Getting started with the X-CUBE-OUT8 industrial digital output software expansion for STM32Cube

Introduction

The X-CUBE-OUT8 expansion software package for STM32Cube runs on the STM32 microcontroller and includes a driver to control the IPS160HF and/or the IPS161HF (single high-side switches for industrial safety compliant systems).

The software provides an affordable and easy-to-use solution for the development of 2 A (X-NUCLEO-OUT08A1) or 0.5 A (X-NUCLEO-OUT10A1) digital output modules, letting you easily evaluate the driving and diagnostic capabilities of IPS160HF and IPS161HF with industrial loads.

The expansion is built on STM32Cube software technology to ease portability across different STM32 microcontrollers.

The software comes with an example implementation of a single channel driver running on the X-NUCLEO-OUT08A1 or X-NUCLEO-OUT10A1 expansion board connected to a NUCLEO-F401RE or NUCLEO-G431RB development board.

You can also evaluate a dual channel digital output module by stacking any combination of X-NUCLEO-OUT08A1 and/or X-NUCLEO-OUT10A1 with shared or independent supply rail and independent loads.

When two X-NUCLEO-OUT08A1 expansion boards are stacked and connected so that the output of the first board supplies the second, the software allows you to evaluate IPS160HF performance in a 2 A digital output module, also by using the additional signals OUT_FB and Nch_DRV.

Similarly, IPS161HF performance can be evaluated for a 0.5 A safety digital output module when two X-NUCLEO-OUT10A1 expansion boards are stacked.

When two expansion boards are stacked, the configuration resistors between signals and Arduino connectors must be properly set between default and alternate positions to guarantee the desired control of the hardware.

RELATED LINKS

Visit the STM32Cube ecosystem web page on www.st.com for further information

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
SPI	Serial peripheral interface

2 X-CUBE-OUT8 software expansion for STM32Cube

2.1 Overview

The X-CUBE-OUT8 software package expands the functionality provided by STM32Cube .

The key features of the package are:

- Complete software to build applications for IPS160HF and IPS161HF high-side smart power solid-state relay
- GPIOs, PWMs and IRQs
- Fault/Diagnostics interrupt handling
- Sample implementation available on the X-NUCLEO-OUT08A1 or X-NUCLEO-OUT10A1 expansion boards when connected to a NUCLEO-F401RE or NUCLEO-G431RB development board
- Easy portability across different MCU families, thanks to STM32Cube
- Free, user-friendly license terms

This software allows control of the digital output channel of a single expansion board (X-NUCLEO-OUT08A1 or X-NUCLEO-OUT10A1) or of two appropriately configured expansion boards (X-NUCLEO-OUT08A1 and/or X-NUCLEO-OUT10A1) stacked on a NUCLEO-F401RE or NUCLEO-G431RB development board.

It also allows you to program the expansion boards to be switched on and off using PWM with a specific frequency in the 0-100 Hz range (0.1 Hz resolution), and specific duty cycle in the 0-100% range (1% resolution).

If two expansion boards are used simultaneously, the specific frequency and duty cycle can be modified independently from the two board channels.

The package includes an example to test device functionality while driving the channels in steady state and PWM.

2.2 Architecture

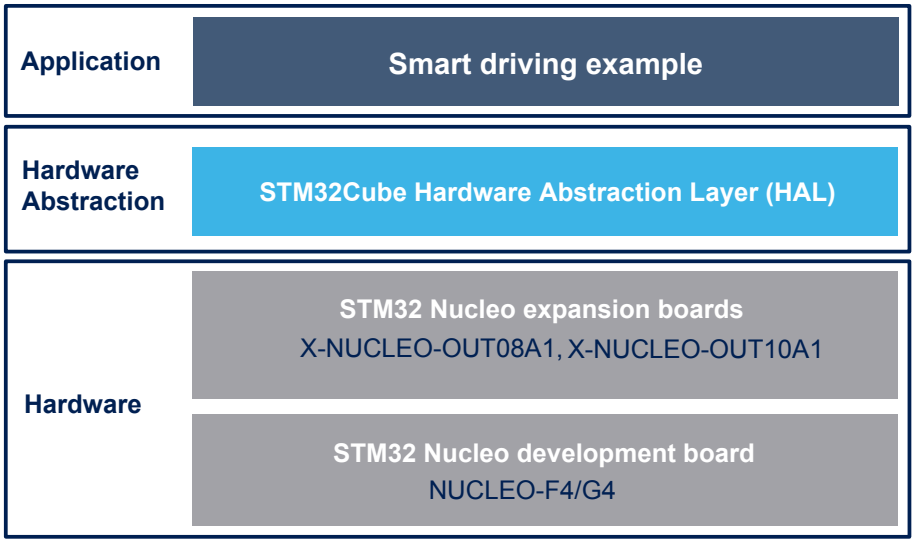
This software is a fully compliant expansion of STM32Cube architecture for the development of applications for single high side driver intelligent power switch (IPS) digital output modules.

