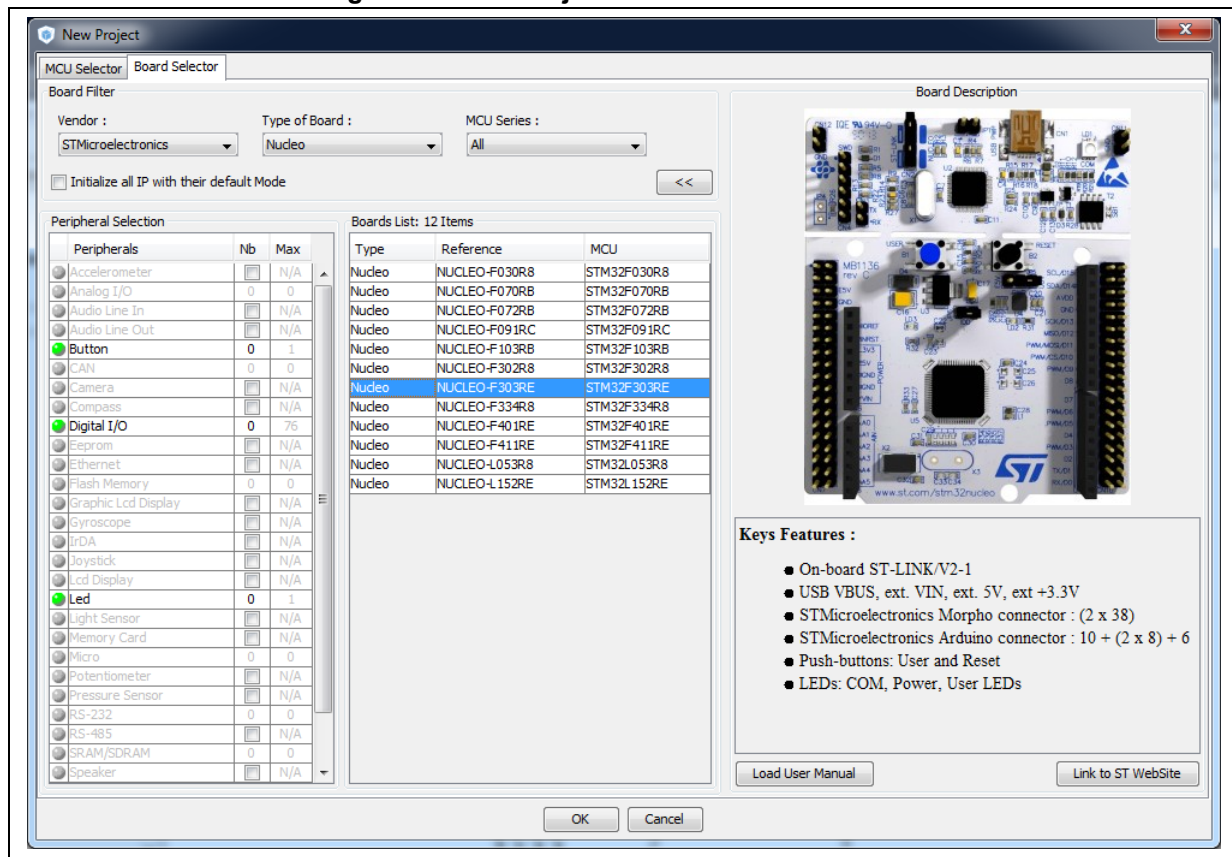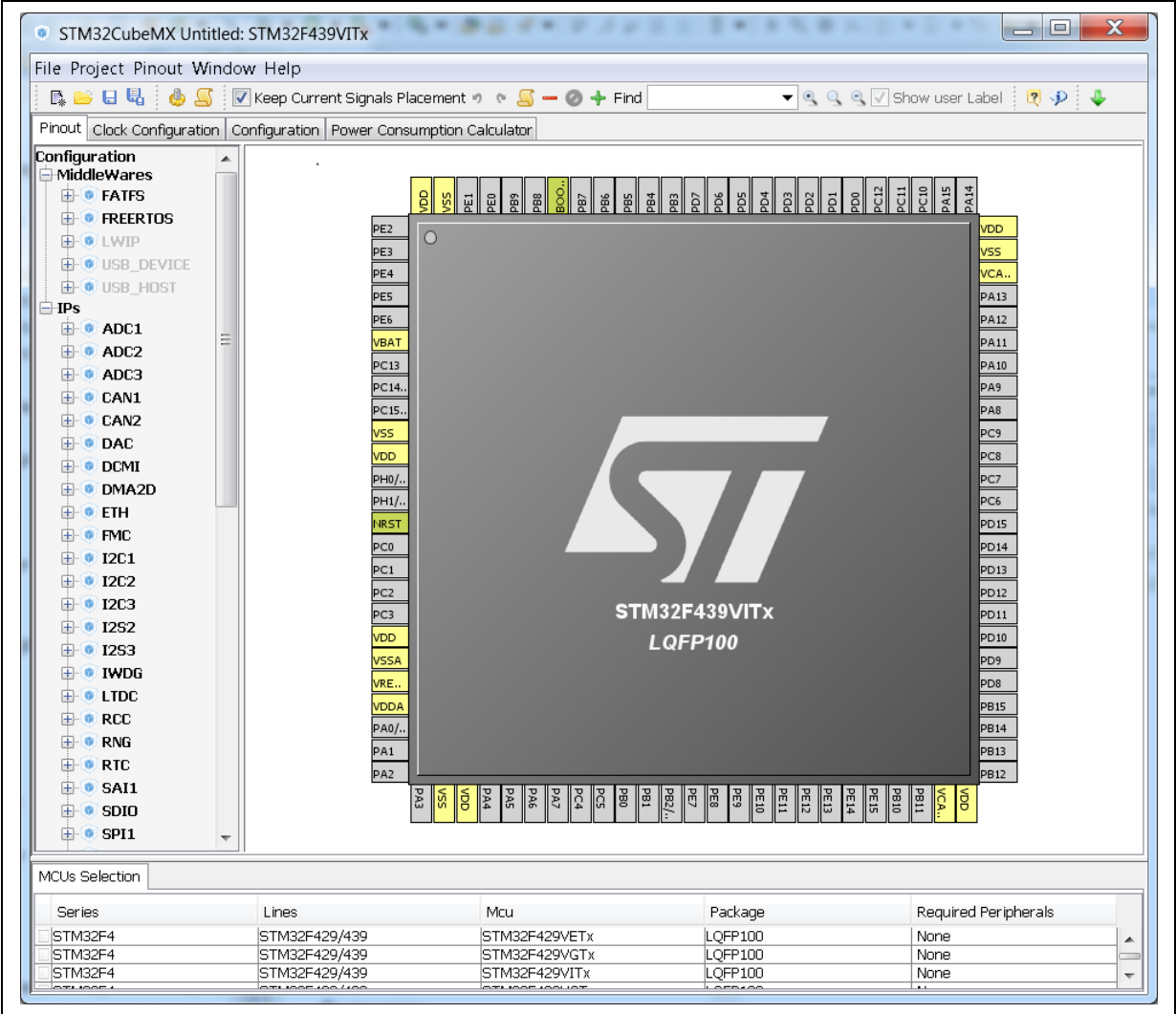**Figure 22. New Project window - board selector**

4.3 Main window

Once an STM32 part number or a board has been selected or a previously saved project has been loaded, the main window displays all STM32CubeMX components and menus (see *Figure 23*). Refer to *Section 4.3* for a detailed description of the toolbar and menus.

**Figure 23. STM32CubeMX Main window upon MCU selection**

Selecting a board while keeping the peripheral default modes option unchecked, automatically sets the pinout for this board. However, only the pins set as GPIOs are marked as configured, i.e. highlighted in green, while no peripheral mode is set. The user can then manually select from the peripheral tree the peripheral modes required for his application (see *Figure 24*).

