iMOTION Solution Designer

User Guide

About this document

Scope and purpose

The purpose of this document is to guide the user with the **iMOTION™ Solution Designer** (iSD) tool usage in the iMOTION™ motor control development process.

Intended audience

This document is written for iMOTION™ users who are dealing with the development process.

Table of contents

Table of contents

Table of contents

Table of contents

1 Introduction

1.1 Purpose

The **iMOTION™ Solution Designer** (iSD) is an integrated GUI-based tool for use in the iMOTION™ motor control development process. This program replaces the MCEWizard and MCEDesigner, and act as new all-in-one tool for iMOTION™ users.

The purpose of this document is to help familiarize the user with the iSD and its suite of all-in-one tools. By the end of this document, the user will be familiar with **Project Creation**, the **Configuration Wizard**, the iSD **Programmer**, the **Dashboard**, the **Oscilloscope**, **Data Capturing**, and **Data Streaming**. Additionally, users will be able to effectively use the **Script Editor** and **Script Debugger** tools where necessary.

2 Project creation

2.1 Welcome Screen

The **Welcome Screen** is the landing page for the iSD, acting as the first page users will see upon starting the program.

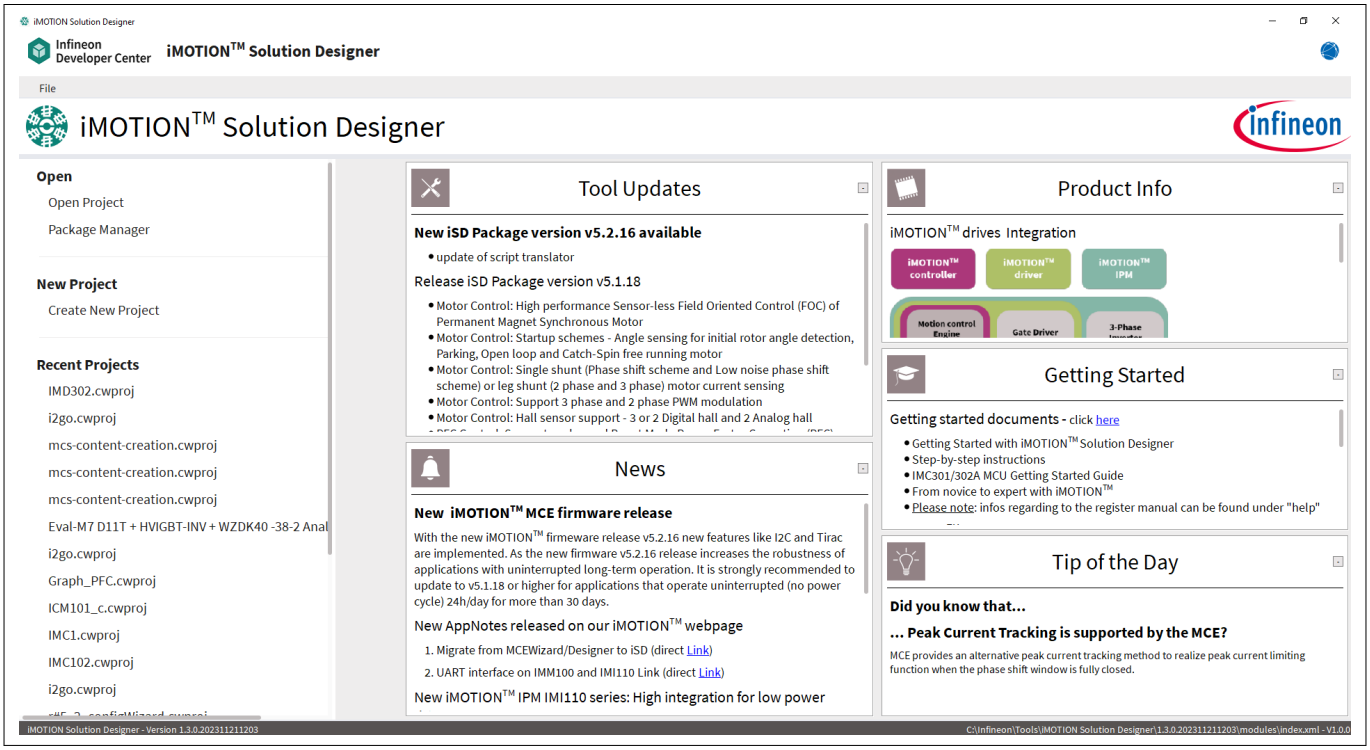

Figure 1 Solution Designer Landing Page

2.1.1 File

Located at the top left of the iSD Welcome Screen, **File** allows the user to access the basic functions of project and file management.

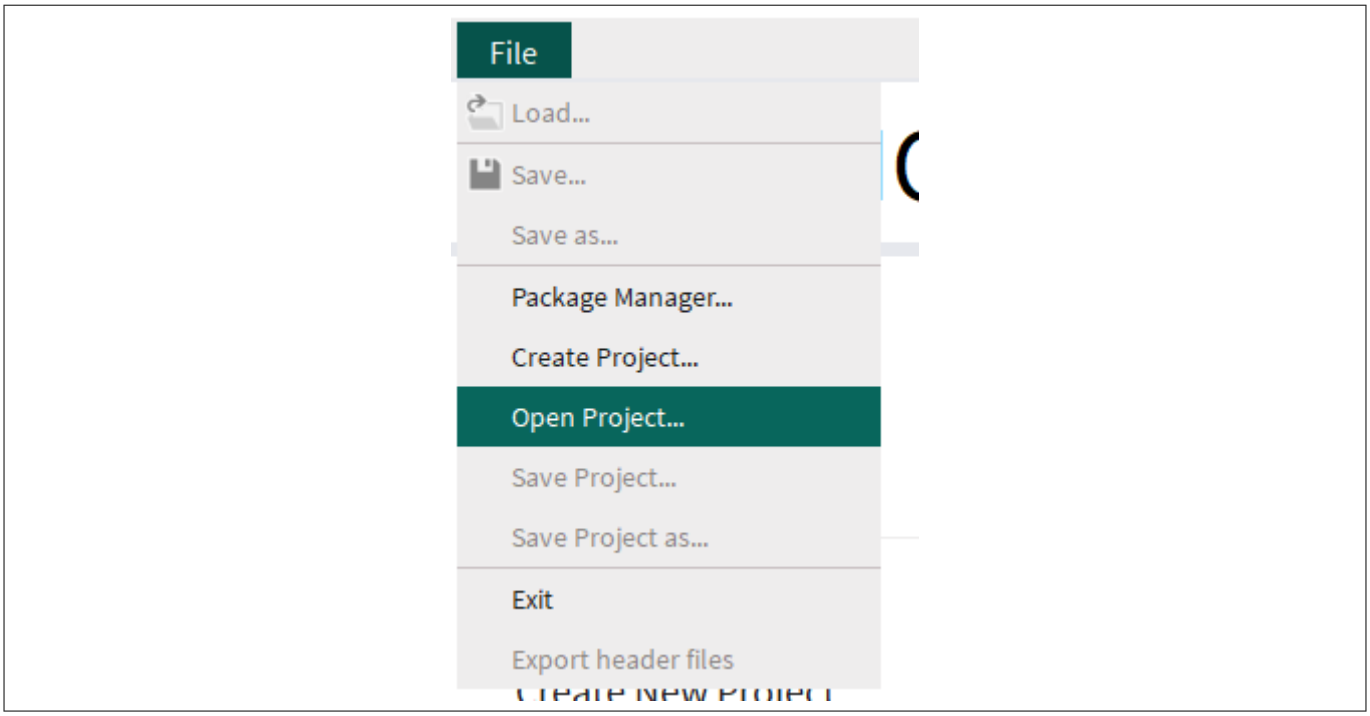

Figure 2 File Drop-Down Menu

- **Load, Save, Save as…:** Allows the user to load and save project configuration files.
- **Package Manager:** Allows the user to install SD Packs, either from the Infineon network or from a local destination.
- **Create Project, Open Project, Save Project, Save Project as…:** Gives the user full control over project file management. Only Package Manager, Open Project, and Create Project will initially be available.

2.1.2 Navigation

The Navigation pane allows users to quickly access their project files. Users can **Open Projects** and access the **Package Manager**. Users can opt to **Create a New Project**. The Navigation pane will also display a number of **Recent Projects** available locally.

2.1.3 News, Tool Updates, Product Info, Getting Started, and Tip of the Day

These panes are regularly updated with information about upcoming and available updates to the iSD and SD Packages, information on upcoming iMOTION™ products, updates to tools within the iSD, and general tips to help users better manage their projects.

2.1.4 Version Number and Installation File Path

The information bar located at the bottom of the screen displays the current iSD version number and program installation path. The user can use this in conjunction with the News pane to see if their iSD is up-to-date, and they can navigate to the file path listed to manually access the program files as needed.

When a project is opened or created, the information bar will show the connection status of the device to the left and the project's installation path to the right.

***Note**: When looking at the version number, users should note that the fourth number section is the date of the version's release displayed in YYYY/MM/DD format. The example below would be **Version 1.1.0**, with the release date being **2022/12/15**. These numbers are subject to change as new versions of the iSD are released.*

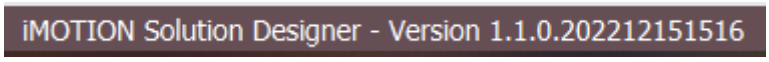

Figure 3 Version Number and Release Date

2.2 Package Manager

The **Package Manager** displays the SD Packages within the IFX server available for installation via the **Available Packages** pane, and the installed SD Package via the **Installed Packages** pane. The Installed Packages will display all currently installed package in release order, with the newest package displayed at the bottom of the list.

SD Packs are an integral part of the iSD operational workflow as they contain the blueprints of the project's structure. The Package Manager should be used to manage SD Packages installed locally, and for the installation of the latest SD Packages.

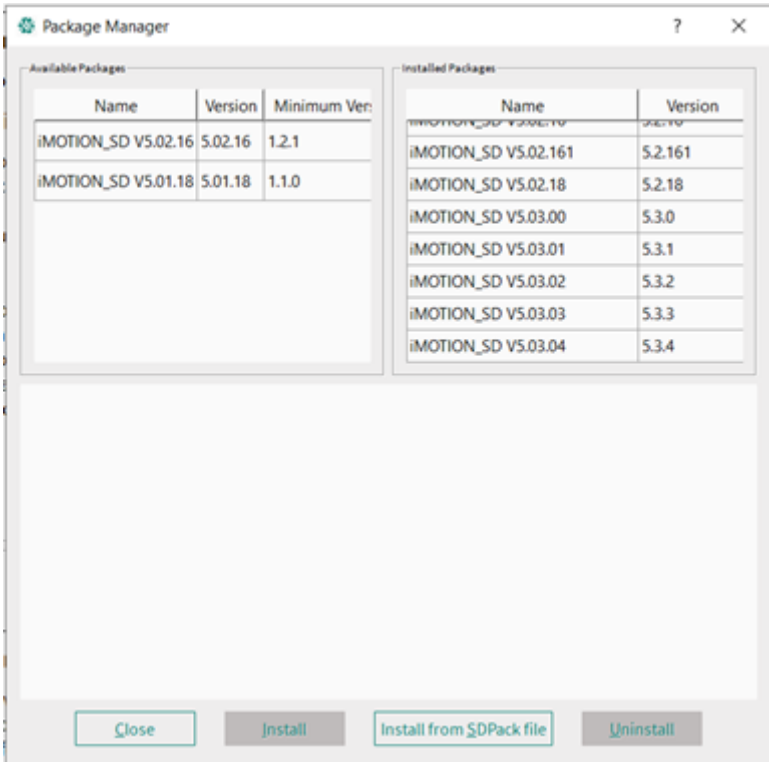

Figure 4 Package Manager

In the **Project Package** pane users can see details regarding the highlighted package. Users can see changes made since the previous version, the packages feature list, and other necessary package information.

Below these panes, users can exit the manager with **Close** and install the highlighted available packages. The user can also choose **Install from SDPack file**, allowing them to install an SD pack from an installation file local to their system.

2.3 Creating a New Project

Using the navigation pane, users can choose **Create New Project** to start their iSD project. Choosing Create New Project will open up the **Create Project** menu. The Create Project menu guides the user through initial project creation, including:

- Selecting one of the installed SD Packages. If the user needs further information on the packages, they can refer to the package information window below the selection.
- Users are encouraged to select the latest SD Pack when possible.
- If needed for compatibility purposes, users are able to access older SD Packs so long as the packs are installed locally or are available for download.
- Choosing a **File name** for the project. This will be how the project is labelled in the file browser. The iSD will use this to name the .cwproj file and project folder.
- Choosing a **Location** to have the project start in. This will create a directory for the project. The iSD will refer to this directory when loading or saving the project.
- Choosing a **Project name** for the project. This will be how the project is labelled in the iSD program. This can be separate from the File name, as it is only used for cosmetic purposes.
- Checking **Create self-contained project** will create a version of the project with the SD Pack included whenever it is saved. This allows users to keep the project as a singular file.

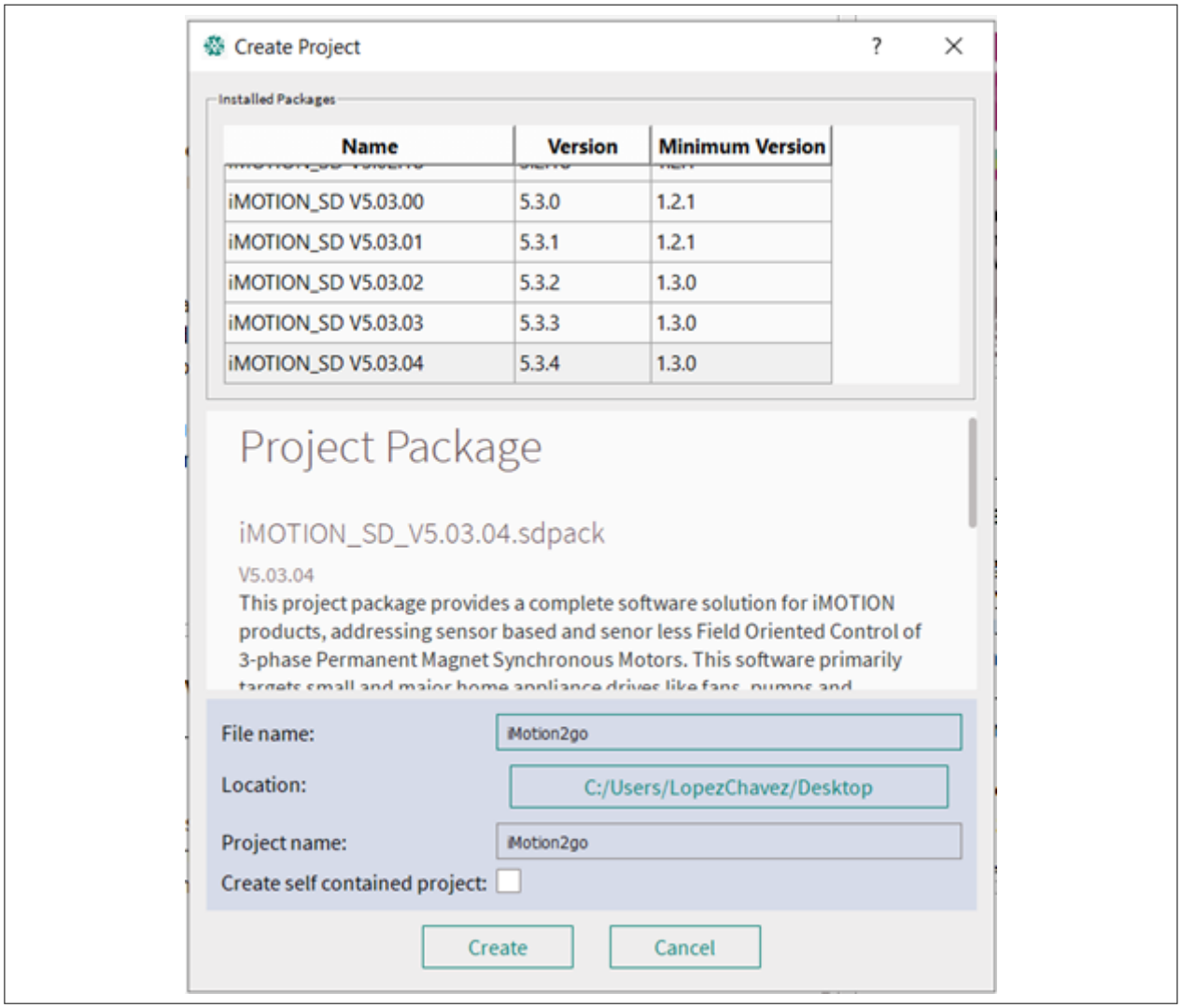

Figure 5 Create Project

2.4 Project Configuration

Once users have created a new project, they will have access to the full suite of project configuration tools the iSD has to offer.

2.4.1 Project Info and Supported Functions

The **Project Info** and **Supported Functions** offer users a quick method of referencing information regarding their project.

Within the Project Info pane, users are able to see the following:

- Project Name – How the project is labelled within the iSD.
- Project Package – Which SD Package the project is based on.
- Evaluation Board – Will display the corresponding Board during the initial project configuration, or will default to 'Custom Board'.
- Requested Device – The device the project is set to affect.
- Latest Build – The last time the project was built for the requested device.
- Parameter Sets – Upon project creation and SD Pack application, will default to 'myFirstParameterSet'.

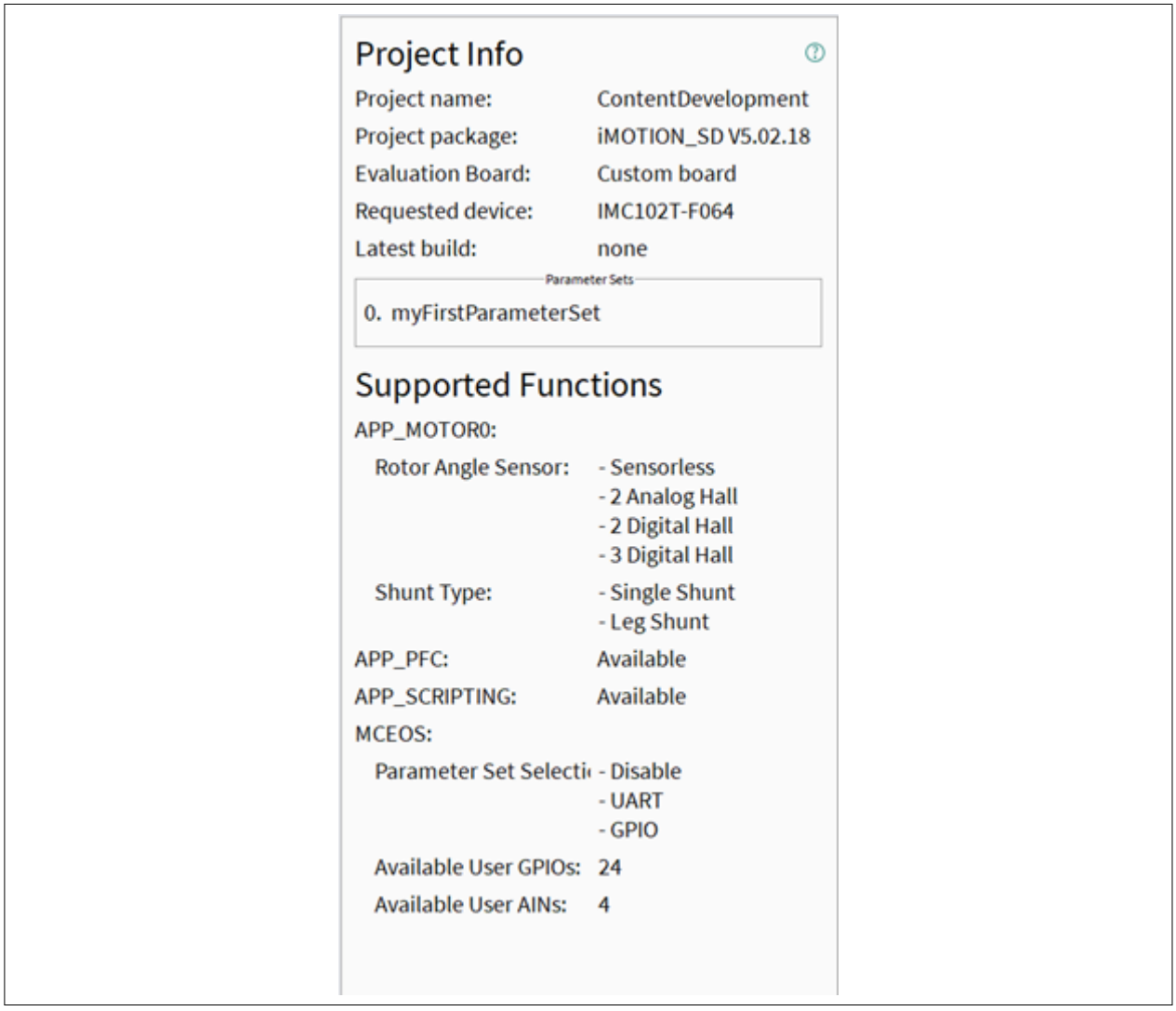

Figure 6 Project Info

2.4.2 Initial Project Configuration

The **Initial Project Configuration** pane offers users three ways to start their project. The method users should make use of depends on the information they have available and the HW board they are using within their project.

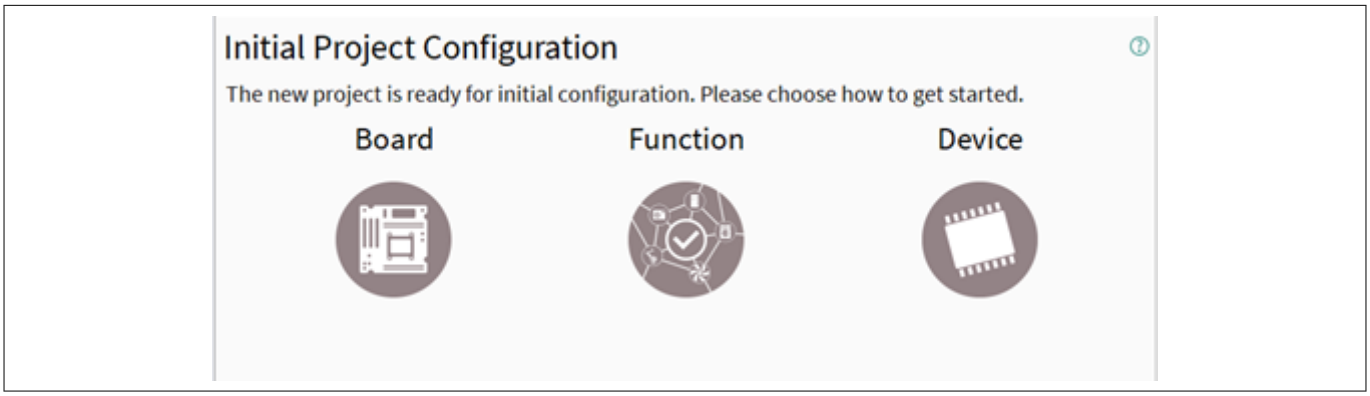

Figure 7 Initial Project Configuration

2.4.2.1 Board

When configuring an iMOTION™ Eval/MADK board or reference board, this option will populate parameters related to board hardware design specifications and components. Users can see what parameters will be imported by checking the box labeled **Show board parameters**. After selecting the board, users will click **Apply** to apply the parameters, then **Close**.

2.4.2.2 Function

If the user lacks specific info regarding the Board or Device of the project, this page highlights various iMOTION™ devices and their uses. Users can select a device based on the needs of their system.

2.4.2.3 Device

If users already know the specific iMOTION™ device for the project, they can use this function to start the project. Users will be able to see all of our available devices and can, as with the previous functions, filter and apply parameters.

2.5 Project Organization

Users can navigate and explore the project's local folders using the file browser of their Operating System (e.g. Windows file Explorer).

- **Settings Folder** – This contains .ozsi Oscilloscope configuration files and .cwproj iSD project configuration files.
- **Lib Folder** - In the instance of a self-contained project, users will find the project's associated SD Pack here.
- **Generated Folder** – Users can find the .ldf files that are downloaded to the target device during device programming. Two .ldf files are generated when building a project – one that contains the firmware and one that contains the project parameters and script ByteCode if the script is used.
- **Script Folder** – Here users can find the projects associated scripts. Users will note that each script project has its own folder, with each task within the script stored as a .mcs file.

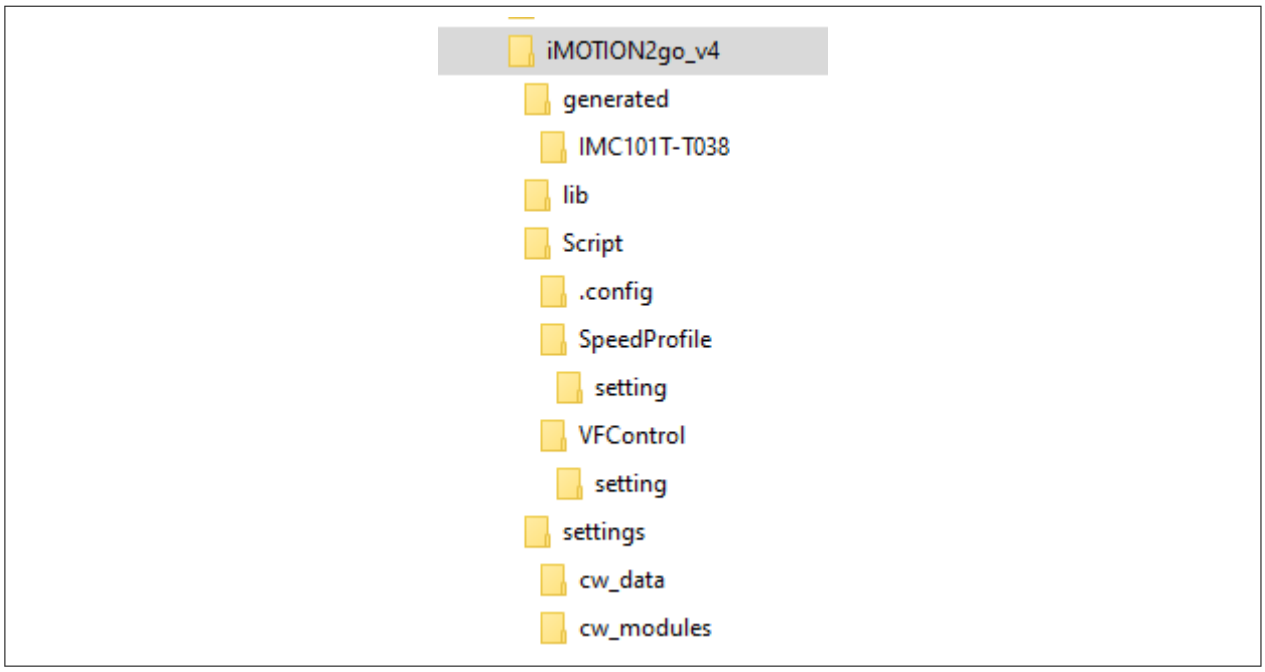

Figure 8 Project Organization and Folder Structure

2.5.1 Sharing a Project

Depending on the method used to create the project, users have two ways of sharing a project:

- **Standard Project** – Users should ensure the recipients software, both the iSD and SDPack, is up-to-date before proceeding. This can be done my navigating to the iMOTION™ home page and ensuring the recipient's version matches the current released version. Once done, users can zip the project files and forward the zip file to the recipient in question. The recipient can then unzip the project file and access the project via their instance of the iSD.
- **Self-Contained Project** – Users do not need to ensure the recipients software is up-to-date using this method. Users can forward the project files, and the recipients can access the project via their instance of the iSD.

3 Menus

The drop-down menus bar at the top of the screen can be used to navigate the iSD's various sub-menus and tools.

3.1 File

Users can use this drop-down menu to access the basic functions of project and file management. This includes the **Package Manager**, **Create Project**, **Open Project**, **Save Project**, and **Save Project as…** functions. While in the Explorer view, only Save, Save Project…, and Exit will be available.

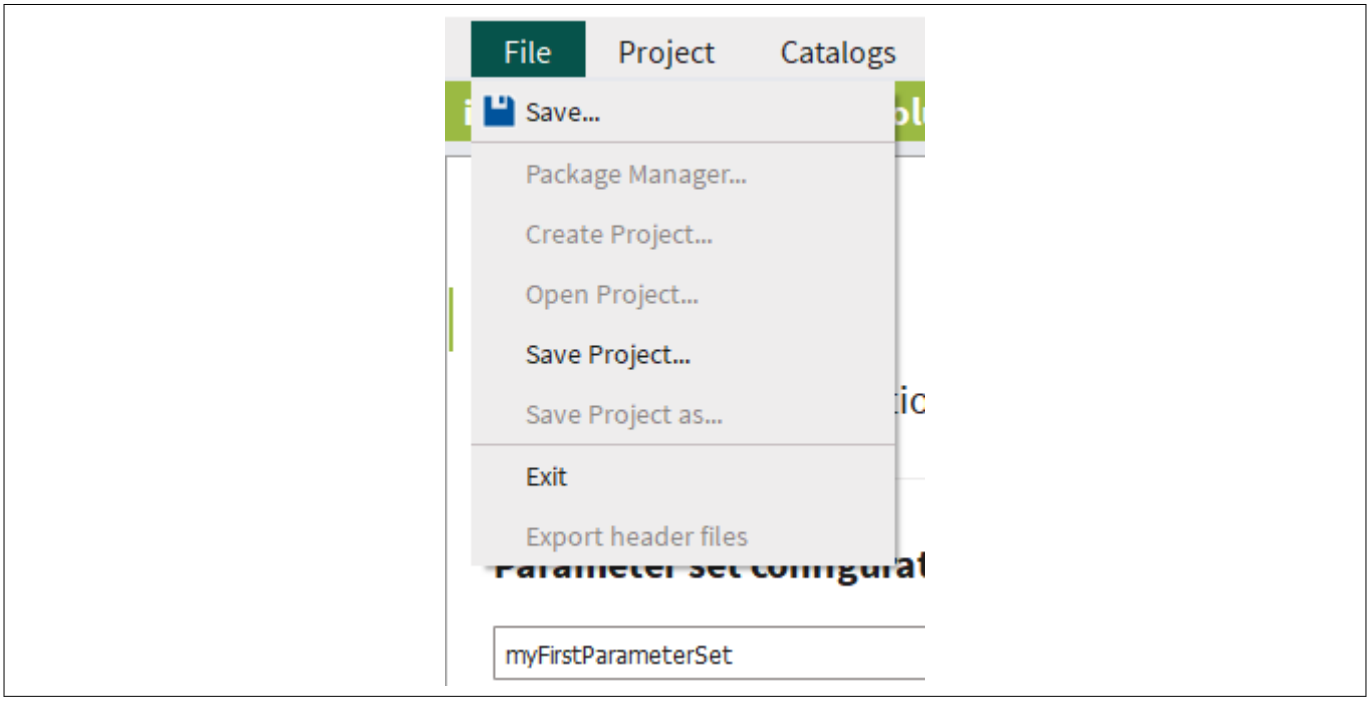

Figure 9 File Drop-Down Menu

3.1.1 Save

Users can use this function to save the changes made by the current tool selected. For example, if users are currently using the Dashboard tool, then the iSD will only save changes made by the Dashboard tool. Users should always save their current changes before switching to a different tool.

3.1.2 Package Manager

This allows users to change their SDpack versions and installations, as well as allowing users to install and uninstall SDpacks.

***Note**: Package Manager is only selectable from the HOME screen, when a project has yet to be opened.*

3.1.3 Create, Open, and Save Project

These options allow users to create a new project, open a project, and save a project. They share the same functionality as the **Create New Project**, **Open Project**, and **Save Project** options on the HOME screen, respectively.

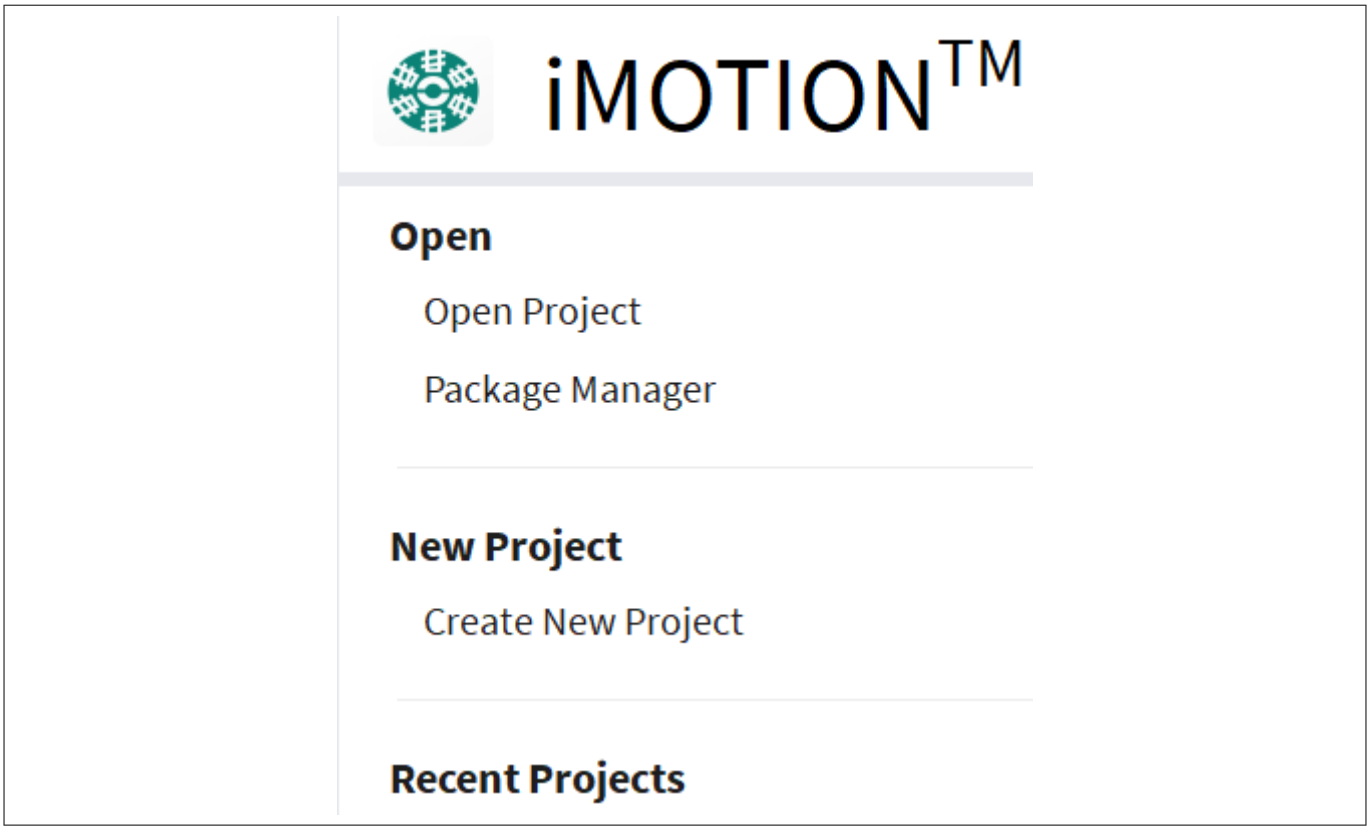

Figure 10 Start a Project

***Note**: Create New Project and Open Project options are only selectable when a project has yet to be opened. The Save Project option is only selectable when a project is open.*

3.2 Project Settings

3.2.1 Project Info

The **Project Info** section allows users to quickly see a brief, but detailed list of the project's information. This includes:

- **Project Name** – The name given to the project during its initial creation.
- **Project Package** – The SDPack used to create the project, as well as its version.
- **Evaluation Board** – Board used to create the initial parameter set.
  - If a device is selected instead of a catalog, then **Custom board** will be displayed.
  - It will **NOT** get updated if a new catalog is selected inside of the Configuration Wizard.
- **Parameter Sets** – This will display a list of parameter sets used within the project.

3.2.2 Supported Functions

The **Supported Function** section contains a list of functions distinct to the projects iMOTION™ device. This includes:

- **APP_MOTOR0** – These are functions related to the motor control. It is broken up into two sections; the **Rotor Angle** and the **Shunt Type**. The Rotor Angle lists the different Motor Sensors that the device supports and the Shunt Type lists the different shunts the device supports.
  - **Rotor Angle** – Options include the 2 Analog, 2 Digital, 3 Digital, and Sensorless angle sensor modes.
  - **Shunt Type** – Options include the Single Shunt and Leg Shunt type.
- **APP_PFC** – This section will tell the user if the PFC is available in the projects iMOTION™ device. It will list the PFC as **Available** or **Not Available**.
- **APP_SCRIPTING** – This section will tell the user if scripting is available for the selected iMOTION™ device. It will list scripting functionality as **Available** or **Not Available**.
- **MCEOS** – This shows users functions related to the iMOTION™ operating system (OS). It is broken up into three sections; the **Parameter Set Selection**, the **Available User GPIOs**, and the **Available User AINs**.
  - **Parameter Set Selection** – This lists the methods available to the project's device in regards to changing parameter sets. The options that can appear are:
    - Disable – Will not change the parameter sets.
    - UART – Will change the parameter set using UART.
    - GPIO – Will change the parameter set using GPIO.
    - PARAM – Change parameter set using PARAM.
  - Available User GPIOs – This lists the number of GPIOs available for scripting.
  - Available User AINs – This lists the number of Analog inputs available for scripting.

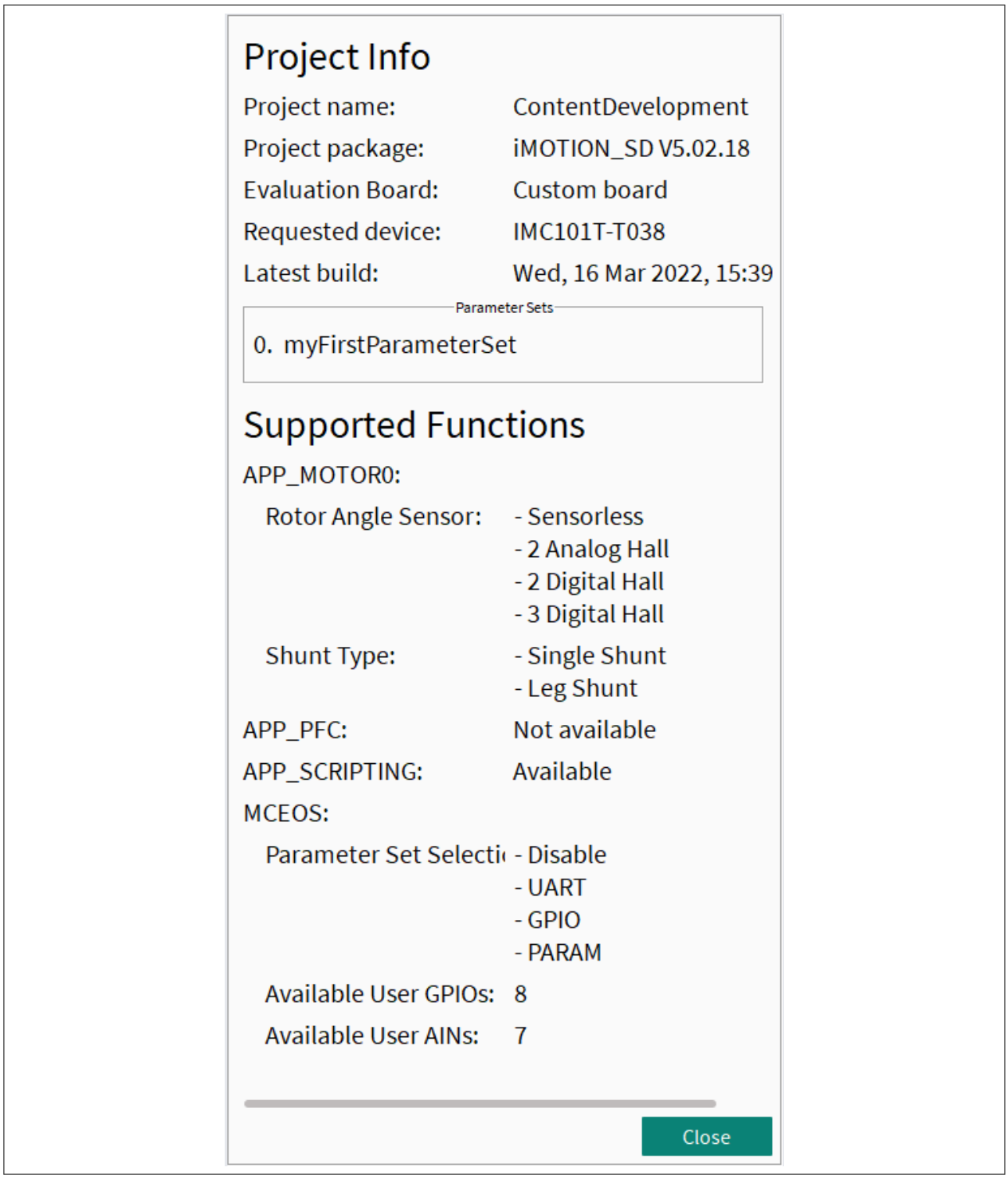

Figure 11 Project Info

3.3 Catalogs

Catalogs allow user to access the iSD database of records and settings for **Boards**, **Motors**, and **Loads**. Users can quickly apply this information to their current Configuration Wizard parameter set via the pop-up menu. Board catalog data only contains system hardware configuration related parameters and does not contain the motor or PFC control and protection configuration related parameters.

***Note**: Catalogs are only selectable when using the Configuration Wizard tool. For more details on Catalog Files, see Chapter 3.4.*

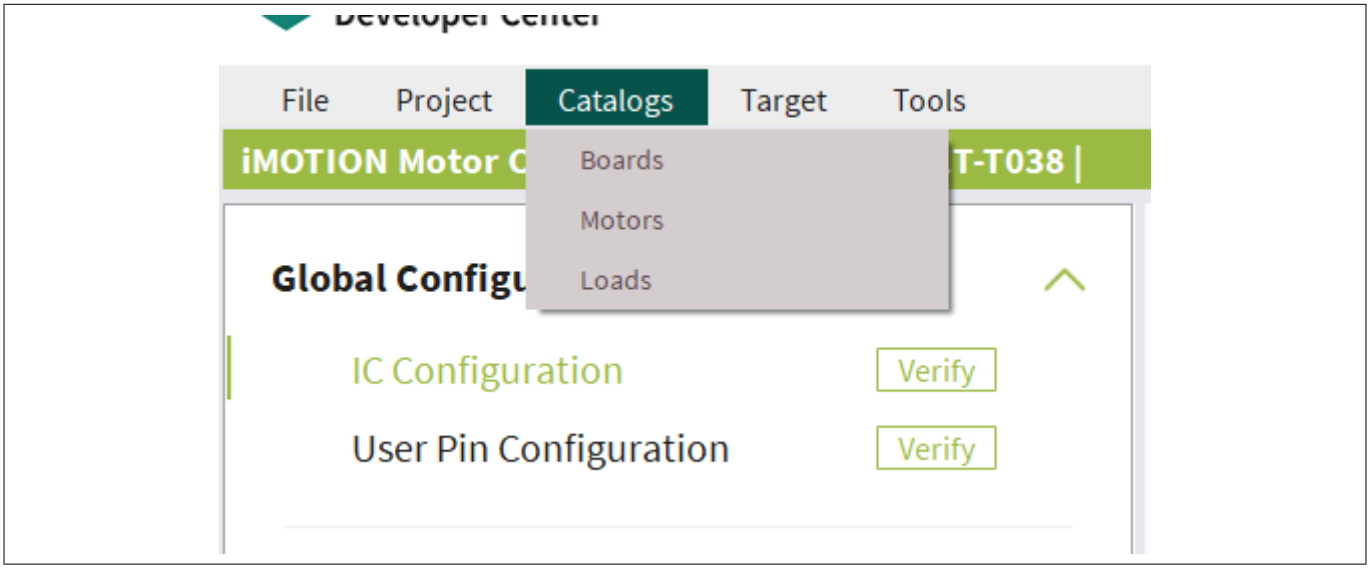

Figure 12 Catalogs Drop-Down Menu

3.4 Target

The Target pop-up window will allow users to quickly connect to the iSD server, associated COM Ports, and the device they are attempting to edit. The Target pop-up window appears automatically if the target is not connected when the Programmer, Dashboard, or Oscilloscope is selected.

