UM3101

User manual

Getting started with D-Power Vienna expansion package in eDesignSuite and STM32Cube software tools

Introduction

The D-Power Vienna expansion package is an STM32Cube expansion pack, which lets you speed up the configuration and first operation of the STDES-VRECTFD, digital power converter platform based on a unidirectional three-level, three-phase Vienna rectifier PFC application.

STM32Cube expansion packages are embedded software packages that complement the STM32Cube MCU packages with additional software bricks, including specific drivers for external companion chips or application-specific middleware. This STM32CubeMX tool enables the user to define the power converter configuration generated by eDesignSuite. Application algorithms and parameters are managed by STM32CubeMX thanks to the STM32Cube expansion pack.

The D-Power Vienna expansion package also provides a fully customizable firmware to control the power platform. The main features, such as the output voltage regulation and the current control for the PFC and protection, can be set directly through the STM32CubeMX environment. The software runs on the STM32 microcontroller and includes all the necessary drivers to operate with an STM32G4 MCU.

1 System requirements

- STM32CubeMX v.6.6 or above
- STM32CubeIDE v.1.8 or above
- IAR v.9.20.2 or above
- Arm Keil v5.36

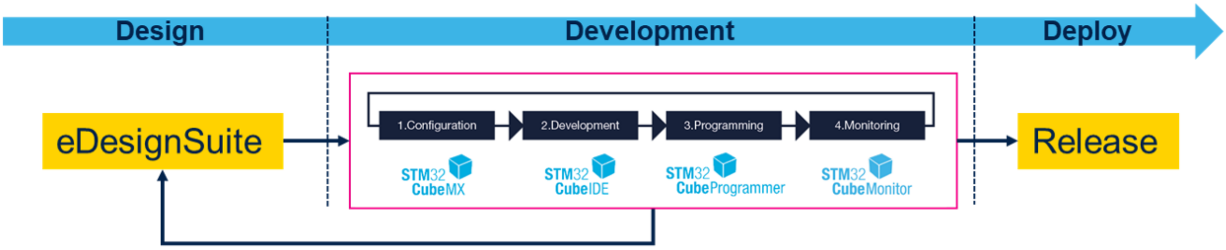
Note: This documentation refers to the above configuration. In case of tools updating, perform a preliminary system check to verify compatibility requirements.

2 eDesignSuite for D-Power converter development

Thanks to the D-Power expansion package, eDesignSuite system level configuration is integrated with STM32Cube environment.

The firmware deployment of eDesignSuite allows generating the STM32CubeMX peripheral configuration file. The multi-toolchain source code generation enabled by STM32CubeMX allows obtaining a fully configured IDE project also for STM32CubeIDE. Additional tools, such as STM32CubeProg and STM32CubeMonitor, complete the user development interface for programming and monitoring.

Figure 1. DPC Vienna rectifier design flow

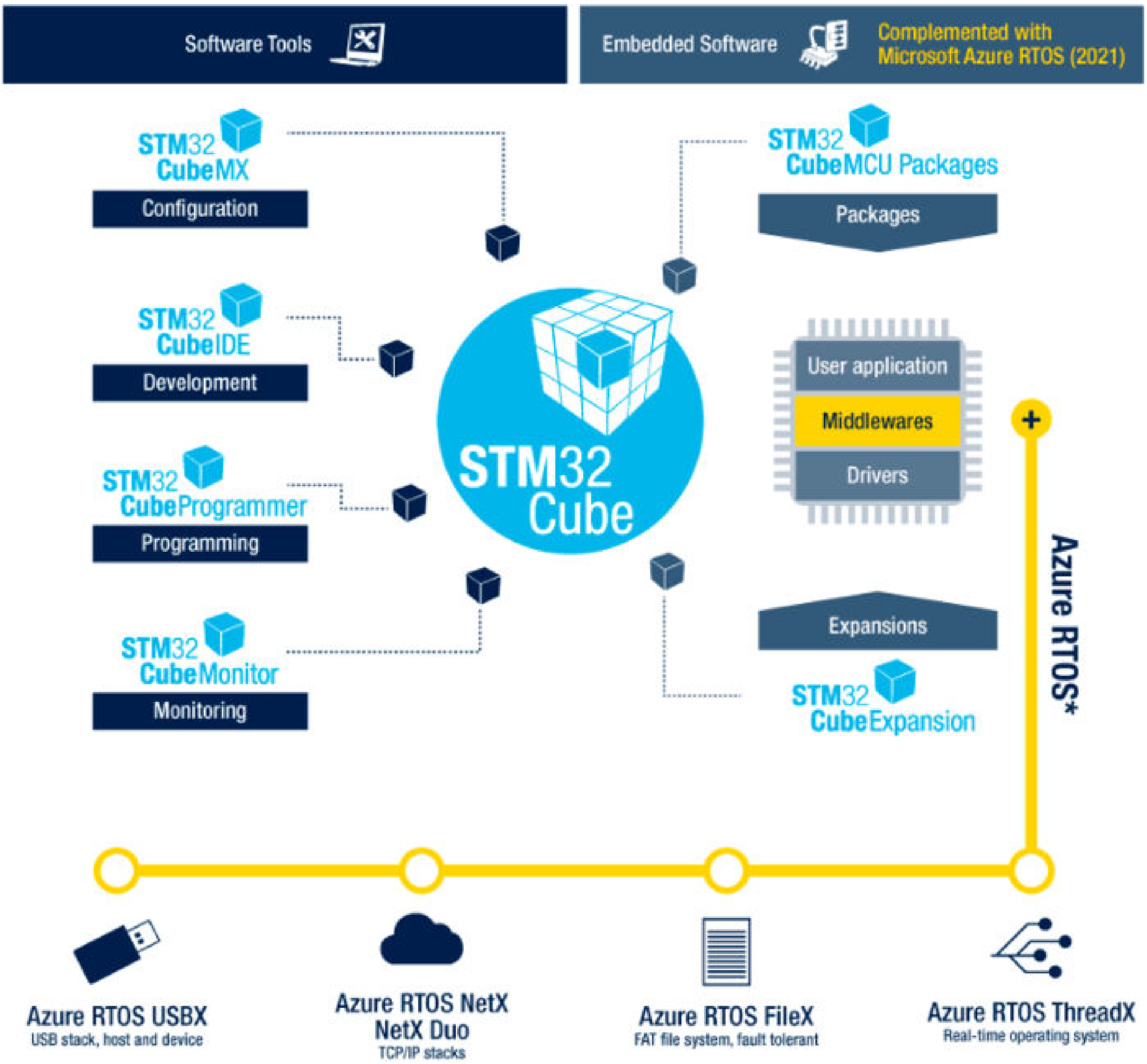


3 STM32Cube development environment overview

STM32Cube is a combination of a full set of PC software tools and embedded software blocks running on STM32 microcontrollers and microprocessors:

- STM32CubeMX configuration tool for any STM32 device; it generates initialization C code for Cortex-M cores and the Linux device tree source for Cortex-A cores
- STM32CubeIDE integrated development environment based on open-source solutions like Eclipse or the GNU C/C++ toolchain, including compilation reporting features and advanced debug features
- STM32CubeProg programming tool that provides an easy-to-use and efficient environment for reading, writing and verifying devices and external memories via a wide variety of available communication media (JTAG, SWD, UART, USB DFU, I2C, SPI, CAN, etc.)
- STM32CubeMonitor family of tools (STM32CubeMonRF, STM32CubeMonUCPD, STM32CubeMonPwr) to help developers customize their applications in real-time
- STM32Cube MCU and MPU packages specific to each STM32 series with drivers (HAL, low-layer, etc.), middleware, and lots of example code used in a wide variety of real-world use cases
- STM32Cube expansion packages for application-oriented solutions

Figure 2. STM32Cube overview



The default software and middleware stacks can be extended thanks to STM32Cube expansion packages. STMicrolectronics or its partner packages can be downloaded directly from a dedicated package manager available within STM32CubeMX. The other packages can be installed from a local drive.

Moreover, a unique utility in STM32CubeMX delivery, STM32PackCreator, helps developers to build their own enhanced STM32Cube expansion packages.

4 STDES-VRECTFD topology

The STDES-VRECTFD reference design represents a complete solution for high-power, three-phase active front end (AFE) rectifier applications based on the three-level Vienna topology.

This reference design topology is mostly used for DC fast charging applications related to industrial and electric vehicles.

It features full digital control. The embedded STM32G474RET3 mixed-signal high-performance microcontroller provides the full control of the power factor (PF), the DC voltage, and the auxiliary task to manage the protections and the soft start-up procedure.

The high-bandwidth continuous conduction mode (CCM) current regulation allows the maximum power quality in terms of total harmonic distortion (THD) and power factor (PF).

Figure 3. DC charging station

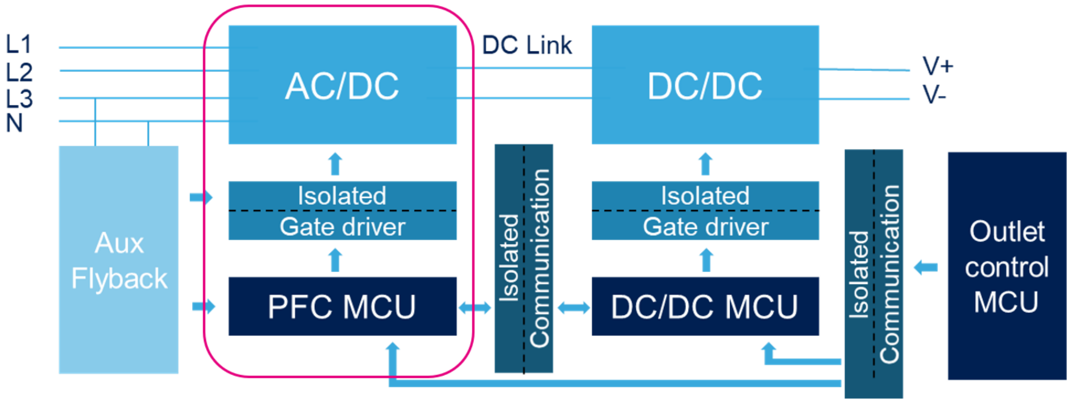
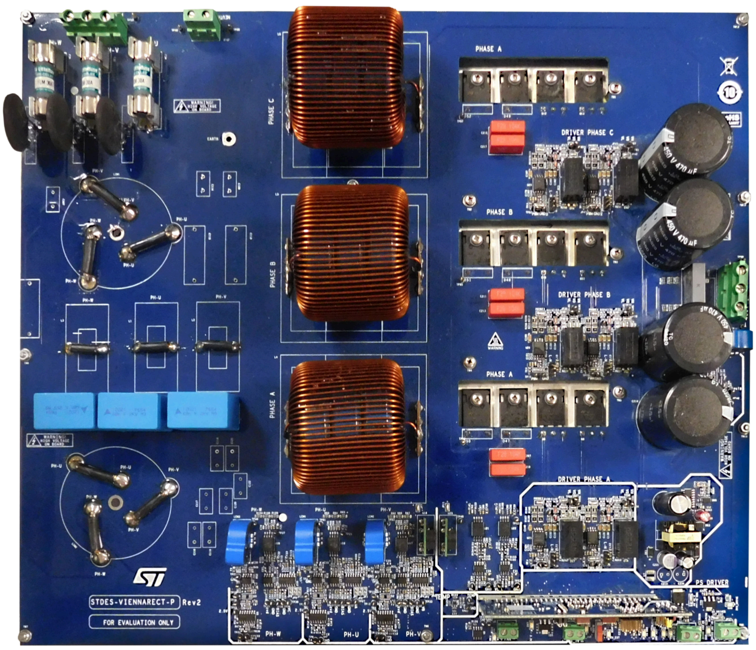
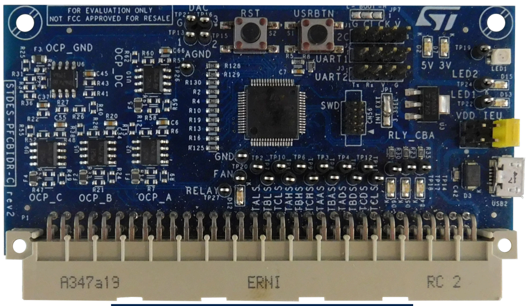


Figure 4. STDES-VRECTFD reference design - power board



Fully assembled board developed for performance evaluation only, not available for sale

Figure 5. **STDES-VRECTFD reference design - control board**



Fully assembled board developed for performance evaluation only,
not available for sale

5 STDES-VRECTFD in eDesignSuite

5.1 Overview

Thanks to the eDesignSuite development support tool, you can define the application project specifications.

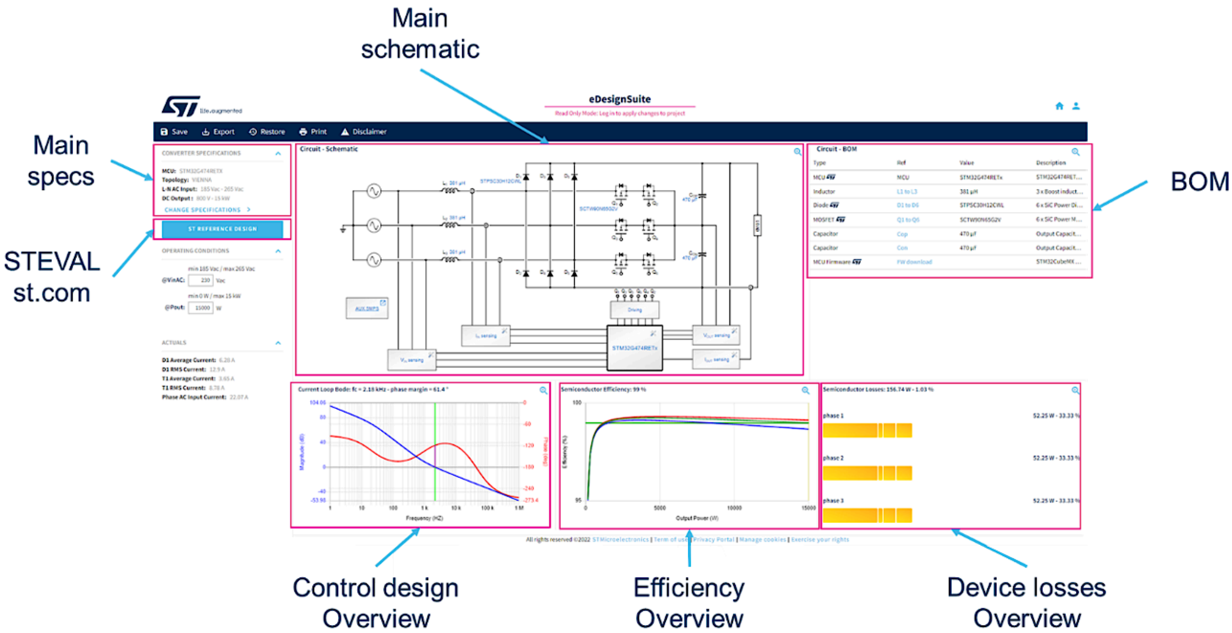
Starting from these specifications, the eDesignSuite configurator returns the hardware and software parameters necessary to get the user's desired specifications.