**Figure 24. STM32CubeMX Main window upon board selection
(Peripheral default option unchecked)**
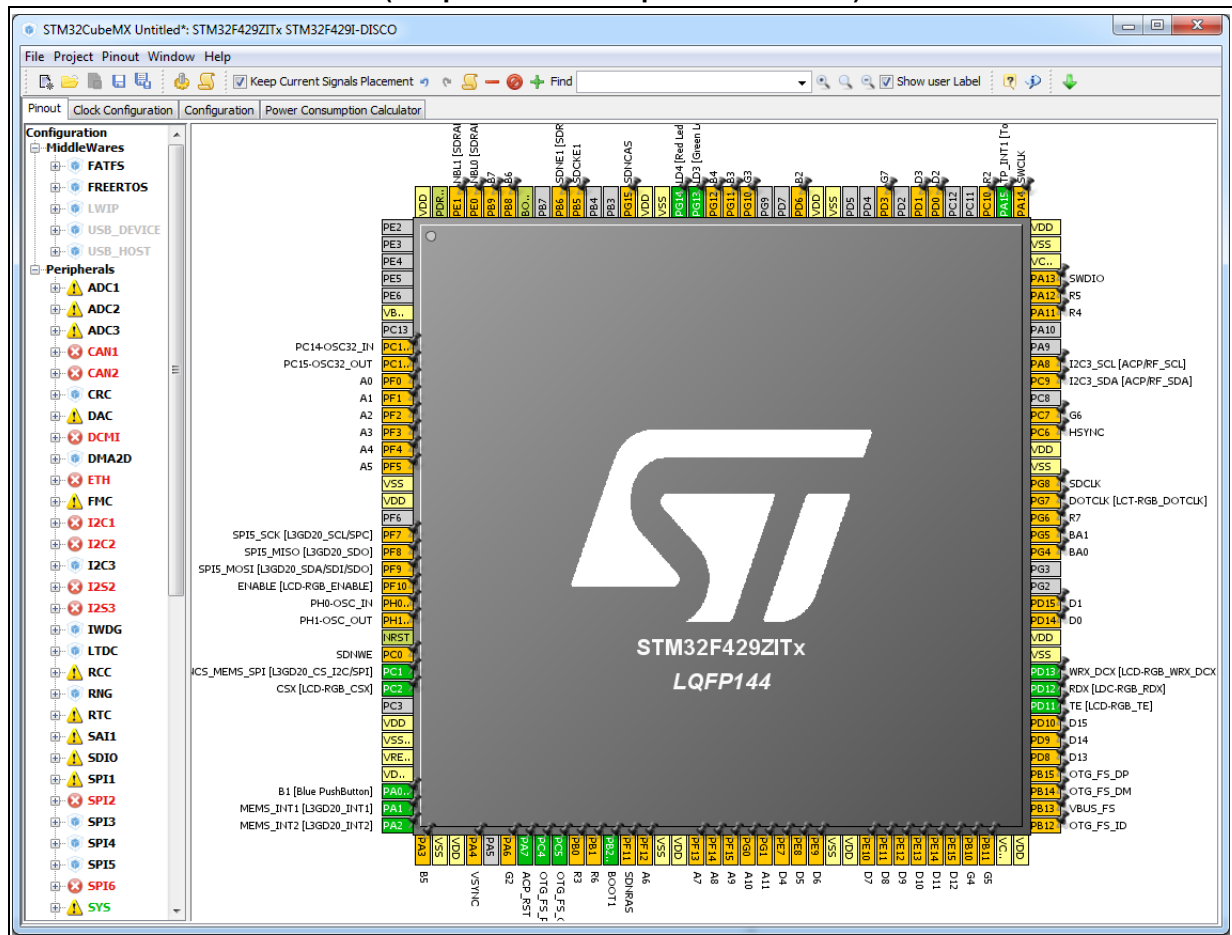
Selecting a board with the peripheral default modes option checked, automatically sets both the pinout and the default modes for the peripherals available on the board. This means that STM32CubeMX will generate the C initialization code for all the peripherals available on the board and not only for those relevant to the user application (see *Figure 25*).

Figure 25. STM32CubeMX Main window upon board selection
(Peripheral default option checked)
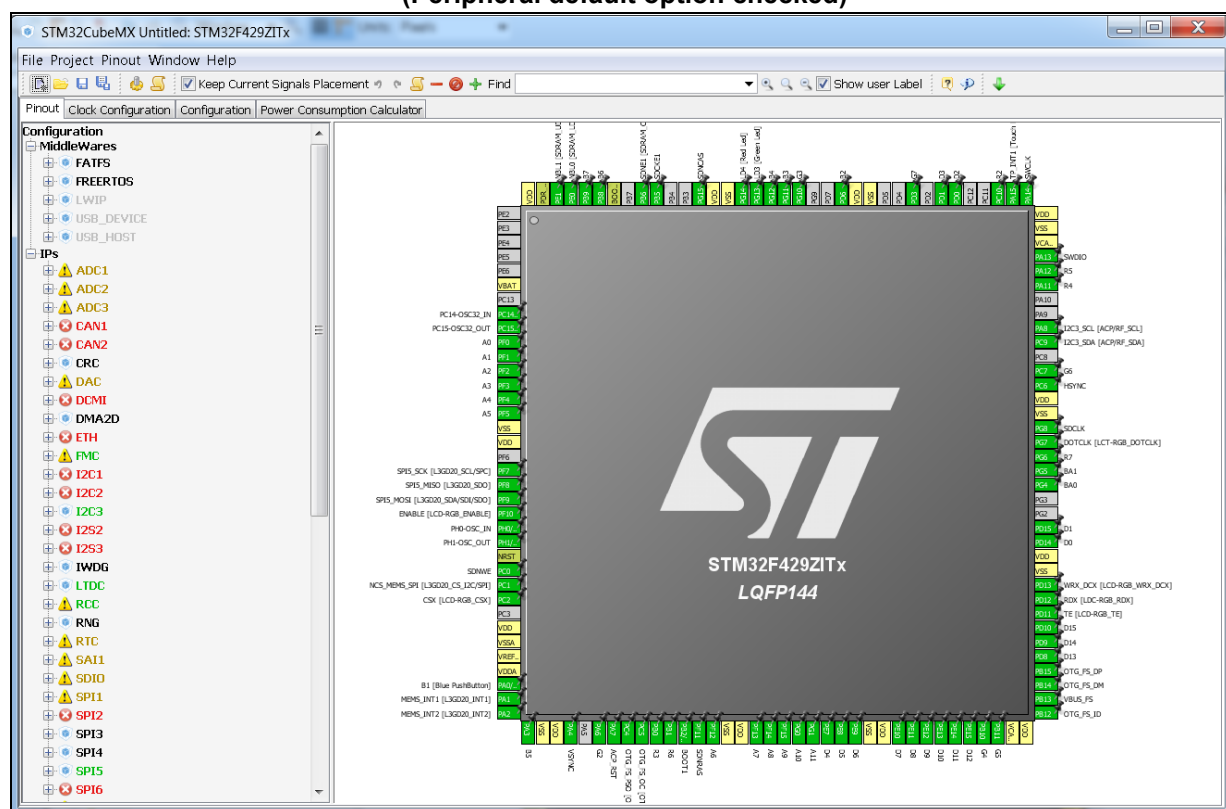
4.4 Toolbar and menus

The following menus are available from STM32CubeMX menu bar:

- **File** menu
- **Project** menu
- **Pinout** menu (displayed only when the **Pinout** view has been selected)
- **Window** menu
- **Help** menu

STM32CubeMX menus and toolbars are described in the sections below.

4.4.1 File menu

Refer to *Table 3* for a description of the **File** menu and icons.