The 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 digital output expansion boards based on IPS160HF and IPS161HF.

The software layers used by the application software to access and use the industrial digital output expansion boards are:

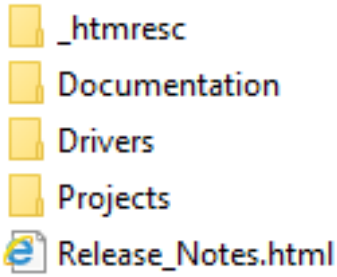
- **STM32Cube HAL layer**: consists of simple, generic and multi-instance APIs (application programming interfaces) which interact with the upper layer applications, libraries and stacks. These generic and extension APIs are based on a common framework so that overlying layers like middleware can function without requiring specific microcontroller unit (MCU) hardware information. This structure improves library code reusability and guarantees easy portability across other devices.
- **Board support package (BSP) layer**: provides software support for the STM32 Nucleo board peripherals, excluding the MCU. These specific APIs provide a programming interface for certain board specific peripherals like LEDs, user buttons, etc., and can also be used to fetch individual board version information. It also provides support for initializing, configuring and reading data.

Figure 1. **X-CUBE-OUT8 expansion software architecture**



2.3 Folder structure

Figure 2. **X-CUBE-OUT8 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:
  - an **STM32Cube HAL** folder, located in the subfolders STM32G4xx_HAL_Driver or STM32F4xx_HAL_Driver. These files are not described here as they are not specific to the X-CUBE-OUT8 software but come directly from the STM32Cube framework.
  - a **CMSIS** folder which contains the Cortex® microcontroller software interface standard files from ARM. These files are vendor-independent hardware abstraction layer for the Cortex-M processor series. This folder comes also unchanged from the STM32Cube framework.
  - a **BSP** folder containing the codes required for the X-NUCLEO-OUT08A1 and X-NUCLEO-OUT10A1 configuration, the IPS160HF and IPS161HF drivers and the relay API.
- **Projects** contains sample applications for IPS160HF and IPS161HF, provided for NUCLEO-F401RE and NUCLEO-G431RB platforms.

2.3.1 BSPs

For the X-CUBE-OUT8 software, different BSPs are used:

- STM32F4XX-Nucleo/STM32G4XX-Nucleo
- IPS160HF/IPS161HF
- OUT08_10

2.3.1.1 *STM32F4XX-Nucleo/STM32G4XX-Nucleo*

Depending of the STM32 Nucleo development board used, these BSPs provide an interface to configure and use the development board peripherals with the X-NUCLEO-OUT08A1 and X-NUCLEO-OUT10A1 expansion boards.

Each STM32F4xx-Nucleo/STM32G4xx-Nucleo subfolder contains couples of .c/.h files (stm32XXxx_nucleo.c/h) that come from the STM32Cube framework without modification and provide the functions to handle the user button and LEDs of the corresponding development board.

2.3.1.2 *IPS160HF/IPS161HF*

The IPS160HF/IPS161HF BSP component provides the driver functions for the IPS160HF or IPS161HF intelligent power switch drivers in the folder X-CUBE_IPS\Drivers\BSP\Components\ips160hf_161hf .

This folder contains:

- **ips160hf_161hf.c :** core functions of the IPS160HF/IPS161HF drivers
- **ips160hf_161hf.h :** declaration of the IPS160HF/IPS161HF driver functions and their associated definitions

2.3.1.3 *OUT08_10*

The OUT08_10 BSP component contains a couple of **out08_10_switch.c/h** files, which are dedicated to the functions necessary to drive the power switch in steady-state and in PWM mode using GPIOs.

