UM1914
User manual

Getting started with the X-CUBE-PLC1 industrial input/output software expansion for STM32Cube

Introduction

This document describes how to get started with the X-CUBE-PLC1 software.

X-CUBE-PLC1 provides a complete software package for STM32 to build applications using industrial input/output expansion boards (the industrial input stage is implemented through the CLT01-38SQ7 device, while industrial outputs are implemented through the VNI8200XP device).

It is easily portable across different MCU families thanks to STM32Cube. This package contains software functions implementing the most common PLC operations which can be combined with the 8 inputs and 8 outputs available with the devices listed above.

The software provides implementation examples for STM32 Nucleo platforms equipped with the X-NUCLEO-PLC01A1 expansion board, featuring logic operators, counters and output driven at fixed frequency.

The software is based on STM32Cube technology and expands the STM32Cube-based set of packages.

Contents

1 What is STM32Cube?

1.1 STM32Cube overview

The STMCube™ initiative was developed by STMicroelectronics to help developer reduce development effort, time and cost. STM32Cube covers the entire STM32 portfolio.

STM32Cube Version 1.x includes:

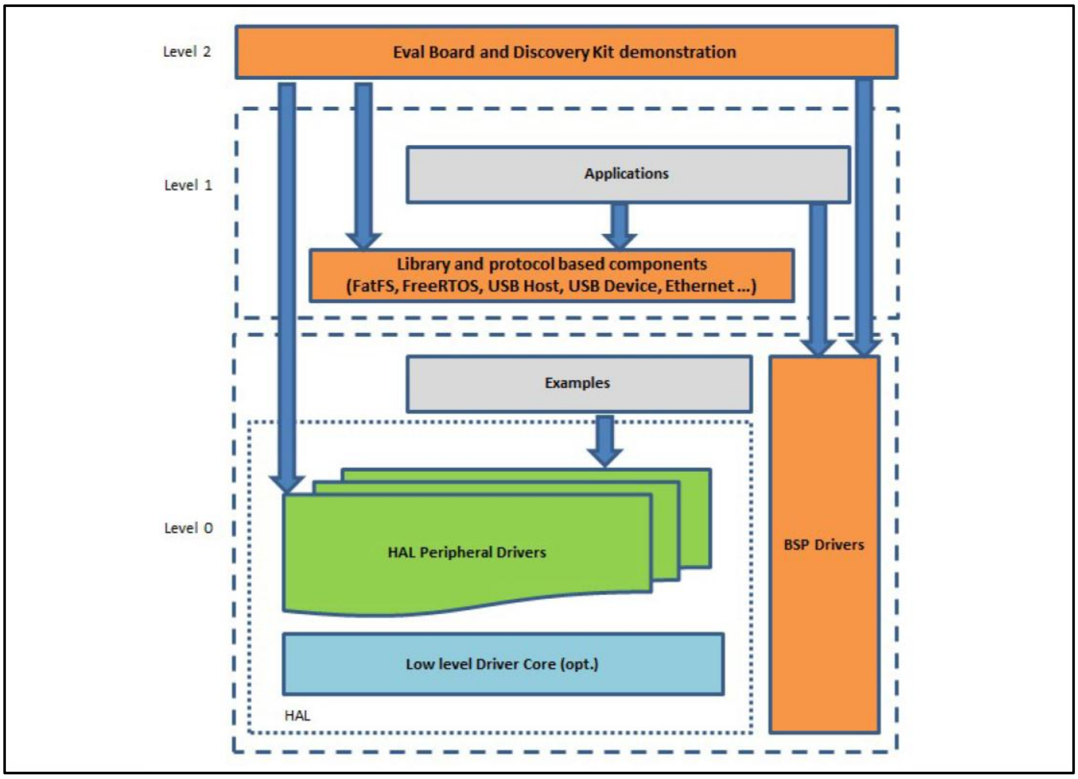
- The STM32CubeMX graphical software configuration tool for generating C initialization code using graphical wizards.
- A comprehensive embedded software platform for each series (e.g., STM32CubeF4 for the STM32F4 series)
  - STM32Cube HAL, an STM32 abstraction layer embedded software, ensuring maximum portability across the STM32 portfolio
  - A consistent set of middleware components such as RTOS, USB, TCP/IP and Graphics
  - All embedded software utilities along with a full set of examples

Information regarding STM32Cube is available on st.com at: http://www.st.com/stm32cube

1.2 STM32Cube architecture

The STM32Cube firmware solution is built around three independent levels that can easily interact with each other, as described in the figure below:

Figure 1: Firmware architecture


Level 2
Eval Board and Discovery Kit demonstration
Level 1
Applications
Library and protocol based components
(FatFS, FreeRTOS, USB Host, USB Device, Ethernet...)
Examples
Level 0
HAL Peripheral Drivers
Low level Driver Core (opt.)
HAL
BSP Drivers

Level 0: This level is divided into three sub-layers:

- Board support package (BSP): this layer offers a set of APIs relative to the components in the hardware boards (Audio codec, IO expander, Touchscreen, SRAM driver, LCD drivers, etc.) and is composed of two parts:
  - Component driver: is the driver relative to the external device on the board and not the STM32; the component drivers provide specific APIs to the BSP driver external components and can be ported to any other board.
  - BSP driver: links the component driver to a specific board and provides a set of user friendly APIs. The API naming convention is BSP_FUNCT_Action() (e.g., BSP_LED_Init(), BSP_LED_On(), etc.)

The BSP is based on modular architecture which is easily ported to any hardware by just implementing the low level routines.

- Hardware abstraction layer (HAL): this layer provides the low level drivers and the hardware interfacing methods to interact with the upper layers (application, libraries and stacks). It provides generic, multi instance and function-oriented APIs which remove the need for user applications by providing ready to use processes.
  For instance, for communication peripherals (I²S, UART, etc.), it provides APIs to initialize and configure the peripheral, manage data transfer based on polling, interrupt or DMA processes, and manage communication errors that may arise during communication.
  The HAL driver APIs are split in two categories:
  - generic APIs for common and generic functions across the STM32 series
  - xtension APIs for specific and customized functions for a specific family or a specific part number
- Basic peripheral usage examples: this layer includes the examples built around the STM32 peripheral using only the HAL and BSP resources.

Level 1 is divided into two sub-layers:

- Middleware components: a set of libraries covering USB host and device libraries, STemWin, FreeRTOS, FatFS, LwIP, and PolarSSL. Horizontal interaction between the components in this layer is performed directly by calling the feature APIs, while the vertical interaction with the low level drivers is handled through specific callbacks and static macros implemented in the library system call interface. For example, the FatFs implements the disk I/O driver to access a microSD drive or the USB Mass Storage Class.
- Examples based on the middleware components: each middleware component comes with one or more examples (also called applications) demonstrating how to use it. Integration examples involving several middleware components are also provided.

Level 2 is composed of a single layer providing a global, real-time and graphical demonstration based on the middleware service layer, the low level abstraction layer and the basic peripheral usage applications for board-based functions.

2 X-CUBE-PLC1 software, expansion for STM32Cube

2.1 Overview

The X-CUBE-PLC1 software package expands the STM32Cube functionality.

The key features of the package are:

- Complete board support package layer to build applications using 8x input with CLT01-38SQ7 (high-speed protected digital termination array) and 8x output with VNI8200XP (high-side solid state relay)
- Easy portability across different MCU families, thanks to STM32Cube
- Sample applications to implement the most common PLC functions
- Free user-friendly license terms
- Example applications for the X-NUCLEO-PLC01A1 board plugged on top of either a NUCLEO-F030R8, NUCLEO-F103RB or NUCLEO-F401RE board.

This software enables the collection and management of industrial input and output data from CLT01-38SQ7 and VNI8200XP devices respectively, running on STM32.

2.2 Architecture

This software is a fully compliant expansion for the STM32Cube, expanding its functionality to enable the development of applications using industrial input and output devices. Please see the previous chapter for information regarding the STM32Cube architecture.

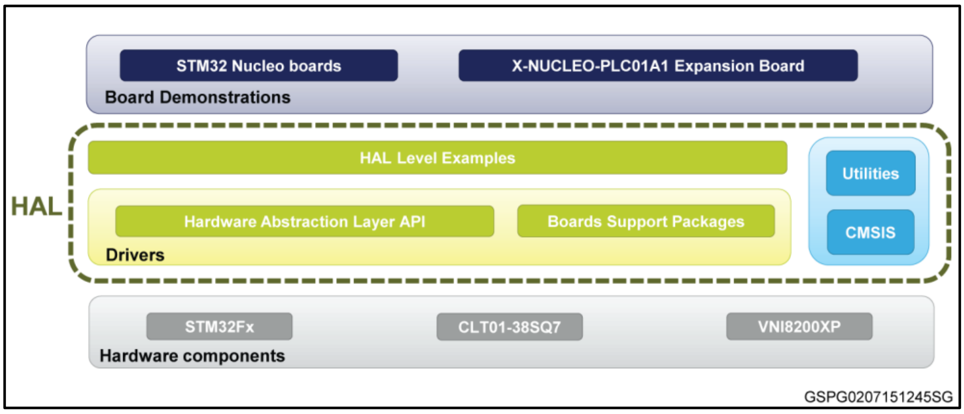
The software is based on the STM32CubeHAL hardware abstraction layer for the STM32 microcontroller. The package extends STM32Cube by providing a board support package (BSP) for the industrial input/output expansion board.