Table 3. File menu functions

Icon	Name	Description
	New Project	Opens a new project window showing all supported MCUs and well as a set of STMicroelectronics boards to choose from
	Load Project …	Loads an existing STM32CubeMX project configuration by selecting an STM32CubeMX configuration .ioc file. **Caution:** **When upgrading to a new version of STM32CubeMX, make sure to always backup your projects before loading the new project (especially when the project includes user code).**
	Import Project …	Opens a new window to select the configuration file to be imported as well as the import settings. The import is possible only if you start from an empty MCU configuration. Otherwise, the menu is disabled. A status window displays the warnings or errors detected when checking for import conflicts. The user can then decide to cancel the import.
	Save Project as …	Saves current project configuration (pinout, clock tree, IP, PCC) as a new project. This action creates an .ioc file with user defined name and located in the destination folder
	Save Project	Saves current project
No icon	Close Project	Closes current project and switch back to the welcome page
No icon	Recent Projects >	Displays the list of five most recently saved projects
No icon	Exit	Proposes to save the project if needed then close the application

4.4.2 Project menu

Refer to *Table 4* for a description of the **Project** menu and icons.

Table 4. Project menu

Icon	Name	Description
	Generate Code	This menu generates C initialization C code for current configuration (pinout, clocks, peripherals and middleware). Opens a window for project settings if they have not been defined previously. *Note: It is recommended to backup the current projects when upgrading to a new version of STM32CubeMX. The user will be prompted to migrate to a new firmware library version if any is available. Select "Continue" to keep the previously used version.*
	Generate Report[1]	This menu generates current project configuration as a pdf file and a text file.
	Settings	This menu opens the project settings window to configure project name, folder, select a toolchain and C code generation options

1. If the project was previously saved, the reports are generated at the same location as the project configuration .ioc file. Otherwise, the user can choose the destination folder, and whether to save the project configuration as an .ioc file or not.

4.4.3 Pinout menu

The **Pinout** menu and sub-menus shortcuts are available only when the **Pinout** tab is selected (see *Figure 26*). They are hidden otherwise (see *Figure 27*). Refer to *Table 5* for a description of the **Pinout** menu and icons.
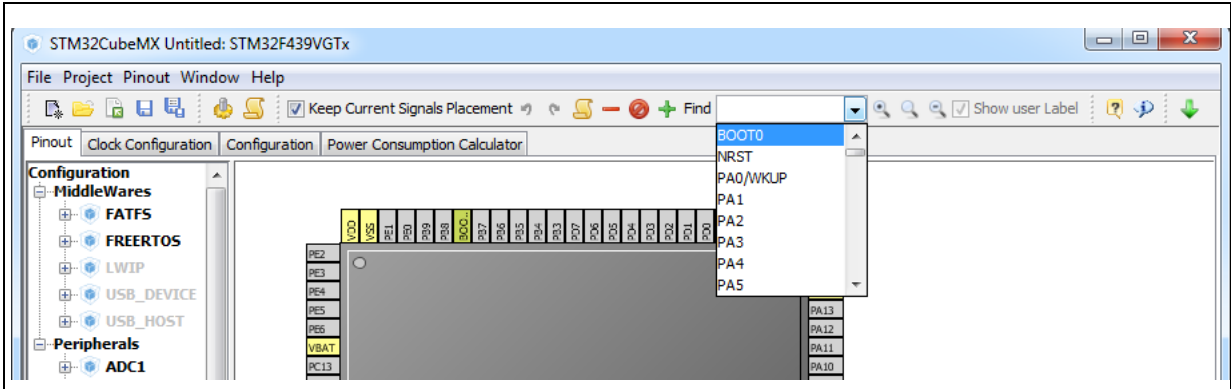
Figure 26. Pinout menus (Pinout tab selected)
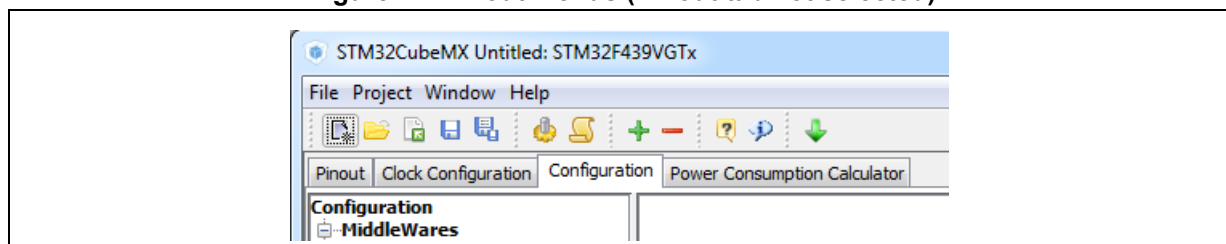
Figure 27. Pinout menus (Pinout tab not selected)

Table 5. Pinout menu
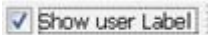
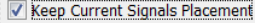| Icon	Name	Description
	Undo	Undoes last configuration steps (one by one)
	Redo	Redoes steps that have been undone (one by one)
No icon	Pins/Signals Options	Opens a window showing the list of all the configured pins together with the name of the signal on the pin and a Label field allowing the user to specify a label name for each pin of the list. For this menu to be active, at least one pin must have been configured. Click the pin icon to pin/unpin signals individually. Select multiple rows then right click to open contextual menu and select action to pin or unpin all selected signals at once. Click column header names to sort alphabetically by name or according to placement on MCU.
Find ⯆	Pinout search field	Allows the user to search for a pin name, signal name or signal label in the **Pinout** view. When it is found, the pin or set of pins that matches the search criteria blinks on the **Chip** view. Click the **Chip** view to stop blinking.
☑ Show user Label	Show user labels	Allows showing on the **Chip** view, the user-defined labels instead of the names of the signals assigned to the pins.
No icon	Clear Pinouts	Clears user pinout configuration in the Pinout window. Note that this action clears from the configuration window the IPs that have an influence on the pinout.
No icon	Clear Single Mapped Signals	Clears signal assignments to pins for signals that have no associated mode (highlighted in orange and not pinned).
No icon	Set unused GPIOs	Opens a window to specify the number of GPIOs to be configure among the total number of GPIO pins that are not used yet. Specify their mode: Input, Output or Analog (recommended configuration to optimize power consumption). **Caution:** Before using this menu, make sure the debug pins (available under SYS peripheral) are set to access microcontroller debug facilities.
No icon	Reset used GPIOs	Opens a window to specify the number of GPIOs to be freed among the total number of GPIO pins that are configured.
	Generate csv text pinout file	Generates pin configuration as a .csv text file

Table 5. Pinout menu (continued)

Icon	Name	Description
—	Collapse All	Collapses the IP / Middleware tree view
⊘	Disable Modes	Resets to "Disabled" all peripherals and middleware modes that have been enabled. The pins configured in these modes (green color) are consequently reset to "Unused" (gray color). IPs and middleware labels change from green to black (when unused) or gray (when not available).
➕	Expand All	Expands the IP/Middleware tree view to display all functional modes.
🔍	Zooming in	Zooms in the chip pinout diagram
🔍	Best Fit	Adjusts the chip pinout diagram to the best fit size
🔍	Zooming out	Zooms out the chip pinout diagram
☑ Keep Current Signals Placement	Keep current signals Placement	Available from toolbar only. Prevents moving pin assignments to match a new IP operating mode. It is recommended to use the new pinning feature that can block each pin assignment individually and leave this checkbox unchecked.

4.4.4 Window menu

The **Window** menu allows to access the **Outputs** function (see *Table 6*).

Table 6. Window menu

Name	Description
Outputs	Opens the MCUs selection window at the bottom of STM32CubeMX Main window. Opens two tabs at the bottom of STM32CubeMX main window: – **MCUs selection** tab that lists the MCUs that match the user criteria selected via the MCU selector. – **Outputs** tab that displays STM32CubeMX messages, warnings and errors encountered upon users actions.