A complete GUI allows the user to use the tool easily.

As the first configuration, the tool proposes the project specifications of the reference design based on the STDES-VRECTFD Vienna rectifier topology.

The figure below shows the tool workspace, which highlights the main specifications of the AC-DC converter project, including its interactive wiring diagram and various additional sections proposed for the design. In addition, the BOM and interactive graphs show an estimation of the semiconductor losses and the specific control loop for the PFC application.

Figure 6. eDesignSuite workspace



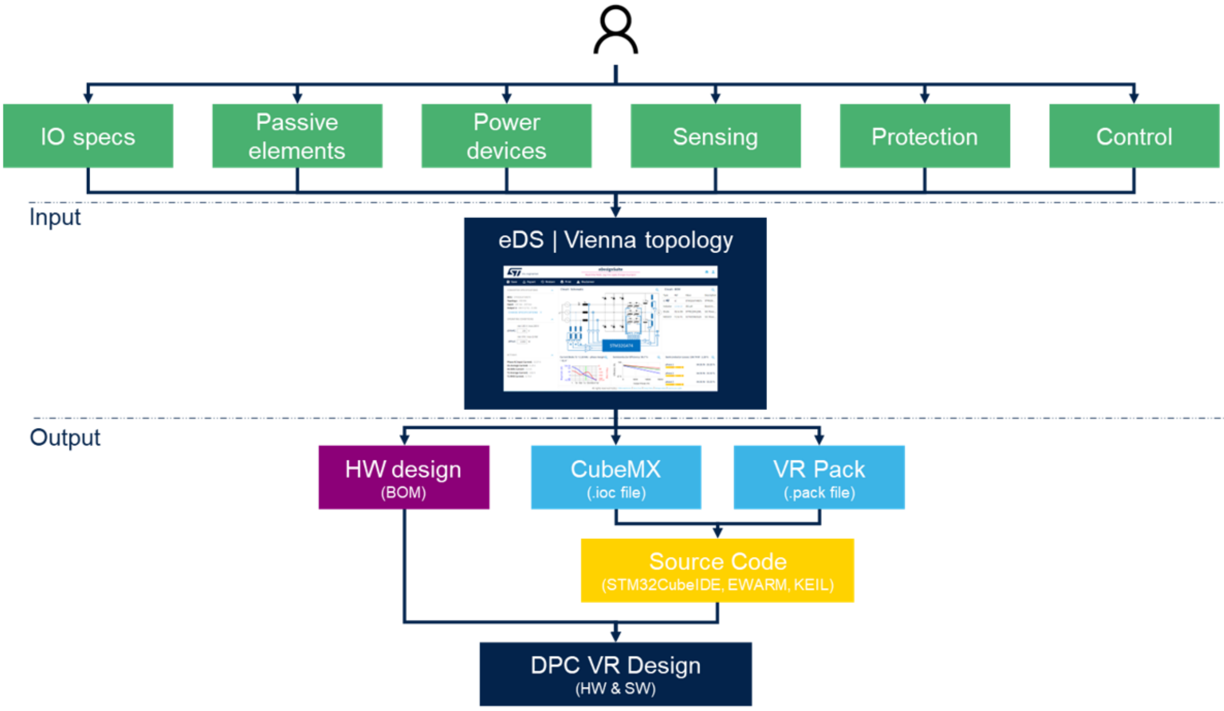
You can provide all design inputs in a sequential logic based on a very comprehensive graphic wizard. Then, eDesignSuite is able to generate the project files. The last design step provides the project files based on STM32CubeMX that allows you to generate the application source code.

eDesignSuite directly provides the parameters used by the firmware. STM32CubeMX uses these parameters to configure the application properly, according to the user's specifications.

5.2 Wizard

You can manage the main power converter design rules thanks to a step-by-step wizard. User requirements in power, signal conditioning, control, and protection sections are linked to ensure the consistency of the overall design.

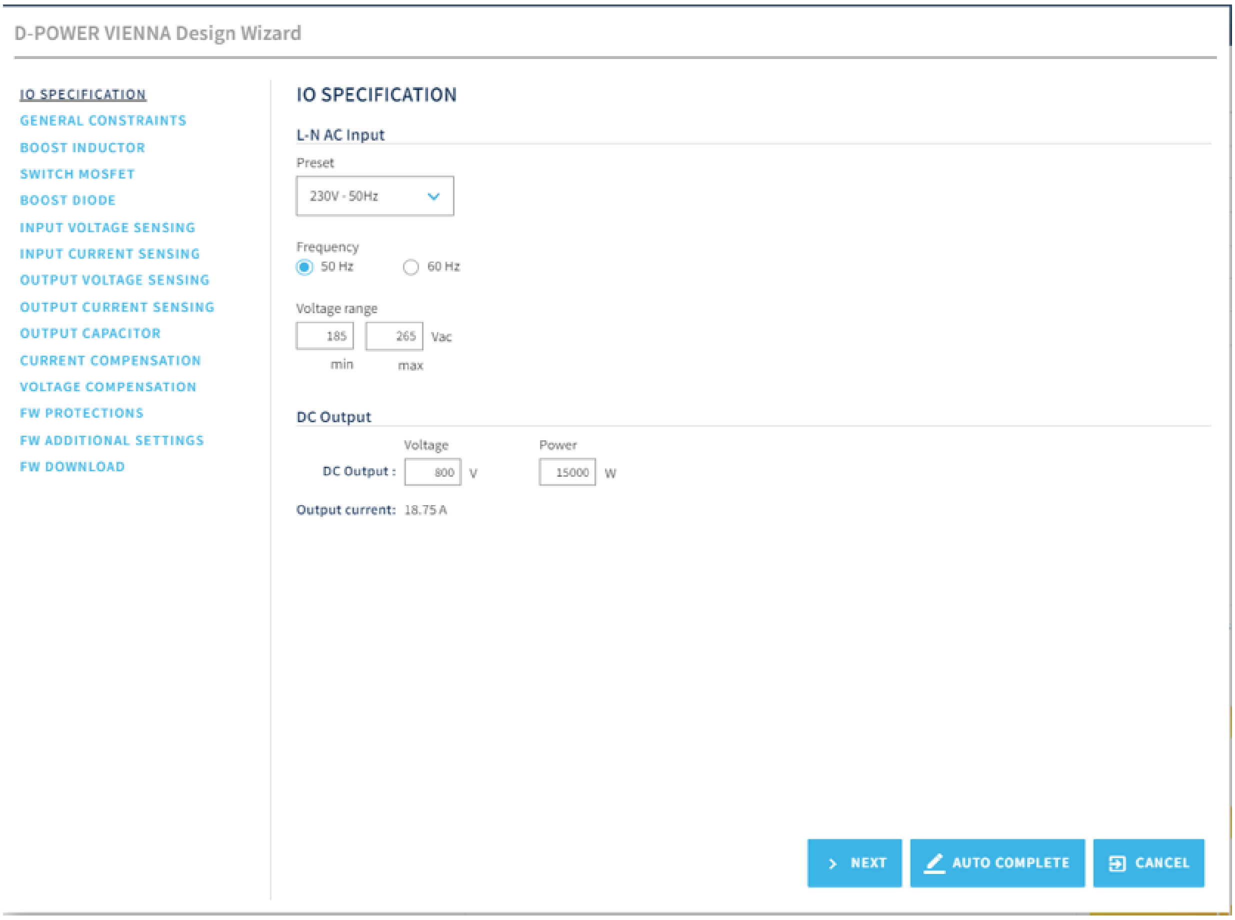
Figure 7. eDesignSuite wizard flow



5.2.1 IO specification

In this tool section, you can set the main converter input and output specifications, such as input voltage requirements and the DC voltage for the application. You can also define the output power for the efficiency estimation, the best fitted power device selection, and the passive design current rated information.

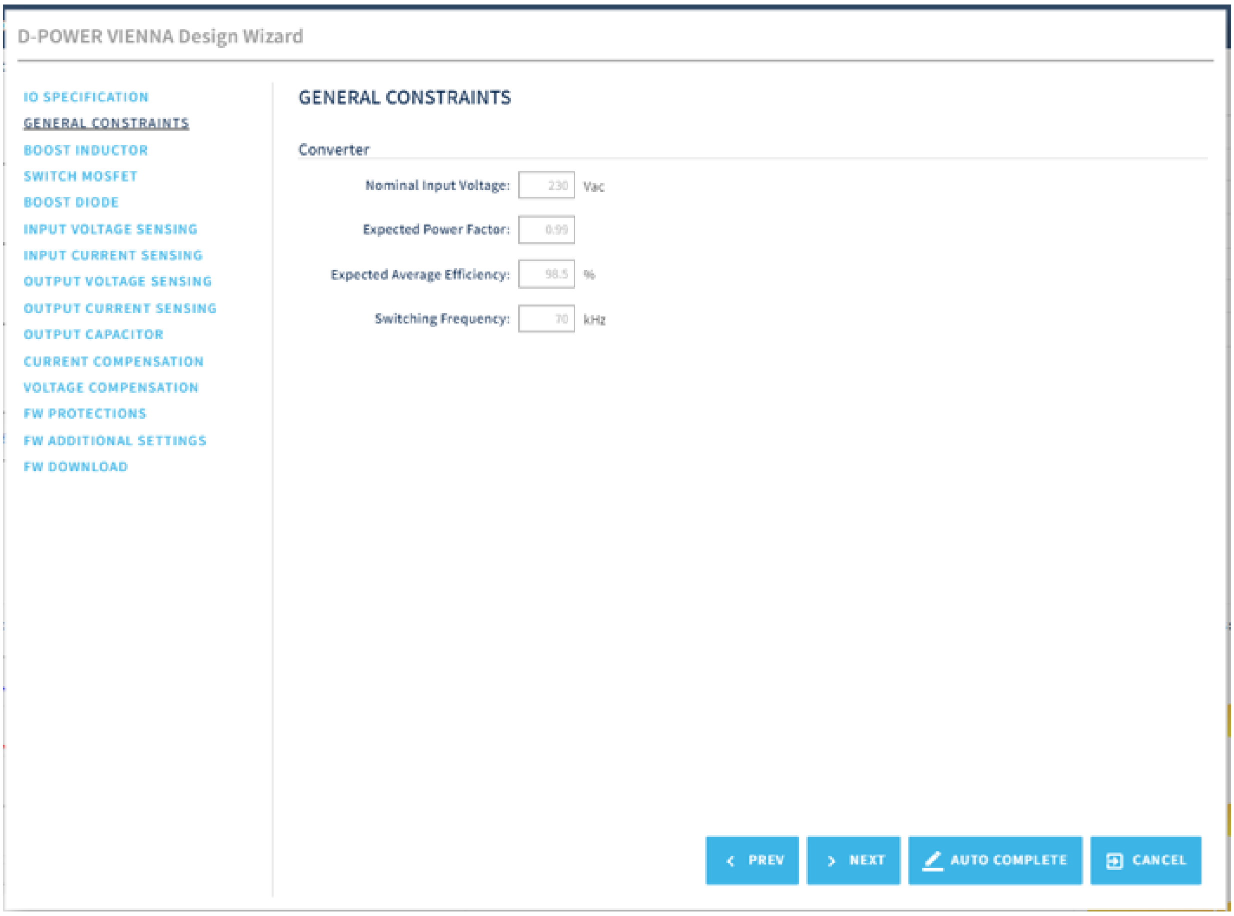
Figure 8. IO specification



5.2.2 General constraints

In this section, you can add other specifications, such as the power factor and efficiency requirements. You can also set the switching frequency.

Figure 9. General constraints



5.2.3 Boost inductor and output capacitor

Boost inductors represent the core design element for a unidirectional Vienna PFC topology. The current ripple is considered to obtain the proper inductance value. In this section, you can also force the preferred magnetic element details. The actual ripple is calculated and considered accordingly. The same design approach is used also for the output capacitors.

