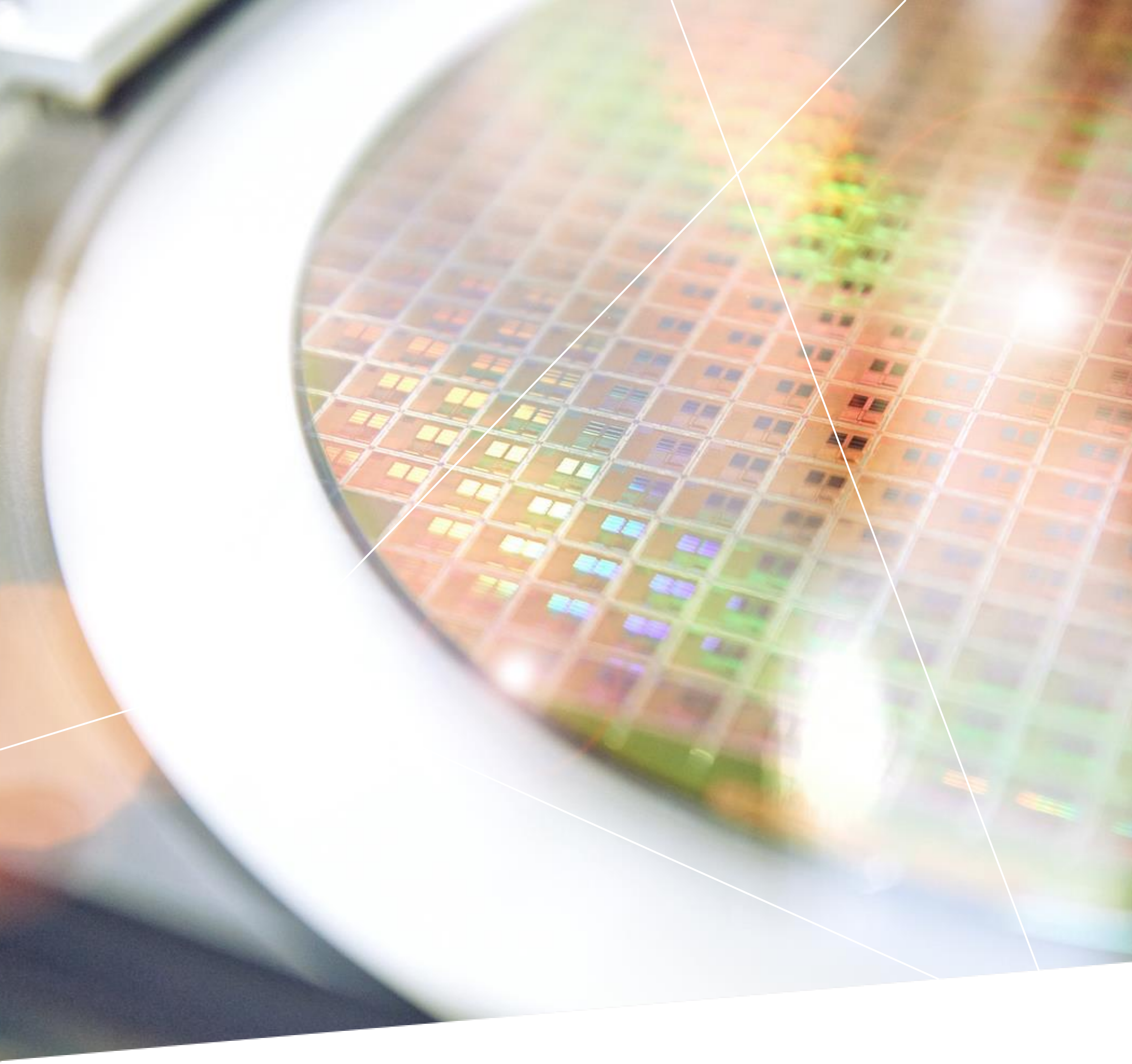


iMOTION™ Solution Designer

User Guide

2022-12-22

www.infineon.com

1 Introduction

1.1 Purpose

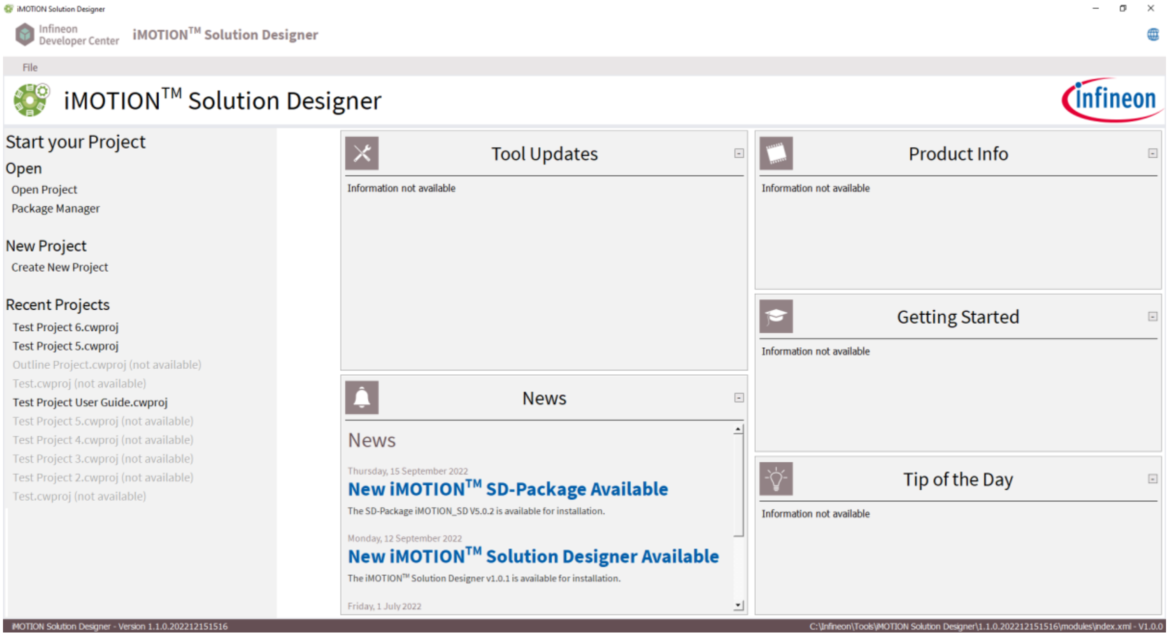
The **iMOTION™ Solution Designer** (iSD) is an integrated GUI-based tool for use in the iMOTION™ motor control development process. This program replaces the MCEWizard and MCEDesigner, and act as new all-in-one tool for iMOTION™ users.

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.


iMOTION Solution Designer
Infineon Developer Center
iMOTIONTM Solution Designer
File
iMOTIONTM Solution Designer
infineon
Start your Project
Open
Open Project
Package Manager
New Project
Create New Project
Recent Projects
Test Project 6.cwproj
Test Project 5.cwproj
Outline Project.cwproj (not available)
Test.cwproj (not available)
Test Project User Guide.cwproj
Test Project 5.cwproj (not available)
Test Project 4.cwproj (not available)
Test Project 3.cwproj (not available)
Test Project 2.cwproj (not available)
Test.cwproj (not available)
Tool Updates
Information not available
Product Info
Information not available
Getting Started
Information not available
News
Tip of the Day
Information not available

News

Thursday, 15 September 2022
New iMOTIONTM SD-Package Available
The SD-Package iMOTION_SD V5.0.2 is available for installation.

Monday, 12 September 2022
New iMOTIONTM Solution Designer Available
The iMOTIONTM Solution Designer v1.0.1 is available for installation.

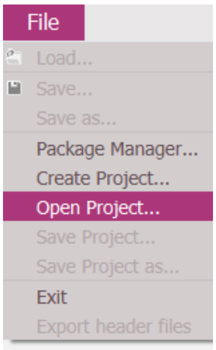
Friday, 1 July 2022

iMOTION Solution Designer - Version 1.1.0.202212151516

C:\Infineon\Tools\iMOTION Solution Designer\1.1.0.202212151516\modules\index.xml - V1.0.0

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.



- **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

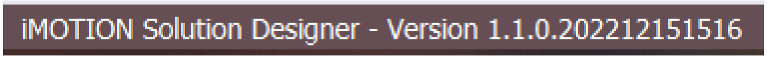
The Navigation pane allows users to quickly access their project files. Users can **Open Projects** and access the **Package Manager.** Users can opt to **Create a New Project.** The Navigation pane will also display a number of **Recent Projects** available locally.

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

These panes are regularly updated, informing users 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.

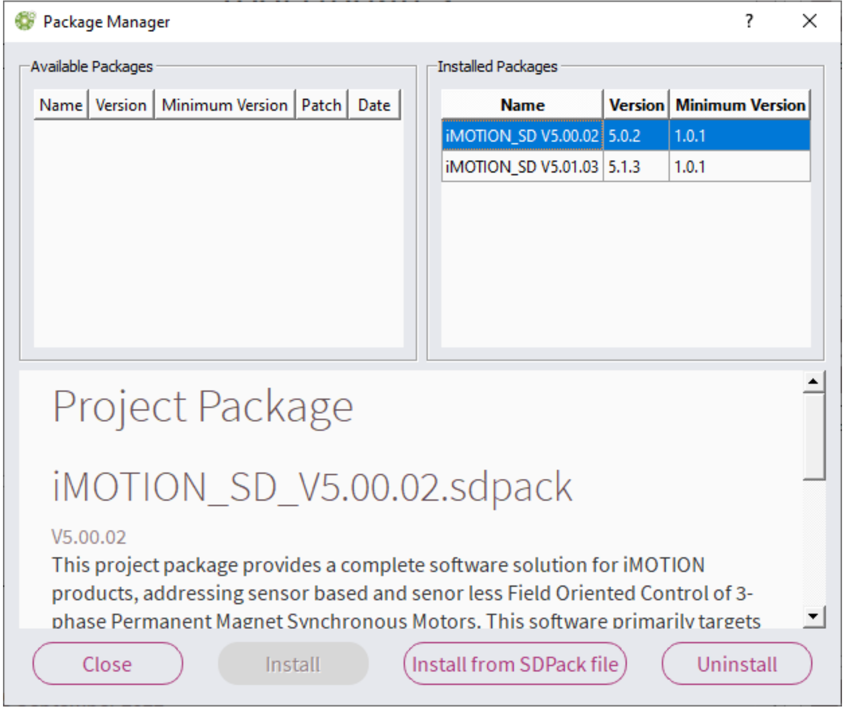
2.1.4 Version Number and Installation File Path

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.


iMOTION Solution Designer - Version 1.1.0.202212151516

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.



Name	Version	Minimum Version	Patch	Date

Name	Version	Minimum Version
iMOTION_SD V5.00.02	5.0.2	1.0.1
iMOTION_SD V5.01.03	5.1.3	1.0.1

Project Package

iMOTION_SD_V5.00.02.sdpack

V5.00.02

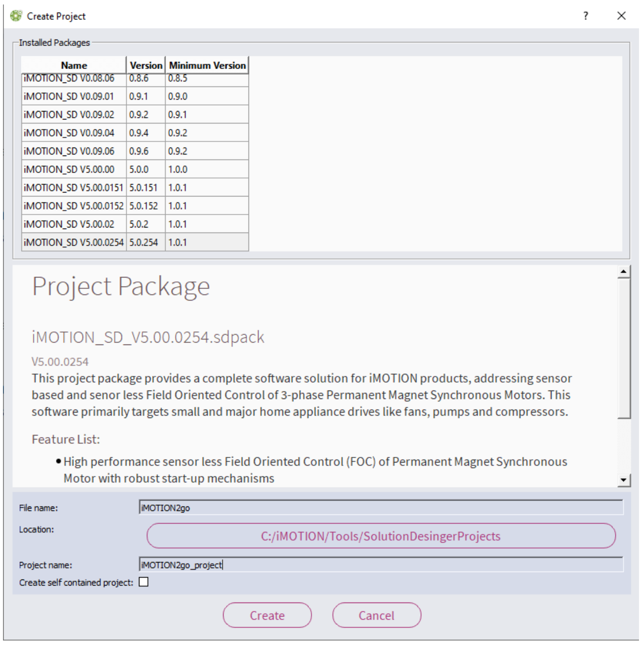
This project package provides a complete software solution for iMOTION products, addressing sensor based and senor less Field Oriented Control of 3-phase Permanent Magnet Synchronous Motors. This software primarily targets

Below, 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, you 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.



Name	Version	Minimum Version
iMOTION_SD V0.08.06	0.8.6	0.8.5
iMOTION_SD V0.09.01	0.9.1	0.9.0
iMOTION_SD V0.09.02	0.9.2	0.9.1
iMOTION_SD V0.09.04	0.9.4	0.9.2
iMOTION_SD V0.09.06	0.9.6	0.9.2
iMOTION_SD V5.00.00	5.0.0	1.0.0
iMOTION_SD V5.00.0151	5.0.151	1.0.1
iMOTION_SD V5.00.0152	5.0.152	1.0.1
iMOTION_SD V5.00.02	5.0.2	1.0.1
iMOTION_SD V5.00.0254	5.0.254	1.0.1

Project Package

iMOTION_SD_V5.00.0254.sdpack

V5.00.0254

This project package provides a complete software solution for iMOTION products, addressing sensor based and senor less Field Oriented Control of 3-phase Permanent Magnet Synchronous Motors. This software primarily targets small and major home appliance drives like fans, pumps and compressors.

Feature List:

- High performance sensor less Field Oriented Control (FOC) of Permanent Magnet Synchronous Motor with robust start-up mechanisms

File name: iMOTION2go
Location: C:/iMOTION/Tools/SolutionDesingerProjects
Project name: iMOTION2go_project
Create self contained project: □

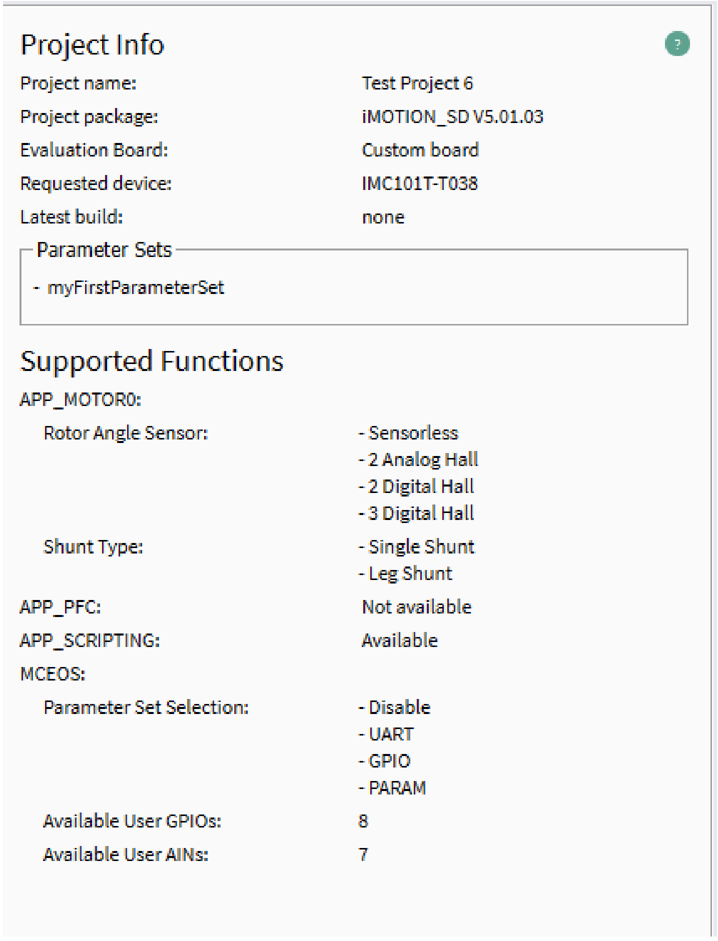
2.4 Project Configuration

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 projects settings were applied to the requested device/were applied to a device.
- Parameter Sets – Upon project creation and SD Pack application, will default to 'myFirstParameterSet'.



Project Info

Project name:	Test Project 6
Project package:	iMOTION_SD V5.01.03
Evaluation Board:	Custom board
Requested device:	IMC101T-T038
Latest build:	none

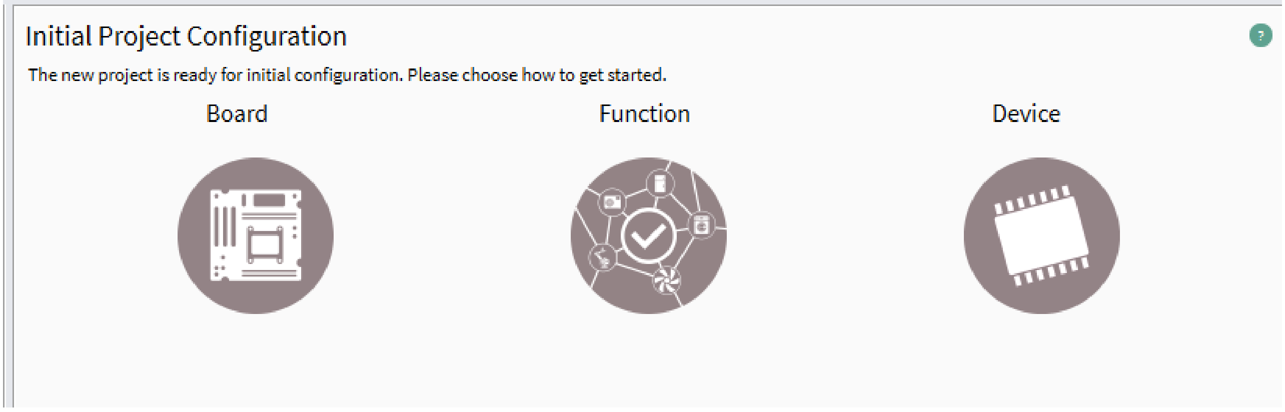
Parameter Sets
- myFirstParameterSet

Supported Functions

APP_MOTOR0:	
Rotor Angle Sensor:	- Sensorless - 2 Analog Hall - 2 Digital Hall - 3 Digital Hall
Shunt Type:	- Single Shunt - Leg Shunt
APP_PFC:	Not available
APP_SCRIPTING:	Available
MCEOS:	
Parameter Set Selection:	- Disable - UART - GPIO - PARAM
Available User GPIOs:	8
Available User AINs:	7

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.