4.4.5 Help menu

Refer to *Table 7* for a description of the **Help** menu and icons.

Table 7. Help menu

Icons	Name	Description
🔖	Help Content	Opens the STM32CubeMX user manual

Table 7. Help menu

	About...	Shows version information
	Check for Updates	Shows the software and firmware release updates available for download.
	Install New Libraries	Shows all STM32CubeMX and firmware releases available for installation. Green check box indicates which ones are already installed on you PC and up-to-date.
	Updater Settings...	Opens the updater settings window to configure manual versus automatic updates, proxy settings for internet connections, repository folder where the downloaded software and firmware releases will be stored.

4.5 Output windows

4.5.1 MCUs selection pane

This window lists all the MCUs of a given family that match the user criteria (series, peripherals, package..) when an MCU was selected last.

Note: *Selecting a different MCU from the list resets the current project configuration and switches to the new MCU. The user will be prompted to confirm this action before proceeding.*

Figure 28. MCU selection menu
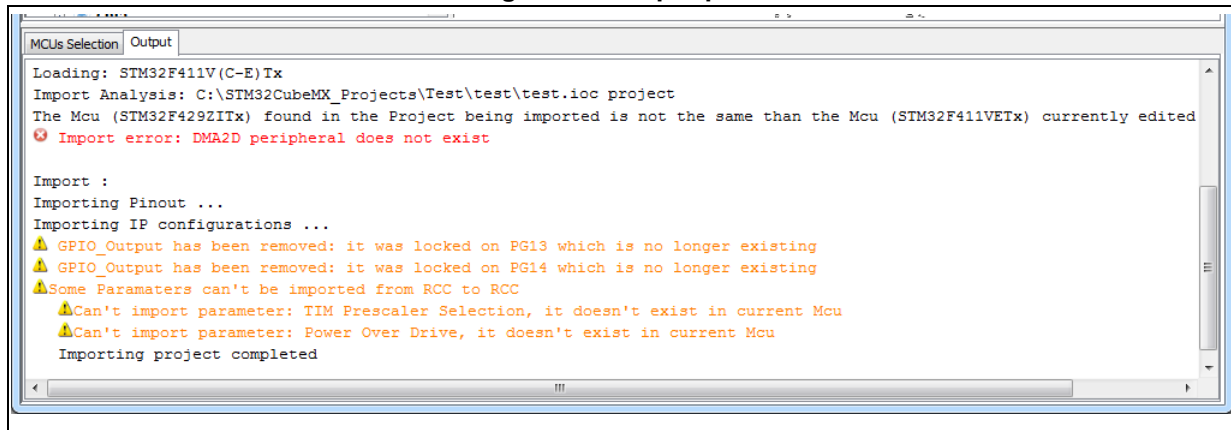

This window can be shown/hidden by selecting/unselecting **Outputs** from the Window menu.

4.5.2 Output pane

This pane displays a non exhaustive list of the actions performed, errors and warnings raised (see *Figure 29*).

Figure 29. Output pane

4.6 Import Project window

The **Import Project** menu eases the porting of a previously-saved configuration to another MCU.

By default the following settings are imported:

- Pinout tab: MCU pins and corresponding peripheral modes.The import fails if the same peripheral instances are not available in the target MCU.
- Clock configuration tab: clock tree parameters.
- Configuration tab: peripherals and middleware libraries initialization parameters.
- Project settings: choice of toolchain and code generation options.

To import a project, proceed as follows:

1. Select the **Import project** icon that appears under the **File** menu after starting a New Project and once an MCU has been selected.

   The menu remains active as long as no user configuration settings are defined for the new project, that is just after the MCU selection. It is disabled as soon as a user action is performed on the project configuration.

2. Select **File** > **Import Project** for the dedicated Import project window to open. This window allows to specify the following options:

   – The STM32CubeMX configuration file (.ioc) pathname of the project to import on top of current empty project.
   – Whether to import the PCC configuration defined in the Power Consumption Calculator tab or not.
   – Whether to import the project settings defined through the Project > Settings menu: IDE selection, code generation options and advanced settings.
   – Whether to import the project settings defined through the **Project** > **Settings** menu: IDE selection and code generation options.
   – Whether to attempt to import the whole configuration (Automatic import) or only a subset (Manual Import).
   a) Automatic project import (see *Figure 30*)

Figure 30. Automatic project import
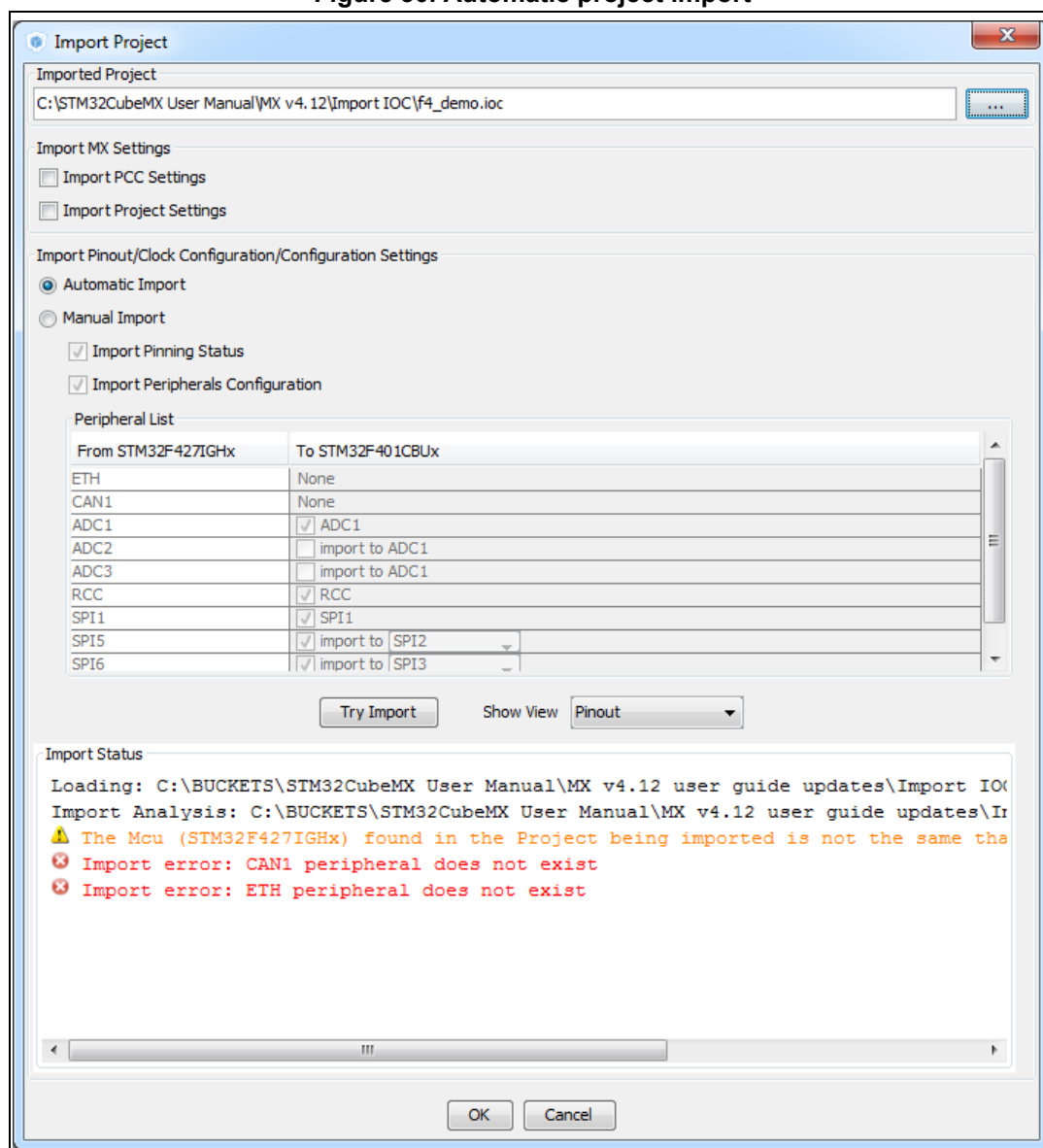
b)   Manual project import

In this case, checkboxes allow to manually select the set of peripherals (see *Figure 31*).