Figure 10. Boost inductor

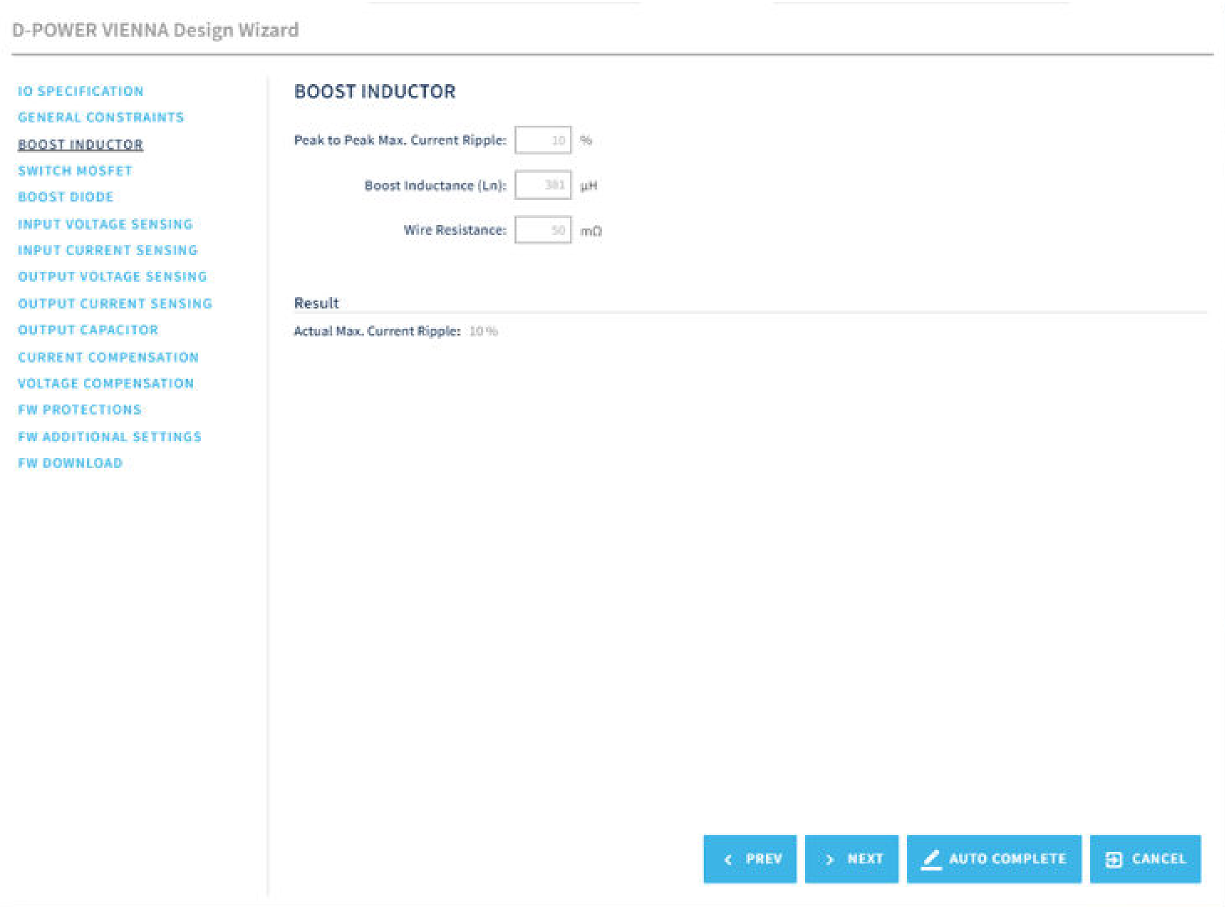
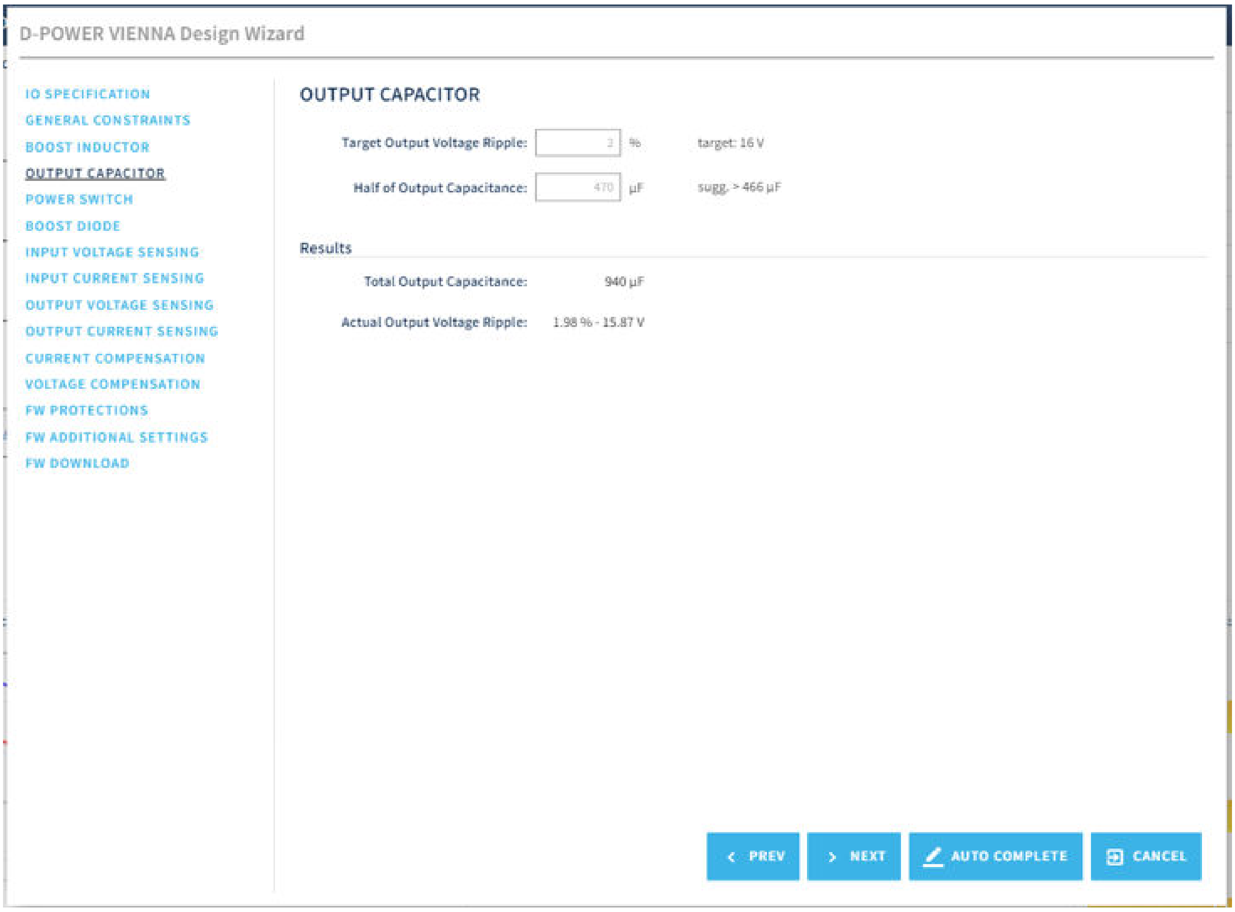


Figure 11. Output capacitor



5.2.4 Switch and diode selector

High power high switching power converters efficiency is strictly related to a good semiconductor selection. The new generation SiC technology available for diodes and MOSFETs allows you to reach a very high performance. In this section, you can compare the analysis splitting conduction and switching power losses contribution for each selection to choose the right one.

Figure 12. Power switch selector

Figure 13. Diode selector

5.2.5 Sensing design

Several feedback signals are required for power factor correction in DC voltage regulated power. Each signal must be properly scaled to optimize performance in ADC section. In eDesignSuite, a dedicated section is available for each sensing part. The tool proposes a condition circuit and BOM components according to the market availability. Custom adjustments are also available to fit a specific user-defined configuration. In addition, firmware parameters are provided or customized according to specifications.

