STM32CubeMX2 documentation

Release 1.1.2

Oct 13, 2026

Table of contents

1 STM32CubeMX2 documentation **4**
1.1 Introduction . . . 4

2 How to get started with STM32CubeMX2 **6**
2.1 LED toggle example . . . 6
2.2 I²C example . . . 17
2.3 Additional articles . . . 20

3 How to start **21**
3.1 System requirements . . . 21
3.2 Installation . . . 22
3.3 Uninstallation . . . 31
3.4 Settings and preferences . . . 37
3.5 Navigation and page structure . . . 38
3.6 How to start from example projects . . . 43

4 User manual **49**
4.1 Home . . . 49
4.2 Project settings . . . 55
4.3 Clock . . . 58
4.4 Pinout . . . 62
4.5 Menu bar and left bar . . . 67
4.6 Peripherals . . . 70
4.7 DMA . . . 73
4.8 NVIC . . . 76
4.9 EXTI . . . 77
4.10 Middleware and utilities . . . 77
4.11 HW Parts . . . 80
4.12 IDE project generation . . . 82
4.13 Pack manager . . . 89
4.14 Code preview . . . 90

5 Command lines interface **92**
5.1 CLI: clock . . . 92
5.2 CLI: composer . . . 95
5.3 CLI: data-analytics . . . 100
5.4 CLI: dma . . . 101
5.5 CLI: exti . . . 103
5.6 CLI: finder . . . 104
5.7 CLI: global-services . . . 105
5.8 CLI: hw-platform . . . 106
5.9 CLI: ide-project . . . 106
5.10 CLI: middleware . . . 113
5.11 CLI: nvic . . . 114
5.12 CLI: pack-manager . . . 116
5.13 CLI: parts . . . 117
5.14 CLI: peripherals . . . 119
5.15 CLI: pinout . . . 121
5.16 CLI: preferences . . . 125
5.17 CLI: project . . . 126
5.18 CLI: sw-config . . . 130
5.19 CLI: utilities . . . 133

6 How to **135**
6.1 How to configure a clock . . . 135

6.2 How to select an IDE and generate code . . . 137
6.3 How to reserve a GPIO pin . . . 138
6.4 How to access peripherals configuration . . . 138
6.5 How to access hardware parts . . . 139
6.6 How to access middleware & utilities . . . 141
6.7 How to configure DMA . . . 142
6.8 How to configure EXTI . . . 144
6.9 How to configure NVIC . . . 145
6.10 How is errors and conflicts propagation managed ? . . . 146

7 Release Note **157**
7.1 Release details . . . 157
7.2 Release limitations . . . 158

8 FAQ **161**
8.1 Frequently asked questions . . . 161

9 Community **163**
9.1 Support link . . . 163
9.2 Community link . . . 163
9.3 Knowledge articles link . . . 163

1 STM32CubeMX2 documentation

Welcome to the documentation portal for **STM32CubeMX2.**

This document covers the **STM32CubeMX2** tool only.

For information about **STM32CubeMX**, see the STM32CubeMX user manual.

To understand which **STM32** series are supported by **STM32CubeMX** or **STM32CubeMX2**, see the STM32CubeMX page on ST website.

1.1 Introduction

STM32CubeMX2 is a software tool for **STM32** products. It is available as a standalone application and offers several important features:

- **Project creation and opening**.
- **Fast configuration:** Set up **MCU** pins and the clock tree.
- **Peripheral setup:** Add and configure peripherals.
- **Middleware and Utilities:** View and activate default middleware and utilities. Add additional middleware and utilities if needed.
- **Parts** View and activate mounted parts and their configurations. Add additional standalone parts if needed.
- **Code preview:** View generated code before project generation to verify configurations or copy code snippets.
- **Project generation:** Generate a complete project with the required libraries and code initialization.
- **Pack management:** Check, install, uninstall, and update software packs.

Note

Software packs are bundles of software components, libraries, drivers, middleware, and utilities. They facilitate development for specific hardware platforms. These packs use the open standard **Open-CMSIS** format. This format provides a uniform structure and management.

How to get started with STM32CubeMX2

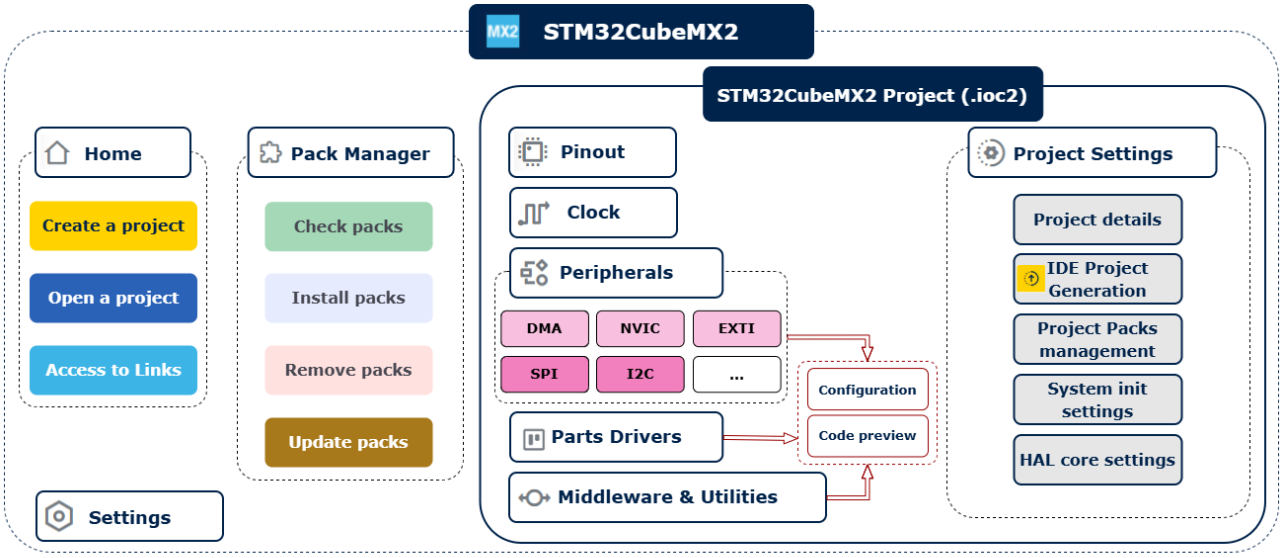


Fig. 1: STM32CubeMX2 overview

2 How to get started with STM32CubeMX2

2.1 LED toggle example

2.1.1 Project creation

Device selection

From the **Home** view, click on **Board** to initiate a board-based project

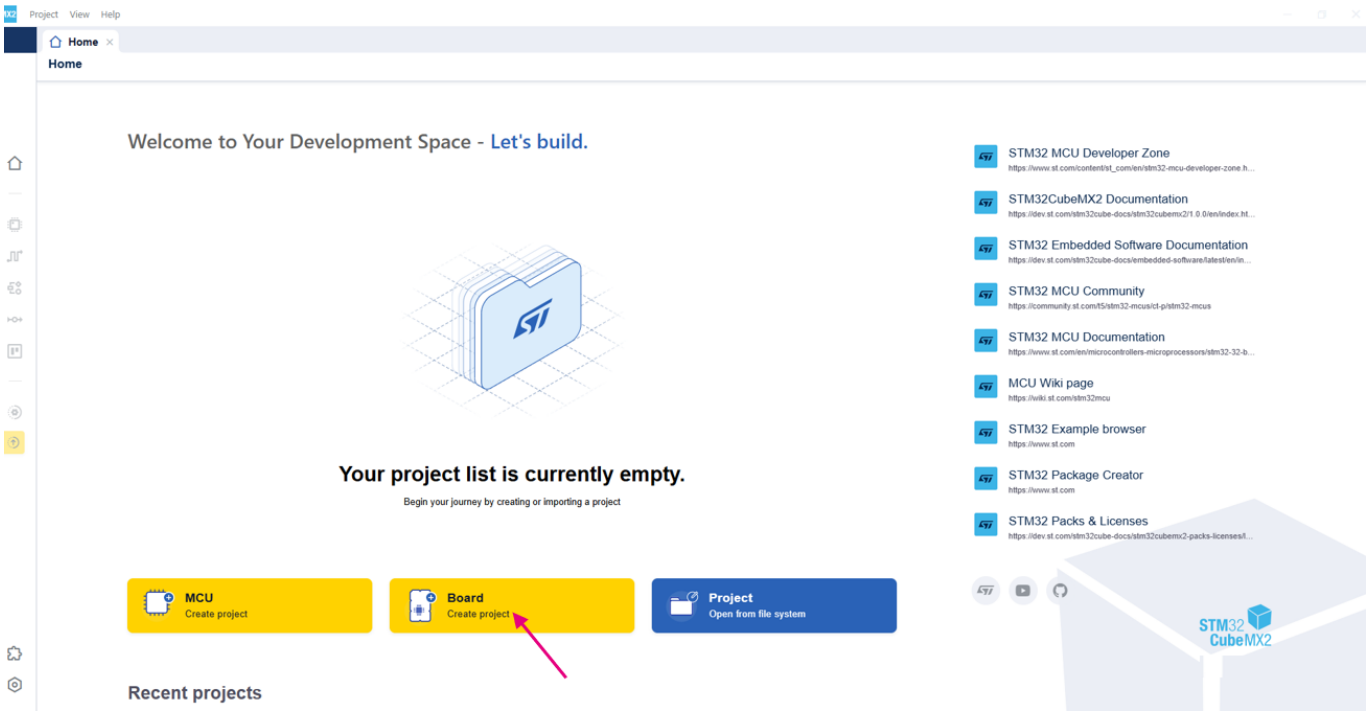


Fig. 1: Create a project from a board

- **Select the board (1):** For this example , we use **NUCLEO-C562RE**.
- **Click Continue to provide the project details (2).**
- **Enter the project name (3):** example **Toggle_LED.**
- **Specify the project location (4):** example **C:\Projects.**
- **Click Automatically Download, Install & Create Project (5).**

After creating the project, open **Project settings** to display and manage the project. This tab contains the following:

- **Project settings (1):** Displays the project settings, including the project name, location, selected device and the software project.
- **IDE project generation settings (2):** Manages the **IDE** project. See the *IDE project generation* for more details.
- **Left bar (3):** contains quick access icons for frequently used features. See *Menu bar and left bar* for further details.
- **Packs management (4):** Manages the installed packs.
- **Global services settings (5): Manages the global services settings.**
  - **HAL System init settings (6):** Manages Code Generation parameters for common System files.

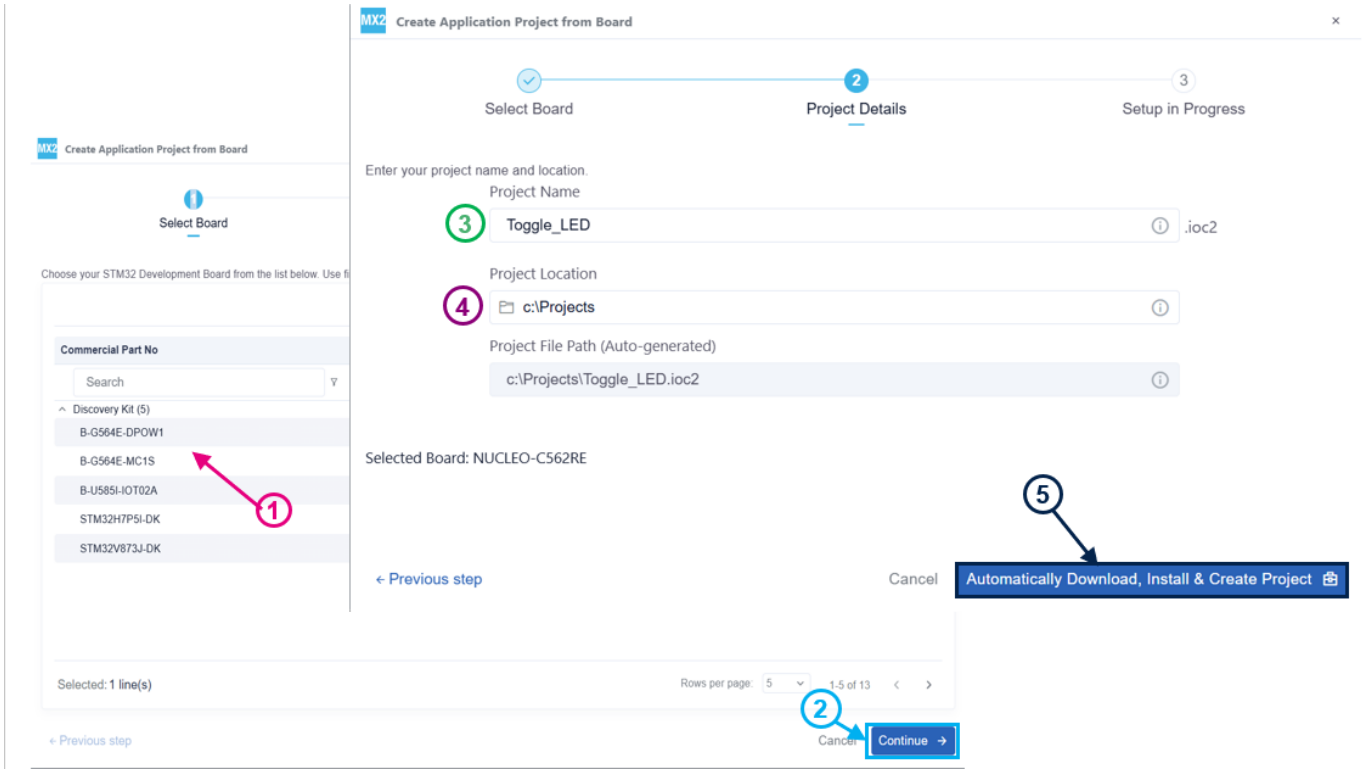


Fig. 2: Project setup

– **HAL Common definitions (7):** Manages Code Generation parameters for Core Files.

For this tutorial, only the pinout configuration is required.

Pinout assignment

- **Open the pinout view (1):** Click on its icon from the left bar.
- **Locate PA5 (2):** Use the search function to find its location. **PA5** is the GPIO that is connected to the LED we want to use on this NUCLEO board.

> **Note**
>
> The Reset Pinout action clears reserved and **GPIO** signals while preserving peripheral-configured signals and user labels.

- **Assign PA5 to GPIO (3):** Right-click the pin and select **Assign signal** > **GPIO.**

> **Note**
>
> The table view can also be used to configure pins. See *Pinout* for further details.

GPIO configuration

To configure GPIO component:

- **Open the configuration (1):** Right-click on the pin and select **Configure GPIO.**
- **Locate the pins PA5 (2):** Extend the **Pins** section to find the pins.
- **Set PA5 mode to output (3):** Extend the configuration then change the mode to **output.**

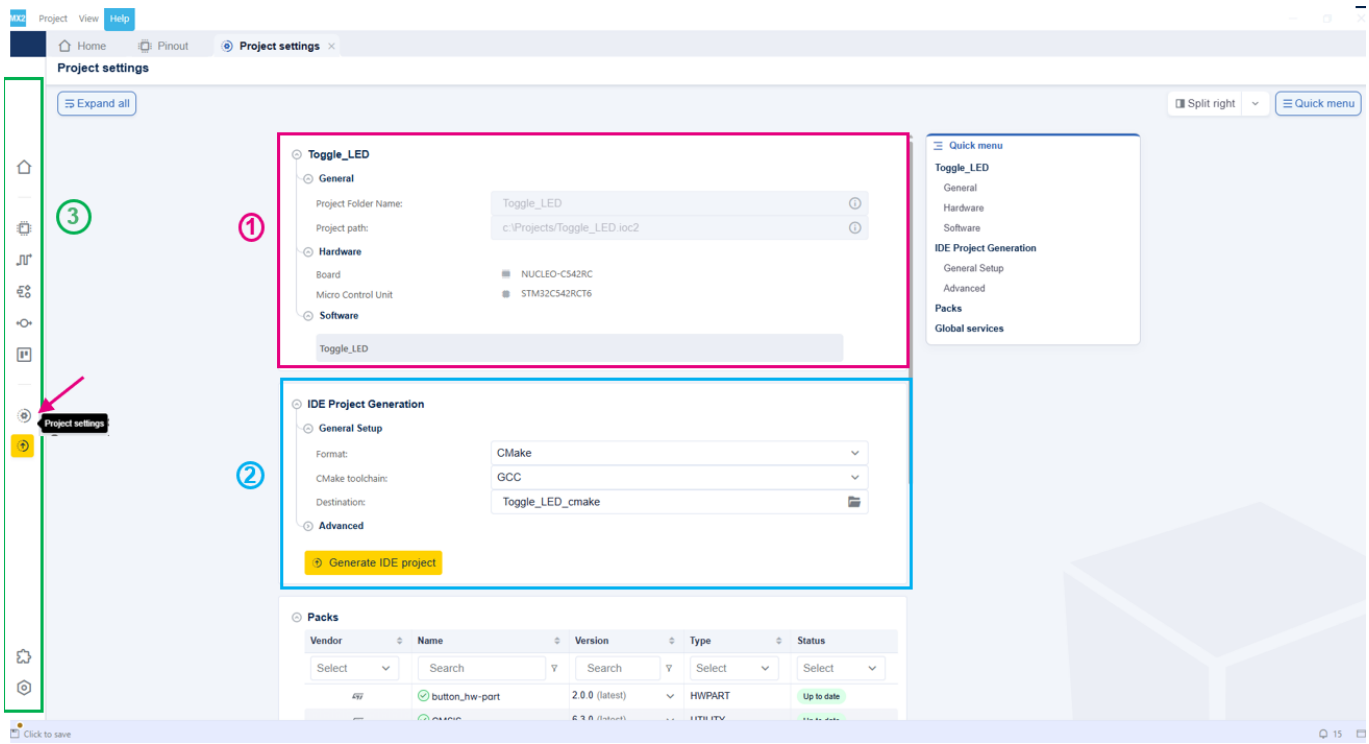


Fig. 3: Project & IDE project settings

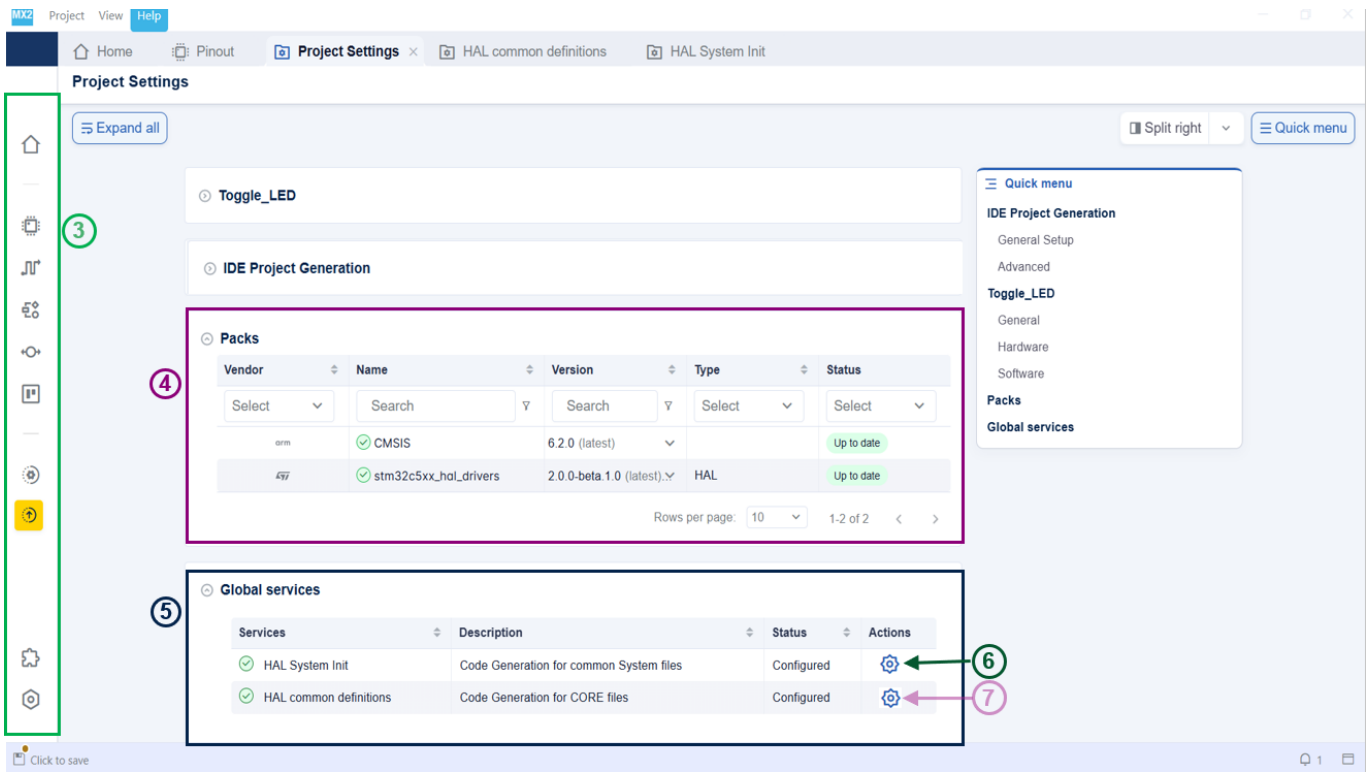


Fig. 4: Packs management & Global services settings

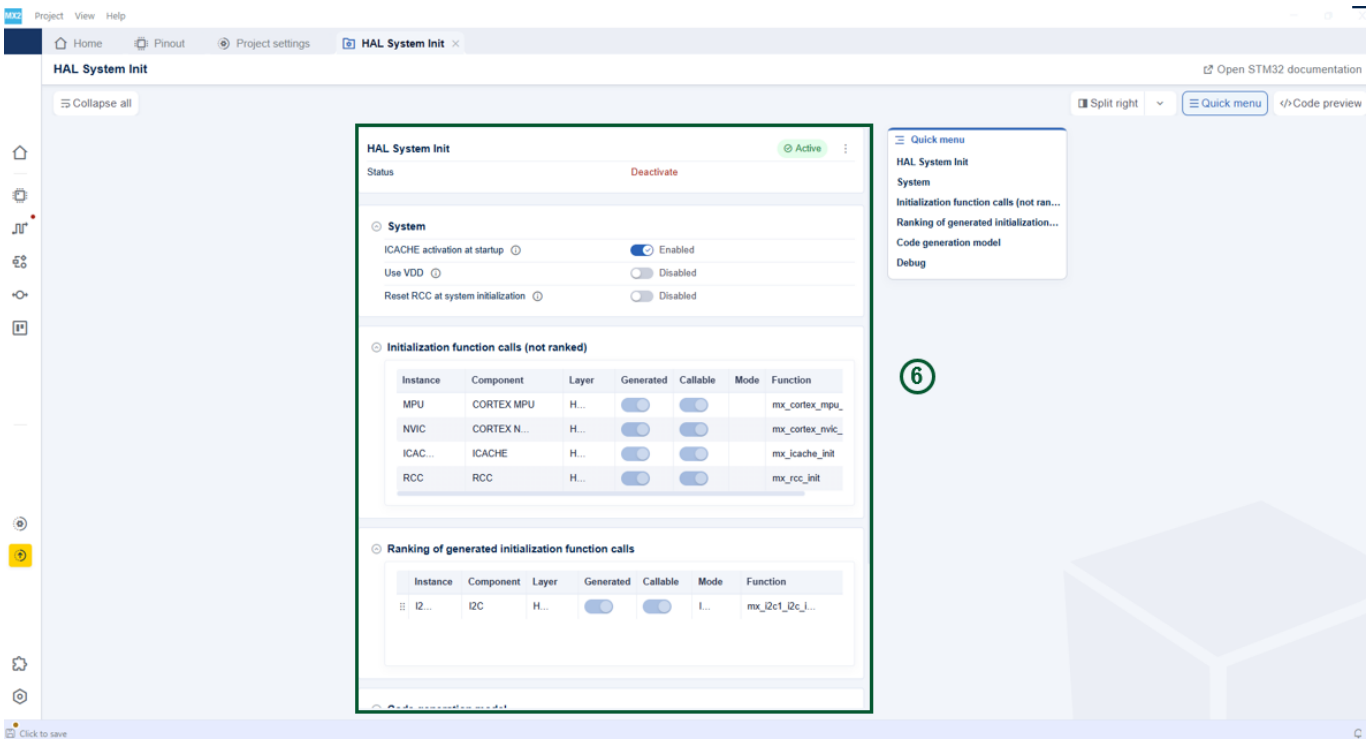


Fig. 5: HAL system init settings

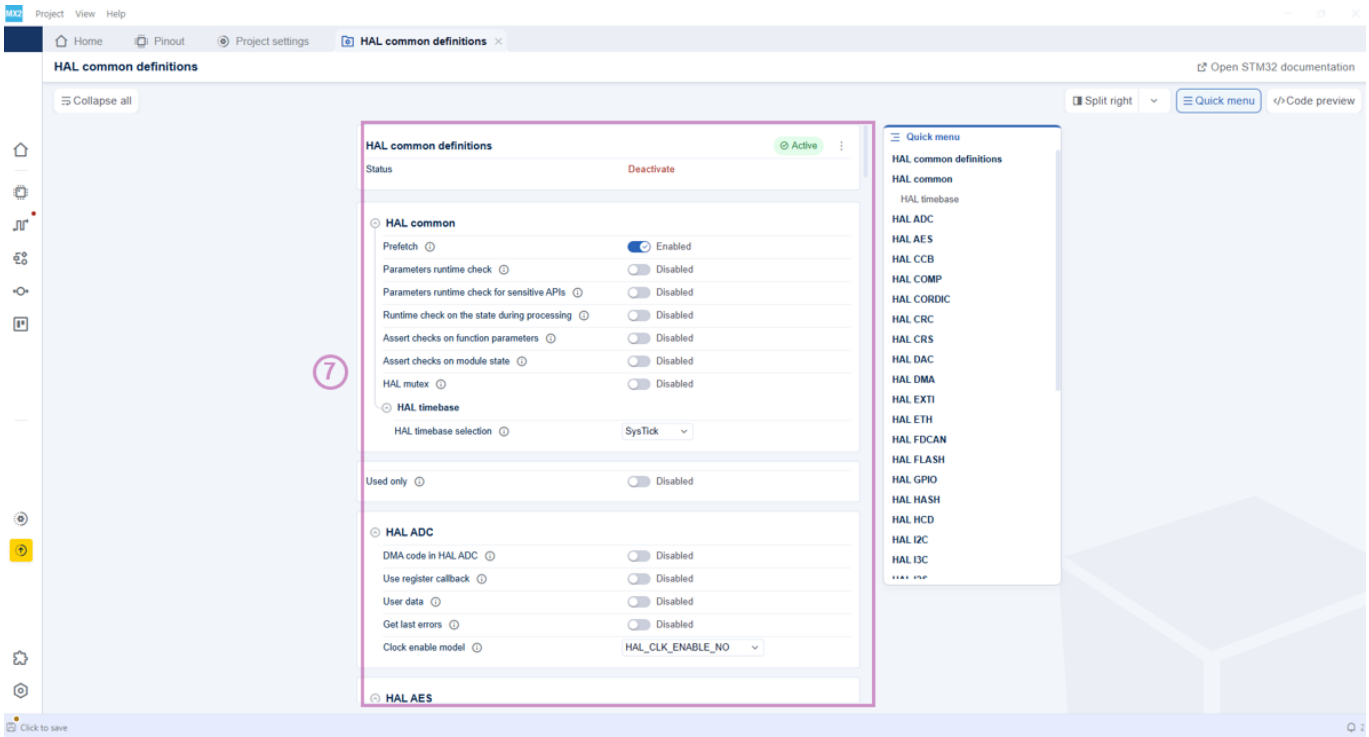


Fig. 6: HAL common definitions

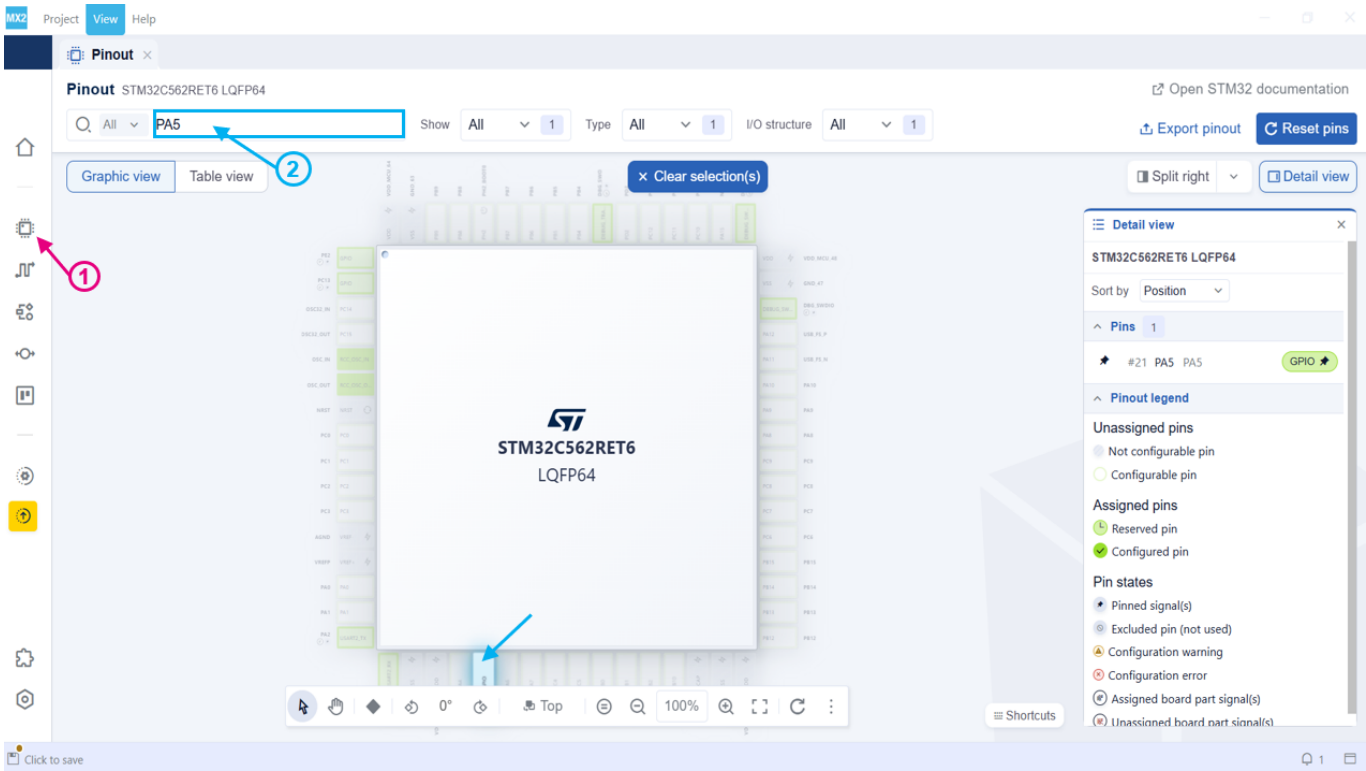


Fig. 7: Pinout view

2.1.2 IDE project generation

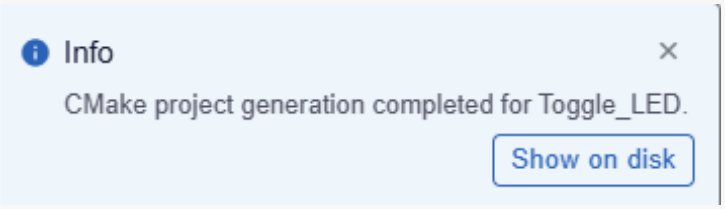
Project generation

After configuring PA5 pins, follow these steps to generate the **IDE** project:

- **Select format (1):** Choose the desired format. Available formats include **IAR Embedded Workbench** for ARM, **CMake** and **Open-CMSIS.** The default is **CMake.**
- **Select toolchain (2):** A default toolchain is automatically created. However, the *Open CMSIS* format currently supports multiple toolchains. In this case, the user can explicitly choose GCC or IAR instead of the default AC6.
- **Select destination (3):** Browse and select the folder for the IDE project. The default path is at the root folder of the created **STM32CubeMX2** project.
- **Generate IDE project (4):** Click the **Generate** button to create the **IDE** project.

Note

- The Show on disk option allows you to view the generated files on your disk when the IDE project generation is completed.



- Check the *IDE project generation* to learn more about the advanced settings.

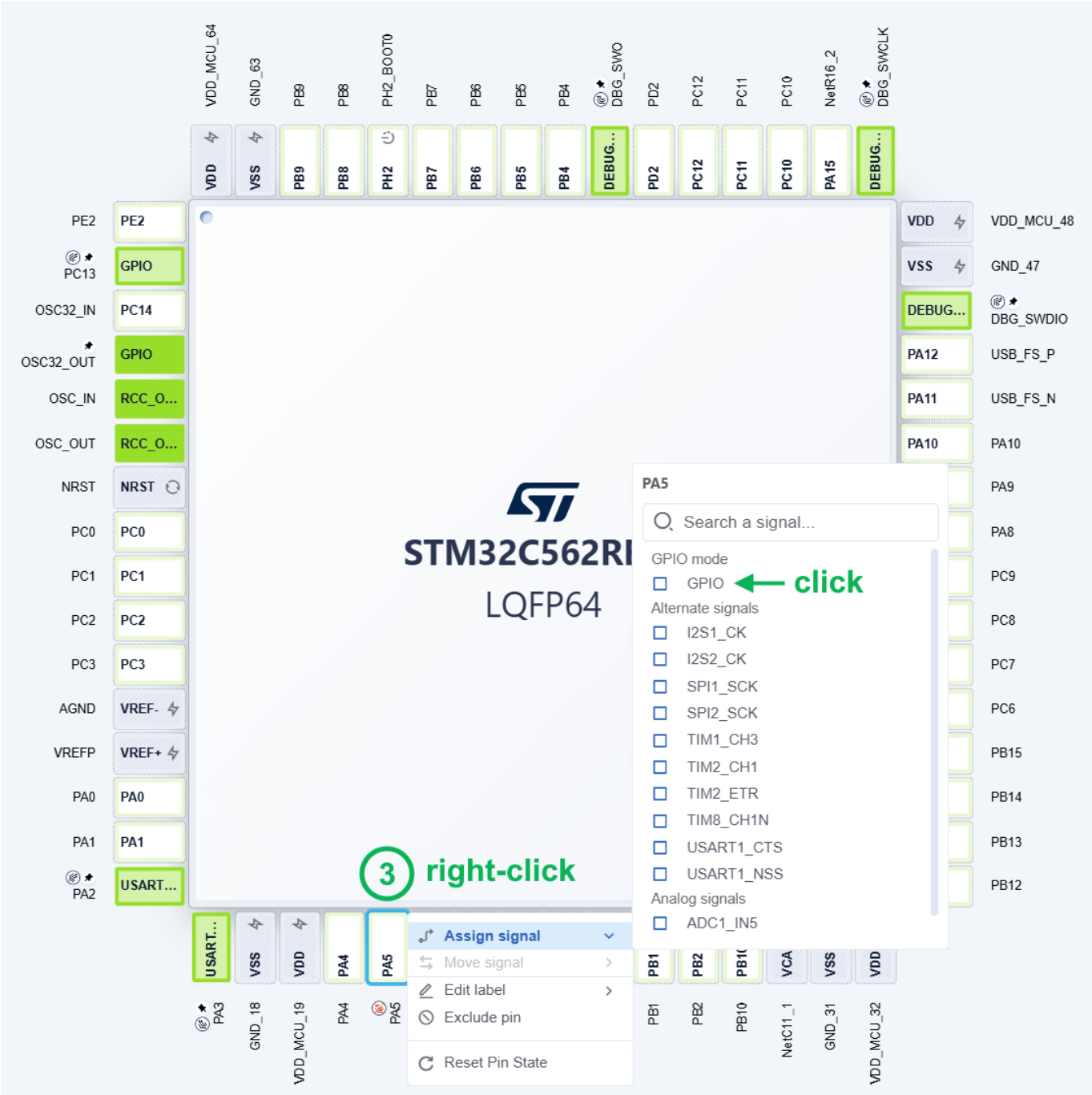


Fig. 8: Pinout configuration

Fig. 9: Access to GPIO configuration

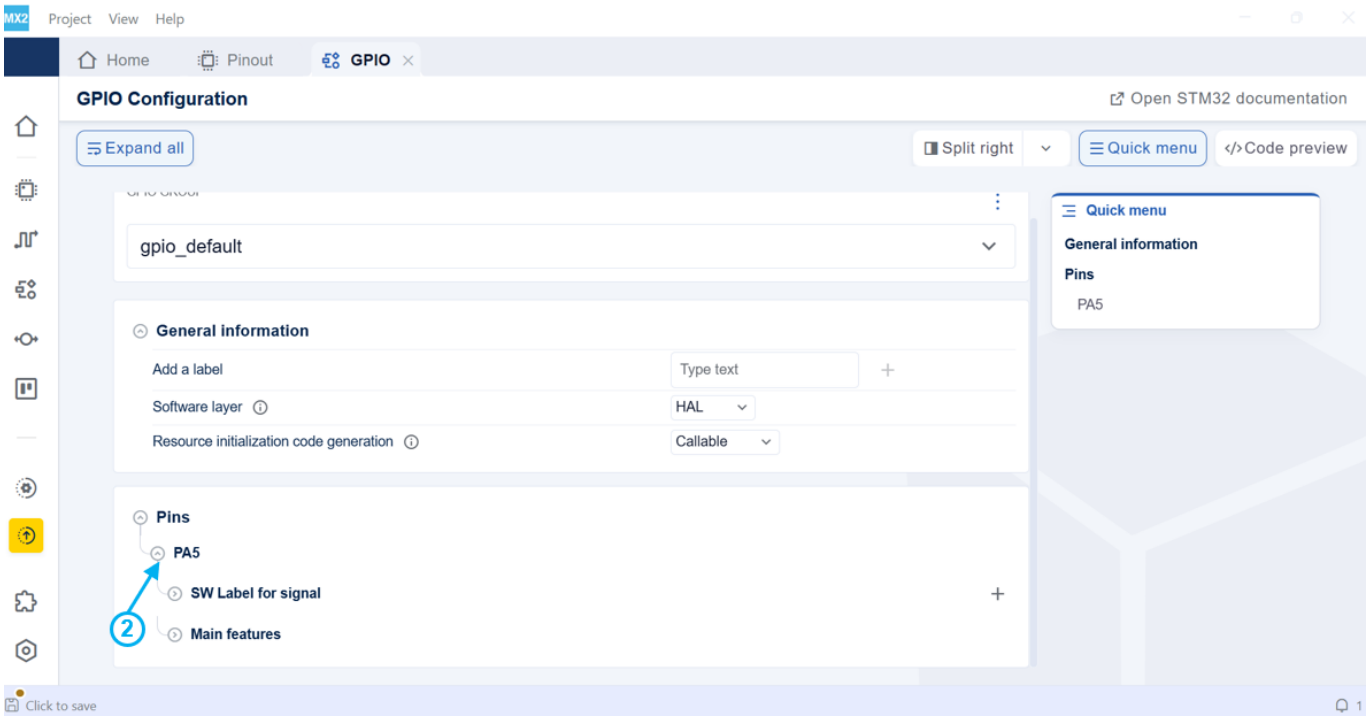


Fig. 10: GPIO configuration

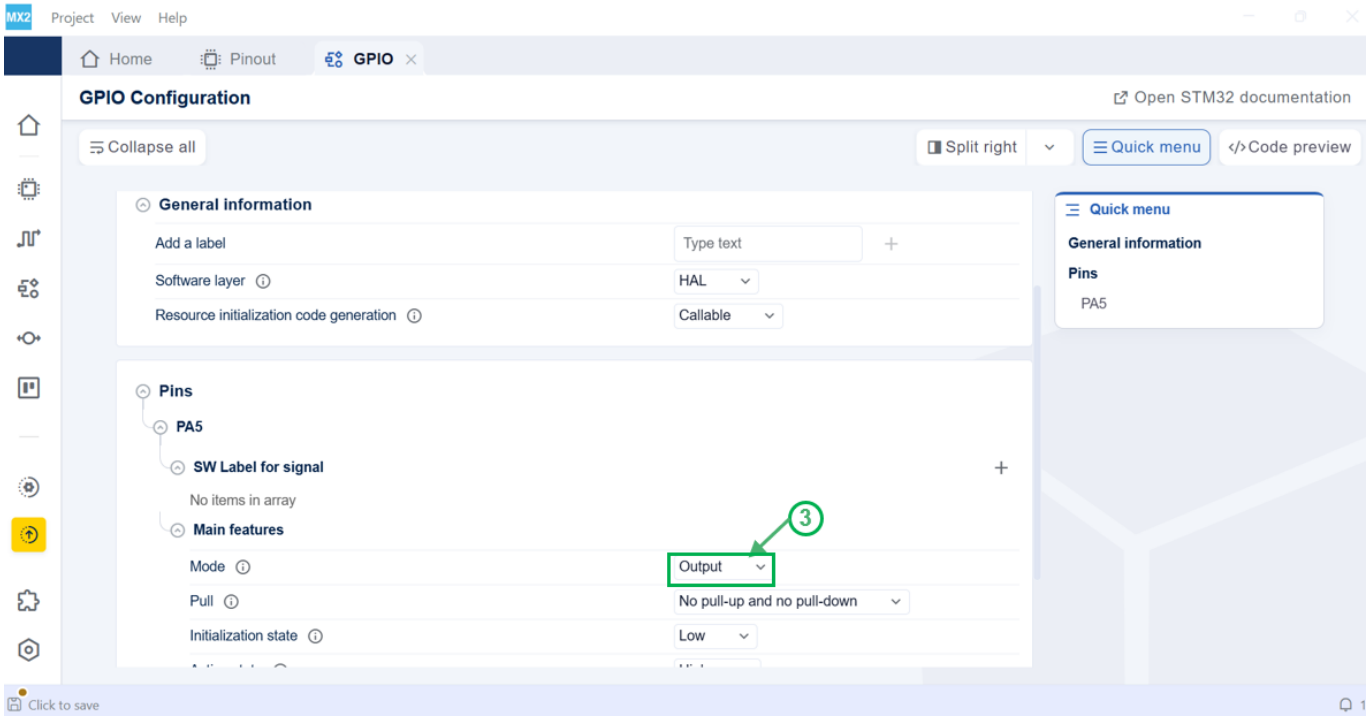


Fig. 11: PA5 pin configuration

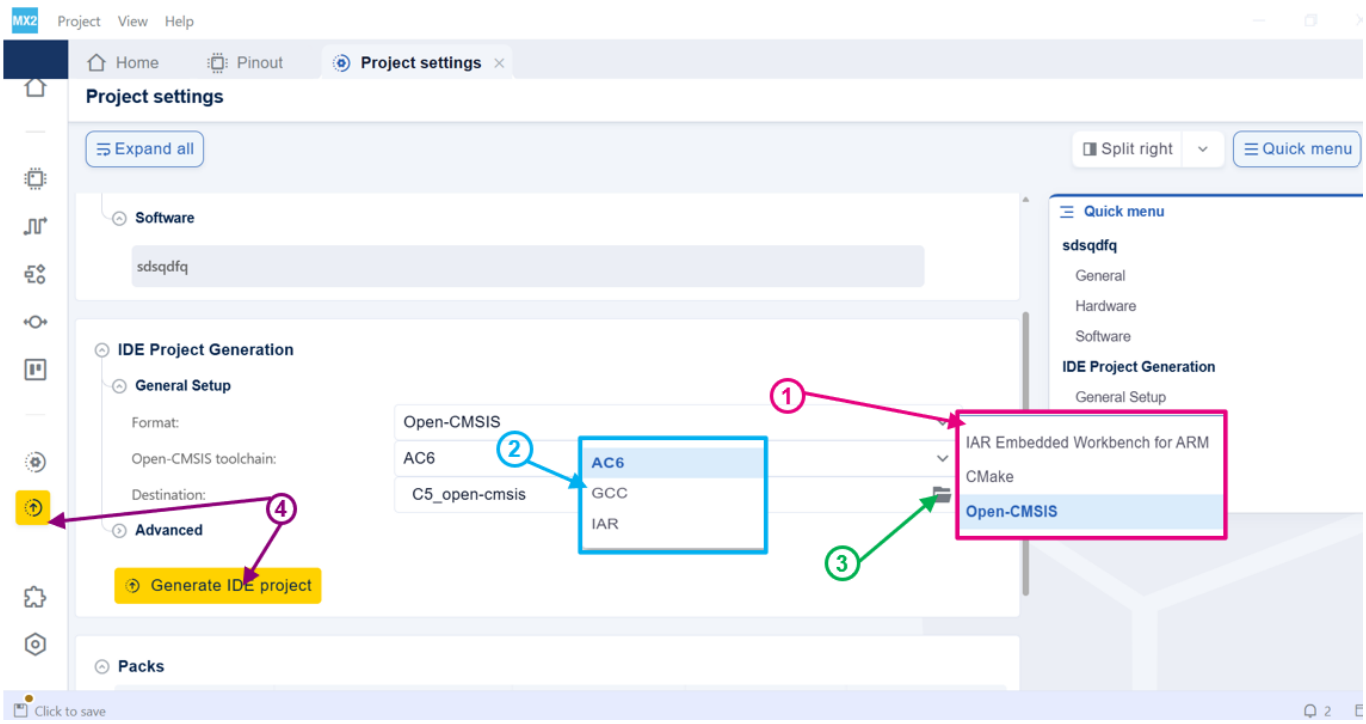


Fig. 12: Project settings

Unified CLI workflow

This section groups all command-line operations used in the tutorial: board-based project creation, pin assignment, GPIO configuration, and IDE project generation.

1) Create the project from a board

Use `cube mx project create-from-board` to create a new STM32CubeMX project based on a board CPN (Commercial Part Number).

```
cube mx project create-from-board \
--cpn NUCLEO-C562RE  \
--project-location /c/Workspace_MX2/MySecondCLIProject \
--project-name Toggle_LED
```

Arguments:

- `--cpn`: Board identifier (e.g. `NUCLEO-C562RE`).
- `--project-location`: Base folder where the project will be created.
- `--project-name`: Logical project name.

After this step, an `.ioc2` file is created, for example:

```
/c/Projects/Toggle_LED.ioc2
```

2) Assign or unassign a pin signal

You can modify pin assignments using the `cube mx pinout` commands.

Unassign a signal from a pin

```
cube mx pinout unassign-signal \
  -p /c/Workspace_MX2/MySecondCLIProject/Toggle_LED.ioc2 \
  --pin PA5 \
  --signal GPIO
```

Arguments:

- `-p`: Path to the `.ioc2` configuration file.
- `--pin`: Pin name (e.g. `PA5`).
- `--signal`: Currently assigned signal type to remove (e.g. `GPIO`).

Assign a signal to a pin

```
cube mx pinout assign-signal \
-p /c/Workspace_MX2/MySecondCLIProject/Toggle_LED.ioc2 \
--pin PA5 --signal GPIO
```

This assigns the `GPIO` function to pin `PA5` in the project configuration.

3) Configure GPIO parameters

Use `cube mx sw-config` to set software configuration parameters, such as pin modes.

```
cube mx sw-config set-parameter-value \
-p /c/Workspace_MX2/MySecondCLIProject/Toggle_LED.ioc2 -r GPIO \
--parameter-path pins/0/basic/mode -v OUTPUT
```

Arguments:

- `-r GPIO`: Selects the GPIO configuration resource.
- `--parameter-path`: Path of the parameter in the configuration tree (for example `pins/0/basic/mode`).
- `-v OUTPUT`: Sets the parameter value (mode) to `OUTPUT` for that pin entry.

> **Note**
>
> The exact `parameter-path` may depend on the project structure (pin index, naming, and mode options).

4) Generate the IDE project

Generate buildable project files using `cube mx ide-project generate`.

```
cube mx ide-project generate \
-p /c/Workspace_MX2/MySecondCLIProject/Toggle_LED.ioc2 \
--format CMake --build-target .debug_GCC+NUCLEO-C562RE \
--source include-packs-from-local --destination /c/Workspace_MX2/MySecondCLIProject/Toggle_LED_
↪cmake
```

Arguments:

- `--format CMake`: Generates a CMake-based project.
- `--build-target`: Name of the build configuration/target (for example `.debug_GCC+NUCLEO-C562RE`).
- `--source include-packs-from-local`: Use installed local packs as source.
- `--destination`: Output folder for the generated CMake project.

After this step, you can open or build the generated project from:

```
/c/Workspace_MX2/MySecondCLIProject/Toggle_LED_cmake
```

Project opening

Navigate to the project's location:

- **Project location (1):** The project is located at **C:\Projects.**

- **Project configuration file (2):** This **.ioc2** file is the main configuration file for the **STM32CubeMX2** project.
- **IDE project folder (3):** Contains the **IDE-specific** files and folders.

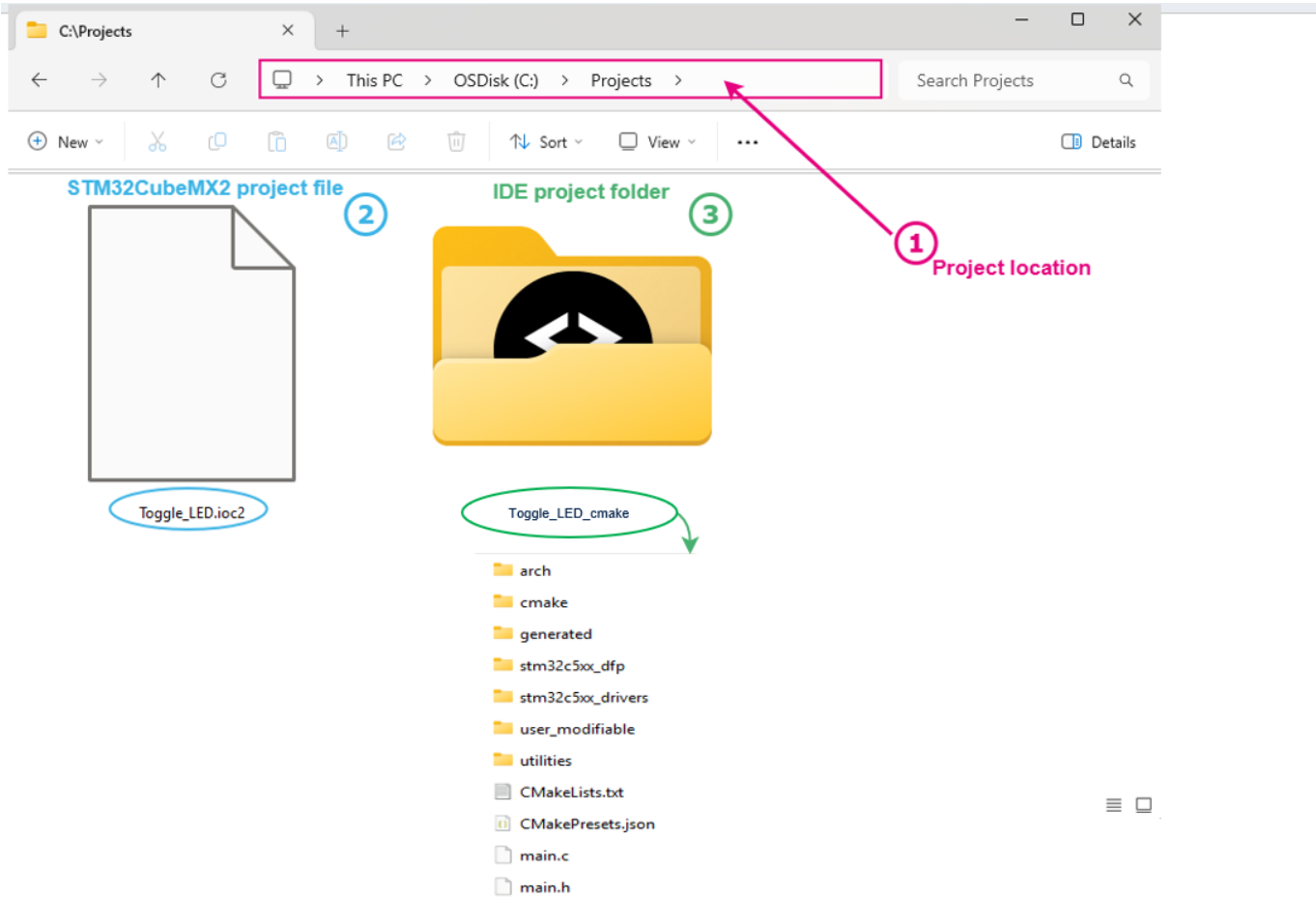


Fig. 13: Project directory

Open Toggle_LED project with **STM32CubeIDE for VSCode**, and edit the **main.c** file.

- **Write the application code (1):** In the main function, add the highlighted code to toggle **GPIO** pins with delays.
- **Build the code (2):** Click the **Build** button to compile the code.
- **Run the code (3):** Click the **Launch** button to execute the code on the target hardware.

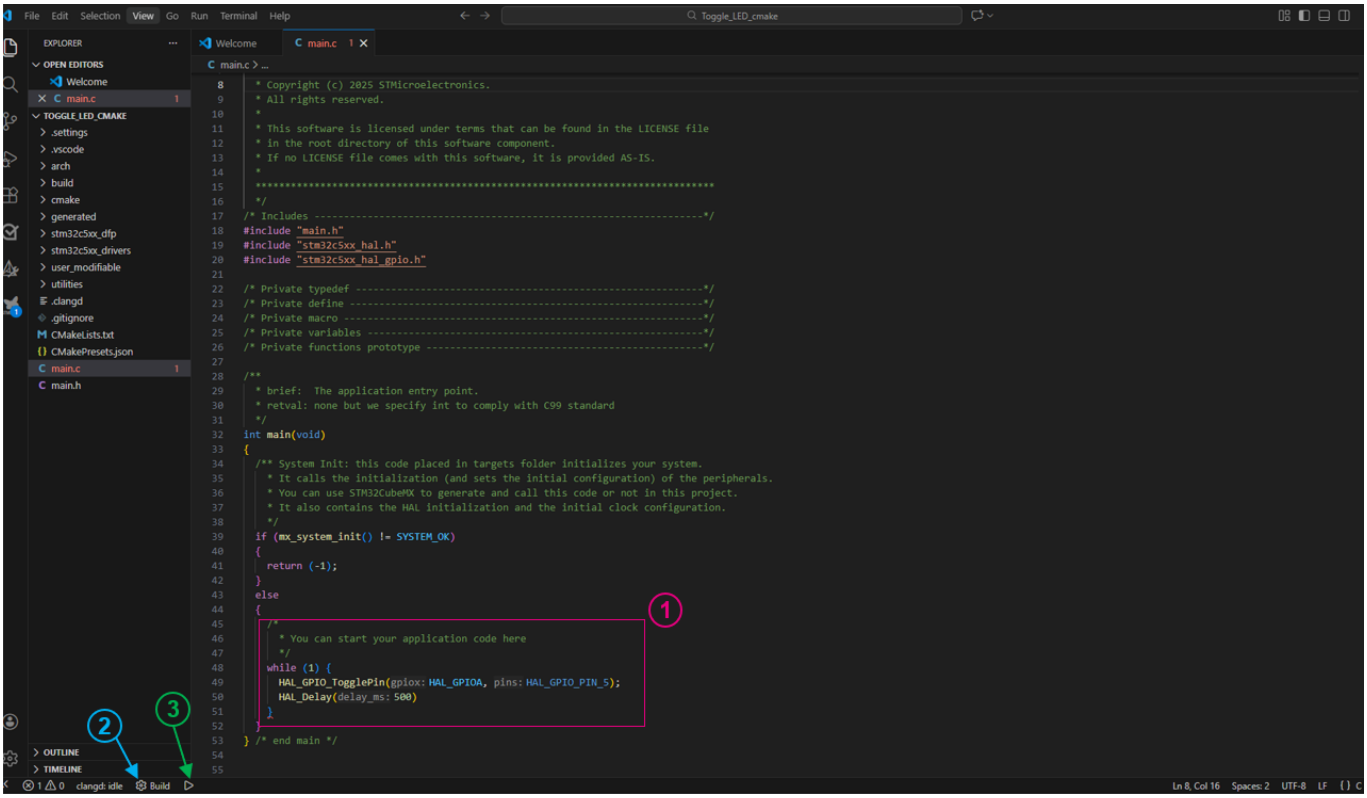


Fig. 14: main.c file

2.2 I²C example

2.2.1 Project creation

create a project by following the steps described in *Create a project from a board* and *Project setup*.

For this demonstration, pinout and clock configurations remain as default.

I²C activation

- **Open the peripheral view (1):** Click its icon from the left bar.
- **Extend connectivity section (2).**
- **Enable I²C1 (3):** Locate **I²C1** and change its status from **Disabled** to **I²C** by selecting the option from the dropdown menu.

> **Note**
>
> Refer to *Peripherals* for more details on how to activate peripherals.

> **Note**
>
> The system displays a notification indicating that the peripheral **I2C1** signals are assigned to specific pins.

