UM1950
User manual

Getting started with the X-CUBE-ANALOG1 multifunctional software expansion for STM32Cube

Introduction

This user manual describes how to get started with the X-CUBE-ANALOG1 expansion software for STM32Cube.

X-CUBE-ANALOG1 provides a complete middleware to build applications such as instrumentation amplifier, current sensing (TSZ124), LED driver (TSV734), photodiode/UV and window comparator (TSU104). It is easily portable across different MCU families, thanks to STM32Cube.

This package contains a sample application enabling access to the following data:

- output voltage in mV, when the board is configured in instrumentation amplifier configuration
- value of current consumed by the application in mA, when the board is configured in current sensing configuration
- value of GPIO pin PC1, when the board is configured in photodiode/UV sensor configuration
- sets duty cycle in percentage, when board is configured in LED driver configuration
- reads value of pin D2 and D4, when board is configured in window comparator mode

The software provides implementation examples for STM32 Nucleo platforms equipped with the X-NUCLEO-IKA01A1 expansion board, featuring instrumentation amplifier, current sensing, LED driver, photodiode/UV and window comparator operational amplifier devices.

The expansion software is based on STM32Cube technology and expands STM32Cube-based packages.

Contents

List of tables

List of figures

1 What is STM32Cube?

1.1 STM32Cube overview

STMCube™ represents an original by STMicroelectronics to ease developers' life by reducing development effort, time and cost. STM32Cube covers the STM32 portfolio.

Version 1.x of STM32Cube includes:

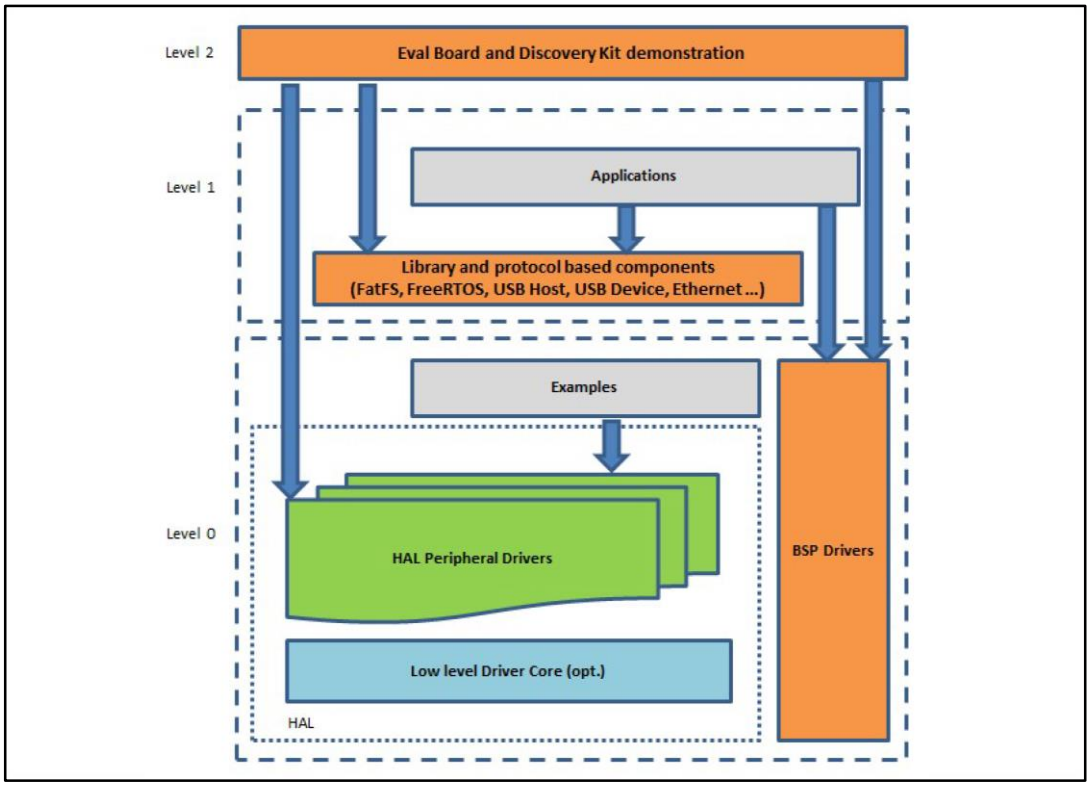
- The STM32CubeMX, a graphical software configuration tool that allows the generation of C initialization code using graphical wizards.
- A comprehensive embedded software platform, delivered per series (such as STM32CubeF4 for the STM32F4 series)
  - The STM32Cube HAL, an STM32 abstraction layer embedded software, ensuring maximized portability across the STM32 portfolio
  - A consistent set of middleware components such as RTOS, USB, TCP/IP, graphics
  - All embedded software utilities coming with a full set of examples