Figure 14. AC voltage sensing

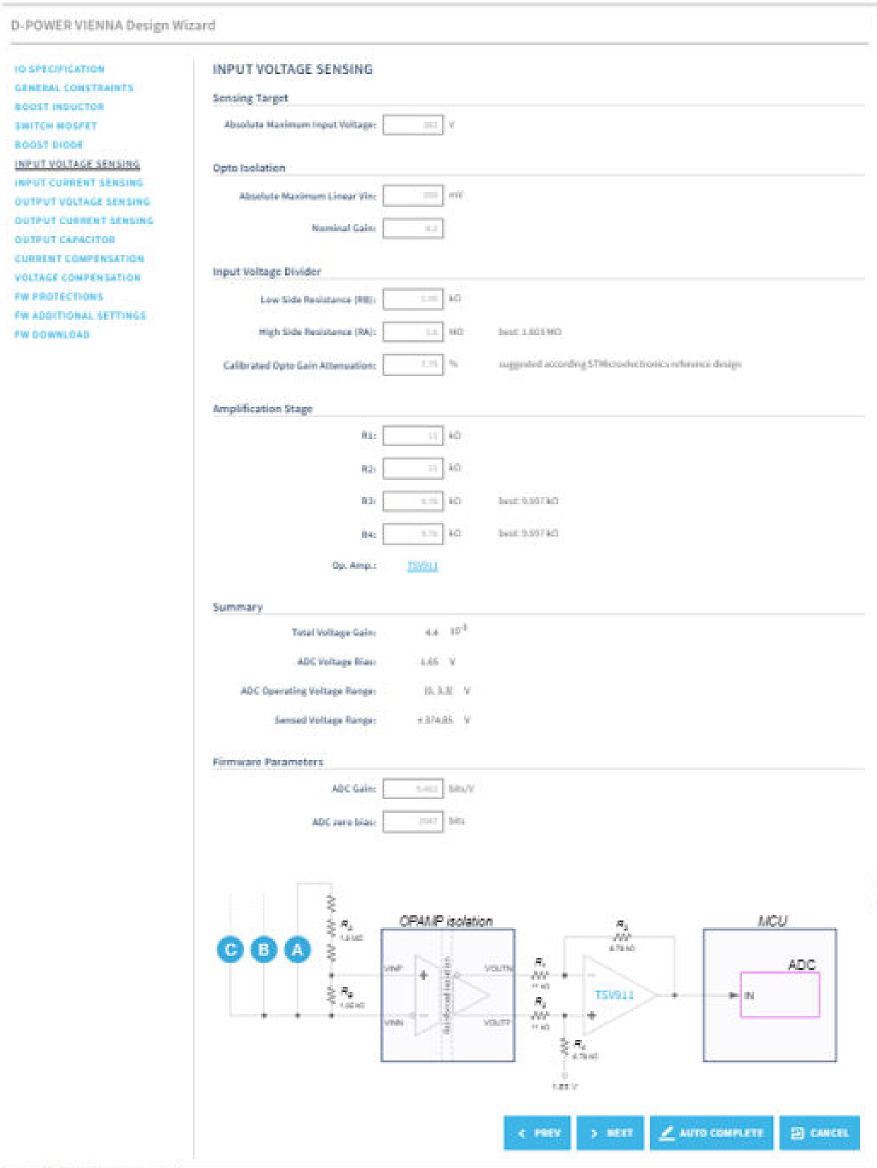


Figure 15. AC current sensing

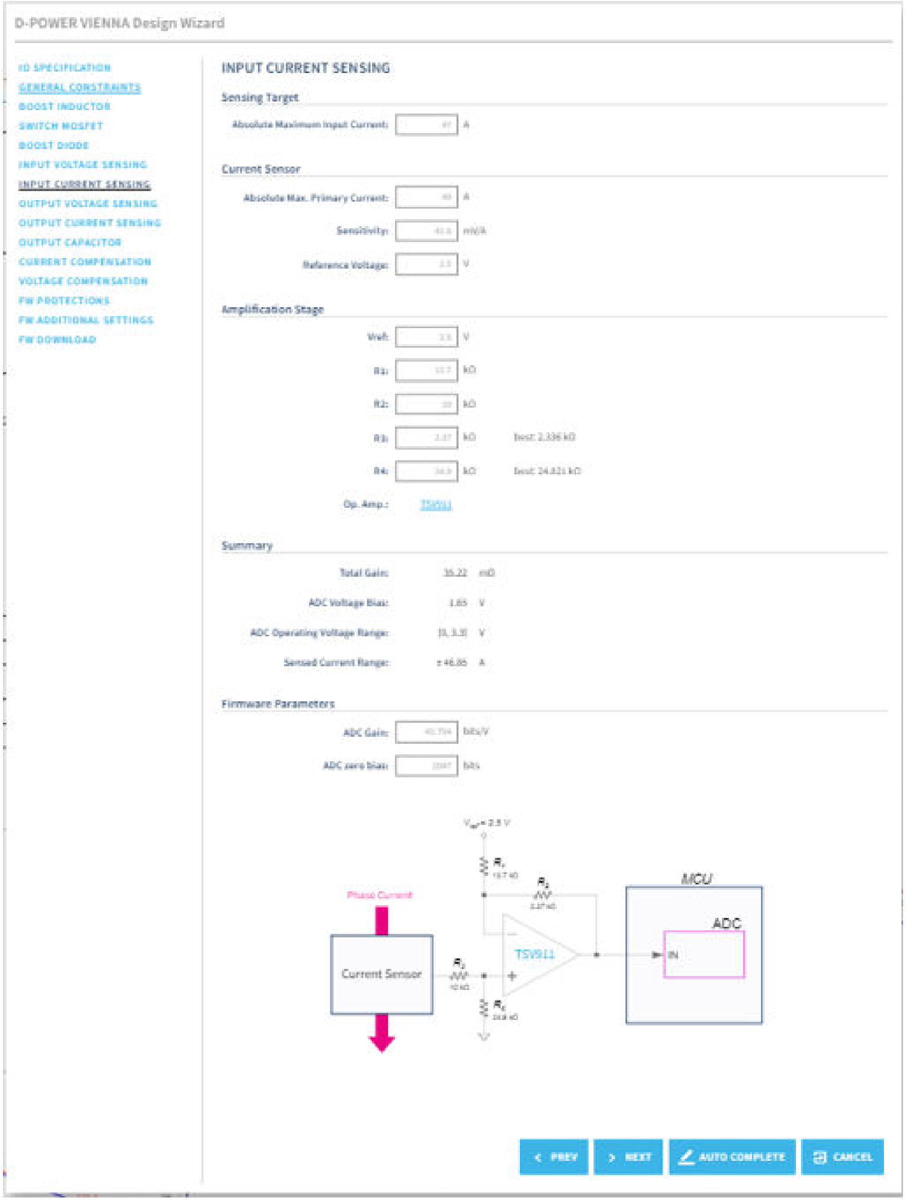


Figure 16. DC voltage sensing

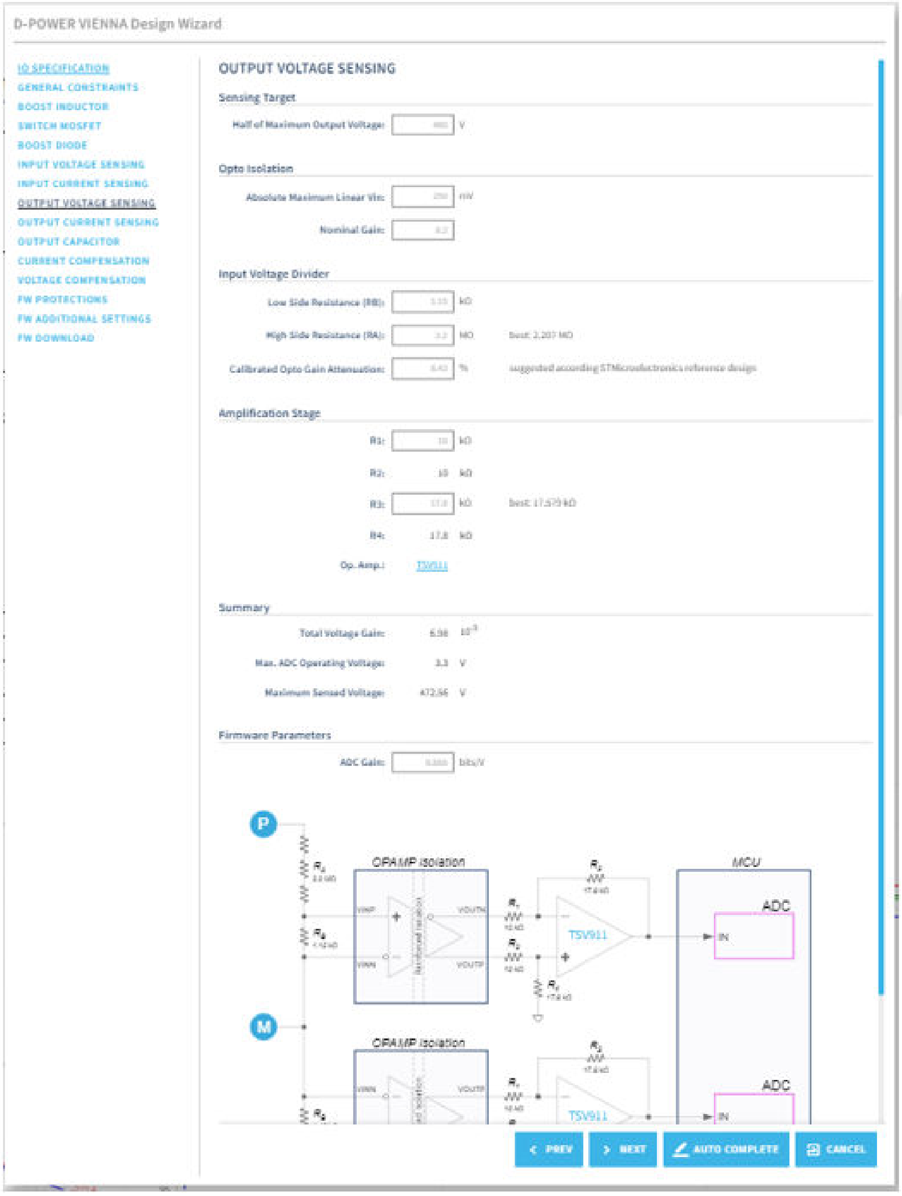
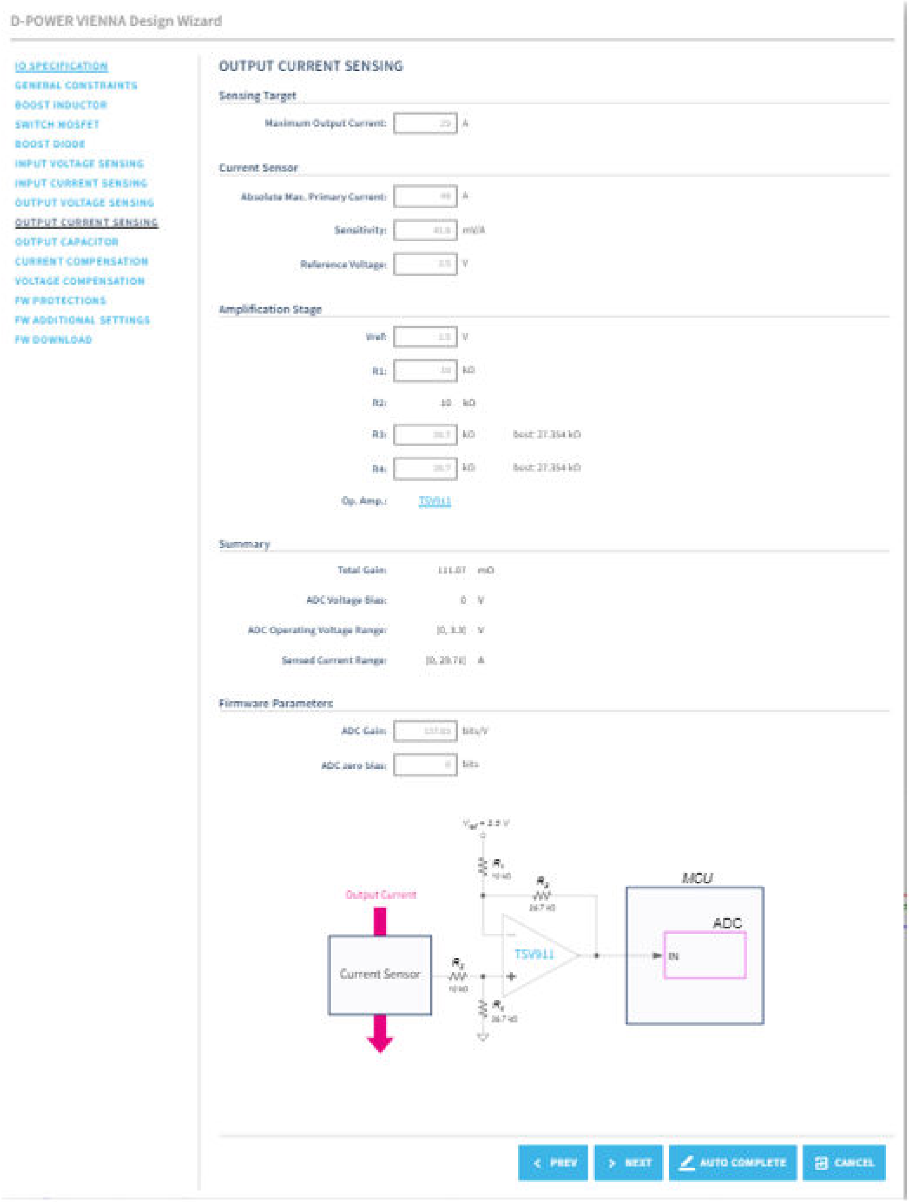


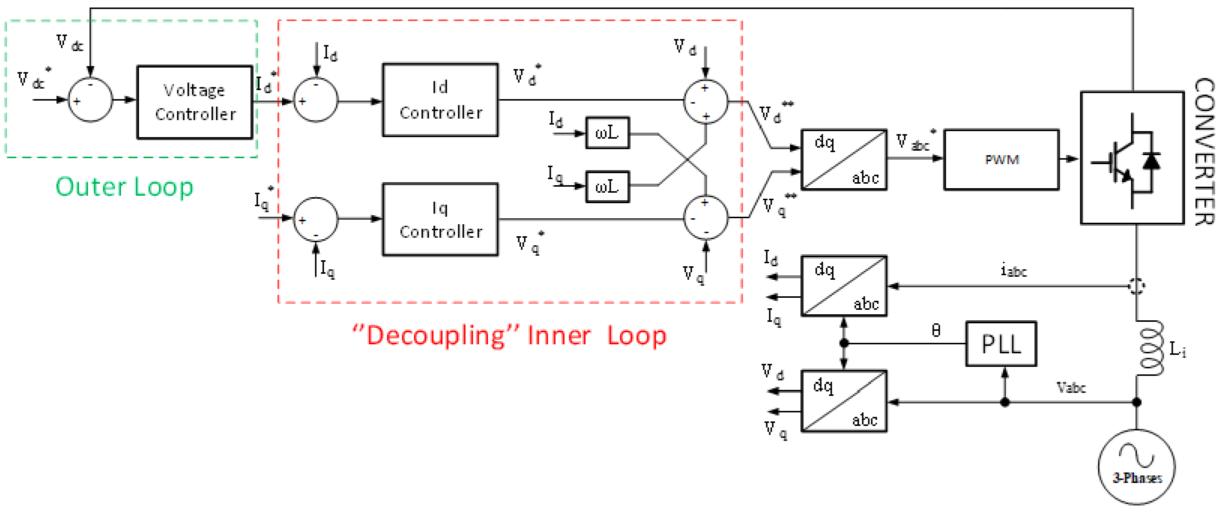
Figure 17. DC current sensing



5.2.6 Closed loop control design

The control algorithm design proposed for this power converter can be represented as a second order dynamic system, which consists of inductors and capacitors. With a good approximation, the theoretical different dynamic behavior of this two-system element allows considering two fully decoupling first order systems. For this reason, a current control and a voltage control are considered in a separate design.

Figure 18. Control loop



A current compensator to enable the PFC capability and proper operation is mandatory to obtain good results in the power quality of the power converter. Dynamic and static control behaviors are defined according to the frequency response parameters, bandwidth, and phase margin. Thanks to the integration of the model behavior of the converter topology, consistent also with the hardware design parameters, eDesignSuite allows obtaining firmware parameters accordingly.

Figure 19. Current compensation

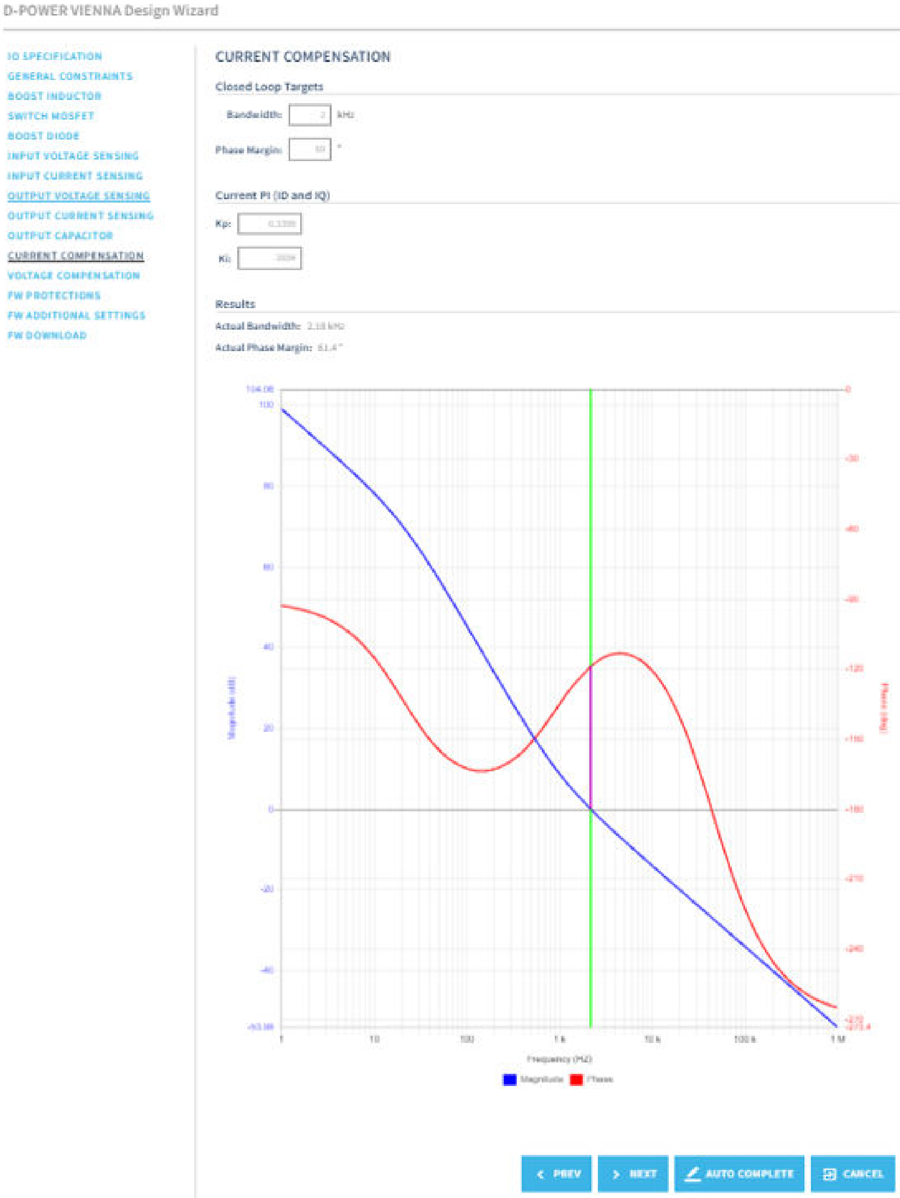
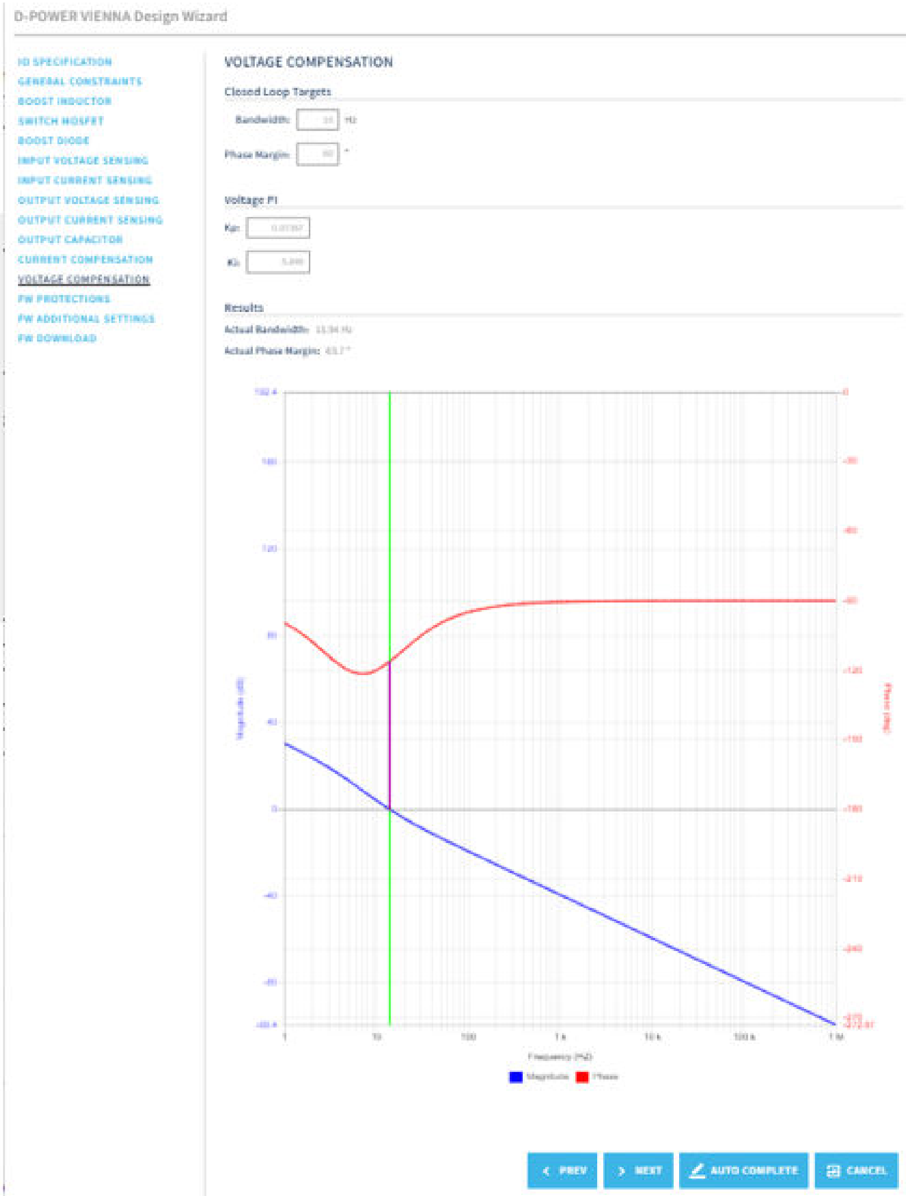


