UM1998
User manual

Getting started with X-CUBE-LED1 DC-DC LED driver software expansion for STM32Cube

This document describes how to get started with the X-CUBE-LED1 software expansion for STM32Cube.

X-CUBE-LED1 provides the STM32 middleware required to build applications using X-NUCLEO-LED61A1 expansion board for STM32 Nucleo. It is easily ported across different MCU families thanks to STM32Cube. The software provides implementation examples for STM32 Nucleo platforms equipped with the X-NUCLEO-LED61A1 expansion board used to drive the string of several LEDs using LED6001 LED driver.

The software is based on STM32Cube technology and expands the range of STM32Cube-based packages, available at www.st.com (*http://www.st.com/stm32cube*).

Contents

List of figures

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.

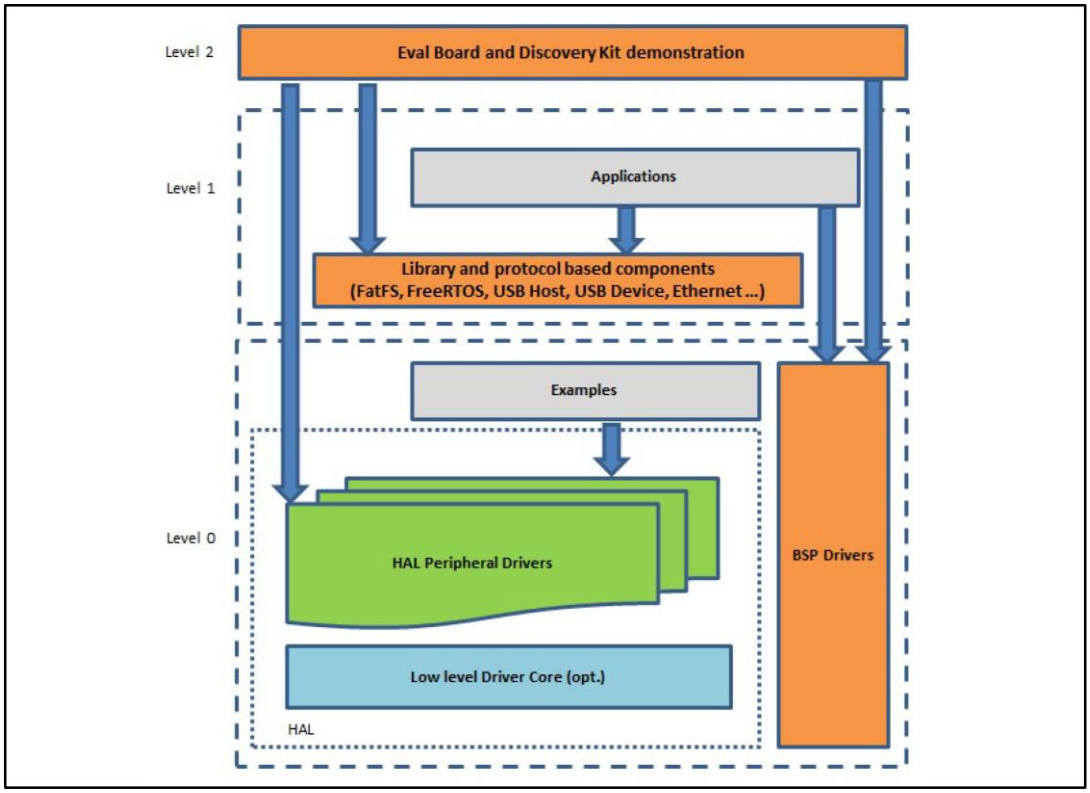
STM32Cube version 1.x includes:

- 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…) and 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().

It is based on modular architecture allowing is to be easily ported on 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 to help offload user application development time by providing ready to use processes. For example, for the communication peripherals (I2S, 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-LED1 software expansion for STM32Cube

2.1 Overview

The X-CUBE-LED1 software package expands the functionality provided by STM32Cube.

The key features of the package are:

- A complete middleware to build applications involving several LEDs arranged in a single string (e.g. off-grid street lighting, advertisement panels, signs, gaming, etc.) based on LED driver (LED6001) and hardware expansion board (X-NUCLEO-LED61A1)
- High portability across different MCU families, thanks to STM32Cube
- Free user-friendly license terms
- Example implementations available for the X-NUCLEO-LED61A1 board plugged on top a NUCLEO-F401RE or NUCLEO-L053R8 STM32 Nucleo board.

