STM8 configuration tool

Introduction

STM8CubeMX is a graphical tool for STM8 core microcontrollers. It is part of STM8Cube™ initiative (see Section 1 STM8CubeMX overview) and is available as a standalone application.

STM8CubeMX has the following key features:

- Easy microcontroller selection covering whole STM8 portfolio
- Board selection from a list of STMicroelectronics boards
- **Easy microcontroller configuration** (pins and peripherals functional modes, clock tree)
- **Easy switching to another microcontroller belonging to the same series** by importing a previously-saved configuration to a new MCU project
- Easy exporting of current configuration to a compatible MCU
- Generation of configuration reports
- **Power consumption calculation** for a user-defined application sequence
- **Self-updates** allowing the user to keep the STM8CubeMX up-to-date

Although STM8CubeMX offers a user interface compliant with STM8 MCU designs, it is recommended to refer to the product technical documentation for details on the actual implementation of microcontroller peripherals.

STM8 microcontroller reference manuals and datasheets are available from http://www.st.com.

1 STM8CubeMX overview

STM8Cube™ is an STMicroelectronics's original initiative to ease developers' life by reducing development efforts, time and cost. STM8Cube covers STM8 portfolio.

The STM8CubeMX, a graphical software configuration tool that allows configuring STM8 microcontrollers very easily and generating the corresponding configuration reports.

2 Getting started with STM8CubeMX

2.1 Principles

Customers need to quickly identify the MCU that best meets their requirements (core architecture, features, memory size, performance…). Board designers main concerns are to optimize the microcontroller pin configuration for their board layout and to fulfill the application requirements (choice of peripherals operating modes). Users may also be interested in migrating existing designs to different microcontrollers:

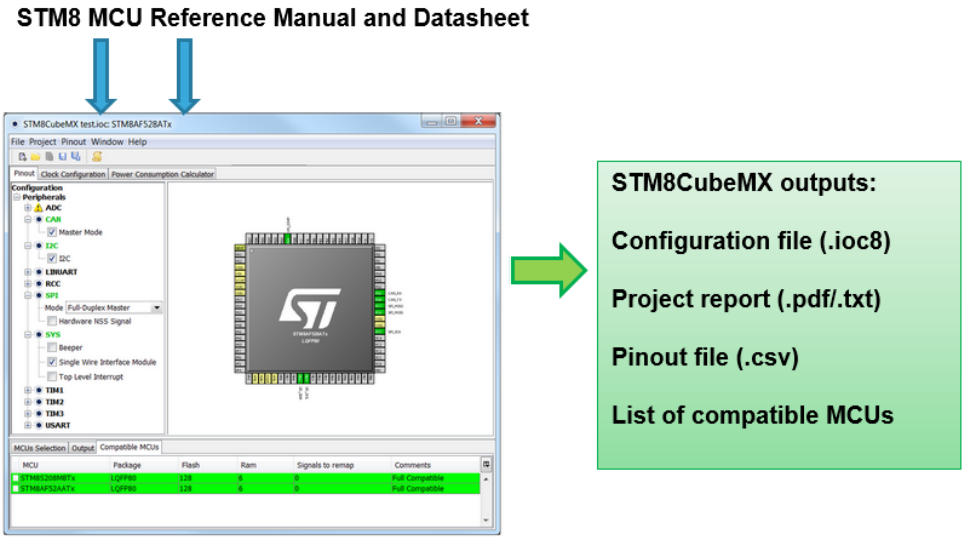
- Optimized migration time is achieved thanks to STM8CubeMX built-in knowledge of STM8 microcontrollers and peripherals.

STM8CubeMX graphical interface performs the following functions:

- Fast and easy configuration of the MCU pins, clock tree and peripheral selection
- Generation of pin configuration report for board designers

During the configuration process, STM8CubeMX detects conflicts and invalid settings and highlights them through meaningful icons and useful tool tips.

Figure 1. Overview of STM8CubeMX



2.2 Key features

STM8CubeMX comes with the following features:

- **Project management**

  STM8CubeMX allows creating, saving and importing previously saved projects:

  – When STM8CubeMX is launched, the user can choose to create a new project or to load a previously saved project.

  – Saving the project saves user settings and configuration performed within the project in an .ioc8 file that will be used the next time the project will be loaded in STM8CubeMX.

  – STM8CubeMX projects comes with an .ioc8 file that can be saved anywhere, next to other .ioc8 files.

  – STM8CubeMX does not support C code generation.

  - It allows easily selecting an STMicroelectronics MCU and board: when starting a new project, a dedicated window opens to select either a microcontroller or an STMicroelectronics board from the STM8 portfolio. Different filtering options are available to ease MCU and board selection.
- **Easy pinout configuration**
  - From the **Pinout** view, the user can select the peripherals from a list and configure the peripheral modes required for the application. STM8CubeMX assigns and configures the pins accordingly.
  - For more advanced users, it is also possible to directly map a peripheral function to a physical pin using the **Chip** view. The signals can be locked on pins to prevent STM8CubeMX conflict solver from moving the signal to another pin.
  - Pinout configuration can be exported as a .csv file.
- **Power consumption calculation**
  - Starting with the selection of a microcontroller part number and a battery type, the user can define a sequence of steps representing the application life cycle and parameters (choice of frequencies, enabled peripherals, step duration).
  - STM8CubeMX Power Consumption Calculator returns the corresponding power consumption and battery life estimates.
- **Clock tree configuration**
  - STM8CubeMX offers a graphical representation of the clock tree as it can be found in the device reference manual.
  - The user can change the default settings (clock sources, prescaler and frequency values). The clock tree is then updated accordingly. Invalid settings and limitations are highlighted and documented with tooltips.
  - Clock tree configuration conflicts can be solved by using the solver feature. When no exact match is found for a given user configuration, STM8CubeMX proposes the closest solution.
- **Automatically updating STM8CubeMX**
  - STM8CubeMX comes with an updater mechanism that can be configured for automatic or on-demand check for updates. It supports STM8CubeMX self-updates.
- **Report generation**
  - .pdf and .csv reports can be generated to document user configuration work.

2.3 Rules and limitations

- Refer to Section 7 STM8CubeMX pin assignment rules for a description of pin assignment rules.

3 Installing and running STM8CubeMX

3.1 System requirements

3.1.1 Supported operating systems and architectures

- Windows ® 7: 32-bit (x86), 64-bit (x64)
- Windows ® 8: 32-bit (x86), 64-bit (x64)
- Windows ® 10: 32-bit (x86), 64-bit (x64)
- Linux ®: 32-bit (x86) and 64-bit (x64) (tested on RedHat, Ubuntu and Fedora)

  Since STM8CubeMX is a 32-bit application, some versions of Linux 64-bit distributions require to install 32-bit compliant packages such as ia32-libs.
- MacOS: 64-bit (x64) (tested on OS X Yosemite)

3.1.2 Memory prerequisites

- Recommended minimum RAM: 2 Gbytes.

3.1.3 Software requirements

The following software must be installed:

- For Windows and Linux, install Java Run Time Environment for 1.7.0_45 or later.

  If Java is not installed on your computer or if you have an old version, STM8CubeMX installer will open the Java download web page and stop.
- For MacOS, install Java Development Kit 1.7.0_45 or later.

3.2 Installing/uninstalling STM8CubeMX standalone version

3.2.1 Installing STM8CubeMX standalone version

To install STM8CubeMX, follow the steps below:

1. Download STM8CubeMX installation package from www.st.com/stm8cubemx.
2. Extract (unzip) STM8cubemx.zip whole package into the same directory.
3. Check your access rights and launch the installation wizard:

   On windows:

   a. Make sure you have administrators rights.

   b. Double-click the SetupSTM8CubeMX-VERSION.exe file to launch the installation wizard.

   On Linux:

   a. Make sure you have access rights to the target installation directory. You can run the installation as root (or sudo) to install STM8CubeMX in shared directories.

   b. Double-click (or launch from the console window) on the SetupSTM8CubeMX-VERSION.linux file.

   On MacOS:

   a. Make sure you have administrators rights.

   b. Double-click SetupSTM8CubeMX-VERSION application file to launch the installation wizard.
4. Upon successful installation of STM8CubeMX on Windows, STM8CubeMX icon is displayed on your desktop and STM8CubeMX application is available from the Program menu. STM8CubeMX .ioc8 files are displayed with a cube icon. Double-click them to open up them using STM8CubeMX.
5. Delete the content of the zip from your disk.

Note: If the proper version of the Java Runtime Environment (version 1.7_45 or newer) is not installed, the wizard will propose to download it and stop. Restart STM8CubeMX installation once Java installation is complete. Refer to Section 6 FAQ for issues when installing the JRE.

Note: When working on Windows, only the latest installation of STM8CubeMX will be enabled in the program menu. Previous versions can be kept on your PC (not recommended) when different installation folders have been specified. Otherwise, the new installation overwrites the previous ones.

3.2.2 Installing STM8CubeMX from command line

There are two ways to launch an installation from a console window: either in console interactive mode or via a script.

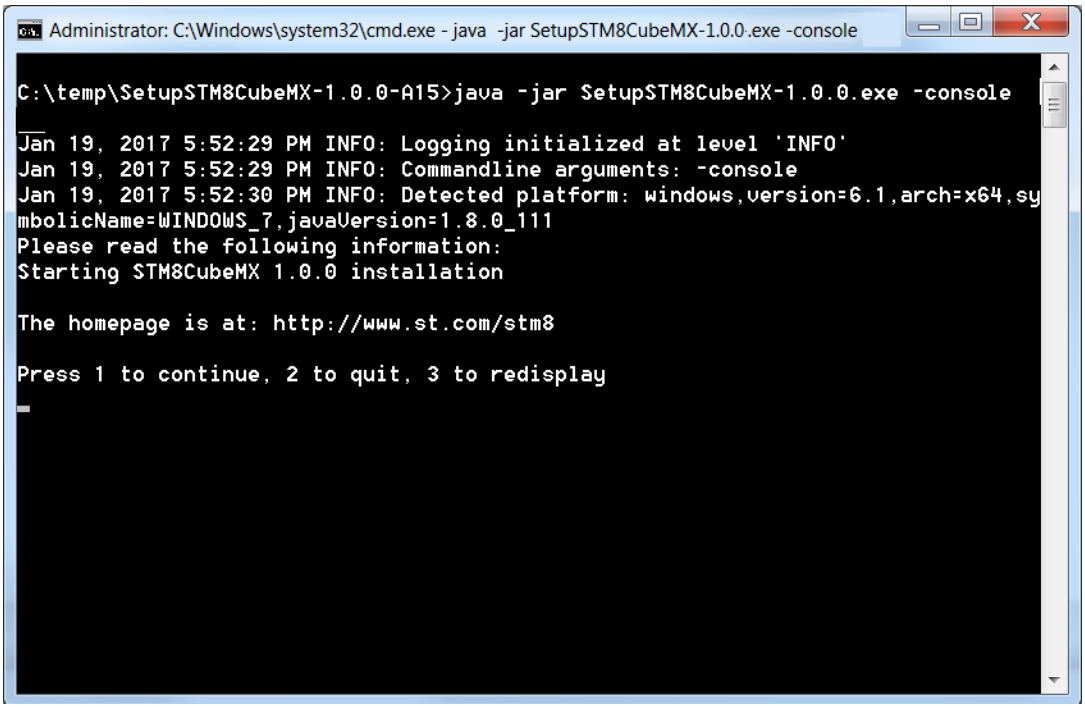
Interactive mode

To perform interactive installation, type the following command:

```
java -jar SetupSTM8CubeMX-1.0.0.exe -console
```

At each installation step, an answer is requested (see Figure 2. Example of STM8CubeMX installation in interactive mode below).

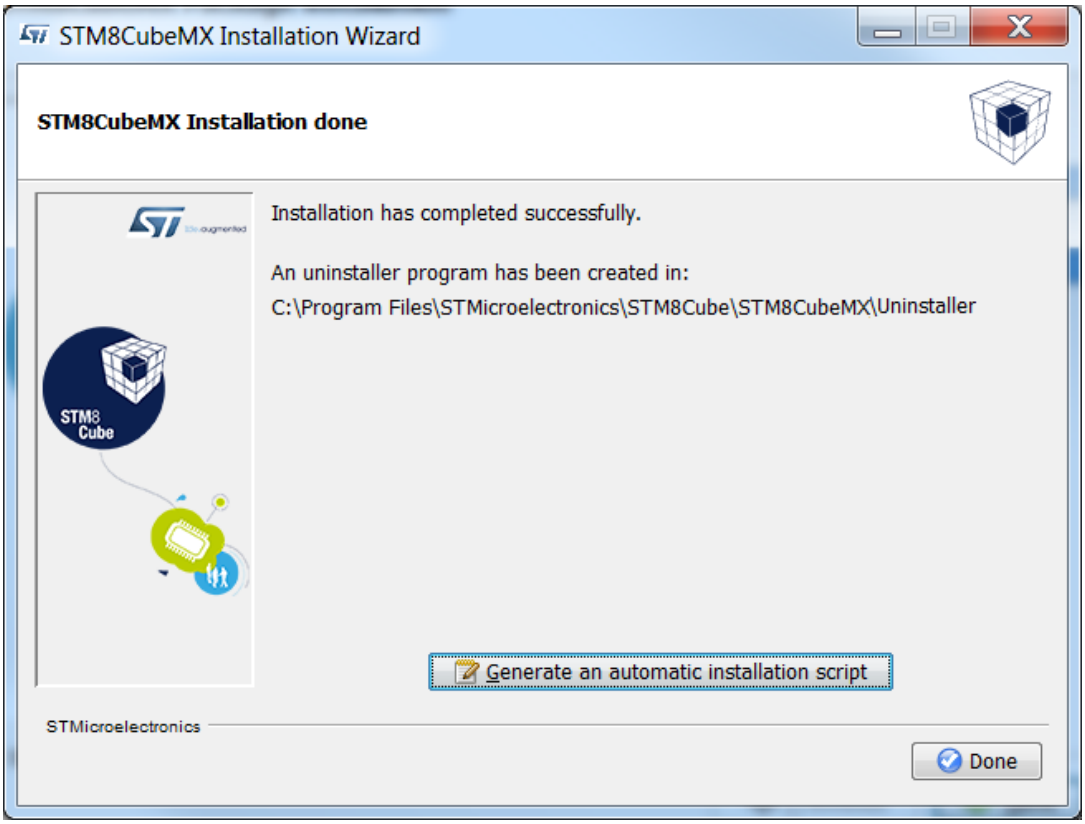
Figure 2. Example of STM8CubeMX installation in interactive mode



Auto-install mode

When the installation is complete, either using STM8CubeMX graphical wizard or the console mode, it is possible to generate an auto-installation script containing user installation preferences (see Figure 3. STM8Cube Installation Wizard below):

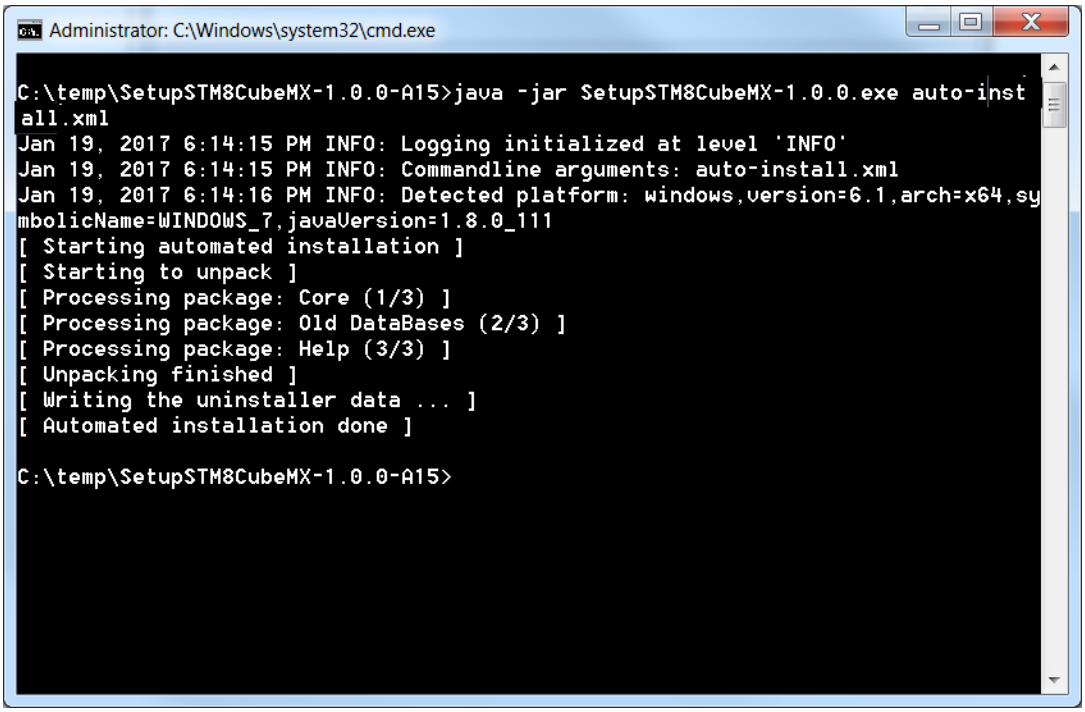
Figure 3. **STM8Cube Installation Wizard**



You can then launch the installation just by typing the following command:

```
java -jar SetupSTM8CubeMX-1.0.0.exe auto-install.xml
```

Figure 4. **Auto-install command line**



3.2.3 Uninstalling STM8CubeMX standalone version

Uninstalling STM8CubeMX on MacOS

To uninstall STM8CubeMX on MacOS, use the following command line:

```
java -jar <STM8CubeMX installation path>/Uninstaller/uninstaller.jar.
```

Uninstalling STM8CubeMX on Linux

There are two means to uninstall STM8CubeMX on Linux:

- By using the following command line

```
java -jar <STM8CubeMX installation path>/Uninstaller/uninstaller.jar.
```

- Through a File Explorer window:
  1. Use a file explorer.
  2. Go to the Uninstaller directory of STM8CubeMX installation.
  3. Double-click the start uninstall desktop shortcut.

Uninstalling STM8CubeMX on Windows

There are three means to uninstall STM8CubeMX on Windows:

- By using the following command line

```
java -jar <STM8CubeMX installation path>/Uninstaller/uninstaller.jar.
```

- Through a Windows Explorer window:
  1. Use a file explorer.
  2. Go to the Uninstaller directory of STM8CubeMX installation.
  3. Double-click the start uninstall desktop shortcut.
- Through the Windows Control Panel:

1. Select Programs and Features from the Windows Control Panel to display the list of programs installed on your computer.
2. Right-click STM8CubeMX and select uninstall.

3.3 Launching STM8CubeMX

3.3.1 Running STM8CubeMX as standalone application

To run STM8CubeMX as a standalone application on Windows:

- select STM8CubeMX from Program Files > ST Microelectronics > STM8Cube > STM8CubeMX.
- or double-click STM8CubeMX icon on your desktop.

To run STM8CubeMX as a standalone application on Linux, launch the STM8CubeMX executable from STM8CubeMX installation directory.

3.3.2 Running STM8CubeMX in command-line mode

To ease its integration with other tools, STM8CubeMX provides a command-line mode. Using a set of commands, you can:

- load an MCU
- load an existing configuration
- save a current configuration

Three command-line modes are available:

- To run STM8CubeMX in interactive command-line mode, use the following command line:
  On Windows:

```
java -jar STM8CubeMX.exe -i
```

```
java -jar STM8CubeMX -i
```

  The “MX>” prompt is then displayed to indicate that the application is ready to accept commands.
- To run STM8CubeMX in command-line mode getting commands from a script, use the following command line:
  On Windows:

```
java -jar STM8CubeMX.exe -s <script filename>
```

  On Linux and MacOS:

```
java -jar STM8CubeMX -s <script filename>
```

All the commands to be executed must be listed in the script file. An example of script file content is shown below:

```
load STM8AF5268Tx
config save C:\\STM8CubeProjects\\MyFirstProject.ioc8
exit
```

- To run STM8CubeMX in command-line mode getting commands from a scripts and without UI, use the following command line:
  On Windows:

```
java -jar STM8CubeMX.exe -q <script filename>
```

  On Linux and MacOS:

```
java -jar STM8CubeMX -q <script filename>
```

Here again, the user can enter commands when the MX prompt is displayed.

See Table 1. Command line summary for available commands.

Table 1. Command line summary

Command line	Purpose	Example
help	This command displays the list of available commands	help
load <mcu>	This command loads the selected MCU	load STM8AF5268Tx
config load <filename>	This command loads a previously saved configuration	config load C:\\Cube\\ccmram\ \ccmram.ioc8
config save <filename>	This command saves the current configuration	config save C:\\Cube\\myProjects\ \demo1.ioc8
config saveext <filename>	This command saves the current configuration with all parameters, including those for which values have been kept to defaults (unchanged by the user).	config saveext C:\\Cube\\myProjects\ \demo1.ioc8
config saveas <filename>	This command saves the current project under a new name	config saveas C:\\Cube\\myProjects\ \demo2.ioc8
csv pinout <filename>	This command exports the current pin configuration as a csv file. This file could later be imported into a board layout tool.	Csv pinout mypinout.csv
script <filename>	This command runs all commands in the script file. There must be one command per line.	script myscript.txt
exit	End STM8CubeMX process	exit

3.4 Getting STM8Cube updates

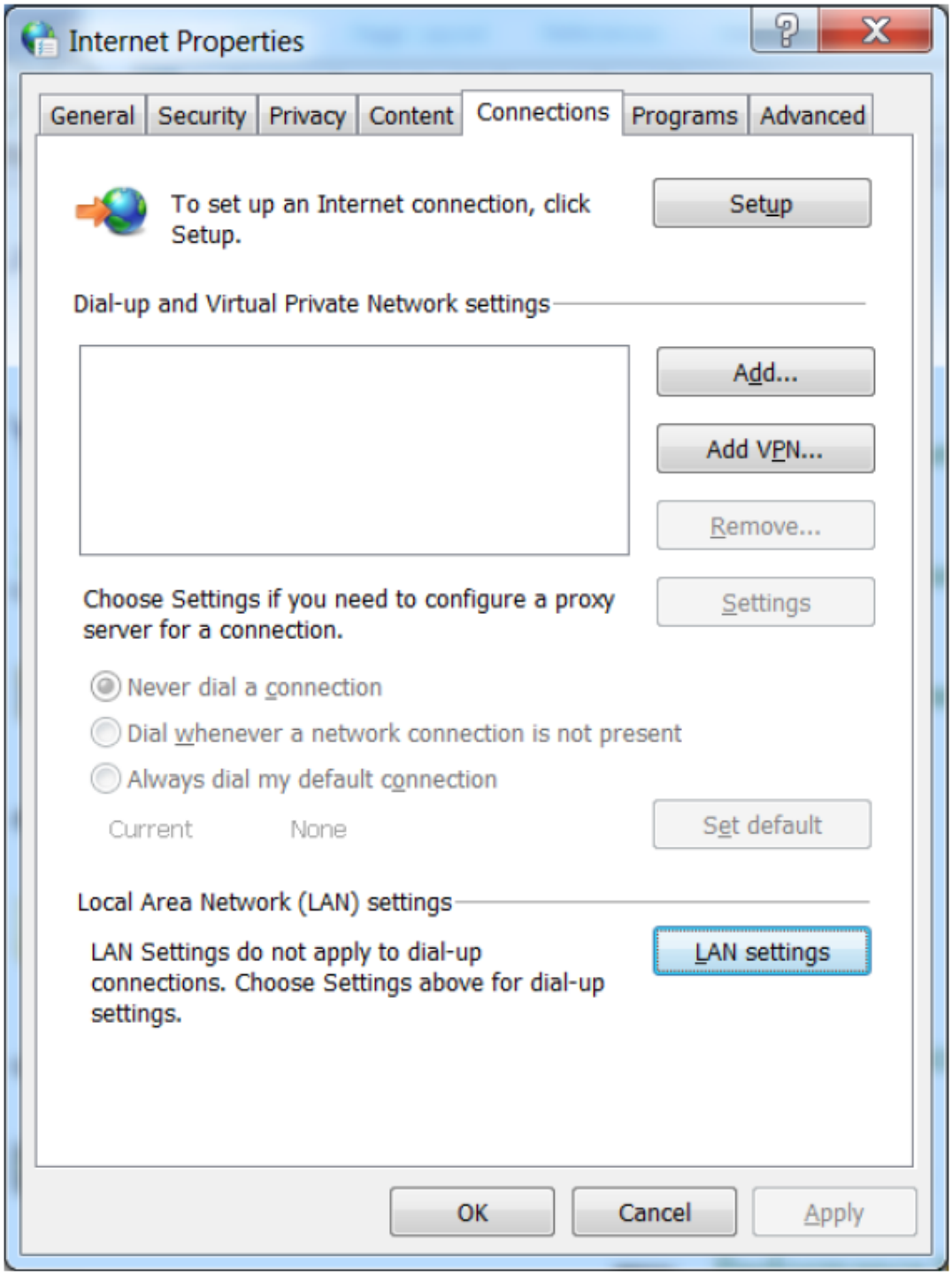
STM8CubeMX implements a mechanism to access the internet for performing STM8CubeMX self-updates.

Installation and update related sub-menus are available under the Help menu.

If the PC on which STM8CubeMX runs is connected to a computer network using a proxy server, STM8CubeMX needs to connect to that server to access the internet and get STM8CubeMX self-update package. Refer to Section 3.4.1 Updater configuration for a description of this connection configuration.

To view Windows default proxy settings, select Internet options from the Control panel and select LAN settings from the Connections tab (see Figure 5. Displaying Windows default proxy settings).

Figure 5. **Displaying Windows default proxy settings**



Several proxy types exist and different computer network configurations are possible:

- Without proxy: the application directly accesses the web (Windows default configuration).
- Proxy without login/password
- Proxy with login/password: when using an internet browser, a dialog box opens and prompts the user to enter his login/password.
- Web proxies with login/password: when using an internet browser, a web page opens and prompts the user to enter his login/password.

If necessary, contact your IT administrator for proxy information (proxy type, http address, port).
STM8CubeMX does not support web proxies.

3.4.1 Updater configuration

To perform STM8CubeMX self-updates, the updater must be configured as follows:

1. Select **Help > Updater Settings** to open the **Updater Settings** window.
2. From the **Updater Settings** tab (see Figure 6. Updater Settings window) Enable/Disable the automatic check for updates.
3. In the **Connection Parameters** tab, specify the proxy server settings appropriate for your network configuration by selecting a proxy type among the following possibilities:
   – No Proxy (see Figure 7. Connection Parameters tab - No proxy)
   – Use System Proxy Parameters (see Figure 8. Connection Parameters tab - Use System proxy parameters)
   On Windows, proxy parameters will be retrieved from the PC system settings.
   Uncheck “Require Authentication” if a proxy server without login/password configuration is used.
   – Manual Configuration of Proxy Server (see Figure 9. Connection Parameters tab - Manual Configuration of Proxy Server)
   Enter the Proxy server http address and port number. Enter login/password information or uncheck “Require Authentication” if a proxy server without login/password configuration is used.
4. Uncheck **Remember my credentials** to prevent STM8CubeMX to save encrypted login/password information in a file. This implies reentering login/password information each time STM8CubeMX is launched.
5. Click the Check Connection button 

to verify if the connection works. A green check mark appears to confirm that the connection operates correctly.
6. If the tool is configured for manual checks, select **Help > Check for Updates** to find out about new tool versions available for installation.