Initial Project Configuration

The new project is ready for initial configuration. Please choose how to get started.

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.

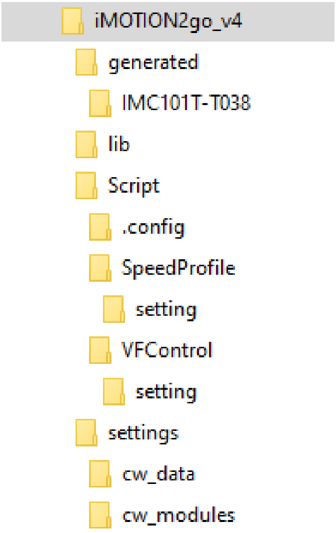
2.4.2.3 Device

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 uploaded 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.



```
iMOTION2go_v4
  generated
    IMC101T-T038
  lib
  Script
    .config
    SpeedProfile
      setting
    VFControl
      setting
  settings
    cw_data
    cw_modules
```

2.5.1 Sharing a Project

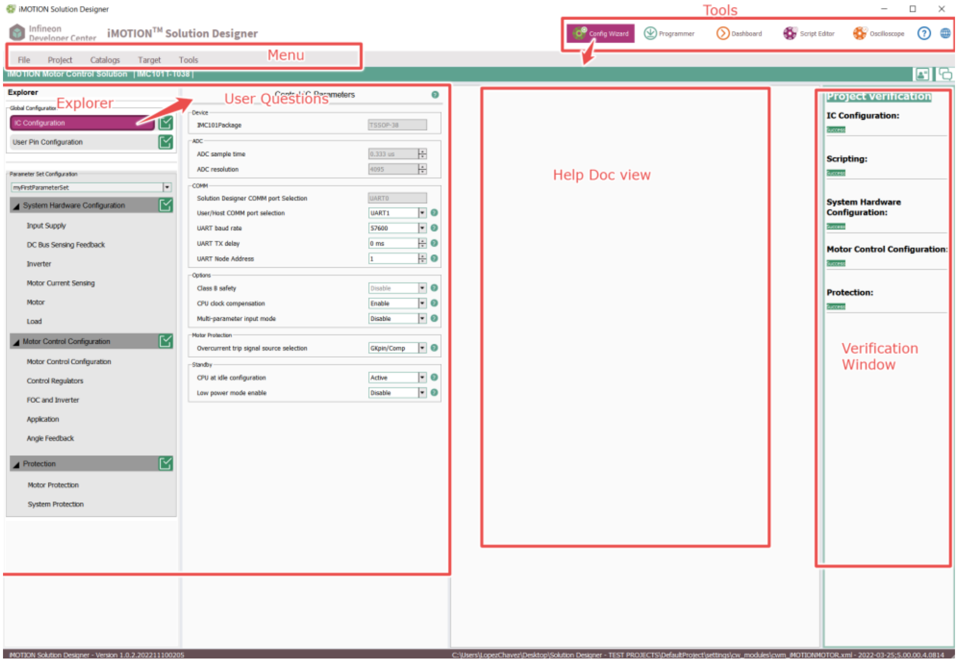
Depending on the method used to create the project, users have two ways of sharing a project:

- **Standard Project** – Users should ensure the recipients software 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 filevto 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 Working with the Solution Designer

3.1 Introduction

The **Solution Designer** 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.



Device

IMC101Package: TSSOP-38

ADC

ADC sample time: 0.333 us

ADC resolution: 4095

COMM

Solution Designer COMM port Selection: UART0

User/Host COMM port selection: UART1

UART baud rate: 57600

UART TX delay: 0 ms

UART Node Address: 1

Options

Class B safety: Disable

CPU clock compensation: Enable

Multi-parameter input mode: Disable

Motor Protection

Overcurrent trip signal source selection: GKpin/Comp

Standby

CPU at idle configuration: Active

Low power mode enable: Disable

IC Configuration: Success

Scripting: Success

System Hardware Configuration: Success

Motor Control Configuration: Success

Protection: Success

iMOTION Solution Designer - Version 1.0.2.202211100205

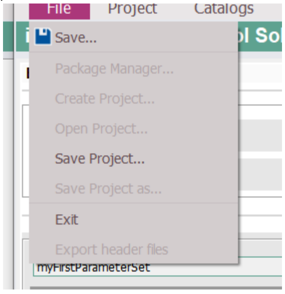
C:\Users\LopezChavez\Desktop\Solution Designer - TEST PROJECTS\DefaultProject\settings\cw_modules\cwm_iMOTIONMOTOR.xml - 2022-03-25;5.00.00.4.0814

3.2 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.

3.2.1 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.



3.2.1.1 Save

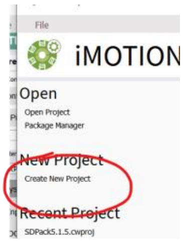
Users can use this function to save the changes made by the current tool selected. For example, if users are currently using the Dashboard tool, then the iSD will only save changes made by the Dashboard tool. Users should always save their current changes before switching to a different tool.

3.2.1.2 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.

3.2.1.3 Create Project

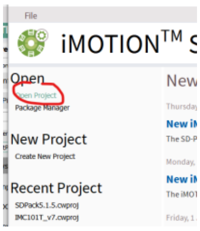
This allows users to create a new project. It shares the same functionality as the **Create New Project** option on the HOME screen.



Note: Create Project is only selectable when a project has yet to be opened.

3.2.1.4 Open Project

This allows users to open an existing project. It shares the same functionality as the **Open Project** option on the HOME screen.



Note: Open Project is only selectable when a project has yet to be opened.

3.2.1.5 Save Project

This will let users to save all the parameter sets changed in the Dashboard settings.
Note: Save Project is only selectable when a project is open.

3.2.2 Project Settings

3.2.2.1 Project Info

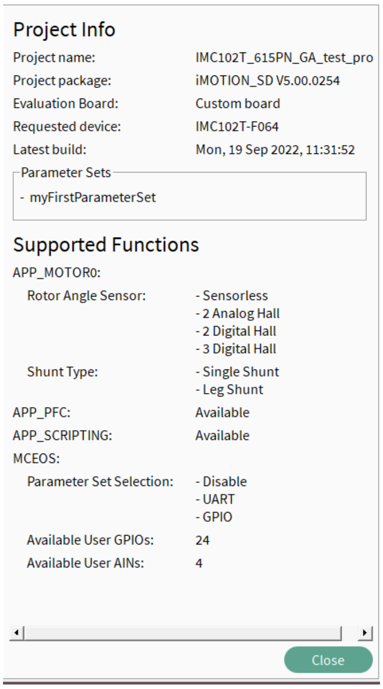
The **Project Info** section allows users to quickly see a brief, but detailed list of the project's information. This includes:

- **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.

3.2.2.2 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.



Project Info

Project name:	IMC102T_615PN_GA_test_pro
Project package:	iMOTION_SD V5.00.0254
Evaluation Board:	Custom board
Requested device:	IMC102T-F064
Latest build:	Mon, 19 Sep 2022, 11:31:52

Parameter Sets
- myFirstParameterSet

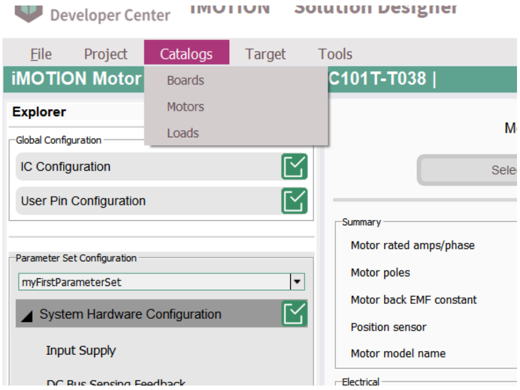
Supported Functions

APP_MOTOR0:	
Rotor Angle Sensor:	- Sensorless - 2 Analog Hall - 2 Digital Hall - 3 Digital Hall
Shunt Type:	- Single Shunt - Leg Shunt
APP_PFC:	Available
APP_SCRIPTING:	Available
MCEOS:	
Parameter Set Selection:	- Disable - UART - GPIO
Available User GPIOs:	24
Available User AINs:	4

3.2.3 Catalogs

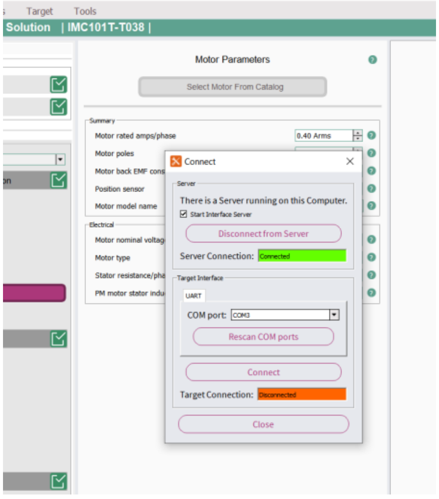
Catalogs allow user to access the iSD database of records and settings for **Boards**, **Motors**, and **Loads**. Users can quickly apply this information to their current Configuration Wizard parameter set via the pop-out menu. Board catalog data only contains system hardware configuration related parameters 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



3.2.4 Target

The Target pop-out 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-out window appears automatically if the target is not connected when the Programmer, Dashboard, or Oscilloscope is selected.



There is a Server running on this Computer.

☑ Start Interface Server

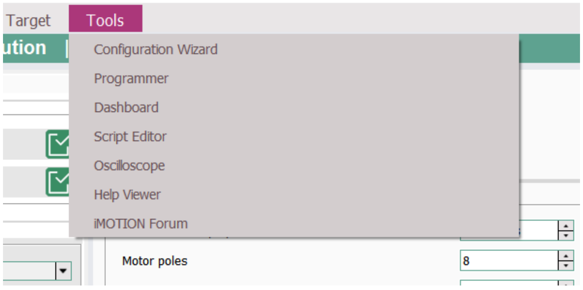
Server Connection: Connected

COM port: COM3

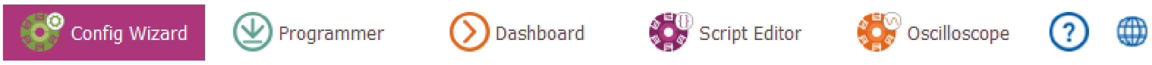
Target Connection: Disconnected

3.2.5 Tools

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



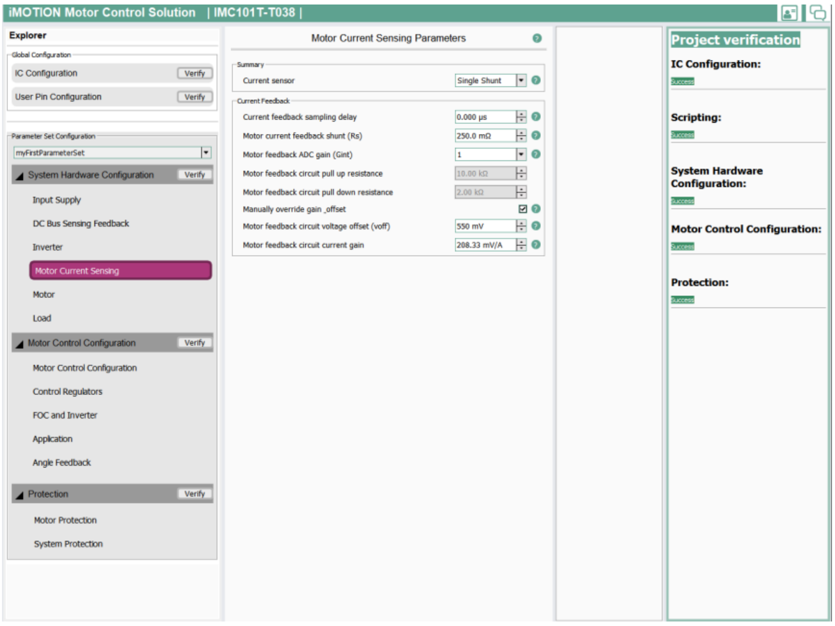
Additionally, several of the tools can be accessed via the icons at the top-right of the user's interface.



3.3 Configuration Wizard

The **Configuration Wizard** will allow users to access and edit specific parameters within their projects. The information provided by the Configuration Wizard can be further modified via **Normal Mode** and **Expert Mode.**

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.



Motor Current Sensing Parameters

Summary

Current sensor: Single Shunt

Current Feedback

Current feedback sampling delay: 0.000 µs
Motor current feedback shunt (Rs): 250.0 mΩ
Motor feedback ADC gain (Gint): 1
Motor feedback circuit pull up resistance: 10.00 kΩ
Motor feedback circuit pull down resistance: 2.00 kΩ
Manually override gain_offset: ☑
Motor feedback circuit voltage offset (voff): 550 mV
Motor feedback circuit current gain: 208.33 mV/A

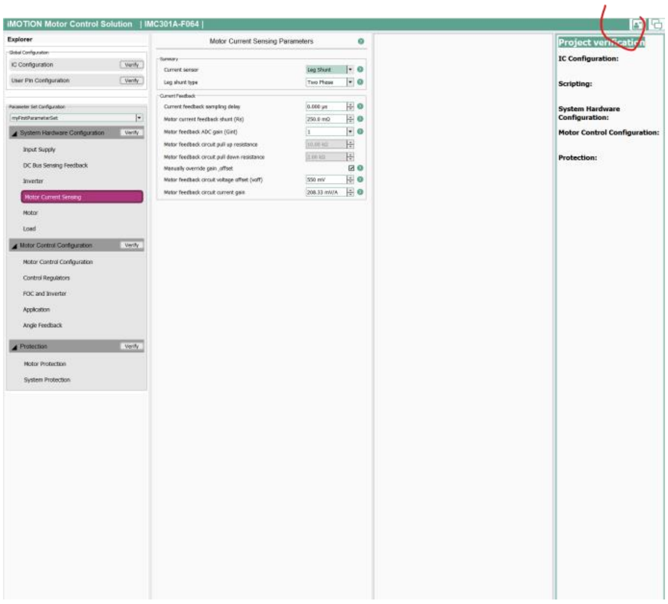
Project verification

IC Configuration: Success
Scripting: Success
System Hardware Configuration: Success
Motor Control Configuration: Success
Protection: Success

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.

3.3.1 Normal Mode vs. Expert Mode



Motor Current Sensing Parameters

Summary

Current sensor: Leg Shunt
Leg shunt type: Two Phase

Current Feedback

Current feedback sampling delay: 0.000 µs
Motor current feedback shunt (Rs): 250.0 mΩ
Motor feedback ADC gain (Gint): 1
Motor feedback circuit pull up resistance: 10.00 kΩ
Motor feedback circuit pull down resistance: [illegible]
Manually override gain_offset: ☑
Motor feedback circuit voltage offset (voff): 550 mV
Motor feedback circuit current gain: 208.33 mV/A

Project verification

IC Configuration:
Scripting:
System Hardware Configuration:
Motor Control Configuration:
Protection:

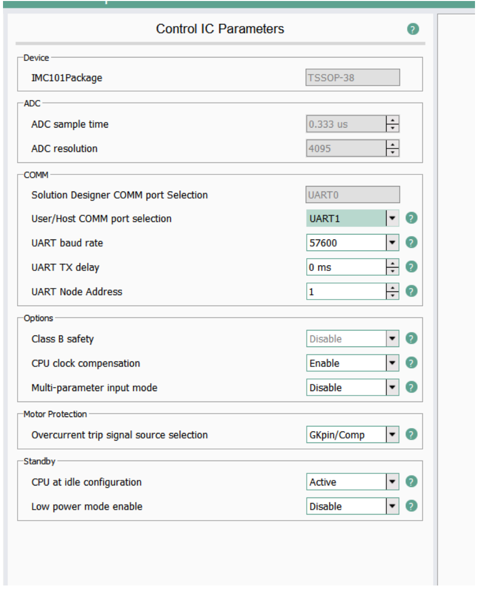
- **Normal Mode** – This will show users the most common and required parameters.
- **Expert Mode** – This will open up the full parameter tree for users, allowing for more detailed editing.

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 the 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.

3.3.2 IC Configuration

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



Control IC Parameters

Device

IMC101Package	TSSOP-38

ADC

ADC sample time	0.333 us
ADC resolution	4095

COMM

Solution Designer COMM port Selection	UART0
User/Host COMM port selection	UART1
UART baud rate	57600
UART TX delay	0 ms
UART Node Address	1

Options

Class B safety	Disable
CPU clock compensation	Enable
Multi-parameter input mode	Disable

Motor Protection

Overcurrent trip signal source selection	GKpin/Comp

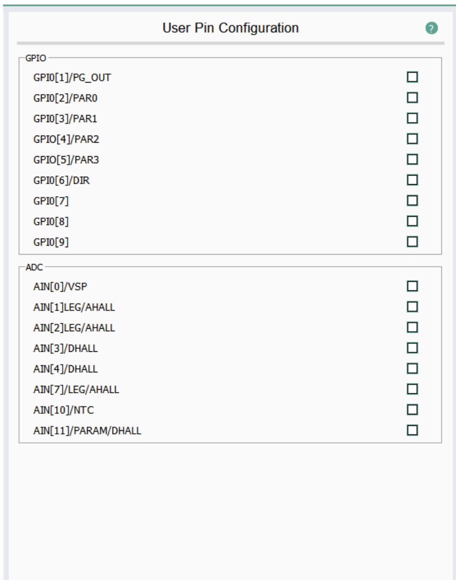
Standby

CPU at idle configuration	Active
Low power mode enable	Disable

3.3.3 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.



User Pin Configuration

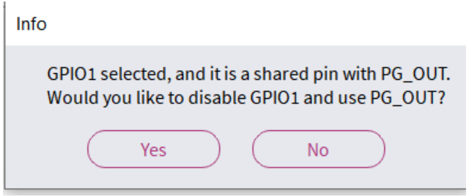
GPIO

GPI0[1]/PG_OUT □
GPI0[2]/PAR0 □
GPI0[3]/PAR1 □
GPIO[4]/PAR2 □
GPIO[5]/PAR3 □
GPI0[6]/DIR □
GPI0[7] □
GPI0[8] □
GPI0[9] □

ADC

AIN[0]/VSP □
AIN[1]LEG/AHALL □
AIN[2]LEG/AHALL □
AIN[3]/DHALL □
AIN[4]/DHALL □
AIN[7]/LEG/AHALL □
AIN[10]/NTC □
AIN[11]/PARAM/DHALL □

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



Info

GPIO1 selected, and it is a shared pin with PG_OUT.
Would you like to disable GPIO1 and use PG_OUT?