The files are also used to obtain the status of the diagnostic and output feedback pins.

Through these functions, the channel can be set on/off or configured with a specific frequency and duty cycle PWM mode.

2.3.2 Projects

For each STM32 Nucleo platform, one example project is available in the folders:

- \X-CUBE_IPS\Projects\STM32F401RE-Nucleo\Examples\Out08_10
- \X-CUBE_IPS\Projects\STM32G431RB-Nucleo\Examples\Out08_10

Each example has a folder dedicated to the targeted IDE:

- **EWARM** containing the project files for IAR
- **MDK-ARM** containing the project files for Keil
- **STM32CubeIDE** containing the project files for OpenSTM32

Each example also contains the following code files:

- **Inc\main.h**: main header file
- **Inc\out08_10_conf.h**: header file device specific function prototypes
- **Inc\ stm32f4xx_hal_conf.h** or **stm32g4xx_hal_conf.h**: HAL configuration file
- **Inc\stm32f4xx_it.h** or **stm32g4xx_it.h**: header for the interrupt handler
- **Src\main.c**: main program (code of the example based on the library for IPS160HF and IPS161HF)
- **Src\out08_10_conf.c**: device specific function implementation
- **Src\stm32f4xx_hal_msp.c** or **stm32g4xx_hal_msp.c**: code for the MSP initialization and de-initialization
- **Src\stm32f4xx_it.c** or **stm32g4xx_it.c**: interrupt handler
- **Src\system_stm32f4xx.c** or **system_stm32g4xx.c**: system initialization

2.4 Software required resources

The MCU controls IPS160HF and IPS161HF via GPIOs.

Thus, when using one X-NUCLEO-OUT08A1 expansion board and one X-NUCLEO-OUT10A1 expansion board, two GPIO signals (IN, NCh_DRV pins) plus two GPIOs dedicated to the interrupt management (DIAG, OUT_FB pins) are needed.

The two expansion boards must be properly configured: in the second expansion board, it is necessary to unsolder four resistors and solder them in a new position (refer to the readme.txt documentation file for details). It is possible to choose one of the two alternative configurations.

To handle the DIAG and OUT_FB signals, the software enables the associated interrupt lines.

The software also uses a PWM timer (TIM3) to generate the periodic patterns on the different output channels for the the expansion boards.

2.5 APIs

The X-CUBE-OUT8 software API is defined in the BSP\OUT08_10\out08_10_switch.h file. Its functions are prefixed by `OUT08_10_SWITCH_`.

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.

2.6 Sample application description

A sample application using the X-NUCLEO-OUT08A1 or X-NUCLEO-OUT10A1 expansion board with either NUCLEO-F401RE or NUCLEO-G431RB boards is provided in the “Projects” directory. Ready to be built projects are available for multiple IDEs.

In this example, a sequence of commands are applied to the IN and Nch_DRV channels of X-NUCLEO-OUT08A1 or X-NUCLEO-OUT10A1 expansion boards. An operation change is requested by a user button press.

At startup, the IN and Nch_DRV channels are all switched off. Each time the user button is pressed, the program performs a consecutive action in the sequence below:

1. switches ON the IN channel
2. switches OFF the IN channel
3. switches ON the Nch_DRV channel
4. switches OFF the Nch_DRV channel
5. switches ON the IN and Nch_DRV channels
6. switches OFF the IN and Nch_DRV channels
7. switches ON the IN channel in PWM mode
8. changes PWM duty cycle
9. changes PWM frequency
10. changes duty cycle to 100%
11. changes duty cycle to 0%
12. switches OFF PWM mode
13. sequence restarts at step 1

