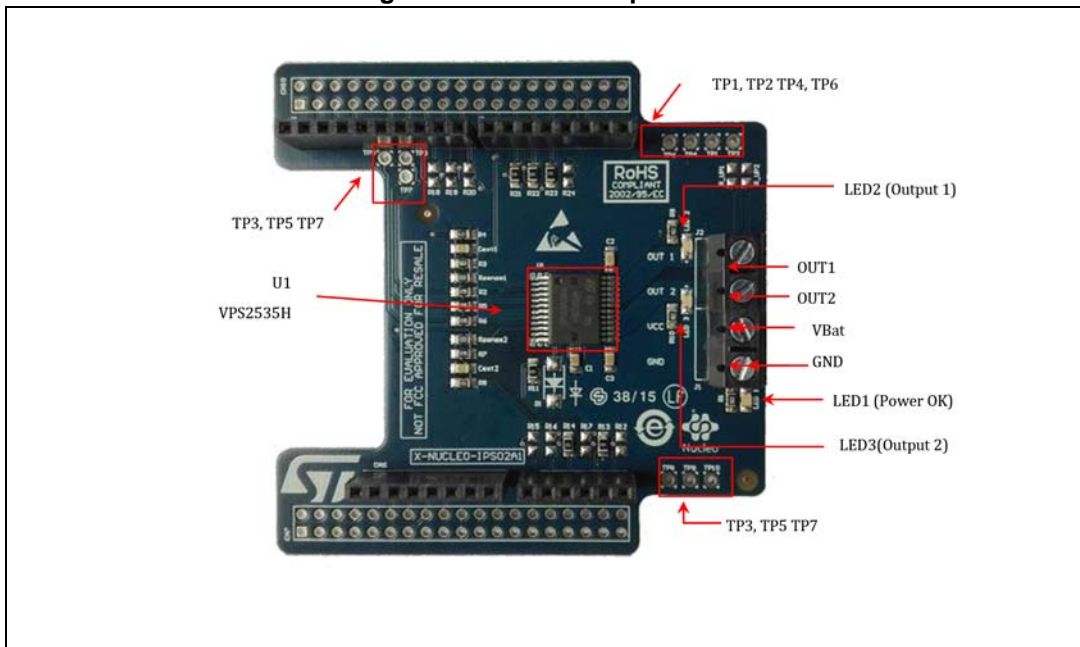


Figure 7. Jumpers position

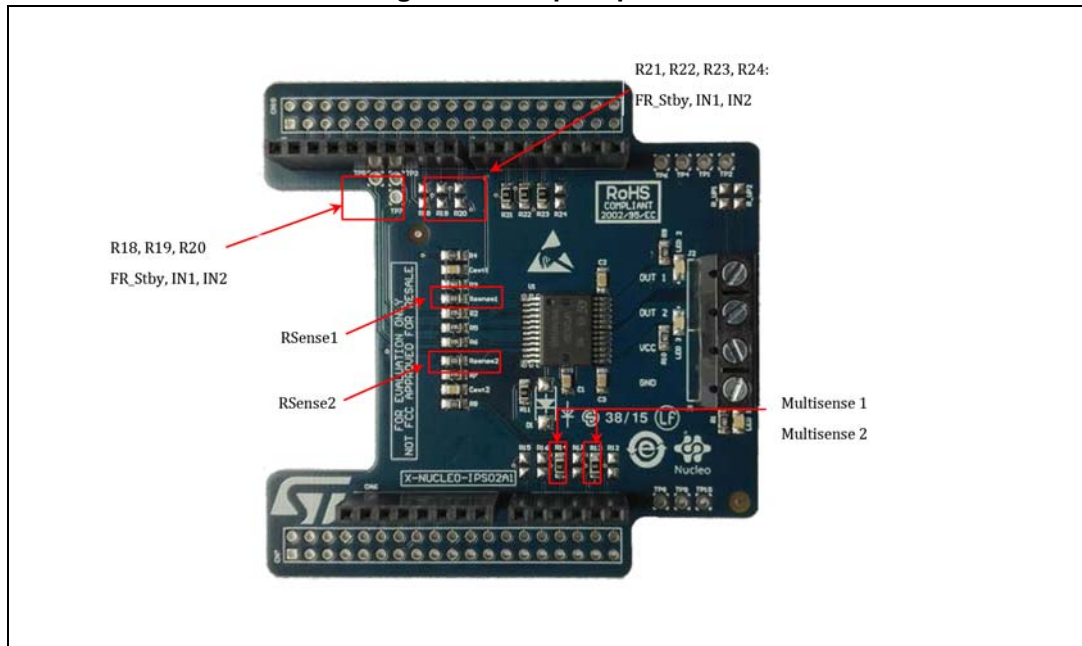


Table 1 details for the Arduino UNO R3 and the ST morpho connectors.

Table 1. Arduino UNO R3 connector table

Connector	Pin ⁽¹⁾	Signal	Remarks
CN5	1 (D8)	FR_Stby (option)	Option Configuration See Section 3.1: Jumpers configurations in UM2078 (see Section A.1: Document reference).
	2 (D9)	IN 1 (option)	
	3 (D10)	IN 2 (option)	