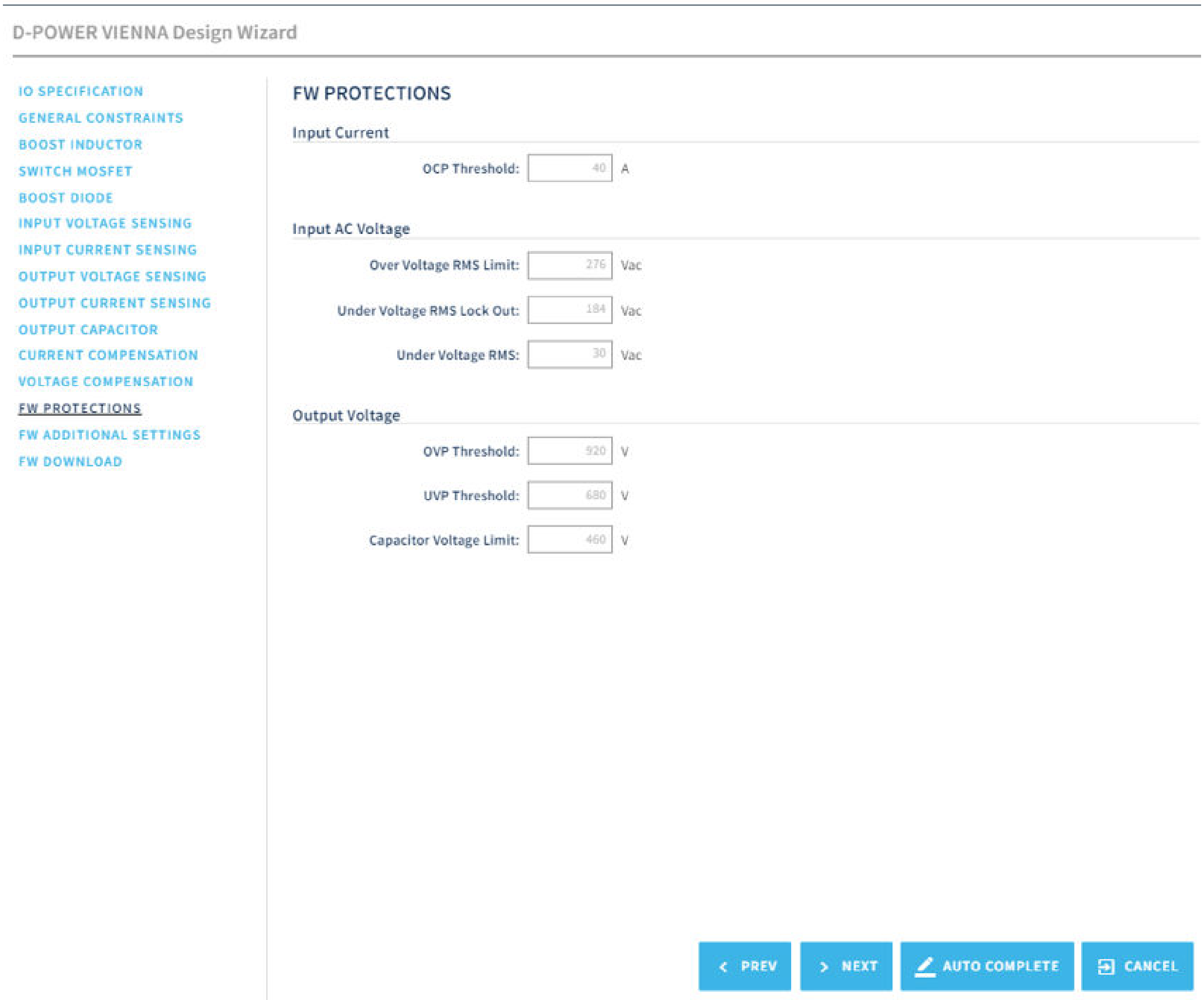
Figure 20. **Voltage compensation**



5.2.7 Firmware protections

The firmware also provides several protections that can be configured according to user specifications through the eDesignSuite user interface. Each configuration is provided to the firmware in the .ioc and .pack files.

Figure 21. **Firmware protections**

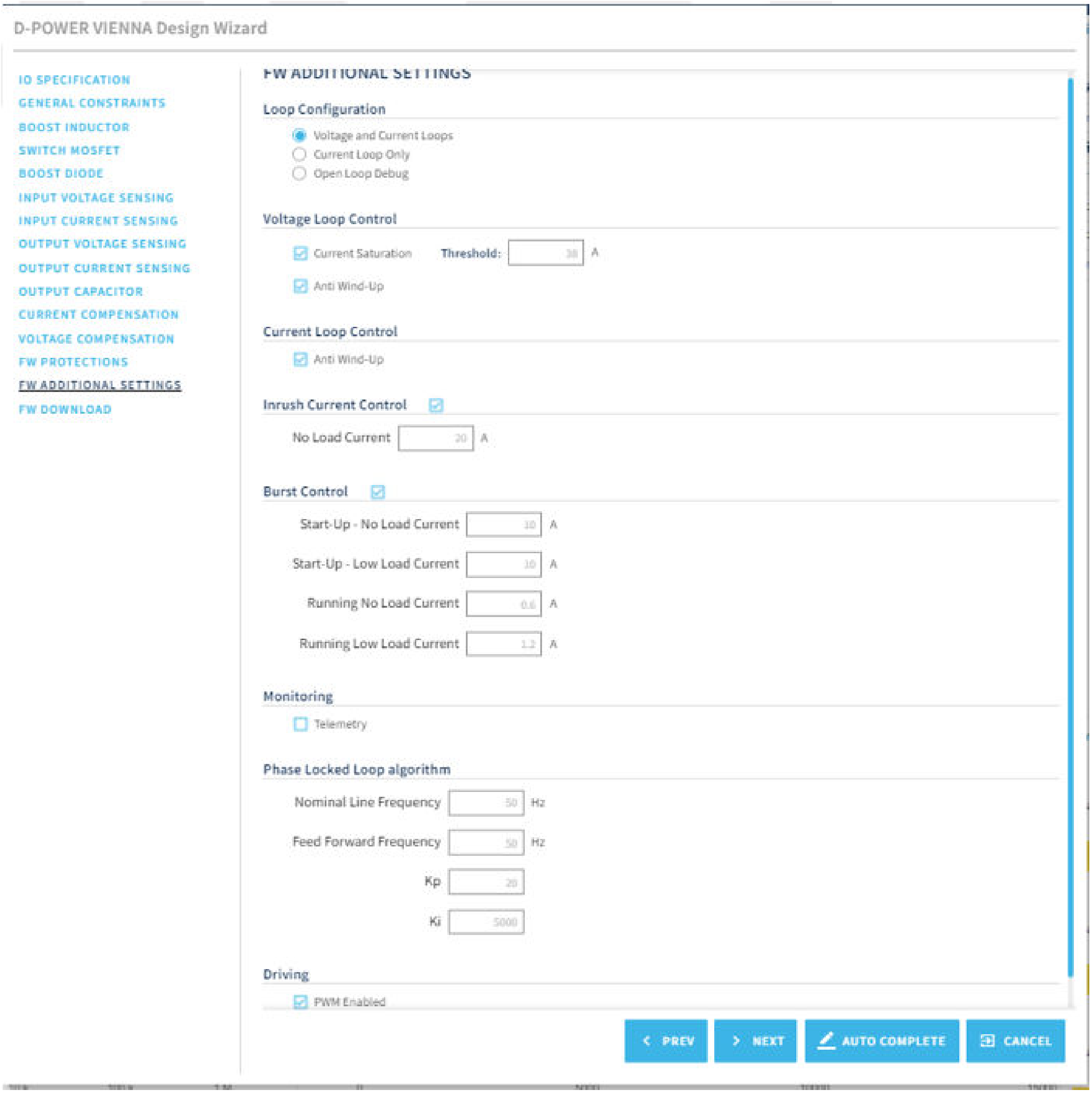


5.2.8 Additional settings

Additional settings of the power converter are available in a dedicated wizard tab.

Parameters are related to the control section startup procedure, monitoring, etc.

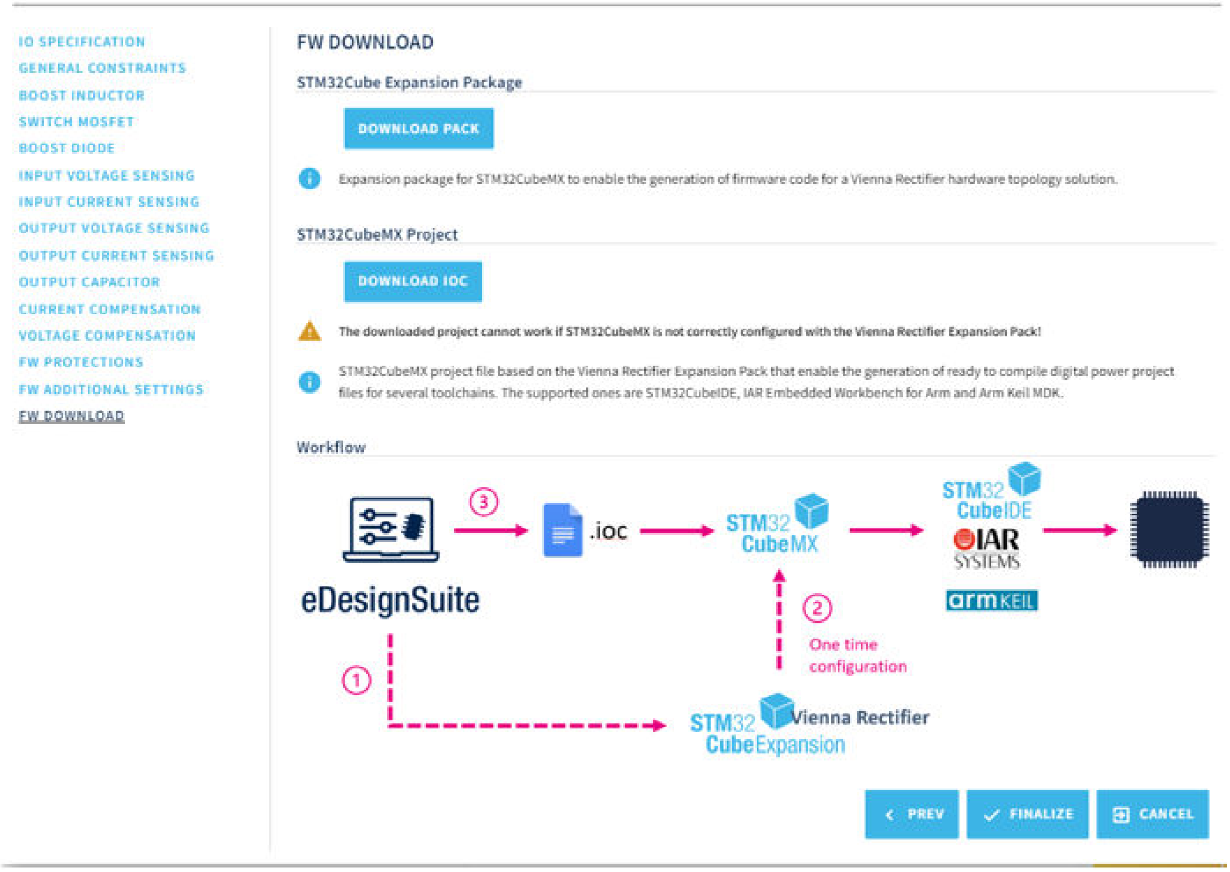
Figure 22. Firmware additional settings



5.2.9 Firmware download

This section is dedicated to the final step. You can download STM32CubeMX and the STM32Cube expansion pack according to the previous configuration.