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

- The STM32Cube HAL layer: provides a simple, generic and multi-instance set of APIs (application programming interfaces) to interact with the upper layers (applications, libraries and stacks). It is composed of generic and extension APIs built around a generic architecture and allows the layers that are built upon to implement their functions independent of the specific hardware configuration of a given Microcontroller Unit (MCU). This structure improves the library code reusability and guarantees easy portability to other devices.
- The board support package (BSP) layer provides support for the peripherals on the STM32 Nucleo board, other than the MCU. It is a limited set of APIs providing a programming interface for certain board-specific peripherals, like the timer. It also provides support for reading and managing industrial inputs and outputs.

The figure below outlines the software architecture of the package:

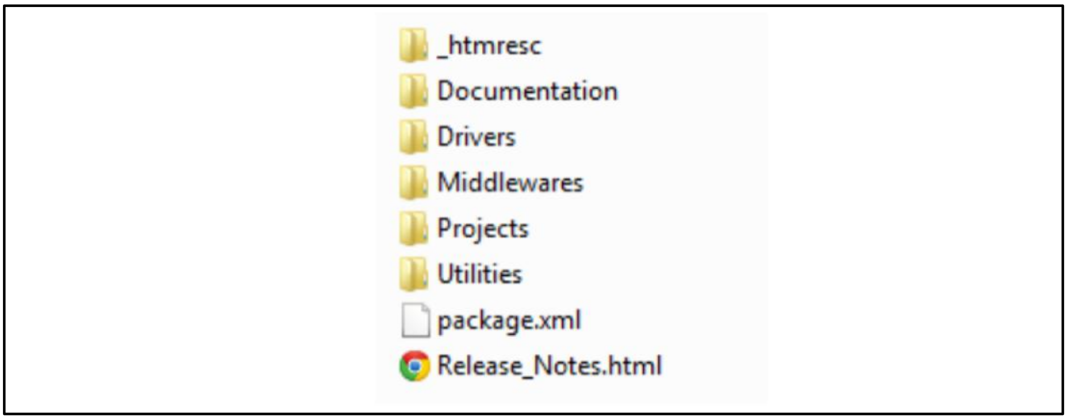
Figure 2: X-CUBE-PLC1 software architecture



2.3 Folders structure

The figure below outlines the architecture of the package:

Figure 3: X-CUBE-PLC1 package folders structure



_htmresc
Documentation
Drivers
Middlewares
Projects
Utilities
package.xml
Release_Notes.html

The following folders are included in the software package:

Documentation: this folder contains a compiled HTML file generated from the source code and documentation in details the software components and APIs.

Drivers: this folder contains the HAL drivers, the board-specific drivers for each supported board or hardware platform, including the on-board components and the CMSIS layer, which is a vendor-independent hardware abstraction layer for the Cortex-M processor series.

Projects: this folder contains a sample application used to handle industrial inputs and outputs using SPI communication for the CLT01-38SQ7 and the VNI8200XP devices. It has been developed for the NUCLEO-F030R8, NUCLEO-F103RB and NUCLEO-F401RE platforms with three development environments (IAR Embedded Workbench for ARM, Keil uVision5 Development Tool (MDK-ARM), SW4STM32 Development Tool).

2.4 APIs

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.

2.5 Firmware description

An application example using the X-NUCLEO-PLC01A1 expansion board with either the NUCLEO-F030R8, NUCLEO-F103RB or NUCLEO-F401RE board is provided in the *Projects* directory. Ready-to-build projects are available for multiple IDEs.

The firmware includes a list of functions covering the most common operations involving a PLC:

- Signal Mirror: transfer the selected inputs to the corresponding output channels
- Output Freeze: the selected output channel is kept high for the selected duration
- Output Regroup: the VNI8200XP gives the possibility to parallelize outputs in order to drive loads requiring higher current
- Inputs Sum: some PLCs; calculate sum of the inputs for statistical purposes or for security applications
- Output ON: select the number of output channels to be kept high until other commands occur
- Output OFF: select the number of output channels to be kept low until other commands occur
- Inputs AND: sets the selected input channels to AND logic, transferring these values to the corresponding output channels
- Inputs OR: sets the selected input channels to OR logic, transferring these values to the corresponding output channels
- Inputs NOT: sets the selected inputs to NOT logic, transferring these values to the corresponding output channels
- Inputs XOR: ets the selected input channels to XOR logic, using the selected output as the result
- Output Cycling: the selected output channel is driven at a fixed frequency and duty cycle. The maximum frequency allowed by the current firmware version is 8 kHz. The duty cycle associated with this maximum frequency is 5% below that expected.
- Building any dedicated firmware easily starting from the functions listed above and taking advantage of STM32Cube.