I²C configuration

Once enabled, the user can open the software configuration following these steps:

- **Open the I²C1 Settings (1).**
- **Update the resources allocation (2):** Change the default assignment if needed.
- **Update the configuration (3):** Expend the different sections to list all available parameters.

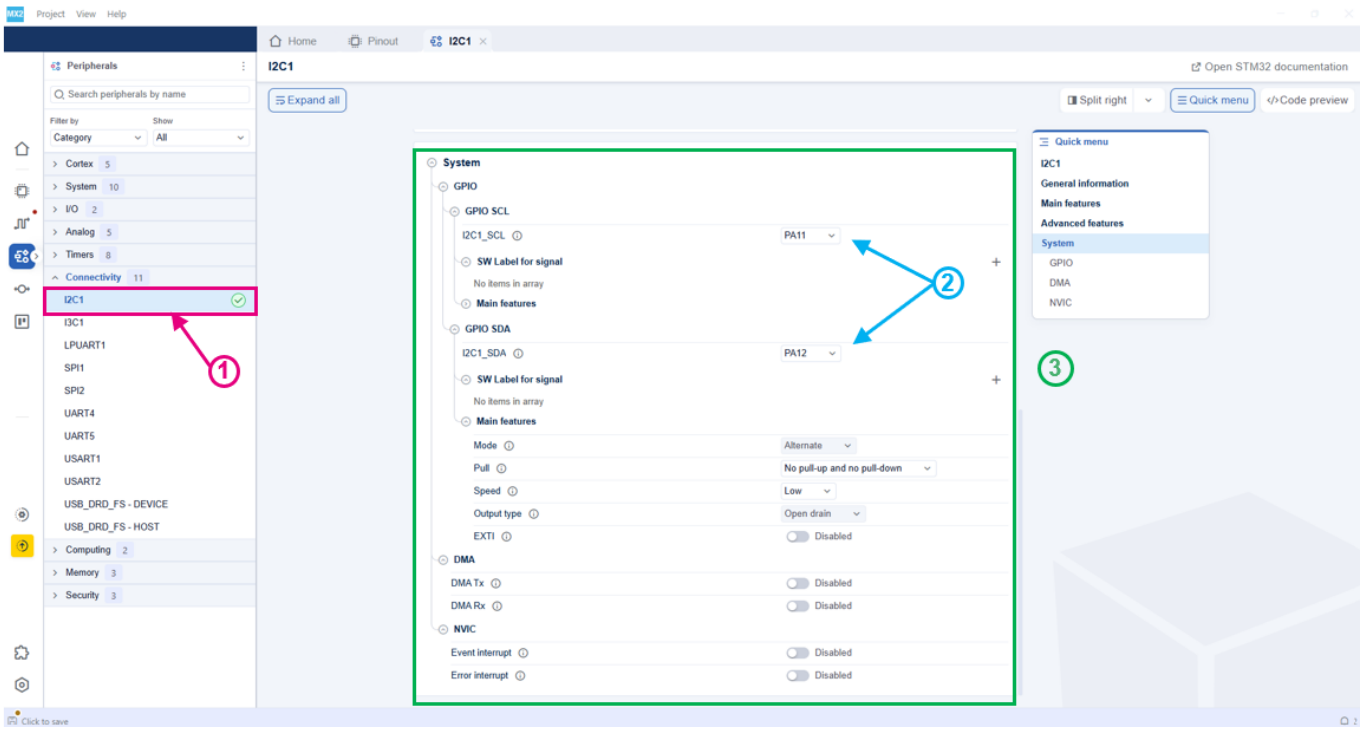


Fig. 15: Access I²C configuration

- **Enable event interruption and error detection (4):** Extend the system section to get the list of interruptions.
- The priority level can be modified either from the **I²C** configuration or from the **NVIC** view.

> **Note**
>
> The *NVIC* view can be accessed from the **Peripheral View** under the **System** category.

2.2.2 IDE project generation

Apply the same steps described in the *IDE project generation* section from *LED toggle example*.

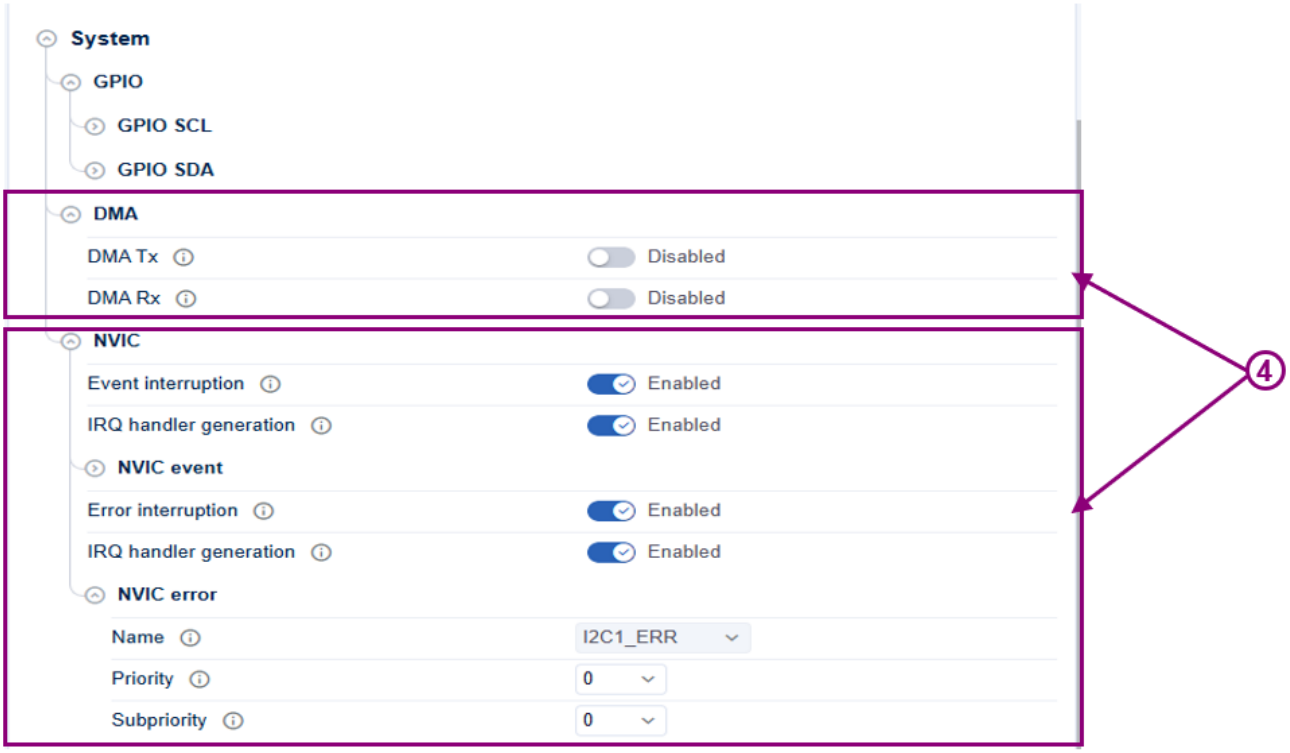


Fig. 16: Enable I²C2 event and error interrupts

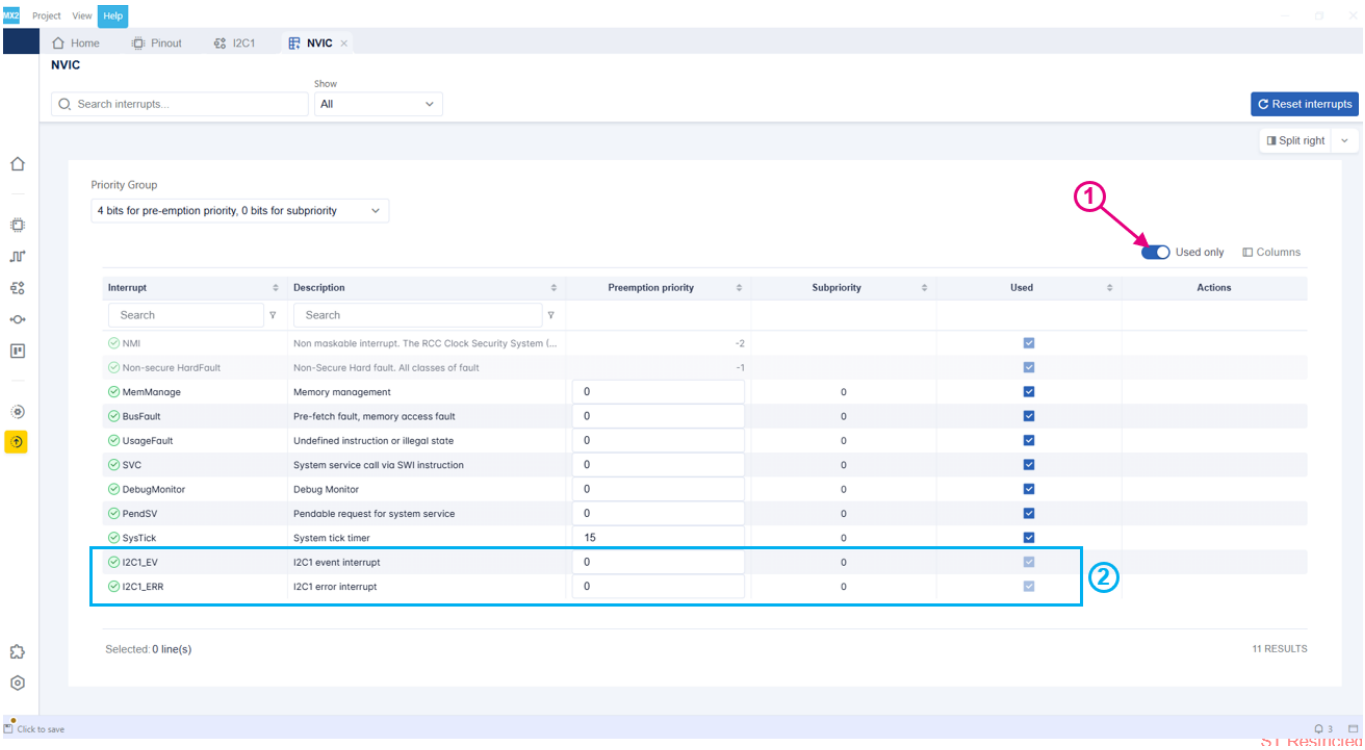


Fig. 17: I²C2 interrupts in NVIC view

2.3 Additional articles

How to create a new project with STM32CubeMX2

3 How to start

3.1 System requirements

3.1.1 Supported operating systems and architectures

- **Windows®**: Tested on both **Windows® 10** and **11 64-bit**.
- **Linux®**: Tested on **Ubuntu® 22.04**, **Ubuntu® 24.04**, **Ubuntu® 26.04**, **Fedora® 43** and **Fedora® 44**.
- **macOS® (x86_64 and aarch64)**: Tested on **macOS® version 15** and **macOS® version 26** on both Intel® and Apple® silicon systems.

> **Note**
>
> - Windows® 64-bit: Intel® Core by default.

3.1.2 Processor (CPU)

- **Minimum configuration**
  - **Windows® / Linux®**: Multi-core x86_64 processor, for example Intel® Core i3, with a frequency of at least 1.6 GHz
  - **macOS®**: Apple® Silicon M1 or newer, or Intel® Core i5
- **Recommended configuration**
  - **Windows® / Linux®**: Intel® Core i5 or equivalent
  - **macOS®**: Apple® silicon M1 or newer, or Intel® Core i7

3.1.3 Memory (RAM)

- Minimum: 8 GB RAM
- Recommended: 16 GB RAM

> **Note**
>
> In typical development environments where multiple applications run in parallel, **STM32CubeMX2** may require more than 10 GB of total system memory. 16 GB RAM or higher is therefore recommended for a comfortable user experience.

3.1.4 Disk space

Users must ensure that:

- There is enough free disk space to install **STM32CubeMX2.**
- There is enough free disk space to install or update several device and software packs.
- There is enough free disk space to create, open and build multiple user projects.
- The drive hosting **STM32CubeMX2** and its data have sufficient free space margin to handle temporary files and future updates.

Warning

STM32CubeMX2 is not compatible with **Ubuntu® 20.04** due to its reliance on a more recent version of **glibc**. Upgrading the **glibc** version on **Ubuntu® 20.04** is complex and may cause issues with other system libraries. Proceed with caution if attempting this upgrade.

3.1.5 Minimal resolution

- **STM32CubeMX2** has been designed for a **15 inches HD screen** (1920*1080 px) in landscape.
- It's possible to use it with a lower resolution or screen size, but your experience may be degraded.

Note

- Using the Zoom can help if STM32CubeMX2 is used with a smaller resolution or screen size.
- The zoom is available in the 'View' menu at the top of the interface.

3.2 Installation

3.2.1 Installation process

Whitelist of executables

The following executables must be whitelisted to ensure proper functioning of **STM32CubeMX2**:

- stm32cubemx2.exe
- cube.exe
- cube-cmsis-scanner.exe
- node.exe
- csolution.exe
- project_extractor.exe

Installation prerequisites for special characters

On some systems, **STM32CubeMX2** may encounter issues if project or working paths contain user profile folders with non-ASCII characters, such as accented characters or non-Latin scripts in the Windows account name or in the **STM32CubeMX2** project name.

Workaround: Enable "Use Unicode UTF-8 for worldwide language support"

Note

- To resolve this issue for **Windows 11**, enable the following Windows setting:
- In Settings go to Time & language > Language & region
- Activate "Beta: Use Unicode UTF-8 for worldwide language support," then restart Windows.

Impacted characters

This workaround is especially relevant if your Windows user account name (`C:\Users\<AccountName>`) or an **STM32CubeMX2** project name contains:

- Accented or diacritic characters (for example, é, è, ä, ö, ü, ñ, ç, ą, ę, ł, ś, ž,...).

- Non-Latin scripts (for example, Greek, Cyrillic, Chinese, Japanese, Korean, Arabic).

If your account name uses only unaccented Latin letters (A–Z, a–z), digits (0–9), underscore (_) or dash (-), this issue is unlikely to occur.

Potential side effects

Enabling this option changes the system ANSI code page to UTF-8. Most modern applications work correctly with this configuration. However, certain older non-Unicode applications that rely on legacy code pages may:

- Display garbled or incorrect characters,
- Misinterpret characters in filenames or configuration files, or
- Encounter problems with text import or export.

If such issues occur, disable this option.

Windows 10 and Windows 11 support

This setting is available on Windows 10 (version 1903 and later) and Windows 11. It is disabled by default. Users may enable this configuration if necessary to resolve encoding issues with **STM32CubeMX2**.

Install on Windows

Following these steps initiates the installation process for **STM32CubeMX2.** Proceed with the installation by following the on-screen instructions:

From GUI installer

- Open file explorer and navigate to the directory where the **GUI installer** file is located.
- Double-click on the **stm32cubemx2-[x]-X64-Windows-installer.exe** file to launch the installation wizard.
- Click the **Continue** button to begin the installation process **(1)**.

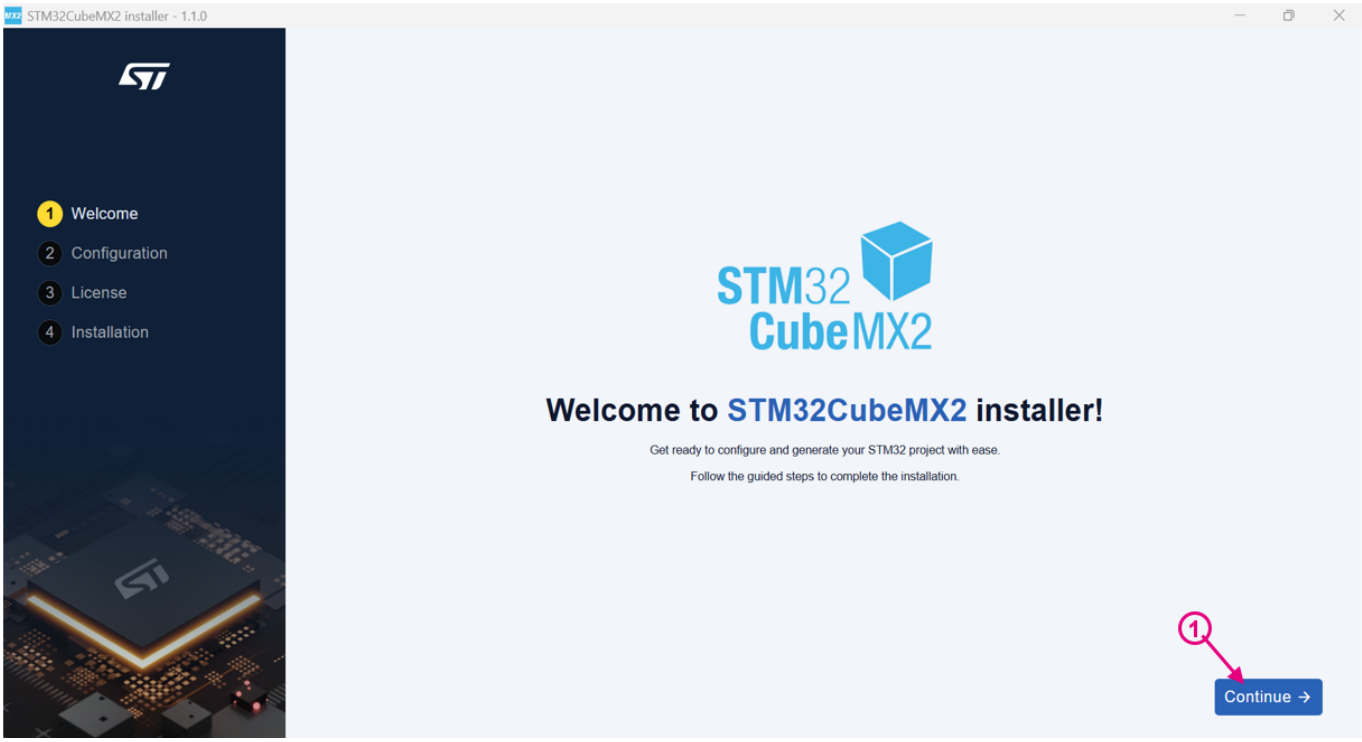


Fig. 1: STM32CubeMX2 installer on Windows - Welcome screen

- Ensure that there is enough disk space available.
- Change the installation directory if needed. Click the **change** button and select a different path. The default path is: 'C:\Users\<mylogin>\AppData\Local\STMicroelectronics\STM32CubeMX2_[<version>]' **(1)**.
- Click the **Back** button, to go back to the previous step **(2)**.
- Click on **Continue** to proceed with the installation for the current user **(3)**.
- Data collection is activated by default **(4)**.

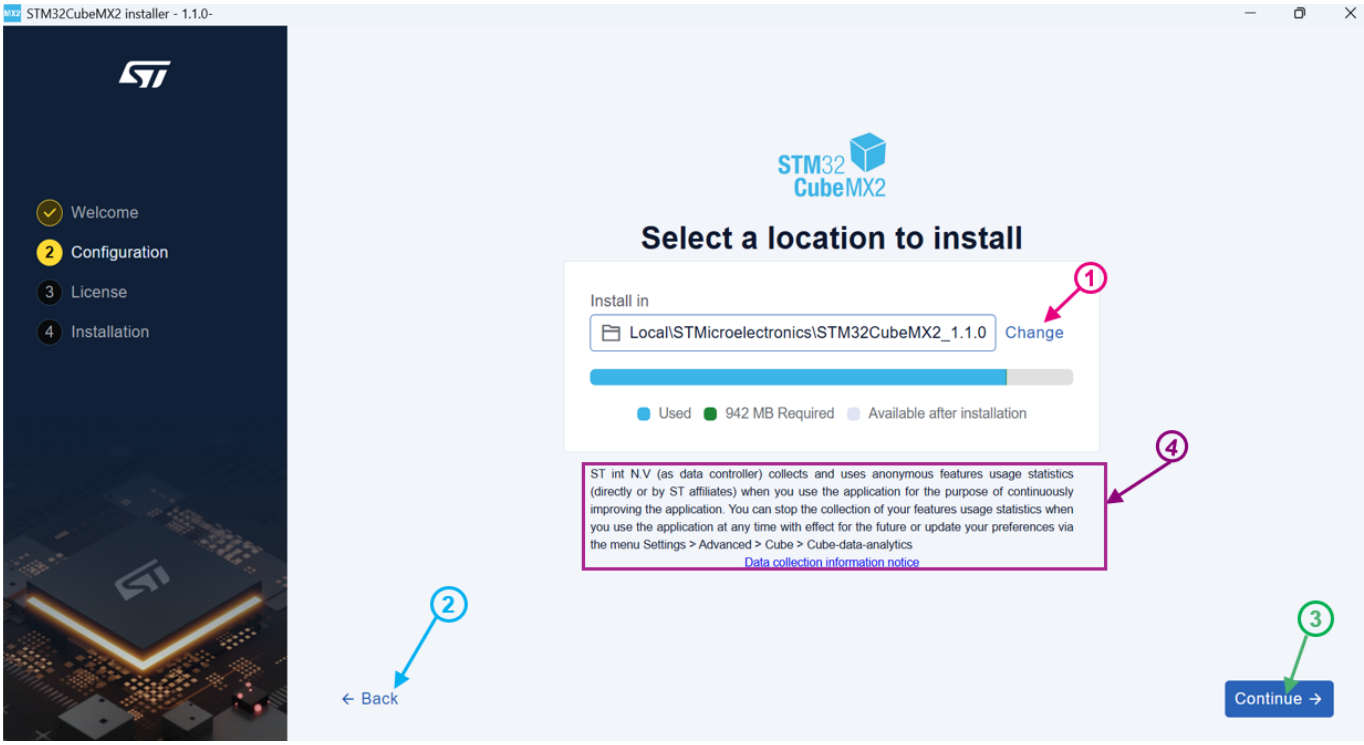


Fig. 2: STM32CubeMX2 installer on Windows - Installation location

- Read and **accept the license** terms Then click **the Begin installation** button to start the installation process.
- During the installation, the progress screen provides real-time feedback on the status of the installation.
- Once installation is complete, **launch (2)** the tool or **close (1)** the installer.

Install on Linux

To install **STM32CubeMX2** on **Linux:**

From GUI installer

- Open a terminal and navigate to the installer path.
- Type **./stm32cubemx2-[x]-X64-Linux-installer** to launch the installer.
- Alternatively, you can start the installation by double-clicking **stm32cubemx2-[x]-X64-Linux-installer**.
- Follow the installation wizard as illustrated in the figures above.

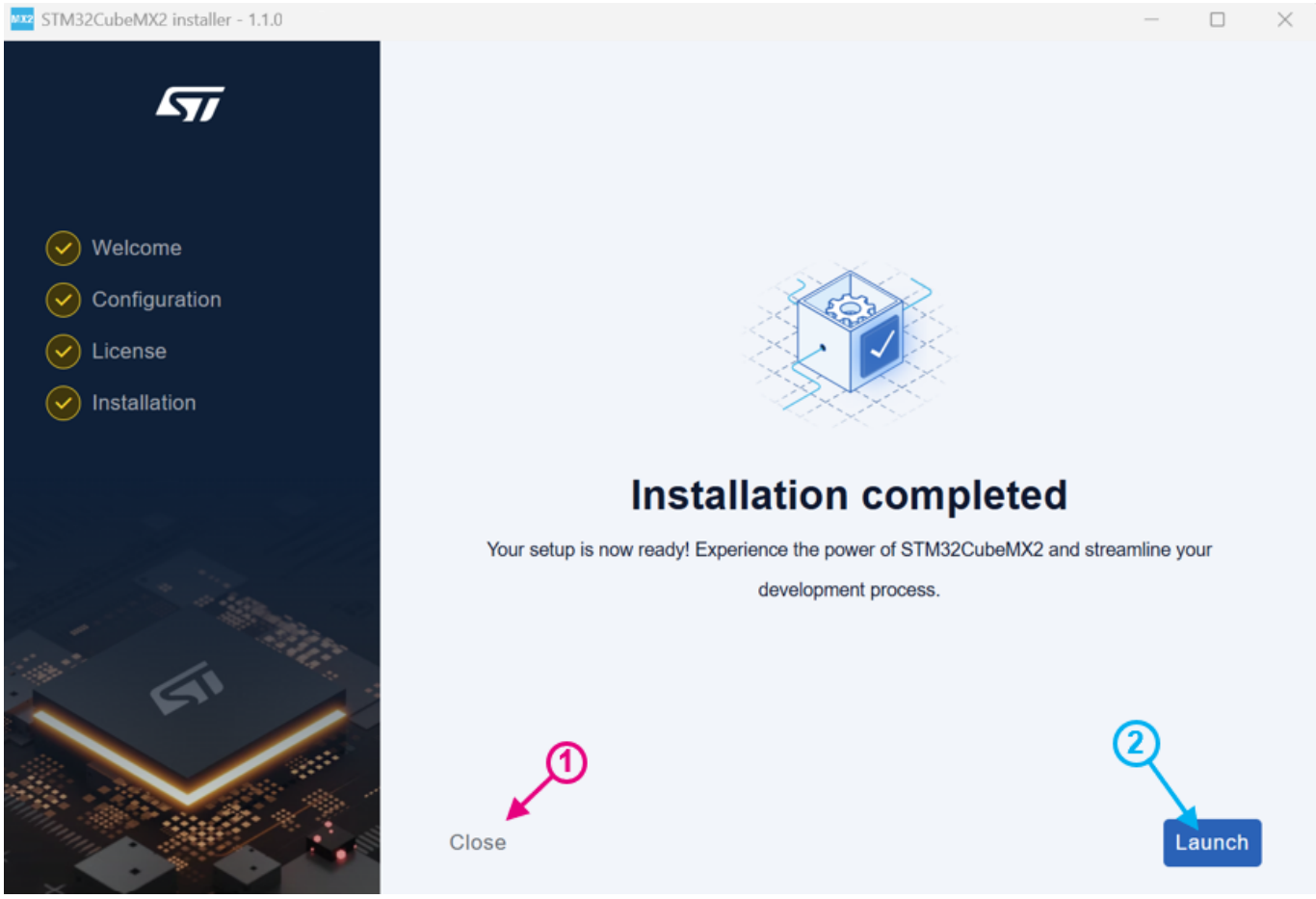


Fig. 3: STM32CubeMX2 GUI installer on Windows - Installation complete

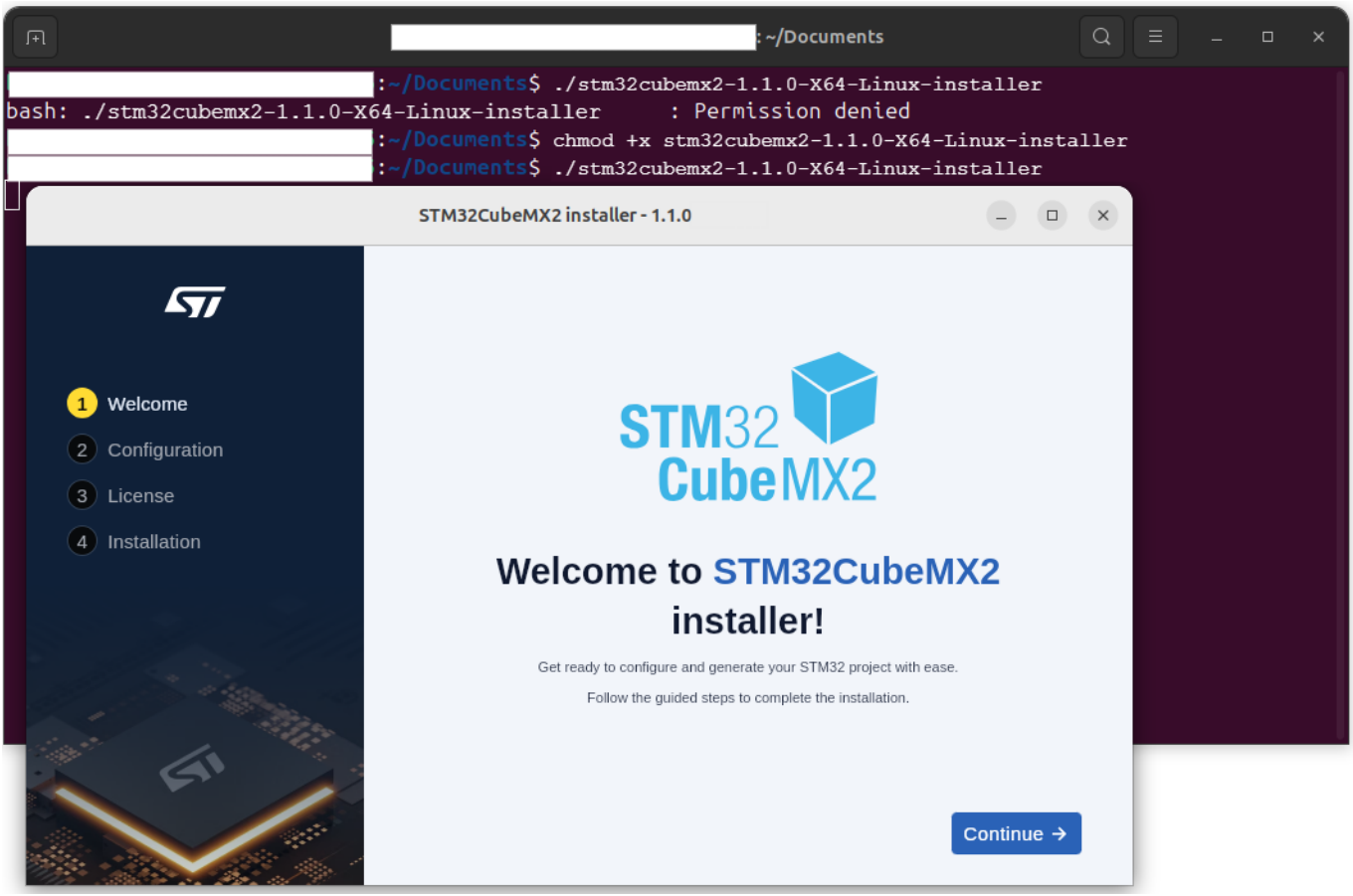


Fig. 4: STM32CubeMX2 installer on Linux

> **Note**
>
> **If you encounter a permission issue, type the following command to make the installer executable:**
> **chmod +x stm32cubemx2-[x]-X64-Linux-installer**

> **Note**
>
> **Users must install libwebkit2gtk-4.1-0 before launching the installer for Ubuntu 22.04.**
> **sudo apt install libwebkit2gtk-4.1-0**

Install on macOS

To install **STM32CubeMX2** on **macOS:**

From GUI installer

- Navigate to the **GUI installer** path.
- Double-click **stm32cubemx2-[x]-X64-macOS-installer.app.tar.gz** to extract the **tar.gz** file.
- Double-click the extracted app to start the installation process.
- Follow the installation wizard.

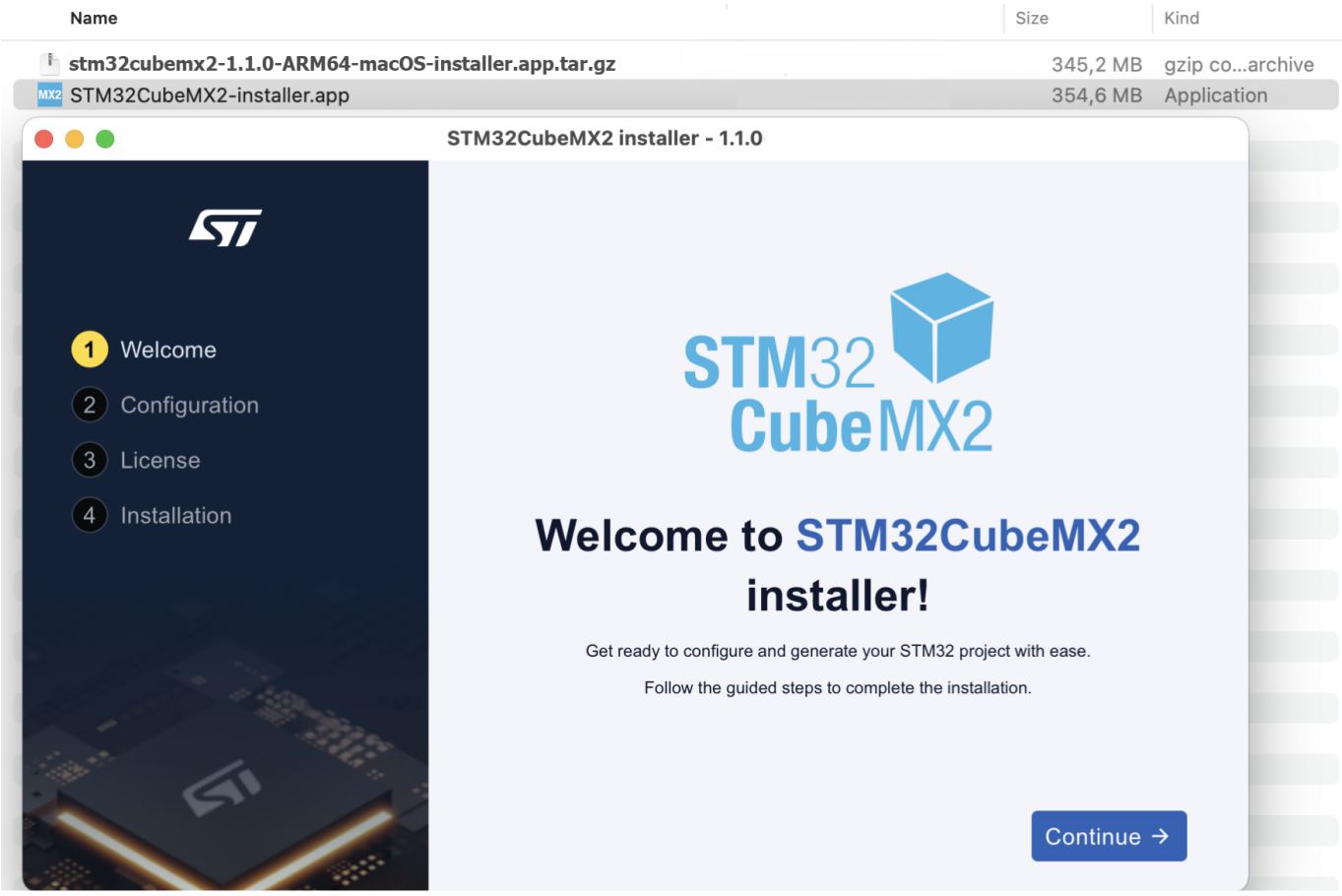


Fig. 5: STM32CubeMX2 installer on macOS

Installation Status and Error Handling

If the installation does not complete with 100% success, an installation status summary is displayed. Depending on the reported error level, the installation may still be partially operational (for example if minor errors are shown).

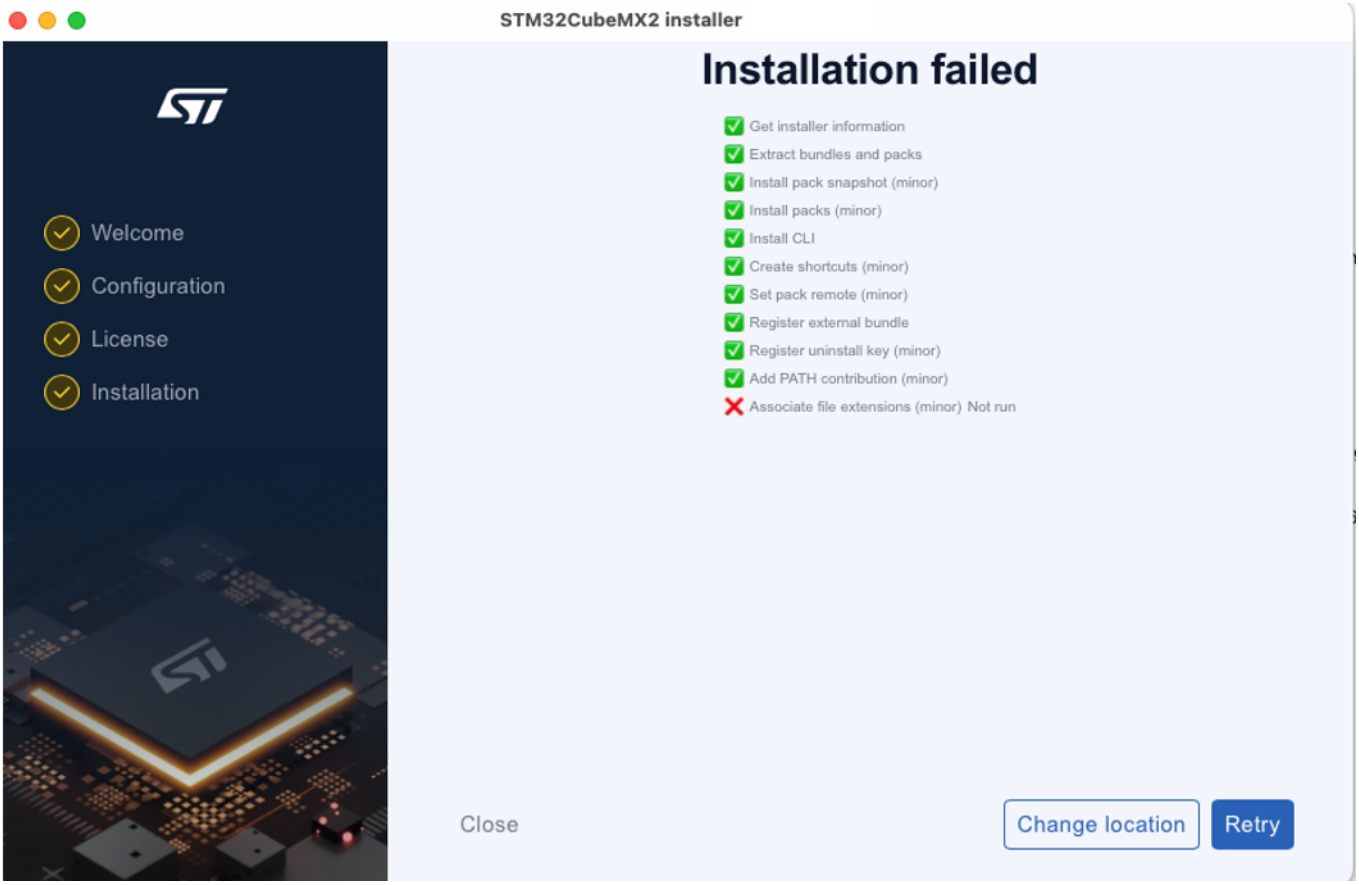


Fig. 6: Installation status summary

3.2.2 STM32CubeMX2 installation directory

The following figure shows the contents of the default installation directory for **STM32CubeMX2**

- **Default path:**
  - **Windows:** 'C:\Users\<mylogin>\AppData\Local\STMicroelectronics\STM32CubeMX2_[<version>]'
  - **Linux:** 'Home/.local/share/STMicroelectronics/STM32CubeMX2_[<version>]'
  - **macOS:** 'Users/<mylogin>/Library/Application Support/STMicroelectronics/STM32CubeMX2_[<version>]'
- The installation directory is organized as follows (Windows example):
  - **.bin:** folder: A folder containing the cube.exe command.
  - **data:** A folder containing **installation.json**, which includes information about the **STM32CubeMX2** installer.
  - **resources:** A folder that contains the necessary resources for **STM32CubeMX2.**
  - **LICENSES.txt:** A text document containing license information.
  - **stm32cubemx2_[<version>].exe:** The executable file to launch **STM32CubeMX2.**
  - **uninstall.exe:** The executable file to uninstall **STM32CubeMX2.**

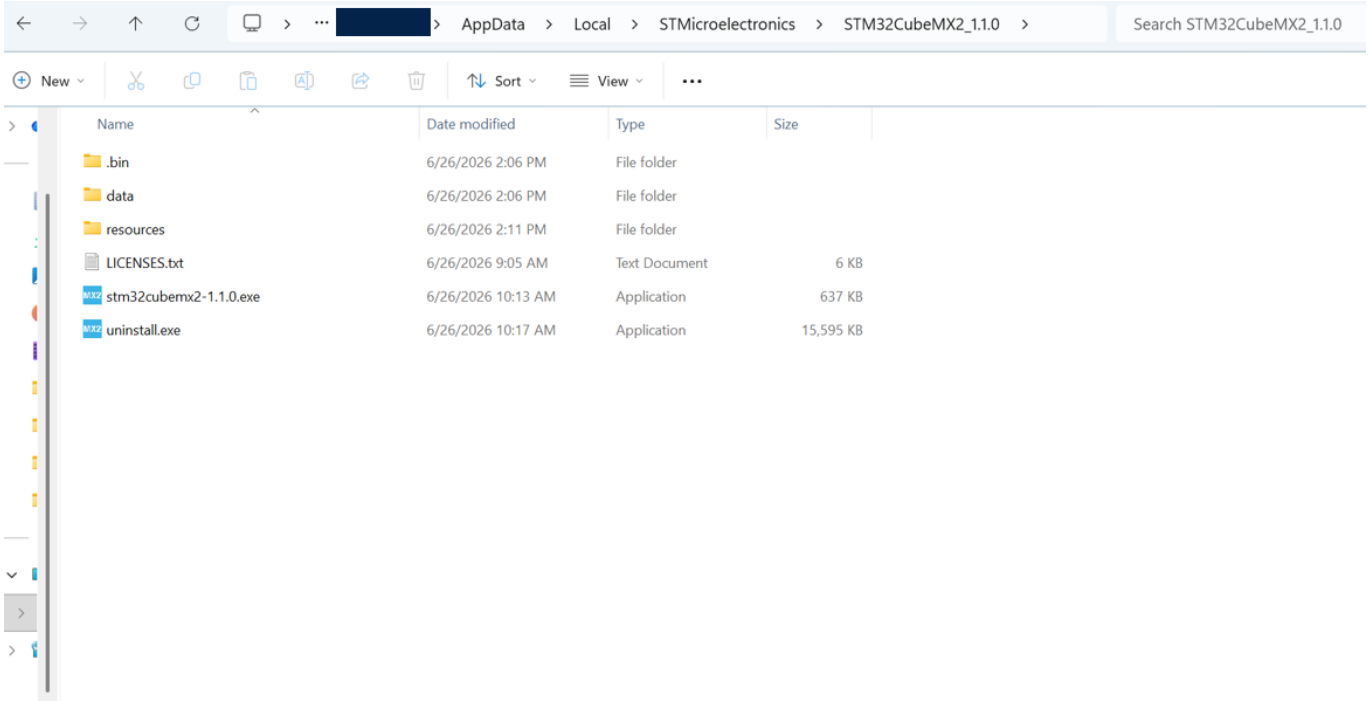


Fig. 7: STM32CubeMX2 installation directory

3.2.3 Packs installation

STM32CubeMX2 includes an automatic synchronization mechanism with the Pack Server. **STM32CubeMX2** will regularly connect to the Pack Server to:

- Check for new or updated packs
- Download and install required packs
- Keep your development environment aligned with the latest validated content

This mechanism and the pack management workflow are described in detail in the *Pack manager* section of the documentation.

- **The packs are installed under:**
  - **Windows:** 'C:\Users\<mylogin>\AppData\Local\stm32cube\packs'
  - **Linux:** 'Home/.local/share/stm32cube/packs'
  - **macOS:** 'Users/<mylogin>/Library/Application Support/stm32cube/packs'

3.2.4 Launch STM32CubeMX2

To launch **STM32CubeMX2,** use one of the following methods:

- **Using the Desktop Shortcut:** Double-click the shortcut icon on the desktop to launch **STM32CubeMX2**.
- You can also pin it to the Windows taskbar for quicker access.
- **Using the command-line interface (CLI):** Open a command prompt and run one of the following commands:
  - **cube mx start:** Launches **STM32CubeMX2** and returns control to the command prompt.
  - **cube mx start –no-detached:** Launches **STM32CubeMX2** and allows you to see logs.
  - **cube mx start –port <port number>:** Launches **STM32CubeMX2** on a specified custom port number.

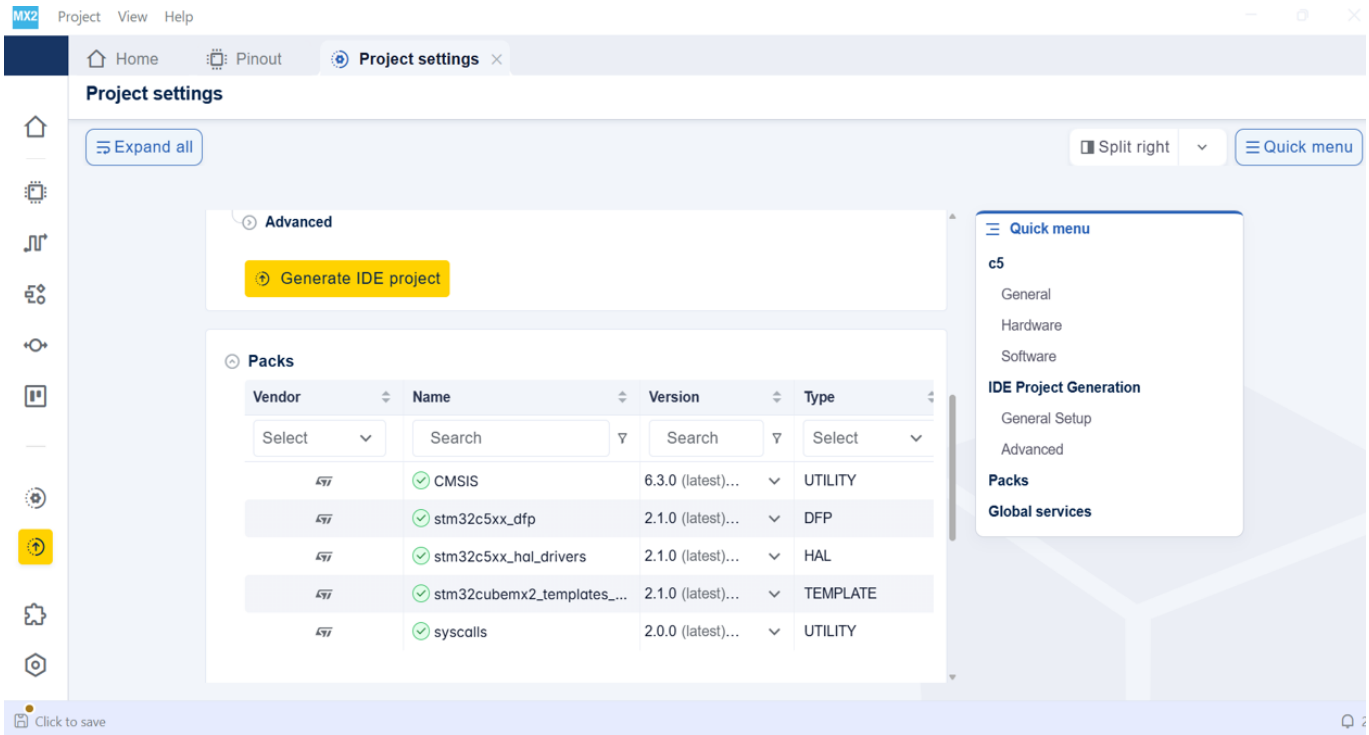


Fig. 8: Pack manager

Fig. 9: STM32CubeMX2 from the desktop shortcut

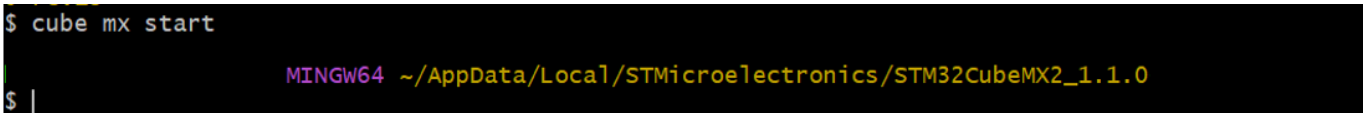
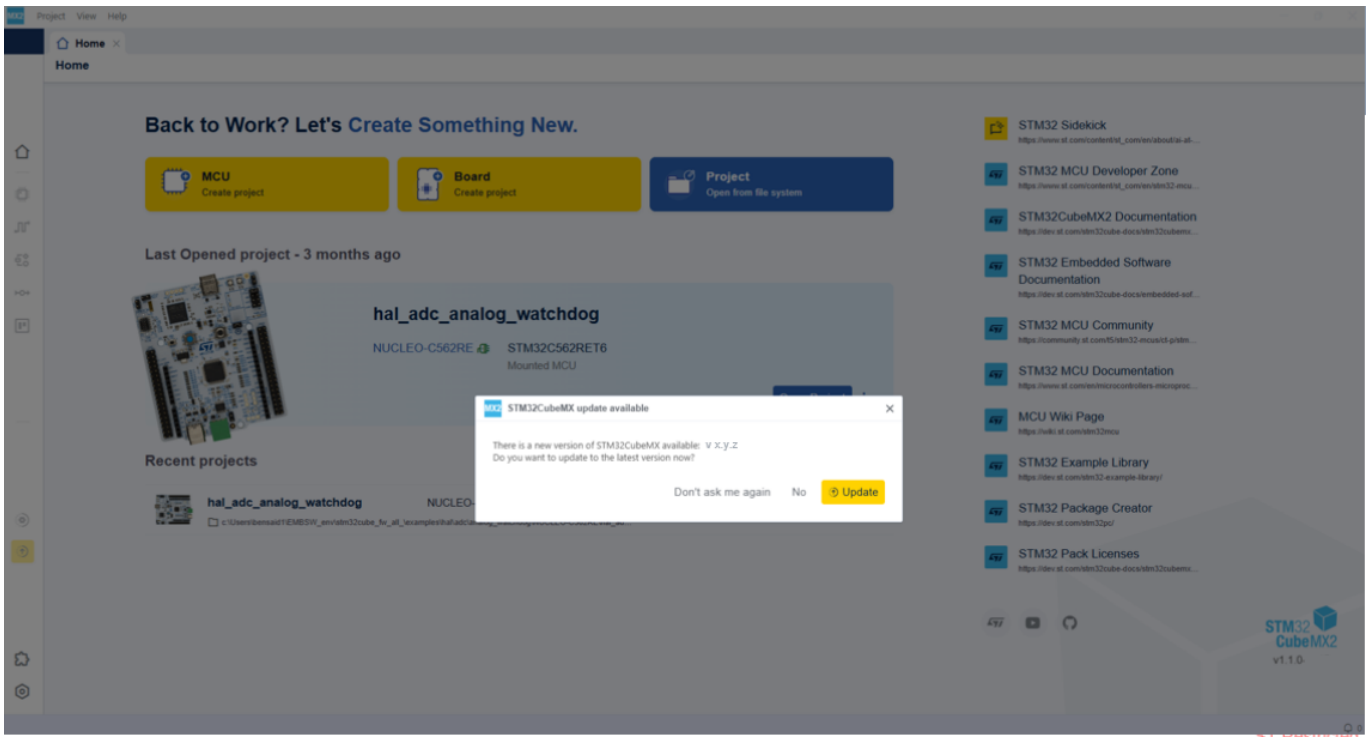


Fig. 10: Launch STM32CubeMX2 from the CLI

> **Note**
>
> For more details about the STM32CubeMX2 CLI, refer to the *Command lines interface* .

3.2.5 How to download and install new STM32CubeMX2 version



A notification is displayed when a new **STM32CubeMX2** version becomes available. To update **STM32CubeMX2** to the latest version, follow these steps:

1. Click the **'Update' button** in the message.
2. The download and installation of the new version will start automatically.

3.3 Uninstallation

3.3.1 Uninstallation process

Uninstall on Windows

To uninstall **STM32CubeMX2** using the **uninstaller.exe** utility:

1. Navigate to the **STM32CubeMX2** installation directory.
2. Locate and run **uninstaller.exe**.
3. Follow the on-screen instructions to complete the uninstallation. The process removes files and shortcut

Uninstall on Linux

To uninstall **STM32CubeMX2** from **Linux:**

- Open a terminal and navigate to the installation folder.
- Enter **./uninstall** to launch the uninstallation process.
- Alternatively, you can start the uninstallation by double-clicking the **uninstall**.
- Follow the on-screen instructions to complete the uninstallation.

