UM2864

User manual

Getting started with X-CUBE-OUT5 industrial digital output software for STM32 Nucleo

Introduction

The X-CUBE-OUT5 expansion software package for STM32Cube runs on the STM32 microcontroller and includes a driver to control the IPS1025H and/or the IPS1025H-32 high efficiency single high-side switches with extended diagnostics and smart driving for capacitive loads.

The software provides an affordable and easy-to-use solution for the development of 2.5 A (X-NUCLEO-OUT05A1) or 5.7 A (X-NUCLEO-OUT06A1) digital output modules, letting you easily evaluate the driving and diagnostic capabilities of IPS1025H and IPS1025H-32 with industrial loads.

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

The software comes with a sample implementation of a single channel driver running on the X-NUCLEO-OUT05A1 or X-NUCLEO-OUT06A1 expansion board connected to a NUCLEO-F401RE or NUCLEO-G431RB development board.

You can stack up to four expansion boards, X-NUCLEO-OUT05A1 and/or X-NUCLEO-OUT06A1 with shared or independent supply rail and independent loads, to evaluate up to a four-channel digital output module.

When two or more expansion boards are stacked, the configuration resistors between signals and Arduino connectors of the second, third or fourth board must be properly placed in 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
SPI	Serial peripheral interface

2 X-CUBE-OUT5 software expansion for STM32Cube

2.1 Overview

The X-CUBE-OUT5 software package expands the functionality provided by STM32Cube.

The key features of the package are:

- Complete software to build applications for IPS1025H and IPS1025H-32 high efficiency single high-side switches
- GPIOs, PWMs and IRQs
- Fault/Diagnostics interrupt handling
- Sample implementation available on the X-NUCLEO-OUT05A1 or X-NUCLEO-OUT06A1 expansion boards when connected to a NUCLEO-F401RE or NUCLEO-G431RB development boards
- Easy portability across different MCU families, thanks to STM32Cube
- Free, user-friendly license terms

This software enables control of the digital output channel of a single channel expansion board (X-NUCLEO-OUT05A1 or X-NUCLEO-OUT06A1) or of up to four appropriately configured expansion boards (X-NUCLEO-OUT05A1 and/or X-NUCLEO-OUT06A1) 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).

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

2.2 Architecture

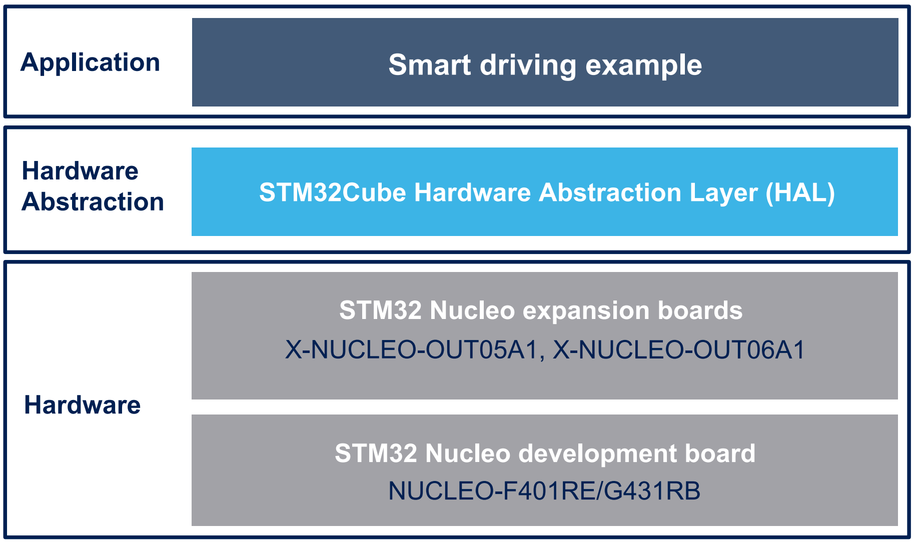
This software is a fully compliant expansion of STM32Cube architecture for the development of applications for single high-side driver intelligent power switch (IPS) digital output modules.