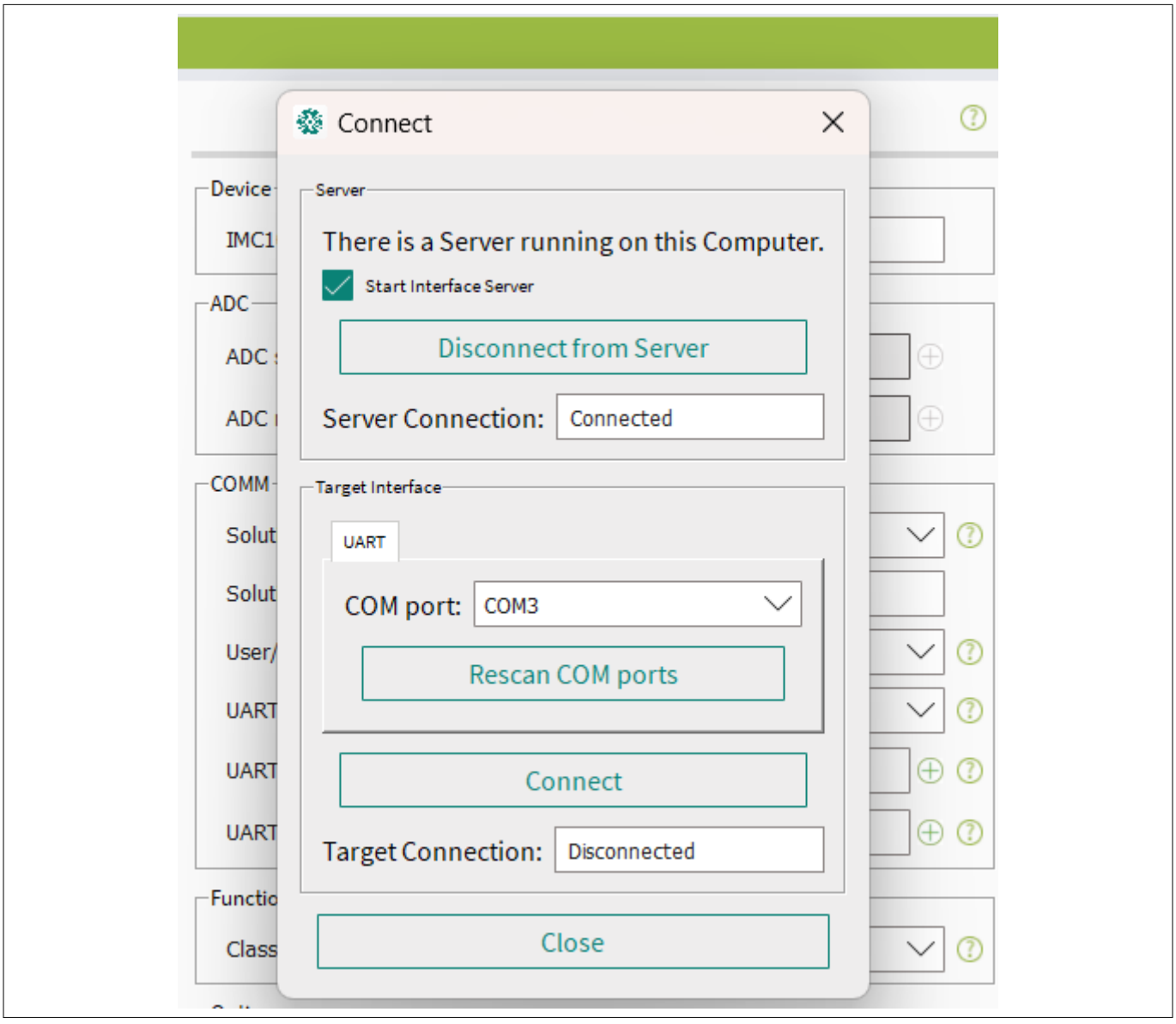

Figure 13 Target Pop-Up Window

3.5 Tools

The Tools drop-down menu allows users to access the Configuration Wizard, Programmer, Dashboard, Script Editor, Oscilloscope, Help Viewer, and iMOTION™ Forum.

Figure 14 Tools Drop-Down Menu

Additionally, several of the tools can be accessed via the icons at the top-right of the user's interface.

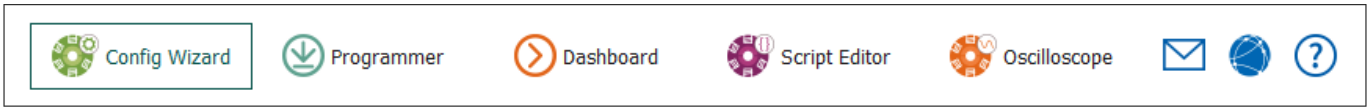

Figure 15 Tools Icons

4 Log View

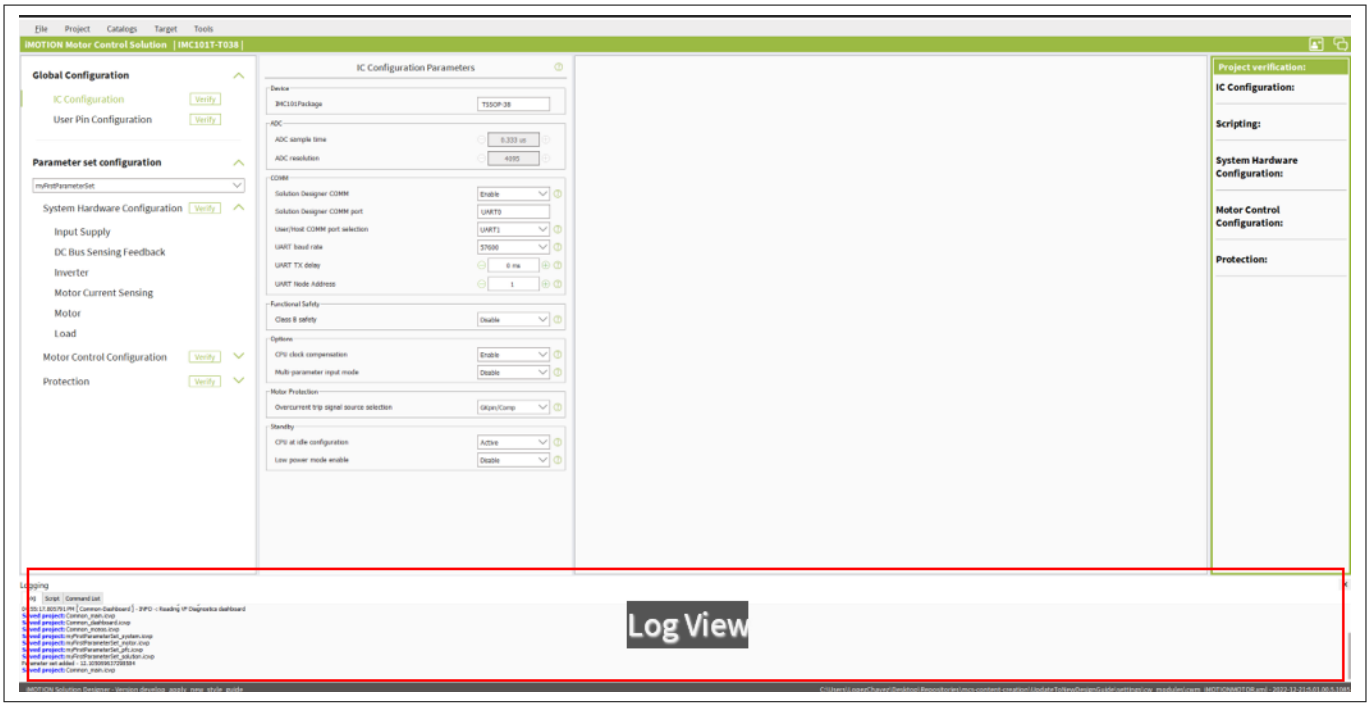

Figure 16 Log View

4 Log View

The Log View in provides diagnostic information in conjunction with several situations and tools:

- When validating parameters and observing if there is a count-out-of-range error.
- When experiencing connection issues.
- When encountering building or programming issues.
- Observing when a file has been saved.

5 Configuration Wizard

5.1 Introduction

The **Configuration Wizard** allows users to set the parameters for their project and device. These tools allow for both manual parameter adjustments and a more automated options, where users can use the program's internal catalogs to apply the projects essential parameters. The Solution Designer also provides users with a validation tool, which will review the parameters to ensure they are appropriate for the chosen device.

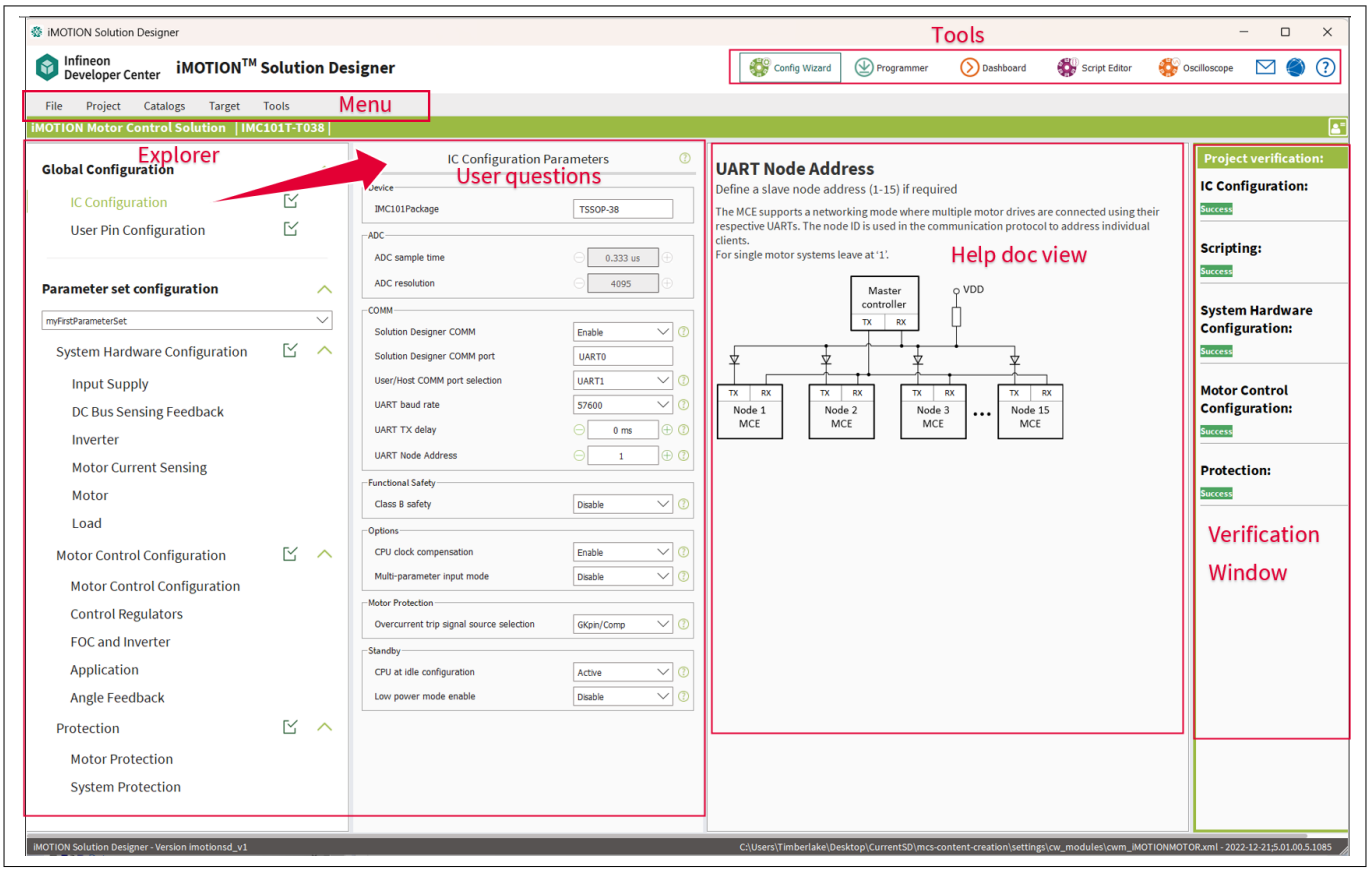

Figure 17 Configuration Wizard Main Page

There are five top-level parameter groups that users are able to navigate through, split across two parent groups. The **Global Configuration** contains the IC Configuration and User Pin Configuration parameter groups. The **Parameter Set Configuration** contains the PFC Control Configuration (if a device that supports it is selected), System Hardware Configuration, Motor Control Configuration, and Protection parameter groups.

When users begin adjusting the parameter sets at the start of a project, it is suggested they start with the IC Configuration parameters and work their way down to the Protection parameters. The parameters available for users to edit may change depending on the device and certain content will be unavailable when a device does not support it as a function.

***Note**: The green question marks within each section can provide users with more in-depth information.*

5.2 Catalog Files

Catalog Files contain the hardware parameters for the evaluation and reference design boards. Users can use these parameters to run the motor after selecting **Boards**, **Motors**, and **Loads**. Users should note that selecting any of the three options will produce a pop-up window, where users can see the database of available

parameters in the **Database** pane. Depending on which of the three catalogs is chosen, the Database pane will display a list of available types.

Users can also apply various filter options to the Database pane via the **Filter** pane. Depending on which of the three catalogs is chosen, the Filter pane will display a list of available filters.

- Boards – Board Name, Board Type, Connector Component Type, Control Board, Application Device.
- Motors – Motor Model Name, Position Sensor, Motor Rated Amps/Phase, Stator Resistance/Phase (Rs), LPM Motor Stator LQ Inductance/Phase, LPM Motor Stator ID Inductance/Phase.
- Loads – Model Name, Load Type, Rated Power, Minimum Speed, Maximum Speed.

Board Catalogs come in two different types: **PFC/Motor** related parameters, which are 02 devices, and **Only Motor/Non-PFC** parameters, which are 01 devices.

Motors and PFC control boards will have catalogs that cover the following:

- Input Supply
- PFC Basic Input and Output
- PFC Power Stage
- PFC Current Sensing
- PFC VAC Sensing
- DC Bus Sensing Feedback
- Inverter
- Motor Current Sensing

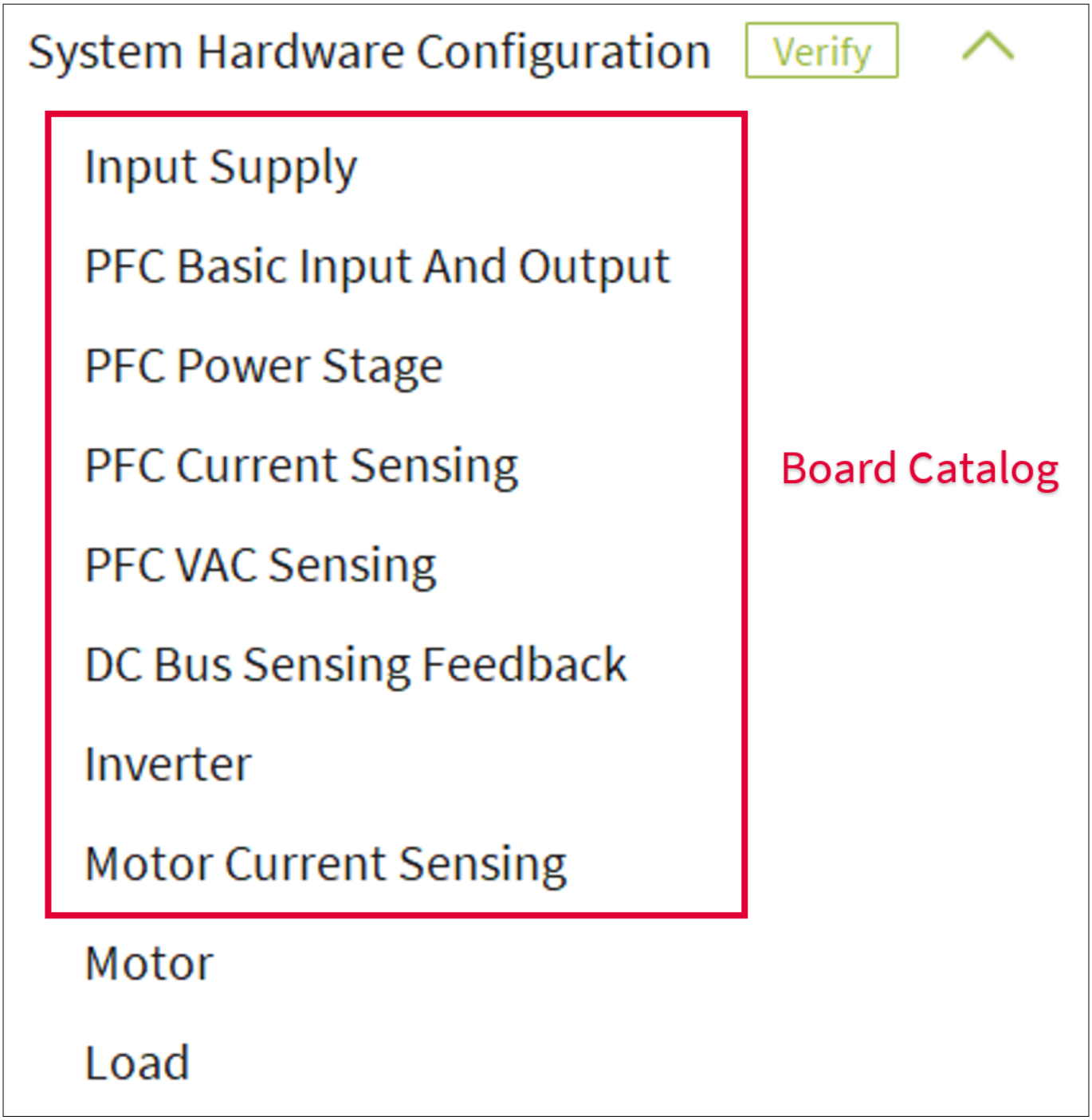

Figure 18 Board Catalog with PFC Parameters

When a Motor control only board is being used, the catalog will cover the following:

- Input Supply
- DC Bus Sensing Feedback
- Inverter
- Motor Current Sensing

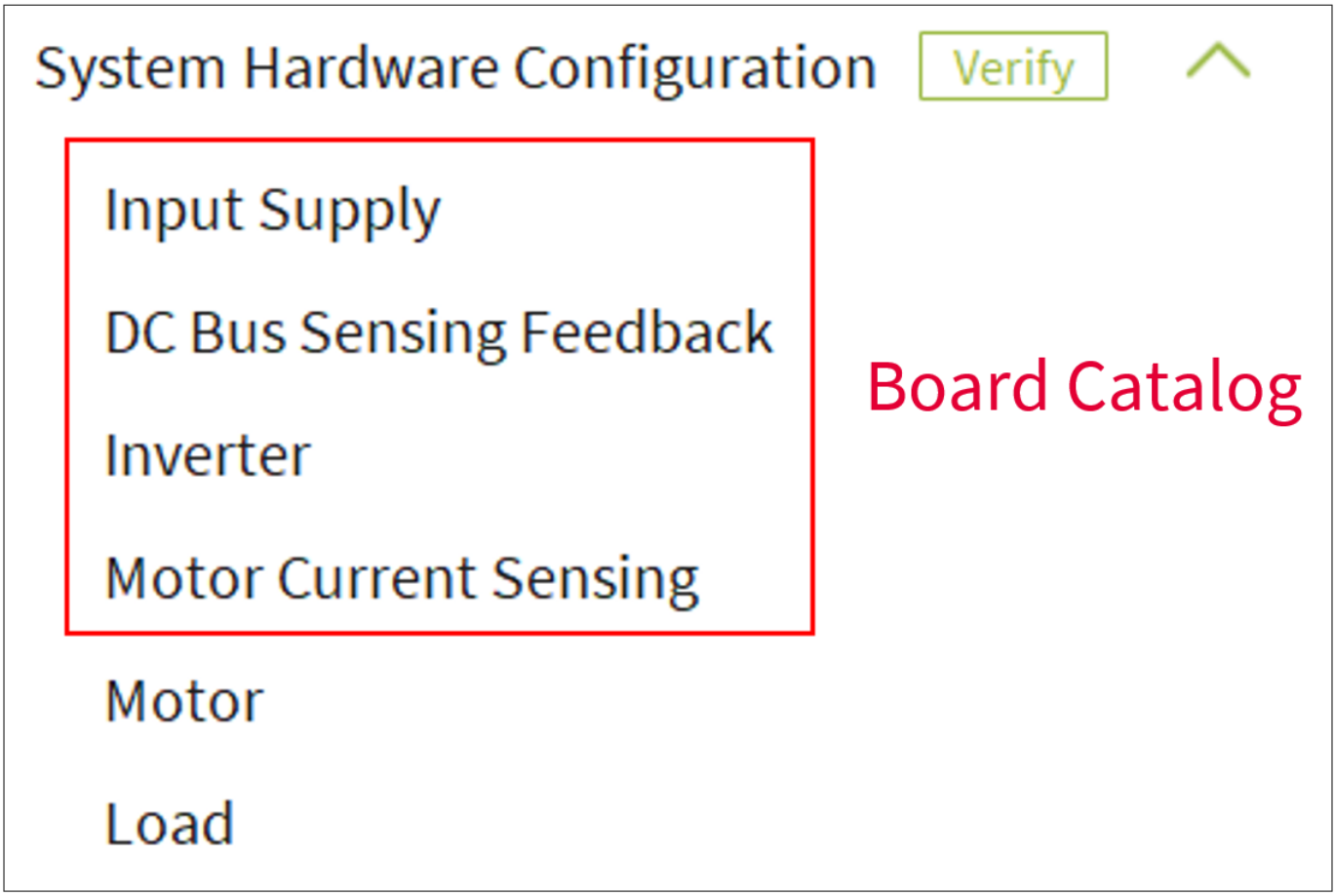

Figure 19 Board Catalog with Motor Control Only Parameters

Motor Catalog files will configure the parameters of the following sections:

- Motor

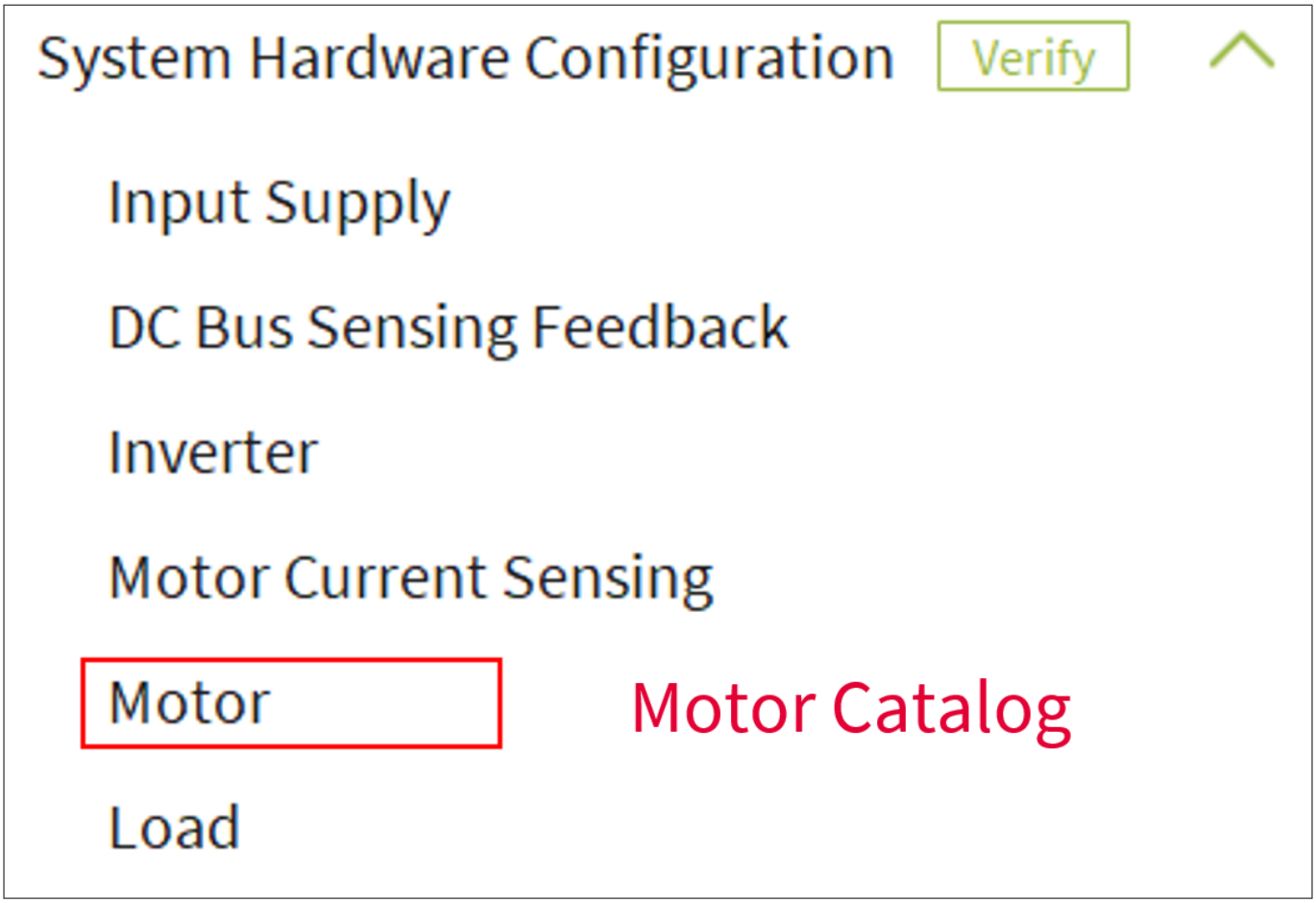

Figure 20 Motor Catalog

Load Catalog files will configure the parameters of the following sections:

- Load

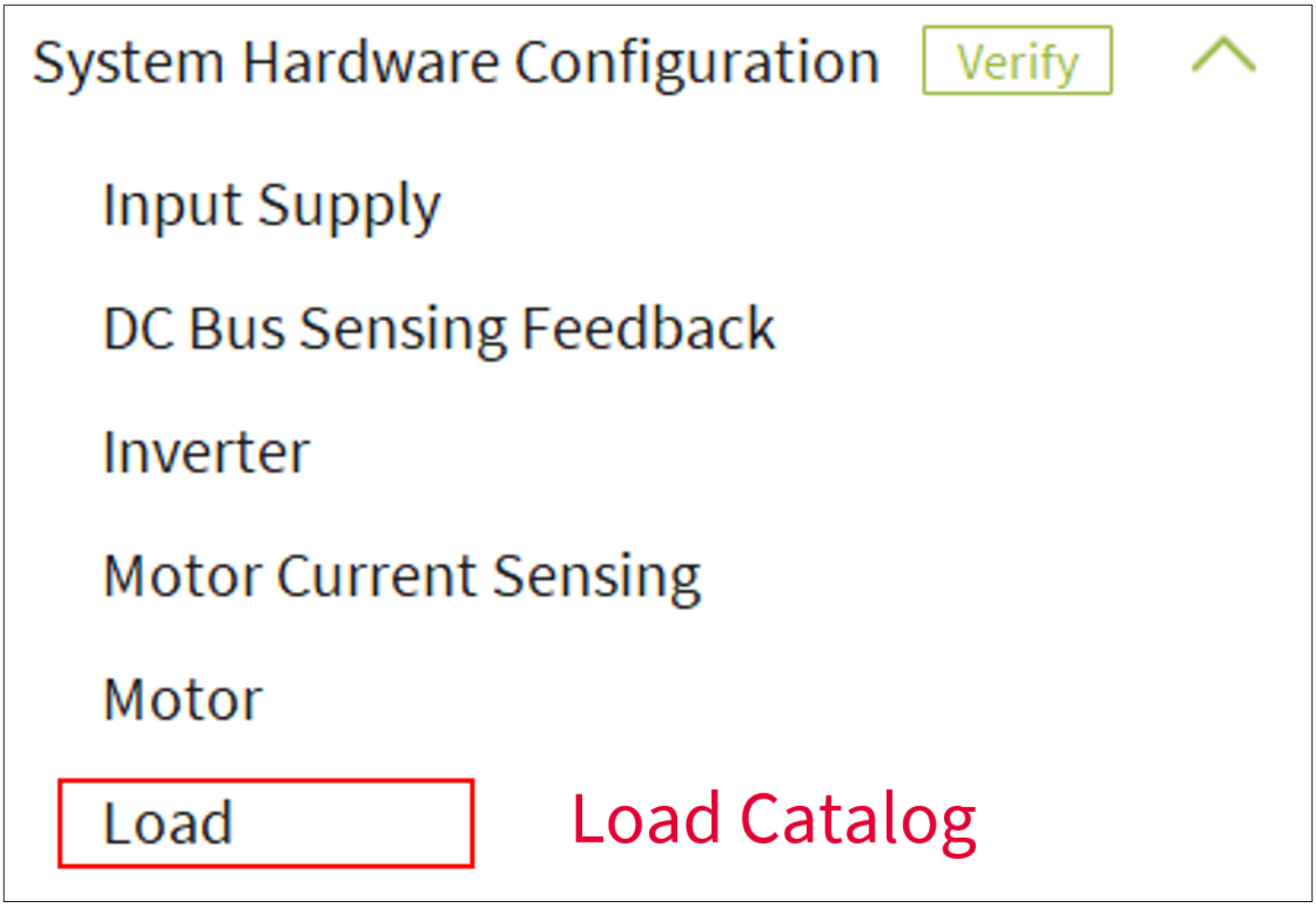

Figure 21 Load Catalog

5.3 Normal Mode vs. Expert Mode

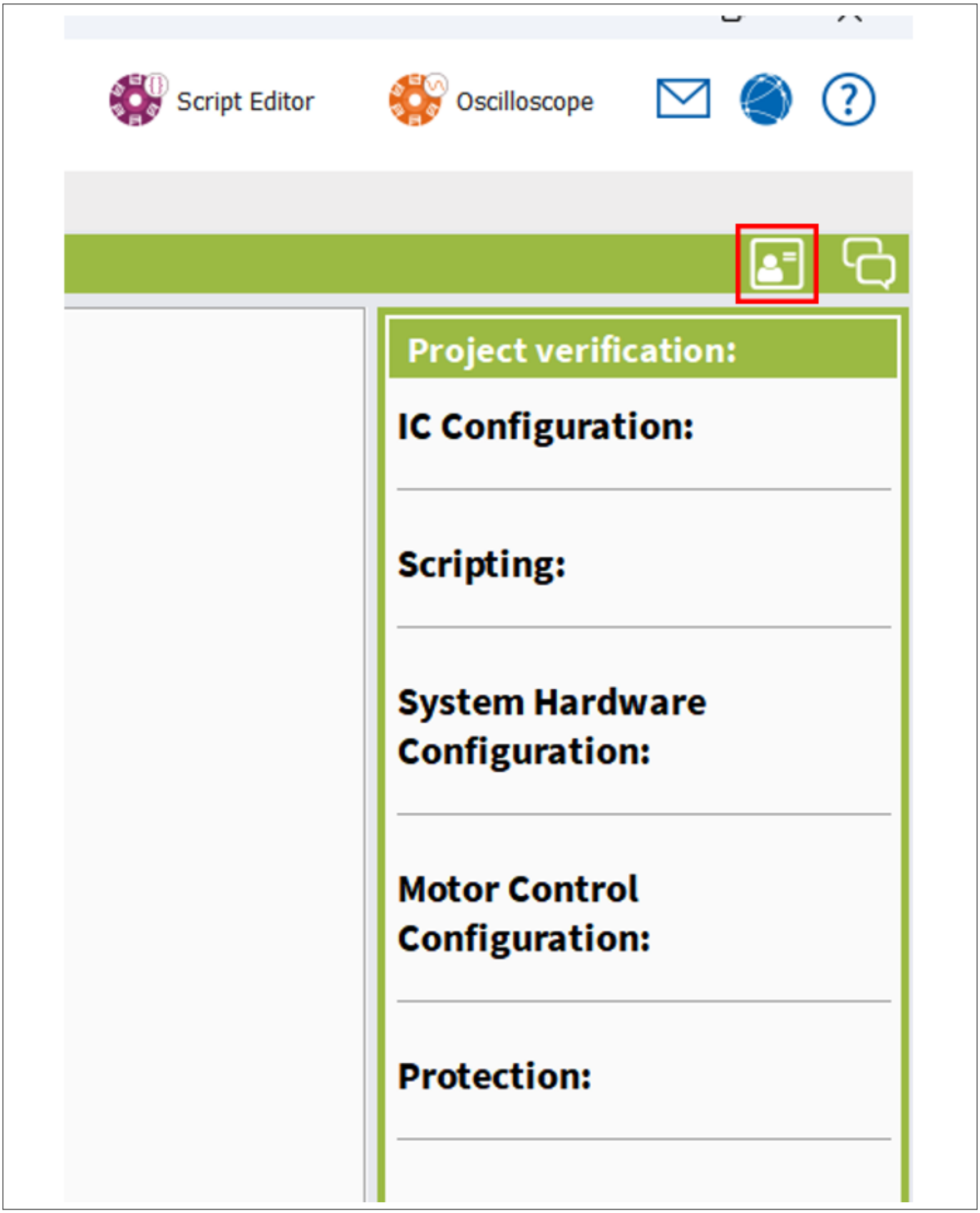

Figure 22 Normal Mode/Expert Mode Button above Project Verification

- **Normal Mode** – This will show users the most common and required parameters.
- **Expert Mode** – This will open up the full top-level parameters for users, allowing for more detailed editing.

Users can toggle between the modes via the  button at the top-right of the screen. 

signifies that Expert Mode is disabled, whereas  signifies that Expert Mode is enabled.

When to use each mode will depend on the needs of the user. Users should use Expert Mode when they want to utilize the advanced functions available in the iSD, such as greater customization of the Motor or PFC behavior. Normal Mode should be used when users want a quick start on getting the motor running.

5.4 IC Configuration

This parameter group gives users access to general **IC Device** configuration, **ADC** settings, **COMM** settings, **IC Options**, and **Motor Protection** settings.

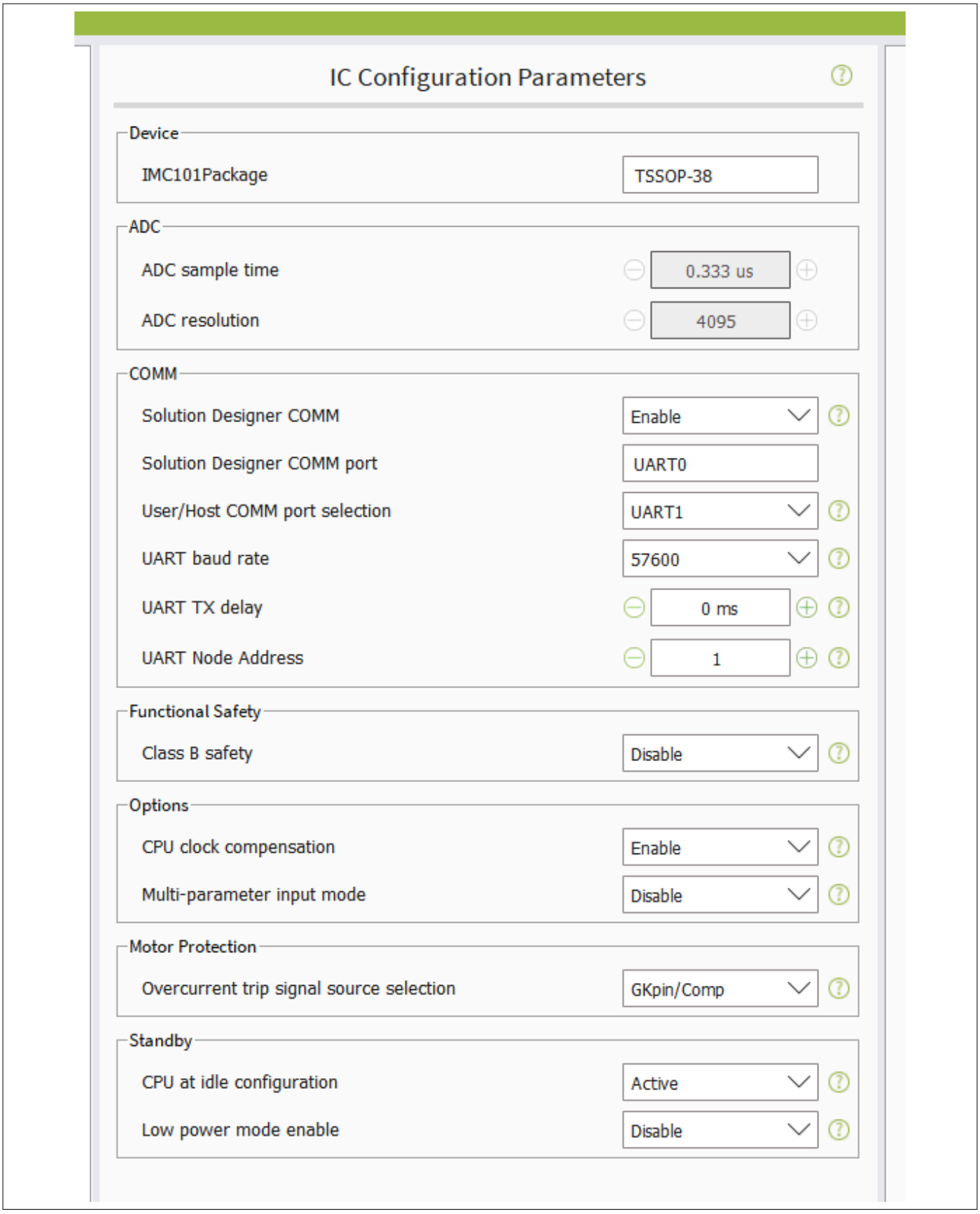

Figure 23 Control IC Parameters

5.5 User Pin Configuration

This parameter group gives users the ability to select the pins used for scripting.

- If a pin (GPIO/AIN) is not available for this device, then it will not be shown in the User Pin Configuration section.
- If a pin is multiplexed with another option (e.g., GPIO1 is multiplexed with PG_OUT) and its shared option is used, then it will not be shown in the User Pin Configuration section.

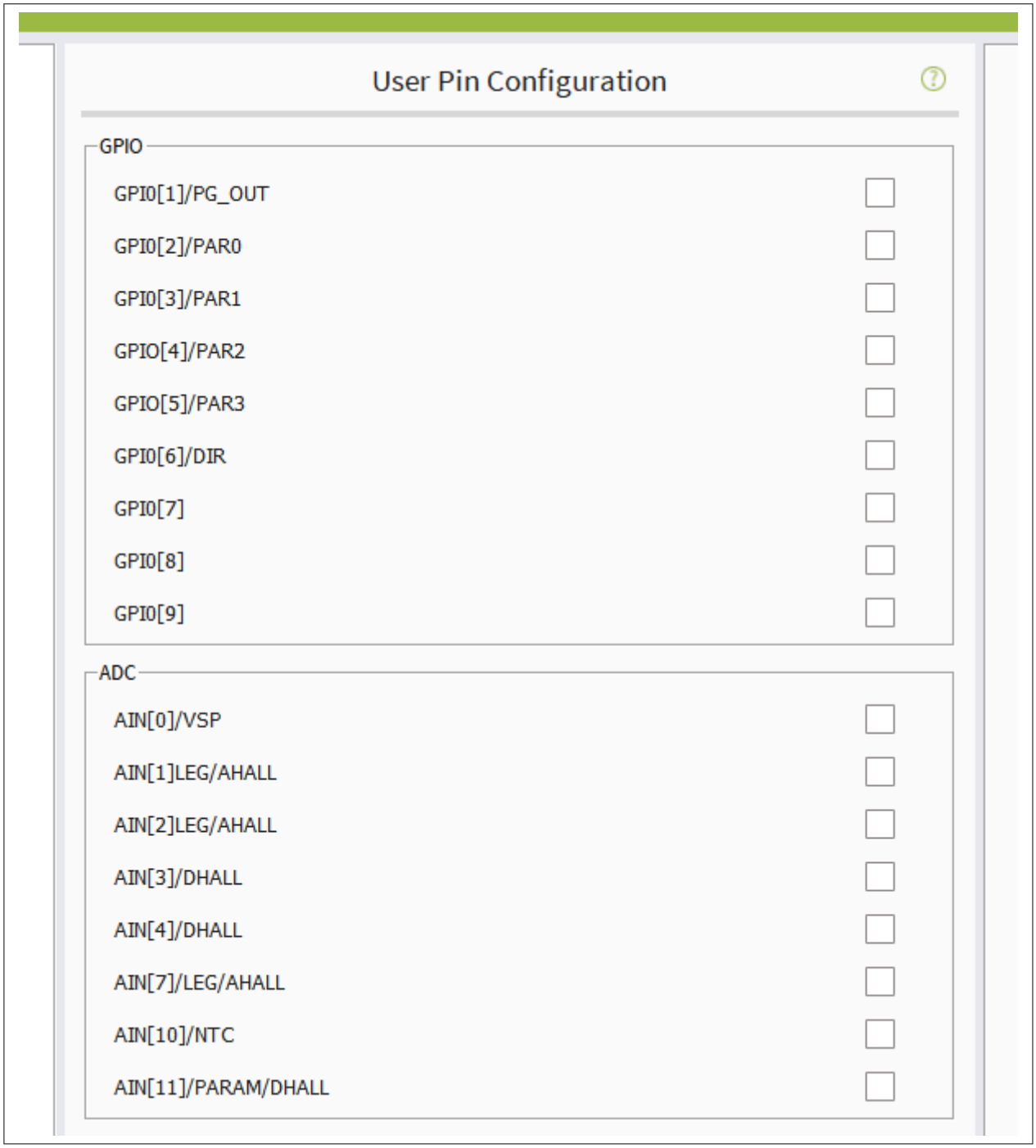

Figure 24 User Pin Configuration

***Note**: If users select a pin (e.g., GPIO1), then later select a shared option (e.g., PG_OUT), then a pop-up prompt will be shown asking the user if they want to disable the related GPIO/ADC pin.*

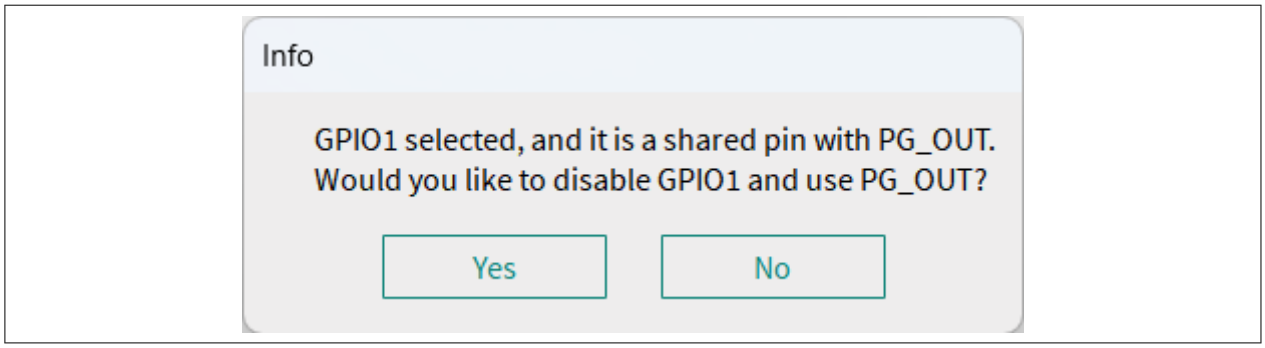

Figure 25 Pin Selection Confirmation

5.6 Parameter Set Configuration

On top of housing the parameter groups, the Parameter Set Configuration section allows users to select the parameter set via the drop-down menu situated at the top.