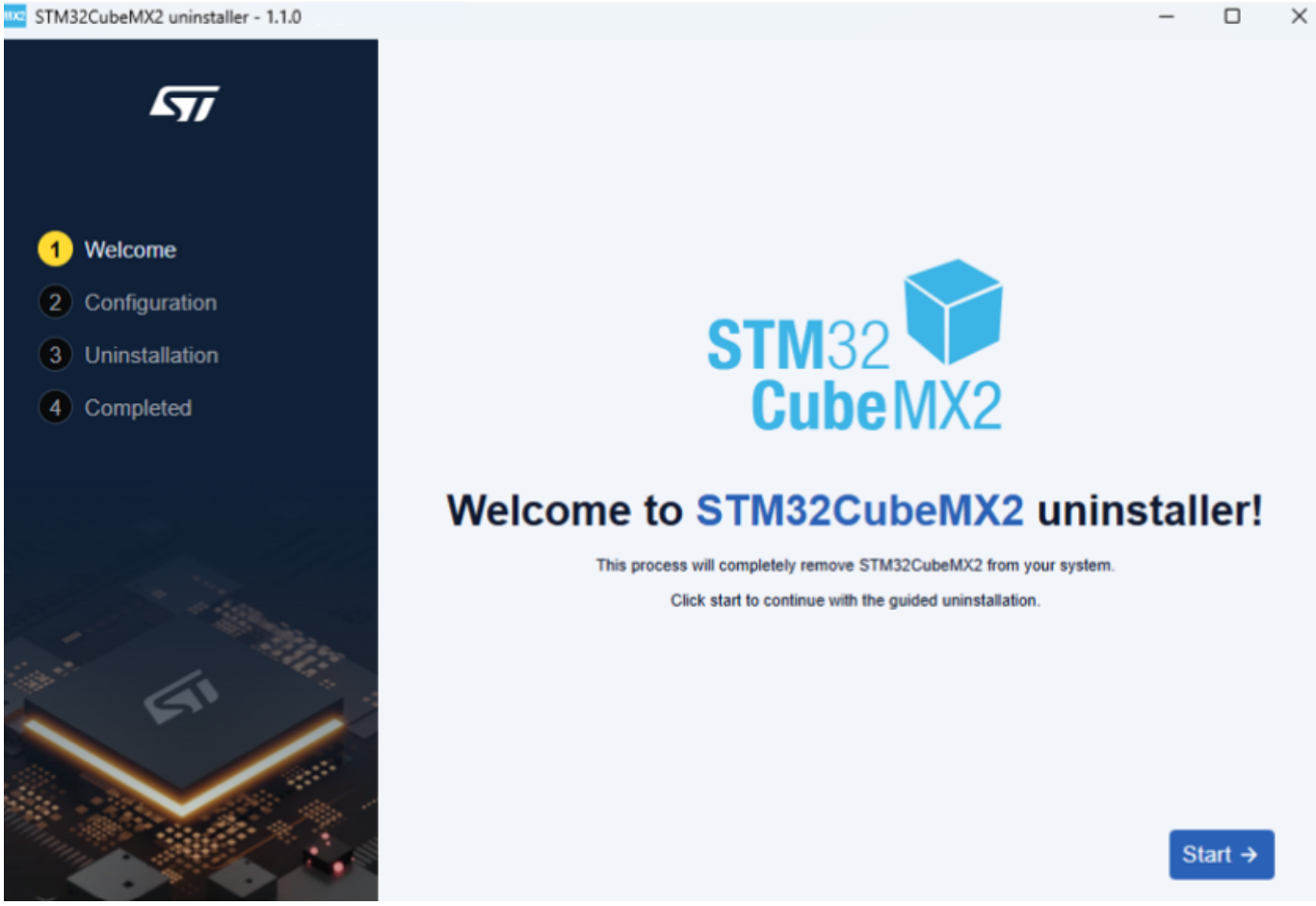


Fig. 11: STM32CubeMX2 uninstaller start on Windows

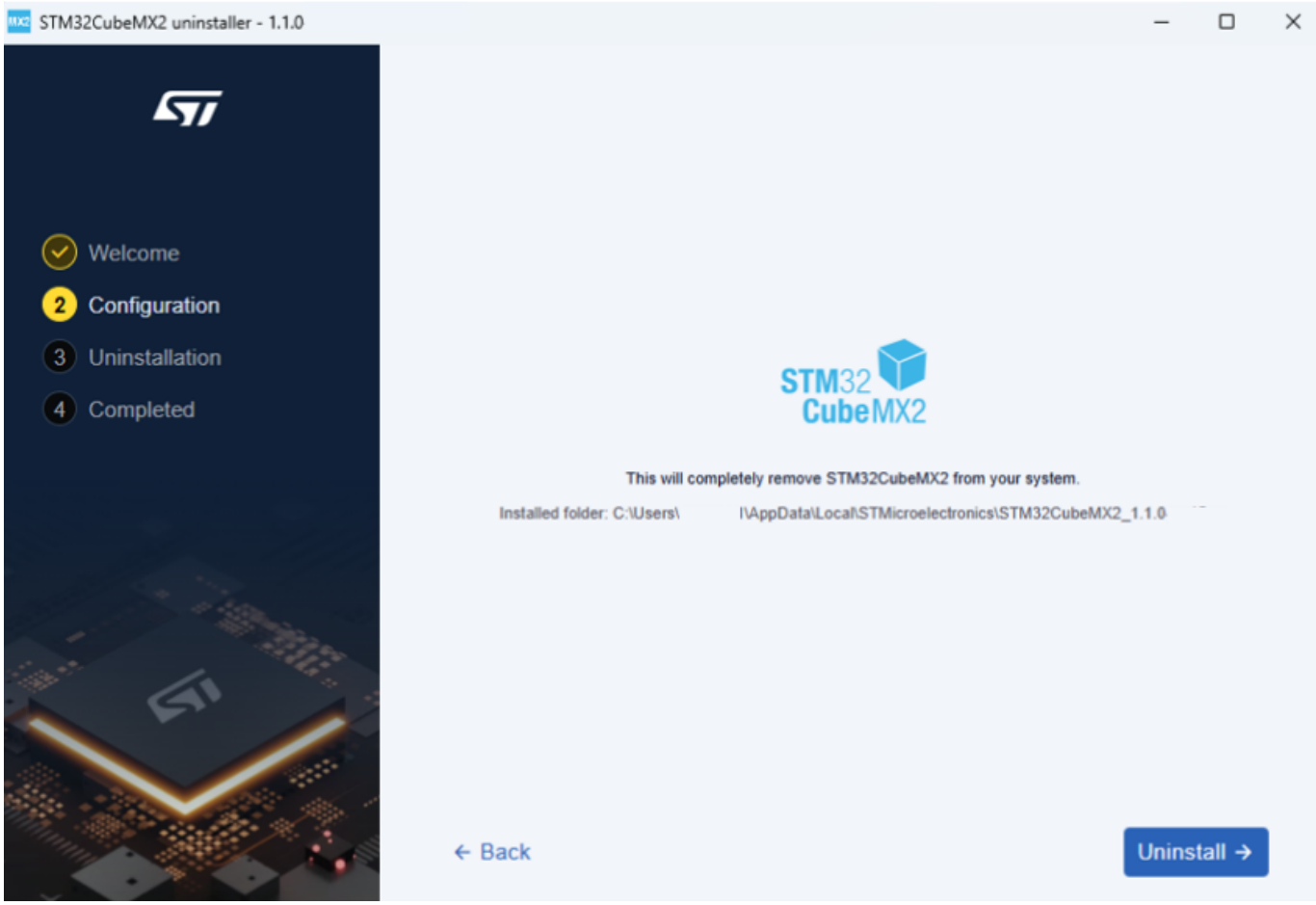


Fig. 12: STM32CubeMX2 uninstaller on Windows

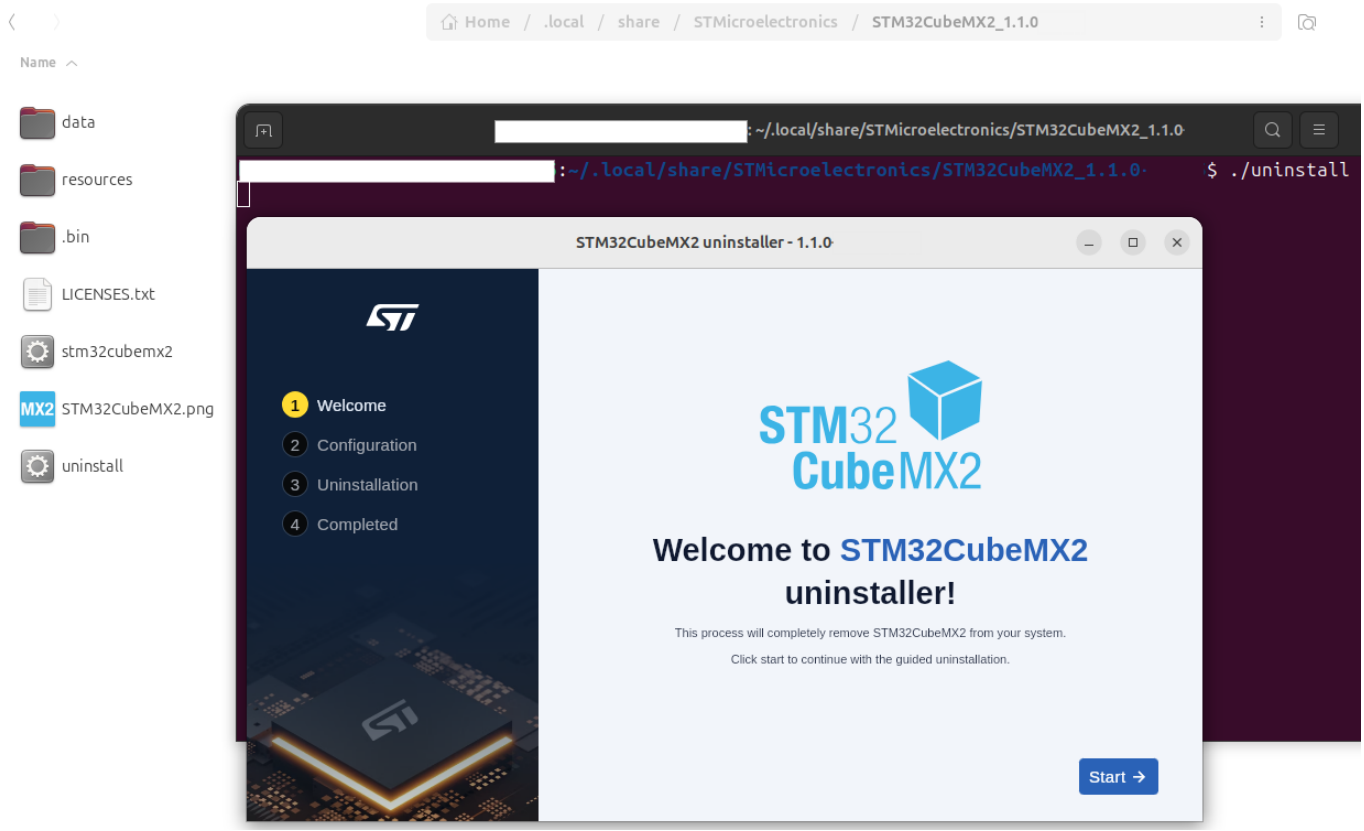


Fig. 13: STM32CubeMX2 uninstaller on Linux

Uninstall on MacOS

To uninstall **STM32CubeMX2** from **MacOS:**

- Open a terminal and navigate to the installation folder.
- Enter **./uninstall** to start the uninstallation process.
- Alternatively, you can start the uninstallation by double-clicking the **uninstall**.
- Follow the on-screen instructions to complete the uninstallation.

Uninstallation Status and Error Handling

If the uninstallation does not complete with 100% success, an uninstallation status summary is displayed.

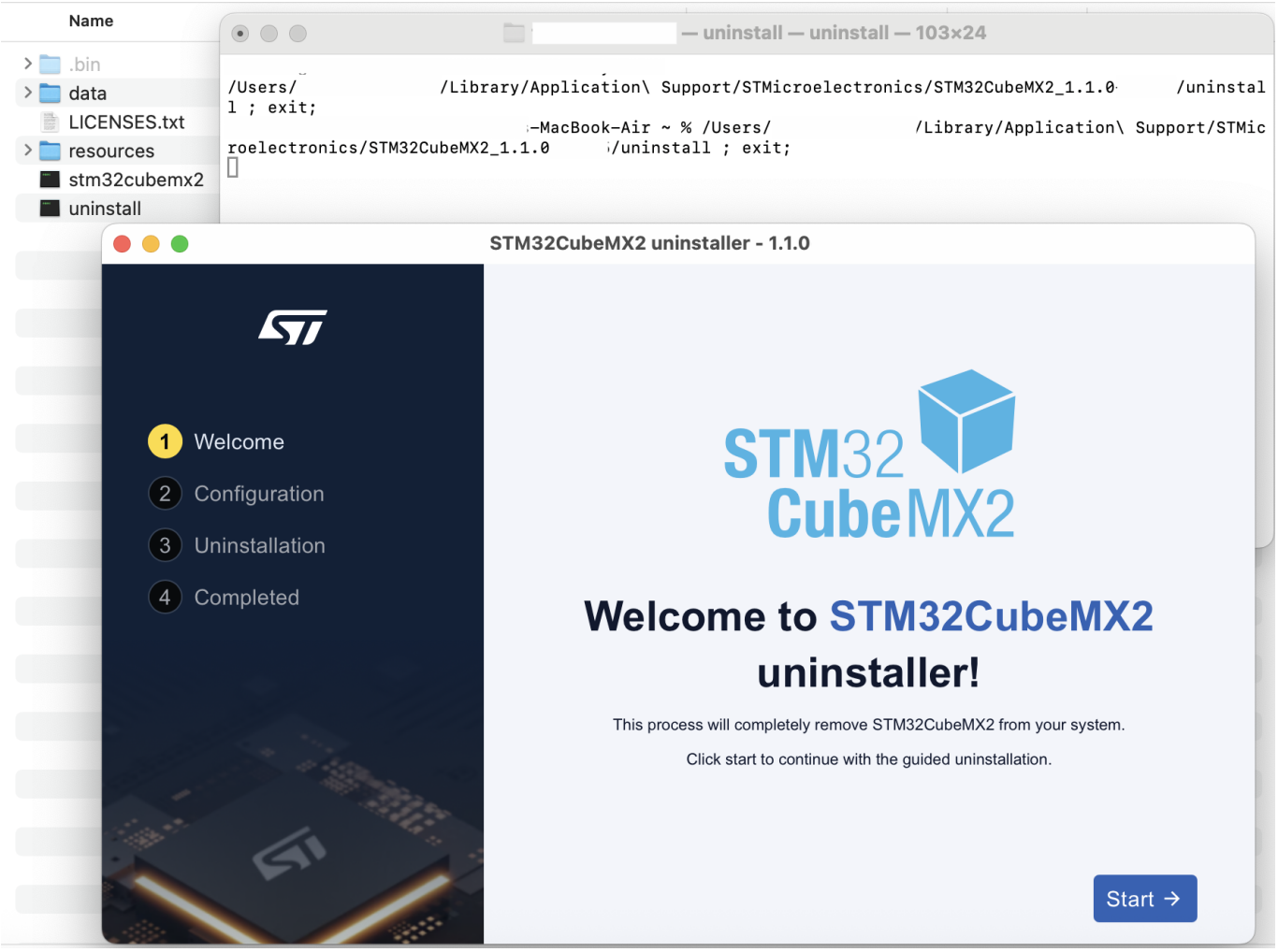


Fig. 14: STM32CubeMX2 uninstaller on MacOS

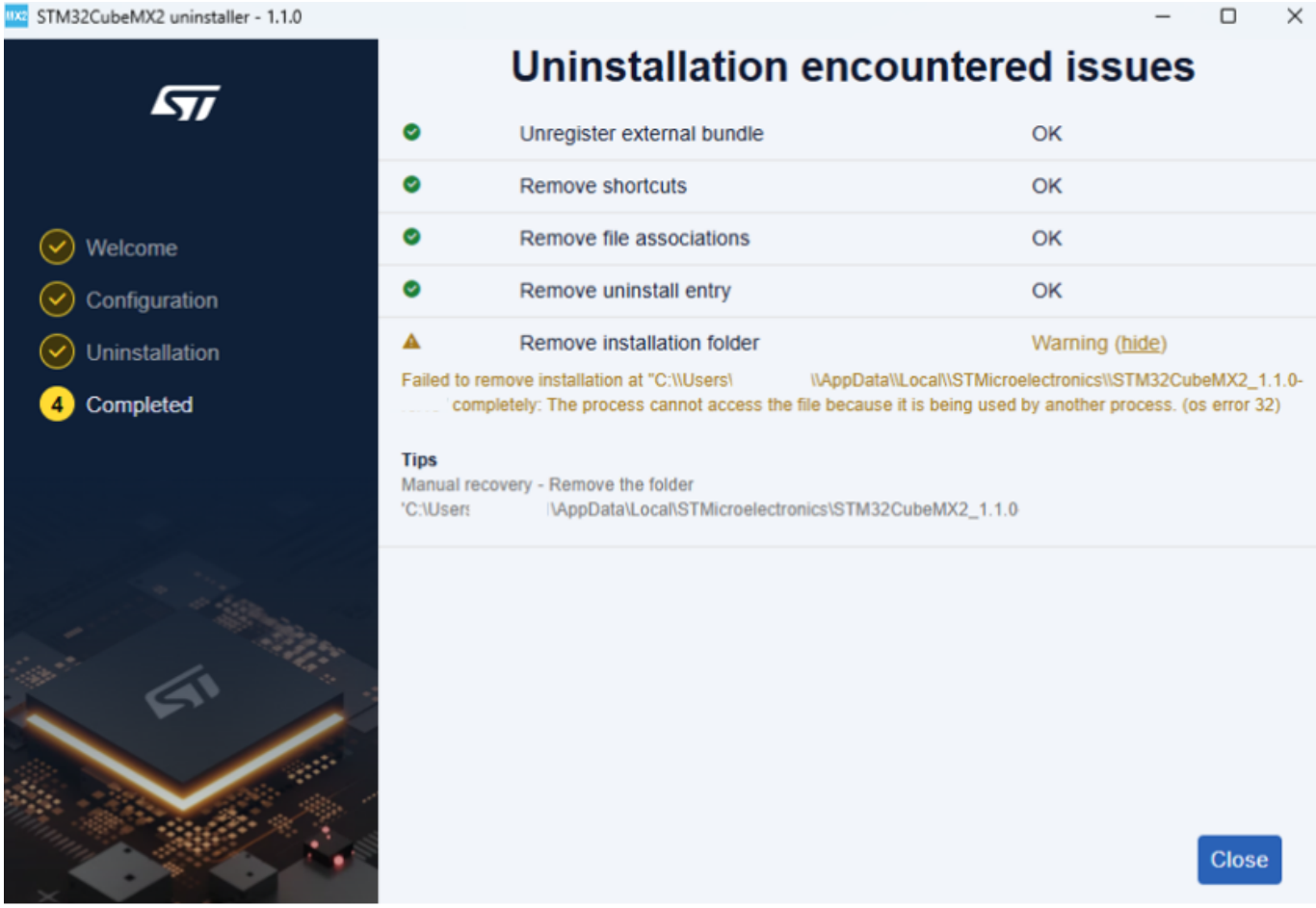


Fig. 15: Uninstallation status summary

3.4 Settings and preferences

3.4.1 Settings

The settings window can be accessed in two ways:

- **In the left bar (1):** Click the gear icon at the bottom of the left bar.
- **In the menu bar (2):** In the menu bar at the top of the application window, select **Project** → **Preferences** → **Settings** (**Ctrl and the comma key(Ctrl + ‘,’)**).

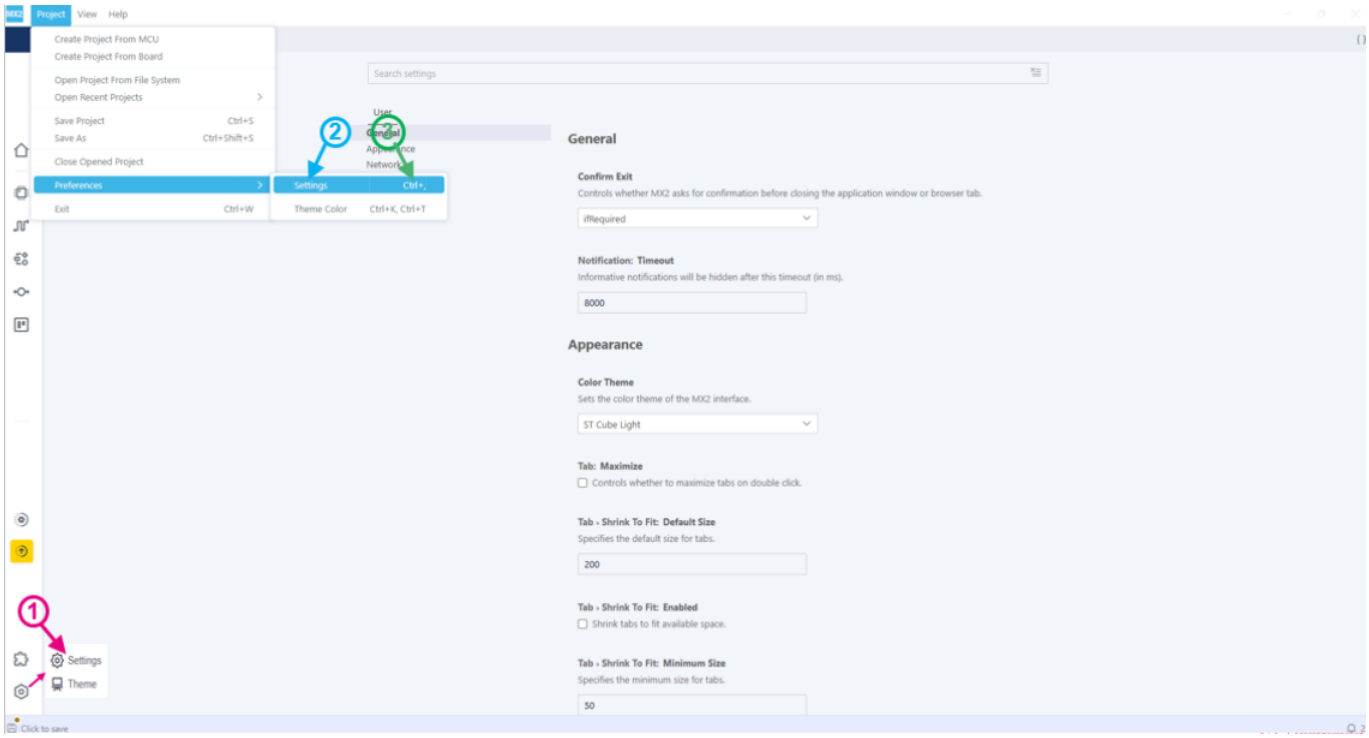


Fig. 16: Settings window

3.4.2 Theme color

To customize the application appearance:

- Open the settings by clicking the gear icon in the bottom-left corner **(1)**.
- Select from available themes **(2)**:
  - **Light themes.**
  - **Dark themes.**

3.4.3 Additional articles

- How to select the dark theme color on STM32CubeMX2

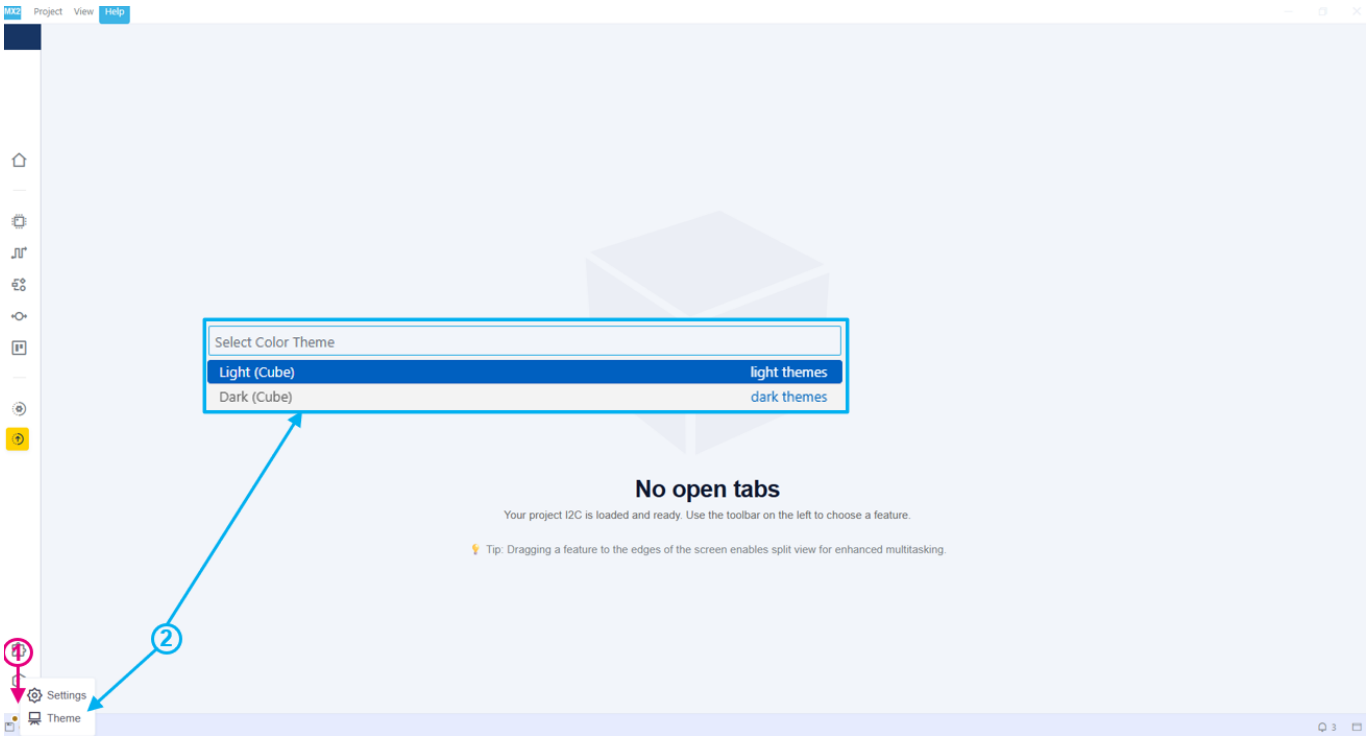


Fig. 17: STM32CubeMX2 themes

3.5 Navigation and page structure

3.5.1 General interface

Some elements of the STM32CubeMX2 interface are common to most pages. These elements are the **top menu, tab bar, sidebar, and footer.**

Top menu (1)

The top menu provides access to high-level features:

- **Project options:** create, open, save, save as.
- **View options:** Zoom In, Zoom Out, tab management.
- **Help:** access documentation, software version, pack, and licenses.

Tab bar (2)

All STM32CubeMX2 pages open in dedicated tabs. Tab headers are displayed in this area. The following actions are available:

- **Close a tab:** click the cross icon on the right of the tab header.
- **Display a tab:** click the tab header.
- **Move a tab:** click and hold the tab header, then drag it to another position.

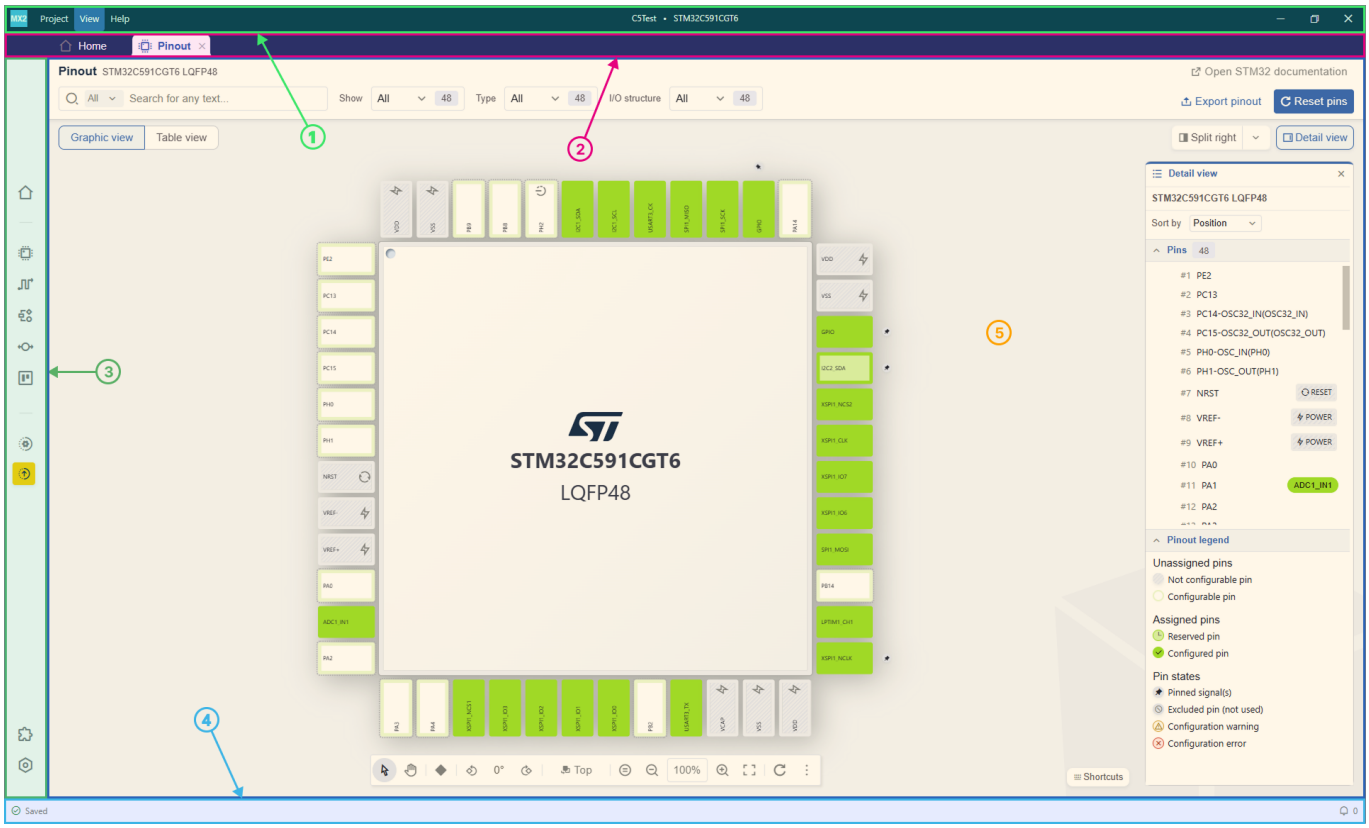


Fig. 18: General interface of STM32CubeMX2

Fig. 19: Tab bar

Note

- If too many tabs are open, some tab headers may not be visible. Use the mouse scroll wheel to navigate across tab headers.
- A scrollbar will be displayed below the tab headers.

Side bar (3)

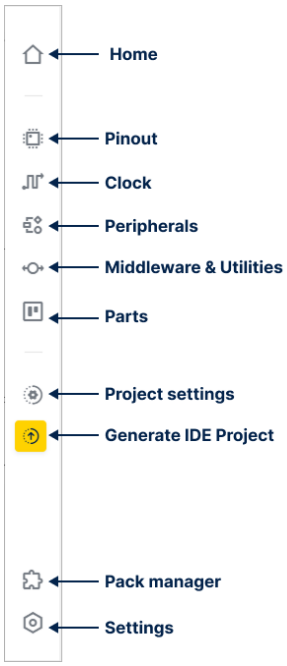


Fig. 20: Side bar with an opened project

The sidebar is used to open new tabs in STM32CubeMX2. The available tabs are:

1. **Home:** manage projects.
2. **Pinout:** configure and manage microcontroller pin assignments and properties.
3. **Clock:** configure and manage MCU clock.
4. **Peripherals:** access peripheral configuration pages.
5. **Middleware:** access middleware and utilities configuration pages.
6. **Parts:** access parts configuration pages.
7. **Project settings:** view project information, including generation settings.
8. **Generate IDE project:** trigger IDE project generation.
9. **Pack manager:** add, remove or update software packs.
10. **Settings:** configure STM32CubeMX2 options.

> **Note**
>
> - Except the Pack manager and the settings, all sidebar icons are disabled until a project is opened.

Footer (4)

Fig. 21: Footer in a saved state & 11 notifications

The **footer** displays different types of information:

- **Save status:** on the left part of the footer, shows the current project's save status. Possible statuses are "Saved" (all information saved) and "Click to save" (changes made since last save). Click "Click to save" to save the project.
- **Notifications:** on the right part of the footer, there's an icon used to access notifications. Click to display all notifications. The notifications are detailed in the next section.

Main workspace (5)

The **main workspace**

- is the place where the tabs will be displayed.
- It's possible to display several tab side by side, this will be detailed in the next chapter.

3.5.2 How to use the split view

It is possible to **display several tabs at a time.** Tabs can also be resized to adapt to the current task.

There are two main ways to use this system: by using the **Split view** buttons in the tabs, or by using **drag and drop.**

Split view button

There is a **split view** button in the top part of most pages. Use this button to create standard split views. Clicking this button splits the interface into **two blocks of identical proportions**.

There are several types of splits:

- For left and right splits, the screen is split **vertically.**
- For top and bottom splits, the screen is split **horizontally.**

Steps :

1. Click the **split view** button.
2. Select a **type of split**: left, right, top, or bottom.
3. The current tab is **displayed in the selected part.**

> **Note**
>
> It is possible to select a type of split: left, right, top, or bottom.

Drag and drop

It is also possible to use a split view by using drag and drop.

Steps :

1. **Click and hold** the left button of your mouse while hovering over a tab header.
2. A **blue frame** appears when you hover over the rest of the interface. The frame represents **where the tab will appear.**
3. **Drop** the tab.

> **Note**
>
> - Each tab can be resized vertically or horizontally.

3.5.3 How to use the zoom

It's possible to Zoom In (increase the size of the elements) or Zoom Out (decrease the size of the elements) in STM32CubeMX2.

There's several way to do it:

With the menu

To Zoom In or Out, it's possible to use the menu.

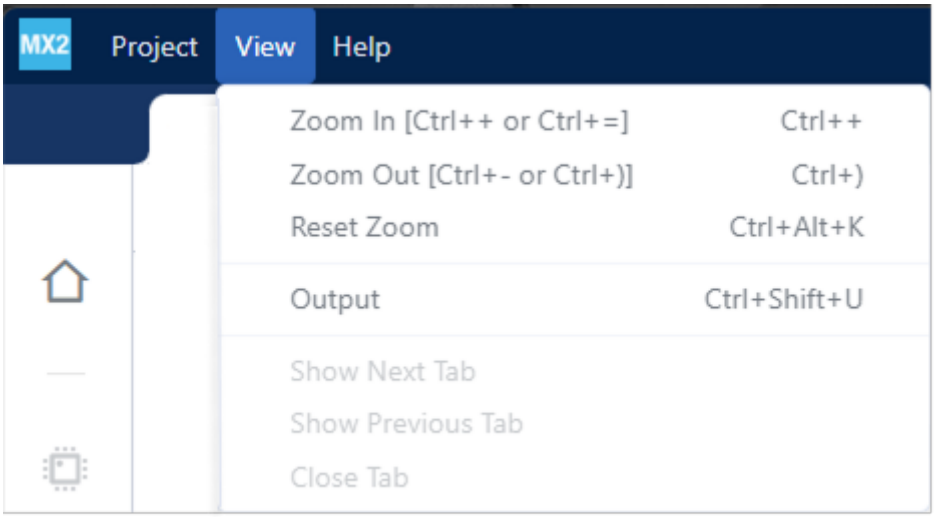


Fig. 22: Zoom In and Out in the menu

Click on 'View' in the menu, and:

- Click on **Zoom In** to increase the size of the elements.
- Click on **Zoom Out** to decrease the size of the elements.

With the shortcuts

It's also possible to Zoom In or Out with keyboards shortcuts. Depending on your keyboard configuration, you can use:

- Click on **Ctrl** + + or **Ctrl** + = to increase the size of the elements.
- Click on **Ctrl** + - or **Ctrl** +) to decrease the size of the elements.

3.6 How to start from example projects

This chapter explains how to open example projects in STM32CubeMX2 from STM32Cube software packages and how to work with them. The chapter covers the following actions:

- **Open** an existing ioc2 file from STM32Cube software packages.
- **Save as** the opened project to another location.
- **Generate IDE projects**.

3.6.1 Prerequisites

Before starting, complete the following steps:

- Install an STM32Cube software package that contains the desired example.
- Ensure STM32CubeMX2 is installed.

3.6.2 Search for an example with ioc2 file inside STM32Cube software package

- Locate the examples folder within the embedded software package containing ioc2 files.
- Select an example containing an .ioc2 file.

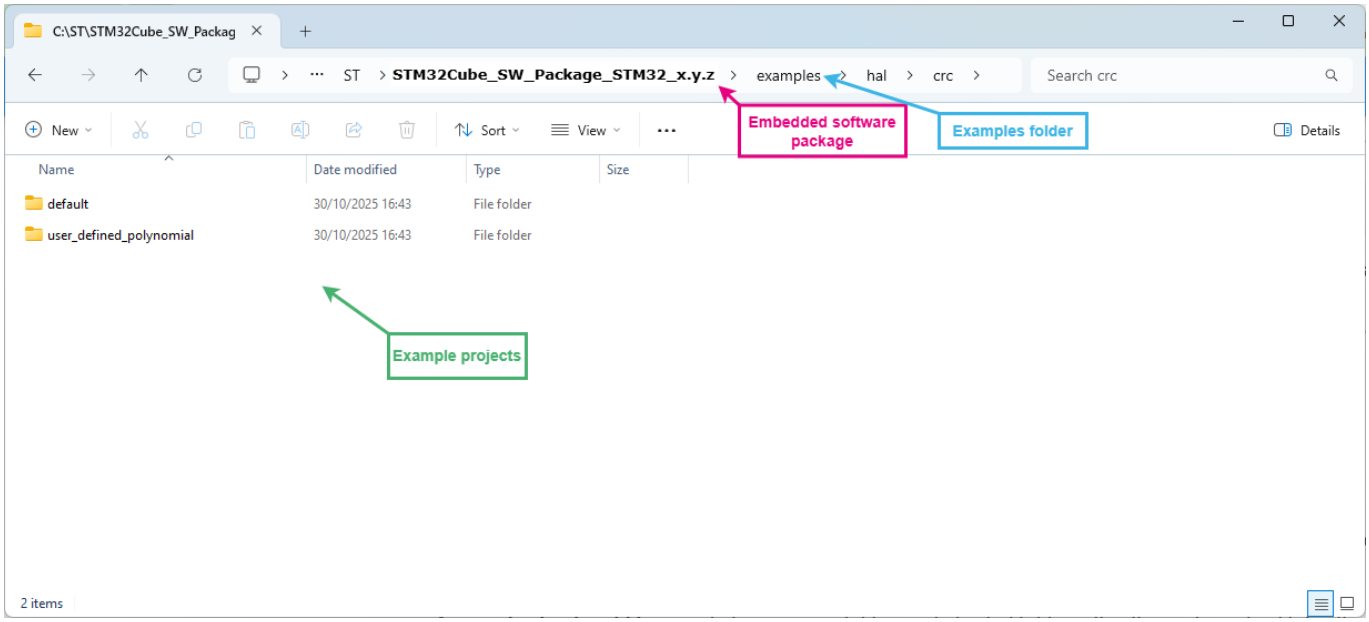


Fig. 23: Locate an example project

3.6.3 Open located example with STM32CubeMX2

Open STM32CubeMX2 and select the corresponding ioc2 file

- Go to the Home page and select Project.
- Browse to the embedded software package examples folder and select the desired example .ioc2 file.

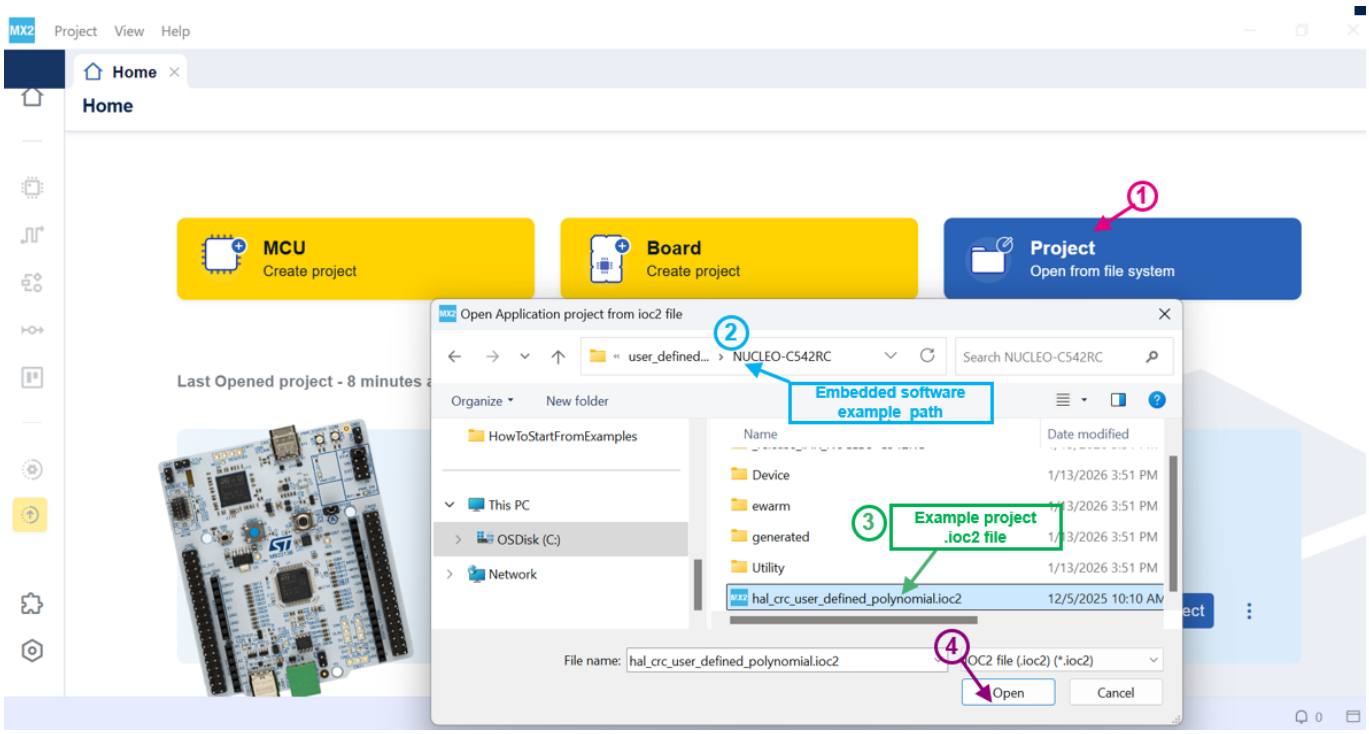


Fig. 24: Open an example project in STM32CubeMX2

- To protect your reference example project, the opened project is in read-only state.
- An information banner appears under the Project Settings tab indicating that the project is read-only **(1)**.
- All fields under "IDE Project Generation" are disabled **(2)**.
- The pack list is frozen, settings are visible in read-only mode, and no pack updates are possible **(3)**.
- Only the **Save As** function is allowed.
- When you try to save the project, a warning appears with a button to **Save As** the project.
- If the user chooses to generate the project, a warning appears with a button to **Save As** the project.

Save As the project

- After a Save As, the new project is loaded.
- All actions are possible on the project.

Generate saved project

- Choose the project format.
- Click the Generate button.
- The example is extracted from the STM32Cube software package using its original example path, and IDE project will be generated.
- Once generation is completed, you can view the generated files on disk.

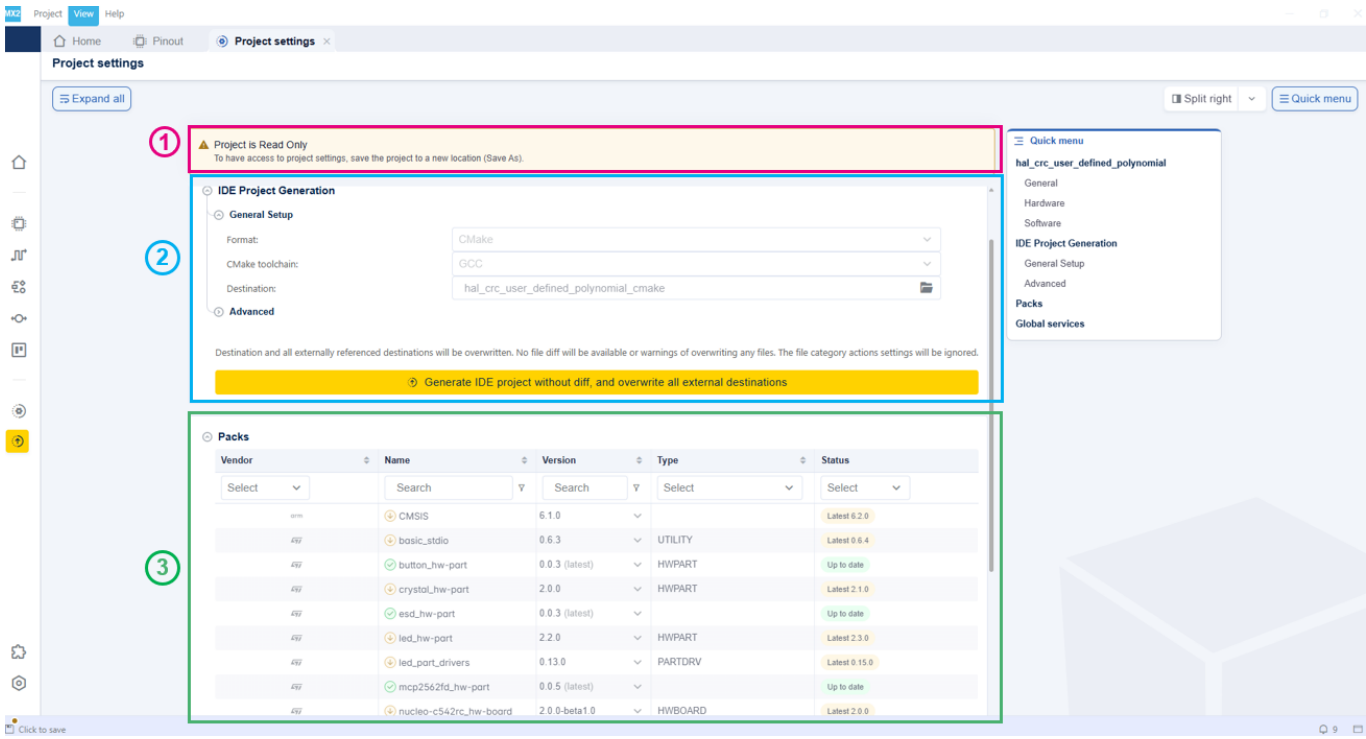


Fig. 25: Read-Only project

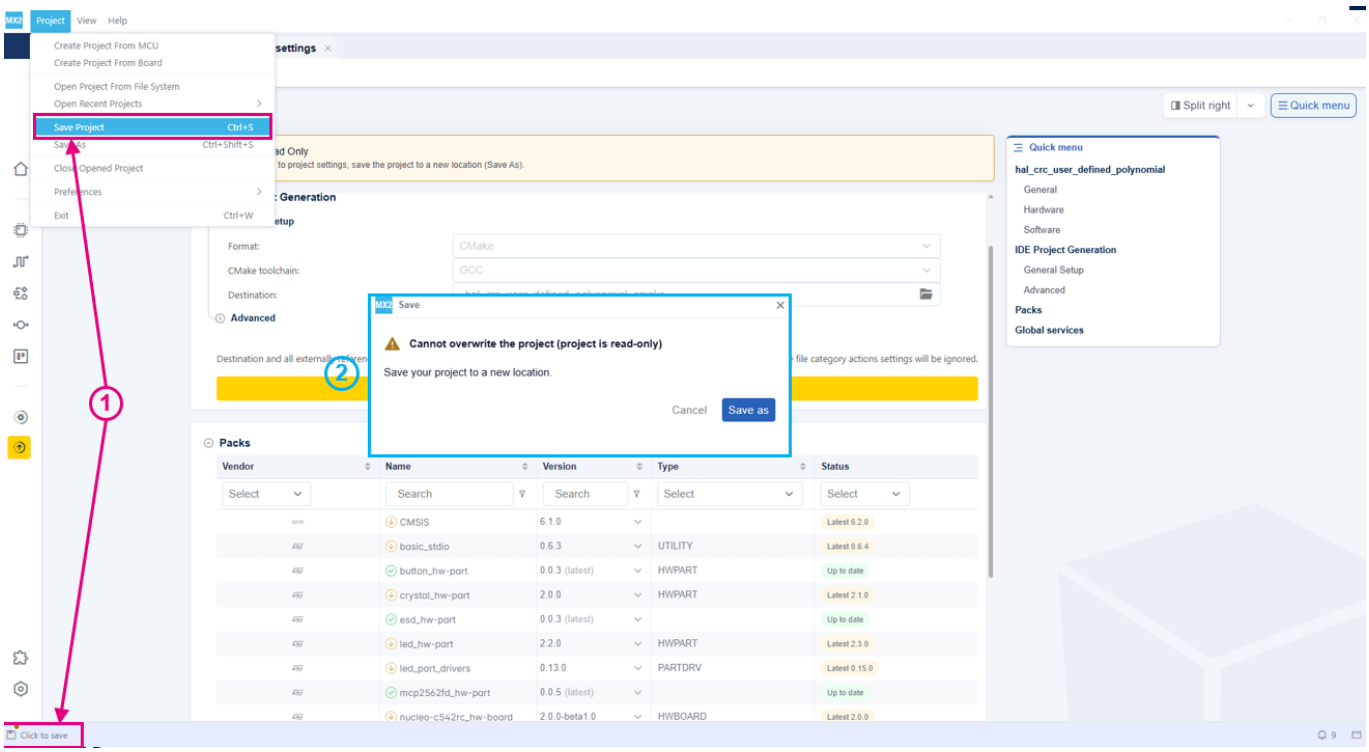


Fig. 26: Save Read-Only project

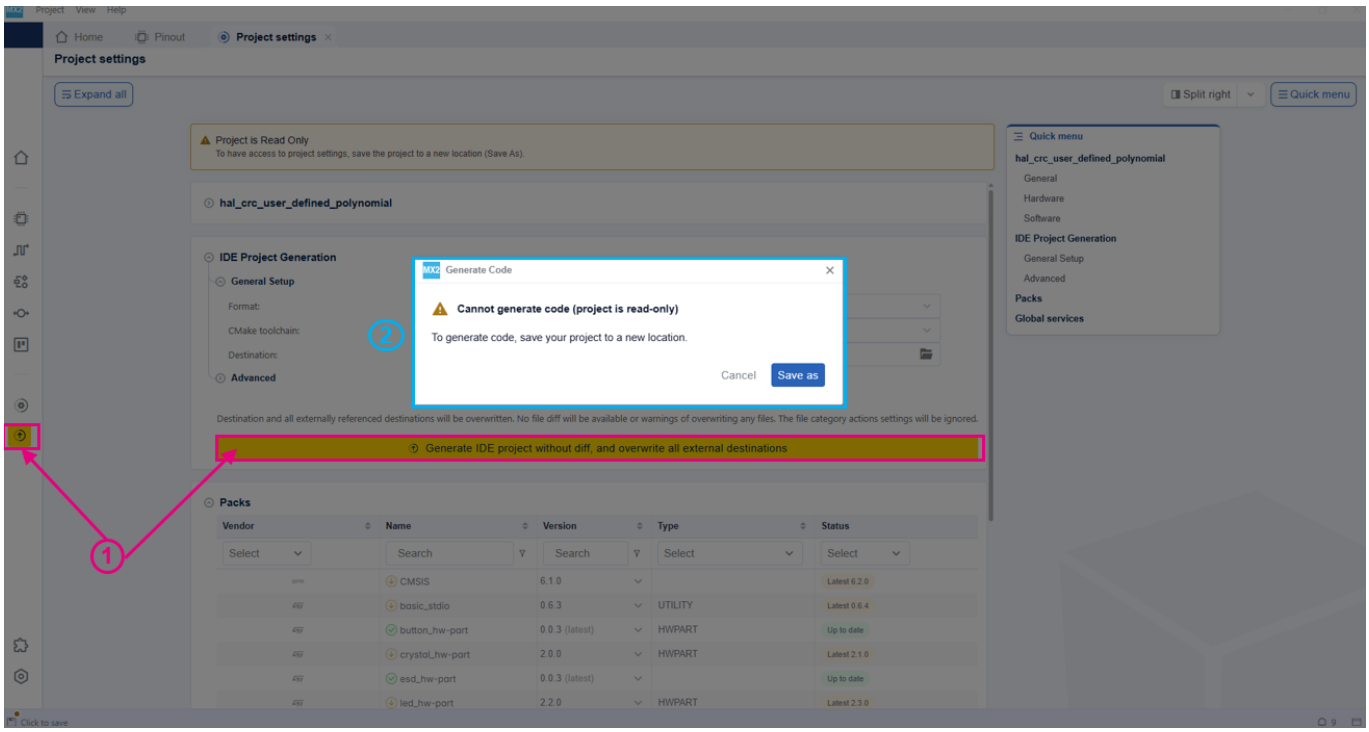


Fig. 27: Generate project with Read-Only status

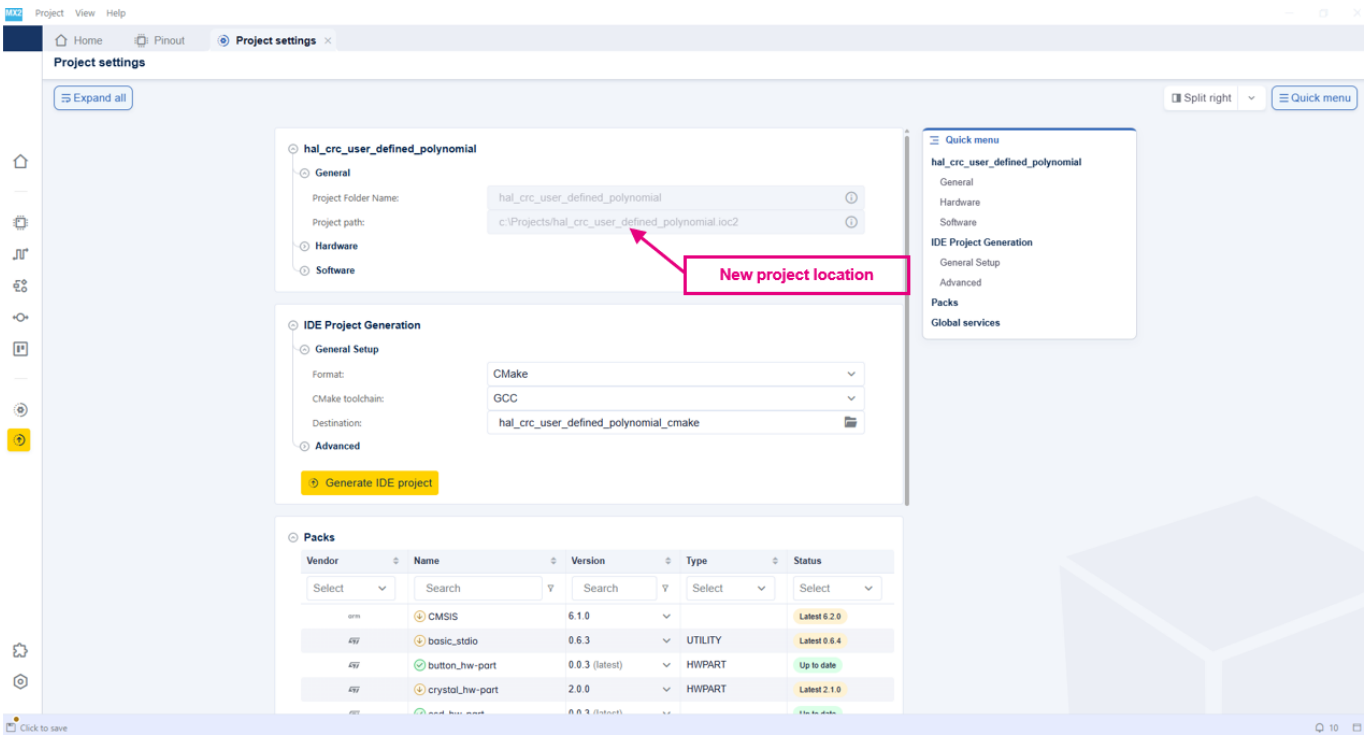


Fig. 28: Example saved under the new location

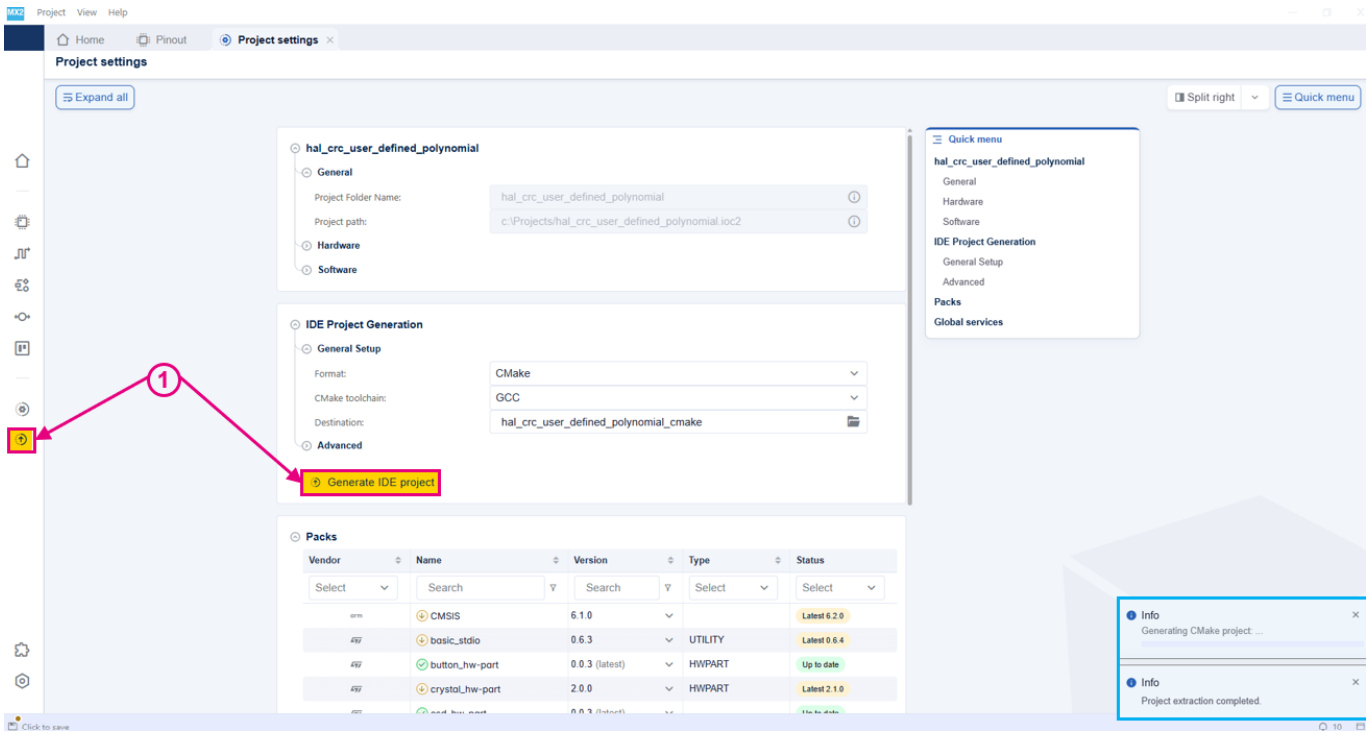


Fig. 29: Project Generation

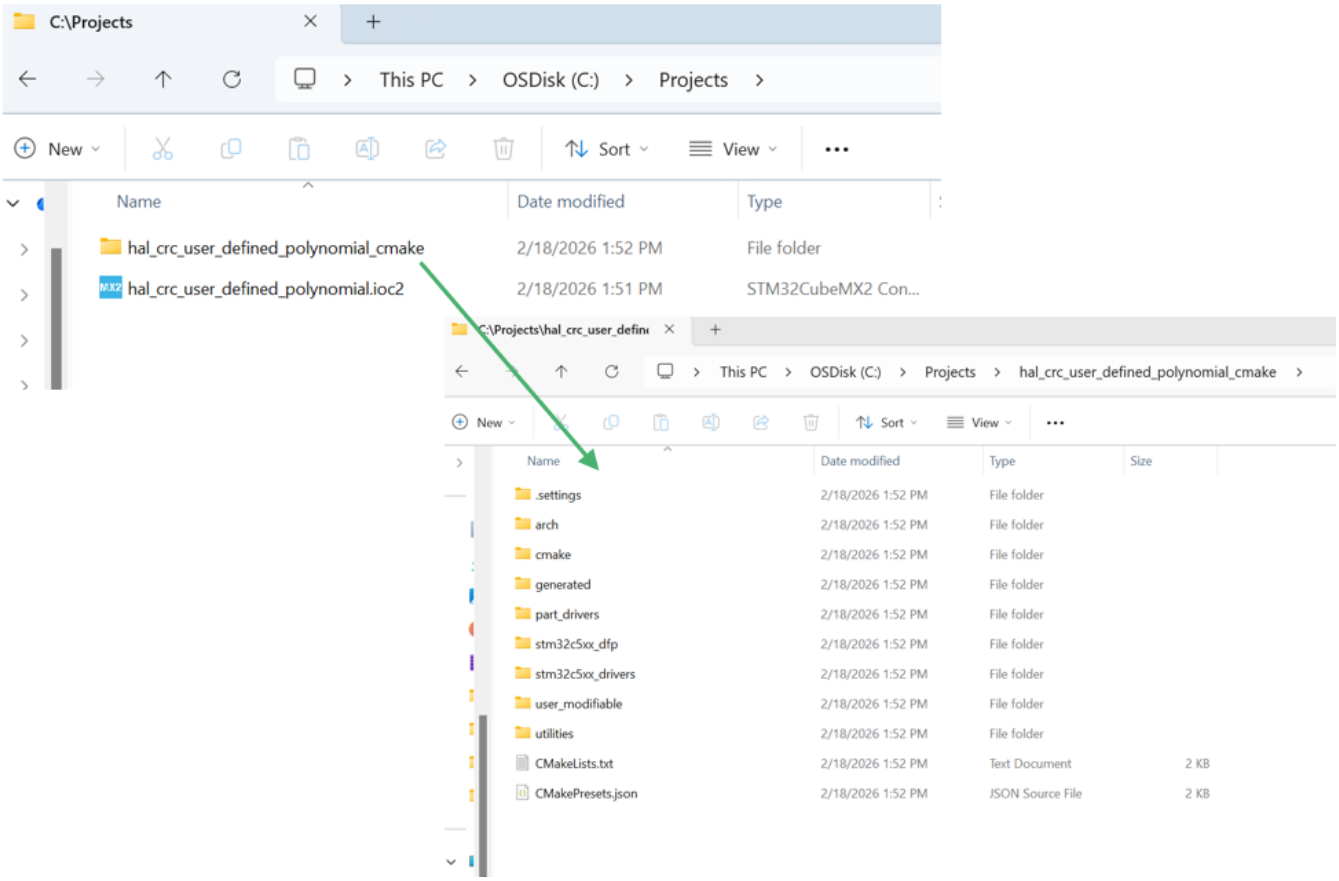


Fig. 30: generated project

Note

- The newly created ioc2 and IDE project are independent from the original STM32Cube example.
- The original example remains unchanged.

