# UM2189
# User manual

## Getting started with the X-CUBE-PLM1 power line communication software expansion for STM32Cube

## Introduction

The X-CUBE-PLM1 expansion software package for STM32Cube runs on STM32 and includes drivers that allow interfacing a ST7580 power line networking system-on-chip.

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

The software comes with all the drivers needed to communicate with the ST7580 power line device on the X-NUCLEO-PLM01A1 expansion board. After downloading the drivers to a NUCLEO-F401RE or NUCLEO-L053R8 development board, you can start evaluating the communication features of the ST7580 using a DC two-wire link between two boards.

# Contents

1 What is STM32Cube? ..... 4
1.1 STM32Cube architecture ..... 4
2 X-CUBE-PLM1 software expansion for STM32Cube ..... 6
2.1 Overview ..... 6
2.2 Architecture ..... 6
2.3 Folder structure ..... 7
2.4 APIs ..... 8
2.5 Sample application description ..... 8
3 System setup guide ..... 11
3.1 Hardware description ..... 11
3.1.1 STM32 Nucleo platform ..... 11
3.1.2 X-NUCLEO-PLM01A1 expansion board ..... 12
3.2 Hardware and software setup ..... 12
3.2.1 Hardware setup ..... 12
3.2.2 Software setup ..... 13
3.3 System setup guide ..... 13
3.3.1 STM32 Nucleo and X-NUCLEO-PLM01A1 expansion board setup 13
4 Revision history ..... 14

# List of figures

Figure 1: Firmware architecture ..... 4
Figure 2: X-CUBE-PLM1 software architecture ..... 7
Figure 3: X-CUBE-PLM1 package folder structure ..... 7
Figure 4: X-NUCLEO-PLM01A1 expansion board and STM32 Nucleo: master and slave interaction flow ..... 9
Figure 5: X-NUCLEO-PLM01A1 expansion board and STM32 Nucleo: debug prints during sample communication ..... 10
Figure 6: STM32 Nucleo board ..... 11
Figure 7: X-NUCLEO-PLM01A1 expansion board ..... 12
Figure 8: Power line communication environment ..... 13

# 1 What is STM32Cube?

STMCube™ represents the STMicroelectronics initiative to make developers' lives easier by reducing development effort, time and cost. STM32Cube covers the STM32 portfolio.

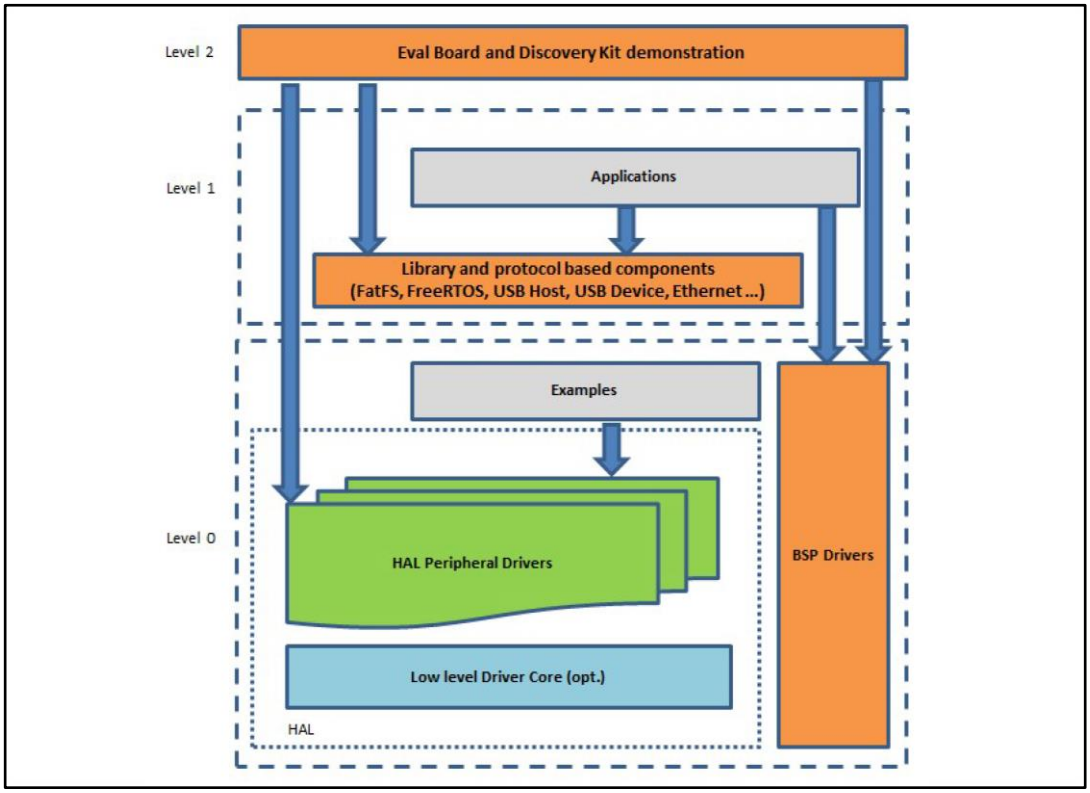
STM32Cube version 1.x includes:

- STM32CubeMX, a graphical software configuration tool that allows the generation of C initialization code using graphical wizards.
- A comprehensive embedded software platform specific to each series (such as the STM32CubeF4 for the STM32F4 series), which includes:
  - the STM32Cube HAL embedded abstraction-layer software, ensuring maximized portability across the STM32 portfolio
  - a consistent set of middleware components such as RTOS, USB, TCP/IP and graphics
  - all embedded software utilities with a full set of examples

## 1.1 STM32Cube architecture

The STM32Cube firmware solution is built around three independent levels that can easily interact with one another, as described in the diagram below.

**Figure 1: Firmware architecture**




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

- Board Support Package (BSP): this layer offers a set of APIs relative to the hardware components in the hardware boards (Audio codec, IO expander, Touchscreen, SRAM driver, LCD drivers. etc…); it is based on modular architecture allowing it to be easily

ported on any hardware by just implementing the low level routines. It is composed of two parts:

  - Component: is the driver relative to the external device on the board and not related to the STM32, the component driver provides specific APIs to the external components of the BSP driver, and can be ported on any other board.
  - BSP driver: links the component driver to a specific board and provides a set of easy to use APIs. The API naming convention is BSP_FUNCT_Action(): e.g., BSP_LED_Init(), BSP_LED_On().

- 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 to help offload user application development time by providing ready to use processes. For example, for the communication peripherals (I²C, UART, etc.) it provides APIs for peripheral initialization and configuration, data transfer management based on polling, interrupt or DMA processes, and communication error management. The HAL Drivers APIs are split in two categories: generic APIs providing common, generic functions to all the STM32 series and extension APIs which provide special, customized functions for a specific family or a specific part number.
- Basic peripheral usage examples: this layer houses the examples built around the STM32 peripherals using the HAL and BSP resources only.

**Level 1**: This level is divided into two sub-layers:

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

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

# 2 X-CUBE-PLM1 software expansion for STM32Cube

## 2.1 Overview

The X-CUBE-PLM1 software package extends the STM32Cube functionality.

The key features are:

- Complete software to drive the ST7580 power line device hosted by X-NUCLEO-PLM01A1 expansion board
- Example firmware available for point-to-point communication, compatible with STM32Cube firmware
- Easy portability across different MCU families, thanks to STM32Cube
- Sample application to transmit real-time communication data to a PC
- Free, user-friendly license terms
- Sample implementation available on XNUCLEO- PLM01A1 board connected to a NUCLEO-F401RE or NUCLEO-L053R8 development board

The software comes with all the drivers needed to interface with the ST7580 system-on-chip device on the X-NUCLEO-PLM01A1 expansion board. After downloading the software libraries to a NUCLEO-F401RE or NUCLEO-L053R8 development board, you can have full access to the device functions.

The package also includes a sample application to configure the device[a], to perform transmission and reception during a power line communication process. The developer can use it to start experimenting with the code.

## 2.2 Architecture

This software is a fully compliant expansion for STM32Cube enabling development of power line communication applications[b].

