Getting started with the X-CUBE-IOD02 industrial IO-Link device transceiver software expansion for STM32Cube

Introduction

The X-CUBE-IOD02 package is a software expansion for STM32Cube with driver for the L6364 transceiver, a mini-stack library and an IODD configuration file.

The package allows you to develop IO-Link sensor applications based on the L6364 mounted on the X-NUCLEO-IOD02A1 expansion board when connected to a NUCLEO-L073RZ or NUCLEO-G071RB development board.

The software architecture is based on a mini-stack library combined with source code communicating via APIs, and is designed to accommodate custom application development.

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

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
MSP	MCU support package
SPI	Serial peripheral interface

2 X-CUBE-IOD02 software expansion for STM32Cube

2.1 Overview

The X-CUBE-IOD02 software package expands the functionality provided by STM32Cube .

The software allows control of the CQ input/output line of a single channel SIO or IO-Link sensor application when the X-NUCLEO-IOD02A1 expansion board is stacked on a NUCLEO-L073RZ or NUCLEO-G071RB development board.

The software package includes an IO-Link demo-stack library supporting the three transmission modes of the L6364 (Single octet, Multi octet and Transparent UART mode) and the two communication speeds (COM2 and COM3).

The software package key features are:

- Complete software to build applications for the L6364 IO-Link transceiver
- GPIOs, SPI, UART and IRQs configuration
- Smart software architecture based on a mini-stack library combined with source code (communicating through API) and IODD configuration file
- Sample implementation available for X-NUCLEO-IOD02A1 expansion board connected to a NUCLEO-L073RZ or NUCLEO-G071RB development board
- Easy portability across different MCU families, thanks to STM32Cube
- Free, user-friendly license terms

Software and hardware jumper settings can be modified for developing joint mode (CQ = DIO) and dual channel applications.

Sensor applications can be evaluated by stacking sensor expansion boards such as X-NUCLEO-IKS01A2 or X-NUCLEO-IKS02A1.

You can test more complex applications by stacking a sensor expansion board, including the related software package in the X-CUBE-IOD02 and enabling the demo-stack library via the related API.

2.2 Architecture

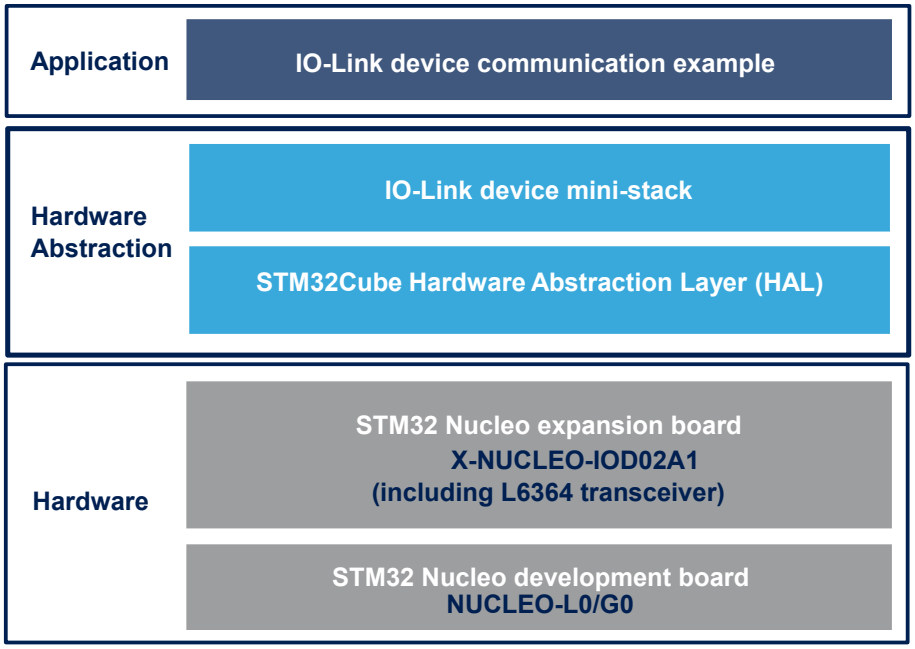
This software is a fully compliant expansion of STM32Cube architecture for the development of SIO and IO-Link Device applications.