2.2 Architecture

This software is a fully compliant STM32Cube expansion for application development involving DC-DC LED driver devices. The STM32Cube architecture is described above.

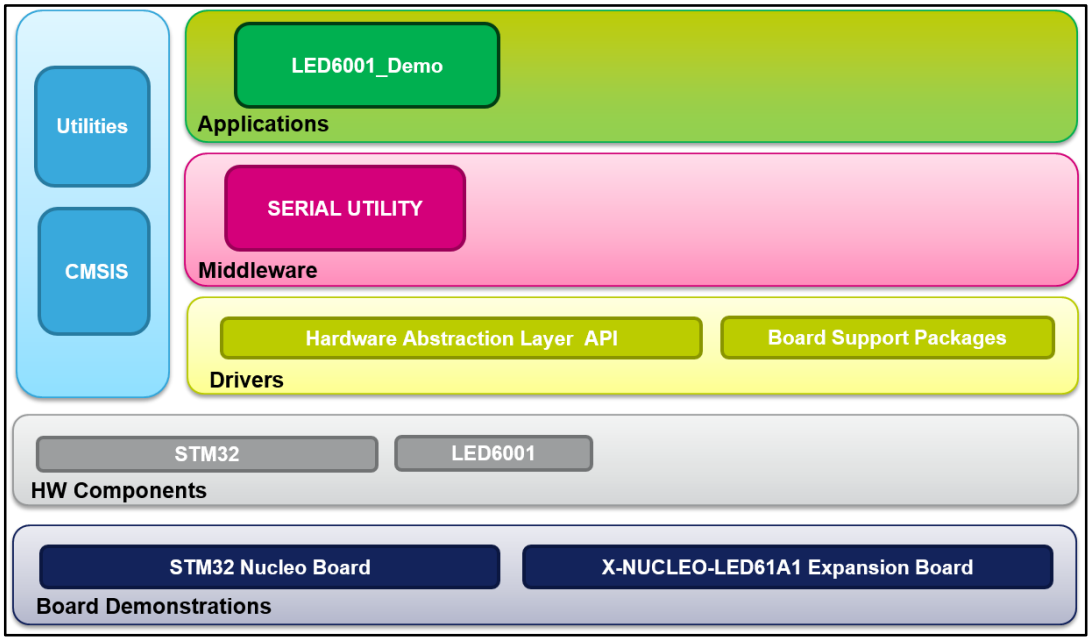
The software is based on the STM32CubeHAL hardware abstraction layer for the STM32 microcontroller. The package extends STM32Cube by providing a Board Support Package (BSP) for the X-NUCLEO-LED61A1 expansion board and a sample application for serial communication with a PC.

The software layers used by the application software to access the X-NUCLEO-LED61A1 expansion board are:

- **STM32Cube HAL layer**: consists of a set of simple, generic, 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 which allows any layers they built on, such as the middleware layer, to implement their functions without requiring specific hardware information for a given microcontroller unit (MCU). 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.

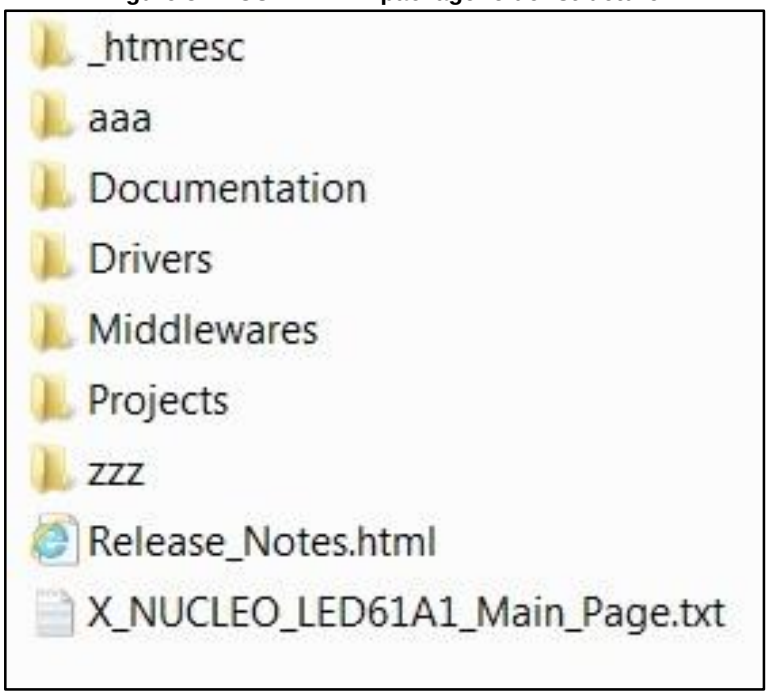
The figure below outlines the software architecture of the package:

Figure 2: X-CUBE-LED1 architecture



2.3 Folders structure

Figure 3: X-CUBE-LED1 package folder structure



The following folders are included in the software package:

- The **Documentation** folder contains a compiled HTML file generated from the source code and detailed documentation regarding the software components and APIs.
- The **Drivers** folder contains the HAL drivers, the board-specific drivers for each supported board or hardware platform, including those for the on-board components

and the CMSIS vendor-independent hardware abstraction layer for the Cortex -M processor series.

- The **Projects** folder contains a sample application for the NUCLEO-L053R8 and NUCLEO-F401RE platforms; it is provided with three development environments (IAR Embedded Workbench for ARM, RealView MDK-ARM Microcontroller Development Kit and System Workbench for STM32).

2.4 APIs

Detailed technical information about the APIs available to the user can be found in the compiled HTML file "X-CUBE-LED1.chm" in the "Documentation" folder of the software package, where all the functions and parameters are fully described.

2.5 Sample application description

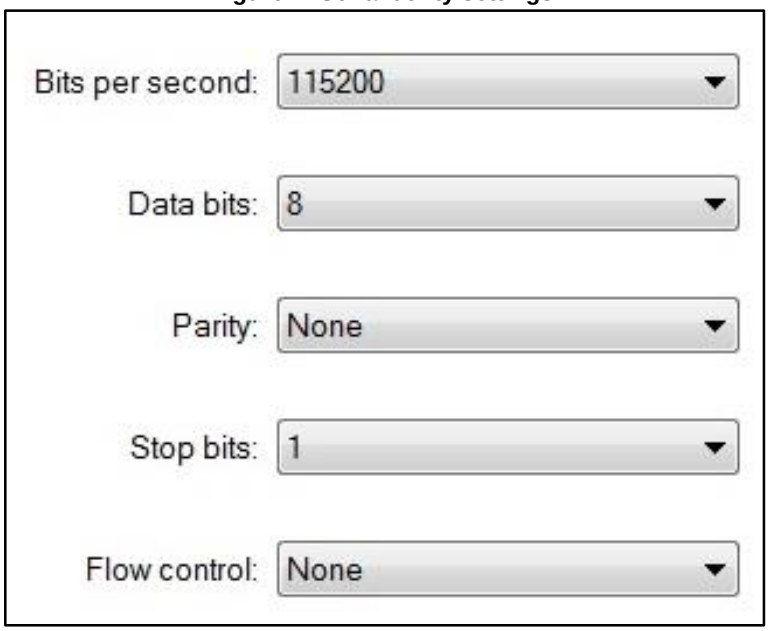
An example application using the X-NUCLEO-LED61A1 expansion board with either NUCLEO-F401RE or the NUCLEO-L053R8 STM32 Nucleo board is provided in the “Projects” directory. Ready to use projects are provided for multiple IDEs.

This application operates a LED driver expansion board, implementing a state machine to demonstrate various demos, including PWM and analog dimming. The state machine input parameters are a user button (short push and long push) and photo sensor values using ADC. Depending on the demo selected, PWM and analog dimming values are varied.

For fault events (XFAULT goes low), PWM is stopped and the LED driver restarts after a brief pause.

All the relevant demo runtime information can be viewed on a PC with a standard serial utility like HyperTerminal.

