
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 and L6362A transceivers, mini-stack libraries and IODD configuration files.

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 or NUCLEO-L452RE or NUCLEO-F303RE development board.

The package can also be used to develop IO-Link sensor applications based on the L6362A mounted on the STEVAL-IOD003V1 expansion board when connected to a NUCLEO-L073RZ or NUCLEO-L452RE development board.

The software architecture is based on mini-stack libraries 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](http://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 the 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 or NUCLEO-L452RE or NUCLEO-F303RE development board.

The software also allows the control of the CQ input/output line of a single channel SIO or IO-Link sensor application when the STEVAL-IOD003V1 expansion board is stacked on a NUCLEO-L073RZ or NUCLEO-L452RE development board.

The software package includes IO-Link demo-stack libraries supporting the three transmission modes of the L6364 (single octet, multioctet, and transparent UART mode) and the two communication speeds (COM2 and COM3).

For the L6362A three communication speeds (COM1, COM2 and COM3) are supported.

The software package key features are:

- Complete software to build applications for the L6364 and L6362A IO-Link transceiver
- GPIOs, SPI, UART and IRQs configuration
- Smart software architecture based on mini-stack libraries 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 or NUCLEO-L452RE or NUCLEO-F303RE development board
- Sample implementation available for STEVAL-IOD003V1 expansion board connected to a NUCLEO-L073RZ or NUCLEO-L452RE development board
- Easy portability across different MCU families, thanks to STM32Cube
- Free, user-friendly license terms

Software and hardware jumper settings for X-NUCLEO-IOD02A1 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

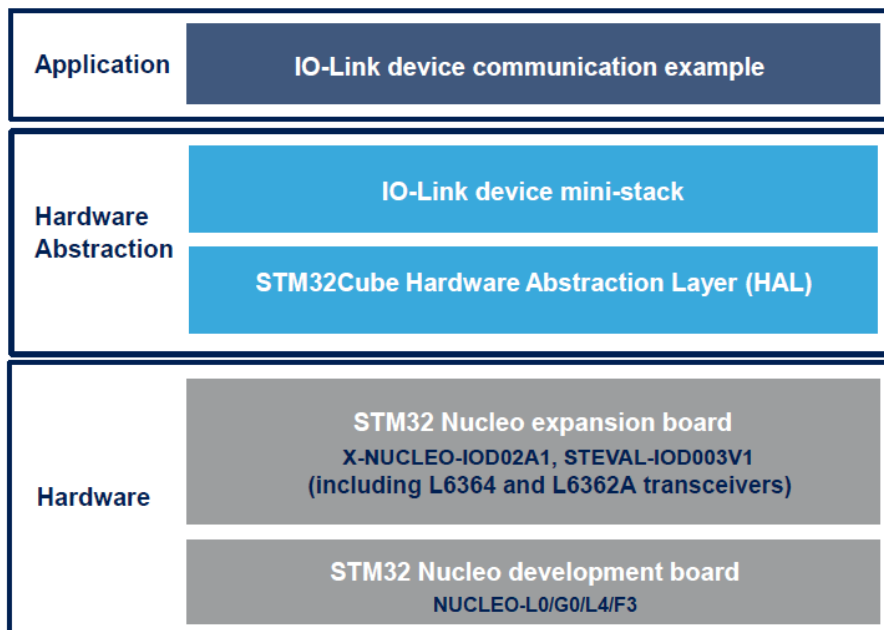
This software is a fully compliant expansion of STM32Cube architecture for the development of SIO and IO-Link Device applications.

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 and L6362A.

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

-  _htmresc
-  Documentation
-  Drivers
-  Middlewares
-  Projects
-  Additional_License_Terms_forX-CUBE-IOD02.html
-  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 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, STM32L0xx_HAL_Driver, STM32L4xx_HAL_Driver and STM32F3xx_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](#) and [STEVAL-IOD003V1](#) configuration, the [L6364](#) and [L6362A](#) drivers and the related API.
- **Middlewares** contains the IO-Link demo stack libraries
 - Middlewares\ST\IoLinkLib\IOD02A1 contains the libraries for [L6364](#)
 - Middlewares\ST\IoLinkLib\IOD003V1 contains the libraries for [L6362A](#)
- **Projects** contains sample applications for:
 - [L6364](#), provided for [NUCLEO-L073RZ](#) and [NUCLEO-G071RB](#) and [NUCLEO-L452RE](#) and [NUCLEO-F303RE](#) platforms
 - [L6362A](#), provided for [NUCLEO-L073RZ](#) and [NUCLEO-L452RE](#) platforms.

2.3.1 BSPs

For the [X-CUBE-IOD02](#) software, different BSPs are used:

- STM32L0xx_Nucleo, STM32G0xx_Nucleo, STM32L4xx_Nucleo, STM32F3xx_Nucleo
- [L6364](#) (in the **Components** subfolder)
- [L6362A](#) (in the **Components** subfolder)
- [IOD02A1](#)
- [IOD003V1](#)

2.3.1.1 [STM32L0xx_Nucleo](#), [STM32G0xx_Nucleo](#), [STM32L4xx_Nucleo](#), [STM32F3xx_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](#) or [STEVAL-IOD003V1](#) expansion board.