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

1.2 STM32Cube architecture

The STM32Cube firmware solution is built on three independent levels that can easily interact with each other, as illustrated 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.) and is composed of two parts:
  - Component: the driver relative to the external device on the board and not related to the STM32. The component driver provides specific APIs to the BSP driver external components and could be portable to any other board.
  - BSP driver: permits linking the component driver to a specific board and provides a set of user-friendly APIs. The API naming rule is BSP_FUNCT_Action(): ex. BSP_LED_Init(), BSP_LED_On().

It is based on a modular architecture, which allows easy portability 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 functionality-oriented APIs which permit the user to offload the application implementation by providing a ready-to-use process. For example, for the communication peripherals (I^2S, UART, etc.) it provides APIs which allow initialization and configuration of the peripheral, manages data transfer based on polling, interrupt or DMA process, and manages communication errors that may arise during communication. The HAL driver APIs are split into two categories, generic APIs which provide common and generic functions to all the STM32 series, and extension APIs which provide specific and customized functions for a specific family or a specific part number.
- Basic peripheral use examples: this layer encloses the examples build over the STM32 peripheral using only the HAL and BSP resources.

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 between the components in this layer is done directly by calling the feature APIs, while the vertical interaction with the low level drivers is done 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 the microSD drive or the USB mass storage class.
- Examples based on the middleware components: each middleware component comes with one or more examples (called also Applications) showing how to use it. Integration examples that use several middleware components are provided as well.

Level 2: This level is composed of a single layer which is a global real-time and graphical demonstration based on the middleware service layer, the low level abstraction layer and the basic peripheral use applications for board-based functionalities.

2 X-CUBE-ANALOG1 software expansion for STM32Cube

2.1 Overview

The X-CUBE-ANALOG1 is a software package that expands the functionality provided by STM32Cube.

The key features of the package are:

- Complete middleware to build applications such as instrumentation amplifier, current sensing (TSZ124), LED driver (TSV734), photodiode/UV and window comparator (TSU104)
- Easy portability across different MCU families, thanks to STM32Cube
- Transmit real-time voltage readings to a PC using serial communication
- Free, user-friendly license terms
- Example implementation available for X-NUCLEO-IKA01A1 expansion board plugged to a NUCLEO-F401RE, NUCLEO-F103RB, NUCLEO-L053R8 or NUCLEO-L476RG board

This software is used for reading and configuring various analog functions such as instrumentation amplifier, current sensing, LED driver, photodiode/UV and window comparator operational amplifier drivers for the TSZ124, TSV734 and TSU104 devices, running on a STM32 microcontroller.

The package also includes a sample application that the developer can use to start experimenting with the code. The sample application was developed to enable voltage reading data logging on a PC. The data logging can be viewed in a serial communication terminal on the PC.

2.2 Architecture

This software is an expansion for STM32Cube, and as such it fully complies with the architecture of STM32Cube and expands it in order to enable development of applications using operational amplifiers. Please see the previous chapter for an introduction to the STM32Cube architecture.

The software is based on the STM32CubeHAL, the hardware abstraction layer for the STM32 microcontroller. The package extends STM32Cube by providing a board support package (BSP) for the multifunctional expansion board and some middleware components for serial communication with a PC.