3.3.4 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.

3.3.5 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 DC bus voltage, Control supply voltage, Input supply, and ADC gain.
- **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.

3.3.6 Motor Control Configuration

The sections in this parameter group gives users finer control over the mechanisms of the device's motor.

- **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.

3.3.7 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.

3.3.8 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.

3.3.9 Frequency Response Diagrams

A **Frequency Response Diagram** allows users to monitor the output for the PFC current and voltage control loop. Frequency Response Diagrams are presented via **Bode Plots**, where the magnitude and phase of the relevant control loop transfer function are shown on a logarithmic frequency scale.

3.3.9.1 Bode Plots

A Bode plot for a given control loop transfer function includes two plots; the top plot will show the magnitude of the transfer function, and the bottom plot will show the phase shift of the transfer function. Frequency response diagrams as Bode plots help visualize the locations of the zero(s), pole(s), and the cross-over frequency, as well as the gain and phase margin of a given control loop transfer function. This further helps users understand the stability of the control loop and design the compensator accordingly to optimize the performance of the control loop.

Note: In order to access control related parameters, please switch from Normal Mode to Expert Mode.

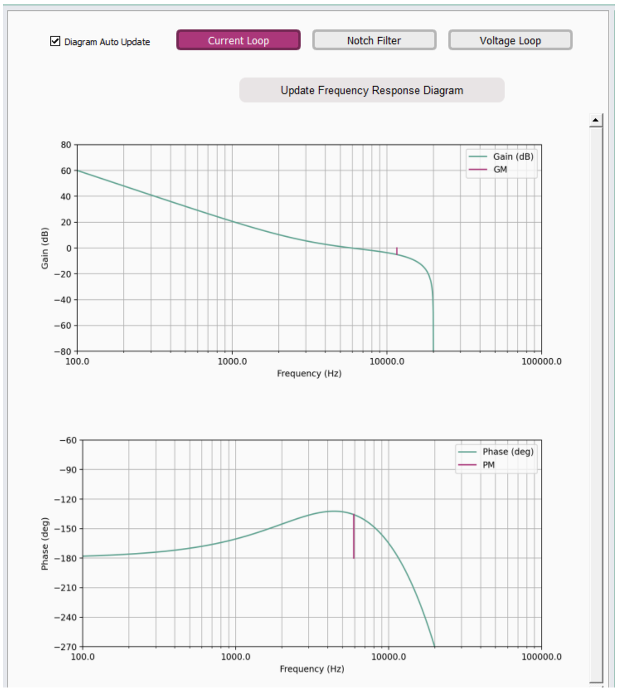
Users can click the **Current Loop**, **Notch Filter**, and **Voltage Loop** buttons to view the PFC current control loop Bode plot, the notch filter Bode Plot, or the PFC voltage control loop Bode plot respectively.

By default, the **Diagram Auto Update** checkbox is unchecked. If users make a change to a parameter that is related to the PFC control loop or the notch filter design, they will need to click the **Update Frequency Response** Diagram button to update the Bode plots accordingly. If the **Diagram Auto Update** checkbox is checked, then the control loop and the notch filter Bode plots will automatically be updated when a parameter related to the PFC control loop or the notch filter design is adjusted.

Note: Because the Bode plot update calculation is resource intensive, it is recommended that users uncheck the Diagram Auto Update checkbox if they are experiencing sluggish iSD GUI performance when adjusting those parameters related to the PFC control loop or the notch filter design.

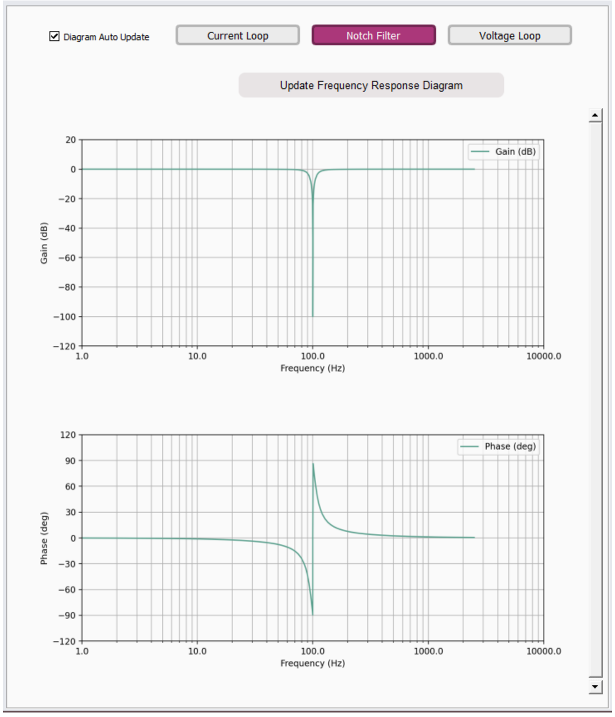
An example of the PFC current control loop Bode plot is shown in the image below. The magnitude of the PFC current control loop is shown in green, at the top plot with the unit of Decibel (dB). Users can estimate the cross-over frequency of the PFC current control loop using the top plot. The exact crossover frequency value of the PFC current control loop can be found from the parameter called **Estimated current loop crossover frequency** in the **PFC Control Regulator** section. The gain margin (GM) of the PFC current control loop is also shown in magenta at the top plot. The exact gain margin value of the PFC current control loop can be found in the parameter labelled **Estimated current loop gain margin** in the PFC Control Regulator section. The min. and max. frequency range of the top plot are fixed to 100 Hz and 100 kHz respectively. The min. and max. magnitude range of the top plot are fixed to -80 dB and 80 dB accordingly with a division of 20 dB.

The phase shift of the PFC current control loop is shown in green at the bottom plot, measured in Degrees (deg). The phase margin (PM) of the PFC current control loop is also shown in magenta in the bottom plot. The exact phase margin value of the PFC current control loop can be found from the parameter named **Estimated current loop phase margin** in the PFC Control Regulator section. The min. and max. frequency range of the bottom plot are fixed to 100 Hz and 100 kHz accordingly. The min. and max. phase range of the bottom plot are fixed to -60 deg and -270 deg accordingly with a division of 30 deg.



An example of the notch filter Bode plot is shown in the image below. The notch filter is part of the PFC voltage control loop and it is used explicitly to attenuate the second harmonics of the DC bus voltage feedback signal. The magnitude of the notch filter is shown in green at the top plot, measured in Decibels (dB). Users can conveniently estimate the bandwidth and the gain at the center frequency of the notch filter using the top plot. The min. and max. frequency range of the top plot are fixed to 1 Hz and 10 kHz respectively. The min. magnitude range of the top plot is dynamic (gain at notch filter center frequency – 20 dB). The max. magnitude range of the top plot is fixed to 20 dB.

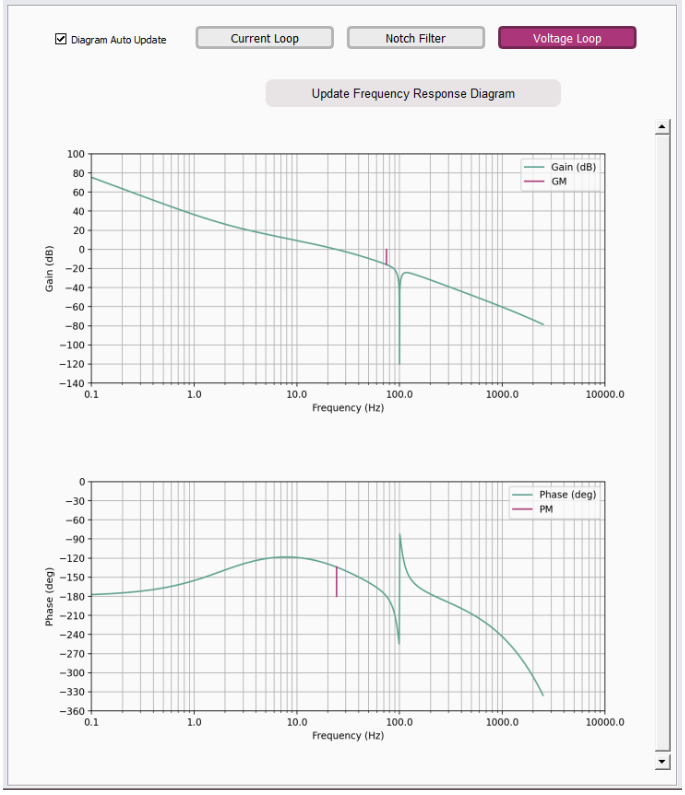
The phase shift of the notch filter is shown in green at the bottom plot, measured in Degrees (deg). The min. and max. frequency range of the bottom plot are fixed to 1 Hz and 10 kHz respectively. The min. and max. phase range of the bottom plot are fixed to -120 deg and 120 deg accordingly with a division of 30 deg.



An example of the PFC voltage control loop Bode plot is shown in the image below. The magnitude of the PFC voltage control loop is shown in green in the top plot, measured in Decibels (dB). Users can conveniently estimate the cross-over frequency of the PFC voltage control loop using the top plot.

The exact crossover frequency value of the PFC voltage control loop can be found from the parameter called **Estimated voltage loop crossover frequency** in PFC Control Regulator section. The gain margin of the PFC voltage control loop is also shown in magenta at the top plot. The exact gain margin value of the PFC voltage control loop can be found in the parameter named **Estimated voltage loop gain margin** in PFC Control Regulator section. The min. and max. frequency range of the top plot are fixed to 0.1 Hz and 10 kHz respectively. The min. and max. magnitude range of the top plot are fixed to -140 dB and 100 dB accordingly with a division of 20 dB.

The phase shift of the PFC voltage control loop is shown in green in the bottom plot measured in Degrees (deg). The phase margin of the PFC voltage control loop is also shown in magenta at the bottom plot. The exact phase margin value of the PFC voltage control loop can be found from the parameter named **Estimated voltage loop phase margin** in PFC Control Regulator section. The min. and max. frequency range of the bottom plot are fixed to 0.1 Hz and 10 kHz respectively. The min. and max. phase range of the bottom plot are fixed to 0 deg and -360 deg accordingly with a division of 30 deg.



3.4 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-out 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.

- 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.

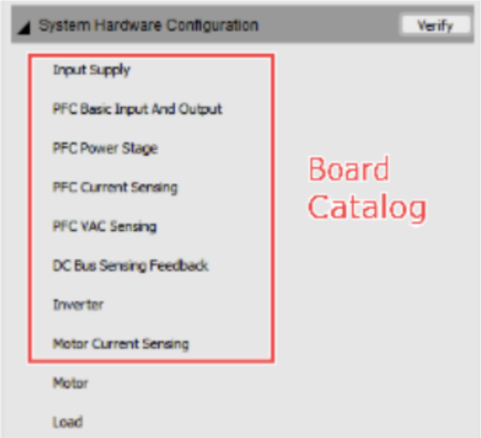
Users can also apply various filter options to the Database pane via the **Filter** pane. The Filter pane has the **Min** filter setting and the **Max** filter setting which allow users to set a range of objects to display. Users can use multiple Min or Max filters to further narrow their search. Depending on which of the three catalogs is chosen, the Filter pane will display a list of available filters.

- Boards – Board, Board Type, Connector, Power Component, Control Board, Application, iMOTION™ Device.
- Motors – Motor Model, Position Sensor, Motor Rated, Motor Poles, Motor Back EMF Constant, Motor Nominal Voltage, Motor Type, Stator Resistance/Phase, PM Motor Stator Inductance/Phase (Ls), LPM Motor, Rotor Inertia, Rotor Dynamic Friction
- Loads – Model Name, Load Type, Rated Power, Minimum Speed, Maximum Speed, Load Inertia, Load Dynamic Friction, Load Static Friction

3.4.1 Boards

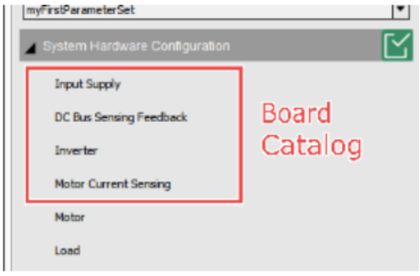
Board catalogs come in two different types: **PFC/Motor** related parameters, which are 02 devices, and **Only Motor/Non-PFC** parameters, which are 01 devices.
PFC boards will have catalogs that cover the following:

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

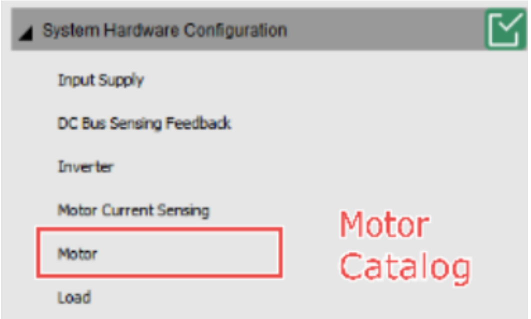


When a non-PFC board is being used, the catalog will cover the following:

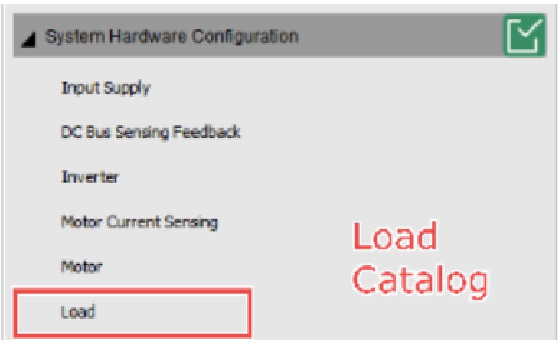
- Input Supply
- DC Bus Sensing Feedback
- Inverter
- Motor Current Sensing



3.4.2 Motors



3.4.3 Loads



3.5 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 issues.

3.5.1 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**.

3.5.1.1 Contents

- **Reference Manual MCE2** – This manual list any and all updates to the current MCE version. Users can see the raw information for many of the app's settings and functions.

- **iMOTION™ Solution Help** – Users can use this tab to get specific help in regards to their project and project parameters.

3.5.1.2 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.

3.5.1.3 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.

3.5.2 iMOTION™ Forum

Users can visit the iMOTION™ 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. **Screenshot of forum, more pop, more Marketable**

3.6 Project Verification

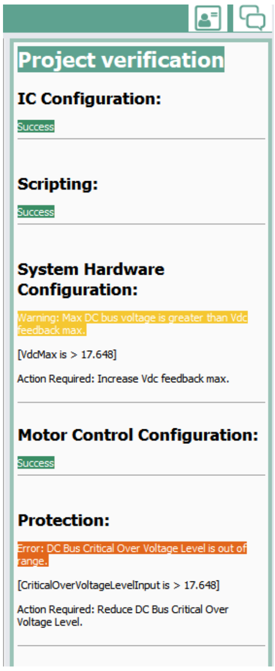
3.6.1 Initiating Verification

On each of the parameter groups, users will find the **Verify** function. Verify is a powerful tool that helps users find errors 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** 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.

3.6.2 Project Verification Information and Outcomes

3.6.2.1 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.



Project verification

IC Configuration:

Success

Scripting:

Success

System Hardware Configuration:

Warning: Max DC bus voltage is greater than Vdc feedback max.

[VdcMax is > 17.648]

Action Required: Increase Vdc feedback max.

Motor Control Configuration:

Success

Protection:

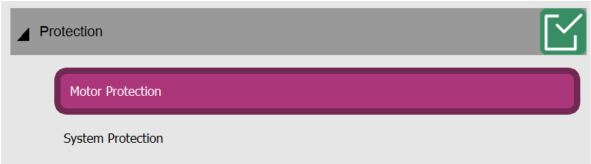
Error: DC Bus Critical Over Voltage Level is out of range.

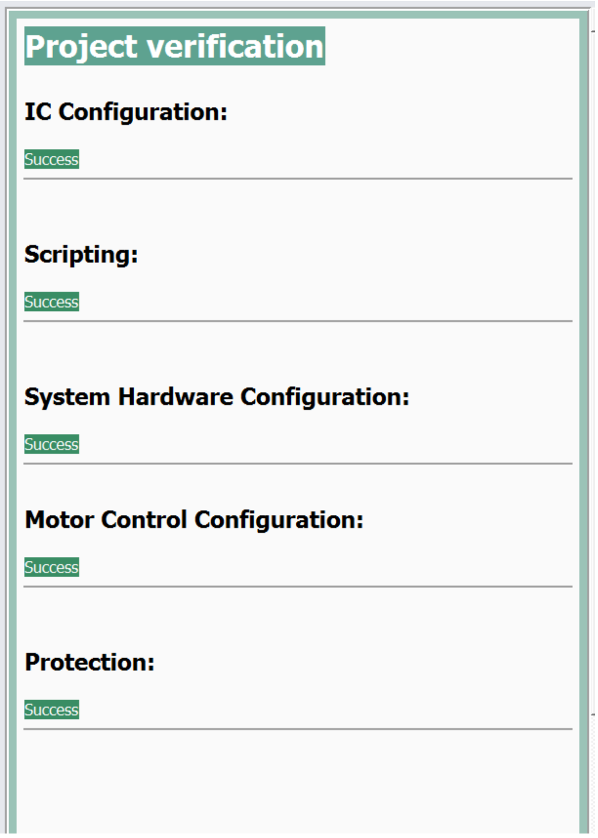
[CriticalOverVoltageLevelInput is > 17.648]

Action Required: Reduce DC Bus Critical Over Voltage Level.

3.6.2.2 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.