Each folder ([STM32L0xx_Nucleo](#), [STM32G0xx_Nucleo](#), [STM32L4xx_Nucleo](#), [STM32F3xx_Nucleo](#)) contains couples of .c/.h files (stm32[code]xx_nucleo.c/.h, where [code] is the MCU family code L0, G0, L4 or F3) 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

2.3.1.3 [L6362A](#)

The [L6362A](#) BSP component provides the driver functions for the [L6362A](#) transceiver in the Drivers\BSP\Components\L6362a folder, which contains:

- **L6362a.c**: core functions of the [L6362A](#) drivers
- **L6362a.h**: declaration of the [L6362A](#) driver functions and their associated definitions

2.3.1.4

IOD02A1

The IOD02A1 BSP component contains:

- **iod02a1.c**: core functions of the IOD02A1 driver
- **iod02a1.h**: header file for the IOD02A1 driver
- **iod02a1_bus.c**: core functions for IOD02A1 bus read/write operations (SPI, GPIO, UART)
- **iod02a1_bus.h**: header file for bus read/write functions

2.3.1.5

IOD003V1

The IOD003V1 BSP component contains:

- **iod003v1.c**: core functions of the IOD003V1 driver
- **iod003v1.h**: header file for the IOD003V1 driver
- **iod003v1_bus.c**: core functions for IOD003V1 bus read/write operations (GPIO, UART)
- **iod003v1_bus.h**: header file for bus read/write functions

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
- Projects\STM32L452RE-Nucleo\Applications\IOD02A1
- Projects\STM32F303RE-Nucleo\Applications\IOD02A1
- Projects\STM32L073RZ-Nucleo\Applications\IOD003V1
- Projects\STM32L452RE-Nucleo\Applications\IOD003V1

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 code files listed in the tables below.

Table 2. Projects\STM32G071RB-Nucleo\Applications\IOD02A1

File	Description
Projects\STM32G071RB-Nucleo\Applications\IOD02A1\Inc\app_iolink.h	Header for main application program body
Projects\STM32G071RB-Nucleo\Applications\IOD02A1\Inc\demoapp.h	Header for demonstration application
Projects\STM32G071RB-Nucleo\Applications\IOD02A1\Inc\iod02a1_conf.h	Header for the components bus interface
Projects\STM32G071RB-Nucleo\Applications\IOD02A1\Inc\iolink_int_interface.h	Header for IO-Link interrupt interface module
Projects\STM32G071RB-Nucleo\Applications\IOD02A1\Inc\iolink_interface.h	Header for iolink_interface.c module
Projects\STM32G071RB-Nucleo\Applications\IOD02A1\Inc\l6364_conf.h	Header for L6364 parameters
Projects\STM32G071RB-Nucleo\Applications\IOD02A1\Inc\main.h	Header for main.c module
Projects\STM32G071RB-Nucleo\Applications\IOD02A1\Inc\stm32g0xx_hal_conf.h	HAL configuration file for STM32G0xx
Projects\STM32G071RB-Nucleo\Applications\IOD02A1\Inc\stm32g0xx_it.h	Interrupt handlers header file for STM32G0xx
Projects\STM32G071RB-Nucleo\Applications\IOD02A1\Inc\stm32g0xx_nucleo_bus.h	Header for platform definitions
Projects\STM32G071RB-Nucleo\Applications\IOD02A1\Inc\stm32g0xx_nucleo_conf.h	Configuration file
Projects\STM32G071RB-Nucleo\Applications\IOD02A1\Inc\stm32g0xx_nucleo_erno.h	Nucleo board management error codes
Projects\STM32G071RB-Nucleo\Applications\IOD02A1\Src\app_iolink.c	Main application program body
Projects\STM32G071RB-Nucleo\Applications\IOD02A1\Src\demoapp.c	Demonstration application
Projects\STM32G071RB-Nucleo\Applications\IOD02A1\Src\iolink_int_interface.c	IO-Link interrupt interface file
Projects\STM32G071RB-Nucleo\Applications\IOD02A1\Src\iolink_interface.c	IO-Link interface file with hardware dependencies

File	Description
Projects/STM32G071RB-Nucleo/Applications/IOD02A1/Src/main.c	Main program
Projects/STM32G071RB-Nucleo/Applications/IOD02A1/Src/stm32g0xx_hal_msp.c	HAL MSP module for STM32G0xx
Projects/STM32G071RB-Nucleo/Applications/IOD02A1/Src/stm32g0xx_it.c	Interrupt handlers for STM32G0xx
Projects/STM32G071RB-Nucleo/Applications/IOD02A1/Src/stm32g0xx_nucleo_bus.c	Bus operations
Projects/STM32G071RB-Nucleo/Applications/IOD02A1/Src/system_stm32g0xx.c	System source file for STM32G0xx
Projects/STM32G071RB-Nucleo/Applications/IOD02A1/IODD/STMicroelectronics-X-NUCLEO-IOD02A1-38kBd-20240423-IODD1.1.xml	IODD file for X-NUCLEO-IOD02A1 (IO-Link COM2 mode)
Projects/STM32G071RB-Nucleo/Applications/IOD02A1/IODD/STMicroelectronics-X-NUCLEO-IOD02A1-230kBd-20240423-IODD1.1.xml	IODD file for X-NUCLEO-IOD02A1 (IO-Link COM3 mode)