5.6.1 System Hardware Configuration

The sections in this parameter group relate to the hardware being used by the project.

- **Input Supply** – Allows users to adjust the Control supply voltage. ADC gain is shown for informative purposes only.
- **DC Bus Sensing Feedback** – Allows users to adjust the upper and lower resisters for Vout sensing within the DC Bus.
- **Inverter** – Allows users to adjust the Catalog name, Power Switch settings, and Gate Driver.
- **Motor Current Sensing** – Allows users to adjust the Current sensor shunt and Current Feedback settings.
- **Motor** – Allows users to adjust the motor settings and associated electrical settings.
- **Load** – Allows users to adjust the power settings.

If a device that supports PFC control is selected, then additional sections to configure PFC hardware related parameters will become available.

- **PFC Basic Input and Output** – Allows users to configure parameters related to the AC input voltage, line frequency, power rating, and desired DC bus voltage.
- **PFC Power Stage** – Allows users to configure the parameters related to power stage components including the DC bus capacitor, inductor, and gate driver propagation delay.
- **PFC Current Sensing** – Allows users to configure the gain and offset of the inductor current sensing circuitry as well as the ADC sample offset time for the current sensing operation.
- **PFC VAC Sensing** – Allows users to configure the upper and lower resistors and the filter capacitor values for VAC sensing circuitry.

5.6.2 Motor Control Configuration

The sections in this parameter group allows users to configure the system's motor control algorithm.

- **Motor Control Configuration** – Allows users to adjust the Control Rates and Control Modes.
- **Control Regulators** – Allows users to adjust Velocity Regulator and Current Regulator settings.
- **FOC and Inverter** – Allows users to adjust the Voltage Generator, SVPWM, and Signal Sampling settings.
- **Application** – Allows users to enable or disable certain motor functions and allows users to adjust the Motor Starting, Current Limits, and Command Input settings.
- **Angle Feedback** – Allows users to adjust the Angle Estimator settings.

5.6.3 PFC Control Configuration

The section in this parameter group allows users to configure PFC control loop related parameters. This section is only visible If a device that supports PFC control is selected.

- **PFC Control Configuration** – Allows users to configure PWM switching frequency, minimum Ton / Toff time, current and voltage loop update rates, and inductor current sensing offset calculation time.
- **PFC Control Regulators** – Allows users to configure the current loop and voltage loop compensator related parameters and the soft-start time.
- **Zero-Crossing Detection** – Allows users to configure zero-crossing detection thresholds and deglitch time.
- **VAC Feed Forwarding** – Allows users to configure the AC input voltage feedforwarding (VFF) control valid range and the permitted AC input current 3rd harmonics from VFF.

5.6.4 Protection

The sections in this parameter group relate to the faults and fault levels of the device.

- **PFC Protection** – Allows users to configure the Over Current, Line Frequency Fault, inductor current offset tolerance, VAC over-voltage and brown-out, and DC bus open-loop and over-voltage related parameters.

  ***Note**: This section is only visible if a device that supports PFC control is selected.*

- **Motor Protection** – Allows users to adjust the Over Current, Voltage, Rotor Lock, Phase Loss, and Current Offset Protection, as well as adjust the Flux Fault and NTC settings.
- **System Protection** – Allows users to adjust certain System Protection settings.

5.7 HelpDoc

Users have a variety of help options at their disposal when using the iSD. The first is the **Green '?'** at the top-right corner of several panes. This will display HelpDoc information for the pane within the center of the screen.

Users also will find links to the **iMOTION™ MCE Help** and the **iMOTION™ Forum** at the upper-right of the program UI. These make up the HelpDoc, a suite of options that allow users to quickly find answers to a variety of questions.

5.7.1 iMOTION™ MCE Help

The iMOTION™ MCE Help is the program's local database of solutions and guides. Users can readily browse this feature by selecting the question mark icon at the upper-right of the program UI. Doing so will open the Help pane.

The iMOTION™ Help pane is divided into three columns; **Contents**, **Index**, and **Search**.

5.7.1.1 Contents

- **Parameter Reference Manual** – This manual documents all of the low-level parameters and registers of the device.
- **iMOTION™ Solution Help** – Users can use this tab to get specific help in regards to their project and project parameters.

5.7.1.2 Index

The Index lists the information from the Contents tab divorced from their parent category. Items are indexed in alphabetical order for ease of use.

5.7.1.3 Search

The Search allows users to search through information in the Contents tab using keywords. The Search tab will display content if the keyword is found in the content's title or body text.

5.7.2 Motor Control ICs Forum

Users can visit the Motor Control ICs Forum by clicking the Web icon at the upper-right of the program UI. Here users can engage with the Infineon community of application engineers. Users can find questions posted by other users and their associated answers.

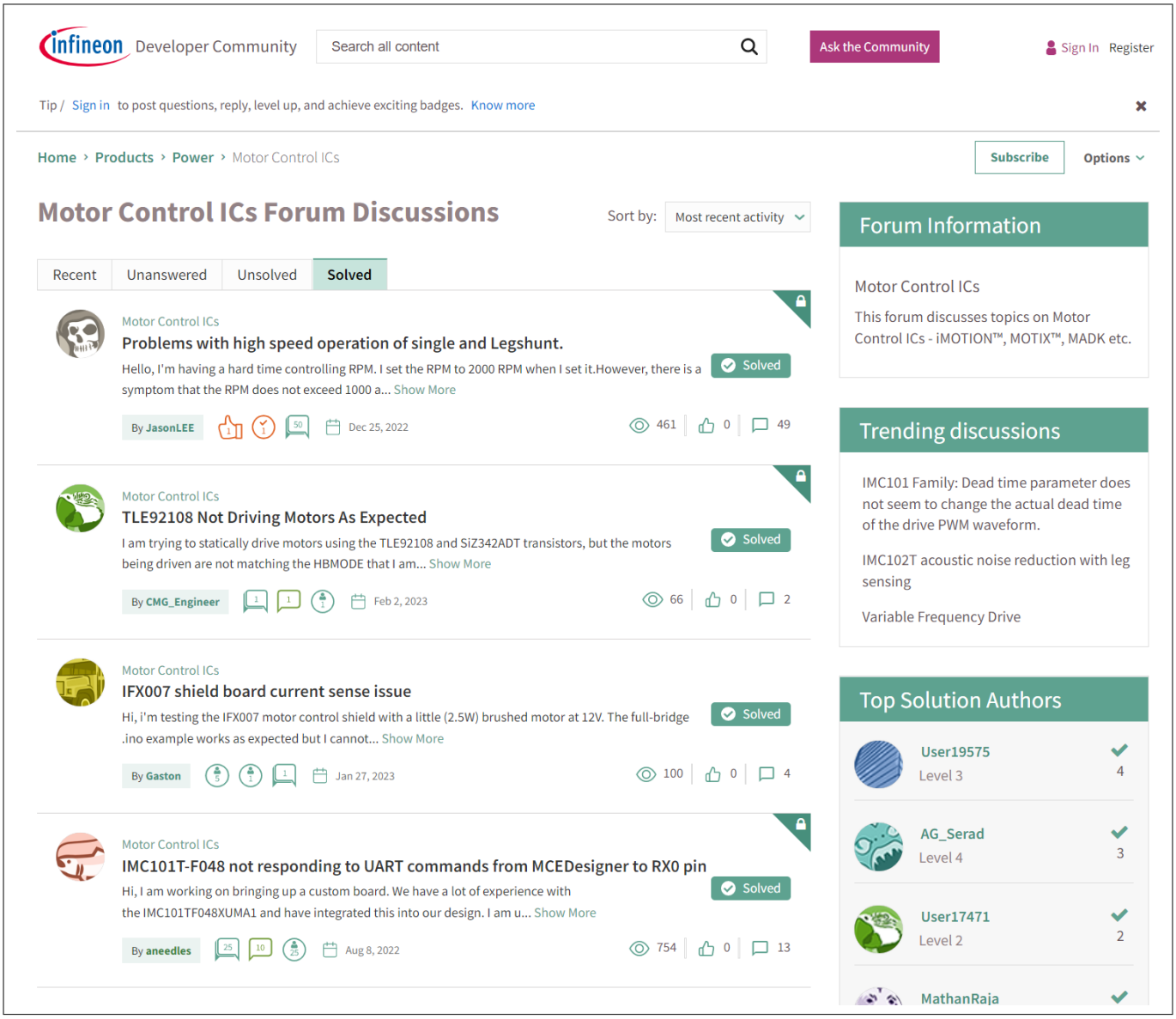

Figure 26 Motor Control ICs Forum Frontpage

The Motor Control ICs Forum is organized with the user in mind, prioritizing usability and readability. The **Discussions** section of the page is separated into four tabs:

- **Recent** – Recent discussions in the Motor Control ICs forum.
- **Unanswered** – Discussions that do not yet have comments. This tab is a great place for users to start interacting with the community.

- **Unsolved** – Discussions that have not been marked as ‘Solved’. Discussions are usually marked as ‘Solved’ by either the discussion creator or a moderator when a comment has sufficiently answered the creator’s question.
- **Solved** – Discussions that have been marked as ‘Solved’ by either the discussion creator or a moderator. If a user is looking for an answer to a question, they should always check here first.

Other sections include **Forum Information**, which summarizes the usage and purpose of the Motor Control ICs Forum, **Trending Discussion**, which show the most active discussions within the forum, and **Top Solution Authors**, which shows the users who have answered the most questions.

5.8 Project Verification

5.8.1 Initiating Verification

On each of the parameter groups, users will find the **Verify** function. Verify is a powerful tool that helps users find errors and undocumented settings in their project. The Verify buttons will be visible by default or when a change has been made to one of the parameters, allowing users to reverify the project.

Users may encounter an **Error** that will appear at the start of the verification process. This means that the count value of a setting has exceeded the maximum range value or is below the minimum value.

5.8.2 Project Verification Information and Outcomes

5.8.2.1 Project Verification Window

- **Information** - The **Project Verification Window** will let users quickly see the results of the verification process. If there are no errors or warnings, then a green **Success** will be displayed beneath the appropriate section.
- **Hide/Show Verification Window** – To hide or show the Project Verification window, users can click on the  button at the top-right of the window. The  icon means that the verification window is visible, and the  icon means that the verification window is hidden.

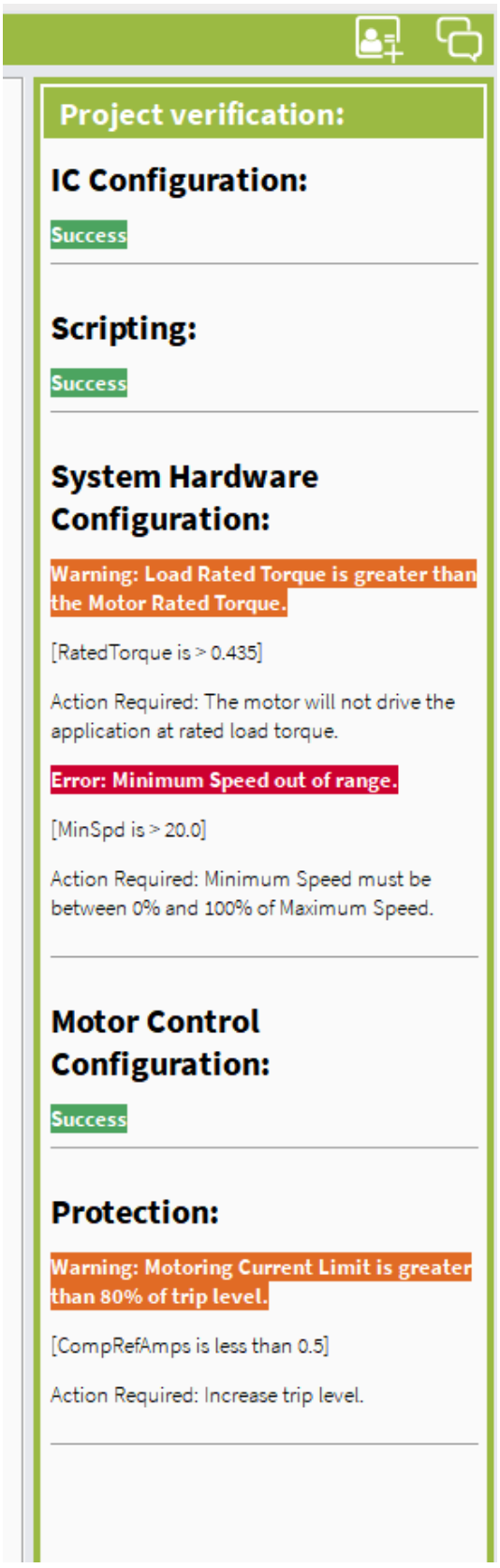

Figure 27 Project Verification Window

5.8.2.2 Successful Verification

If a parameter group passes a verification check, green

checkmark box will appear over the Verify function. The **Project Verification** pane, which appears at the far right of the explorer during this process, will display a green Success if the parameter group has passed.

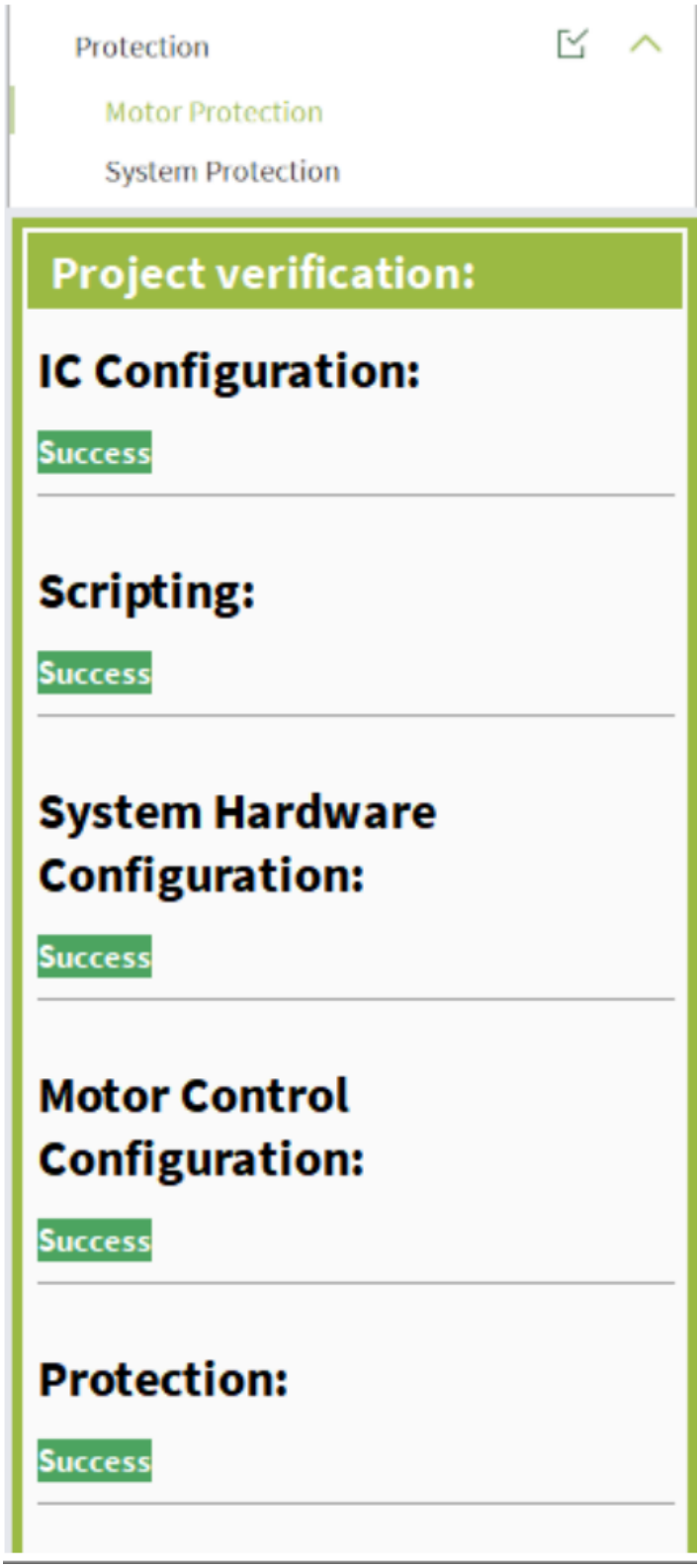

Figure 28 Successful Project Verification

5.8.2.3 Warnings and Errors

When a parameter fails a verification check, the tool will alert the user via **Warnings** and **Errors**:

- Warnings will be highlighted in yellow and display a yellow

  

  exclamation point in place of a green checkmark. A yellow warning will still allow users to build their parameters and program their device, but will ask users to acknowledge the warning and the risks involved.
- Errors will be highlighted in orange and an orange

  

  exclamation point will appear over in place of a green checkmark. While an orange warning will usually not allow users to build their parameters and program their device, users can uncheck the **'Verify before build**' to build without verification. This can be useful for diagnosing various concerns. Users should use this option with caution, as the errors involved could have significant detrimental effects on the project.

Users are encouraged to adjust the settings that caused the warning or error. This will reset the Verify function, allowing users to attempt the verification process again.

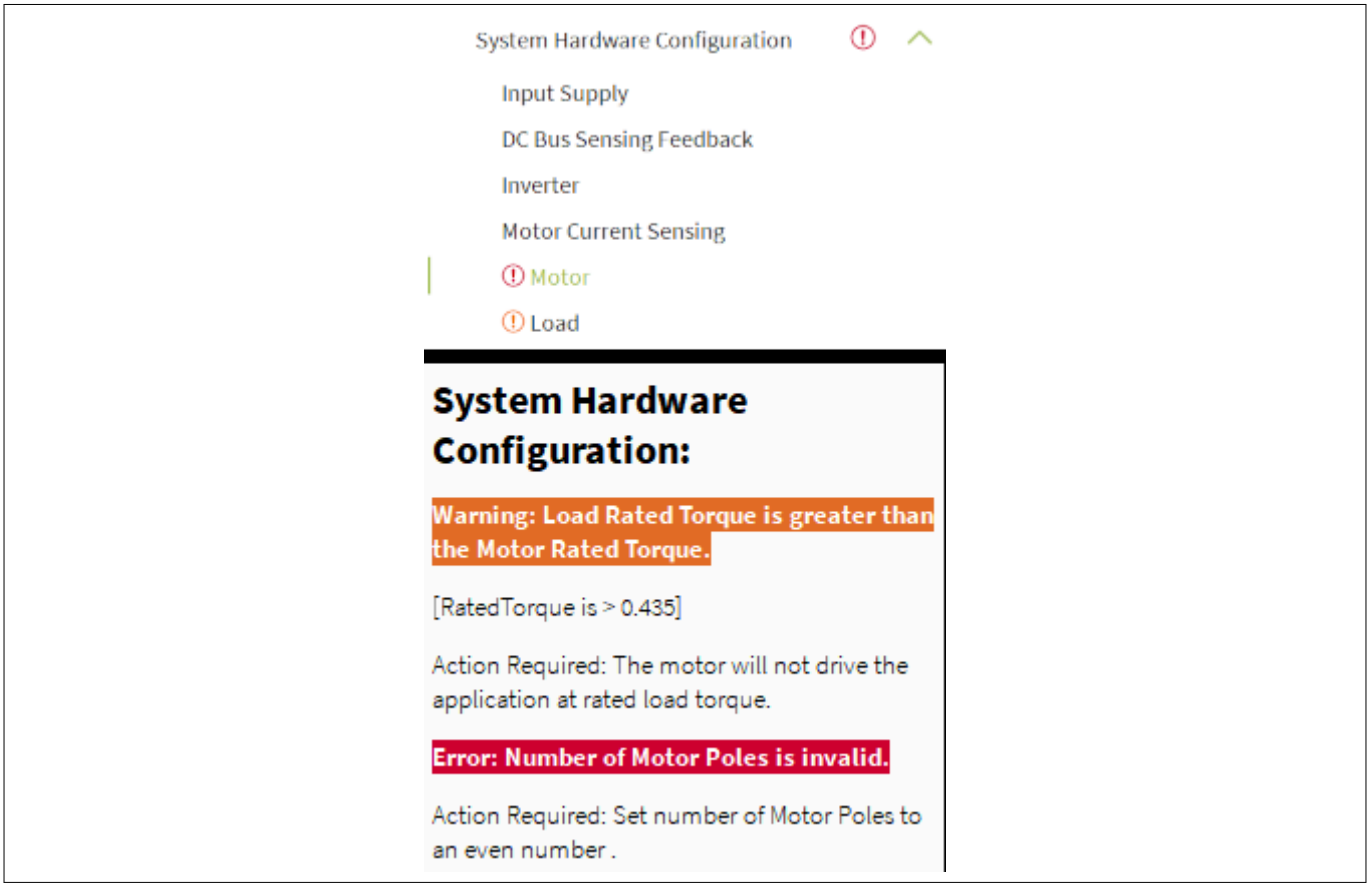

Figure 29 Verification that has Produced Warnings and Errors

5.9 Build Project

When a user starts the build process, the **MCE parameter binary file** (or configuration file) is generated. The file generated is based on the Configuration Wizards settings and compiles the **script ByteCode file** (if applicable) into the **build output file**.

The build output **.ldf file** consists of the configured parameter set date and script binary file. This file is located with the project directory (Project Location → generated) folder.

In the build process, only the script ByteCode file is compiled into the output file. Users are required to successfully compile the script project in the script editor before starting the build process.

5.9.1 Build Process

1. To start, users will configure all the required parameter set functions in the Configuration Wizard and, if required, will compile the script project in the script editor.
2. Users will then open the **Build Project** screen using Project drop-down menu.
3. Next, select the Parameter Set the project uses when the MCE first starts up. The program will default to myFirstParameterSet as the selected set.
4. If required, select the appropriate script project. The program will default to the active script project in the script editor.
5. Users will then start the build. This generates the .ldf output file.

***Note**: The program will attempt to Verify the project before it begins the build process. If an error is found, then the build process will be cancelled and no output file will be generated. If the user has already performed the verification process with the current settings, then users can skip the default verification process by unchecking the **Verify before build process** option.*

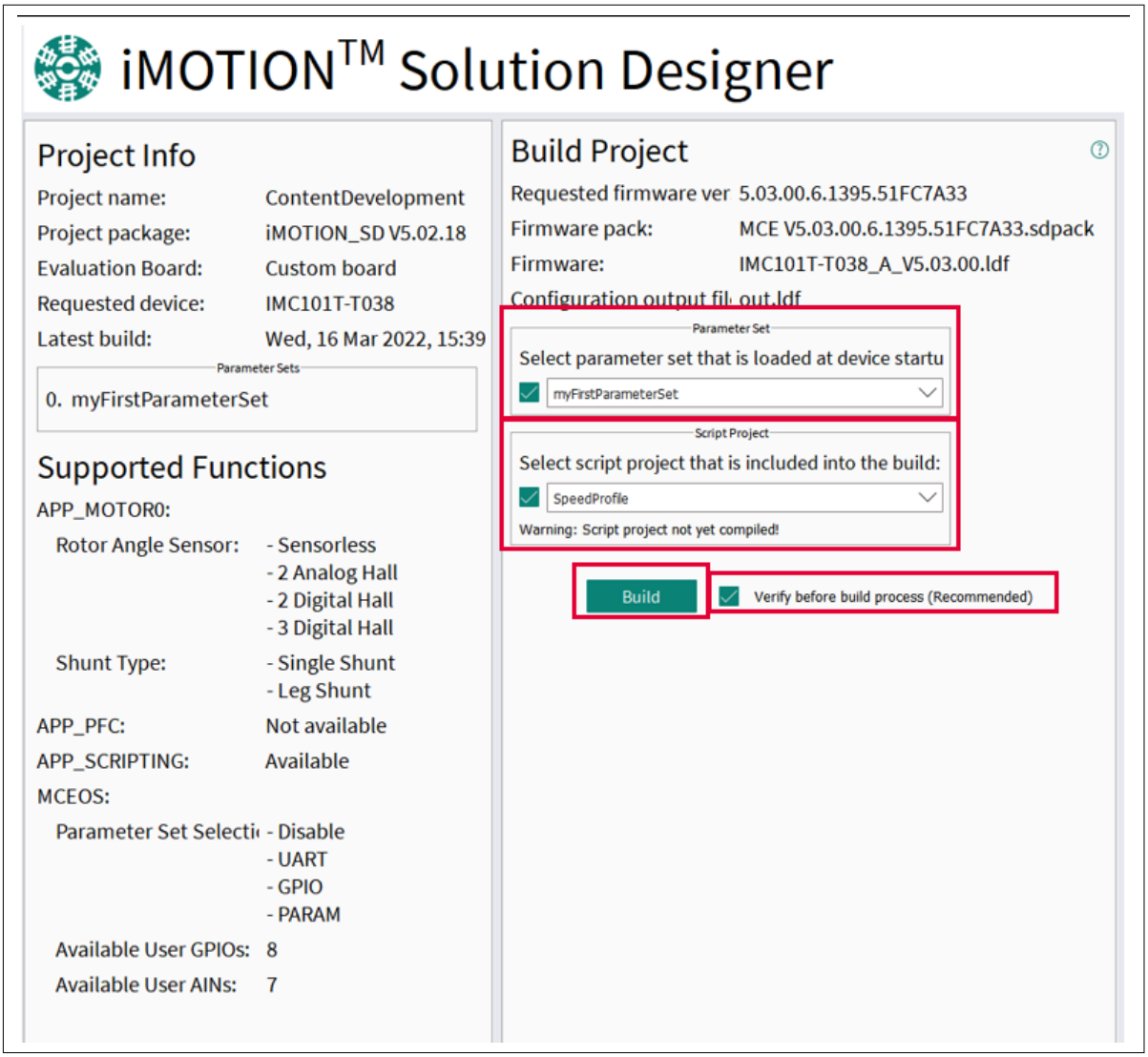

Figure 30 Build Project Pane

5.10 Parameter Set Manager

The **Parameter Set Manager** allows users to add, remove, edit, or import parameter sets. Users can access the manager via the Project drop-down menu on the menu bar.

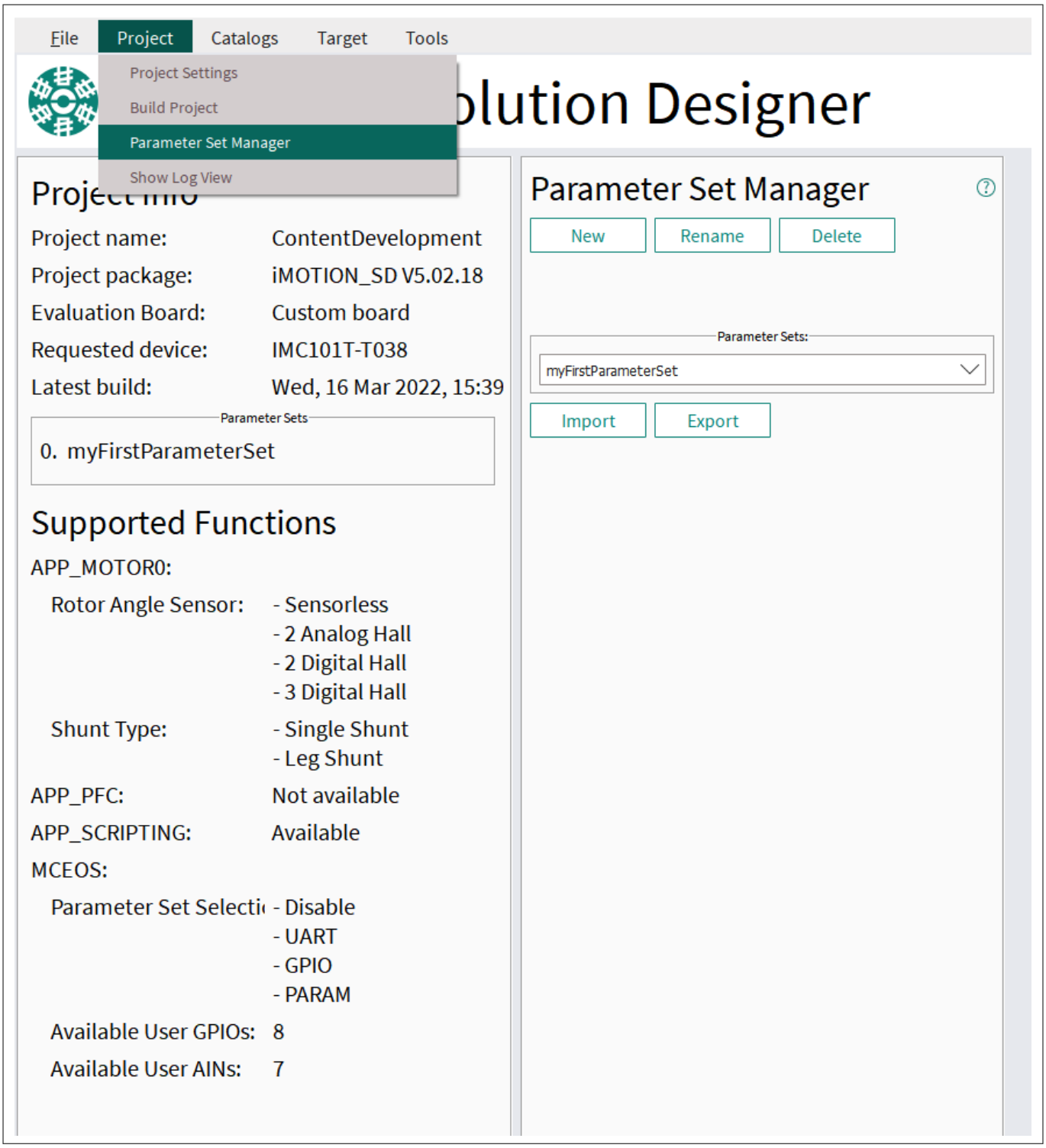

Figure 31 Parameter Set Manager

5.10.1 Working with Multiple Parameter Sets

In order to select which parameter is used when the device boots up, users will navigate to the **Build Project** tool from the Project drop-down menu.

Users can also change parameter sets they want to configure via the Configuration Wizard. Users will navigate to the Configuration Wizard and use the **Parameter Set Configuration** drop-down box to switch to the desired set.

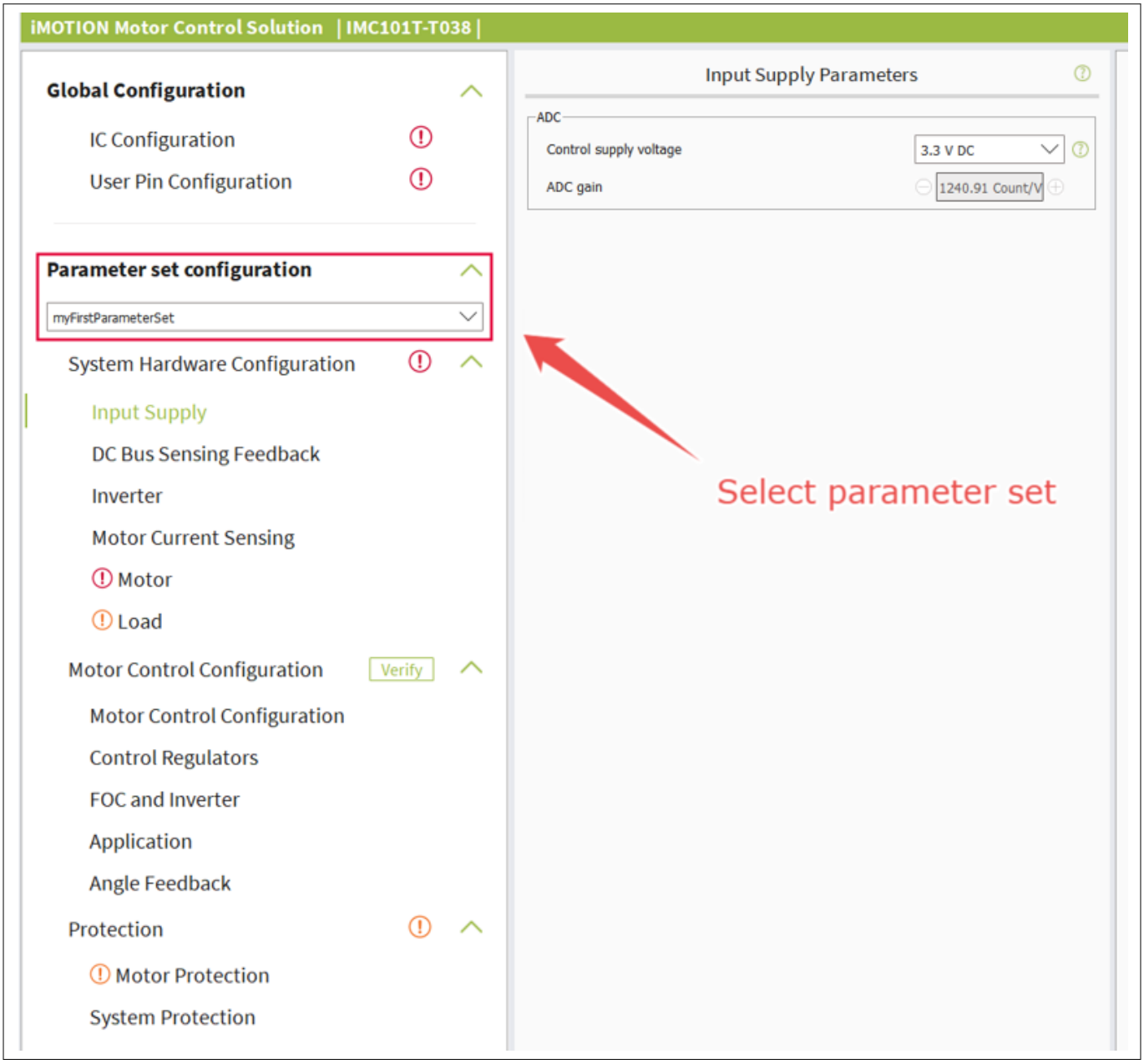

Figure 32 Parameter Set Configuration Drop-down Box

5.10.2 Global Parameter Set and Regular Parameter Set

Global Parameters have only one configuration per project, staying the same across all parameter sets. The global sections within Global Parameters are not included in the parameter sets, and are not included when a parameter set is exported.

Parameter Sets can hold up to fifteen different configurations per project. Users will find these configurations throughout the Hardware, Configuration, and Protection sections of the Configuration Wizard. These Parameter Sets will be included when exporting a parameter set.

5.10.3 Parameter File Location and Importing/Exporting Parameter Sets

File Location

Users can find the parameter set files by navigating to the project location, settings folder, then cw_data folder. This folder will contain all saved information regarding the current project. The file extension for parameter sets is **.icwp**.

Importing and Exporting Parameter Sets

***Note**: Before users begin, they should **Save** their project to ensure the exported settings are up-to-date.*

Users can begin exporting a parameter set by opening the Parameter Set Manager (Menu Bar → Project → Parameter Set Manager). From there, users will click the **Export** button, select a location, then name the file using the systems file explorer. Once done, users will click Save.

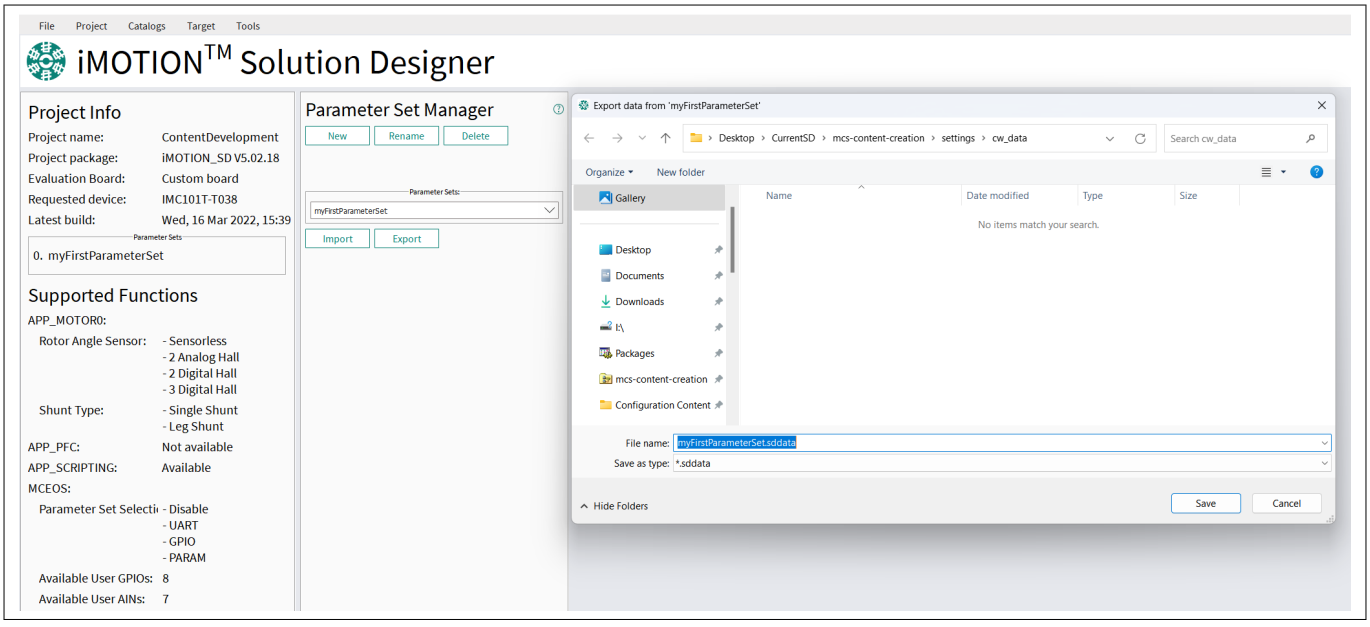

Figure 33 Saving a Parameter Set

Sharing a Parameter Set

After exporting a parameter set, users will find a file with the extension **.sddata** in the chosen location. This file can be shared to other iSD users.

5.11 Target Connection

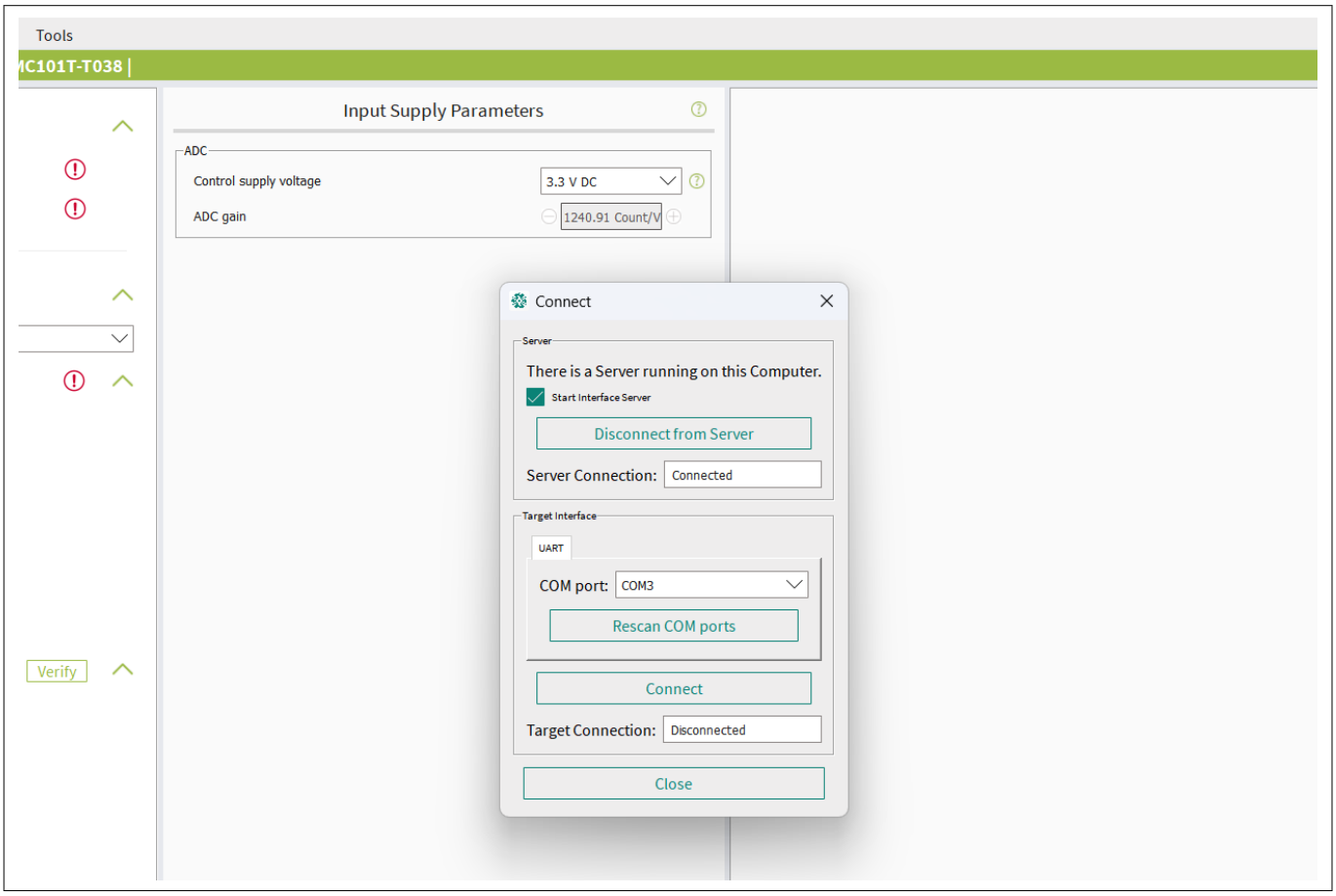

Figure 34 Connect Pop-up Window

Users will need to connect to a device (Target) when they want to use certain iSD tools and tasks. The list of actions that require a target connection are:

- Programing the device.
- Using the Oscilloscope tool.
- Using the Debugger for the Script Editor tool.
- Using the Tuning Dashboard.