Figure 6. Updater Settings window

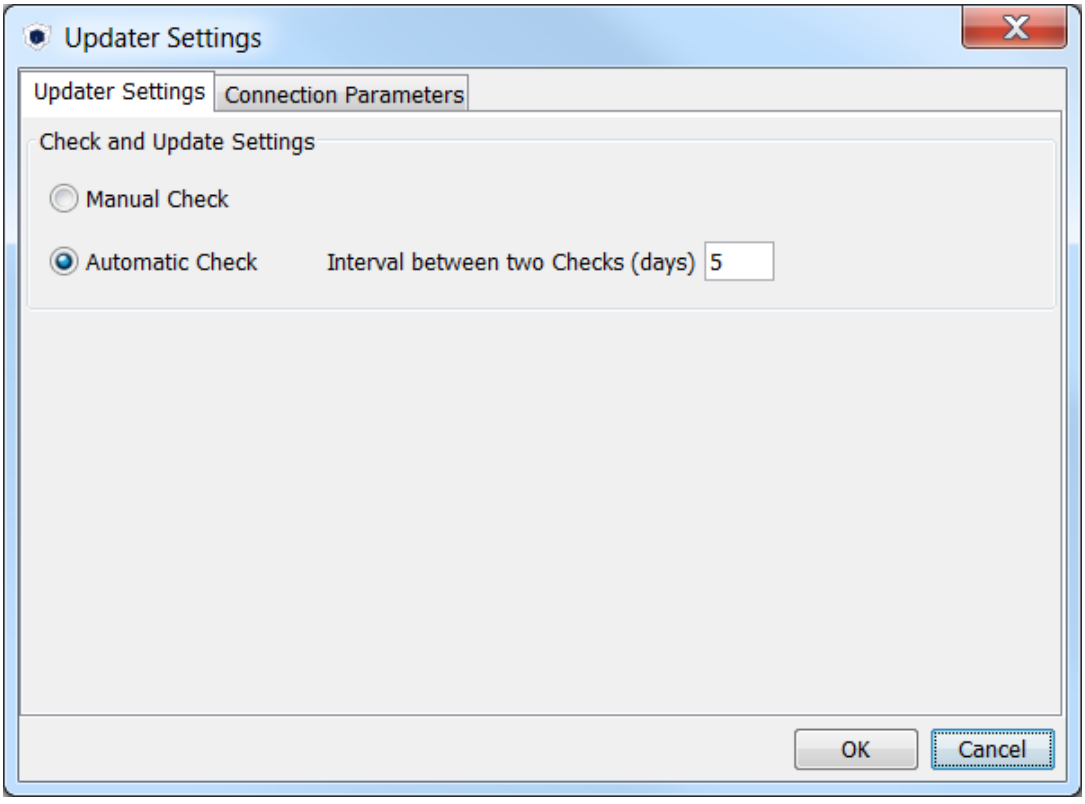


Figure 7. **Connection Parameters tab - No proxy**

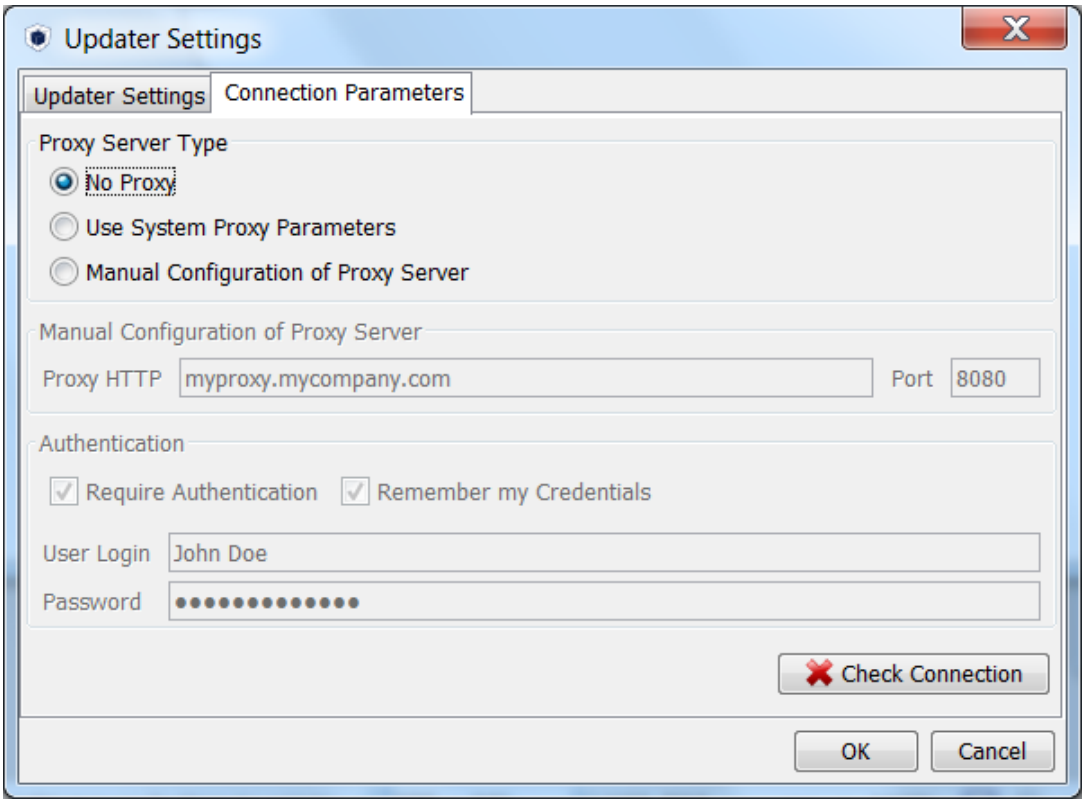


Figure 8. **Connection Parameters tab - Use System proxy parameters**

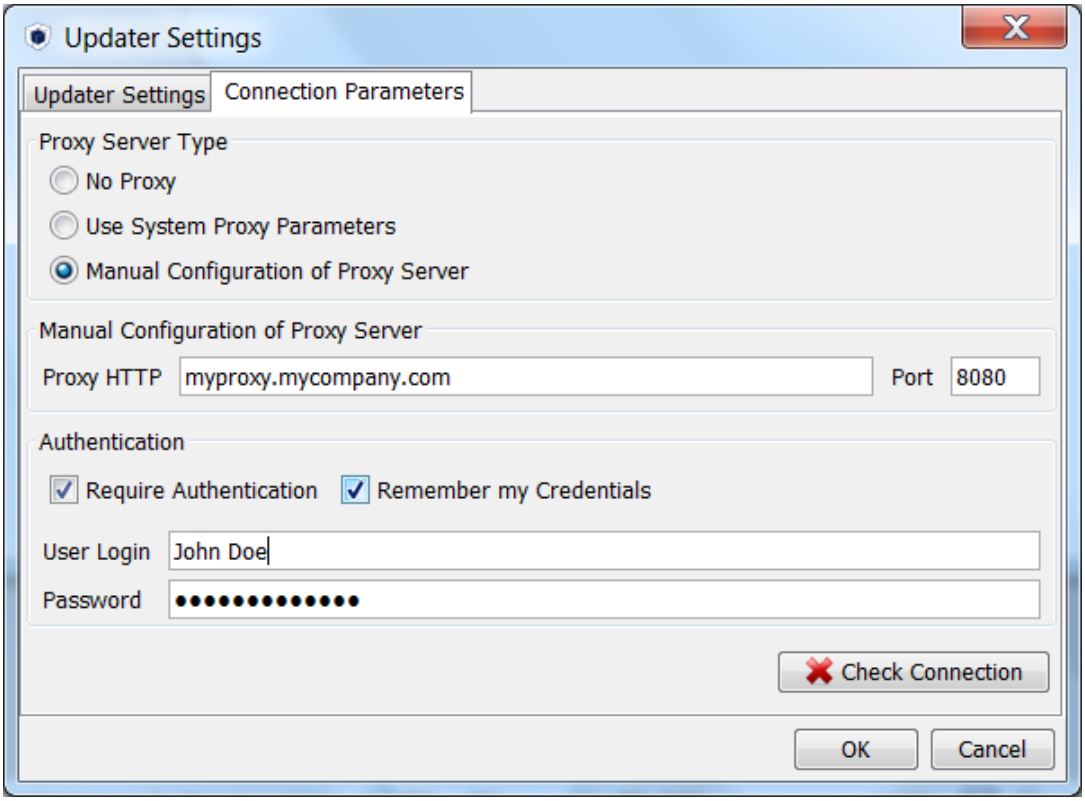
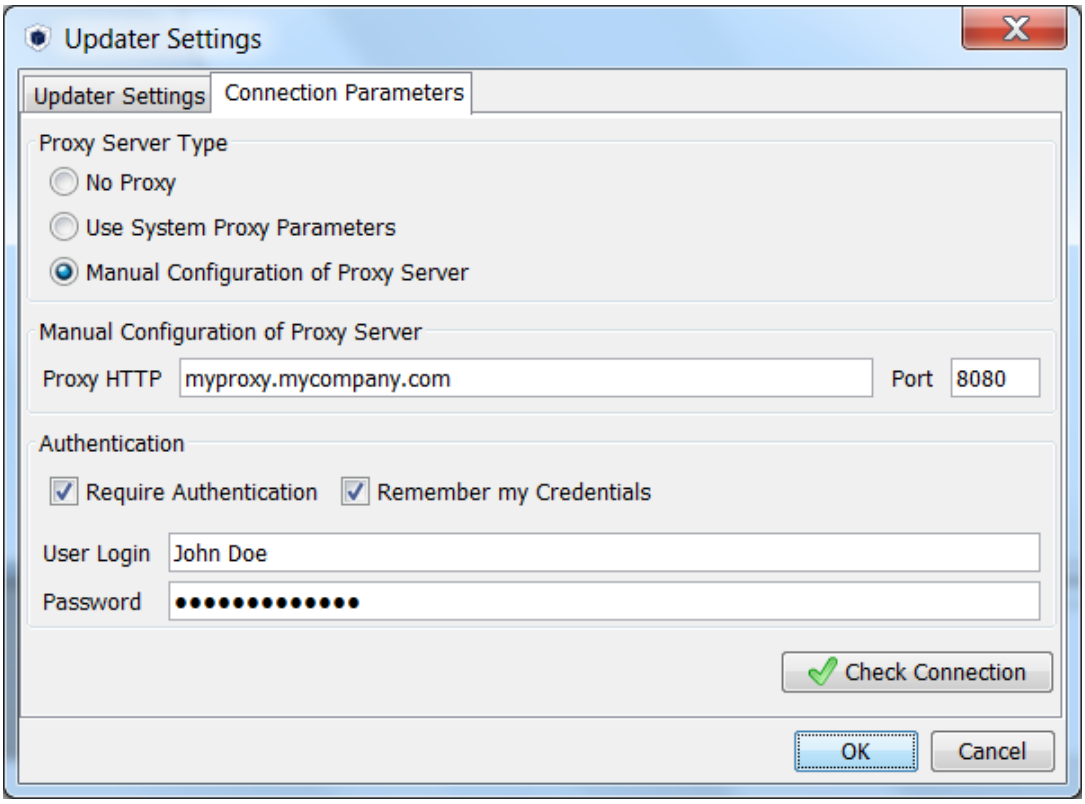


Figure 9. **Connection Parameters tab - Manual Configuration of Proxy Server**



3.4.2 Checking for updates

When the updater is configured for automatic checks, it regularly verifies if updates are available. In this case, a green arrow icon appears on the tool bar ( ).

When automatic checks have been disabled in the updater settings window, the user can manually check if updates are available:

1. Click the icon to open the **Update Manager** window or Select **Help > Check for** updates. All the updates available for the user current installation are listed.
2. Click the check box to select a new version of the package, and then **Install Now** to download the update.

4 STM8CubeMX User Interface

STM8CubeMX user interface consists of a main window, a menu bar, a toolbar, three views (Pinout, Clock Configuration, Power Consumption Calculator) and a set of help windows (MCUs selection, Update manager, About). All these menus are described in the following sections.

4.1 Welcome page

The Welcome page is the first window that opens up when launching STM8CubeMX program. It remains open as long as the application is running. Closing it closes down the application. Refer to Figure 10. STM8CubeMX Welcome page and to Table 2. Welcome page shortcuts for a description of the Welcome page.

Figure 10. STM8CubeMX Welcome page

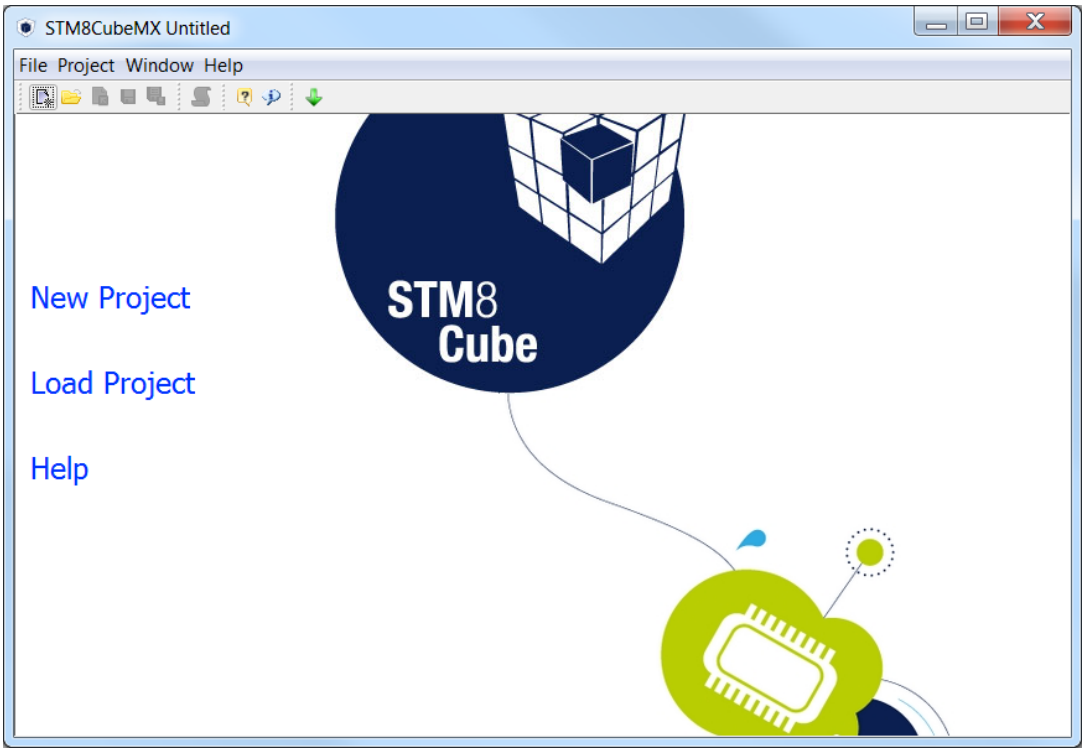


Table 2. Welcome page shortcuts

Name	Description
New Project	This shortcut launches STM8CubeMX new project creation by opening the New project window (select an MCU from the MCU selector tab or a board configuration from the Board selector tab).
Load Project	This shortcut opens a browser window to select a previously saved configuration (.ioc8 file) and loads it.
Help	This shortcut opens the user manual.

4.2 New project window

This window shows two tabs to choose from:

- The MCU selector tab offering a list of target processors
- A Board selector tab showing a list of STMicroelectronics boards.

The MCU selector allows filtering on various criteria: series, lines, packages, peripherals and additional MCU characteristics such as memory size or number of I/Os (see Figure 11. New Project window - MCU selector).

The Board selector allows filtering on STM8 board types, series and peripherals (see Figure 12. New Project window - board selector). Only the default board configuration is proposed. Alternative board configurations obtained by reconfiguring jumpers or by using solder bridges are not supported.

When a board is selected, the **Pinout** view is initialized with the relevant MCU part number (see Section 4.3 Main window).

When a board configuration is selected, the signals change to 'pinned', i.e. they cannot be moved automatically by STM8CubeMX constraint solver (user action on the peripheral tree, such as the selection of a peripheral mode, will not move the signals). This ensures that the user configuration remains compatible with the board.

Figure 11. New Project window - MCU selector

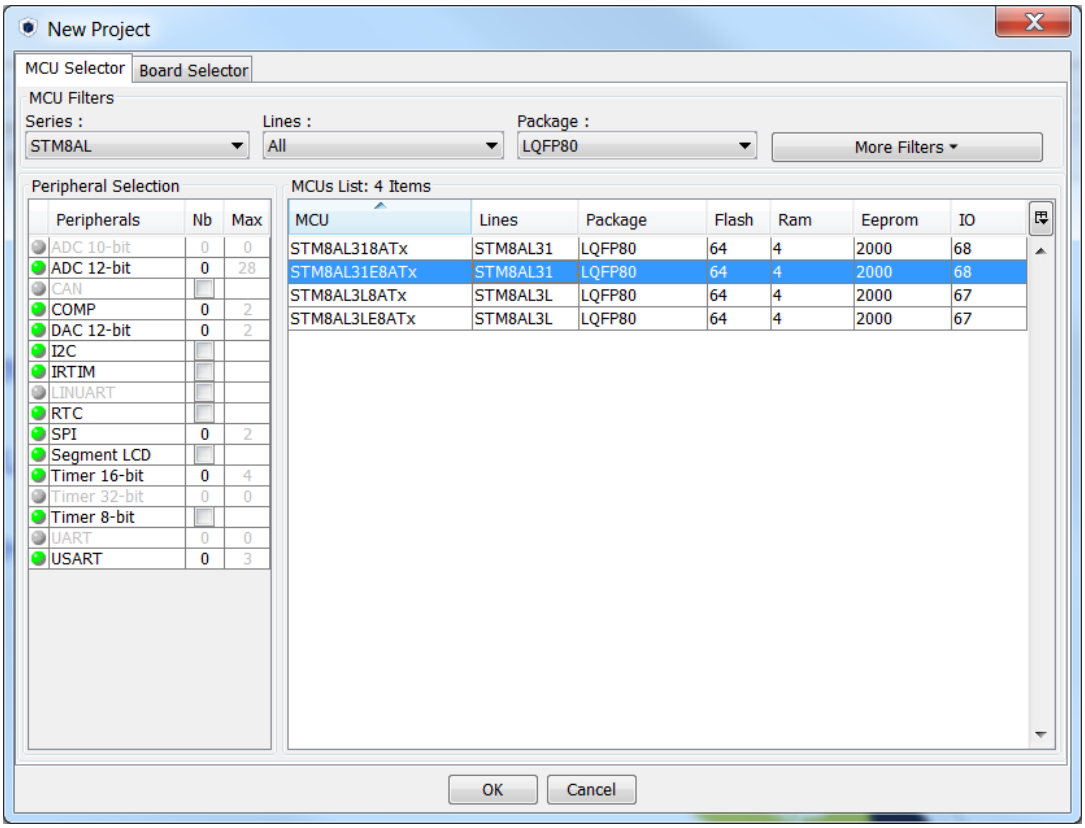
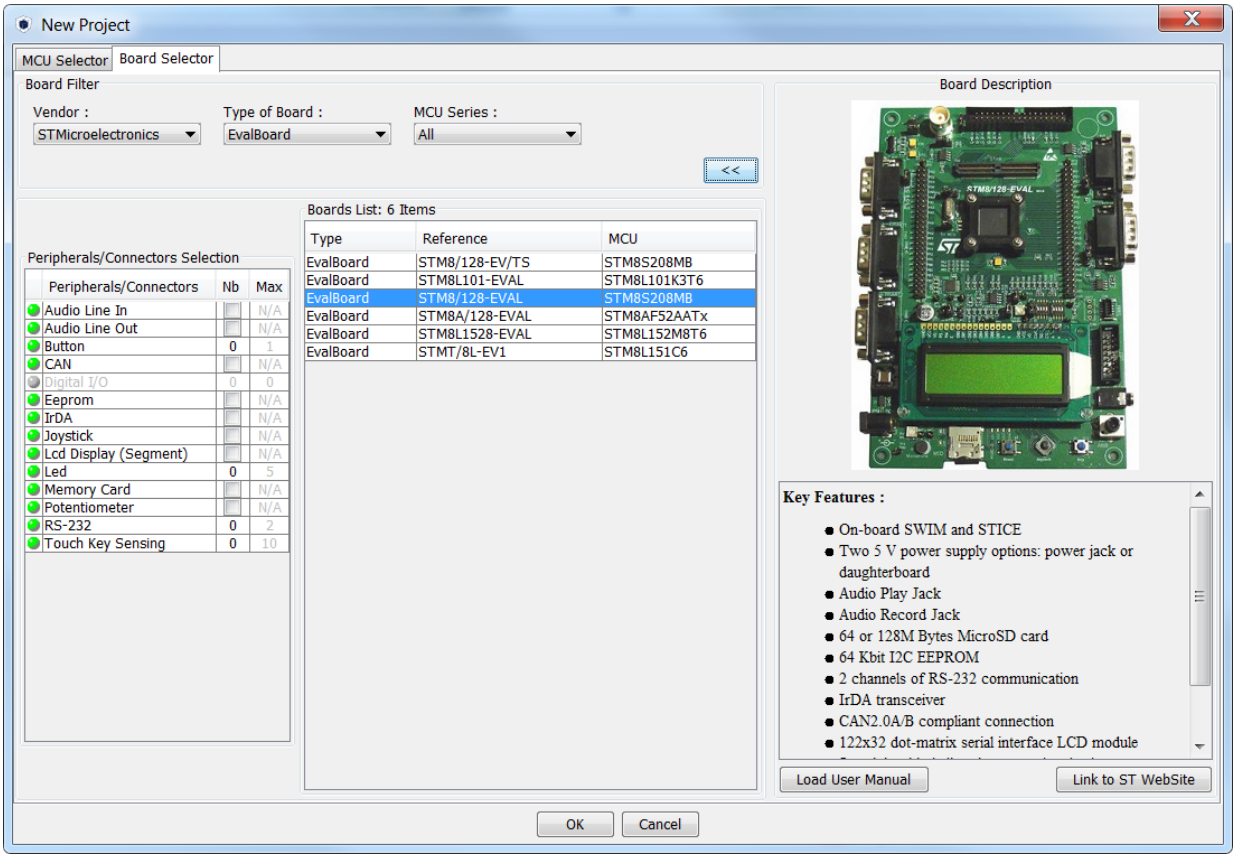


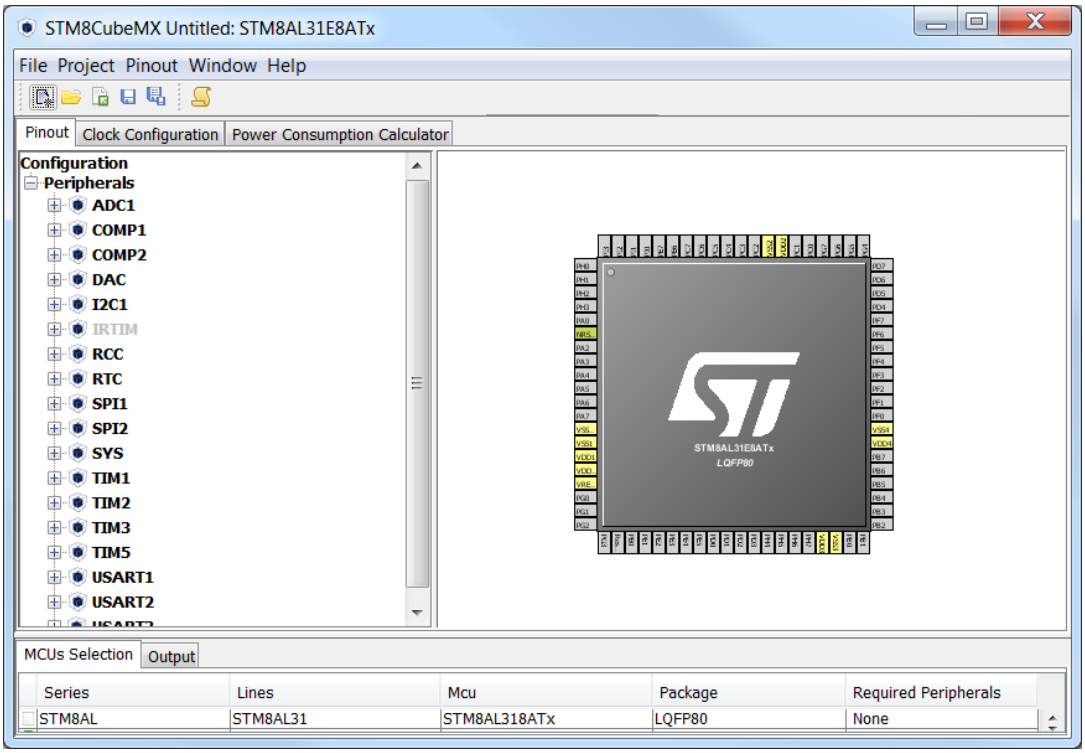
Figure 12. **New Project window - board selector**



4.3 Main window

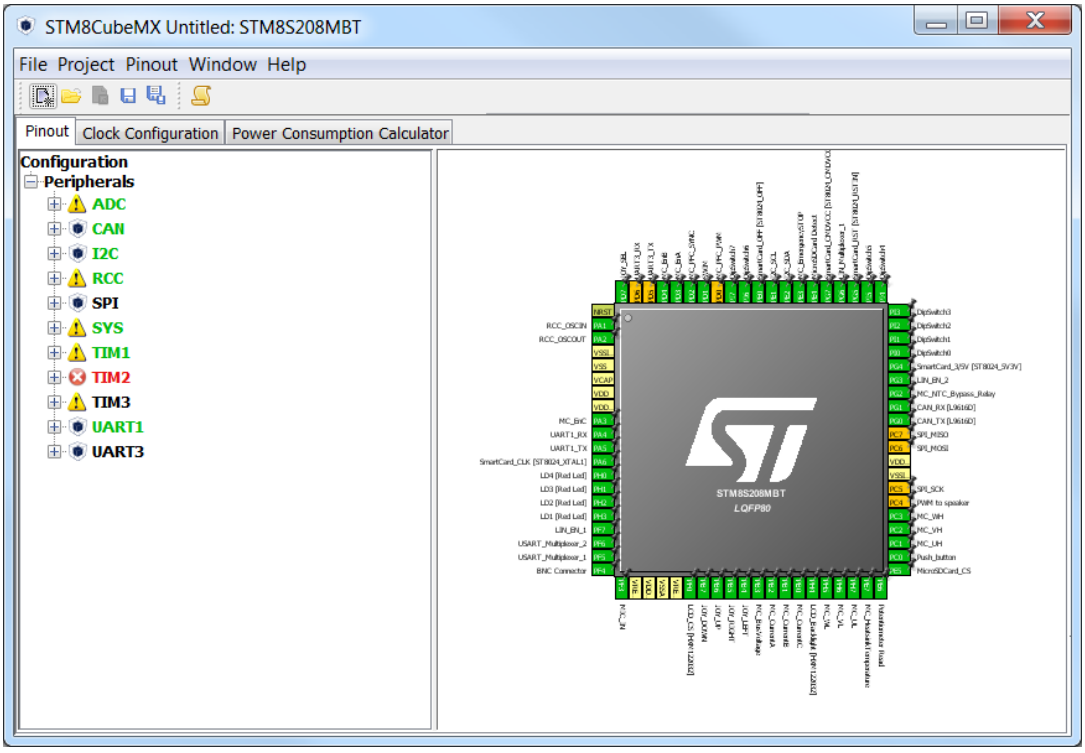
Once an STM8 part number or a board has been selected or a previously saved project has been loaded, the main window displays all STM8CubeMX components and menus (see Figure 13. STM8CubeMX Main window upon MCU selection). Refer to Section 4.4 Toolbar and menus for a detailed description of the toolbar and menus.

Figure 13. **STM8CubeMX Main window upon MCU selection**



Selecting a board, automatically sets the pinout for this board. The user can then manually select from the peripheral tree the peripheral modes required for his application (see Figure 14. STM8CubeMX Main window upon board selection).

Figure 14. **STM8CubeMX Main window upon board selection**



4.4 Toolbar and menus

The following menus are available from STM8CubeMX menu bar:

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

STM8CubeMX menus and toolbars are described in the sections below.

4.4.1 File menu

Refer to Table 3. File menu functions 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 STM8CubeMX project configuration by selecting an STM8CubeMX configuration .ioc8 file.
	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, peripherals, Power Consumption Calculator) as a new project. This action creates an .ioc8 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. Project menu for a description of the **Project** menu and icons.

Table 4. Project menu

Icon	Name	Description
	Generate Report (1)	This menu generates current project configuration as a pdf file and a text file.

1. *If the project was previously saved, the reports are generated at the same location as the project configuration .ioc8 file. Otherwise, the user can choose the destination folder, and whether to save the project configuration as an .ioc8 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 15. Pinout menus (Pinout tab selected)). Refer to Table 5. Pinout menu for a description of the **Pinout** menu and icons.

Figure 15. Pinout menus (Pinout tab selected)

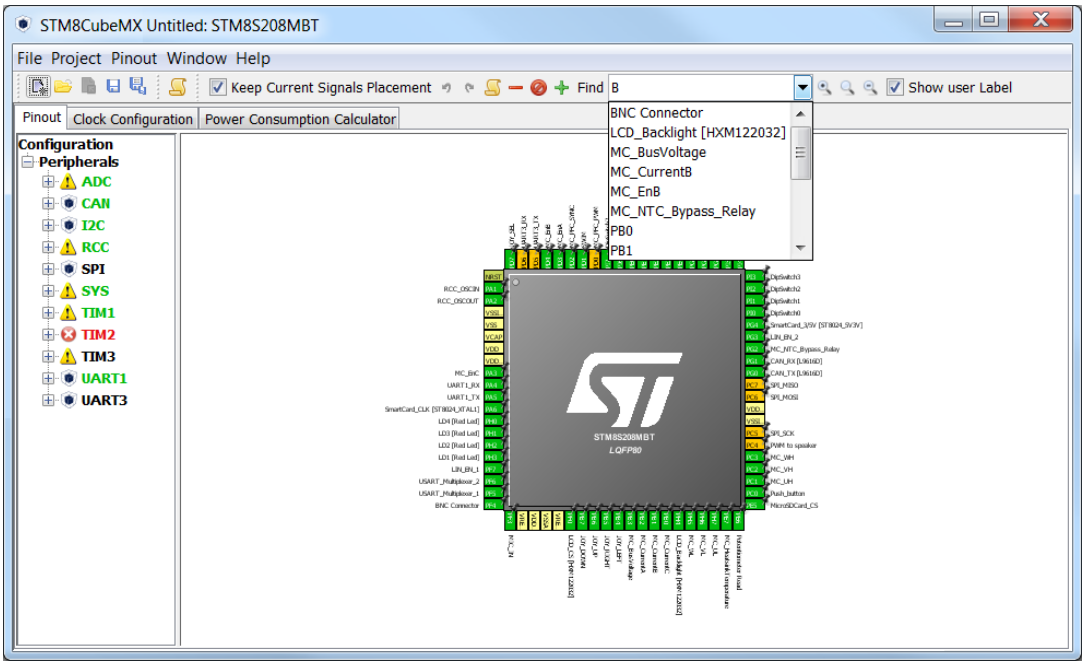


Table 5. Pinout menu

Icon	Name	Description
	Undo	Undoes last configuration steps (one by one)
	Redo	Redoes steps that have been undone (one by one)
	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.
	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 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 puts all configured pins back to their reset state and disables all the peripheral modes previously enabled (whether they were using signals on pins or not).
No icon	Clear Single Mapped Signals	Clears signal assignments to pins for signals that have no associated mode (highlighted in orange and not pinned).

Icon	Name	Description
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). 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: 2 sub-menus to generate with or without alternate functions.
No Icon	List pinout compatible MCUs	Provides a list of MCUs that best match the pin configuration of the current project. The matching can be: • An exact match • A partial match with hardware compatibility: pin locations are the same, pin names may have been changed • A partial match without hardware compatibility: all signals could be mapped but not all at the same pin location
	Collapse All	Collapses the Peripheral tree view
	Disable Modes	Resets to “Disabled” all peripherals modes that have been enabled. The pins configured in these modes (green color) are consequently reset to “Unused” (gray color). Peripheral labels change from green to black (when unused) or gray (when not available).
	Expand All	Expands the Peripheral tree view to display all functional modes.
	Zoom in	Zooms in the chip pinout diagram
	Best Fit	Adjusts the chip pinout diagram to the best fit size
	Zoom out	Zooms out the chip pinout diagram
	Keep current signals Placement	Available from toolbar only. Prevents moving pin assignments to match a new peripheral 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. Window menu).

Table 6. Window menu

Name	Description
Outputs	Opens the MCUs selection window at the bottom of STM8CubeMX Main window. Opens two tabs at the bottom of STM8CubeMX main window: • **MCUs selection** tab that lists the MCUs that match the user criteria selected via the MCU selector. • **Outputs** tab that displays STM8CubeMX messages, warnings and errors encountered upon users actions.

4.4.5 Help menu

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

Table 7. **Help menu**

Icons	Name	Description
	Help Content	Opens the STM8CubeMX user manual
	About...	Shows version information
	Check for Updates	Shows the software available for download.
	Updater Settings...	Opens the updater settings window to configure manual versus automatic updates, proxy settings for internet connections.

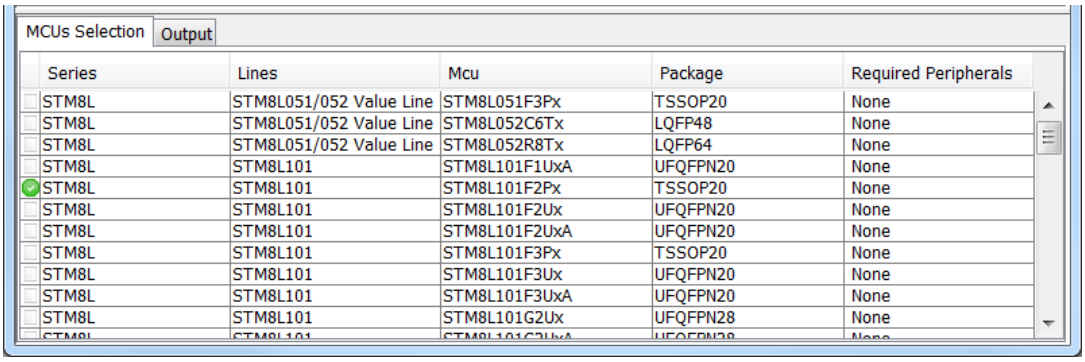
4.5 Output windows

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 29. **MCU selection menu**

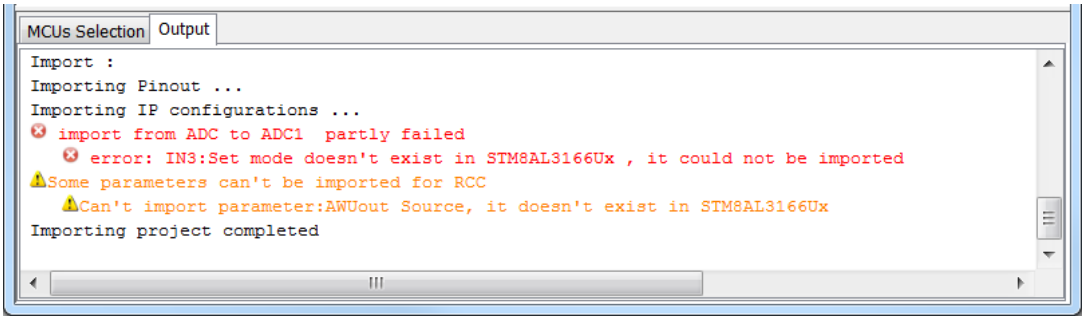


This window can be shown/hidden by selecting/unselecting **Outputs** from the Window menu.