Table 3. Projects\STM32L073RZ-Nucleo\Applications\IOD02A1

File	Description
Projects/STM32L073RZ-Nucleo/Applications/IOD02A1/Inc/app_iolink.h	Header for main application program body
Projects/STM32L073RZ-Nucleo/Applications/IOD02A1/Inc/demoapp.h	Header for demonstration application
Projects/STM32L073RZ-Nucleo/Applications/IOD02A1/Inc/iod02a1_conf.h	Header for the components bus interface
Projects/STM32L073RZ-Nucleo/Applications/IOD02A1/Inc/iolink_int_interface.h	Header for IO-Link interrupt interface module
Projects/STM32L073RZ-Nucleo/Applications/IOD02A1/Inc/iolink_interface.h	Header for iolink_interface.c module
Projects/STM32L073RZ-Nucleo/Applications/IOD02A1/Inc/l6364_conf.h	Header for L6364 parameters
Projects/STM32L073RZ-Nucleo/Applications/IOD02A1/Inc/main.h	Header for main.c module
Projects/STM32L073RZ-Nucleo/Applications/IOD02A1/Inc/stm32l0xx_hal_conf.h	HAL configuration file for STM32L0xx
Projects/STM32L073RZ-Nucleo/Applications/IOD02A1/Inc/stm32l0xx_it.h	Header for platform definitions
Projects/STM32L073RZ-Nucleo/Applications/IOD02A1/Inc/stm32l0xx_nucleo_bus.h	Header for platform definitions
Projects/STM32L073RZ-Nucleo/Applications/IOD02A1/Inc/stm32l0xx_nucleo_conf.h	Configuration file
Projects/STM32L073RZ-Nucleo/Applications/IOD02A1/Inc/stm32l0xx_nucleo_errno.h	Nucleo board management error codes
Projects/STM32L073RZ-Nucleo/Applications/IOD02A1/Src/app_iolink.c	Main application program body
Projects/STM32L073RZ-Nucleo/Applications/IOD02A1/Src/demoapp.c	Demonstration application
Projects/STM32L073RZ-Nucleo/Applications/IOD02A1/Src/iolink_int_interface.c	IO-Link interrupt interface file
Projects/STM32L073RZ-Nucleo/Applications/IOD02A1/Src/iolink_interface.c	IO-Link interface file with hardware dependencies
Projects/STM32L073RZ-Nucleo/Applications/IOD02A1/Src/main.c	Main program
Projects/STM32L073RZ-Nucleo/Applications/IOD02A1/Src/stm32l0xx_hal_msp.c	HAL MSP module for STM32L0xx
Projects/STM32L073RZ-Nucleo/Applications/IOD02A1/Src/stm32l0xx_it.c	Interrupt handlers for STM32L0xx
Projects/STM32L073RZ-Nucleo/Applications/IOD02A1/Src/stm32l0xx_nucleo_bus.c	Bus operations
Projects/STM32L073RZ-Nucleo/Applications/IOD02A1/Src/system_stm32l0xx.c	System source file for STM32L0xx
Projects/STM32L073RZ-Nucleo/Applications/IOD02A1/IODD/STMicroelectronics-X-NUCLEO-IOD02A1-38kBd-20240423-IODD1.1.xml	IODD file for X-NUCLEO-IOD02A1 (IO-Link COM2 mode)
Projects/STM32L073RZ-Nucleo/Applications/IOD02A1/IODD/STMicroelectronics-X-NUCLEO-IOD02A1-230kBd-20240423-IODD1.1.xml	IODD file for X-NUCLEO-IOD02A1 (IO-Link COM3 mode)

Table 4. Projects\STM32L452RE-Nucleo\Applications\IOD02A1

File	Description
Projects/STM32L452RE-Nucleo/Applications/IOD02A1/Inc/app_iolink.h	Header for main application program body
Projects/STM32L452RE-Nucleo/Applications/IOD02A1/Inc/demoapp.h	Header for demonstration application

File	Description
Projects/STM32L452RE-Nucleo/Applications/IOD02A1/Inc/iod02a1_conf.h	Header for the components bus interface
Projects/STM32L452RE-Nucleo/Applications/IOD02A1/Inc/iolink_int_interface.h	Header for IO-Link interrupt interface module
Projects/STM32L452RE-Nucleo/Applications/IOD02A1/Inc/iolink_interface.h	Header for iolink_interface.c module
Projects/STM32L452RE-Nucleo/Applications/IOD02A1/Inc/l6364_conf.h	Header for L6364 parameters
Projects/STM32L452RE-Nucleo/Applications/IOD02A1/Inc/main.h	Header for main.c module
Projects/STM32L452RE-Nucleo/Applications/IOD02A1/Inc/stm32l4xx_hal_conf.h	HAL configuration file for STM32L4xx
Projects/STM32L452RE-Nucleo/Applications/IOD02A1/Inc/stm32l4xx_it.h	Interrupt handlers header file for STM32L4xx
Projects/STM32L452RE-Nucleo/Applications/IOD02A1/Inc/stm32l4xx_nucleo_bus.h	Header for platform definitions
Projects/STM32L452RE-Nucleo/Applications/IOD02A1/Inc/stm32l4xx_nucleo_conf.h	Configuration file
Projects/STM32L452RE-Nucleo/Applications/IOD02A1/Inc/stm32l4xx_nucleo_errno.h	Nucleo board management error codes
Projects/STM32L452RE-Nucleo/Applications/IOD02A1/Src/app_iolink.c	Main application program body
Projects/STM32L452RE-Nucleo/Applications/IOD02A1/Src/demoapp.c	Demonstration application
Projects/STM32L452RE-Nucleo/Applications/IOD02A1/Src/iolink_int_interface.c	IO-Link interrupt interface file
Projects/STM32L452RE-Nucleo/Applications/IOD02A1/Src/iolink_interface.c	IO-Link interface file with hardware dependencies
Projects/STM32L452RE-Nucleo/Applications/IOD02A1/Src/main.c	Main program
Projects/STM32L452RE-Nucleo/Applications/IOD02A1/Src/stm32l4xx_hal_msp.c	HAL MSP module for STM32L4xx
Projects/STM32L452RE-Nucleo/Applications/IOD02A1/Src/stm32l4xx_it.c	Interrupt handlers for STM32L4xx
Projects/STM32L452RE-Nucleo/Applications/IOD02A1/Src/stm32l4xx_nucleo_bus.c	Bus operations
Projects/STM32L452RE-Nucleo/Applications/IOD02A1/Src/system_stm32l4xx.c	System source file for STM32L4xx
Projects/STM32L452RE-Nucleo/Applications/IOD02A1/IODD/STMicroelectronics-X-NUCLEO-IOD02A1-38kBd-20240423-IODD1.1.xml	IODD file for X-NUCLEO-IOD02A1 (IO-Link COM2 mode)
Projects/STM32L452RE-Nucleo/Applications/IOD02A1/IODD/STMicroelectronics-X-NUCLEO-IOD02A1-230kBd-20240423-IODD1.1.xml	IODD file for X-NUCLEO-IOD02A1 (IO-Link COM3 mode)