Select the **Try Import** option to attempt importing.

Figure 31. Manual project import
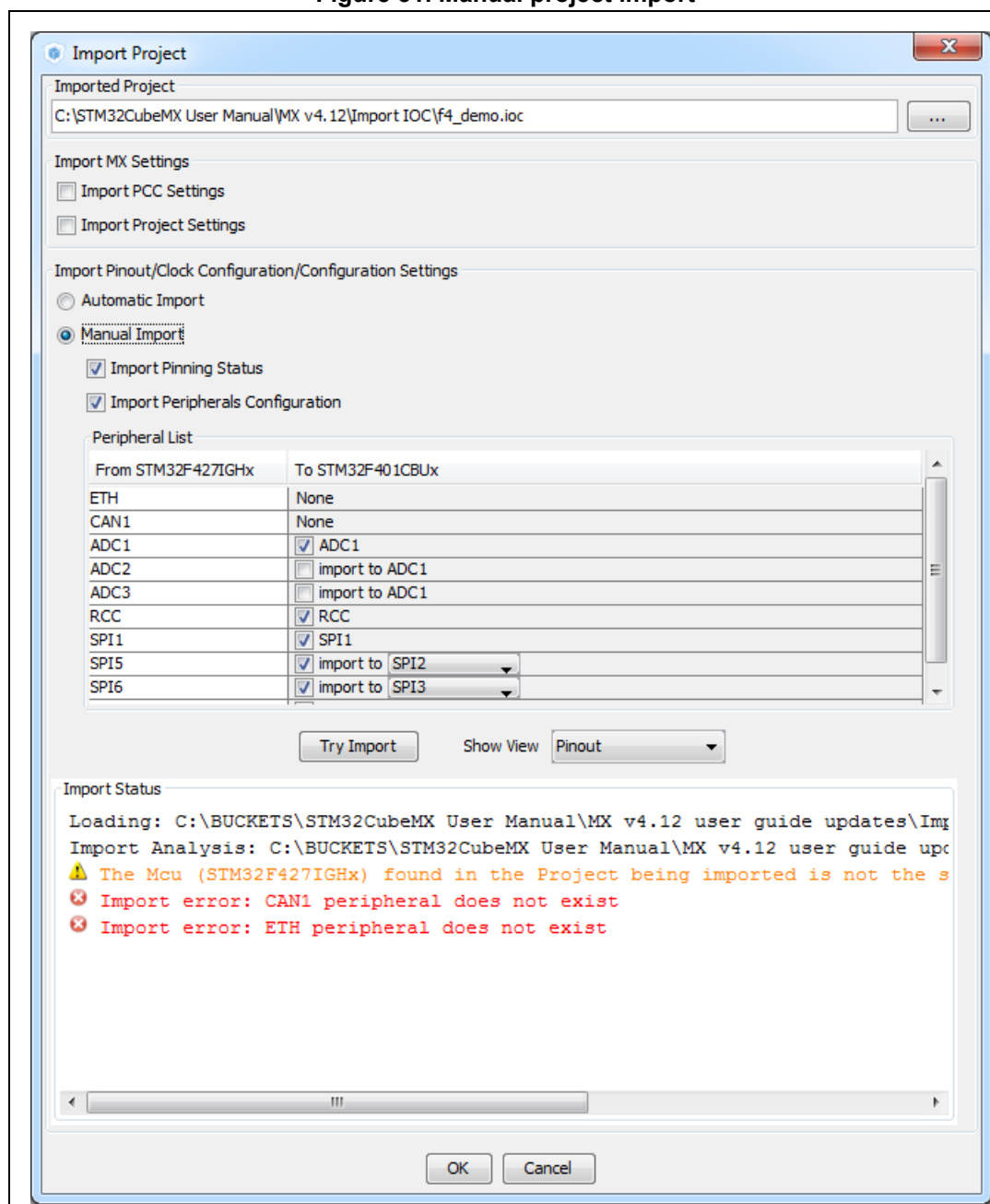
The Peripheral List indicates:

– The peripheral instances configured in the project to be imported

– The peripheral instances, if any exists for the MCU currently selected, to which the configuration has to be imported. If several peripheral instances are candidate for the import, the user needs to choose one.

Conflicts might occur when importing a smaller package with less pins or a lower-end MCU with less peripheral options. Click the **Try Import** button to check for such

conflicts: the Import Status window and the Peripheral list get refreshed to indicate errors, warnings and whether the import has been successful or not:

– Warning icons indicate that the user has selected a peripheral instance more than once and that one of the import requests will not be performed. *Figure 32* shows an example where the ADC1 instance has been selected twice.

– A cross sign indicates that there is a pinout conflict and that the configuration can not be imported as such. In *Figure 32*, the SPI6 instance configuration can not be imported on SPI3 because it conflicts with the previously selected SPI1 configuration.

The manual import can be used to refine import choices and resolve the issues raised by the import trial. *Figure 33* shows how to complete the import successfully, that is, in this case, by unselecting the request for ADC2 and SPI1 imports.

The **Show View** function allows switching between the different configuration tabs (pinout, clock tree, peripheral configuration) for checking influence of the "Try Import" action before actual deployment on current project (see *Figure 33*).