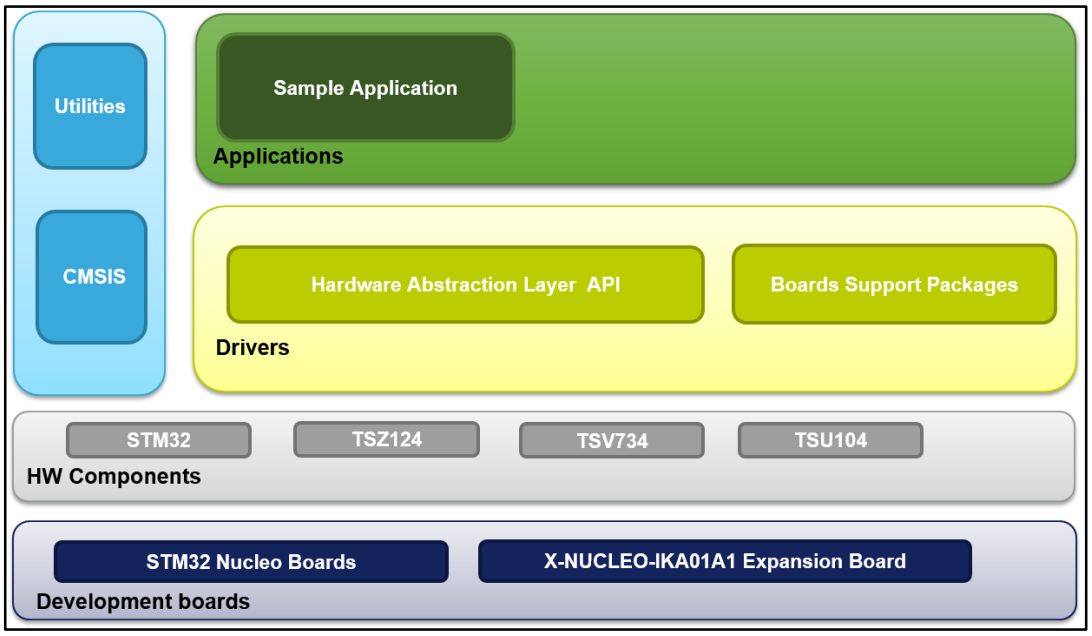
The software layers used by the application software to access and use the multifunctional expansion board are the following:

- STM32Cube HAL layer: The HAL driver 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 dependency on the specific hardware configuration for a given microcontroller unit (MCU). This structure improves library code reusability and guarantees easy portability to other devices.
- Board support package (BSP) layer: The software package needs to support the peripherals on the STM32 Nucleo board apart from the MCU. This software is

included in the BSP. This is a limited set of APIs which provides a programming interface for certain board specific peripherals, e.g. the LED, the user button, etc. This interface also helps in identifying the specific board version. When using the multifunctional board, it provides the programming interface for various operational amplifiers. It provides support for initializing and reading operational amplifier data.

The diagram below outlines the software architecture of the package:

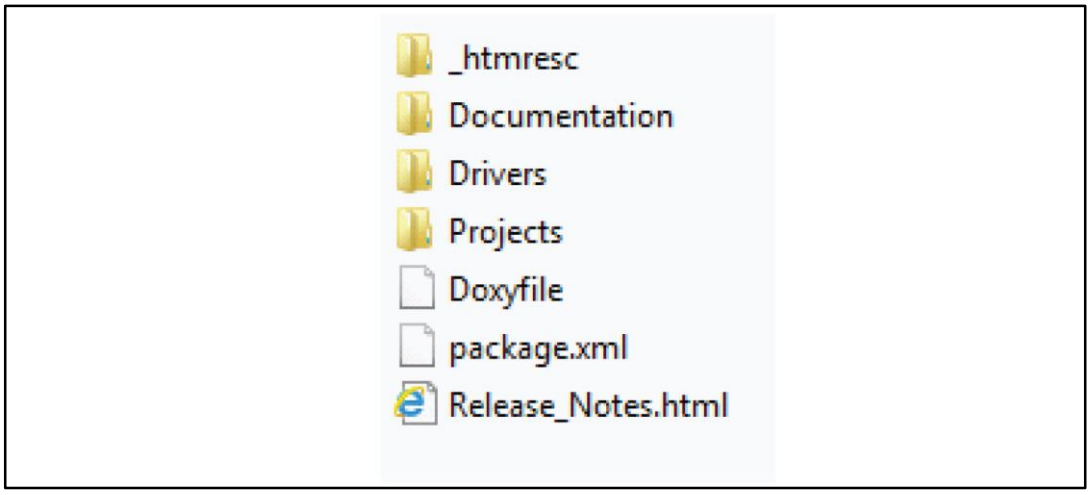
Figure 2: X-CUBE-ANALOG1 software architecture