3.6.4 Additional articles

- How to create a new project with STM32CubeMX2

4 User manual

4.1 Home

The **STM32CubeMX2 home** page provides an easy-to-use starting point for **STM32** projects.

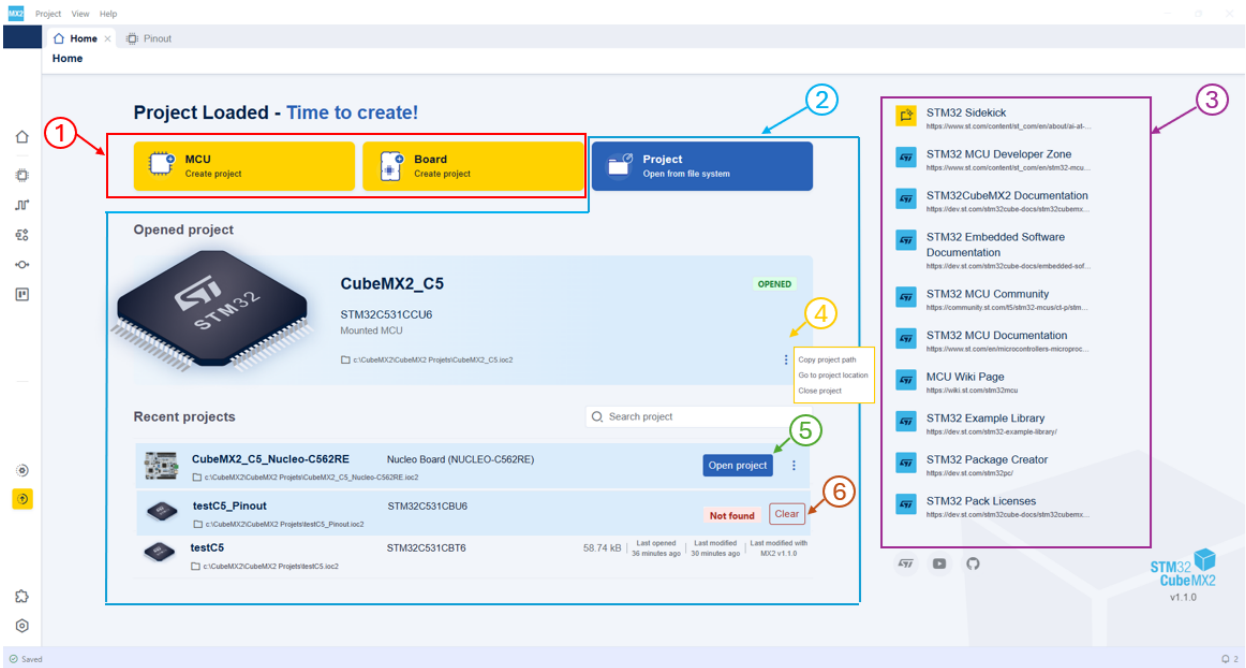


Fig. 1: STM32CubeMX2 home page

This window provides options for the following actions:

- **Start a new project from (1)**
  - **MCU:** The tool displays a list of supported **MCUs.**
  - **Board:** The tool displays a list of supported boards.
- **Open a project (2)**
  - **Open project:** Browse and select existing projects.
  - **Recent/current projects:** Open a recent or current project.
- **Access useful links to documentation and ST Community (3)**
- **Opened projects (4)** possible actions.
- **Open a project (5)** Moving the mouse over a line displays the button open project button.
- **Clear a project (6)** When a recent project is deleted, the **Clear** button appears. It informs the user that the project was removed and can be cleared from the list.

> **Note**
>
> - The left bar displays only the **Home**, **Pack Manager** and **Settings** icons when no project is opened.
> - The Current project is shown only if a project is open.

4.1.1 Create project from an MCU

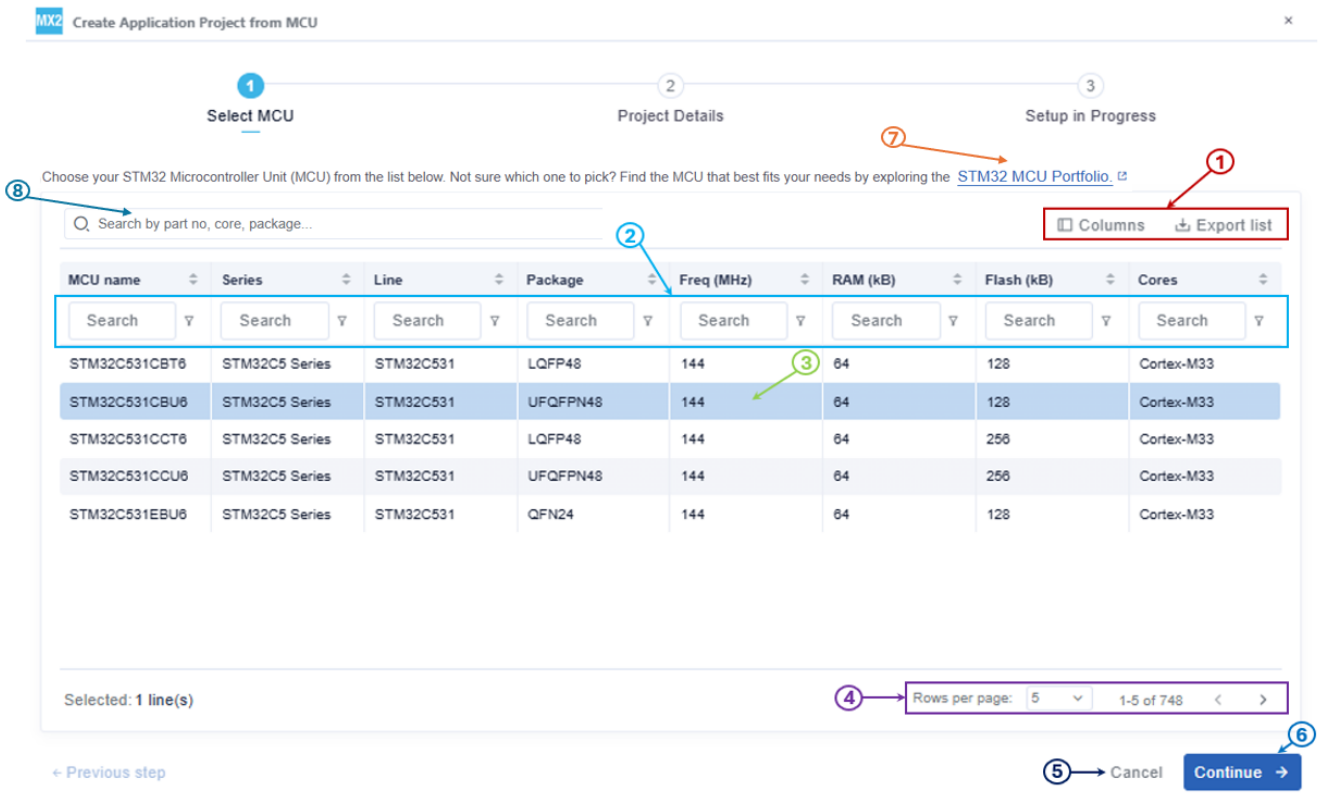


Fig. 2: Create a project from an MCU - select MCU

This panel assists users in selecting an **STM32** microcontroller and configuring project details for the **STM32** development environment.

- **Columns table settings (1):** Customize visible table columns.
- **Search & filter the MCU list (2):** Access the list of available **MCUs** by clicking on the **MCU** option. Search, apply filters, or scroll through the list to find the desired **MCU.**
- **Select an MCU (3):** Select the desired **MCU** from the list.
- **Control pagination (4):** Use pagination controls to navigate through different pages of the table view. Adjust the number of rows displayed per page.
- **Cancel creation (5):** Click the **Cancel** button to abort the current setup process.
- **Continue the project creation (6):** After selecting an **MCU**, click **Continue** button to continue with project creation.
- **Link to the STM32 MCU Portfolio (7)**
- **Search in all fields (8)**

After selecting an **MCU** configure the project details.

- **Enter project name (1):** Input the desired project name in the project name field.
- **Specify project location (2):** Select the directory for storing project files.

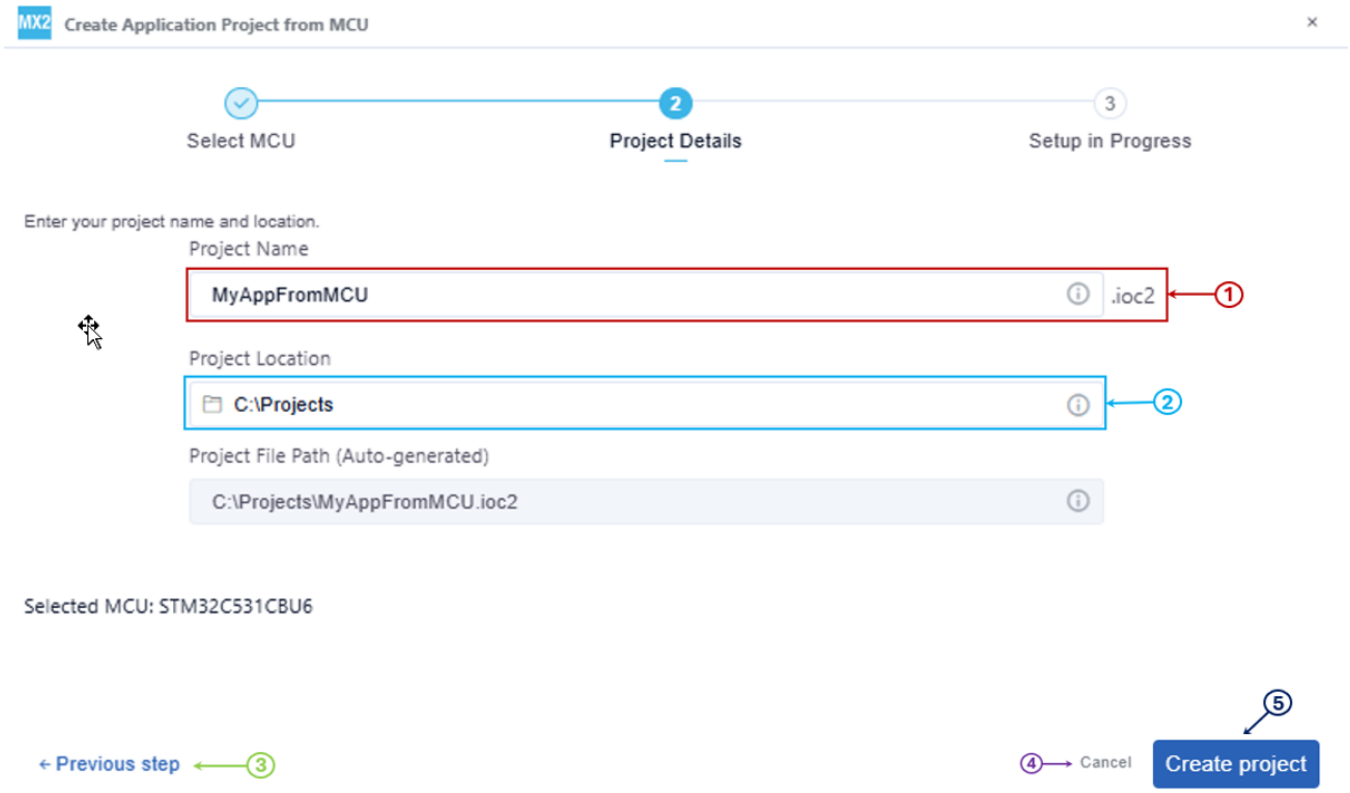


Fig. 3: Create a project from an MCU - project details

- **Navigate to the previous step (3):** Use the previous step option to return to the earlier page.
- **Cancel (4):** Click this button to abort the current project setup process.
- **Finalize and create a project (5):** Click **Create Project** to initiate the download and installation of necessary components. Wait for the progress bar to complete, then the project opens.

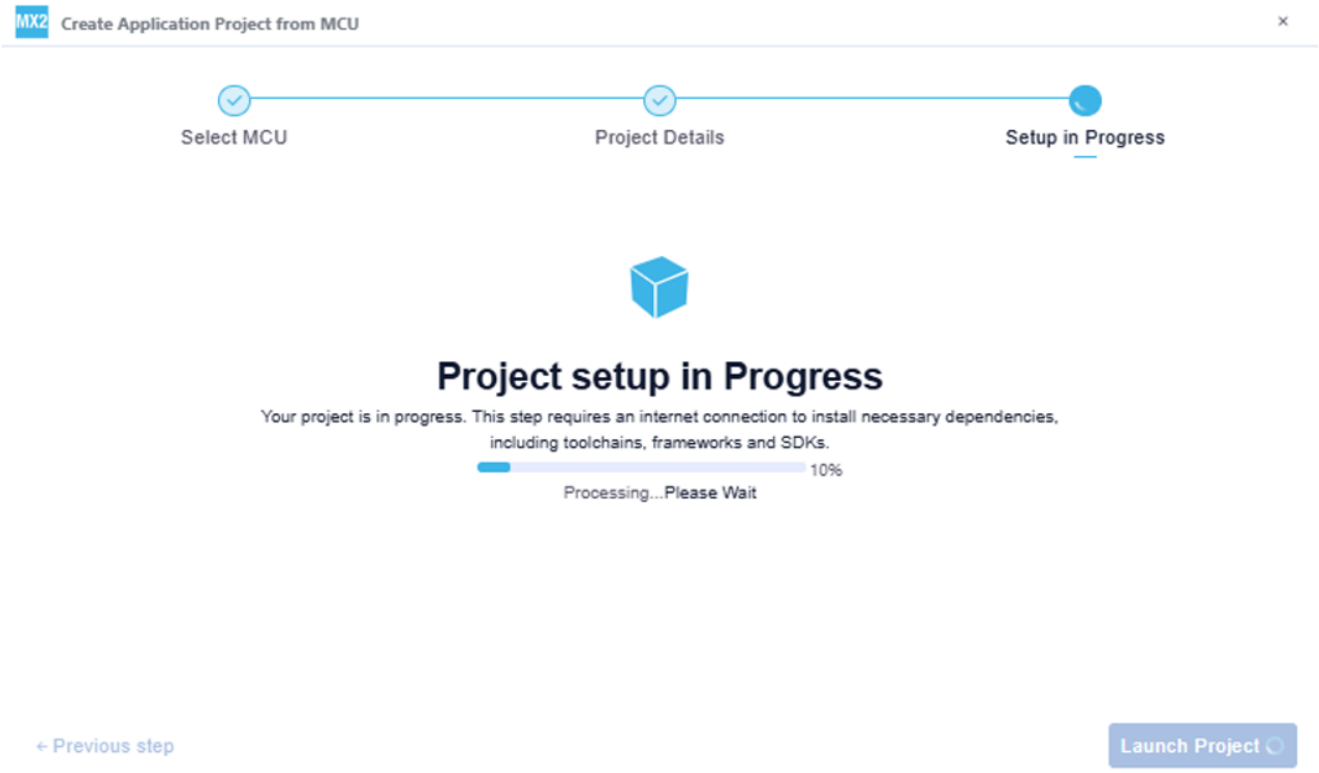
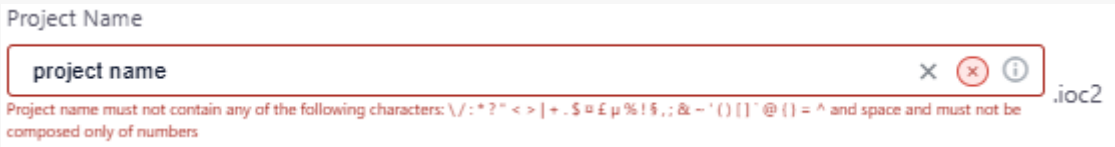


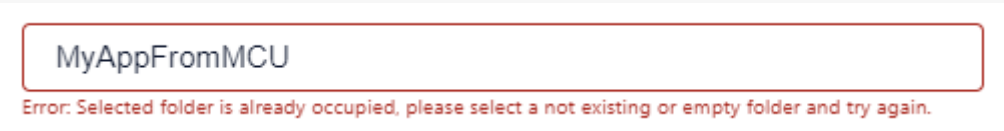
Fig. 4: Create a project from an MCU - setup in progress

Note

- When naming a project, do not use special characters. These characters are not permitted in project names.



- When creating a project with a name that already exists, a notification alerts you to potential conflicts. To proceed smoothly, select a unique project folder name.



4.1.2 Create project from a Board

This panel assists users in selecting an **STM32** board and configuring project details for the **STM32** development environment.

- **Columns table settings (1):** Customize visible table columns.
- **Search & filter the board list (2):** Access the list of available boards by clicking the **Board** option. Search, apply filters, or scroll through the list to find the desired board.
- **Selection of a board (3):** Select the desired board from the list.

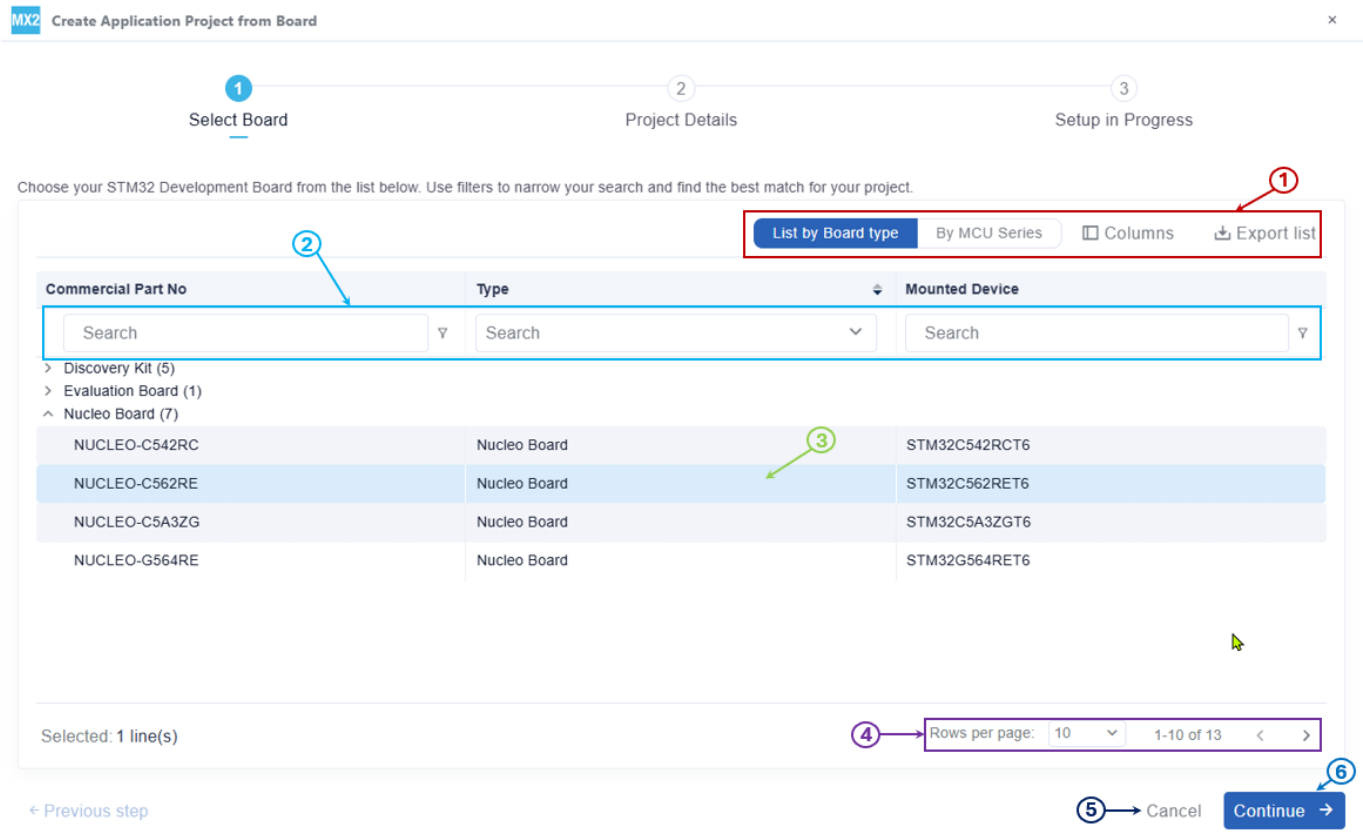


Fig. 5: Create a project from a board - select board

- **Pagination controls (4):** Use pagination controls to navigate through different pages of the table view. Adjust the number of rows displayed per page.
- **Cancel project creation (5):** Click the **Cancel** button to abort the current setup process.
- **Continue project creation (6):** After selecting a board, click the **Continue** button to proceed with project creation.

After selecting a board, configure the project details as described in the *Create project from an MCU* section, as shown in *Create a project from an MCU - project details* and *Create a project from an MCU - setup in progress*.

4.1.3 Open project

A project can be opened in the following ways:

- **File system (1):** Click the project button to browse and open a project from the file system.
- **Current project :** Open the current project in two ways.
  - Click **Open Project (2)** to open the application current project.
  - Click **Open Project (3)** to open a recent project by moving the mouse over the project line.
- **Clear deleted projects (4):** When a recent project is deleted, overpassing the mouse displays the **Clear** button. It informs the user that the project was removed and can be cleared from the list.

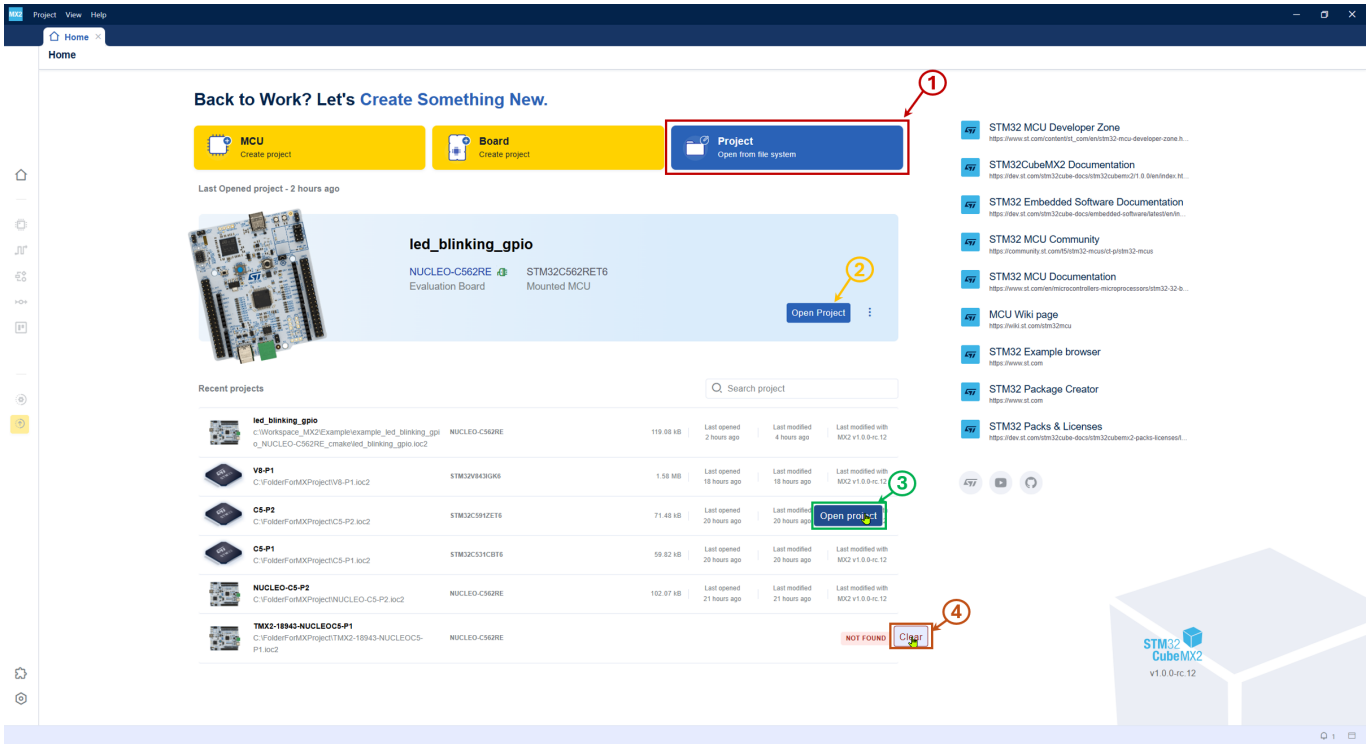
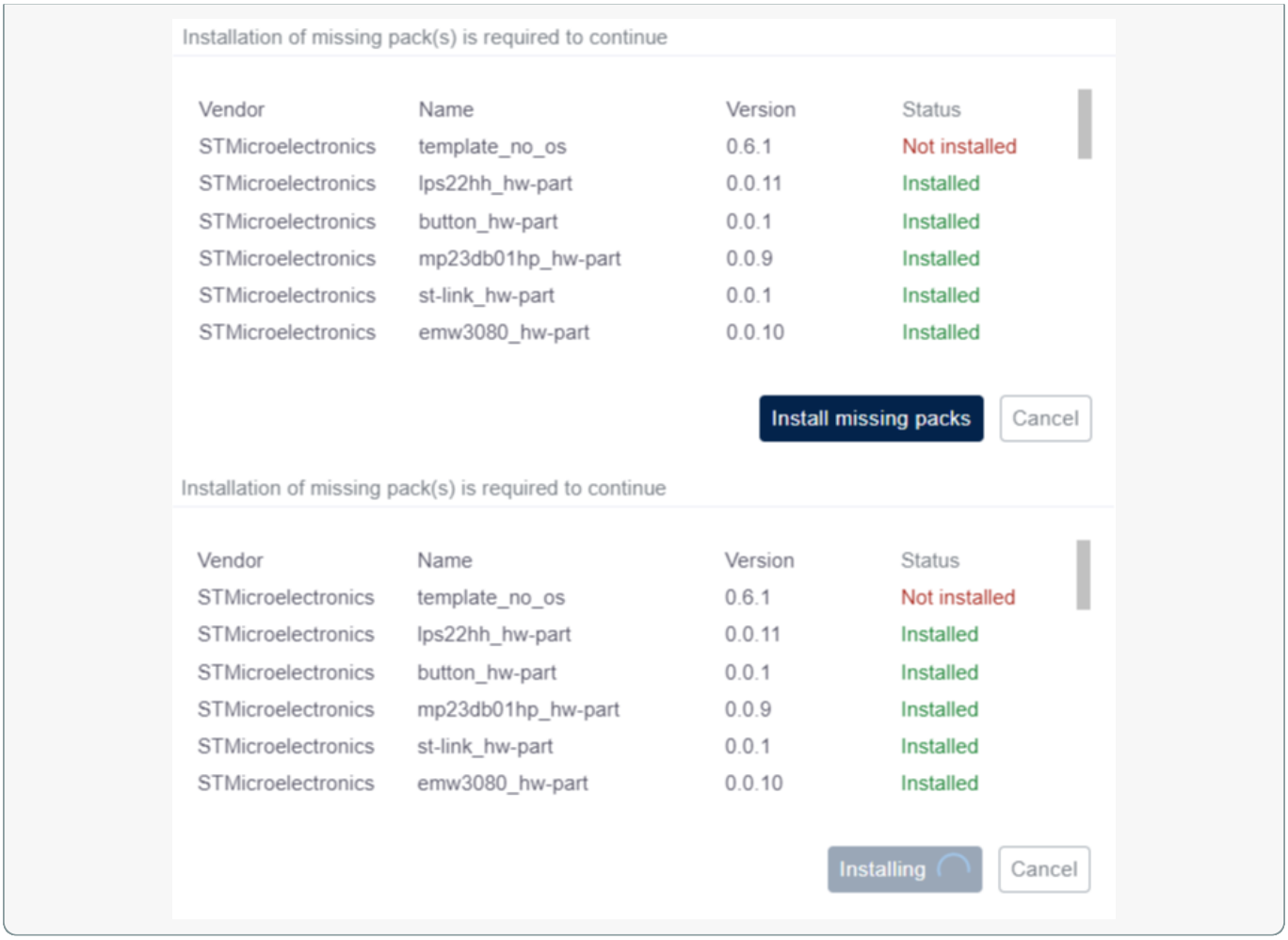


Fig. 6: Open project

Note

When opening a project, if any necessary packs are missing or have been deleted, **STM32CubeMX2** displays a notification. The missing packs must be installed to open and work with the project.



4.2 Project settings

The project settings page includes:

- **Project details (1):** Displays the project details, including the project name, location, and selected device.
- **IDE project generation settings (2):** Manages the **IDE** project. Check the *IDE project generation* for more details.
- **Packs details used in the project (3):** Manages the versions of packs used in the project. For more information, see *Packs details used in the project*.
- **Global Services (4):** Manages HAL Common and System Init parameters. Configure HAL System Init and HAL Common definitions by clicking the respective configuration button. For more information, see *Global Services configuration*.

Note

Use the scroll bar on the right side to navigate the page.

4.2.1 Global Services configuration

The **Global Services** section in the **Project settings** page allows users to configure the **HAL Common definitions** and **HAL System Init** parameters for the STM32CubeMX2 project. See *HAL System init configuration* flow.

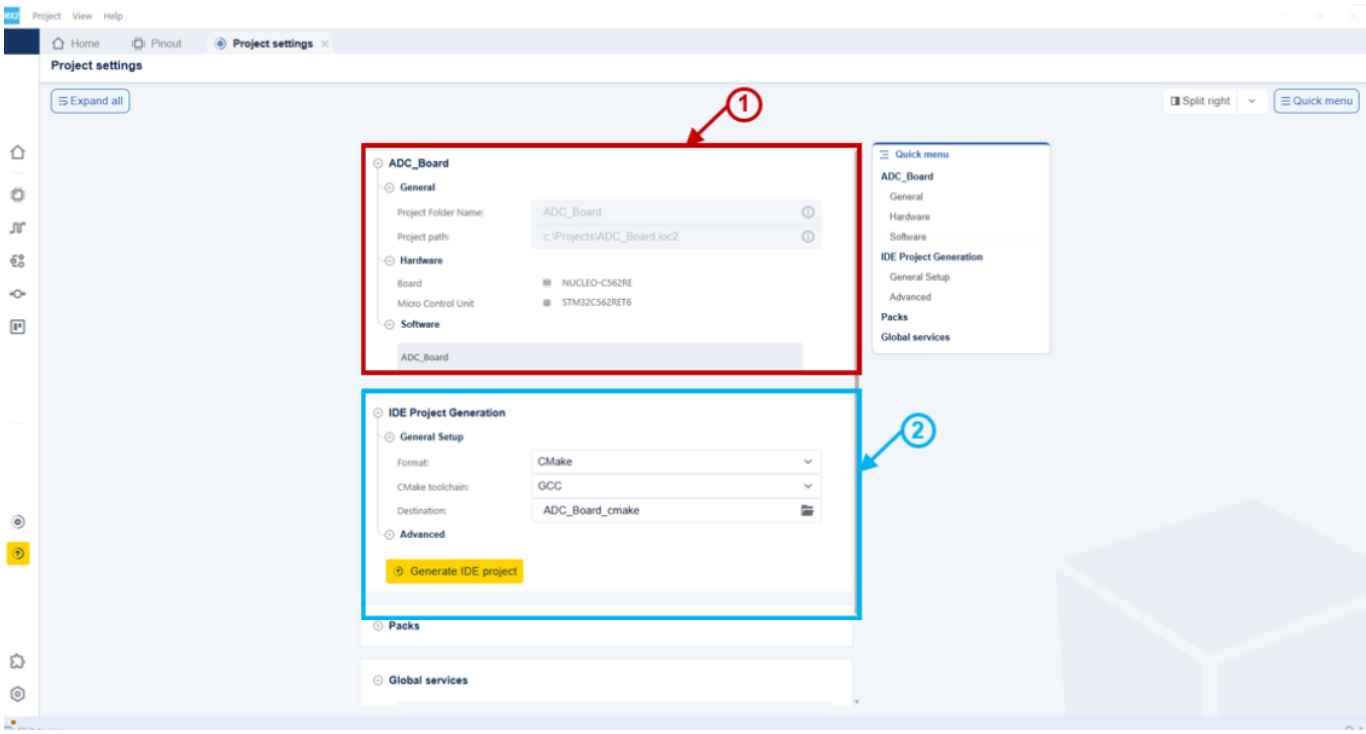


Fig. 7: IDE settings and Project details - project details & system init settings

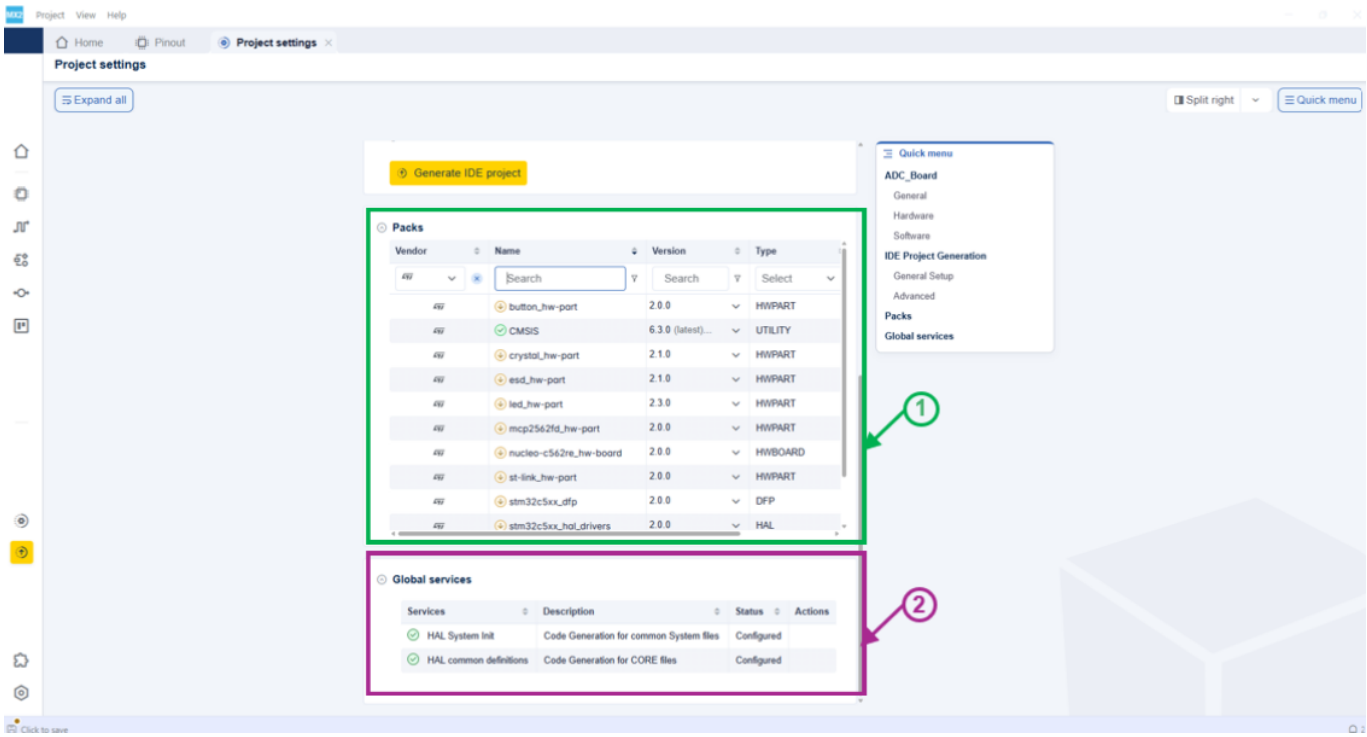


Fig. 8: Packs details and Global Services (HAL Common & System Init settings)

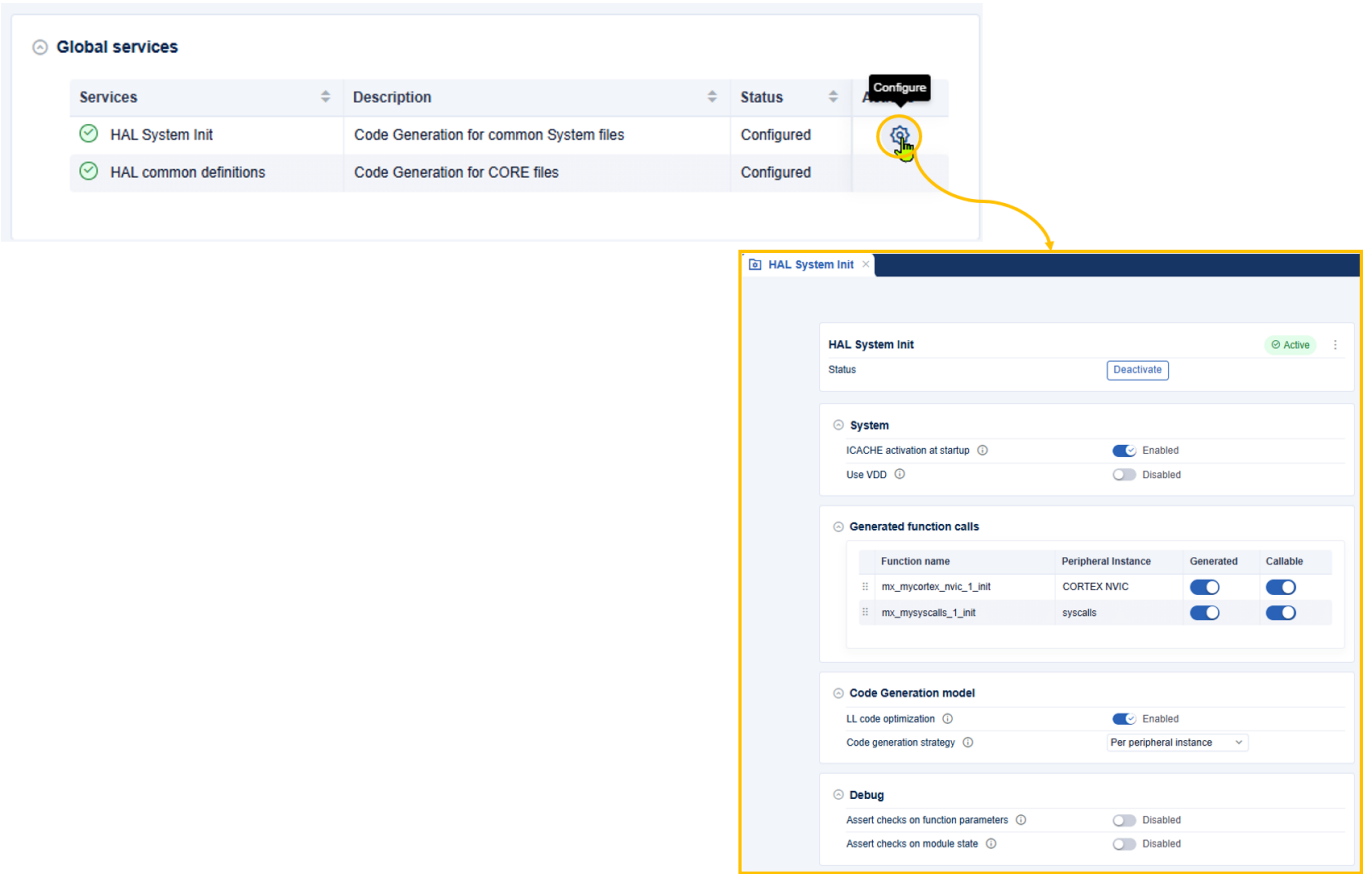


Fig. 9: HAL System init configuration

4.2.2 Packs details used in the project

The **Packs details used in the project** section allows users to update packs to a newer version for the STM32CubeMX2 project. See *Packs details used in the project* flow.

If a new pack is available, STM32CubeMX2 indicates that an update is possible by displaying the available pack version in the status column. Select the desired pack version from the dropdown list in the version column. Finally, click the **Update** button to apply the selected pack version to the project.

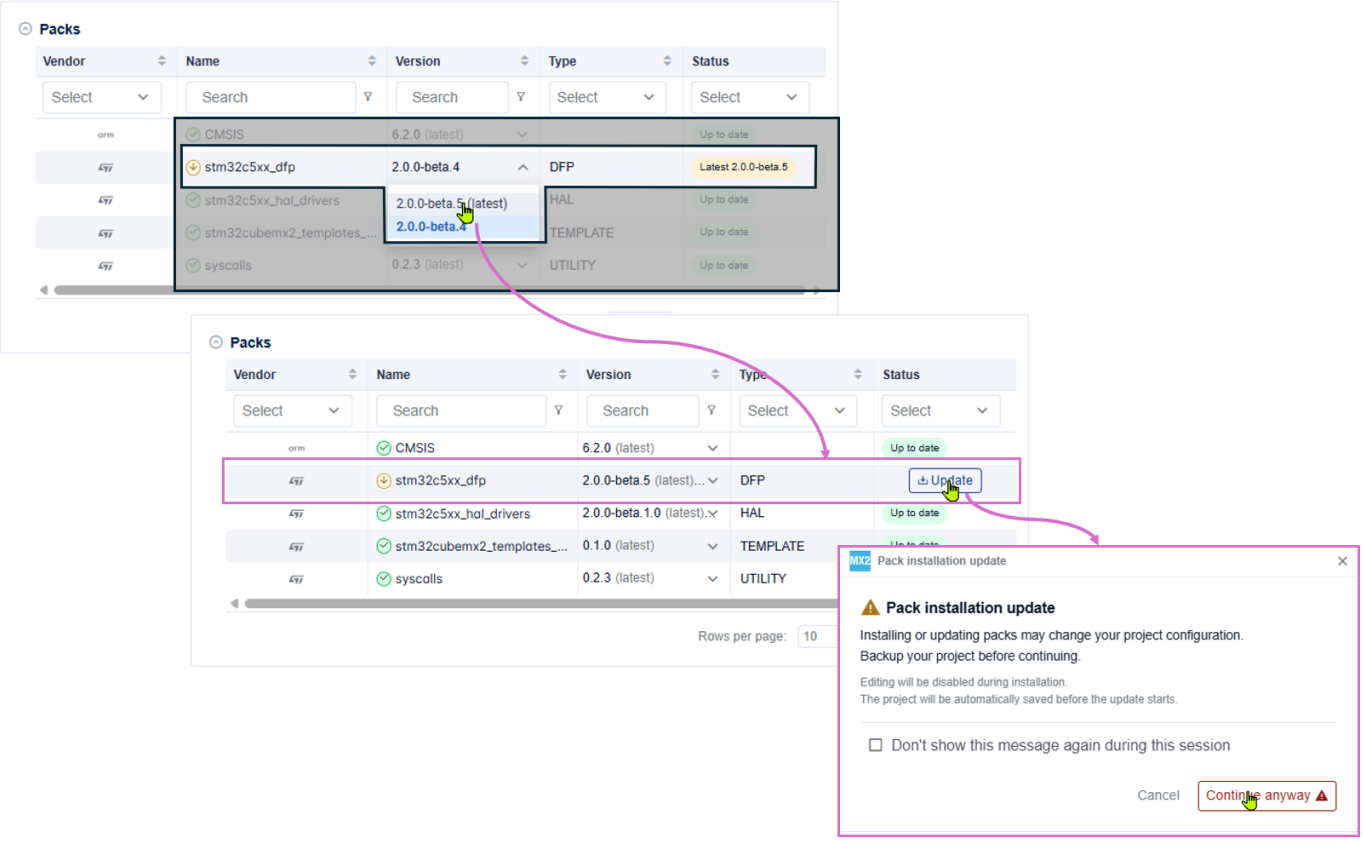


Fig. 10: Packs details used in the project

4.3 Clock

This interface enables microcontroller clock configuration and management, offering two view modes:

- **Graphic view:** Provides a schematic overview of the clock paths, clock sources, dividers, and multipliers(see *Clock tree*).
  - Drop-down menus and buttons allow modification of clock element values.
  - A zoom feature is available to enhance visibility.
- **Table view:** Displays all clock sources in a tabular format (see *Clock table*).

In both views, users can:

- **Switch view (1):** Switch between clock tree or table view.
- **Search (2):** Search for specific clock components.
- **Resolve clock issues (3):** Address any existing clock issues.
  - Conflicts are highlighted with red borders.
- **Reset clock (4):** Reset the clock configuration to default settings.

- **Lock/unlock (5):** Lock an element to prevent changes. It is possible to lock any type of clock element (for example, multiplexers, dividers, etc.)
- **Toggle detail view (6):** Display or hide the detail view panel.

The following functionalities are only available in the clock tree:

- **Configure RCC (7):** Navigate to the RCC configuration tab.
- **Highlight clock path (8):** Visually emphasize the selected clock path.
  - **Highlight all paths:** Highlight all related paths.
  - **Highlight occurrences:** Highlight every occurrence of the selected element.
  - **Highlight to origin:** Highlight the path tracing back to its origin source.

The following functionalities are only available in the graphic view:

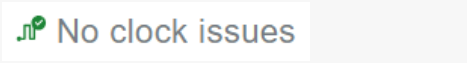
- **Navigation and zoom controls (9):** The bottom toolbar provides tools for zooming (see *Clock tree toolbar*).
- **Keyboard shortcuts (10):** Access a list of available shortcuts to speed up operations within the graphic view (see *Clock tree shortcuts*).

The following functionalities are only available in the table view:

- **Columns table settings (11):** Customize visible table columns.

Note

When no issues are detected in the clock view, the **Resolve clock issues** button changes to **No clock issues**.



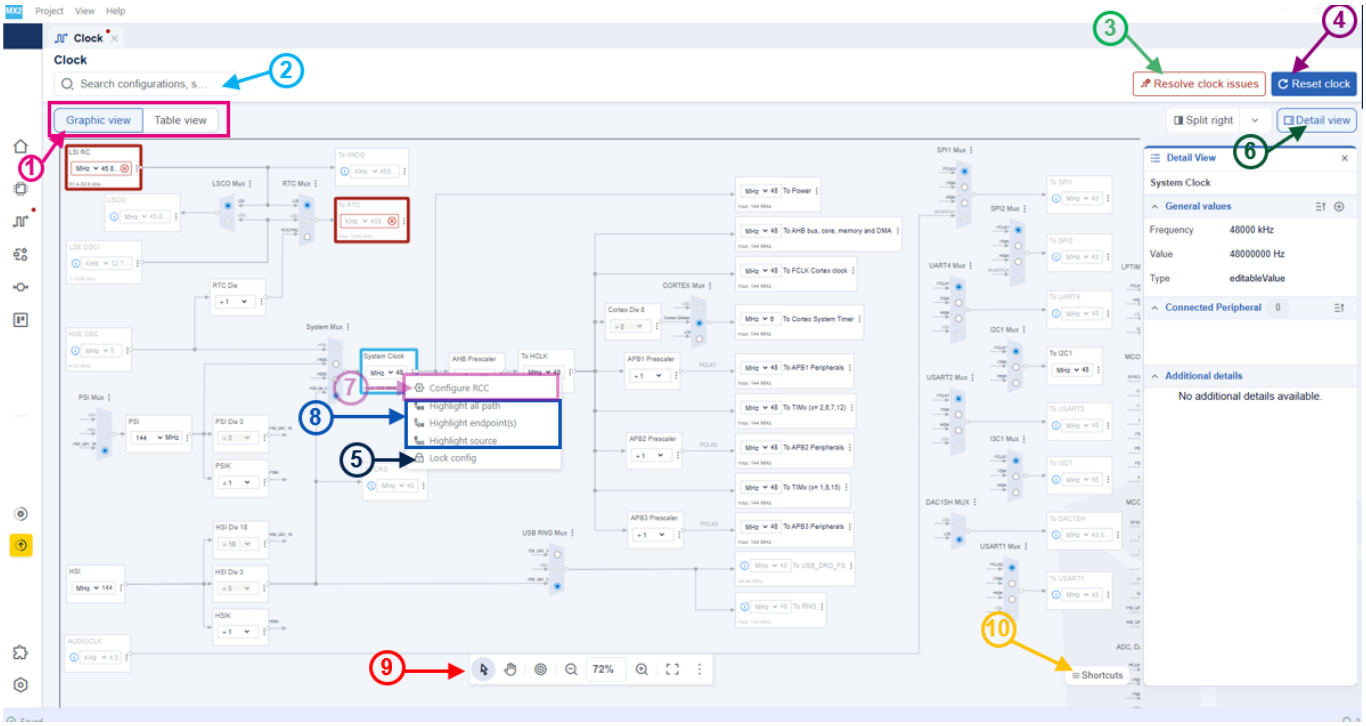


Fig. 11: Clock tree

To toggle between the hand cursor (move mode) and the selection cursor (edit mode), user can press the Esc key , which makes it easier to pan the graph and then quickly return to selecting and editing graph elements.

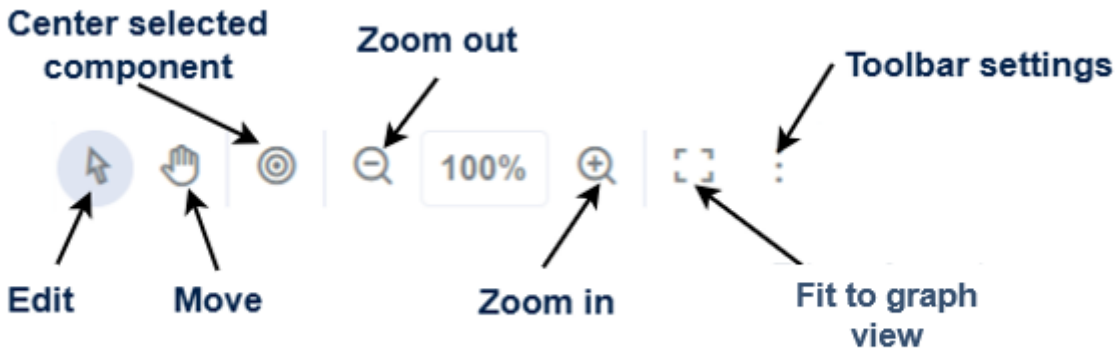


Fig. 12: Clock tree toolbar

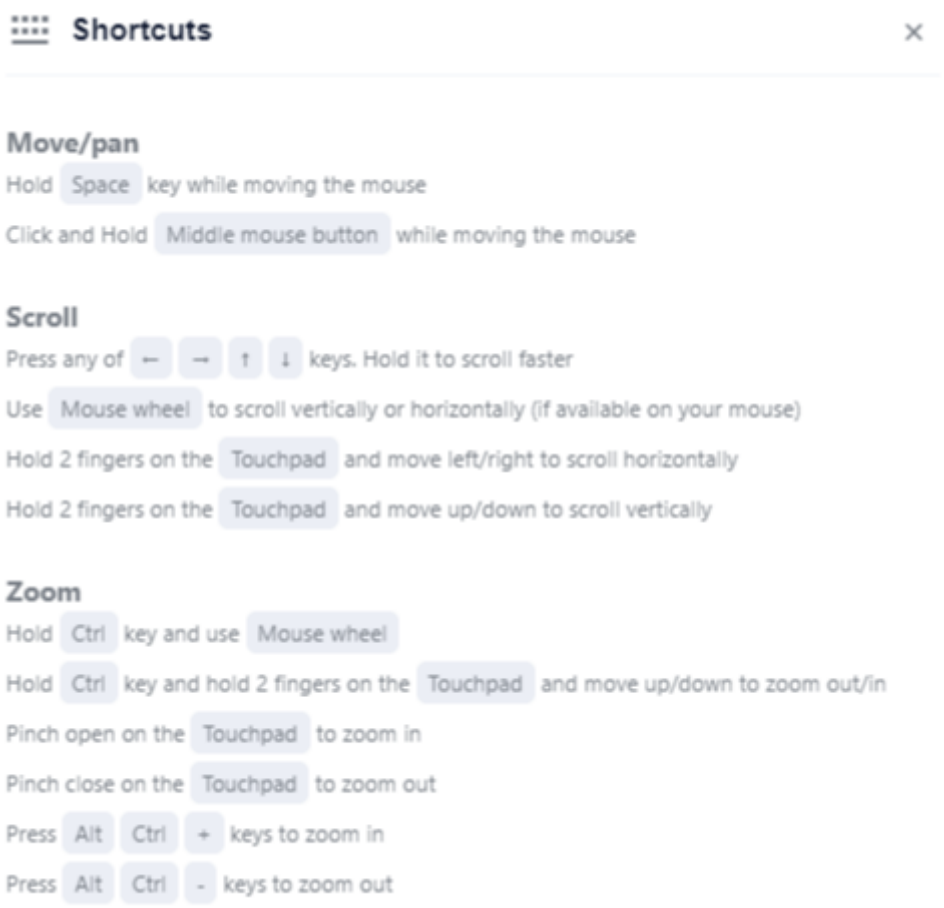


Fig. 13: Clock tree shortcuts

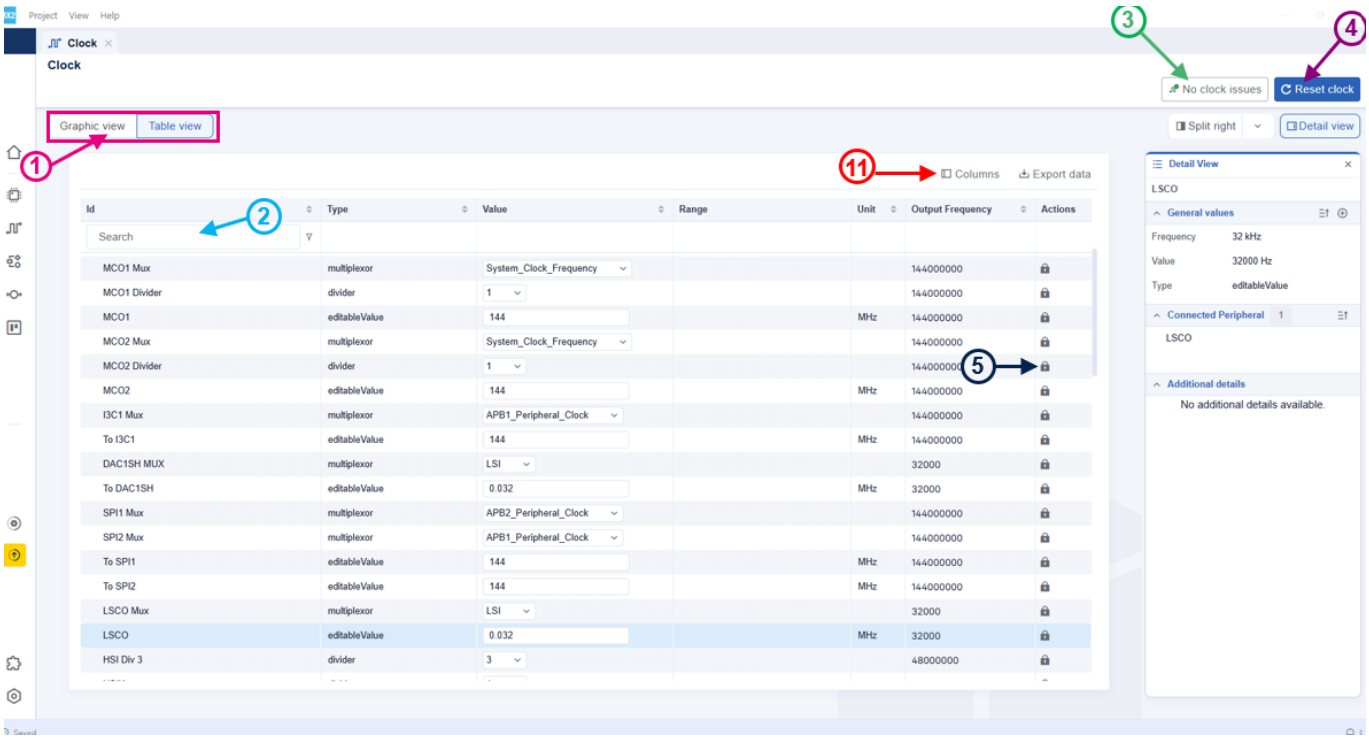
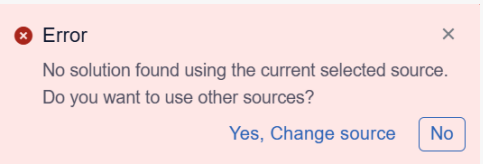


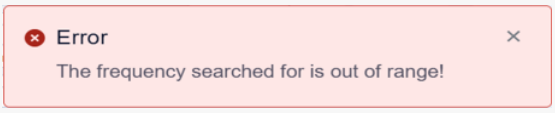
Fig. 14: Clock table

Note

- If you enter an invalid frequency for the selected clock sources, a message indicates that the current clock configuration does not provide a valid solution. You can change the clock source or keep the current configuration.