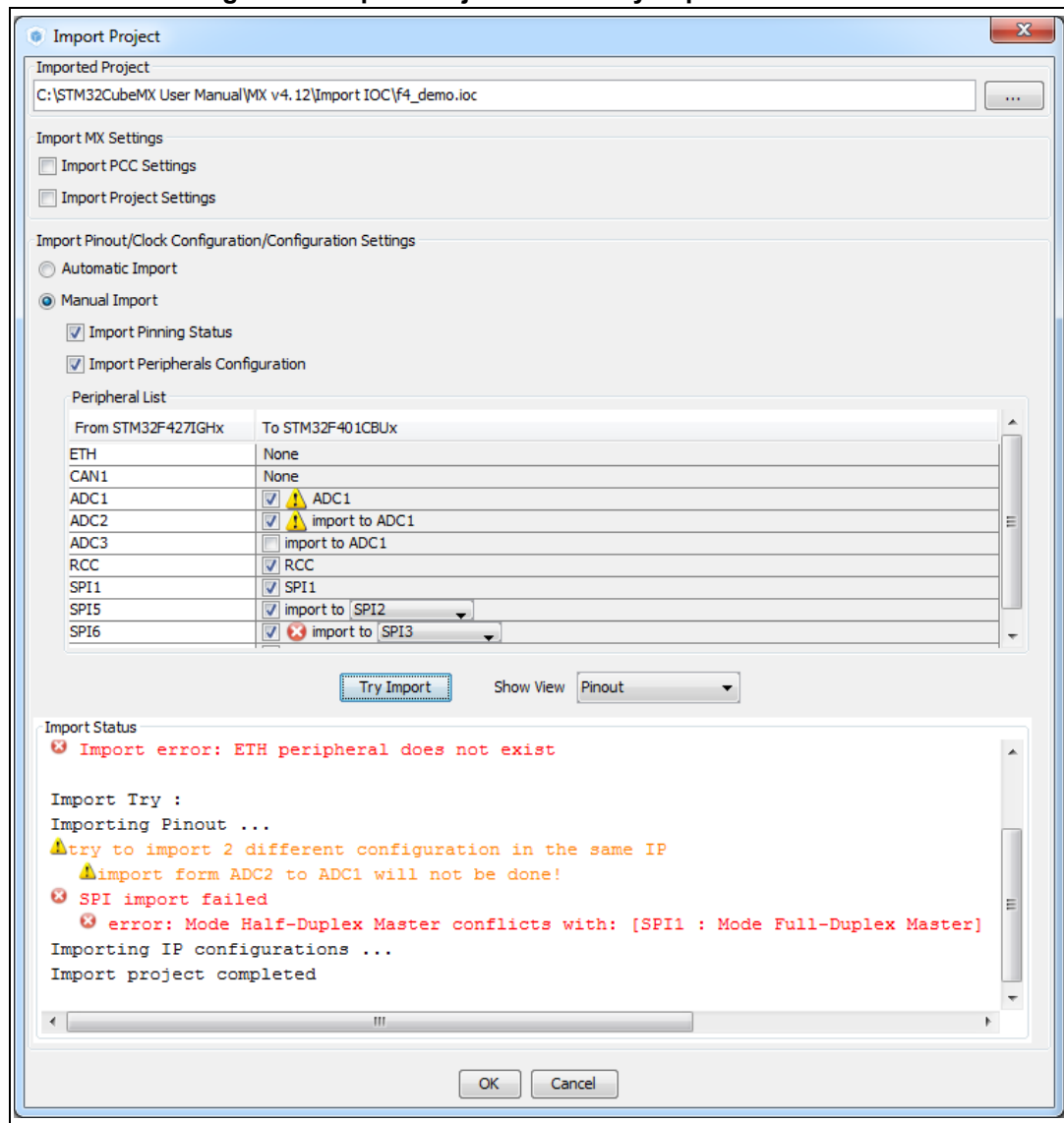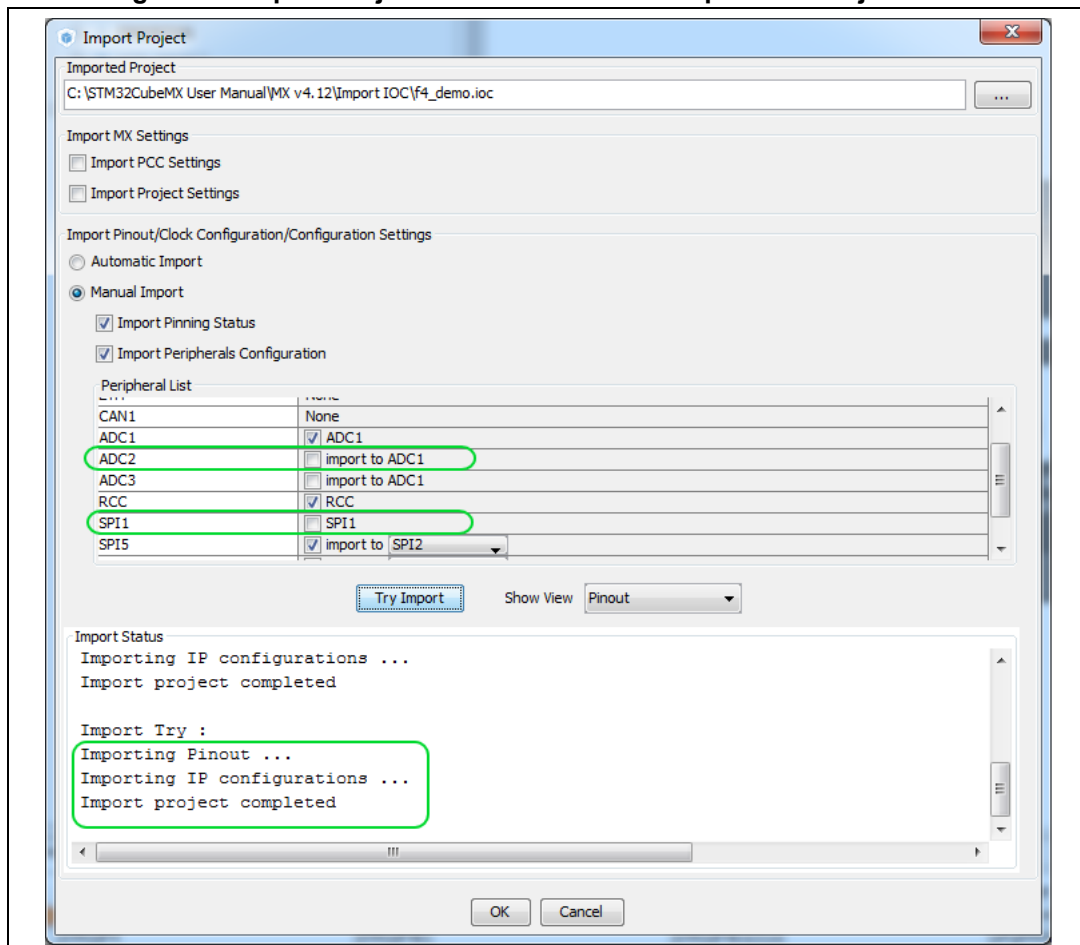**Figure 32. Import Project menu - Try import with errors**

Figure 33. Import Project menu - Successful import after adjustments

3. Choose **OK** to import with the current status or **Cancel** to go back to the empty project without importing.

    Upon import, the Import icon gets grayed since the MCU is now configured and it is no more possible to import a non-empty configuration.
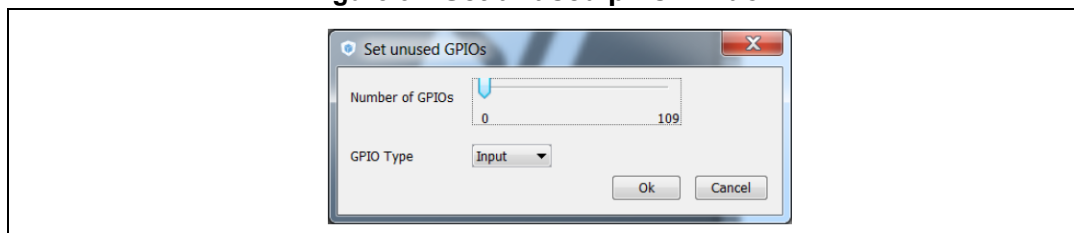
4.7 Set unused / Reset used GPIOs windows

These windows allow configuring several pins at a time in the same GPIO mode.

To open them:

- Select **Pinout** > **Set unused GPIOs** from the STM32CubeMX menu bar.

Note: *The user selects the number of GPIOs and lets STM32CubeMX choose the actual pins to be configured or reset, among the available ones.*

Figure 34. Set unused pins window

- Select **Pinout > Reset used GPIOs** from the STM32CubeMX menu bar.

Depending whether the Keep Current Signals Placement option is checked or not on the toolbar, STM32CubeMX conflict solver will be able to move or not the GPIO signals to other unused GPIOs:

- When Keep Current Signals Placement is off (unchecked), STM32CubeMX conflict solver can move the GPIO signals to unused pins in order to fit in another peripheral mode.

- When Keep Current Signals Placement is on (checked), GPIO signals will not be moved and the number of possible peripheral modes becomes limited.