2.3 Folder structure

This section provides an overview of the package folder structure.

Figure 3: X-CUBE-ANALOG1 package folder structure



The following folders are included in the software package:

- **_htmresc**: contains image resources used by Release_Notes.html.
- **Documentation**: contains a compiled HTML file generated from the source code and documentation that details the software components and APIs.

- **Drivers**: 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**: contains a sample application used to access various configurations of X-CUBE-ANALOG1 software for the NUCLEO-L053R8, NUCLEO-F103RB and NUCLEO-F401RE platforms.

2.4 APIs

Detailed technical information regarding the APIs can be found in a compiled HTML file named "X-CUBE-ANALOG1.chm" and located inside the "Documentation" folder of the software package where all the functions and parameters are fully described.

2.5 Sample application description

An example application using the X-NUCLEO-IKA01A1 expansion board with a NUCLEO-F401RE, NUCLEO-F103RB, NUCLEO-L053R8 or NUCLEO-L476RG board is provided in the "Projects" directory. Ready-to-build projects are available for multiple IDEs.

This application transmits meaningful data in various configurations of the X-NUCLEO-IKA01A1 expansion board.

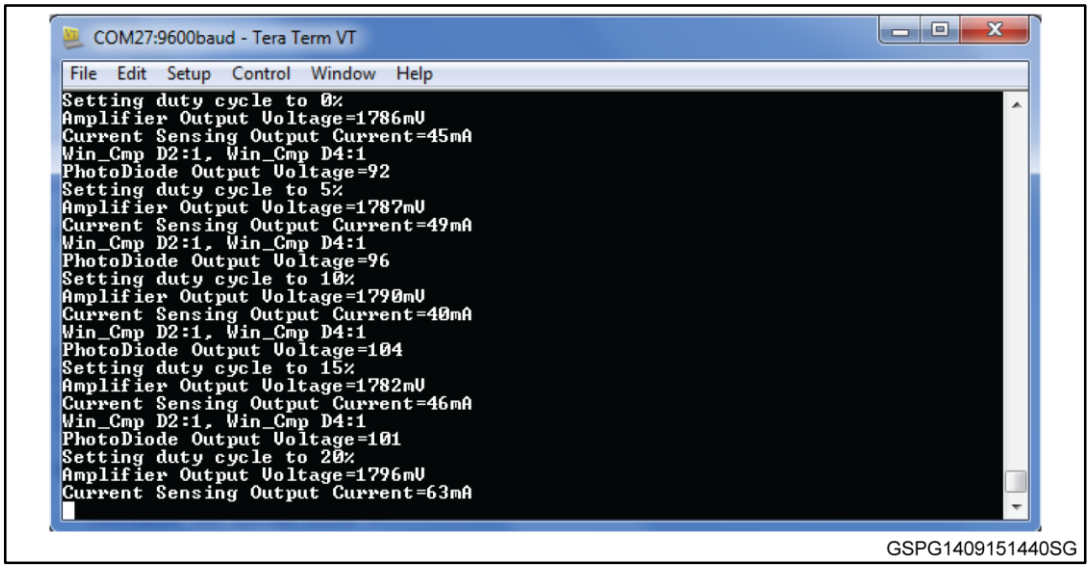
This data is:

1. Amplified differential voltage (in mV) when the X-NUCLEO-IKA01A1 expansion board is configured in instrumentation amplifier configuration
2. Current (in mA) consumed by an application when the X-NUCLEO-IKA01A1 expansion board is configured in current sensing configuration
3. Value of GPIO pin PC1 when the X-NUCLEO-IKA01A1 expansion board is configured in photodiode/UV sensor configuration
4. Values of GPIO pin D2 and D4 when the X-NUCLEO-IKA01A1 expansion board is configured in photodiode/UV sensor configuration

In LED driver configuration of the X-NUCLEO-IKA01A1 expansion board, this application varies light intensity of LED D2 by varying the duty cycle from 0 to 100 in steps of 5.