Table 5. Projects\STM32F303RE-Nucleo\Applications\IOD02A1

File	Description
Projects/STM32F303RE-Nucleo/Applications/IOD02A1/Inc/app_iolink.h	Header for main application program body
Projects/STM32F303RE-Nucleo/Applications/IOD02A1/Inc/demoapp.h	Header for demonstration application

File	Description
Projects/STM32F303RE-Nucleo/Applications/IOD02A1/Inc/iod02a1_conf.h	Header for the components bus interface
Projects/STM32F303RE-Nucleo/Applications/IOD02A1/Inc/iolink_int_interface.h	Header for IO-Link interrupt interface module
Projects/STM32F303RE-Nucleo/Applications/IOD02A1/Inc/iolink_interface.h	Header for iolink_interface.c module
Projects/STM32F303RE-Nucleo/Applications/IOD02A1/Inc/l6364_conf.h	Header for L6364 parameters
Projects/STM32F303RE-Nucleo/Applications/IOD02A1/Inc/main.h	Header for main.c module
Projects/STM32F303RE-Nucleo/Applications/IOD02A1/Inc/stm32f3xx_hal_conf.h	HAL configuration file for STM32F3xx
Projects/STM32LF303RE-Nucleo/Applications/IOD02A1/Inc/stm32f3xx_it.h	Interrupt handlers header file for STM32F3xx
Projects/STM32F303RE-Nucleo/Applications/IOD02A1/Inc/stm32f3xx_nucleo_bus.h	Header for platform definitions
Projects/STM32F303RE-Nucleo/Applications/IOD02A1/Inc/stm32f3xx_nucleo_conf.h	Configuration file
Projects/STM32F303RE-Nucleo/Applications/IOD02A1/Inc/stm32f3xx_nucleo_errno.h	Nucleo board management error codes
Projects/STM32F303RE-Nucleo/Applications/IOD02A1/Src/app_iolink.c	Main application program body
Projects/STM32F303RE-Nucleo/Applications/IOD02A1/Src/demoapp.c	Demonstration application
Projects/STM32F303RE-Nucleo/Applications/IOD02A1/Src/iolink_int_interface.c	IO-Link interrupt interface file
Projects/STM32F303RE-Nucleo/Applications/IOD02A1/Src/iolink_interface.c	IO-Link interface file with hardware dependencies
Projects/STM32F303RE-Nucleo/Applications/IOD02A1/Src/main.c	Main program
Projects/STM32F303RE-Nucleo/Applications/IOD02A1/Src/stm32f3xx_hal_msp.c	HAL MSP module for STM32F3xx
Projects/STM32F303RE-Nucleo/Applications/IOD02A1/Src/stm32f3xx_it.c	Interrupt handlers for STM32F3xx
Projects/STM32F303RE-Nucleo/Applications/IOD02A1/Src/stm32f3xx_nucleo_bus.c	Bus operations
Projects/STM32F303RE-Nucleo/Applications/IOD02A1/Src/system_stm32f3xx.c	System source file for STM32F3xx
Projects/STM32F303RE-Nucleo/Applications/IOD02A1/IODD/STMicroelectronics-X-NUCLEO-IOD02A1-38kBd-20240423-IODD1.1.xml	IODD file for X-NUCLEO-IOD02A1 (IO-Link COM2 mode)
Projects/STM32F303RE-Nucleo/Applications/IOD02A1/IODD/STMicroelectronics-X-NUCLEO-IOD02A1-230kBd-20240423-IODD1.1.xml	IODD file for X-NUCLEO-IOD02A1 (IO-Link COM3 mode)

Table 6. Projects\STM32L073RZ-Nucleo\Applications\IOD003V1

File	Description
Projects/STM32L073RZ-Nucleo/Applications/IOD003V1/Inc/app_iolink.h	Header for main application program body
Projects/STM32L073RZ-Nucleo/Applications/IOD003V1/Inc/demoapp.h	Header for demonstration application