Refer to *Figure 36* and *Figure 37* and check the limitation in available peripheral modes.
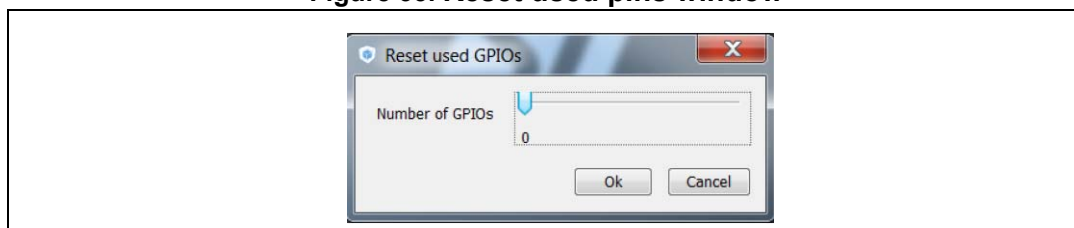
Figure 35. Reset used pins window

Figure 36. Set unused GPIO pins with Keep Current Signals Placement checked
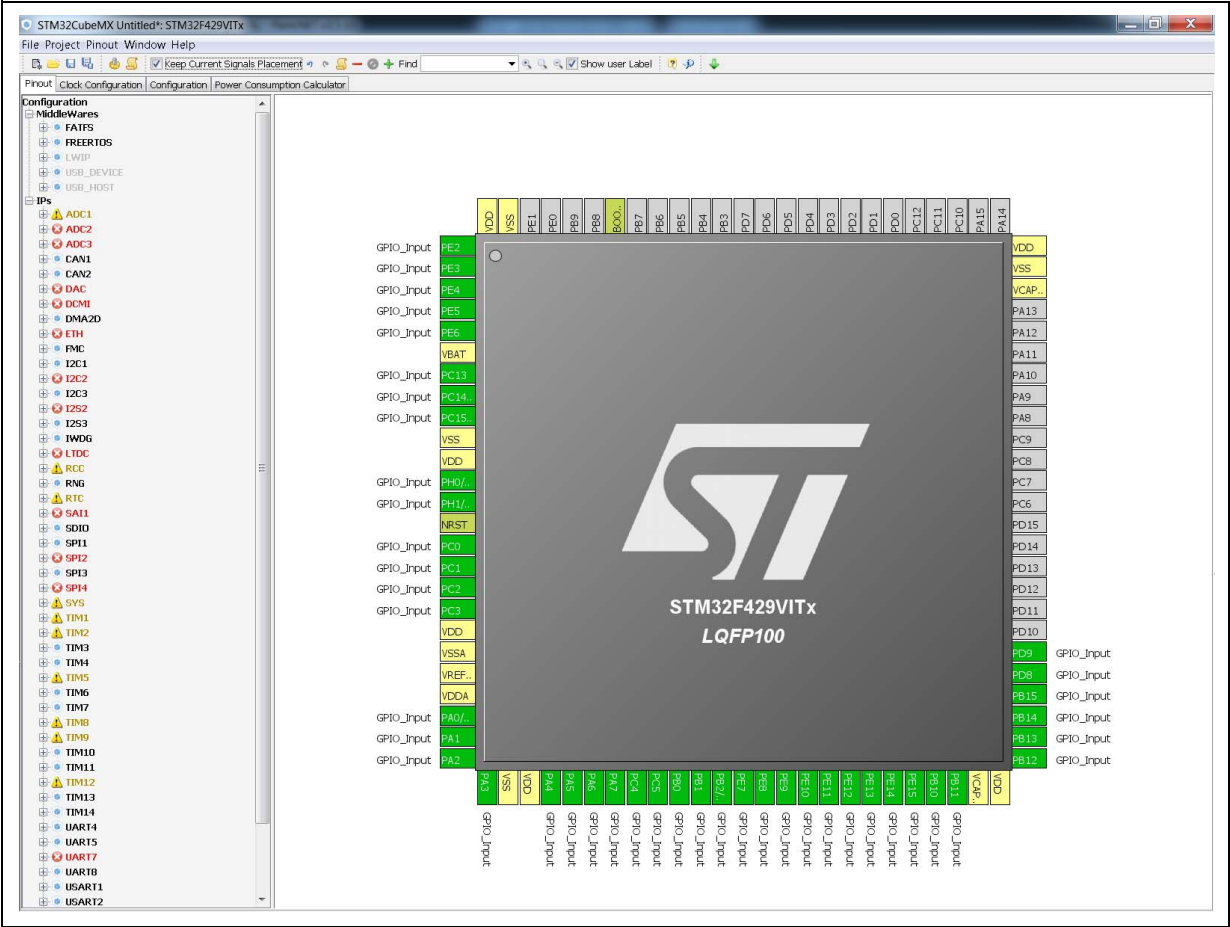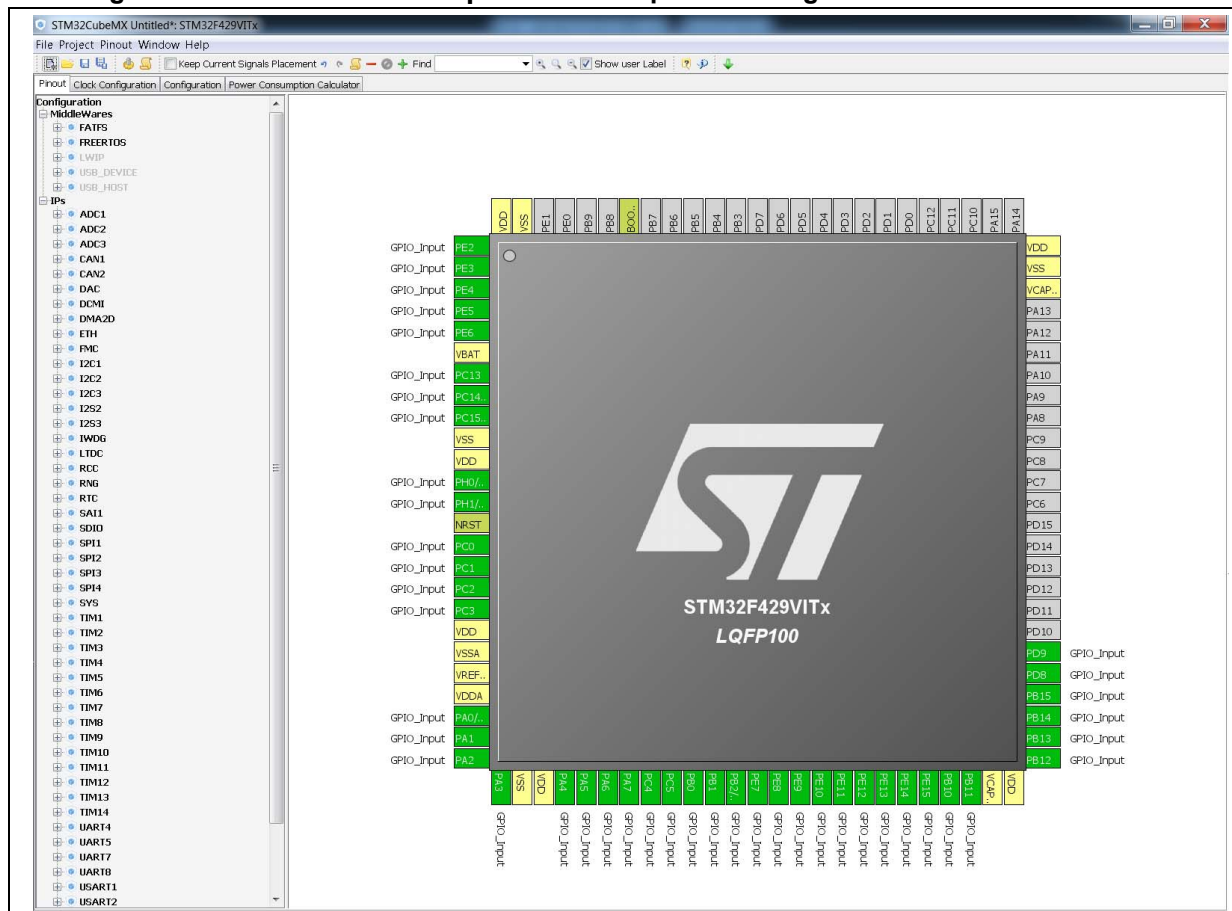
Figure 37. Set unused GPIO pins with Keep Current Signals Placement unchecked

4.8 Project Settings window

This Project Settings windows includes 3 tabs:

- A general project setting tab allowing to specify the project name, the location, the toolchain, and the firmware version.
- A code generation tab allowing to set code generation options such as the location of peripheral initialization code, library copy/link options, and to select templates for customized code.
- An advanced settings tab dedicated to ordering STM32CubeMX initialization function calls.