4.5.2 Output pane

This pane displays a non exhaustive list of the actions performed, errors and warnings raised (see Figure 30. Output pane).

Figure 30. **Output pane**



4.6 Import Project window

The **Import Project** menu eases the porting of a previously-saved configuration to another MCU. This menu becomes active when an MCU has been selected and remains so as long as no user action is performed on the project configuration (pinout, clock..).

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.

To import a project, proceed as follows:

1. After starting a new project and once an MCU has been selected, select **File > Import Project** or the **Import project** icon that appears under the **File** menu to open the dedicated **Import project** window.
2. In the dedicated **Import project** window, enter the following information:
   – The STM8CubeMX configuration file (.ioc8) pathname of the project to import on top of current empty project.
   – Whether to import the configuration defined in the Power Consumption Calculator tab or not.
   – Whether to perform an automatic project import (see Figure 31. Automatic project import) or a manual project import. In case of manual import, checkboxes allow manually selecting the set of peripherals (see Figure 32. Manual project import).
3. As 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.
   – The manual import can be used to refine import choices and resolve the issues raised by the import trial. Figure 34. Import Project menu - Successful import after adjustmentsshows how to complete the import successfully, that is, in this example, by unselecting the request for ADC2 and SPI1 imports.
   – The **Show View** function allows switching between the different configuration tabs (pinout, clock tree) for checking the influence of the Try Import action before its actual deployment on current project (see Figure 34. Import Project menu - Successful import after adjustments).
   – The Peripheral List shows:
     ◦ 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; in this case the user needs to choose one
     ◦ warning icons, which indicate that the user has selected a peripheral instance more than once and that one of the import requests will not be performed (Figure 33. Import Project menu - Try import with errors shows an example where the ADC1 instance has been selected twice)
     ◦ Cross signs, which indicate that there is a pinout conflict and that the configuration can not be imported as such (in Figure 33. Import Project menu - Try import with errors, the SPI6 instance configuration can not be imported on SPI3 because it conflicts with the previously selected SPI1 configuration).
4. 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.

Figure 31. **Automatic project import**

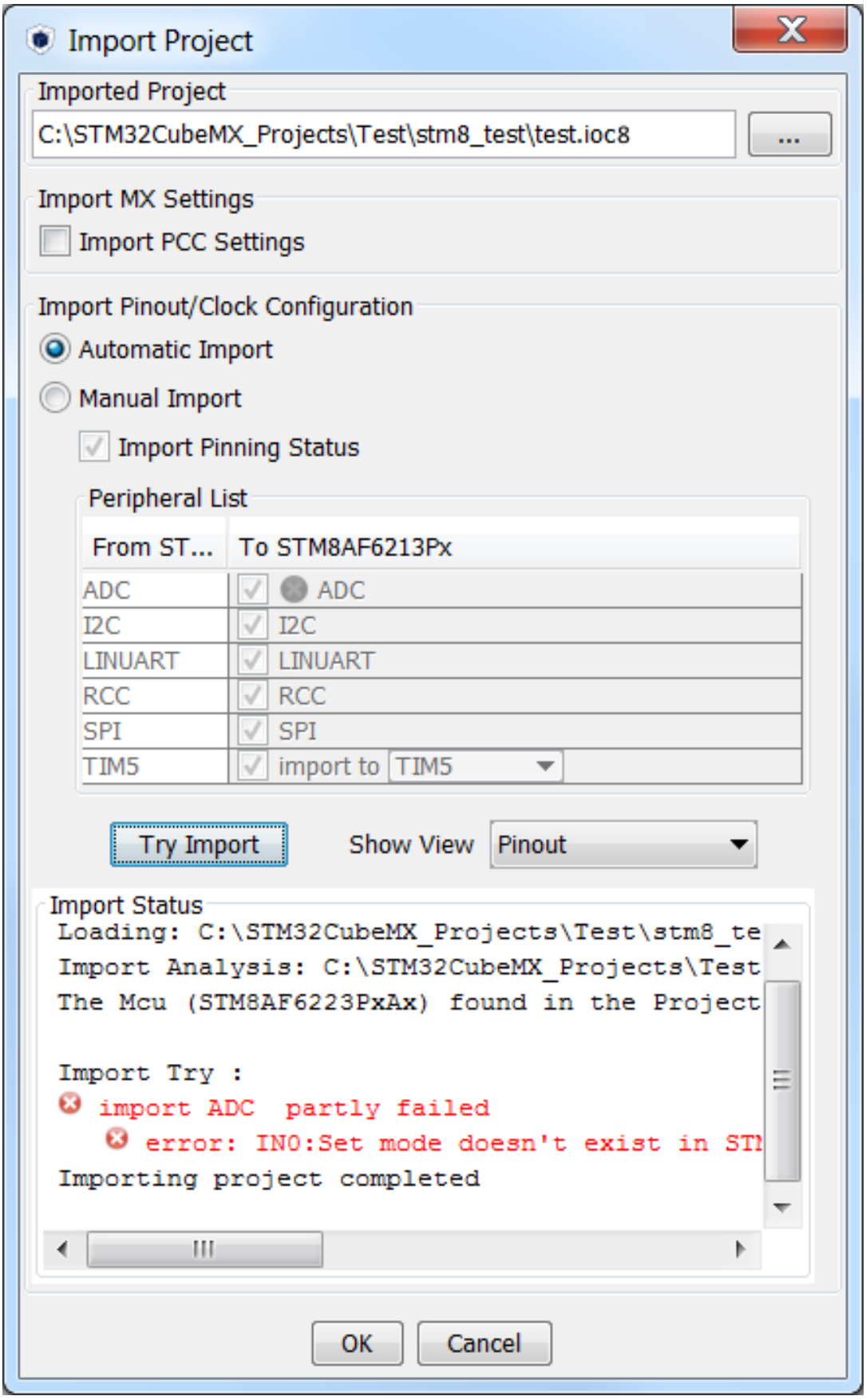


Figure 32. **Manual project import**

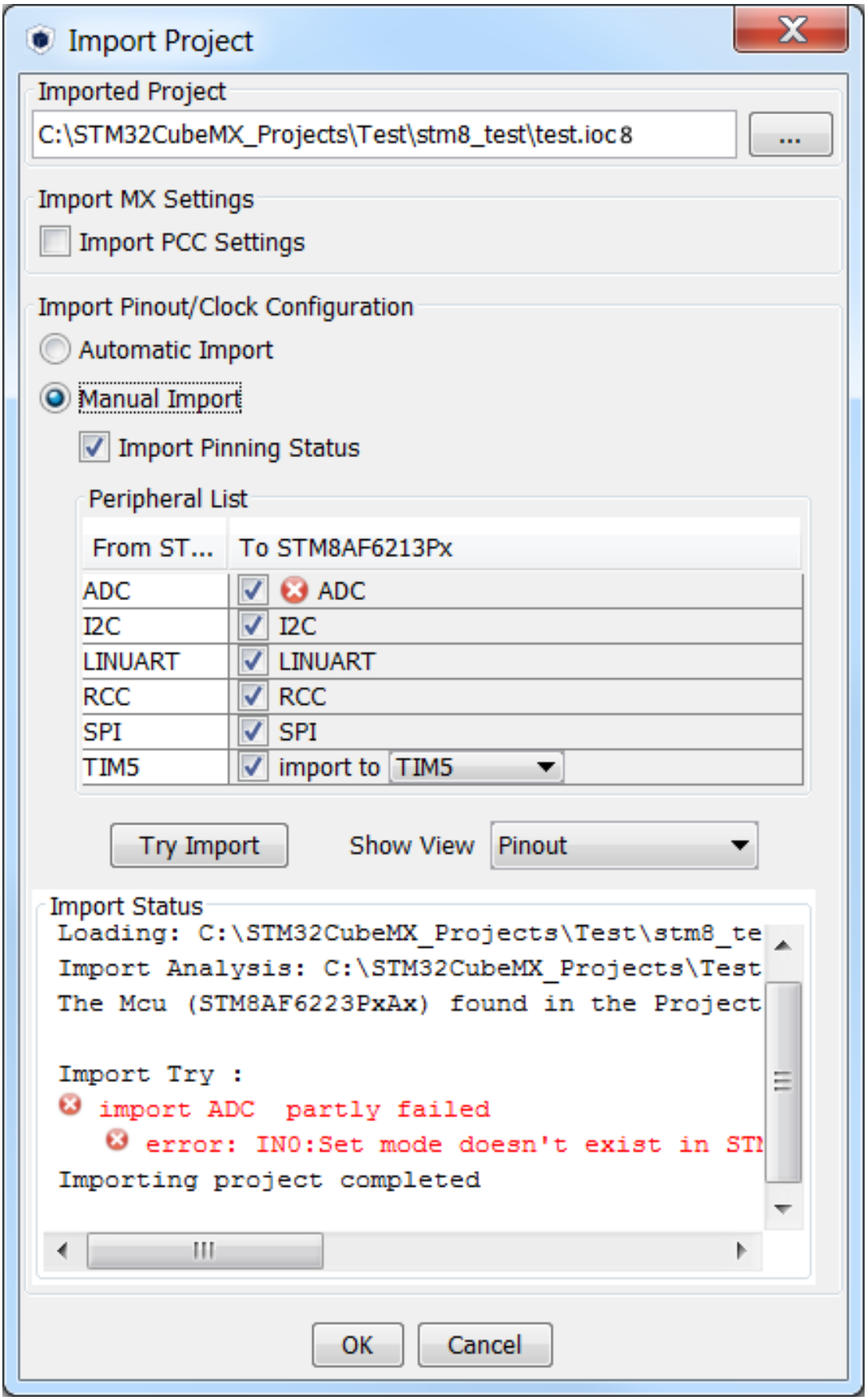


Figure 33. **Import Project menu - Try import with errors**

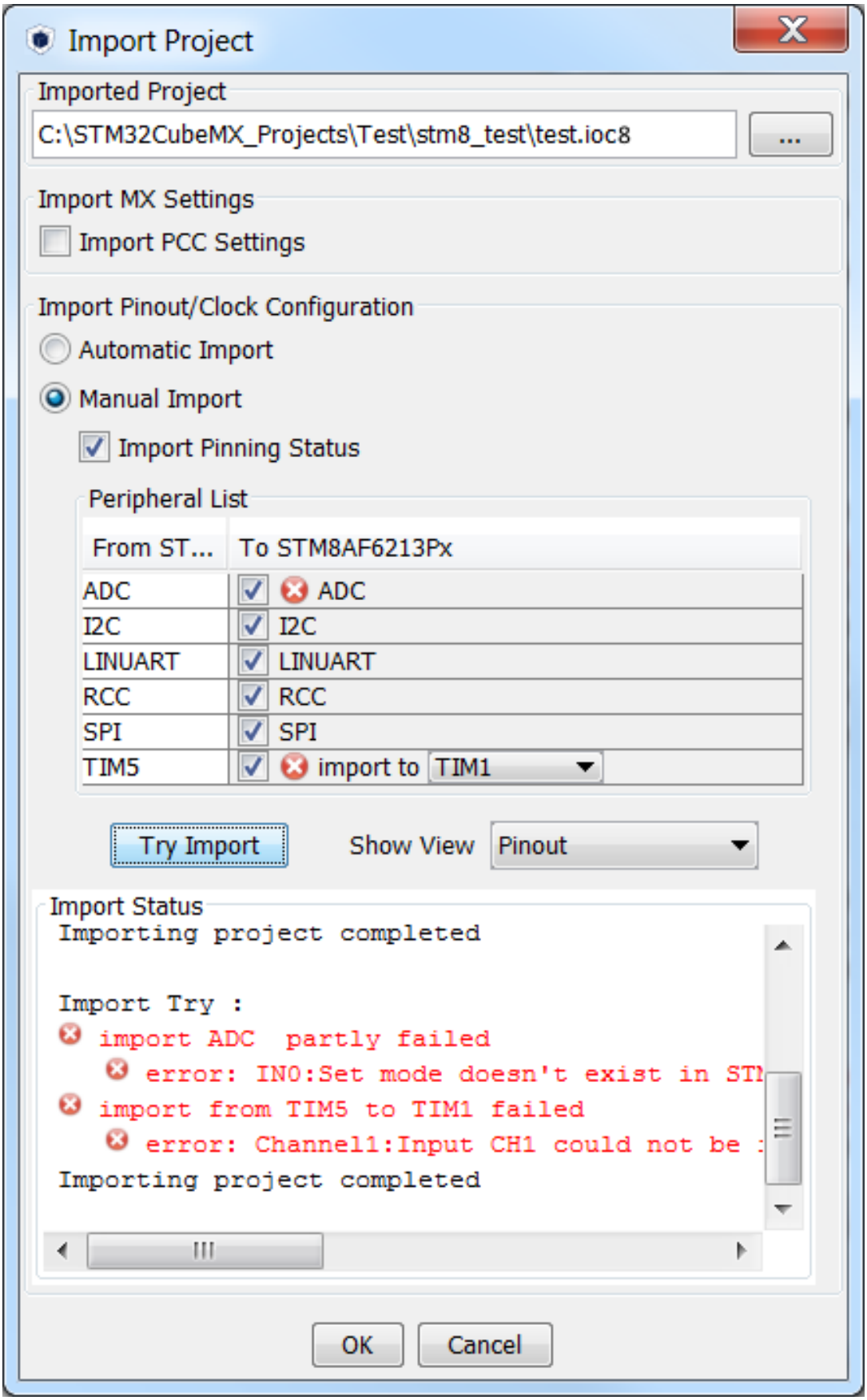
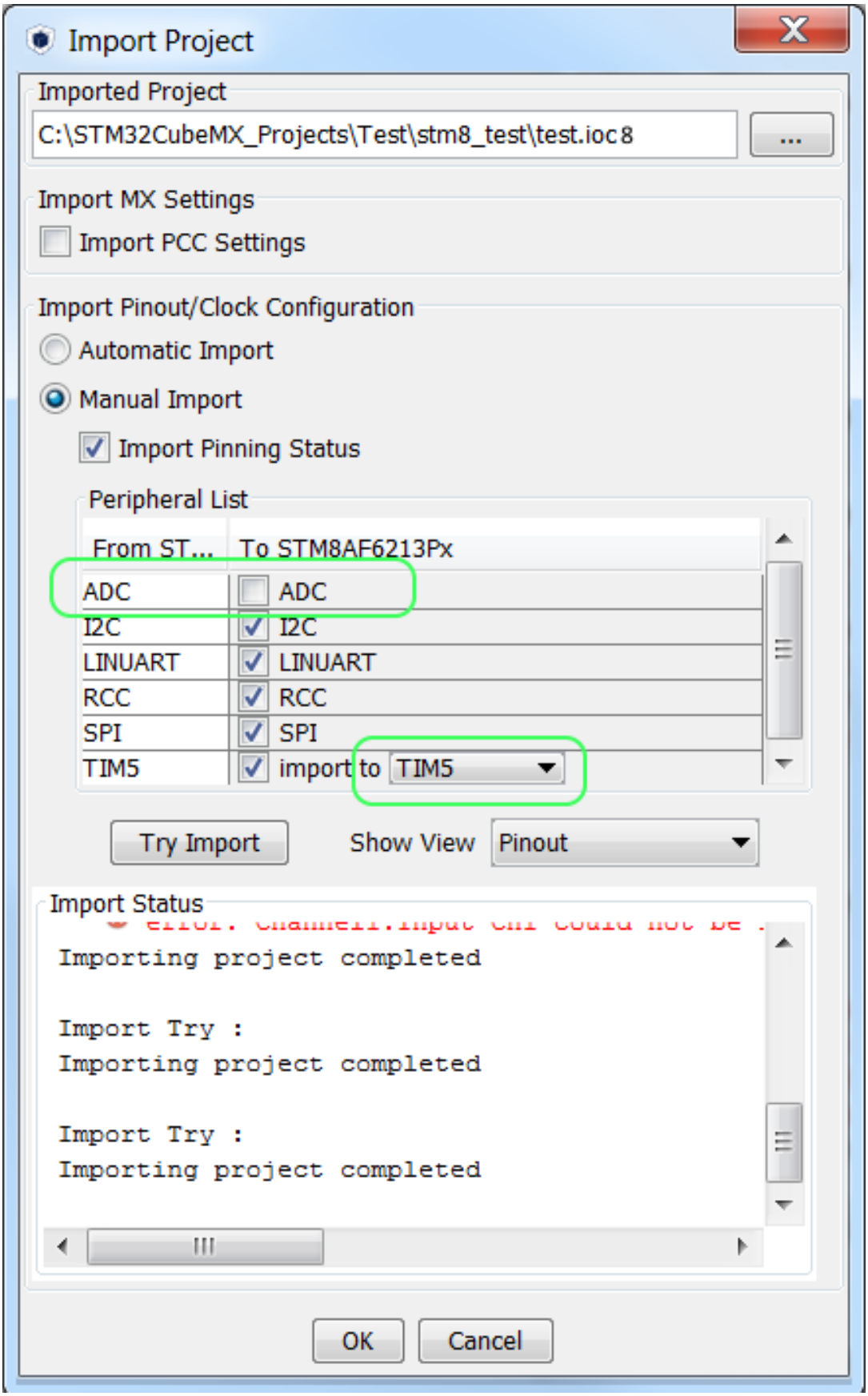


Figure 34. **Import Project menu - Successful import after adjustments**



4.7 Set unused / Reset used GPIOs windows

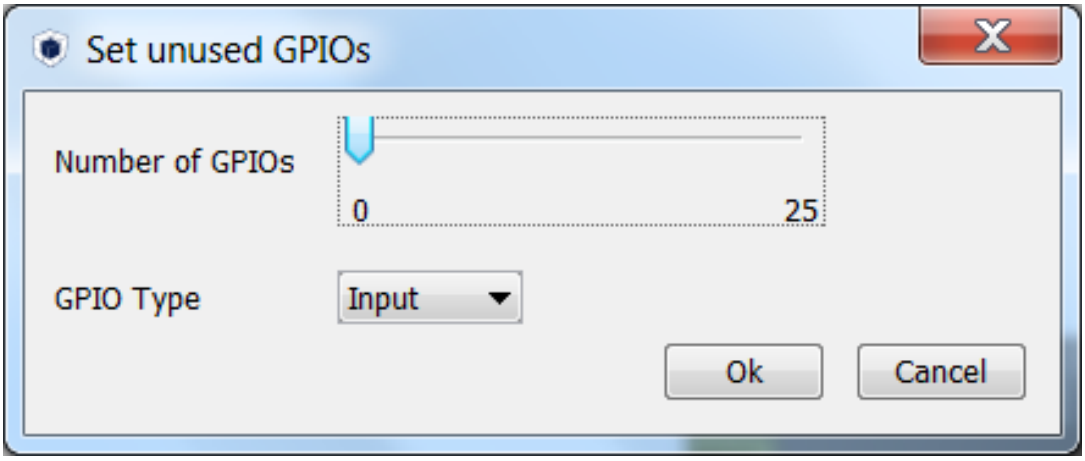
These windows allow configuring several pins at a time in the same GPIO mode.

To open them:

- Select **Pinout**> **Set unused GPIOs** from the STM8CubeMX menu bar.

Note: The user selects the number of GPIOs and lets STM8CubeMX choose the actual pins to be configured or reset, among the available ones.

Figure 35. Set unused pins window



- Select **Pinout > Reset used GPIOs** from the STM8CubeMX menu bar.

Depending whether the Keep Current Signals Placement option is checked or not on the toolbar, STM8CubeMX conflict solver will be able to move or not the GPIO signals to other unused GPIOs:

- When Keep Current Signals Placement is off (unchecked), STM8CubeMX 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. Reset used pins window and Figure 35. Set unused pins window and check the limitation in available peripheral modes.

Figure 36. Reset used pins window

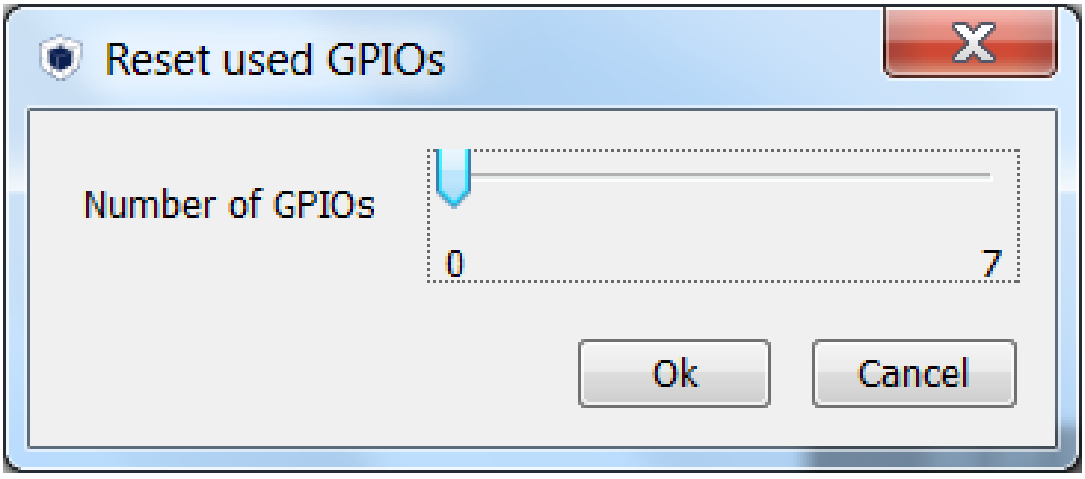


Figure 37. **Set unused GPIO pins with Keep Current Signals Placement checked**

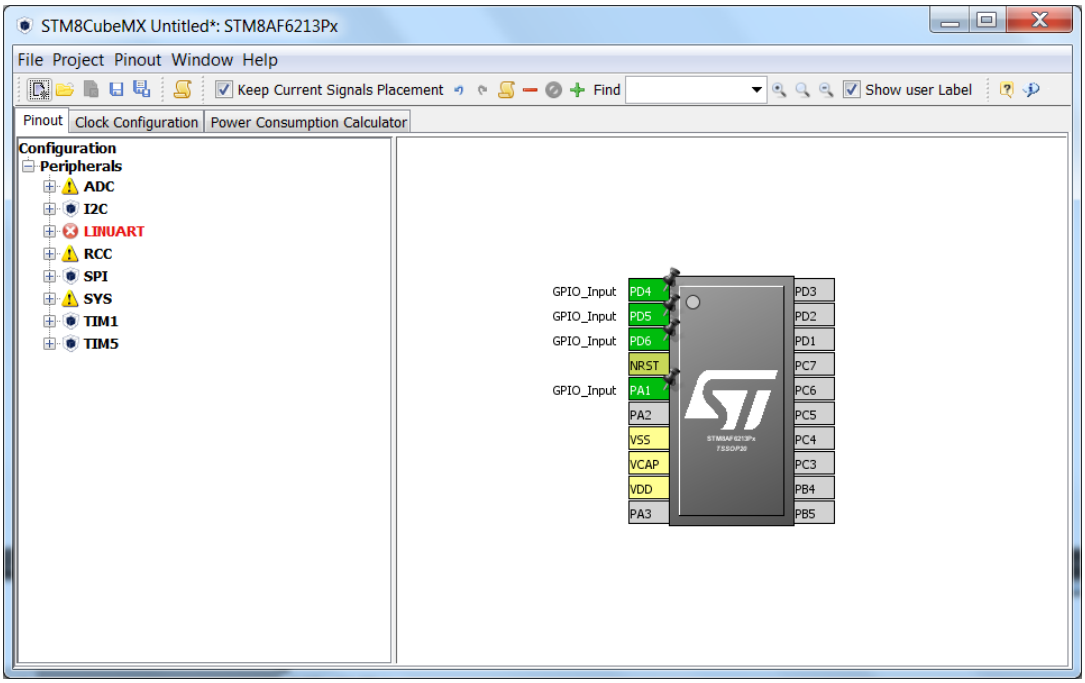
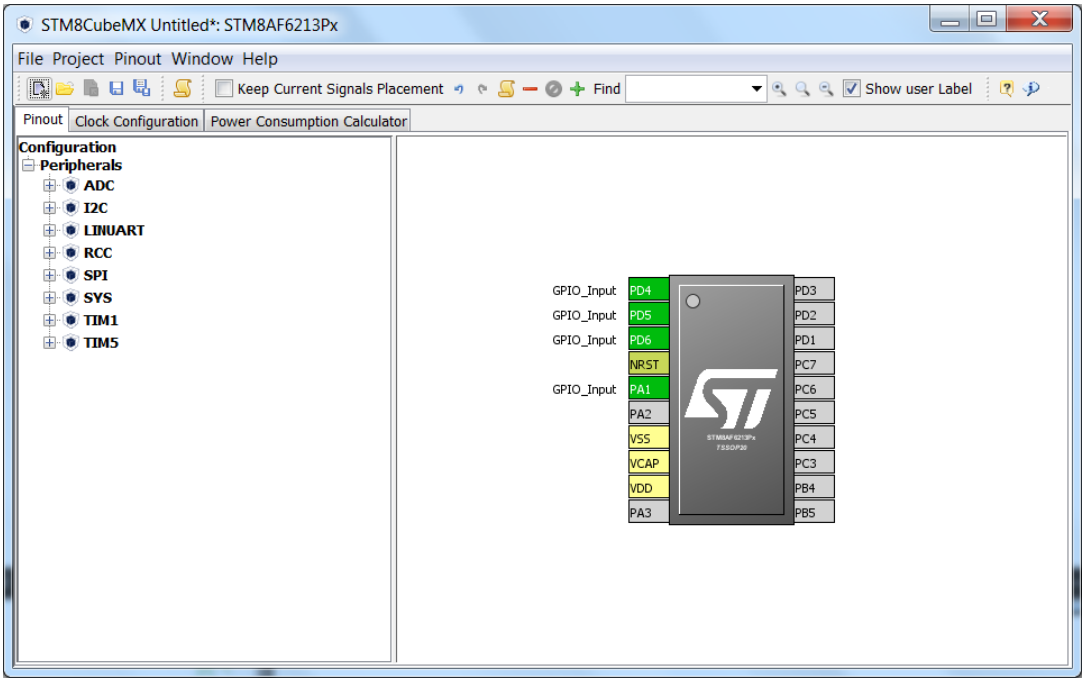


Figure 38. **Set unused GPIO pins with Keep Current Signals Placement unchecked**



4.8 Update Manager windows

Two windows can be accessed through the Help menu available from STM8CubeMX menu bar:

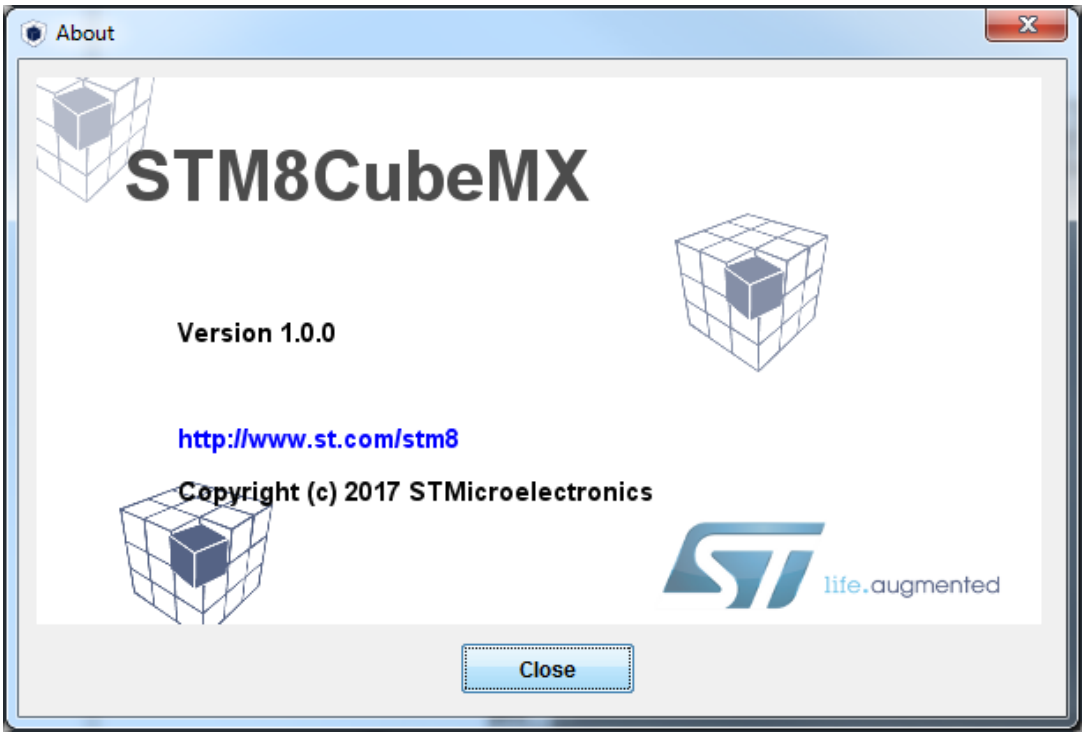
- Select **Help > Check for updates** to open the **Check Update Manager** window and find out about the latest software versions available for download.
- Select **Help > Updater settings** to open the **Updater settings** window and configure update mechanism settings (proxy settings, manual versus automatic updates).

4.9 About window

This window displays STM8CubeMX version information.

To open it, select **Help**> **About** from the STM8CubeMX menu bar.

Figure 39. About window



4.10 Pinout view

The **Pinout** view helps the user configuring the MCU pins based on a selection of peripherals and of their operating modes.

Since STM8 MCUs allow a same pin to be used by different peripherals and for several functions (alternate functions), the tool searches for the pinout configuration that best fits the set of peripherals selected by the user. STM8CubeMX highlights the conflicts that cannot be solved automatically.

The **Pinout** view left panel shows the **Peripheral tree** and the right pane, a graphical representation of the pinout for the selected package (e.g. BGA, QFP...) where each pin is represented with its name (e.g. PC4) and its current alternate function assignment if any.