Figure 23. Firmware download



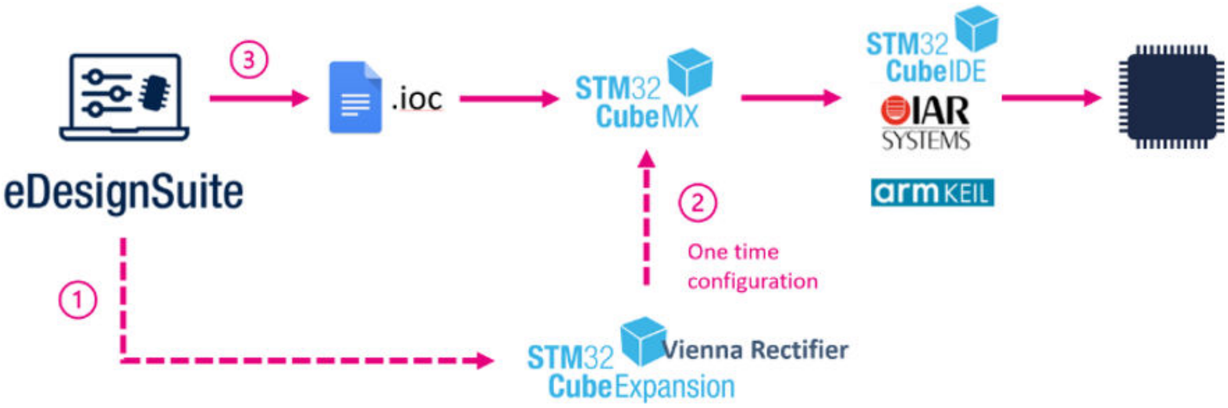
6 Design files and tools configuration

After the configuration, the wizard last section is dedicated to the design files handling.

Two files are deployed by the cloud-based application. These files are used locally by ST tools.

The workflow is designed to download and install the STM32Cube expansion pack into STM32CubeMX. This step is required only the first time to add the support of this specific topology. After this, STM32CubeMX can work properly with the IOC file that contains the actual configuration of the custom design.

Figure 24. eDesignSuite workflow

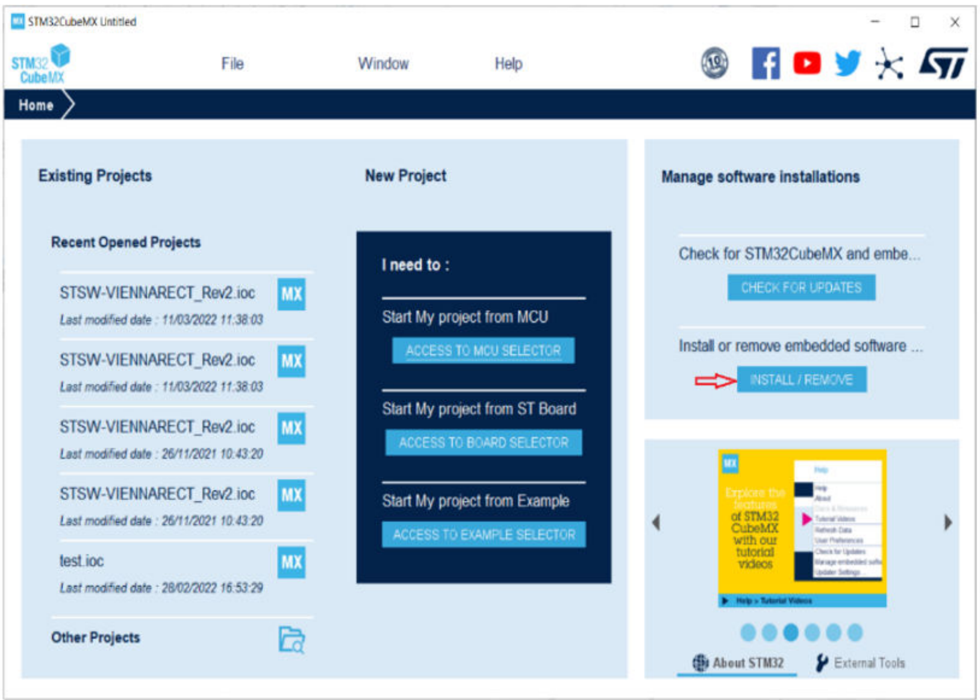


6.1 How to add Vienna topology in STM32CubeMX

Step 1. Open STM32CubeMX.

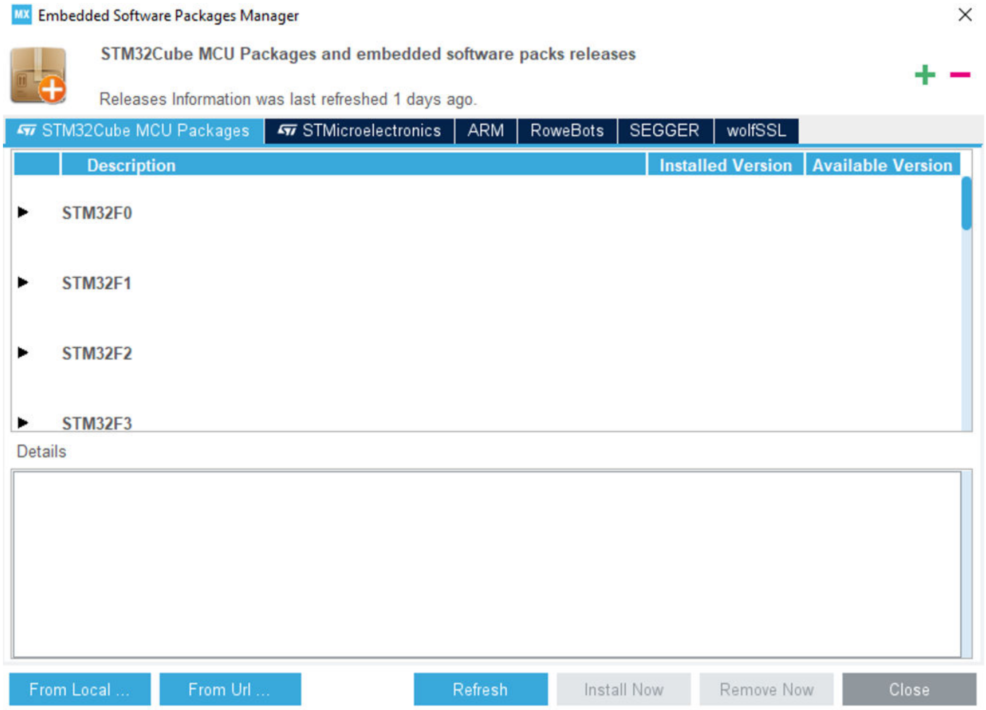
Step 2. Open "Install/Remove" wizard.

Figure 25. STM32CubeMX home page



Step 3. Click **Install/Remove** to open the Embedded Software Package Management page.

Figure 26. Embedded Software Packages Manager page



Step 4. Drag and drop the file or select the software package from a local file.

Figure 27. Drag and drop the software pack

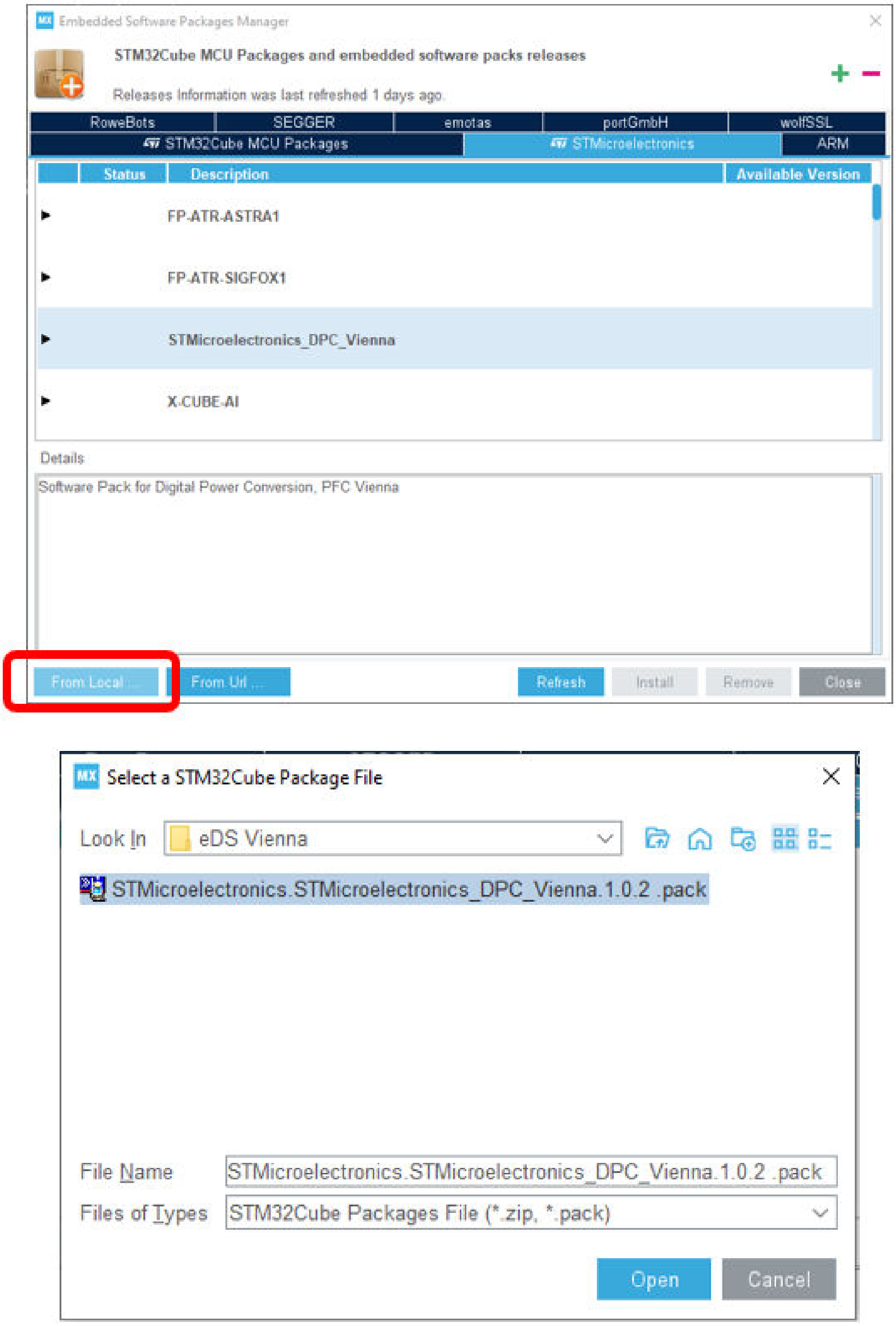
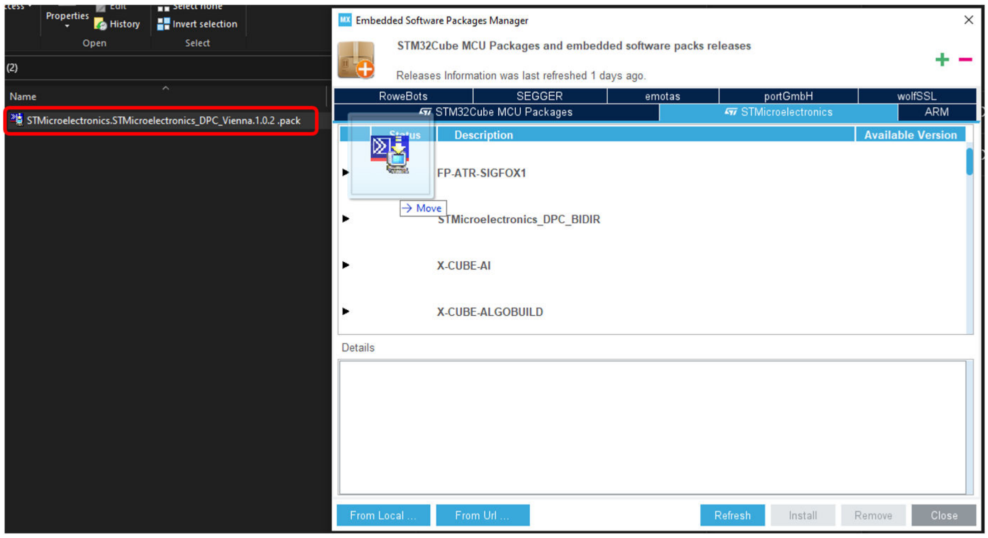


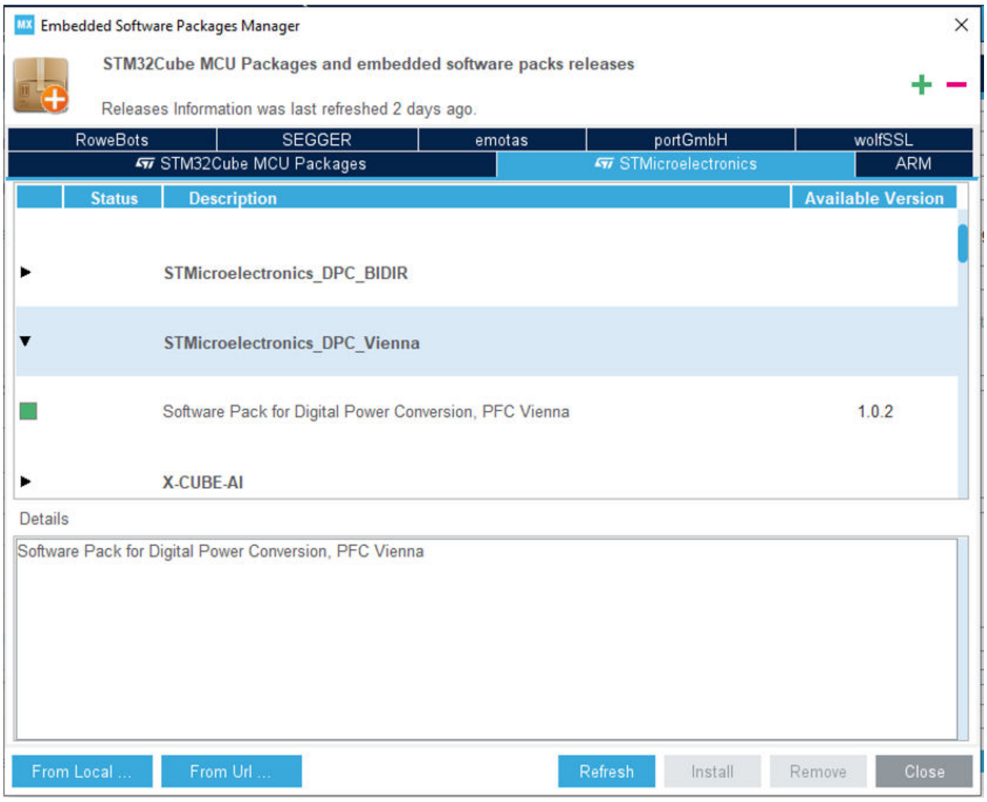
Figure 28. Select the software package from a local file



After the installation, the software pack is available in the packs areas.