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 power line expansion board and a sample application for point-to-point communication.

The software layers used by the application software to access the power line expansion board are:

- **STM32Cube HAL driver layer**: this layer provides a generic multi instance simple set of APIs (application programming interfaces) to interact with the upper layers (application, libraries and stacks). It is composed of generic and extension APIs. It is directly built around a generic architecture and allows the layers that are built upon, such as the middleware layer, to implement their functionalities without dependencies on the specific hardware configuration for a given Microcontroller Unit (MCU). This structure improves the library code reusability and guarantees high portability on other devices.
- **Board Support Package (BSP) layer**: the software package needs to support the peripherals on the STM32 Nucleo board apart from the MCU. It contains a limited set of APIs which provides a programming interface for certain board specific peripherals

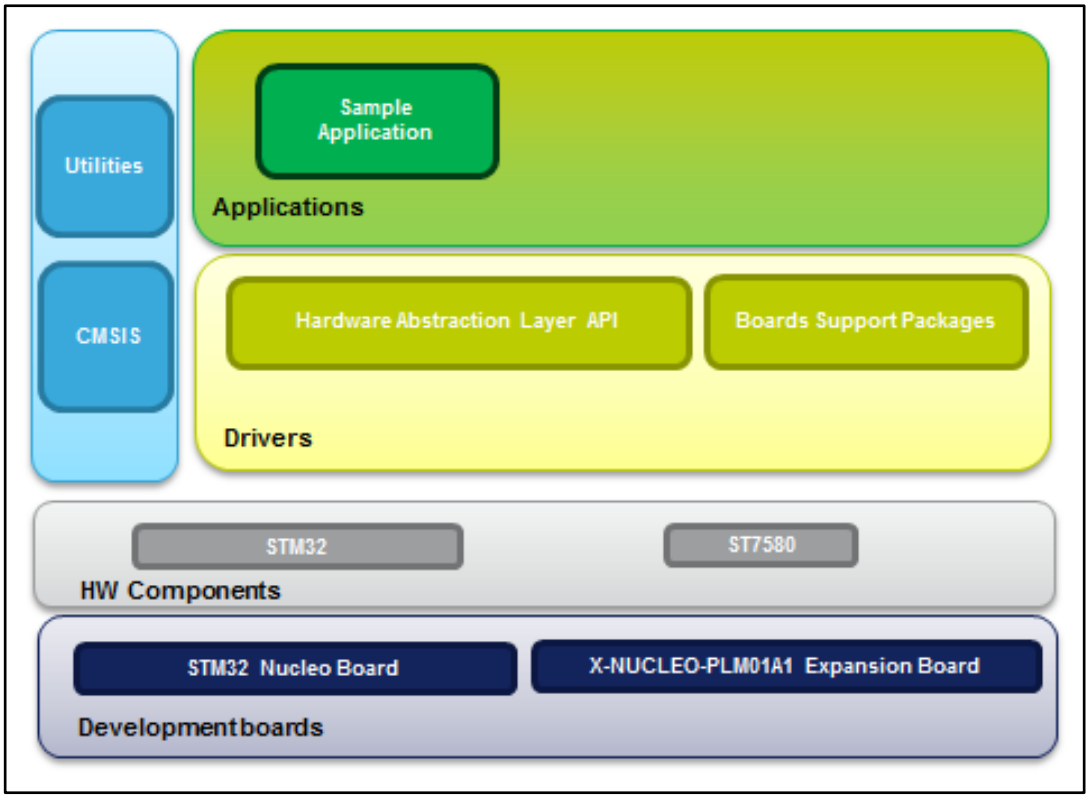
[a] Refer to ST7580 datasheet on www.st.com for the device complete configuration parameters and commands

[b] Refer to the previous chapter for an overview of the STM32Cube architecture.

(e.g. the LED, the user button etc.) and allows identifying the specific board version. For the power line expansion board, it provides the programming interface for various inertial and environmental sensors and provides support for initializing and reading sensor data.