STM8CubeMX offers two ways to configure the microcontroller:

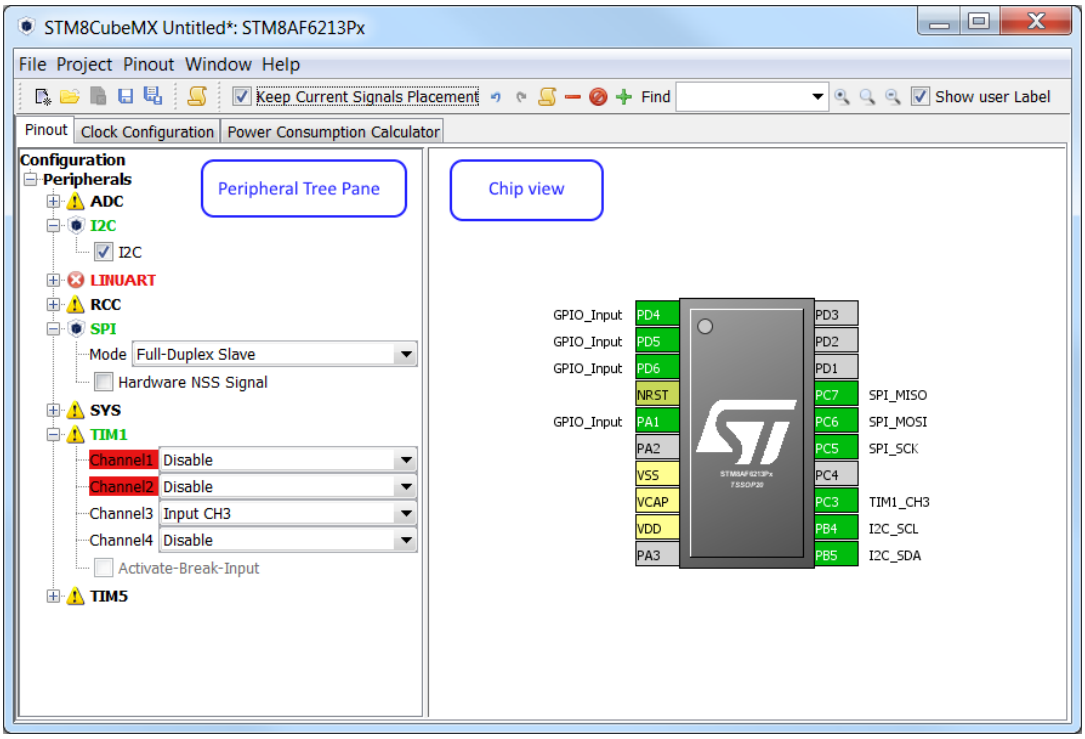
- From the **Peripheral** tree by clicking the peripheral names and selecting the operating modes (see Section 4.10.1 Peripheral tree pane).
- For advanced users, by clicking a pin on the **Chip** view to manually map it to a peripheral function (see Section 4.10.2 Chip view).

In addition, selecting **Pinout > Set unused GPIOs** allows configuring in one shot several unused pins in a given GPIO mode.

*Note: The **Pinout** view is automatically refreshed to display the resulting pinout configuration.*

*Note: Pinout relevant menus and shortcuts are available when the **Pinout** view is active (see the menu dedicated sections for details on the **Pinout** menus).*

Figure 40. **STM8CubeMX Pinout view**



4.10.1 Peripheral tree pane

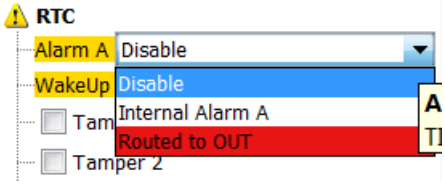
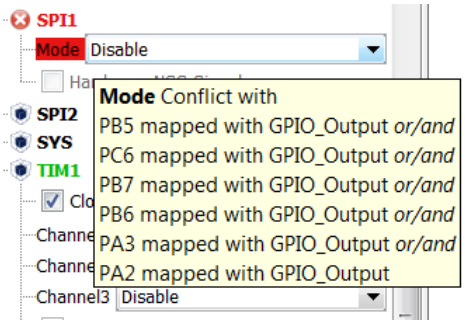
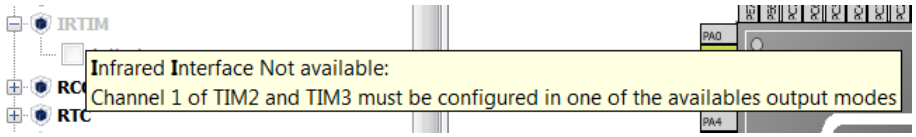
In this pane, the user can select the peripherals in the modes required for the application.

Icons and color schemes

Table 8. Peripheral - icons and color scheme shows the icons and color scheme used in the **Peripheral tree** pane.

Table 8. **Peripheral - icons and color scheme**

Display	Peripheral status
SPI Mode Disable Hardware NSS Signal	The peripheral is not configured (no mode is set) and all modes are available.
LINUART Mode Asynchronous	The peripheral is configured (at least one mode is set) and all other modes are available
RCC	The peripheral is configured (one mode is set) and at least one of its other modes is unavailable.
TIM3	The peripheral is not configured (no mode is set) and at least one of its modes is unavailable.
LINUART	The peripheral is not configured (no mode is set) and no mode is available. Move the mouse over the peripheral name to display the tooltip describing the conflict.

Display	Peripheral status
	Available peripheral mode configurations are shown in plain black.
	The warning yellow icon indicates that at least one mode configuration is no longer available.
	When no more configurations are left for a given peripheral mode, this peripheral is highlighted in red.
	Tooltips are available to indicate the conflict.
	Some peripheral modes depend on the configuration of other peripheral modes. A tooltip describes the dependency when conditions are not fulfilled.

4.10.2 Chip view

The **Chip** view shows, for the selected part number:

- The MCU in a specific package (LQFP, WLCSP…)
- The graphical representation of its pinout, each pin being represented with its name (e.g. PC7: pin 7 of GPIO port C) and its current function assignment (e.g. SPI_MISO) (see Figure 51. Chip viewfor an example).

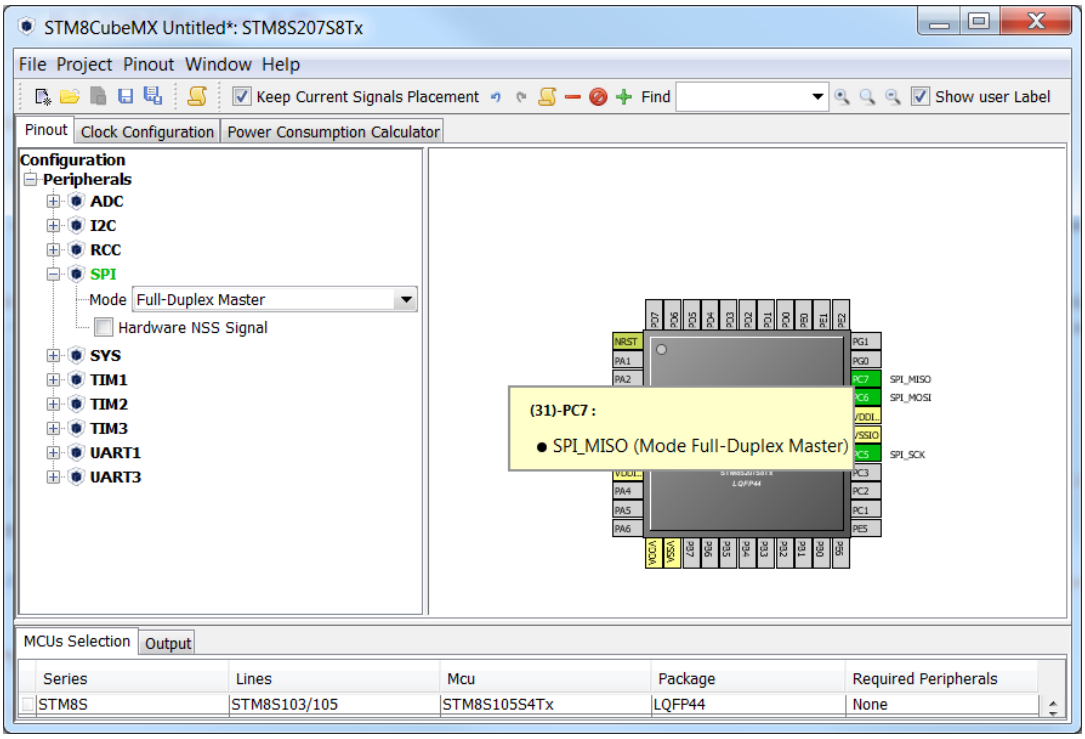
The **Chip** view is automatically refreshed to match the user configuration performed via the peripheral tree. It shows the pins current configuration state.

Assigning pins through the **Chip** view instead of the peripheral pane requires a good knowledge of the MCU since each individual pin can be assigned to a specific function.

Tips and tricks

- Use the mouse wheel to zoom in and out.
- Click and drag the chip diagram to move it. Click **best fit** to reset it to best suited position and size (see Table 5. Pinout menu).
- Use **Pinout > Generic CSV pinout text file** to export the pinout configuration into text format.
- Some basic controls, such as insuring blocks of pins consistency, are built-in. See Section 7 STM8CubeMX pin assignment rules for details.

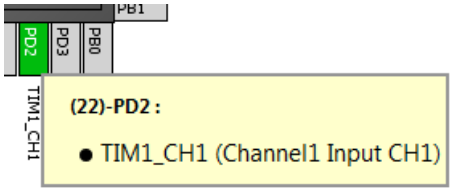
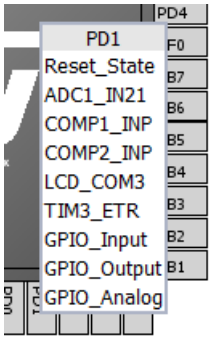
Figure 51. **Chip view**

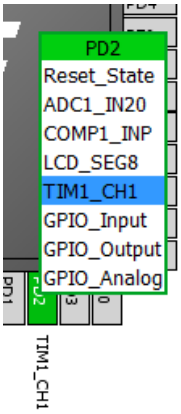
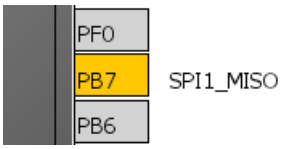
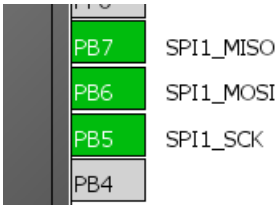


Icons and color schemes

Table 9. STM8CubeMX Chip view - Icons and color scheme shows the icons and color scheme used in the **Chip** view.

Table 9. **STM8CubeMX Chip view - Icons and color scheme**

Display	Pin information
	**Tooltip** indicates the selected pin current configuration: alternate function name, Reset state or GPIO mode. Move your mouse over the pin name to display it. When a pin features alternate pins corresponding to the function currently selected, a popup message prompts the user to perform a CTRL + click to display them. The alternate pins available are highlighted in blue.
	**List** of alternate functions that can be selected for a given pin. By default, no alternate function is configured (pin in reset state). Click the pin name to display the list.

Display	Pin information
	When a function has been mapped to the pin, it is highlighted in blue. When it corresponds to a well configured peripheral mode, the list caption is shown in green.
	Boot and reset pins are highlighted in khaki. Their configuration cannot be changed.
	Power dedicated pins are highlighted in yellow. Their configuration cannot be changed.
	Non-configured pins are shown in gray (default state).
	When a signal assignment corresponds to a peripheral mode without ambiguity, the pin color switches to green.
	When the signal assignment does not correspond to a valid peripheral mode configuration, the pin is shown in orange. Additional pins need to be configured to achieve a valid mode configuration.
	When a signal assignment corresponds to a peripheral mode without ambiguity, the pins are shown in green.

Tooltips

On the peripheral tree view, move the mouse over peripherals and peripheral modes that are unavailable or partially available to display the tooltips describing the source of the conflict: which pins are already being used and by which peripherals.

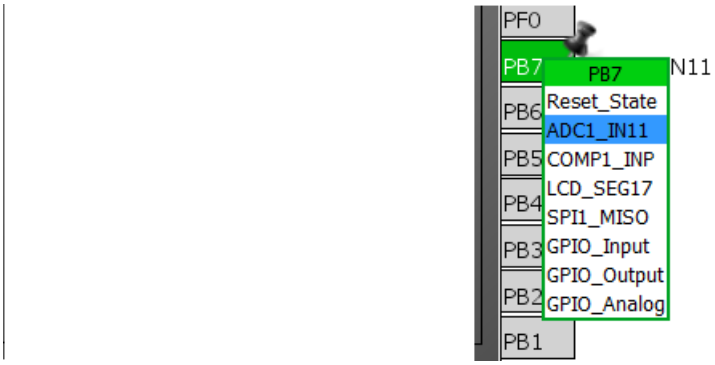
4.10.3 Chip view advanced actions

Manually modifying pin assignments

To manually modify a pin assignment, follow the sequence below:

1. Click the pin in the **Chip** view to display the list of all other possible alternate functions together with the current assignment highlighted in blue (see Figure 61. Modifying pin assignments from the Chip view).
2. Click to select the new function to assign to the pin.

Figure 61. **Modifying pin assignments from the Chip view**

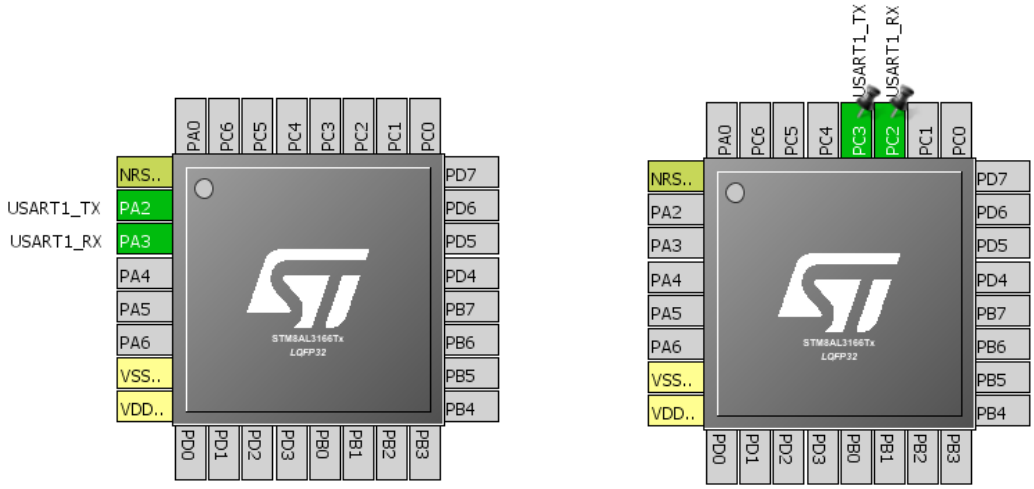


Manually remapping a function to another pin

To manually remap a function to another pin, follow the sequence below:

1. Press the CTRL key and click the pin in the **Chip** view. Possible pins for relocation, if any, are highlighted in blue.
2. Drag the function to the target pin.

Figure 62. **Modifying pin assignments from the Chip view**

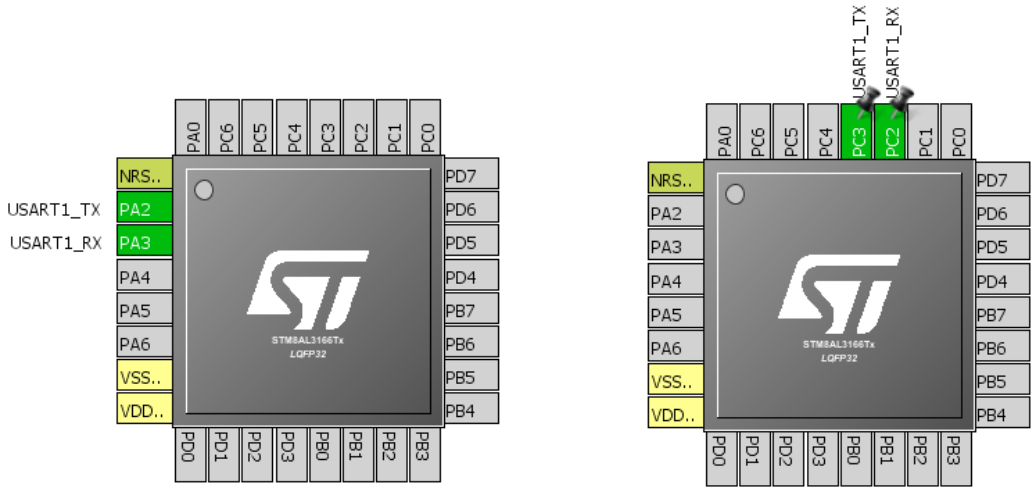


Caution: A pin assignment performed from the Chip view overwrites any previous assignment.

Manual remapping with destination pin ambiguity

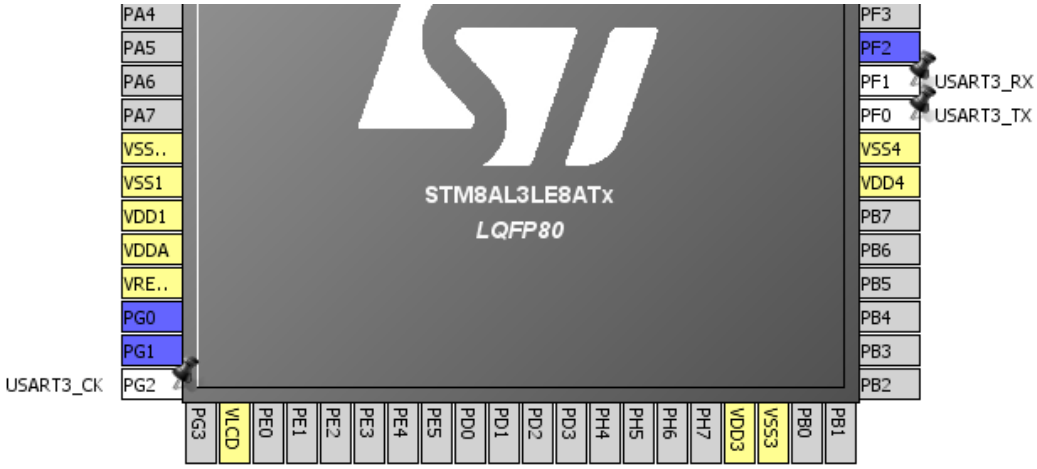
A "block of pins" is a group of pins that must be assigned together to achieve a given peripheral mode (see Figure 63. Example of remapping of a block of pins).

Figure 63. **Example of remapping of a block of pins**



For MCUs with block of pins consistency, the destination pin can be ambiguous,e.g. there can be more than one destination block including the destination pin. To display all the possible alternative remapping blocks, move the mouse over the target pin.

Figure 64. **Example of remapping with destination pin ambiguity**



Resolving pin conflicts

To resolve the pin conflicts that may occur when some peripheral modes use the same pins, STM8CubeMX attempts to reassign the peripheral mode functions to other pins. The peripherals for which pin conflicts could not be solved are highlighted in red or orange with a tooltip describing the conflict.

If the conflict cannot be solved by remapping the modes, the user can try the following:

- If the 

box is checked, try to select the peripherals in a different sequence.
- Uncheck the box and let STM8CubeMX try all the remap combinations to find a solution.
- **Manually remap** a mode of a peripheral when you cannot use it because there is no pin available for one of the signals of that mode.

4.10.4 Keep Current Signals Placement

This checkbox is available from the toolbar when the **Pinout** view is selected (see Figure 15. Pinout menus (Pinout tab selected) and Table 5. Pinout menu). It can be selected or unselected at any time during the configuration. It is unselected by default.

It is recommended to keep the checkbox unchecked for an optimized placement of the peripherals (maximum number of peripherals concurrently used).

The **Keep Current Signals Placement** checkbox should be selected when the objective is to match a board design.

Keep Current Signals Placement is unchecked

This allows STM8CubeMX to remap previously mapped blocks to other pins in order to serve a new request (selection of a new peripheral mode or a new peripheral mode function) which conflicts with the current pinout configuration.

Keep Current Signals Placement is checked

This ensures that all the functions corresponding to a given peripheral mode remain allocated (mapped) to a given pin. Once the allocation is done, STM8CubeMX cannot move a peripheral mode function from one pin to another. New configuration requests are served if it is feasible within current pin configuration.

This functionality is useful to:

- Lock all the pins corresponding to peripherals that have been configured using the **Peripherals** panel.
- Maintain a function mapped to a pin while doing manual remapping from the **Chip** view.

Tip

If a mode becomes unavailable (highlighted in red), try to find another pin remapping configuration for this mode by following the steps below:

1. From the **Chip** view, unselect the assigned functions one by one until the mode becomes available again.
2. Then, select the mode again and continue the pinout configuration with the new sequence (see Section 7 STM8CubeMX pin assignment rules for a remapping example). This operation being time consuming, it is recommended to unselect the **Keep Current Signals Placement** checkbox.

Note: Even if Keep Current Signals placement is unchecked, GPIO_ functions are not moved by STM8CubeMX.

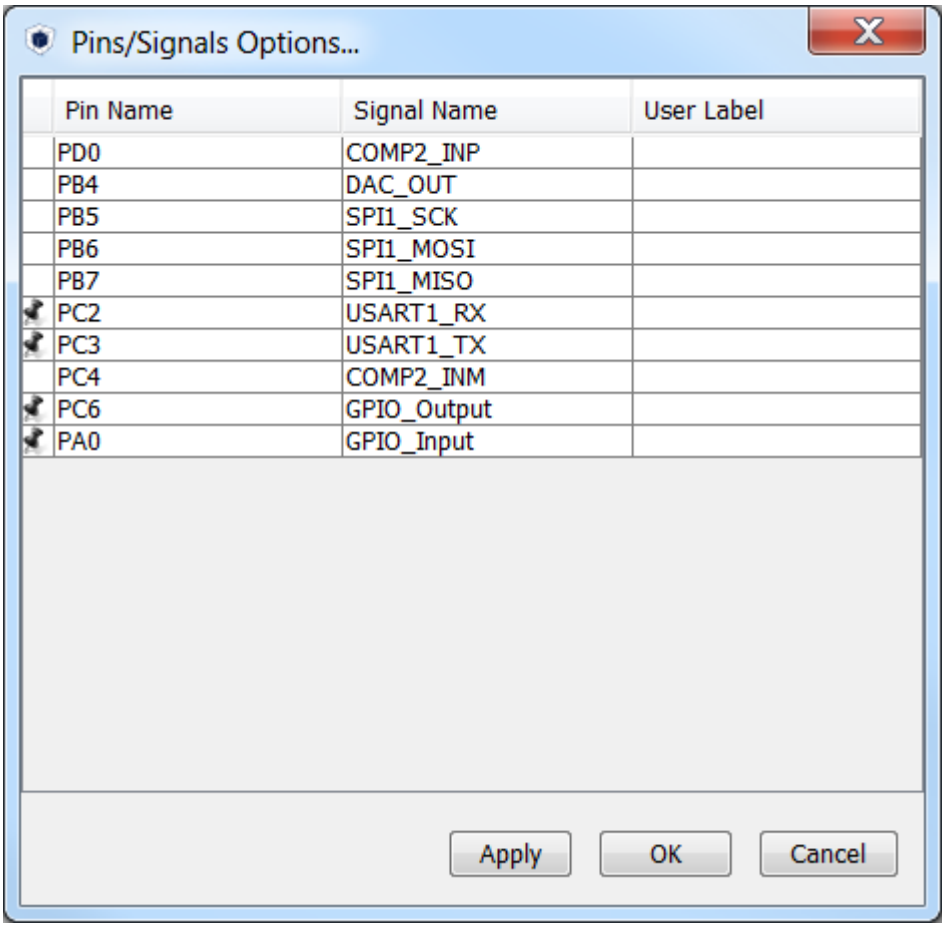
4.10.5 Pinning and labeling signals on pins

STM8CubeMX comes with a feature allowing the user to selectively lock (or pin) signals to pins: This will prevent STM8CubeMX from automatically moving the pinned signals to other pins when resolving conflicts. There is also the possibility to label the signals: user labels will be available in generated reports.

There are several ways to pin, unpin and label the signals:

1. From the **Chip** view, right-click a pin with a signal assignment. This opens a contextual menu:
   a. For unpinned signals, select **Signal Pinning** to pin the signal. A pin icon is then displayed on the relevant pin. The signal can no longer be moved automatically (for example when resolving pin assignment conflicts).
   b. For pinned signals, select **Signal Unpinning** to unpin the signal. The pin icon is removed. From now on, to resolve a conflict (such as peripheral mode conflict), this signal can be moved to another pin, provided the Keep user placement option is unchecked.
   c. Select **Enter User Label** to specify a user defined label for this signal. The new label will replacing the default signal name in the **Chipview**.
2. From the **Pinout** menu, select **Pins/Signals Options**
   The Pins/Signals Options window (see Figure 65. Pins/Signals Options window) lists all configured pins.
   a. Click the first column to individually pin/unpin signals.
   b. Select multiple rows and right-click to open the contextual menu and select Signal(s) Pinning or Unpinning.
   c. Select the User Label field to edit the field and enter a user-defined label.
   d. Order list alphabetically by Pin or Signal name by clicking the column header. Click once more to go back to default i.e. to list ordered according to pin placement on MCU.

Figure 65. **Pins/Signals Options window**



Note: Even if a signal is pinned, it is still possible however to manually change the pin signal assignment from the Chip view: click the pin to display other possible signals for this pin and select the relevant one.

4.10.6 Listing pinout-compatible MCUs

STM8CubeMX allows listing the MCUs which pinout is compatible with the current project.

1. Start from a given project (see Figure 66. STM8CubeMX Project example).
2. Select the pinout tab. Then from the **Pinout** menu select **List Pinout Compatible MCUs** to open the dedicated **Pinout Compatible Dialog** window (see Figure 67. Pinout Compatible Dialog window): fully pinout compatible MCUs are highlighted in green. On the left panel, a package criteria is set to match the current project.
3. Changing the package or series criteria as well as the search options, then clicking the Search button, refreshes the list of compatible MCUs according to the choices made.

   When the package criteria is set to All, some additional MCUs are listed. They are highlighted in yellow since they do not offer a full pinout match with the current project (see Figure 68. Full and partial pinout compatibility). The user will have to perform manual configurations to migrate to a non-fully compatible MCU.
4. Select an MCU from this list and click **Import** (see Figure 69. Selecting a compatible MCU).

   This migrates the current pinout configuration to the selected MCU. STM8CubeMX view and project are updated for the new MCU (see Figure 70. Importing current configuration to a compatible MCU).

   The list of compatible MCUs is shown on the **Compatible MCUs** tab and can be hidden by unchecking **Outputs** under the **Window** menu. Double-clicking an MCU from this list automatically imports the current configuration on this MCU and refreshes the project accordingly .

