Getting started with the X-CUBE-DISPLAY STM32Cube Expansion Package

Introduction

This user manual gives an overview of the X-CUBE-DISPLAY Expansion Package and explains how to get started in STM32CubeMX.

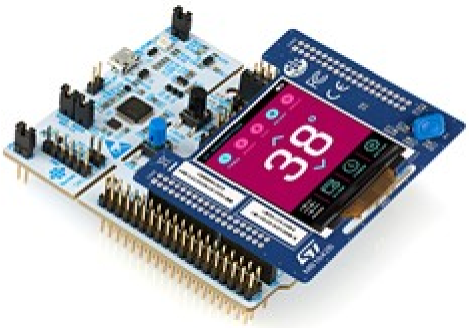
The X-CUBE-DISPLAY Expansion Package enables display and user interface features on top of STM32 Nucleo-64 and Nucleo-144 boards. Various user interfaces are supported by the X-CUBE-DISPLAY Expansion Package such as LCD controllers, touch panel, user buttons, joystick, external memories, and others.

The X-CUBE-DISPLAY Expansion Package supports the X-NUCLEO-GFX01M1, X-NUCLEO-GFX01M2, and X-NUCLEO-GFX02Z1 expansion boards, offering the needed drivers as well as simple examples.

The X-CUBE-DISPLAY main features are:

- Software building block supporting the addition of display expansion boards to Nucleo-64 and Nucleo-144 boards, including
  - drivers for the supported LCD controllers
  - drivers for the supported external flash memories
- Simple "hello world" example available on
  - the X-NUCLEO-GFX01M1 expansion board connected to the NUCLEO-G071RB or NUCLEO-L476RG Nucleo-64 development board
  - the X-NUCLEO-GFX01M2 expansion board connected to the NUCLEO-G071RB or NUCLEO-WB55RG Nucleo-64 development board
  - the X-NUCLEO-GFX02Z1 expansion board connected to the NUCLEO-L4R5ZI or NUCLEO-U575ZI-Q Nucleo-144 development board
- FreeRTOS™ examples available on
  - the X-NUCLEO-GFX01M1 expansion board connected to the NUCLEO-L476RG Nucleo-64 development board
  - the X-NUCLEO-GFX02Z1 expansion board connected to the NUCLEO-L4R5ZI Nucleo-144 development board
- Azure® RTOS ThreadX examples available on
  - the X-NUCLEO-GFX02Z1 expansion board connected to the NUCLEO-U575ZI-Q Nucleo-144 development board
- Easy portability across STM32 microcontrollers with STM32CubeMX in the STM32Cube ecosystem
- Free and user-friendly license terms

Figure 1. X-CUBE-DISPLAY running on NUCLEO-G071RB equipped with X-NUCLEO-GFX01M1



Picture is not contractual.

1 General information

This user manual briefly describes the X-CUBE-DISPLAY Expansion Package. It focuses on X-CUBE-DISPLAY usage. It does not explain the LCD controllers and the input devices.

Table 1 presents the definitions of the acronyms that are relevant for a better understanding of this document.

Table 1. List of acronyms

Acronym	Definition
API	Application programming interface
BSP	Board support package
FMC	Flexible memory controller
HAL	Hardware abstraction layer
IDE	Integrated development environment
LCD	Liquid-crystal display
SPI	Serial peripheral interface
UIX	User interaction and experience

The X-CUBE-DISPLAY Expansion Package runs on STM32 32-bit microcontrollers based on the Arm® Cortex® processor. Some of its examples run with the FreeRTOS™ or Azure® RTOS ThreadX middleware.

Note: Arm is a registered trademark of Arm Limited (or its subsidiaries) in the US and/or elsewhere.

FreeRTOS is a trademark of Amazon in the United States and/or other countries.

Azure is a trademark of the Microsoft group of companies.

2 Package description

This section details the X-CUBE-DISPLAY Expansion Package contents and how to use it.

2.1 General description

The X-CUBE-DISPLAY Expansion Package provides simple and generic support software for the X-NUCLEO-GFX01M1, X-NUCLEO-GFX01M2, and X-NUCLEO-GFX02Z1 expansion boards. It runs on the STM32 microcontroller.

The following integrated development environments are supported:

- IAR Systems® - IAR Embedded Workbench® for Arm® (EWARM)
- Keil® - Microcontroller Development Kit for Arm®-based microcontrollers (MDK-ARM)
- STMicroelectronics - STM32CubeIDE

Note: *Refer to the release note available in the root folder of the delivery package for information about the IDE versions supported.*

2.2 Architecture

The software runs on STM32 Nucleo-64 and Nucleo-144 boards. It allows the display and control of graphics and texts via add-on display, memories, and controlling modules.

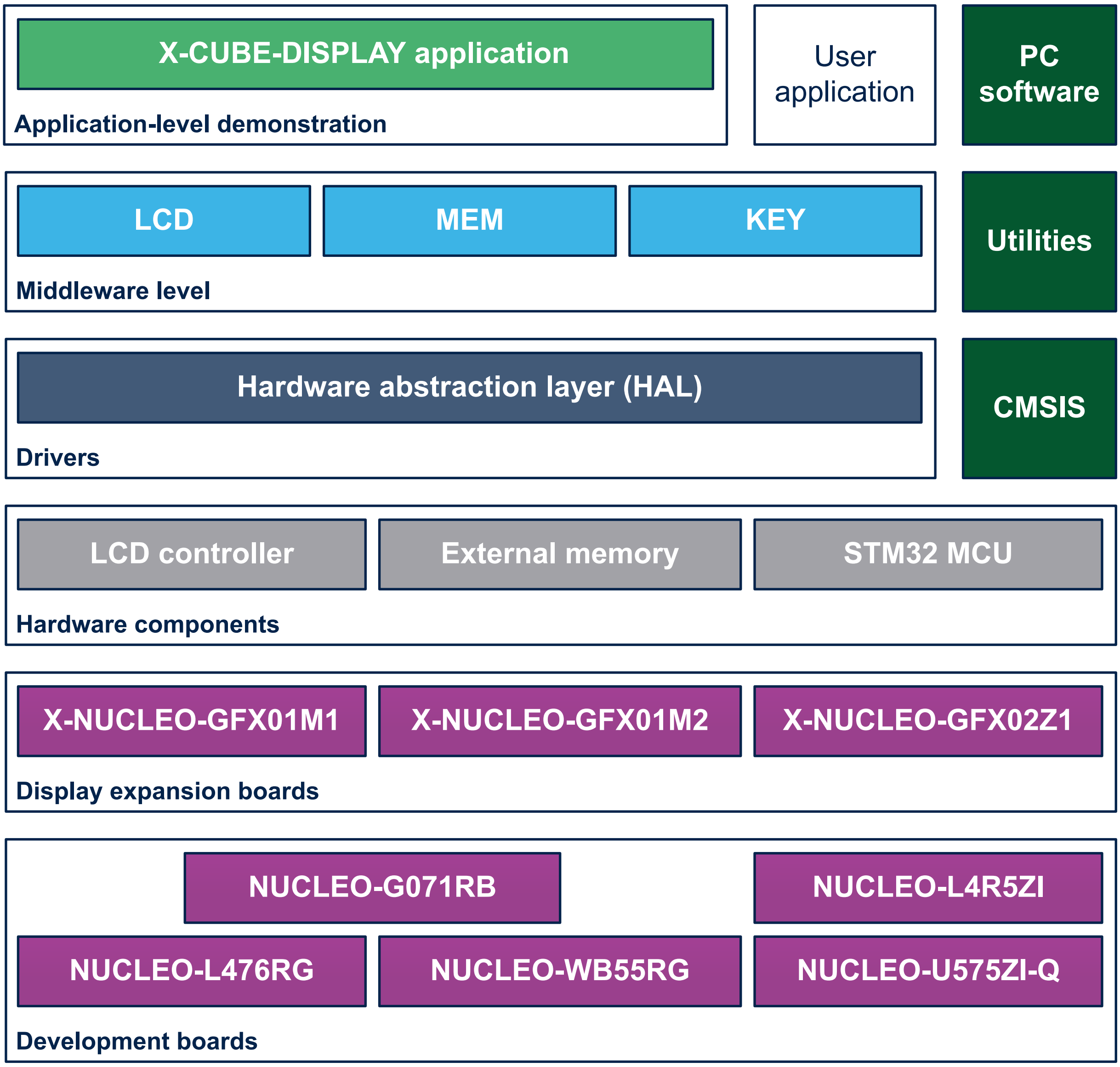
The package is split into the following components:

- LCD controllers
- External memories
- Buttons and joysticks

2.2.1 Architecture concept

This section provides a high-level view of the software architecture to support the display expansion boards.

Figure 2. Architecture concept



The display expansion boards expose three main interfaces to applications:

- Display interface: via LCD controllers or dedicated display bus
- External memory interface: external memory devices to store graphical assets
- User interface: touch panel, joystick, and buttons

The `LCD_IO` layer abstracts the actual hardware bus interface used with the LCD controller.

The `MEM_IO` layer abstracts the actual hardware bus interface used with external memory devices.

The `KEY_IO` layer abstracts the actual hardware button interface used with the expansion board.

2.3 X-CUBE-DISPLAY Expansion Package contents

The X-CUBE-DISPLAY Expansion Package is an expansion for STM32CubeMX with the following main characteristics:

- It complies with the STM32CubeMX architecture recommendation (code generator and folders)
- It expands STM32CubeMX to enable the development of UIX applications
- It is based on STM32CubeHAL, which is the hardware abstraction layer for STM32 microcontrollers

The software components used by the application software to exchange with remote applications are the following ones:

- **STM32CubeHAL**: 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 dependencies on the specific hardware configuration for a given microcontroller.

  This structure improves the library code reusability and guarantees easy portability onto other devices.
- **Configuration files**:
  - `lcd_conf.h` defines the mapping of the GPIO and hardware interface specific to a logical name to ease software porting to another board. It provides also the configuration for the hardware bus interface (for example SPI) to be used for communication with the LCD controller.
  - `mem_conf.h` defines the mapping of the GPIO and hardware interface specific to a logical name to ease software porting to another board. It provides also the configuration for the hardware bus interface (for example SPI) to be used for communication with the external memory device.
  - `key_conf.h` defines the mapping of the GPIO specific to a logical name to ease software porting to another board. It provides the GPIO configurations to be used to provide controls for the user.
- **Example projects**: This release provides the "hello world" examples for the board associations listed in the table below.

Table 2. **Board associations demonstrated in X-CUBE-DISPLAY**

STM32 Nucleo boards		Expansion boards		
		X-NUCLEO-GFX01M1	X-NUCLEO-GFX01M2	X-NUCLEO-GFX02Z1
Nucleo-64	NUCLEO-G071RB	Yes	Yes	-
	NUCLEO-L476RG	Yes	-	-
	NUCLEO-WB55RG	-	Yes	-
Nucleo-144	NUCLEO-L4R5ZI	-	-	Yes
	NUCLEO-U575ZI-Q	-	-	Yes

3 Sample application description

This section provides a short overview of the sample applications and examples included in X-CUBE-DISPLAY.

The sample applications and examples

- are ready-to-use projects that can be generated with STM32CubeMX for any supported STM32 Nucleo-64 or Nucleo-144 board used together with the X-NUCLEO-GFX01M1, X-NUCLEO-GFX01M2, or X-NUCLEO-GFX02Z1 expansion board
- show the users how to use the APIs to initialize correctly and use the X-NUCLEO-GFX01M1, X-NUCLEO-GFX01M2, and X-NUCLEO-GFX02Z1 peripherals

3.1 GFX01M1_HelloWorld application

This application provides a "hello world" example that configures the LCD controller and displays an image stored in the X-NUCLEO-GFX01M1 expansion board external memory. The example provides controls to the user through the joystick buttons.

3.2 GFX01M2_HelloWorld application

This application provides a "hello world" example that configures the LCD controller and displays an image stored in the X-NUCLEO-GFX01M2 expansion board external memory. The example provides controls to the user through the joystick buttons.

3.3 GFX02Z1_HelloWorld application

This application provides a "hello world" example that configures the LCD controller and displays an image stored in the X-NUCLEO-GFX02Z1 expansion board external memory. The example provides controls to the user through the joystick buttons.

3.4 GFX01M2_FreeRTOS application

This application is similar to GFX01M2_HelloWorld with the exception that it runs with FreeRTOS™.

3.5 GFX02Z1_FreeRTOS application

This application is similar to GFX02Z1_HelloWorld with the exception that it runs with FreeRTOS™.