- When you enter a value that is not within the acceptable limits for the system, a notification appears stating:



4.3.1 Additional articles

- To get more information about RCC config, please refer to the RCC chapter in https://dev.st.com/stm32cube-docs/embedded-software/latest/en/index.html
- How to use clock configuration in STM32CubeMX2 for STM32C5
- How to use the MCO with STM32CubeMX2
- How to change the HAL2 TimeBase from Systick to a timer in STM32CubeMX2
- How to use RTC and backup registers with STM32CubeMX2

4.4 Pinout

This interface allows users to configure and manage the microcontroller pin assignments and properties. The **pinout view** is available in two formats:

- **Graphic view:** Displays the pinout in a graphical layout representing the physical package of the microcontroller, such as **BGA** or **QFP** (see *Pinout - graphic view*)).

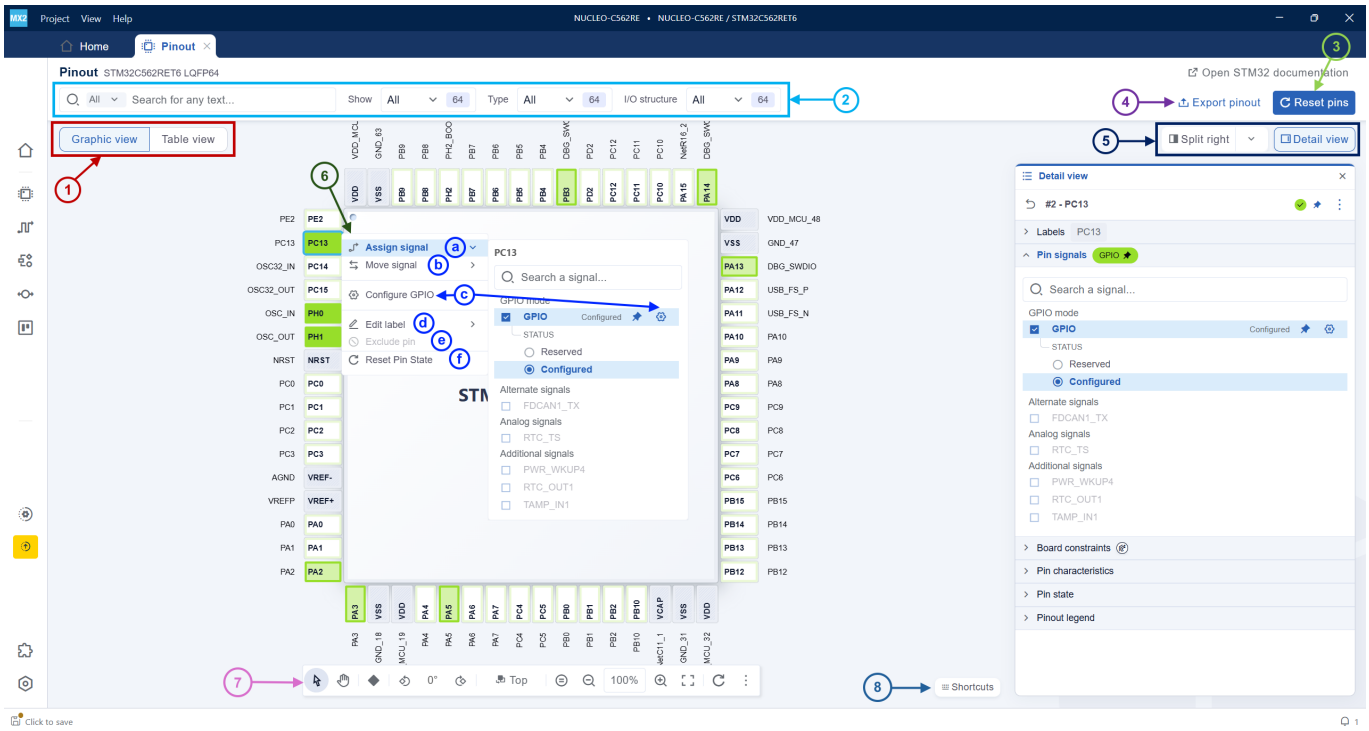


Fig. 15: Pinout - graphic view

- **Table view:** Presents the pinout information in a tabular format for easier reading and editing (see *Pinout - table view*).

In both views, users can perform the following actions:

- **Switch view (1):** Switch between graphic and table views.
- **Search and filter options (2):** Use dropdown menus to filter pins by type, function, I/O structure, or power.
- **Reset pins (3): Reset the pinout configuration. This action:**
  - Resets all reserved signals.
  - Preserves all configured signals except GPIO signals.
  - Preserves pin labels.
  - Restores excluded pins.
- **Export pinout (4): Export the pinout configuration to a CSV file.**
  - **Exports without alternate functions (4a):** The output file contains for each pin its position, name, type, the assigned signal(s) and the user label(s).
  - **Exports with alternate functions (4b):** The output file contains 16 additional columns (AF0..AF15) showing a map of all assignable signals on each pin.
- **Toggle detail view and split screen (5):** Enable or disable the detail view panel on the right side of the interface (see *Pinout - property view*), and split the screen horizontally or vertically.
- **Configure pinout (6): Includes the following options:**

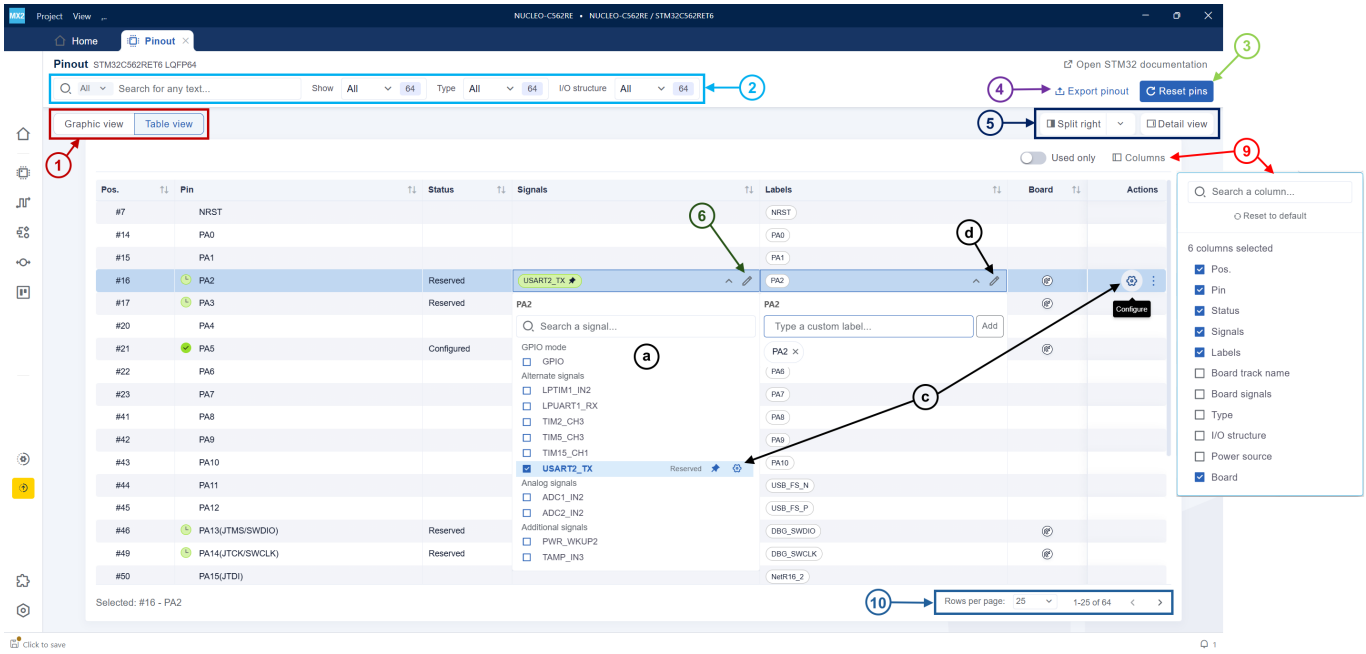
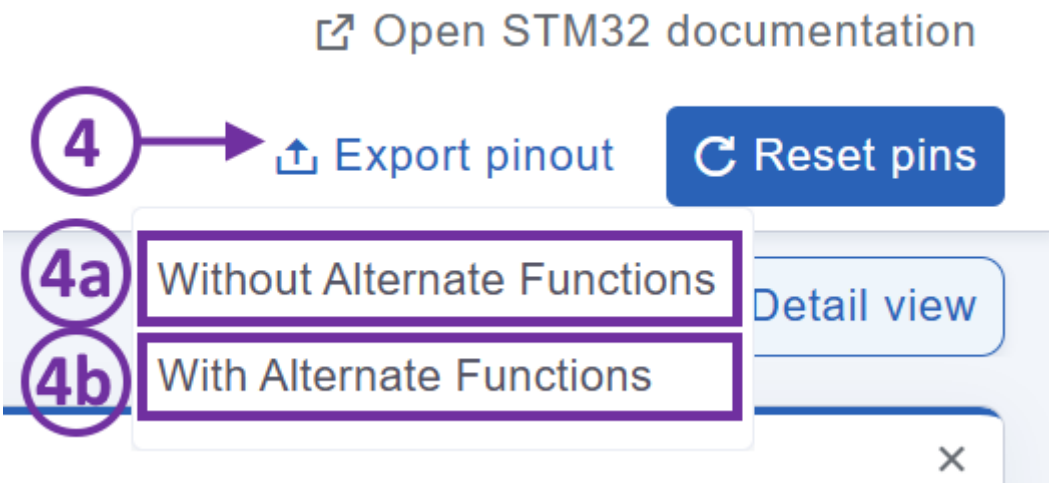


Fig. 16: Pinout - table view



- **Assign signal (a):** When clicked, a dropdown displaying available signals is presented. Navigate through this dropdown and choose the appropriate signal to link with the pin.
- **Move signal (b):** When clicked, the user can move the signal to an alternate pin.
- **Configure (c):** It opens the software configuration in a new window.
- **Edit label (d):** Assign a custom label to the pin.
- **Exclude pin (e):** Clicking this option excludes the pin from the pinout configuration.
- **Reset pin state (f):** Return the pin to unassigned state.

Note

- When opening the **graphic view**, locate the desired pin for signal assignment. **Right-click** the pin to access the context menu.
- **In STM32CubeMX2, signal allocation is distinct from configuration:**
  - Reserved pins are used to ensure that all necessary signals can be mapped. This is mainly used by board designers, as they do not need the generated code.
  - Configured pins are used to generate code for **GPIOs** and for signals used by peripherals configured in the project.

The following functionalities are only available in the graphic view (see *Pinout - graphic view*):

- **Navigation and zoom controls (7):** The bottom toolbar provides tools for zooming and package orientation. (See *Pinout - graphic view toolbar*).
- **Keyboard shortcuts (8):** Access a list of available shortcuts to speed up operations within the graphic view. (See *Pinout - graphic view shortcuts*).

The following functionalities are only available in table view (See *Pinout - table view*):

- **Columns table settings (9):** Customize the columns displayed in the table view by enabling or disabling specific columns.
- **Pagination controls (10):** Navigate through different pages of the table view.

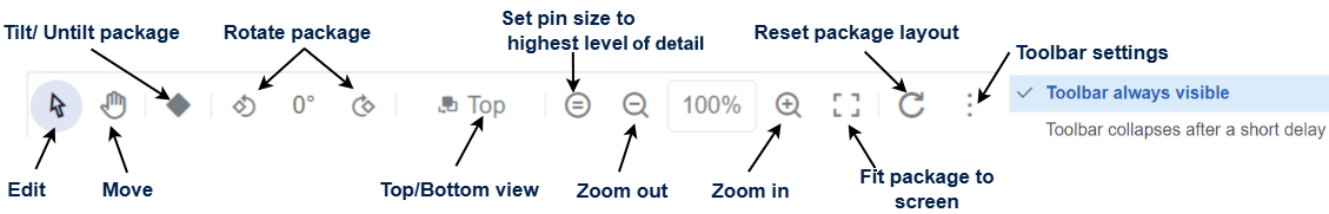


Fig. 17: Pinout - graphic view toolbar

Note

The **Tilt/Untilt package** option is only available for **QFP** and **QFN** packages.

The detail view serves three main roles in the pinout view:

- **Pins (1):** List all pins for the selected microcontroller and allow selection of a single pin.
- **Pinout legend (2):** Provide a detailed explanation of the **color coding** and **icons** used in the pinout graphic view.
- **Details on selected pins (3): When a pin is selected, the detail view displays information about its configuration and characteristics, including:**

Shortcuts

Move/pan

Hold Space key while moving the mouse

Hold Middle mouse button while moving the mouse

Press Home key to center the package in window

Scroll

Press any of ← → ↑ ↓ keys. Hold it to scroll faster

Use Mouse wheel to scroll vertically

Hold Shift key and use Mouse wheel to scroll horizontally

Hold 2 fingers on the Touchpad and move left/right to scroll horizontally

Hold 2 fingers on the Touchpad and move up/down to scroll vertically

Zoom

Hold Ctrl key and use Mouse wheel

Hold Ctrl key and hold 2 fingers on the Touchpad and move up/down to zoom out/in

Pinch open on the Touchpad to zoom in

Pinch close on the Touchpad to zoom out

Press Ctrl Alt + keys to zoom in

Press Ctrl Alt - keys to zoom out

Select

Press Tab key to put focus on the package or any pin

Press Enter key to select the focused pin

Press Escape key to unselect the selected pin

Other

Press ←Backspace key to flip the package

Press any of Page ↑ Page ↓ keys to rotate the package

Fig. 18: Pinout - graphic view shortcuts

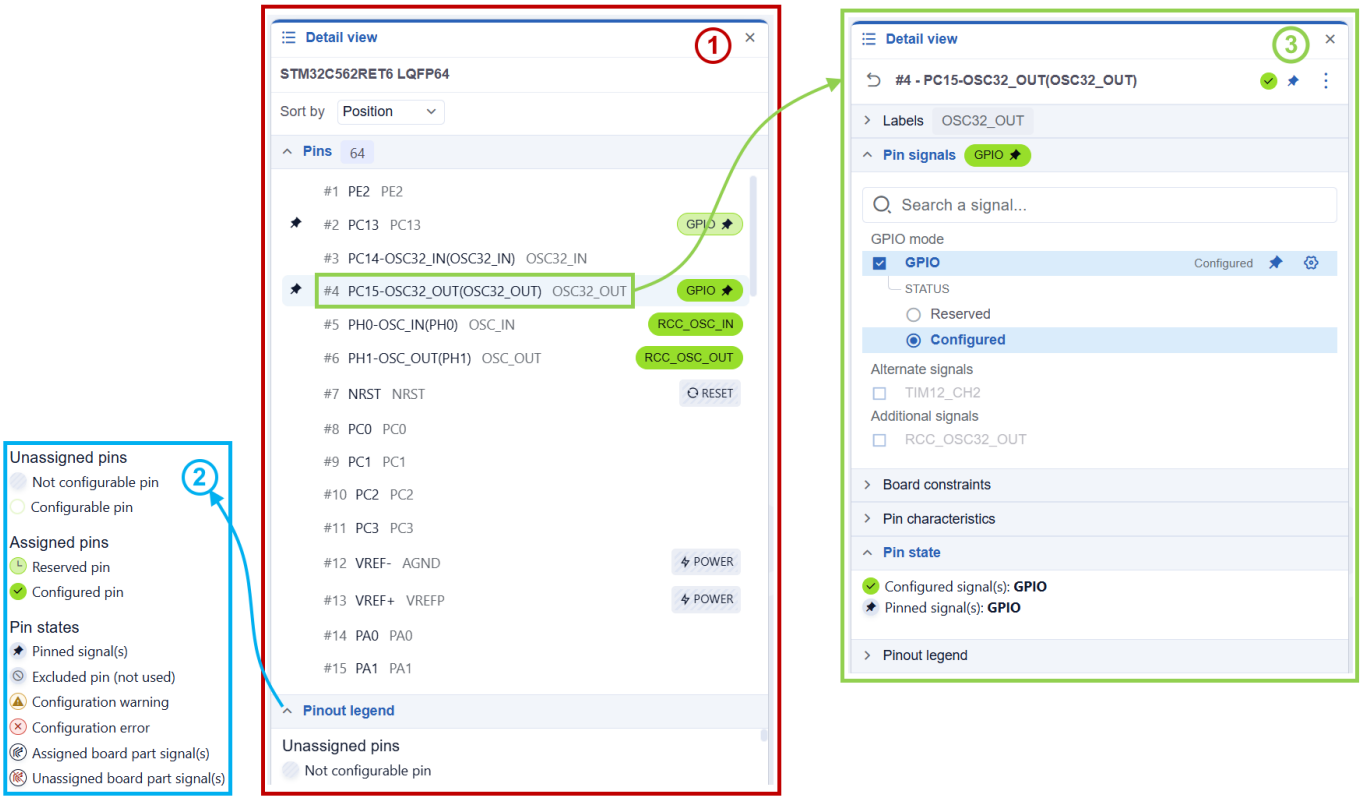


Fig. 19: Pinout - property view

- **Labels:** Add labels to the selected pin.
- **Pin signals:** Perform all actions described in **point 6: configure pinout**, such as assigning signals, adding or removing pin labels, and resetting pin states.
- **Board constraints (only if the project was created from a board):** Display constraints such as the track name.
- **Pin characteristics:** Show details such as input/output type and voltage supply.
- **Pin state:** It shows the current state of the pin.

4.4.1 Pinout multibonding

Multibonding refers to the configuration where multiple **MCU** pads share the same physical pin on a package. This feature is particularly relevant for packages with low pin count, as it allows efficient pin multiplexing to optimize limited pin availability in these compact packages.

To configure multibonding in STM32CubeMX2, follow these steps:

- **Select the pin (1):** Click the desired pin (for example, **PA1/PA2/PA3**).
- **Open assign signal menu (2):** Choose **Assign signal**.
- **Select the bonded pin (3):** From the list, select the desired pin (**PA2**).
- **View signals for PA2 (4):** Pick the required signal from the list and switch between options (for example, ADC1-IN2). You can select signals only on one pin.

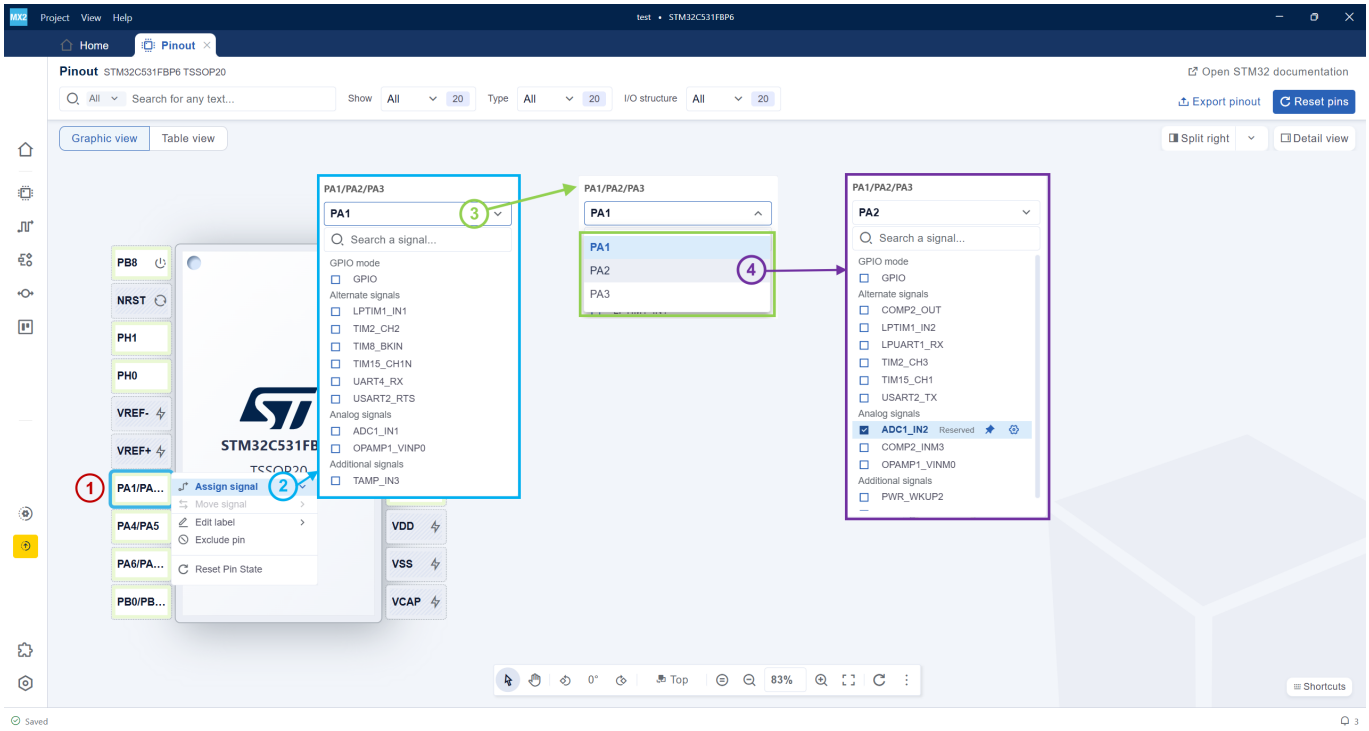


Fig. 20: Pinout multibonding

4.4.2 Pinout Conflicts

When the same pin is used for different functions, a pinout conflict arises. STM32CubeMX2 shows conflicts directly in the pinout or table view using visual indicators (for example, color coding or icons) to highlight the conflicting pins. To resolve pinout conflicts, users can take the following steps: - **Identify conflicting pins:** Look for visual indicators in the pinout view that highlight pins with conflicts.

- **Review peripheral configurations:** Check the configurations of the peripherals associated with the conflicting pins to understand their requirements.
- **Reassign pins:** Use the pinout view to reassign one or more of the conflicting pins to alternative available pins that do not have conflicts.
- **Validate changes:** After reassigning pins, validate the changes to ensure that the conflicts have been resolved and that all peripherals are correctly configured.

4.5 Menu bar and left bar

4.5.1 Menu bar overview

The menu bar in **STM32CubeMX2** is organized into three categories: **Project, View,** and **Help.** Each category contains options and submenus that facilitate different functionalities within the application.

- **Project menu:** The Project menu provides quick access to features available on the **Home** page. It focuses on project creation, management, and preferences. The following options are available:
  - **Create Project From MCU:** Start a new project using a specific **MCU**.
  - **Create Project From Board:** Start a new project based on a selected board.
  - **Open Project From File System:** Open an existing project from the file system.
  - **Open Recent Projects:** Access recently opened projects.
  - **Save Project:** Store changes to the current file.
  - **Save As:** Save a new copy with a different name or location.

  - **Close Opened Project:** Close the active project.
  - **Preferences:** Open the preferences settings for customization.
    * **Settings:** Open the settings dialog (shortcut: **Ctrl and the comma key (Ctrl + ‘,’)**).
    * **Theme Color:** Adjust the application theme color (shortcut: **Ctrl+K**, **Ctrl+T**).
  - **Exit:** Close the application (shortcut: **Ctrl+W**).
- **View menu:** The view menu is designed to manage various views and settings within the application. It includes:
  - **Zoom In:** Increases the view size to magnify on screen content (shortcut: **CTRL ++ or CTRL + =**).
  - **Zoom Out:** Decreases the view size to show more content at once (shortcut: **CTRL + - or CTRL +)**).
  - **Reset Zoom:** Restores the view to the default 100% zoom level (shortcut: **CTRL + ALT + K**).
  - **Output:** Display output logs (shortcut: **Ctrl+Shift+U**).
  - **Show Next Tab:** Switches to the next tab in the tab bar (shortcut: **CTRL + Tab**).
  - **Show Previous Tab:** Switches to the previous tab in the tab bar. (shortcut: **CTRL + Shift + Tab**).
  - **Close Tab:** Closes the currently active tab (shortcut: **CTRL + ALT + P**).
- **Help menu:** The help menu provides access to documentation and support options. It includes:
  - **About:** Display information about the application.
  - **Documentation:** Open the application related documentation.
    * **STM32CubeMX2 :** MX2 online documentation.
    * **STM32 Embedded Software :** Embedded SW top level Page.
    * **STM32 MCU:** Online resources related to STM32 microcontrollers.
  - **STM32 Packs & Licences:** STM32Cube MCU software packages information (HAL/LL drivers, middleware, BSPs, examples) and their associated license terms and conditions.
  - **Issue reporter:** Export the cube environment as a zip file to facilitate issue reporting.

Report issue

To report an issue with **STM32CubeMX2,** in the **Help menu,** select **Issue Reporter,** and follow the prompts.

4.5.2 Left bar overview

The left sidebar contains quick access icons for frequently used features. The following table summarizes these icons:

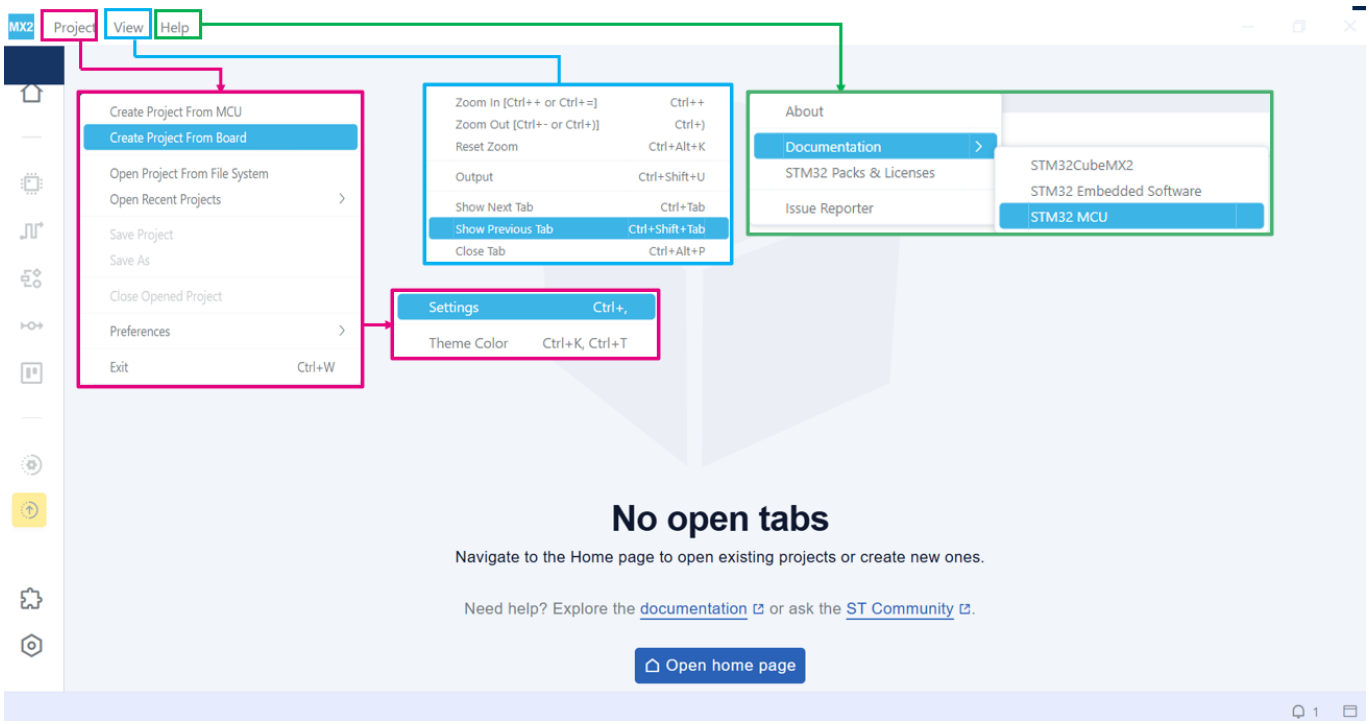


Fig. 21: Menu bar overview

	Ico

> **Note**
>
> The left bar displays only the **Home, Pack manager,** and **Settings** icons when no project is open.

4.6 Peripherals

The **Peripherals** panel in **STM32CubeMX2** provides the following features and capabilities (see *Peripheral panel*):

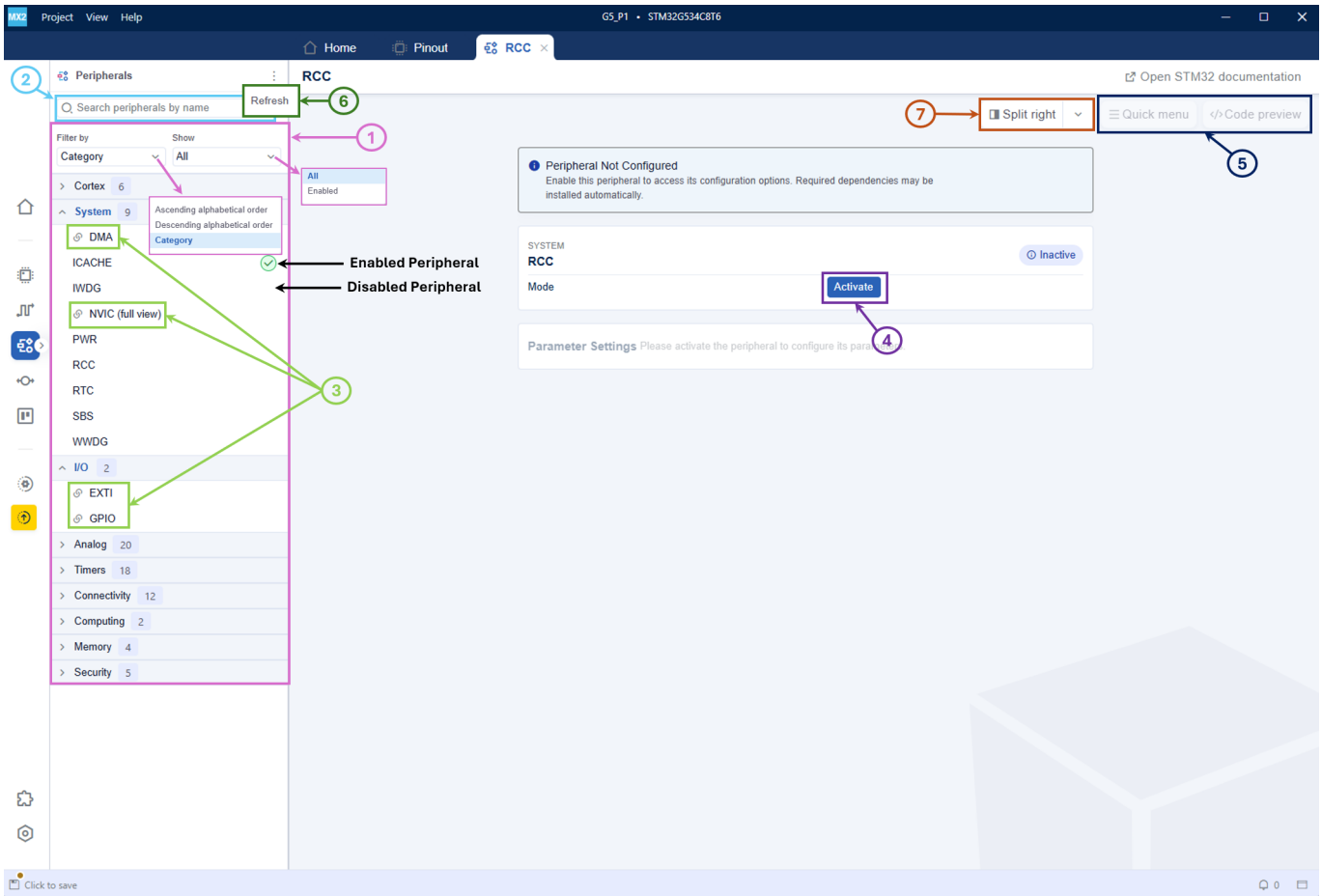


Fig. 22: Peripheral panel

- **Display the list of supported peripherals (1): The list includes components such as ADC, timer, UART, I²C interface, and others.**
  - Users can filter the list by category or sort it alphabetically in ascending or descending order.
  - Users can filter the list by status to display enabled peripherals or all peripherals.
- **Search for a peripheral (2):** Use the search function to find a peripheral without scrolling through the entire list.
- **Access GPIO. DMA, EXTI, and NVIC views (3):** Click to open the corresponding view in a separate window. See *DMA*, *EXTI*, *NVIC* sections for further information.
- **Enable or disable a peripheral (4):** Click a peripheral to enable it and access its configuration, or disable it if already enabled. See *Peripherals configuration* section below for further information.
- **Code preview and Quick Menu (5):** When a peripheral is enabled, users preview the content of the associated *.h* and *.c* files and navigate the settings panel. See *Code preview* and *Quick Menu* sections below for further information.

- **Refresh (6):** If the UI takes a long time to refresh, use this button to update the display.
- **Split view (7):** Split the display horizontaly or verticaly.

> **Note**
>
> - Activating a peripheral automatically assigns its corresponding **MCU** pins (if needed) in the **Pinout view**, displaying the required connections for operation. The software detects and signals resource conflicts when pin assignments or peripheral settings overlap.
> - The **Code preview** button is grayed out when a peripheral is disabled.

4.6.1 Peripherals configuration

The **Peripherals configuration page** in **STM32CubeMX2** provides the following features and capabilities (see *Peripheral configuration window*):

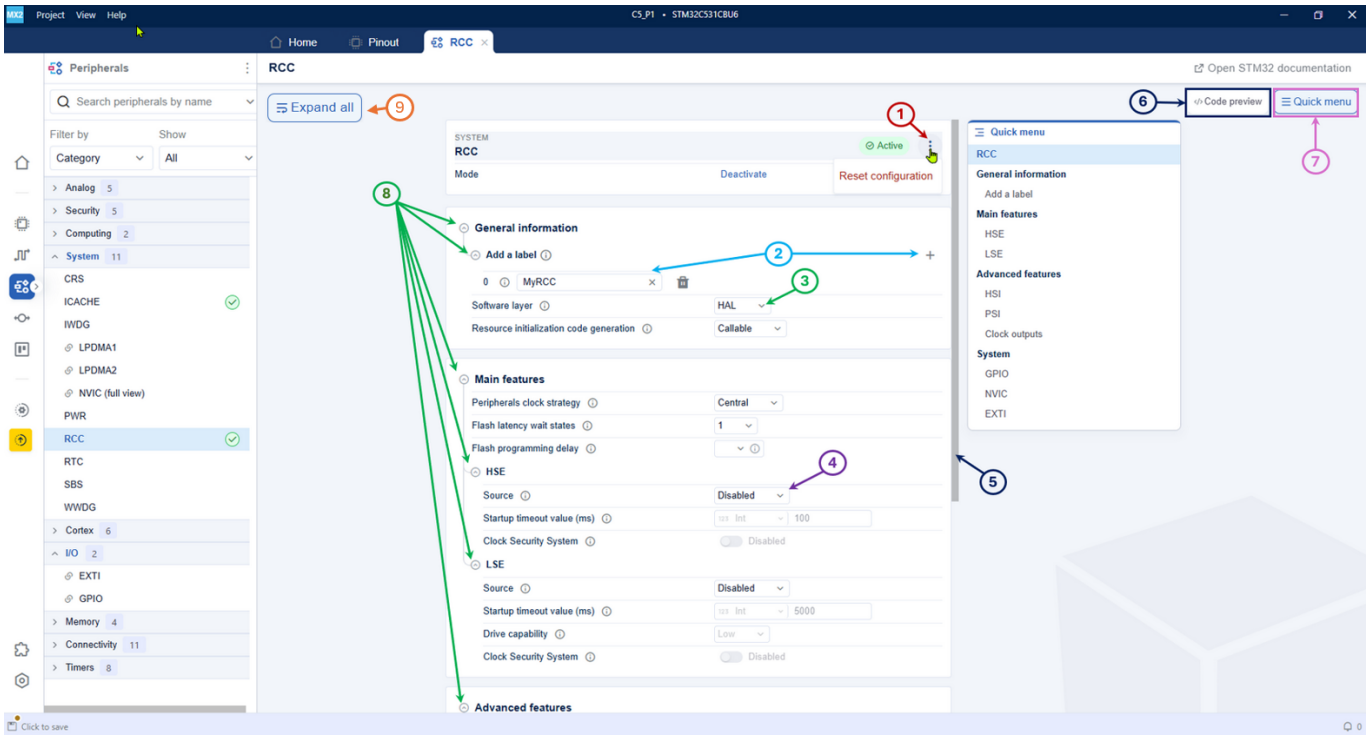


Fig. 23: Peripheral configuration window

- **Reset configuration (1):** Reset the settings of the selected peripheral to their default state.
- **Input fields (2):** Define specific values or text for parameters.
- **Dropdown fields (3):** Configure specific parameters of the peripheral by selecting a value from a dropdown list.
- **Checkbox fields (4):** Enable or disable specific parameters.
- **Scroll bar for additional settings (5):** Access additional configuration options for the peripheral that are not visible
- **Code preview (6):** Open the **Code preview** to view the *.h* and *.c* files.
- **Quick menu (7):** Toggle the Quick menu window, that ease the settings panel navigation.
- **Expand/collapse sections (8):** Show or hide additional configuration options by expanding or collapsing sections.

- **Expand All / Collapse all (9):** Expand all the configuration to see all parameters or Collapse All sections to keep main sections.

4.6.2 Code preview

The **Code preview** panel in **STM32CubeMX2** provides the following features and capabilities (see *Code preview*):

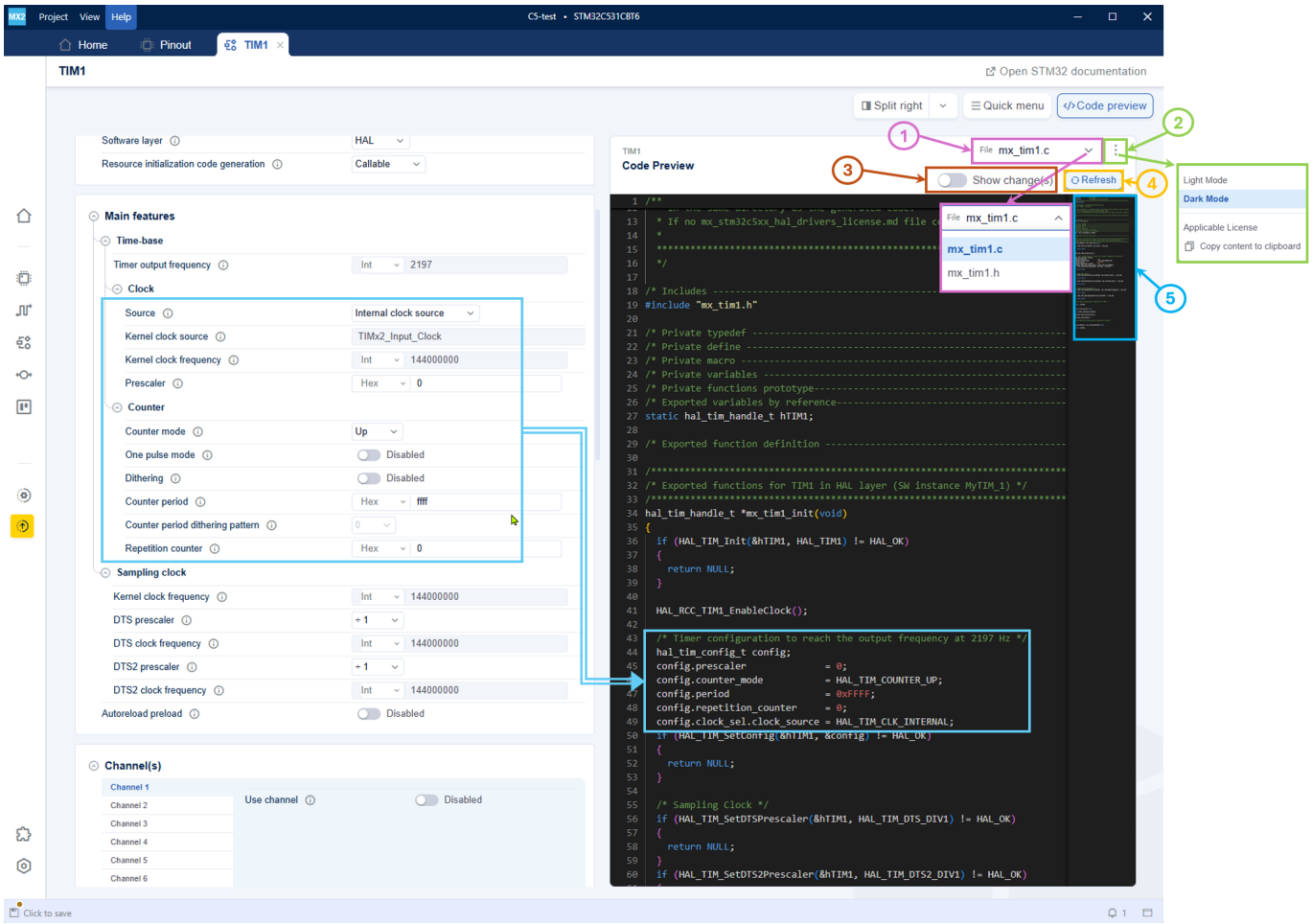


Fig. 24: Code preview

- **File selection (1):** The file selection dropdown allows the user to toggle between *.h* and *.c* files.
- **Settings menu (2):** The settings menu provides options such as toggling between **Light Mode** and **Dark Mode,** applicable license, and copy content to clipboard.
- **Show change(s) (3):** To show changes after the modification of a parameter.
- **Refresh view (4):** The refresh button updates the static code preview to reflect the latest changes made in the configuration settings.
- **Minimap (5):** The minimap allows users to navigate through the preview file.

4.6.3 Quick Menu

The **Quick Menu** panel in **STM32CubeMX2** provides easy navigation in the parameter page. Click any line to move the settings panel to the corresponding section.

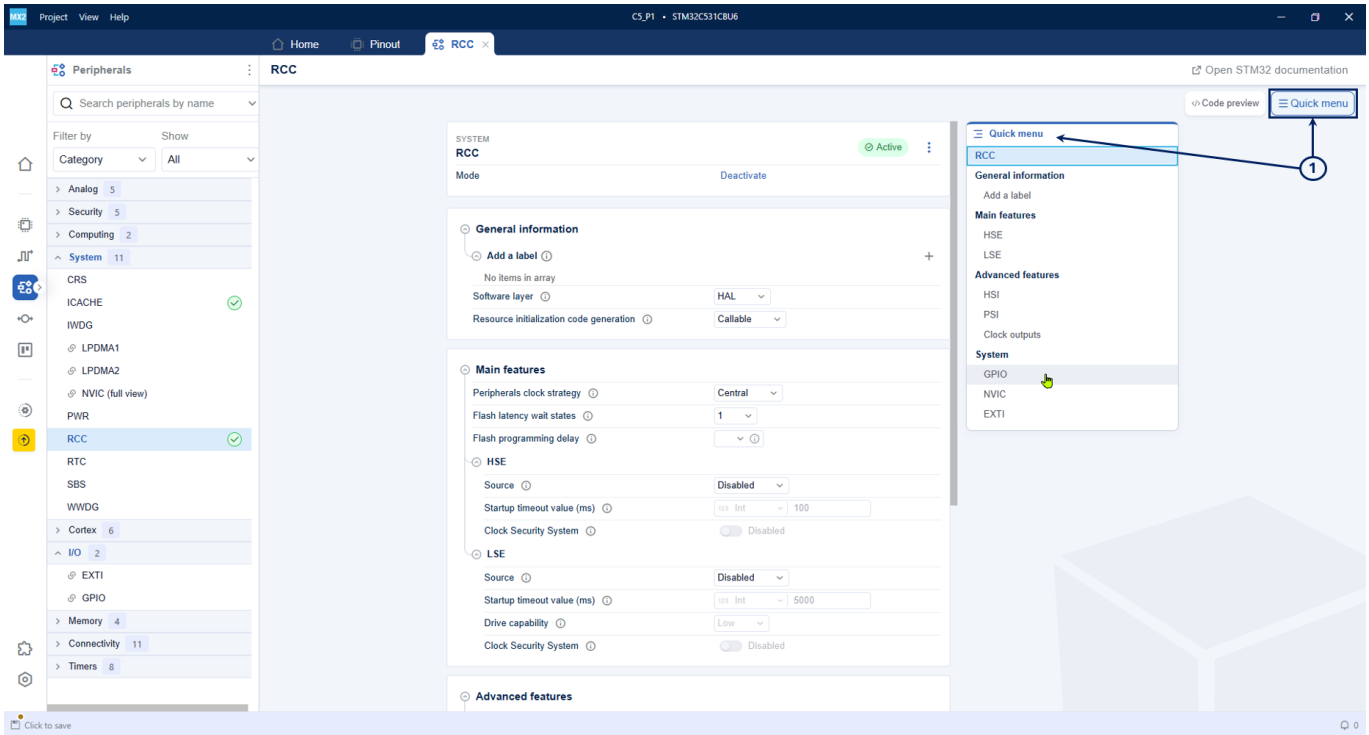


Fig. 25: Quick Menu

4.6.4 Additional articles

- How to use UART with STM32CubeMX2
- How to use TIM with STM32CubeMX2
- How to use SPI with STM32CubeMX2
- How to use I2C with STM32CubeMX2
- How to use ADC with STM32CubeMX2
- How to use DAC with STM32CubeMX2
- How to use FDCAN and HSE with STM32CubeMX2

4.7 DMA

This interface displays the **DMA channels.** The **DMA** view is accessible from the peripheral view and provides the following features and capabilities (see *DMA system view*) :

- **Select DMA controller (1):** Toggle between **DMA** controllers to manage channels, including reserving or configuring channels.
- **Hide unused channels (2):** Hide unconfigured channels to focus on active or reserved channels.
- **Columns (3):** Update the list of visible columns.
- **Filter & search (4):** Use dropdown menus and input fields to filter and search through the displayed channels.
- **Requests(5):** View the list of requests associated with each channel, potential conflicts, and reservation status.
- **Manage DMA requests through Actions column (5.1):** Reassign DMA requests from one channel to another by moving or switching them. This allows for better management of DMA resources and helps resolve conflicts by reallocating requests to available channels.
- **Manage DMA requests through Request column (5.2):** Reserve or move DMA requests.

- **Go to Configuration (6):** Open the peripheral configuration for the selected DMA channel in a new tab.
- **Reset Channel (7):** Remove the reservation for a channel and return it to its default state.

> **Note**
>
> The **Reset Channel** option is grayed out when a channel is configured and used by a peripheral. This functionality is available only for channels in the reserved state or for memory-to-memory configurations.

- **Split (8):** Split the tab into two separate views for better visibility.
- **Toggle detail view panel (9):** Toggle the detail panel for the selected DMA controller and its channels (see *DMA system view - detail view*).
- **Open STM32 documentation (10):** Open the STM32 documentation related to the selected DMA controller and its channels.

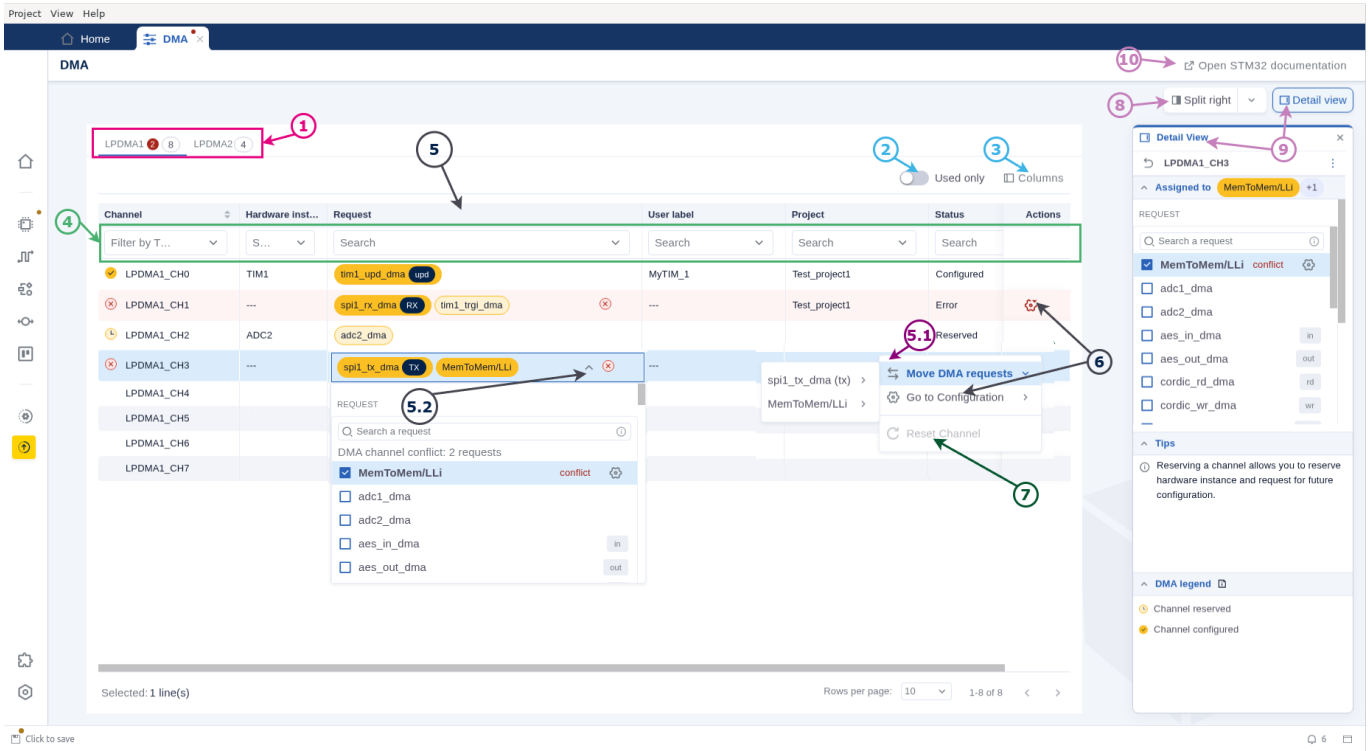


Fig. 26: DMA system view

The detail view provides the following features and capabilities:

- **List of available channels (1):** If no channel is selected, the panel displays all channels of the DMA controller and their statuses (see left side of the *DMA system view - detail view*). Conflicts per channel are visible.
- **Tips section (2):** Provides guidance and contextual information about the **DMA** configuration process.
- **DMA legend (3):** Explains the status of channels with visual indicators.
- **Channel reservation (4):** When a specific channel is selected, the panel shows detailed information about it, including its current assignment. You can reserve a signal for future configuration, view existing configurations, reset the channel, or unassign reserved channels.
- **Return to all channels (5):** Switch the view back to the main detailed view.

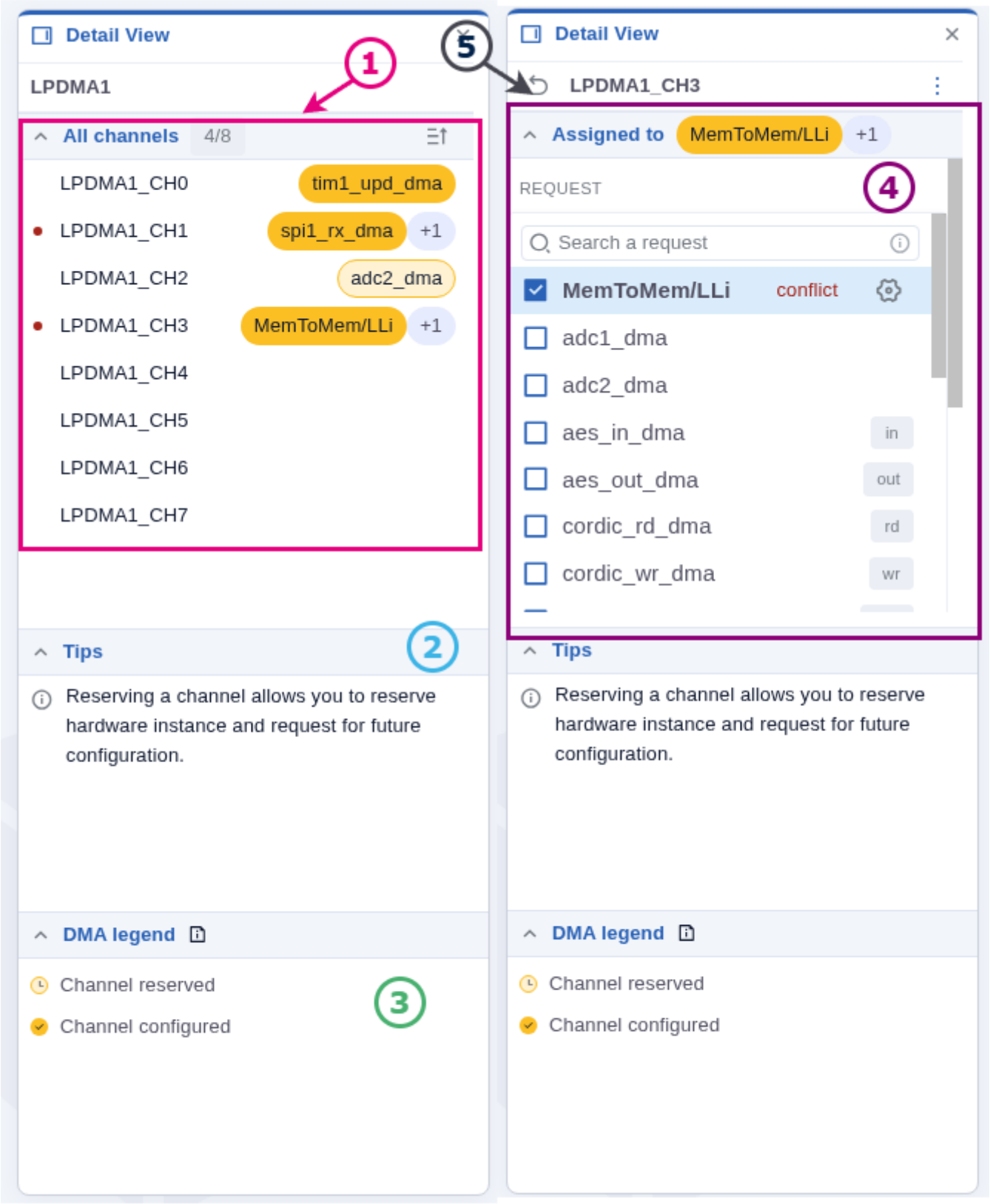


Fig. 27: DMA system view - detail view

> **Note**
>
> Actions menu is also available in the detail view to move, swap, reset, or go to configuration.

4.7.1 Additional articles

- How to use the DMA in normal mode with STM32CubeMX2

4.8 NVIC

The **NVIC** view, illustrated in *NVIC view*, allows users to manage interrupts. It is accessible from the *Peripherals* view.

- **Search and filter (1):** Provides search and filter options to locate specific interrupts by acronym or description.
- **Display options (2):** Filters the view to display all interrupts, exceptions, or peripherals.
- **Reset all interrupts (3):** Resets the NVIC configuration to its default state, discarding any changes.
- **Priority group selection (4):** A dropdown menu for selecting the priority group.
- **Configuration access (5):** Opens the peripheral configuration for the selected interrupt in a new tab.
- **Update preemption priority and subpriority (6):** Modifies the preemption priority and subpriority values directly in the table.
- **Columns table settings (7):** Customizes the columns displayed in the table view, enabling, or disabling specific columns as needed.