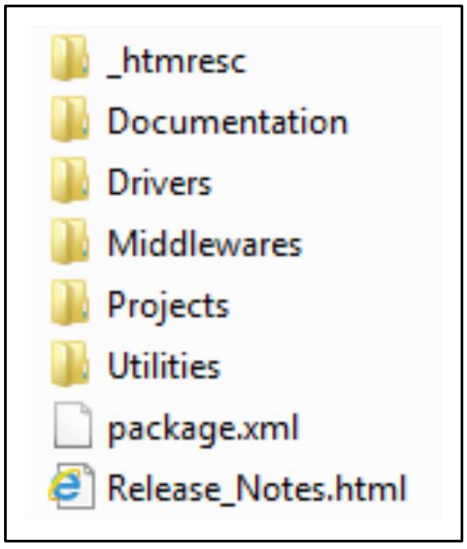
The following figure outlines the software architecture of the package:

**Figure 2: X-CUBE-PLM1 software architecture**




## 2.3 Folder structure

**Figure 3: X-CUBE-PLM1 package folder structure**



The following folders are included in the software package:

- Documentation: contains a compiled HTML file generated from the source code and documentation detailing the software components and APIs.
- Drivers: contains the HAL drivers and the board specific drivers for each supported board or hardware platform, including the on-board components and the CMSIS layer - a vendor-independent hardware abstraction layer for the ARM Cortex-M processor series.
- Projects: contains a sample application for the NUCLEO-L053R8 and NUCLEO-F401RE platforms to access power line device data and functionalities; it is provided with three development environments (IAR Embedded Workbench for ARM, RealView MDK-ARM Microcontroller Development Kit, System Workbench for STM32).

## 2.4 APIs

To write a PLM application based on the STM32 Nucleo board equipped with an X-NUCLEO-PLM01A1 expansion board, a list of APIs is available to the user.[a]

A brief description for each API and relative prototype follows.

**Initialization**

- `void BSP_PLM_Init(void)`: initializes the PLM module.
- `int BSP_PLM_Reset(void)`: resets the PLM module.
- `int BSP_PLM_Ping(const uint8_t* pingBuf, uint8_t pingLen)`: pings the PLM module.

**MIB configuration**

- `int BSP_PLM_Mib_Write(uint8_t indexMib, const uint8_t* bufMib, uint8_t lenBuf)`: writes a PLM MIB parameter.
- `int BSP_PLM_Mib_Read(uint8_t indexMib, uint8_t* bufMib, uint8_t lenBuf)`: reads a PLM MIB parameter.
- `int BSP_PLM_Mib_Erase(uint8_t indexMib)`: erases a PLM MIB parameter.

**PLM communication**

- `int BSP_PLM_Send_Data(uint8_t plmOpts, const uint8_t* dataBuf, uint8_t dataLen, uint8_t* confData)`: sends data via PLM communication.
- `int BSP_PLM_Send_Secure_data(uint8_t plmOpts, const uint8_t* dataBuf, uint8_t clrLen, uint8_t encLen, uint8_t* retData)` sends encrypted data via PLM communication.
- `ST7580Frame *BSP_PLM_Receive_Frame(void)`: receives data via PLM communication.

## 2.5 Sample application description

A sample application using the X-NUCLEO-PLM01A1 expansion board with the NUCLEO-F401RE or NUCLEO-L053R8 board is provided in the "Projects" directory. Ready-to-use projects are available for multiple IDEs.

The sample application contains a simple point-to-point communication between two nodes equipped with an STM32 Nucleo board and an X-NUCLEO-PLM01A1 expansion board.

[a] Detailed technical information decribing the user API functions and parameters can be found in the “X-CUBE-PLM1.chm” compiled HTML file located in the software package “Documentation” folder.

To create a demo (using a PC to monitor the activity on the power line), the developer has to use the STM32 Nucleo LEDs and/or buttons. He can assign each node a role as master or slave by pushing the STM32 Nucleo USER button at boot time:

- USER button pressed = master
- USER button released = slave

These options allow having only a single precompiled binary for both roles.

After the role assignment, in an endless loop, the master board sends a TRIGGER frame to the power line (toggling the X-NUCLEO-PLM01A1 green LED) that the slave will receive and recognize (toggling the X-NUCLEO-PLM01A1 red LED).