3.6 GFX02Z1_ThreadX application

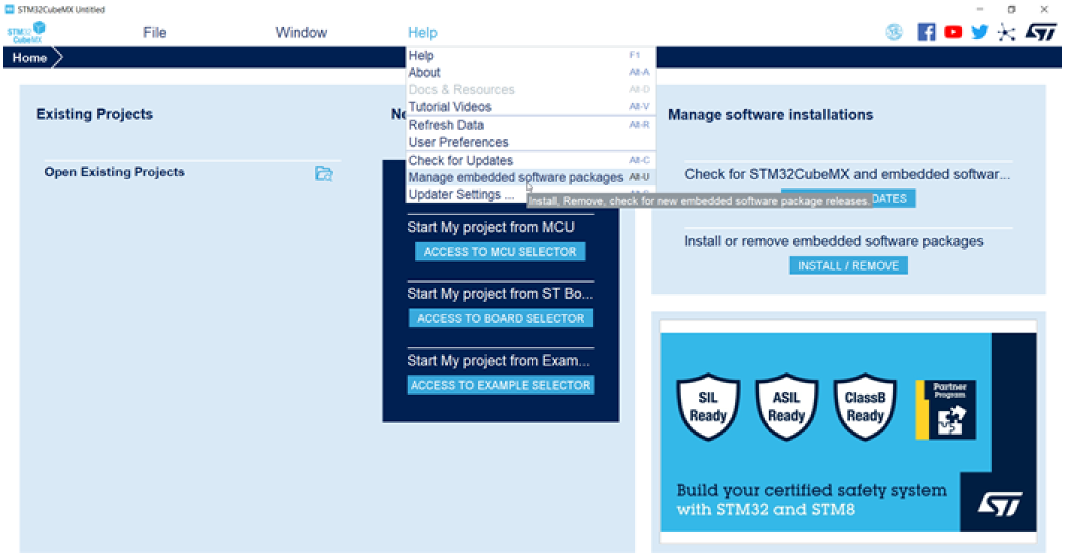
This application is similar to GFX02Z1_HelloWorld with the exception that it runs with Azure® RTOS ThreadX.

4 Installing X-CUBE-DISPLAY in STM32CubeMX

After downloading (from *www.st.com*), installing and launching STM32CubeMX (V6.0.0 or above), the X-CUBE-DISPLAY package can be installed following the few steps described below.

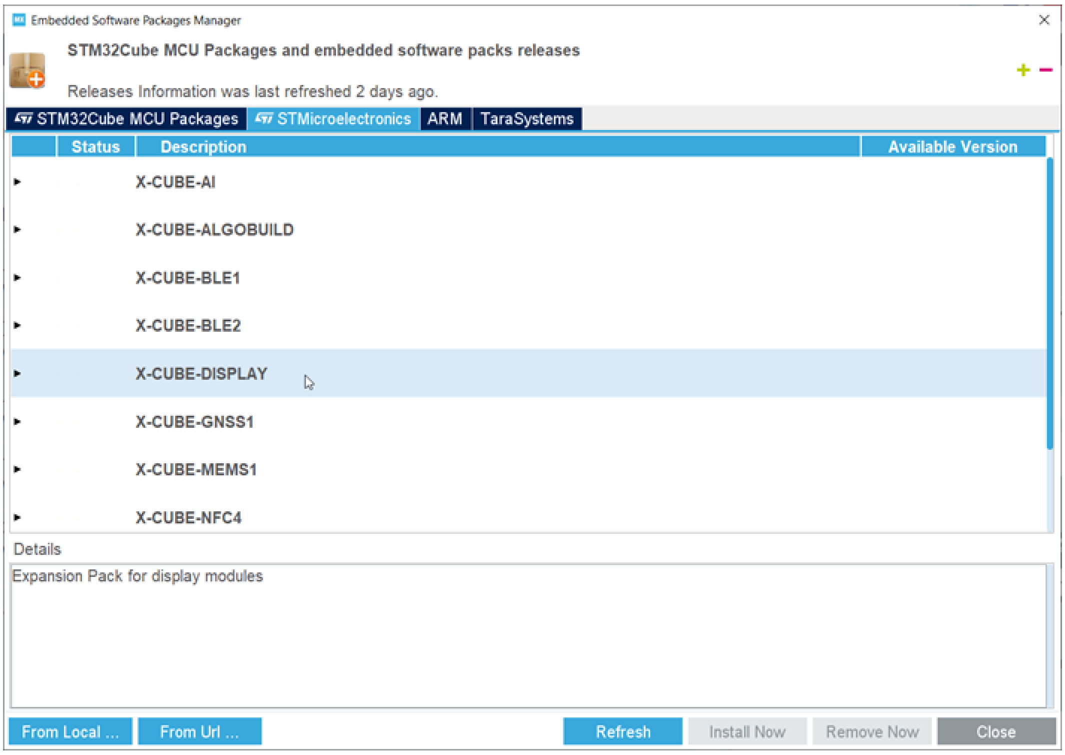
Step 1. From the menu, select [**Help**]>[**Manage embedded software packages**].

Figure 3. **Managing embedded software packages in STM32CubeMX**


STM32CubeMX Untitled
File
Window
Help
Home
Help F1
About Alt-A
Docs & Resources Alt-D
Tutorial Videos Alt-V
Refresh Data Alt-R
User Preferences
Check for Updates Alt-C
Manage embedded software packages Alt-U
Updater Settings ...
Install, Remove, check for new embedded software package releases.
Existing Projects
Open Existing Projects
Start My project from MCU
ACCESS TO MCU SELECTOR
Start My project from ST Bo...
ACCESS TO BOARD SELECTOR
Start My project from Exam...
ACCESS TO EXAMPLE SELECTOR
Manage software installations
Check for STM32CubeMX and embedded softwar...
DATES
Install or remove embedded software packages
INSTALL / REMOVE
SIL Ready
ASIL Ready
ClassB Ready
Partner Program
Build your certified safety system with STM32 and STM8

Step 2. From the *Embedded Software Packages Manager* window, press the [**Refresh**] button to get the updated add-on package list. Go to the *STMicroelectronics* tab to find the X-CUBE-DISPLAY package.