What can be used without Connecting to a Target

- Configuration Wizard
- Parameter Set Manager
- Build Project
- Dashboard
- Script Editor

5.11.1 How to Connect

Before starting, users should ensure their firmware version is newer than FW5.0. If this is the user's first time connecting to the device, they must use the programmer to update the firmware.

***Note**: There must be an available UART for the Solution Designer and iMOTION™ device to interact.*

1. Using the menu bar, navigate to the **Target** drop-down menu and click the **Connect** button.
2. The Connect tool will default to being connected to the server. Users can disconnect from the server via the **Disconnect from Server** button. Users can connect or reconnect to the server via the **Connect to Server** button.
3. In the Connect pop-up window, under **Target Interface**, select the appropriate COM port for the device. Users can hover over a COM port to learn its associated profile name. By default, the iMOTION™ profile name is **JLink CDC UART Port**.
4. Click on the bottom **Connect** button to connect the iMOTION™ device.

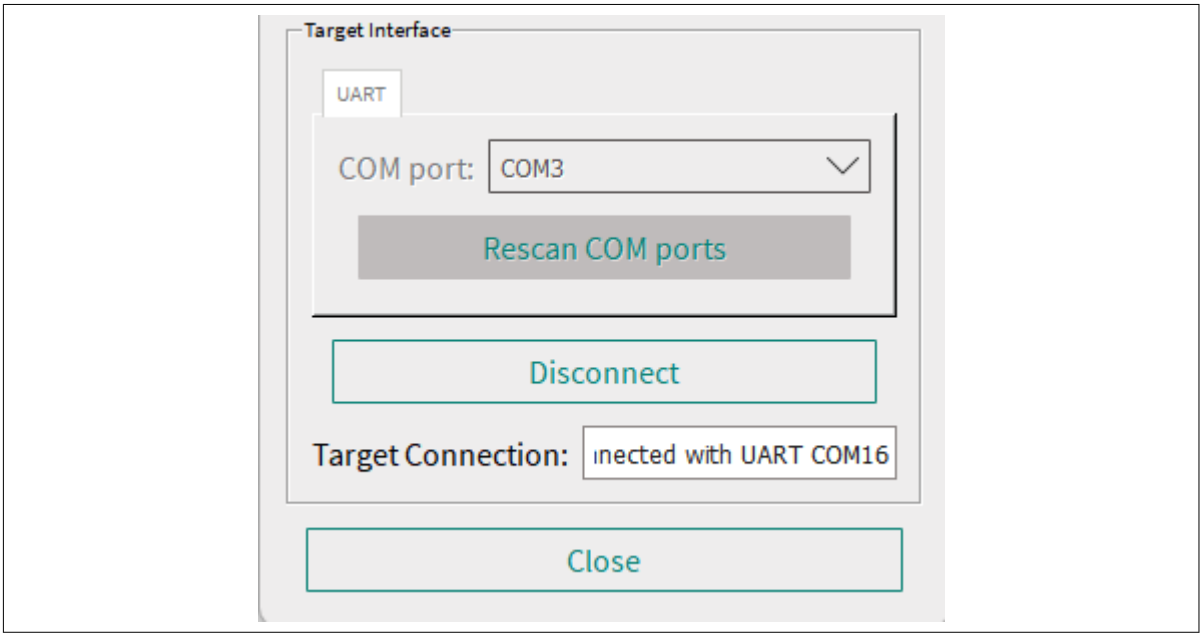

Figure 35 COM Port Drop-down Box

5. The **Target Connection** section will show the message **Connected with UART COM#** if the connection was successful.

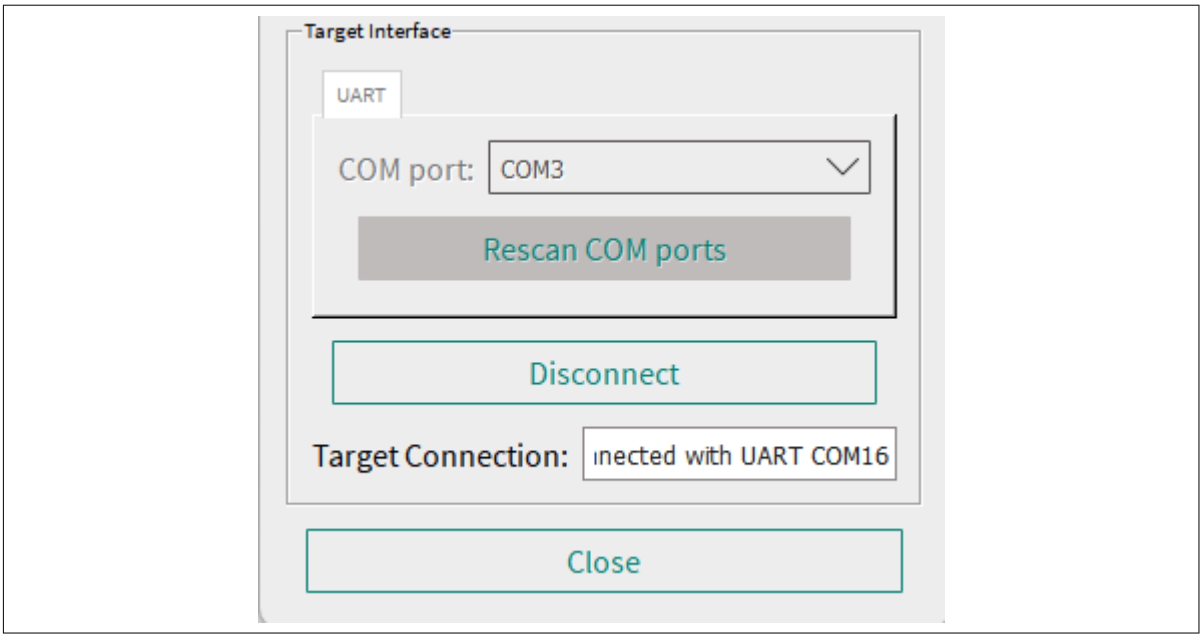

Figure 36 Target Connected to COM Port

5.11.2 How to tell when a Target is Connected

Users can see if a Target is connected by looking at the bottom of the Solution Designer window. In the bottom-left of the menu bar, users will see **Target Connected** or **Target Not Connected**.

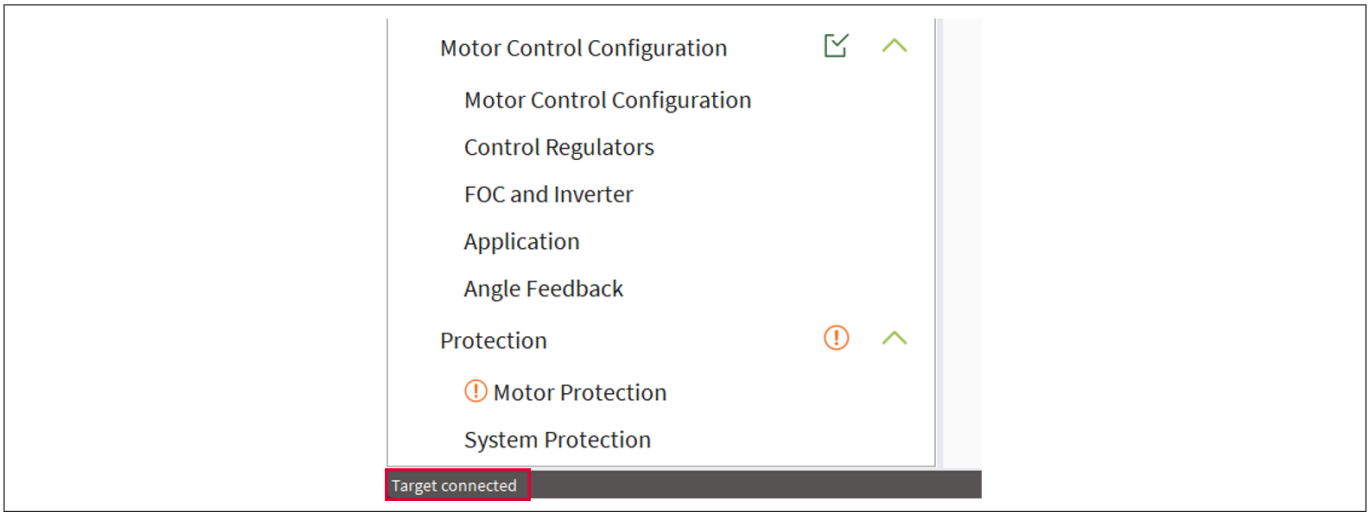

Figure 37 Target showing as Connected

6 Programmer

The **Programmer** allows users to download firmware, parameters, and script code to iMOTION™ hardware. There are two options to initiate device programming. First option is to go directly to the Programmer user interface which is launched by clicking the Programmer icon at the top of the Solution Designer window or by going to the 'Tools' drop-down menu. The second option is to launch the Dashboard by clicking the Dashboard icon at the top of the window or by going to the 'Tools' drop-down menu. Before launching the Dashboard, iSD checks the latest build against the actual firmware on the device and if they do not match, the user is asked if firmware programming is requested before proceeding to the Dashboard. The programming sequence depends on which launch method is used to initiate programming.

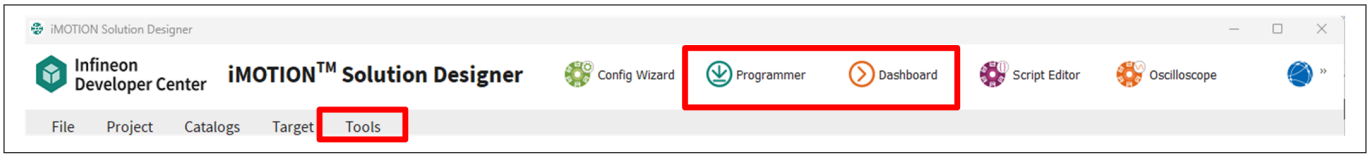

Figure 38 Options For Initiating Programming

In addition to the Programmer launch method, the initial state of the Device also affect the programming sequence. The possible initial states of a device are:

- Previously programmed with firmware version FW5.0 or newer (new firmware)
- Previously programmed with firmware version FW1.03.07 or older (old firmware)
- Never been programmed (boot loader mode)

The programming sequences for each scenario will be discussed in the following sections.

The main Programmer window is shown in the figure below. It provides a simple user interface with status information and 3 controls that allows users to:

- Connect – Establish connection with the iMOTION™ hardware
- Program – Download all the relevant code and parameters to the iMOTION™ hardware
- Close – Closes programming session and allows user to switch back to other iSD tools

The watch window provides additional status information and pop-ups inform about the progress of the connection or download process.

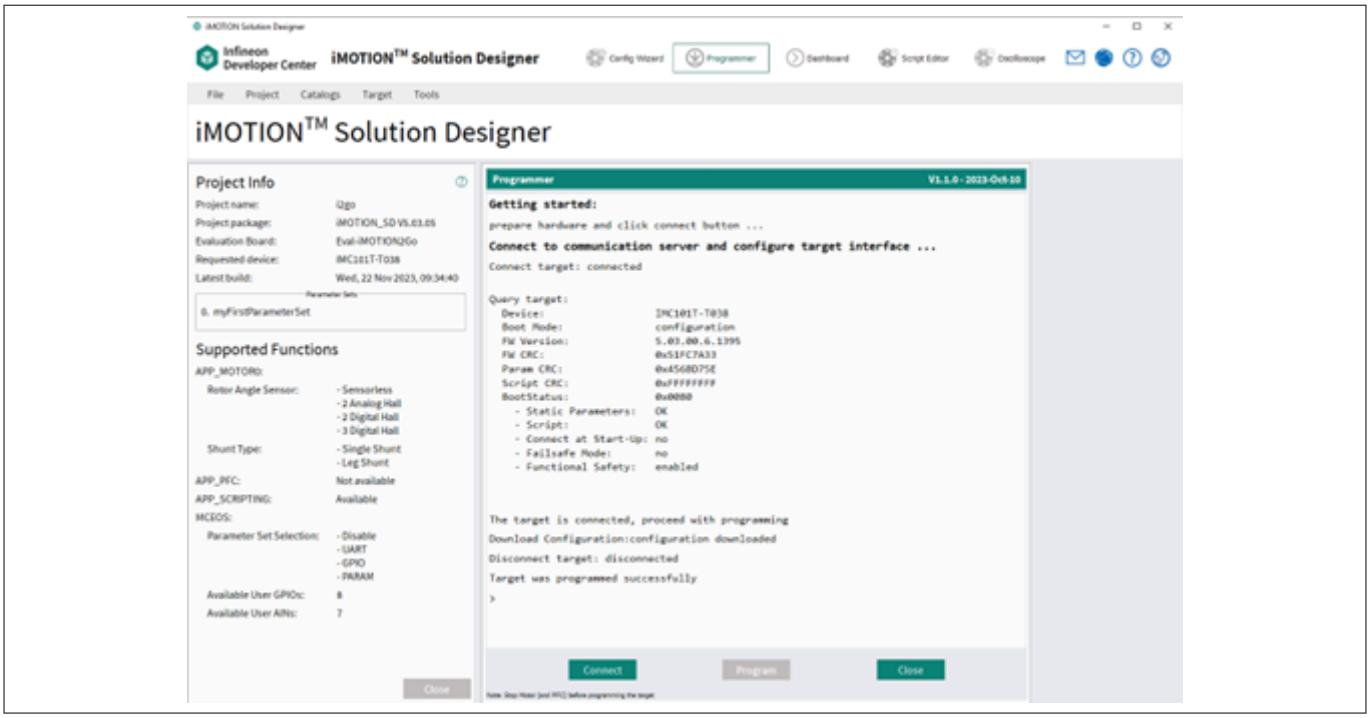

Figure 39 Programmer Landing Page

When connecting to the device the user is presented with the pop-up window shown below. The primary user options are:

- COM port – Communication port selection
- Connect –Connect to the device using the specified COM port
- Close –The action of 'Close' depends on the device under programming. The possible actions are:
  - Closes programming session and allows the user to switch back to other iSD tools
  - Connects to a device in boot loader mode

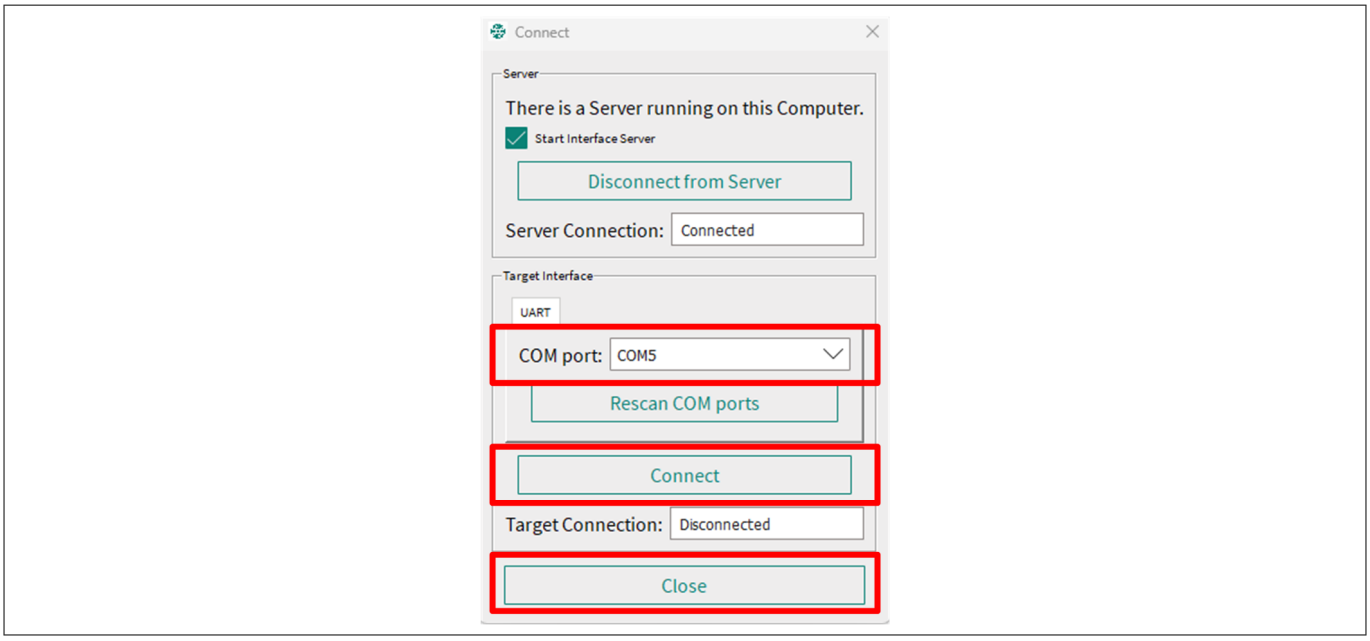

Figure 40 Connect Menu With Relevant Options For Programming Highlighted

6.1 Devices Previously Programmed With New Firmware

If the device has previously been programmed with firmware version FW5.0 or newer, and programming is done by launching the Programmer directly, the sequence is as follows:

1. Build the iSD project and apply power to the device
2. Click the ‘Programmer’ icon at the top of the iSD window or go to the 'Tools' menu to launch the Programmer
3. Click ‘Connect’ in the programmer window
4. Connect window pops up. Select Com Port and click 'Connect' in the pop-up window
5. If connection is successful, device information is shown in the programmer window
6. Click ‘Program’ in the Programmer window
7. After successful programming, click 'Close' to close the programmer window
8. iSD remains connected to the device. If switching to the Dashboard after programming, it is not necessary to re-connect to the device

If the device has previously been programmed with firmware version FW5.0 or newer, and programming is done by launching the Dashboard, the sequence is as follows:

1. Build the iSD project and apply power to the device
2. Click the 'Dashboard’ icon at the top of the iSD window or go to the 'Tools' menu to launch the Dashboard
3. iSD asks if user wants to download parameters to target before switching to the Dashboard. Click ‘Yes’ to download parameters and firmware
4. Connect window pops up. Select Com Port and click 'Connect' in the pop-up window
5. Programming happens in the background and iSD automatically switches to the Dashboard when programming is done

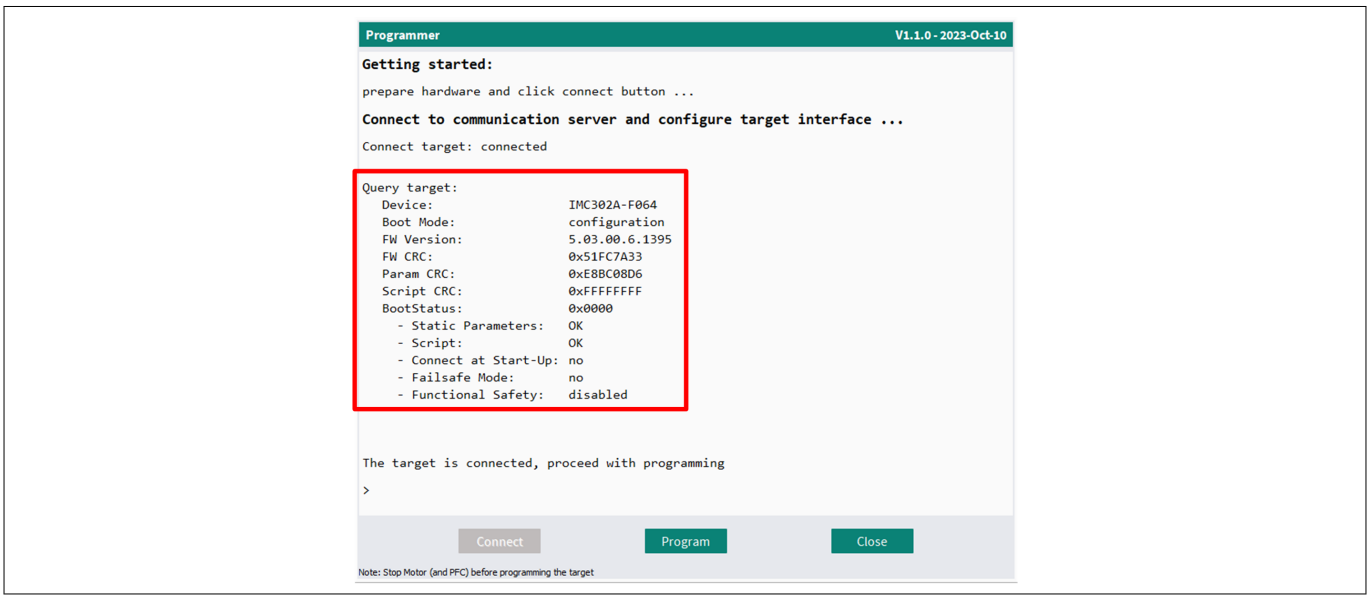

Figure 41 Programming Window After Connecting To A Device With Firmware Version FW5.0 Or Newer

6.2 Devices Previously Programmed With Old Firmware

Users who are using iMOTION™ hardware that has previously been used in combination with the legacy tools MCE Designer and MCE Wizard, will need to upgrade the firmware in order to be able to use the device with the iSD. Upgrading is done in Solution Designer and access to the legacy tools is not required.

If the device has previously been programmed with firmware version FW1.03.07 or older, and programming is done by launching the Programmer directly, the sequence is as follows:

1. Build the iSD project and apply power to the device
2. Click the ‘Programmer’ icon at the top of the iSD window or go to the 'Tools' menu to launch the Programmer
3. Click ‘Connect’ in the programmer window
4. Connect window pops up. Select Com Port and click 'Connect' in the pop-up window
5. If connection is successful, device information is shown in the programmer window
6. Click ‘Program’ in the programmer window
7. After successful programming, click 'Close' to close the programmer window
8. iSD automatically disconnects from the device after programming
9. If switching to the Dashboard after programming, iSD will display a pop-up message saying 'Target not connected. Do you want to connect to the target?’ Click 'Yes'
10. Connect window pops up. Reselect Com Port and click 'Connect' in the pop-up window
11. iSD switches to the Dashboard

If the device has previously been programmed with firmware version FW1.03.07 or older, and programming is done by launching the Dashboard, the sequence is as follows:

1. Build the iSD project and apply power to the device
2. Click the ‘Dashboard’ icon at the top of the iSD window or go to the 'Tools' menu to launch the Dashboard
3. iSD asks if the user wants to download parameters to target before switching to the Dashboard. Click ‘Yes’ to download parameters and firmware
4. Connect window pops up. Select Com Port and click 'Connect' in the pop-up window
5. Connect window pops up again. Reselect Com Port and click 'Connect' in the pop-up window
6. Programming happens in the background
7. iSD disconnects from the device
8. A pop-up window appears saying 'Target not connected. Do you want to connect to the target?’ Click 'Yes'
9. Connect window pops up again. Reselect Com Port and click 'Connect' in the pop-up window
10. iSD switches to the Dashboard

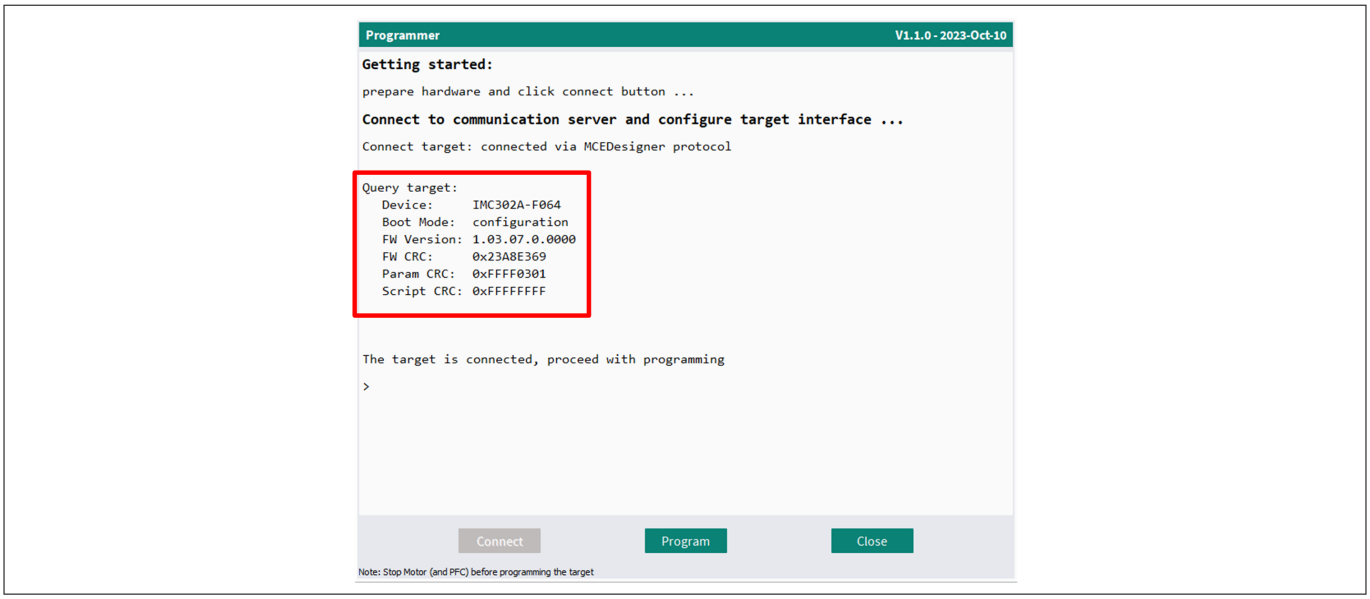

Figure 42 Programming Window After Connecting To A Device With Firmware Version FW1.03.07 Or Older

6.3 Devices That Have Never Been Programmed

If the device has never been programmed, the device is in Boot Loader mode (SBSL mode). When programming is done by launching the Programmer directly, the sequence is as follows:

1. Build the iSD project and apply power to the device
2. Click the ‘Programmer’ icon at the top of the iSD window or go to the 'Tools' menu to launch the Programmer
3. Click ‘Connect’ in the programmer window
4. Connect window pops up. Select Com Port and click ‘Close’ in the Connect window. Do NOT click ‘Connect’
5. If connection is successful, iSD connects to the device in Boot Loader mode. Device information is shown in the programmer window
6. Click ‘Program’ in the programmer window
7. After successful programming, click 'Close' to close the programmer window
8. iSD disconnects from the device after programming
9. If switching to the dashboard after programming, iSD will display a pop-up message saying 'Target not connected. Do you want to connect to the target?’ Click 'Yes'
10. Connect window pops up. Reselect Com Port and click 'Connect' in the pop-up window
11. iSD switches to the Dashboard

If the device has never been programmed and programming is done by launching the Dashboard, the sequence is as follows:

1. Build the iSD project and apply power to the device
2. Click the ‘Dashboard’ icon at the top of the iSD window or go to the 'Tools' menu to launch the Dashboard
3. iSD asks if the user wants to download parameters to target before switching to the Dashboard. Click ‘Yes’ to download parameters and firmware
4. Connect window pops up. Select Com Port and click ‘Close’ in the connect window. Do NOT click ‘Connect’

5. Connect window pops up again. Reselect Com Port and click ‘Close’ in the connect window. Do NOT click ‘Connect’
6. Programming happens in the background
7. iSD automatically disconnects from the device
8. A pop-up window appears saying 'Target not connected. Do you want to connect to the target?’ Click 'Yes'
9. Connect window pops up. Reselect Com Port. This time click 'Connect' in the pop-up window. Do NOT click 'Close'
10. iSD switches to the dashboard

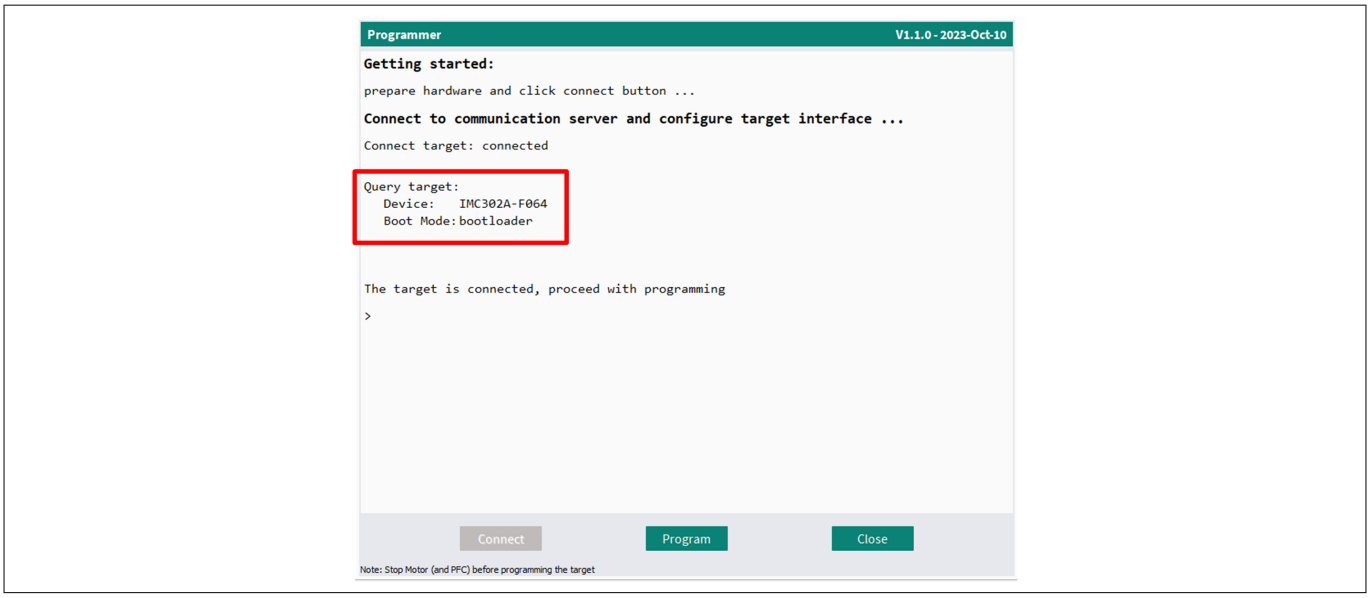

Figure 43 Programming Window After Connecting To A Blank Device

6.4 Downgrading Firmware

Users who are leveraging iMOTION™ hardware that has been used in combination with the iSD, and who want to revert back to the legacy tools (MCE Designer and MCE Wizard), will need to downgrade the firmware. This will allow them to connect to the hardware.

To downgrade the firmware, users can follow these steps:

1. Change the **BootMode** from “Application” to “Configure” in Solution Designer when connected with board. Users will then find “Target disconnected” displayed in the dashboard.
2. Close Solution Designer and open MCE Designer
3. Ignore the warning information after opening MCE Designer. Use MCE Designer to download the firmware (FW1.03.07 or older) along with the parameter file.
4. The firmware will be downgraded.

6 Programmer

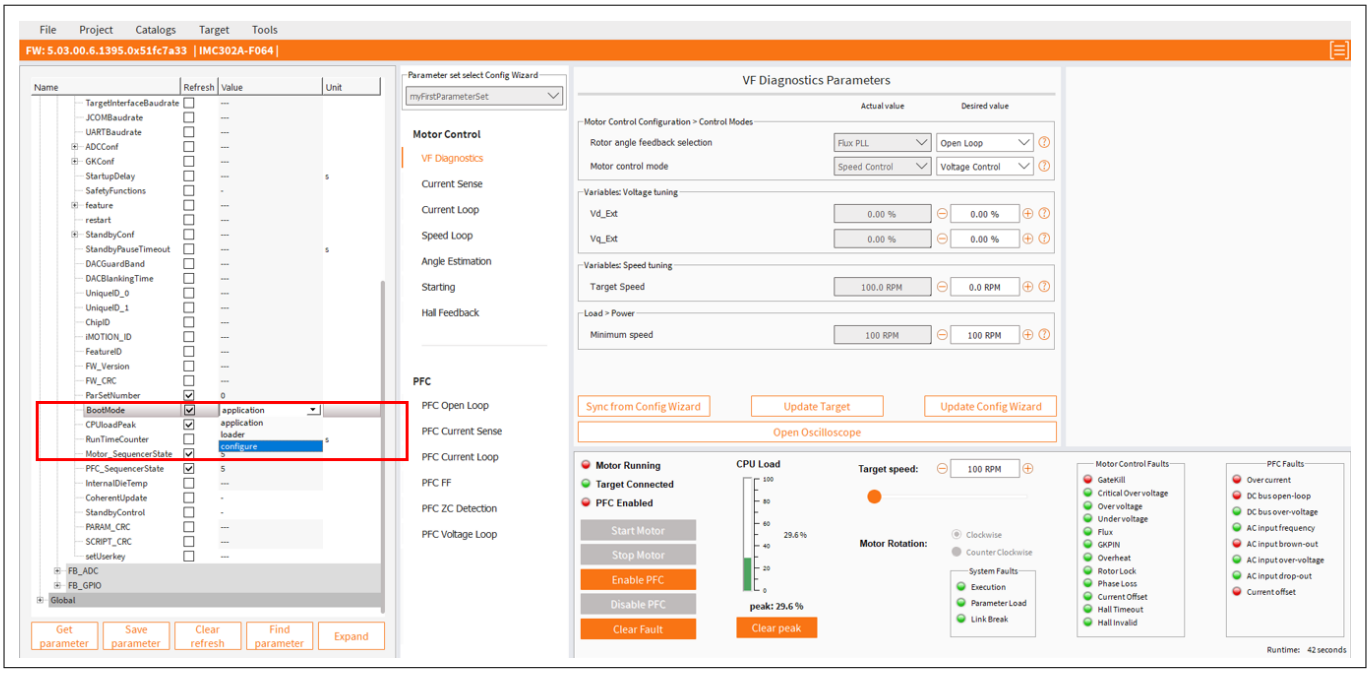

Figure 44 Firmware Downgrading Process

7 Dashboard

7.1 Introduction

The **Dashboard** tool can be used to monitor and edit MCE parameters and variables during runtime. Additionally, changes made in the Dashboard can be transferred into the Config Wizard with a single button click to modify the current system configuration. Each tuning case has its own Oscilloscope configuration which contain custom channel selections and settings .The tuning case includes a parameter tree for updating/ viewing variables and parameters at runtime, an tuning case panel for changing the active tuning case, a tuning case content window for updating a set of parameters and variables to help tune a motor or PFC, a help window for viewing help related to tuning parameters, a verification window for looking at verification results, and a control panel for sending basic commands to start/stop a motor or clear/view faults. The image below shows how these sections are split.

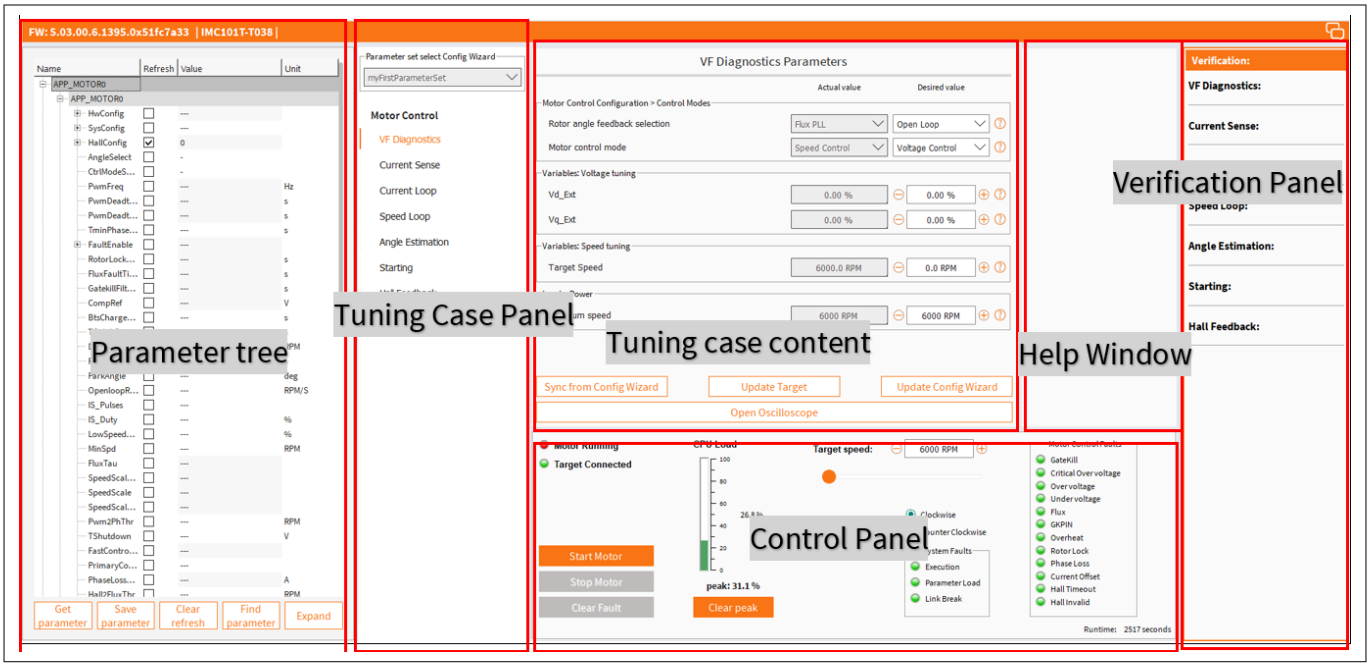

Figure 45 Dashboard overview

***Note**: Editing the parameters in the Dashboard section will change them in the RAM, and not in the flash. In case of a restart of the device, the parameters from the flash will be read and used to update the Dashboard.*

7.2 Parameter Tree

The **Parameter Tree** displays all available parameters and variables of the target. The parameter tree with Motor Control Parameters is displayed in the image below.

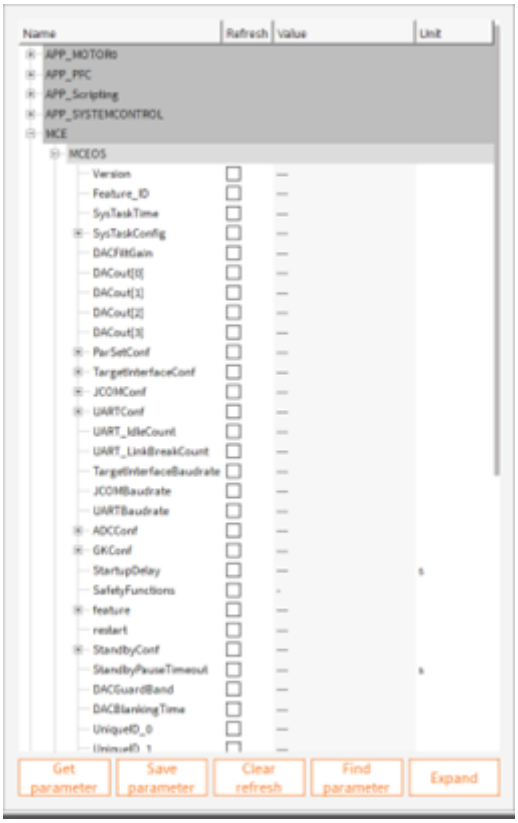

Figure 46 Parameter Tree

In order to update the parameters, users must click the boxes in the **Refresh** column. By right-clicking on the name of any column of the Parameter Tree, additional columns can be added or removed. Menu to add or remove columns is displayed below.

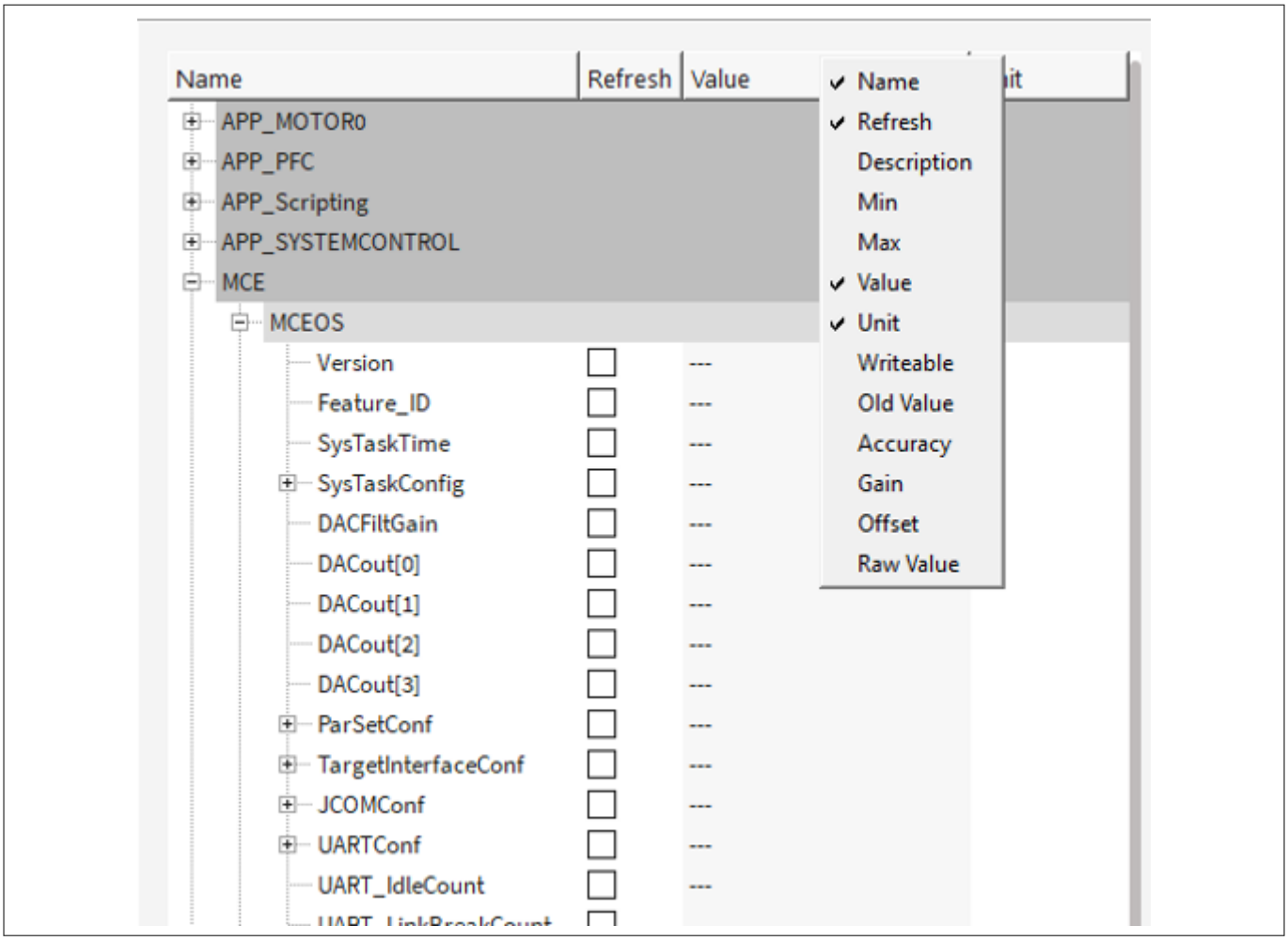

Figure 47 Adding and Removing Columns

The buttons at the bottom of the pane allow users to quickly navigate and change their parameter settings:

- Get parameter - Read all parameters from the target and display them in the parameter tree.
- Save parameter - Save all parameters as a .csv file.
- Clear refresh - Uncheck all checkboxes in the refresh column.
- Find parameter - Search for a parameter in the parameter tree.
- Expand - Expand the parameter tree structure.
- Collapse - Collapse the parameter tree structure.