Figure 4: Serial utility settings



2.6 Preconfigured demos

X-CUBE-LED1 supports following demo routines running on an X-NUCLEO-LED61A1 board:

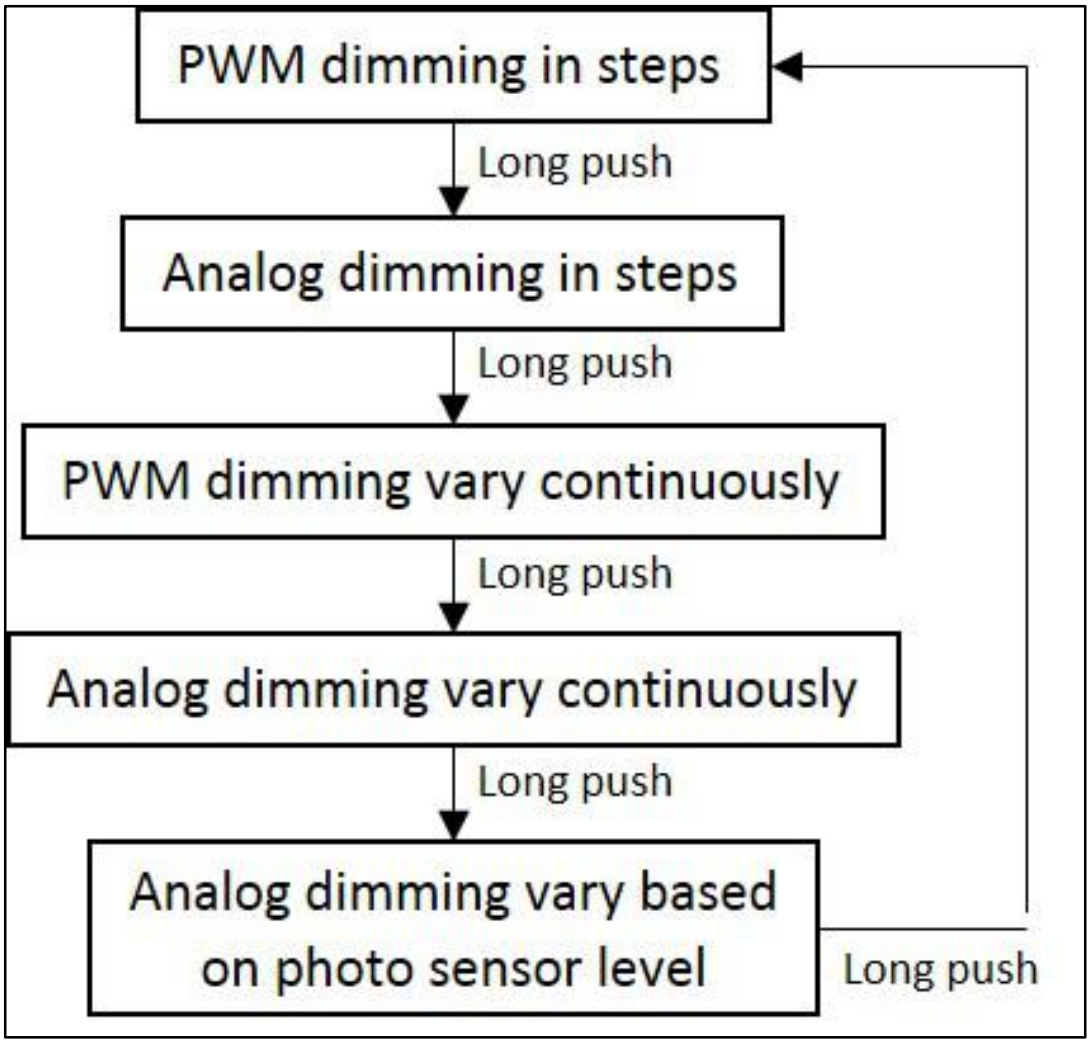
1. PWM dimming in step variation by briefly pressing the blue user button
2. Analog dimming in step variation by briefly pressing the blue user button
3. PWM dimming continuous variation
4. Analog dimming continuous variation
5. Analog dimming variation based on photo sensor level

Each demo can be selected by pressing the blue user button on the STM32 Nucleo board for more than 1 second (considered a “long” push)

By default, the “PWM dimming in steps” demo is selected.

