Getting started with the STMicroelectronics X-CUBE-ALS software package for STM32CubeMX

What is STM32Cube?

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

Version 1.x of STM32Cube includes:

- 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 the STM32CubeF4 for STM32F4 series).
  - 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, including a full set of examples.

1 Acronyms and abbreviations

Acronym/abbreviation	Definition
ALS	ambient light sensing
CCT	correlated color temperature
HAL	hardware abstraction layer
I2C	inter integrated circuit
PCB	printed circuit board
TCP	transmission control protocol
U(S)ART	universal (synchronous) asynchronous receiver transmitter
USB	universal serial BUS

2 License

The software provided in this package is licensed under SLA0081 for X-CUBE-ALS.

3 Sample applications description

This section provides a short overview of the sample applications included in the X-CUBE-ALS pack.The sample applications are ready-to-use projects that can be generated through the STM32CubeMX for any Nucleo board and using the X-NUCLEO-6283A1 expansion board.

3.1 6283A1_ALSValues

This application shows how to use a X-NUCLEO-6283A1 expansion board to send sensor data from an STM32 Nucleo board using UART to a connected PC and display it on generic applications like Tera Term. Once the connection is established, the user can view the data from the on-board light sensor using a hyper terminal. The application provides real-time values and allows the user to change the gain, exposure time, and sample rate through the keyboard. The application serial settings can be configured by changing the settings of USART2 in the STM32CubeMX GUI.

3.2 6283A1_LuxCCT

This application shows how to use a X-NUCLEO-6283A1 expansion board to compute CCT and Lux measurement from an STM32 Nucleo board using UART to a connected PC and display it on generic applications like Tera Term. Once the connection is established, the user can view the CCT and Lux values from the on-board light sensor using a hyper terminal. The application provides real-time values and allows the user to change exposure time, and sample rate through the keyboard. The application serial settings can be configured by changing the settings of USART2 in the STM32CubeMX GUI.

3.3 6283A1_AnalogFlicker

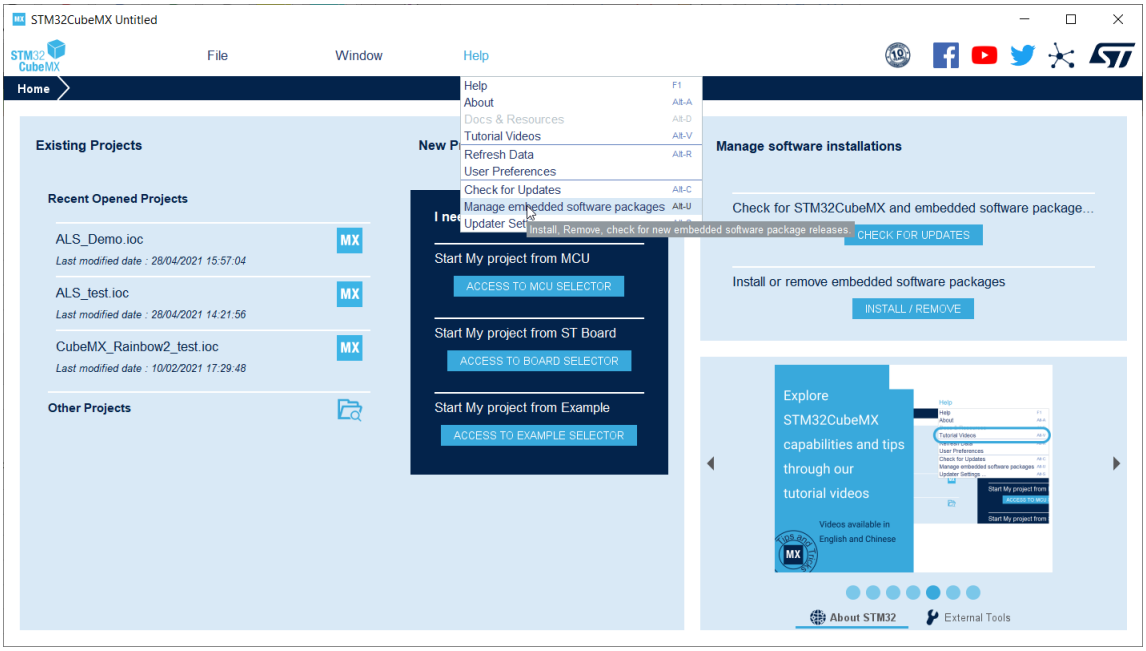
This application shows how to use a X-NUCLEO-6283A1 expansion board to compute flicker frequency extraction from an STM32 Nucleo board using UART to a connected PC and display it on generic applications like Tera Term. Once the connection is established, the user can view the data from the on-board light sensor using a hyper terminal. The application provides real-time values and allows the user to change the gain, time-exposure, and sample rate through the keyboard. This example can be accessed only in the pack installation folder.

4 Installing the X-CUBE-ALS pack in STM32CubeMX

Once the STM32CubeMX (version 6.2.0 or after) has been downloaded (from www.st.com), installed, and launched, the X-CUBE-ALS pack can be installed using the following steps.

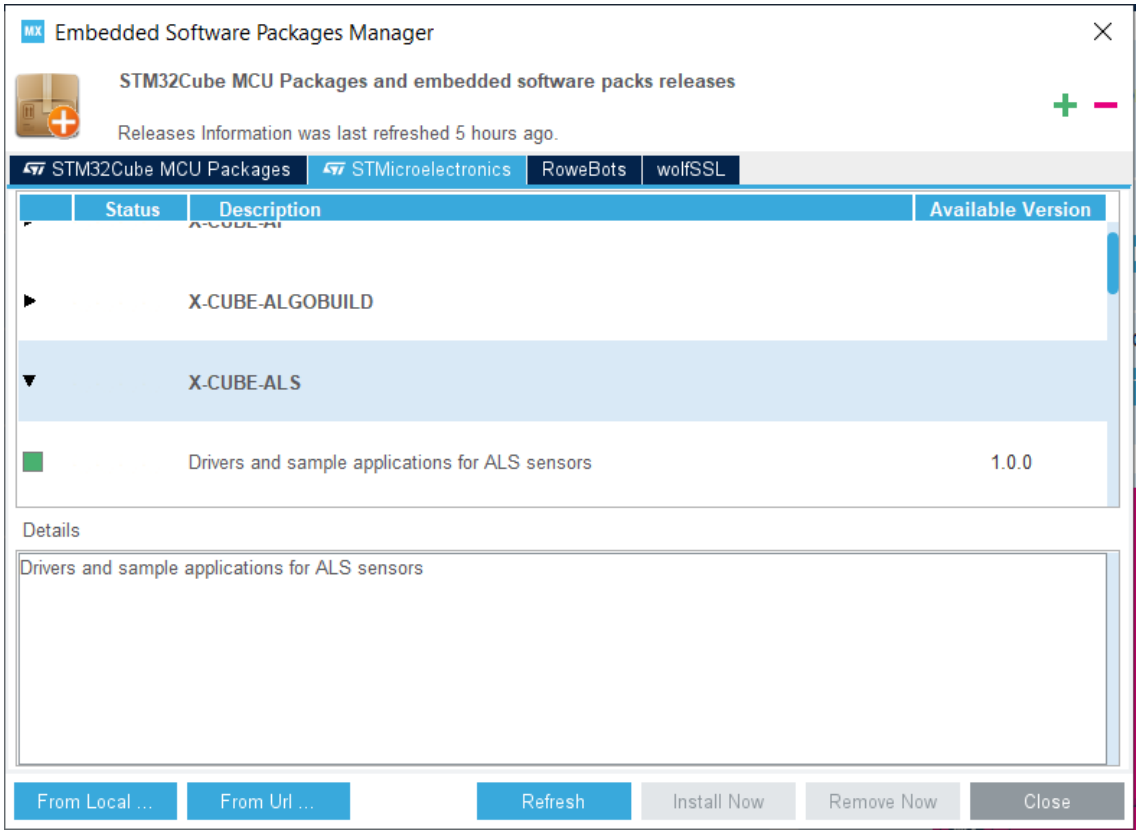
1. From the menu, select Help > Manage embedded software packages

Figure 1. Managing embedded software packs in STM32CubeMX



2. On the Embedded Software Packages Manager window, click Refresh to get an updated list of the add-on packs. Go to the STMicroelectronics tab to find the X-CUBE-ALS pack.