3 System setup guide

3.1 Hardware description

3.1.1 STM32 Nucleo

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

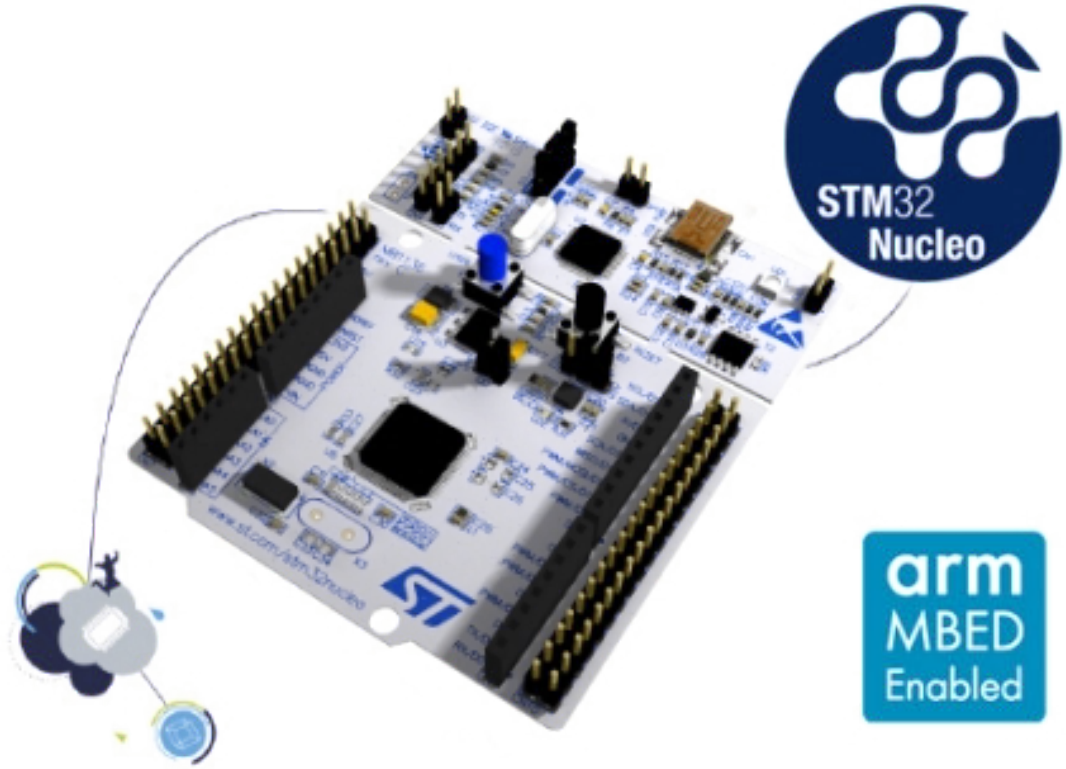
The Arduino™ connectivity support and ST morpho connectors make it easy to expand the functionality of the STM32 Nucleo open development platform with a wide range of specialized expansion boards to choose from.

The NUCLEO-F401RE development board does not require separate probes as it integrates the ST-LINK/V2-1 debugger/programmer.

The NUCLEO-G431RB development board does not require separate probes as it integrates the STLINK-V3 debugger/programmer.

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

Figure 3. STM32 Nucleo board



3.1.2 X-NUCLEO-OUT08A1 expansion board

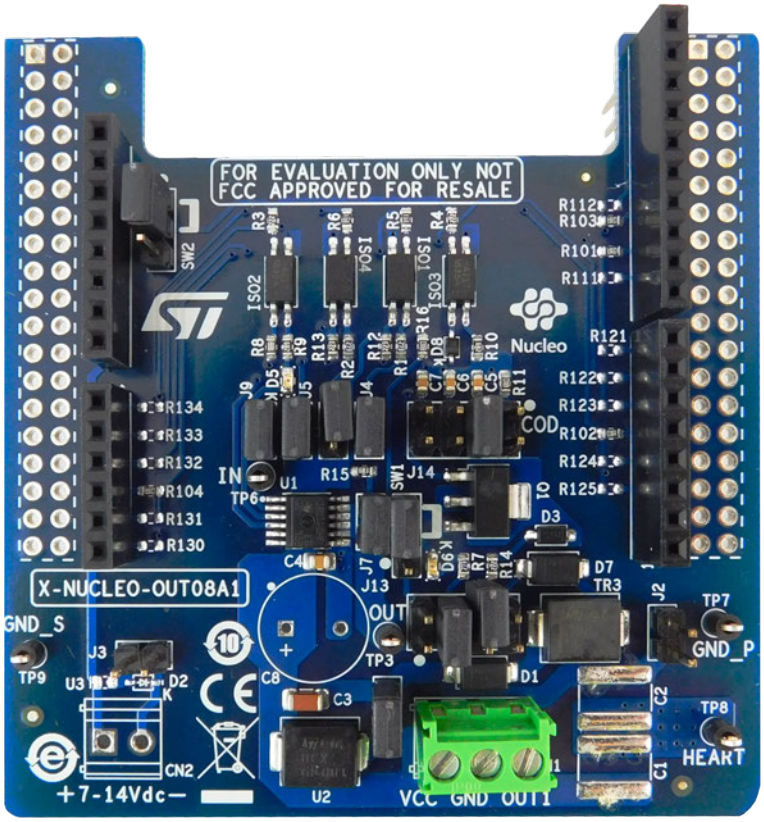
The X-NUCLEO-OUT08A1 industrial digital output expansion board for STM32 Nucleo provides a powerful and flexible evaluation and development environment for 2 A (typ.) digital output modules, featuring the safe driving and smart diagnostic capabilities of the IPS160HF single high-side switch.