Project verification

IC Configuration:

Success

Scripting:

Success

System Hardware Configuration:

Success

Motor Control Configuration:

Success

Protection:

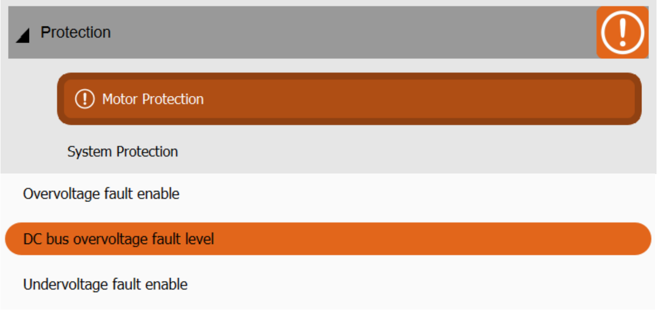
Success

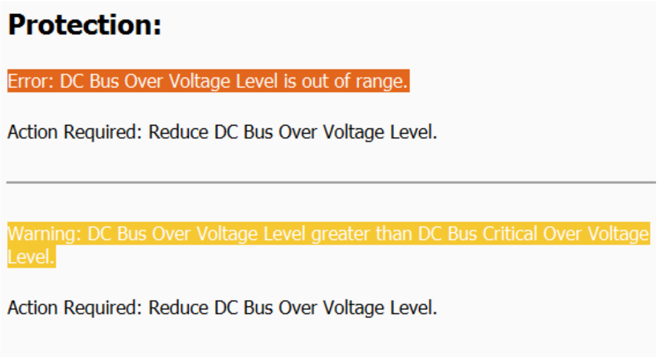
3.6.2.3 Warnings and Errors

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. An orange warning will **not** allow users to build their parameters and program their device, 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.





Protection:

Error: DC Bus Over Voltage Level is out of range.

Action Required: Reduce DC Bus Over Voltage Level.

Warning: DC Bus Over Voltage Level greater than DC Bus Critical Over Voltage Level.

Action Required: Reduce DC Bus Over Voltage Level.

3.7 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**.

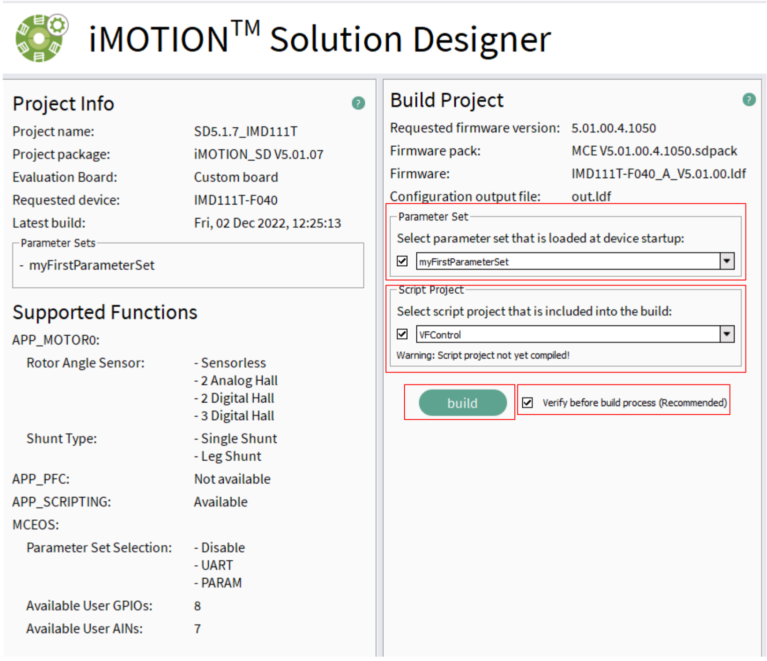
The build output **.ldf file** consists of the configured parameter set date and script binary file. This file is located with the project directory (Project Location -> generated) folder.

In the build process, only the script binary 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.

3.7.1 Build Process

1. To start, users will onfigure 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.



iMOTION™ Solution Designer

Project Info

Project name:	SD5.1.7_IMD111T
Project package:	iMOTION_SD V5.01.07
Evaluation Board:	Custom board
Requested device:	IMD111T-F040
Latest build:	Fri, 02 Dec 2022, 12:25:13

Parameter Sets
- myFirstParameterSet

Supported Functions

APP_MOTOR0:

Rotor Angle Sensor:	- Sensorless - 2 Analog Hall - 2 Digital Hall - 3 Digital Hall
Shunt Type:	- Single Shunt - Leg Shunt
APP_PFC:	Not available
APP_SCRIPTING:	Available
MCEOS:	
Parameter Set Selection:	- Disable - UART - PARAM
Available User GPIOs:	8
Available User AINs:	7

Build Project

Requested firmware version:	5.01.00.4.1050
Firmware pack:	MCE V5.01.00.4.1050.sdpack
Firmware:	IMD111T-F040_A_V5.01.00.ldf
Configuration output file:	out.ldf

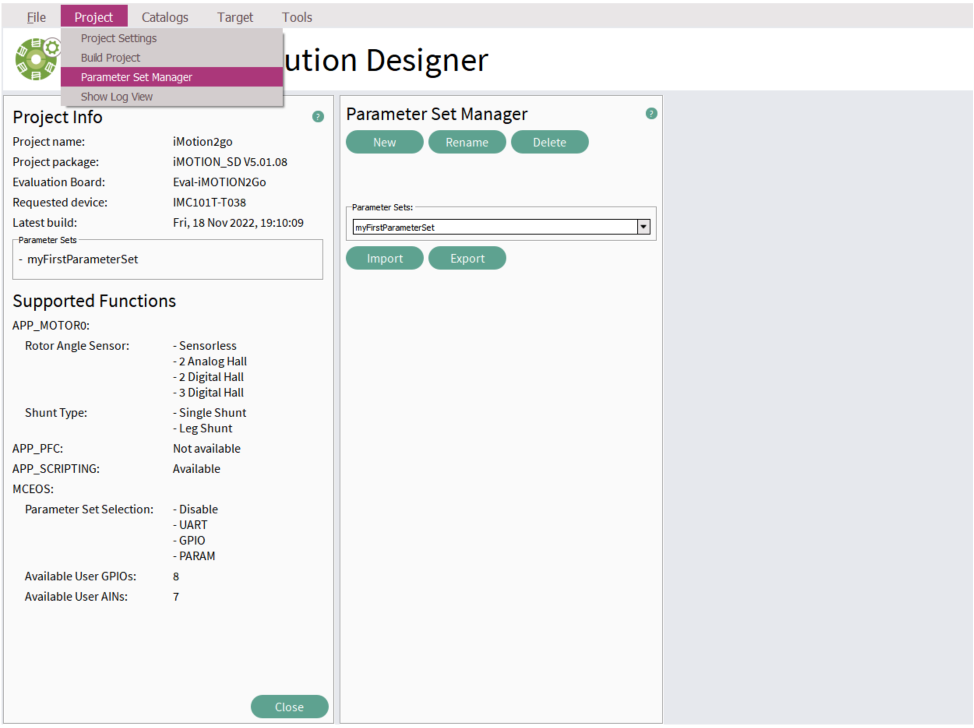
Parameter Set
Select parameter set that is loaded at device startup:
☑ myFirstParameterSet

Script Project
Select script project that is included into the build:
☑ VFControl
Warning: Script project not yet compiled!

☑ Verify before build process (Recommended)

3.8 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.



Project Info

Project name:	iMotion2go
Project package:	iMOTION_SD V5.01.08
Evaluation Board:	Eval-iMOTION2Go
Requested device:	IMC101T-T038
Latest build:	Fri, 18 Nov 2022, 19:10:09

Parameter Sets
- myFirstParameterSet

Supported Functions

APP_MOTOR0:

Rotor Angle Sensor:	- Sensorless - 2 Analog Hall - 2 Digital Hall - 3 Digital Hall
Shunt Type:	- Single Shunt - Leg Shunt
APP_PFC:	Not available
APP_SCRIPTING:	Available
MCEOS:	
Parameter Set Selection:	- Disable - UART - GPIO - PARAM
Available User GPIOs:	8
Available User AINs:	7

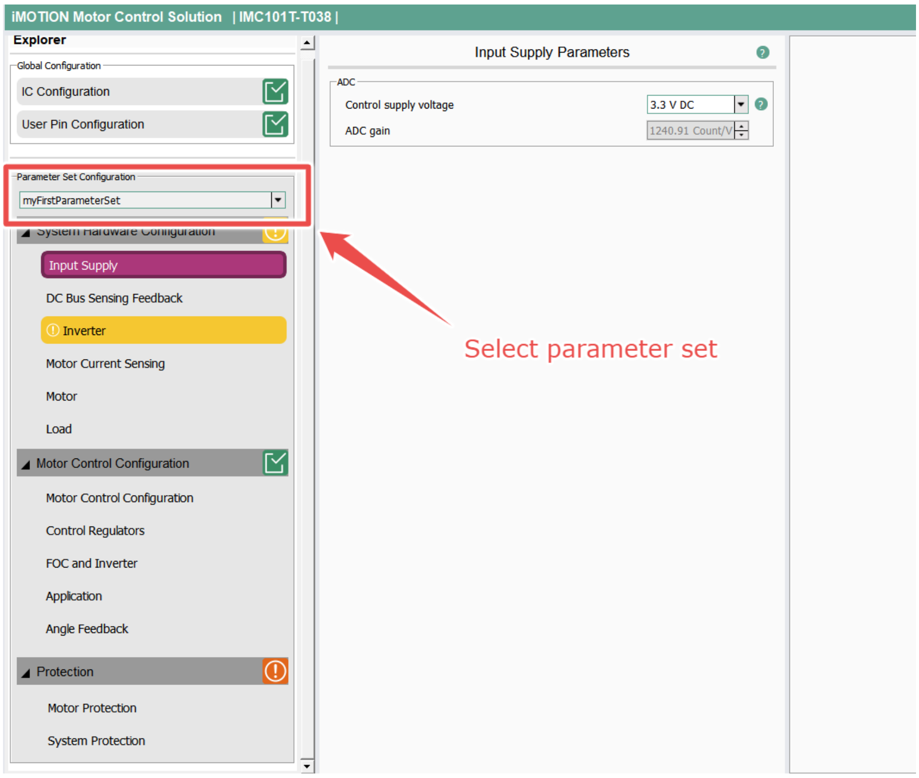
Parameter Set Manager

Parameter Sets:
myFirstParameterSet

3.8.1 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.



3.8.2 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.

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

3.8.3 Parameter File Location and Importing/Exporting Parameter Sets

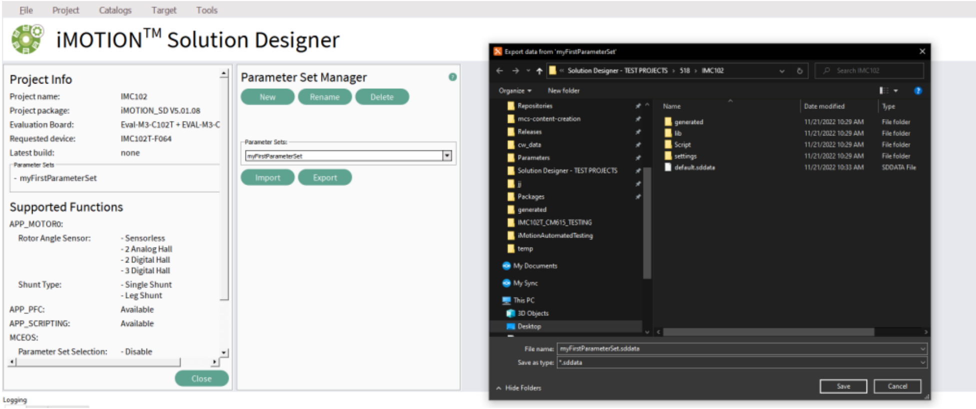
File Location

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.



Project Info

Project name: IMC102
Project package: iMOTION_SD V5.01.08
Evaluation Board: Eval-M3-C102T + EVAL-M3-C
Requested device: IMC102T-F064
Latest build: none
Parameter Sets
- myFirstParameterSet

Supported Functions

APP_MOTOR0:
Rotor Angle Sensor: - Sensorless - 2 Analog Hall - 2 Digital Hall - 3 Digital Hall
Shunt Type: - Single Shunt - Leg Shunt
APP_PFC: Available
APP_SCRIPTING: Available
MCEOS:
Parameter Set Selection: - Disable

Parameter Set Manager

Parameter Sets: myFirstParameterSet

Export data from 'myFirstParameterSet'

« Solution Designer - TEST PROJECTS › 518 › IMC102

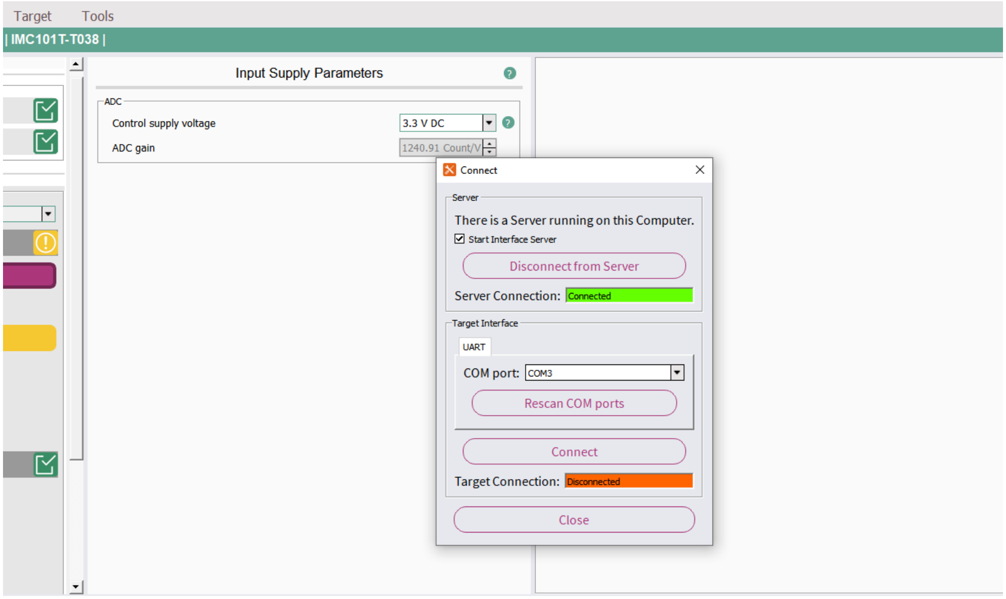
Name	Date modified	Type
generated	11/21/2022 10:29 AM	File folder
lib	11/21/2022 10:29 AM	File folder
Script	11/21/2022 10:29 AM	File folder
settings	11/21/2022 10:29 AM	File folder
default.sddata	11/21/2022 10:33 AM	SDDATA File

File name: myFirstParameterSet.sddata
Save as type: *.sddata

Sharing a Parameter Set

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

3.9 Target Connection



Control supply voltage: 3.3 V DC
ADC gain: 1240.91 Count/V

There is a Server running on this Computer.
☑ Start Interface Server
Server Connection: Connected
COM port: COM3
Target Connection: Disconnected

3.9.1 When to Connect to a Target

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

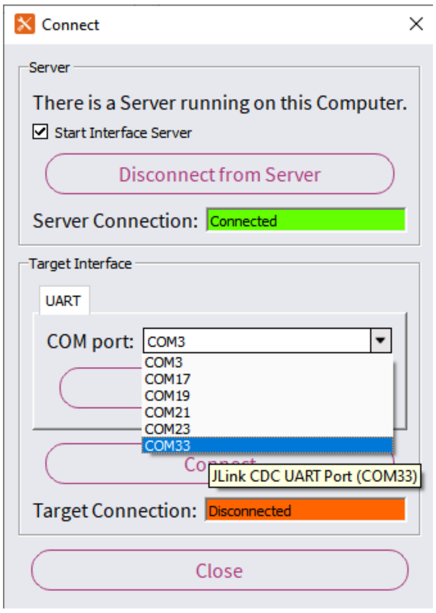
- Build Project
- Dashboard
- Script Editor

3.9.2 How to Connect

Before starting, users should ensure their firmware version is newer than 5.0. If this is the user's first time connecting the device, they should 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-out window, under **Target Interface**, select the appropriate COM port for your 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.



Connect

Server

There is a Server running on this Computer.

☑ Start Interface Server

Disconnect from Server

Server Connection: Connected

Target Interface

UART

COM port: COM3

COM3
COM17
COM19
COM21
COM23
COM33

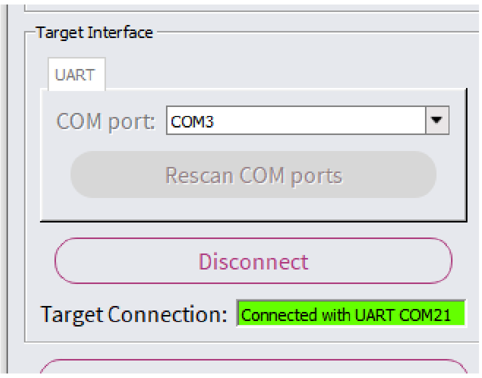
JLink CDC UART Port (COM33)

Connect

Target Connection: Disconnected

Close

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



Target Interface

UART

COM port: COM3

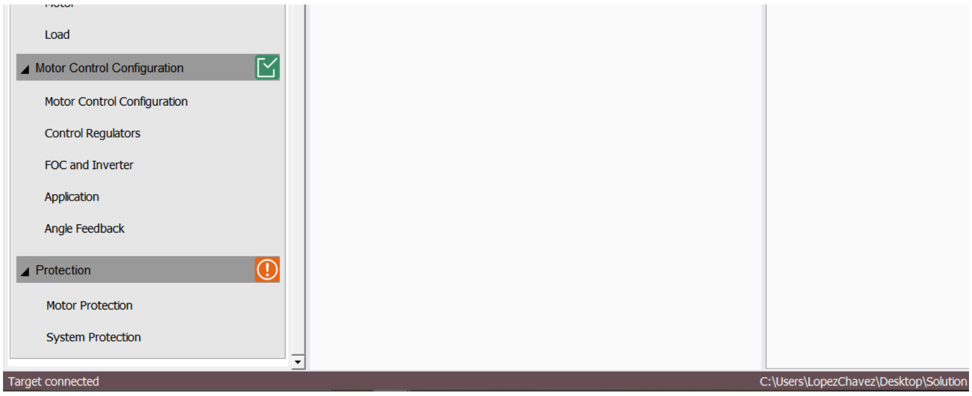
Rescan COM ports

Disconnect

Target Connection: Connected with UART COM21

3.9.3 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**.