This data can be viewed on a PC by using a terminal emulation program like Tera Term. Please see *Section 3.3.4: "Tera Term"* for information on the proper configuration of Tera Term. The following snapshot shows the output screen of Tera Term when the example application is running on an NUCLEO-F401RE board equipped with a X-NUCLEO-IKA01A1 expansion board.

Figure 4: Tera Term window screenshot



3 System setup guide

3.1 Hardware description

This section describes the hardware components needed for developing an operational amplifier-based application. The following subsections describe the individual components.

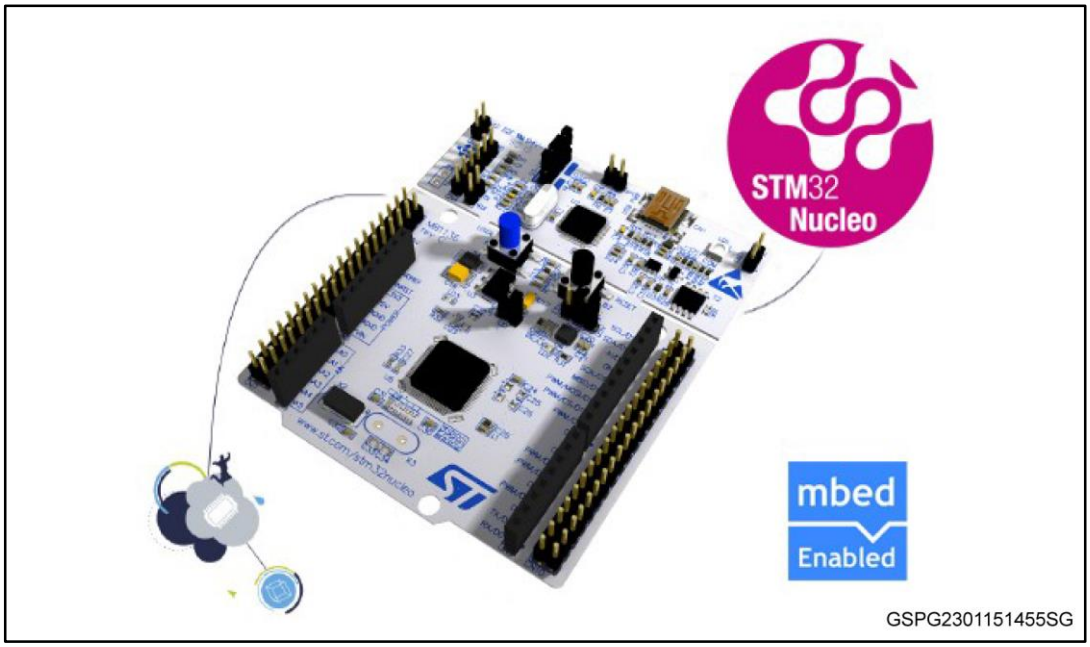
3.1.1 STM32 Nucleo platform

The STM32 Nucleo boards provide an affordable and flexible way for users to try out new ideas and build prototypes with any of the 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 a separate probe, 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 at:

http://www.st.com/stm32nucleo

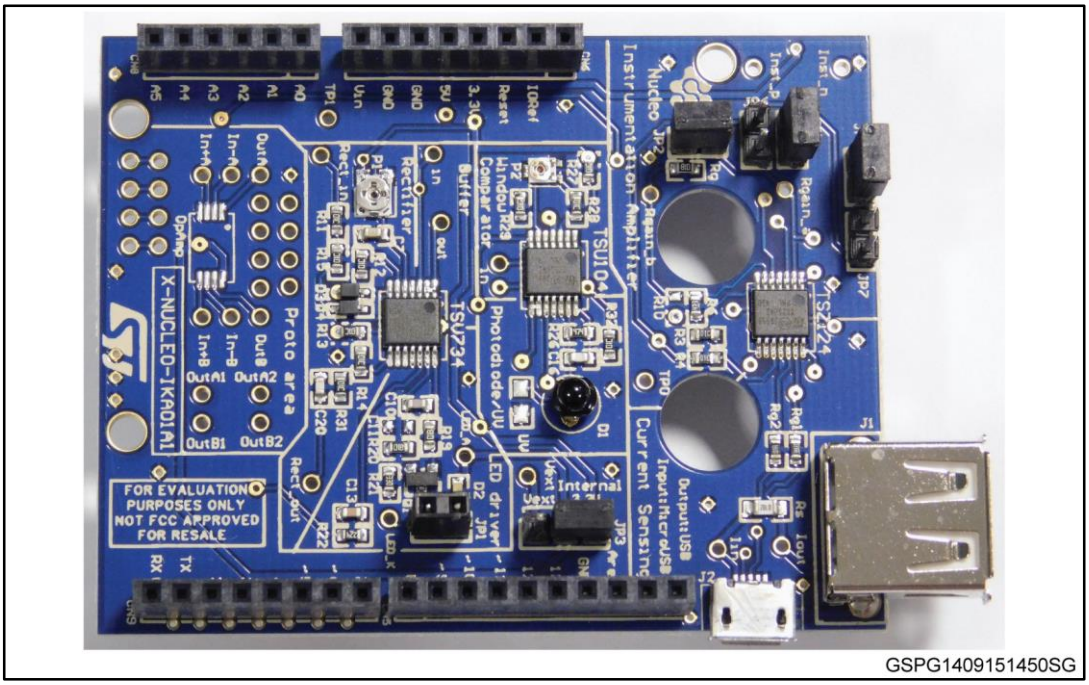
Figure 5: STM32 Nucleo board



3.1.2 X-NUCLEO-IKA01A1 expansion board

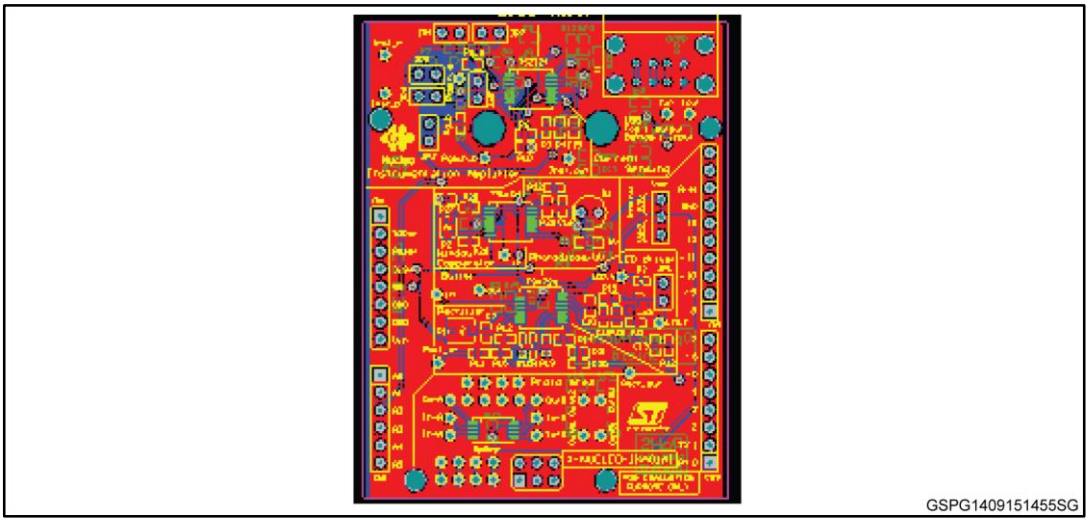
The X-NUCLEO-IKA01A1 is a multifunctional expansion board based on operational amplifiers. It provides an affordable and easy-to-use solution for different use cases with STM32 Nucleo or Arduino projects. It can be used as analog front-end by conditioning signals as actuator to drive LEDs, or used in a comparator configuration. It also supports a current sensing configuration, to allow current measurement of any device which has a Micro-USB port.