Figure 66. **STM8CubeMX Project example**

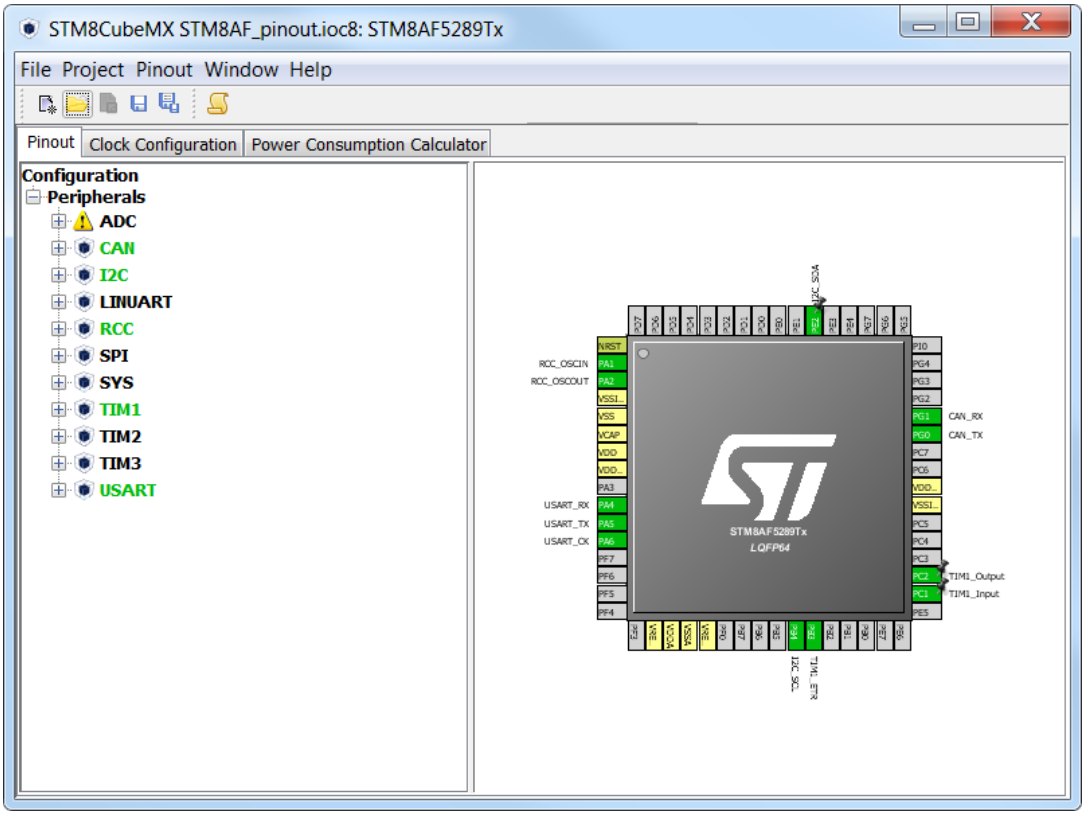


Figure 67. **Pinout Compatible Dialog window**

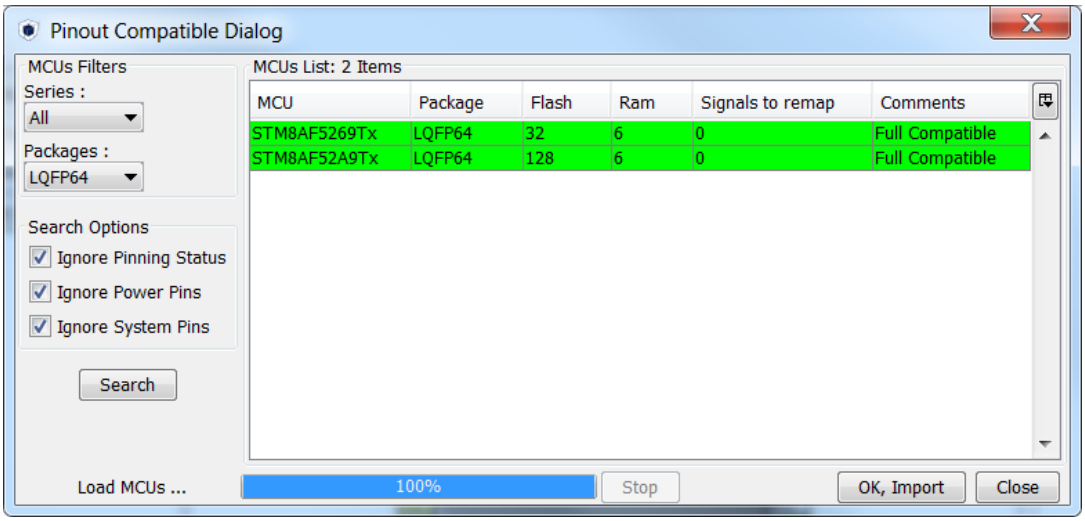


Figure 68. **Full and partial pinout compatibility**

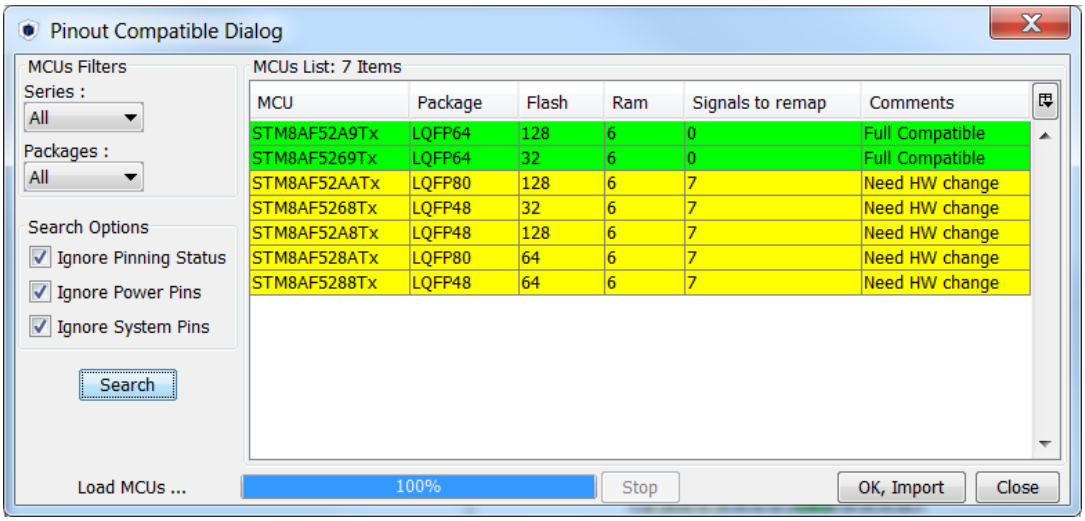


Figure 69. **Selecting a compatible MCU**

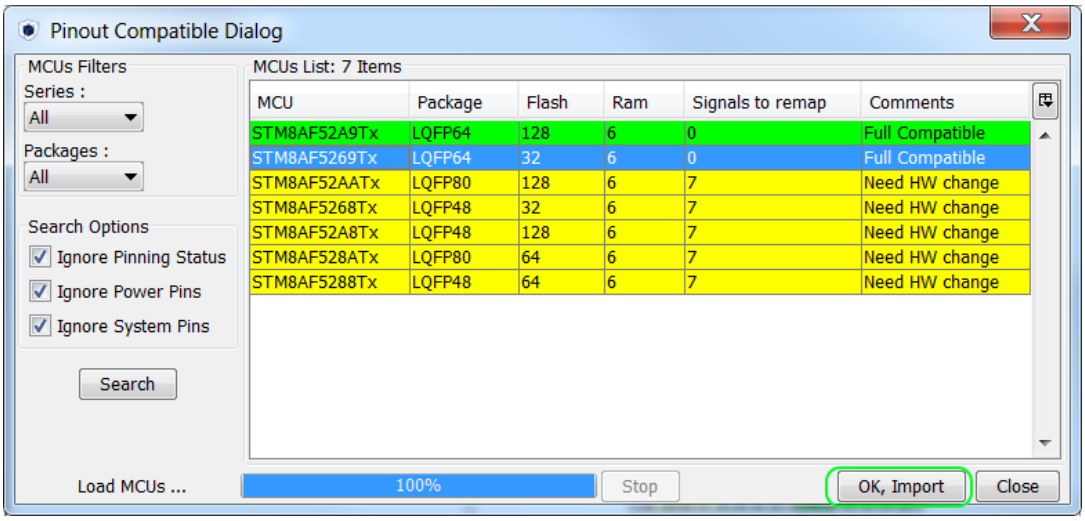
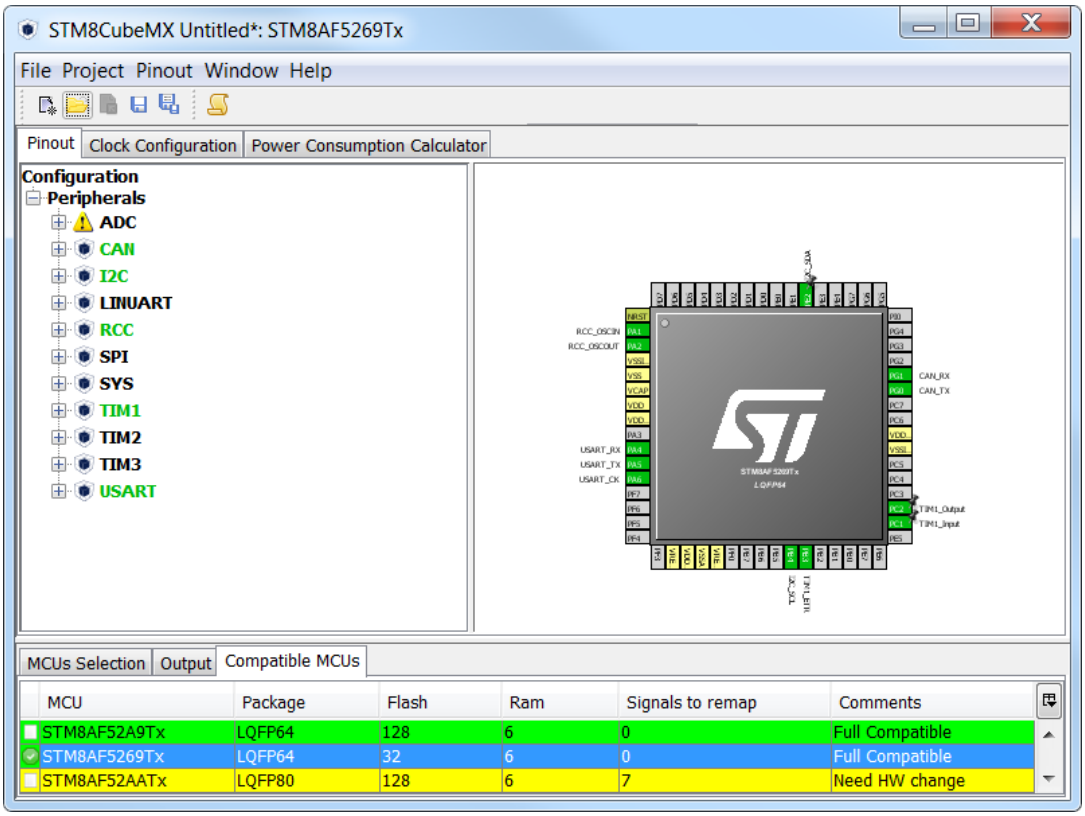


Figure 70. **Importing current configuration to a compatible MCU**



4.11 Clock tree configuration view

STM8CubeMX **Clock configuration** window (see Figure 71. STM8AL3166Tx Clock Tree configuration view) provides a schematic overview of the clock paths, clock sources, dividers, and multipliers. Drop-down menus and buttons allow modifying the actual clock tree configuration to meet user application requirements.

Actual clock speeds are displayed and active. The clock signals that are used are highlighted in blue.

Out-of-range configured values are highlighted in red to flag potential issues. A solver feature is proposed to automatically resolve such configuration issues (see Figure 72. Clock Tree configuration view with errors).

Reverse path is supported: just enter the required clock speed in the blue filed and STM8CubeMX will attempt to reconfigure multipliers and dividers to provide the requested value. The resulting clock value can then be locked by right clicking the field to prevent modifications.

4.11.1 Clock tree configuration functions

External clock sources

When external clock sources are used, the user must previously enable them from the **Pinout** view available under the RCC peripheral.

Peripheral clock configuration options

Some other paths, corresponding to clock peripherals, are grayed out. To become active, the peripheral must be properly configured in the **Pinout** view (e.g. RTC).

This view allows to:

- Enter a frequency value for the CPU Clock (HCLK), buses or peripheral clocks

  STM8CubeMX tries to propose a clock tree configuration that reaches the desired frequency while adjusting prescalers and dividers and taking into account other peripheral constraints (such as USB clock minimum value). If no solution can be found, STM8CubeMX proposes to switch to a different clock source or can even conclude that no solution matches the desired frequency.

- Lock the frequency fields for which the current value should be preserved

  Right click a frequency field and select **Lock** to preserve the value currently assigned when STM8CubeMX will search for a new clock configuration solution.

  The user can unlock the locked frequency fields when the preservation is no longer necessary.
- Select the clock source that will drive the system clock (SYSCLK)
  - External oscillator clock (HSE) for a user defined frequency.
  - Internal oscillator clock (HSI) for the defined fixed frequency.
  - Main PLL clock
- Select secondary sources (as available for the product)
  - Low-speed internal (LSI) or external (LSE) clock
  - I2S input clock
  - …
- Select prescalers, dividers and multipliers values.
- Enable the Clock Security system (CSS) on HSE when it is supported by the MCU

  This feature is available only when the HSE clock is used as the system clock source directly or indirectly through the PLL. It allows detecting HSE failure and inform the software about it, thus allowing the MCU to perform rescue operations.
- Enable the CSS on LSE when it is supported by the MCU

  This feature is available only when the LSE and LSI are enabled and after the RTC or LCD clock sources have been selected to be either LSE or LSI.
- **Reset the Clock tree default settings by using the toolbar Reset button** ( )

  This feature reloads STM8CubeMX default clock tree configuration.
- **Undo/Redo user configuration steps by using the toolbar** Undo/Redo buttons ( )
- Detect and resolve configuration issues

  Errors are highlighted in red and the **Clock Configuration** view is marked with a red cross (see Figure 72. Clock Tree configuration view with errors).

  Issues can be resolved manually or automatically by clicking the **Resolve Clock Issue** button ( ) which is enabled only if issues have been detected.

The tool will automatically perform the following operations:

- Adjust bus frequencies and master output clocks according to user selection of clock sources, clock frequencies and prescalers/multipliers/dividers values.
- Check the validity of user settings.
- Highlight invalid settings in red and provide tooltips to guide the user to achieve a valid configuration.

The Clock tree view is adjusted according to the RCC settings configured in the pinout view:

- If in RCC **Pinout** view, the external and output clocks are enabled, they become configurable in the clock tree view.

Figure 71. STM8AL3166Tx Clock Tree configuration view gives an example of Clock tree configuration view for an STM8ALx MCU and Table 10. Clock tree view widget describes the widgets that can be used to configure each clock.

Figure 71. **STM8AL3166Tx Clock Tree configuration view**

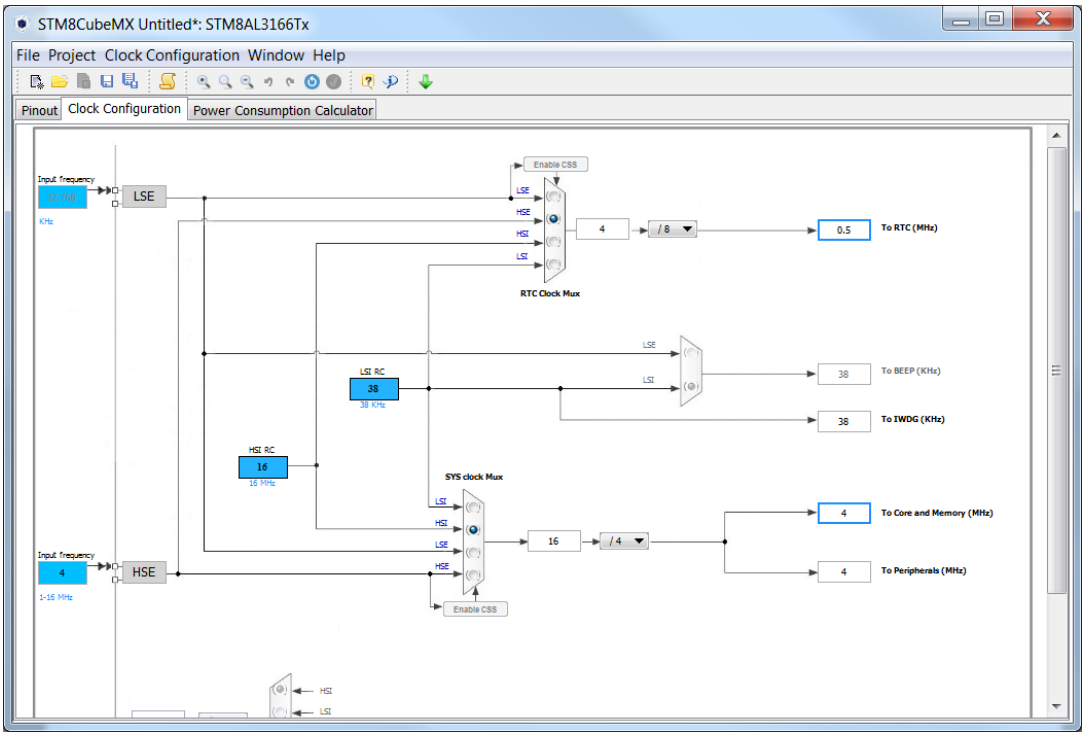


Figure 72. **Clock Tree configuration view with errors**

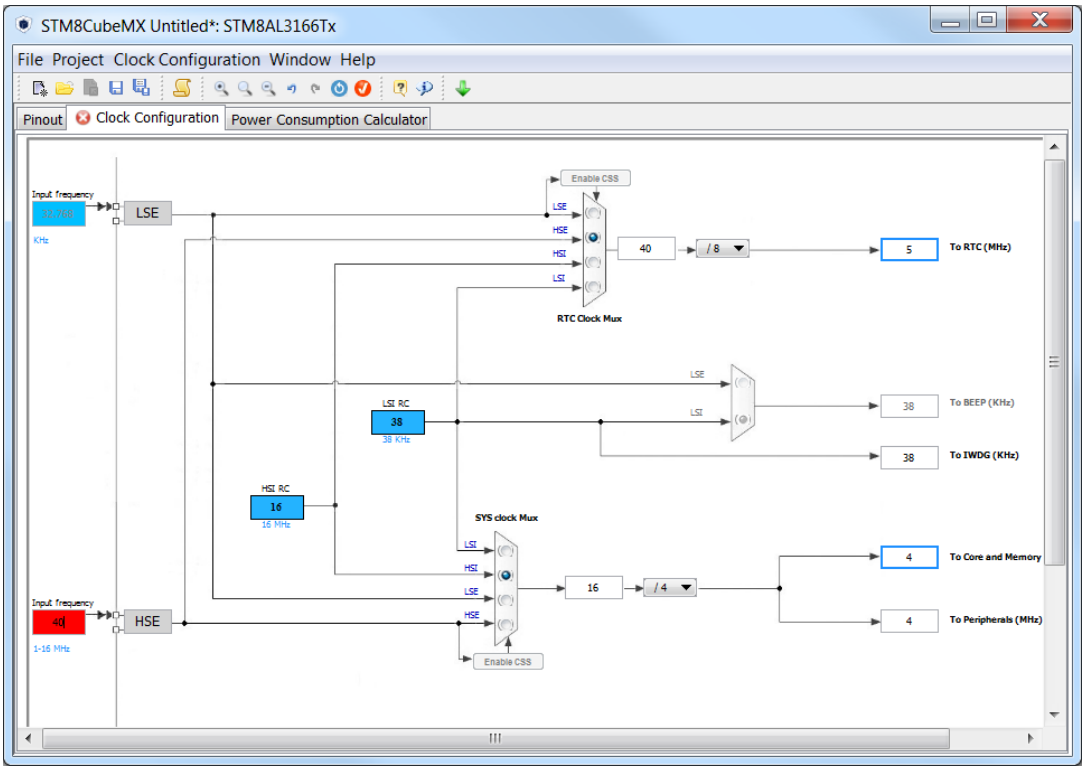
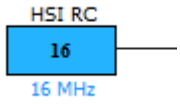
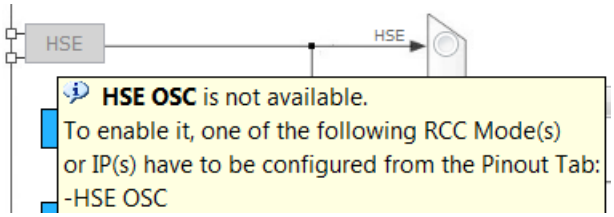
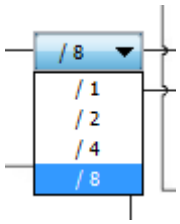
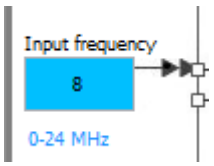
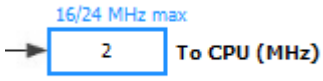
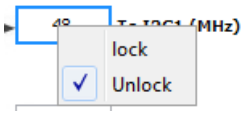


Table 10. Clock tree view widget

Format	Configuration status of the Peripheral Instance
	Active clock sources
	Unavailable settings are blurred or grayed out (clock sources, dividers,…)
	Gray drop down lists for prescalers, dividers, multipliers selection.
	User defined frequency values
	Automatically derived frequency values
	User-modifiable frequency field
	Right click blue border rectangles, to lock/unlock a frequency field. Lock to preserve the frequency value during clock tree configuration updates.

4.12 Power Consumption Calculator view

For an ever-growing number of embedded systems applications, power consumption is a major concern. To help minimizing it, STM8CubeMX offers the **Power Consumption** Calculator tab (see Figure 80. Power Consumption Calculator default view), which, given a microcontroller, a battery model and a user-defined power sequence, provides the following results:

- Average current consumption

  Power consumption values can either be taken from the datasheet or interpolated from a user specified bus or core frequency.
- Battery life
- Average DMIPs

  DMIPs values are directly taken from the MCU datasheet and are neither interpolated nor extrapolated.

- Maximum ambient temperature (T_{AMAX})

  According to the chip internal power consumption, the package type and a maximum junction temperature of 105 °C, the tool computes the maximum ambient temperature to ensure good operating conditions.

  Current T_{AMAX} implementation does not account for I/O consumption. For an accurate T_{AMAX} estimate, I/O consumption must be specified using the Additional Consumption field. The formula for I/O dynamic current consumption is specified in the microcontroller datasheet.

The **Power Consumption Calculator** view allows developers to visualize an estimate of the embedded application consumption and lower it further at each power sequence step:

- Adjust clock sources and frequencies based on the step requirements.
- Enable the peripherals necessary for each phase.

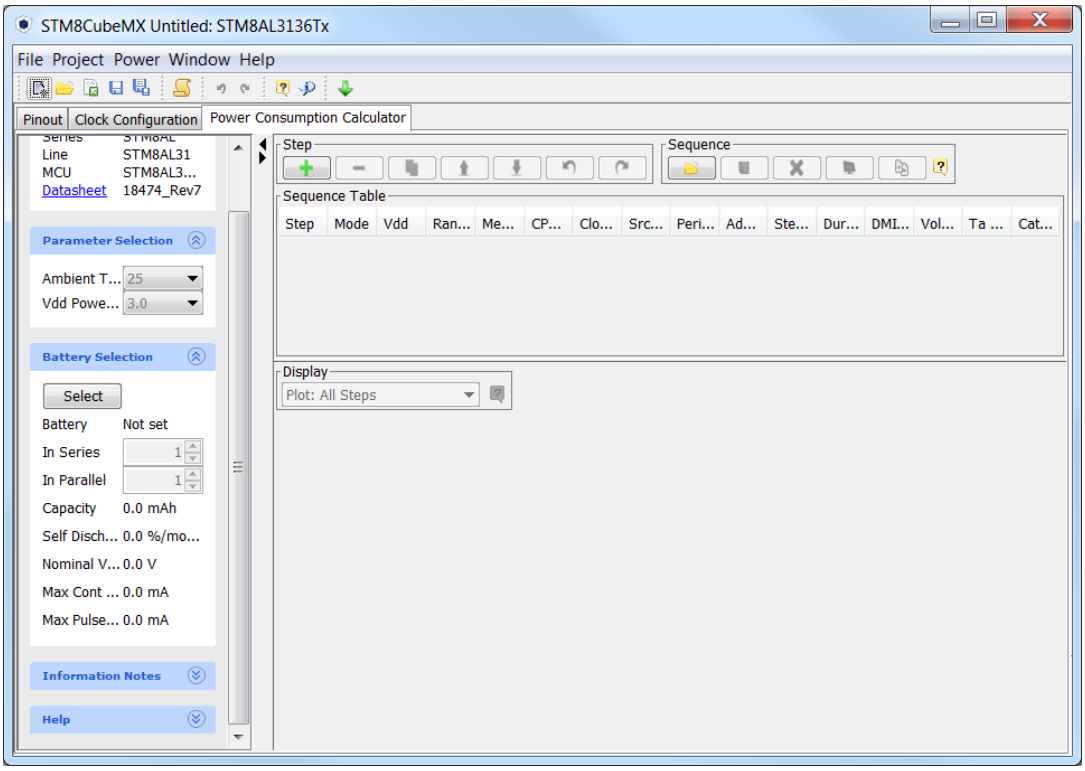
If power consumption measurements are available at different voltage levels, STM8CubeMX will also propose a choice of voltage values (see Figure 83. Power consumption sequence: new step default view).

4.12.1 Building a power consumption sequence

Power sequence default view

The default starting view is shown in Figure 80. Power Consumption Calculator default view.

Figure 80. Power Consumption Calculator default view



Selecting a V_{DD} value

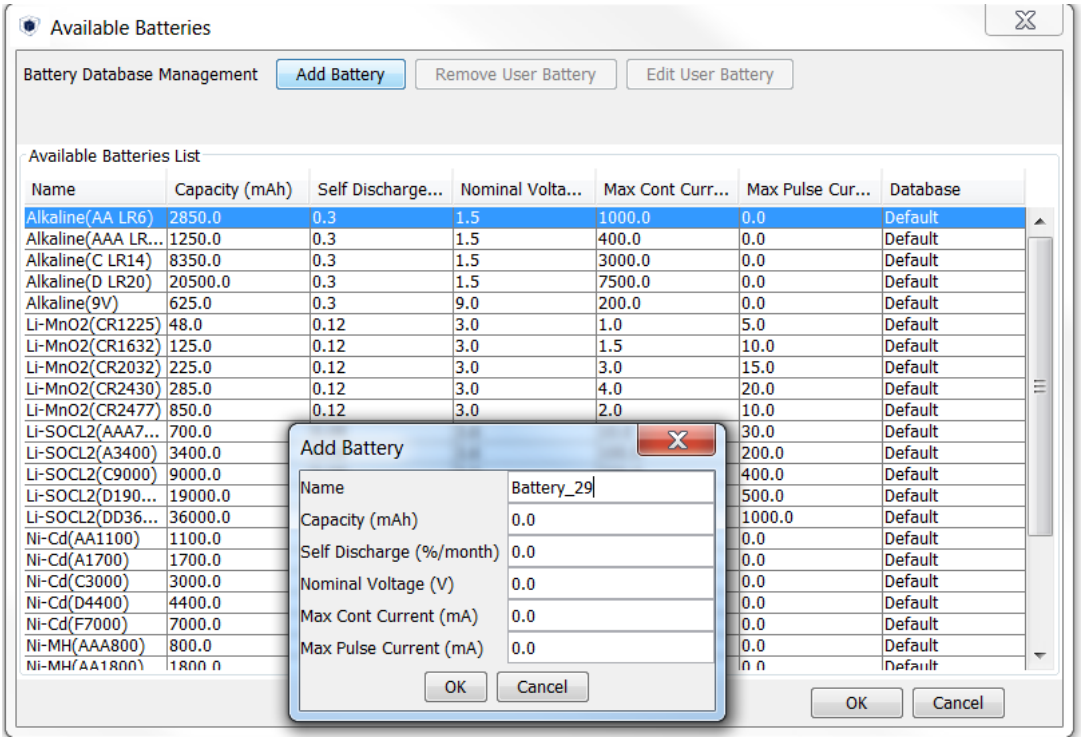
From this view and when multiple choices are available, the user must select a V_{DD} value.

Selecting a battery model (optional)

Optionally, the user can select a battery model. This can also be done once the power consumption sequence is configured.

The user can select a predefined battery or choose to specify a new battery that best matches his application (see Figure 81. Battery selection).

Figure 81. **Battery selection**

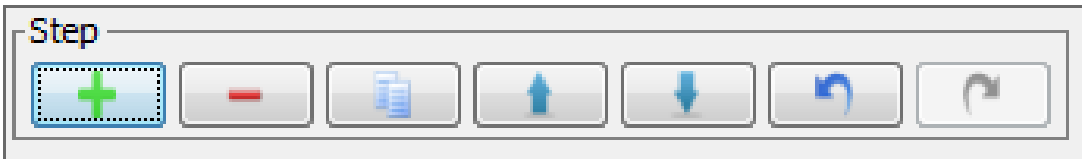


Managing sequence steps

Steps can be reorganized within a sequence (**Add** new, **Delete** a step, **Duplicate** a step, move **Up** or **Down** in the sequence) using the set of Step buttons (see Figure 82. Step management functions).

The user can undo or redo the last configuration actions by clicking the **Undo** button in the Power Consumption Calculator view or the Undo icon from the main toolbar

Figure 82. **Step management functions**

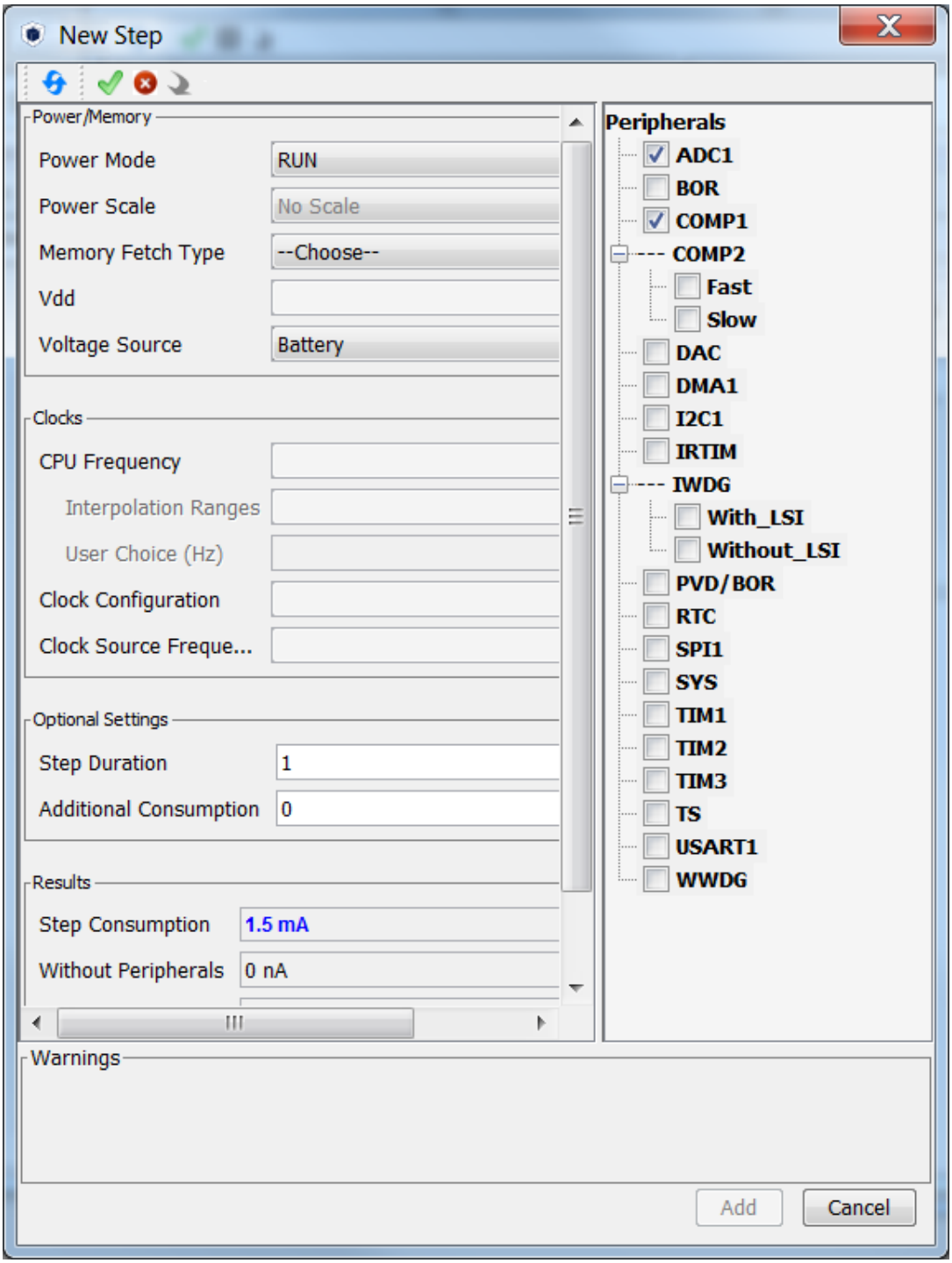


Adding a step

There are two ways to add a new step:

- Click **Add** in the Power Consumption panel. The **New Step** window opens with empty step settings.
- Or, select a step from the sequence table and click **Duplicate**. A **New Step** window opens duplicating the step settings. (see Figure 83. Power consumption sequence: new step default view).