You can scroll through the following sequence of demos with “long” pushes of the user button:

Figure 5: Demo sequence



3 System setup guide

3.1 Hardware description

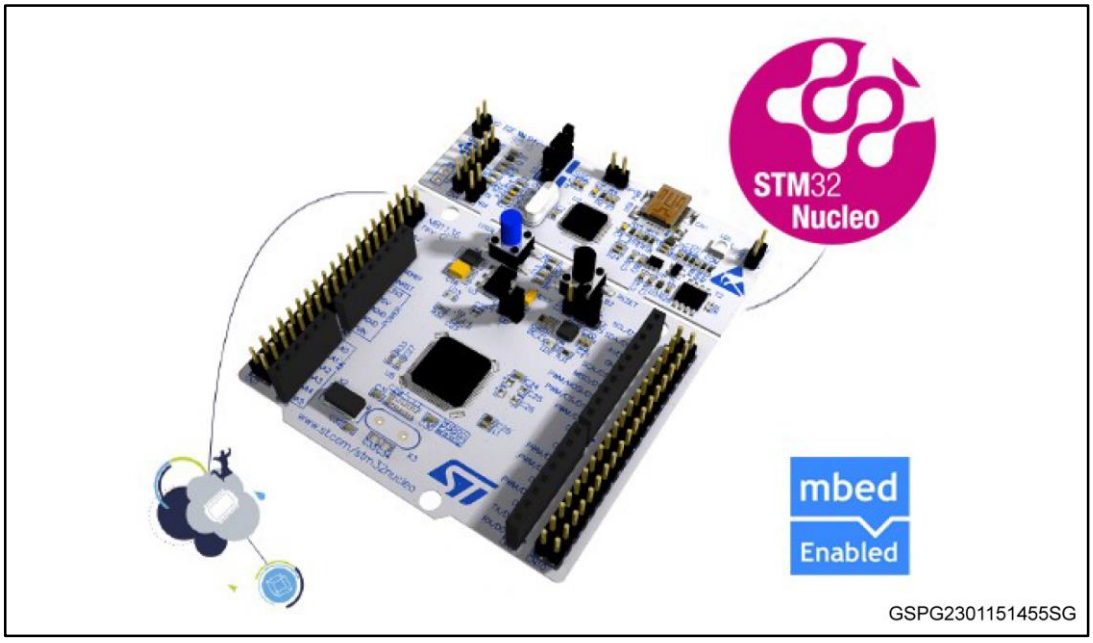
This section describes the hardware components needed for developing an LED driver application.

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 range of specialized expansion boards to choose from. The STM32 Nucleo board does not require any separate probe 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.

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

Figure 6: STM32 Nucleo board

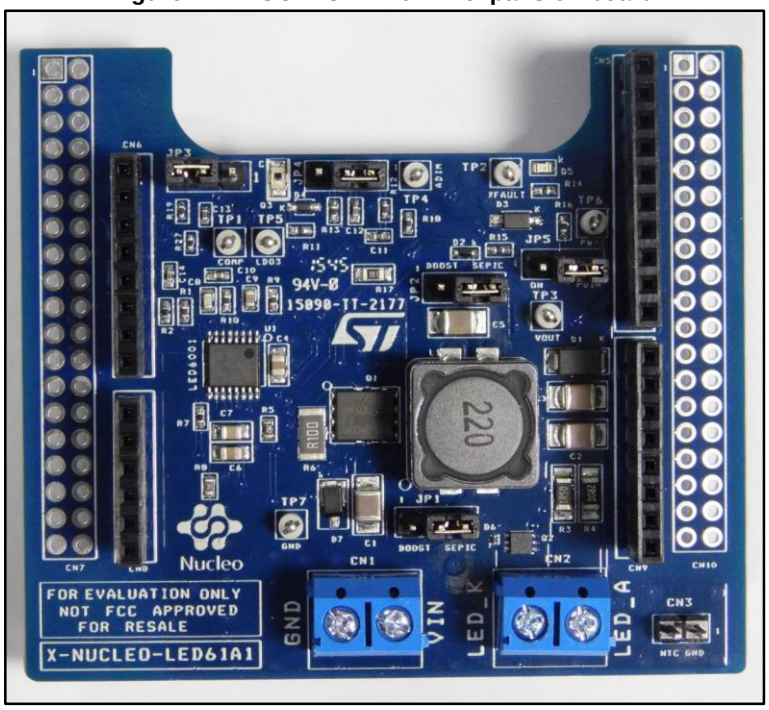


3.1.2 X-NUCLEO-LED61A1 expansion board

The X-NUCLEO-LED61A1 is a DC-DC LED driver expansion board based on LED6001 for STM32 Nucleo. It can be used independently or with the STM32 Nucleo board; it is also compatible with the Arduino UNO R3 platform.