Target connected

C:\Users\LopezChavez\Desktop\Solution

3.10 Log View

3.10.1 Using the Log when Verifying, Building, and Programming

Users can use the Log View in conjunction with several situations and tools:

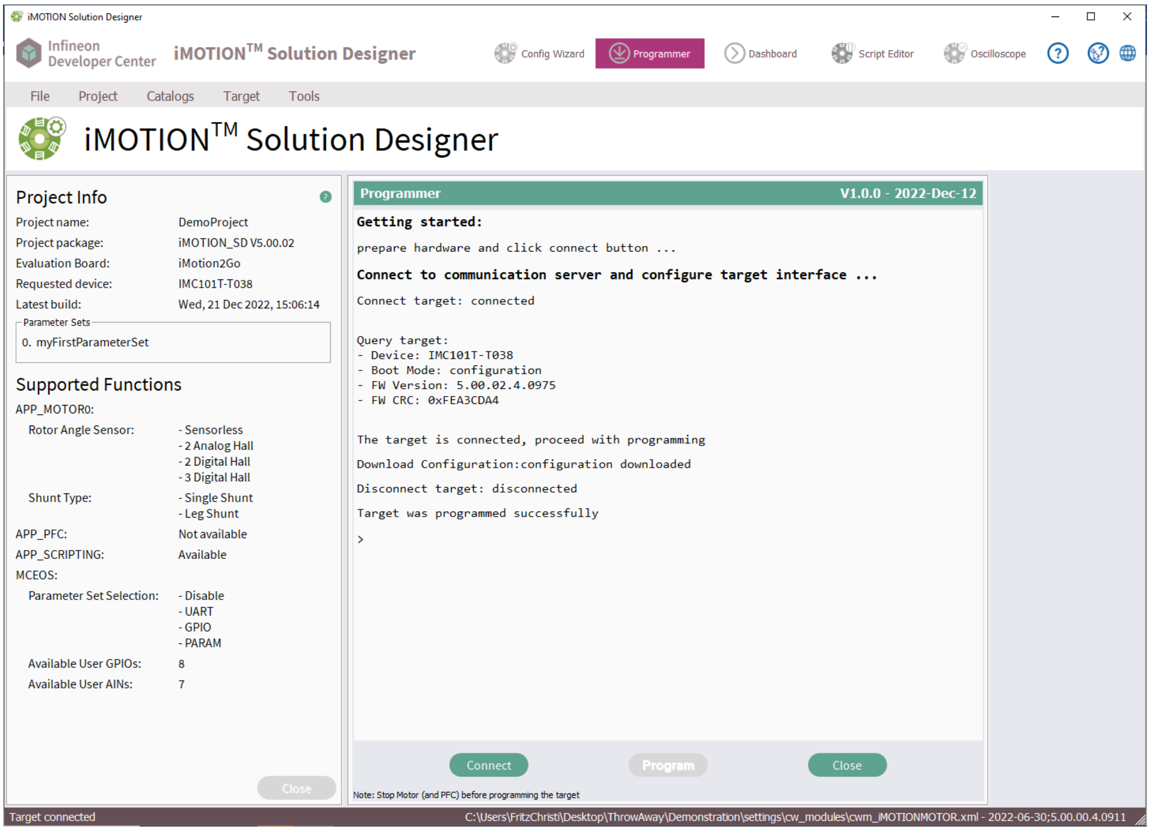
- When validating parameters and observing if there is a count-out-of-range error.
- When experiencing connection issues.
- When encountering building or programming issues.
- Observing when a file has been saved.

4 Programmer

The **Programmer** allows users to deploy firmware, parameters, and script code to iMOTION™ hardware. 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 useful status information and pop-ups inform about the progress of the connection or download process.



Project name:	DemoProject
Project package:	iMOTION_SD V5.00.02
Evaluation Board:	iMotion2Go
Requested device:	IMC101T-T038
Latest build:	Wed, 21 Dec 2022, 15:06:14

0. myFirstParameterSet

APP_MOTOR0:	
Rotor Angle Sensor:	- Sensorless - 2 Analog Hall - 2 Digital Hall - 3 Digital Hall
Shunt Type:	- Single Shunt - Leg Shunt
APP_PFC:	Not available
APP_SCRIPTING:	Available
MCEOS:	
Parameter Set Selection:	- Disable - UART - GPIO - PARAM
Available User GPIOs:	8
Available User AINs:	7

```
Getting started:

prepare hardware and click connect button ...

Connect to communication server and configure target interface ...

Connect target: connected


Query target:
- Device: IMC101T-T038
- Boot Mode: configuration
- FW Version: 5.00.02.4.0975
- FW CRC: 0xFEA3CDA4


The target is connected, proceed with programming

Download Configuration:configuration downloaded

Disconnect target: disconnected

Target was programmed successfully

>
```

Note: Stop Motor (and PFC) before programming the target

Target connected C:\Users\FritzChristi\Desktop\ThrowAway\Demonstration\settings\cw_modules\cwm_iMOTIONMOTOR.xml - 2022-06-30;5.00.00.4.0911

4.1 Special Use Case: Upgrading and Downgrading Firmware

For users who want to leverage iMOTION™ hardware that has previously been programmed and has existing firmware on it, there are two special use cases that require a dedicated firmware up- or downgrade procedure.

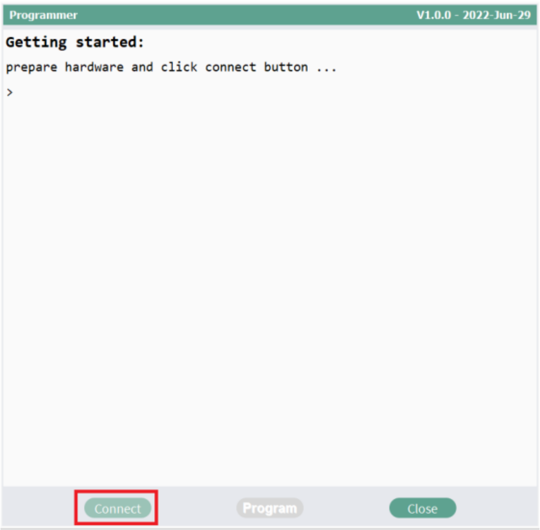
4.1.1 Upgrading Firmware

Users who are using iMOTION™ hardware that has previously been used in combination with the MCE Designer and MCE Wizard, will need to upgrade the firmware in order to be able to use the device with the iSD.

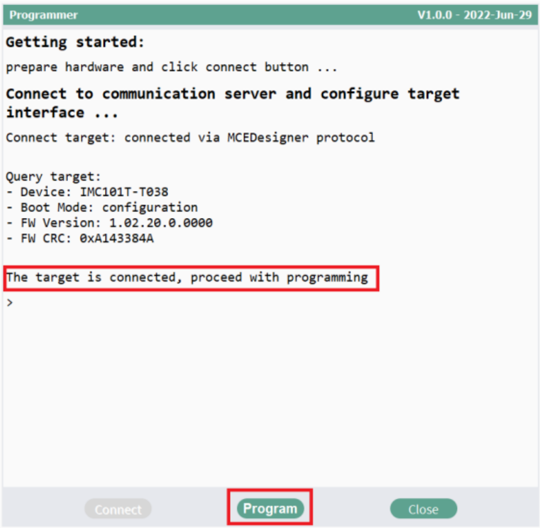
To upgrade the firmware, users can follow these steps:

1. Open the Solution Designer and connect with the device.
    a. **Note:** Ensure the firmware version is FW1.03.03 (or older).

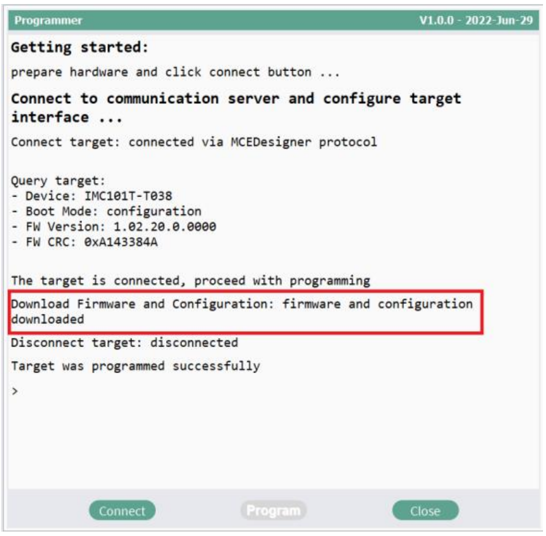
2. Because the Solution Designer cannot connect with the device when the firmware version is FW1.03.03, “Target not connected” will be displayed as below. Users can ignore this information and go straight to the next step.
3. Open the **Programmer**.
4. Follow the programming procedure as shown below. Users can then upgrade FW to the latest release (currently #5.1).



```
Getting started:
prepare hardware and click connect button ...
>
```



```
Getting started:
prepare hardware and click connect button ...
Connect to communication server and configure target
interface ...
Connect target: connected via MCEDesigner protocol

Query target:
- Device: IMC101T-T038
- Boot Mode: configuration
- FW Version: 1.02.20.0.0000
- FW CRC: 0xA143384A

The target is connected, proceed with programming
>
```



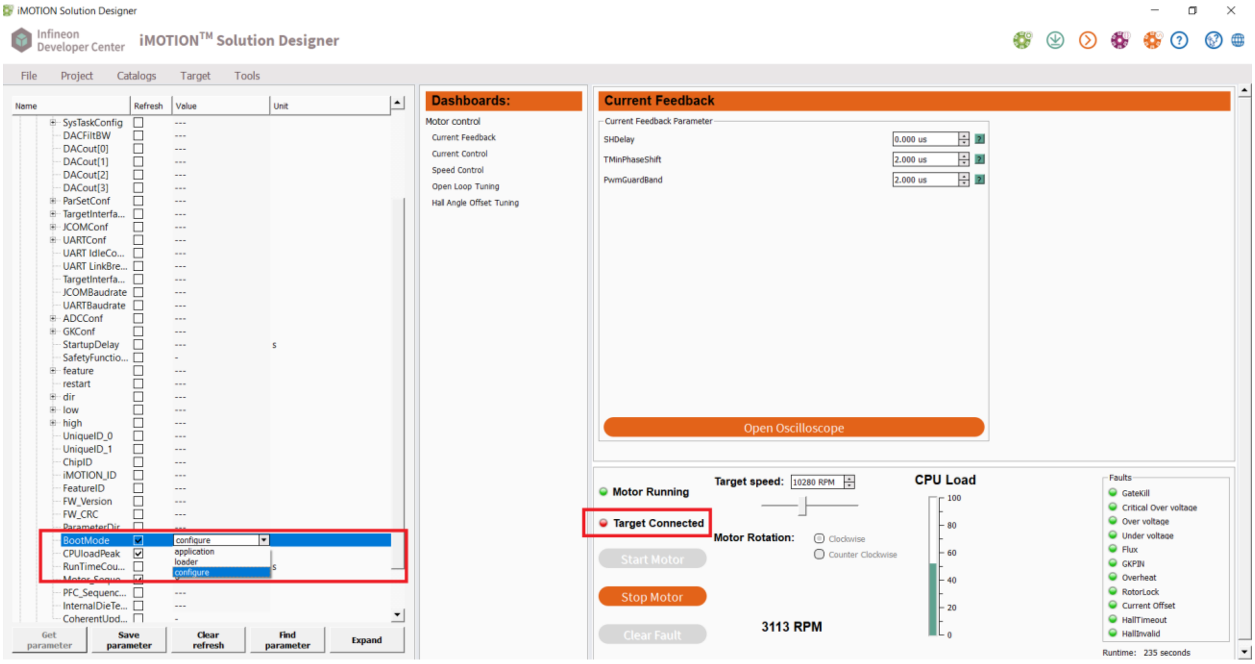
```
Getting started:
prepare hardware and click connect button ...
Connect to communication server and configure target
interface ...
Connect target: connected via MCEDesigner protocol

Query target:
- Device: IMC101T-T038
- Boot Mode: configuration
- FW Version: 1.02.20.0.0000
- FW CRC: 0xA143384A

The target is connected, proceed with programming
Download Firmware and Configuration: firmware and configuration
downloaded
Disconnect target: disconnected
Target was programmed successfully
>
```

4.1.2 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 the Solution Designer when connected with board. Users will then find “Target disconnected” displayed in the dashboard.



2. Ignore the warning information after opening the MCE Designer. Use the MCE Designer to download the firmware (FW1.03.03) with the parameter file.

3. The firmware will be successfully downgraded to FW1.03.03.

5 Dashboard

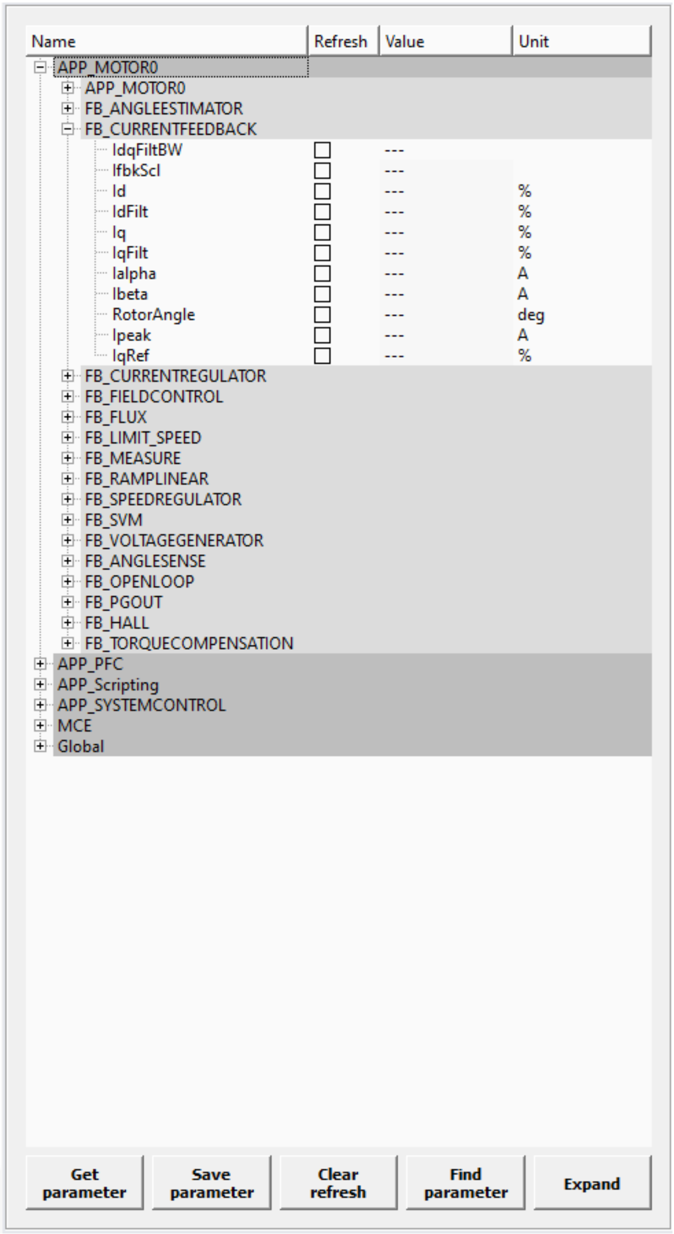
5.1 Introduction

The **Dashboard** tool can be used to monitor and edit the parameters during runtime. Furthermore, the Oscilloscope tool can be opened from any Dashboard. Depending on which Dashboard is active, different parameters and trigger settings will appear in the Oscilloscope.

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

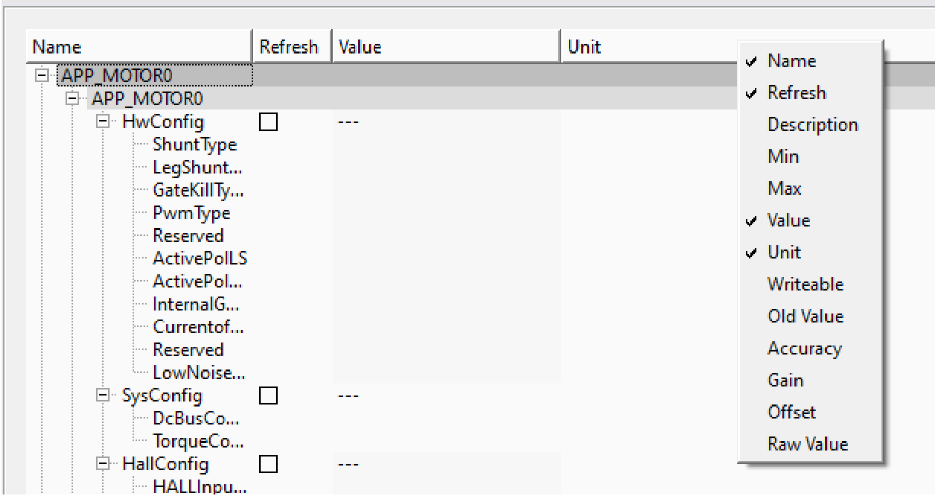
5.2 Parameter Tree

The **Parameter Tree** displays all available parameters of the target.



Name	Refresh	Value	Unit
⊟ APP_MOTOR0			
⊞ APP_MOTOR0			
⊞ FB_ANGLEESTIMATOR			
⊟ FB_CURRENTFEEDBACK			
IdqFiltBW	□	---	
IfbkScl	□	---	
Id	□	---	%
IdFilt	□	---	%
Iq	□	---	%
IqFilt	□	---	%
Ialpha	□	---	A
Ibeta	□	---	A
RotorAngle	□	---	deg
Ipeak	□	---	A
IqRef	□	---	%
⊞ FB_CURRENTREGULATOR			
⊞ FB_FIELDCONTROL			
⊞ FB_FLUX			
⊞ FB_LIMIT_SPEED			
⊞ FB_MEASURE			
⊞ FB_RAMPLINEAR			
⊞ FB_SPEEDREGULATOR			
⊞ FB_SVM			
⊞ FB_VOLTAGEGENERATOR			
⊞ FB_ANGLESENSE			
⊞ FB_OPENLOOP			
⊞ FB_PGOUT			
⊞ FB_HALL			
⊞ FB_TORQUECOMPENSATION			
⊞ APP_PFC			
⊞ APP_Scripting			
⊞ APP_SYSTEMCONTROL			
⊞ MCE			
⊞ Global			

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.