Figure 83. **Power consumption sequence: new step default view**

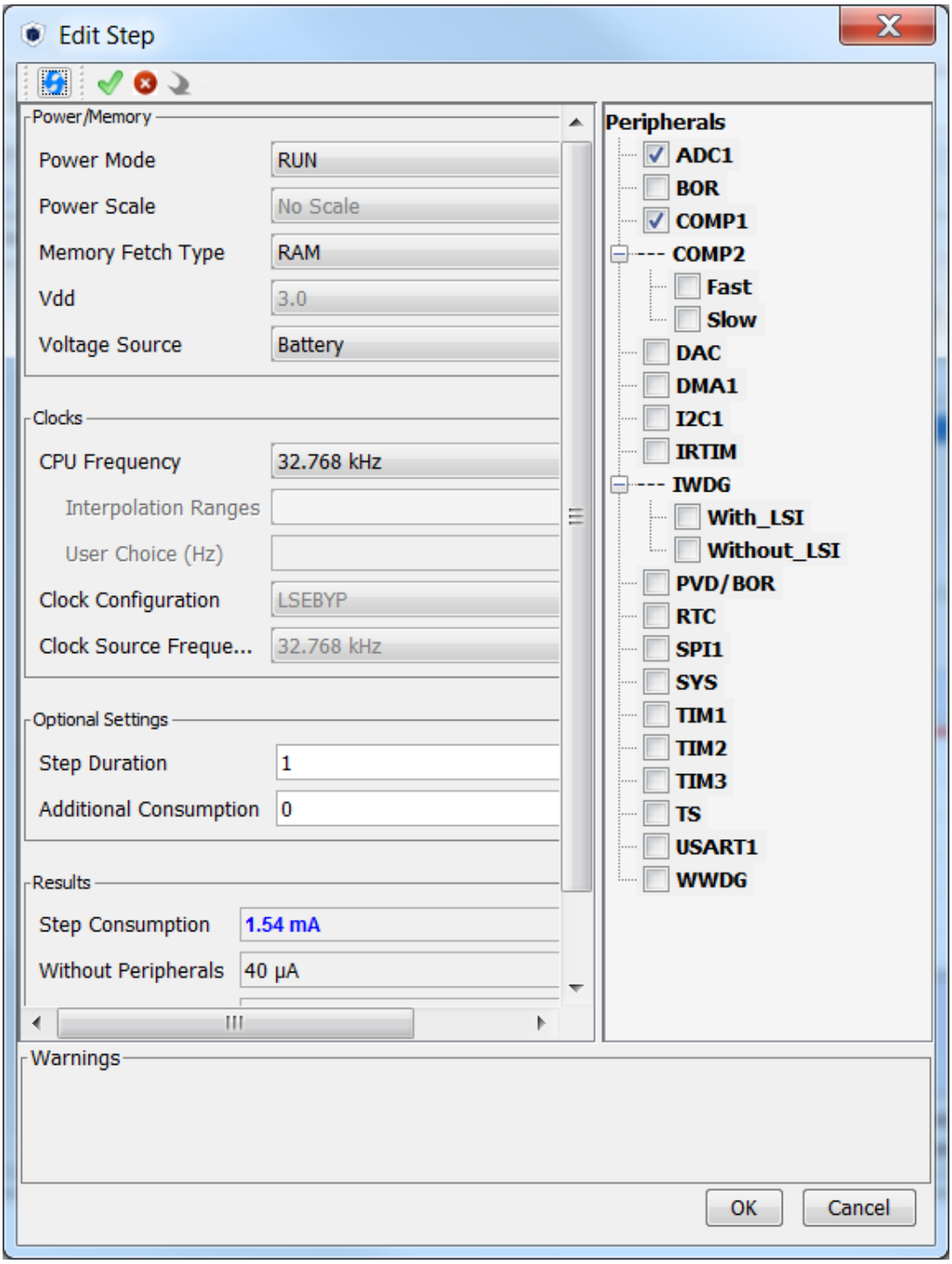


Once a step is configured, resulting current consumption and T_{AMAX} values are provided in the window.

Editing a step

To edit a step, double-click it in the sequence table. The **Edit Step** window opens (see Figure 84. Edit Step window).

Figure 84. **Edit Step window**



Moving a step

By default, a new step is added at the end of a sequence.

Click the step in the sequence table to select it and use the **Up** and **Down** buttons to move it elsewhere in the sequence.

Deleting a step

Select the step to be deleted and click the **Delete** button.

4.12.2 Configuring a step in the power sequence

The step configuration is performed from the **Edit Step** and **New Step** windows. The graphical interface guides the user by forcing a predefined order for setting parameters.

Their naming may differ according to the selected MCU series. For details on each parameter, refer to Section 4.12.4 Power sequence step parameters glossary glossary and to the electrical characteristics section of the MCU datasheet.

The parameters are set automatically by the tool when there is only one possible value (in this case, the parameter cannot be modified and is grayed out). The tool proposes only the configuration choices relevant to the selected MCU.

Proceed as follow to configure a new step:

1. Click **Add** or **Duplicate** to open the **New step** window or double-click a step from the sequence table to open the **Edit step** window.
2. Within the open step window, select in the following order:
   - The **Power Mode**

     Changing the Power Mode resets the whole step configuration.
   - The **Peripherals**

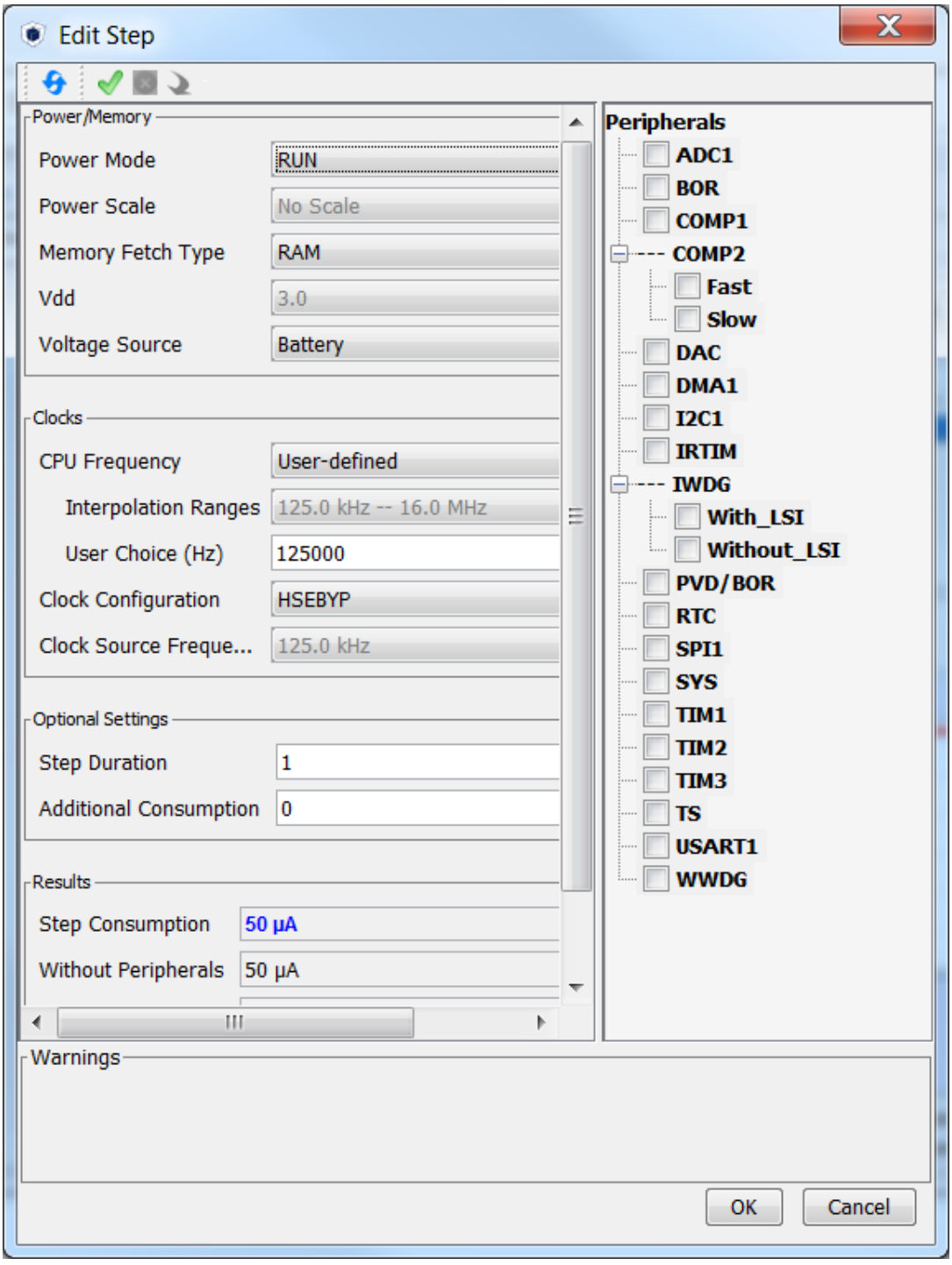
     Peripherals can be selected/unselected at any time after the Power Mode is configured.
   - The **Memory Fetch Type**
   - The V_{DD} value if multiple choices available
   - The voltage source (battery or VBUS)
   - A **Clock Configuration**

     Changing the Clock Configuration resets the frequency choices further down.
   - When multiple choices are available, the CPU Frequency (STM8F4) and the AHB Bus Frequency/CPU Frequency (STM8L1) or, for active modes, a user specified frequency. In this case, the consumption value will be interpolated.
   - Depending on the selected mode, multiple choices can be available for the clock source configuration and frequencies. When the CPU frequency is defined by the user, the consumption may result from an interpolation instead of a value published in datasheet.
3. Optionally set
   - A **step duration** (1 ms is the default value)
   - An **additional consumption** value (expressed in mA) to reflect, for example, external components used by the application (external regulator, external pull-up, LEDs or other displays). This value added to the microcontroller power consumption will impact the step overall power consumption.
4. Once the configuration is complete, the Add button becomes active. Click it to create the step and add it to the sequence table.

Using interpolation

For steps configured for active modes (Run, Sleep), frequency interpolation is supported by selecting CPU frequency as User Defined and entering a frequency in Hz (see Figure 85. Interpolated Power Consumption).

Figure 85. **Interpolated Power Consumption**

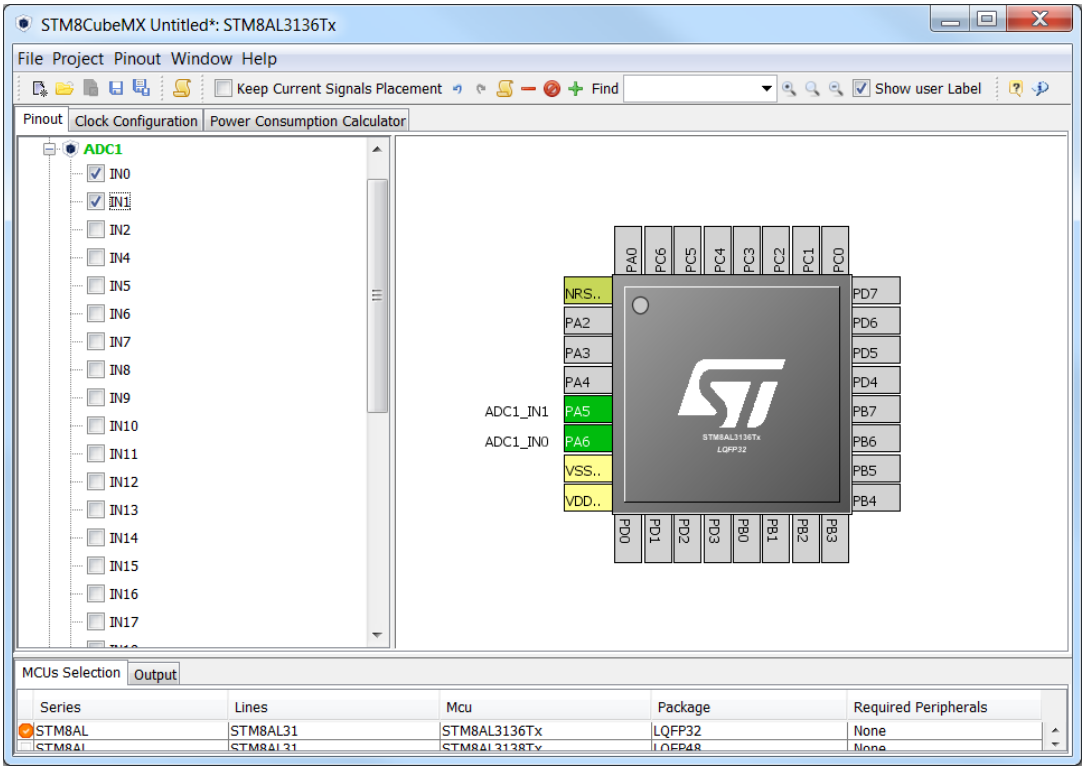


Importing pinout

Figure 86. ADC selected in Pinout view illustrates the example of the ADC configuration in the **Pinout** view: clicking **Import Pinout** in the Power Consumption Calculator view selects the ADC peripheral (Figure 87. Power Consumption Calculator Step configuration window: ADC enabled using the import pinout function).

The **Import pinout** button allows to automatically select the peripherals that have been configured in the **Pinout** view.

Figure 86. **ADC selected in Pinout view**

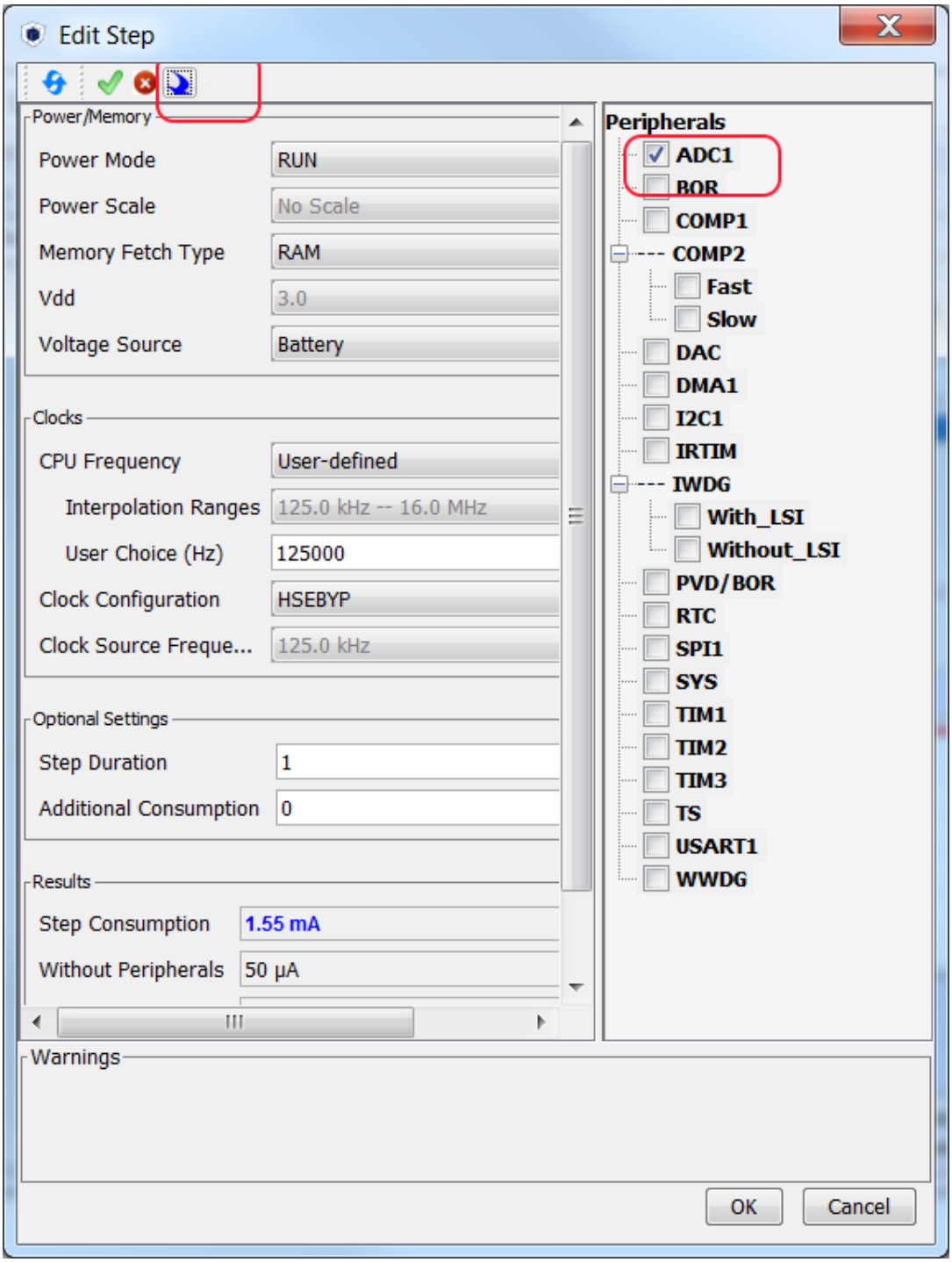


Selecting/deselecting all peripherals

Clicking the **Select All** button allows selecting all peripherals at once.

Clicking **Deselect All** removes them as contributors to the step consumption.

Figure 87. **Power Consumption Calculator Step configuration window: ADC enabled using the import pinout function**



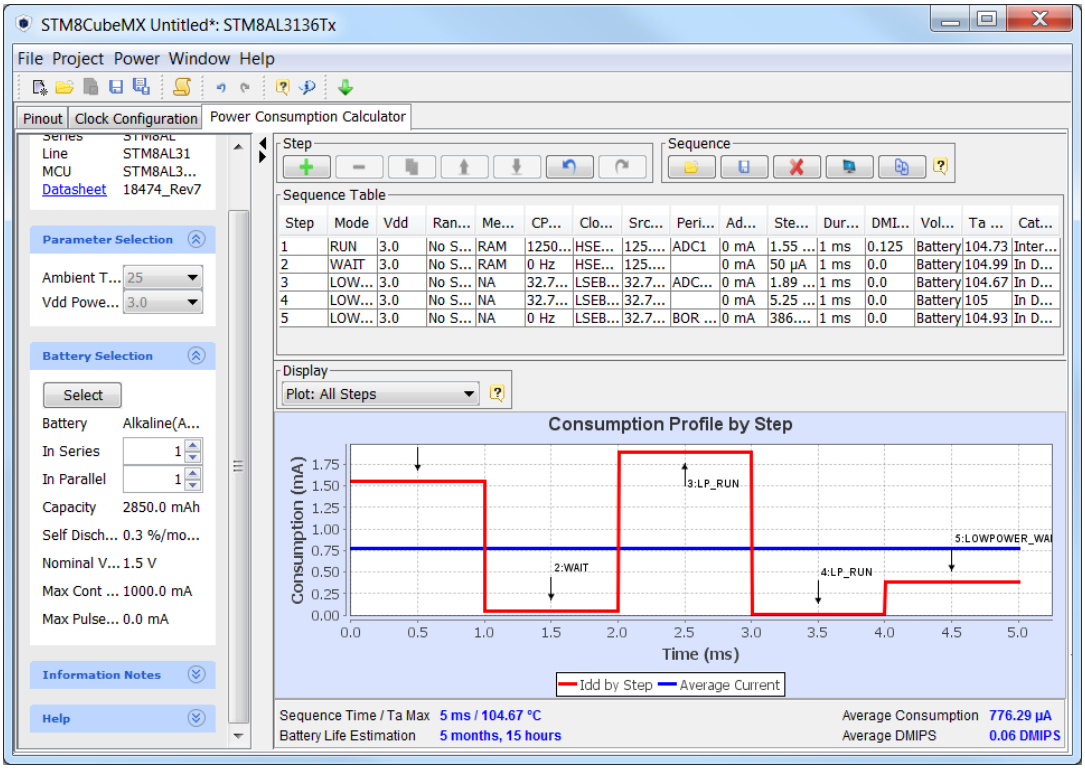
4.12.3 Managing user-defined power sequence and reviewing results

The configuration of a power sequence leads to an update of the Power Consumption Calculator view (see Figure 88. Power Consumption Calculator view after sequence building):

- The sequence table shows all steps and step parameters values. A category column indicates whether the consumption values are taken from the datasheet or are interpolated.
- The sequence chart area shows different views of the power sequence according to a display type (e.g. plot all steps, plot low power versus run modes, ..)

- The results summary provides the total sequence time, the maximum ambient temperature (T_{AMAX}), plus an estimate of the average power consumption, DMIPS, and battery lifetime provided a valid battery configuration has been selected.

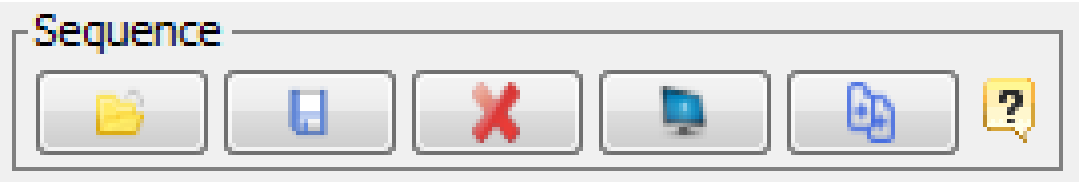
Figure 88. Power Consumption Calculator view after sequence building



Managing the whole sequence (load, save and compare)

The current sequence can be saved or deleted by clicking  and  , respectively.
In addition, a previously saved sequence can be either loaded in the current view or opened for comparison by clicking  (see Figure 89. Sequence table management functions).

Figure 89. Sequence table management functions



To load a previously saved sequence:

1. Click the load button  .
2. Browse to select the sequence to load.

To open a previously saved sequence for comparison:

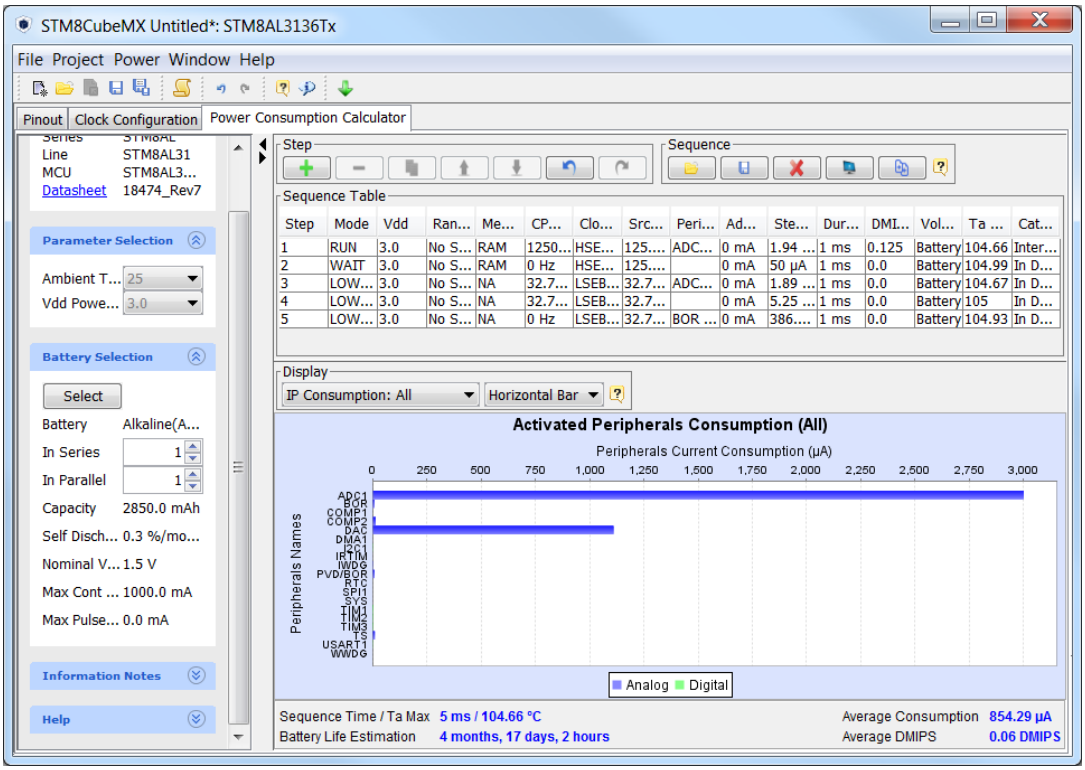
1. Click the **Compare** button  .
2. Browse and select the .pcs sequence file to be compared with the current sequence. A new window opens showing the selected sequence details.

Managing the results charts and display options

In the Display area, select the type of chart to display (sequence steps, pie charts, consumption per peripherals, ...). You can also click **External Display** to open the charts in dedicated windows (see Figure 90. Power Consumption: Peripherals Consumption Chart).

Right-click on the chart to access the contextual menus: **Properties**, **Copy**, **Save** as png picture file, **Print**, **Zoom** menus, and **Auto Range** to reset to the original view before zoom operations. **Zooming** can also be achieved by mouse selecting from left to right a zone in the chart and **Zoom reset** by clicking the chart and dragging the mouse to the left.

Figure 90. Power Consumption: Peripherals Consumption Chart



Overview of the Results summary area

This area provides the following information (see Figure 91. Description of the Results area):

- Total sequence time as the sum of the sequence steps durations.
- Average consumption as the sum of each step consumption weighed by the step duration.
- The average DMIPS (Dhrystone Million Instructions per Second) based on Dhrystone benchmark, highlighting the CPU performance for the defined sequence.
- Battery life estimation for the selected battery model, based on the average power consumption and the battery self-discharge.
- T_{AMAX}: highest maximum ambient temperature value encountered during the sequence.

Figure 91. Description of the Results area

4.12.4 Power sequence step parameters glossary

The parameters that characterize power sequence steps are the following:

- Power modes

  To save energy, it is recommended to switch the microcontroller operating mode from running mode, where a maximum power is required, to a low-power mode requiring limited resources.

- Memory Fetch Type

  This field proposes the possible memory locations for application C code execution. It can be either RAM, FLASH or FLASH with ART ON or OFF (only for families that feature a proprietary Adaptive real-time (ART) memory accelerator which increases the program execution speed when executing from Flash memory).

  The performance achieved thanks to the ART accelerator is equivalent to 0 wait state program execution from Flash memory. In terms of power consumption, it is equivalent to program execution from RAM. In addition, STM8CubeMX uses the same selection choice to cover both settings, RAM and Flash with ART ON.

- Clock Configuration

  This operation sets the AHB bus frequency or the CPU frequency that will be used for computing the microcontroller power consumption. When there is only one possible choice, the frequencies are automatically configured.

  The clock configuration drop-down list allows to configure the application clocks:

  – The internal or external oscillator sources: HSI, LSI, HSE or LSE),
  – The oscillator frequency,
  – Other determining parameters: PLL ON, LSE Bypass, AHB prescaler value, LCD with duty...

- Peripherals

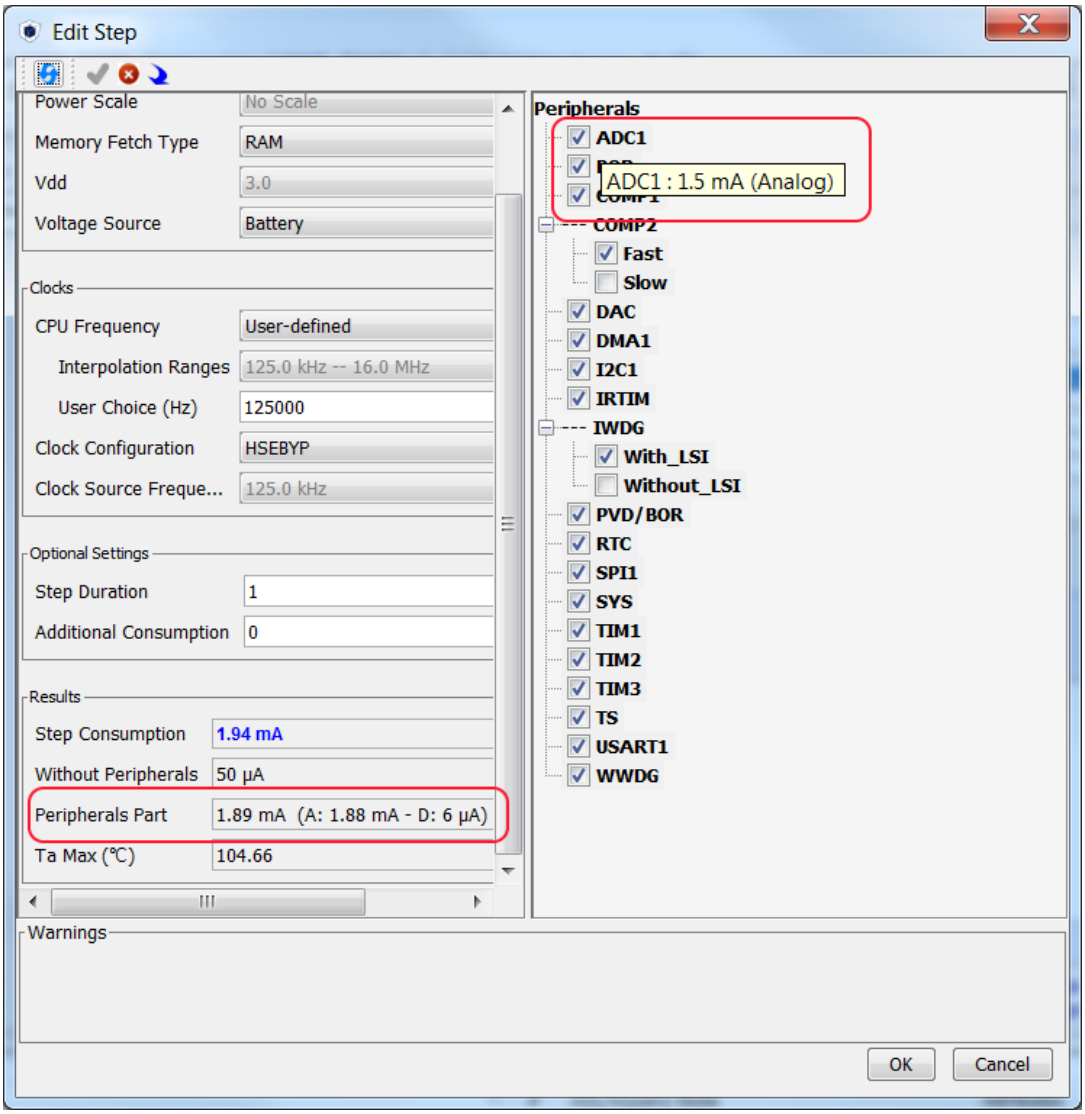
  The peripheral list shows the peripherals available for the selected power mode. The power consumption is given assuming that peripherals are only clocked (e.g. not in use by a running program). Each peripheral can be enabled or disabled. Peripherals individual power consumptions are displayed in a tooltip. An overall consumption due to peripheral analog and digital parts is provided in the step Results area (see Figure 92. Peripheral power consumption tooltip).

  The user can select the peripherals relevant for the application:

  – None (**Disable All)**,
  – Some (using peripheral dedicated checkbox),
  – All (**Activate All**),
  – Or all from the previously defined pinout configuration (**Import Pinout**).

  Only the selected and enabled peripherals are taken into account when computing the power consumption.

Figure 92. **Peripheral power consumption tooltip**



- Step duration

The user can change the default step duration value. When building a sequence, the user can either create steps according to the application actual power sequence or define them as a percentage spent in each mode. For example, if an application spends 30% in Run mode, 20% in Sleep and 50% in Stop, the user must configure a 3-step sequence consisting in 30 ms in Run, 20 ms in Sleep and 50 ms in Stop.

- Additional Consumption

This field allows entering an additional consumption resulting from specific user configuration (e.g. MCU providing power supply to other connected devices).

4.12.5 Battery glossary

- Capacity (mAh)

  The capacity is the amount of energy that can be delivered in a single battery discharge.

- Self-discharge (%/month)

  This percentage, over a specified period, represents the loss of battery capacity when the battery is not used (open-circuit conditions), as a result of internal leakage.

- Nominal voltage (V)

  Voltage supplied by a fully charged battery.

- Max. Continuous Current (mA)

  This current corresponds to the maximum current that can be delivered during the battery lifetime period without damaging the battery.
- Max. Pulse Current (mA)

  This is the maximum pulse current that can be delivered exceptionally, for instance when the application is switched on during the starting phase.

5 Tutorial 1: STM8CubeMX project creation and miscellaneous features

This section shows how to configure the pins and clock tree of an STM8 microcontroller through STM8CubeMX user interface, how to generate the configuration report and how to export the current configuration to another STM8 microcontroller part number.