***Note**:* *Note: Users will find some Refresh boxes are checked by default. They should be left as is. Avoid clicking "Clear Refresh" button since it will remove all the checked boxes. These refresh options are crucial for the tuning cases to function properly.*

7.3 Tuning Cases Menu

The **Dashboard Menu** lists all available control tuning use cases for the selected device. Additional dashboard options will be available when a PFC-compatible device is used, as shown in the image below.

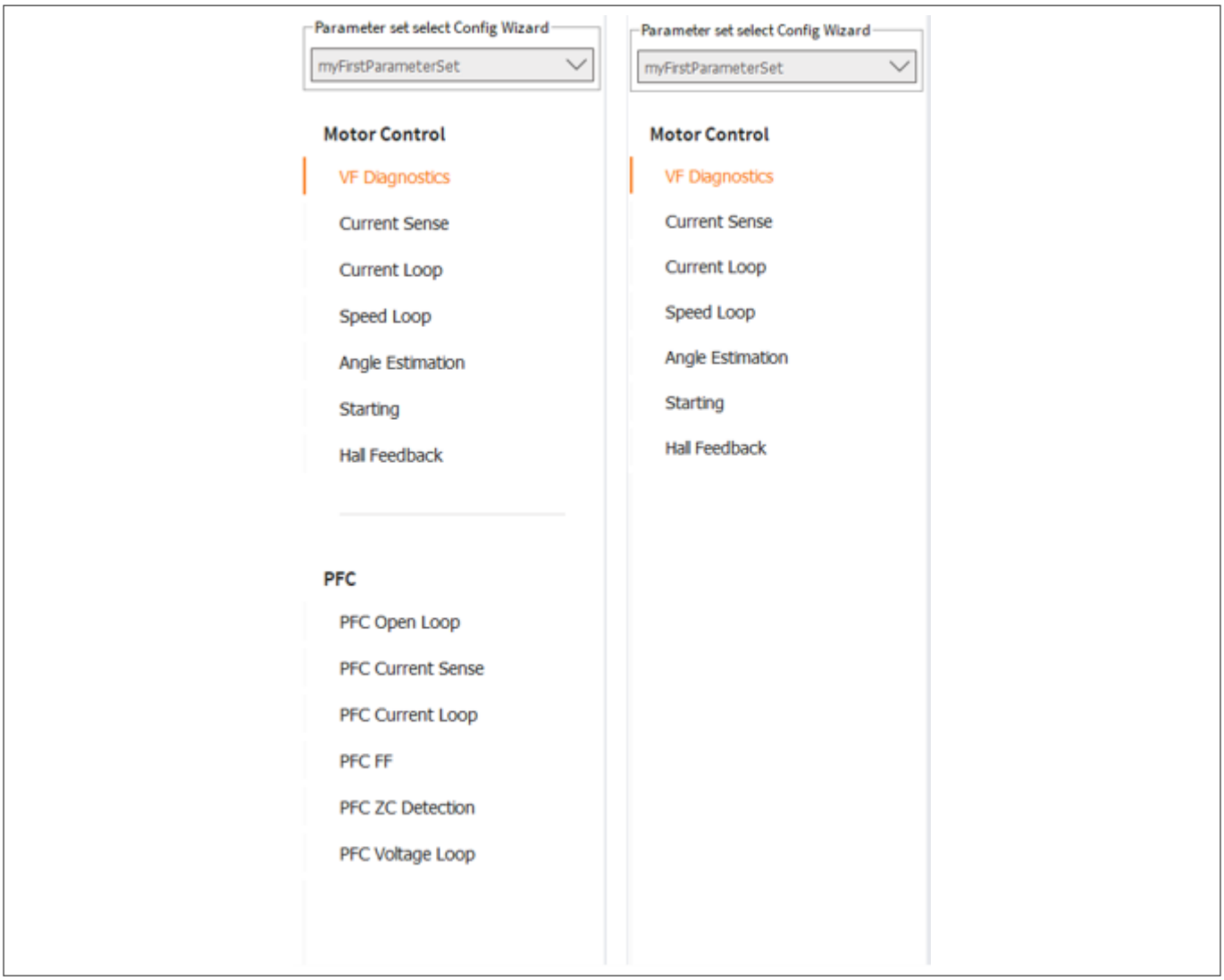

Figure 48 PFC Dashboard Menu and Motor Control Dashboard Menu

7.4 Tuning Cases Content

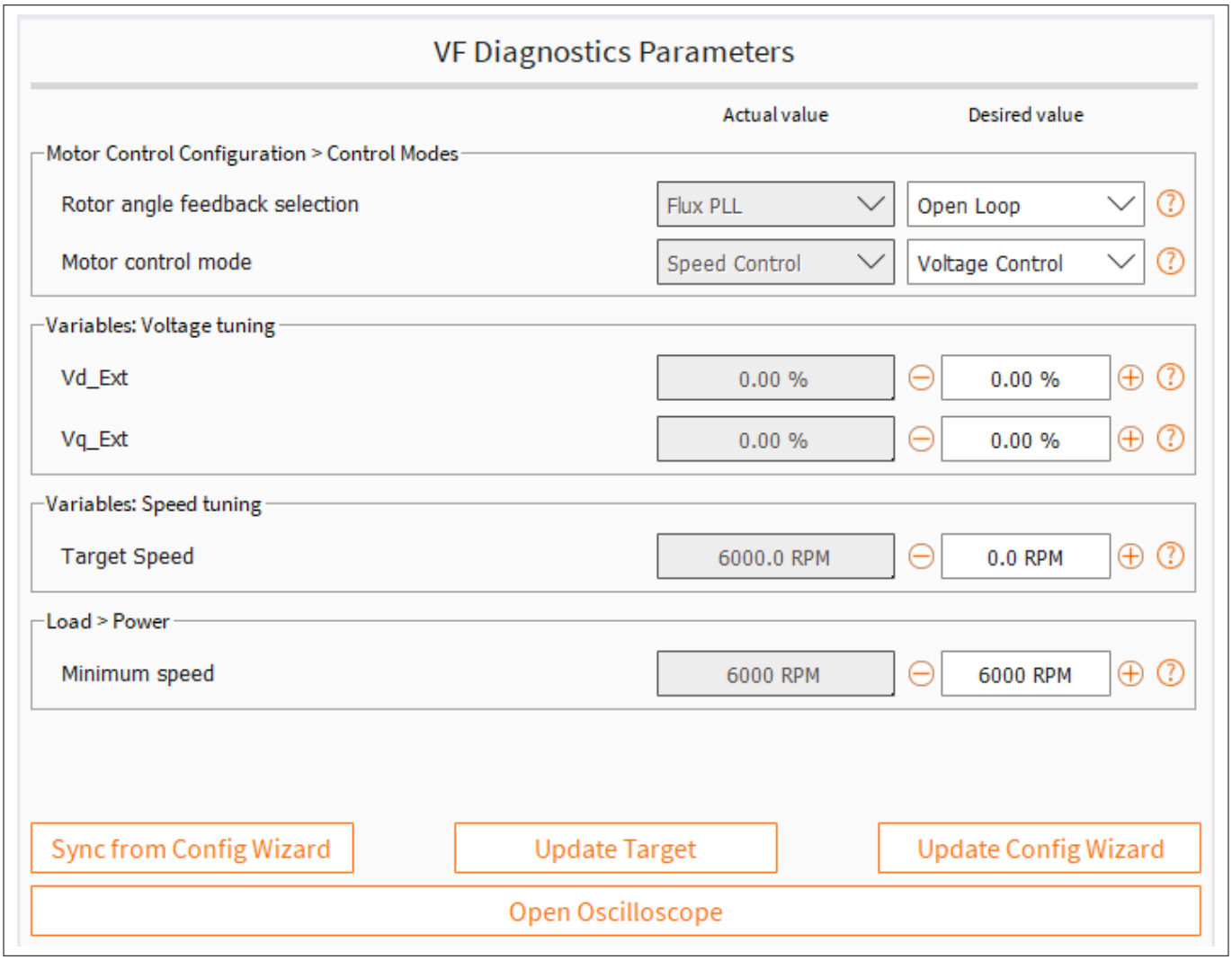

Figure 49 VF Diagnostics Parameters

Each Dashboard displays select parameters and variables based on the tuning case. Parameters can be changed in the config wizard, while variables are only for run-time use and can be adjusted when tuning for a specific tuning case. To differentiate between a variable and a parameter, simply check the group name. If it begins with "Variable," it's a variable. If it's a parameter, the group will include the Configuration wizard's location followed by a ' > ' and the name of the specific group.

Each entry can be edited by a spin box or a drop-down box. The user has the options to edit the desired value and to view the actual value stored in the target's RAM. This makes it possible to detect minor differences between the actual and desired values.

***Note**: Due to conversion from physical values into integer values, the desired value and the actual value may vary.*

7.4.1 Tracking changes

To keep track of any differences between the parameter values entered into Dashboard and the parameter values of the of the Config Wizard, dashboard parameters are highlighted when not in alignment with the Config Wizard. Additionally, an asterisk will appear next to the tuning case name after a parameter has been modified within the Dashboard as shown in image below. .

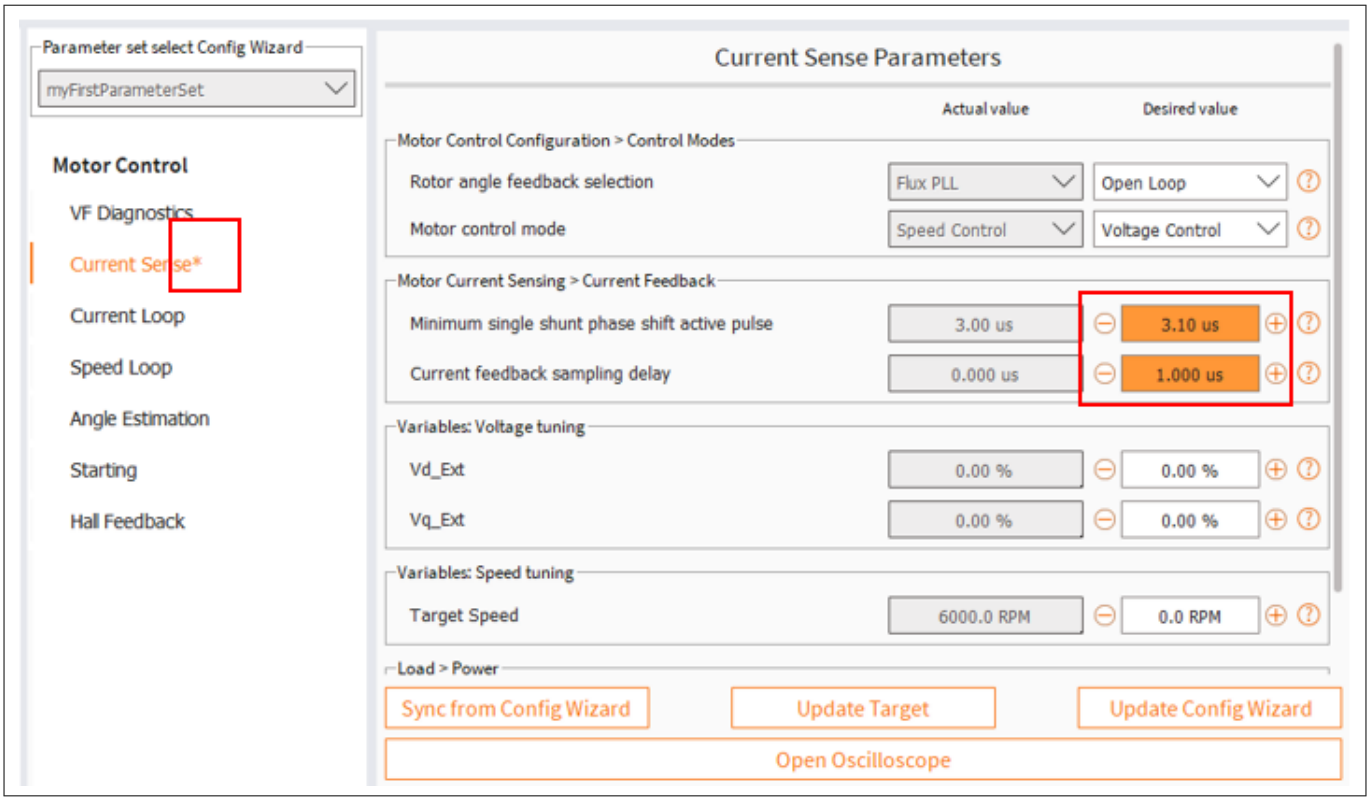

Figure 50 Identification of edits and difference between the Dashboard and Config Wizard

A few parameters are not tracked as they are not part of the configuration but rather used to enable a tuning case in the dashboard. The parameters that are not tracked by Config Wizard are:

- Motor Control
  - Rotor Angle feedback selection
  - Motor Control Mode
- PFC
  - Control Mode

Variables are not tracked and are not highlighted when edited.

Color highlighting is cleared by bringing the parameters of the Config Wizard and the Dashboard in alignment. In other words, highlight only happens when values differ between Config wizard and dashboard. The methods to do this are:

- Use the "Update Config wizard" button, which will synchronize the values in the Config wizard with those in the dashboard
- Use the "Sync from Config wizard" button, which will synchronize the values in dashboard with those in Config wizards
- Manually change the values to match those in the Config wizard

7.4.2 Sync from config wizard

The Dashboard's "Sync From Config Wizard" feature populates the desired value column (only the current tuning case) with the values from the active Config Wizard parameter set. Also, MCE Variables will be set to zero. Additionally, all tuning cases will be automatically synced each time a user opens Dashboard.

7.4.3 Update Config Wizard

The Dashboard's "Update Config Wizard" feature overwrites the parameters in the Configuration Wizard's active parameter set with the current values in the current tuning case without saving Solution Designer Project. This feature only updates Config Wizard from the current tuning case, if changes are made in other tuning cases, the "Update Config Wizard" must be used in each tuning case.

***Note**:* *Note: Since this feature makes the parameters in the Configuration Wizard and the Dashboard to be equal, it will clear the highlighting.*

7.4.4 Update Target

The Dashboard's "Update Target" feature allows users to update the target's RAM with the current tuning case values. The process involves verification and checks for parameters that should not be updated while motor is running (ControlMode, MinSpd, and AngleSelect). Once all values are converted to counts, they are stored in the target using a coherent update method that updates all parameters/variables at once.

***Note**:* *Note: Since updating is only storing the values in RAM, these values will be reset to the previous programmed values if target is reset.*

7.4.4.1 Verification

After the user clicks "Update Target", a set of verifications are performed to ensure that the configuration entered to the iMotion device is correct. If the verification fails, a warning or error message will be shown. If it is classified as a warning, the device can still be updated, but if there is at least one error, the "Update Target" process will not be allowed to proceed. In other words, errors will prevent the update process, while warnings will still allow it to continue. Warnings are displayed in yellow, while errors are displayed in orange. If an error or warning is encountered, a suggestion will be displayed as shown in the image below (Error on the left, Warning in the center, Success on the right).

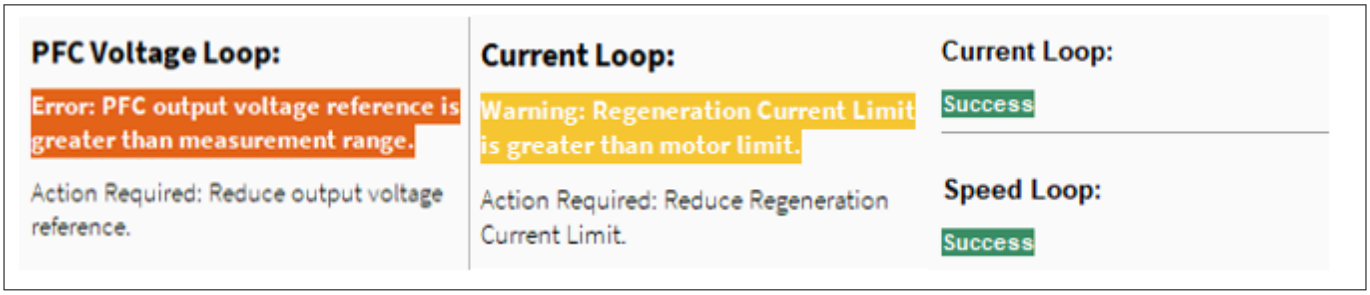

Figure 51 PFC Validations.

7.5 Uncommitted Changes

It is important to update the parameters in Configuration Wizard before leaving the Dashboard to avoid data loss. Tuning cases' data loss can happen when you navigate to Configuration Wizard, programmer, or when you close the program. Therefore, it is crucial to ensure that all changes are saved before exiting the Dashboard. Upon leaving the dashboard, a message will appear indicating which parameters have been modified, with the original and new values displayed for each. See the image below for an example.

Figure 52 Uncommitted changes to config wizard

7.6 Open Oscilloscope

The Open Oscilloscope button will let users access the Oscilloscope tool. Each dashboard has a predefined set of parameters and triggers already defined which will be used to populate the oscilloscope.

7.7 Help Doc

To access help for any questions, simply click on the question mark button located on the right side of the question as shown in Figure 52. This will bring up a help section on the right side of the screen. If you want to access help documentation in a separate window, double-click the question mark button.

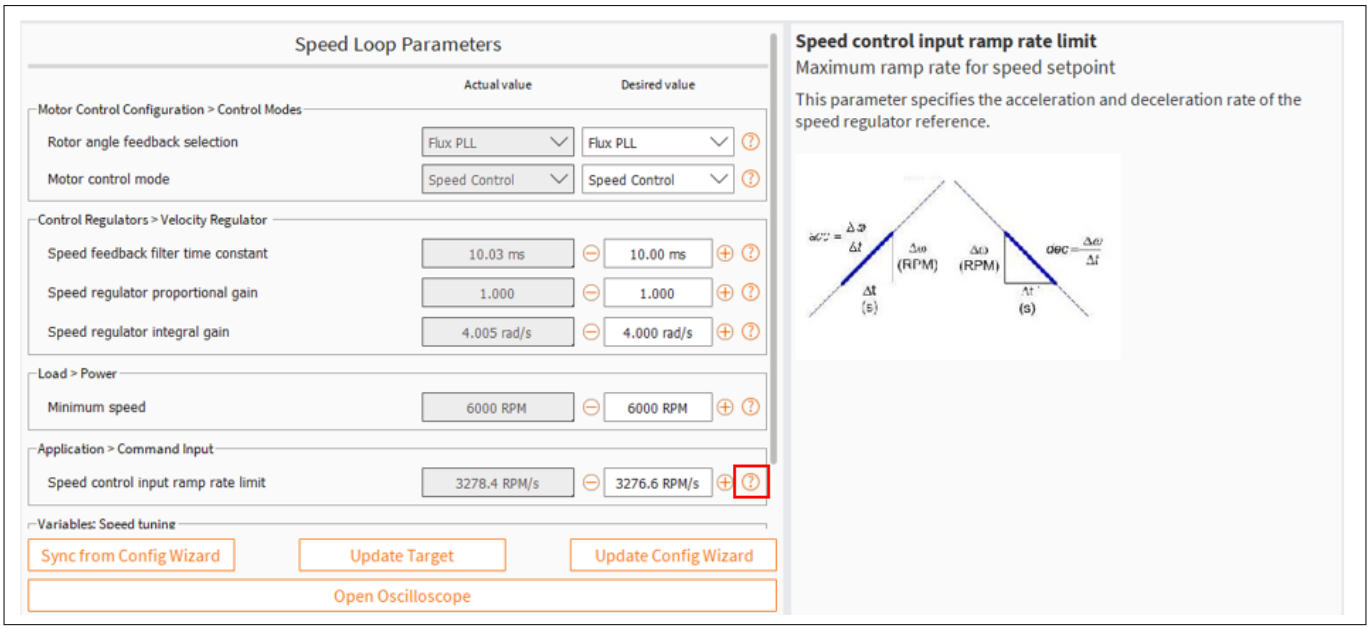

Figure 53 Dashboard Help doc

7.8 Working With Multiple Parameter Set

The active parameter set is chosen based on the ‘multiple parameter input mode’ selection in the Config Wizard. The options are ‘PARAM’, ‘GPIO’, ‘UART’ or ‘Disabled’. When ‘Disabled’ is selected the user manually specifies which parameter set to use.

When tuning multiple parameters sets using the dashboards, the user must set the ‘multiple parameter input mode’ to ‘Disable’.

The desired value columns of the dashboards are populated based on the active parameter set in the Config Wizard. Selecting ‘Disable’ guarantees the parameter values on the target matches those of the Config Wizard as well as those pre-filled into dashboard’s desired values.

To determine which parameter set is being used to pre-fill the desired values, users can check the combo box located under the "Tuning cases" title section.

7.9 Control Panel

Via the control panel, direct commands can be sent to the target. In addition to that, basic information from the target will be displayed as well.

7.9.1 Input Features

- Start/Stop Motor: Sending Start/Stop motor command to the target
- Enable/Disable PFC: Sending Start/Stop PFC command to the target
- Clear Fault: Clear Motor Control Faults
- Target Speed: Set the target speed (via spin box or via slider)
- Motor Rotation: set motor rotation to clockwise/counter clockwise

7.9.2 Output Features

- Runtime
  - Displays the current runtime (in seconds)
- Motor running LED
  - Green: Motor enabled
  - Red: Motor not enabled
- Target connected LED
  - Green: Target connected
  - Red: Target not connected
- PFC running LED
  - Green: PFC enabled
  - Red: PFC not enabled
- CPU load bar
  - Display the current CPU load of the target
  - Display CPU load peak of the target as text information
- Motor Control Fault LEDs
  - Green: no fault
  - Red: fault
- PFC Fault LEDs
  - Green: no fault
  - Red: fault

8 Oscilloscope

The **Oscilloscope** tool allows users to monitor control and measurement signals while the system is running. It emulates the function of a laboratory sampling oscilloscope used to debug hardware. However, instead of sampling voltages on a circuit board it samples control system variables stored in the device memory. The iMOTION™ software on the device includes a debug agent that samples the data and transmits frames of data to the PC hosting the iSD software. The Oscilloscope displays the signals on the screen and allows the user to select the desired signal sources, samples rates, and trigger events just like a laboratory oscilloscope. However, the Oscilloscope cannot continuously display captured data because the debug agent needs to pause sampling operations while it transmits a complete frame of sampled data to the PC. The Oscilloscope includes a variety of functions to manage signal selection and capture, as well as mathematical tools to analyze captured data.

The user interface is mainly divided into two parts. These are the header (1) and the space below (2), which contains all areas needed for working with channels, signals, control, and logging.

Figure 54 Overview of the Oscilloscope

8.1 Layout

The content areas contain all spaces that are needed to adjust and control the data acquisition using the oscilloscope. Furthermore, analysis and measurement can be done on the data received. This functionality regards the following areas.

- Channels section (1): This section shows all channels that can be adjusted. Channels can be assigned to signals. Adding or deleting of channels is possible
- Overview section (2): This section shows the traced buffer content for all available channels. It provides an overview and a zoom window to allow the inspection of certain areas of interest
- Panels section (3): This section provides space to arrange and group channels individually. Every panel contains a separate table for data analysis results, when doing measurements using cursors. This is the main section for working on a signal's content
- Control section (4): This section provides all needed functionality to control acquisition (run, stop, time base, triggering) and cursor functionality to be used

Figure 55 Different content areas of Infineon oscilloscope

8.2 Setup

The Oscilloscope will only work properly with a complete motor control system. If using an iMOTION™ EVAL board, ensure that both the control board and power board are powered on and connected to each other, and that a motor is connected to your board.

If the iMOTION™ device is not powered or connected to the PC, opening the oscilloscope will not be possible as connecting to a target is required before the oscilloscope can open.

8.3 Getting Help

The tool header bar offers various ways of getting help on the tool. You can access help from any screen in the tool.

Figure 56 Header of Oscilloscope

Access the tool forum to exchange information with peers

Open the tool user guide

8.4 Panel

Users will initially see two panels when opening the Oscilloscope tool. Panels are containers used for signal monitoring, both as a group or alone. Users are able to create multiple panels.

8.4.1 Working with Panels

The Infineon Oscilloscope supports the depiction of more than one chart at a time. Therefore a panel concept is available, where different child windows can be used to structure the view.

A panel (1) is the space containing a chart (2). This enables the grouping of channels with the same physical background, and scaling in one chart, without the need for a different y-axis for each signal.

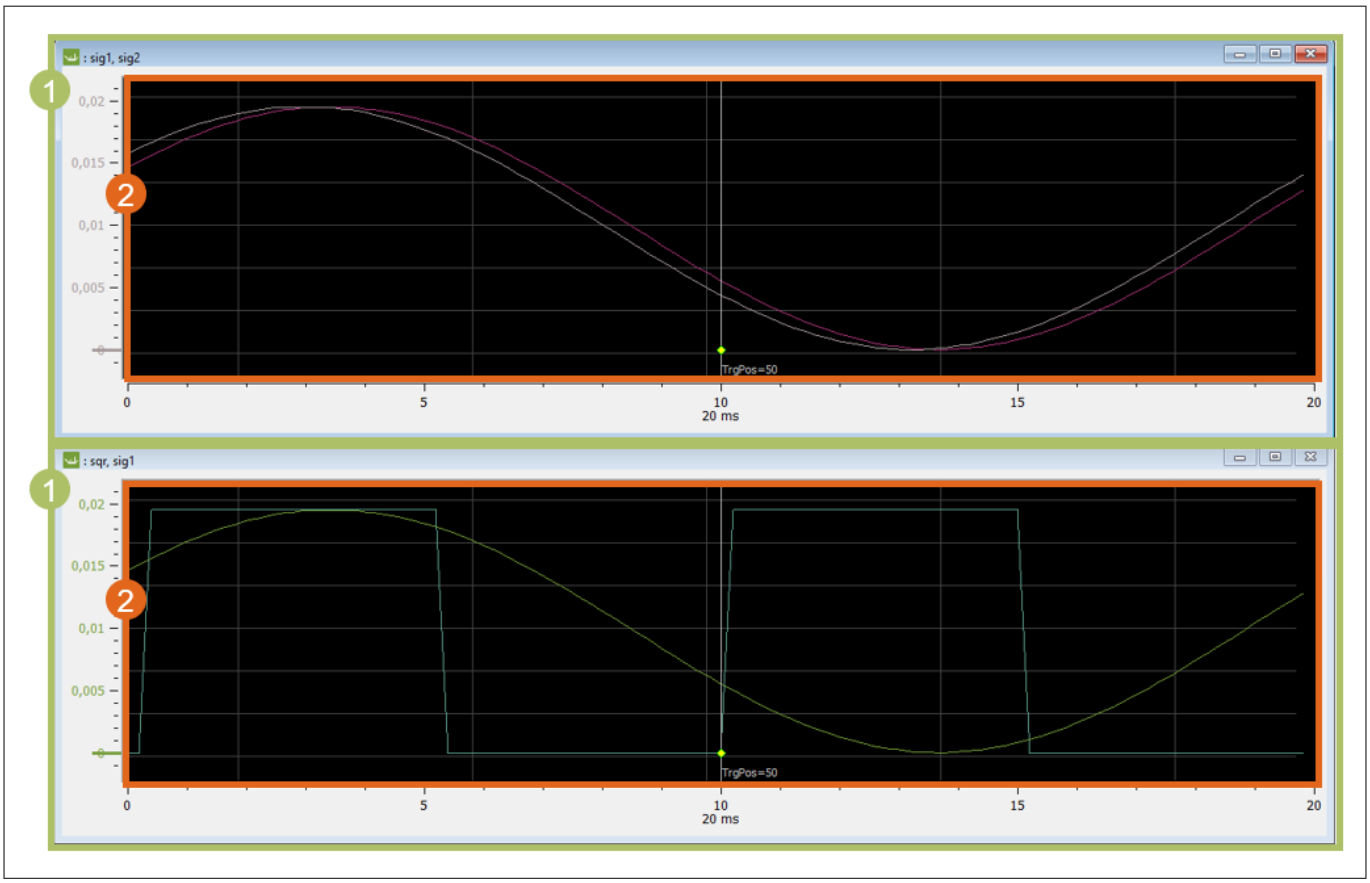

Figure 57 Grouping channels in panels

Various panels can be created by the user who can assign varying channels to each panel. After creation, all panels are arranged in rows automatically with connected time axis per default to ensure the same base for further work on the acquired data.

Users can add panels to the Oscilloscope project by clicking **Add Panel** in the File drop-down menu.

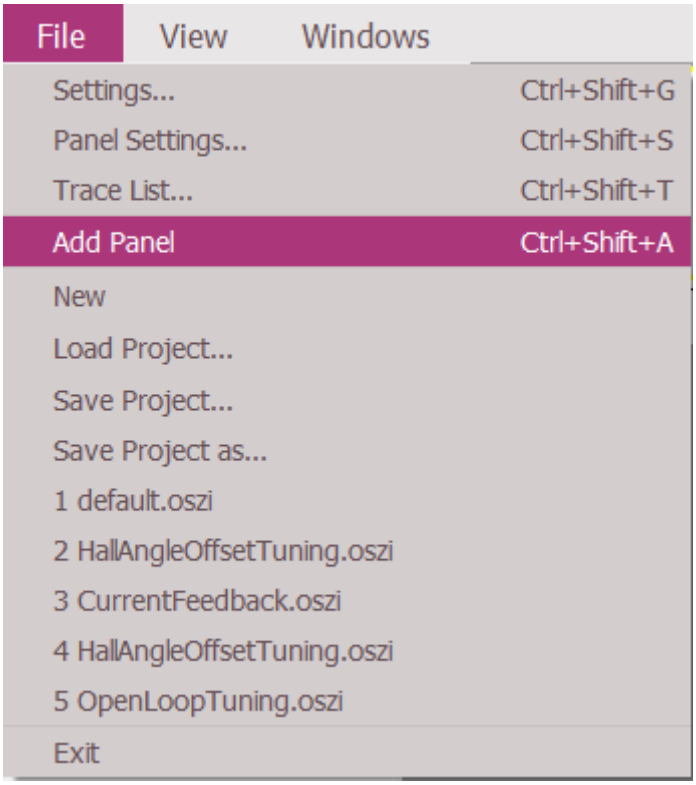

Figure 58 Add Panel in the File Menu

A pop-up pane will appear. Users will then fill out the required panel information, including the name of the panel and which signals will be displayed on the panel. This done in the **Y/T Signal Selection** section. Here users can scroll through all applicable signals, choosing each signal manually or selecting/deselecting all signals using the **Set All** and **Clear All** buttons respectively.

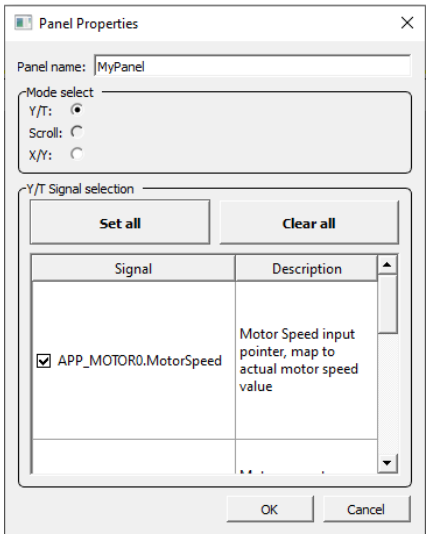

Figure 59 Panel Properties Window

The new panel will be added to the bottom of the panel list. The panel's name and signals will be displayed at the top of the panel window.

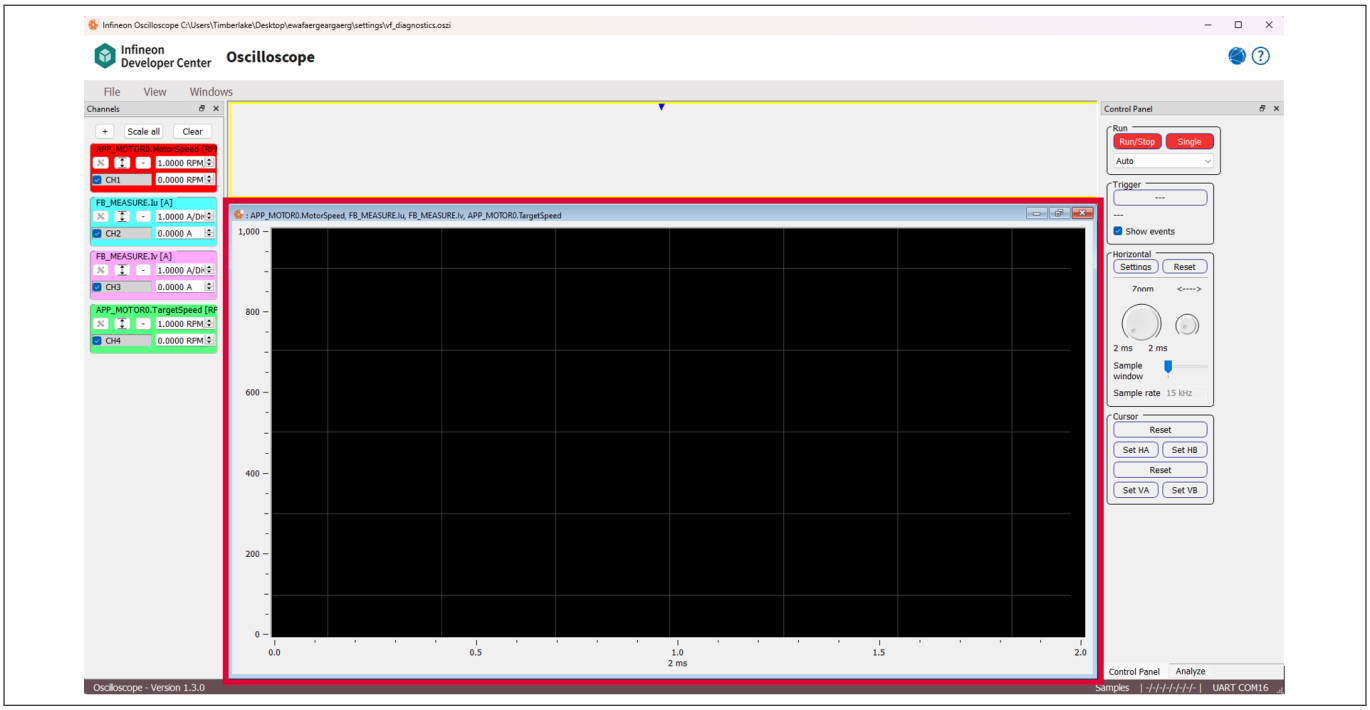

Figure 60 Oscilloscope with Panel Added

Users can remove a panel from the oscilloscope project section by clicking the exit **(X)** button in the upper-right corner of the panel.

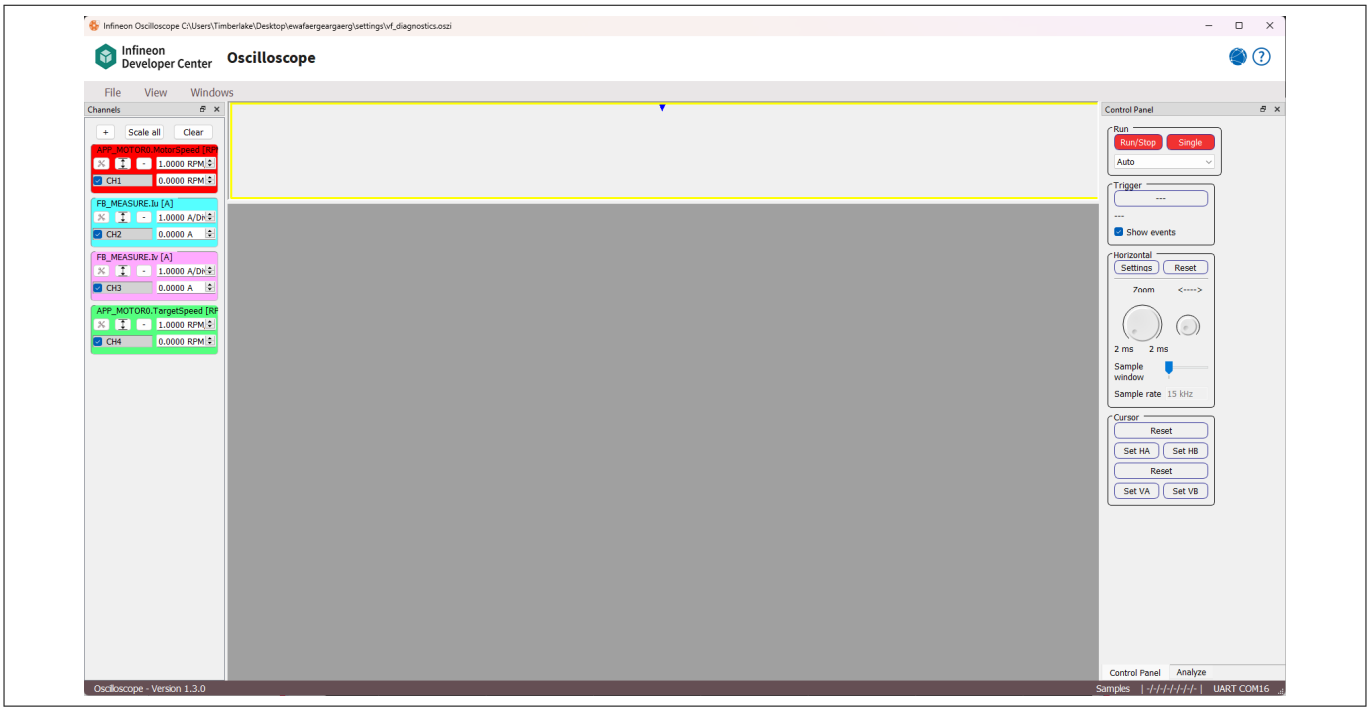

Figure 61 Oscilloscope with Panel Removed

8.5 Channel Setup

Channels are used to monitor a specific signal. Users can adjust channels with the **Add Channel**, **Scale All**, and **Clear** buttons.

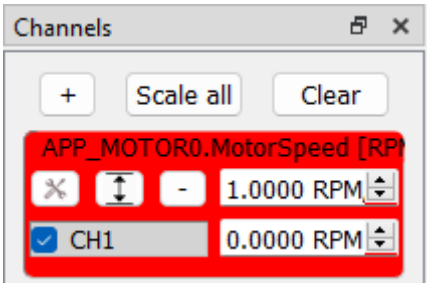

Figure 62 Channels Section

- + - Add Channel is used to create a channel to monitor a signal.
- Scale all - Scale All is used to fit the signal chart in the plotting area.
- Clear - Clear is used to remove the plots of data in all channels.

8.5.1 Channel Components

Within the individual channel tabs are several functions users can use to configure the individual channels.

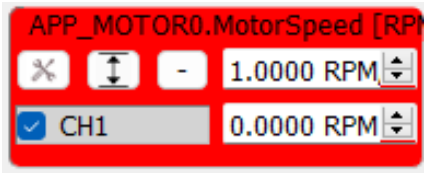

Figure 63 Channel Components

- - **Signal Settings** is used to configure various properties within the channel (Label, Signal, Color, etc).
- - **Autoscale** will fit the signal to the plotting area automatically.
- - **Remove Channel** will delete the associated channel.
- CH1 - This button will allow users to enable or disable the plotting of the channel.
- 1.0000 RPM - **Gain Adjustment** is the textbox in the channel, and **Offset Adjustment** is the second.

8.5.2 Adding a Channel

To add a channel, users can click on the **Add Signal** + button. Users can add up to eight channels and signals, with channels able to monitor one signal at a time.

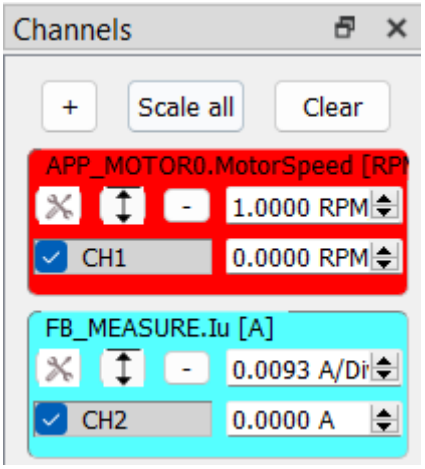

Figure 64 Multiple Channels within the Channels Section

Newly created channels will be colored dark gray and will lack an associated signal.

8.5.3 Removing a Channel

Users can remove a channel by clicking on the **Remove Channel** - button. A pop-up window will appear confirming the user's selection. Users can click **Yes** to remove the channel, or **No** to cancel the operation and keep the channel.

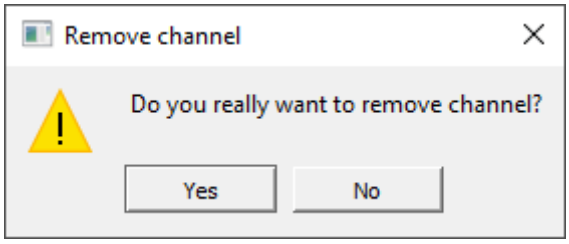

Figure 65 Remove Channel Confirmation Window

8.6 Signal

8.6.1 Adding a Signal

After adding a new channel, users can click the **Add Signal** button to open the channel configuration pop-up window.