Figure 4. **Installing the X-CUBE-DISPLAY package in STM32CubeMX**



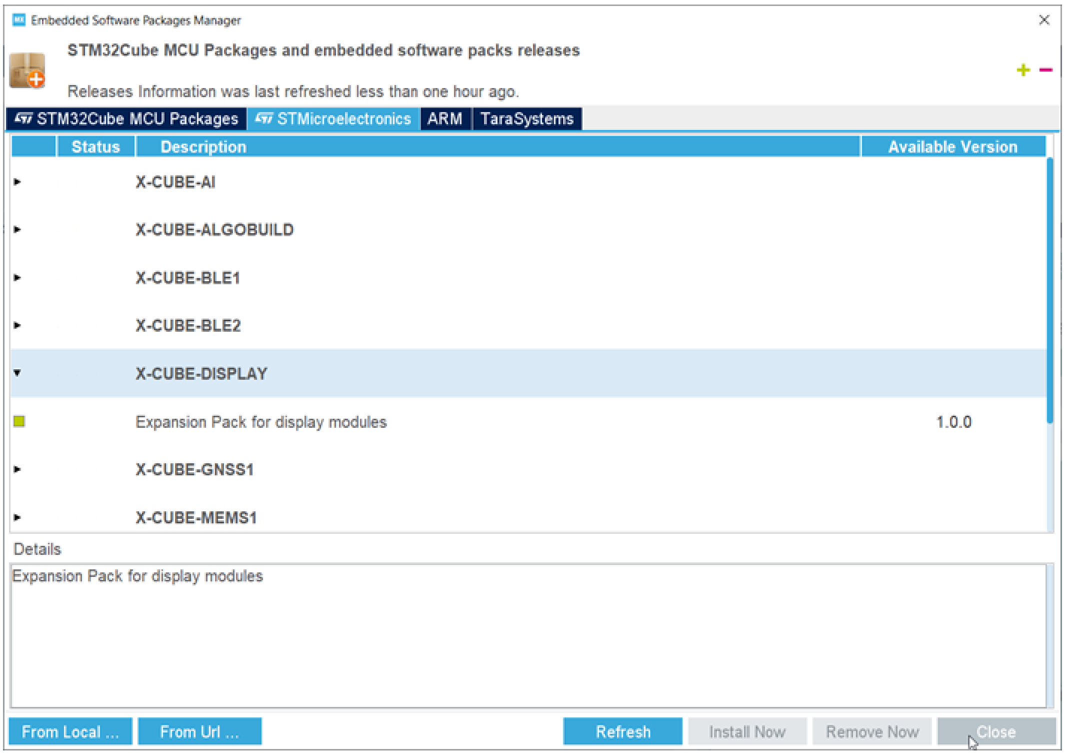
Releases Information was last refreshed 2 days ago.

Status	Description	Available Version
	X-CUBE-AI	
	X-CUBE-ALGOBUILD	
	X-CUBE-BLE1	
	X-CUBE-BLE2	
	X-CUBE-DISPLAY	
	X-CUBE-GNSS1	
	X-CUBE-MEMS1	
	X-CUBE-NFC4	

Expansion Pack for display modules

Step 3. Checking the corresponding checkbox to select X-CUBE-DISPLAY and install it by pressing the [**Install Now**] button. Once the installation is completed, the corresponding box becomes green, the [**Close**] button can be pressed, and the configuration of a new project can start.

Figure 5. **The X-CUBE-DISPLAY package in STM32CubeMX**



STM32Cube MCU Packages and embedded software packs releases

Releases Information was last refreshed less than one hour ago.

	Status	Description	Available Version
▸		X-CUBE-AI	
▸		X-CUBE-ALGOBUILD	
▸		X-CUBE-BLE1	
▸		X-CUBE-BLE2	
▾		X-CUBE-DISPLAY	
🟩		Expansion Pack for display modules	1.0.0
▸		X-CUBE-GNSS1	
▸		X-CUBE-MEMS1	

Details

Expansion Pack for display modules

5 Starting a new project

After launching STM32CubeMX, start a new project as described in the steps below.

Step 1. Choose to start a new project from the *MCU Selector* or *Board Selector*.

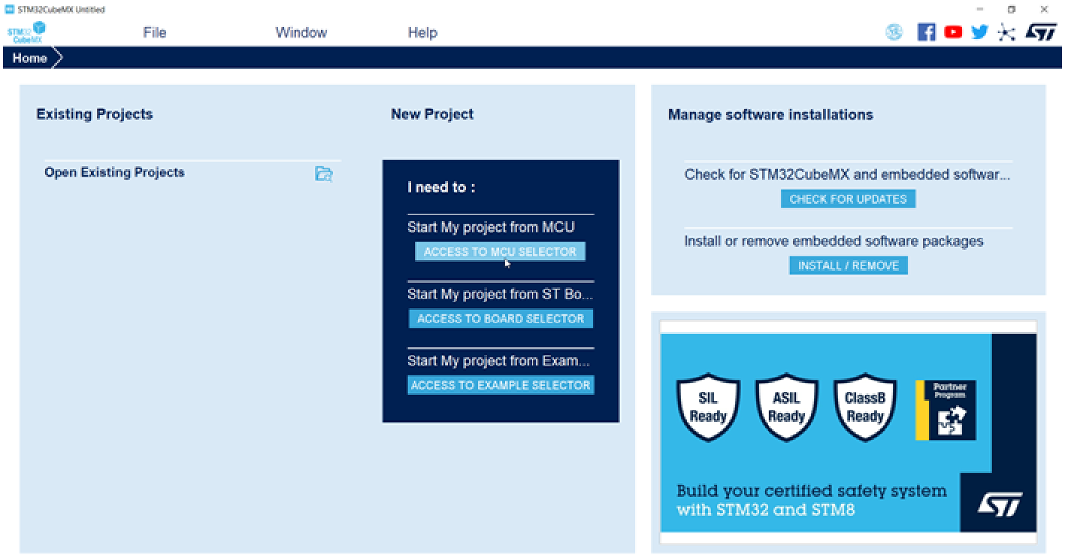
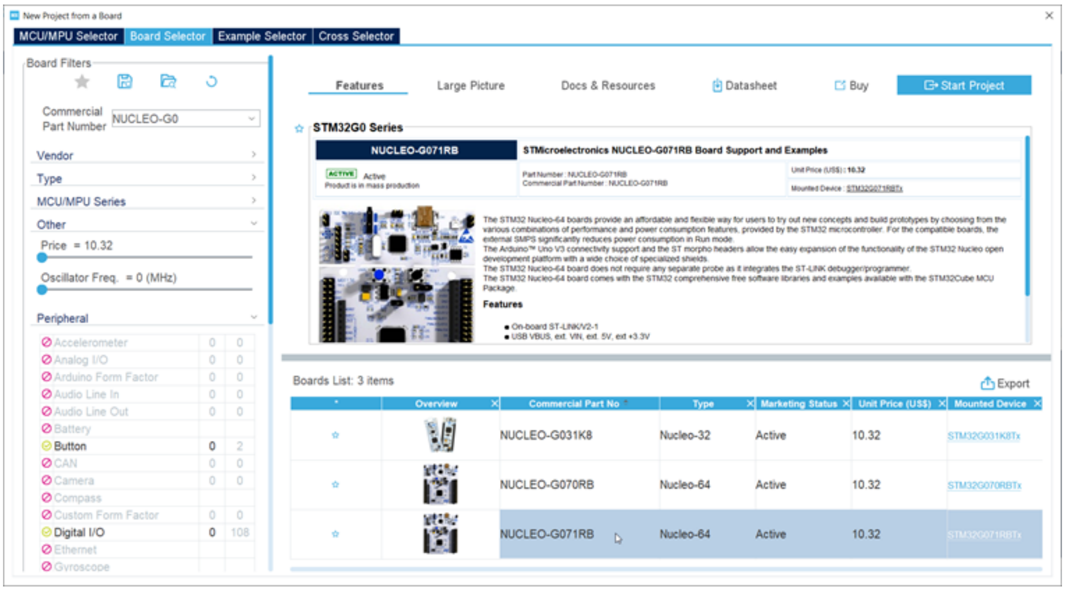