Figure 2. Installing the X-CUBE-ALS pack in STM32CubeMX

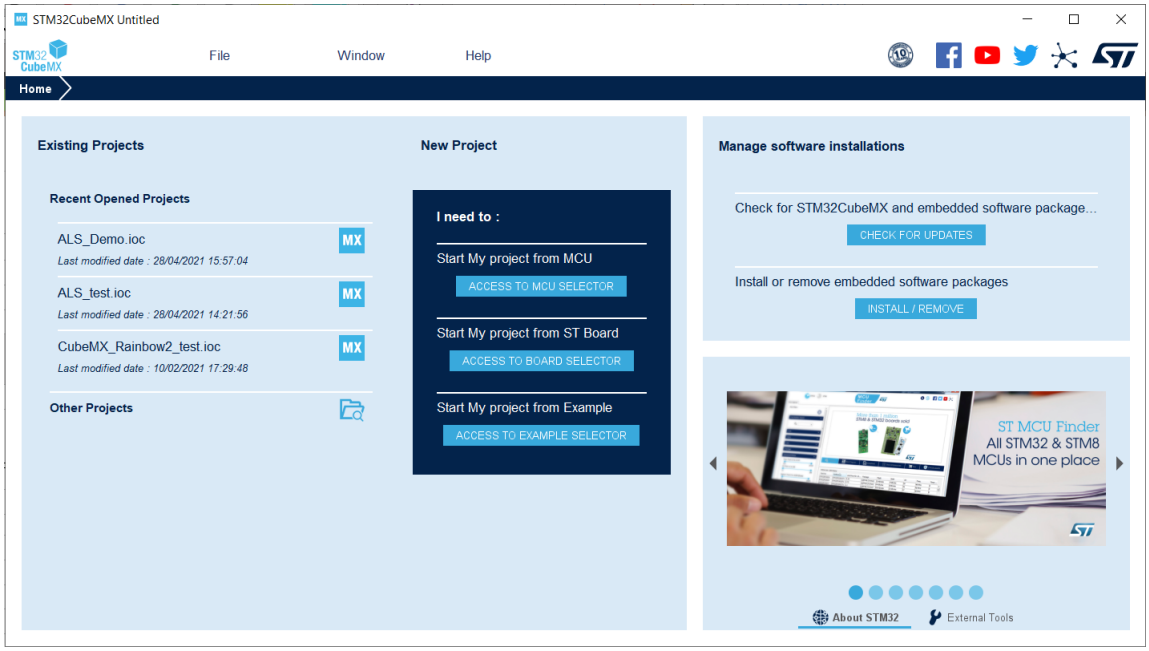


3. Select the X-CUBE-ALS pack by checking the corresponding box, and install it by clicking Install Now. Once the installation is complete the corresponding box becomes green.
4. Click Close. The configuration of a new project can be started.

5 Starting a new project

1. Launch the STM32CubeMX.
2. Choose if you want to start a new project from the MCU Selector, or from the Board Selector.