Figure 6: X-NUCLEO-IKA01A1 multifunctional expansion board



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

Figure 7: X-NUCLEO-IKA01A1 multifunctional expansion board silkscreen



3.2 Software description

The following software components are needed in order to setup the suitable development environment for creating applications for the STM32 Nucleo equipped with the X-NUCLEO-IKA01A1 expansion board:

- X-CUBE-ANALOG1: an expansion for STM32Cube dedicated to X-NUCLEO-IKA01A1 applications development. The X-CUBE-ANALOG1 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) toolchain + ST-LINK
- AC6 System Workbench for STM32 + ST-LINK

3.3 Hardware and software setup

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

3.3.1 Hardware setup

The following hardware components are required:

1. One STM32 Nucleo Development platform (order code: NUCLEO-F401RE, NUCLEO-F103RB, NUCLEO-L053R8 or NUCLEO-L476RG)
2. One multifunctional expansion board (order code: X-NUCLEO-IKA01A1)
3. One USB type A to Mini-B USB cable to connect the STM32 Nucleo to the PC
4. One USB type A to Micro-B USB extension cable (for current sensing configuration)

3.3.2 Software setup

This section lists the minimum requirements for the developer to set up the SDK, run the sample testing scenario and customize applications.

3.3.3 Development toolchains and compilers

Please select one of the integrated development environments (IDEs) supported by the STM32Cube expansion software.

Please also read the system requirements and setup information provided by the selected IDE provider.

3.3.4 Tera Term

The PC must have Tera Term or some other serial communication terminal emulator program to visualize UART data transmitted by the sample application running on NUCLEO-F401RE, NUCLEO-F103RB, NUCLEO-L053R8 or NUCLEO-L476RG boards with X-NUCLEO-IKA01A1 expansion boards.

3.3.5 System setup guide

This section describes how to set up the different hardware components before writing and executing an application on the STM32 Nucleo board with the multifunctional expansion board.

3.3.6 STM32 Nucleo and sensor expansion boards 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 looking up STSW-LINK008 or STSW-LINK009 on www.st.com (based on the version of Microsoft® Windows® OS).

The multifuctional expansion board X-NUCLEO-IKA01A1 can be easily connected to the STM32 Nucleo board through the Arduino UNO R3 extension connector, see *Figure 7: "X-NUCLEO-IKA01A1 multifunctional expansion board silkscreen"*.

3.3.7 Tera Term setup

The Tera Term terminal emulation program can be used to view the streaming serial data transmitted by the sample application program running on the STM32 Nucleo board equipped with the X-NUCLEO-IKA01A1 expansion board.

The following screenshots show the Tera Term serial port and terminal setup configurations to be used for viewing the serial UART data transmitted by the sample application.