The X-NUCLEO-OUT08A1 interfaces with the microcontroller on the STM32 Nucleo via 3 kV optocouplers driven by GPIO pins and Arduino™ UNO R3 (default configuration) and ST morpho (optional, not mounted) connectors.

The expansion board should be connected to either a NUCLEO-F401RE or NUCLEO-G431RB development board, and can also be stacked with another X-NUCLEO-OUT08A1 or X-NUCLEO-OUT10A1.

Two X-NUCLEO-OUT08A1 expansion boards allows you to evaluate a dual channel digital output module with 2 A (typ.) capability each, or a 2 A (typ.) single channel safety digital output module. In the second scenario, the first shield output is connected to the supply of the second one. Dedicated on-board hardware can be enabled or disabled to activate fast discharge of high capacitive loads, output voltage sensing and additional surge pulse output line protection.

Figure 4. X-NUCLEO-OUT08A1 expansion board



3.1.3 X-NUCLEO-OUT10A1 expansion board

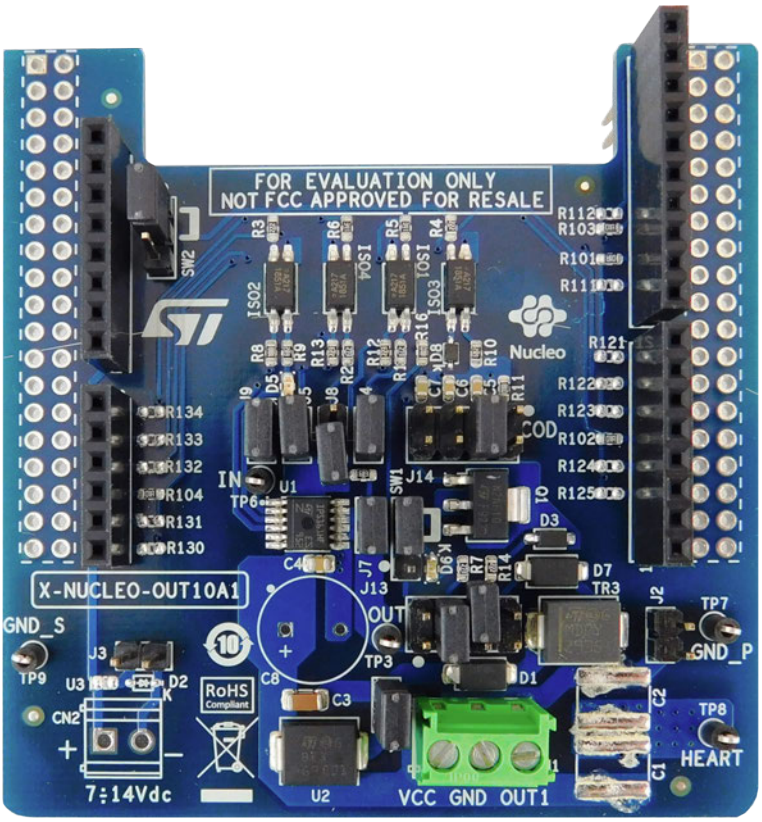
The X-NUCLEO-OUT10A1 industrial digital output expansion board for STM32 Nucleo provides an affordable and easy-to-use solution for the development of 0.5 A (typ.) digital output modules, letting you easily evaluate the IPS161HF driving and diagnostic capabilities with industrial loads.