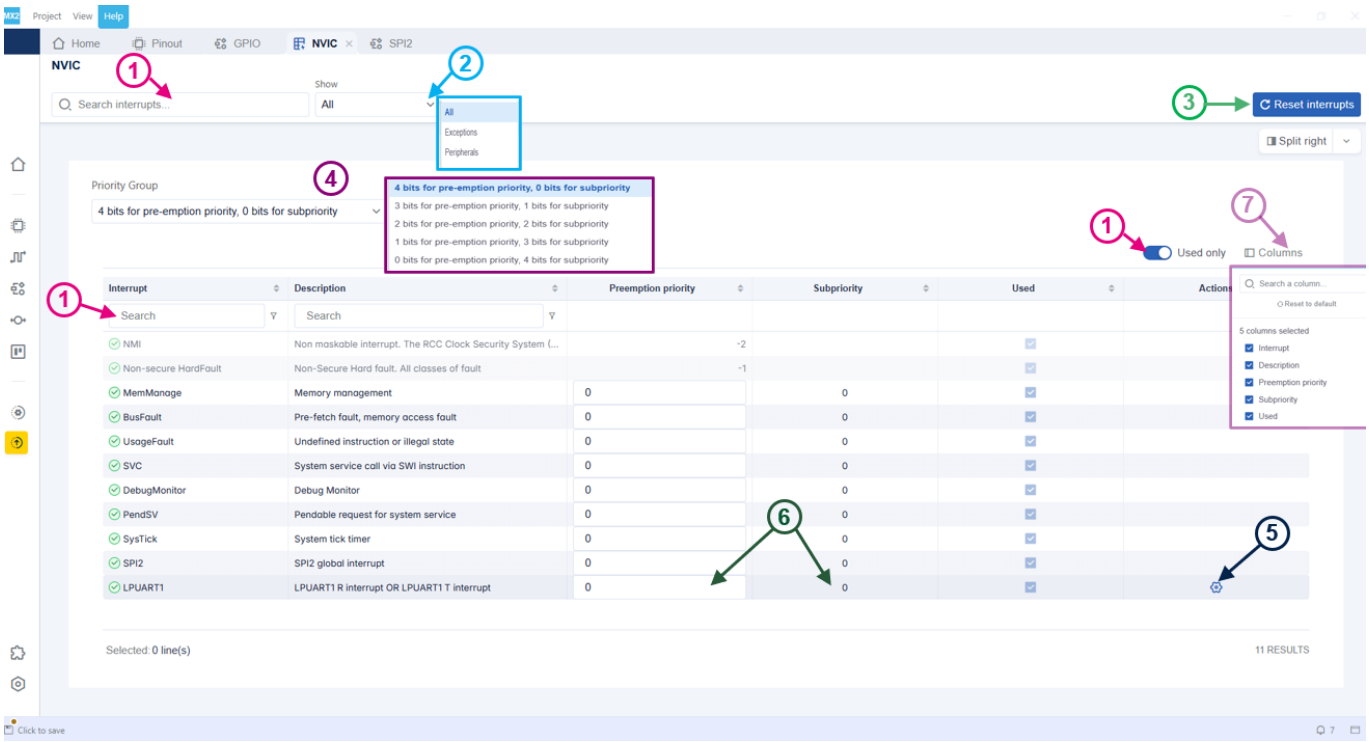


Fig. 28: NVIC view

4.9 EXTI

The **EXTI view** displays a read-only list of available **EXTI lines.** The **EXTI lines** are enabled through the **GPIO configuration.**

The **EXTI view** is accessible from the peripheral view. In the **EXTI view,** users can perform the following actions:

- **Search (1):** Find specific **EXTI lines** by name or description.
- **Filter (2):** Display only the used EXTI lines.
- **Display EXTI lines (3):** Show the list of all available **EXTI lines.**
- **Sort (4):** Sort the list of **EXTI lines.**
- **Columns table settings (5):** Customize the columns displayed in the table view by enabling or disabling specific columns as needed.

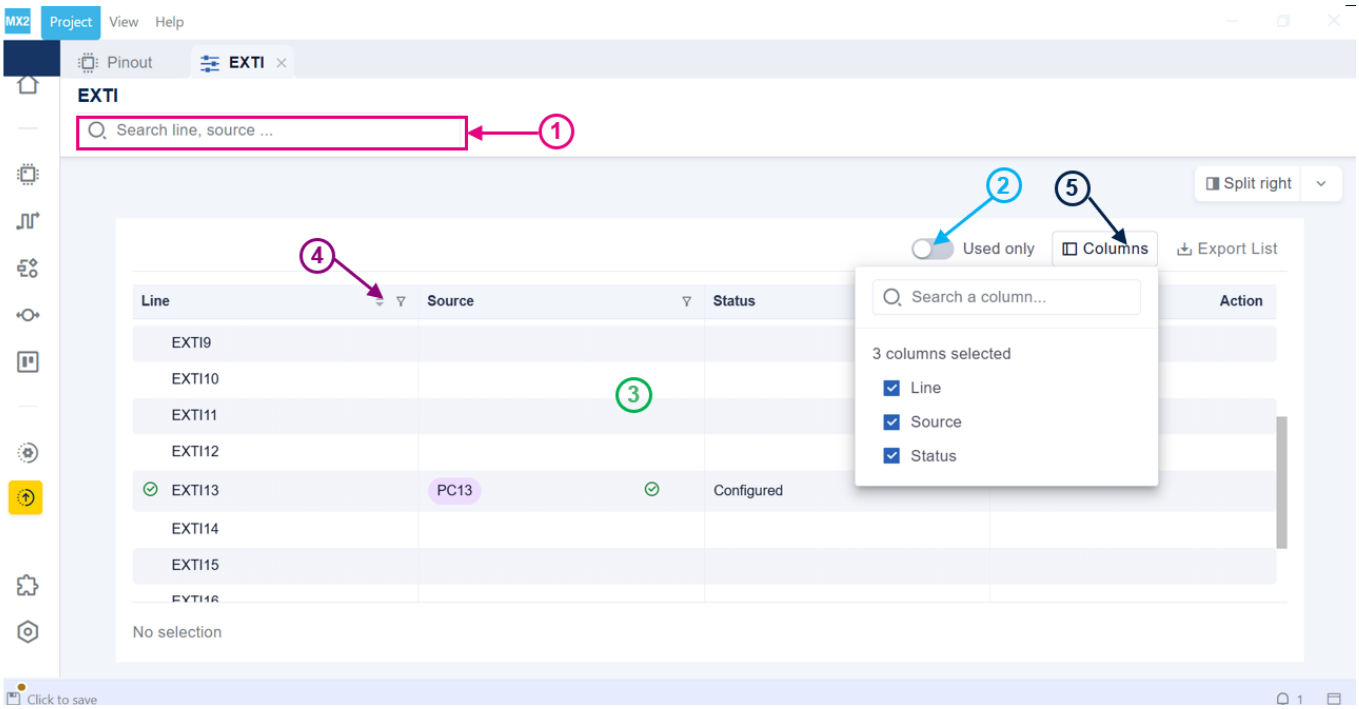


Fig. 29: EXTI view

4.10 Middleware and utilities

In the **Middleware and utilities** view, the user can perform the following actions:

- **Filter (1a):** Search for specific middleware or utility by name, category, or status (active/inactive).
- **Middleware section (1b):** View the list of predefined or manually added middlewares.
- **Utilities section (1c):** View the list of predefined or manually added utilities.
- **Add middleware or utility (2):** Select one or more middlewares or utilities from the available list and add them to the project.
- **Sort (3):** Sort the list of middlewares or utilities in ascending or descending alphabetical order.
- **Activate/Deactivate middleware or utility (4):** Create or delete a configuration for the selected middleware or utility. When a middleware or utility is activated, its configuration panel appears, and a check mark icon is displayed next to the middleware or utility name in the left panel.

- **Remove from list (5):** Remove an added middleware or utility from the list. A middleware or utility cannot be removed if it is activated. This feature is only available for added middlewares or utilities. The predefined list cannot be removed and is prefixed with a padlock icon.

> **Note**
>
> The **Remove from list** option is grayed out for the default list, but it becomes available for added middlewares or utilities.

- **Reset Configuration (6):** Reset configuration settings to their default values.
- **Toggle the detail view (7):** Display or hide the detail view panel.
- **Toggle the code preview (8):** Display or hide the code preview panel.
- **Toggle the quick menu (9):** Display or hide the quick menu panel.
- **Access STM32Cube documentation (10):** Open the STM32Cube documentation.
- **Expand All / Collapse all (11):** Expand/Collapse all sections in the middleware and utilities configuration cards.

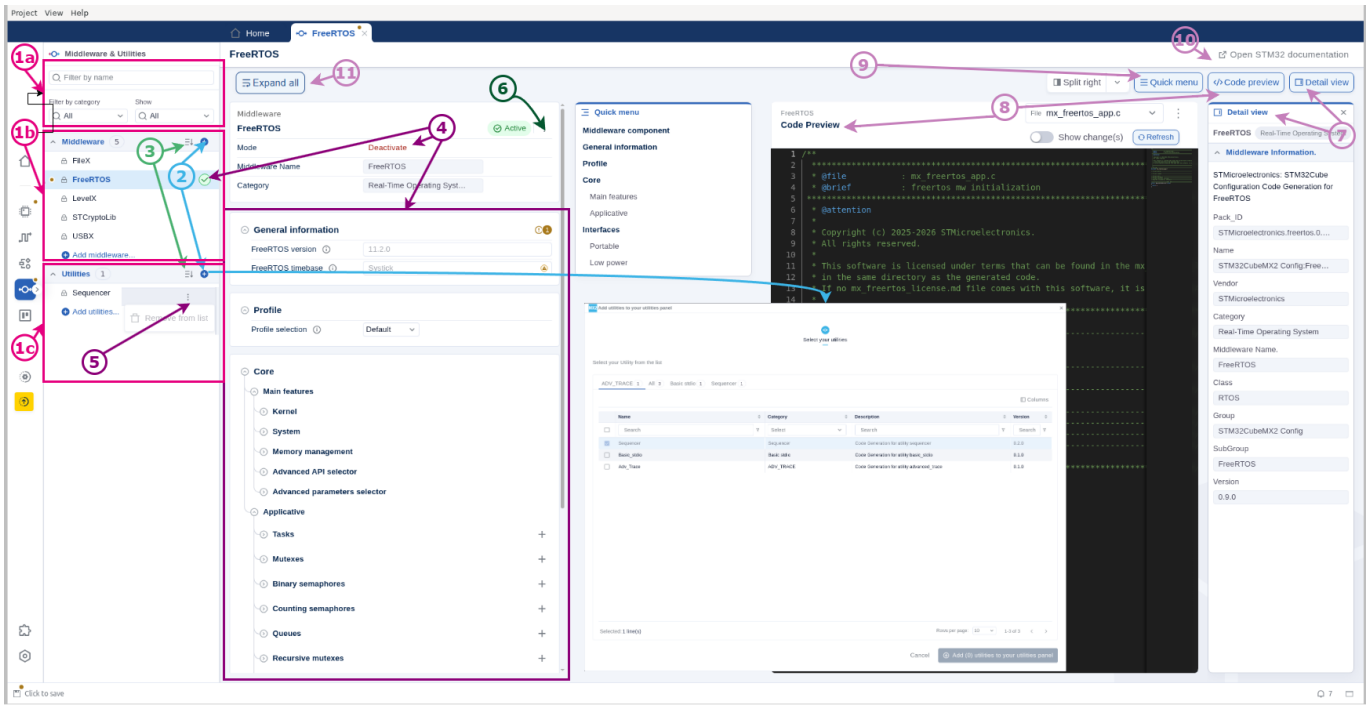
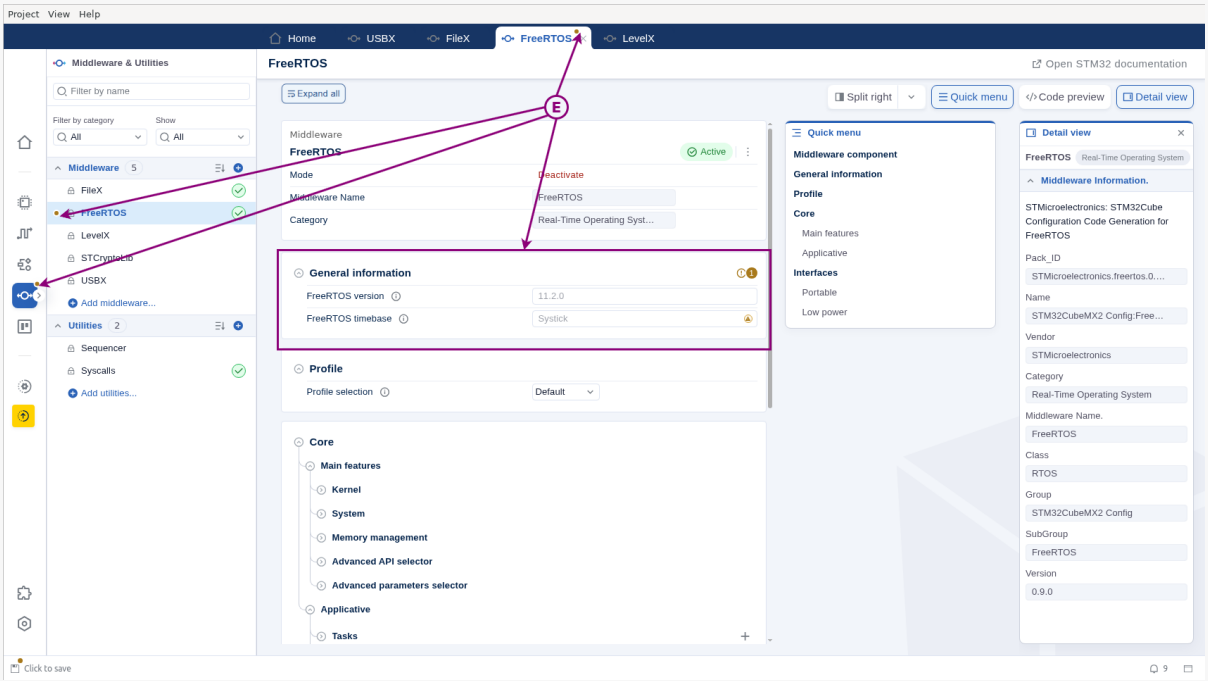


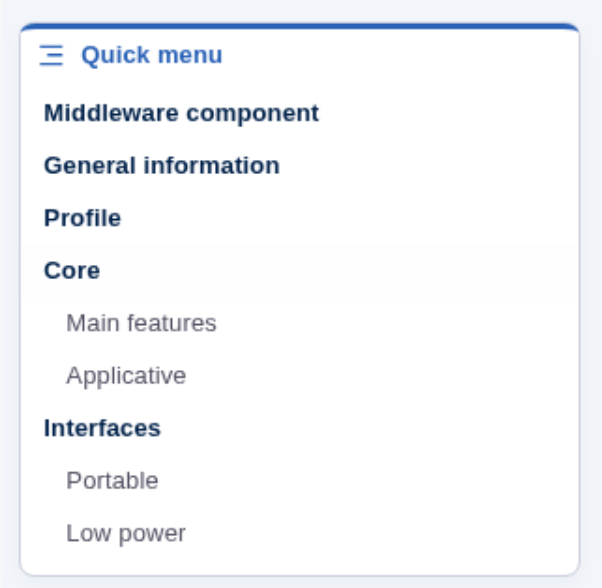
Fig. 30: Middlewares and utilities panel

> **Note**
>
> - The **Errors & Warnings icons (E)** allow the user to highlight errors and warnings related to the selected middleware or utility.

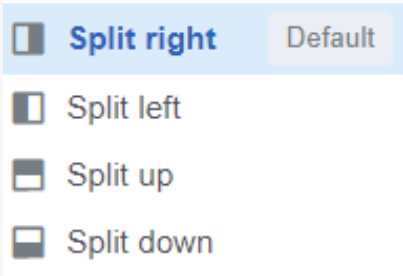


- The **Banners** provide information to help users quickly identify the status of the selected middleware or utility.

- The **Quick Menu** provides a convenient way to navigate middleware-related or utility-related options.



- The **Split view** allows the user to divide the current workspace or editor window into multiple panes for multitasking and comparison.



4.10.1 Additional articles

- How to use Middleware with STM32CubeMX2
- How to use EEPROM Emulation with STM32CubeMX2
- How to use FreeRTOS with STM32CubeMX2

4.11 HW Parts

In the **HW Parts** view, users can perform the following actions:

- **Filter (1a):** Search for a specific HW part by name, category, or status (active or inactive).
- **Section Mounted Parts (1b):** Display one section per board in the selected project. Each section shows the mounted parts on the specific board. If only the MCU is present, only the standalone part section is displayed.
- **Section Standalone Parts (1c):** Display one section showing standalone parts that users can add to a custom board.
- **Add part (2):** Select one or more standalone parts and add them to a custom board.
- **Sort (3):** Sort the list of parts in ascending or descending alphabetical order.
- **Activate/Deactivate part (4):** Activate or deactivate the selected HW part. Create or delete a configuration for the selected HW part. When a part is activated, its configuration panel is displayed, and a check mark icon appears next to the part name in the left panel.
- **Remove from list (5):** Remove a standalone part from the list. This feature is only available for standalone parts. A part cannot be removed if it is activated. Mounted parts cannot be removed.
- **Reset configuration (6):** Reset configuration settings to default values.
- **Toggle the detail view (7):** Display or hide the detail view panel.
- **Toggle the code preview (8):** Display or hide the code preview panel.
- **Toggle the quick menu (9):** Display or hide the quick menu panel.
- **Access STM32Cube documentation (10):** Open the STM32Cube documentation.
- **Expand All / Collapse all (11):** Expand/Collapse all sections in the HW parts configuration cards.

Note

- The **Errors and Warnings icons (E)** allow users to highlight errors and warnings related to the selected HW part.

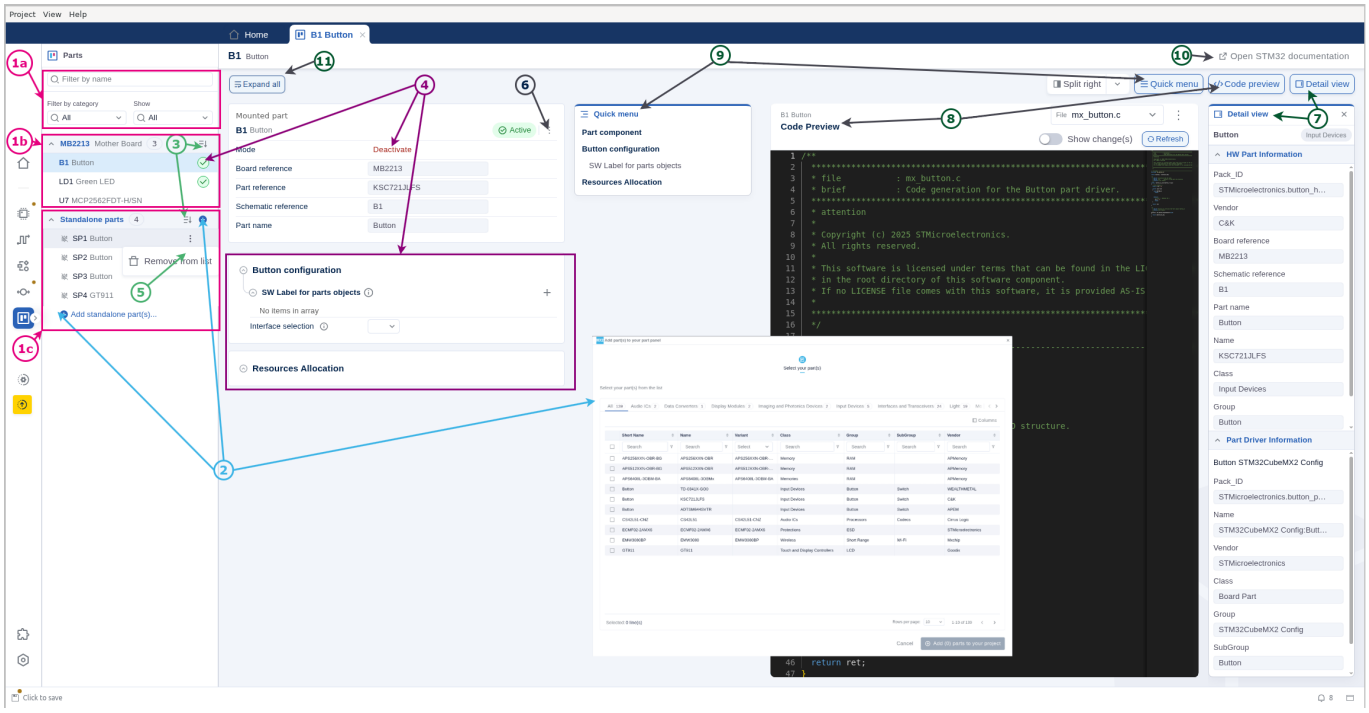
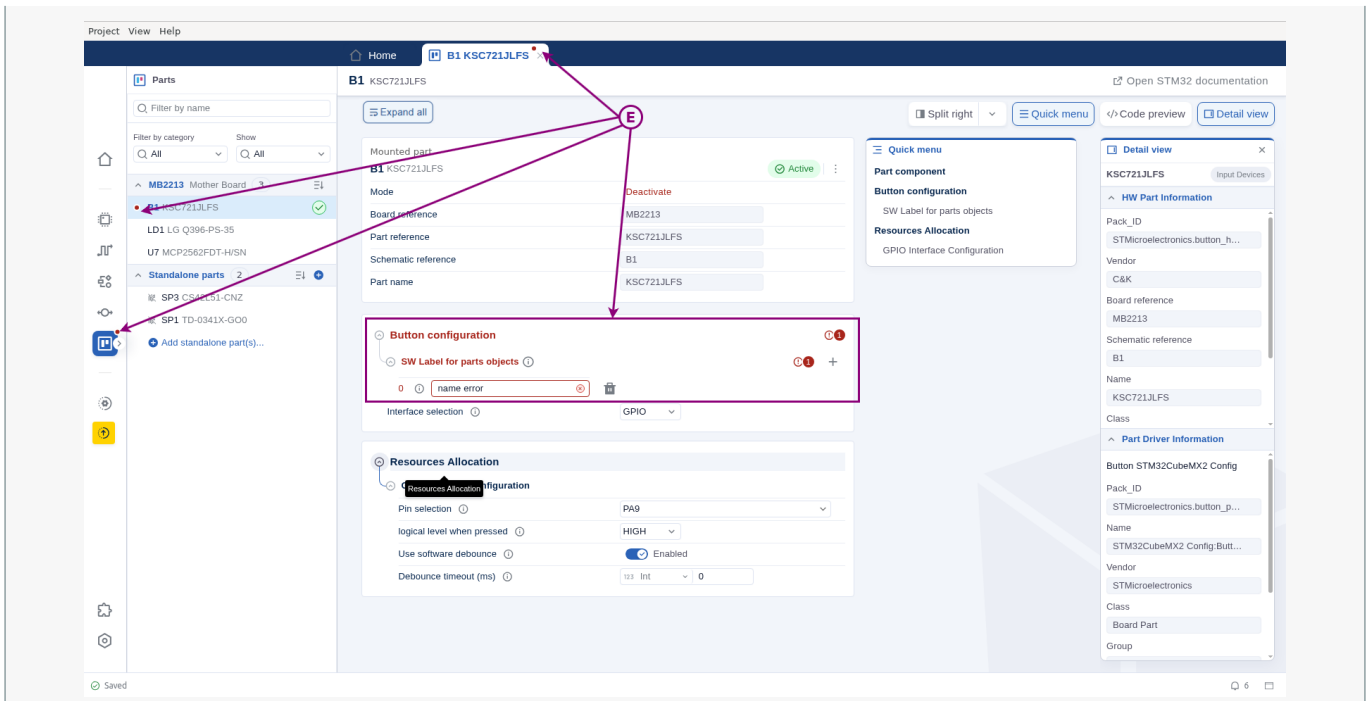
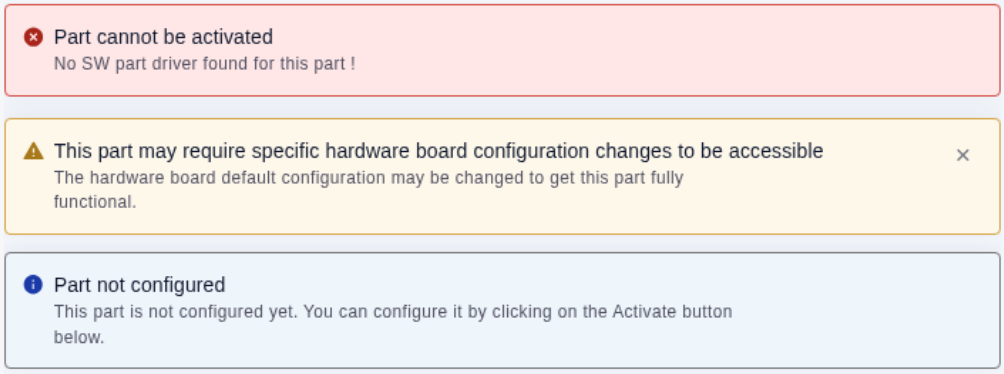


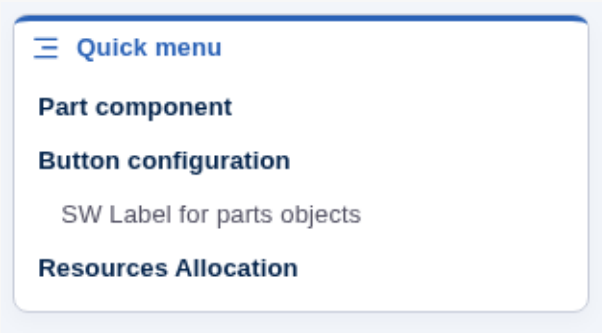
Fig. 31: HW Parts panel



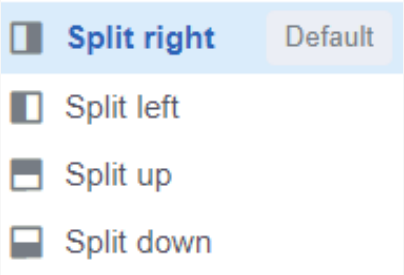
- The **Banners** provide information to help users quickly identify the status of the selected HW part.



- The **Quick Menu** provides a convenient way to navigate part-related options.



- The **Split View** allows users to divide the current workspace or editor window into multiple panes for multitasking and comparison.



4.12 IDE project generation

From the project settings, users select the development toolchain and, if applicable, the target integrated development environment (IDE). STM32CubeMX then generates the corresponding toolchain settings and integrated development environment project files for import as a new project.

4.12.1 IDE project settings

For **IDE project generation**, a minimal configuration is required. Additional advanced settings available for further customization.

General setup

- **Select format (1)** in *'IDE project settings' figure*
  - Choose the desired format.
  - Available formats include:

    * IAR Embedded Workbench® (default)
    * CMake
    * Open-CMSIS

> **Note**
>
> For STM32CubeIDE for VS Code, CMake is the required project format.
>
> Open-CMSIS projects are compatible with Keil µVision and Keil Studio.

- **Select toolchain: (2):** in *'IDE project settings' figure*
  - For the IAR Embedded Workbench only IAR is available as a toolchain.
  - For CMake, select GCC.
  - For Open-CMSIS, select IAR, GCC or AC6.
- **Select destination (3):** in *'IDE project settings' figure*
  - Browse and select the folder for the **IDE** project.
  - The default path is the root folder of the created **STM32CubeMX2** project.
- **Advanced setup (4)** in *'IDE project settings' figure*
  - **User modifiable component files (5):** in *'IDE project settings' figure*

    Optionally, change the path for user modifiable components.
  - **Generated code directories (6):** in *'IDE project settings' figure*

    Optionally, change the default name and directory path for storing generated code.
  - **Select external sources (7):** in *'IDE project settings' figure*

    Select the packs source. There are two options to manage packs for the exported project: - **Include fresh packs:**

    * Packs are always installed from the **.pack** file (downloaded if needed) inside the created project.
    * Relative pack paths cannot be outside of the created project (they are made local).
    * User code is always copied to the created project.
    * Generated code is always generated in the created project.
    * The created project is easily movable and is self-contained.
    * **Include packs from local:**
      · Packs are copied from **$CMSIS_PACK_ROOT** to the created project. For more details on **$CMSIS_PACK_ROOT,** see *Packs installation*.
      · Relative pack paths cannot be outside of the created project (they are made local).
      · User code is always copied to the created project.
      · Generated code is always generated in the created project.
      · Created project is easily movable and is self-contained.
  - **Remove board packs (8):** in *'IDE project settings' figure*

    Exclude board support packages from the project.
  - **Select packs installation directories (9):** in *'IDE project settings' figure*

    Optionally, browse and select a path for the packs installation folder. The default folder is named **misc** and is placed under the **IDE** project folder.

- **Use custom directories (10):** in *'IDE project settings' figure*

  Organize installed packs by category into customized folders and modify folder paths and names.

- **Automatic Conflict Handling Rules (11):** in *'IDE project settings' figure*
  - By default all files update on change.
  - By selecting **Custom...**, multiple ways are available to handle different types of conflict resolution.
    - **Types of conflict resolution:**

      Show conflicting files: Launches the conflict resolution tool to handle each case.

      Don't generate: Keeps any existing files.

      Backup: Creates backups of existing files and overwrites.

      Overwrite: Overwrites existing files.

    - **Types of files:**

      User files: Files owned by the user, for example, `main.c`, `main.h`, `my_lib.cpp` etc.

      Project files: Project files, for example, `CMakeLists.txt`, `.csolution.yml`, `cproject.yml`, `.eww`, `.ewp`)

      Pack: Files within embedded software packs

      User-modifiable files: User-modifiable files from packs.

      Generated: Generated files from packs that have a generator run during project export, for example, a tailored file based on the configuration in MX.

- **Generate project (12):** in *'IDE project settings' figure* Click the **Generate** button to start the project creation process with all configured settings.

Note

- When exporting, ensure to **keep the default values for generator directories** to avoid issues.
- To prevent export problems, **make sure that the generation directories are different for each configurable part.**

Note

- With the **IDE** project settings configured, users can skip the **Project settings** tab and click the **Generate** button to start project generation for the selected format.
- During the generation process, a popup appears to inform the user about the ongoing project generation and to provide progress updates.

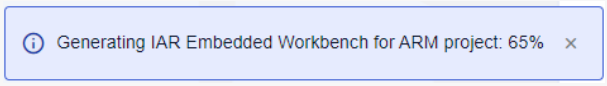


Fig. 33: Project generation popup

- After the generation process completes, a popup appears to notify the user that the **IDE** project is finalized. The user can view the generated files on disk.

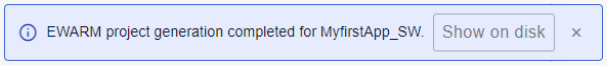


Fig. 34: Project successfully finalized

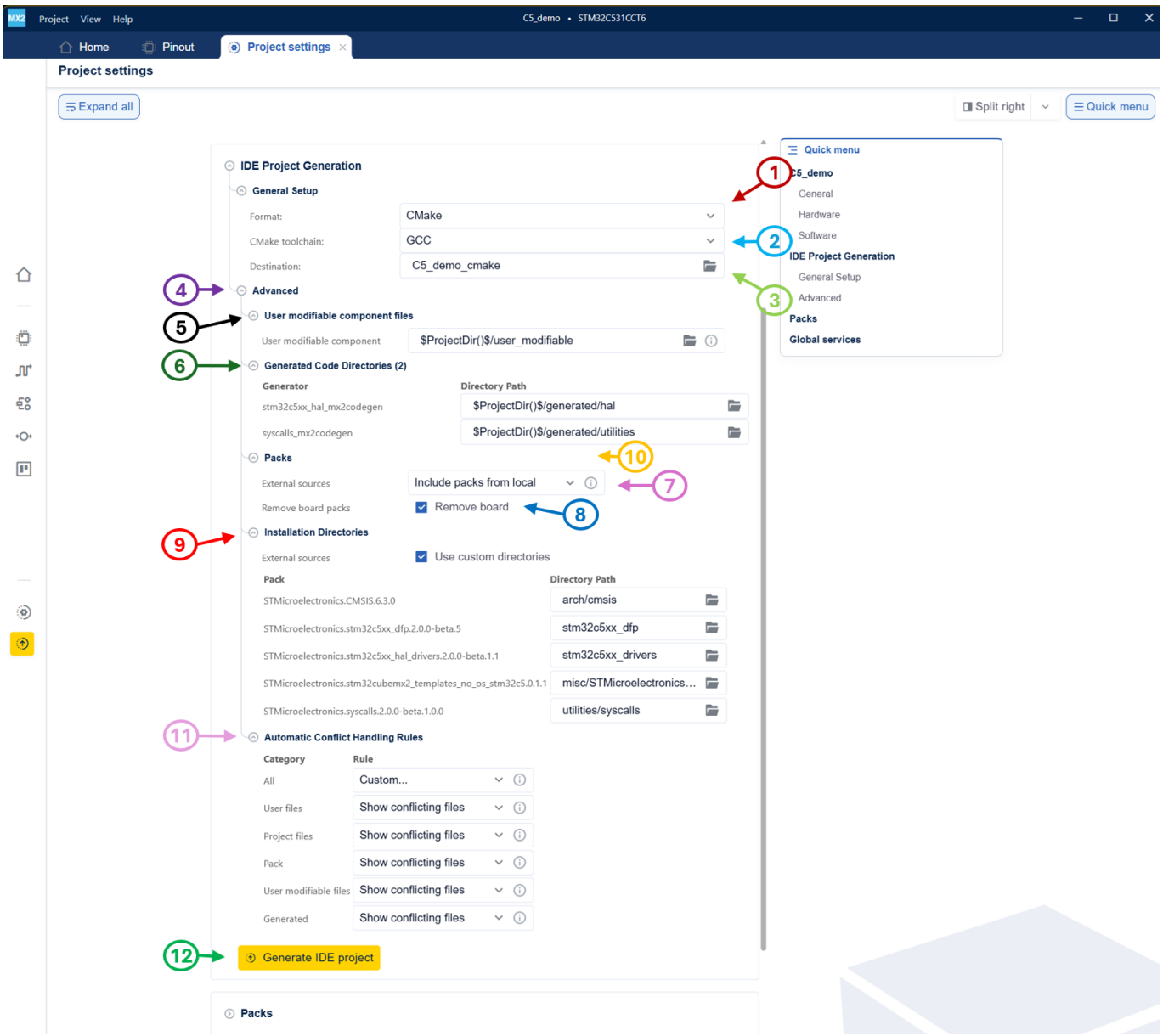


Fig. 32: IDE project settings

4.12.2 IDE project regeneration

During the code regeneration process in **STM32CubeMX2**, conflicts may arise when the user previously modified files that the tool attempts to update. These conflicts occur because the regeneration process overwrites or modifies existing project files to incorporate the latest changes or configurations made in the tool.

To handle such situations, **STM32CubeMX2** provides a conflict resolution mechanism that allows users to decide how each conflicting file is treated.

During code regeneration, a popup appears to inform the user of the following options:

- **Go to project settings:** Open the project settings menu to configure automatic conflict handling rules.
- **Show differences:** Enables the user to view the differences between the existing files and the newly generated ones after the regeneration process completes. For each file, the user can choose an action:
  - **Overwrite without backup:** The existing file is directly overwritten with the new file containing the latest changes.
  - **Overwrite with backup:** The existing file is renamed to **FILE_bckp**, and a new file (**FILE**) with the latest changes is created.
  - **Don't generate:** No action is taken, and the existing file remains unchanged.
- **Overwrite all:** Overwrites all files with the latest changes without creating backups.

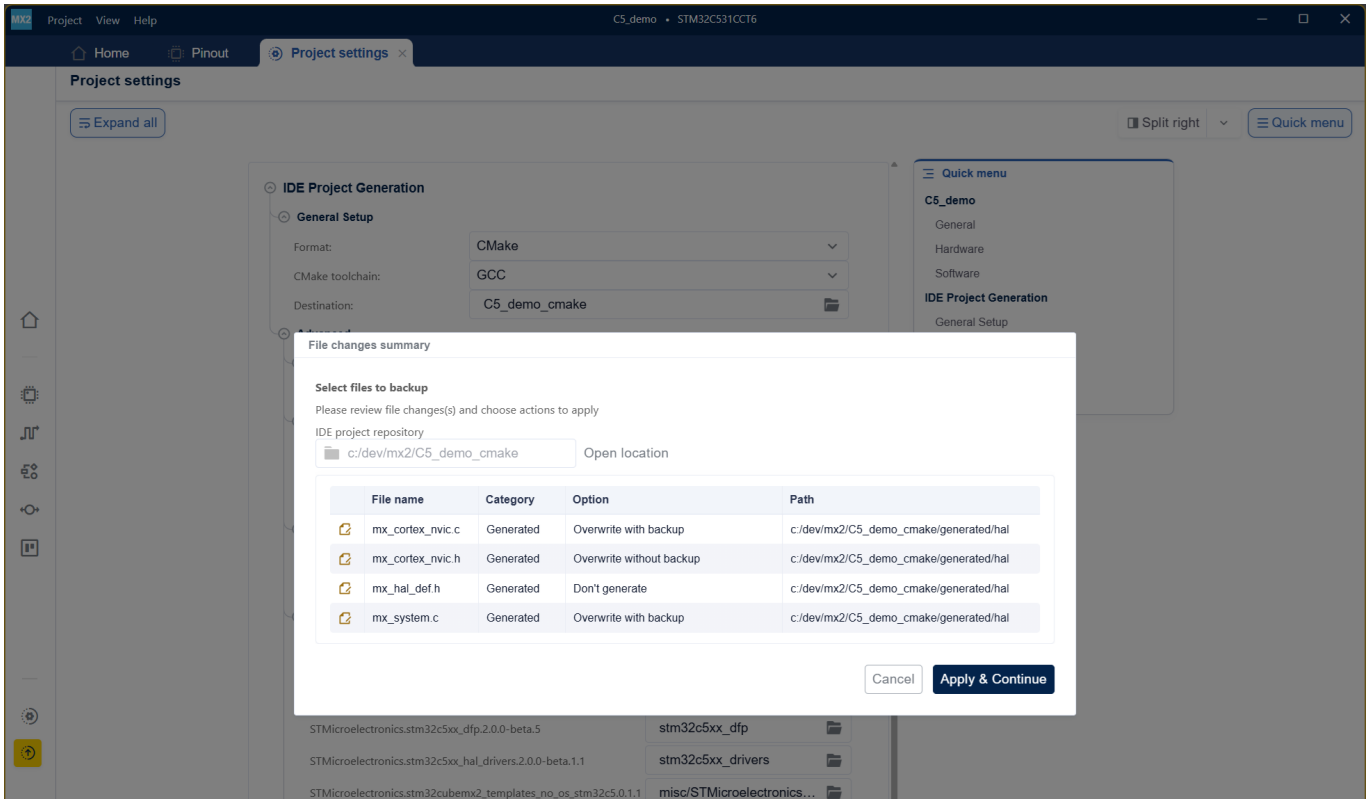


Fig. 35: IDE project regeneration

For more information about customizing this conflict resolution, refer to step 11 of the *IDE project settings* guide.

4.12.3 IDE project structure

The **STM32CubeMX2** project is organized to support multiple **IDE** environments and build systems.

Below is an explanation of the project components and folder structures as illustrated in *the figure below*.

- **Project configuration file: MyFirstApp.ioc2** is the main configuration file for the **STM32CubeMX2** project.

- **IDE project folders:** The project generates multiple folders corresponding to different IDE and builds.

  Each folder contains files and subfolders tailored to the specific environment:

  - **CMake format — Default folder: MyFirstApp_cmake**
    - **.settings/:** CubeMX2 metadata used to track and regenerate project files.
    - **arch/:** Architecture-specific files.
    - **cmake/:** User file inclusion and CMake support files.
    - **generated/:** Generated source and header files.
    - **stm32mcuxx_dfp/:** Device support pack for the **STM32 MCU.**
    - **stm32mcuxx_drivers/:** Device drivers for the **STM32 MCU.**
    - **user_modifiable/:** Project files intended for user edits (application code, device configuration, and build variant).
    - **utilities/:** Utility source files.
    - **CMakeLists.txt**, **CMakePresets.json:** Build configuration files for CMake.
    - **main.c**, **main.h:** Application source and header files.
  - **EWARM format — Default folder: MyFirstApp_ewarm**
    - **.settings/:** CubeMX2 metadata used to track and regenerate project files.
    - **arch/:** Architecture-specific files.
    - **project/:** Software project files including generated code.
    - **stm32mcuxx_dfp/:** Device support pack for the **STM32 MCU.**
    - **stm32mcuxx_drivers/:** Device drivers for the **STM32 MCU.**
    - **utilities/:** Utility source files.
    - **MyFirstApp_SW.eww:** EWARM project workspace file to open the project in the IDE.
  - **Open-CMSIS format — Default folder: MyFirstApp_open-cmsis**
    - **.settings/:** CubeMX2 metadata used to track and regenerate project files.
    - **arch/:** Architecture-specific files.
    - **project/:** Software project files including generated code.
    - **stm32mcuxx_dfp/:** Device support pack for the **STM32 MCU.**
    - **stm32mcuxx_drivers/:** Device drivers for the **STM32 MCU.**
    - **utilities/:** Utility source files.
    - **MyFirstApp.csolution.yml:** CMSIS solution file describing the project structure.

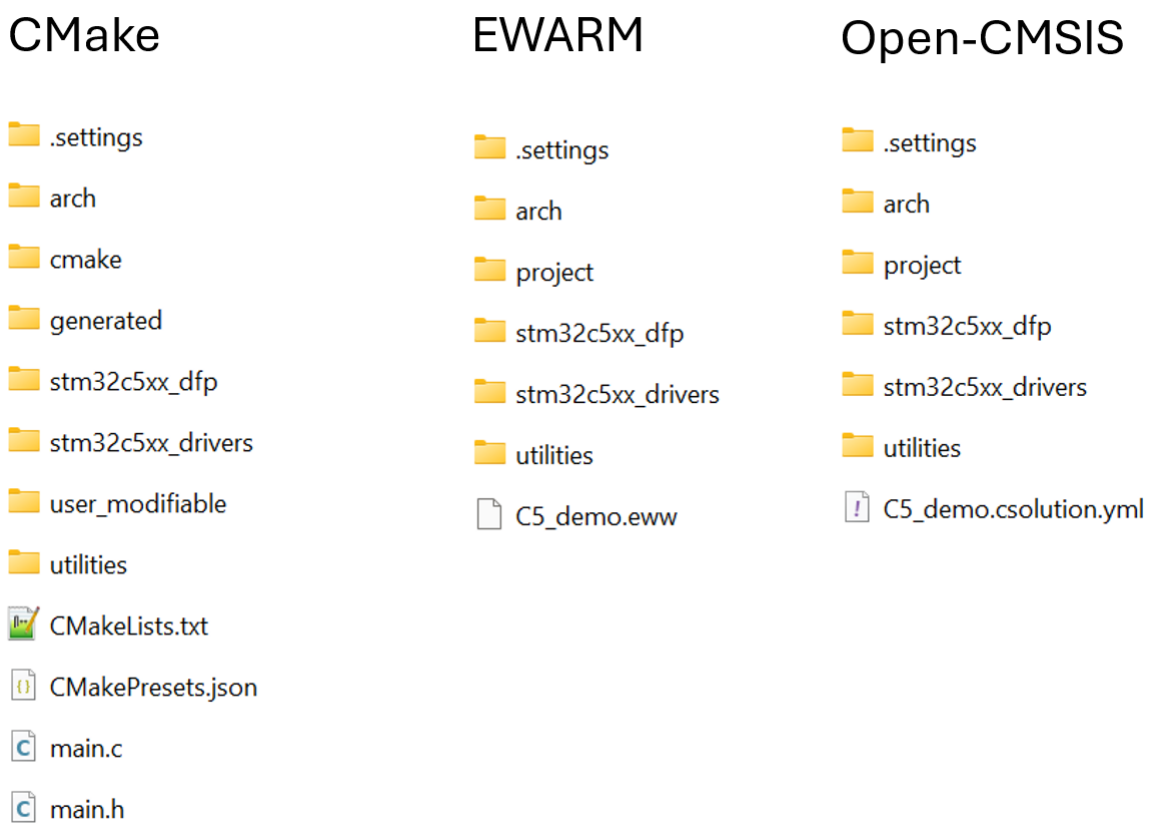


Fig. 36: IDE project folder structure

4.13 Pack manager

This panel enables users to manage software packs, which are essential for developing applications with different **STM32** microcontrollers.

- **Pack manager access (1):** Access the pack manager from the left bar to view and manage available software packs.
- **Search and filter (2):** Use the top section of the panel to search for specific packs and filter them by various criteria.
- **Pack management (3.1):** Use the **Action** column to install, update, or remove packs. The Status column indicates if a pack is **Not installed, Up to date,**, or **Update available.**
- **Pack management (3.2):** The **Compatible** column indicates if a pack is compatible with the current version of STM32CubeMX2.
- **Detail view toggle (4):** Use the details view to display additional information about each pack, such as description, release version, and release date.
- **Local pack installation (5):** Use this button to browse and select a software pack from local storage for installation.
- **Refresh button (6):** Use this button to update the list of available software packs.
- **Columns (7):** Use this button to update the list of visible columns.
- **Access to Licenses (8):** Use this button to view all licenses from all packs before downloading them.
- **Split (9):** Use this button to split the tab into two separate views for better visibility.
- **Pagination controls (10):** Use pagination controls to navigate through different pages.

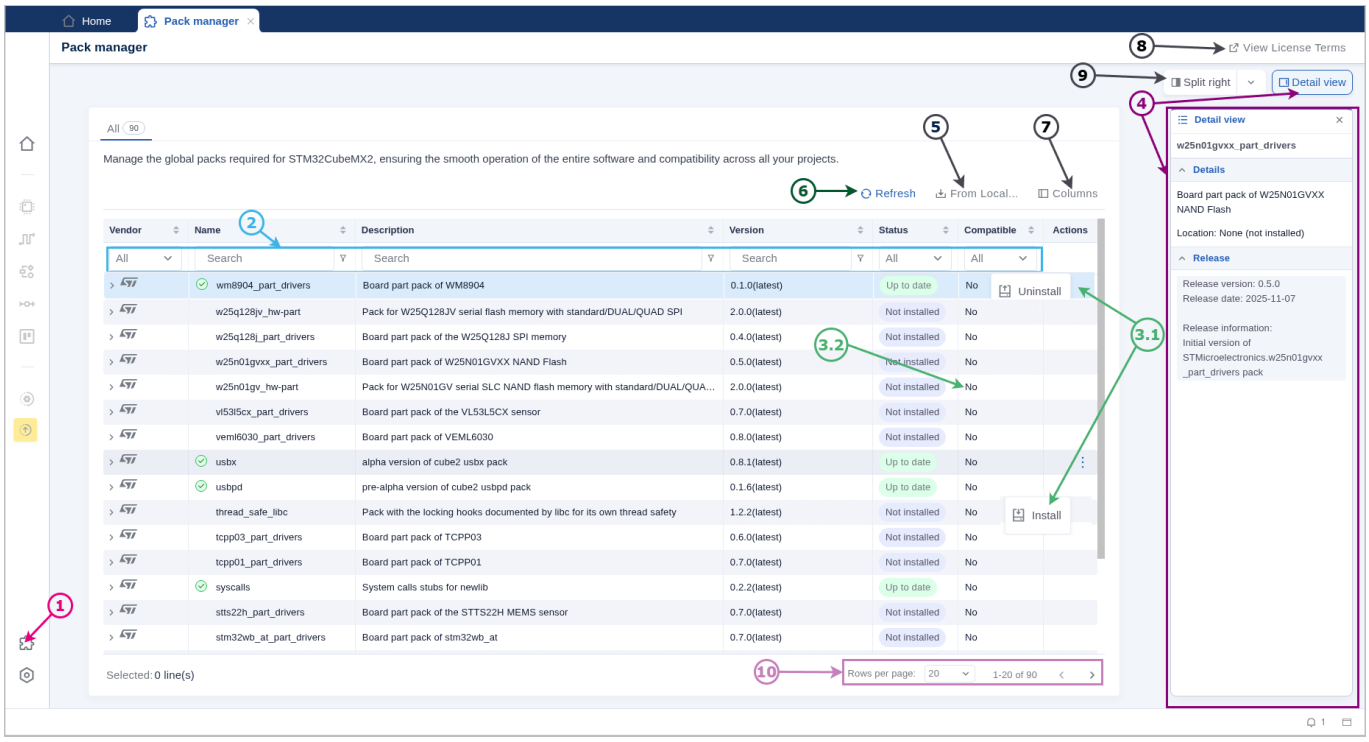


Fig. 37: Pack manager panel

4.14 Code preview

4.14.1 Code preview

The **Code preview** panel in **STM32CubeMX2** provides the following features and capabilities (see *Code preview*):

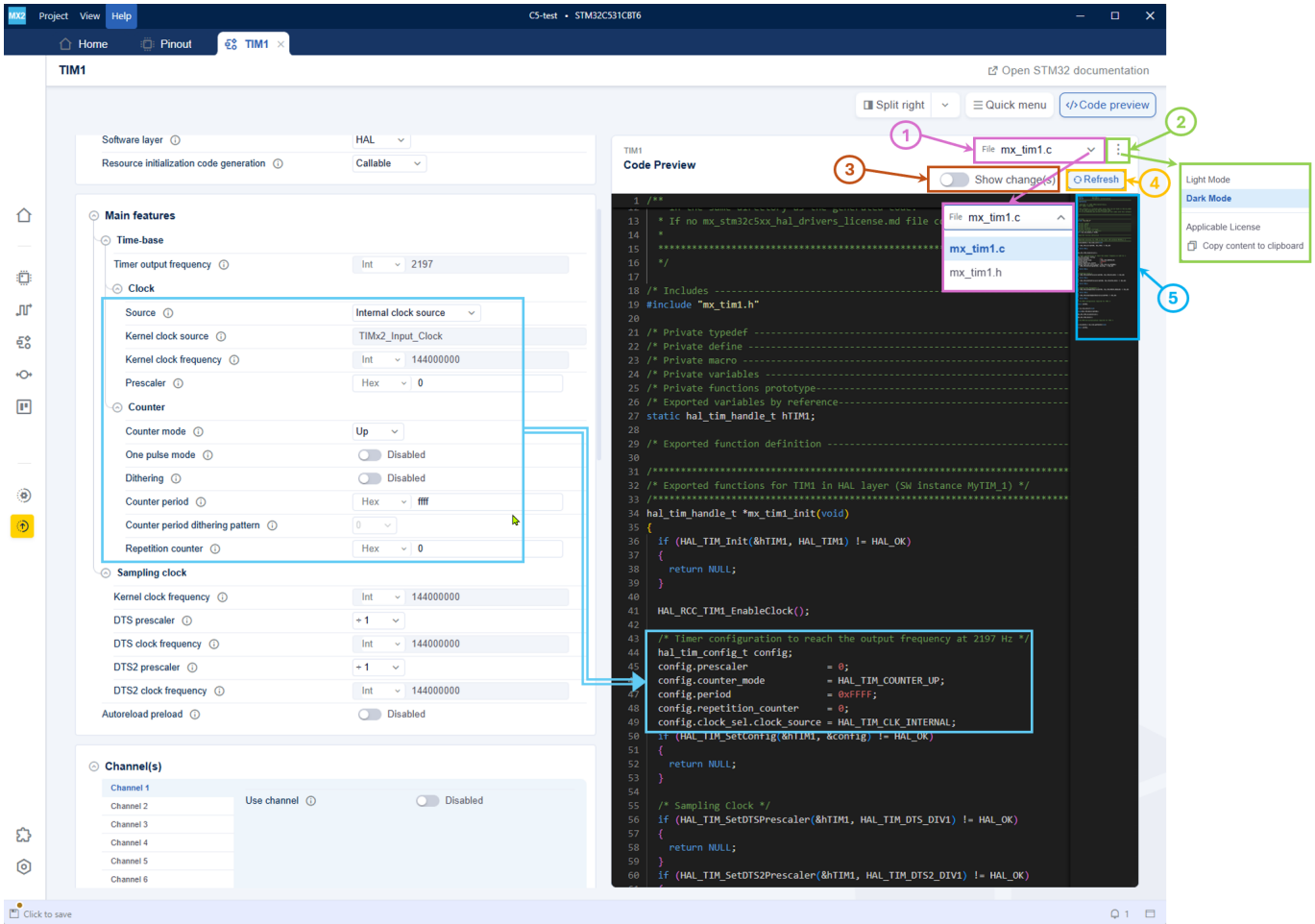


Fig. 38: Code preview

- **File selection (1):** The file selection dropdown allows the user to toggle between *.h* and *.c* files.
- **Settings menu (2):** The settings menu provides options such as toggling between **Light Mode** and **Dark Mode,** applicable license, and copy content to clipboard.
- **Show change(s) (3):** When enabled, it displays the code differences with color highlighting (see *Code preview Diff*):.
- **Refresh view (4):** The code preview is refreshed automatically in real time reflecting the latest changes made in the configuration settings. The **Refresh** button remains available as a reassurance in case of any doubt.
- **Minimap (5):** The minimap allows users to navigate through the preview file.

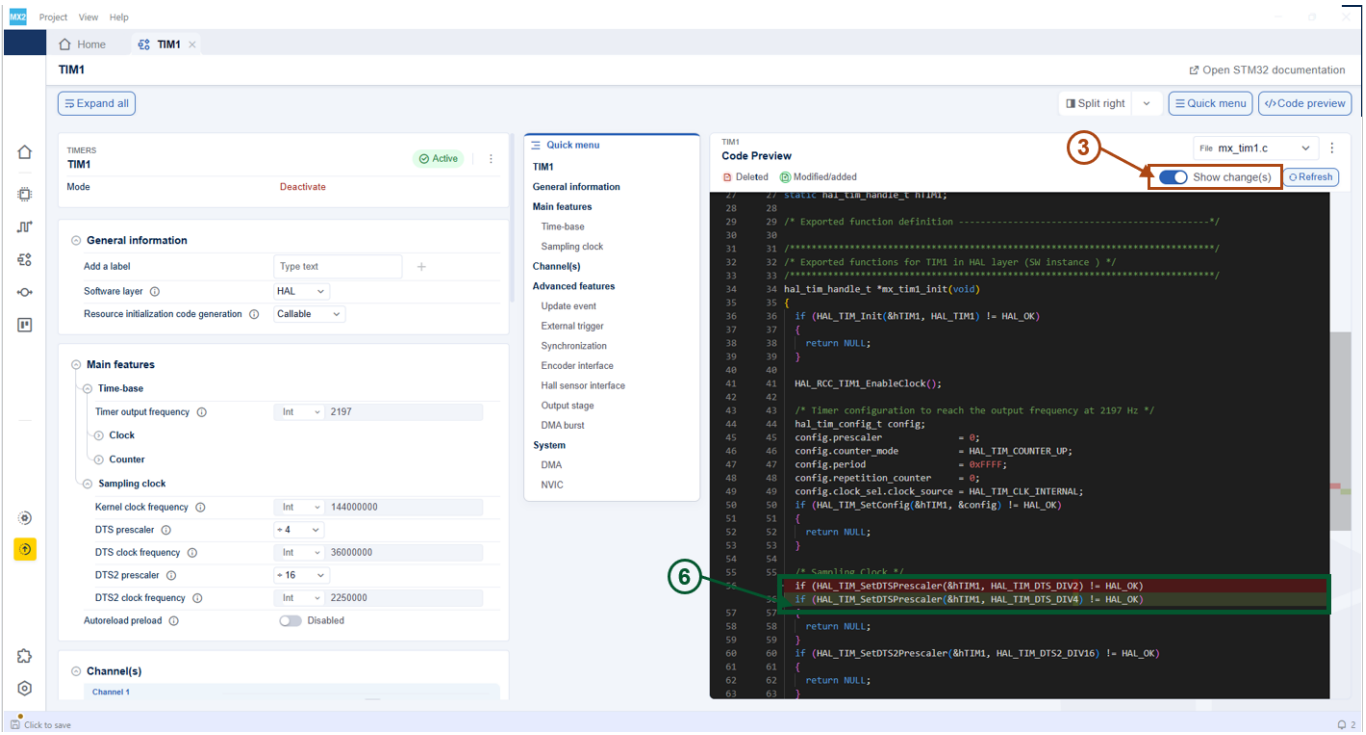


Fig. 39: Show Code preview Diff

5 Command lines interface

5.1 CLI: clock

CLI for SW clock validation

5.1.1 find-clock

Returns an array of filtered configurations based on ids (ids containing the provided string), role, clocked_Instance, type.

Options:

Name
–clocked-instance <clocked_Instance>
–exact
–host <host>
–id <id>
–port <port>
-p, –project-path <projectPath>
–role <role>
–type <type>

5.1.2 get-clock

Returns the configuration of a specific clock element in a formatted object.

Options:

Name
–format, <string>
–host <host>
–id, <string>
–port <port>
-p, –project-path <projectPath>

5.1.3 get-clock-constraints

Returns a list of clock elements that have applied constraints

Options:

Name	
–host <host>	MX
–port <port>	MX
-p, –project-path <projectPath>	Path to

5.1.4 get-clock-items

Returns a list of clock items in {key, value} format.

Options:

Name	
–host <host>	
–ids, <string>	exac
–port <port>	
-p, –project-path <projectPath>	

5.1.5 get-clock-status

Returns the global status of the clock tree, checking for out-of-range configurations.