Figure 3. STM32CubeMX main page



3. The MCU/Board selector window will pop up. From this window, the STM32 MCU or platform can be selected.

Figure 4. STM32CubeMX MCU selector windows

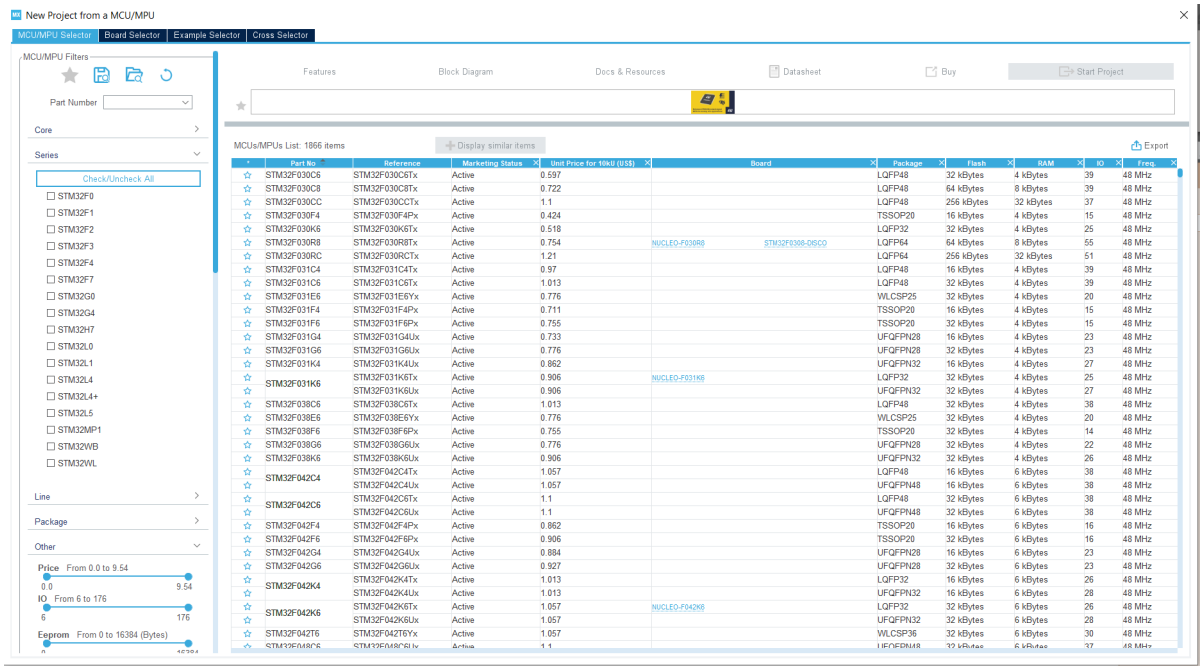
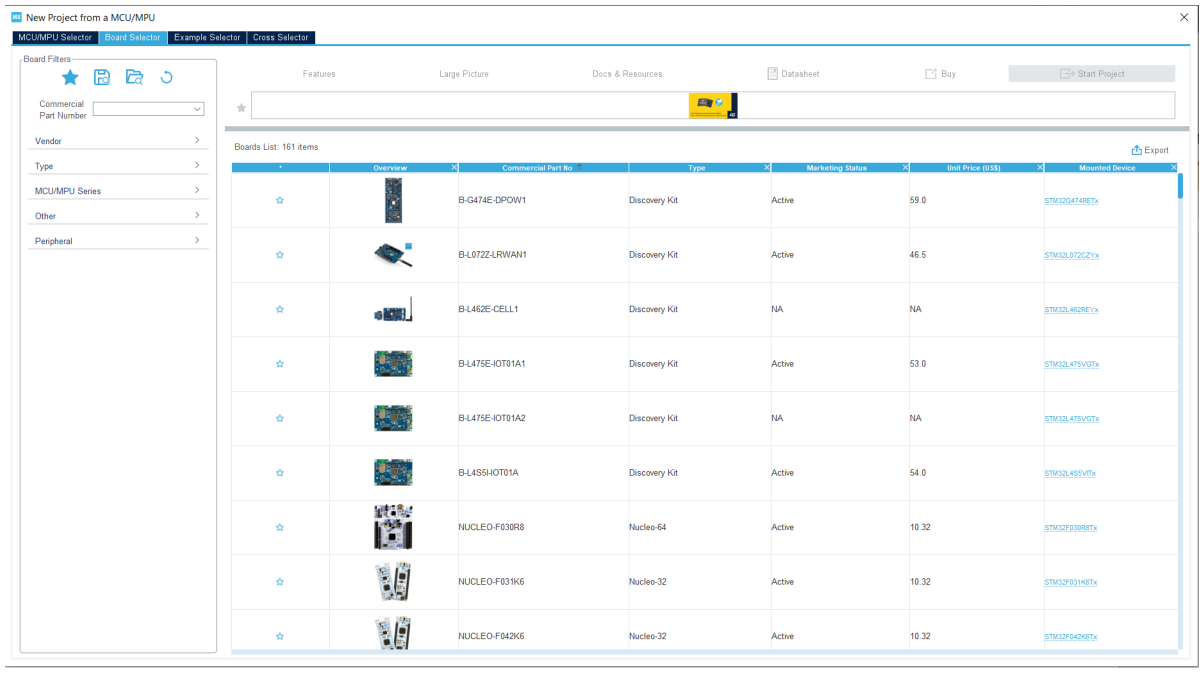
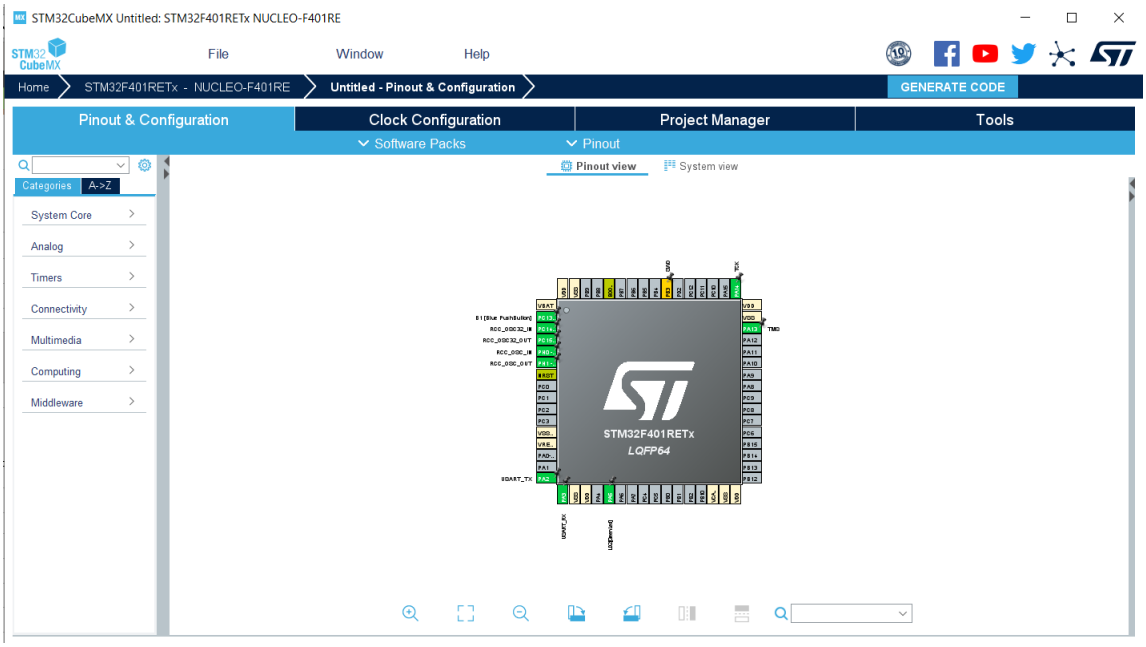


Figure 5. STM32CubeMX Board selector windows



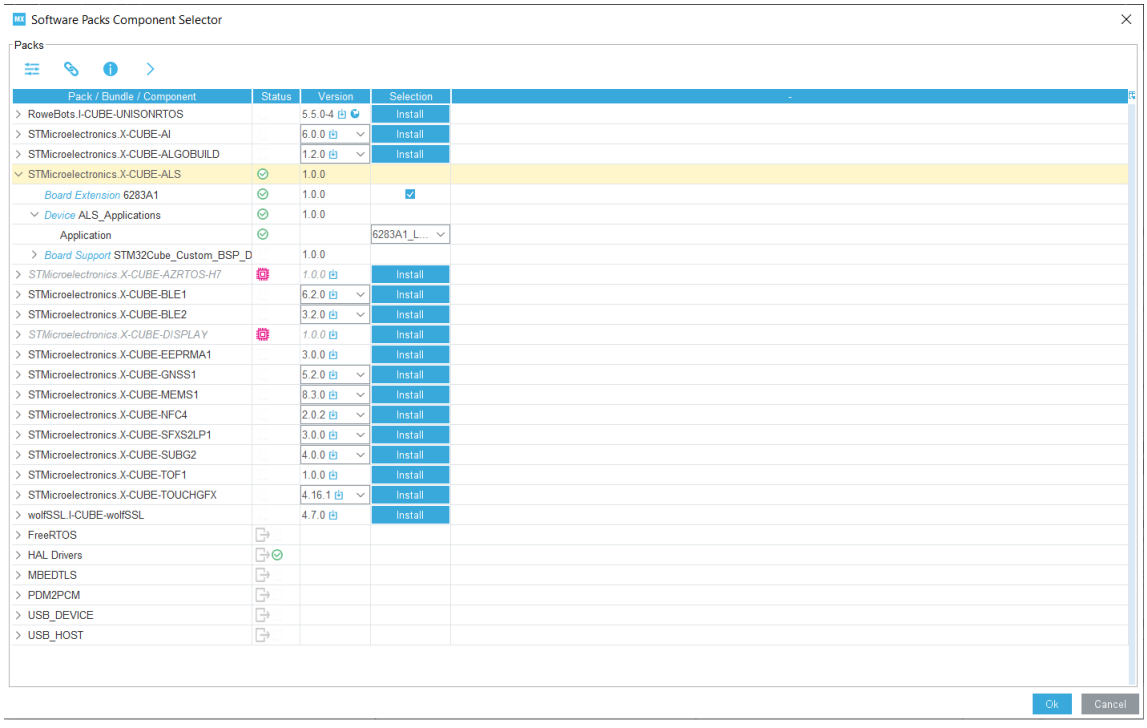
4. After selecting the MCU or the Board, the selected STM32 pinout will appear. From this window the user can set up the project, by adding one or more Software Packs and peripherals and configuring the clock.