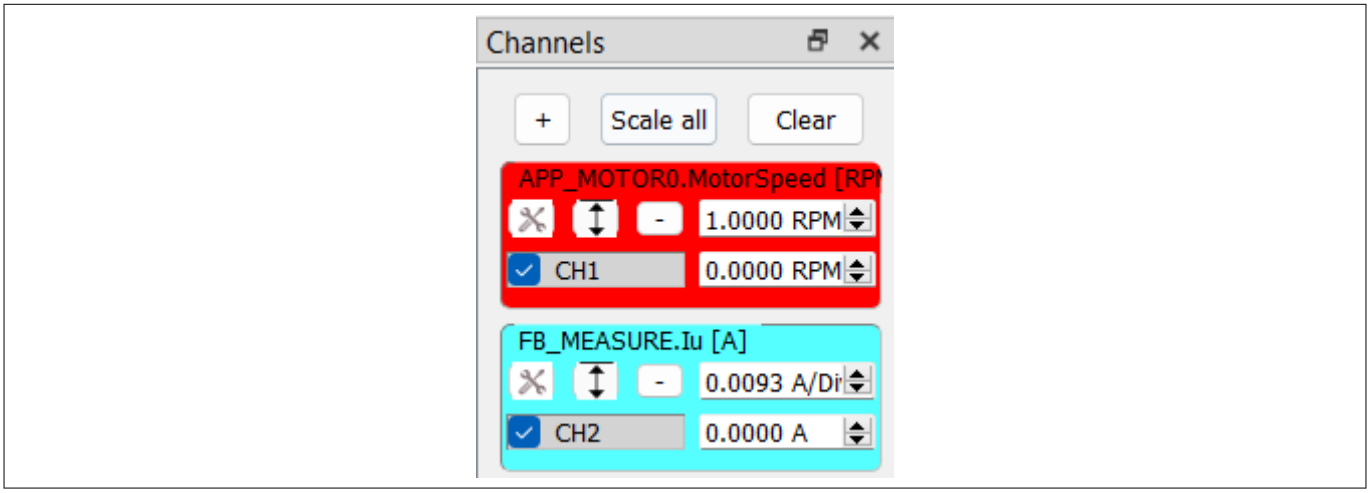

Figure 66 Channels Section

The channel configuration window allows users to edit the associated signal, channel color, Acquisition Mode, Line Style, and Line Width.

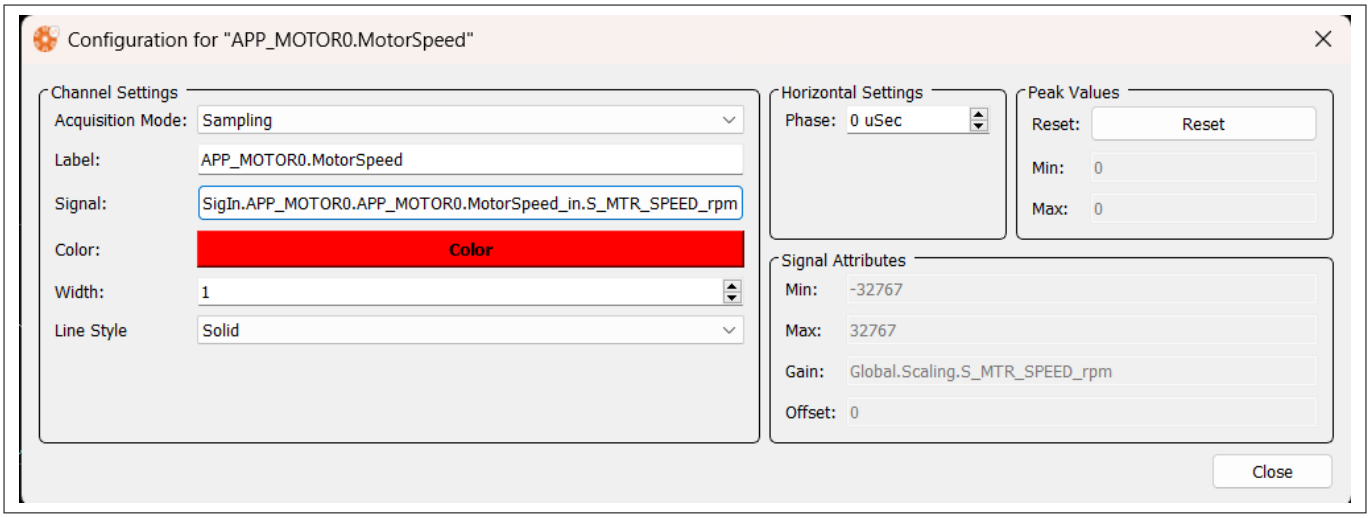

Figure 67 Channel Configuration Window

Users can click the **Signal** button to display the list of supported signals. After navigating and expanding the lists as needed, users can select a signal to monitor and confirm their selection with the **Ok** button.

Clicking on the **Color** button will allow users to choose the display color of the associated channel. Users can choose from the basic selection of colors, or create their own to better suit their project. Once finished, users can confirm their selection by clicking **Close**.

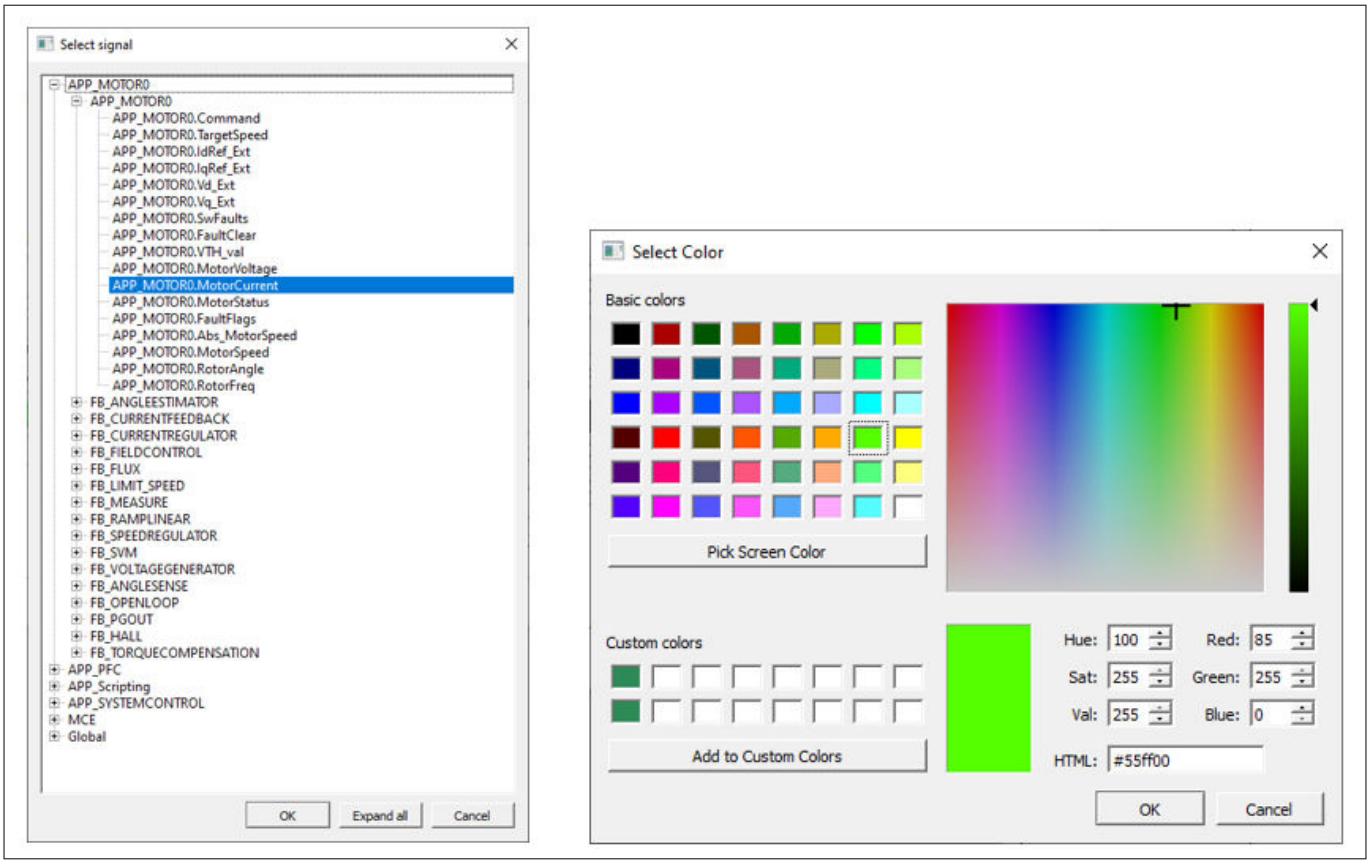

Figure 68 Signal Selection and Color Selection

8.6.2 Removing a Signal

Before users remove a signal, the associated channel must also be removed. If a user wishes to remove a signal without removing the channel, the channel must be excluded by unchecking the **CH#** checkbox.

8.6.3 Changing a Signal

Users can change the signal using the Add Signal configuration window and Signal button within the configuration window. The oscilloscope must not be in capture mode when changing the signal.

8.7 Control Panel

The control panel allows users to quickly change the active function settings of the oscilloscope.

8.7.1 Run Control

The **Run** function allows users to change the trigger mode for the oscilloscope. Users can switch between **Auto** and **Normal** mode via the drop-down menu. The **Run/Stop** function of the Run function can be used in Auto and Normal mode, while the **Single** function can be run in Normal mode when required for certain triggers.

Figure 69 Normal, Auto, and Single Function Control

8.7.2 Trigger Modes

The trigger mode for the project determines how the oscilloscope captures a waveform when it does not detect a trigger.

- **Normal Mode** – In this mode the oscilloscope will only display the signal capture if it meets the trigger point. The panel will otherwise be blank as it awaits the trigger or the panel will display the last captured waveform.

  ***Note**: Users should be aware of a timeout that exists in Normal Mode. If no trigger is detected before this timeout, plotting is done regardless of the trigger.*

- **Auto Mode** – In this mode the oscilloscope will continue to capture regardless of the triggers state. This mode is useful when users are monitoring multiple signals that would require multiple trigger conditions.

8.7.3 Zoom and Sample Window

The sample window (1) is a chart that contains all existing channels to make it easier for users to see the whole stream of data currently buffered. A zoom rectangle (2) is provided to mark certain regions of interest and to get a zoomed view within this section on all panels (3) with connected time axis.

In case of data acquisition of more than a few seconds or in the range of minutes the Infineon Oscilloscope supports a rolling mode and a mechanism to scroll back in data history. Once recording is started from an empty overview window, incoming data will be filled into this area. When the amount of data reaches the right hand side of the screen all further data will shift the current content to the left. To get to points of interest from the history a scroll bar can be used for navigation.

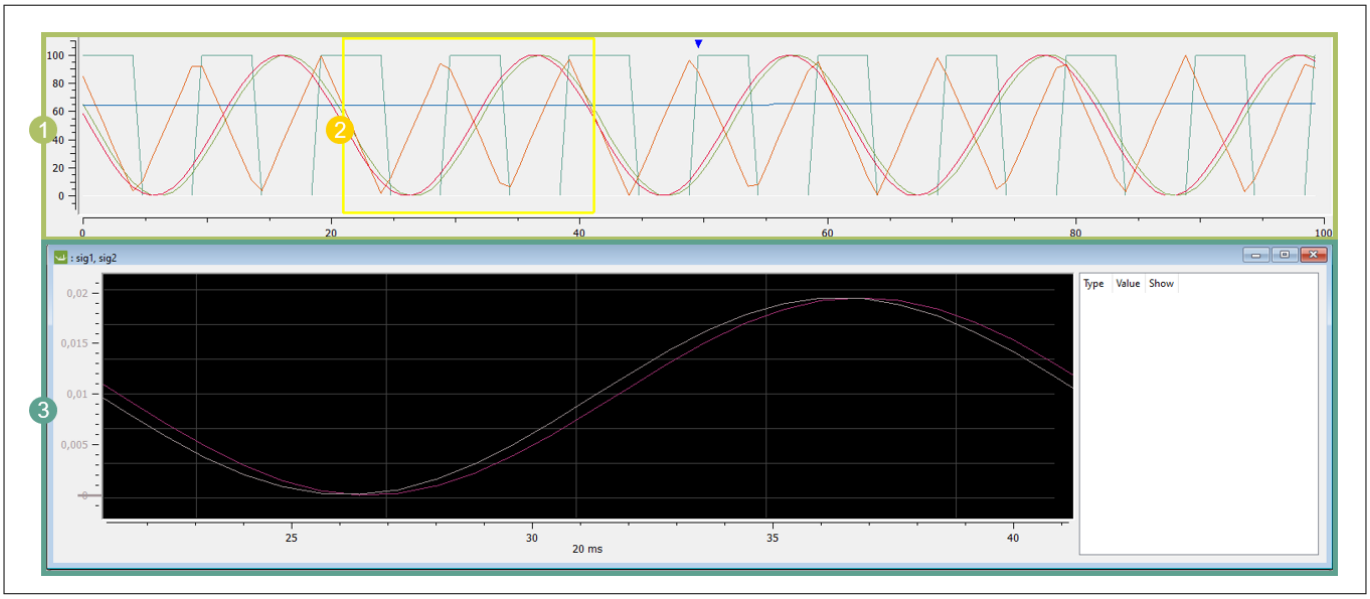

Figure 70 Zooming into Regions of Interest

There are three ways to adjust the yellow zoom rectangle: zoom, offset, and sample window size. Note that the zoom knob only changes where the panel is looking and does not change how the signals are sampled.

Zooming in and out of the sample window will change the width of the zoom rectangle. To do this the first knob on the Horizontal control panel can be turned, or the **Ctrl** key can be held while hovering the mouse over the sample window and scrolling with the scroll wheel.

Adjusting the offset will move the zoom rectangle right or left. To do this, scroll the second knob on the Horizontal control panel, or drag the yellow zoom rectangle to the right or left with the mouse.

The last adjustment, sample window, changes the duration that the buffer is filled. Adjusting the "Sample window" slider allows you to choose from the following durations for the buffer: 2ms, 5ms, 10ms, 20ms, 50ms, 100ms, 200ms, 500ms, 1s, 2s, and 5s,

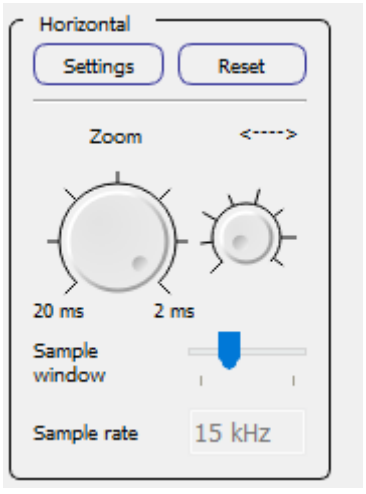

Figure 71 Horizontal Axis Adjustment Panel

8.8 Triggers

8.8.1 Trigger

The trigger control allows users to capture data based on selection conditions. Trigger settings and conditions will be shown in the Trigger Section of the Control Panel pane.

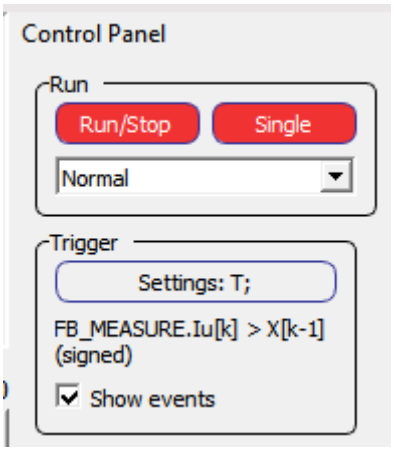

Figure 72 Trigger Section of the Control Panel

8.8.1.1 Trigger Types

The oscilloscope tool offers different types of triggers. This includes the Edge, Level, and Comparison triggers. When the trigger signal matches the project's settings, the oscilloscope will generate the appropriate trigger.

- **Edge Triggering** will have a trigger point that occurs on the rising edge (positive change) or falling edge (negative change) of the signal.
- **Level Triggering** will have a trigger point that occurs on a pre-determined level, when the input signal is equal to the selected trigger level.
- **Comparison Triggering** will have a trigger point that occurs where both signals meet the trigger condition.

8.8.1.2 Trigger Sources

The oscilloscope does not necessarily have to trigger on the signal being measured. The trigger source is a signal internally generated by MCEOS where no external signals are supported. Trigger sources supported are signals from APP MOTOR, PFC, MCE, Scripting and APP SYSTEM CONTROL.

8.8.2 Creating an Edge Trigger

To create an Edge Trigger, users will click on the Settings button located in the Trigger section of the Control Panel.

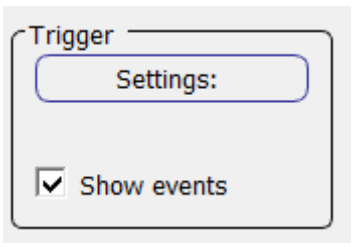

Figure 73 Trigger Section

This will open the Event Trigger Configuration pop-up window. Select the proper Edge Trigger under the Edge section of the Trigger tab. Users will then select the appropriate Signal X and Trigger Position for their project under the Trigger Event Settings. Clicking the OK button will confirm the trigger condition.

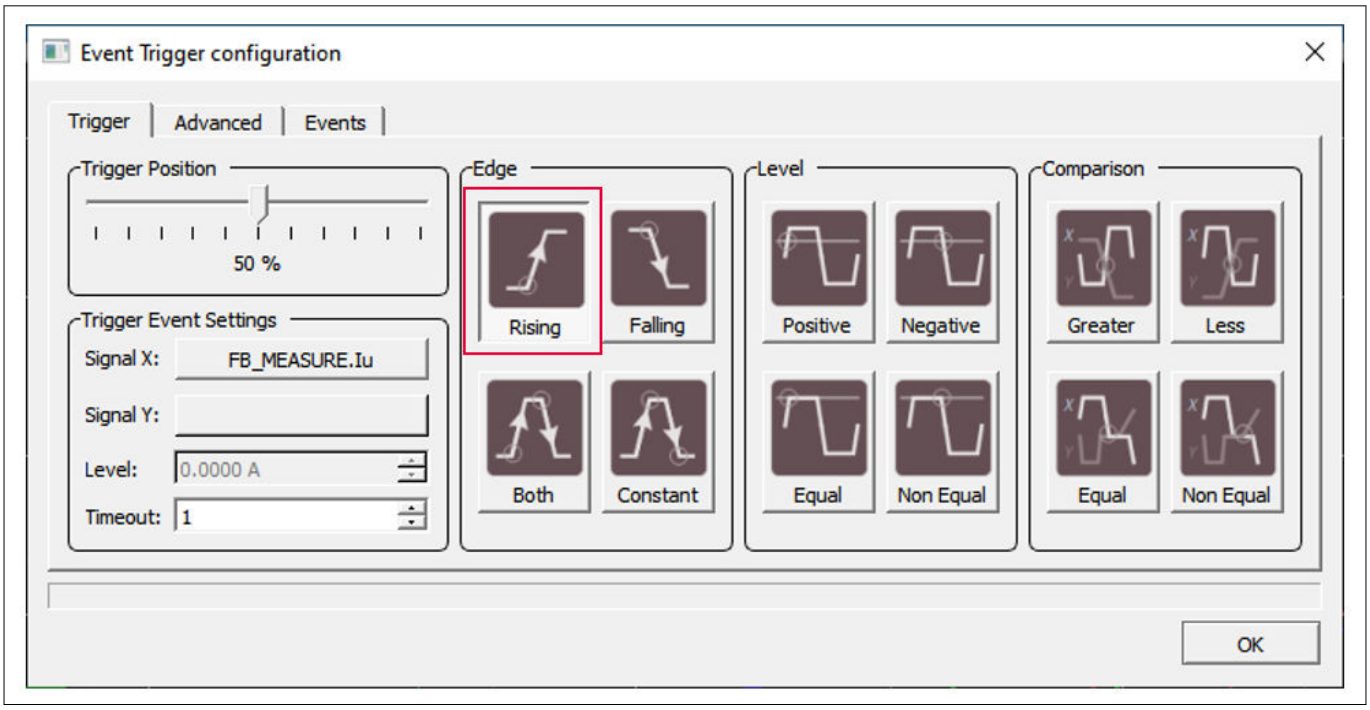

Figure 74 Event Trigger Configuration with Edge Trigger

The trigger settings and condition will be shown in the Trigger section of the Control Panel.

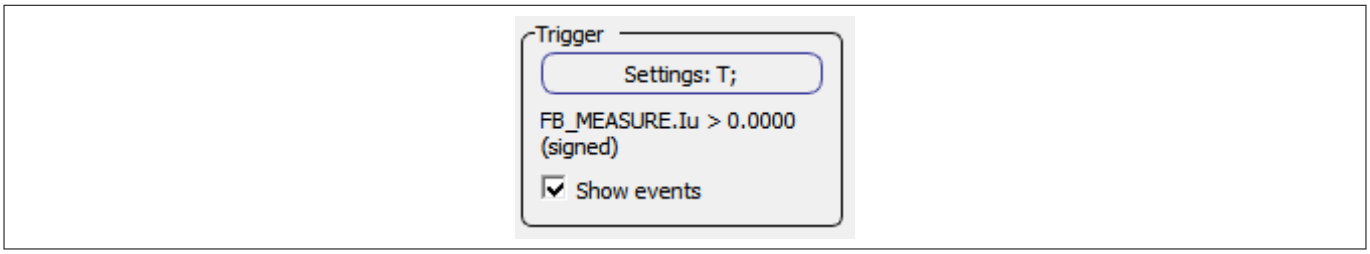

Figure 75 Trigger Section with Edge Condition

8.8.3 Creating a Level Trigger

Users will navigate to the Event Trigger Configuration window. Select the proper Level Trigger under the Level section of the Trigger tab. Users will then select the appropriate Signal X, Level Value, and Trigger Position for their project under the Trigger Event Settings. Clicking the OK button will confirm the trigger condition.

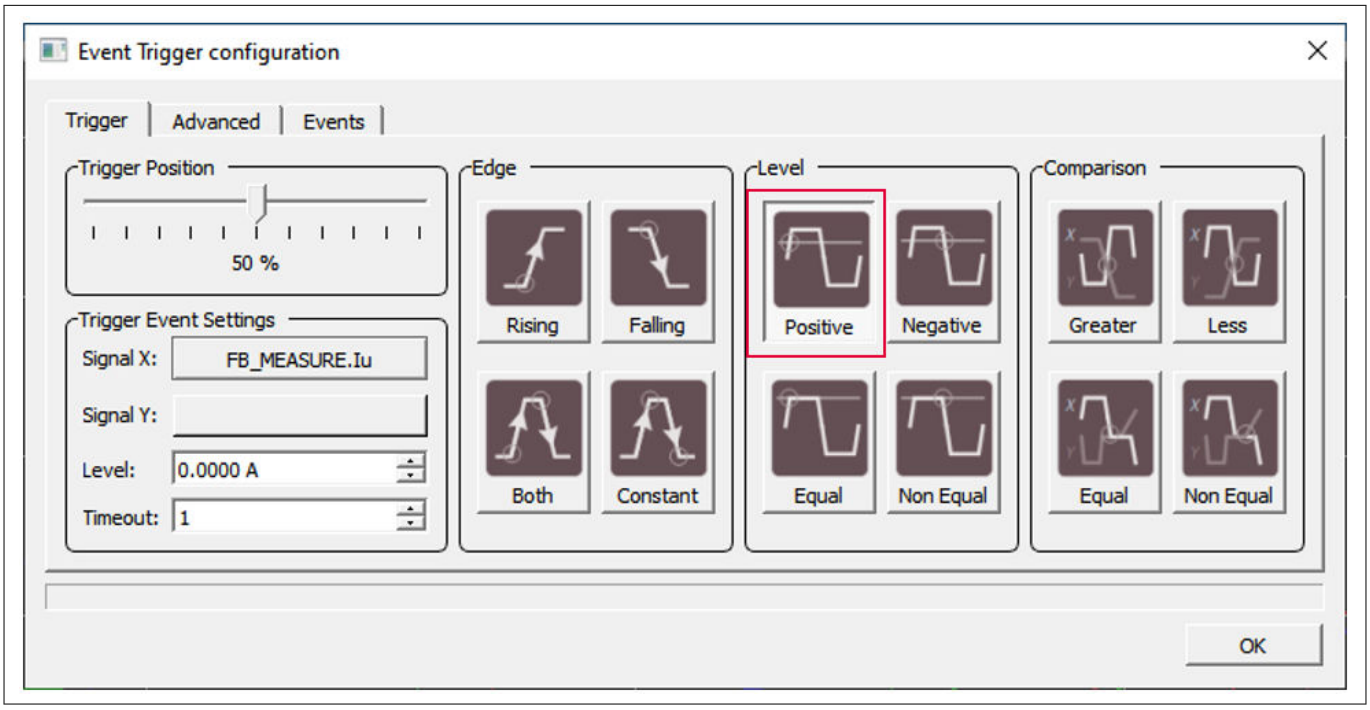

Figure 76 Event Trigger Configuration with Level Trigger

The trigger settings and condition will be shown in the Trigger section of the Control Panel.

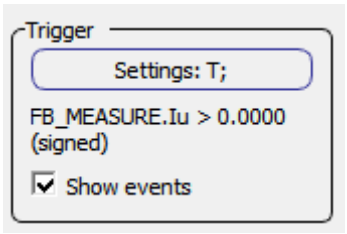

Figure 77 Trigger Section with Level Condition

8.8.4 Creating a Comparison Trigger

Users will navigate to the Event Trigger Configuration window. Select the proper Comparison Trigger under the Comparison section of the Trigger tab. Users will then select the appropriate Signal X, Signal Y, and Trigger Position for their project under the Trigger Event Settings. Clicking the OK button will confirm the trigger condition.

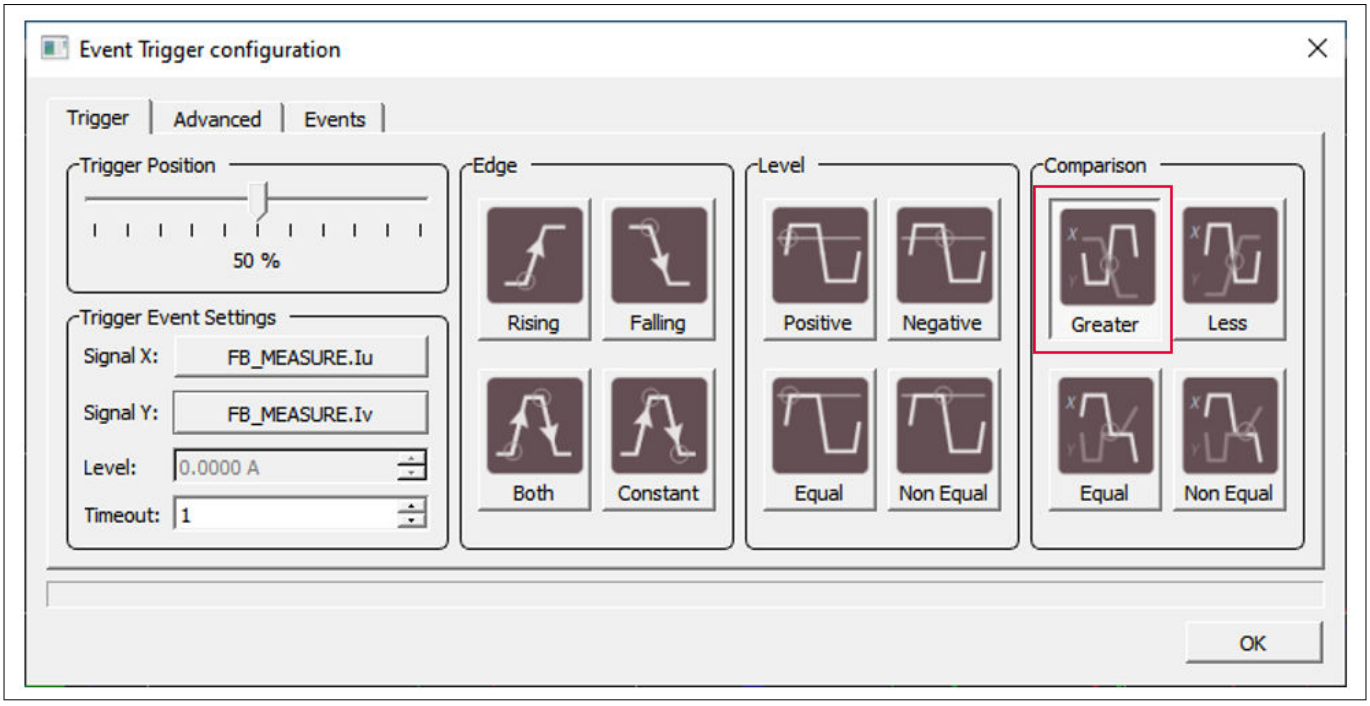

Figure 78 Event Trigger Configuration with Comparison Trigger

The trigger settings and condition will be shown in the Trigger section of the Control Panel.

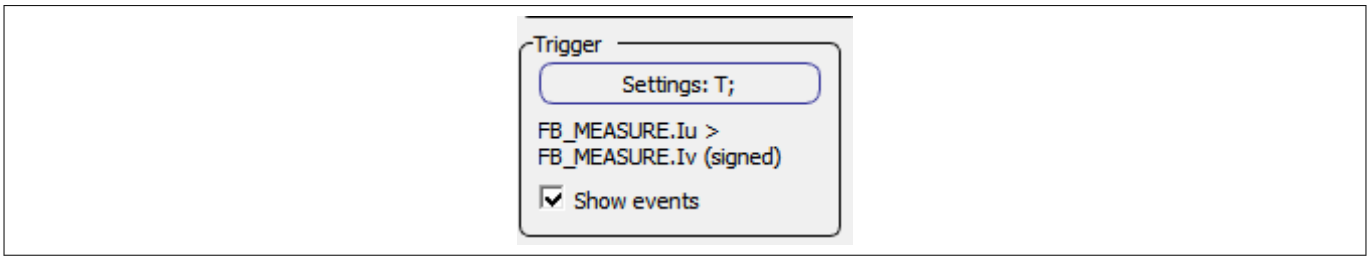

Figure 79 Trigger Section with Comparison Condition

8.8.5 Removing a Trigger

Users can remove a trigger that is capturing via Auto Mode.

8.8.6 Advanced Trigger Configuration

The Advanced tab of the Event Trigger Configuration allows users to change the Event Selection, Stream Event Selection, and the Events Clock Divider. When an event is enabled in either of the Selection sections, users should activate the associated event in the Events tab.

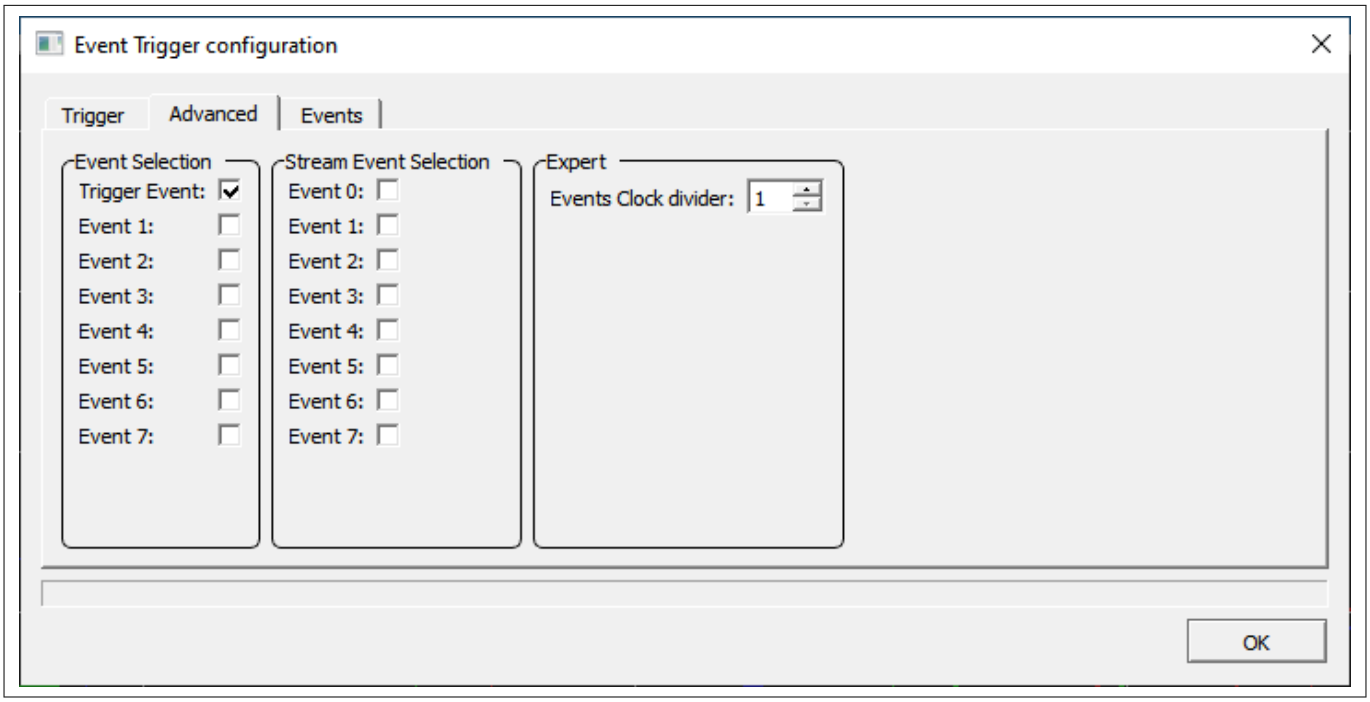

Figure 80 Advanced Tab of Event Trigger Configuration

8.9 Events

Users can create events in the Events tab of the Event Trigger Configuration pop-up window. Edge, Level, and Comparison events are supported, and each category follows the same rules as its trigger counterpart. Users can create up to 7 events.

***Note**: Users should know that every created and enabled event increases the CPU load by an average of 2%. For example: if 4 events are enabled, the CPU load will be increased by 8%.*

8.9.1 Creating an Event

Users can create an event using the following steps:

1. Start by setting the Event Number (Event Nr). This acts as the events identifier.
2. Check Show Event.
3. Change the Label to the appropriate setting.
4. Select the corresponding Signal X and Signal Y. The appropriate signal will depend on the type of event initially chosen (Edge, Level, or Comparison).
5. If it is required by the type of event, set the Level to the appropriate number.

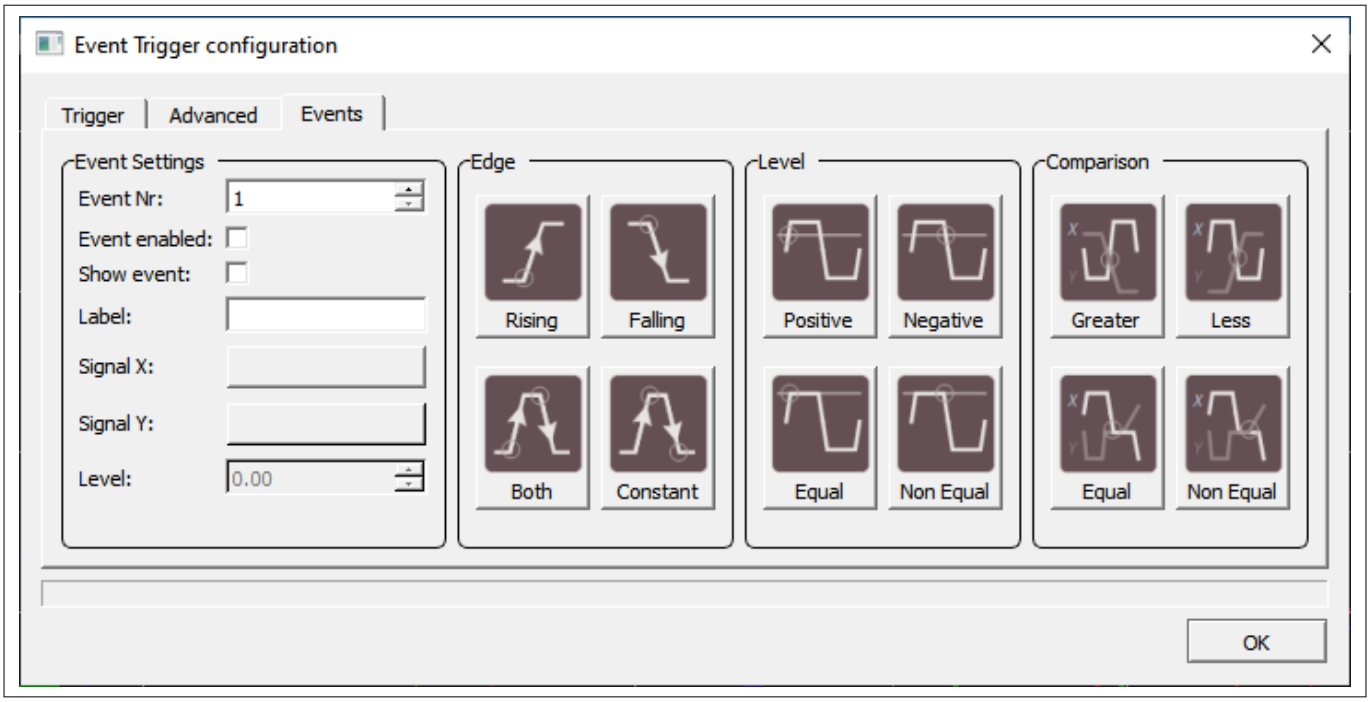

Figure 81 Events Tab of Event Trigger Configuration

8.9.2 Removing an Event

To remove an event, users can uncheck the Event Enabled option from both the Events tab and the Advanced tab.

8.10 Analyze Functionality

Cursors, triggers, and measurement functionalities support the user to analyze the acquired data stream, to find the data of interest and to perform measurements on the signals are provided.

8.10.1 Cursors

The first type of available markers are the two horizontal and the two vertical cursors that can be placed on every panel's chart individually. They help to mark certain points of interest and to highlight their values in the *t* and *y* domain as well as to span areas of the chart in between areas of analysis.

Placing a cursor requires a click on the corresponding button from the control area (1) of the oscilloscope after selecting the relevant panel where this cursor must be placed. Moving the cursor along the signals leads to snapping behavior to precisely catch the required values. Then the target position on the chart is marked with a mouse click to fix the cursor (2). The cursor will always show the measured values at this point for each panel contained signal.

Using the right click on the chart and opening the **Settings** dialog snapping can be activated/deactivated for every cursor independently.

Placing a cursor can then be done in different ways

- Drag and drop the cursor via mouse onto the surface of the chart
- Double right click on the mouse and
  - Use the mouse wheel when the mouse is over the appearing spinbox (different increments can be used when simultaneously clicking the ctrl key) or
  - Typing in a required position into the spinbox using the num-pad

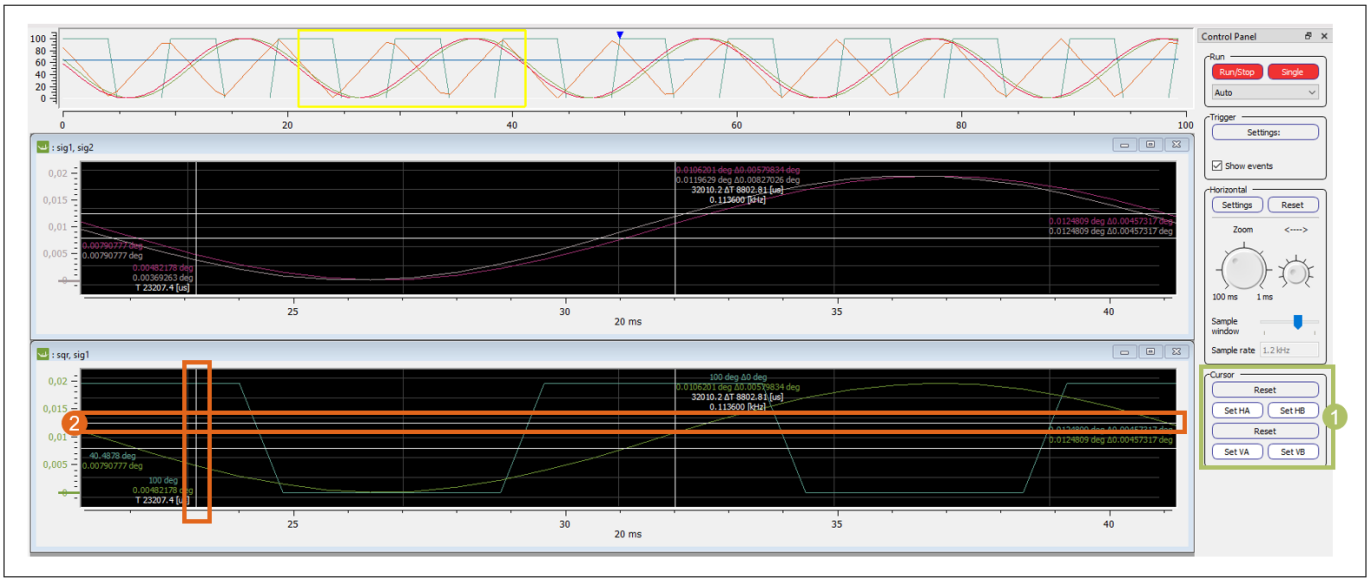

Figure 82 Working with Cursors

Once the cursor is placed it can be adjusted per drag-and-drop operation with the mouse.

The **Couple Cursors** function is activated from the panel context menu if the cursor placement is needed over several different panels. All panels enabled for this option are synchronized when moving the cursors from one of them.

The **Lock Cursors** option is selected from the panel context menu to move cursors while keeping the distance between them constant. This helps to move cursors, yet maintaining constant distance.

8.10.2 Measurement

The Infineon Oscilloscope offers further measurement functionalities in the investigation of acquired signals. Therefore a set of analyze functions can be selected by users to calculate certain values for each panel independently.

Each function may have different dependencies on the cursors to be executed correctly. Some measurement is based on the position of one cursor, other measurements need two cursors to execute actions between them.