There are several ways to open the Project Settings window:

1. By selecting **Project > Settings** from the STM32CubeMX menu bar (see *Figure 38*). The code generation will then be generated in the project folder tree shown in *Figure 39*.
2. By clicking **Project** > **Generate code** for the first time.
3. By selecting **Save As** for a project that includes C code generation (and not only pin configuration).

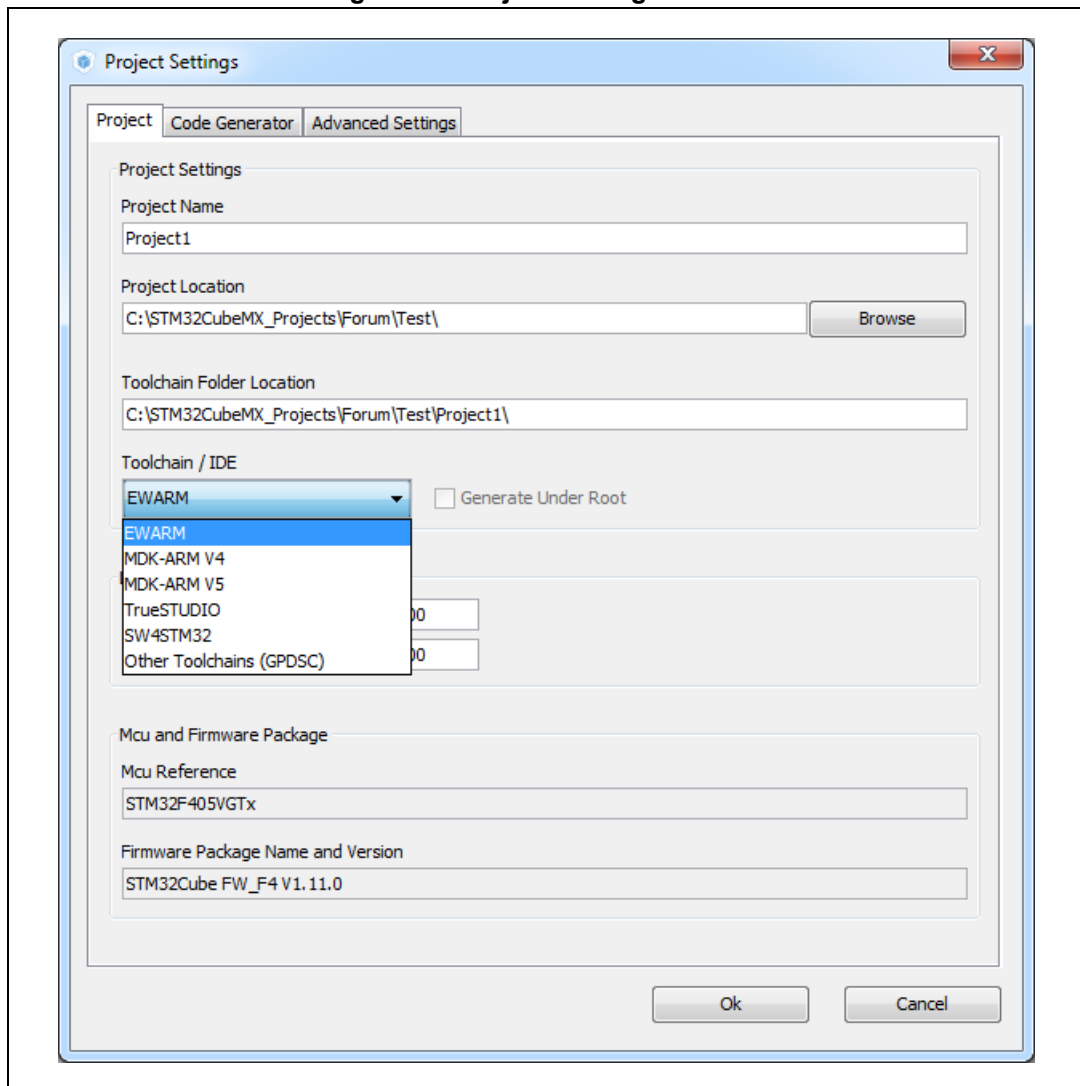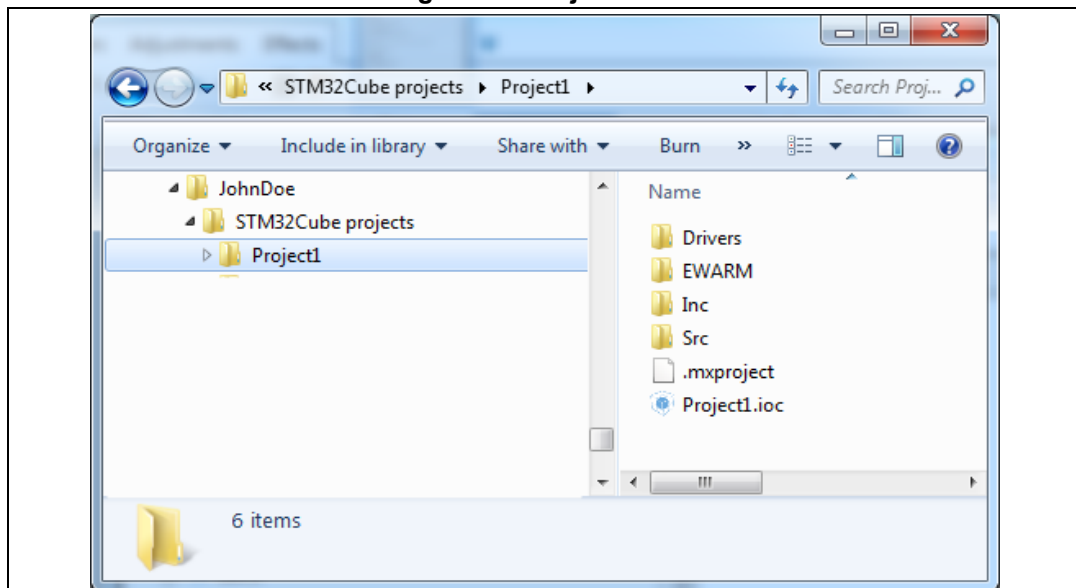**Figure 38. Project Settings window**

Figure 39. Project folder

4.8.1 Project tab

The Project tab of the Project Settings window allows configuring the following options (see *Figure 38*):

- Project settings: project name, location, toolchain folder for toolchain specific generated files, and toolchain to be used for project generation.

  Selecting *Other Toolchains (GPDSC)* generates a gpdsc file. The gpdsc file provides a generic description of the project, including the list and paths of drivers and other files (such as startup files) that are required for building the project. This allows extending STM32CubeMX project generation to any toolchain supporting gpdsc since the toolchain will be able to load a STM32CubeMX generated C project by processing the gpdsc file information. To standardize the description of embedded projects, the gpdsc solution is based on CMSIS-PACK.

- Additional project settings for SW4STM32 and Atollic TrueSTUDIO toolchains:

  Select the optional *Generate under root* checkbox to generate the toolchain project files in STM32CubeMX user project root folder or unselect it to generate them under a dedicated toolchain folder.

  STM32CubeMX project generation under the root folder allows to benefit from the following Eclipse features when using Eclipse-based IDEs such as SW4STM32 and TrueStudio:

  – Optional copy of the project into the Eclipse workspace when importing a project.
  – Use of source control systems such as GIT or SVN from the Eclipse workspace.

  However, it shall be noted that choosing to copy the project into workspace will prevent any further synchronization between changes done in Eclipse and changes done in STM32CubeMX as there will be 2 different copies of the project.

- Linker settings: value of minimum heap and stack sizes to be allocated for the application. The default values proposed are 0x200 and 0x400 for heap and stack