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 IPS1025H and IPS1025H-32.

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

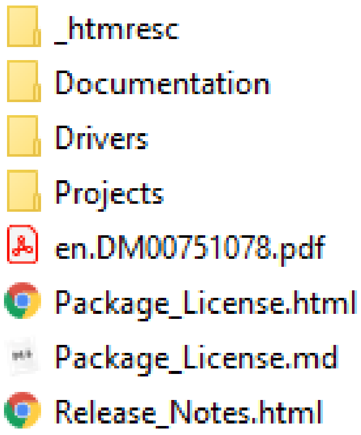
- **STM32Cube HAL layer**: consists of simple, generic and multi-instance APIs (application programming interfaces) which interact with the upper layer applications, libraries and stacks. These generic and extension APIs are based on a common framework so that overlying layers like middleware can function 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-OUT5 software architecture**


Application
Smart driving example
Hardware Abstraction
STM32Cube Hardware Abstraction Layer (HAL)
Hardware
STM32 Nucleo expansion boards
X-NUCLEO-OUT05A1, X-NUCLEO-OUT06A1
STM32 Nucleo development board
NUCLEO-F401RE/G431RB

2.3 Folder structure

Figure 2. **X-CUBE-OUT5 package folder structure**



_htmresc
Documentation
Drivers
Projects
en.DM00751078.pdf
Package_License.html
Package_License.md
Release_Notes.html

The following folders are included in the software package:

- **_htmresc** contains graphics for html pages.
- **Documentation** contains a compiled HTML file generated from the source code, detailing the software components and APIs.
- **Drivers** contains:
  - **STM32Cube HAL** sub-folders, located in the subfolders (STM32G4xx_HAL_Driver and STM32F4xx_HAL_Driver). These files are not specific to the X-CUBE-OUT5 software but come directly from the STM32Cube framework and represent the Hardware Abstraction Layer code for the STM32 MCUs.
  - 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-OUT05A1 and X-NUCLEO-OUT06A1 configuration, the IPS1025H and IPS1025H-32 drivers and the switch API functions.
- **Projects** contains sample applications for IPS1025H and IPS1025H-32, provided for NUCLEO-F401RE and NUCLEO-G431RB platforms.

2.3.1 BSPs

For the X-CUBE-OUT5 software, different BSPs are used:

1. STM32F4xx-Nucleo/STM32G4xx_Nucleo
2. IPS1025H_2050H
3. OUT0xA1

2.3.1.1 *STM32F4xx-Nucleo/STM32G4xx_Nucleo*

Depending on the STM32 Nucleo development board used, these BSPs provide an interface to configure and use the development board peripherals with the X-NUCLEO-OUT05A1 and X-NUCLEO-OUT06A1 expansion boards.

Each STM32FXX-Nucleo sub-folder contains couples of .c/.h files (stm32XXxx_nucleo.c/h) which come from the STM32Cube framework without modification and provide the functions to handle the corresponding user button and LEDs of the corresponding development board.

2.3.1.2 *IPS1025H_2050H*

The IPS1025H_2050H BSP component provides the driver functions for the STMicroelectronics intelligent power switch devices in the X-CUBE_IPS\Drivers\BSP\Components\ips1025h_2050h folder, which contains:

- **ips1025h_2050h.c**: core functions of the IPS1025H and IPS1025H-32 drivers
- **ips1025h_2050h.h**: declaration of the IPS1025H and IPS1025H-32 driver functions and their associated definitions

2.3.1.3 *OUT0xA1*

The OUT0xA1 BSP component contains board support package files for X-NUCLEO-OUT0xA1 board family (X-NUCLEO-OUT05A1, X-NUCLEO-OUT06A1, etc.), which are dedicated to the functions necessary to drive the power switches in steady-state and in PWM mode using GPIOs.

The files are also used to obtain the status of diagnostics and output feedback pins. Through these functions, the channel can be set and reset or configured in PWM mode with a specific frequency and duty cycle.