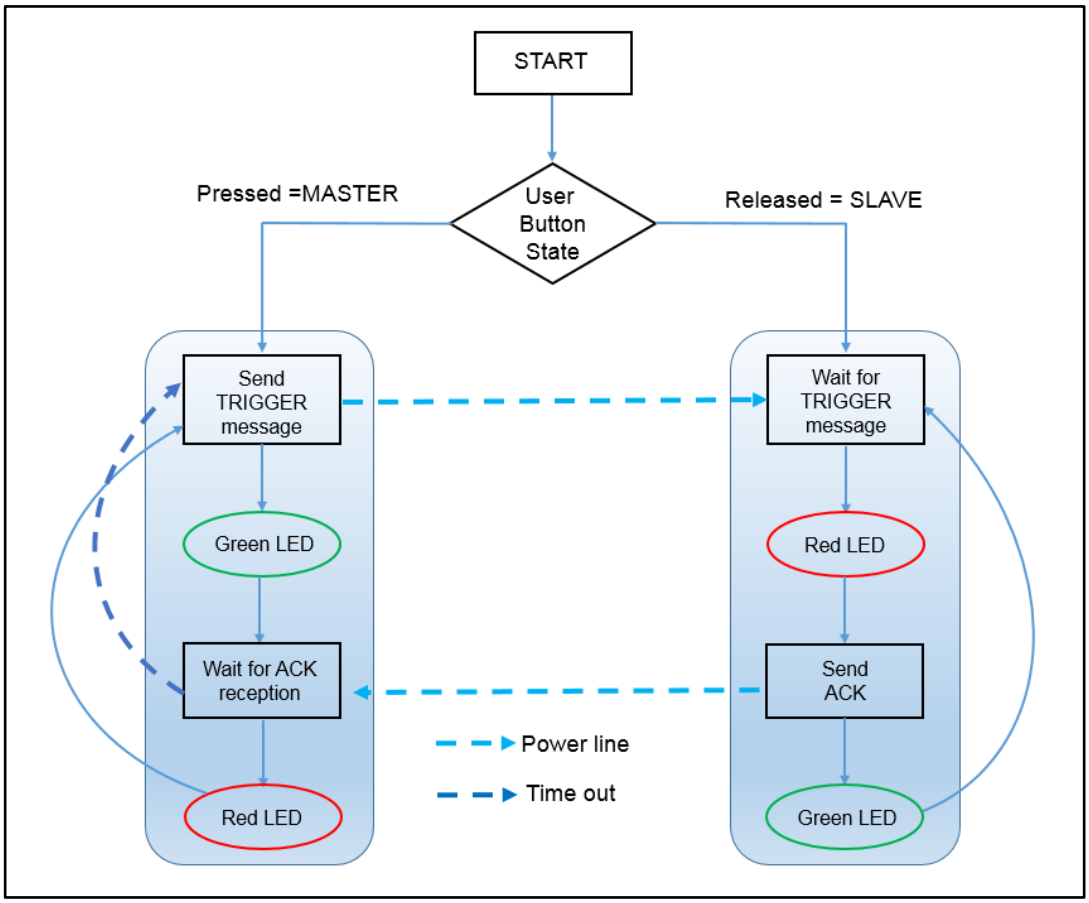
The slave will send back an ACK frame as a confirmation (toggling the X-NUCLEO-PLM01A1 green LED).

The master node will recognize the ACK frame (toggling the X-NUCLEO-PLM01A1 red LED).

If the two nodes are connected through USB cables, they print power line status messages on a debug serial port (e.g. message sent, message received, etc).

The following figure shows the master and slave routine described above.