Figure 6. STM32CubeMX Pinout & Configuration window



5. To add the X-CUBE-ALS software pack to the project:
   – Click Software Packs and select the Select Components item.
   – From the Software Packs Component Selector window, the user can either choose to generate (for the selected MCU/Board) one of the enclosed sample applications, or a new project. In the latter case, the user must simply implement the main application logic. The pinout and peripherals configuration code are automatically generated by STM32CubeMX.

Figure 7. STM32CubeMX Software Packs Component Selector



6 STM32 configuration steps

The X-NUCLEO-6283A1 interfaces with the STM32 microcontroller via the I2C bus. If a user wants to interface the X-NUCLEO-6283A1 expansion board with an STM32 Nucleo 64 pin board (e.g. a Nucleo-F401RETx) or an STM32 Nucleo 144 pin board (e.g. Nucleo F429ZITx), no hardware modification is needed.

Figure 8. STM32 Nucleo 64 pin and X-NUCLEO-6283A1

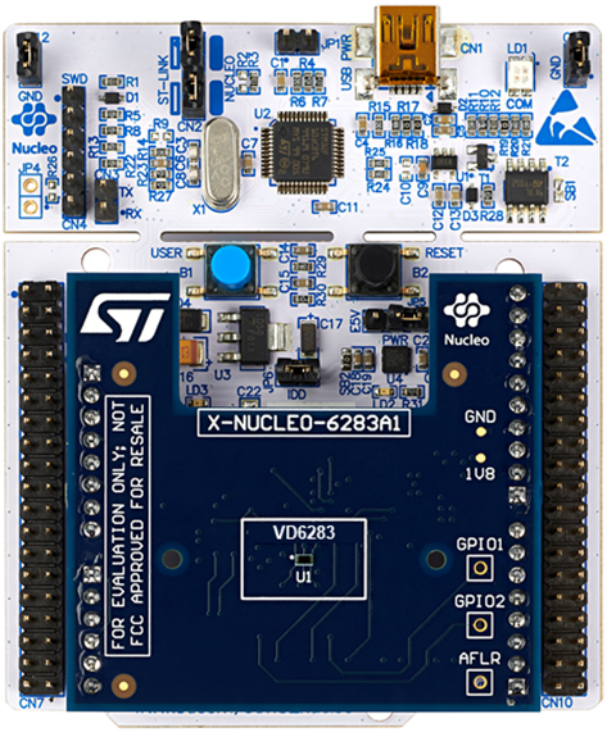
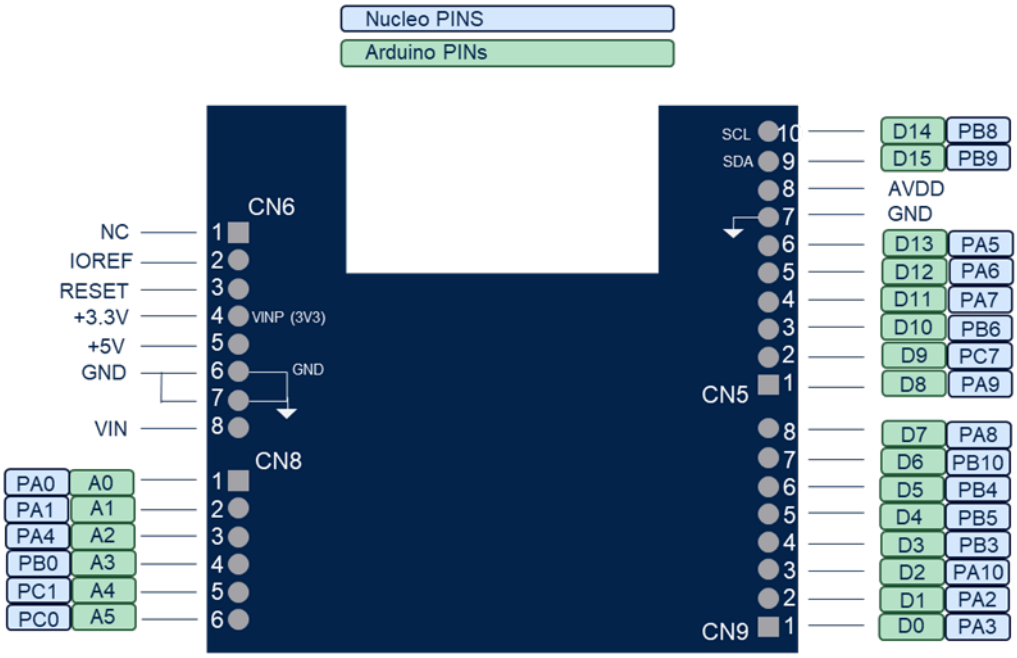


Figure 9. **X-NUCLEO-6283A1 pinout**



6.1 Use of expansion software without sample applications for X-NUCLEO-6283A1

This section outlines how to configure the STM32CubeMX with X-NUCLEO-6283A1 when sample applications are not required. With such a setup, only middleware and driver layers are configured. This setup is used when the user does not intend to leverage the sample application provided in the package.

To add the X-CUBE-ALS software pack to the project:

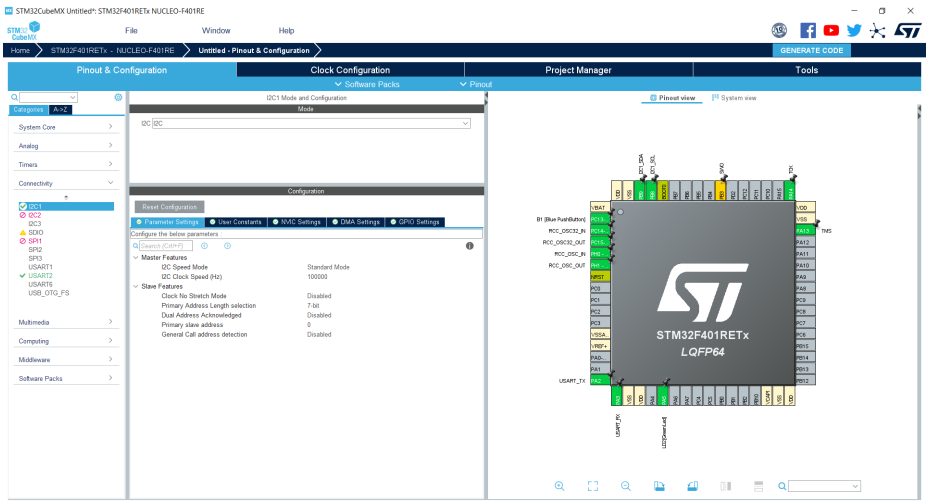
1. Select Software Packs and then Select Components.
2. On the Software Packs Component Selector window, select the Board Extension class as shown in Figure 7. STM32CubeMX Software Packs Component Selector.
3. On the Pinout & Configuration tab:
   - From the Pinout scheme, click on PB8 and set it as I2C1_SCL
   - From the Pinout scheme, click on PB9 and set it as I2C1_SDA
   - Enable the I2C1 as I2C from the Connectivity category
   - Configure the I2C1 settings with I2C speed at 400 kHz (Fast Mode) from the configuration view