3 System setup guide

3.1 Hardware description

This section describes the hardware components needed for developing a PLC based application.

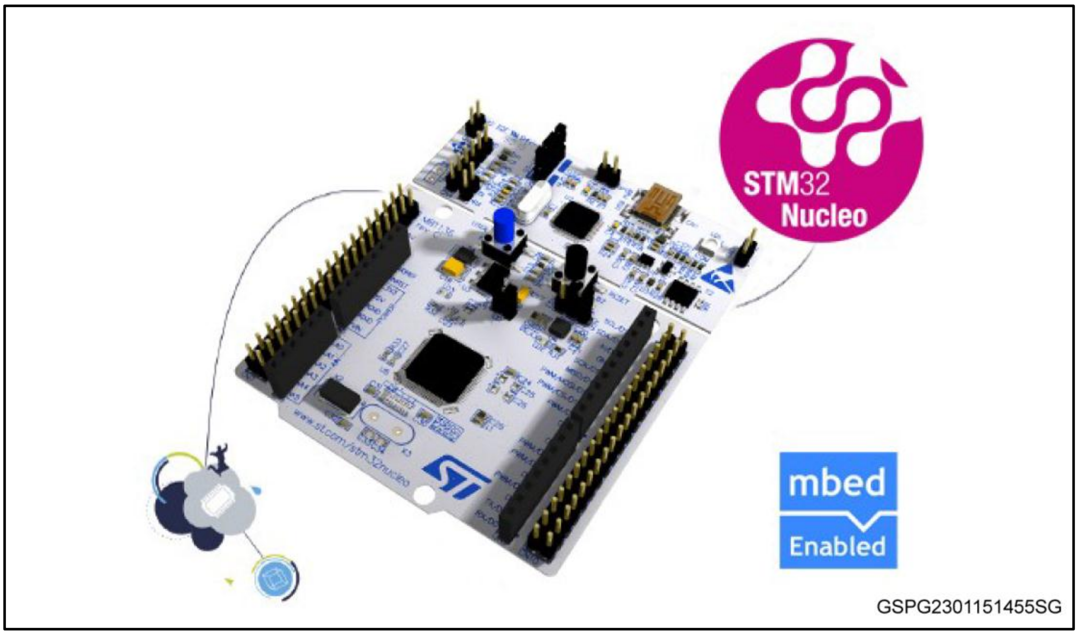
The following sub-sections describe the individual components.

3.1.1 STM32 Nucleo platform

The STM32 Nucleo boards provide an affordable and flexible way for users to try out new ideas and build prototypes with any STM32 microcontroller lines. The Arduino™ connectivity support and ST Morpho headers make it easy to expand the functionality of the STM32 Nucleo open development platform with a wide choice of specialized expansion boards. The STM32 Nucleo board does not require any separate probes as it integrates the ST-LINK/V2-1 debugger/programmer. The STM32 Nucleo board comes with the STM32 comprehensive software HAL library together with various packaged software examples.