2.3.2 Projects

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

- Projects\STM32F401RE-Nucleo\Examples\Out05_06
- Projects\STM32G431RB-Nucleo\Examples\Out05_06

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:

- **Projects\STM32F401RE-Nucleo\Examples\Out05_06**
  - **Inc\main.h** - Header for main.c module
  - **Inc\out05_06a1_conf.h** - Header for BSP/OUT0xA1 driver configuration
  - **Inc\app_switch.h** - Header for app_switch.c module
  - **Inc\stm32f4xx_hal_conf.h** - HAL configuration file for STM32F4xx
  - **Inc\stm32f4xx_it.h** - Interrupt handlers header file for STM32F4xx
  - **Inc\stm32f4xx_nucleo_errno.h** - Error codes for STM32F4xx-Nucleo
  - **Inc\ips1025h_conf.h** - Header for BSP/Components/ips1025h_2050h driver configuration
  - **Src\main.c** - Main program
  - **Src\app_switch.c** - Init and switch functions
  - **Src\stm32f4xx_hal_msp.c** - HAL MSP module for STM32F4xx
  - **Src\stm32f4xx_it.c** - Interrupt handlers for STM32F4xx
  - **Src\system_stm32f4xx.c** - System source file for STM32F4xx

- **Projects\STM32G431RB-Nucleo\Examples\Out05_06**
  - **Inc\main.h** - Header for main.c module
  - **Inc\out05_06a1_conf.h** - Header for BSP/OUT0xA1 driver configuration
  - **Inc\app_switch.h** - Header for app_switch.c module
  - **Inc\stm32g4xx_hal_conf.h** - HAL configuration file for STM32G4xx
  - **Inc\stm32g4xx_it.h** - Interrupt handlers header file for STM32G4xx
  - **Inc\stm32g4xx_nucleo_conf.h** - Configuration file for STM32G4xx-Nucleo
  - **Inc\ips1025h_conf.h** - Header for BSP/Components/ips1025h_2050h driver configuration
  - **Src\main.c** - Main program
  - **Src\app_switch.c** - Init and switch functions
  - **Src\stm32g4xx_hal_msp.c** - HAL MSP module for STM32G4xx
  - **Src\stm32g4xx_it.c** - Interrupt handlers for STM32G4xx
  - **Src\system_stm32g4xx.c** - System source file for STM32G4xx

2.4 Software required resources

The MCU controls IPS1025H and IPS1025H-32 via GPIOs.

Thus, when using one X-NUCLEO-OUT05A1 expansion board or one X-NUCLEO-OUT06A1 expansion board, one GPIO signal (IN1 pin) plus two GPIOs dedicated to the interrupt management (FLT1, FLT2 pins) are required.

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

It is also possible to evaluate a four channel digital output module by stacking up to four X-NUCLEO-OUT05A1 and\or X-NUCLEO-OUT06A1 with shared or independent supply rail and independent loads.

In this case, the additional expansion boards must be properly configured: for the second, third or fourth board, it is necessary to unsolder three resistors for each board from the default position and solder them in different positions related to the board number following the scheme described below.

Table 2. **Configuration of a stack of four expansion boards**

Board no.	IN1	FLT1	FLT2
Board 0	R101	R103	R114
Board 1	R102	R104	R117
Board 2	R115	R116	R107
Board 3	R120	R119	R118

For further details, see the jumper configuration described in Section 3.4 Board setup and the documentation file (readme.txt).

2.5 APIs

The X-CUBE-OUT5 API software is defined in the Drivers\BSP\OUT0xA1\out0xa1.h file. Its functions are prefixed by OUT03_05_SWITCH_.

Detailed technical information about the APIs available to the user can be found in a compiled HTML Help (chm) 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-OUT05A1 or X-NUCLEO-OUT06A1 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 is applied to the X-NUCLEO-OUT05A1 or X-NUCLEO-OUT06A1 input channel. An operation change is requested by pressing the user button.