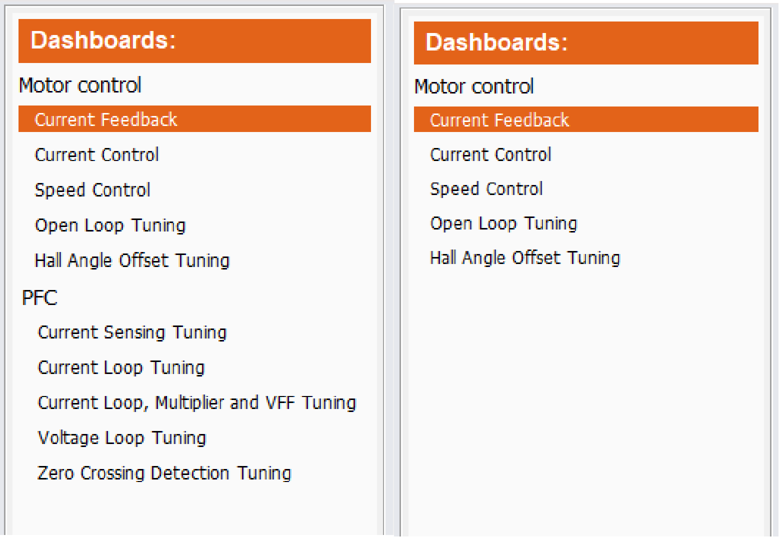
The buttons at the bottom of the pane allow users to quickly navigate and change their parameter settings:

- 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: Users will find some Refresh boxes are checked by default. They should be left as is.

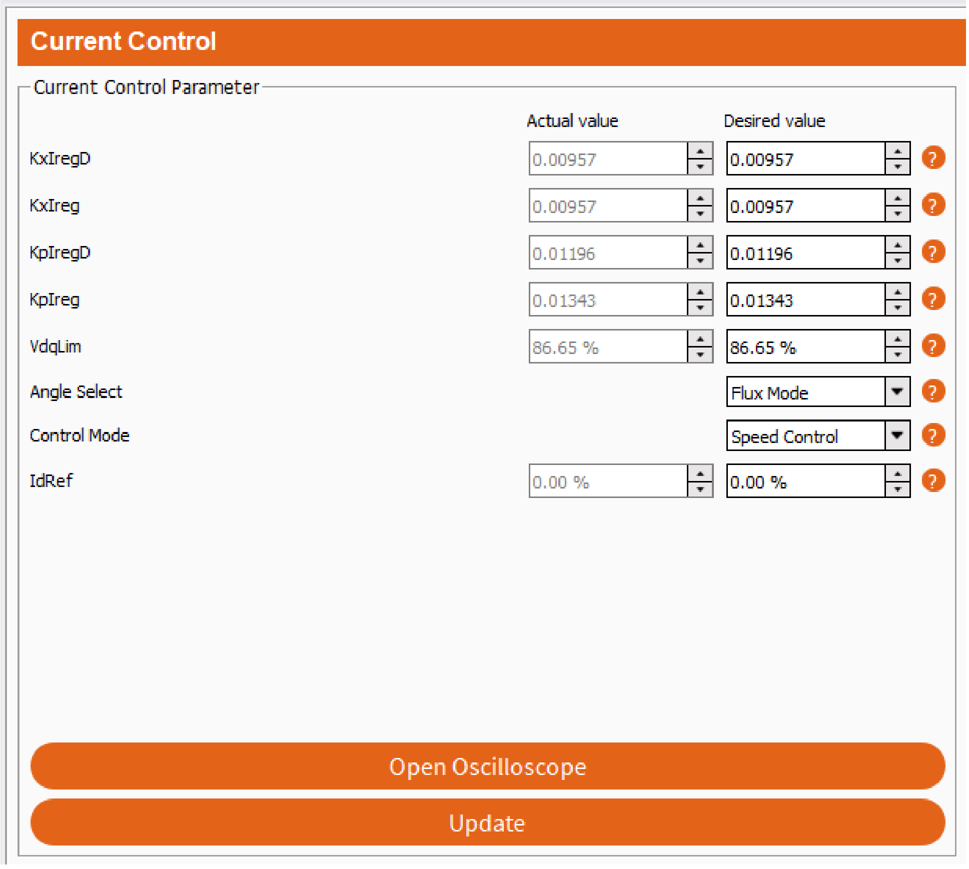
5.3 Dashboard Menu

The **Dashboard Menu** lists all available Motor or PFC control tuning use cases and relevant parameters / variables.



When a PFC compatible device is used, additional dashboard options will be available.

5.4 Dashboard Content



Current Control

Current Control Parameter

	Actual value	Desired value
KxIregD	0.00957	0.00957
KxIreg	0.00957	0.00957
KpIregD	0.01196	0.01196
KpIreg	0.01343	0.01343
VdqLim	86.65 %	86.65 %
Angle Select		Flux Mode
Control Mode		Speed Control
IdRef	0.00 %	0.00 %

Each dashboard displays different parameters, based on its purpose. Each parameter can be edited by a spin box or a drop-down box. In case of a spin box, in addition to an input spin box (**Desired value**) an additional output spin box (**Actual value**) is displayed

By clicking the **Update** button, the values from the Desired value spin boxes will be written to the RAM of the target. The Actual value spin boxes show the real value which has been written into the target.

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

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.

5.5 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.

5.5.1 Input Features

- Start/Stop Motor: Sending Start/Stop motor command to the target
- Start/Stop 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

5.5.2 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 error
  - Red: error
- PFC Fault LEDs
  - Green: no error
  - Red: error

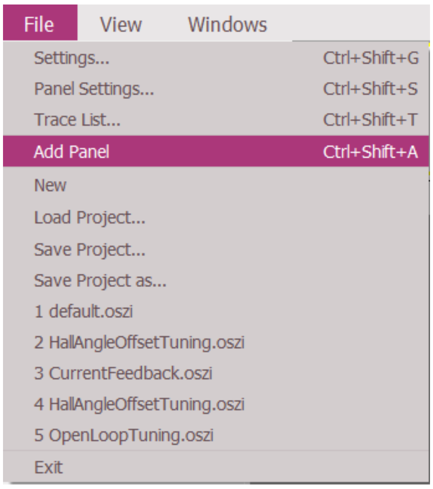
6 Oscilloscope

6.1 Panel

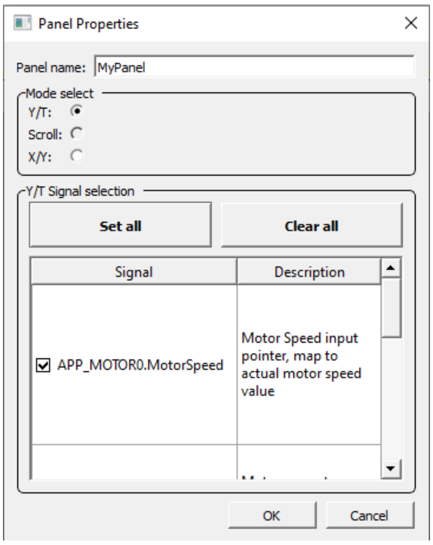
Users will initially see two panels when opening the Oscilloscope tool. Panels are containers used for signal monitoring, both as a group or alone. Users are able to create multiple panels.

6.1.1 Adding a Panel.

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



A pop-out pane will appear. Users will then fill out the required panel information, including the name of the project and which signals will be monitored by 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.



Panel Properties

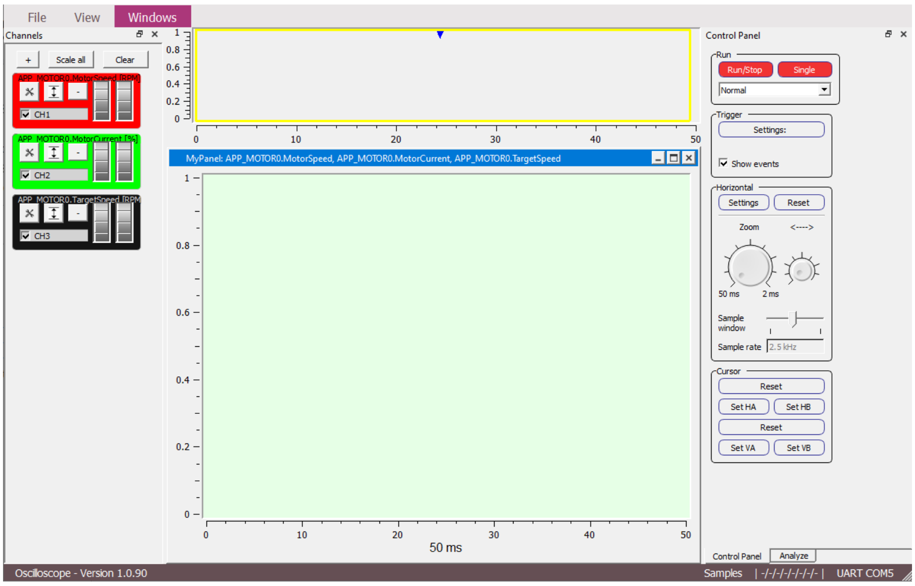
Panel name: MyPanel

Mode select
Y/T: ●
Scroll: ○
X/Y: ○

Y/T Signal selection

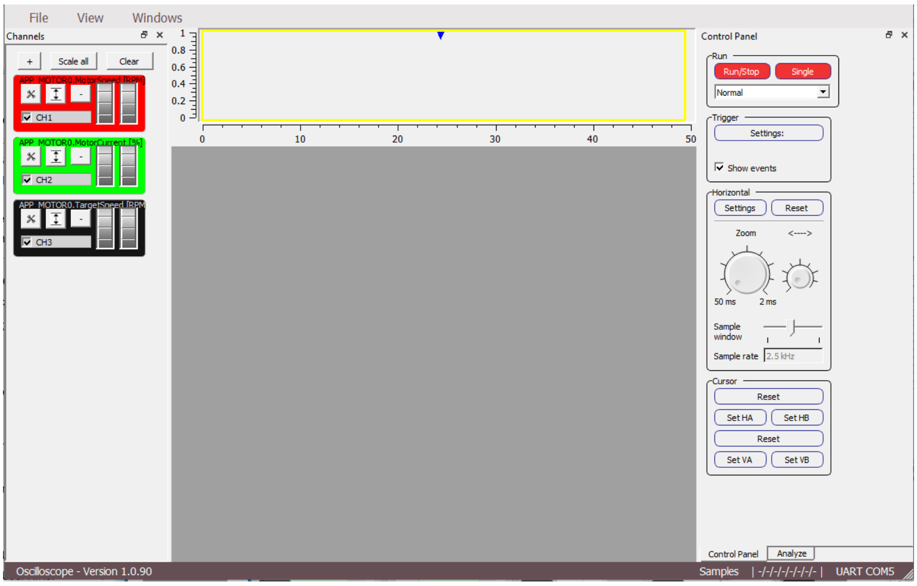
Signal	Description
☑ APP_MOTOR0.MotorSpeed	Motor Speed input pointer, map to actual motor speed value

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.



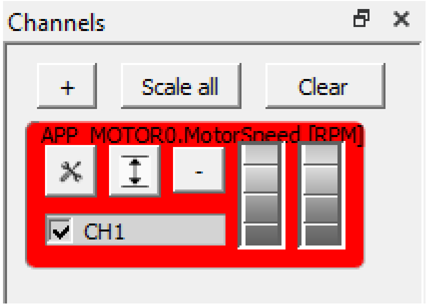
6.1.2 Removing a Panel

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



6.2 Channel Setup

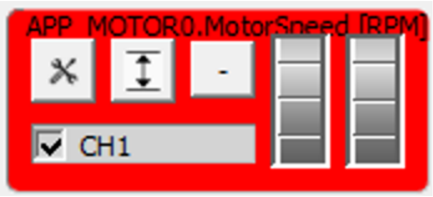
Channels are used to monitor a specific motor's signal. Users can adjust channels with the **Add Channel** +, **Scale All**, and **Clear** buttons.



- + - 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 the plots of data in all channels.

6.2.1 Channel Components

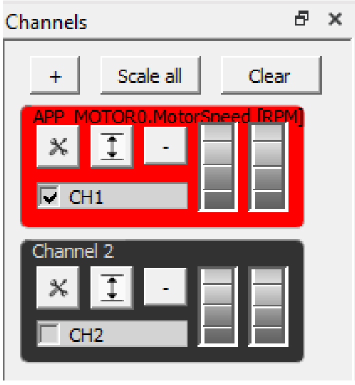
Within the individual channel tabs are several functions users can use to configure the individual channels.



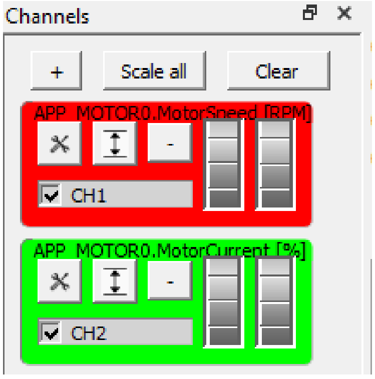
- ⚒ - **Signal Settings** is used to configure various properties within the channel (Label, Signal, Color, etc).
- ↕ - **Autoscale** will fit the signal to the plotting area automatically.
- - - **Remove Channel** will delete the associated channel.
- ☑ CH1 - This button will allow users to enable or disable the signals ability to capture the associated channel.
- ▤ - **Gain Adjustment** is the first wheel from left to right, and **Offset Adjustment** is the second wheel from left to right.

6.2.2 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.

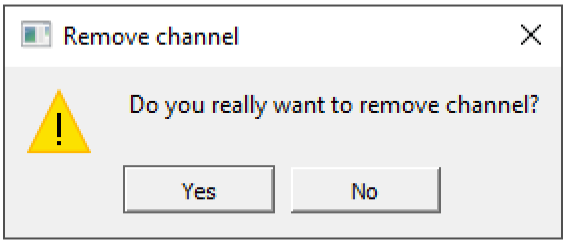


Newly created channels will be colored dark gray and will lack an associated signal.



6.2.3 Removing a Channel

Users can remove a channel by clicking on the **Remove Channel** - button. A pop-out 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.

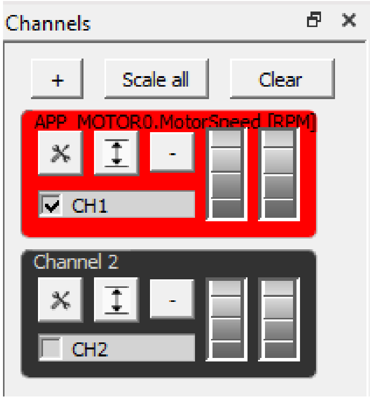


Do you really want to remove channel?

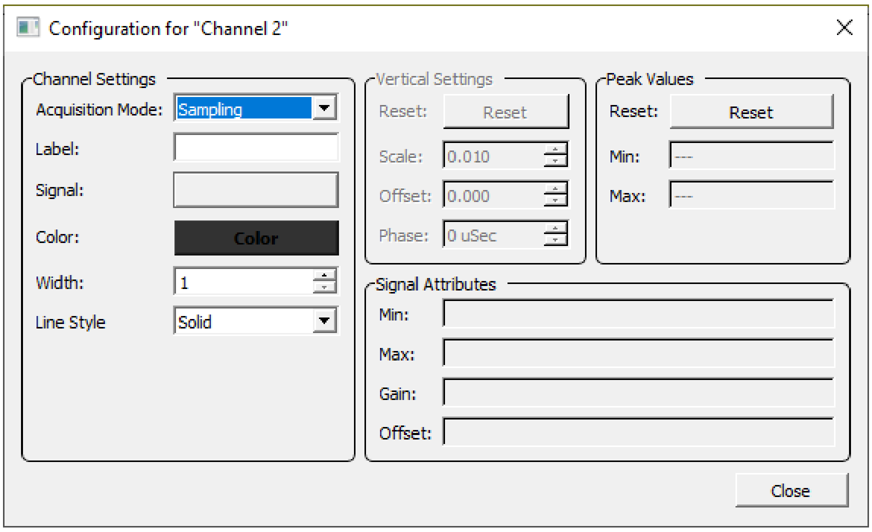
6.3 Signal

6.3.1 Adding a Signal

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



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



Configuration for "Channel 2"

Channel Settings

Acquisition Mode: Sampling

Label:

Signal:

Color: Color

Width: 1

Line Style: Solid

Vertical Settings

Reset: Reset

Scale: 0.010

Offset: 0.000

Phase: 0 uSec

Peak Values

Reset: Reset

Min: ---

Max: ---

Signal Attributes

Min:

Max:

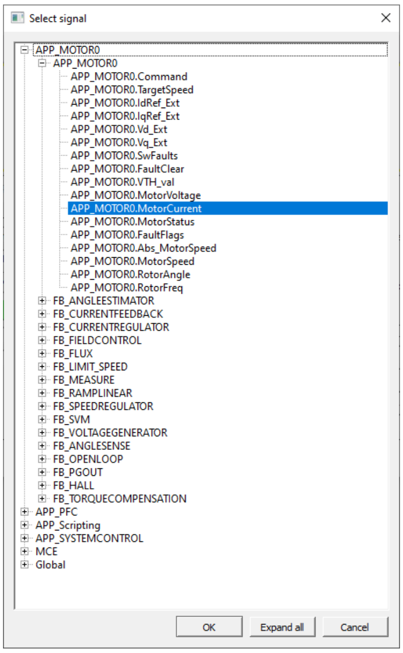
Gain:

Offset:

Close

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.