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

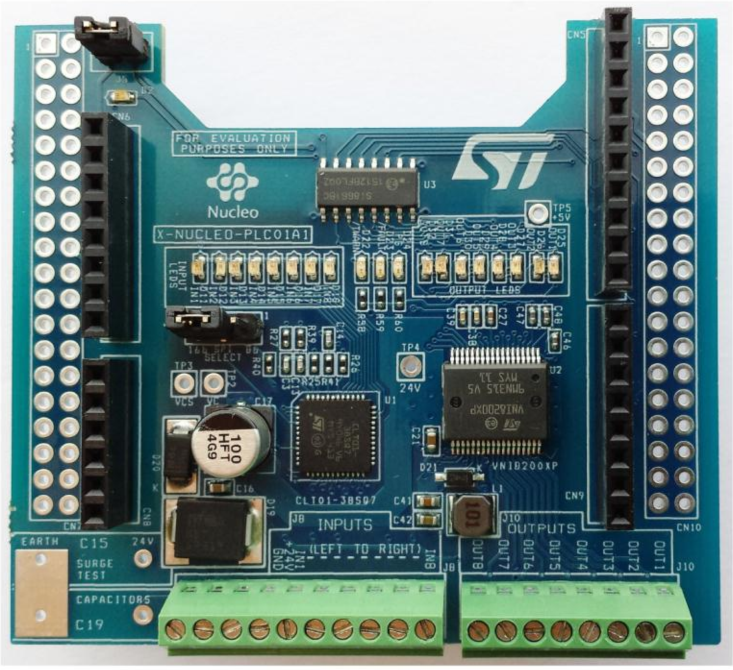
Figure 4: STM32 Nucleo board



3.1.2 X-NUCLEO-PLC01A1 expansion board

The X-NUCLEO-PLC01A1 is an industrial input / output expansion board usable with the STM32 Nucleo system. It is also compatible with Arduino UNO R3 connector layout, and is designed around the STMicroelectronics CLT01-38SQ7 (current limited termination array device) and VNI8200XP (high-side solid state relay device), allowing the management of 8x industrial input and 8x industrial output respectively. The X-NUCLEO-PLC01A1 interfaces with the STM32 MCU via SPI pins.

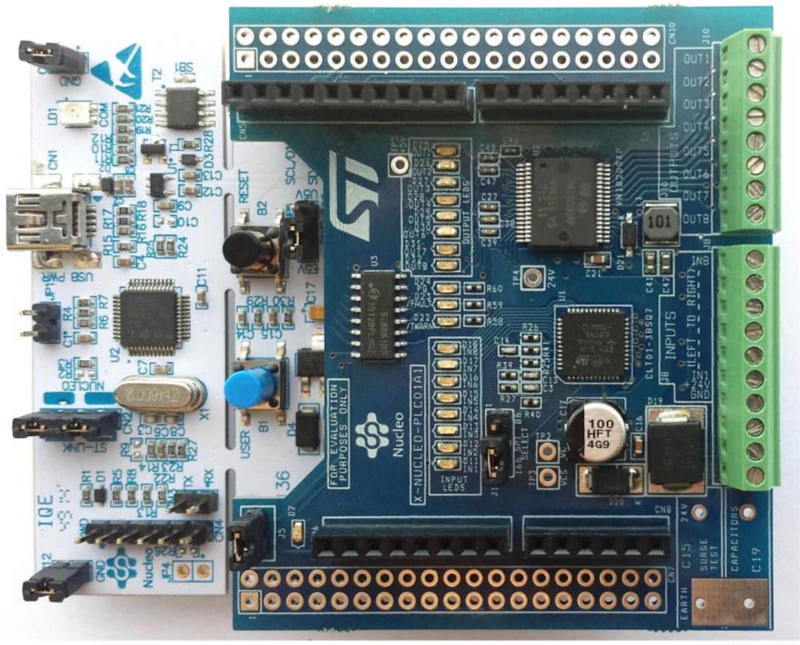
Figure 5: X-NUCLEO-PLC01A1, industrial input/output expansion board



GSPG0207151250SG

Information about the X-NUCLEO-PLC01A1 expansion board is available on st.com at http://www.st.com/x-nucleo.

Figure 6: X-NUCLEO-PLC01A1 expansion board plugged on STM32 Nucleo board



GSPG0207151255SG

3.2 Software description

The following software components are required in order to set up a suitable development environment for creating applications for the STM32 Nucleo equipped with the industrial input/output expansion board:

- X-CUBE-PLC1: an expansion for STM32Cube dedicated to PLC application developments. The X-CUBE-PLC1 firmware and related documentation is available on st.com.
- Development tool-chain and compiler: The STM32Cube expansion software supports the following environments:
  - IAR Embedded Workbench for ARM® (EWARM) toolchain + ST-Link
  - Keil uVision5 Development Tool (MDK-ARM) toolchain + ST-LINK
  - SW4STM32 Development Tool + ST-LINK

3.3 Hardware and Software setup

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

3.3.1 Hardware setup

The following hardware components are needed:

1. One STM32 Nucleo development platform (suggested order code: NUCLEO-F030R8 or NUCLEO-F103RB or NUCLEO-F401RE)
2. One industrial input/output expansion board (order code: X-NUCLEO-PLC01A1)
3. One USB type A to Mini-B USB cable to connect the STM32 Nucleo to the PC

4 Acronyms and abbreviations

Table 1: Acronyms

Acronym	Description
PLC	Programmable Logic Controller

5 Revision history

Table 2: Document revision history

Date	Revision	Changes
06-Aug-2015	1	Initial release.