CN9	5 (D4)	FR_Stby (Default)	Option Configuration See Section 3.1: Jumpers configurations in UM2078 (see Section A.1: Document reference).
	6 (D5)	IN 1 (Default)	
	7 (D6)	IN 2 (Default)	
CN6	2	+3V3	
	5	+5V	
	6, 7	GND	
CN8	1 (A0)	Multisense_1 (Option)	See Section 3.1: Jumpers configurations in UM2078 (see Section A.1: Document reference)
	2 (A1)	Multisense_2 (Option)	
	3 (A2)	Multisense_1 (Default) GND_Sense (Option)	
	4 (A2)	Multisense_2 (Default)	
	5 (A4)	GND_Sense (Option)	

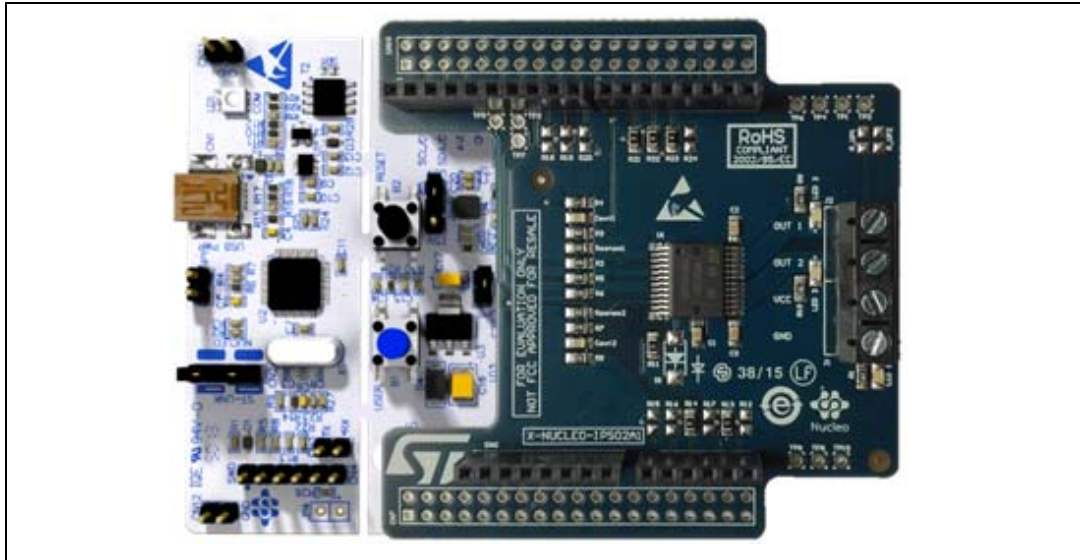
1. All the unlisted pins are not connected