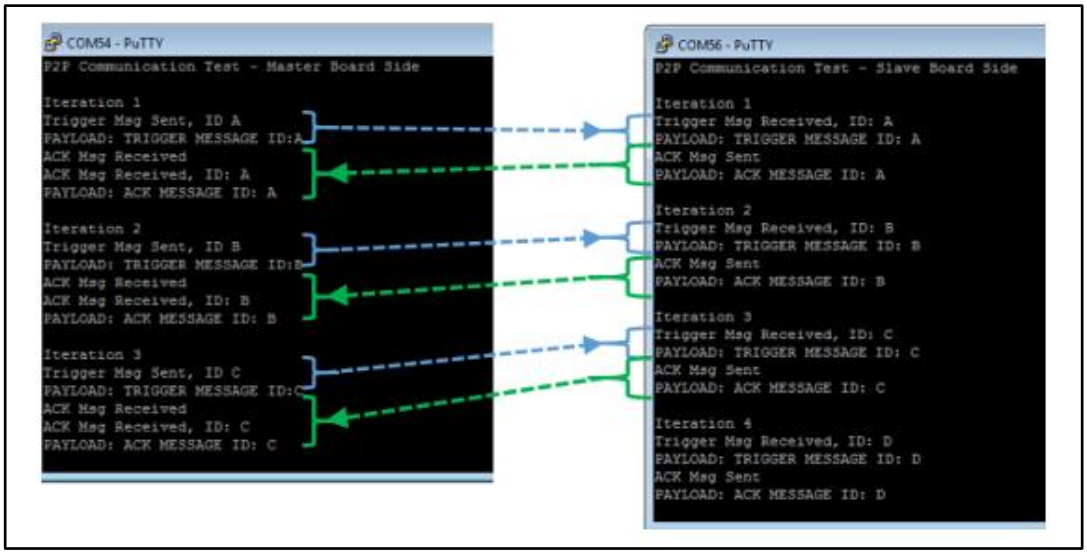
**Figure 4: X-NUCLEO-PLM01A1 expansion board and STM32 Nucleo: master and slave interaction flow**




There is also the possibility to monitor the modem power line activity, through serial sniffers connected to the boards: baudrate 115200, 8 bit and no flow control.

Both the master and the slave will send debug prints, notifying their activity.

**Figure 5: X-NUCLEO-PLM01A1 expansion board and STM32 Nucleo: debug prints during sample communication**



# 3 System setup guide

## 3.1 Hardware description

This section describes the hardware components required for developing power line communication-based application.

The following subsections describe the individual components.

### 3.1.1 STM32 Nucleo platform

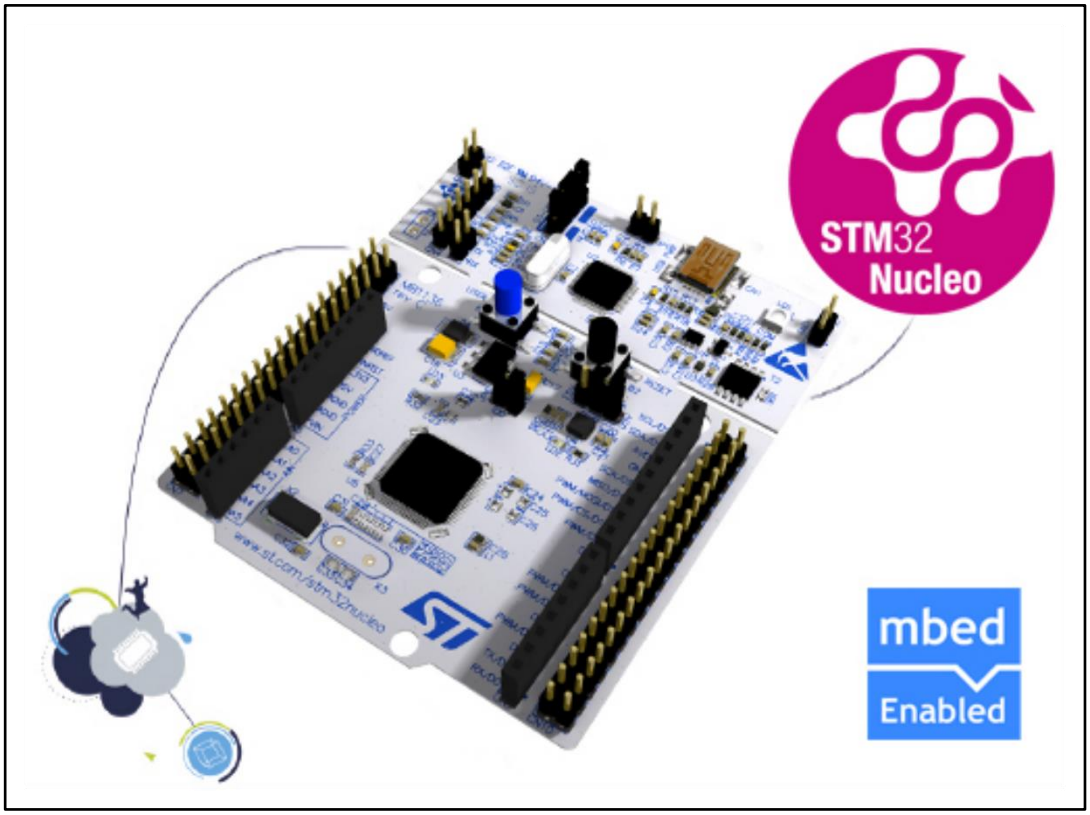
STM32 Nucleo development boards provide an affordable and flexible way for users to test solutions and build prototypes with any STM32 microcontroller line.