4. From the Pinout scheme set:

Table 1. Pinout scheme

Name	Supported IPs	Nucleo 64	Nucleo 144
INT_PIN	GPIO:EXTI1	PB0	PF3
I2C_SCL	I2C1_SCL	PB8	PB8
I2C_SDA	I2C1_SDA	PB9	PB9

Figure 10. STM32CubeMX Pinout & Configuration tab and I2C settings

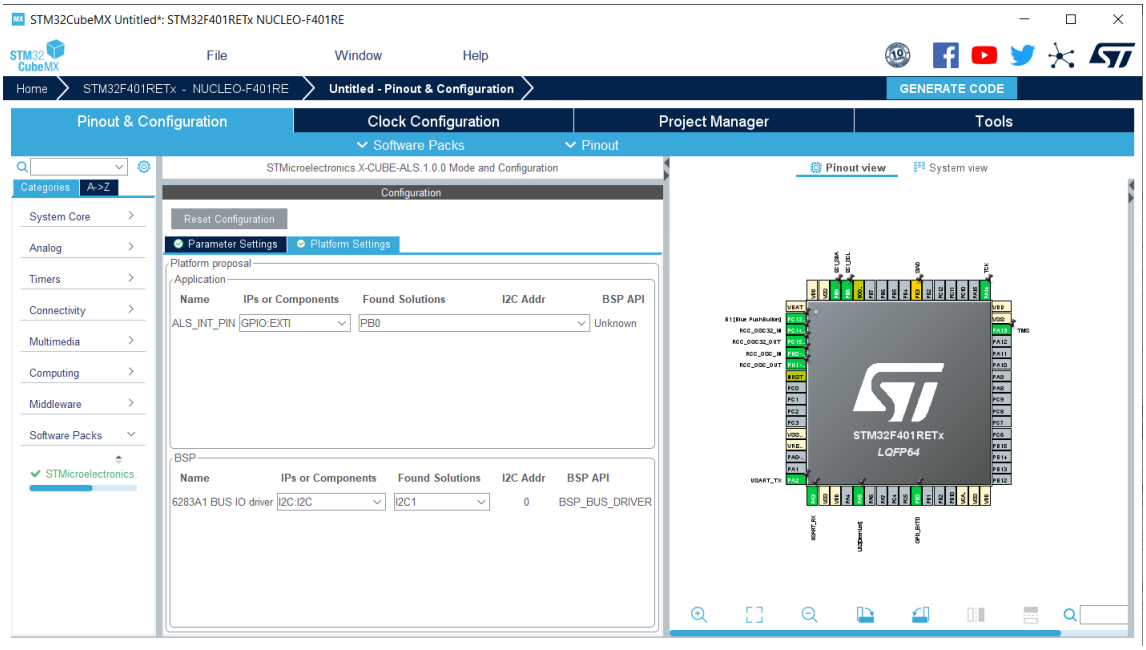


5. On the Software Packs tab, click on the STMicroelectronics.X-CUBE-ALS.1.0.0 item, enable the Board Extension X-NUCLEO-6283A1 checkbox from the Mode view and set the following platform settings from the Configuration view.

Table 2. Platform settings

Name	BSP_API	Supported IPs	Nucleo 64	Nucleo 144
ALS_INT_PIN	Unknown	GPIO:EXTI	PB0	PF3
6283A1 BUS IO driver	BSP_BUS_DRIVER	I2C:I2C	I2C1	I2C1

Figure 11. STM32CubeMX Pinout & Configuration tab and Software Packs settings



6. Once all the steps above have been performed, the source code of the project using the STMicroelectronics X-CUBE-ALS software can be generated by clicking Generate Code.

6.2 Use of expansion software with sample applications for X-NUCLEO-6283A1

This section outlines how to configure STM32CubeMX with X-NUCLEO-6283A1 when the use of the sample applications is required. With such a setup, all the components of the expansion software package, including applications, will be properly configured.

6.2.1 Pinout & Configuration setup

To add the X-CUBE-ALS software pack to the project:

1. Select Software Packs and then Select Components.
2. From the Software Packs Component Selector window, select the Board Extension class as shown in Figure 7. STM32CubeMX Software Packs Component Selector.
3. From the Pinout & Configuration tab:
   - From the Pinout scheme, click on PB8 and set it as I2C1_SCL
   - From the Pinout scheme, click on PB9 and set it as I2C1_SDA
   - Enable the I2C1 as I2C from the Connectivity category
   - Configure the I2C1 settings with I2C speed at 400 kHz (Fast Mode) from the configuration view
   - If not enabled yet:
     - Enable the USART2 in Asynchronous mode (for Nucleo 64) from the Connectivity category (115200 bps, 8N1)
     - Enable the USART3 in Asynchronous mode (for Nucleo 144) from the Connectivity category (115200 bps, 8N1)

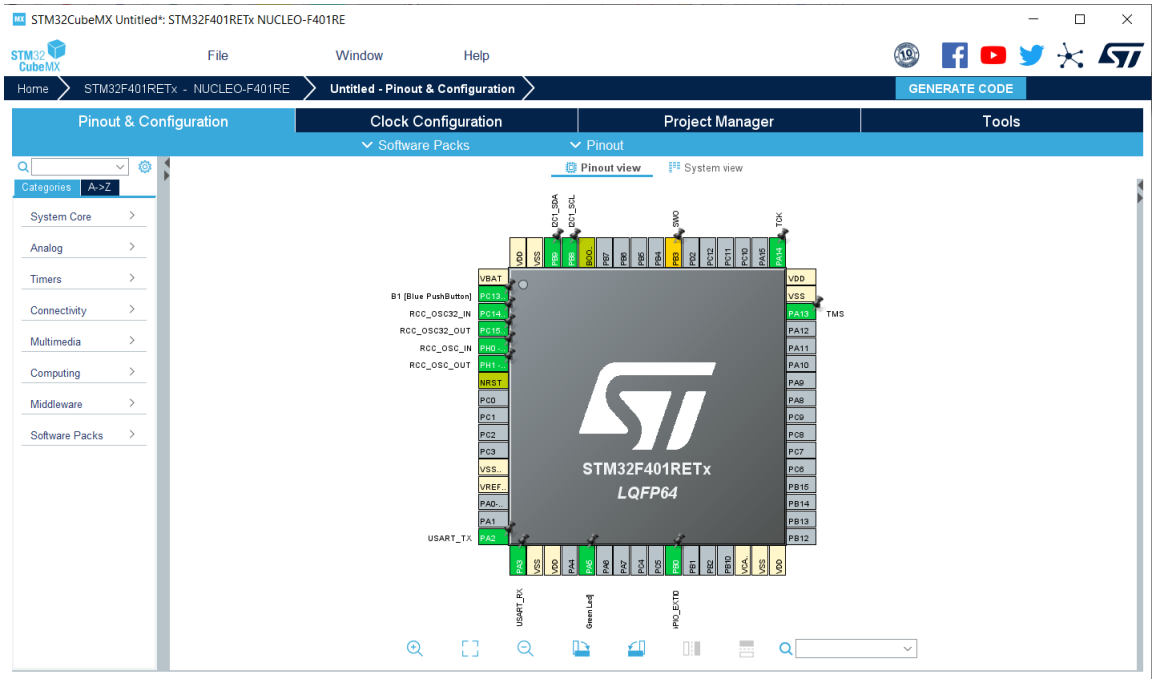
Note: The schematics may differ, please check the schematics of your board.