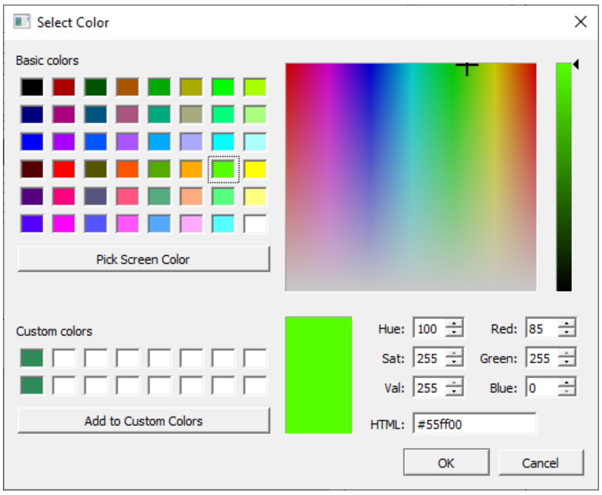
Clicking on the **Color** button will allow users to choose the display color of the associated channel. Users can choose from the basic selection of colors, or create their own to better suit their project. Once finished, users can confirm their selection by clicking **Close.**



Select signal

```
⊟ APP_MOTOR0
  ⊟ APP_MOTOR0
      APP_MOTOR0.Command
      APP_MOTOR0.TargetSpeed
      APP_MOTOR0.IdRef_Ext
      APP_MOTOR0.IqRef_Ext
      APP_MOTOR0.Vd_Ext
      APP_MOTOR0.Vq_Ext
      APP_MOTOR0.SwFaults
      APP_MOTOR0.FaultClear
      APP_MOTOR0.VTH_val
      APP_MOTOR0.MotorVoltage
      APP_MOTOR0.MotorCurrent
      APP_MOTOR0.MotorStatus
      APP_MOTOR0.FaultFlags
      APP_MOTOR0.Abs_MotorSpeed
      APP_MOTOR0.MotorSpeed
      APP_MOTOR0.RotorAngle
      APP_MOTOR0.RotorFreq
  ⊞ FB_ANGLEESTIMATOR
  ⊞ FB_CURRENTFEEDBACK
  ⊞ FB_CURRENTREGULATOR
  ⊞ FB_FIELDCONTROL
  ⊞ FB_FLUX
  ⊞ FB_LIMIT_SPEED
  ⊞ FB_MEASURE
  ⊞ FB_RAMPLINEAR
  ⊞ FB_SPEEDREGULATOR
  ⊞ FB_SVM
  ⊞ FB_VOLTAGEGENERATOR
  ⊞ FB_ANGLESENSE
  ⊞ FB_OPENLOOP
  ⊞ FB_PGOUT
  ⊞ FB_HALL
  ⊞ FB_TORQUECOMPENSATION
⊞ APP_PFC
⊞ APP_Scripting
⊞ APP_SYSTEMCONTROL
⊞ MCE
⊞ Global
```

OK | Expand all | Cancel



Select Color

Basic colors

Pick Screen Color

Custom colors

Add to Custom Colors

Hue: 100

Sat: 255

Val: 255

Red: 85

Green: 255

Blue: 0

HTML: #55ff00

OK | Cancel

6.3.2 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.

6.3.3 Changing a Signal

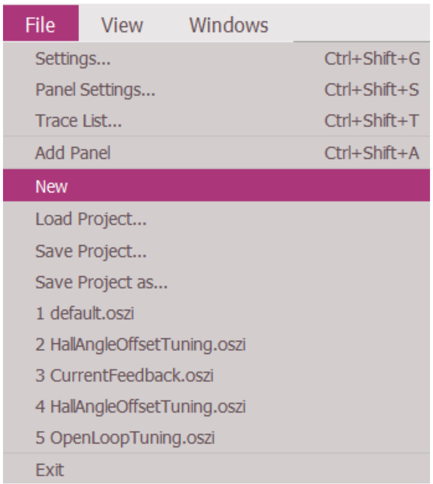
Users can change the signal using the Add Signal configuration window and Signal button within the configuration window. The oscilloscope must not be in capture mode when changing the signal.

6.4 Oscilloscope Project Files

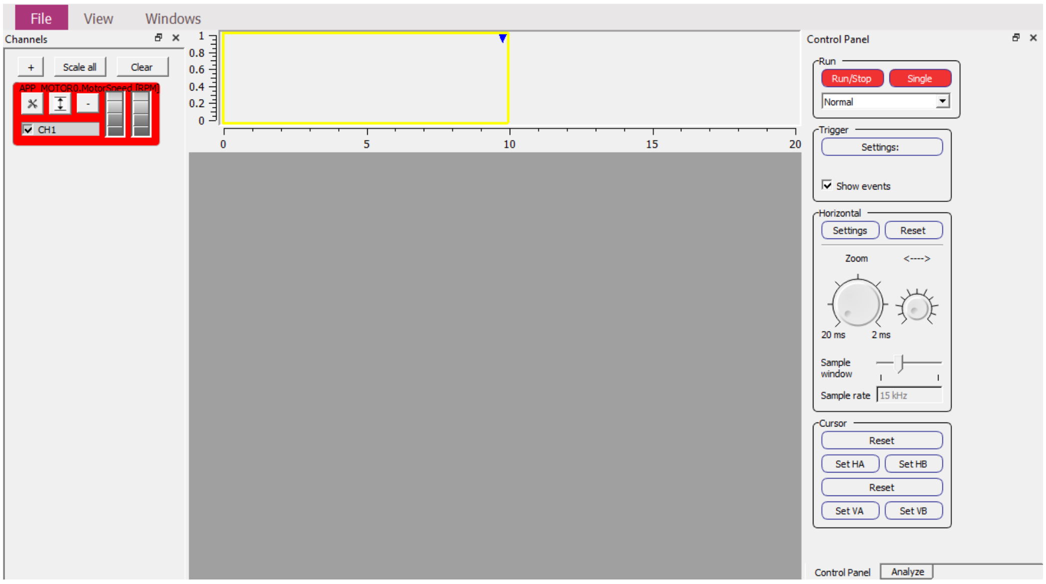
Users can find the oscilloscope project files inside the **Settings** folder of the project directory. The files are in .ozsi format and contain the channel, signal, and trigger settings, as well as other valuable project information.

6.4.1 Creating an Oscilloscope Project File

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

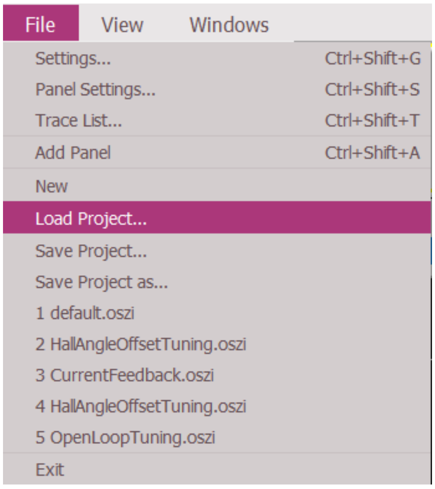


The tool will create a blank oscilloscope project. Users can continue generating the project by creating a panel (**6.1**), creating channels (**6.2**), and creating signals for monitoring (**6.3**).



6.4.2 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.

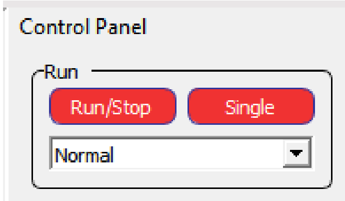


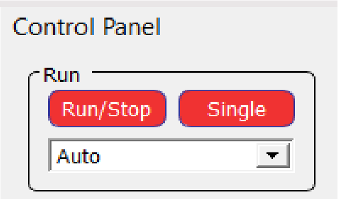
6.5 Control Panel

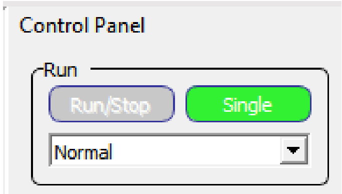
The control panel allows users to quickly change the active function settings of the oscilloscope.

6.5.1 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.







6.5.1.1 Trigger Modes

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

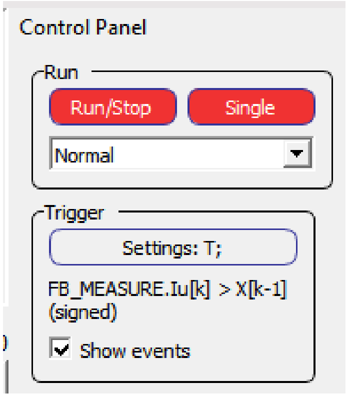
- **Normal Mode** – In this mode the oscilloscope will only display the signal capture if it meets the trigger point. The panel will otherwise be blank as it awaits the trigger or the panel will display the last captured waveform.

- **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.

6.6 Triggers

6.6.1 Trigger

The trigger control allows users to capture single-shot waveforms. Trigger settings and conditions will be shown in the Trigger Section of the Control Panel pane.



Normal

Settings: T;

FB_MEASURE.Iu[k] > X[k-1] (signed)

☑ Show events

6.6.1.1 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.

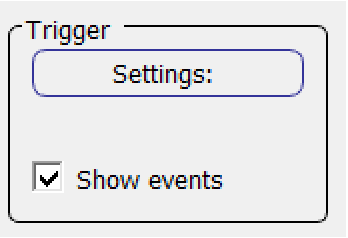
- **Edge Triggering** will have a trigger point that occurs on the rising edge or falling edge of the signal.
- **Level Triggering** will have a trigger point that occurs on a pre-determined point.
- **Comparison Triggering** will ha a trigger point that occurs where both signals meet the trigger condition.

6.6.1.2 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.

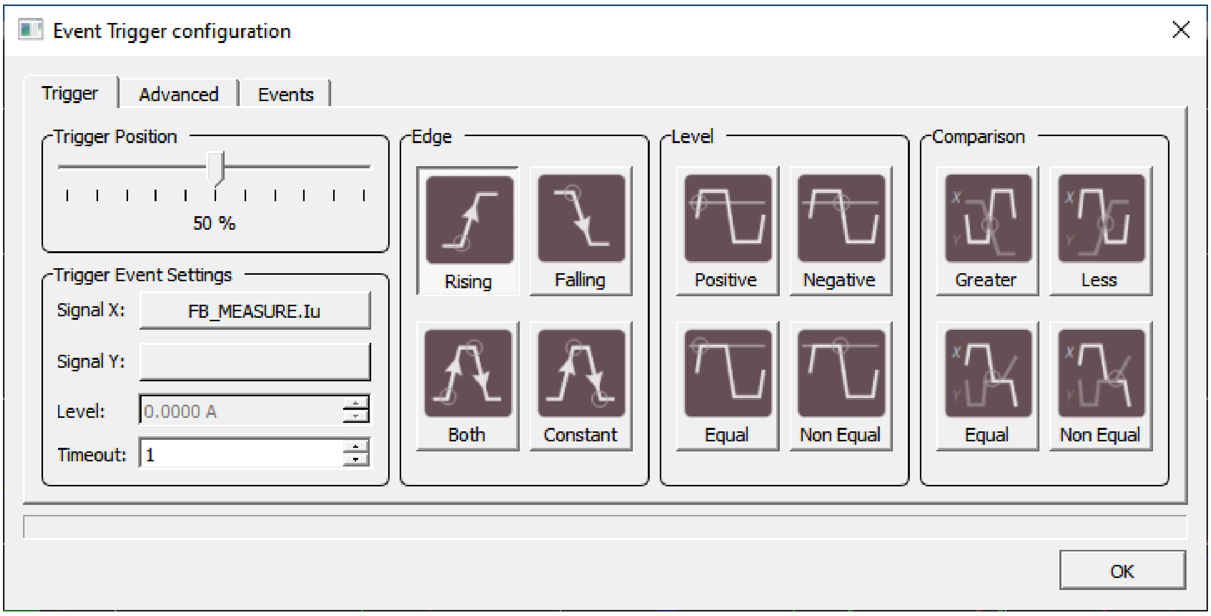
6.6.2 Creating an Edge Trigger

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



☑ Show events

This will open the Event Trigger Configuration pop-out window. Select the proper Edge Trigger for your project 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.



Event Trigger configuration

Trigger Position: 50 %

Trigger Event Settings

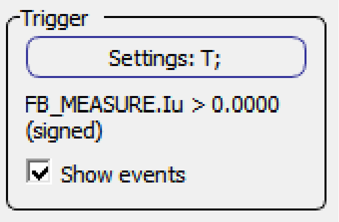
Signal X: FB_MEASURE.Iu

Signal Y:

Level: 0.0000 A

Timeout: 1

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



Trigger

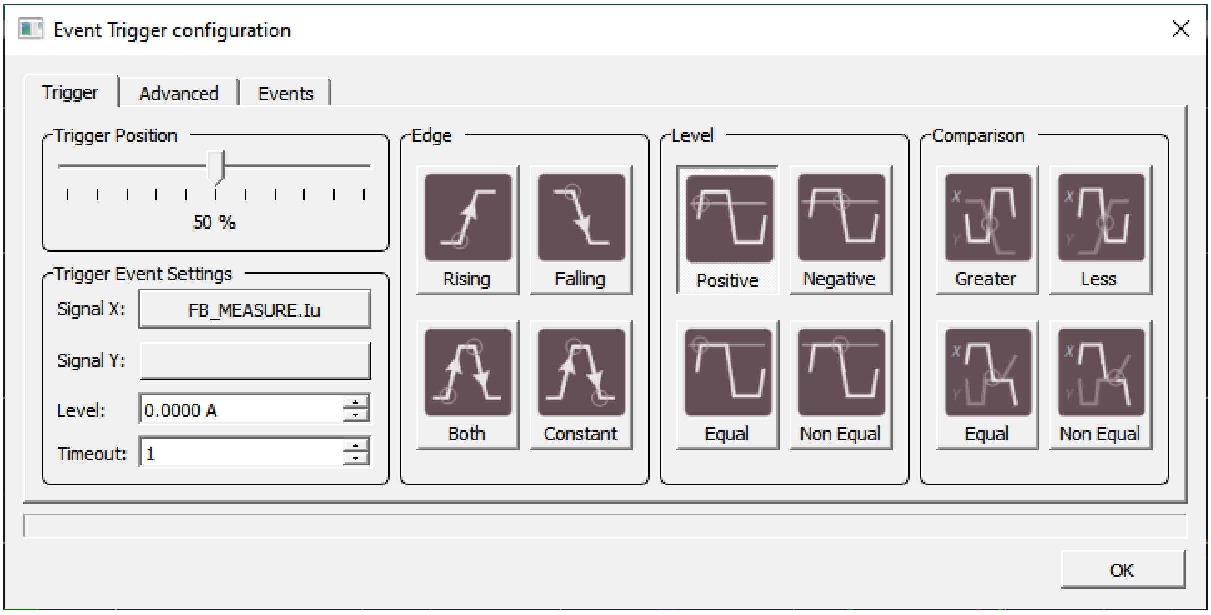
Settings: T;

FB_MEASURE.Iu > 0.0000 (signed)

☑ Show events

6.6.3 Creating a Level Trigger

Users will navigate to the Event Trigger Configuration window. Select the proper Level Trigger for your project 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.



Event Trigger configuration

Trigger Position: 50 %

Trigger Event Settings

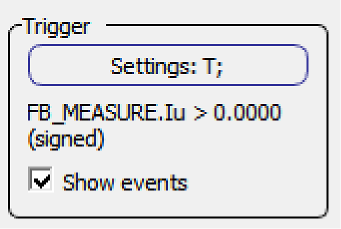
Signal X: FB_MEASURE.Iu

Signal Y:

Level: 0.0000 A

Timeout: 1

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



Trigger

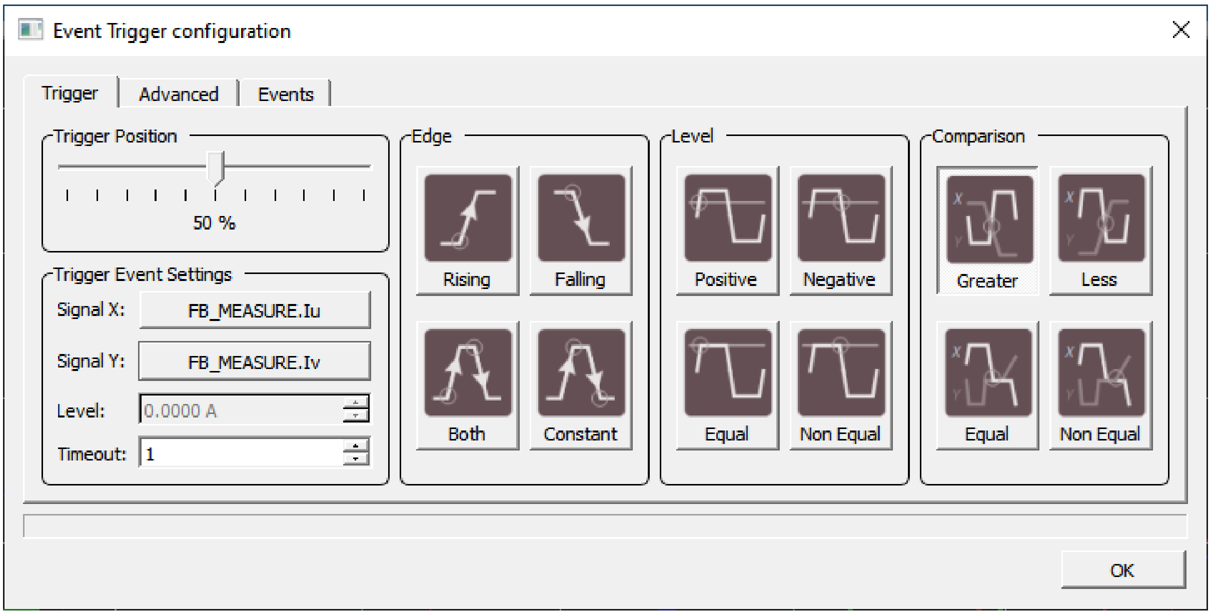
Settings: T;

FB_MEASURE.Iu > 0.0000 (signed)

☑ Show events

6.6.4 Creating a Comparison Trigger

Users will navigate to the Event Trigger Configuration window. Select the proper Comparison Trigger for your project 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.