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 SIO and IO-Link Device expansion boards based on L6364.

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

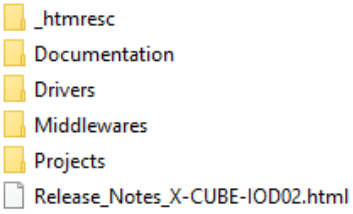
- **STM32Cube HAL layer**: consists of simple, generic and multi-instance APIs (application programming interfaces) which interact with the upper layer applications, libraries and stacks. These generic and extension APIs are based on a common framework so that overlying layers like middleware can function 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-IOD02 expansion software architecture**



2.3 Folder structure

Figure 2. **X-CUBE-IOD02 package folder structure**



The following folders are included in the software package:

- **Documentation** contains compiled HTML files for each supported STM32 Nucleo board, generated from the source code, detailing the software components and APIs.
- **Drivers** contains:
  - STM32G0xx_HAL_Driver and STM32L0xx_HAL_Driver subfolders. These files are not described here as they are not specific to the X-CUBE-IOD02 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-IOD02A1 configuration, the L6364 drivers and the related API.
- **Middlewares** contains the IO-Link demo stack library
- **Projects** contains sample applications for L6364, provided for NUCLEO-L073RZ and NUCLEO-G071RB platforms.

2.3.1 BSPs

For the X-CUBE-IOD02 software, different BSPs are used:

- STM32L0XX-Nucleo/STM32G0XX-Nucleo
- L6364 (in the **Components** subfolder)

- X-NUCLEO-IOD02A1

2.3.1.1 *STM32L0XX-Nucleo/STM32G0XX-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-IOD02A1 expansion board.

Each STM32L0xx-Nucleo/STM32G0xx-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 *L6364*

The L6364 BSP component provides the driver functions for the L6364 transceiver in the Drivers\BSP \Components\L6364 folder, which contains:

- **L6364.c**: core functions of the L6364 drivers
- **L6364.h**: declaration of the L6364 driver functions and their associated definitions
- **l6364_reg.c**: core functions for L6364 read/write register
- **l6364_reg.h**: definitions related to register read/write operations
- **l6364_target_config.h** for communication configuration

2.3.1.3 *X-NUCLEO-IOD02A1*

The X-NUCLEO-IOD02A1 BSP component contains a couple of IOD02_switch.c/h files dedicated to the functions necessary to drive the transceiver using SPI, GPIOs (and UART in Transparent Mode).

The files are also used for the transceiver interrupt.

2.3.2 Projects

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

- Projects\STM32L073RZ-Nucleo\Applications\IOD02A1
- Projects\STM32G071RB-Nucleo\Applications\IOD02A1

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\iod02_conf.h**: header file device specific function prototypes
- **Inc\ stm32l0xx_hal_conf.h** or **stm32g0xx_hal_conf.h**: HAL configuration file
- **Inc\stm32l0xx_it.h** or **stm32g0xx_it.h**: header for the interrupt handler
- **Src\main.c**: main program (code of the example based on the library for L6364)
- **Src\stm32l0xx_hal_msp.c** or **stm32g0xx_hal_msp.c**: code for the MSP initialization and de-initialization
- **Src\stm32l0xx_it.c** or **stm32g0xx_it.c**: interrupt handler
- **Src\system_stm32l0xx.c** or **system_stm32g0xx.c**: system initialization

2.4 Software required resources

The MCU controls the L6364 device via SPI, GPIOs (and UART in case of Transparent Mode).

The X-NUCLEO-IOD02A1 expansion board needs SPI interface (MISO, MOSI, SCLK of SPI#1 configured as SPI slave), a GPIO signal (SS acting as chip selection of the SPI) and an interrupt signal (INT, external I8) when Single octet, Multi octet and Transparent UART mode is selected.