STM32CubeMX Untitled
File
Window
Help
Home
Existing Projects
Open Existing Projects
New Project
I need to :
Start My project from MCU
ACCESS TO MCU SELECTOR
Start My project from ST Bo...
ACCESS TO BOARD SELECTOR
Start My project from Exam...
ACCESS TO EXAMPLE SELECTOR
Manage software installations
Check for STM32CubeMX and embedded softwar...
CHECK FOR UPDATES
Install or remove embedded software packages
INSTALL / REMOVE
SIL Ready
ASIL Ready
ClassB Ready
Partner Program
Build your certified safety system with STM32 and STM8

Figure 6. STM32CubeMX main page

Step 2. Select the desired STM32 microcontroller or platform from the pop-up *MCU Selector* or *Board Selector* window.



Commercial Part Number: NUCLEO-G0

Price = 10.32

Oscillator Freq. = 0 (MHz)

Accelerometer	0	0
Analog I/O	0	0
Arduino Form Factor	0	0
Audio Line In	0	0
Audio Line Out	0	0
Battery		
Button	0	2
CAN	0	0
Camera	0	0
Compass		
Custom Form Factor	0	0
Digital I/O	0	108
Ethernet		
Gyroscope		

STM32G0 Series

NUCLEO-G071RB — STMicroelectronics NUCLEO-G071RB Board Support and Examples

ACTIVE Active
Product is in mass production

Part Number : NUCLEO-G071RB
Commercial Part Number : NUCLEO-G071RB

Unit Price (US$) : 10.32
Mounted Device : STM32G071RBTx

The STM32 Nucleo-64 boards provide an affordable and flexible way for users to try out new concepts and build prototypes by choosing from the various combinations of performance and power consumption features, provided by the STM32 microcontroller. For the compatible boards, the external SMPS significantly reduces power consumption in Run mode.
The Arduino™ Uno V3 connectivity support and the ST morpho headers allow the easy expansion of the functionality of the STM32 Nucleo open development platform with a wide choice of specialized shields.
The STM32 Nucleo-64 board does not require any separate probe as it integrates the ST-LINK debugger/programmer.
The STM32 Nucleo-64 board comes with the STM32 comprehensive free software libraries and examples available with the STM32Cube MCU Package.

Features

- On-board ST-LINK/V2-1
- USB VBUS, ext. VIN, ext. 5V, ext +3.3V

Boards List: 3 items

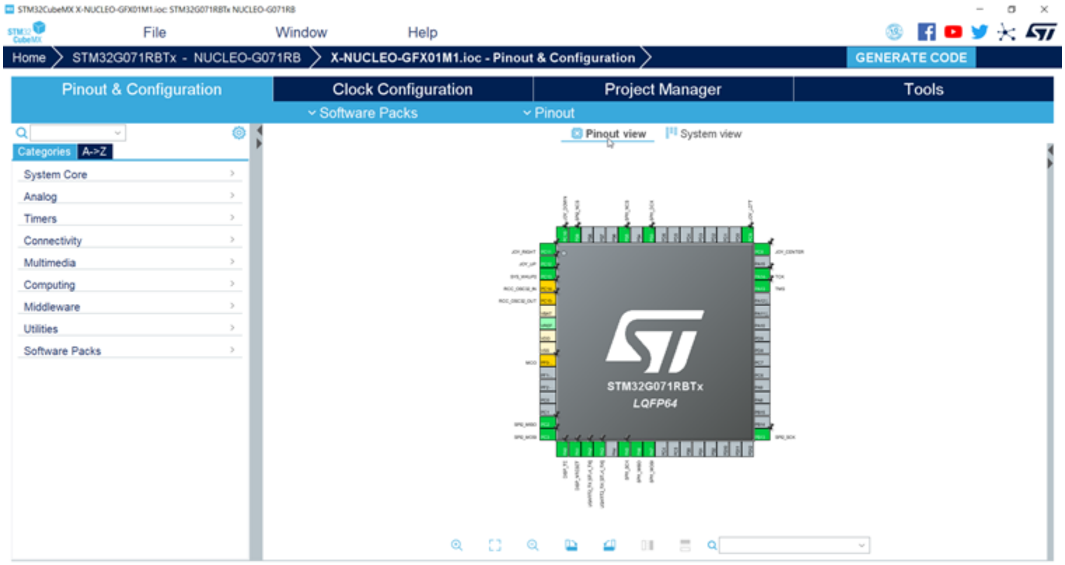
*	Overview	Commercial Part No	Type	Marketing Status	Unit Price (US$)	Mounted Device
☆		NUCLEO-G031K8	Nucleo-32	Active	10.32	STM32G031K8Tx
☆		NUCLEO-G070RB	Nucleo-64	Active	10.32	STM32G070RBTx
☆		NUCLEO-G071RB	Nucleo-64	Active	10.32	STM32G071RBTx

Figure 7. STM32CubeMX *Board Selector* window example

Step 3. Select the microcontroller or the board to display the pinout of the corresponding STM32 microcontroller.

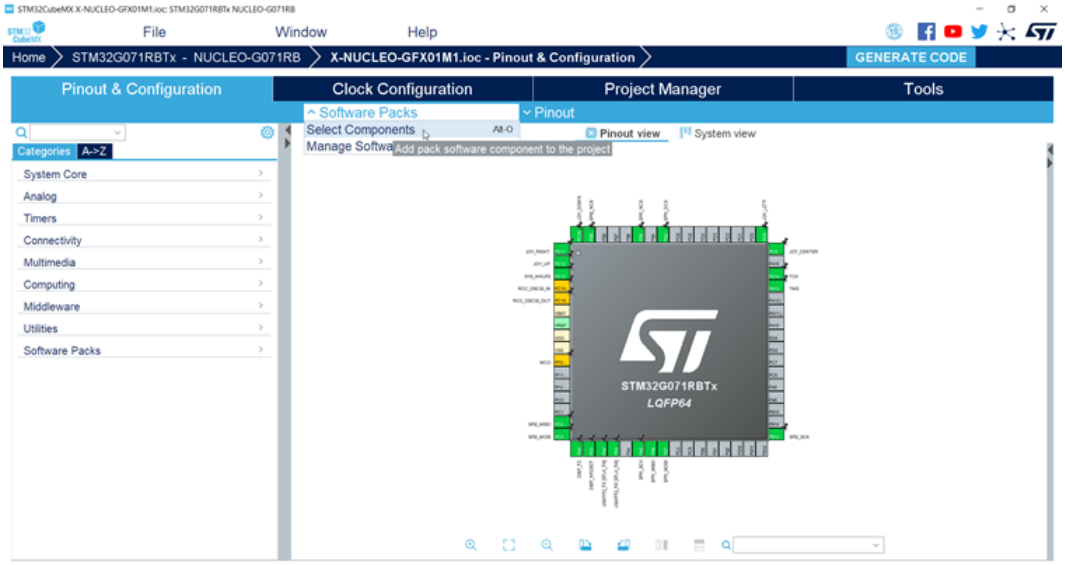
From the window, the user can set up the project by adding one or more additional software and peripherals, and configuring the clock.