Options:

Name	
–host <host>	MX
–port <port>	MX
-p, –project-path <projectPath>	Path to

5.1.6 get-clocks

Returns the configuration of all clock elements in a formatted object.

Options:

Name	
–format, <string>	
–host <host>	
–port <port>	
-p, –project-path <projectPath>	

5.1.7 list-possible-values

Lists possible values for a given clock element. Possible values can be a frequency range or a set of multiple values or a fixed value depending on the clock element.

Options:

Name	
–host <host>	
–id, <string>	exac
–port <port>	
-p, –project-path <projectPath>	

5.1.8 reset

Resets clock to default state.

Options:

Name	
–host <host>	
–port <port>	
-p, –project-path <projectPath>	
–skip-confirmation	U

5.1.9 set-clock-constraints

Applies a constraint to a clock element. Supported constraints are: minimize, maximize.

Options:

Name
–allow-unlock
–constraints <format>
–flexible-path
–host <host>
–id, <string>
–port <port>
-p, –project-path <projectPath>
–role, <string>

5.1.10 set-max-value-by-hw-instance

Sets a hardware-instance clock element to its defined maximum value.

Options:

Name
–allow-unlock
–flexible-path
–host <host>
–hw-instance, <string>
–port <port>
-p, –project-path <projectPath>
–type, <string>

5.1.11 set-value

Sets the value of a specific configuration item.

Options:

Name
–allow-unlock
–flexible-path
–fraction-denominator, <string>
–fraction-value, <string>
–host <host>
–id, <string>
–port <port>
-p, –project-path <projectPath>
–value, <string>

5.1.12 set-value-by-hw-instance

Sets the value of a specific configuration item.

Options:

Name	
–allow-unlock	
–flexible-path	
–host <host>	
–hw-instance, <string>	
–port <port>	
-p, –project-path <projectPath>	
–type, <string>	
–value, <string>	

5.1.13 set-values

Sets new values to a set of clock items

Options:

Name	
–allow-unlock	
–flexible-path	
–fraction-denominators, <string>	
–fraction-values, <string>	
–host <host>	
–ids, <string>	
–port <port>	
-p, –project-path <projectPath>	
–values, <string>	

5.1.14 solver-on-demand

Calls the clock solver to fix erroneous values in the clock tree.

Options:

Name	
–host <host>	MX
–port <port>	MX
-p, –project-path <projectPath>	Path to

5.2 CLI: composer

CLI for sw composer feature

5.2.1 add-component

Add one component to an existing software component project

Options:

Name
-c,–component-id <componentId>
–host <host>
–port <port>
-p, –project-path <projectPath>
-r,–resolve-dependency-strategy <resolveDependencyStrategy>
–skip-configuration-instance <skipConfigurationInstance>
-s, –software-project <softwareProjectName>

5.2.2 add-files

Add any library or dependency files to a SW project

Options:

Name
-b, –build-target <string>
-c,–category <string>
-f,–files '[<string>]'
-g,–group <string>
–host <host>
–port <port>
-p, –project-path <projectPath>
-s, –software-project <softwareProjectName>

5.2.3 apply-quick-fix

Apply quick fix for a single or multiple actions

Options:

Name
-a,–actions <string>
–host <host>
–port <port>
-p, –project-path <projectPath>
-s, –software-project <softwareProjectName>

5.2.4 delete-configuration-instances

Delete a list of configuration instances of a components in a software project

Options:

Name
-c,–component-id <string>
–host <host>
-i,–instance-id <string>
–port <port>
-p, –project-path <projectPath>
-s, –software-project <softwareProjectName>

5.2.5 get-component-label

Get the label of a component

Options:

Name
-c,–component-id <string>
–host <host>
–port <port>
-p, –project-path <projectPath>
-s, –software-project <softwareProjectName>

5.2.6 get-rtos

Get Real-Time Operating System (RTOS) component of a SW project

Options:

Name
–host <host>
–port <port>
-p, –project-path <projectPath>
-s, –software-project <softwareProjectName>

5.2.7 list-component-conditions

List the conditions for a specific component

Options:

Name
-c,–component-id <string>
–host <host>
–port <port>
-p, –project-path <projectPath>
-s, –software-project <softwareProjectName>

5.2.8 list-component-fixes

List the available fixes for a specific component

Options:

Name
-c,–component-id <string>
–host <host>
–port <port>
-p, –project-path <projectPath>
-s, –software-project <softwareProjectName>

5.2.9 list-components

List all the components

Options:

Name
-e,–extended <boolean>
–host <host>
–port <port>
-p, –project-path <projectPath>
-s, –software-project <softwareProjectName>

5.2.10 list-files

List any library or dependency files of a SW project

Options:

Name
-c,–category <string>
–host <host>
–port <port>
-p, –project-path <projectPath>
-s, –software-project <softwareProjectName>

5.2.11 list-packs

List all packs of a SW project

Options:

Name
–host <host>
–port <port>
-p, –project-path <projectPath>
-s, –software-project <softwareProjectName>

5.2.12 list-project-fixes

List the available fixes for all the components of a specific software project

Options:

Name
–host <host>
–port <port>
-p, –project-path <projectPath>
-s, –software-project <softwareProjectName>

5.2.13 remove-all-components

Remove all the components

Options:

Name
–host <host>
–port <port>
-p, –project-path <projectPath>
-s, –software-project <softwareProjectName>

5.2.14 remove-component

Remove a specific component from an existing software component project

Options:

Name
-c,–component-id <string>
–host <host>
–keep-dependencies <boolean>
–port <port>
-p, –project-path <projectPath>
-s, –software-project <softwareProjectName>

5.2.15 remove-components

remove a list of components from an existing software component project

Options:

Name
–component-ids <string>
–host <host>
–port <port>
-p, –project-path <projectPath>
-s, –software-project <softwareProjectName>

5.2.16 remove-files

Remove any files or libraries from a SW project

Options:

Name
-f,–files ‘[<string>]’
-g,–group <string>
–host <host>
–port <port>
-p, –project-path <projectPath>
-s, –software-project <softwareProjectName>

5.2.17 reorder-component-instances

Reorder component software instances

Options:

Name
-c,–component-id <string>
–host <host>
-i,–instance-id <string>
-o,–overlapping-instance-id <string>
–port <port>
-p, –project-path <projectPath>
-s, –software-project <softwareProjectName>

5.2.18 resolve-dependencies

Resolve dependencies for a software project

Options:

Name
–host <host>
–port <port>
-p, –project-path <projectPath>
-r,–resolve-dependency-strategy <string>
-s, –software-project <softwareProjectName>

5.2.19 search-components

Search components

Options:

Name
–host <host>
–port <port>
-p, –project-path <projectPath>
–search-text <string>
-s, –software-project <softwareProjectName>

5.2.20 set-component-label

Change the component label

Options:

Name
-c,–component-id <string>
–host <host>
-l,–label <string>
–port <port>
-p, –project-path <projectPath>
-s, –software-project <softwareProjectName>

5.3 CLI: data-analytics

CLI for data analytics feature

5.3.1 get-telemetry

Retrieve the current telemetry data collection status. This command allows you to check whether telemetry data collection is enabled or disabled for your project.

Options:

Name	Description	Required	Default value
–host <host>	MX backend host	false	127.0.0.1
–port <port>	MX backend port	false	

5.3.2 set-telemetry

Opt-in or opt-out of telemetry data collection. This command allows you to enable or disable the collection of telemetry data, which helps us improve our products and services. By opting in, you agree to share anonymous usage data with us. You can opt out at any time using this command.

Arguments:

Name
<enabled>

Options:

Name	Description	Required	Default value
–host <host>	MX backend host	false	127.0.0.1
–port <port>	MX backend port	false	

5.4 CLI: dma

CLI for dma feature

5.4.1 bind-resource

Bind a specific DMA channel to a MemToMem software configuration.(–project-path (-p) and –channel (-c) are required)

Options:

Name
-c,–channel <string>
–host <host>
–port <port>
-p, –project-path <projectPath>
-s, –software-project <softwareProjectName>

5.4.2 extract-store

Returns a copy of the DMA store file generated by the tool. (–project-path (-p) is required)

Options:

Name	
–host <host>	MX
–port <port>	MX
-p, –project-path <projectPath>	Path to

5.4.3 get-allocations

Returns all the DMA channels allocations for a specific software project. (–project-path (-p) is required)

Options:

Name	
–host <host>	
–port <port>	
-p, –project-path <projectPath>	
-s, –software-project <softwareProjectName>	

5.4.4 list-channels

Returns the list of dma channels .(–project-path (-p) is required and –sw-project-path (-s) are required))

Options:

Name	
-d,–dma-controller <string>	
–host <host>	
–hw-instance <string>	
-n,–not-used <boolean>	
–port <port>	
-p, –project-path <projectPath>	
-s, –software-project <softwareProjectName>	
-u,–used <boolean>	

5.4.5 list-configurable-channels

Returns the list of configurable channels by IP. (–project-path (-p) is required)

Options:

Name		
–host <host>		MX
–port <port>		MX
-p, –project-path <projectPath>		Path to

5.4.6 list-controllers

Returns the list of dma controllers available in the SoC. (–project-path (-p) is required)

Options:

Name		
–host <host>		MX
–port <port>		MX
-p, –project-path <projectPath>		Path to

5.4.7 list-hw-resources

Returns the list of all the hardware resources allocated for a specific channel. (–project-path (-p) and –channel (-c) are required)

Options:

Name
-c,–channel <string>
–host <host>
–port <port>
-p, –project-path <projectPath>
-s, –software-project <softwareProjectName>

5.4.8 list-sw-instances

Returns the list of all the software instances ids used for a specific channel. (–project-path (-p) and –channel (-c) are required)

Options:

Name
-c,–channel <string>
–host <host>
–port <port>
-p, –project-path <projectPath>
-s, –software-project <softwareProjectName>

5.4.9 unbind-resource

unbind a specific DMA channel configured in a specific MemToMem configuration. (–project-path (-p) ,-sw-project-path (-s),–channel (-c) are required) and –sw-instance are required.

Options:

Name
-c,–channel <string>
–host <host>
–port <port>
-p, –project-path <projectPath>
-s, –software-project <softwareProjectName>
–sw-instance <string>

5.5 CLI: exti

CLI for exti feature

5.5.1 is-used

Returns if an exti line is used

Options:

Name
–host <host>
-l,–line-id <string>
–port <port>
-p, –project-path <projectPath>
-s, –software-project <softwareProjectName>

5.5.2 list-hw-resources

Returns the list of all the hardware resources using an exti line

Options:

Name
–host <host>
-l,–line-id <string>
–port <port>
-p, –project-path <projectPath>
-s, –software-project <softwareProjectName>

5.5.3 list-lines

Returns the list of exti lines being used|list all exti lines available in the HW

Options:

Name
–host <host>
–hw-instance <string>
-n,–not-used
–port <port>
-p, –project-path <projectPath>
-s, –software-project <softwareProjectName>
-u,–used

5.5.4 list-sw-instances

Returns the list of all the software instances ids used for a specific exti line

Options:

Name
–host <host>
-l,–line-id <string>
–port <port>
-p, –project-path <projectPath>
-s, –software-project <softwareProjectName>

5.6 CLI: finder

The finder command is designed to extract and display marketing information for boards and devices. It is particularly useful for developers and engineers who need specific details about supported MX products.

5.6.1 device

Groups all finder's commands related to device.

info

get device information per CPN or per PPN

Options:

Name	Description	Required	Default value
–cpn <cpn>	Filter by CPN	false	
–host <host>	MX backend host	false	127.0.0.1
–port <port>	MX backend port	false	
–ppn <ppn>	Filter by PPN	false	

list

get all devices

Options:

Name	Description	Required	Default value
–host <host>	MX backend host	false	127.0.0.1
–line <line>	Filter by line	false	
–port <port>	MX backend port	false	
–series <series>	Filter by series	false	

5.7 CLI: global-services

Provide users with the ability to manage/view project-level configuration options, enabling them to activate, configure, or deactivate settings as needed

5.7.1 disable

Disable a project global service

Arguments:

Name	Description
<globalServiceId>	Global service to disable

Options:

Name
–host <host>
–port <port>
-p, –project-path <projectPath>
-s, –software-project <softwareProjectName>

5.7.2 enable

Enable a project global service

Arguments:

Name	Description
<globalServiceId>	Global service to enable

Options:

Name	
–host <host>	
–port <port>	
-p, –project-path <projectPath>	
-s, –software-project <softwareProjectName>	

5.7.3 list

List the project global services

Options:

Name	
–host <host>	
–port <port>	
-p, –project-path <projectPath>	
-s, –software-project <softwareProjectName>	

5.8 CLI: hw-platform

CLI for hardware platform feature

5.8.1 board-info

get Board informations for a given project

Options:

Name		
–host <host>		MX
–port <port>		MX
-p, –project-path <projectPath>		Path to

5.8.2 mcu-info

get Mcu informations for a given project

Options:

Name		
–host <host>		MX
–port <port>		MX
-p, –project-path <projectPath>		Path to

5.9 CLI: ide-project

Cube IDE Project CLI

5.9.1 build-target

Manage the build targets of a software project

add

Add a new build target to the list of build targets

Options:

Name	
–force	
–host <host>	
–name <name>	
–port <port>	
–profile <profile>	
-p, –project-path <projectPath>	
–toolchain <toolchain>	

list

List all build targets for the project

Options:

Name	
–host <host>	MX
–port <port>	MX
-p, –project-path <projectPath>	Path to

remove

Remove a build target from the list of build targets

Options:

Name	
–host <host>	
–name <name>	
–port <port>	
–profile <profile>	
-p, –project-path <projectPath>	
-s, –software-project <softwareProjectName>	

5.9.2 flag

Manage the compiler flags of a build setup

add

Add flags to the build setup

Arguments:

Name	Description
<flags>	the list of flags

Options:

Name
–force
–from-scratch
–host <host>
–port <port>
-p, –project-path <projectPath>
–setup <build setup name>
-s, –software-project <softwareProjectName>
–tool <tool…>

list

List the flags of a build setup

Options:

Name
–host <host>
–port <port>
-p, –project-path <projectPath>
–setup <build setup name>
-s, –software-project <softwareProjectName>
–tool <tool…>

remove

Remove flags from the build setup

Arguments:

Name	Description
<flags>	the list of flags

Options:

Name
–host <host>
–port <port>
-p, –project-path <projectPath>
–setup <build setup name>
-s, –software-project <softwareProjectName>
–tool <tool…>

5.9.3 fmt

FMT stands for File Mapping Table, which is a set of rules to customize the placement inside the exported IDE project of packs, user modifiable files and generated files.

reload

Reload FMT rules defined in the running STM32CubeMX instance or provided in the file option and overwrite values in project

Options:

Name	
–file <fmtFilePath>	
–host <host>	
–port <port>	
-p, –project-path <projectPath>	

show

Show FMT rules values stored in the project

Options:

Name	
–host <host>	MX
–port <port>	MX
-p, –project-path <projectPath>	Path to

show-schema

Show the FMT JSON Schema

Options:

Name	Description	Required	Default value
–host <host>	MX backend host	false	127.0.0.1
–port <port>	MX backend port	false	

validate

Validate FMT rules file defined in the running STM32CubeMX instance or provided in the file option

Options:

Name	
–file <fmtFilePath>	The FMT ru
–host <host>	
–port <port>	

validate-project

Validate FMT rules file defined in the running STM32CubeMX instance or provided in the file option and compare it with the values stored in the project

Options:

Name	
–file <fmtFilePath>	
–host <host>	
–port <port>	
-p, –project-path <projectPath>	

5.9.4 generate

Generate the project

Options:

Name
–build-target <buildTarget…>
–destination <destination>
–diff <diff>
-f, –force
–format <format>
–host <host>
–keep-folder-structure <value>
–make-user-code-local <value>
–port <port>
-p, –project-path <projectPath>
–remove-board
-s, –software-project <softwareProjectName>
–source <source>

5.9.5 list-available-toolchains-by-format

List the available toolchains by format

Options:

Name	Description	Required	Default value
–host <host>	MX backend host	false	127.0.0.1
–port <port>	MX backend port	false	

5.9.6 list-format

List all available project formats

Options:

Name	Description	Required	Default value
–host <host>	MX backend host	false	127.0.0.1
–port <port>	MX backend port	false	

5.9.7 list-generator

List the generators for the project

Options:

Name	
–host <host>	MX
–port <port>	MX
-p, –project-path <projectPath>	Path to

5.9.8 list-source

List all available project sources

Options:

Name	Description	Required	Default value
–host <host>	MX backend host	false	127.0.0.1
–port <port>	MX backend port	false	

5.9.9 list-toolchain

List the available toolchains

Options:

Name	Description	Required	Default value
–host <host>	MX backend host	false	127.0.0.1
–port <port>	MX backend port	false	

5.9.10 set-destination

Set the destination folder for project generation

Options:

Name	
–destination <destination>	the g
–format <format>	
–host <host>	
–port <port>	
-p, –project-path <projectPath>	

5.9.11 set-format

Set the default format for the project - see list-format for available formats

Options:

Name	
–format <format>	The for
–host <host>	M
–port <port>	M
-p, –project-path <projectPath>	Path t

5.9.12 set-modifiable-base-dir

Set the directory for user modifiable code

Options:

Name	
–base-dir <baseDir>	
–host <host>	
–port <port>	
-p, –project-path <projectPath>	
-s, –software-project <softwareProjectName>	

5.9.13 set-remove-board

Set the removal of board and associated packs when generating the project

Options:

Name	
–format <format>	
–host <host>	
–port <port>	
-p, –project-path <projectPath>	
–remove-board	Remo

5.9.14 set-source

Set the source - see list-source for available sources

Options:

Name	
–format <format>	The for
–host <host>	M
–port <port>	M
-p, –project-path <projectPath>	Path t
–source <source>	the sour

5.9.15 set-target

Set the target on format [projectName][.buildType][+targetType] to assign to the project

Options:

Name	
–format <format>	The for
–host <host>	M
–port <port>	M
-p, –project-path <projectPath>	Path t
–target <target>	the

5.9.16 show-config

Show the export configuration of the project

Options:

Name	
–host <host>	MX
–port <port>	MX
-p, –project-path <projectPath>	Path to

5.9.17 status

Get status

Options:

Name
–host <host>
–only-application-project
–port <port>
-p, –project-path <projectPath>
-s, –software-project <softwareProjectName>

5.10 CLI: middleware

CLI for middleware feature

5.10.1 disable-middleware

Disable a middleware from the software project

Options:

Name
–host <host>
-n,–name <readableName>
–port <port>
-p, –project-path <projectPath>
-s, –software-project <softwareProjectName>

5.10.2 enable-middleware

Enable a middleware in the software project

Options:

Name
–host <host>
-n,–name <readableName>
–port <port>
-p, –project-path <projectPath>
-s, –software-project <softwareProjectName>

5.10.3 get-available-middleware

Get the list of all available middleware for the project

Options:

Name
–host <host>
–port <port>
-p, –project-path <projectPath>
-s, –software-project <softwareProjectName>

5.10.4 get-default-middleware

Get the list of default middleware for the project

Options:

Name	
–host <host>	
–port <port>	
-p, –project-path <projectPath>	
-s, –software-project <softwareProjectName>	

5.11 CLI: nvic

CLI for nvic feature

5.11.1 get-grouping

Returns grouping.

Options:

Name	
–host <host>	MX
–port <port>	MX
-p, –project-path <projectPath>	Path to
-v, –values	get

5.11.2 get-interrupt

Returns the nvic interrupt details.

Options:

Name	
–description	
–host <host>	
-n,–name <string>	
–port <port>	
–priority	get
-p, –project-path <projectPath>	
–status	

5.11.3 get-interrupts-by-instance

Returns nvic interrupts by instance.

Options:

Name	
–host <host>	MX
-i,–instance <string>	
–port <port>	MX
-p, –project-path <projectPath>	Path to

5.11.4 get-priority

Returns priority.

Options:

Name	
–host <host>	MX
-n,–name <string>	name of
–port <port>	MX
-p, –project-path <projectPath>	Path to
-v, –values	get

5.11.5 list-interrupts

Returns the list of nvic interrupts.

Options:

Name
–full
–host <host>
–peripheral
–port <port>
-p, –project-path <projectPath>
-s, –software-project <softwareProjectName>
–system
–used

5.11.6 set-grouping

Change priority grouping.

Options:

Name	
–host <host>	M
–port <port>	M
-p, –project-path <projectPath>	Path to
-v, –value <number>	priority

5.11.7 set-interrupts

Enable/Disable NVIC system interrupts.

Options:

Name
–host <host>
-n,–names <names...>
–port <port>
-p, –project-path <projectPath>
-s, –software-project <softwareProjectName>
–status <number>

5.11.8 set-priorities

Set priorities of NVIC interrupts.

Options:

Name	
–host <host>	
-i, –interrupts <string…>	
–port <port>	
-p, –project-path <projectPath>	
-s, –software-project <softwareProjectName>	

5.12 CLI: pack-manager

CLI for pack manager feature

5.12.1 add-registry

Add one or more registries to the pack manager configuration

Options:

Name		
–cmsis-pack-root <path>		Use speci
–host <host>		
–port <port>		
–url <url…>		

5.12.2 install

Install one or multiple packs using either URL, VENDOR.NAME.VERSION identifier, or a local extracted package folder.

Options:

Name		
–cmsis-pack-root <string>		
–destination <string>		Use sp
–force		
–host <host>		
–mcu-package <string>		
–pack-ids <string…>		
–port <port>		

5.12.3 list-boards

List of available board(s).

Options:

Name		
–cmsis-pack-root <path>		Use speci
–host <host>		
–port <port>		
–verbose		

5.12.4 list-devices

List of available devices(s).

Options:

	Name	
	–cmsis-pack-root <path>	Use speci
	–host <host>	
	–port <port>	

5.12.5 list-packs

List of available pack(s).

Options:

	Name	
	–cmsis-pack-root <path>	Use speci
	–host <host>	
	–port <port>	

5.12.6 remove

Remove selected pack(s).

Options:

	Name	
	–cmsis-pack-root <string>	Path
	–host <host>	
	–pack-ids <string...>	
	–port <port>	

5.13 CLI: parts

CLI for parts feature

5.13.1 add-standalone-part

Add a standalone hardware part to the project by applying case-insensitive substring filters (metadata & pack id). Fails if zero or multiple parts match; succeeds only when exactly one candidate is found.

Options:

Name
–hclass <readableName>
–hgroup <readableName>
–hname <readableName>
–host <host>
–hrevision <readableName>
–hsub <readableName>
–hvariant <readableName>
–hvendor <readableName>
–packidName <readableName>
–packidVendor <readableName>
–packidVersion <readableName>
–port <port>
-p, –project-path <projectPath>
–shortname <readableName>
-s, –software-project <softwareProjectName>

5.13.2 disable-part

Disable a part for the targeted sw project

Options:

Name
–host <host>
–pcb <readableName>
–port <port>
-p, –project-path <projectPath>
-r,–referenceOnBoard <readableName>
-s, –software-project <softwareProjectName>

5.13.3 enable-part

Enable a part for the targeted sw project

Options:

Name
–host <host>
–pcb <readableName>
–port <port>
-p, –project-path <projectPath>
-r,–referenceOnBoard <readableName>
-s, –software-project <softwareProjectName>

5.13.4 get-mounted-parts

Get the list of mounted parts for the project

Options:

Name
–host <host>
–port <port>
-p, –project-path <projectPath>
-s, –software-project <softwareProjectName>

5.13.5 get-standalone-parts

List all standalone parts currently added to the project; excludes mounted parts already available through the board definition.

Options:

Name
–host <host>
–port <port>
-p, –project-path <projectPath>
-s, –software-project <softwareProjectName>

5.13.6 remove-standalone-part

Remove a standalone hardware part by its exact referenceOnBoard; only succeeds if the reference matches one and only one standalone part.

Options:

Name
–host <host>
–port <port>
-p, –project-path <projectPath>
-r,–referenceOnBoard <readableName>
-s, –software-project <softwareProjectName>

5.13.7 search-standalone-parts

List compatible standalone parts applying case-insensitive substring filters; returns metadata summaries.

Options:

Name
–hclass <readableName>
–hgroup <readableName>
–hname <readableName>
–host <host>
–hrevision <readableName>
–hsub <readableName>
–hvariant <readableName>
–hvendor <readableName>
–packidName <readableName>
–packidVendor <readableName>
–packidVersion <readableName>
–port <port>
-p, –project-path <projectPath>
–shortname <readableName>
-s, –software-project <softwareProjectName>

5.14 CLI: peripherals

CLI for peripherals feature

5.14.1 description

Display the informations about a peripheral.

Options:

Name	Description	Required	Default value
–cpn <cpn>	Specify the CPN of a board	true	
–host <host>	MX backend host	false	127.0.0.1
–peripheral <peripheral>	The id of the peripheral	true	
–port <port>	MX backend port	false	

5.14.2 disable

Disable a peripheral.

Options:

Name
–host <host>
–peripheral <peripheral>
–port <port>
-p, –project-path <projectPath>
–software-project <softwareProject>

5.14.3 enable

Enable a peripheral.

Options:

Name
–host <host>
–mode <mode>
–override
–peripheral <peripheral>
–port <port>
-p, –project-path <projectPath>
–software-project <softwareProject>

5.14.4 list

List peripherals in an MX2 project.

Options:

Name	
–enabled	Fil
–host <host>	
–port <port>	
-p, –project-path <projectPath>	
–search <search>	

5.14.5 view

Display the id, name, and other information about a peripheral.

Options:

Name	
–host <host>	
–peripheral <peripheral>	T
–port <port>	
-p, –project-path <projectPath>	
–software-project <softwareProject>	

5.15 CLI: pinout

Pinout CLI for STM32CubeMX2

5.15.1 add-label

Add a label to a pin.

Options:

Name	
–host <host>	
–label <label>	label (
–pin <pinName>	
–port <port>	
-p, –project-path <projectPath>	

5.15.2 add-to-group

Adds a pin to a GPIO group (uses the default group if –group is omitted).

Options:

Name	
–group <group>	group
–host <host>	
–pin <pinName>	
–port <port>	
-p, –project-path <projectPath>	

5.15.3 all-pins

Returns all the pins.

Options:

Name	
–host <host>	MX
–port <port>	MX
-p, –project-path <projectPath>	Path to
–short	

5.15.4 assign-signal

Assigns a signal to a pin.

Options:

Name	
–host <host>	
–pin <pinName>	pin
–port <port>	
-p, –project-path <projectPath>	
–signal <signal>	signal

5.15.5 assigned-pins

Returns all pins with assigned signal(s).

Options:

Name	
–host <host>	MX
–port <port>	MX
-p, –project-path <projectPath>	Path to
–short	

5.15.6 exclude-pin

Excludes a pin from pinout configuration and automatic resolution algorithm.

Options:

Name	
–host <host>	
–pin <pinName>	pin nam
–port <port>	
-p, –project-path <projectPath>	Pat

5.15.7 group

Returns a GPIO group description (default group if groupName is omitted).

Options:

Name	
–group <group>	group
–host <host>	
–port <port>	
-p, –project-path <projectPath>	
–short	

5.15.8 include-pin

Includes a pin (previously excluded) for pinout configuration and automatic resolution algorithm.

Options:

	Name	
	–host <host>	
	–pin <pinName>	pin nam
	–port <port>	
	-p, –project-path <projectPath>	Pat

5.15.9 needs

Get for each configured pin, the SW instance using it and with which signal.

Options:

	Name	
	–host <host>	
	–instance [instanceName]	instanc
	–port <port>	
	-p, –project-path <projectPath>	
	–short	

5.15.10 pin

Returns a pin description.

Options:

	Name	
	–host <host>	
	–pin <pinName>	pin nam
	–port <port>	
	-p, –project-path <projectPath>	Pat
	–short	

5.15.11 pin-signal

Pins a signal to a pin.

Options:

	Name	
	–host <host>	
	–pin <pinName>	pin
	–port <port>	
	-p, –project-path <projectPath>	
	–signal <signal>	signal

5.15.12 remove-from-group

Removes a pin from a GPIO group (default group if groupName is omitted).

Options:

	Name	
	–group <group>	group
	–host <host>	
	–pin <pinName>	
	–port <port>	
	-p, –project-path <projectPath>	

5.15.13 remove-label

Remove a label from a pin.

Options:

	Name	
	–host <host>	
	–label <label>	label (
	–pin <pinName>	
	–port <port>	
	-p, –project-path <projectPath>	

5.15.14 reset-pin

Reset a pin. Reserved signals are reset, configured signals are preserved, labels are preserved, excluded pins are restored

Options:

	Name	
	–host <host>	
	–pin <pinName>	pin nam
	–port <port>	
	-p, –project-path <projectPath>	Pat

5.15.15 reset-pinout

Reset all the pins. Reserved signals are reset, configured signals are preserved, labels are preserved, excluded pins are restored

Options:

	Name	
	–host <host>	MX
	–port <port>	MX
	-p, –project-path <projectPath>	Path to

5.15.16 unassign-signal

Unassigns a signal from a pin.

Options:

Name	
–host <host>	
–pin <pinName>	pin
–port <port>	
-p, –project-path <projectPath>	
–signal <signal>	signal

5.15.17 unpin-signal

Unpins a signal from a pin.

Options:

Name	
–host <host>	
–pin <pinName>	pin
–port <port>	
-p, –project-path <projectPath>	
–signal <signal>	signal

5.16 CLI: preferences

CLI for preferences feature

5.16.1 get-preference

Return the current value of a specified preference

Options:

Name	
–default-value	Return the defa
–host <host>	
–port <port>	
–possible-values	Return the possib
–pref <preference>	Return the curr

5.16.2 get-preferences

Return the whole list of preferences with their current values

Options:

Name	
–default-values	Return the who
-e, –export <path>	
–host <host>	
–port <port>	
–possible-values	Return the whol

5.16.3 list-preferences

Display the list of user preferences

Options:

Name	Description	Required	Default value
–host <host>	MX backend host	false	127.0.0.1
–port <port>	MX backend port	false	

5.16.4 set-preference

Change the value of a specified preference

Options:

Name	
–host <host>	M
–port <port>	M
–pref <preference>	The
–value <value>	New value for the spe

5.16.5 set-preferences

Set several preferences at once

Options:

Name	
–host <host>	
–port <port>	
–prefs <preferences>	JSON file path

5.17 CLI: project

5.17.1 close

close a project

Options:

Name	
–discard	If
–host <host>	
–port <port>	
-p, –project-path <projectPath>	

5.17.2 create-from-board

Create an Application Project from Board

Options:

Name
–cpn <cpn>
–empty-project
–host <host>
–overwrite
–pack-name <packName>
–pack-version <packVersion>
–port <port>
-l, –project-location <projectLocation>
-n, –project-name <projectName>

5.17.3 create-from-mcu

Create an Application Project from MCU

Options:

Name
–cpn <cpn>
–empty-project
–host <host>
–overwrite
–pack-name <packName>
–pack-version <packVersion>
–port <port>
-l, –project-location <projectLocation>
-n, –project-name <projectName>

5.17.4 create-software

create a software Project

Options:

Name
–host <host>
–port <port>
-p, –project-path <projectPath>
–software-project <softwareProject>

5.17.5 delete

delete a project

Options:

Name
–host <host>
–port <port>
-p, –project-path <projectPath>
-s, –software-project <softwareProjectName>

5.17.6 get-sw-project-list

list software projects in the project

Options:

Name	
–host <host>	MX
–port <port>	MX
-p, –project-path <projectPath>	Path to

5.17.7 pack

Manage application project packs

add

The command adds a pack inside the project csolution.

Options:

Name	
–host <host>	MX
–pack-id <packId>	T
–port <port>	MX
-p, –project-path <projectPath>	Path to

list

The command enumerates all the packs utilized within a project.

Options:

Name	
–host <host>	
–port <port>	
-p, –project-path <projectPath>	
–upgradable	lists p

update

The command updates a pack to a newer version.

Options:

Name	
–host <host>	
–newer-pack-version <newerPackVersion>	
–pack-name <packName>	
–pack-vendor <packVendor>	
–port <port>	
-p, –project-path <projectPath>	

5.17.8 save

save a project

Options:

Name	
–host <host>	MX
–port <port>	MX
-p, –project-path <projectPath>	Path to

5.17.9 set-linked

Set IOC2 linked attribute to true

Options:

Name	
–host <host>	MX
–port <port>	MX
-p, –project-path <projectPath>	Path to

5.17.10 set-read-only

Set IOC2 read-only attribute to true

Options:

Name	
–host <host>	MX
–port <port>	MX
-p, –project-path <projectPath>	Path to

5.17.11 unset-linked

Unset IOC2 linked attribute (remove the flag)

Options:

Name	
–host <host>	MX
–port <port>	MX
-p, –project-path <projectPath>	Path to

5.17.12 unset-read-only

Unset IOC2 read-only attribute (remove the flag)

Options:

Name	
–host <host>	MX
–port <port>	MX
-p, –project-path <projectPath>	Path to

5.18 CLI: sw-config

CLI for sw-configuration feature

5.18.1 add-array-element

Add an element to an array

Options:

Name
-e, –element-to-add <elementToAdd>
–host <host>
–parameter-path <pathToParameter>
–port <port>
-p, –project-path <projectPath>
-r, –resource-name <resourceName>
-s, –software-project <softwareProjectName>

5.18.2 get-parameter-actions

Get the actions (if any, from the parameters.json) of a given parameter

Options:

Name
–host <host>
–parameter-path <pathToParameter>
–port <port>
-p, –project-path <projectPath>
-r, –resource-name <resourceName>
-s, –software-project <softwareProjectName>

5.18.3 get-parameter-possible-values

Get the possible values of a given parameter

Options:

Name
–host <host>
–parameter-path <pathToParameter>
–port <port>
-p, –project-path <projectPath>
-r, –resource-name <resourceName>
-s, –software-project <softwareProjectName>
–use-internal-value

5.18.4 get-parameter-value

Get the value of a given parameter

Options:

Name
–host <host>
–parameter-path <pathToParameter>
–port <port>
-p, –project-path <projectPath>
-r, –resource-name <resourceName>
-s, –software-project <softwareProjectName>
–use-internal-value

5.18.5 list-available-parameters

List the visible parameters in a given component instance

Options:

Name
-d, –details <string>
–host <host>
–port <port>
-p, –project-path <projectPath>
-r, –resource-name <resourceName>
-s, –software-project <softwareProjectName>
–use-internal-value

5.18.6 list-configurations

List the configurations of the project (the whole ones or for an optional component)

Options:

Name
–host <host>
–port <port>
-p, –project-path <projectPath>
-s, –software-project <softwareProjectName>
-c, –sw-component-id <string>

5.18.7 list-parameters

List all the parameters in a given component instance

Options:

Name
–host <host>
–port <port>
-p, –project-path <projectPath>
-r, –resource-name <resourceName>
-s, –software-project <softwareProjectName>

5.18.8 move-array-element

Move an element to a given rank

Options:

Name
-e, –element-to-move <elementToMove>
–host <host>
–parameter-path <pathToParameter>
–port <port>
–position <position>
-p, –project-path <projectPath>
-r, –resource-name <resourceName>
-s, –software-project <softwareProjectName>

5.18.9 move-array-elements

Move several elements to the given ranks

Options:

Name
–host <host>
-m, –moves <moves>
–parameter-path <pathToParameter>
–port <port>
-p, –project-path <projectPath>
-r, –resource-name <resourceName>
-s, –software-project <softwareProjectName>

5.18.10 remove-array-element

Remove an element from an array

Options:

Name
-i, –element-index <elementIndex>
–host <host>
–parameter-path <pathToParameter>
–port <port>
-p, –project-path <projectPath>
-r, –resource-name <resourceName>
-s, –software-project <softwareProjectName>

5.18.11 reset-configuration

Reset a configuration (restores default; same action than UI button Reset Configuration)

Options:

Name
–host <host>
–port <port>
-p, –project-path <projectPath>
-r, –resource-name <resourceName>
-s, –software-project <softwareProjectName>

5.18.12 set-multiple-parameters

Set the value of several parameters

Options:

Name
–host <host>
–parameters <parameters>
–port <port>
-p, –project-path <projectPath>
-r, –resource-name <resourceName>
-s, –software-project <softwareProjectName>

5.18.13 set-parameter-value

Set the value of a given parameter

Options:

Name
–host <host>
–parameter-path <pathToParameter>
–port <port>
-p, –project-path <projectPath>
-r, –resource-name <resourceName>
-s, –software-project <softwareProjectName>
-v, –value <value>

5.19 CLI: utilities

CLI for utilities feature

5.19.1 disable-utility

Disable a utility from the software project

Options:

Name
–host <host>
-n,–name <readableName>
–port <port>
-p, –project-path <projectPath>
-s, –software-project <softwareProjectName>

5.19.2 enable-utility

Enable a utility in the software project

Options:

Name
–host <host>
-n,–name <readableName>
–port <port>
-p, –project-path <projectPath>
-s, –software-project <softwareProjectName>

5.19.3 get-available-utilities

Get the list of all available utilities for the project

Options:

Name
–host <host>
–port <port>
-p, –project-path <projectPath>
-s, –software-project <softwareProjectName>

5.19.4 get-default-utilities

Get the list of default utilities for the project

Options:

Name
–host <host>
–port <port>
-p, –project-path <projectPath>
-s, –software-project <softwareProjectName>

Last updated 2026-10-13 b'16:33:50 CEST'

6 How to

The main points to get started are:

6.1 How to configure a clock

Configuring the clock system is a fundamental step in STM32 microcontroller projects, as it determines the operating frequency and timing for the entire system. STM32CubeMX2 provides a graphical interface to configure clocks easily.

Here's a detailed guide to identify and configure common clock scenarios:

6.1.1 Understanding STM32 Clock System Basics

- STM32 MCUs have multiple clock sources: - Fixed source/variable source and PLL to reach the max system frequency. - HSI (internal RC oscillator), HSE (external crystal), LSI, LSE, PLL, etc.
- The clock tree controls CPU clock, peripheral clocks, and bus clocks (AHB, APB1, APB2).
- Proper clock configuration ensures correct setup by detecting peripheral frequencies that exceed the maximum limit and suggests automatic fixes.

6.1.2 Clock Configuration access

- Open your project in STM32CubeMX2.
- Go to the clock configuration tab to view the visual clock tree graph (similar to the reference manual).
- Set values for clock elements such as dividers, multipliers, or sources to configure the clock.
- Let STM32CubeMX2 validate the configuration and warn if frequencies are out of range.
- Click the resolve clock issue button to get a solution for an error-free clock tree.
- Revert to the default configuration if needed.
- Enter a desired frequency; the system will find a valid configuration or notify if none exists.

6.1.3 Typical Clock Configuration Scenarios

Scenario A: Use Internal Oscillator (HSI) as a system clock

- Use case: Simple applications without external crystal.
- Steps:
  1. In STM32CubeMX2, go to the Clock Configuration tab.
  2. Select HSI as the clock source for the system clock (SYSCLK).
  3. Configure System Clock if a higher frequency is needed.
  4. Verify bus clocks (AHB, APB1, APB2) frequencies.
  5. CubeMX will show if the configuration is valid.

Scenario B: Configure Peripheral Clocks (e.g., USB, ADC, Timers)

- Use case: Some peripherals need specific clock frequencies.
- Steps:
  1. In clock configuration, check peripheral clocks.
  2. Configure dedicated dividers if needed.
  3. Ensure that peripheral clock frequencies meet specifications.

Scenario C: Clock configuration with errors + resolve issue

1. Introduce an error in the configuration.
2. Click on **Resolve Clock Issues**.

Scenario D: Clock source availability in RCC Config and Clock Tree

The availability of a clock source in the **Clock Tree** depends on its activation state in **RCC Config**.

1. When a clock source is disabled in RCC Config:

- The clock source cannot be edited in the Clock Tree view.
- The clock source cannot be selected as a clock source in the Clock Tree view.
- An information tooltip is displayed, indicating that the user must first enable the clock source in RCC.

2. When a clock source is enabled in RCC Config:

- The clock source becomes editable in the Clock Tree view.
- The clock source can be selected or left unused as a clock source in the Clock Tree.

3. When a clock source is later disabled in RCC Config after being used in the Clock Tree:

- The clock source becomes disabled in the Clock Tree.
- If it is currently selected as a clock source somewhere in the Clock Tree, such as for the **system clock** or a **peripheral clock**, it is highlighted in **Error** on the corresponding **Mux**.

Scenario E: Clock Reset to Default Value

- If you want to return to the default configuration, click on the **Reset Clock** button.

Scenario F: Clock Table

- You can use the table view to set your clock configuration.

Scenario G: lock/ unlock elements

- You can lock or unlock clock element.

Scenario H: Enable/ disable clock elements

- You can enable or disable clock elements in the RCC configuration tab. For example, enable HSE in the RCC configuration tab.
- You can also enable or disable clock peripheral elements in their corresponding peripherals. For example, activate or deactivate DAC.

6.1.4 Tips and Best Practices

- Always check the maximum frequency specifications of your MCU.
- Use external crystal (HSE) for applications needing precise timing.
- Configure low-power oscillators (LSI, LSE) for RTC and watchdog.
- Review peripheral clock requirements (USB, ADC, etc.).
- Use STM32CubeMX2 validation feature to avoid invalid configurations.

6.1.5 Additional articles

- How to use clock configuration in STM32CubeMX2 for STM32C5
- How to use the MCO with STM32CubeMX2
- How to change the HAL2 TimeBase from Systick to a timer in STM32CubeMX2
- How to use RTC and backup registers with STM32CubeMX2

6.2 How to select an IDE and generate code

Selecting an IDE and generating code with STM32CubeMX2, then compiling your project, involves a few clear steps. Here's a comprehensive guide:

6.2.1 Selecting an IDE

STM32CubeMX2 supports generating project code for several popular IDEs:

- IAR Embedded Workbench®
- Open-CMSIS (compatible with Keil® MDK-Arm)
- CMake (compatible with STM32 VS Code and STM32CubeIDE)

How to select IDE in STM32CubeMX2:

1. Open your project in STM32CubeMX2.
2. Navigate to the project manager tab.
3. Locate the IDE project generation section under project settings.
4. Select your desired format from the list (for example, CMake).
5. Configure all needed input fields, or keep the default settings.

6.2.2 Generating Code

- After configuring peripherals, clocks, and middleware, click the **Generate Code** button (gear icon or from **Project Settings** tab).
- STM32CubeMX2 will generate the project files tailored for the selected IDE.

6.2.3 Additional tips

- Keep STM32CubeMX2 and your IDE updated for compatibility.
- Configure debug settings in your IDE to match your hardware.
- Use STM32CubeMX's **Project manager** tab to customize code generation options.

6.2.4 Build project

- Load the previous generated project in VSCode
- Select Debug configuration
- Configure the board/Device with the right board name
- Save configuration
- Once the project is well configured launch the build
- Check that no error occurred

6.2.5 Additional articles

- How to generate code for VS Code from STM32CubeMX2
- How to generate code for IAR from STM32CubeMX2

6.3 How to reserve a GPIO pin

This guide provides a step-by-step procedure to set a GPIO pin in Reserved or Configured mode.

6.3.1 Step 1: Access the Pinout Tab

In the Pinout tab, view the microcontroller pins.

6.3.2 Step 2a: Set a GPIO in Reserved mode

2a-1: Reserve a GPIO Pin using the Detail view in the right of the screen

Left-click the desired pin in the MCU package graphic view. In the detail view on the right of the screen, select GPIO. By default, the pin is set as Configured. Select Reserved. The pin color appears in light green.

Follow the animation to reserve a pin using the right menu.

2a-2: Configure pin with the contextual menu

Right-click the desired pin in the MCU package graphic view to open its contextual menu. In the contextual menu, select GPIO. By default, the pin is set as Configured. Select Reserved. The pin color appears in light green.

Follow the animation to reserve a pin using the contextual menu.

6.3.3 Step 2b: Set a GPIO in Configured mode

2b-1: Configure a GPIO Pin using the Detail view in the right of the screen

Left-click the desired pin in the MCU package graphic view. In the detail view on the right of the screen, select GPIO. By default, the pin is set as Configured. If the pin has been previously Reserved, select Configured. The pin color appears in dark green.

Follow the animation to configure a pin using the right menu.

2b-2: Configure pin with the contextual menu

Right-click the desired pin in the MCU package graphic view to open its contextual menu.

In the contextual menu, select GPIO. By default, Configured is selected. If the pin was previously set as Reserved, select Configured. The pin color appears in dark green.

Follow the animation to configure a pin using the contextual menu.

Troubleshooting

If issues occur, consult the documentation or verify the pin configuration.

6.4 How to access peripherals configuration

This guide provides a step-by-step tutorial **on how to access IP configuration.**

Two cases are shown: - IP with direct enablement (for example, ADC) - IP with protocol selection (for example, I2C)

6.4.1 1: How to enable the IP ADC

Open the **Peripheral** menu by clicking **its icon on the left panel.**

Fig. 1: Icon of the peripheral menu

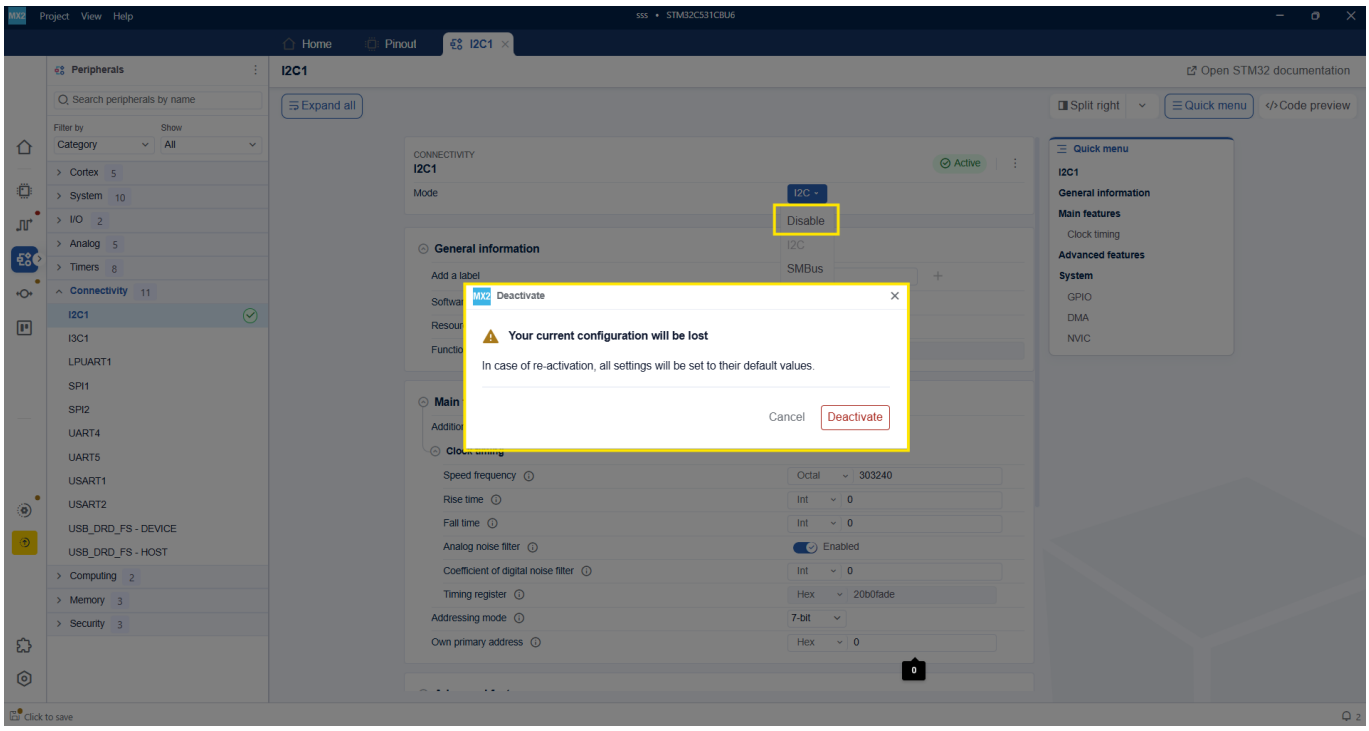
Select **ADC1**. If sorted by category, it belongs to **Analog**. Click the **Activate** button.

6.4.2 2: How to enable the IP I2C

Open the Peripheral menu by clicking on its icon on the left panel. Select **I²C1**. If sorted by category, it belongs to **Communication**. Click the **Activate** button.

6.4.3 3: How to deactivate a Peripheral

To deactivate a peripheral, select the Disable Mode from the configuration main tab. A popup will ask for confirmation before deactivation. Deactivating a peripheral disables its functionality and removes its configurations from the project.



Troubleshooting

If you encounter issues, check the documentation or verify the pin configuration.

6.5 How to access hardware parts

This section provides a step-by-step guide to using hardware parts (also known as Parts) in STM32CubeMX2. Follow the instructions below to manage HW Parts effectively:

6.5.1 1. Access the Parts Panel

- Open the STM32CubeMX2 application.
- Click the “Parts” button on the left side bar to open the left “Parts” panel.

- Starting from the “Board” project, the left “Parts” panel displays a list of available hardware parts mounted on each board, but not yet activated.
- It is possible to add additional hardware parts (also known as standalone parts) to the project. These will be displayed in the Standalone Part section in the left **Parts** panel.

6.5.2 2. Activate a Part

- Locate the part to activate in the left **Parts** panel.
- Click the part to display the configuration main tab.
- Activate the part by using the **Activate** button in the configuration main tab.
- After activation, the part’s configuration options become available.

6.5.3 3. Browse a Part configuration and views

- After activating a Part, explore its configuration options.
- Navigate through different configuration chapters and views, such as code preview and detail view, associated with the part.
- Use Expand/Collapse options to manage the visibility of configuration sections.
- Make necessary adjustments to tailor the part configuration to your project requirements.

6.5.4 4. Deactivate a Part

- To deactivate a part, use the **Deactivate** button next to its name from the configuration main tab. A pop up will ask for confirmation before deactivation.
- Deactivating a part disables its functionality and removes its configurations from the project.

6.5.5 5. Find and Add a new Part

- If the desired part is not displayed, use the **Add Standalone Part(s)** button in the left **Parts** panel.
- Browse the available parts library and select the required part.
- Click **Add** to include it in your project. The part appears in the left **Parts** panel.

> **Note**
>
> Adding a part to the project does not automatically add its software configuration. You must activate the part to include its software configuration in the project.

6.5.6 6. Remove a standalone Part

- To remove a standalone part from the project, first deactivate the part if it is active. In the part configuration main tab, click on the **Deactivate** button.
- Next, in the left **Parts** panel, click on the three dots at the end of the part line to open the dedicated menu. Click on **Remove**.
- The part will disappear from the left **Parts** panel.

> **Note**
>
> this is not possible to remove a board part, only standalone parts can be removed from the project.

6.5.7 7. Conclusion

By following these steps, you can efficiently manage hardware parts in STM32CubeMX2.

6.5.8 Additional articles

How to use the STM32CubeMX2 Part Drivers and BSP

6.6 How to access middleware & utilities

This section provides a step-by-step guide to using middlewares and utilities in STM32CubeMX2. Follow the instructions below to manage middlewares and utilities:

6.6.1 1. Access the Middleware & Utilities Panel

- Open the STM32CubeMX2 application.
- In the left sidebar, click on **Open Middlewares & Utilities** button to open the **Middlewares & Utilities** panel.
- The left panel displays a list of available middlewares and utilities.
- After project creation, the list contains default middlewares and utilities (not yet activated), if any. These middlewares and utilities cannot be removed from the list, and their names are prefixed by a lock icon.
- It is possible to add more middlewares and utilities to the project.

6.6.2 2. Activate a middleware or utility

- Locate the middleware or utility to activate in the **Middlewares & Utilities** left panel.
- Click on the middleware or utility to display its configuration tab.
- Activate the middleware or utility by clicking on **Activate** button in the configuration main tab.
- After activation, the configuration options for the middleware or utility become available.

6.6.3 3. Browse a middleware or utility configuration and views

- After activation, explore the configuration options for the middleware or utility.
- Navigate through configuration chapters and views, such as code preview and detail view, associated with the middleware or utility.
- Use Expand/Collapse options to manage the visibility of configuration sections.
- Adjust the configuration to meet project requirements.

6.6.4 4. Deactivate a middleware or utility

- To deactivate a middleware or utility, click on **Deactivate** next to its name in the configuration tab. A pop up will ask for confirmation before deactivation.
- Deactivating a middleware or utility disables its functionality and removes its configuration from the project.

6.6.5 5. Find and add a new middleware

- If the desired middleware is not listed, click on **Add middleware...** button in the **Middlewares & Utilities** section of the left panel to open the **Add middleware to your middleware panel** selection windows.
- The selection window displays the middleware library available for the project. Middlewares already in the project are selected and cannot be unselected.
- Browse the available middleware library and select one or more to add to the project. Use filters to search for the desired middleware, and add columns for new filters or search criteria.
- Click on **Add (x) middleware to your middleware panel** to include the selected middleware in your project. The new middleware appears in the **Middlewares & Utilities** left panel.
- After adding the middleware, activate it to access its features.

6.6.6 6. Find and add a new utility

- If the desired utility is not listed, click on **Add utility...** button in the **Middlewares & Utilities** section of the left panel to open the **Add utility to your utility panel** selection windows.
- The selection window displays the utility library available for the project. Utilities already in the project are selected and cannot be unselected.
- Browse the available utility library and select one or more to add to the project. Use filters to search for the desired utility, and add columns for new filters or search criteria.
- Click on **Add (x) utility to your utility panel** to include the selected utility in your project. The new utility appears in the **Middlewares & Utilities** left panel.
- After adding the utility, activate it to access its features.

6.6.7 7. Remove a middleware or utility

- To remove a middleware or utility from the left panel, first deactivate the middleware or utility if it is active. In the middleware or utility configuration main tab, click on the **Deactivate** button.
- Next, in the left **Middlewares & Utilities** section panel, click on the three dots at the end of the middleware or utility line to open the dedicated menu. Click **Remove**.
- The middleware or utility will disappear from the left **Middlewares & Utilities** panel.

> **Note**
>
> this is not possible to remove a default middlewares or utilities, only added middlewares or utilities can be removed from the project.

6.6.8 8. Conclusion

By following these steps, you can efficiently manage middleware and utilities in CubeMX2.

6.7 How to configure DMA

6.7.1 1. Access the DMA System View Page

- Launch the CubeMX2 application.
- Navigate to the “DMA” section from the left panel or the system overview.
- Click on the “DMA System View” to open the dedicated page for configuring and monitoring DMA settings.

6.7.2 2. Reserve DMA Channels

- In the DMA system view, locate the list of available DMA channels.
- Select the channel to reserve.
- To reserve a channel, click in the "Request" column to display all possible reservation options, then select the desired request to complete the reservation.
- The channel is marked as reserved and associated with the selected request.
- Alternatively, manage reservations using the detailed view on the right side of the panel.

6.7.3 3. Activate a Memory-to-Memory Configuration

- In the DMA system view, select an available DMA channel.
- Choose the "MemToMem/Lli" request from the list of available requests.
- The configuration is created. To access its settings, select the cogwheel icon at the end of the line.
- Alternatively, perform this operation using the detailed view on the right side of the panel

6.7.4 4. View and Resolve Allocation Conflicts

- There are two levels of issues: **warnings** and **errors**. A warning indicates that there are reservations or a memory-to-memory configuration, and at least one reservation exists. An error indicates that two or more configured requests are involved on one channel.
- Conflicting channels are highlighted, and a status of **Error** or **Warning** is shown.
- **To resolve conflicts:**
  - Change the channel assignment for one of the conflicting requests, or
  - Release a request by unassigning it from the channel (only available for memory-to-memory configurations or reserved requests).

> **Note**
>
> **There are several ways to resolve DMA conflicts:**
>
> - Go to the configuration panel where the conflict occurs and change the channel or the request
> - In the DMA system view, change the channel or the request
> - In the DMA system view, use the **Move** or **Swap** menu.
> - In the DMA detailed view, use the **Move** or **Swap** menu.
> - Release one of the conflicting requests (memory-to-memory configuration or reserved request) from the DMA system view.

Once all conflicts are resolved, the error indication disappears and the configuration can be saved.

6.7.5 5. Conclusion

By following these steps, you can efficiently manage DMA configurations in CubeMX2.

6.7.6 Additional articles

- How to use the DMA in normal mode with STM32CubeMX2

6.8 How to configure EXTI

Configuring the extended interrupts and Event controller (EXTI) in STM32CubeMX2 involves setting GPIO pins to trigger interrupts on signal changes (rising or falling edges). Below is a detailed guide to identify and configure common EXTI scenarios:

6.8.1 EXTI Basics

- EXTI provides a number of interrupt and event lines: 16 lines directly connected to GPIOs. Some lines are connected internally to peripherals.
- You can configure EXTI lines to trigger on rising edge, falling edge, or both.
- EXTI interrupts are managed via NVIC, so priority and enabling must be configured.