5.1 Creating a new STM8CubeMX project

1. Select **File > New project** from the main menu bar or **New project** from the Welcome page. This opens the MCU selector window.
2. From the MCU Selector window, use the MCU filters to filter down the STM8 portfolio (see Figure 93. MCU selection):
   a. Select **CAN** from Peripheral Selection.
   b. Select **STM8AF** as Series.
   c. Select **LQFP64** as Package.
3. Finally, select the **STM8AF5289Tx** from the MCUs list and click **OK**. The STM8CubeMX views are then populated with the selected MCU information (see Figure 94. Pinout view with MCU selection).

Optionally, remove the MCUs Selection bottom window by unselecting **Window> Outputs** sub-menu (see Figure 95. Pinout view without MCUs selection).

Figure 93. MCU selection

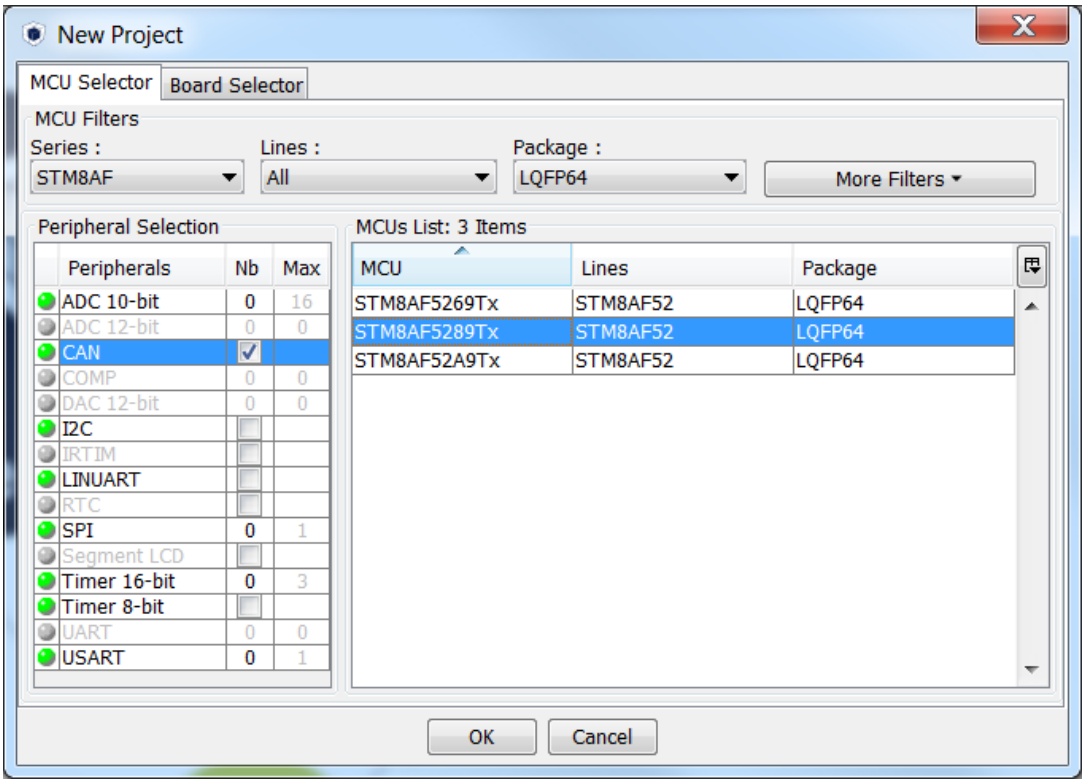


Figure 94. **Pinout view with MCU selection**

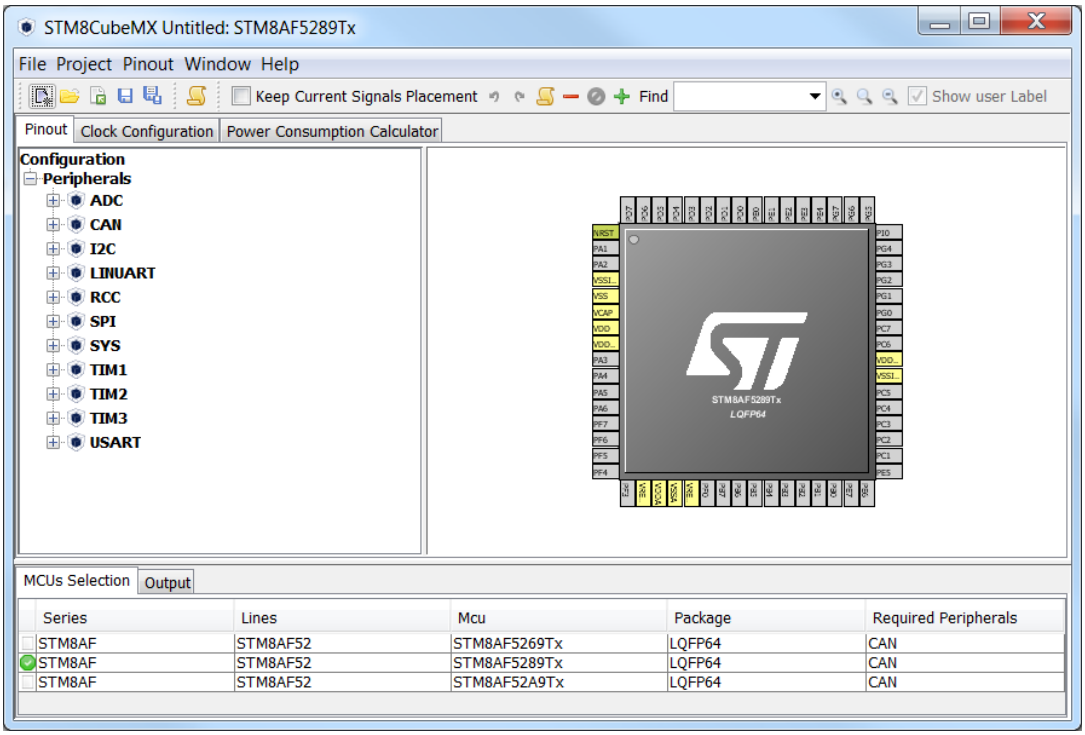
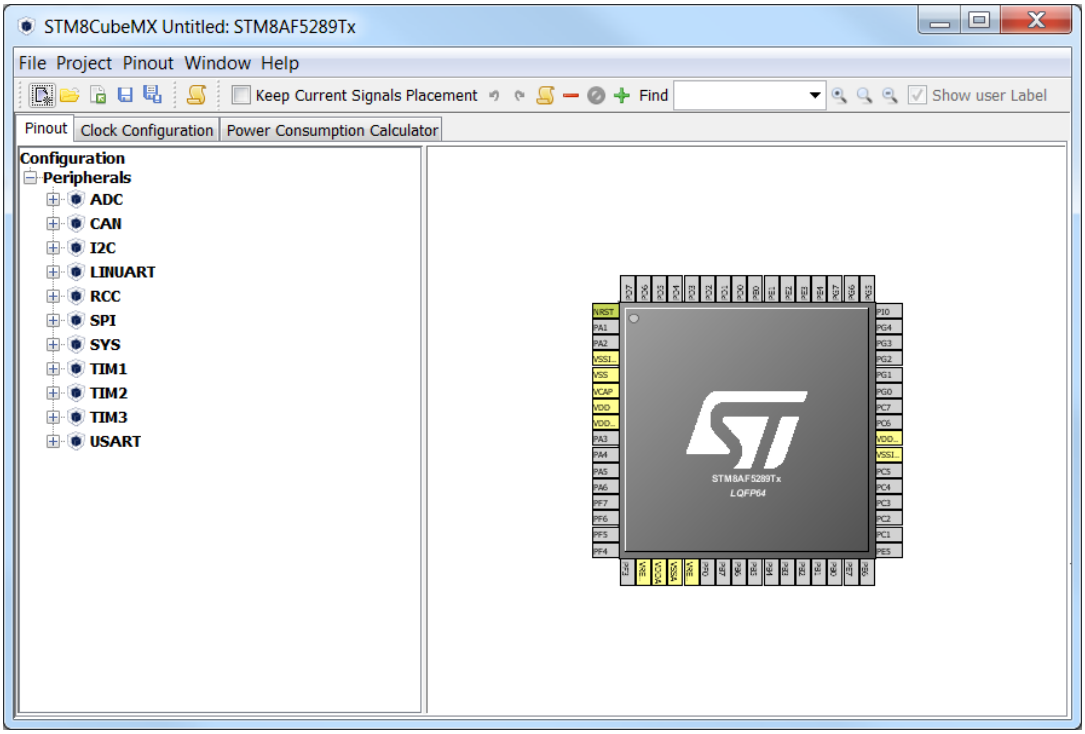
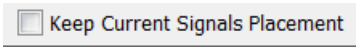


Figure 95. **Pinout view without MCUs selection**



5.2 Configuring the MCU pinout

For a detailed description of menus, advanced actions and conflict resolution, refer to Section 4 STM8CubeMX user Interface and Section 7 STM8CubeMX pin assignment rules.

1. By default, STM8CubeMX shows the **Pinout** view.
2. Leaving 

unchecked allows STM8CubeMX to move the peripheral functions around and to find the optimal pin allocation, that is the one that accommodates the maximum number of peripheral modes
3. Select the following peripherals and peripheral modes from the **Peripheral tree** view (see Figure 96. Initial pin configuration):
   a. CAN in Master mode.
   b. I2C.
   c. USART in Synchronous mode.
   d. RCC with HSE clock as crystal/ceramic resonator.
   e. TIM1 with clock/trigger ETR, one input channel and one output channel.
4. If the default pin allocation for I2C is not adequate, control-click on the I2C pin to move the I2C signal to the pin highlighted in blue (see Figure 97. Pinout changes).
5. The part of the pinout configuration that is considered final can be “pinned” to prevent STM8CubeMX from doing automatic pin re-allocation: select **Pinout > Pins/Signals options** from the main menu bar to open the pin/signals options window. Decide which signals to pin and enter a user-defined label if need be (see Figure 98. Pins/signals option window settings for pinning and labelling).
6. Click **OK** to update the pinout view accordingly (see Figure 99. Final pinout configuration view).

Figure 96. Initial pin configuration

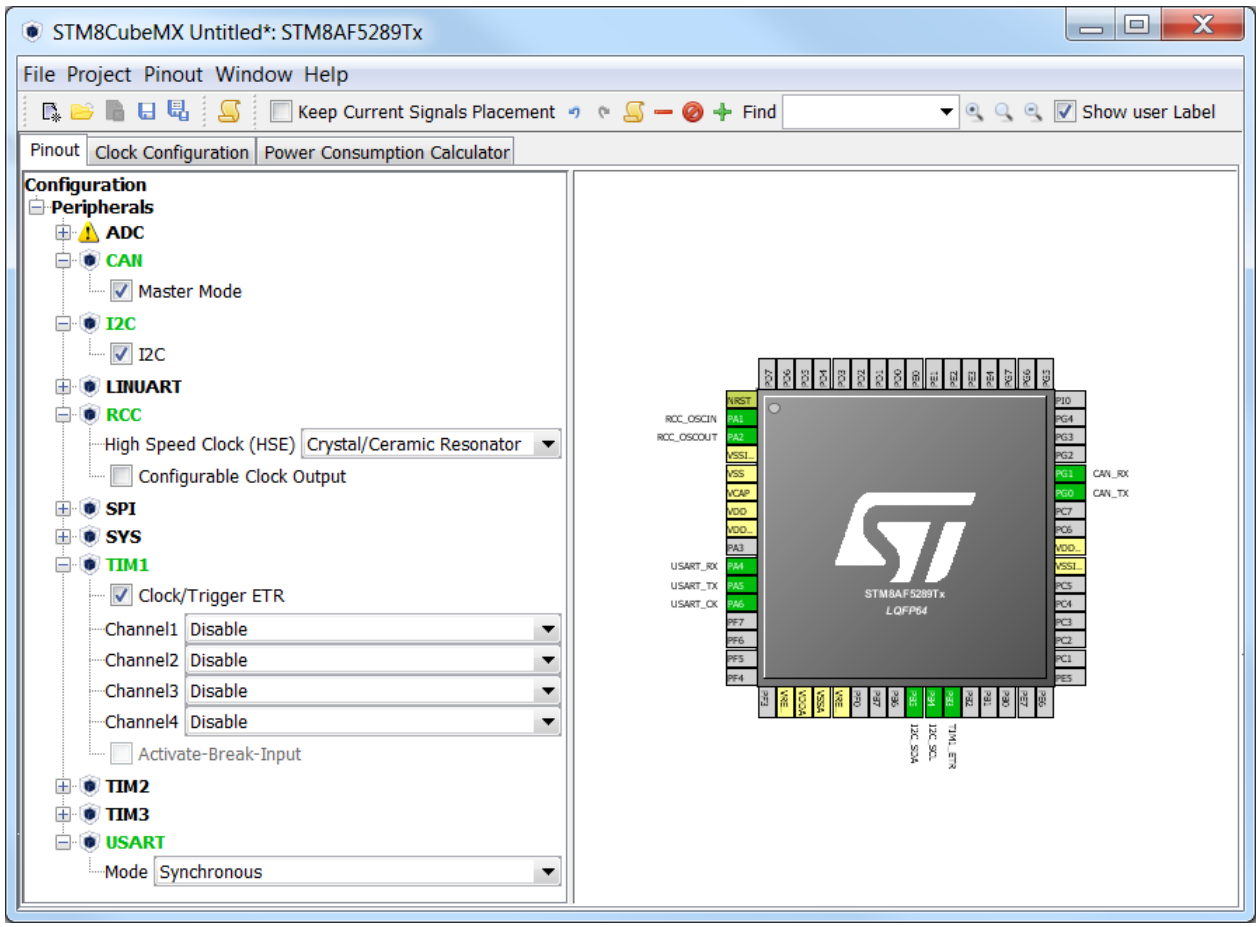


Figure 97. **Pinout changes**

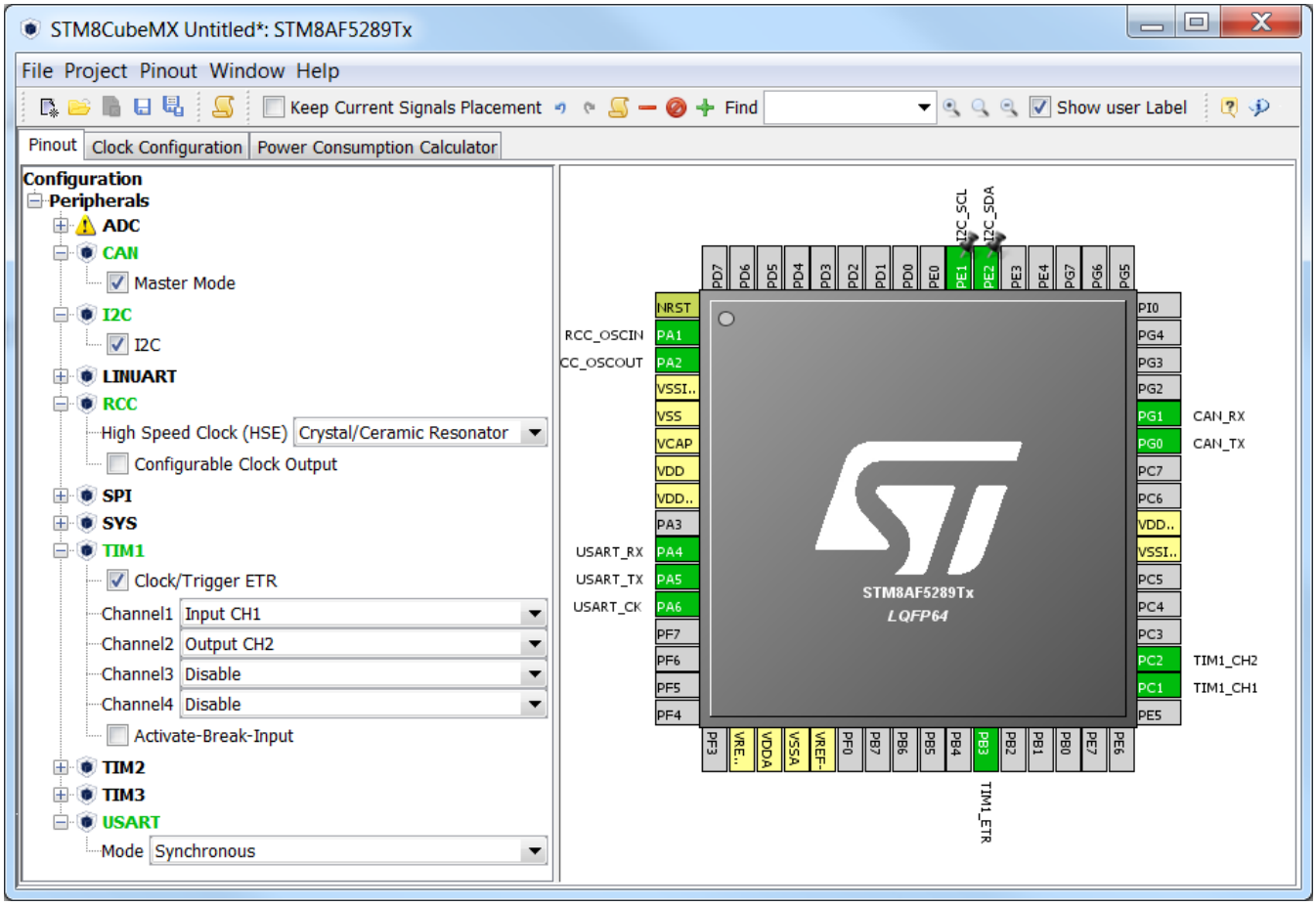


Figure 98. **Pins/signals option window settings for pinning and labelling**

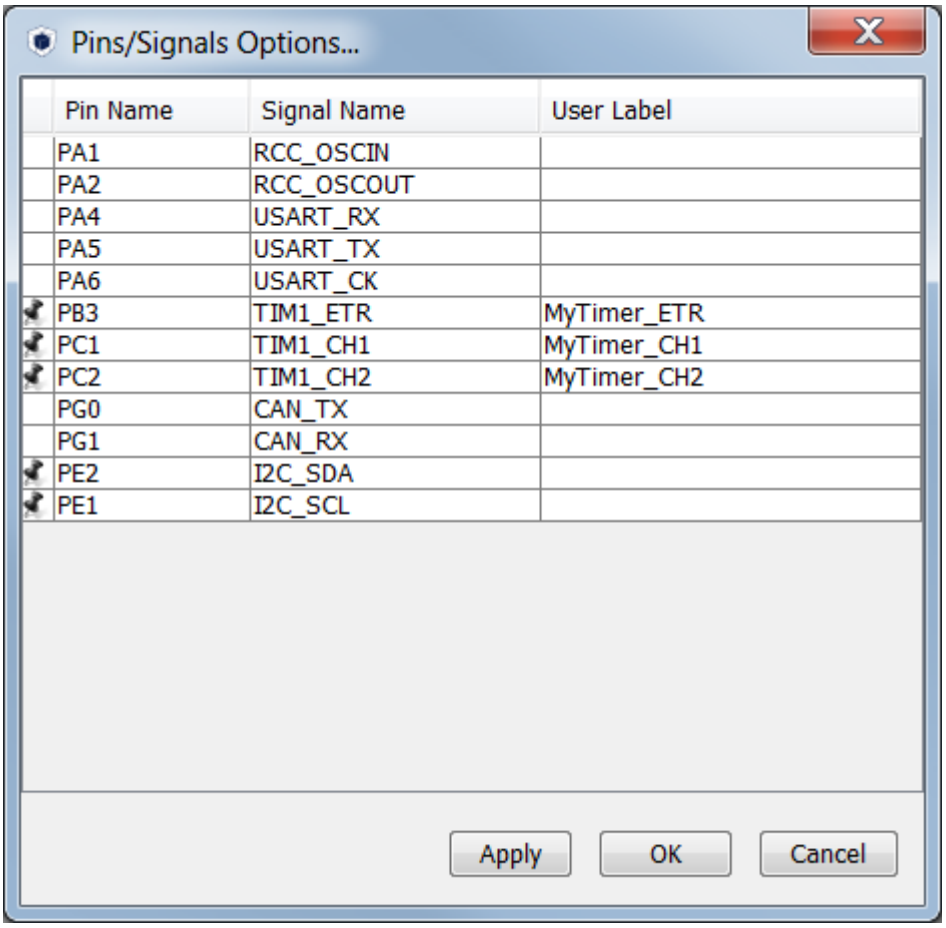
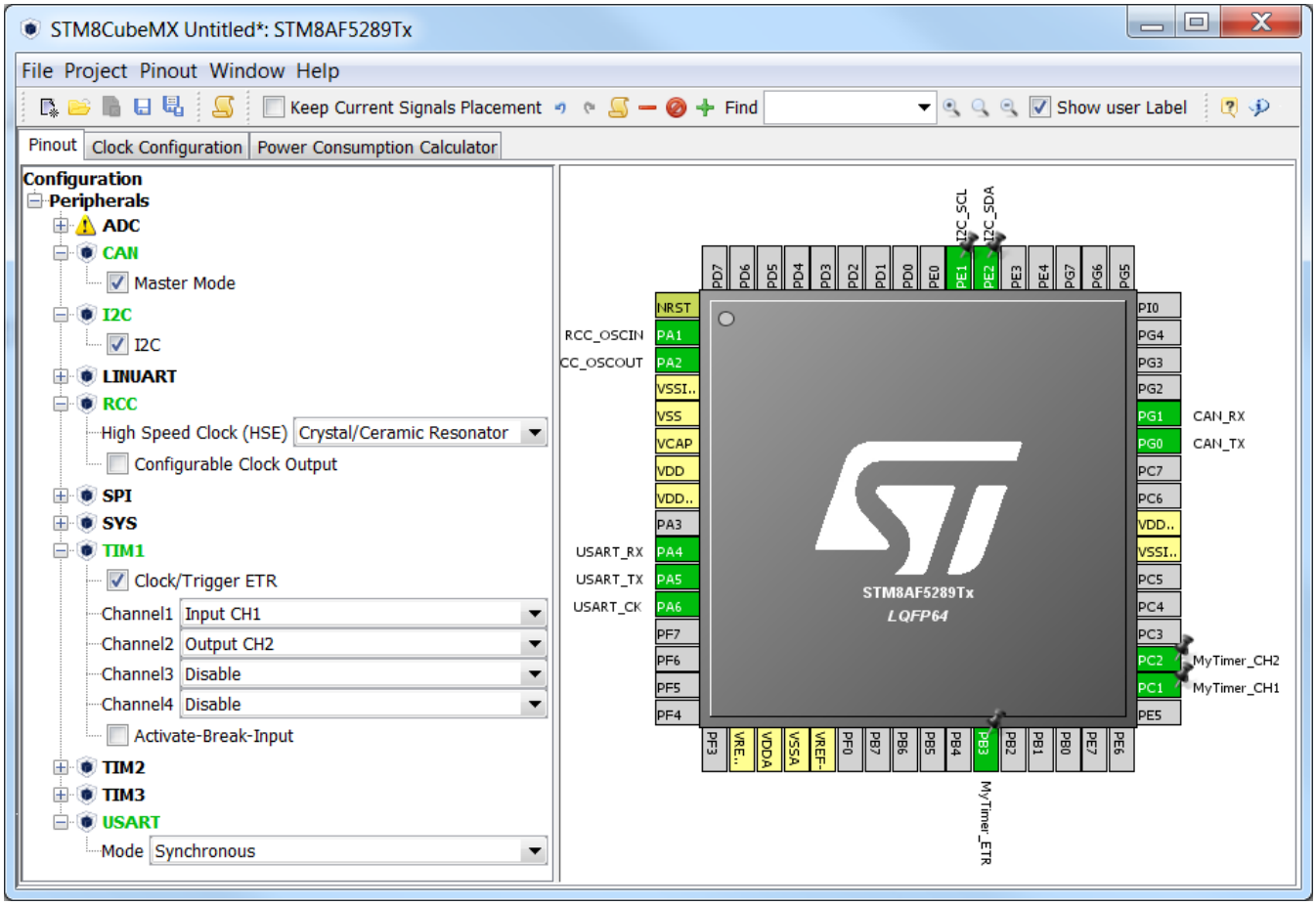


Figure 99. **Final pinout configuration view**

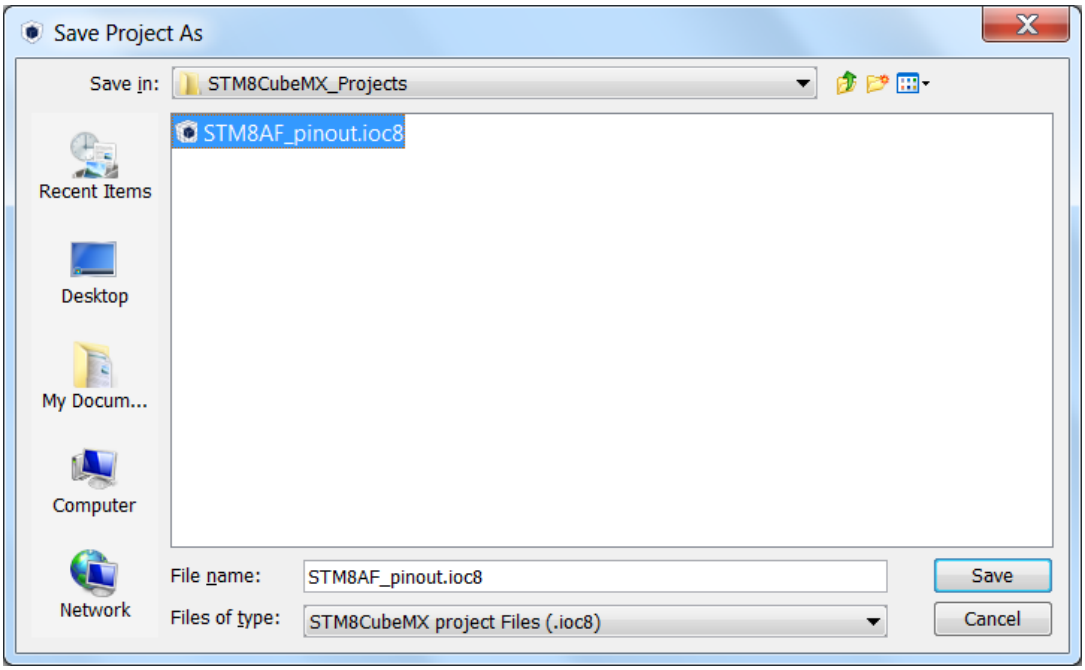


5.3 Saving the project

1. Click  to save the project.

   When saving for the first time, select a destination folder and filename for the project. The .ioc8 extension is added automatically to indicate this is an STM8CubeMX configuration file (see Figure 100. Saving your project).

Figure 100. **Saving your project**



5.4 Configuring the MCU Clock tree

Follow the sequence below to configure the MCU clock tree:

1. Click the **Clock Configuration** tab to display the clock tree (see Figure 101. Clock tree view).
2. Select HSE as clock source for the master clock multiplexer, and change the input frequency to 12 MHz (see Figure 102. Configuring the clock tree).

Figure 101. **Clock tree view**

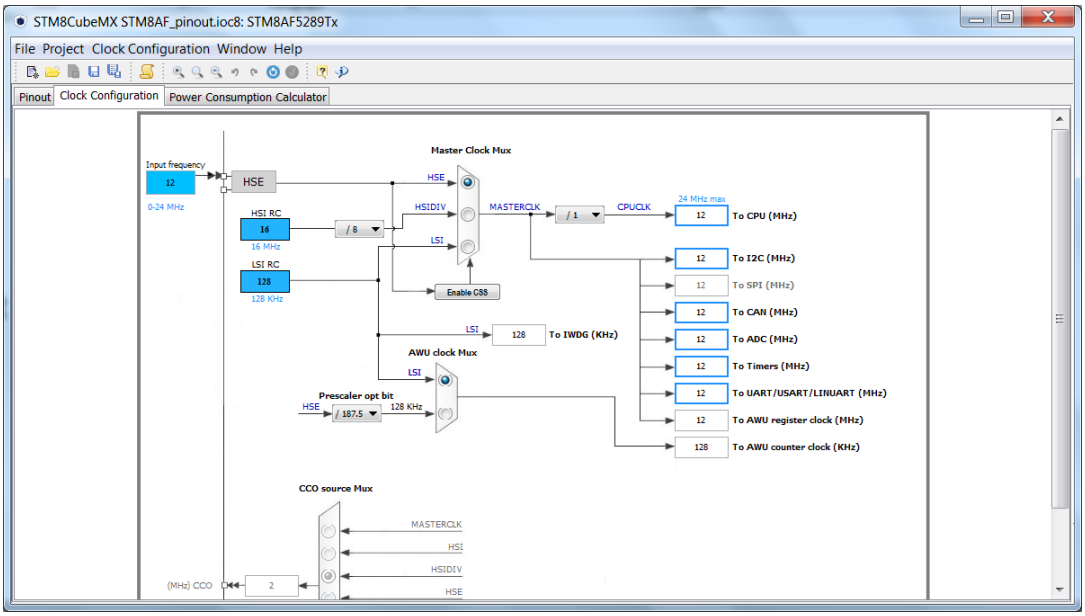
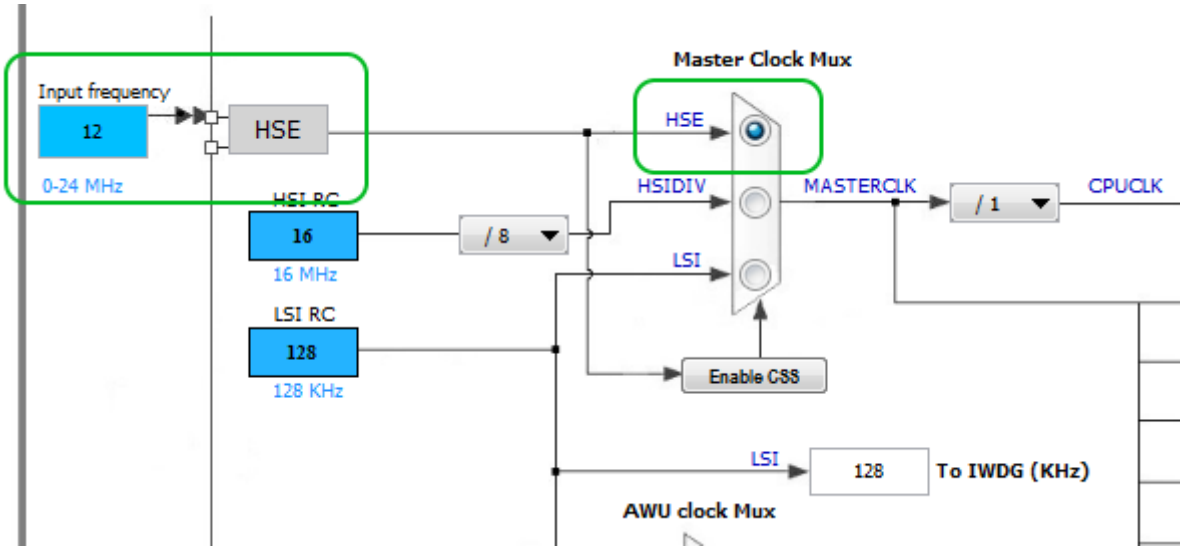


Figure 102. **Configuring the clock tree**



5.5 Generating the report

1. Click to generate .pdf and .txt reports. A dialog window pops up to prompt the user that the generation has been successful (see Figure 103. Generate project reports). When some project modifications have not been saved, a dialog window opens to ask the user whether to save the project before report generation or not.
2. Click **Open folder** to display the project configuration file and reports (see Figure 104. Resulting STM8CubeMX project file and reports).