Figure 8. **STM32CubeMX *Pinout & Configuration* window**



Step 4. Select the [**Software Packs**]>[**Select Components**] menu to add X-CUBE-DISPLAY as additional software to the project.

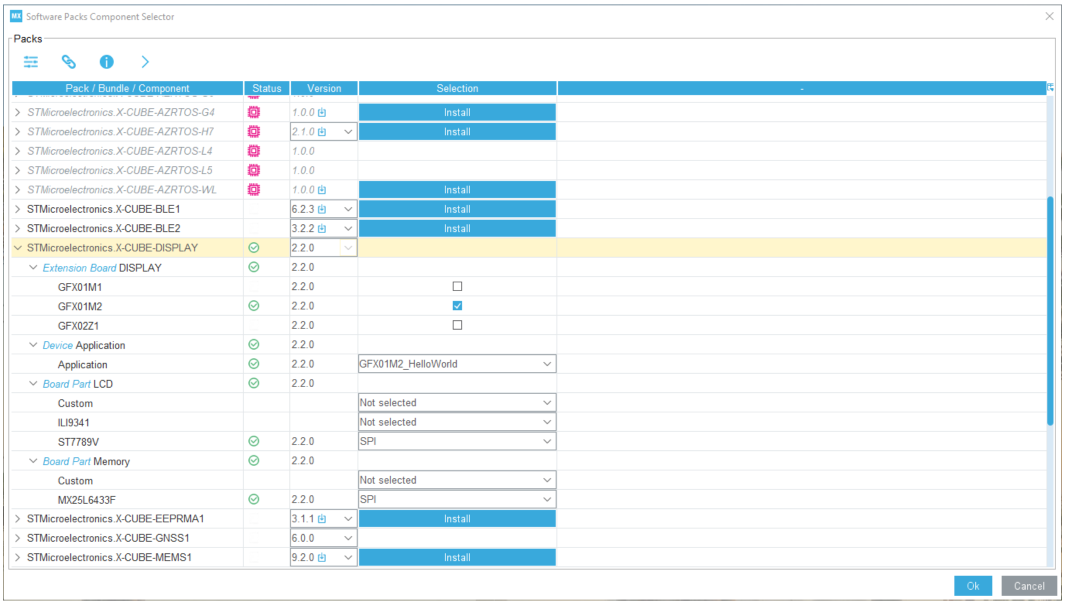
Figure 9. **STM32CubeMX *Software Packs* window**



Step 5. Select a display expansion board from the [**Extension Board DISPLAY**] selector in *Software Packs Component Selector* and resolve its dependencies with the support of STM32CubeMX.

In the example illustrated in Figure 10, expansion board X-NUCLEO-GFX01M1 is selected.

Figure 10. STM32CubeMX *Software Packs Component Selector* window



Software Packs Component Selector

Packs

Pack / Bundle / Component	Status	Version	Selection
STMicroelectronics.X-CUBE-AZRTOS-G4	▣	1.0.0	Install
STMicroelectronics.X-CUBE-AZRTOS-H7	▣	2.1.0	Install
STMicroelectronics.X-CUBE-AZRTOS-L4	▣	1.0.0	
STMicroelectronics.X-CUBE-AZRTOS-L5	▣	1.0.0	
STMicroelectronics.X-CUBE-AZRTOS-WL	▣	1.0.0	Install
STMicroelectronics.X-CUBE-BLE1		6.2.3	Install
STMicroelectronics.X-CUBE-BLE2		3.2.2	Install
STMicroelectronics.X-CUBE-DISPLAY	✓	2.2.0	
Extension Board DISPLAY	✓	2.2.0	
GFX01M1		2.2.0	□
GFX01M2	✓	2.2.0	☑
GFX02Z1		2.2.0	□
Device Application	✓	2.2.0	
Application	✓	2.2.0	GFX01M2_HelloWorld
Board Part LCD	✓	2.2.0	
Custom			Not selected
ILI9341			Not selected
ST7789V	✓	2.2.0	SPI
Board Part Memory	✓	2.2.0	
Custom			Not selected
MX25L6433F	✓	2.2.0	SPI
STMicroelectronics.X-CUBE-EEPRMA1		3.1.1	Install
STMicroelectronics.X-CUBE-GNSS1		6.0.0	
STMicroelectronics.X-CUBE-MEMS1		9.2.0	Install

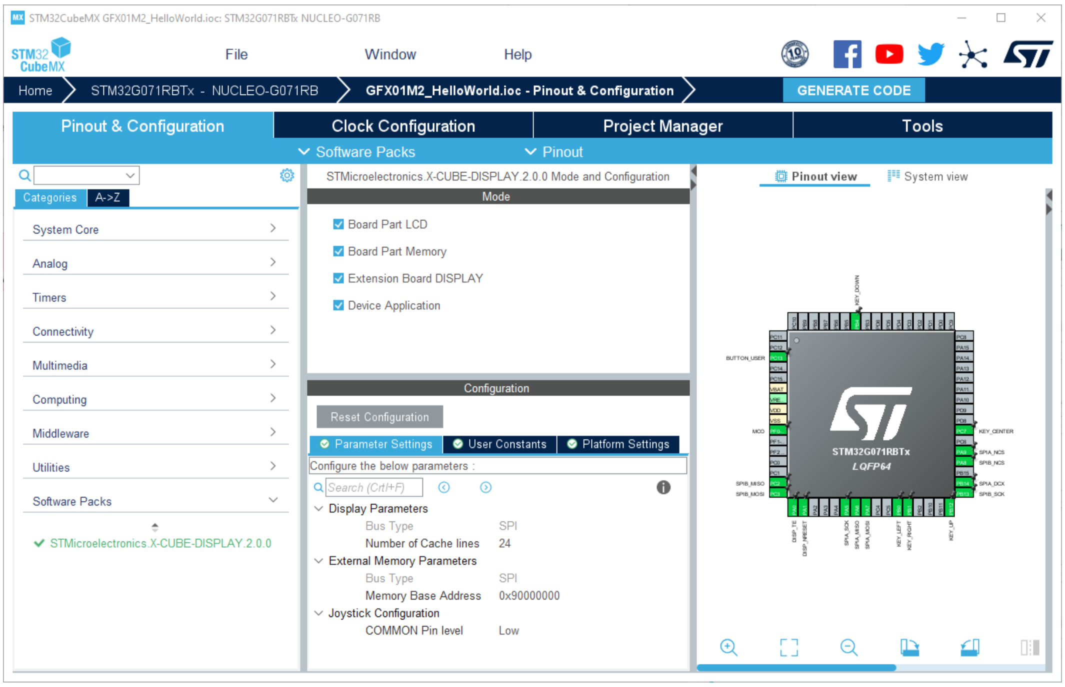
Step 6. Select the [**Device Application**] for the selected display expansion board among the possible ones automatically filtered and proposed by STM32CubeMX.