6.8.2 Typical EXTI Configuration Scenarios in CubeMX

Scenario A: Configure a GPIO pin as an external interrupt source

- Use case: Detect button press or external event.
- Steps:
  1. In STM32CubeMX2, select the GPIO pin you want to use.
  2. Configure the GPIO pull-up/pull-down as needed.
  3. In the **GPIO Configuration** tab, enable the EXTI.
  4. Under EXTI Configuration, enable interruption.
  5. Set interrupt priority if needed.
  6. Go to NVIC Tab and check that the corresponding EXTI interrupt (e.g., *EXTI0_IRQn*) is enabled (optional).

Scenario B: Configure Multiple EXTI Pins

- Use case: Multiple buttons or sensors triggering interrupts.
- Steps:
  1. Configure each GPIO pin as EXTI line (e.g., PA0 as EXTI0, PC13 as EXTI13).
  2. Enable corresponding EXTI interrupts in NVIC (*EXTI0_IRQn*, *EXTI15_10_IRQn*).
  3. Assign priorities accordingly.

Scenario C: Configure Edge Trigger Type

- Use case: Trigger interrupt on rising edge, falling edge, or both.
- Steps:
  1. After selecting GPIO, go to the GPIO configuration tab and enable EXTI.
  2. Set Trigger to: - Rising edge - Falling edge - Rising and falling edge
  3. This configures the EXTI line behavior.

Scenario D: Configure EXTI Internal Line

- Example: RCC interrupt
- The RCC CSS interrupt is internally connected to EXTI line 19.

- This EXTI line is internal, meaning it is not connected to an external GPIO pin but to an internal peripheral event.
- Steps: - Enable RCC interrupt. - Enable LSE. - Enable the Clock Security System feature in LSE.

Scenario E: Disable EXTI Interrupt Temporarily

- Use case: Temporarily ignore interrupts without changing GPIO.
- Steps:
  1. In GPIO Configuration tab, disable interruption.
  2. This disables the NVIC interrupt but keeps GPIO configured as EXTI.

6.8.3 Additional articles

- How to use the EXTI with STM32CubeMX2

6.9 How to configure NVIC

Configuring the Nested Vectored Interrupt controller (NVIC) in STM32CubeMX2 is essential for setting interrupt priorities and enabling interrupts on STM32 microcontrollers. Here's a detailed guide to identify and configure common NVIC scenarios.

6.9.1 Typical Scenarios for NVIC Configuration in STM32CubeMX2

How to access NVIC settings in CubeMX

- Open your project in CubeMX.
- Go to **System > NVIC** tab.
- Here you can: - Configure priorities for each enabled interrupt. - For settable system Interruption (exceptions) you can enable/disable the interrupt.
- You find also NVIC settings in the peripheral configuration tabs when you enable interrupts for specific peripherals.
- To configure NVIC settings for Cortex, go to **Cortex > NVIC** tab.

Scenario A: Configuring Interrupt Priority Grouping

- Priority grouping defines how many bits are allocated to preemption priority vs subpriority.
- Use case: When multiple interrupts with different urgency levels exist.
- Steps:
  1. In STM32CubeMX2, navigate to NVIC settings.
  2. Set the priority grouping (for example, 4-bit preemption priority, 0 subpriority).
  3. This affects how priorities are interpreted and nested.

Scenario B: Enabling interrupt for a peripheral

- Example: Enable USART interrupt for receiving data.
- Steps:
  1. Enable the peripheral (e.g., USART1) in STM32CubeMX2.
  2. In the **Configuration** tab, enable the interrupt (e.g., USART1 global interrupt).

3. STM32CubeMX2 automatically adds the NVIC interrupt global configuration for this peripheral.
4. Optionally, configure the interrupt priority in the NVIC settings.

Scenario C: Order Interrupts

- Example: Assign higher priority to an I2C interrupt than to a UART interrupt.
- Steps:
  1. Enable interrupts for both peripherals.
  2. In NVIC settings, set the Preemption Priority and Subpriority for each interrupt.
  3. Lower numerical value means higher priority (e.g., priority 0 is higher than 1).
  4. Ensure priority grouping supports your priority scheme.

Scenario D: Disabling an Interrupt

- Use case: Temporarily disable an interrupt without disabling the peripheral.
- Steps:
  1. In STM32CubeMX2, uncheck the interrupt enable box from the peripheral config tab.
  2. This disables the NVIC interrupt but keeps the peripheral functional.

Scenario E: Reset Interrupts

- After modifying interrupts configuration, you can cancel all changes using the **Reset all interrupts** button.

6.9.2 Important Tips

- Interrupt priorities: Lower number = higher priority.
- Priority grouping: Determines the split between preemption priority and subpriority.
- Avoid priority conflicts: Make sure critical interrupts have higher priority.
- When exceptions and peripherals IRQ handlers codes are not generated (case of NVIC system exceptions and 'IRQ handler generation' disabled in peripheral configuration panel), these handlers must be implemented in user application files.

6.9.3 Additional articles

- How to set NVIC priority with STM32CubeMX2

6.10 How is errors and conflicts propagation managed ?

This section describes how **STM32CubeMX2** detects configuration errors and conflicts, how these issues are reported in the user interface, and how users can identify and resolve them efficiently.

The error and conflict system uses a **top-down approach**:

- Start from a **global indicator** that shows the project contains issues.
- Drill down step by step through panels and views.
- Reach the **exact parameter, pin, or channel** causing the error or warning.

6.10.1 Objective

The STM32CubeMX2 error and conflict propagation mechanism is designed to help users:

- Detect configuration errors and conflicts.
- Identify **where** issues occur in the project.
- Resolve issues **efficiently** without manually checking every configuration.

The following key questions are addressed:

- How can users determine if a project has warnings or errors?
- How can users verify that the pinout has no conflicts?
- How can users check that the clock configuration is valid?
- Where can user view **DMA**, **EXTI**, or **NVIC** conflicts?
- Is it safe to generate code with the current configuration?

> **Note**
>
> In **STM32CubeMX2**, users do not need to search for issues manually. **Visual indicators** guide users from the global project status down to the detailed configuration level.

6.10.2 Design principles

STM32CubeMX2 follows the principle:

> **Start from a global indicator, then drill down step-by-step until reaching the exact failing field.**

The design relies on three main concepts:

- **Global notification:** The user is notified at the top level when an error or warning exists.
- **Easy navigation to root cause:** Indicators at each level guide the user to the source of the issue.
- **Clear and contextual messages:** Messages explain what is wrong and why it is wrong.

Global notification

Global notifications indicate that the project contains at least one error or warning.

The following areas display global notifications:

- The **left navigation bar**, which is always visible.
- The **top tabs**, corresponding to the currently opened workspaces.

Easy navigation to root cause

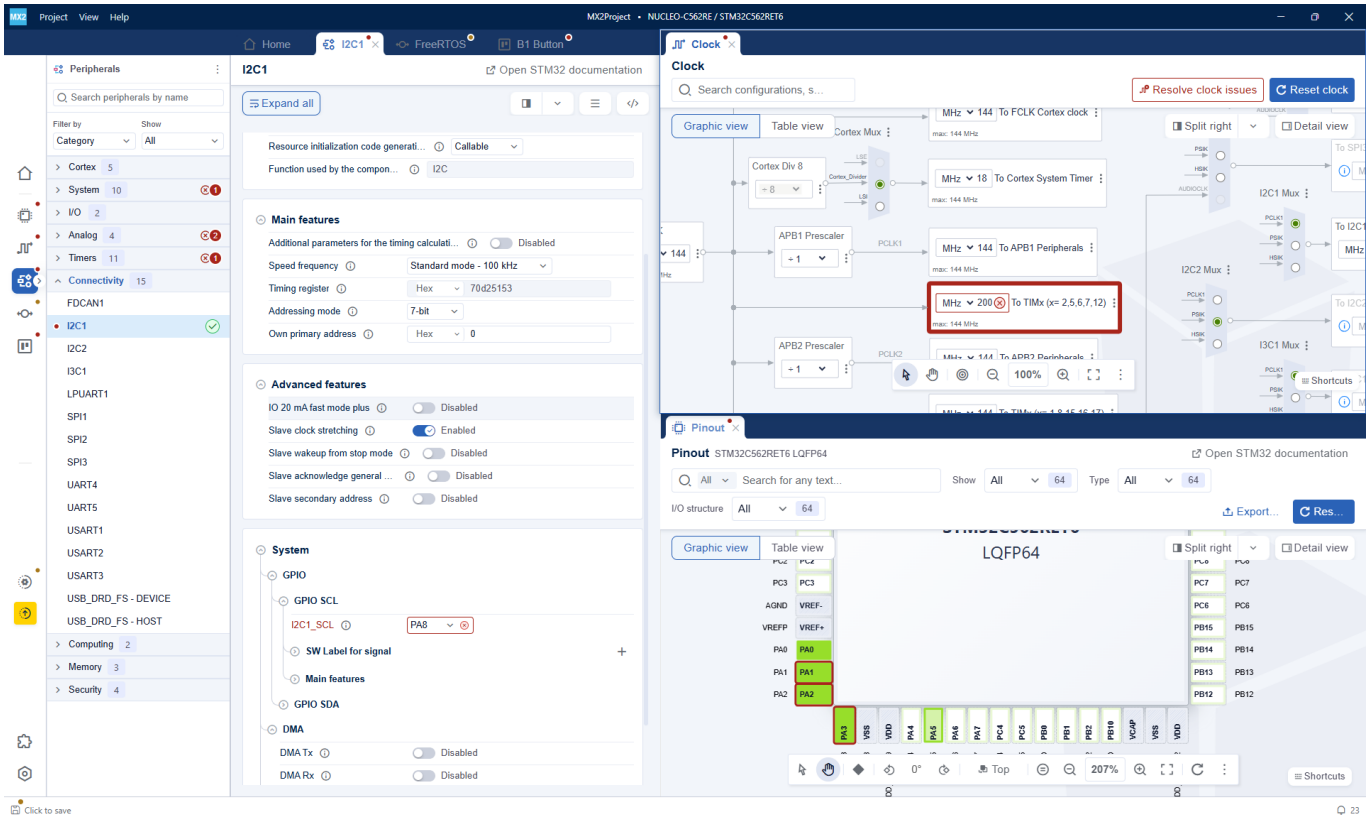
To help users find the root cause of a problem, STM32CubeMX2 combines:

- **Generic indications** on view icons and tab names.
- **Localized indications** inside cards and sections in configuration tabs, tables and channels (for example, **DMA**, **EXTI**, **NVIC**) and individual parameters and pins.

Clear and contextual messages

Error and warning messages are designed to be:

- **Clear:** They describe the issue in simple terms.
- **Contextual:** They explain why the issue occurs, based on the current configuration.



Typical situations include:

- Local parameter errors (for example, value out of range or invalid combination).
- Inconsistency between parameters inside the same configuration screen.
- Inconsistency between different configuration screens or system parameters.

> **Note**
>
> Error counters are displayed in several places in the user interface to help track how many issues remain to be resolved in a given area of the project.

6.10.3 Error propagation overview

The error and conflict propagation mechanism is organized into four levels:

- **Global:** Overall project status, visible in the main navigation area and in the currently opened workspaces.
- **Panel:** Status of a panel, indicating which area of the project still contains issues.
- **Detailed:** Status inside a specific configuration screen, pointing to the affected configuration or parameter.
- **Final gate:** Overall configuration check performed at code generation time.

6.10.4 Global indicators

Left navigation bar

The **left bar** provides a project-wide health overview and is always visible. See **(1)** in *Global indicators in STM32CubeMX2: left navigation bar and workspace tabs*

Each icon (for example, **Pinout**, **Clock**, **Peripherals**, **Middleware**, **Parts**, and **Project Settings**) can display:

- A **red dot**: At least one error exists in the corresponding view.

- An **orange dot**: At least one warning exists in the corresponding view.

Top tabs

The **top tabs** show the status of each open workspace see **(2)** in *Global indicators in STM32CubeMX2: left navigation bar and workspace tabs* (for example, **Pinout**, **Clock**, **Project Settings**, **FreeRTOS**, **DMA**, **NVIC**, **EXTI**, IP configuration tabs).

Each tab can display a red or orange dot:

- The presence of a dot indicates that at least one **error** or **warning** remains in that workspace.

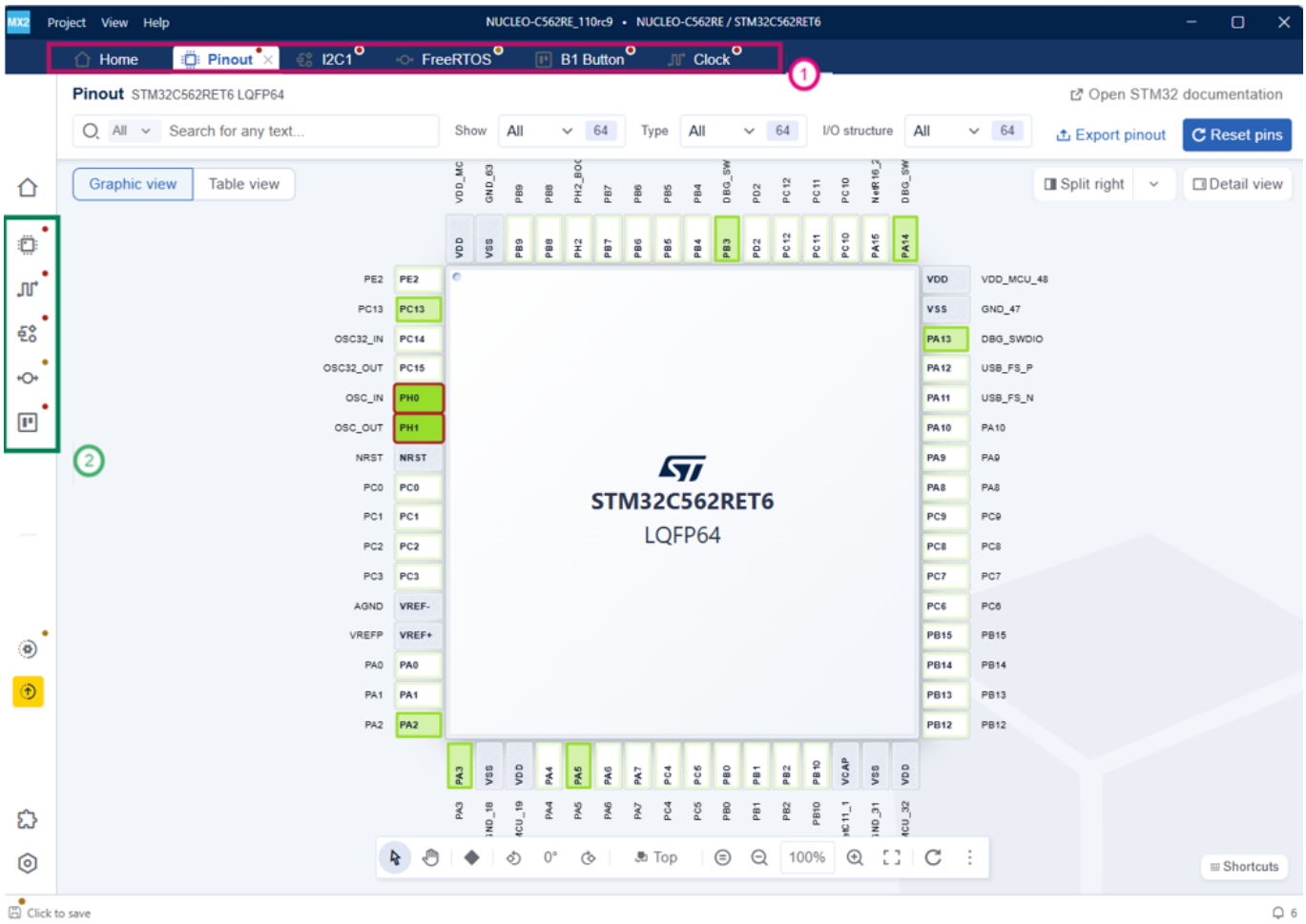


Fig. 2: Global indicators in STM32CubeMX2: left navigation bar and workspace tabs

6.10.5 Panel-level indicators

Peripherals, middleware, and parts panels

In the **Peripherals**, **Middleware**, and **Parts** panels, error indicators are shown at two levels.

- **IP / Middleware / Part level: see (1) in** *Panel-level indicators: category badges and IP-level error icons*
  - A **red error icon or badge** is displayed when the corresponding IP has configuration issues.
  - This tells the user which **peripheral or component** should be opened next to investigate the error.
- **Category level: see (2) in** *Panel-level indicators: category badges and IP-level error icons*
  - A **red badge with a number** indicates how many IPs in this category contain errors.
  - This allows users to quickly identify which **subsystem** concentrates the issues.

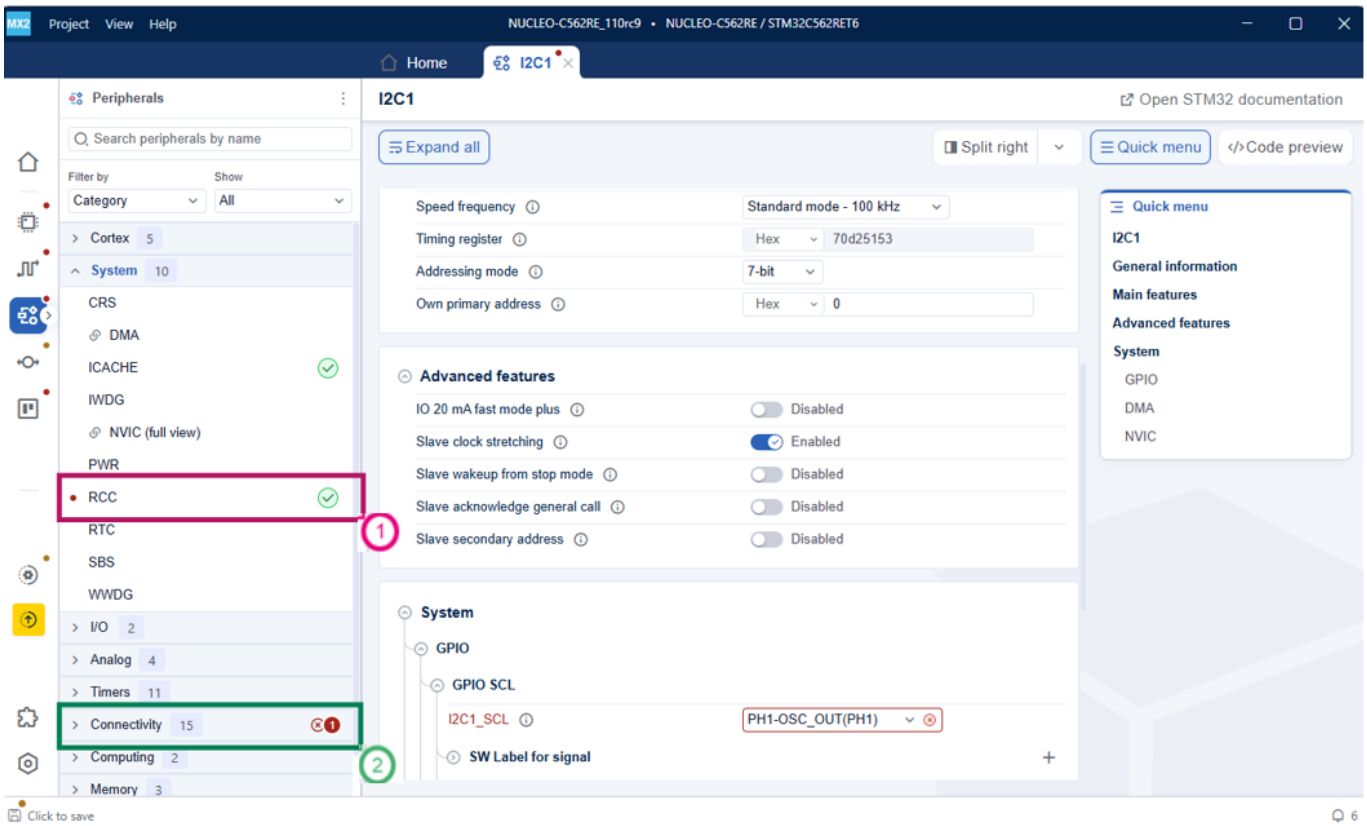


Fig. 3: Panel-level indicators: category badges and IP-level error icons

6.10.6 Detailed indicators

Clock and pinout overview

In complex projects, multiple areas can report issues at the same time. For example, clock configuration, peripheral configuration, and pinout conflicts. STM32CubeMX2 displays these issues in a unified workspace.

In this view:

- The **Peripherals** panel shows which peripherals and categories contain errors or warnings.
- The **Clock** view highlights clock configuration problems and provides tools to resolve them.
- The **Pinout** view indicates pins in conflict or incorrectly assigned.

Linked views for the same issue

A single configuration problem is **propagated across all related views** so that users can identify it from different entry points.

Examples include:

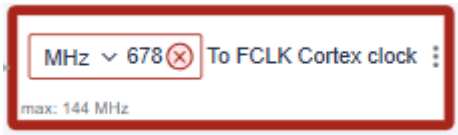
- **System / GPIO conflicts:**
  - Visible in the **IP configuration** (GPIO section).
  - Visible in the **Pinout view**, where the affected pin is highlighted with a red border.
- **DMA channel conflicts:**
  - Visible in the **peripheral configuration** (DMA section).
  - Visible in the **DMA view**, where the corresponding channel row is marked as a conflict.
- **EXTI line conflicts:**

  - Visible in the **IP configuration** (EXTI section).
  - Visible in the **EXTI view**, where the corresponding line is flagged in red.

Clock conflict example

When there is a clock configuration issue:

- The clock elements affected by the issue are **framed in red** and marked with an **error icon**.



- A dedicated button indicates the global status of the clock configuration:
  - **Resolve clock issues:** At least one clock error is present in the tree.

  - **No clock issues:** All clock issues have been resolved and no clock element is bordered in red.

Users have two options to correct these issues:

- Click **Resolve clock issues** to allow STM32CubeMX2 automatically propose valid clock settings.
- Locate the **red elements** in the clock tree and adjust the configuration manually.

Pinout conflict example

In the **Pinout view**:

- Pins that have issues are **highlighted** directly on the package.
- Users can hover over the pin or open the **detail view** to see which signals are in conflict.

In the **System → GPIO** configuration:

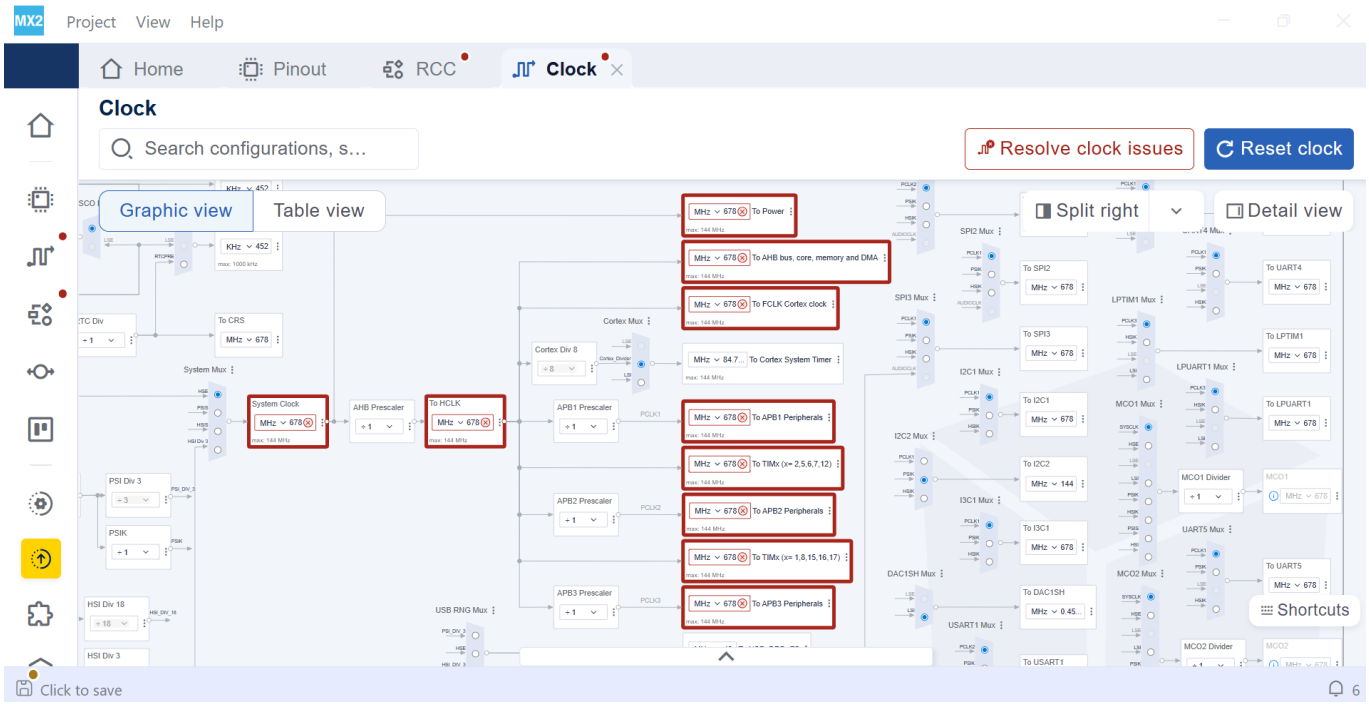
- **The pin selector uses icons to represent the pin status:**
  - No icon: The pin is free (no signal mapped).
  - `x` icon: Indicates a conflict on that pin.
  - **`!` icon: Indicates that a signal is already mapped on that pin, but:**
    - This signal can be moved, and
    - The newly selected signal can be mapped to this pin.

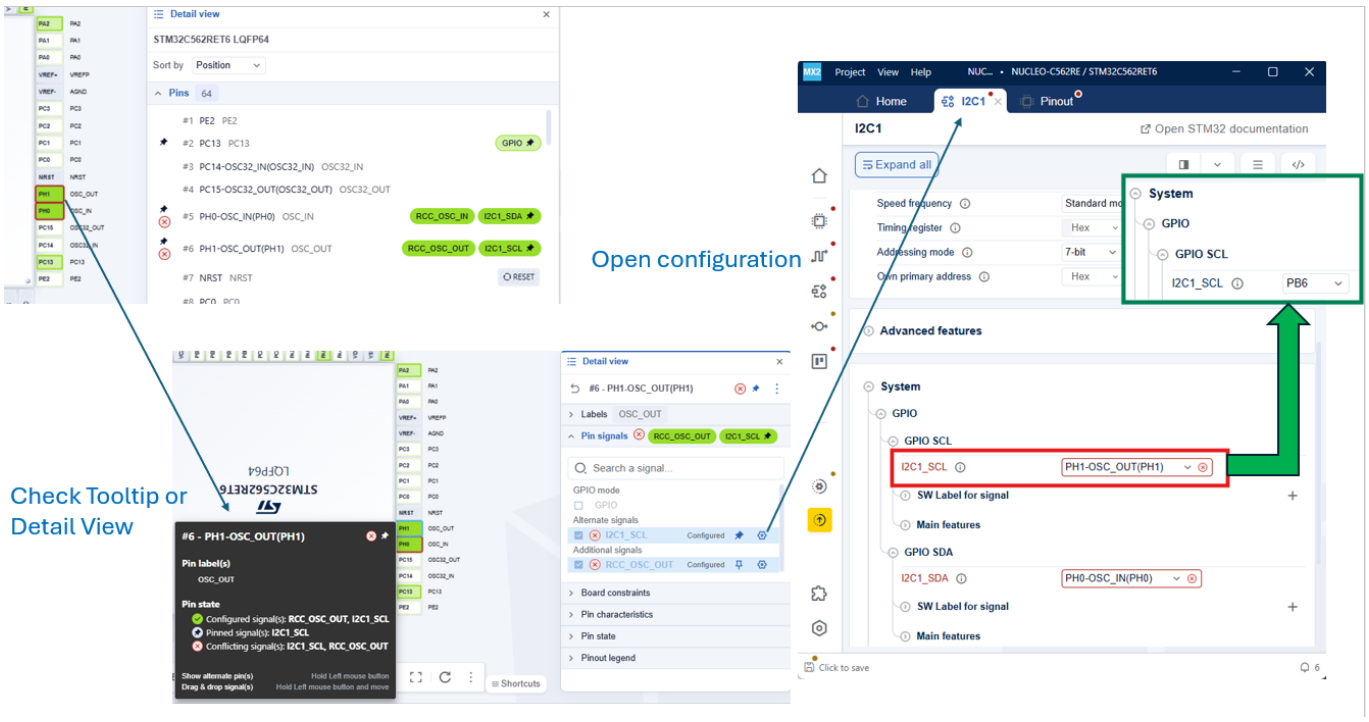
To resolve the conflict, users must choose the appropriate pin and adjust the configuration.

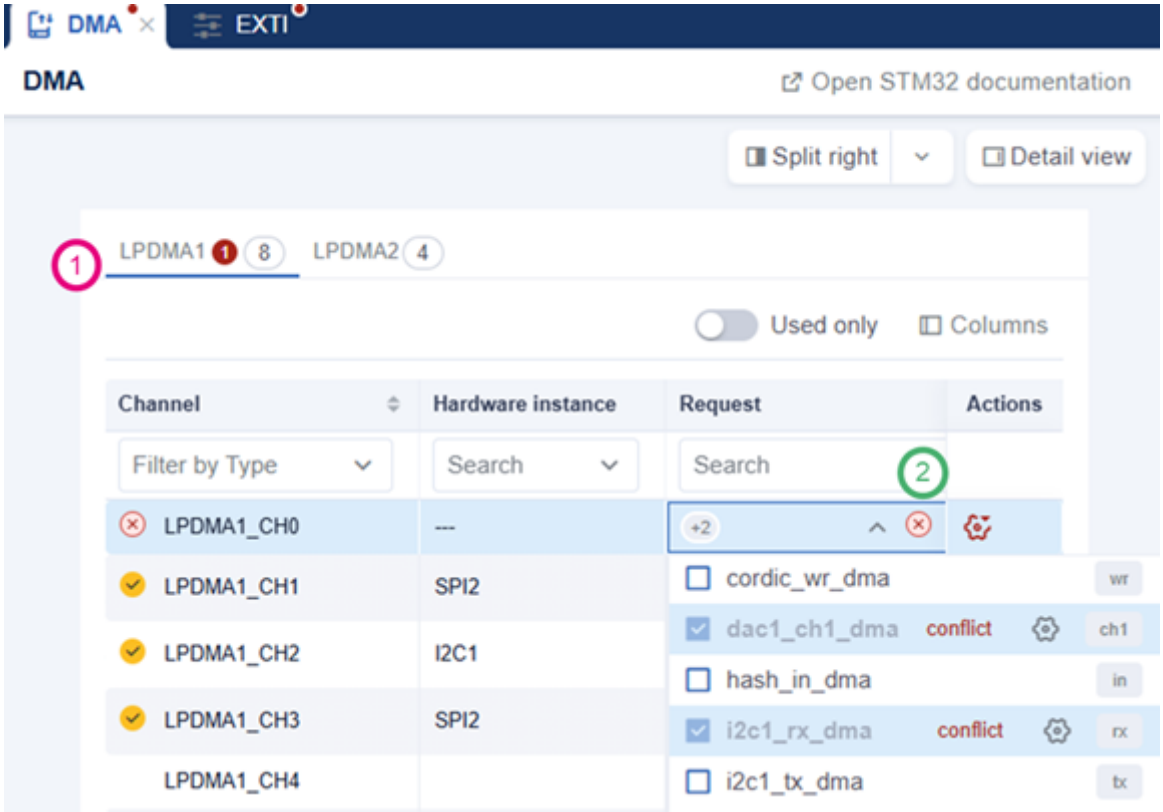
DMA conflict example

In the **DMA view**:

- The **instance tab** shows a **red badge** containing the number of errors. See **(1)** in HowTo_Err_Fig17
- **The channel row in conflict is highlighted. See (2) in HowTo_Err_Fig17:**
  - A red error icon.
  - The number of conflicting requests.


MX2
Project
View
Help
Home
Pinout
RCC
Clock
Clock
Search configurations, s...
Resolve clock issues
Reset clock
Graphic view
Table view
Split right
Detail view
Shortcuts
Click to save


Open configuration
Check Tooltip or
Detail View
Detail view
STM32C562RET6 LQFP64
Home
I2C1
Pinout
Open STM32 documentation
Expand all
System
GPIO
GPIO SCL
I2C1_SCL
PB6
Advanced features
PH1-OSC_OUT(PH1)
SW Label for signal
Main features
GPIO SDA
PH0-OSC_IN(PH0)
Click to save



In the **System → DMA** configuration:

- **The DMA channel selector uses icons to show the channel status:**
  - No icon: The channel is free.
  - `x` icon: Indicates a conflict on that channel.

EXTI conflict example

In the **EXTI view**:

- Any EXTI line in conflict is **marked in red**. see **(1)** in HowTo_Err_Fig12

In the **GPIO configuration**:

- The same issue is visible in the **GPIO / EXTI settings** of each pin.
- The EXTI line associated with the conflict is highlighted as an error.

To solve the conflict, the user can **disable EXTI on one of the pins** sharing the same EXTI line.

6.10.7 Configuration check at code generation

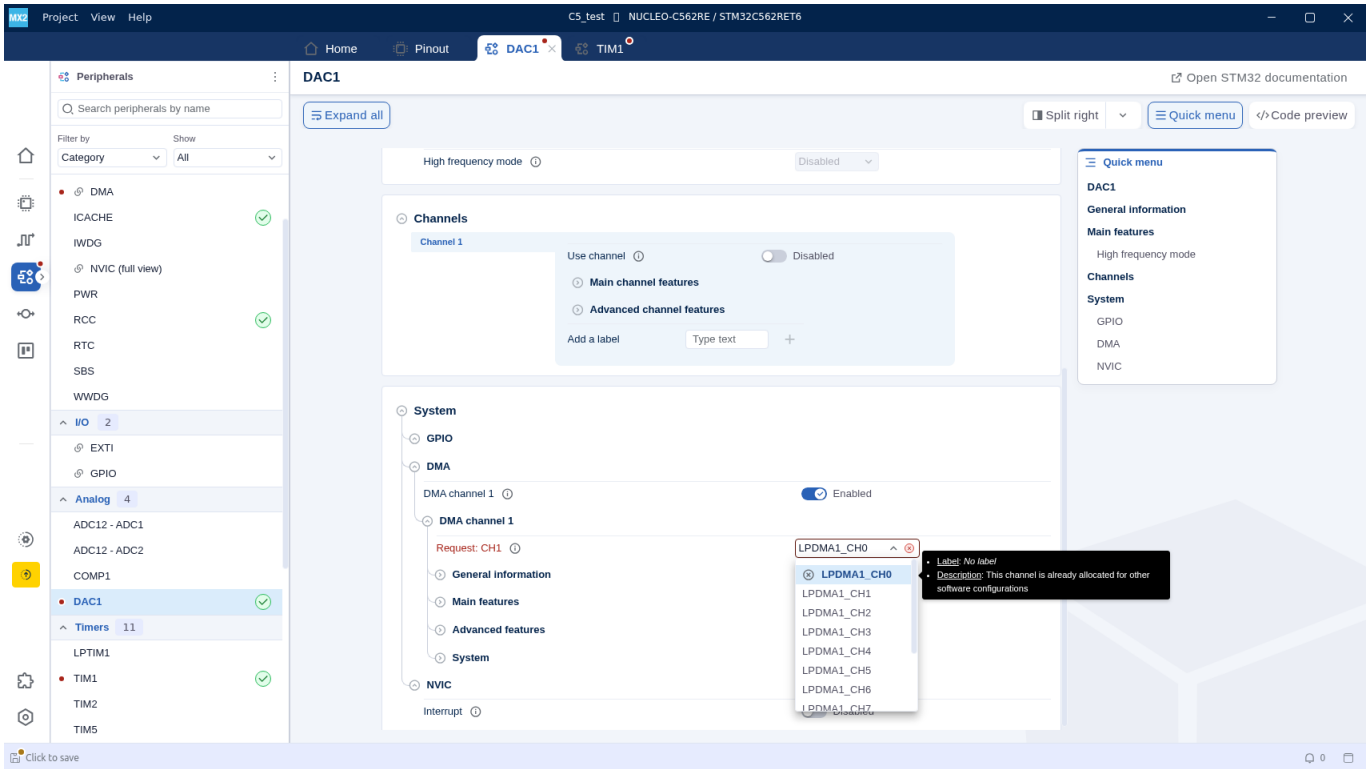
When code generation starts, STM32CubeMX2 performs a **global configuration check** on the entire project.

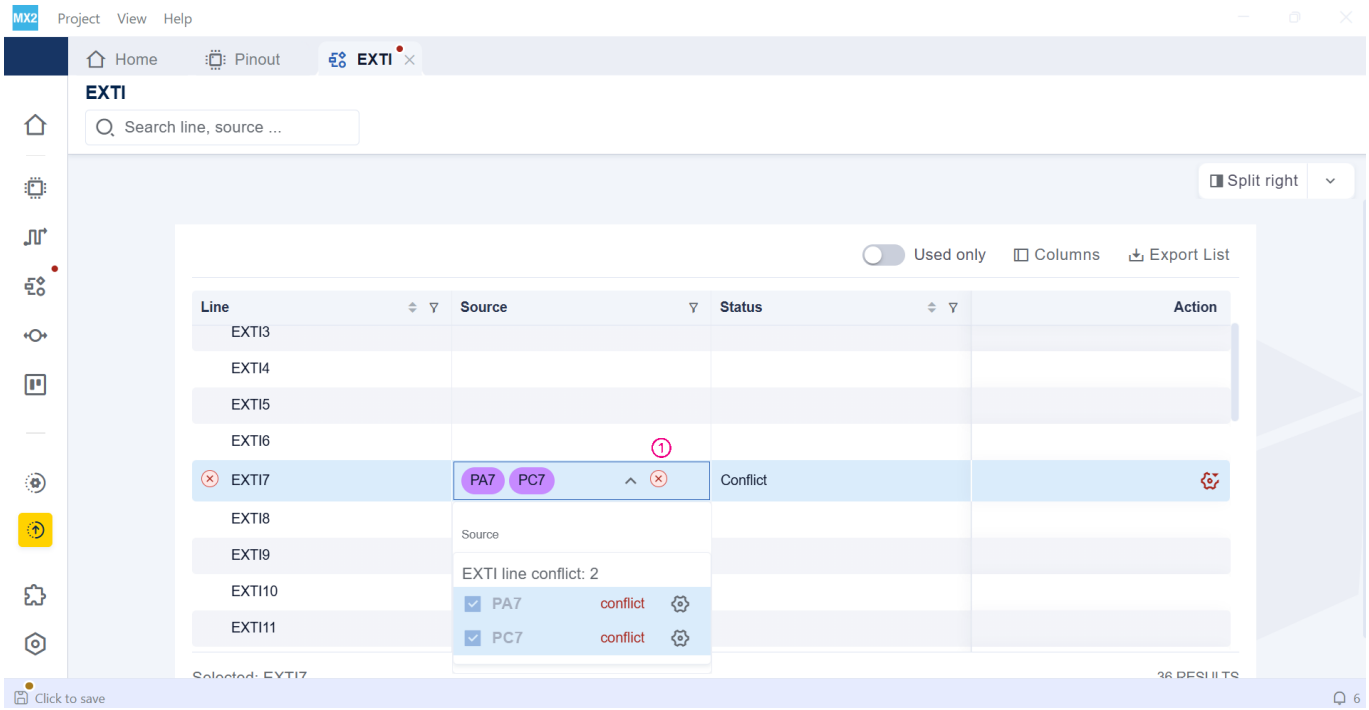
If any error or warning remains, a popup is displayed:

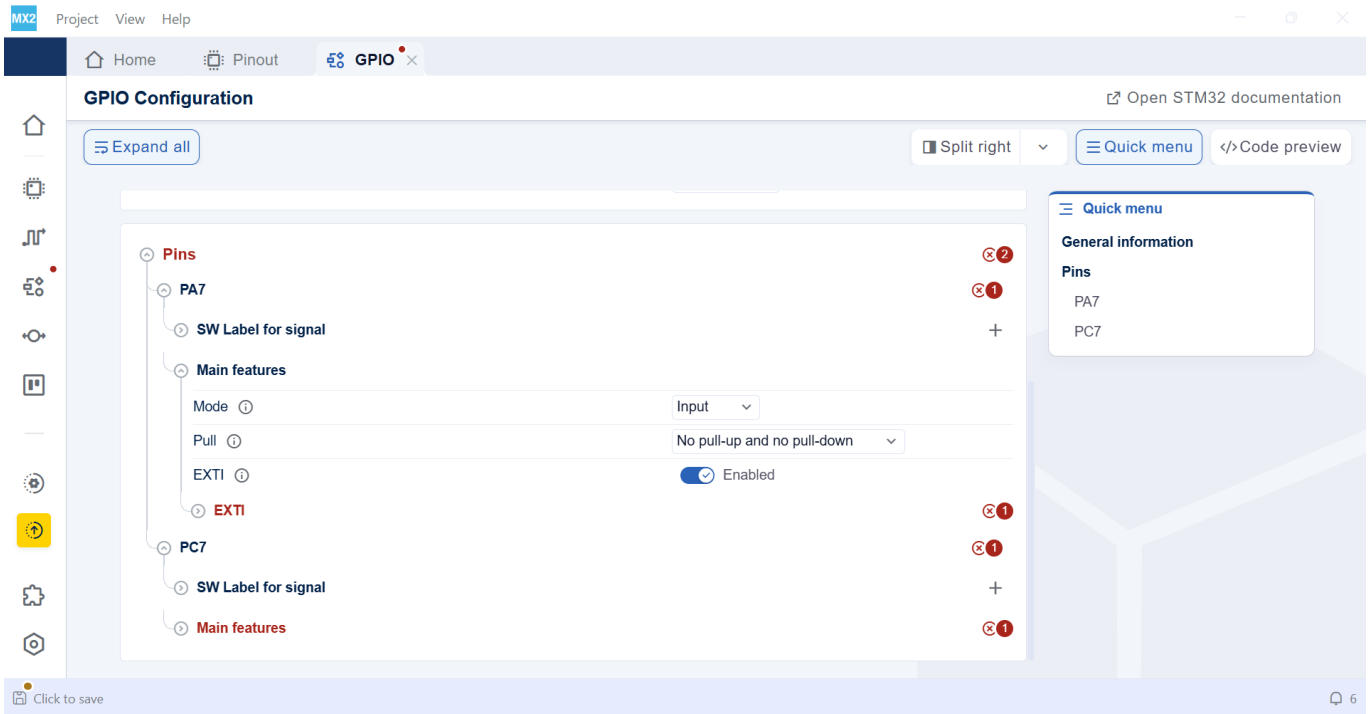
Impact of remaining errors and warnings

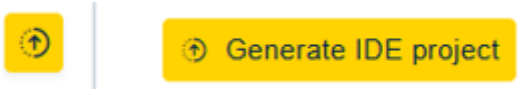
If configurations are generated while errors or warnings are still present, the following issues may occur:

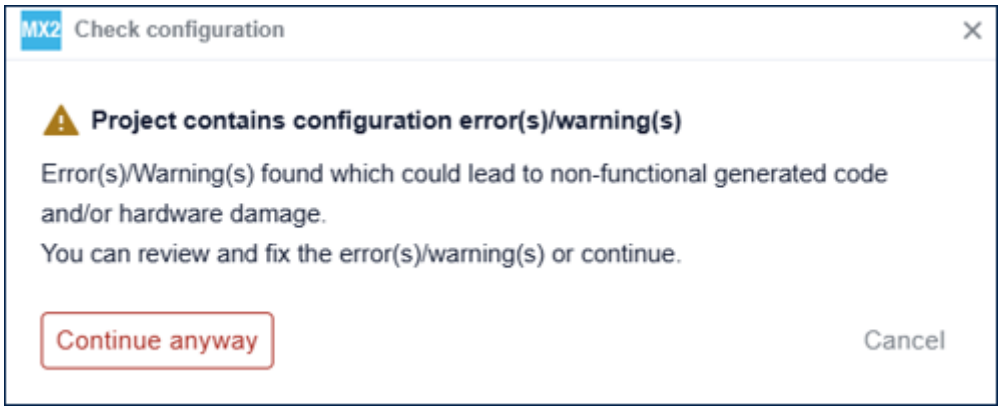
- Nonfunctional generated code.
- Potential **hardware damage** in extreme cases, depending on the configuration.


Project View Help
C5_test NUCLEO-C562RE / STM32C562RET6
Home
Pinout
DAC1
TIM1
Peripherals
Search peripherals by name
Filter by
Category
Show
All
DMA
ICACHE
IWDG
NVIC (full view)
PWR
RCC
RTC
SBS
WWDG
I/O 2
EXTI
GPIO
Analog 4
ADC12 - ADC1
ADC12 - ADC2
COMP1
DAC1
Timers 11
LPTIM1
TIM1
TIM2
TIM5
Open STM32 documentation
Expand all
Split right
Quick menu
Code preview
High frequency mode
Disabled
Channels
Channel 1
Use channel
Disabled
Main channel features
Advanced channel features
Add a label
Type text
System
GPIO
DMA
DMA channel 1
Enabled
DMA channel 1
Request: CH1
LPDMA1_CH0
General information
Main features
Advanced features
System
NVIC
Interrupt
LPDMA1_CH0
LPDMA1_CH1
LPDMA1_CH2
LPDMA1_CH3
LPDMA1_CH4
LPDMA1_CH5
LPDMA1_CH6
Label: No label
Description: This channel is already allocated for other software configurations
Quick menu
DAC1
General information
Main features
High frequency mode
Channels
System
GPIO
DMA
NVIC
Click to save


Project View Help
Home
Pinout
EXTI
EXTI
Search line, source ...
Split right
Used only
Columns
Export List
Line
Source
Status
Action
EXTI3
EXTI4
EXTI5
EXTI6
EXTI7
PA7
PC7
Conflict
EXTI8
EXTI9
EXTI10
EXTI11
Source
EXTI line conflict: 2
PA7 conflict
PC7 conflict
Click to save


Project View Help
Home
Pinout
GPIO
GPIO Configuration
Open STM32 documentation
Expand all
Split right
Quick menu
Code preview
Pins
PA7
SW Label for signal
Main features
Mode
Input
Pull
No pull-up and no pull-down
EXTI
Enabled
EXTI
PC7
SW Label for signal
Main features
Quick menu
General information
Pins
PA7
PC7
Click to save


Generate IDE project


Check configuration
Project contains configuration error(s)/warning(s)
Error(s)/Warning(s) found which could lead to non-functional generated code and/or hardware damage.
You can review and fix the error(s)/warning(s) or continue.
Continue anyway
Cancel

User options

When the popup is displayed, users can:

- **Review and fix** the reported issues, then start code generation again.
- Click **Continue anyway** to proceed with code generation **at their own risk**.

7 Release Note

7.1 Release details

7.1.1 Introduction

This release note is updated periodically to reflect the latest changes, issues, and limitations of **STM32CubeMX2**. For the most recent release summary, see the table below

STM32CubeMX2 version	
1.1.2	
1.1.1	
1.1.0	

- Failure to uninstall STM32CubeMX 1.0.0, especially when the user path contains a space Diff collisions are handled poorly when exporting a project through the command-line interface (CLI) User interface (UI) elements resized incorrectly and overlapped when the window was resized Scrolling the mouse wheel while editing the frequency in the clock tree caused the editing window to overlap the clock tree components Resolved the issue where users were unexpectedly required to use administrator rights for installation or uninstallation, although administrator mode is not supported

Other major fixed issues:

- I²C configuration showed no error when a node was unavailable
- Errors were not visible in the Peripherals tab when interrupt errors occurred
- Mounted parts were not visible when creating a project and then another project on the same MCU

1.0.1 Patch release

- Enables users to install multiple versions of STM32CubeMX2 side by side on the same PC.
- Adds a graphical uninstaller for easier removal.
- Prevents certain user files from being overwritten during Project Export.
- Fixes the “Pack Update” feature for new packs with minor changes, which previously did not work correctly.

1.0.0 Initial release

7.1.2 Customer support

For more information or assistance regarding **STM32CubeMX2,** contact the nearest **STMicroelectronics** sales office or use the STM32CubeMX2 Community.

7.1.3 General information

Overview

STM32CubeMX2 is provided as a standalone tool. To install the tool, see *How to start - Installation*.

Licensing

STM32CubeMX2 is delivered under the SLA0048 software license agreement and its Additional License Terms.

Notice

- The behavior of CMake projects has changed:
  - The current configuration now includes the "." directory when necessary to ensure that the project builds
  - **STM32CubeMX2** now requires at least CMake 3.30 versionto generate projects

7.2 Release limitations

7.2.1 STM32 series configuration and code generation limitations

STM32 series
STM32C5

7.2.2 Generic STM32CubeMX2 limitations

Display

- **STM32CubeMX2** is optimized for 15 inches / 1920*1080 display. Using lower resolution may lead to incomplete display on some panels. Use Zoom Out to adapt tool to your display.

Installer

- Depending on your PC configuration, the first screen of the installer may take some time to appear.
- You cannot install **STM32CubeMX2** 1.0.0 version if **STM32CubeMX2** 1.0.1 (or upper) version is already installed.
- We recommend installing a new version of **STM32CubeMX2** in a separate directory, rather than using one that already contains an installed version on your machine.
- Installation as an administrator is not supported. Do NOT install **STM32CubeMX2** with administrator access on Windows by right-clicking and selecting Run as administrator.
- Users must install **libwebkit2gtk-4.1-0** before launching the installer for **Ubuntu 22.04.**
- **STM32CubeMX2** cannot currently be launched on Debian systems using Wayland
  - As a workaround, set the following environment variable before launching the app : ELECTRON_OZONE_PLATFORM_HINT=x11. This forces Electron to use XWayland instead of native Wayland.
  - An alternative fallback is to launch the application with: –disable-gpu

Uninstaller

- Do not move the uninstaller from the original installation folder. Moving the uninstaller prevents it from running properly and can cause unintended deletion of the installation folder and its contents.
- Uninstalling **STM32CubeMX2** previous version after installing **STM32CubeMX2** 1.1.0 version will uninstall both versions. Please uninstall **STM32CubeMX2** previous version version before installing **STM32CubeMX2** 1.1.0 version.
- Recent Windows policy updates can prevent the uninstaller from running correctly. If issues occur, follow these steps:
  - Manually delete the folder `C:\Users\user_name\AppData\Local\STMicroelectronics\STM32CubeMX2\<Version_of_MX2>`
  - Open the control panel
  - You should get **STM32CubeMX2** unknown icon
  - Double click on the **STM32CubeMX2** unknown icon and confirm by yes

- After that, **STM32CubeMX2** is fully removed from the PC

Documentation

- **STM32CubeMX2** does not include documentation. Connect to the internet to access **STM32CubeMX2** online documentation.

Project manager

- Project name must not contain any of the following characters: / : * ? “ < > | + . $ ¤ £ µ % ! § , ; & ~ ‘ ( ) [ ] ‘ @ { } = ^ and space and must not be composed only of numbers.
- Project creation fails when the project directory is set to the root of the ‘C:\’

Packs management

- Pack versions cannot be selected during project creation, latest packs are used by default.

  While waiting for enhancements to **STM32CubeMX2**, a script is available to create **STM32CubeMX2** projects using the STM32C5 v2.0.0 packs instead of the latest published pack version.

Project generation

- During project regeneration, **STM32CubeMX2** does not automatically merge IDE project files, such as OpenCMSIS csolution, EWARM .ewp, and CMake files. **STM32CubeMX2** detects if files have been modified outside the tool since the last generation. The user can choose to overwrite, skip, or create a backup of the existing files.
- EWARM: Only version 9.20 is supported.
- The main.c and main.h files cannot be relocated.
- **STM32CubeMX2** cannot generate projects for the ST Arm® Clang toolchain (Clang/LLVM).
- Project generation can take time and cannot be stopped. Wait for the generation to complete.
- **STM32CubeMX2** does not prevent configuration changes while project is being generated. Avoid changing project configuration during project generation.
- When generating a project with the OpenCMSIS format, some file filtering is not properly applied, leading to a longer generation process and a larger project size than expected.
- When generating code for a project started from an IOC2 extracted from a hardware abstraction layer (HAL) pack, the **STM32CubeMX2** Output panel is more verbose than usual. This has no functional impact on the generated project.
- **STM32CubeIDE** version 2.1.1 minimum is mandatory to build CMake projects generated by **STM32CubeMX2**.
- CMakePresets.json and flags.cmake are not updated when the user chooses to add a Release or Debug build target through the command-line interface (CLI).
- The release build target is not available from the UI, but it can be added using the CLI command:
  - cube mx ide-project generate –project-path <projectPath> –build-target <buildTarget> –destination <destination> –format <format> –source <source>
  - BuildTarget format: [projectName][.buildType][+targetType]
  - As example: cube mx ide-project generate -p MyProject.ioc2 –build-target .debug_GCC+STM32C562RET6 .release_GCC+STM32C562RET6 –destination MyIDEProject –format CMake –source include-packs-from-local

Code Preview

- When an IP configuration is updated in Code Preview, the “Copy content to clipboard” action does not reflect the latest changes. It copies only the default generated code. As a workaround, manually select the full content by pressing [Ctrl+A] before copying.

Clock

- The RTCPRE input is not grayed out as expected when HSE is disabled.

8 FAQ

8.1 Frequently asked questions

8.1.1 1 - Why are there no user sections in STM32CubeMX2 generated code ?

- **Separation of concerns** ensures that user application logic and hardware-specific configuration are kept in distinct files.
- This approach makes it easier to port an application to a new STM32 microcontroller or board, as only the hardware-specific files must be adapted. You can generate the updated content of these files for a new device by using **STM32CubeMX2**.

The hardware configuration code generated by STM32CubeMX2 leverages **modularity and encapsulation**.

- Each software unit, such as peripheral configuration, is implemented in its own module (mx_[pppi].c and mx_[pppi].h).
- **Encapsulation** allows you to update or replace hardware configuration without affecting application logic.

The benefits of this approach include:

- Maintainability and reusability:
  - **Generated code** can be safely regenerated by STM32CubeMX2 without overwriting user logic.
  - **User code** remains untouched, making updates and maintenance straightforward by leveraging HAL2.
  - **Middleware, utilities, and part drivers** are series-agnostic and reusable across projects.
- Clear porting path
  - When porting to a new microcontroller or board, **only update the /generated/ folder** and related configuration files.
  - **There is no need to modify application logic or infrastructure code**.

By keeping user code and generated code separate, the application remains modular, maintainable, and easy to port to new hardware. Only the hardware-specific configuration must be adapted, which is simplified by STM32CubeMX2 code generation. The core logic remains reusable and protected from accidental overwrites.

STMicroelectronics uses this layered, modular approach in **STM32Cube software examples** to simplify porting and maintenance, as described in the STM32Cube Software Examples - Porting Example Architecture.

On top of this, **STM32CubeMX2** proposes a mechanism of diff/skip/overwrite into the project export settings to be able to handle potential changes done by end-user in some exported files. Refer to the *User manual — IDE Project Generation* for detailed information.

8.1.2 2 - How do I solve DMA conflicts in STM32CubeMX2 ?

For detailed information about solving DMA conflicts refer to *How to configure DMA - View and Resolve Allocation Conflicts*.

8.1.3 3 - How do I solve pinout conflicts in STM32CubeMX2?

For detailed information about solving pinout conflicts refer to *User manual - Pinout - Pinout Conflicts*.

8.1.4 4 - Why do I receive a warning when activating FreeRTOS™ ?

For detailed information about FREERTOS™ refer to the FREERTOS™ User Guide.

8.1.5 5 - Can I use CMSIS-Packs?

STM32CubeMX2 only works with packs that contain STM32Cube-specific metadata.

While other CMSIS-Packs can be installed, they are not visible in the configuration views and are not used for code generation.

Installing such packs has no effect inside **STM32CubeMX2**.

8.1.6 6 - How do I solve issues that may occur during installation ?

Here you will find issues that may occur during the installation of **STM32CubeMX2** and how to resolve them.

	Installation issue
	Installer fails immediately at startup with “Installation failed”
	Installation stops while extracting bundles or packs
	Packs are not available after installation
	Required packs are missing when creating or opening a project
	STM32CubeMX2 CLI is not installed or CLI commands are not recognized
	No desktop or Start menu shortcut is created
	Pack repository is not reachable or packs cannot be downloaded
	STM32CubeMX2 does not appear in system uninstall tools
	CLI commands are not available in a terminal after installation
	Double clicking *.ioc2* files does not open **STM32CubeMX2**

9 Community

9.1 Support link

If you need assistance or have questions regarding STM32CubeMX2, support is available. Access a wide range of resources and receive help from the STMicroelectronics support team.

For more information and to get support, visit the STMicroelectronics Support Home.

9.2 Community link

Feedback is valuable and helps improve STM32CubeMX2. If you have suggestions, issues, or comments, share them with the community.

To provide feedback and participate in discussions, visit the STM32CubeMX2 Community.

9.3 Knowledge articles link

To find information on STM32 microcontroller programming, troubleshooting, ecosystem, and product details, visit the STM32 MCUs Knowledge Base.