A UART interface (UART TX and UART RX of UART#1) is needed when Transparent mode is selected.

The X-CUBE-IOD02 software also uses a PWM timer (TIM3) to generate the timings for IO-Link specifications.

2.5 APIs

The X-CUBE-IOD02 software API is defined in the \Middlewares\ST\IoLink\inc .h files.

Detailed technical information about the available APIs can be found in a compiled HTML file located in the "Middlewares\ST\IoLink\Documentation" folder where all the functions and parameters are fully described.

2.6 Sample application description

A sample application using the X-NUCLEO-IOD02A1 expansion board with a NUCLEO-L073RZ or NUCLEO-G071RB development board is provided in the “Projects” directory. Ready to be built projects are available for multiple IDEs.

In this sample application, the software runs and sets the system as an IO-Link Device. The system remains in SIO mode until a wake-up request from an IO-Link Master is received.

The demo stack library embeds the Startup, Preoperate and Operate functions according to the IO-Link specifications. Once the wake-up request succeeds, the IO-Link Master port can be set as a digital input and you can control the CQ line High/Low modes by pressing/releasing the blue button on the STM32 Nucleo development board.

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.

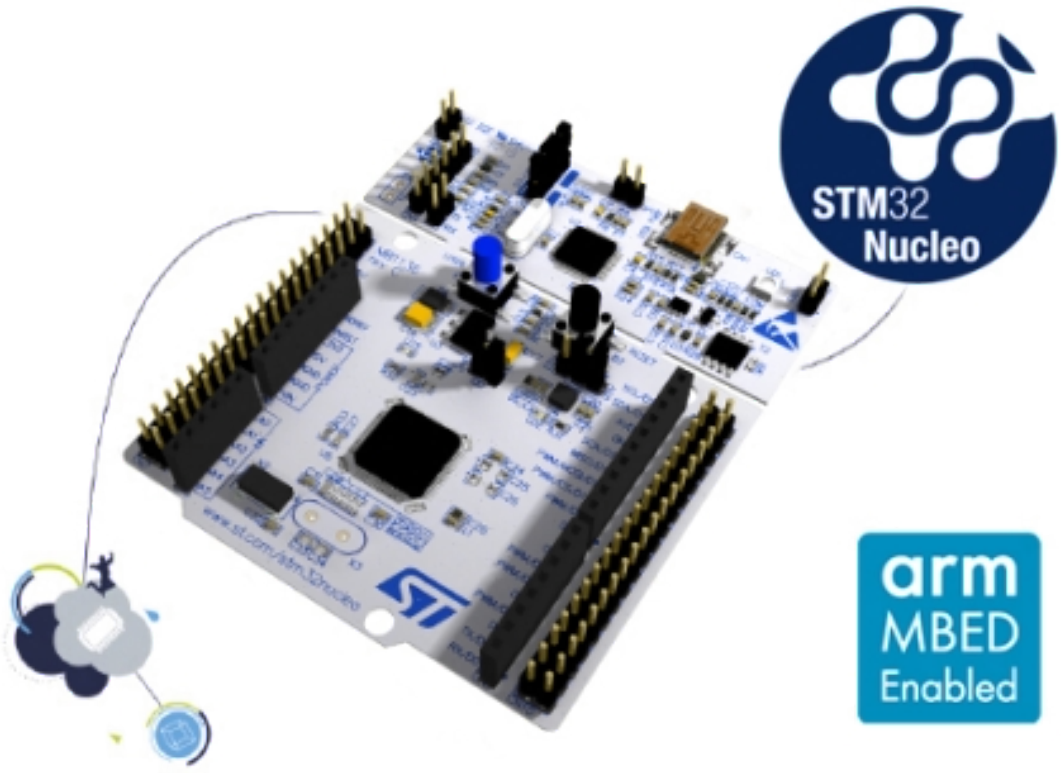
The Arduino™ connectivity support and ST morpho connectors make it easy to expand the functionality of the STM32 Nucleo open development platform with a wide range of specialized expansion boards to choose from.