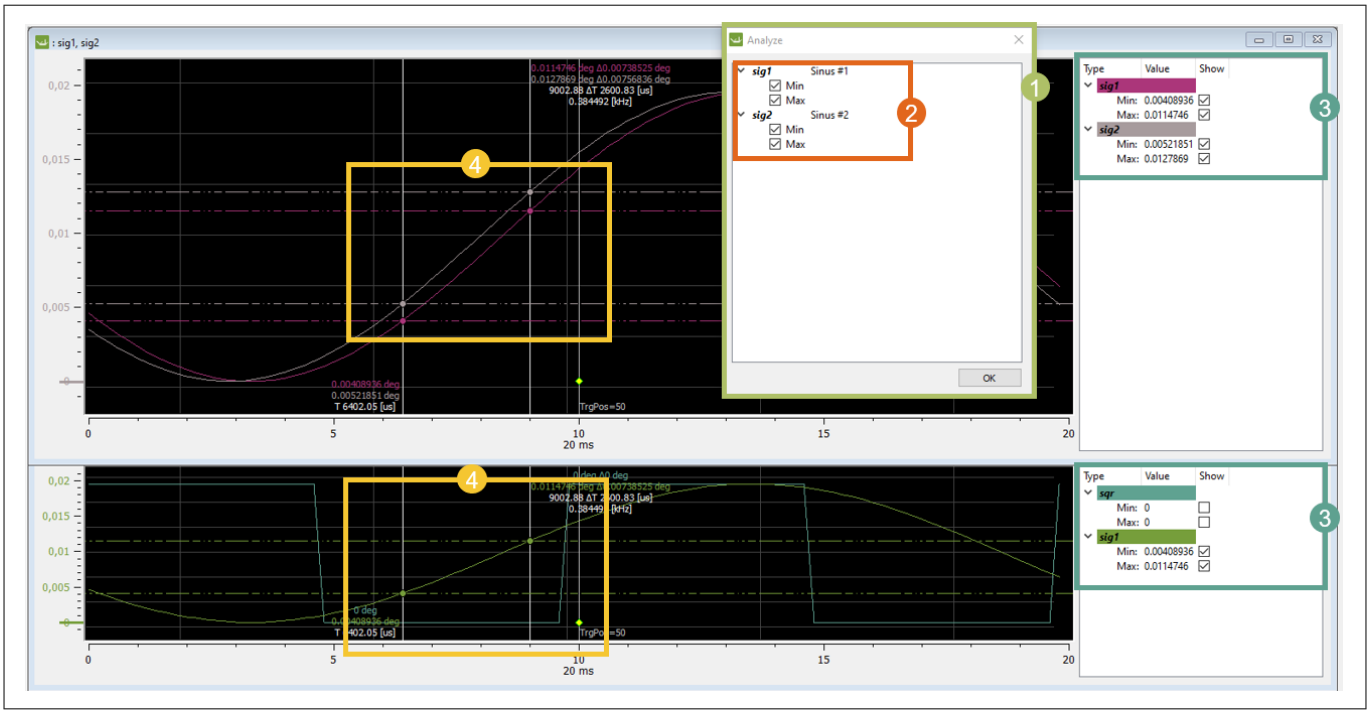

Figure 83 Using Analyze Functions

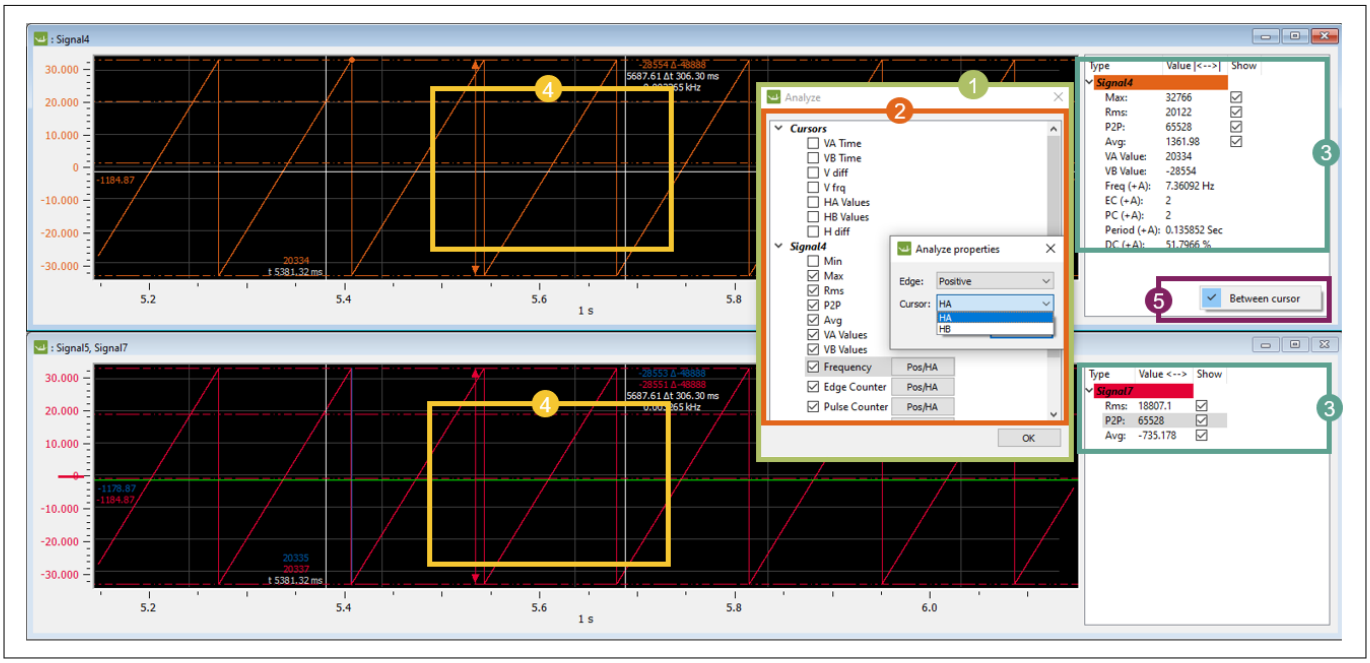

Figure 84 Using Analyze Functions

The **Analyze** dialog (1) is selected by means of a right click on the desired panel's chart to select the cursor. Every channel that is contained in this panel will be provided with its available measurement functions (2). Setting the corresponding check box will add it to the current measurement.

Two modes of representation are supported by the Oscilloscope. These are

1. A tabular view (3): every selected measurement function is displayed in a table right next to the chart of every panel with the result of its analysis
2. A graphical view (4): every measurement function is supported by individual markers to show and mark certain results of analysis on the chart. The visibility of these markers can be adjusted by a **Checkbox** in the tabular view

Functions that need only one cursor are

- **Time (VA Time, VB Time)**: Measures the position of each vertical cursor in time domain
- **Value (HA Values, HB Values)**: Measure the position each horizontal cursor
- **Value on VA/VB (VA Values, VB Values)**: Measures the value of a signal at a vertical cursor position

Functions that need two or more cursors

- **Horizontal difference (V diff)**: Measures the horizontal difference between vertical cursors
- **Frequency (V frq)**: Measures the frequency in which the difference between the vertical cursors appear
- **Vertical difference (H diff)**: Measures the vertical difference between horizontal cursors
- **Minimum (Min)**: Measures the minimum of a signal in a defined range
- **Maximum (Max)**: Measures the maximum of a signal in a defined range
- **Root mean square (RMS)**: Measures the root mean square value of a signal in a defined range
- **Peak-to-Peak (P2P)**: Measures the difference between minimum and maximum of a signal in a defined range
- **Average (AVG)**: Measures the average value of a signal in a defined range
- **Frequency (FRQ)**: Measures the frequency of a signal in a defined range along a selected horizontal cursor
- **Edge counter (EC)**: Measures the number of crossings of a signal in a defined range along a selected horizontal cursor
- **Pulse counter (PC)**: Measures the number of pulses of a signal in a defined range along a selected horizontal cursor
- **Period (PER)**: Measures the length of the period of a signal in a defined range along a selected horizontal cursor
- **Duty cycle (DC)**: Measures the duty cycle of a signal in a defined range along a selected horizontal cursor

***Note**:* *Via right click on the measurement table the range for measurement can be defined as the whole visible range within the current panel's chart or the area spanned by the two vertical cursors (5). Functions that need a horizontal cursor for measurement provide a configuration dialog to select the required cursor and to select between positive and negative interpretation of the corresponding measurement function (2).*

8.11 Oscilloscope Project Files

Users can find the oscilloscope project files inside the **Settings** folder of the project directory. The files are in .ozsi format and contain the channel, signal, and trigger settings, as well as other valuable project information.

8.11.1 Creating an Oscilloscope Project File

To create a new oscilloscope file, users will choose the **New** option from the File menu.

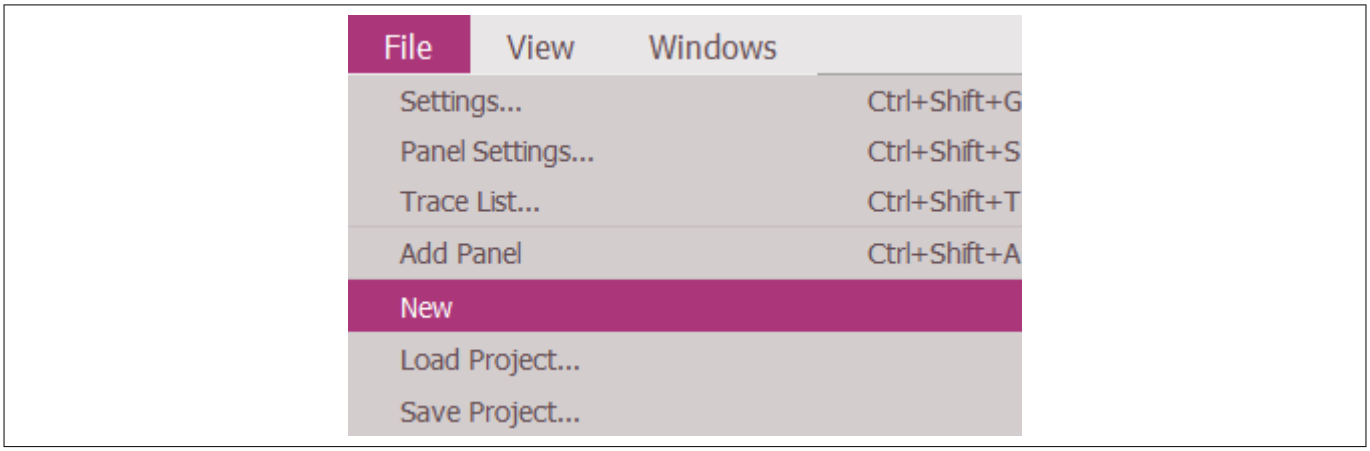

Figure 85 File Menu for Oscilloscope Project Creation

The tool will create a blank oscilloscope project. Users can continue generating the project by creating a panel (**6.1**), creating channels (**6.2**), and creating signals for monitoring (**6.3**).

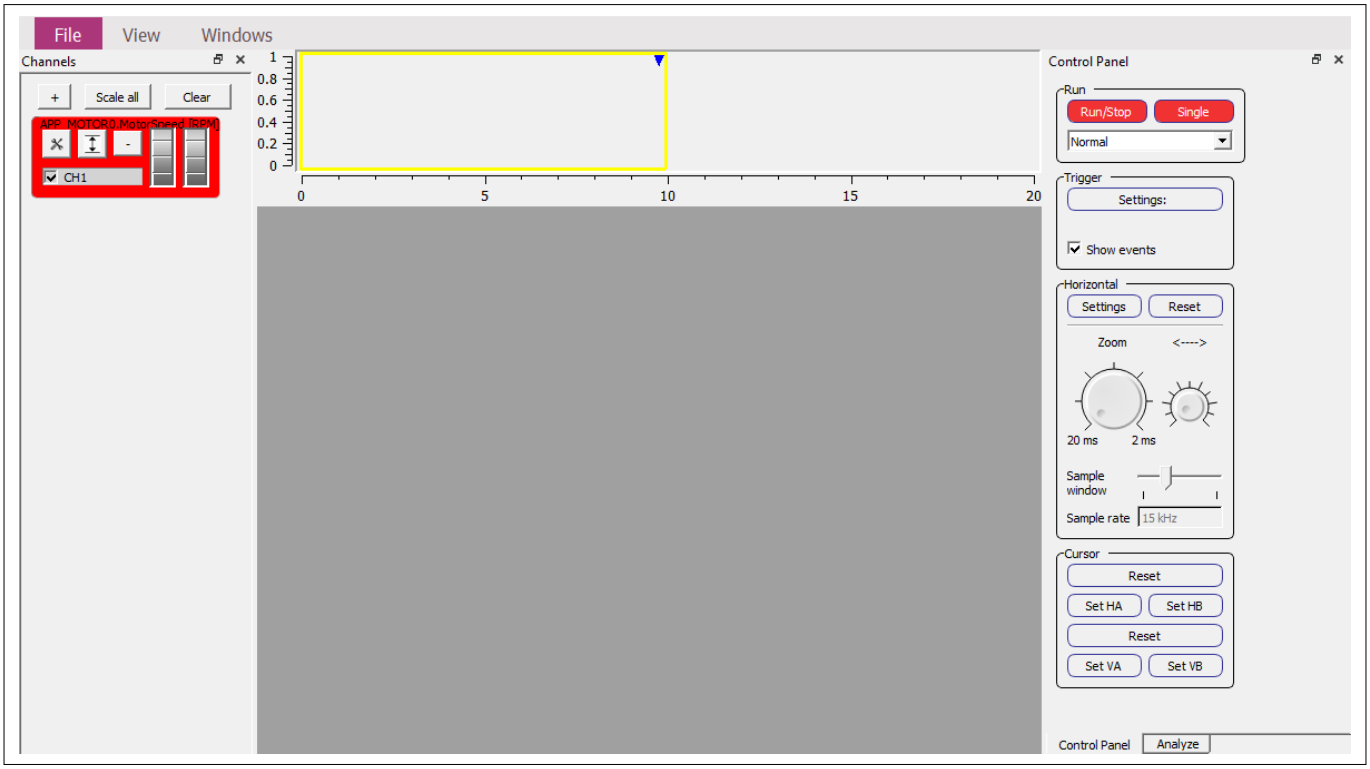

Figure 86 Blank Oscilloscope Project

8.11.2 Loading an Oscilloscope Project File

If users want to load an oscilloscope project file, they can do so by choosing the **Load Project** option from the File menu.

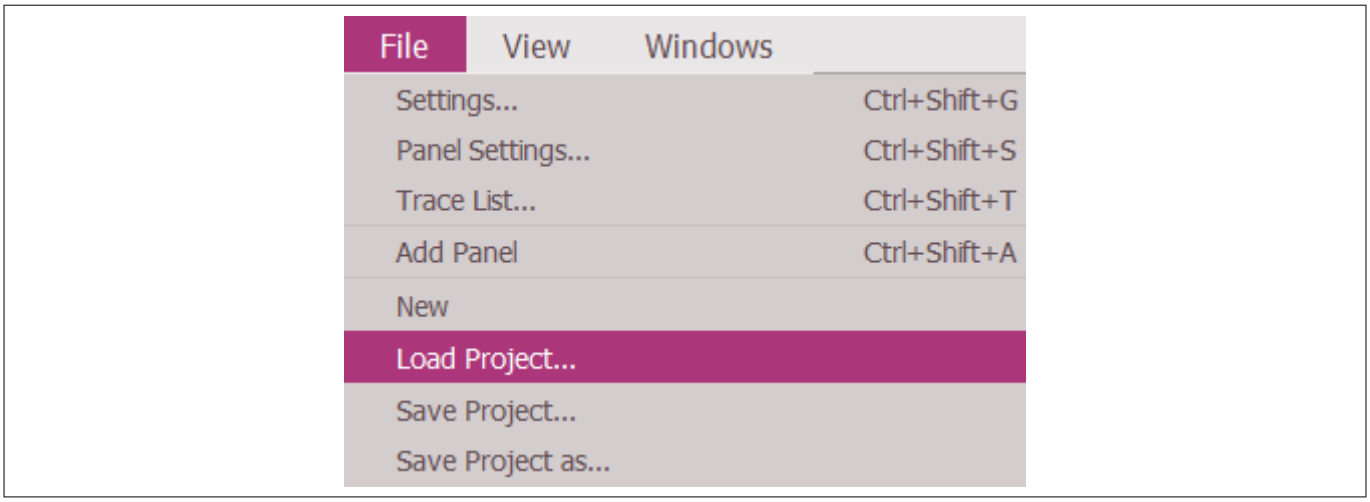

Figure 87 File Menu for Oscilloscope Project Loading

9 Script Editor

9.1 Overview

The Script Editor is a development platform for the MCE script engine that contains a code editor with debug support. It integrates all necessary tools for MCE script code development and generation of byte code. Key features of the script editor are listed below:

- Script code editor with run time syntax and compilation check.
- Project Manager to create, import/export, and maintain script project.
- Complete list of available MCE parameters and variables for ease of use.
- Debug support to verify and optimize script code, including simple break points, variable watch window, and execution control.

9.2 Script Editor Graphical User Interface

The script editor supports two operating modes: **Editor Mode** for script code development, and **Debug Mode** for script code analysis. In Editor Mode, debug related options are disabled and in Debug Mode, editor related options are disabled.

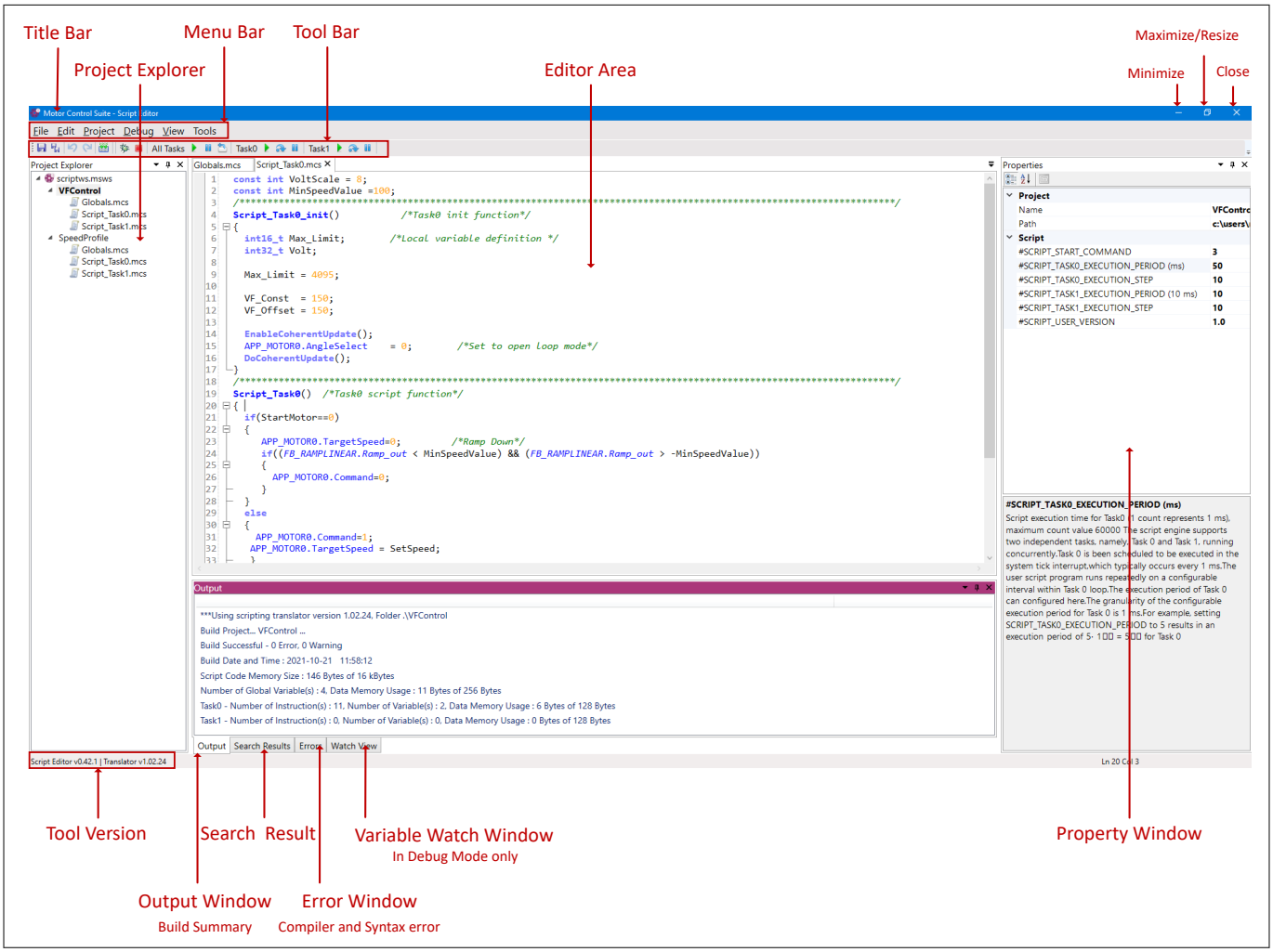

Figure 88 Script Editor Graphical User Interface

9.2.1 Project Explorer

Project Explorer is a project management window for the script editor. The editor only supports script files (*.mcs).

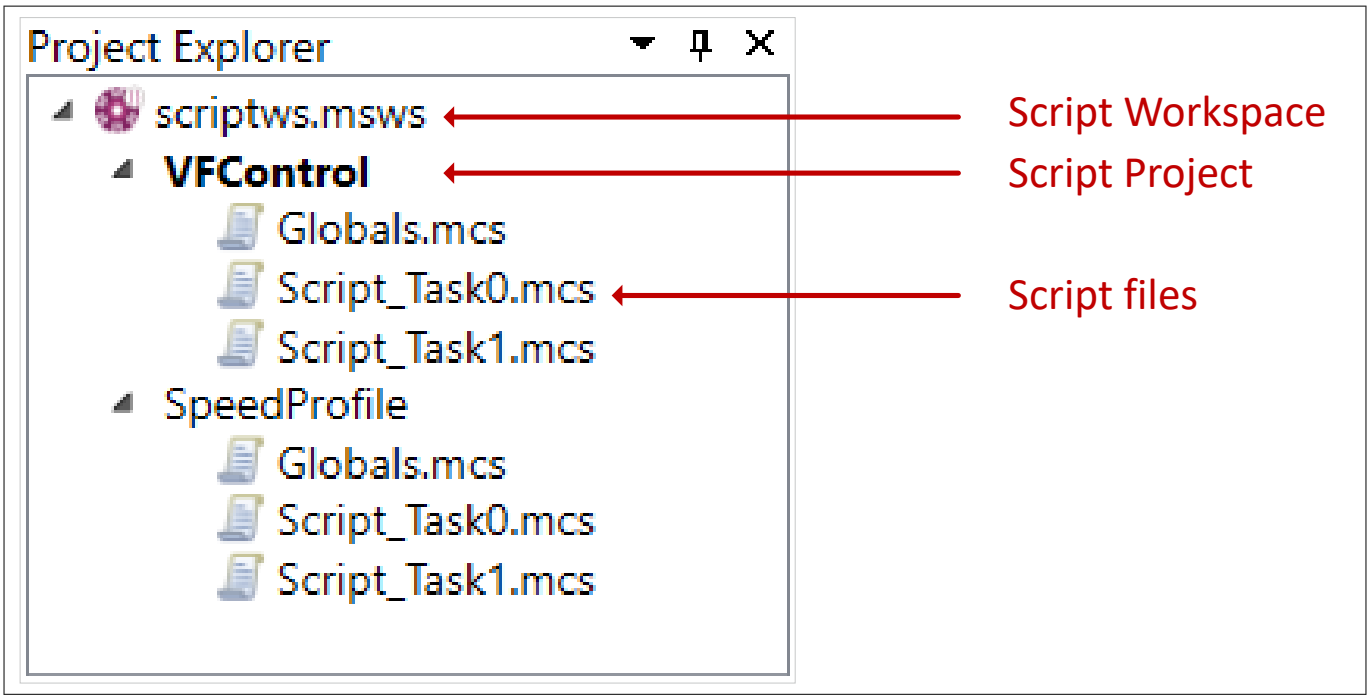

Figure 89 Script Editor: Watch View Window

9.2.2 Output Window

The **Output Window** shows the build output summary of the active project, after building/compiling the project. Build status, data memory usage, and code memory usage are displayed in this window. In case of any compilation errors, they will be displayed here.

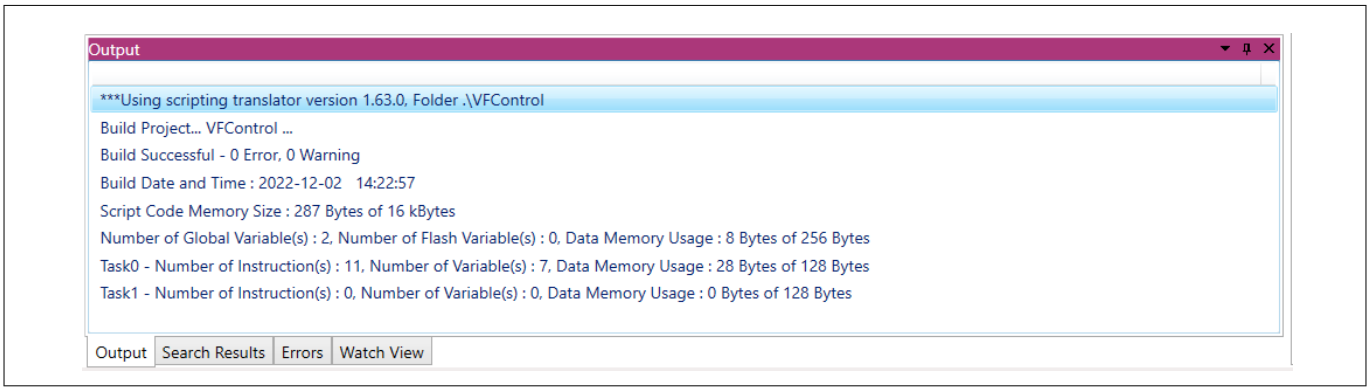

Figure 90 Script Editor: Output Window

9.2.3 Search Results

Search result (find and replace) outputs are listed in this window by file name, line number, and position.

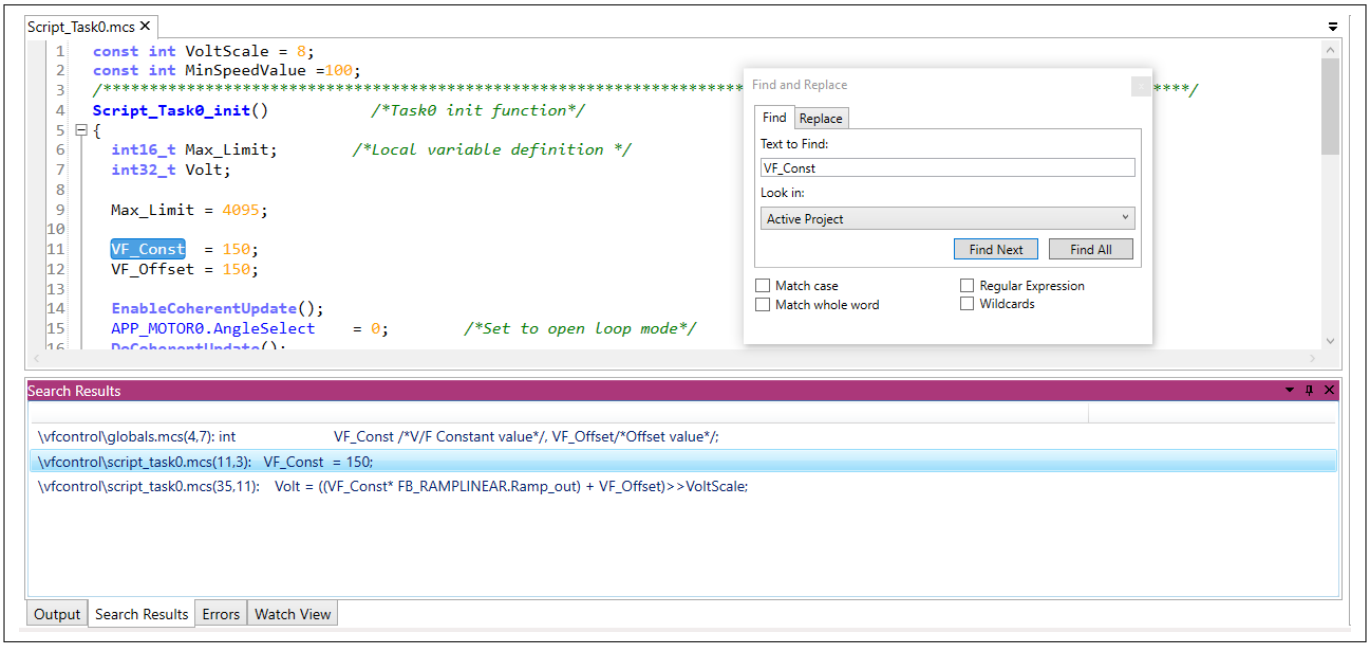

Figure 91 Script Editor: Search Result Window

9.2.4 Errors Window

Runtime syntax and compiler check is performed by the script editor. All the runtime errors are displayed in the **Errors Window** along with the file name and a short description of the error.

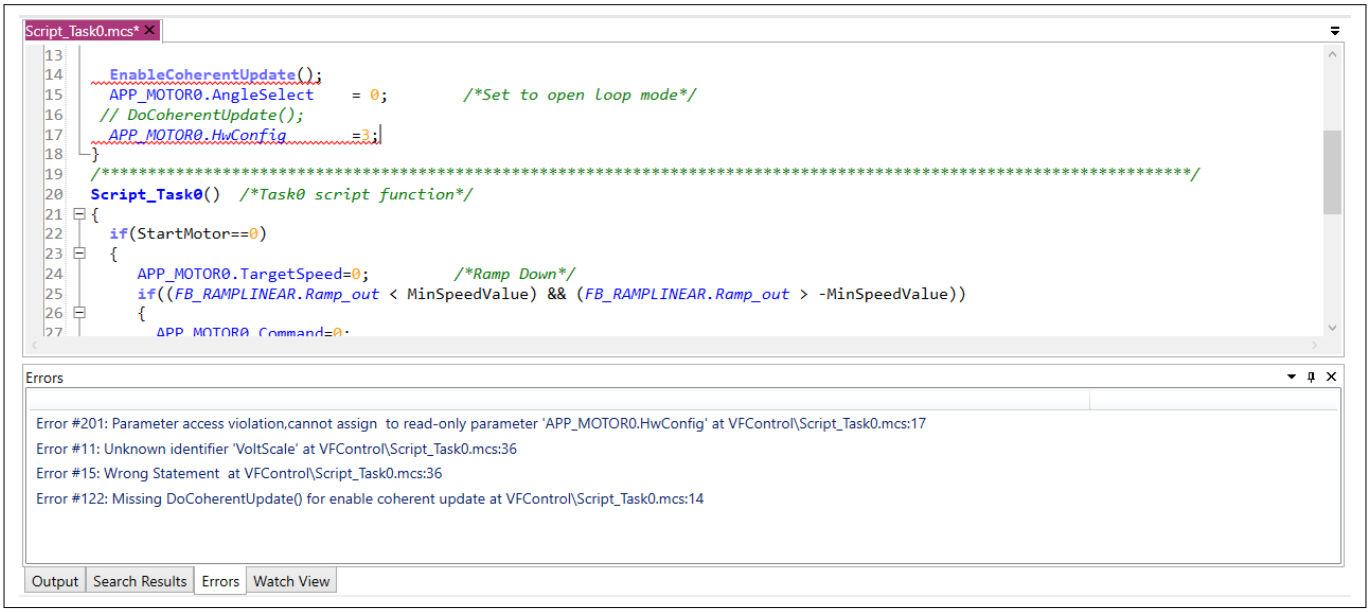

Figure 92 Script Editor: Error Window

9.2.5 Watch View

The **Watch View** window is used to read or write script variables in debug mode. It is also possible to read MCE parameters and variables using this window while in debug mode.

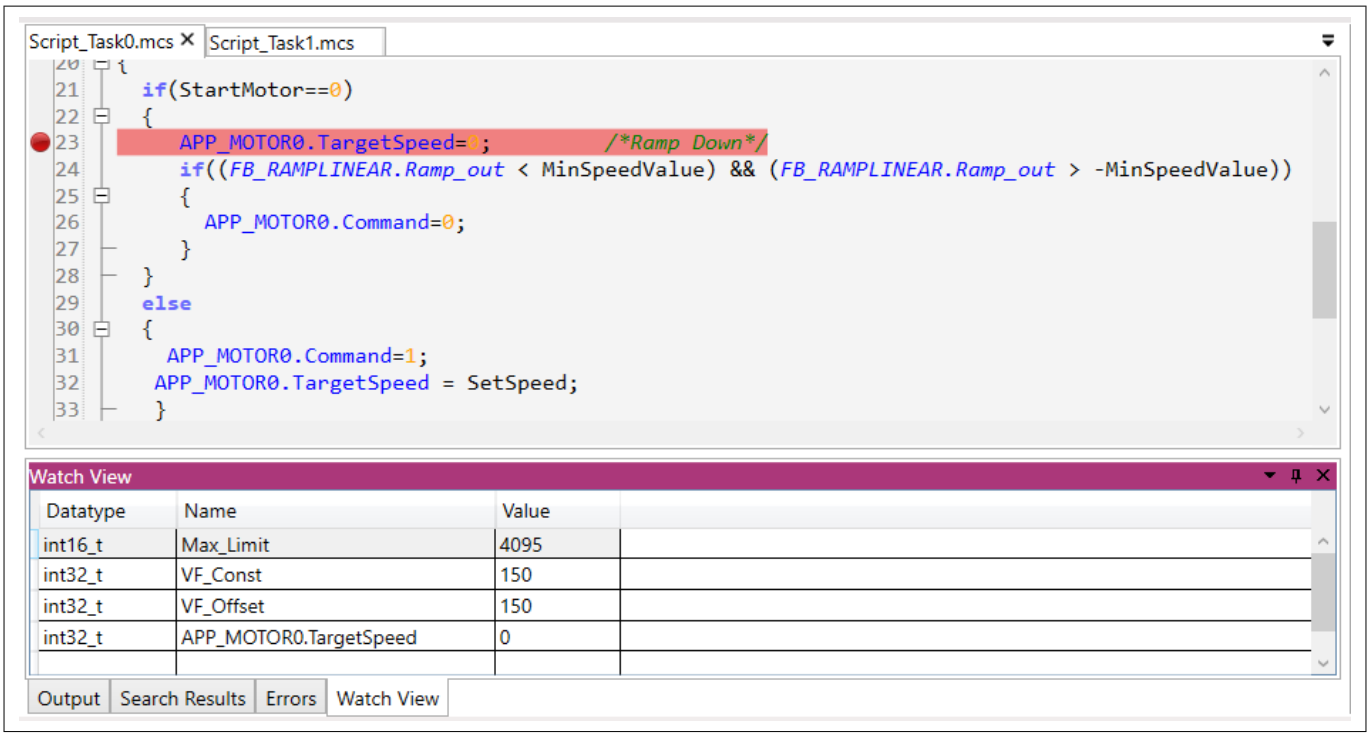

Figure 93 Script Editor: Watch View Window

Variables can be added to the watch view window by clicking the **Plus** (**+**) symbol that appears while hovering over the variable. Right clicking within the window will bring up additional options.

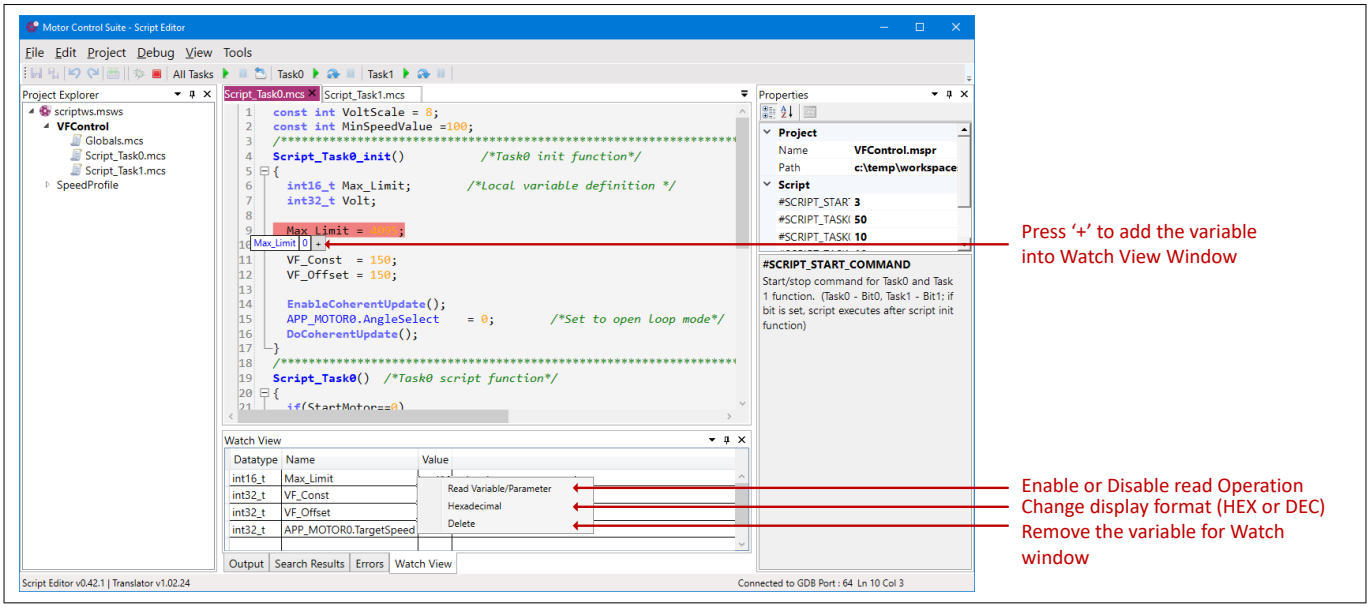

Figure 94 Script Editor: Watch View Window

9.2.6 Property Window

The property window is used to view and modify the property of selected items of the active project, Work Space, Project file, or Script file.

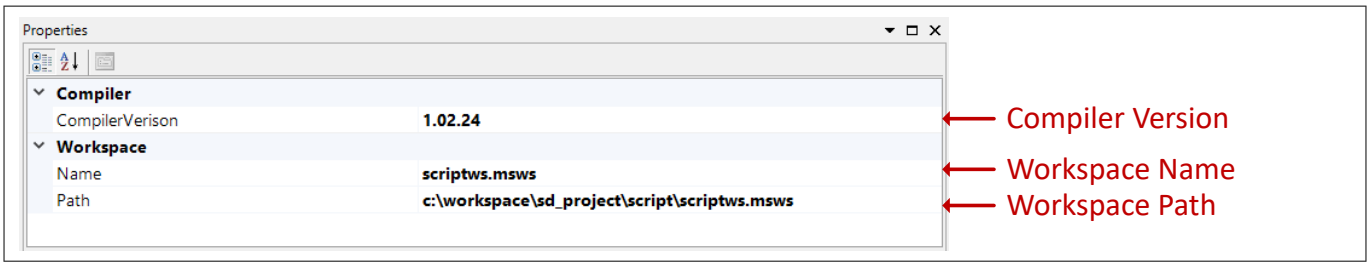

Figure 95 Script Editor: Workspace Property Window

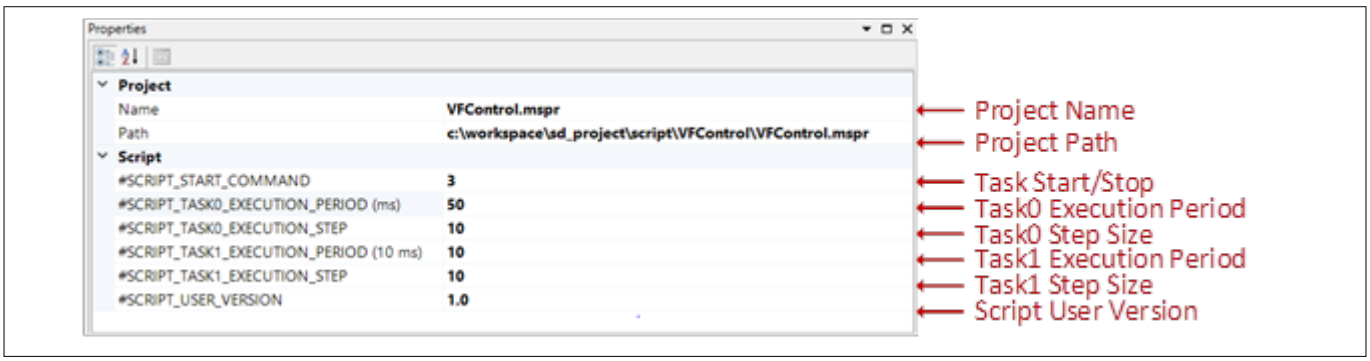

Figure 96 Script Editor: Project Property Window

#SCRIPT_START_COMMAD: Provides configuration to start or stop the script execution after script initialization. Bit0 value of this configuration controls Task0 execution and Bit1 value controls Task1 execution. If the bit value is 1, then Task function will start executing after initialization. **APP_Scripting.Command** variable can be used to start or stop script execution during normal operation.

#SCRIPT_TASK0_EXECUTION_STEP: The number of script instructions that gets executed by each Task0 during every execution step (1ms) can be configured here. Maximum value is 60000.

#SCRIPT_TASK1_EXECUTION_STEP: The number of script instructions that gets executed by each Task1 during every execution step(10ms) can be configured here. Maximum value is 60000.

#SCRIPT_TASK0_EXECUTION_PERIOD (ms): Script execution period of Task0 can be configured here (1 count represents 1ms). Maximum count value is 60000. Task0 is scheduled to be executed in the system task. The user script program runs repeatedly on a configurable interval within Task0 loop. The granularity of the configurable execution period for Task0 is 1 ms. For example, setting SCRIPT_TASK0_EXECUTION_PERIOD to 50 results in an execution period of 50 · 1 ms = 50 ms for Task0.

#SCRIPT_TASK1_EXECUTION_PERIOD (10 ms): Script execution period for Task1 can be configured here (1 count represents 10ms). Maximum count value is 60000. Task1 is scheduled to be executed in the background task. The user script program runs repeatedly on a configurable interval within Task1 loop. The granularity of the configurable execution period for Task1 is 10 ms for example, setting SCRIPT_TASK1_EXECUTION_PERIOD to 10 results in an execution period of 10 · 10 ms = 100 ms for Task 1.

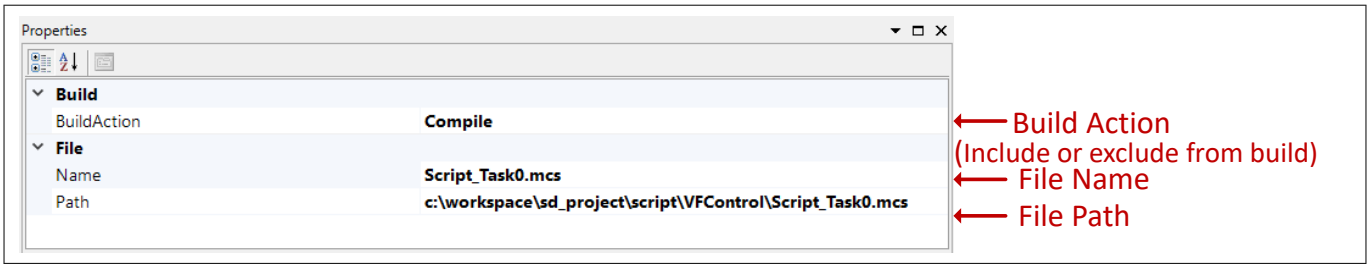

Figure 97 Script Editor: File Property Window

9.2.7 Docking Window

Most windows can be docked to another window or even floated to another screen. When a window is dragged, several docking symbols are displayed as shown in the screenshot below.