4. From the Pinout scheme, if you are using the X-NUCLEO-6283A1 set:

Table 3. Pinout scheme

Nucleo 64			Nucleo 144		
Pin	Mode	Label	Pin	Mode	Label
PB0	GPIO_EXTI0	ALS_INT_PIN	PF3	GPIO_EXTI3	ALS_INT_PIN
PA2	USART2_TX	USART_TX	PD8	USART3_TX	USART_TX
PA3	USART2_RX	USART_RX	PD9	USART3_RX	USART_RX
PA5	GPIO_Output	LD2 [Green Led]	PB7	GPIO_Output	LD2[Blue]
PC13	GPIO_EXTI13	B1 [Blue PushButton]	PC13	GPIO_EXTI13	USER_Btn[B1]

Figure 12. STM32CubeMX Pinout & Configuration tab for X-NUCLEO-6283A1

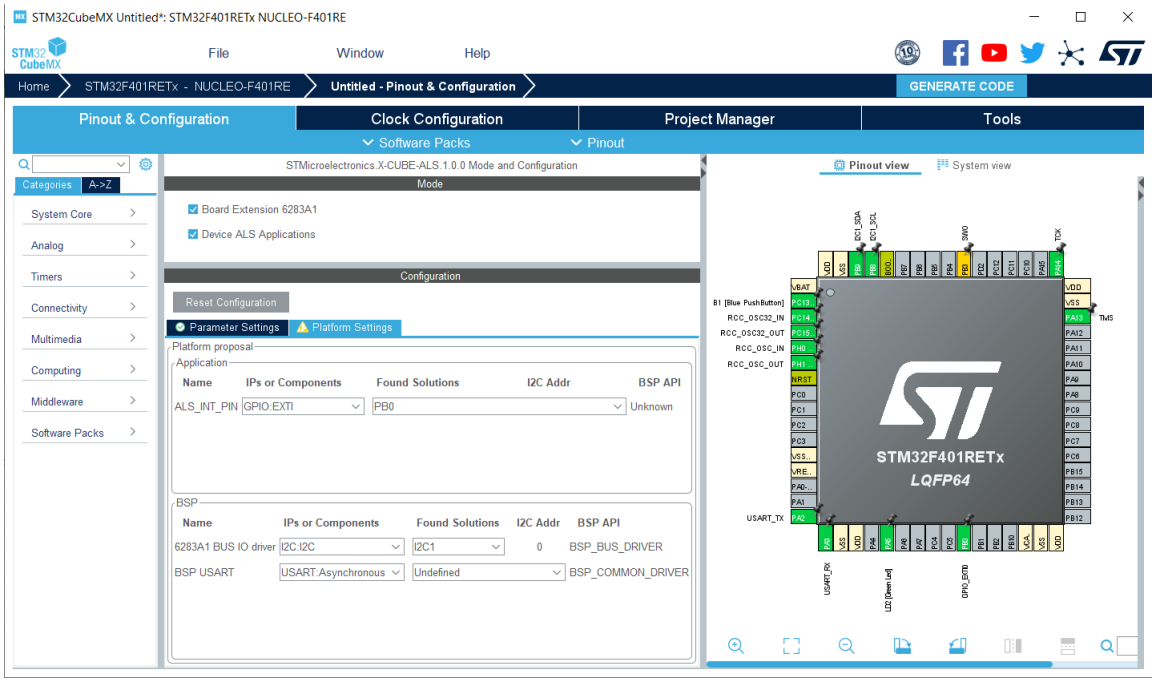


5. From the Software Packs category, click on the STMicroelectronics.X-CUBE-ALS.1.0.0 item, enable the Board Extension X-NUCLEO-6283A1 and the Device ALS Applications checkboxes from the Mode view. Then set the following platform settings from the Configuration view.

Table 4. Platform settings

Name	BSP_API	Supported IPs	Nucleo 64	Nucleo 144
ALS_INT_PIN	Unknown	GPIO:EXTI	PB0	PF3
6283A1 BUS IO driver	BSP_BUS_DRIVER	I2C:I2C	I2C1	I2C1
BSP USART	BSP_COMMON_DRIVER	USART:Asynchronous	USART2	USART3

Figure 13. STM32CubeMX Pinout & Configuration tab and software pack settings for X-NUCLEO-6283A1



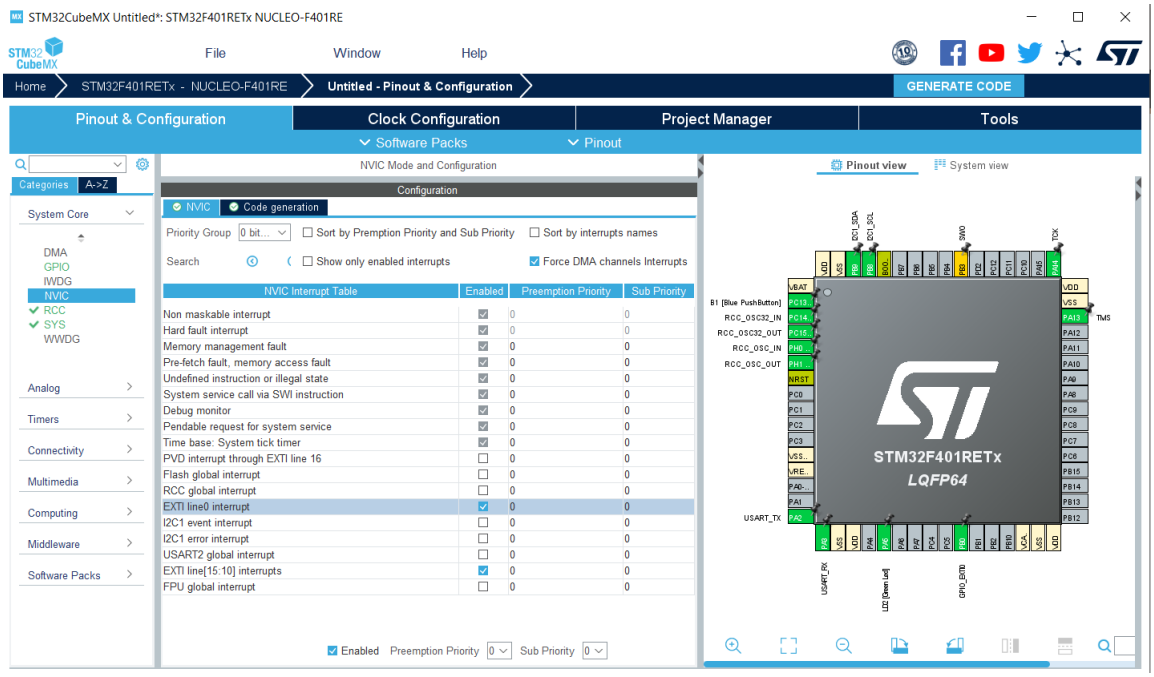
6. From the Parameter Settings tab, some parameters for the routing of the interrupt signals can be changed.

7. From the Configuration & Pinout tab, click on System Core and then on the NVIC item to enable the EXTI line interrupts when the X-NUCLEO-6283A1 is used.