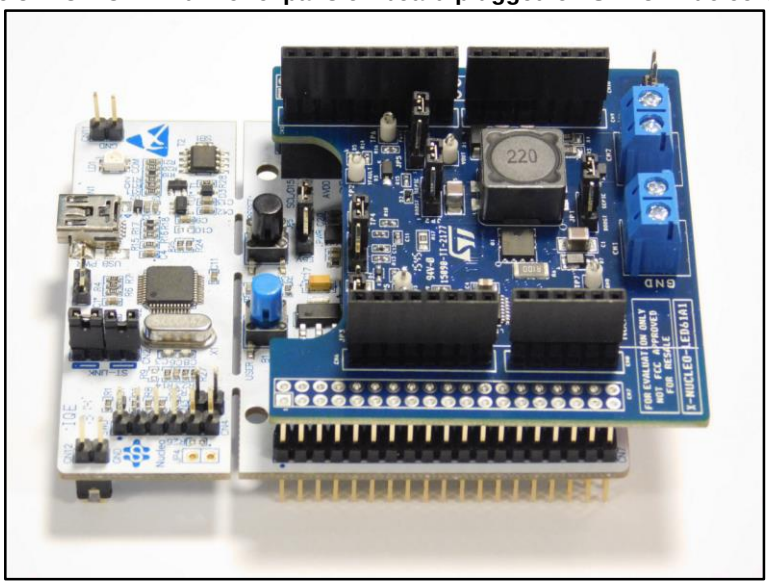
It is designed to provide a sample application for compact LED drivers using the new LED6001 device. The X-NUCLEO-LED61A1 interfaces with the STM32 MCU via the Timer and ADC pins, and the user can change the PWM levels for different dimming actions.

Figure 7: X-NUCLEO-LED61A1 expansion board



Information regarding the X-NUCLEO-LED61A1 expansion board is available on www.st.com at

Figure 8: DC-DC LED driver expansion board plugged on STM32 Nucleo board



3.2 Software description

The following software components are required in order to establish a suitable development environment for creating applications for the STM32 Nucleo equipped with the X-NUCLEO-LED61A1 expansion board:

- X-CUBE-LED1: an STM32Cube expansion for LED driving application development. The X-CUBE-LED1 firmware and associated documentation is available on

- Development tool-chain and compiler: The STM32Cube expansion software supports three following environments:
  - IAR Embedded Workbench for ARM® (EWARM) toolchain + ST-LINK
  - RealView Microcontroller Development Kit (MDK-ARM) toolchain + ST-LINK
  - 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

Set up the hardware with the following components and accessories:

- One STM32 Nucleo Development platform (suggested order code: NUCLEO-F401RE or NUCLEO-L053R8)
- One DC-DC LED driver expansion board (order code: X-NUCLEO-LED61A1)
- One USB type A to Mini-B USB cable to connect the STM32 Nucleo to the PC
- One external power supply, 8-24 V / 2 A
- LED load, varying voltage/ 350 mA

3.3.2 Software setup

This section lists the minimum requirements for the developer to setup the SDK, run the sample testing scenario based on the GUI utility and customize applications.

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 same IDE provider.

PC utility

The PC should have the following minimum requirements:

- PC with Intel or AMD processor running one of following Microsoft® operating systems:
  - Windows XP SP3
  - Windows Vista
  - Windows 7
- At least 128 MB of RAM
- 2 x USB ports
- 40 MB of hard disk space

3.4 System setup guide

This section describes how to setup different hardware components before writing and executing an application on the STM32 Nucleo board with the X-NUCLEO-LED61A1 expansion board.

3.4.1 STM32 Nucleo and hardware expansion boards setup

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

The X-NUCLEO-LED61A1 expansion board can be easily connected to the STM32 Nucleo motherboard through the Arduino UNO R3 extension connector and can interface with the external STM32 microcontroller on STM32 Nucleo via the Timer and ADC pins.

4 Revision history

Table 1: Document revision history

Date	Revision	Changes
10-Dec-2015	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.