At startup, the IN1 channel is switched off. Each time the user button is pressed, the program performs a consecutive action as in the sequence below:

1. set ON IN1 pin on boards 0-2, set OFF IN1 pin on boards 1-3
2. set ON IN1 pin on boards 1-3, set OFF IN1 pin on boards 0-2
3. set ON IN1 pin on all boards
4. set OFF IN1 pin on all boards
5. start PWM on IN1 pin on all boards with different frequency and duty cycle settings:
   - IN1 pin boards 0-3: PWM ON with frequency 2 Hz, DC 25%
   - IN1 pin boards 1-2: PWM ON with frequency 1 Hz, DC 25%
6. IN1 pin on all boards: set DC 50%
7. IN1 pin on all boards: set DC 75%
8. IN1 pin on all boards: set DC 100%
9. stop PWM on IN1 pin on all boards
10. 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.

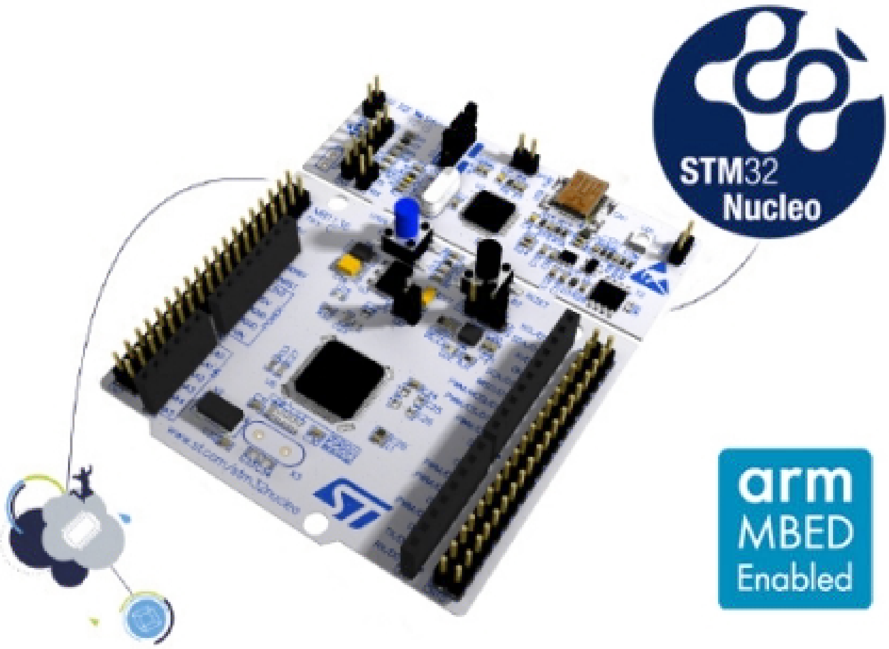
The Arduino connectivity support and ST morpho connectors make it easy to expand the functionality of the STM32 Nucleo open development platform with a wide range of specialized expansion boards to choose from.

The 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 for different IDEs (IAR EWARM, Keil MDK-ARM, STM32CubeIDE, mbed and GCC/ LLVM).

All STM32 Nucleo users have free access to the mbed online resources (compiler, C/C++ SDK and developer community) at www.mbed.org to easily build complete applications.

Figure 3. STM32 Nucleo board



3.1.2 X-NUCLEO-OUT05A1 expansion board

The X-NUCLEO-OUT05A1 industrial digital output expansion board for STM32 Nucleo provides a powerful and flexible environment for the evaluation of the driving and diagnostic capabilities of the IPS1025H (single high-side smart power solid state relay) in a digital output module connected to 2.5 A (max.) industrial loads.