In the example illustrated in Figure 10, application GFX01M2_HelloWorld is proposed for expansion board X-NUCLEO-GFX01M2.

Step 7. Press the [**OK**] button to return to STM32CubeMX *Pinout* view.

To enable and configure the selected X-CUBE-DISPLAY modules ([**Extension Board GFX01M1**], [**Device Application**], or both), select [**STMicroelectronics.X-CUBE-DISPLAY.2.0.0**] in *Software Packs* from the *Categories* list.

Figure 11. STMicroelectronics.X-CUBE-DISPLAY selection



STMicroelectronics.X-CUBE-DISPLAY.2.0.0 Mode and Configuration

Mode

☑ Board Part LCD
☑ Board Part Memory
☑ Extension Board DISPLAY
☑ Device Application

Configuration

Reset Configuration

Parameter Settings | User Constants | Platform Settings

Configure the below parameters :

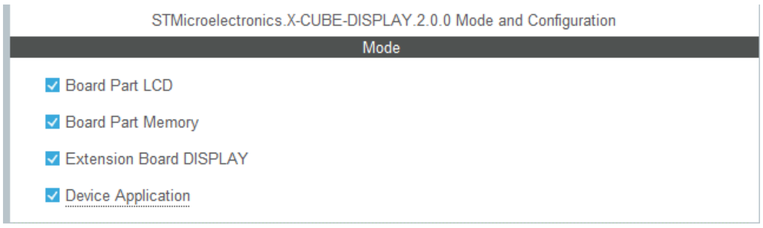
Display Parameters
- Bus Type: SPI
- Number of Cache lines: 24

External Memory Parameters
- Bus Type: SPI
- Memory Base Address: 0x90000000

Joystick Configuration
- COMMON Pin level: Low

Step 8. Enable display modules

Figure 12. Display modules enabled



STMicroelectronics.X-CUBE-DISPLAY.2.0.0 Mode and Configuration

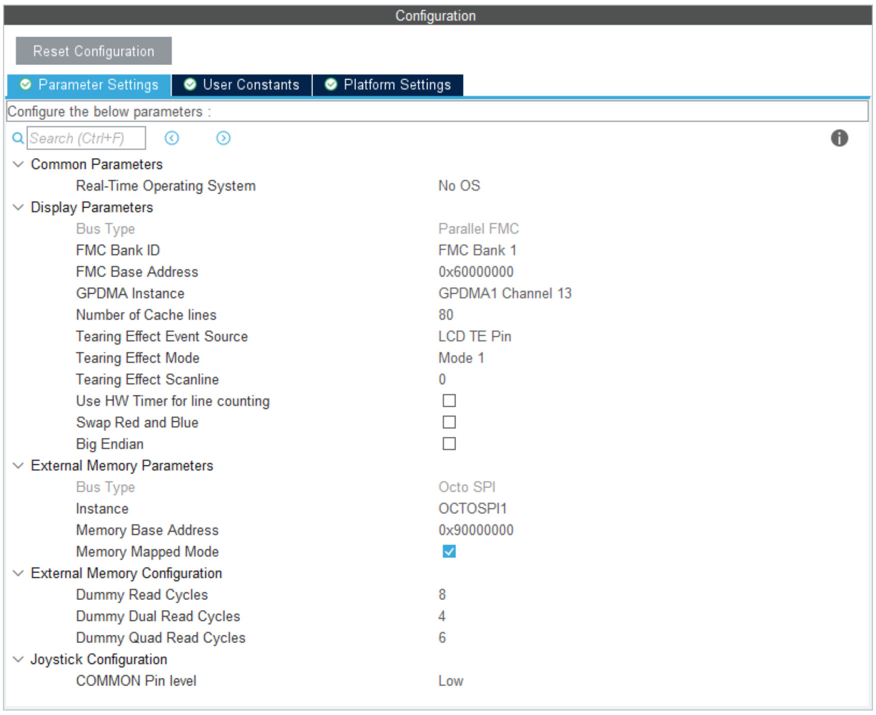
Mode

☑ Board Part LCD
☑ Board Part Memory
☑ Extension Board DISPLAY
☑ Device Application

Step 9. Configure the selected interfaces: LCD, memory, and keys.

Step 9a. Parameters for FMC display and Quad-SPI flash memory.

Figure 13. FMC display and Quad-SPI flash memory parameters

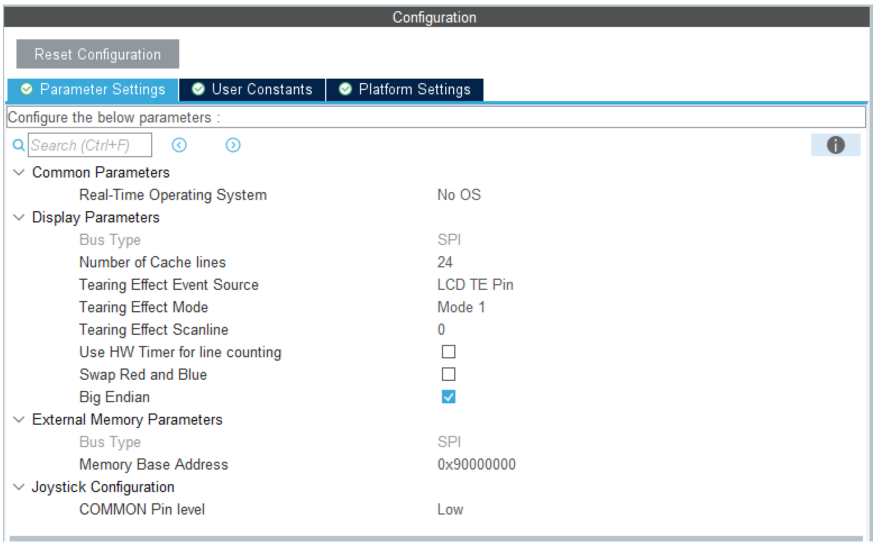


Configure the below parameters :