The X-NUCLEO-OUT10A1 interfaces with the microcontroller on the STM32 Nucleo via 3 kV optocouplers driven by GPIO pins and Arduino™ UNO R3 (default configuration) and ST morpho (optional, not mounted) connectors.

The expansion board should be connected to either a NUCLEO-F401RE or NUCLEO-G431RB development board, and can be stacked with another X-NUCLEO-OUT10A1 or an X-NUCLEO-OUT08A1 .

Two X-NUCLEO-OUT10A1 expansion boards allows you to evaluate a dual channel digital output module with 0.5 A (typ.) capability each, or a 0.5 A (typ.) single channel safety digital output module. In the second scenario, the first shield output is connected to the supply of the second one. Dedicated on-board hardware can be enabled or disabled to activate fast discharge of high capacitive loads, output voltage sensing and additional surge pulse output line protection.

Figure 5. **X-NUCLEO-OUT10A1 expansion board**



3.2 Hardware setup

The following hardware components are needed:

1. One USB type A to Mini-B USB cable to connect the STM32 Nucleo to the PC when using a NUCLEO-F401RE
2. One USB type A to Micro-B USB cable when using a NUCLEO-G431RB
3. An external power supply (8 - 33 V) and the associated wires to supply the X-NUCLEO-OUT08A1 or X-NUCLEO-OUT10A1 expansion boards

3.3 Software setup

The following software components are to set up a suitable development environment for creating applications for the STM32 Nucleo equipped with the X-NUCLEO-OUT08A1 or X-NUCLEO-OUT10A1 industrial digital output expansion boards:

- X-CUBE-OUT8: an expansion for STM32Cube dedicated to applications development which require the use of IPS160HF or IPS161HF. The X-CUBE-OUT8 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-STR) toolchain + ST-LINK
  - STM32CubeIDE + ST-LINK

3.4 Board setup

The STM32 Nucleo must be configured with the following jumper positions:

- JP1 off
- JP5 (PWR) on UV5 side for NUCLEO-F401RE, on 5V_STLK for NUCLEO-G431RB

- JP6 (IDD) on

The X-NUCLEO-OUT08A1 or X-NUCLEO-OUT10A1 expansion board must be configured in the following way:

- J1, J4, J5, J7, J8, J9 closed
- J13 closed: 1-2, 3-4, 5-6
- J14 closed: 1-2, 3-4
- SW1: 2-3
- SW2: 1-2
- All other jumpers open

Step 1. Plug the X-NUCLEO-OUT08A1 or X-NUCLEO-OUT10A1 expansion board on top of theSTM32 Nucleo via the Arduino UNO connectors.

Figure 6. X-NUCLEO-OUT08A1 expansion board connected to an STM32 Nucleo development board