Figure 103. **Generate project reports**

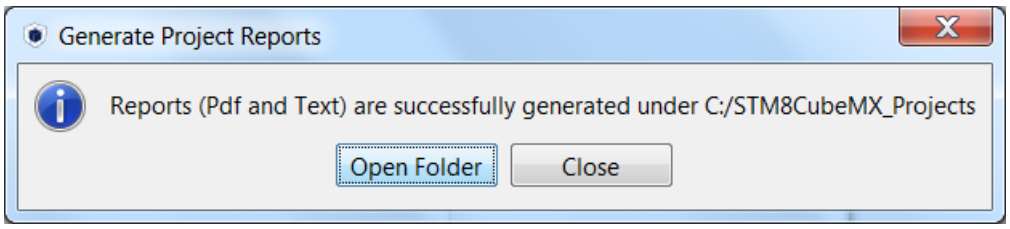
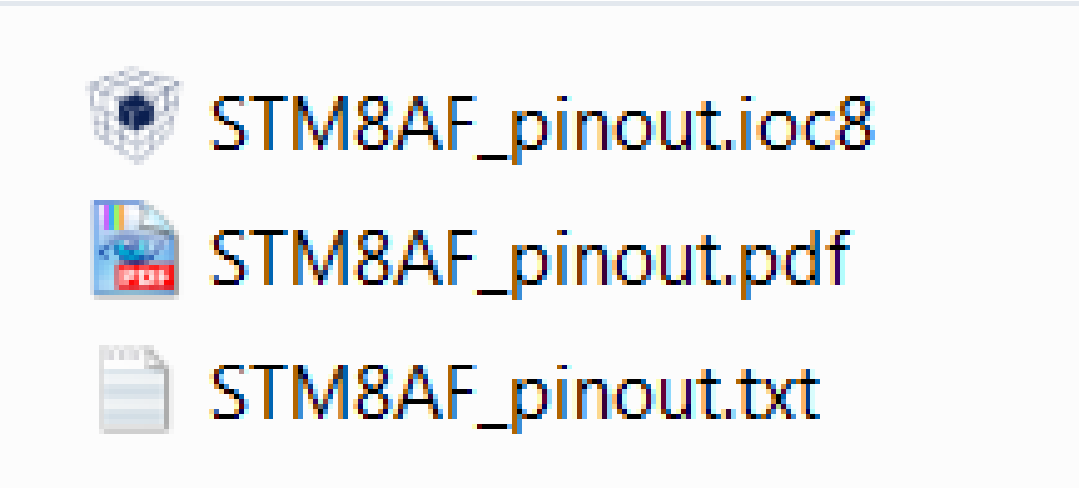


Figure 104. **Resulting STM8CubeMX project file and reports**



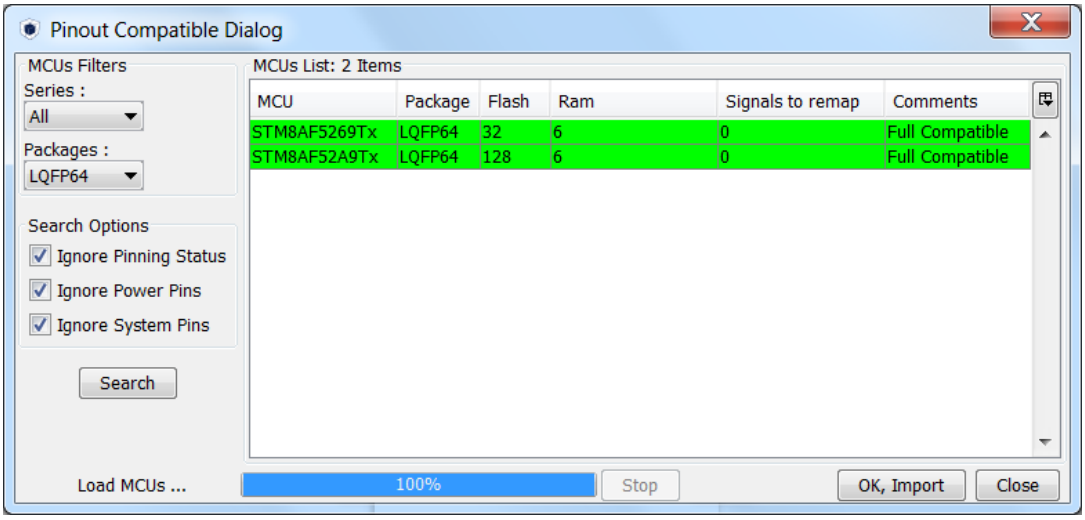
Both pdf and txt reports provide the details of the user-selected MCU, peripherals, peripherals modes and peripherals signals allocation on pins. The pdf version also includes user interface snapshots of the STM8CubeMX project.

5.6 Switching to another MCU

Listing compatible MCUs

1. Select the **Pinout** tab to enable the pinout related menus.
2. Select 'List compatible MCUs' from the **Pinout** menu to open the pinout compatible Dialog (see Figure 105. Showing the list of compatible MCUs).

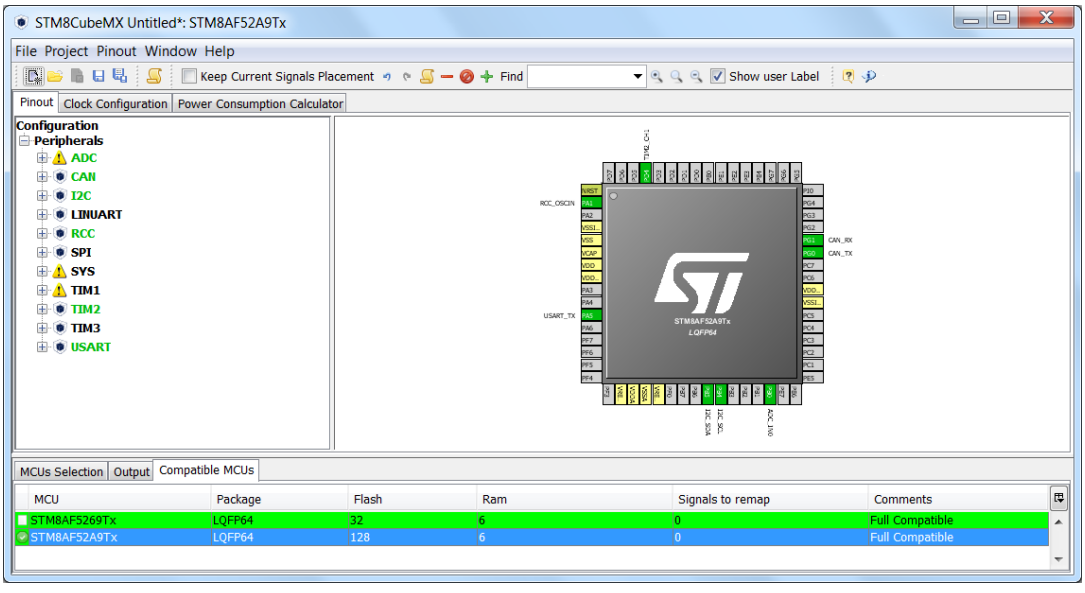
Figure 105. Showing the list of compatible MCUs



Importing current configuration on another MCU

1. Select the STM8AF52A9Tx MCU and click **Ok, Import** (see Figure 106. Importing current configuration to a compatible MCU). This creates a new project using the selected compatible MCU configured as previously saved project.

Figure 106. Importing current configuration to a compatible MCU



6 FAQ

6.1 On the Pinout configuration pane, why does STM8CubeMX move some functions when I add a new peripheral mode?

You may have unselected the **Keep Current signals Placement** box (

). In this case, the tool performs an automatic remapping to optimize your placement.

6.2 How can I manually force a function remapping?

You should use the **Manual Remapping** feature.

6.3 Why are some pins highlighted in yellow or in light green in the Chip view? Why cannot I change the function of some pins (when I click some pins, nothing happens)?

These pins are specific pins (such as power supply or BOOT) which are not available as peripheral signals.

6.4 Why do I get the error “Java 7 update 45’ when installing ‘Java 7 update 45’ or a more recent version of the JRE?

The problem generally occurs on 64-bit Windows operating system, when several versions of Java are installed on your computer and the 64-bit Java installation is too old.

During STM8CubeMX installation, the computer searches for a 64-bit installation of Java.

- If one is found, the ‘Java 7 update 45’ minimum version prerequisite is checked. If the installed version is older, an error is displayed to request the upgrade.
- If no 64-bit installation is found, STM8CubeMX searches for a 32-bit installation. If one is found and the version is too old, the ‘Java 7 update 45’ error is displayed. The user must update the installation to solve the issue.

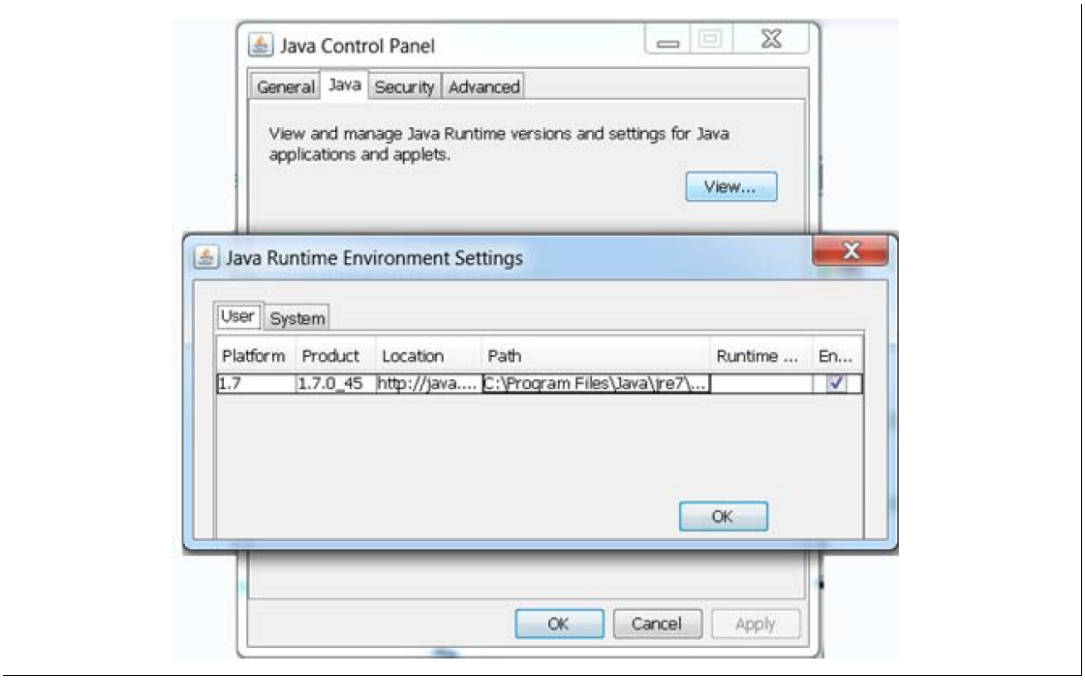
To avoid this issue from occurring, it is recommended to perform one of the following actions:

1. Remove all Java installations and reinstall only one version (32 or 64 bits) (Java 7 update 45 or more recent).
2. Keep 32-bit and 64-bit installations but make sure that the 64-bit version is at least Java 7 update 45.

Note: Some users (such as Java developers) may need to check the PC environment variables defining hard-coded Java paths (e.g. JAVA_HOME or PATH) and update them so that they point to the latest Java installation.

On Windows 7 you can check your Java installation using the Control Panel. To do this, double-click icon  from Control Panel\\All Control Panel to open the Java settings window (see Figure 107. Java Control Panel):

Figure 107. **Java Control Panel**



You can also enter 'java –version' as an MS-DOS command to check the version of your latest Java installation (the Java program called here is a copy of the program installed under C:\\Windows\\System32):

```
java version "1.7.0_45"
```

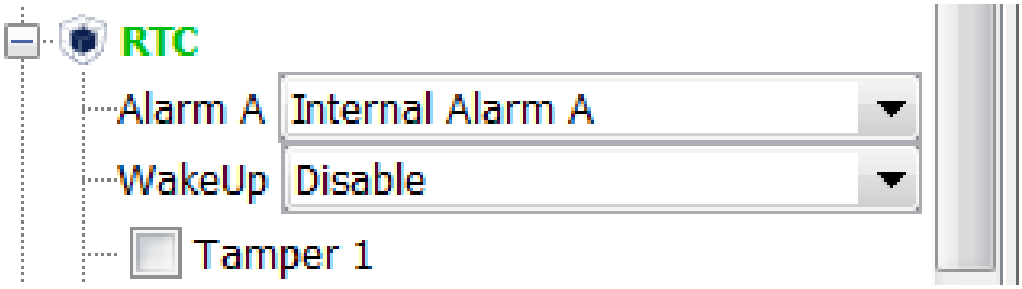
```
Java (TM) SE Runtime Environment (build 1.7.0_45-b18)
```

```
Java HotSpot (TM) 64-Bit Server VM (build 24.45-b08, mixed mode)
```

6.5 Why does the RTC multiplexer remain inactive on the Clock tree view?

To enable the RTC multiplexer, the user shall enable the RTC peripheral in the **Pinout** view as indicated in below:

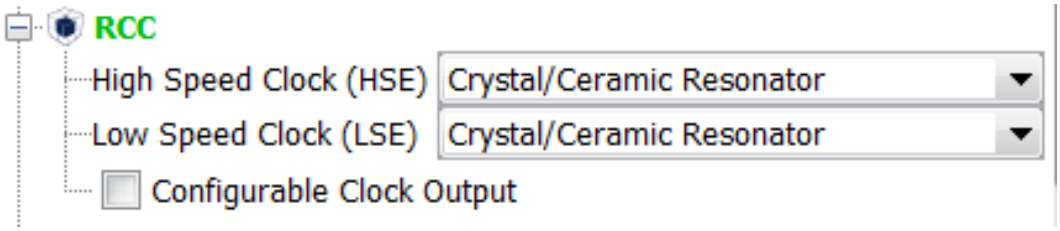
Figure 108. **Pinout view - Enabling the RTC**



6.6 How can I select LSE and HSE as clock source and change the frequency?

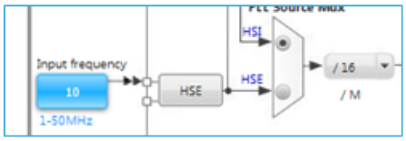
The LSE and HSE clocks become active once the RCC is configured as such in the **Pinout** view. See Figure 109. Pinout view - Enabling LSE and HSE clocks for an example.

Figure 109. **Pinout view - Enabling LSE and HSE clocks**



The clock source frequency can then be edited and the external source selected:

Figure 110. **Pinout view - Setting LSE/HSE clock frequency**



7 STM8CubeMX pin assignment rules

The following pin assignment rules are implemented in STM8CubeMX:

- Block consistency
- Block remapping (only for STM8AL/Lxx)
- Function remapping
- Setting or clearing a peripheral mode
- Mapping a function individually (if Keep Current Placement is unchecked)
- GPIO signals mapping

7.1 Block consistency

When setting a pin signal (provided there is no ambiguity about the corresponding peripheral mode), all the pins/signals required for this mode are mapped and pins are shown in green (otherwise the configured pin is shown in orange).

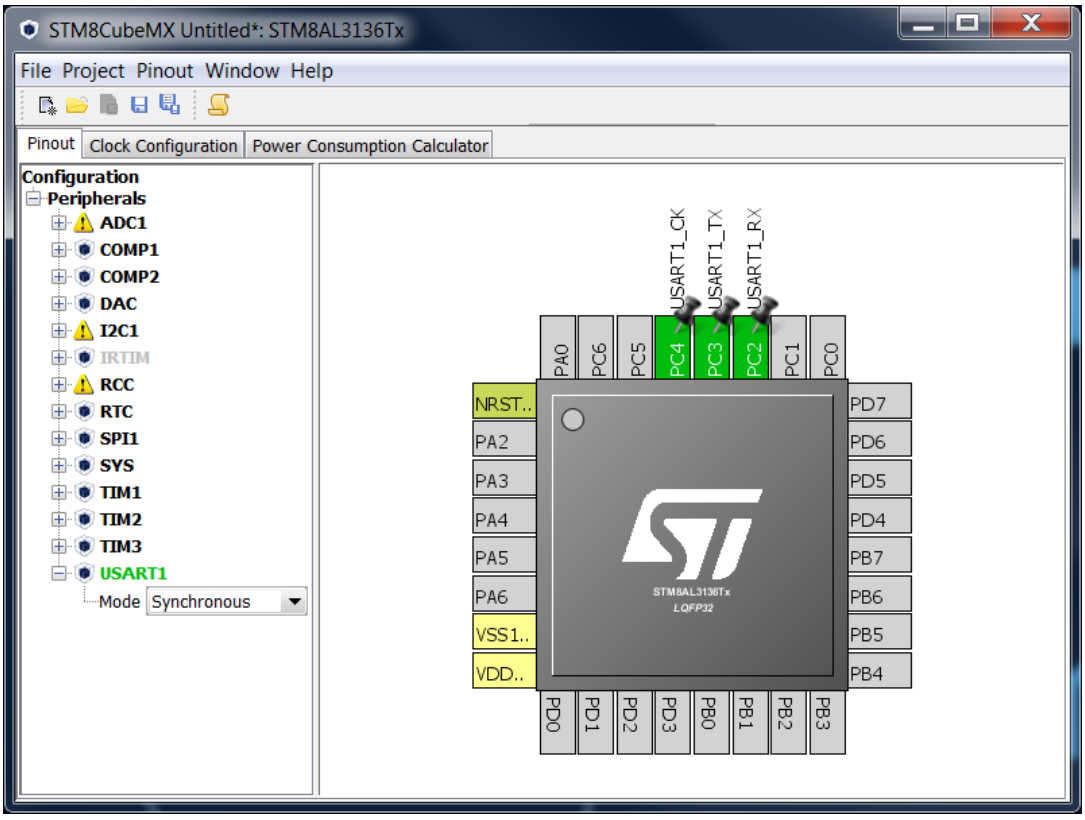
When clearing a pin signal, all the pins/signals required for this mode are unmapped simultaneously and the pins turn back to gray.

Example of block mapping with a STM8AL3136Tx MCU

If the user assigns USART1_CK to PC4 pin, then STM8CubeMX automatically configures pins and modes as follows:

- USART1_TX and USART_RX signals are mapped to the PC3 and PC2 pins, respectively (see Figure 111. Block mapping).
- USART1 peripheral mode is set to Synchronous mode.

Figure 111. Block mapping

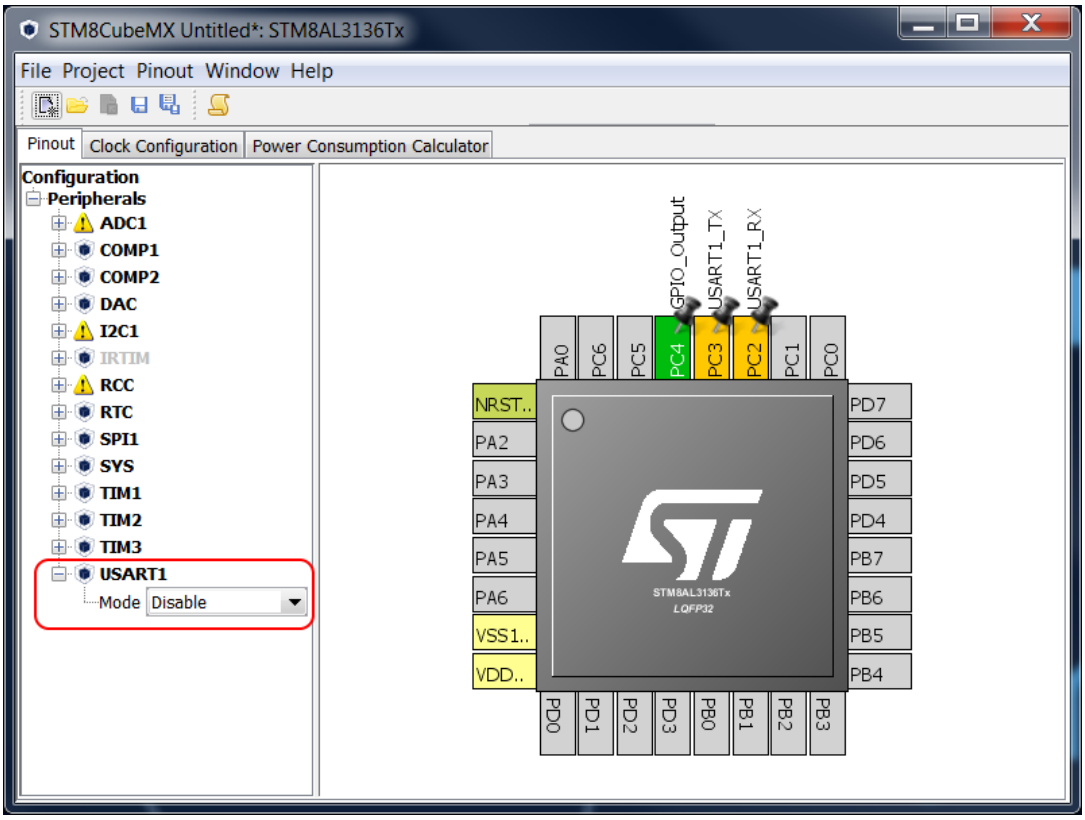


Example of block remapping with a STM8AL3136Tx MCU

If the user assigns GPIO_Output to PC4, STM8CubeMX automatically disables USART1 synchronous mode from the peripheral tree view and updates the other USART1 pins (PC2 and PC3) as follows:

- If they are unpinned, the pin configuration is reset (pin grayed out).
- If they are pinned, the peripheral signal assigned to the pins is kept and the pins are highlighted in orange since they no longer match a peripheral mode (see Figure 112. Block remapping).

Figure 112. Block remapping



For STM8CubeMX to find an alternative solution for the USART1 peripheral mode, the user will need to unpin USART1 pins and select the USART1 mode from the peripheral tree view (see Figure 113. Block remapping - example 1and Figure 114. Block remapping - example 2).

Figure 113. Block remapping - example 1

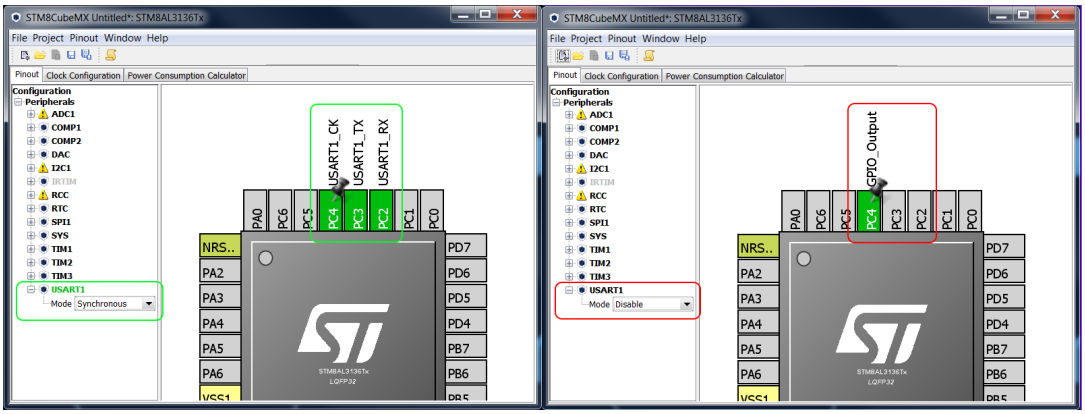
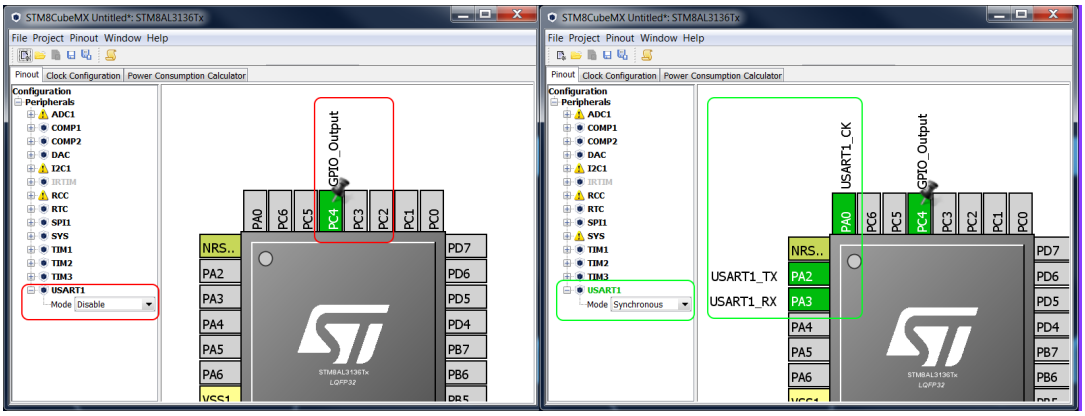


Figure 114. **Block remapping - example 2**



7.2 Block remapping (STM8AL/Lxx)

To configure a peripheral mode, STM8CubeMX selects a block of pins and assigns each mode signal to a pin in this block. In doing so, it looks for the first free block to which the mode can be mapped.

When setting a peripheral mode, if at least one pin in the default block is already used, STM8CubeMX tries to find an alternate block. If none can be found, the user shall either select the functions in a different sequence, or uncheck the Keep current signal placement 

for STM8CubeMX to remap all the blocks to find a solution.

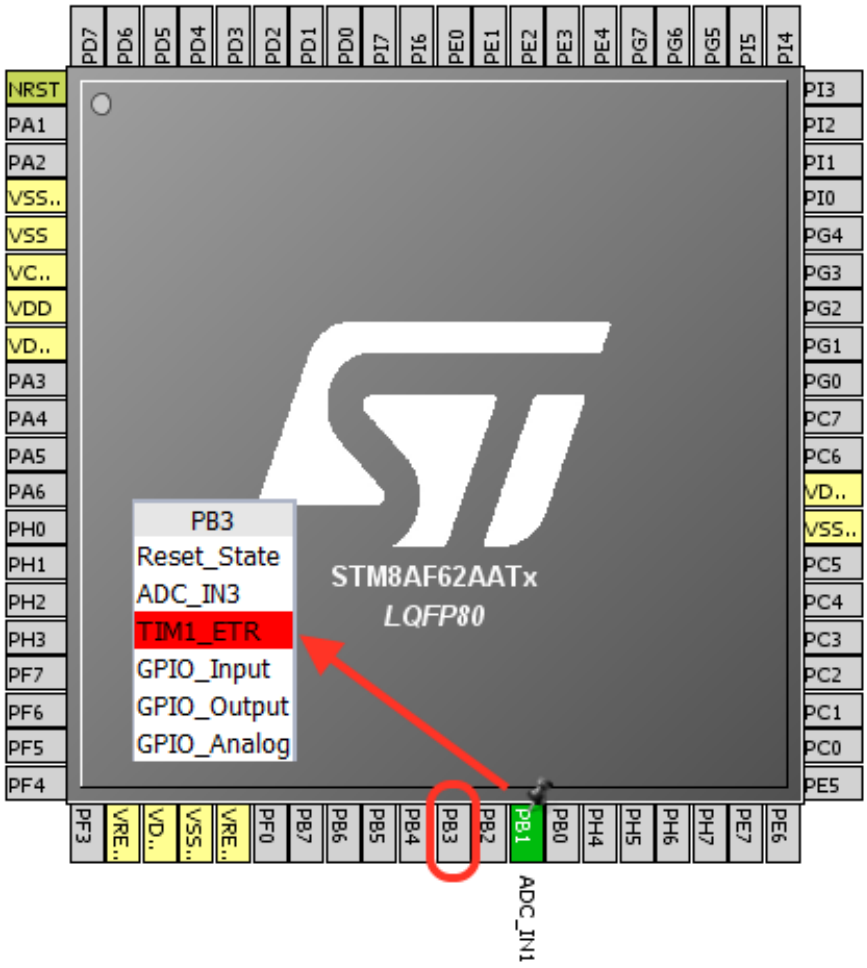
7.3 Alternate Function remapping (STM8AF/Sxx MCUs)

As stated in Datasheet dedicated “alternate function remapping bits” chapter and as illustrated in the following example, some functions may become unavailable on a pin by setting a function on another pin.

Example

Once an ADC channel is configured on PB1, the Tim1 ETR channel becomes unavailable on PB3 pin(see Figure below).

Figure 115. **Alternate function remapping**



7.4 Setting and clearing a peripheral mode

The Peripherals panel and the **Chip**view are linked: when a peripheral mode is set or cleared, the corresponding pin functions are set or cleared.

7.5 Mapping a function individually

When STM8CubeMX needs a pin that has already been assigned manually to a function (no peripheral mode set), it can move this function to another pin, only if 

is unchecked and the function is not pinned (no pin icon).

7.6 GPIO signals mapping

I/O signals (GPIO_Input, GPIO_Output, GPIO_Analog) can be assigned to pins either manually through the **Chip** view or automatically through the Pinout menu. Such pins can no longer be assigned automatically to another signal: STM8CubeMX signal automatic placement does not take into account this pin anymore since it does not shift I/O signals to other pins.

The pin can still be manually assigned to another signal or to a reset state.

Revision history

Table 11. **Document revision history**

Date	Revision	STM8CubeMX release number	Changes
10-Feb-2017	1	1.0.0	Initial release.
18-Sep-2017	2	1.2.0	Changed STM8CubeMx logo together with all figures where it is embedded.
02-Jul-2018	3	1.3.0	Replaced STMCube ™ by STM8Cube ™ in the whole document. Updated Section • Introduction and changed STM8CubeMX logo on cover page.

Contents

List of figures

List of tables

IMPORTANT NOTICE – PLEASE READ CAREFULLY

STMicroelectronics NV and its subsidiaries (“ST”) reserve the right to make changes, corrections, enhancements, modifications, and improvements to ST products and/or to this document at any time without notice. Purchasers should obtain the latest relevant information on ST products before placing orders. ST products are sold pursuant to ST’s terms and conditions of sale in place at the time of order acknowledgement.

Purchasers are solely responsible for the choice, selection, and use of ST products and ST assumes no liability for application assistance or the design of Purchasers’ products.

No license, express or implied, to any intellectual property right is granted by ST herein.

Resale of ST products with provisions different from the information set forth herein shall void any warranty granted by ST for such product.

ST and the ST logo are trademarks of ST. All other product or service names are the property of their respective owners.

Information in this document supersedes and replaces information previously supplied in any prior versions of this document.