Common Parameters	
Real-Time Operating System	No OS
Display Parameters	
Bus Type	Parallel FMC
FMC Bank ID	FMC Bank 1
FMC Base Address	0x60000000
GPDMA Instance	GPDMA1 Channel 13
Number of Cache lines	80
Tearing Effect Event Source	LCD TE Pin
Tearing Effect Mode	Mode 1
Tearing Effect Scanline	0
Use HW Timer for line counting	□
Swap Red and Blue	□
Big Endian	□
External Memory Parameters	
Bus Type	Octo SPI
Instance	OCTOSPI1
Memory Base Address	0x90000000
Memory Mapped Mode	☑
External Memory Configuration	
Dummy Read Cycles	8
Dummy Dual Read Cycles	4
Dummy Quad Read Cycles	6
Joystick Configuration	
COMMON Pin level	Low

Step 9b. Parameters for SPI display and SPI flash memory.

Figure 14. SPI display and SPI flash memory parameters

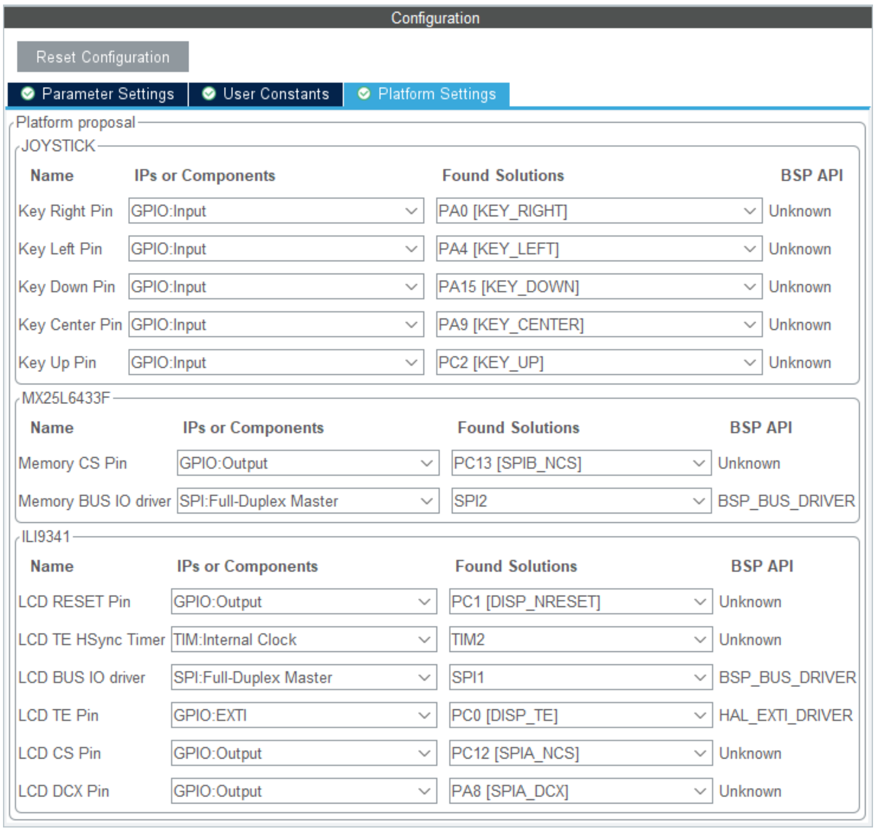


Configure the below parameters :

Common Parameters	
Real-Time Operating System	No OS
Display Parameters	
Bus Type	SPI
Number of Cache lines	24
Tearing Effect Event Source	LCD TE Pin
Tearing Effect Mode	Mode 1
Tearing Effect Scanline	0
Use HW Timer for line counting	□
Swap Red and Blue	□
Big Endian	☑
External Memory Parameters	
Bus Type	SPI
Memory Base Address	0x90000000
Joystick Configuration	
COMMON Pin level	Low

Step 9c. Platform settings.

Figure 15. **Platform settings**



Platform proposal

JOYSTICK

Name	IPs or Components	Found Solutions	BSP API
Key Right Pin	GPIO:Input	PA0 [KEY_RIGHT]	Unknown
Key Left Pin	GPIO:Input	PA4 [KEY_LEFT]	Unknown
Key Down Pin	GPIO:Input	PA15 [KEY_DOWN]	Unknown
Key Center Pin	GPIO:Input	PA9 [KEY_CENTER]	Unknown
Key Up Pin	GPIO:Input	PC2 [KEY_UP]	Unknown

MX25L6433F

Name	IPs or Components	Found Solutions	BSP API
Memory CS Pin	GPIO:Output	PC13 [SPIB_NCS]	Unknown
Memory BUS IO driver	SPI:Full-Duplex Master	SPI2	BSP_BUS_DRIVER

ILI9341

Name	IPs or Components	Found Solutions	BSP API
LCD RESET Pin	GPIO:Output	PC1 [DISP_NRESET]	Unknown
LCD TE HSync Timer	TIM:Internal Clock	TIM2	Unknown
LCD BUS IO driver	SPI:Full-Duplex Master	SPI1	BSP_BUS_DRIVER
LCD TE Pin	GPIO:EXTI	PC0 [DISP_TE]	HAL_EXTI_DRIVER
LCD CS Pin	GPIO:Output	PC12 [SPIA_NCS]	Unknown
LCD DCX Pin	GPIO:Output	PA8 [SPIA_DCX]	Unknown

Revision history

Table 3. **Document revision history**

Date	Revision	Changes
18-Dec-2020	1	Initial release.
1-Dec-2021	2	Extended the document scope to the support of the X-NUCLEO-GFX01M2 and X-NUCLEO-GFX02Z1 display expansion boards and their associated examples: • Updated *Introduction*, *Package description*, and *Starting a new project* • Added *GFX01M2_HelloWorld application* and *GFX02Z1_HelloWorld application*
2-Sep-2022	3	Updated for X-CUBE-DISPLAY version 2.2.0 with the addition of examples running on the FreeRTOS™ and Azure® RTOS ThreadX middleware: • Updated Introduction • Updated Figure 10, Figure 13, Figure 14, and Figure 15 in Starting a new project • Added GFX01M2_FreeRTOS application, GFX02Z1_FreeRTOS application, and GFX02Z1_ThreadX application

Contents

List of tables

List of figures