Step 5. To verify that the installation was successful, select “Software Pack Component Selector” in the Software Packs section of STM32CubeMX.

Figure 29. Embedded Software Package Manager



Step 6. Download or open eDesignSuite Vienna STM32CubeMX "ioc" file.

This section is automatically populated according to the eDesignSuite user configuration.

Note: *All eDesignSuite system configuration parameters are collected and proposed in the STM32CubeMX configuration tab. Do not change them directly from this tab to avoid unexpected power converter behaviors.*

Figure 30. Vienna ioc file - Software Packs selector

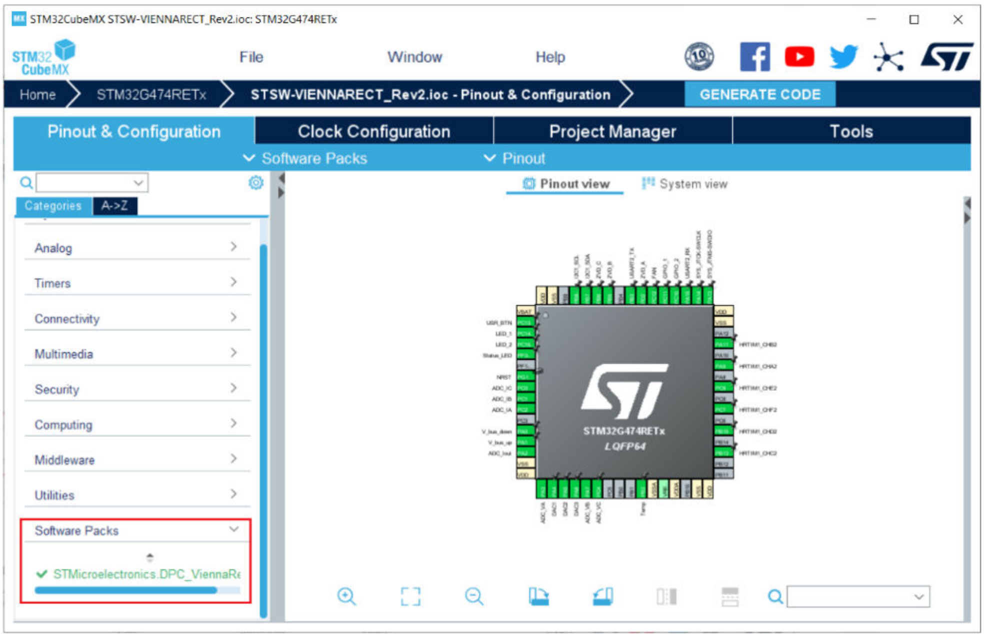
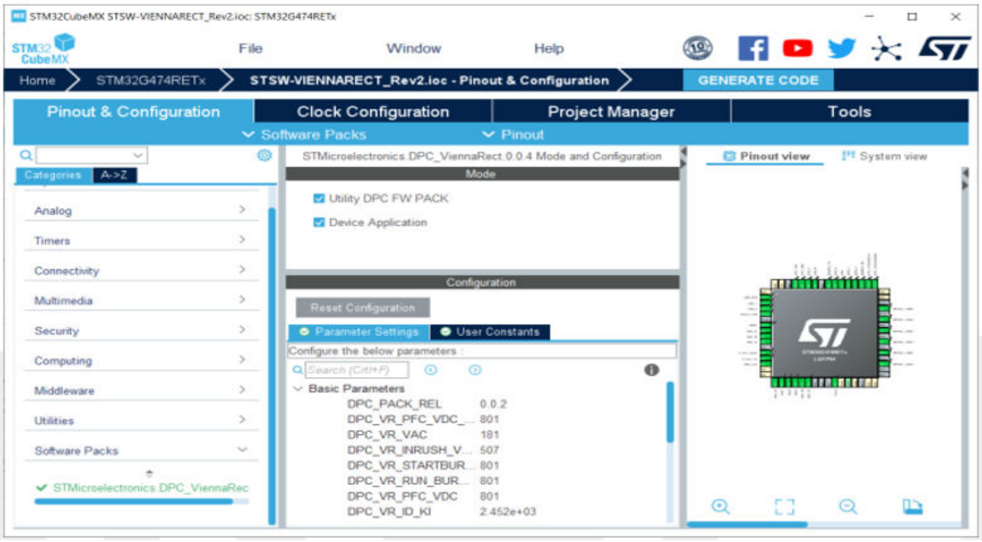


Figure 31. Vienna topology STM32CubeMX parameters



6.2 STM32CubeMX parameter description

All parameters are shown in parameters setting. Several sections are available and collect related parameters.

6.3 Generate toolchain project

The STM32Cube firmware projects are compliant with the requirements hereafter regarding the supported integrated development environments (IDEs):

- STMicroelectronics STM32CubeIDE
- IAR Systems EWARM
- Keil® MDK-ARM IDE

Figure 32. STM32Cube project deployment

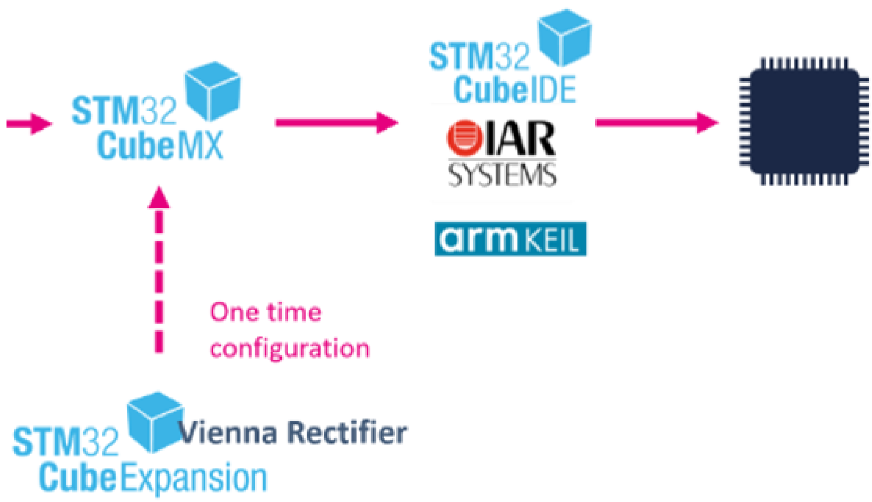
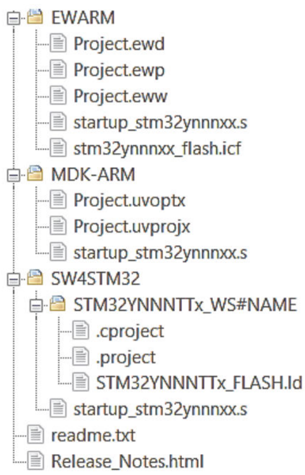


Figure 33. IDE folders in the STM32Cube package



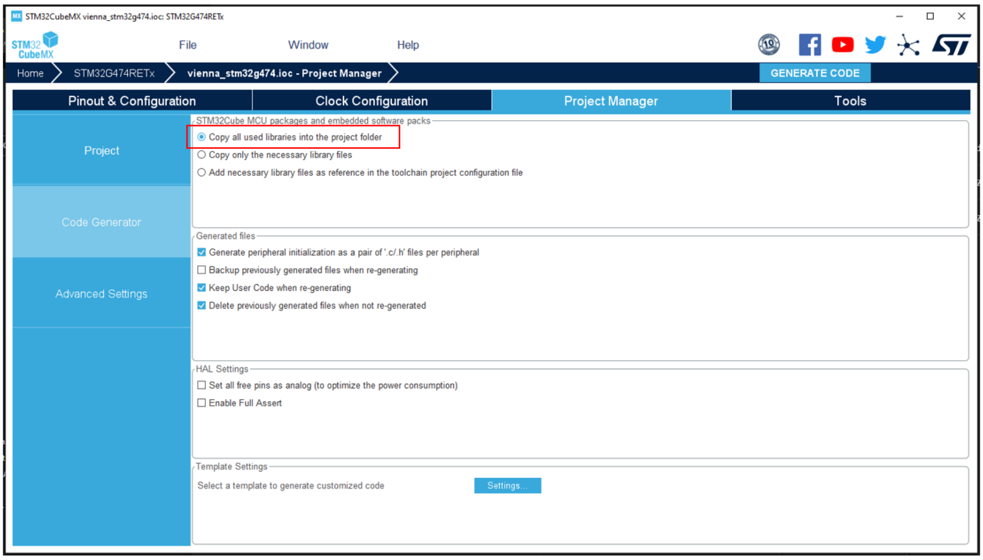
6.4 STM32CubeIDE application from STM32CubeMX

Step 1. Select STM32CubeIDE in Project Manager→Code Generator.

Step 2. The CubeMX project generated by eDesignSuite is already configured to use STMicroelectronics STM32CubeIDE toolchain. In case of other manual toolchain selection, verify "Code Generator" and "Project" sections in the Project Manager tab.

Copy all used libraries into the project folder and save the project.

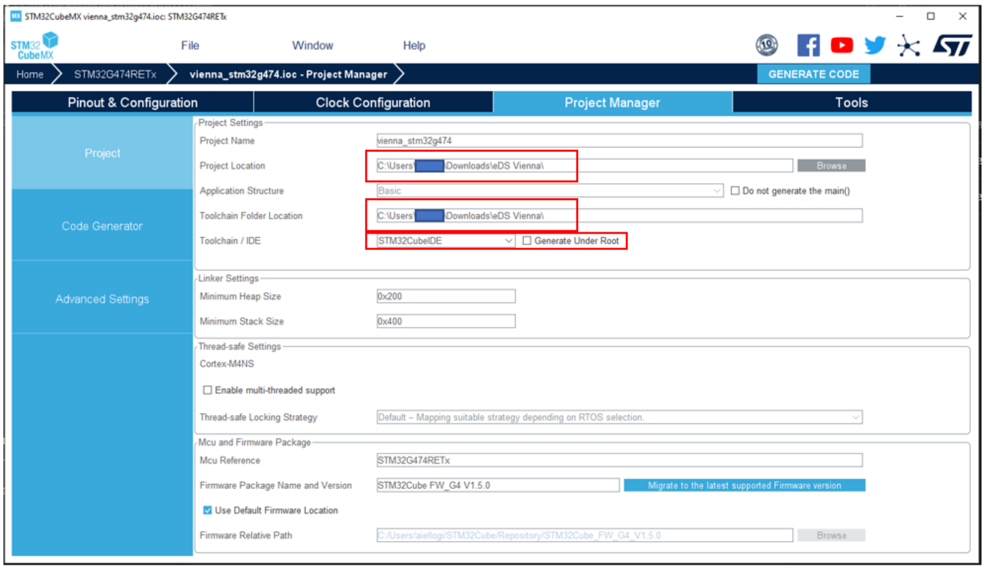
Figure 34. Project manager tab



After saving the project, the project location section and in the toolchain folder location section should have the same path.

Note: *If these sections have different paths, a wrong project creation phase can occur.*

Figure 35. Expected status



If the project creation phase is successful, you can run the generated code and open the project.