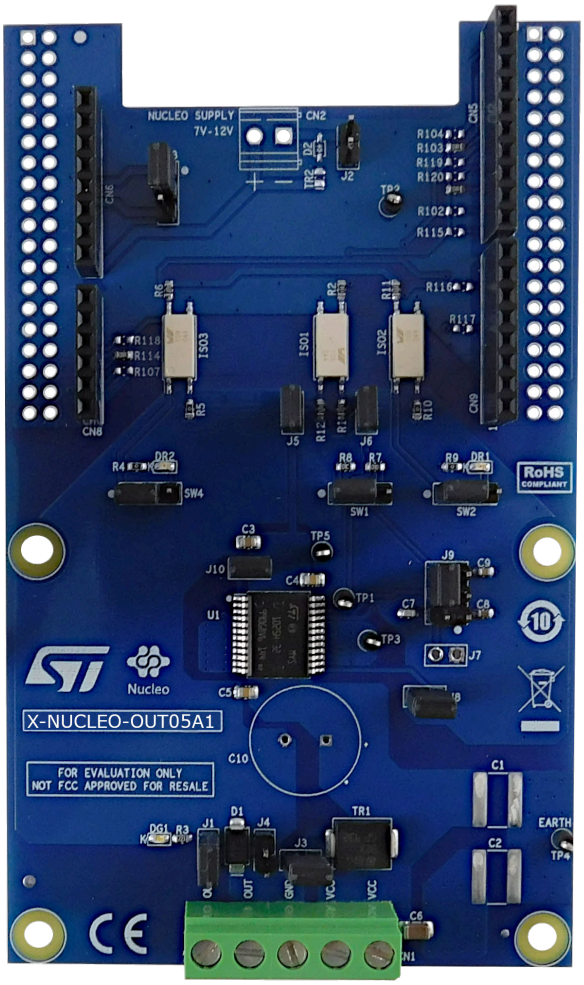
The X-NUCLEO-OUT05A1 interfaces with the microcontroller on the STM32 Nucleo via 5 kV optocouplers driven by GPIO pins and Arduino UNO R3 (default configuration) and ST morpho (optional, not mounted) connectors.

The expansion board can be connected to either a NUCLEO-F401RE or NUCLEO-G431RB development board.

It is also possible to evaluate a system composed of up to four stacked X-NUCLEO-OUT05A1 expansion boards.

As an example, a system with four X-NUCLEO-OUT05A1 expansion boards allows you to evaluate a four-channel digital output module with 2.5 A (max.) capability each.

Figure 4. X-NUCLEO-OUT05A1 expansion board



3.1.3 X-NUCLEO-OUT06A1 expansion board

The X-NUCLEO-OUT06A1 industrial digital output expansion board for STM32 Nucleo provides a powerful and flexible environment for the evaluation of the driving and diagnostic capabilities of the IPS1025H-32 (single high-side smart power solid state relay) in a digital output module connected to 5.7 A (max.) industrial loads.

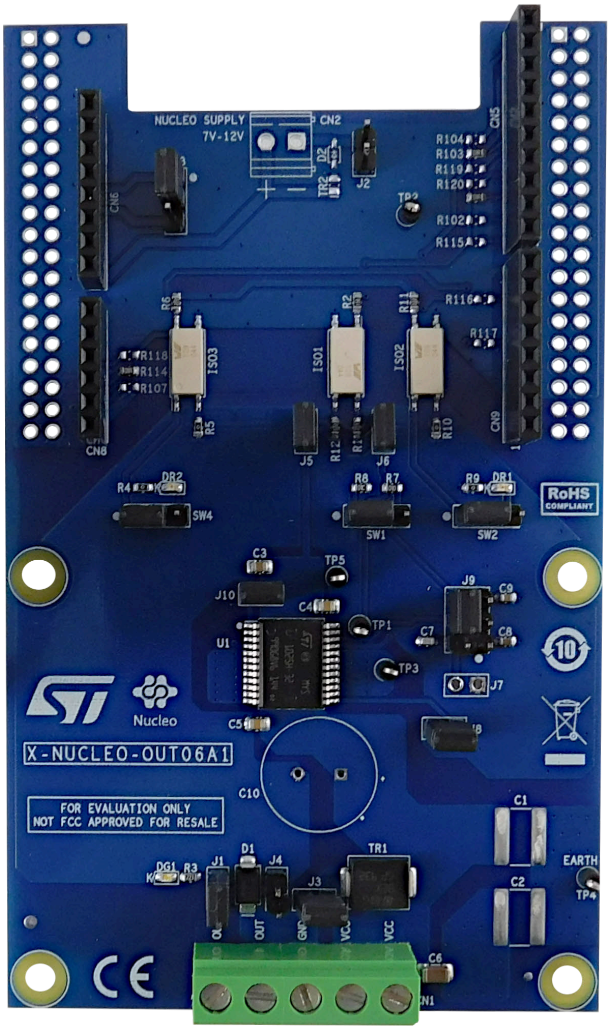
The X-NUCLEO-OUT06A1 interfaces with the microcontroller on the STM32 Nucleo via 5 kV optocouplers driven by GPIO pins and Arduino UNO R3 (default configuration) and ST morpho (optional, not mounted) connectors.