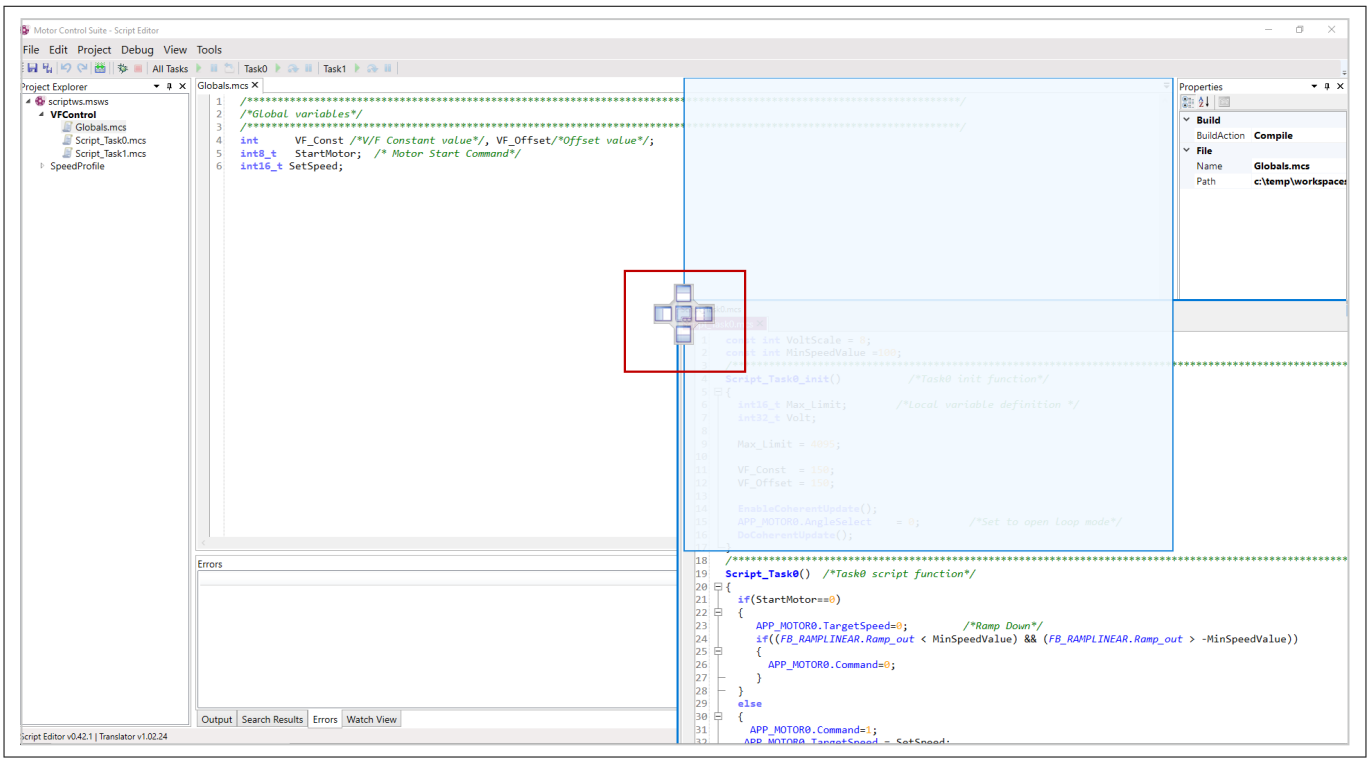

Figure 98 Script Editor: Docking window for script files

Script files are found in the Editor Window. To move a window to another location:

- Click on the Object name of a window.
- Drag the window onto a docking symbol.
- Release the mouse button.

9.3 Script Editor Menu

Menu items in the menu bar and context menu (Right Click Menu) in the Script Editor are descripted in this section.

9.3.1 Project Menu

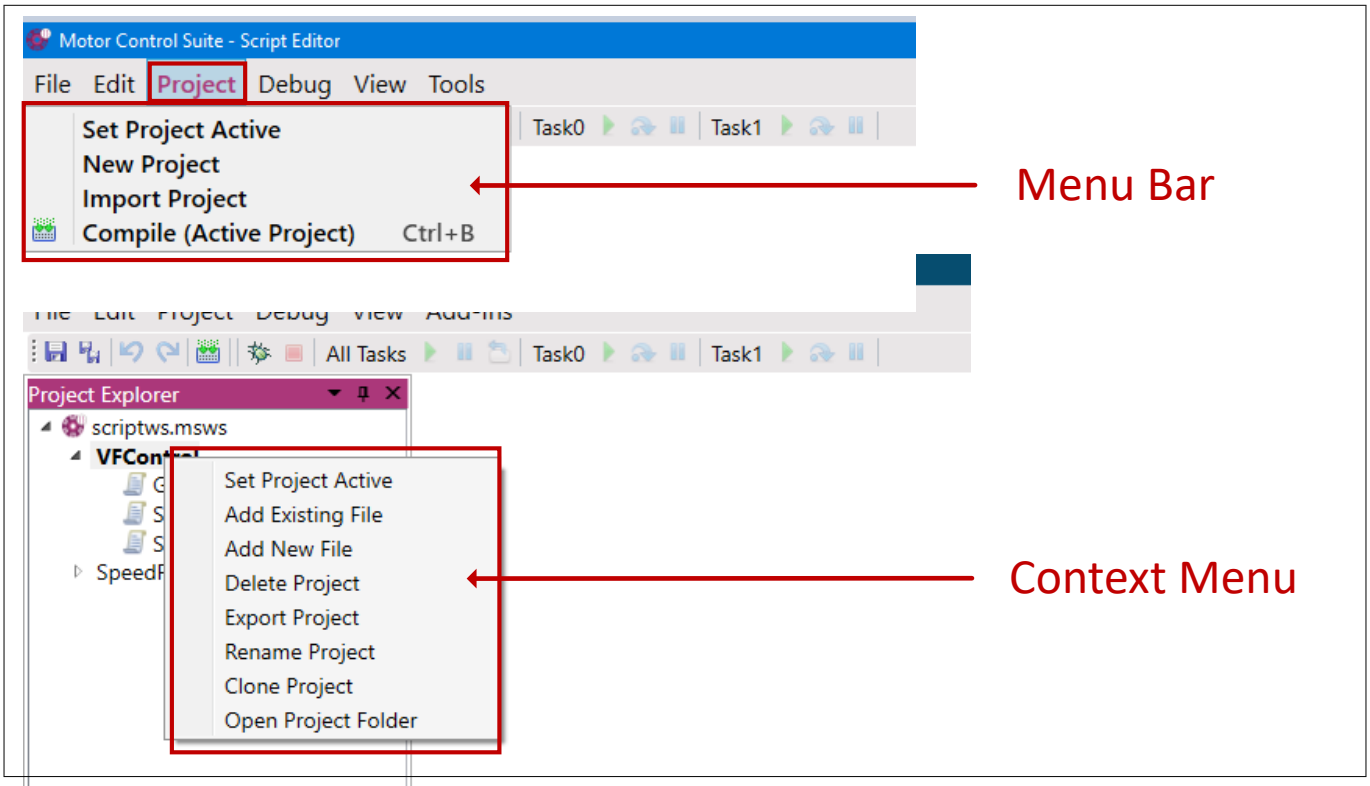

Figure 99 Project Menu Bar and Context Menu

Table 1 Script Editor: Project Menu

Menu	Shortcut	Description
Set Project Active		Set the selected project as active project
New Project		Create new project
Import Project		Import an existing project into current workspace
Compile (Active Project)	Ctrl + B	Compile the active project
Add Existing File		Add existing script file in the selected project
Add New File		Add new script file in the selected project
Delete Project		Delete the selected project
Export Project		Export the selected project
Rename Project		Rename the selected project
Clone Project		Clone the selected project
Open Project Folder		Open project folder in windows explorer

Script projects in a workspace can be exported to other users or other Solution Designer projects. Importing a script project into the current script workspace is also possible. Script projects are exported out as zip files.

***Note**: Do not rename the exported script project (<ProjectName>.zip). Renaming the zip file, will not change the project file name (<ProjectName>.mspr) inside the zip folder.*

9.3.2 File Menu

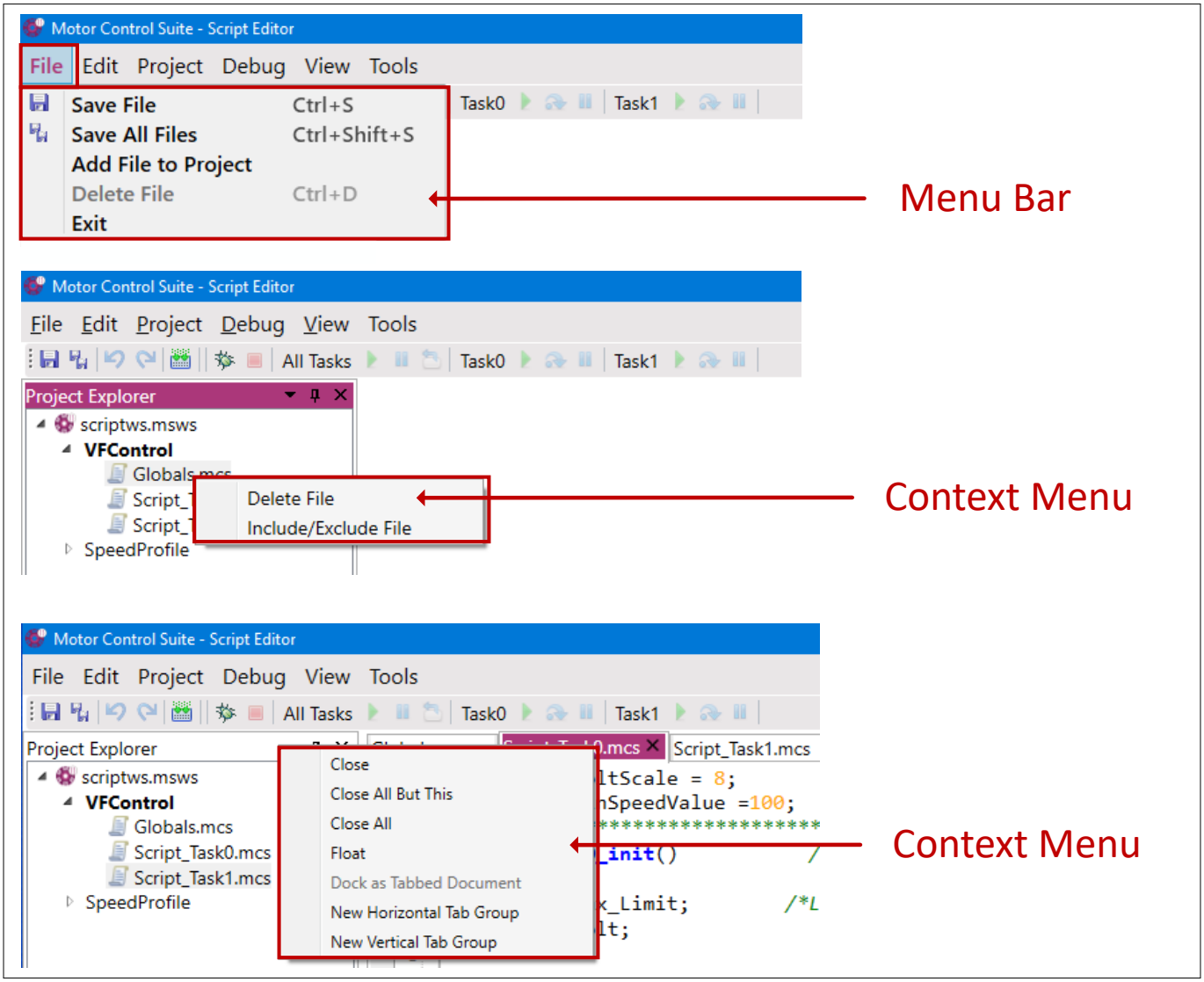

Figure 100 File Menu Bar and Context Menu(s)

Table 2 Script Editor: File Menu

Menu	Shortcut	Description
Save File	Ctrl + S	Save- the active file
Save All Files	Ctrl + Shift + S	Save- all the open file/s
Add File to Project		Add a new file into the active project
Delete File	Ctrl + D	Delete- the active file

(table continues...)

Table 2 (continued) Script Editor: File Menu

Menu	Shortcut	Description
Exit		Close Script Editor
Include/Exclude File		Include or exclude the active file for the build
Close		Close the active file
Close All But This		Close all the open file/s except the active file
Close All		Close all the open file/s
Float		Float the file, as a separate window
Dock as Tabbed Document		Dock the window/file into the default location
New Horizontal Tab Group		Move the selected file to new horizontal tab
New Vertical Tab Group		Move the selected file to new vertical tab

Ctrl + Space Bar shortcut key provides complete list of all the available MCE parameters and variables inside editor area.

9.3.3 Edit Menu

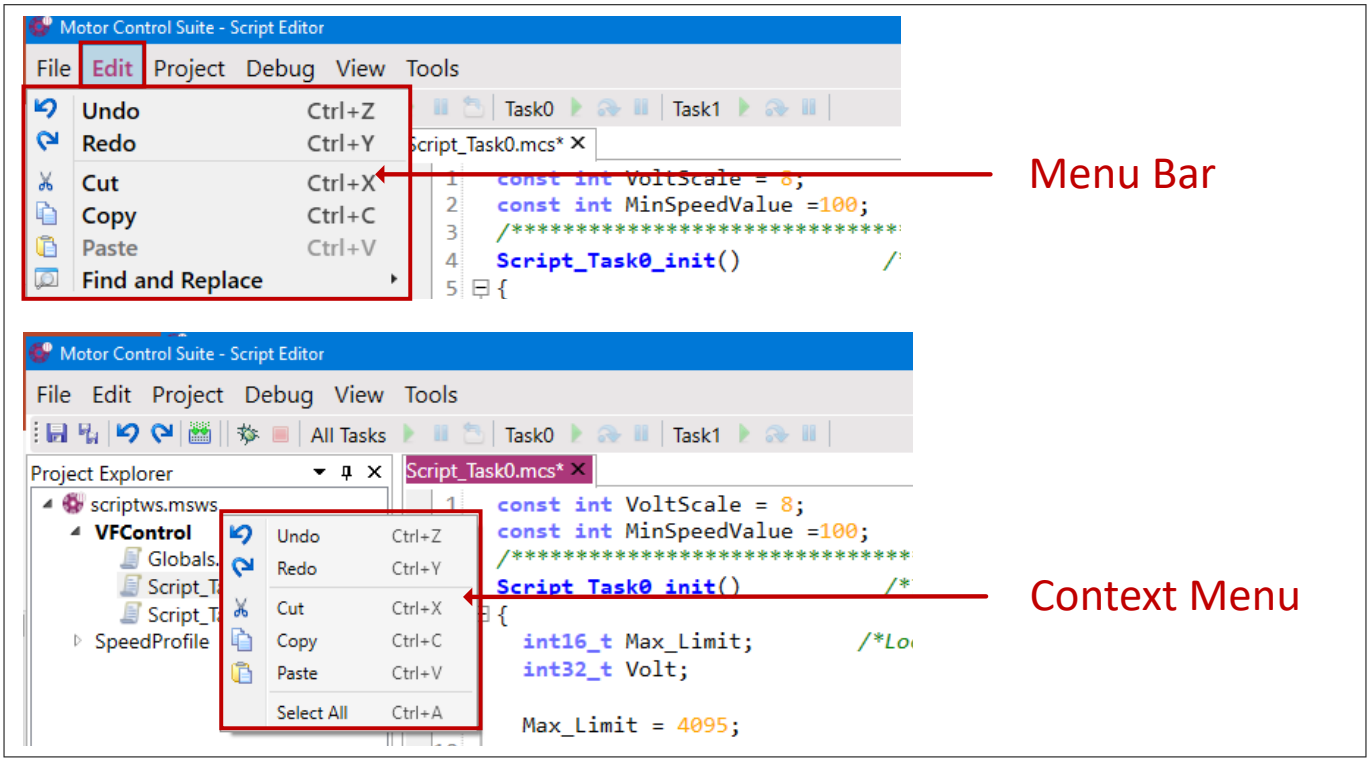

Figure 101 Edit Menu Bar and Context Menu

Table 3 Script Editor: Editor Menu

Menu	Shortcut	Description
Undo	Ctrl + Z	Cancel - the last edit operation.
Redo	Ctrl + Y	Restore - the last undone operation.

(table continues...)

Table 3 (continued) Script Editor: Editor Menu

Menu	Shortcut	Description
Cut	Ctrl + X	Cut the selected text to the clipboard.
Copy	Ctrl + C	Copy the selected text to the clipboard.
Paste	Ctrl + V	Paste the text from the clipboard.
Find and Replace → Find	Ctrl + F	Open find window to search for text patterns in a file or project.
Find and Replace → Replace	Ctrl + H	Open replace window to replaces the specified text with another text.
Select All	Ctrl + A	Select complete text of selected file.

9.3.4 View Menu

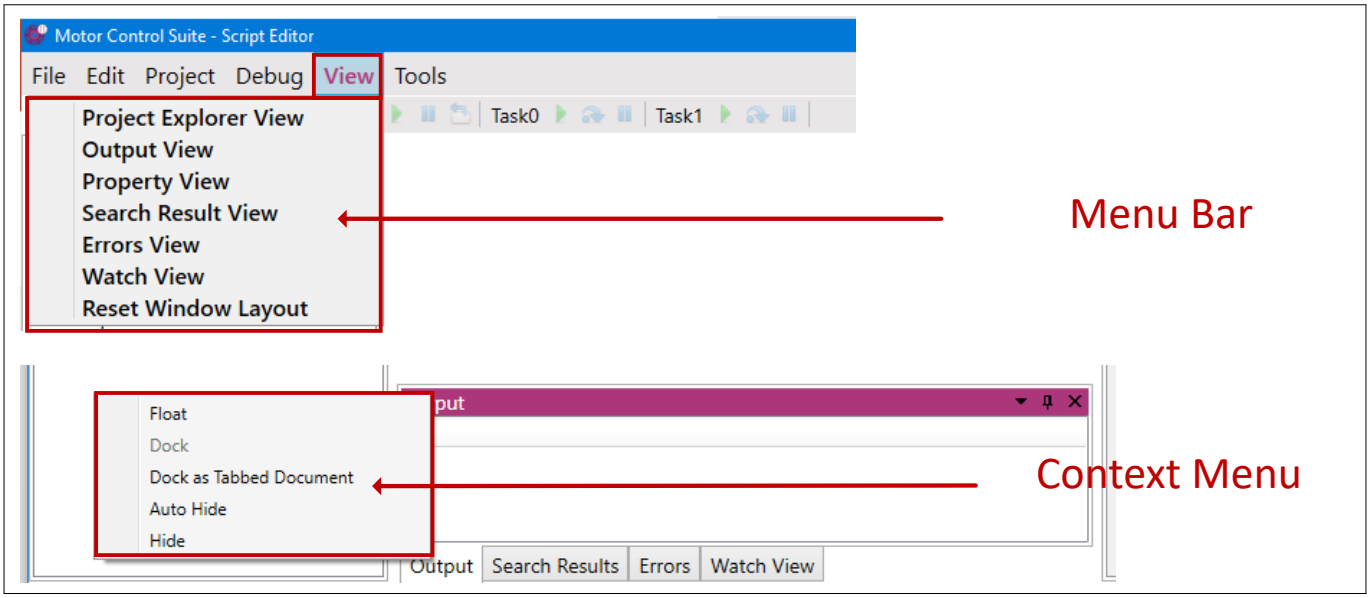

Figure 102 View Menu Bar and Context Menu

Table 4 Script Editor: View Menu

Menu	Shortcut	Description
Project Explorer View		Allow to manage the script projects
Output View		Show the script Build output summary
Property View		Project/File/Workspace property window
Search Result View		Show the search result
Error View		Show the syntax and compilation error
Watch View		Read/Write script variables in Debug Mode
Reset Window Layout		Reset the window perspective to default
Float		Float the window, as a separate window
Dock		Dock the window into the default location

(table continues...)

Table 4 (continued) Script Editor: View Menu

Menu	Shortcut	Description
Dock as Tabbed Document		Dock the window to the Editor
Auto Hide		Hide the window as soon as it loses focus
Hide		Hide the window completely

9.3.5 Debug Menu

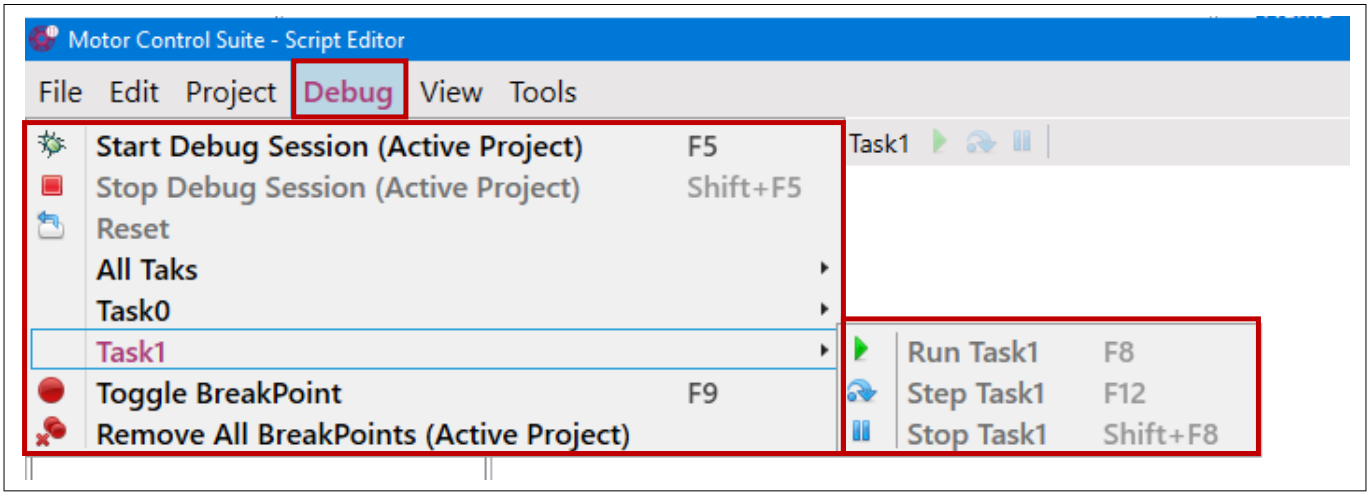

Figure 103 Debug Menu

Table 5 Script Editor: Debug Menu

Menu	Shortcut	Description
Start Debug Session	F5	Start script debug session
Stop Debug Session	Shift + F5	Stop script debug session
All Tasks → Run All Tasks	F6	Run both the script tasks, task0 and task1
All Tasks → Stop All Tasks	Shift + F6	Stop both the script tasks, task0 and task1
Reset		Reset the script code
Task0 → Run Task0	F7	Run script task0
Task0 → Step Task0	F11	Step script task0
Task0 → Stop Task0	Shift + F7	Stop script task0
Task1 → Run Task0	F8	Run script task1
Task1 → Step Task0	F12	Step script task1
Task1 → Stop Task0	Shift + F6	Stop script task1
Toggle Breakpoint	F9	Add or remove breakpoints
Remove All Breakpoints		Remove all the breakpoints

9.3.6 Script Editor Tool Bar

Script editor icons are added in the tool bar and all the available icons are captured in the below diagram.

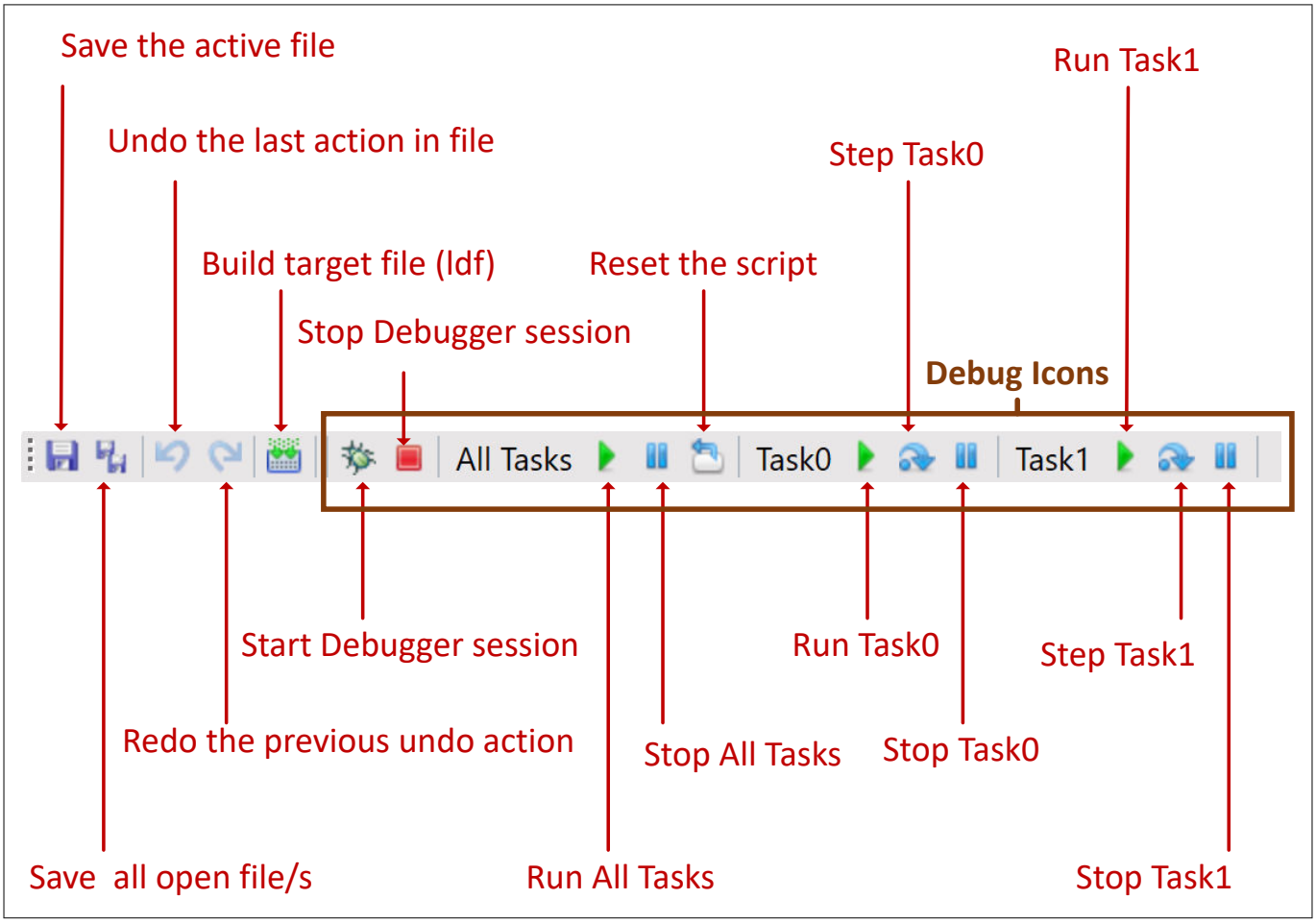

Figure 104 Script Editor: Tool Bar

9.4 Working with a Script Project

9.4.1 Creating a Script Project

- Launch the Script Editor from the iMOTION™ Solution Designer (Tools → Script Editor or using Tool Bar icon)

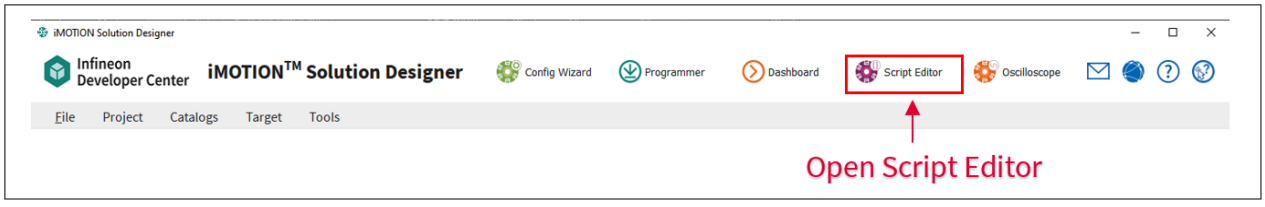

Figure 105 Open Script Editor

- Create a new script project (Menu → Project → New Project or right click on script workspace)

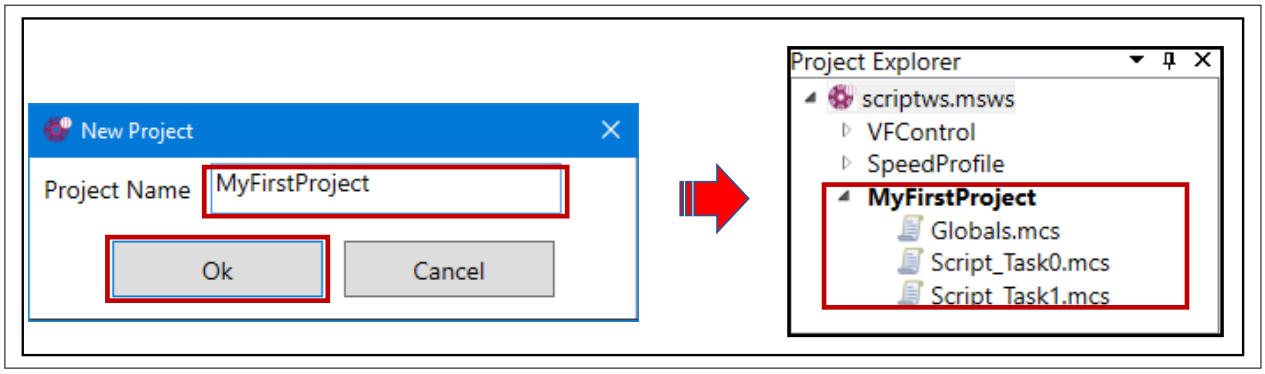

Figure 106 Create a New Script Project

- In the new project, three script files will be added (Globals.mcs, Script_TasK0.mcs and Script_Task1.mcs), by default.
- Add the script code in the relevant section. It is recommended to define all the global variables in Globals.mcs file.
  - Script files can be added or deleted. It is also possible to exclude a file from the build.
  - Task execution period and step can be configured in Project Property window.
- Build the project (Menu → Project → Compile or using Tool Bar icon).
  - Build sequence: Globals.mcs file will be compiled first, with other script files then being compiled based on alphabetical order.

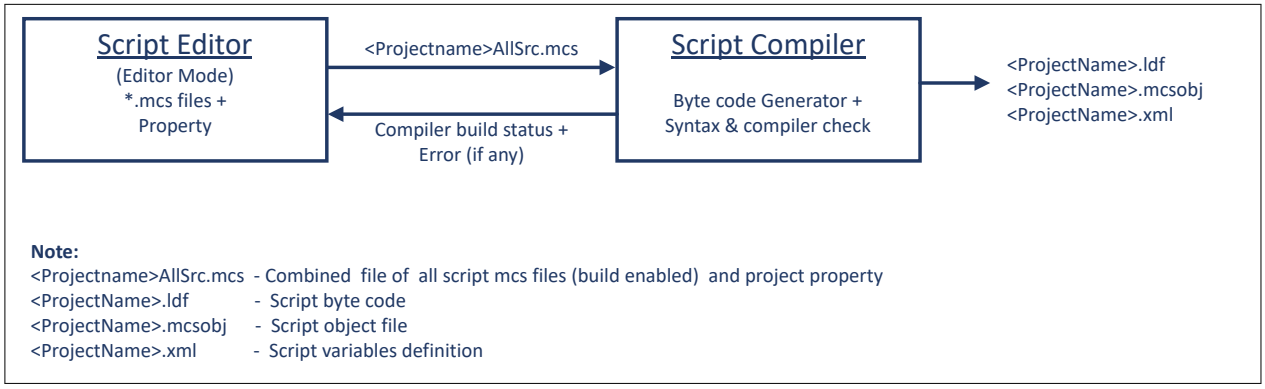

Figure 107 Script editor: Editor mode

Each script project contains following files:

- <projectname>.mspr: Script project file
- *.mcs: Script files
- <Projectname>AllSrc.mcs: Combined mcs files and project property (output folder)
- <ProjectName>.ldf: Script byte code (output folder)
- <ProjectName>.mcsobj: Script object file (output folder)
- <ProjectName>.xml: Script variables definition (output folder)

9.4.2 Debugging a Script Project

- After successfully build of the script project, program the script byte code from Solution Designer (In Solution project, Build Project with script and program the device)

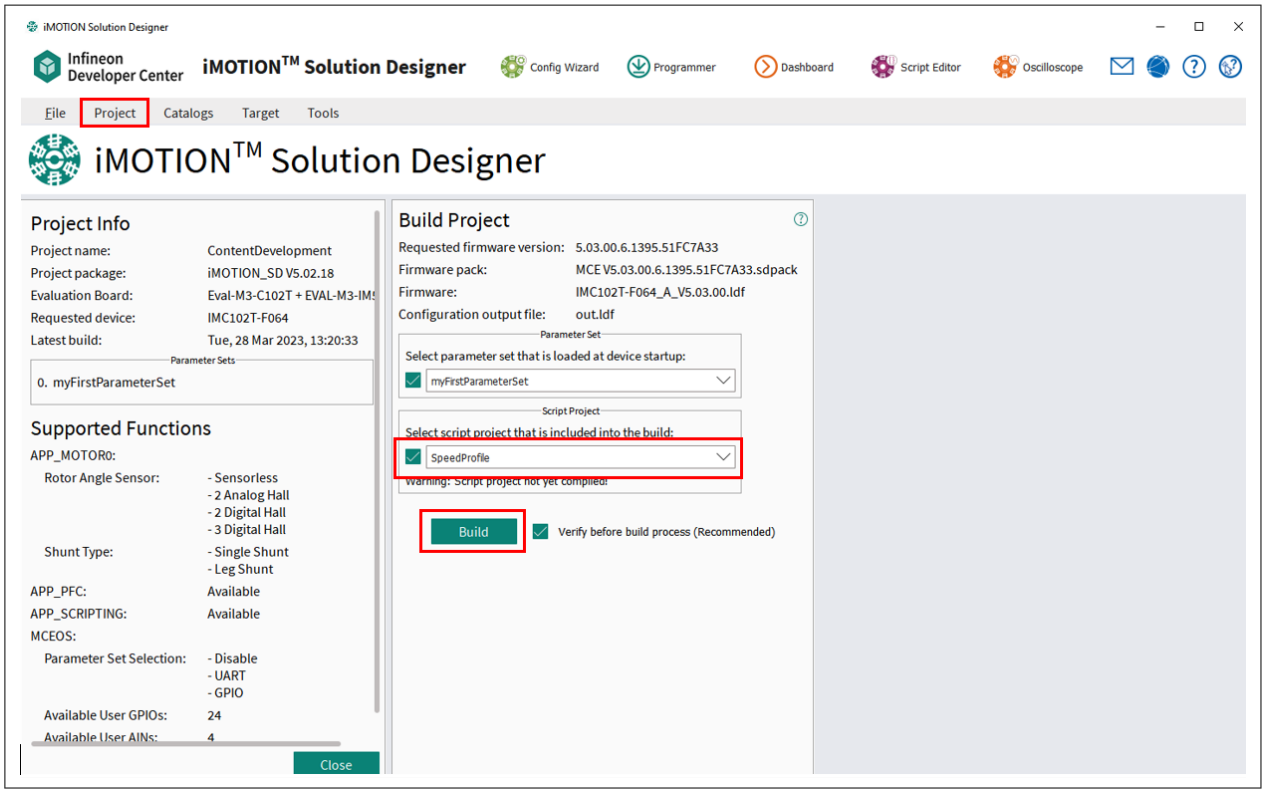

Figure 108 Building a Project

- Start debug session in the script editor using the debug icon or Menu → Debug → Start Debug session.
  - Make sure the Solution Designer is connected to the target, before starting the debug session.

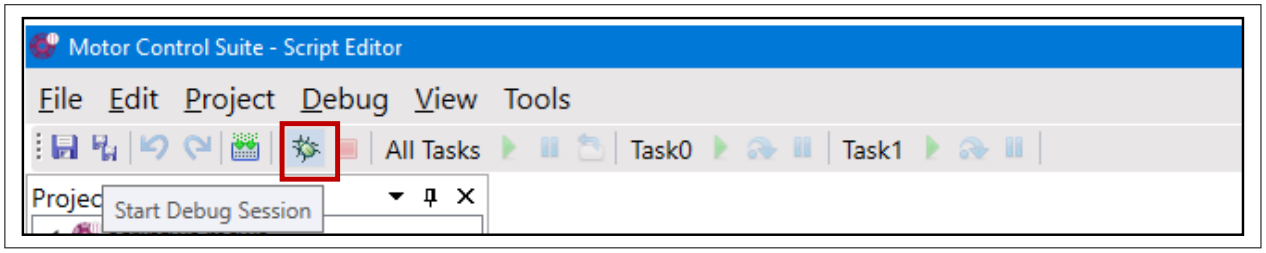

Figure 109 Start Debug Session

- Debug session will be started if the current project byte code matches with the programmed byte code.

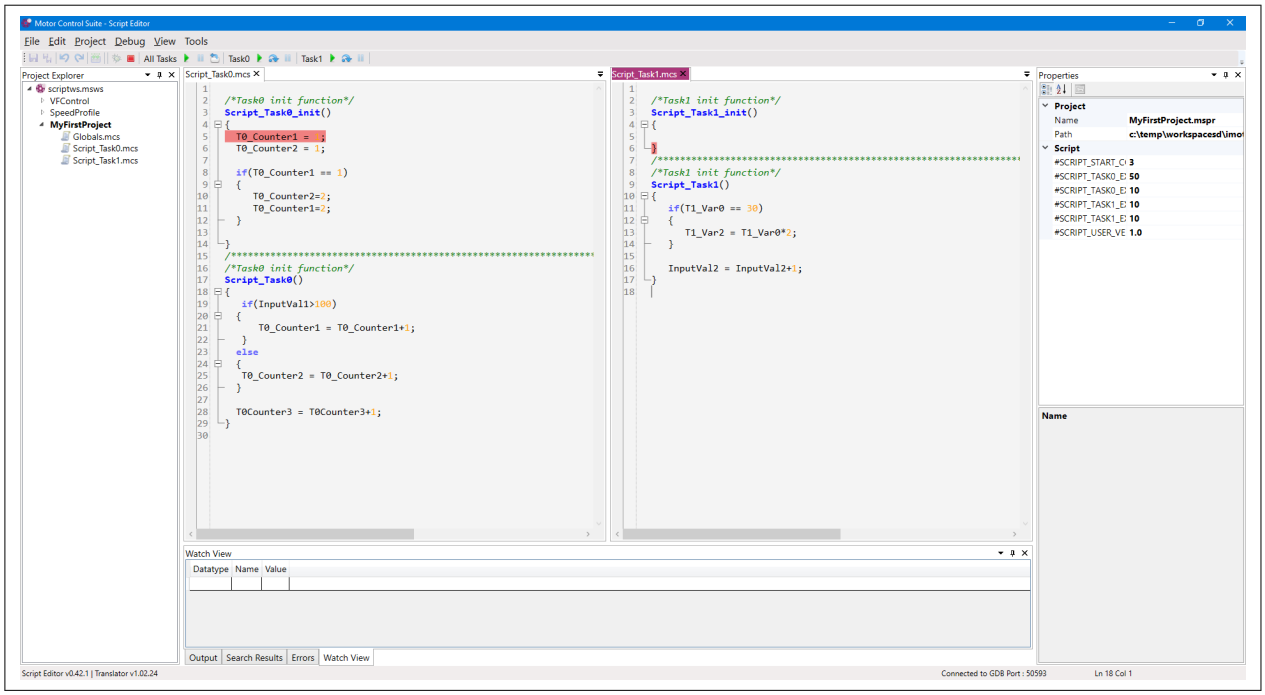

Figure 110 Debug Session

- In the debug session, it is possible to control the execution of Task0 and Task1 functions separately or together. Reset the script code is also possible in debug session.
- It is possible to place up to four breakpoints in each task.
- Script variable and MCE parameter/variables can be monitored, but only script variables are writable in debug session.
- After completing the script debugging, press Stop Debug icon or Menu → Debug → Stop Debug session to stop debug session.

10 Appendix

Table 6 File descriptions

File	Location	Description	Module Owner
.cwproj	/setting/projectName.cwprojc	File used to open project. It contains the link code for parameter sets and Common screens. • One file per project.	Solution Designer
/cw_data	/settings/cw_data	Directory where all the ICWP files are stored.	Config Wizard, Dashboard
.icwp	/settings/cw_data/screen.icwp	File used to store content for the screen (Values from input questions). • One file per screen. **Global Files** • Common_dashboard.icwp • Common_main.icwp • Common_mceos.icwp **ParameterSet** • ConfigWizard → ParSetName_Solution.icwp • APP_MOTOR → ParSetName_motor.icwp • APP_SYSTEM → ParSetName_system.icwp • APP_PFC → ParSetName_pfc.icwp	Config Wizard, Dashboard
.ldf	/generated/out.ldf /generated/IMC101T-T038/ IMC101T-T038_A_V2.ldf	File used to program Board. • out.ldf = parameter file. • IMC101T-T038_A_V2.ldf = firmware.	Programmer
package.xml	/lib/package.xml	package.xml contains information about the sdpack used to create the current solution designer project.	Solution Designer
.oszi	/settings/CurrentControl.oszi	File used by the oscilloscope to load a configuration of parameters. • Ei. The file CurrentControl will select signals related to current control (idRed_Ext, lqFilt).	Oscilloscope
.mcs	/Script/projectName/ Script_Task0.mcs	Script file .ms file contains the content (code) for the user-defined script.	Script
.msws	/Script/scriptws.msws	scriptws.msws is the workspace project. scriptws.msws contains the name of script projects within the solution designer project.	Script
.xshd	/Script/.config/mcs.xshd	mcs.xshd is the style sheet file for the script editor.	Script
.xml	/Script/.config/parameter.xml	parameter.xml contains mce parameters that the user-defined script can use.	Script

(table continues...)

Table 6 (continued) File descriptions

File	Location	Description	Module Owner
.mspr	/Script/ ProejctName.projectName.mspr	projectName.mspr contains the Script project file.	Script
.ldf	/Script/ProjectName/Output/ Projectname.ldf	Projectname.ldf contains Script byte code.	Script
.mcsobj	/Script/ProjectName/Output/ ProjectnName.mcsobj	Script object file.	Script
.xml	/Script/ProjectName/Output/ ProjectnName.xml	ProjectnName.xml contains the Script variables definition.	Script

Revision history

Document revision	Date	Description of changes
1.0	2022-12-21	Initial document release
1.0.1	2023-04-11	Minor formatting update
1.1	2023-09-08	The dashboard section has been updated with the new 5.2 features ("Update Config Wizard", "Sync From Config Wizard"). The "Working with Solution Designer" section has been renamed to "Configuration Wizard"
1.2	2024-02-12	New Programmer section to reflect Programmer behavior. Minor updates to other sections.

Trademarks

All referenced product or service names and trademarks are the property of their respective owners.

Edition 2024-02-12

Published by

Infineon Technologies AG

81726 Munich, Germany

Do you have a question about any aspect of this document?

Email: erratum@infineon.com

Document reference
IFX-lqt1680618724158

Important notice

The information contained in this application note is given as a hint for the implementation of the product only and shall in no event be regarded as a description or warranty of a certain functionality, condition or quality of the product. Before implementation of the product, the recipient of this application note must verify any function and other technical information given herein in the real application. Infineon Technologies hereby disclaims any and all warranties and liabilities of any kind (including without limitation warranties of non-infringement of intellectual property rights of any third party) with respect to any and all information given in this application note.

The data contained in this document is exclusively intended for technically trained staff. It is the responsibility of customer's technical departments to evaluate the suitability of the product for the intended application and the completeness of the product information given in this document with respect to such application.

Warnings

Due to technical requirements products may contain dangerous substances. For information on the types in question please contact your nearest Infineon Technologies office.

Except as otherwise explicitly approved by Infineon Technologies in a written document signed by authorized representatives of Infineon Technologies, Infineon Technologies' products may not be used in any applications where a failure of the product or any consequences of the use thereof can reasonably be expected to result in personal injury.