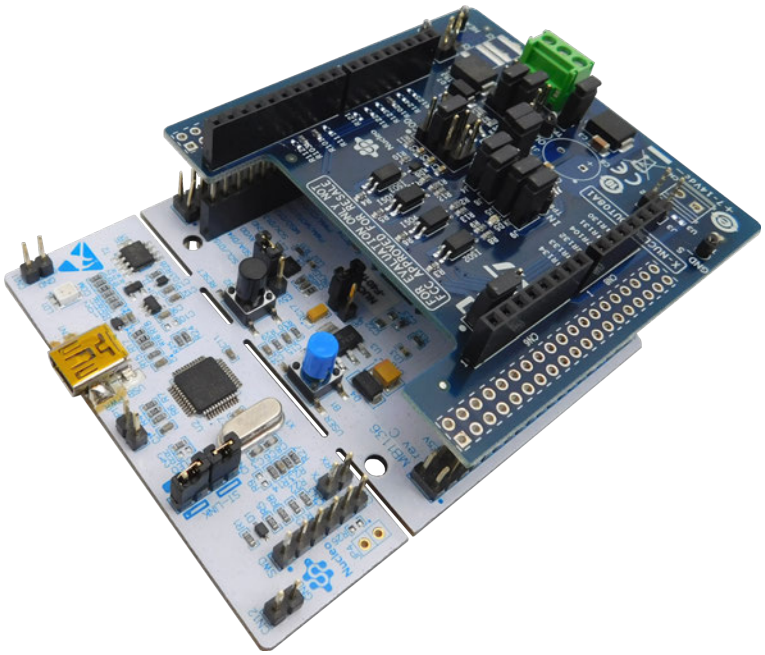
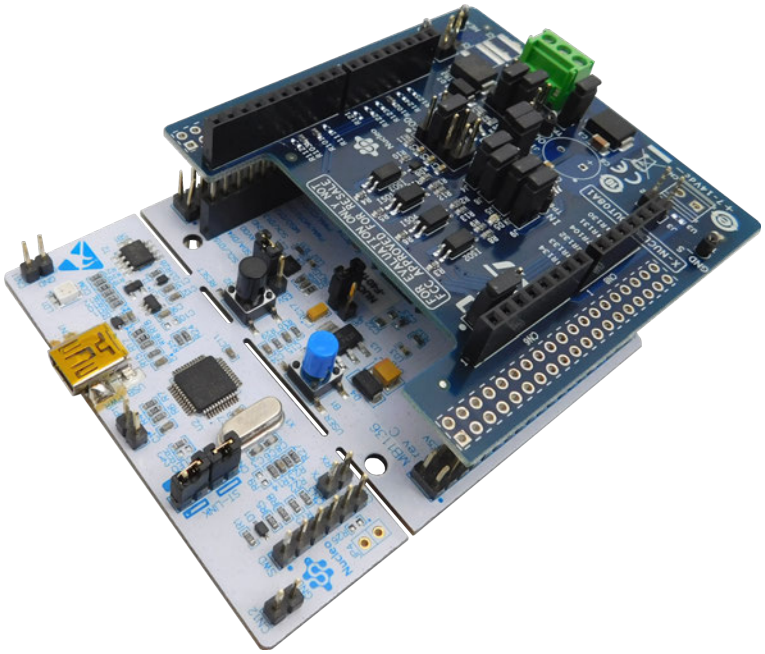


Figure 7. X-NUCLEO-OUT10A1 expansion board connected to an STM32 Nucleo development board



Step 2. Power the STM32 Nucleo board via USB cable between connector CN1 and a PC USB port.

Step 3. Power the X-NUCLEO-OUT08A1 or X-NUCLEO-OUT10A1 expansion board on by connecting its connectors CN1 1(V_{CC}), 2(GND) to the DC power supply (which must be set between 8 and 33 V).

Step 4. Open your preferred toolchain (MDK-ARM from Keil, EWARM from IAR, or STM32CubeIDE)

Step 5. Depending on the STM32 Nucleo board used, open the software project from:

- \Projects\STM32F401RE-Nucleo\Examples\Out08_10 for NUCLEO-F401RE
- \Projects\STM32G431RB-Nucleo\Examples\Out08_10 for NUCLEO-G431RB

Step 6. Rebuild all files and load your image into target memory.

Step 7. Run the example.

Each time the user button is pressed, a new command is applied at the digital output as described in Section 2.6 Sample application description.

Revision history

Table 2. **Document revision history**

Date	Version	Changes
12-Jun-2020	1	Initial release.

Contents

1 **Acronyms and abbreviations** 2
2 **X-CUBE-OUT8 software expansion for STM32Cube** 3
2.1 Overview 3
2.2 Architecture 3
2.3 Folder structure 4
2.3.1 BSPs 4
2.3.2 Projects 5
2.4 Software required resources 5
2.5 APIs 6
2.6 Sample application description 6
3 **System setup guide** 7
3.1 Hardware description 7
3.1.1 STM32 Nucleo 7
3.1.2 X-NUCLEO-OUT08A1 expansion board 7
3.1.3 X-NUCLEO-OUT10A1 expansion board 8
3.2 Hardware setup 9
3.3 Software setup 9
3.4 Board setup 9
Revision history 13

List of tables

Table 1. List of acronyms 2
Table 2. Document revision history 13

List of figures

Figure 1. X-CUBE-OUT8 expansion software architecture 4
Figure 2. X-CUBE-OUT8 package folder structure 4
Figure 3. STM32 Nucleo board 7
Figure 4. X-NUCLEO-OUT08A1 expansion board 8
Figure 5. X-NUCLEO-OUT10A1 expansion board 9
Figure 6. X-NUCLEO-OUT08A1 expansion board connected to an STM32 Nucleo development board 11
Figure 7. X-NUCLEO-OUT10A1 expansion board connected to an STM32 Nucleo development board 11