The expansion board can be connected to either a NUCLEO-F401RE or NUCLEO-G431RB development board.

It is also possible to evaluate a system composed by up to four stacked X-NUCLEO-OUT06A1 expansion boards.

As an example, a system with four X-NUCLEO-OUT06A1 expansion boards allows you to evaluate a four-channel digital output module with 5.7 A (max.) capability each.

Figure 5. X-NUCLEO-OUT06A1 expansion board



3.2 Hardware setup

The following hardware components are required:

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-OUT05A1 or X-NUCLEO-OUT06A1 expansion boards

3.3 Software setup

The following software components are required to set up a suitable development environment for creating applications for the STM32 Nucleo equipped with the X-NUCLEO-OUT05A1 or X-NUCLEO-OUT06A1 industrial digital output expansion boards:

- X-CUBE-OUT5: an expansion for STM32Cube dedicated to application development which requires the use of IPS1025H or IPS1025H-32. The X-CUBE-OUT5 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:

- NUCLEO-F401RE
  - JP5 on U5V for firmware flashing
  - JP1 open
  - JP6 closed
  - CN2: closed 1-2, 3-4
  - CN3 open
  - CN4 open
  - CN11 closed
  - CN12 closed
- NUCLEO-G431RB
  - JP5 closed 1-2 (5V_STLK for firmware flashing)
  - JP1, JP7 open
  - JP3, JP6 closed
  - JP8 closed 1-2
  - CN4 open
  - CN11 closed
  - CN12 closed

The X-NUCLEO-OUT05A1 or X-NUCLEO-OUT06A1 expansion board must be configured in the following way:

- SW1 1-2
- SW2
  - close 1-2 to route FLT1 signal from device to microcontroller only
  - close 2-3 to drive the DR1 red LED only
- SW3 1-2
- SW4
  - close 1-2 to route FLT2 signal from device to microcontroller only
  - close 2-3 to drive the DR2 red LED only
- J1, J3, J5, J6, J8, J10 closed
- J2, J4, J7 open
- J9 closed 4-6

Step 1. Plug the X-NUCLEO-OUT05A1 or X-NUCLEO-OUT06A1 expansion board on top of the STM32 Nucleo via the Arduino connectors.

Figure 6. **X-NUCLEO-OUT05A1 expansion board connected to an STM32 Nucleo development board**