Table 5. EXTI line interrupts settings

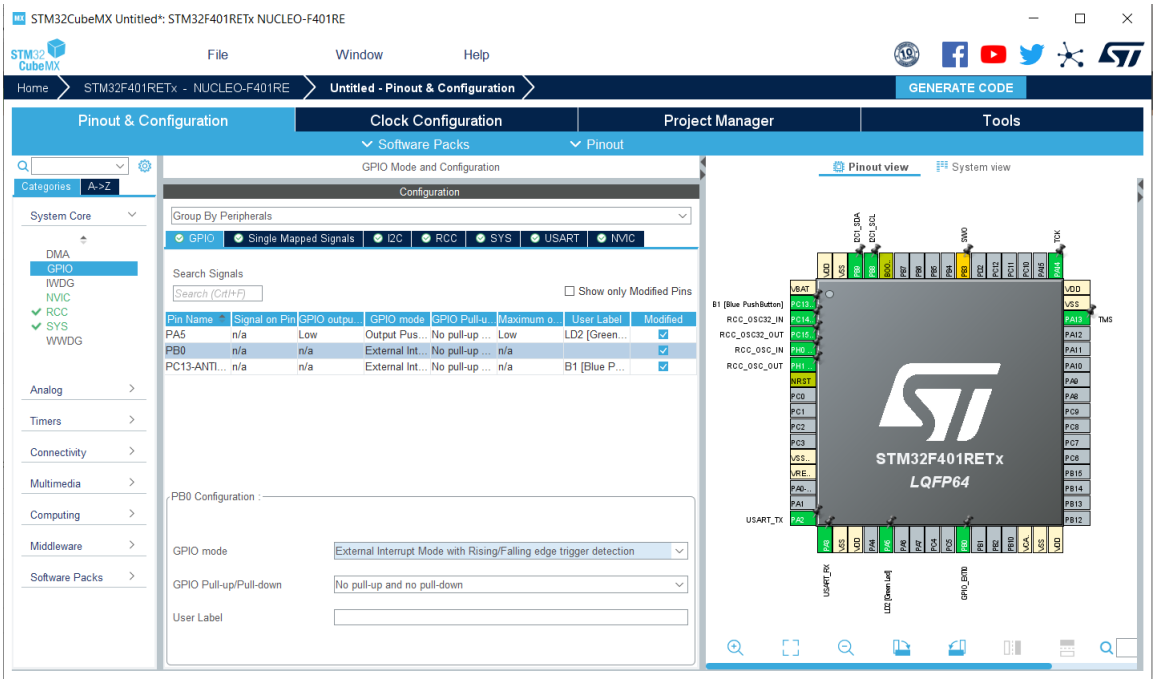
Nucleo 64	Nucleo 144
EXTI line 0 interrupt	EXTI line 3 interrupt
EXTI line 13 interrupt	EXTI line 13 interrupt

Figure 14. STM32CubeMX NVIC Configuration for X-NUCLEO-6283A1



8. The GPIO that manages the interrupt of the VD6283 mounted on the X-NUCLEO-6283A1 must be configured in External Interrupt Mode with Falling edge trigger detection from the GPIO section of the System Core menu.

Figure 15. STM32CubeMX GPIO configuration for X-NUCLEO-6283A1

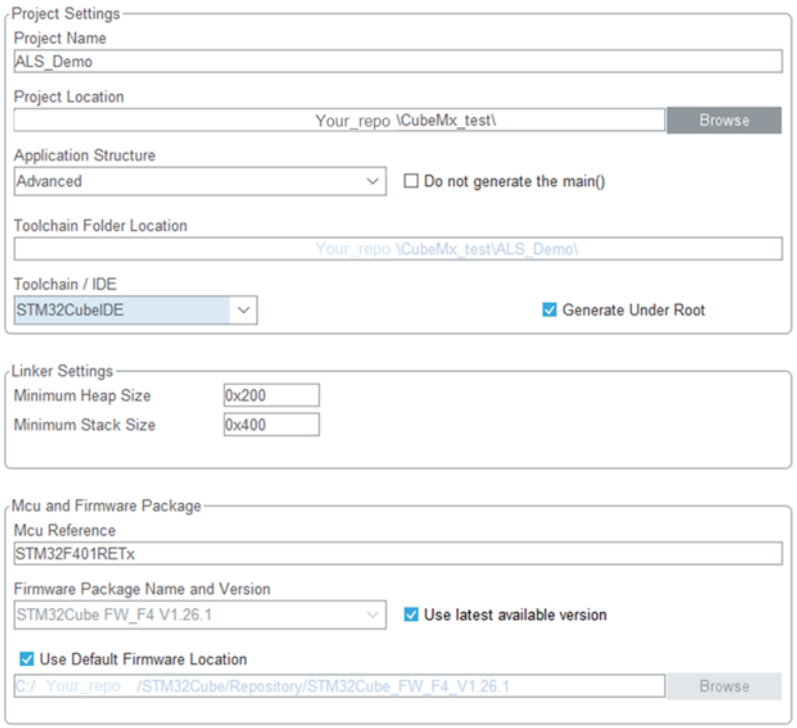


9. Once all the steps above have been performed, the sample applications for X-NUCLEO-6283A1 using the STMicroelectronics X-CUBE-ALS software can be generated clicking Generate Code.

6.2.2 Project manager configuration and code generator

1. Go to the Project manager.
2. Fill the name of the project, and choose the correct toolchain / IDE.

Figure 16. Cube MX Project manager configuration



3. Once all these settings are set, click on Generate code.
4. On the pop up select open project. The STM32CubeIDE project opens.

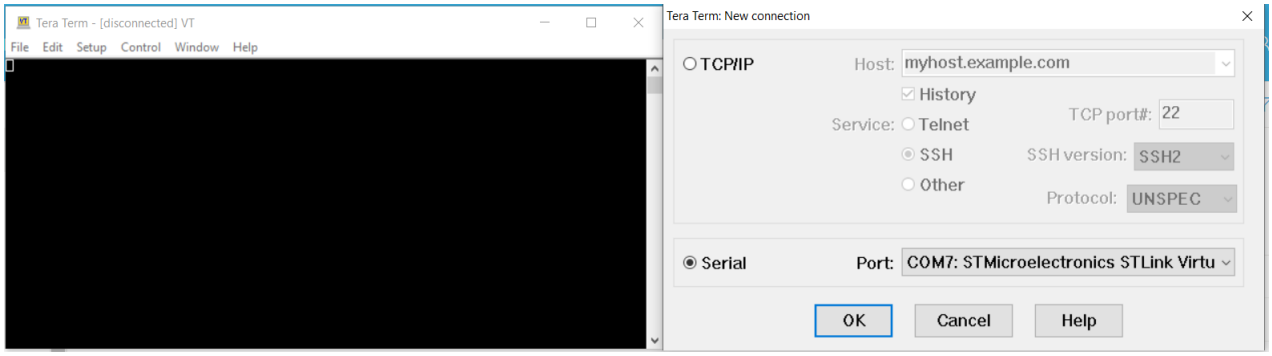
6.2.3 Final configuration

Once you have completed the steps in the previous section, when the STM32CubeIDE project is open:

1. Build the project.
2. Run the project.

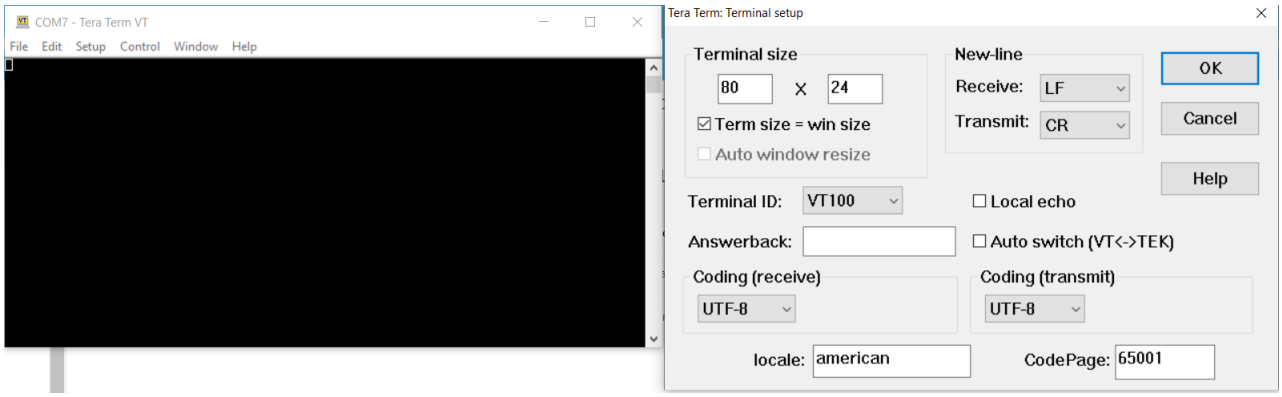
3. Configure a hyper terminal like Tera Term as follows:
   a. Choose a serial connection and select the right COM Port

Figure 17. Hyperterminal communication configuration



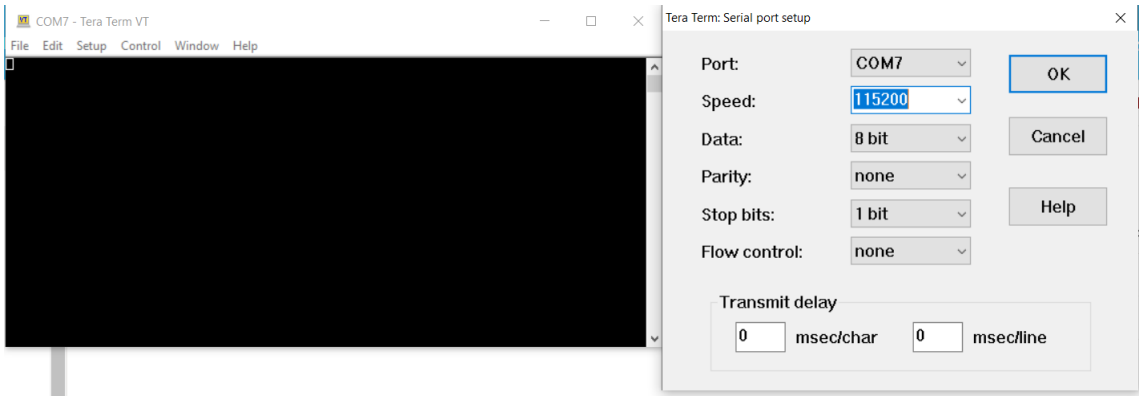
   b. In the Setup menu, go to Terminal and set New line Receive as LF and Transmit as CR

Figure 18. Hyperterminal lines setup



   c. Configure the baud rate to 115200.

Figure 19. Hyperterminal serial port



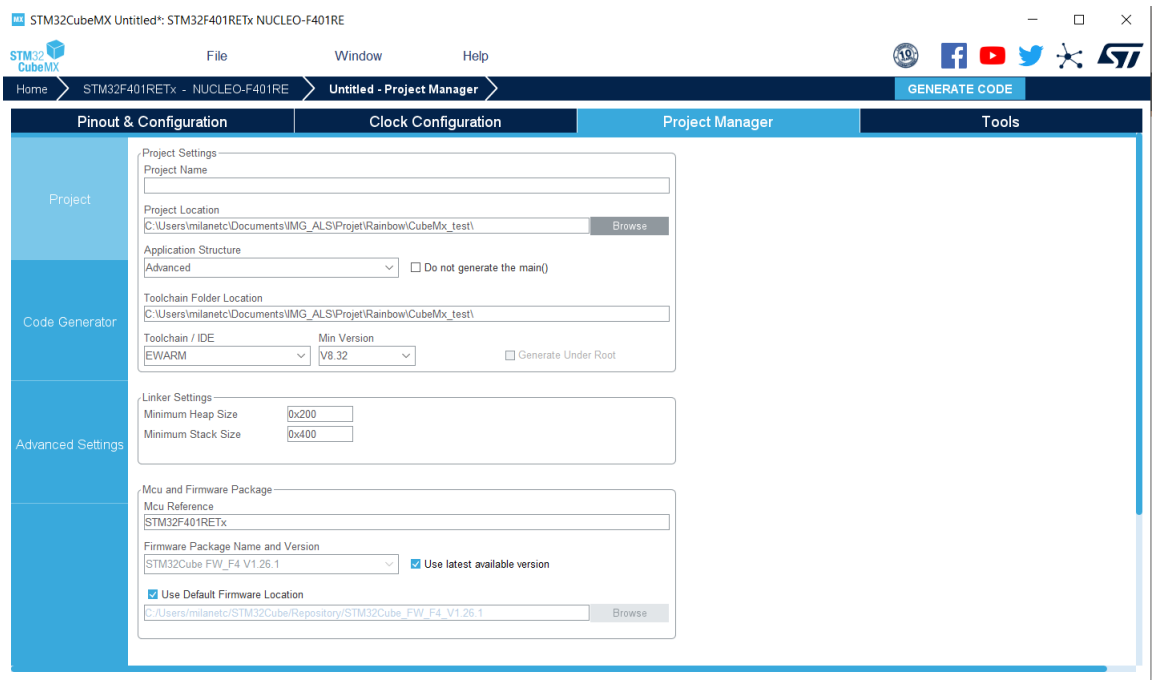
Once these steps are complete, the hyper terminal will display the application values (as shown below).

7 Generated folders structure

When generating a project, two models of folder structure can be adopted when using a high level firmware component (i.e. a middleware in the STM32Cube MCU package):

- Basic Structure: the basic structure is often used with HAL examples and single package projects. This structure consists of having the IDE configuration folder in the same level as the sources (organized in Inc and Src subfolders)
- Advanced Structure: the advanced structure provides a more efficient and organized folders model that allows ease middleware applications integration when several packages are used. In the Advanced mode Src and Inc are generated under folder Core. For this package, the list of the generated files is under X-CUBE-ALS, at the same level as Core and containing inside the App and the Target subfolder

Figure 20. STM32CubeMX application structure configuration



8 References

DS13735 *Hybrid filter multispectral sensor with light flicker engine*

UM2858 *Getting started with X-NUCLEO-6283A1 expansion board based on VD6283 6-channel ambient light sensor with flicker extraction*

AN5638 *How correlated color temperature is calculated by VD6283*

AN5639 *Using flicker with VD6283*

Revision history

Table 6. **Document revision history**

Date	Version	Changes
10-May-2021	1	Initial release

Contents