Table 2. ST morpho connector table

Connector	Pin	Signal	Remarks
CN7	12 16	+3V3	
	14	NRST	
	18	+5V	
	8 19 20 22	GND	
	7	GND	
	28	Multisense_1 (option)	See Section 3.1: Jumpers configurations in UM2078 (see Section A.1: Document reference)
	30	Multisense_2 (option)	
	32	Multisense_1 (default) GND_Sense (option)	
	34	Multisense_2 (default)	
	36	GND_Sense (option)	
	CN10	9 20	GND
17		IN 2 (option)	See Section 3.1: Jumpers configurations in UM2078 (see Section A.1: Document reference)
19		IN 1 (option)	
21		FR_Stby (option)	
25		IN 2 (default)	
27		IN 1 (default)	
29		FR_Stby (default)	
32		A_GND	

Information about the X-NUCLEO-IPS02A1 expansion board is available on www.st.com.

Figure 8. X-NUCLEO-IPS02A1 expansion board connected to STM32 Nucleo board



3.3 Software description

The following software components are needed in order to setup the suitable development environment for creating applications for the STM32 Nucleo equipped with the IPS expansion board:

- X-CUBE-IPS02A1: an expansion for STM32Cube dedicated to Intelligent Power Switch (IPS) 24 V. The X-CUBE-IPS02 firmware and related documentation are 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
 - Real/View Microcontroller Development Kit (MDK-ARM) toolchain + ST-LINK
 - System Workbench for STM32 (SW4STM32) + ST-LINK

3.4 Hardware and software setup

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

3.4.1 Hardware setup

The following hardware components are needed:

1. STM32 Nucleo development platform (NUCLEO-F401RE or NUCLEO-L053R8)
2. One Dual-channel high side driver expansion board (order code: X-NUCLEO-IPS02A1)
3. USB type A to Mini-B USB cable to connect the STM32 Nucleo to the PC
4. Power Supply to connect to the Vbat and GND of the expansion board (CN1)
5. One or two loads (Motor, Light bulbs, pumps, fans,...or just resistors) connected to CN1

3.4.2 Software setup

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

Development Tool-chains and Compilers

Please select one of the Integrated Development Environments supported by the STM32Cube expansion software. Install a terminal emulator like Teraterm of Hyperlink and configure as specified later on this document.

Please read the system requirements and setup information provided by the selected IDE provider.

PC Utility

Supported IDE available in the package have following minimum requirements:

- PC with Intel or AMD processor running one of following Microsoft operating system:
 - Windows XP SP3
 - Windows Vista
 - Windows 7
 - Windows 8
 - At least 128 MBs of RAM
 - 2 USB ports
 - 40 MB of hard disk space

3.4.3 System setup guide

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

STM32 Nucleo and IPS 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 looking STSW-LINK008 or STSW-LINK009 on www.st.com (according to the Microsoft Windows OS).

The IPS expansion board X-NUCLEO-IPS02A1 can be easily connected to the STM32 Nucleo motherboard through the Arduino UNO R3 extension connector. The X-NUCLEO-IPS02A1 expansion board is capable of interfacing with the external STM32 Microcontroller on STM32 Nucleo using Inter-Integrated Circuit (IC) transport layer.

Terminal

Configure Hyper-terminal or whatever as following:

- Baud-rate = 9600
- Data = 8-bit
- Parity = none
- Stop= 1 bit
- Flow Control = none

Appendix A Further information

A.1 Document reference

- *Getting started with the 24 V Intelligent Power Switch expansion board based on VPS2535H for STM32 Nucleo* (UM2078, DocID029463).

Revision history

Table 3. Document revision history

Date	Revision	Changes
29-Aug-2016	1	Initial release.

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.

© 2016 STMicroelectronics – All rights reserved