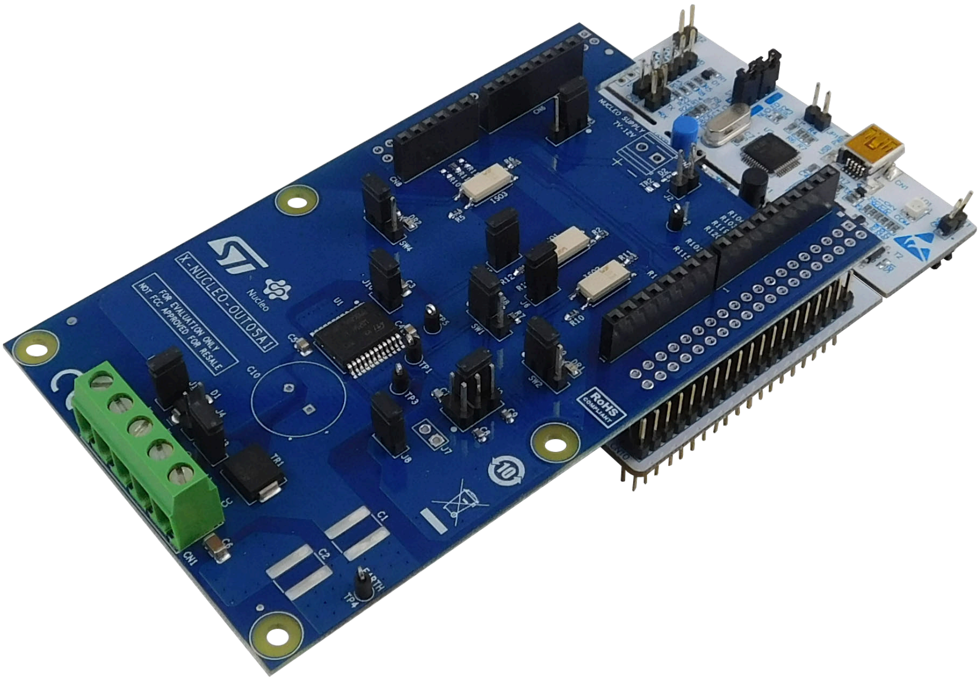
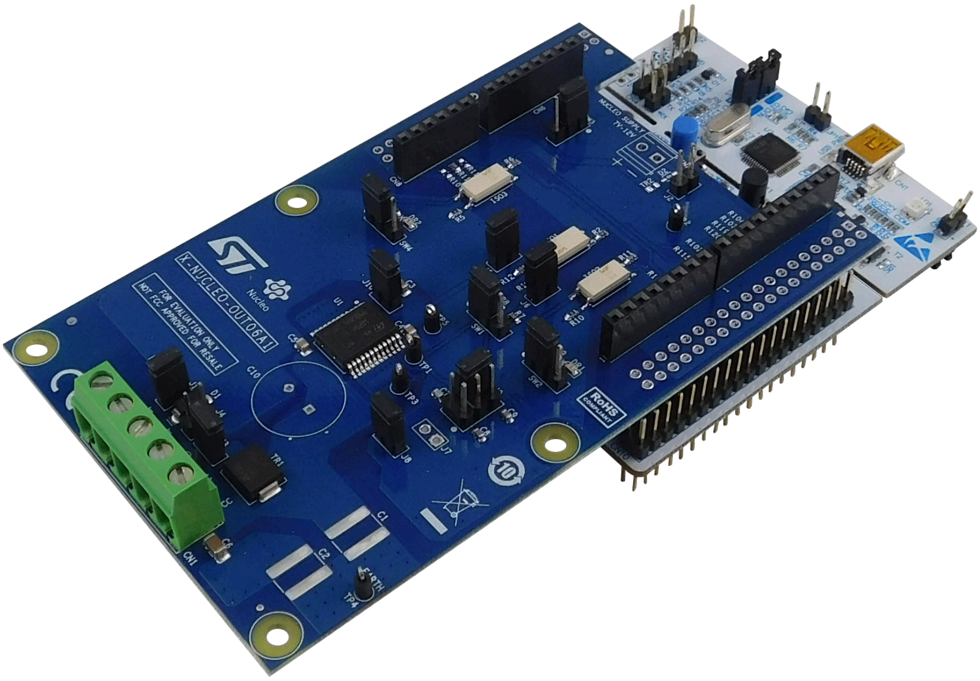


Figure 7. **X-NUCLEO-OUT06A1 expansion board connected to an STM32 Nucleo development board**



Step 2. Power on the STM32 Nucleo development board by connecting a USB cable between the CN1 connector and a PC USB port.

Step 3. Power on the X-NUCLEO-OUT05A1 or X-NUCLEO-OUT06A1 expansion board by properly connecting CN1 connector pin 4 or 5 (V_{CC}) and 3 (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 development board used, open the software project from:

- Projects\STM32F401RE-Nucleo\Examples\Out05_06 for NUCLEO-F401RE
- Projects\STM32G431RB-Nucleo\Examples\Out05_06 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 3. Document revision history

Date	Revision	Changes
24-Mar-2022	1	Initial release.

Contents

List of tables

List of figures