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



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

3.1.2 X-NUCLEO-IOD02A1 expansion board

The X-NUCLEO-IOD02A1 expansion board for STM32 Nucleo is based on the L6364 dual channel SIO and IO-Link PHY device transceiver embedding 50 mA 3.3 V and 5.0 V voltage regulators, DC-DC converter and M-sequence management.

The expansion board provides an affordable and easy-to-use solution for the development of SIO and IO-Link industrial sensor applications, letting you easily evaluate the L6364 communication features and robustness.

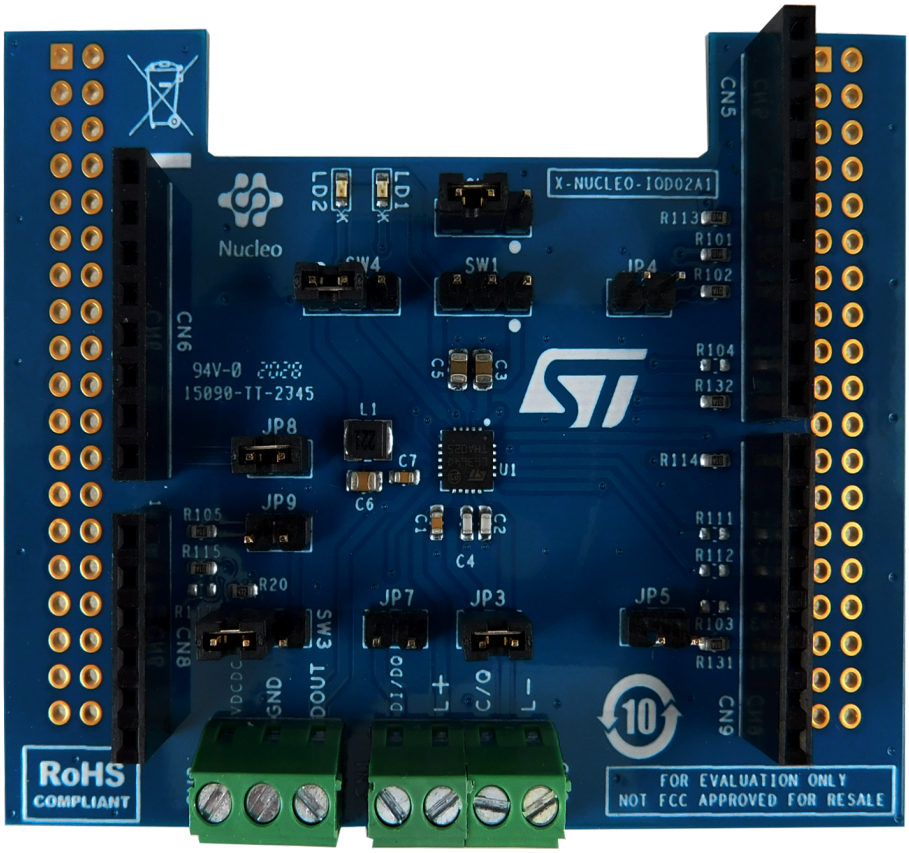
The X-NUCLEO-IOD02A1 communicates with the STM32 controller via SPI and GPIO pins and it is compatible with the Arduino UNO R3 (default configuration) and ST morpho (optional, not mounted) connectors (when connected to a NUCLEO-L073RZ or NUCLEO-G071RB development board).

Communication via IO-Link can be performed in either Multi octet and Single octet UART modes with SPI control of IC configuration and bidirectional sensor data transmission, or in Transparent UART mode with SPI control of IC configuration and UART interfacing for bidirectional sensor data transmission.

The switches on the X-NUCLEO-IOD02A1 conveniently allow you to configure L6364 and expansion board settings such as transmission mode and DC-DC converter enable/disable according to application requirements.