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.

**Figure 6: STM32 Nucleo board**



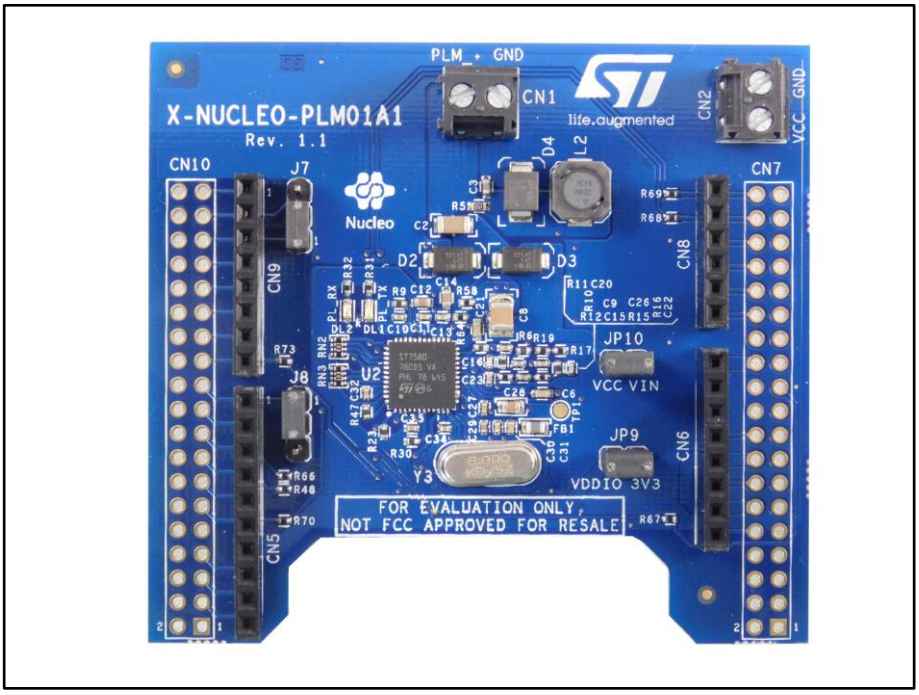

Information regarding the STM32 Nucleo board is available at *www.st.com/stm32nucleo*

### 3.1.2 X-NUCLEO-PLM01A1 expansion board

The X-NUCLEO-PLM01A1 expansion board for STM32 Nucleo is based on the ST7580 FSK, PSK multi-mode power line networking system-on-chip. It provides an affordable and easy-to-use solution for the development of connectivity applications based on power line communication. It lets you easily evaluate the communication features of the ST7580 based on a DC two-wire link between two boards.

You can also perform evaluation on an AC power line by connecting the X-NUCLEO-PLM01A1 to an STEVAL-XPLM01CPL board providing effective AC coupling and isolation. The X-NUCLEO-PLM01A1 is interfaced with the STM32 controller via UART and GPIO pins and is compatible with the Arduino UNO R3 (default configuration) and ST morpho (optional) connectors.