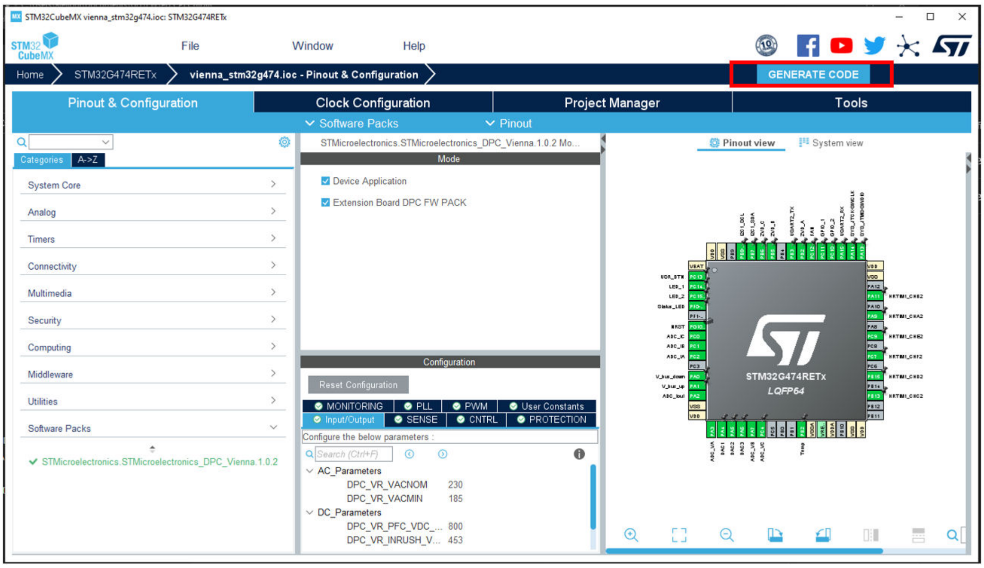


Figure 36. Generating the code

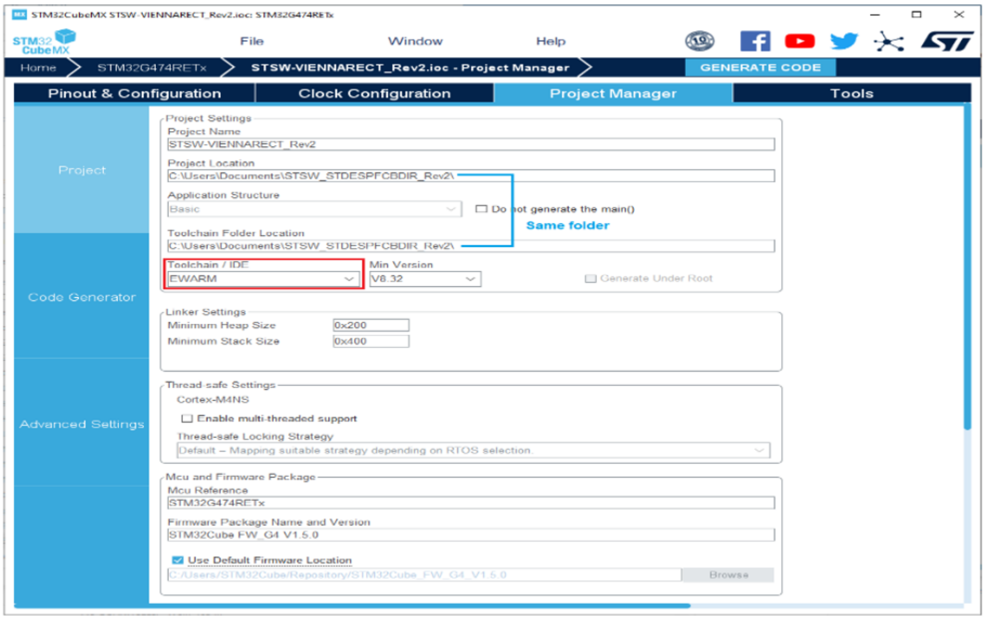
6.5 IAR Workbench application from STM32CubeMX

Step 1. After the software pack installation, move the .ioc file into a new blank folder and open it. Load Application file.ioc from local folder and select the Software Pack needed.

Step 2. Select the correct Toolchain (EWARM) in Project Manager→Code Generator.

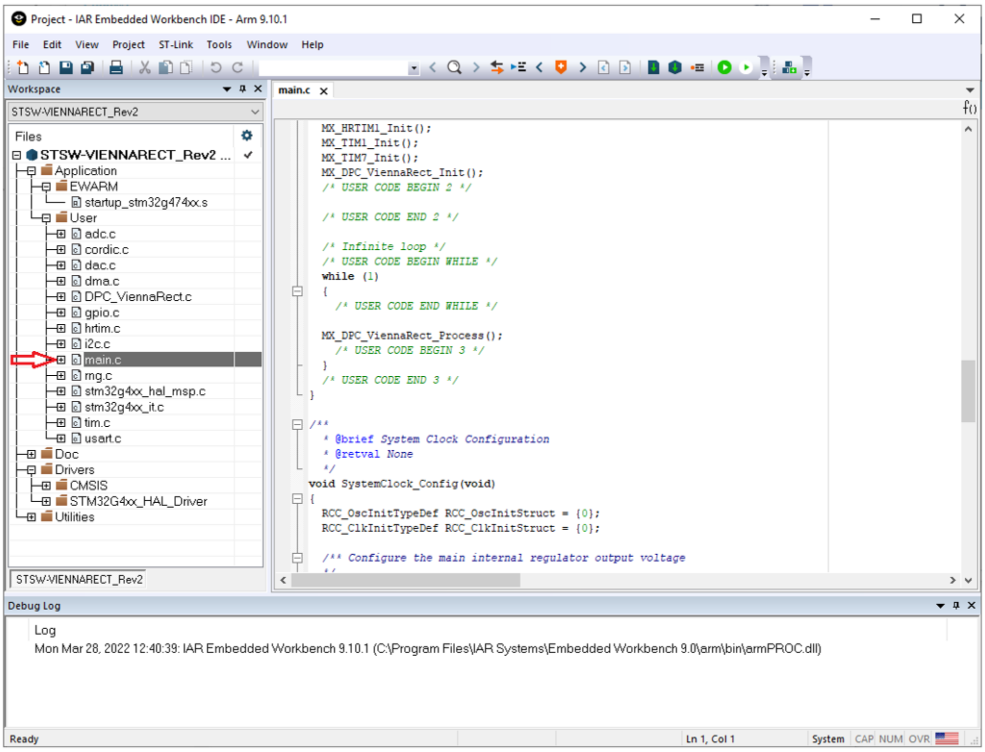
Step 3. Move to Project Manager tab and verify in "Code Generator" that "Copy all used libraries into the project folder" is selected and in "Project" sections that path location are the same.

Figure 37. EWARM configuration



If the project creation phase is successful, you can run the generated code and open the project.

Figure 38. IAR EWARM toolchain



7 STM32Cube Vienna expansion packaging requirements

The STM32Cube package is the backbone of any STM32Cube expansion package. As a result, the folders and file structures must always be organized without modifying the original folder structures as described in "How to develop an STM32Cube Expansion Package" in the STM32 MCU wiki page as well as within the development guidelines for STM32Cube firmware packs user manual.

The STM32Cube expansion package development checklist document provides the list of requirements to follow when building an STM32Cube expansion package.

7.1 Development with STM32CubeMX

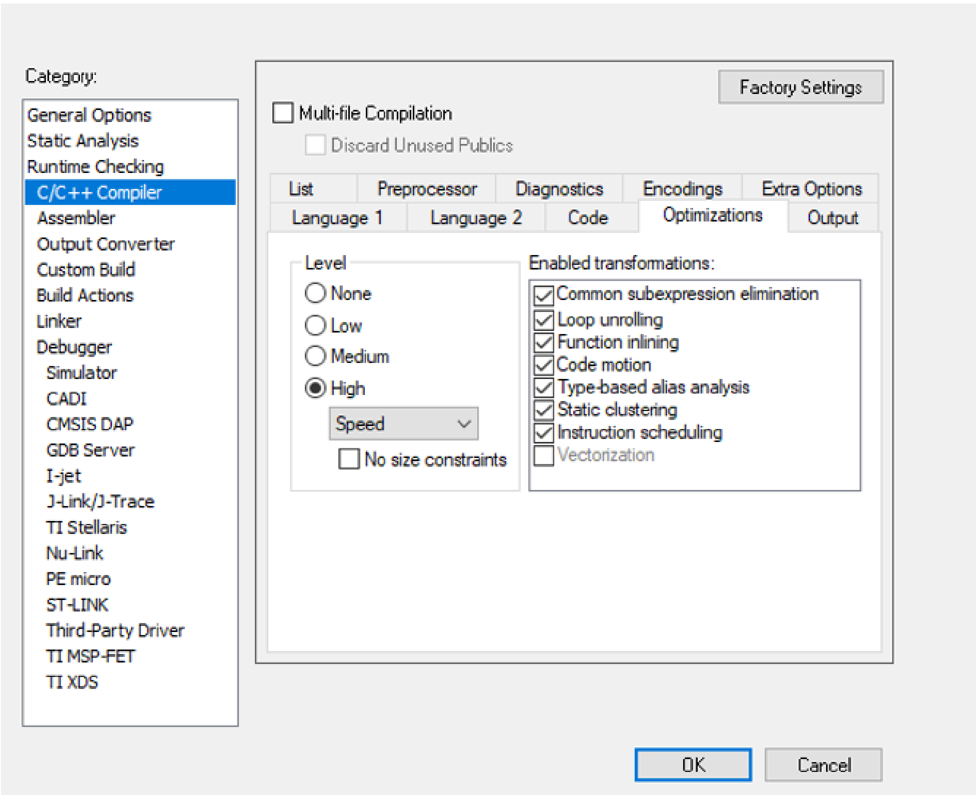
In an STM32Cube expansion package, application must be developed using the STM32CubeMX tool. This requirement implies that the development satisfies the following rules:

- the STM32CubeMX tool must be used to configure the STM32 device and board, and generate the corresponding initialization code
- in the generated *.h and *.c source files, the user must add the applicative code within the sections limited by the /* USER CODE BEGIN */ and /* USER CODE END */ markers
- the associated **.ioc* file must be available at the example root
- if additional STM32CubeMX project settings are used, the *.extSettings* file must be kept at the same level of the **.ioc* file

7.2 Compiling optimization requirements

According to the IDE selection, you must verify the optimization configuration, setting the high level with speed priority.

Figure 39. Building optimization using EWARM



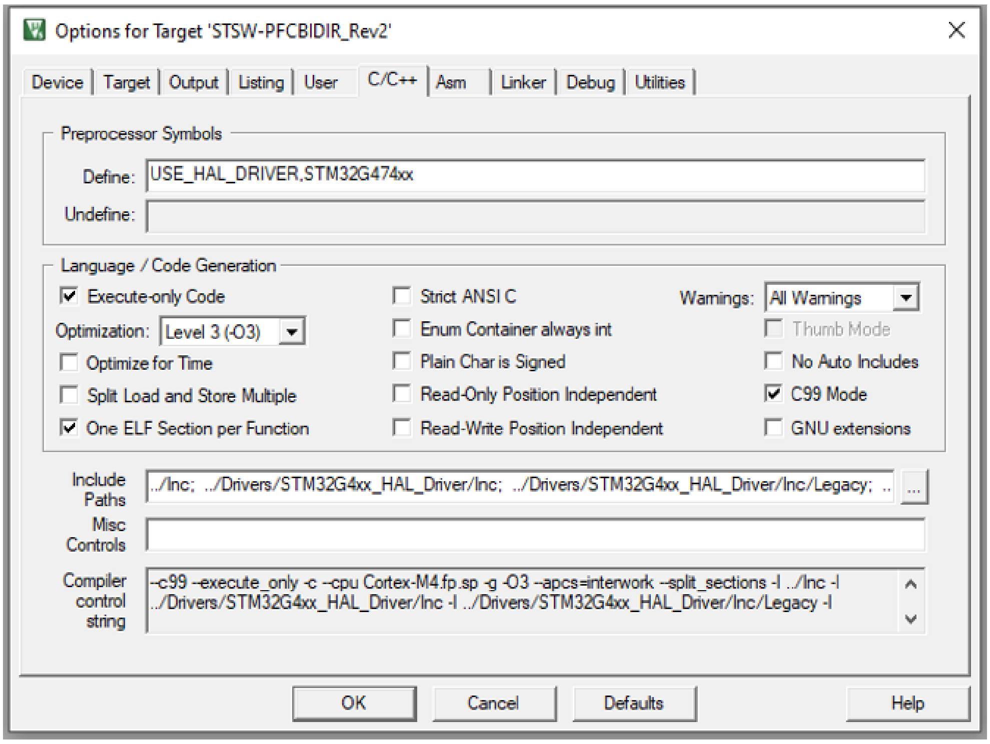


Figure 40. Building optimization using Keil

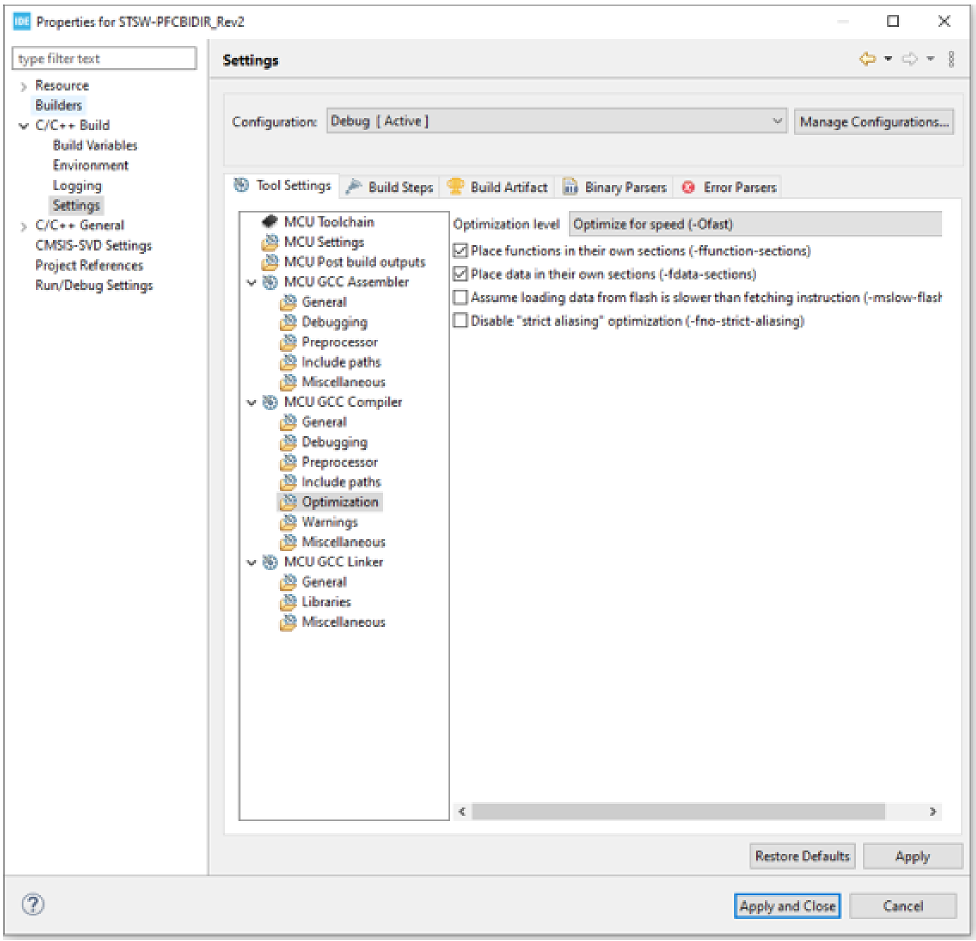


Figure 41. Building optimization using STM32CubeIDE

Revision history

Table 1. Document revision history

Date	Revision	Changes
01-Feb-2023	1	Initial release.

Contents

List of tables

List of figures