You can also perform evaluation of comprehensive industrial sensor modules by connecting the X-NUCLEO-IOD02A1 to the X-NUCLEO-IKS02A1 sensor shield.

Figure 4. X-NUCLEO-IOD02A1 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-L073RZ.
2. One USB type A to Micro-B USB cable when using a NUCLEO-G071RB.

Note: USB cables are strictly necessary only for the download of the example firmware: the system can work without USB connection as it is supplied by the L+ pin (for further details, refer to UM2741, freely available on www.st.com).

3. An IO-Link Master (e.g. P-NUCLEO-IOM01M1) with the associated control software.
4. A set of wires for the connection of L+, L- and CQ lines between the IO-Link Master and the X-NUCLEO-IOD02A1.

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-IOD02A1 expansion board:

- X-CUBE-IOD02: an expansion for STM32Cube dedicated to application development which require the use of L6364. The X-CUBE-IOD02 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 development board has to be configured with the following jumper position:

- JP1 off
- JP5 (PWR) on UV5 side for NUCLEO-L073RZ; on 5V_STLK for NUCLEO-G071RB
- JP6 (IDD) on

The X-NUCLEO-IOD02A1 expansion board has to be configured as follows:

- JP3, JP7, JP9 open
- JP4, JP5, open (Single octet and Multi octet UART modes); J4, JP5 closed (Transparent UART mode)
- JP8 closed
- SW1: open
- SW2, SW3, SW4: close pins 1-2

Step 1. Check the configuration of JP5 (on NUCLEO-L073RZ) and set it to “U5V”, or check the configuration of JP2 (on NUCLEO-G071RB) and close between pins 1-2.

Step 2. Connect the mini-USB (for NUCLEO-L073RZ) or micro-USB (for NUCLEO-G071RB) to your PC and the STM32 Nucleo development board.

Step 3. Download the selected firmware onto the microcontroller.

You can use the tools available in your IDE, STM32 LINK Utility or STM32CubeProgrammer and you can select among different firmware packages according to the MCU (STM32L0x or STM32G0x). In Single octet and Multi octet UART modes, JP4 and JP5 remain open. In Transparent UART mode, JP4 and JP5 must be closed.

Step 4. Disconnect the USB cable from the STM32 Nucleo development board and close JP5 to “E5V” (on NUCLEO-L073RZ), or set JP2 from 1-2 to 3-4 (on NUCLEO-G071RB).

In this setup, the STM32 Nucleo development board is supplied by the X-NUCLEO-IOD02A1 and the step-down converter is active (for SW4 close pins 1-2) to supply Vin (JP8 closed). VDIG must be referred to 3.3 V rail supplied by the STM32 Nucleo development board (SW1 open, for SW2 close pins 1-2).

Step 5. Connect the X-NUCLEO-IOD02A1 to the STM32 Nucleo development board through the Arduino connectors.

Step 6. Connect the X-NUCLEO-IOD02A1 power section (CN1, CN2) to the IO-Link master according to the schematic and serigraphy.

Step 7. Open the control tool of your IO-Link master and upload the IODD XML file (included in the X-CUBE-IOD02 software package).

You can select between the two IODD files according to the communication speed (COM2 or COM3) of the firmware downloaded on the STM32 Nucleo development board.

Step 8. Activate your IO-Link Master (usually requires a connection to a 24 V supply rail).

The IO-Link master control tool lets you supply the X-NUCLEO-IOD02A1 L+ line and launch a wake-up request to initiate communication.

Step 9. Set to the IO-Link port of the Master as digital input and then press the STM32 Nucleo development board blue button to drive the CQ line status to 24 V/0 V.

Revision history

Table 2. **Document revision history**

Date	Version	Changes
01-Sep-2020	1	Initial release.
09-Oct-2020	2	Updated Introduction, Section 2.1 Overview, Section 2.2 Architecture, Section 2.3 Folder structure and Section 3.4 Board setup.

Contents

List of tables

List of figures