File	Description
Projects/STM32L073RZ-Nucleo/Applications/IOD003V1/Inc/iod003v1_conf.h	Header for the components bus interface
Projects/STM32L073RZ-Nucleo/Applications/IOD003V1/Inc/iolink_int_interface.h	Header for IO-Link interrupt interface module
Projects/STM32L073RZ-Nucleo/Applications/IOD003V1/Inc/iolink_interface.h	Header for iolink_interface.c module
Projects/STM32L073RZ-Nucleo/Applications/IOD003V1/Inc/l6362a_conf.h	Header for L6362A parameters
Projects/STM32L073RZ-Nucleo/Applications/IOD003V1/Inc/main.h	Header for main.c module
Projects/STM32L073RZ-Nucleo/Applications/IOD003V1/Inc/stm32l0xx_hal_conf.h	HAL configuration file for STM32L0xx
Projects/STM32L073RZ-Nucleo/Applications/IOD003V1/Inc/stm32l0xx_it.h	Interrupt handlers header file for STM32L0xx
Projects/STM32L073RZ-Nucleo/Applications/IOD003V1/Inc/stm32l0xx_nucleo_bus.h	Header for platform definitions
Projects/STM32L073RZ-Nucleo/Applications/IOD003V1/Inc/stm32l0xx_nucleo_conf.h	Configuration file
Projects/STM32L073RZ-Nucleo/Applications/IOD003V1/Inc/stm32l0xx_nucleo_errno.h	Nucleo board management error codes
Projects/STM32L073RZ-Nucleo/Applications/IOD003V1/Src/app_iolink.c	Main application program body
Projects/STM32L073RZ-Nucleo/Applications/IOD003V1/Src/demoapp.c	Demonstration application
Projects/STM32L073RZ-Nucleo/Applications/IOD003V1/Src/iolink_int_interface.c	IO-Link interrupt interface file
Projects/STM32L073RZ-Nucleo/Applications/IOD003V1/Src/iolink_interface.c	IO-Link interface file with hardware dependencies
Projects/STM32L073RZ-Nucleo/Applications/IOD003V1/Src/main.c	Main program
Projects/STM32L073RZ-Nucleo/Applications/IOD003V1/Src/stm32l0xx_hal_msp.c	HAL MSP module for STM32L0xx
Projects/STM32L073RZ-Nucleo/Applications/IOD003V1/Src/stm32l0xx_it.c	Interrupt handlers for STM32L0xx
Projects/STM32L073RZ-Nucleo/Applications/IOD003V1/Src/stm32l0xx_nucleo_bus.c	Bus operations
Projects/STM32L073RZ-Nucleo/Applications/IOD003V1/Src/system_stm32l0xx.c	System source file for STM32L0xx
Projects/STM32L073RZ-Nucleo/Applications/IOD003V1/IODD/STMicroelectronics-STEVAL-IOD003V1-4kBd-20240423-IODD1.1.xml	IODD file for STEVAL-IOD003V1 (IO-Link COM1 mode)
Projects/STM32L073RZ-Nucleo/Applications/IOD003V1/IODD/STMicroelectronics-STEVAL-IOD003V1-38kBd-20240423-IODD1.1.xml	IODD file for STEVAL-IOD003V1 (IO-Link COM2 mode)
Projects/STM32L073RZ-Nucleo/Applications/IOD003V1/IODD/STMicroelectronics-STEVAL-IOD003V1-230kBd-20240423-IODD1.1.xml	IODD file for STEVAL-IOD003V1 (IO-Link COM3 mode)

Table 7. Projects\STM32L452RE-Nucleo\Applications\IOD003V1

File	Description
Projects/STM32L452RE-Nucleo/Applications/IOD003V1/Inc/app_iolink.h	Header for main application program body
Projects/STM32L452RE-Nucleo/Applications/IOD003V1/Inc/demoapp.h	Header for demonstration application
Projects/STM32L452RE-Nucleo/Applications/IOD003V1/Inc/iod003v1_conf.h	Header for the components bus interface
Projects/STM32L452RE-Nucleo/Applications/IOD003V1/Inc/iolink_int_interface.h	Header for IO-Link interrupt interface module

File	Description
Projects/STM32L452RE-Nucleo/Applications/IOD003V1/Inc/iolink_interface.h	Header for iolink_interface.c module
Projects/STM32L452RE-Nucleo/Applications/IOD003V1/Inc/l6362a_conf.h	Header for L6362A parameters
Projects/STM32L452RE-Nucleo/Applications/IOD003V1/Inc/main.h	Header for main.c module
Projects/STM32L452RE-Nucleo/Applications/IOD003V1/Inc/stm32l4xx_hal_conf.h	HAL configuration file for STM32L4xx
Projects/STM32L452RE-Nucleo/Applications/IOD003V1/Inc/stm32l4xx_it.h	Interrupt handlers header file for STM32L4xx
Projects/STM32L452RE-Nucleo/Applications/IOD003V1/Inc/stm32l4xx_nucleo_bus.h	Header for platform definitions
Projects/STM32L452RE-Nucleo/Applications/IOD003V1/Inc/stm32l4xx_nucleo_conf.h	Configuration file
Projects/STM32L452RE-Nucleo/Applications/IOD003V1/Inc/stm32l4xx_nucleo_errno.h	Nucleo board management error codes
Projects/STM32L452RE-Nucleo/Applications/IOD003V1/Src/app_iolink.c	Main application program body
Projects/STM32L452RE-Nucleo/Applications/IOD003V1/Src/demoapp.c	Demonstration application
Projects/STM32L452RE-Nucleo/Applications/IOD003V1/Src/iolink_int_interface.c	IO-Link interrupt interface file
Projects/STM32L452RE-Nucleo/Applications/IOD003V1/Src/iolink_interface.c	IO-Link interface file with hardware dependencies
Projects/STM32L452RE-Nucleo/Applications/IOD003V1/Src/main.c	Main program
Projects/STM32L452RE-Nucleo/Applications/IOD003V1/Src/stm32l4xx_hal_msp.c	HAL MSP module for STM32L4xx
Projects/STM32L452RE-Nucleo/Applications/IOD003V1/Src/stm32l4xx_it.c	Interrupt handlers for STM32L4xx
Projects/STM32L452RE-Nucleo/Applications/IOD003V1/Src/stm32l4xx_nucleo_bus.c	Bus operations
Projects/STM32L452RE-Nucleo/Applications/IOD003V1/Src/system_stm32l4xx.c	System source file for STM32L4xx
Projects/STM32L452RE-Nucleo/Applications/IOD003V1/IODD/STMicroelectronics-STEVAL-IOD003V1-4kBd-20240423-IODD1.1.xml	IODD file for STEVAL-IOD003V1 (IO-Link COM1 mode)
Projects/STM32L452RE-Nucleo/Applications/IOD003V1/IODD/STMicroelectronics-STEVAL-IOD003V1-38kBd-20240423-IODD1.1.xml	IODD file for STEVAL-IOD003V1 (IO-Link COM2 mode)
Projects/STM32L452RE-Nucleo/Applications/IOD003V1/IODD/STMicroelectronics-STEVAL-IOD003V1-230kBd-20240423-IODD1.1.xml	IODD file for STEVAL-IOD003V1 (IO-Link COM3 mode)