Figure 8: Tera Term serial port setup configuration

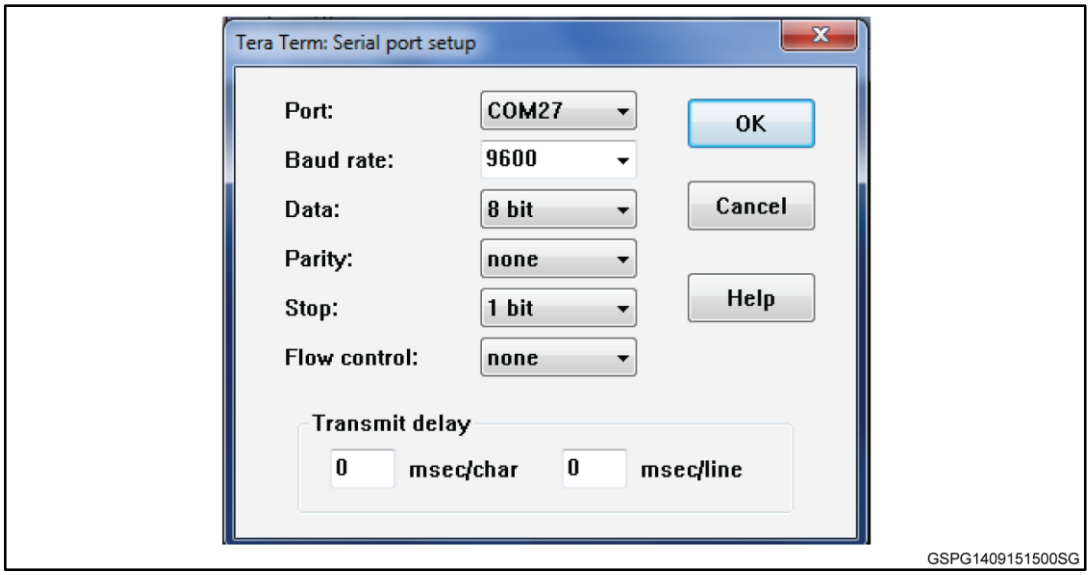
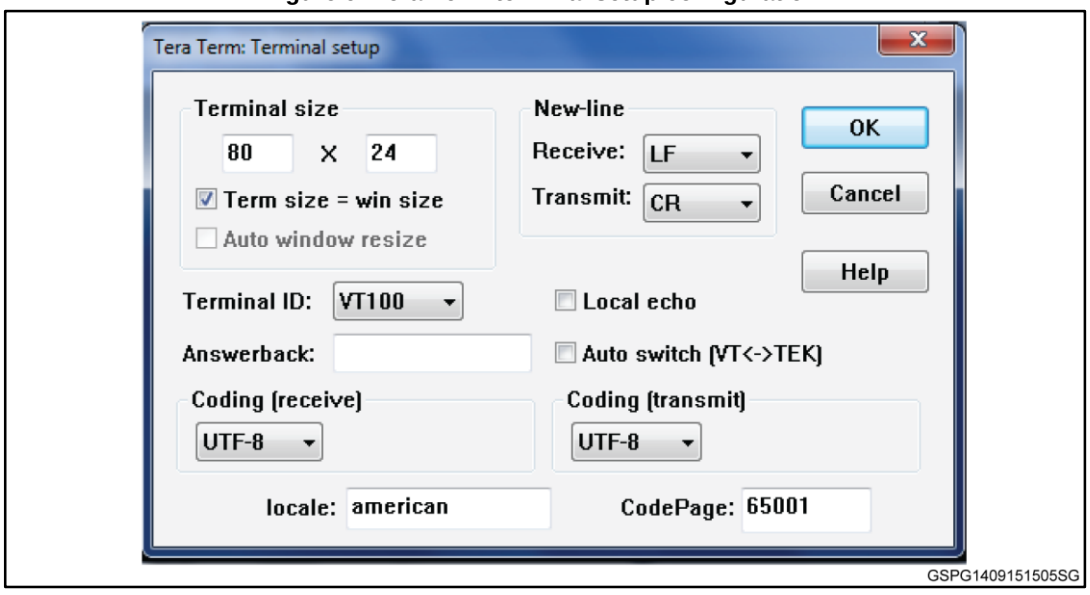


Figure 9: Tera Term terminal setup configuration



4 Acronyms and abbreviations

Table 1: Acronyms

Acronym	Description
BSP	Board support package
HAL	Hardware abstraction layer
I²C	Inter integrated circuit
MCU	Microcontroller unit
UART	Universal asynchronous receiver transmitter

5 Revision history

Table 2: Document revision history

Date	Revision	Changes
02-Oct-2015	1	Initial release.
23-May-2016	2	Minor text edits Updated *Section 2.1: "Overview"* Updated *Section 2.5: "Sample application description"* Updated *Section 3.3.1: "Hardware setup"* Updated *Section 3.3.4: "Tera Term"*

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.