**Figure 7: X-NUCLEO-PLM01A1 expansion board**




## 3.2 Hardware and software setup

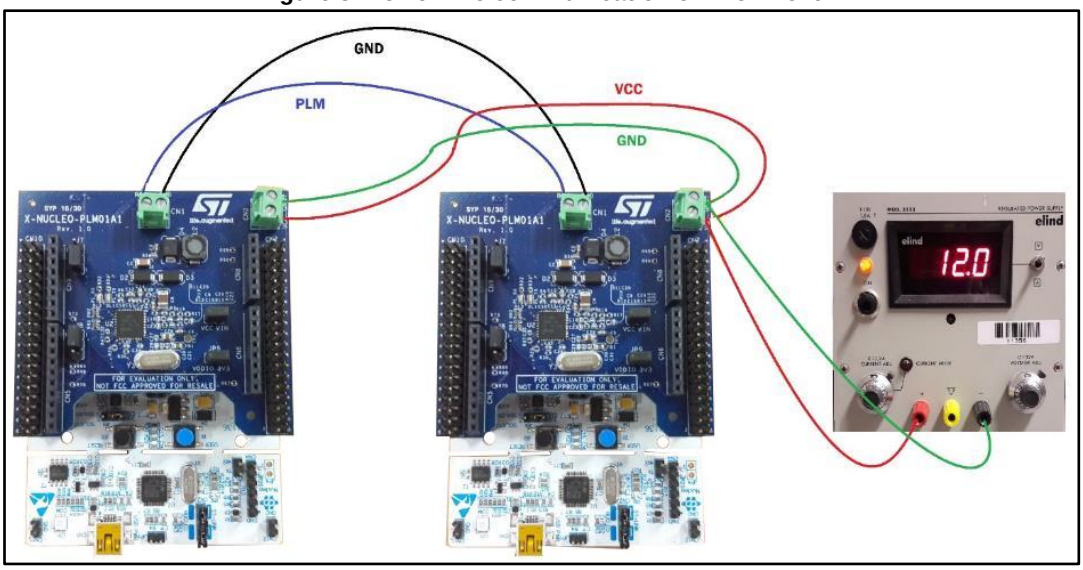
### 3.2.1 Hardware setup

The following hardware components are needed:

- Two STM32 Nucleo development platforms (suggested order code: NUCLEO-F401RE or NUCLEO-L053R8)
- Two power line expansion boards (X-NUCLEO-PLM01A1)
- Two USB type A to Mini-B USB cables to connect the two STM32 Nucleo boards to the PC
- One lab DC power supply to provide 12 V
- Four wires for power supply (2 per board)
- Two wires for PLM connection

The following picture shows the environment detailed above.

**Figure 8: Power line communication environment**




### 3.2.2 Software setup

This section lists the minimum requirements for the developer to set up the SDK, run the sample scenario based on the application demo provided and customize applications.

#### 3.2.2.1 Development Tool-chains and Compilers

Select one of the Integrated Development Environments supported by the STM32Cube expansion software and follow the system requirements and setup information provided by the selected IDE provider.

## 3.3 System setup guide

This section describes how to set up the different components before writing and executing applications on the STM32 Nucleo board with the power line communication expansion board.

### 3.3.1 STM32 Nucleo and X-NUCLEO-PLM01A1 expansion board setup

The STM32 Nucleo board integrates the ST-LINK/V2-1 debugger/programmer. The developer can download the relevant version of the ST-LINK/V2-1 USB driver by searching STSW-LINK008 or STSW-LINK009 on www.st.com (according to used the Microsoft Windows OS version).

The X-NUCLEO-PLM01A1 expansion board can be easily connected to the STM32 Nucleo board through the Arduino UNO R3 extension connector and can interface with the external STM32 microcontroller on the STM32 Nucleo via the I²C transport layer.

# 4 Revision history

**Table 1: Document revision history**

| Date | Version | Changes |
|---|---|---|
| 28-Jun-2017 | 1 | Initial release. |

## IMPORTANT NOTICE – PLEASE 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 acknowledgement.

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. 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.

© 2017 STMicroelectronics – All rights reserved