2.4

Software required resources

The MCU controls the [L6364](#) device via SPI, GPIOs (and UART in case of Transparent Mode) and the [L6362A](#) via UART.

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 interrupt line 8) when Single octet, Multi octet 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 as per IO-Link specifications.

The STEVAL-IOD003V1 expansion board needs UART interface a GPIO signal (EN) and an interrupt signal (external interrupt line 6).

2.5 APIs

The X-CUBE-IOD02 software APIs are defined in the folders:

- Middlewares\ST\IoLinkLib\IOD02A1\inc .h files (X-NUCLEO-IOD02A1 expansion board)
- Middlewares\ST\IoLinkLib\IOD003V1\inc .h files (STEVAL-IOD003V1 expansion board)

Detailed technical information about the available APIs can be found in a compiled HTML file located in the:

- Middlewares\ST\IoLinkLib\IOD02A1\Documentation" folder for X-NUCLEO-IOD02A1 and
- Middlewares\ST\IoLinkLib\IOD003V1\Documentation" folder for STEVAL-IOD003V1

where all the functions and parameters are fully described.

2.6 Sample application description

Sample applications using the X-NUCLEO-IOD02A1 expansion board with a NUCLEO-L073RZ or NUCLEO-G071RB or NUCLEO-L452RE or NUCLEO-F303RE development board or a sample application using the STEVAL-IOD003V1 expansion board with a NUCLEO-L073RZ or NUCLEO-L452RE development board are provided in the "Projects" directory. Ready to be built projects are available for multiple IDEs.

In these sample applications, 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.

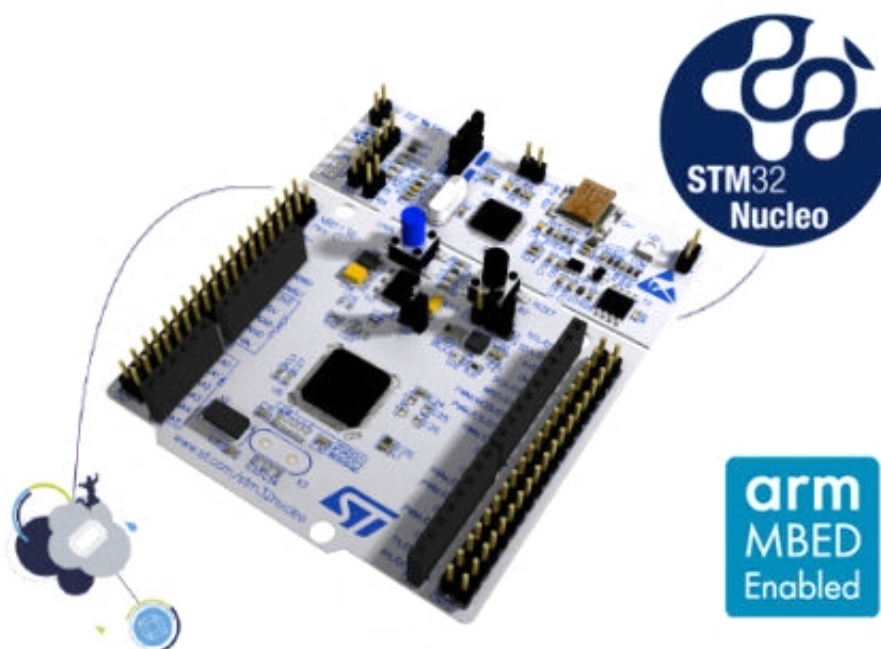
The Arduino connectivity support and ST morpho connectors make it easy to expand the functionality of the STM32 Nucleo open development platform with a wide range of specialized expansion boards to choose from.

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



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.

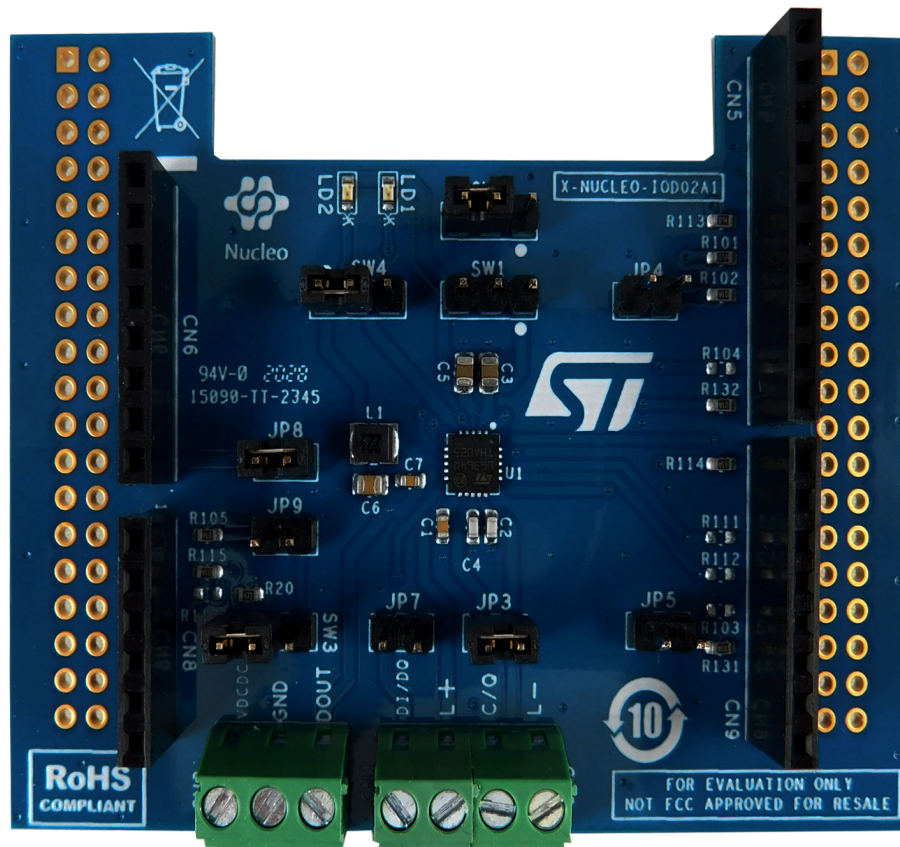
The expansion board provides an affordable and easy-to-use solution for the development of SIO and IO-Link industrial sensor applications, letting you easily evaluate the L6364 communication features and robustness.

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 or NUCLEO-L452RE or NUCLEO-F303RE 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 or a NUCLEO-L452RE or a NUCLEO-F303RE.
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 allow setting 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:

- **NUCLEO-L073RZ, NUCLEO-L452RE, NUCLEO-F303RE**
 - JP5 on U5V for firmware flashing
 - JP1 open
 - JP6 closed
 - CN2: closed 1-2, 3-4
 - CN3 open
 - CN4 1-6 open
 - CN11 closed
 - CN12 closed
- **NUCLEO-G071RB**
 - JP2: closed 1-2 (5V_STLK for firmware flashing)
 - JP1, JP6 open
 - JP3, JP4, JP5 closed
 - CN4 1-2 3-4 closed
 - CN11 1-6 open

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** or **NUCLEO-L452RE** or **NUCLEO-F303RE**) 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 **NUCLEO-L452RE** or **NUCLEO-F303RE**) 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 or STM32L4x or STM32F3x). 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 **NUCLEO-L452RE** or **NUCLEO-F303RE**), 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.

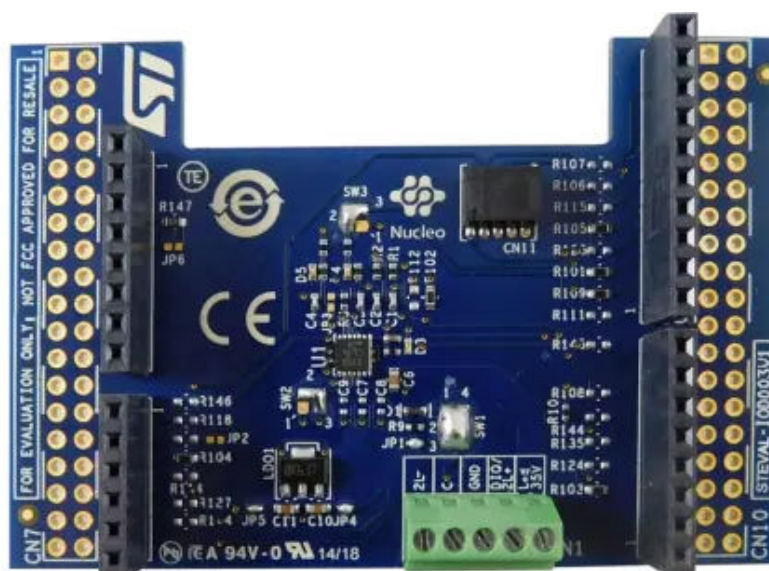
3.5 STEVAL-IOD003V1 expansion board

The **STEVAL-IOD003V1** evaluation board is based on the **L6362A** IO-Link PHY device with full integrated EMC protection (according to IO-Link v1.1 specification) and surge protection (according to IEC 60947-5-2). It provides an affordable and easy-to-use solution for the development of IO-Link and SIO applications, letting you easily evaluate the communication features and robustness of the **L6362A**.

The on-board linear regulators (12 mA-3.3 V from **L6362A** and 100 mA-12 V from **L78L12ABUTR**) can be used to supply the microcontroller via the 24 V bus, instead of via USB. When the **L78L12ABUTR** is enabled (default configuration), you can also perform evaluation of complete industrial sensor modules by connecting the **STEVAL-IOD003V1** to a **NUCLEO-L073RZ** or **NUCLEO-L452RE** board and using an expansion board with sensors.

The **STEVAL-IOD003V1** interfaces with the STM32 controller via UART and GPIO pins and is compatible with the Arduino UNO R3 (default configuration) and ST morpho (optional, not mounted) connectors.

Figure 5. STEVAL-IOD003V1 evaluation board



3.6 Hardware setup

The following hardware components are needed:

- One USB type A to Mini-B USB cable to connect the STM32 Nucleo to the PC when using a [NUCLEO-L073RZ](#) or a [NUCLEO-L452RE](#). The [STEVAL-IOD003V1](#) is connected to the STM32 Nucleo board through the matching Arduino connectors.

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

- An IO-Link master (for example, [P-NUCLEO-IOM01M1](#)) with the associated control software.
- A set of wires for the connection of L+, L-, and CQ lines between the IO-Link master and the [STEVAL-IOD003V1](#).

3.7 Software setup

The following software components allow setting up a suitable development environment for creating applications for the STM32 Nucleo equipped with the [STEVAL-IOD003V1](#) expansion board:

- [STEVAL-IOD003V1](#): an expansion for STM32Cube dedicated to application development, which requires the use of [L6362A](#). The [STEVAL-IOD003V1](#) 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 + STLINK
 - Keil® + STLINK
 - STM32CubeIDE + STLINK

3.8 Board setup

The STM32 Nucleo development board must be configured with the following jumper position for [NUCLEO-L073RZ](#) or [NUCLEO-L452RE](#):

- JP5 on U5V for firmware flashing
- JP1 open
- JP6 closed
- CN2: closed 1-2, 3-4
- CN3 open
- CN4 1-6 open
- CN11 closed
- CN12 closed

The [STEVAL-IOD003V1](#) expansion board has to be configured as follows:

- SW1 all position closed implement the IO-Link configuration
- SW2 close 2-3 (default), [L6362A](#) configured to work with 3.3 V digital signals
- SW3 close 2-3 (default), [L6362A](#) EN pin driven by the microcontroller
- JP1 open, this footprint can be used to host a blocking diode equivalent to D1, for NPN loads
- JP2 open (default), DI/DQ signal is not used for this application board
- JP3 open (default), the [L6362A](#) VDD supplies only the LED D3
- JP4 open, LDO1 disconnected from L+
- JP5 open, LDO1 disconnected from VIN
- JP6 open (default), pull-up of RST signal managed by the STM32 Nucleo

Step 1. Check the configuration of JP5 (on [NUCLEO-L073RZ](#) or [NUCLEO-L452RE](#)) and set it to “U5V”.

Step 2. Connect the mini-USB for [NUCLEO-L073RZ](#) or [NUCLEO-L452RE](#) 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 STM32L4x).
- Step 4.** Disconnect the USB cable from the STM32 Nucleo development board and close JP5 to “E5V” (on NUCLEO-L073RZ or NUCLEO-L452RE).
- Step 5.** Connect the STEVAL-IOD003V1 to the STM32 Nucleo development board through the Arduino connectors.
- Step 6.** Connect the STEVAL-IOD003V1 power section (CN1) 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 three IODD files according to the communication speed (COM1, COM2 or COM3) of the firmware downloaded on the STM32 Nucleo development board.
- Step 8.** Activate your IO-Link master (which usually requires a connection to a 24 V supply rail). The IO-Link master control tool lets you supply the STEVAL-IOD003V1 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 8. 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.
18-Nov-2022	3	Updated Section 2.3 Folder structure, Section 2.3.1 BSPs, Section 2.3.1.1 STM32L0xx_Nucleo, STM32G0xx_Nucleo, Section 2.3.1.2 I6364, Section 2.3.1.3 IOD02A1, Section 2.3.2 Projects, Section 2.4 Software required resources, Section 3.3 Software setup, and Section 3.4 Board setup.
27-Jun-2024	4	Added references to STEVAL-IOD003V1, Section 2.3.1.3: L6362A , Section 2.3.1.5: IOD003V1 , Section 3.5: STEVAL-IOD003V1 expansion board , Section 3.6: Hardware setup , Section 3.7: Software setup and Section 3.8: Board setup . Updated Section Introduction, Section 2.1: Overview, Section 2.2: Architecture, Section 2.3: Folder structure, Section 2.3.1: BSPs, Section 2.3.2: Projects, Section 2.4: Software required resources, Section 2.5: APIs, Section 2.6: Sample application description. Added NUCLEO-L452RE and NUCLEO-F303RE for L6364.

Contents

1	Acronyms and abbreviations	2
2	X-CUBE-IOD02 software expansion for STM32Cube	3
2.1	Overview	3
2.2	Architecture	3
2.3	Folder structure	4
2.3.1	BSPs	5
2.3.2	Projects	6
2.4	Software required resources	11
2.5	APIs	12
2.6	Sample application description	12
3	System setup guide	13
3.1	Hardware description	13
3.1.1	STM32 Nucleo	13
3.1.2	X-NUCLEO-IOD02A1 expansion board	13
3.2	Hardware setup	14
3.3	Software setup	15
3.4	Board setup	15
3.5	STEVAL-IOD003V1 expansion board	16
3.6	Hardware setup	17
3.7	Software setup	17
3.8	Board setup	17
	Revision history	19
	List of tables	21
	List of figures	22
	Disclaimer	23

List of tables

Table 1.	List of acronyms	2
Table 2.	Projects\STM32G071RB-Nucleo\Applications\IOD02A1	6
Table 3.	Projects\STM32L073RZ-Nucleo\Applications\IOD02A1	7
Table 4.	Projects\STM32L452RE-Nucleo\Applications\IOD02A1	7
Table 5.	Projects\STM32F303RE-Nucleo\Applications\IOD02A1	8
Table 6.	Projects\STM32L073RZ-Nucleo\Applications\IOD003V1	9
Table 7.	Projects\STM32L452RE-Nucleo\Applications\IOD003V1	10
Table 8.	Document revision history	19

List of figures

Figure 1.	X-CUBE-IOD02 expansion software architecture.	4
Figure 2.	X-CUBE-IOD02 package folder structure	4
Figure 3.	STM32 Nucleo board	13
Figure 4.	X-NUCLEO-IOD02A1 expansion board	14
Figure 5.	STEVAL-IOD003V1 evaluation board	16

IMPORTANT NOTICE – 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 acknowledgment.

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. For additional information about ST trademarks, refer to www.st.com/trademarks. 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.

© 2024 STMicroelectronics – All rights reserved