Event Trigger configuration

Trigger Position: 50 %

Trigger Event Settings

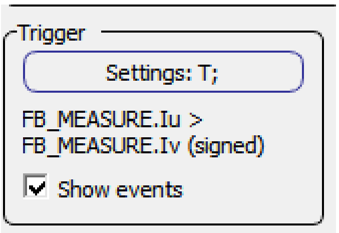
Signal X: FB_MEASURE.Iu

Signal Y: FB_MEASURE.Iv

Level: 0.0000 A

Timeout: 1

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



Trigger

Settings: T;

FB_MEASURE.Iu > FB_MEASURE.Iv (signed)

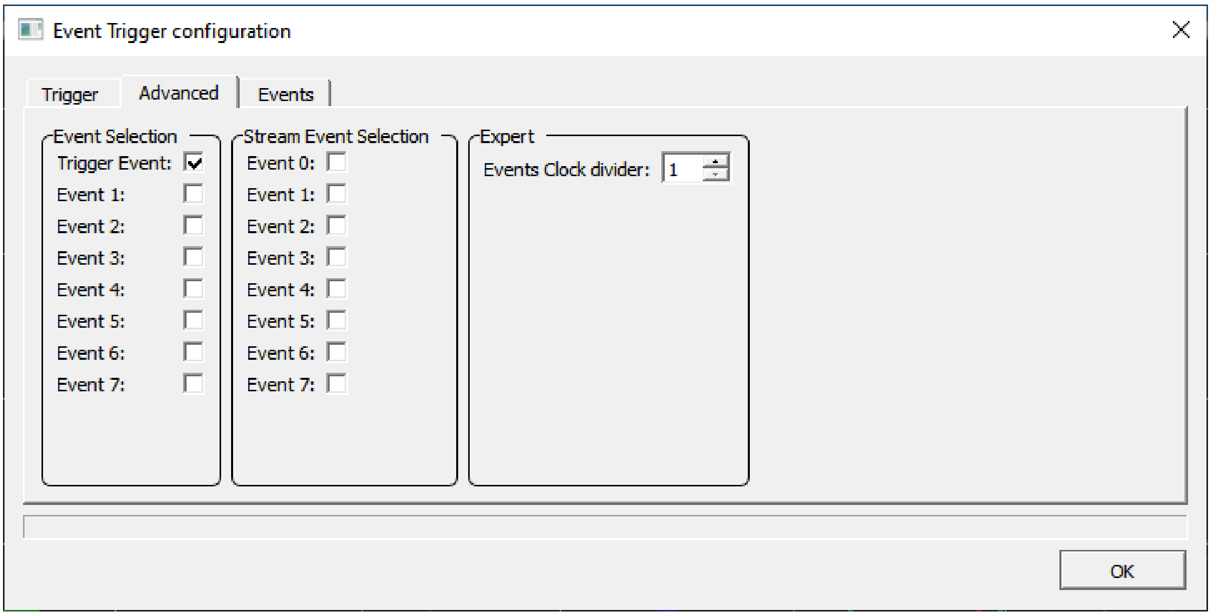
☑ Show events

6.6.5 Removing a Trigger

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

6.6.6 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.



Event Selection

Trigger Event: ☑
Event 1: □
Event 2: □
Event 3: □
Event 4: □
Event 5: □
Event 6: □
Event 7: □

Stream Event Selection

Event 0: □
Event 1: □
Event 2: □
Event 3: □
Event 4: □
Event 5: □
Event 6: □
Event 7: □

Expert

Events Clock divider: 1

6.7 Events

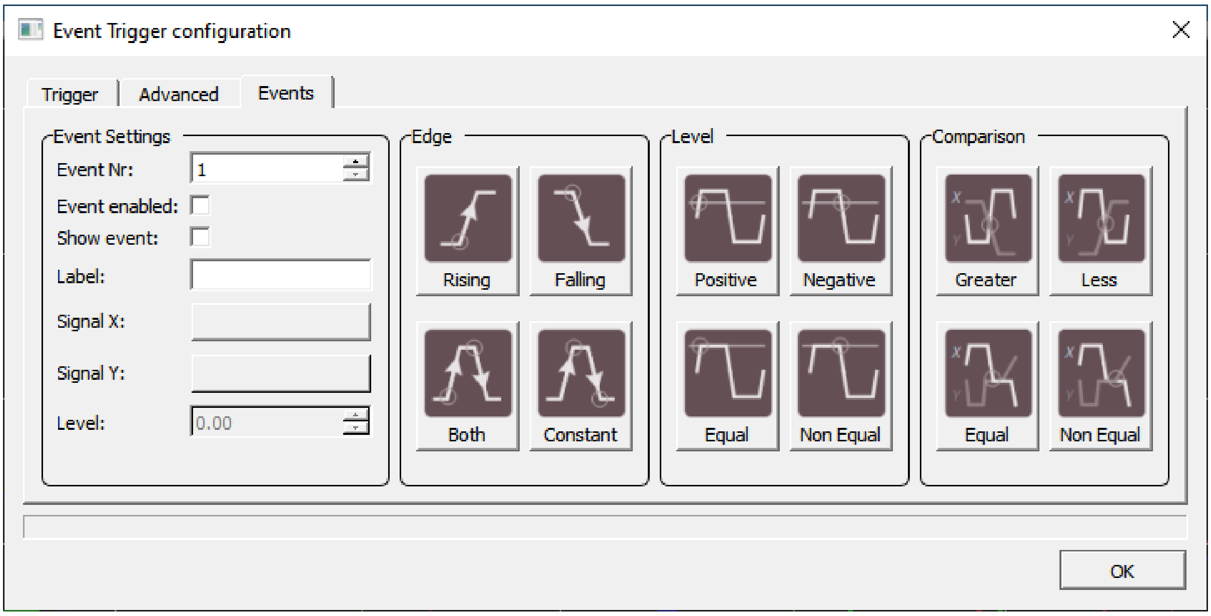
Users can create events in the Events tab of the Event Trigger Configuration pop-out window. Edge, Level, and Comparison events are supported, and each category follows the same rules as its trigger counterpart. Users can create up to 7 events.

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%.

6.7.1 Creating an Event

Users can create an event using the following steps:

1. Users can start by setting the Event Number (Event Nr). This acts as the events identifier.
2. Users will then check Show Event.
3. Users must change the Label to the appropriate setting.
4. Afterwards, users will 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, users must set the Level to the appropriate number.



Event Trigger configuration

Trigger | Advanced | Events

Event Settings

Event Nr: 1

Event enabled: □

Show event: □

Label:

Signal X:

Signal Y:

Level: 0.00

6.7.2 Removing an Event

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

7 Script Editor

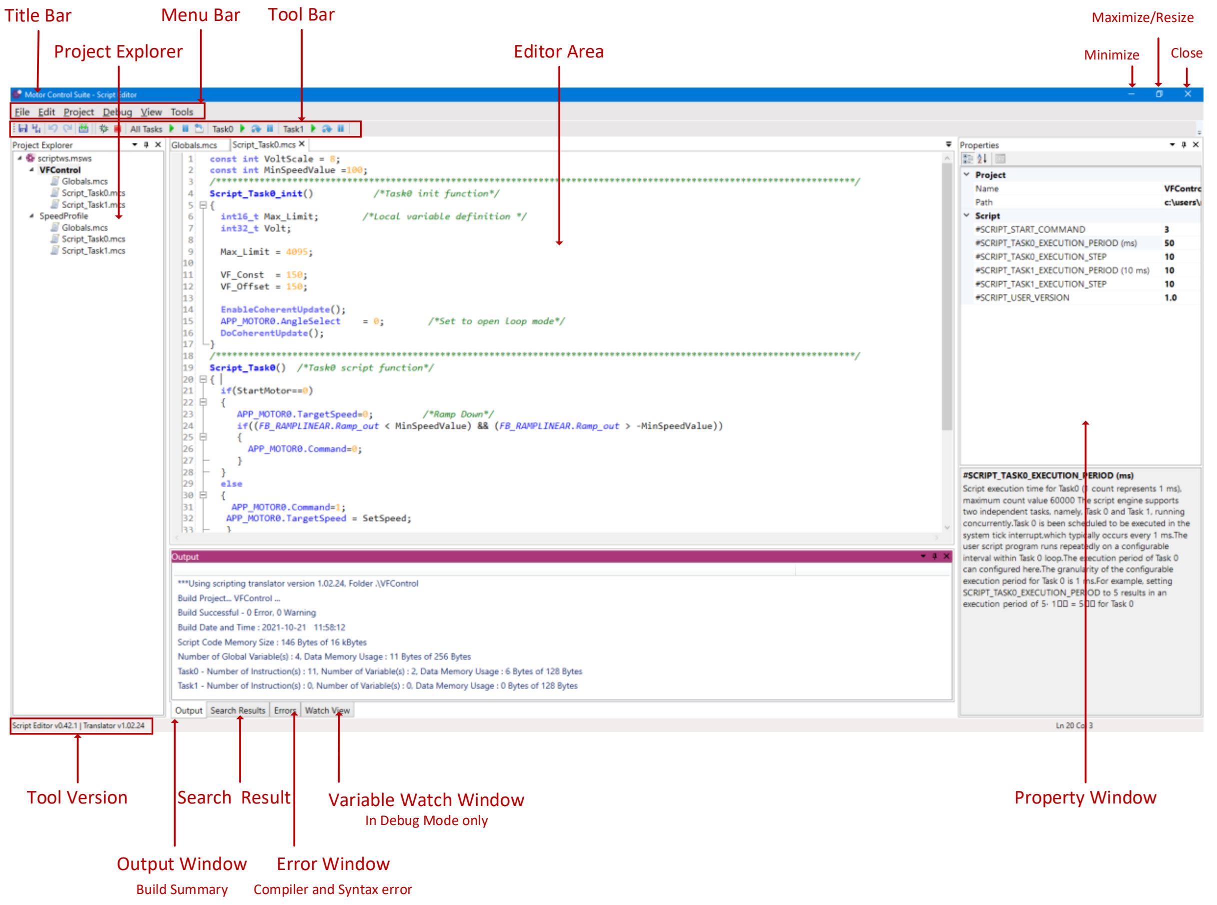
7.1 Overview

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.

7.2 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.



```
const int VoltScale = 8;
const int MinSpeedValue =100;
/*******************************************************************************************/
Script_Task0_init()          /*Task0 init function*/
{
  int16_t Max_Limit;        /*Local variable definition */
  int32_t Volt;

  Max_Limit = 4095;

  VF_Const  = 150;
  VF_Offset = 150;

  EnableCoherentUpdate();
  APP_MOTOR0.AngleSelect    = 0;        /*Set to open loop mode*/
  DoCoherentUpdate();
}
/*******************************************************************************************/
Script_Task0()  /*Task0 script function*/
{ |
  if(StartMotor==0)
  {
     APP_MOTOR0.TargetSpeed=0;         /*Ramp Down*/
     if((FB_RAMPLINEAR.Ramp_out < MinSpeedValue) && (FB_RAMPLINEAR.Ramp_out > -MinSpeedValue))
     {
       APP_MOTOR0.Command=0;
     }
  }
  else
  {
    APP_MOTOR0.Command=1;
   APP_MOTOR0.TargetSpeed = SetSpeed;
  }
```

```
***Using scripting translator version 1.02.24. Folder .\VFControl
Build Project... VFControl ...
Build Successful - 0 Error, 0 Warning
Build Date and Time : 2021-10-21   11:58:12
Script Code Memory Size : 146 Bytes of 16 kBytes
Number of Global Variable(s) : 4, Data Memory Usage : 11 Bytes of 256 Bytes
Task0 - Number of Instruction(s) : 11, Number of Variable(s) : 2, Data Memory Usage : 6 Bytes of 128 Bytes
Task1 - Number of Instruction(s) : 0, Number of Variable(s) : 0, Data Memory Usage : 0 Bytes of 128 Bytes
```

Project	
Name	**VFContro**
Path	**c:\users**
Script	
#SCRIPT_START_COMMAND	**3**
#SCRIPT_TASK0_EXECUTION_PERIOD (ms)	**50**
#SCRIPT_TASK0_EXECUTION_STEP	**10**
#SCRIPT_TASK1_EXECUTION_PERIOD (10 ms)	**10**
#SCRIPT_TASK1_EXECUTION_STEP	**10**
#SCRIPT_USER_VERSION	**1.0**

#SCRIPT_TASK0_EXECUTION_PERIOD (ms)
Script execution time for Task0 (1 count represents 1 ms), maximum count value 60000 The script engine supports two independent tasks, namely, Task 0 and Task 1, running concurrently.Task 0 is been scheduled to be executed in the system tick interrupt.which typically occurs every 1 ms.The user script program runs repeatedly on a configurable interval within Task 0 loop.The execution period of Task 0 can configured here.The granularity of the configurable execution period for Task 0 is 1 ms.For example, setting SCRIPT_TASK0_EXECUTION_PERIOD to 5 results in an execution period of 5· 1□□ = 5□□ for Task 0

Script Editor v0.42.1 | Translator v1.02.24

Figure 1 Script Editor Graphical User Interface

7.2.1 Project Explorer

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

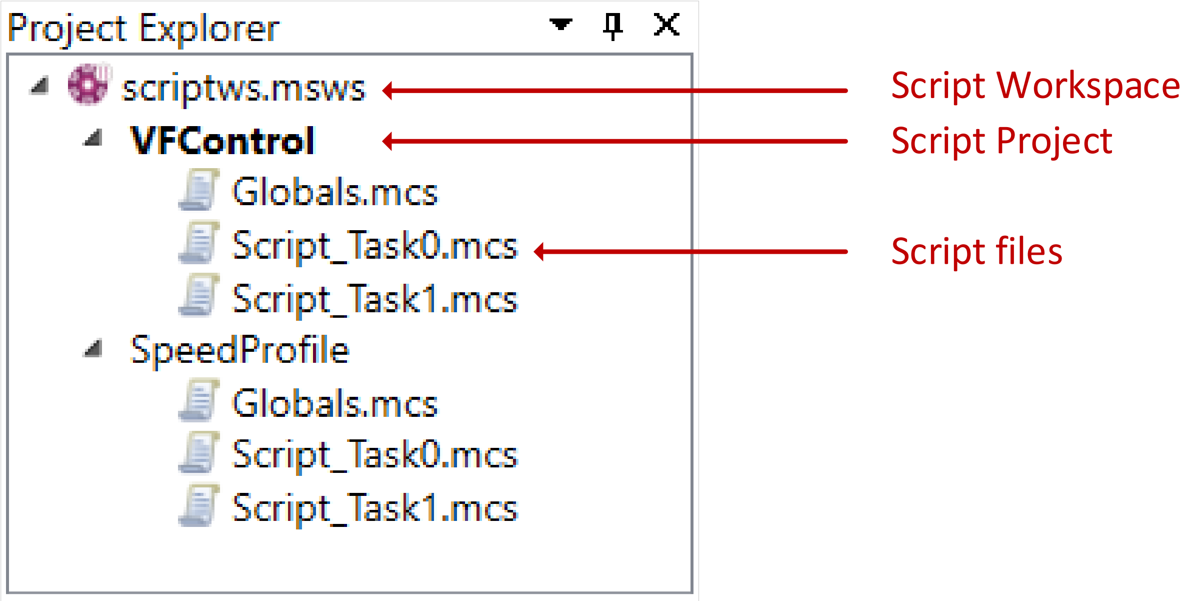
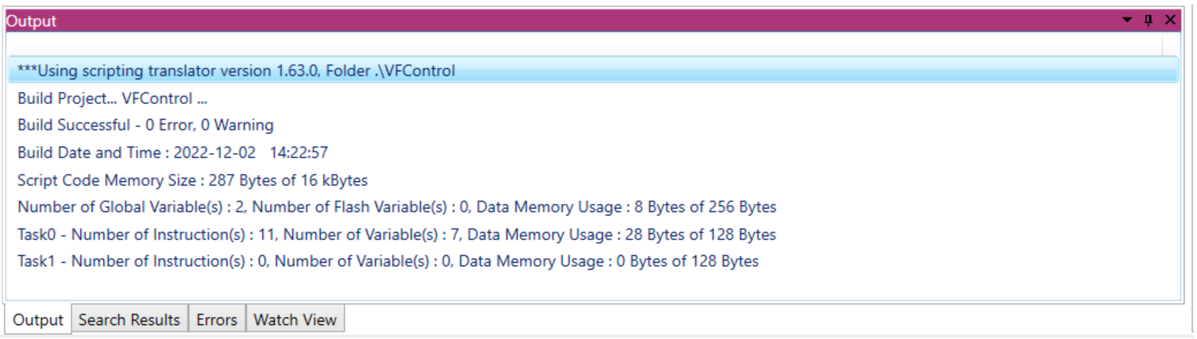


Figure 2 Script Editor: Watch View Window

7.2.2 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.

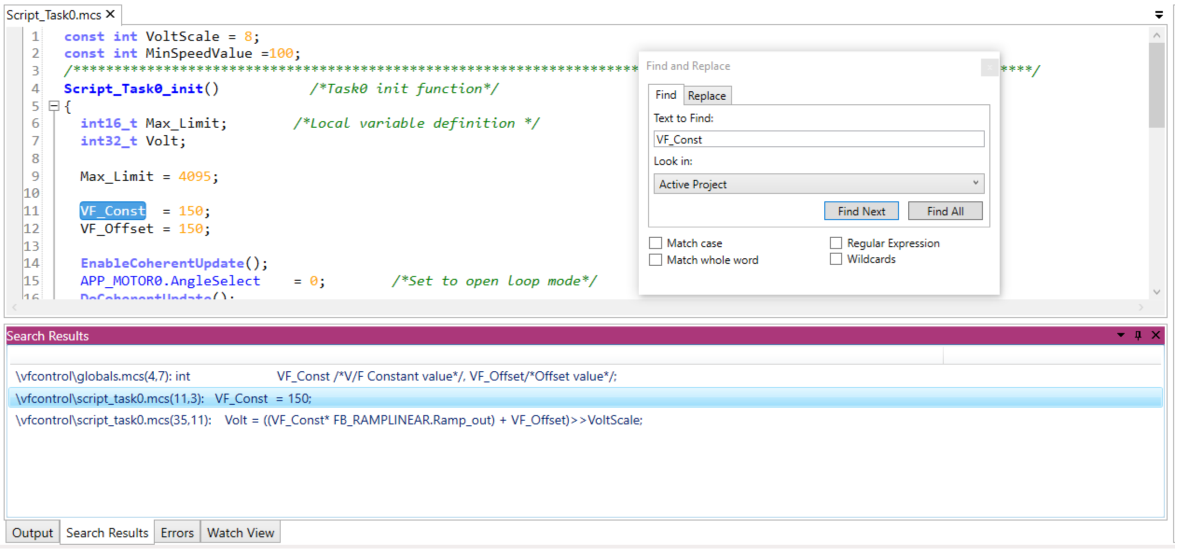


```
***Using scripting translator version 1.63.0, Folder .\VFControl
Build Project... VFControl ...
Build Successful - 0 Error, 0 Warning
Build Date and Time : 2022-12-02   14:22:57
Script Code Memory Size : 287 Bytes of 16 kBytes
Number of Global Variable(s) : 2, Number of Flash Variable(s) : 0, Data Memory Usage : 8 Bytes of 256 Bytes
Task0 - Number of Instruction(s) : 11, Number of Variable(s) : 7, Data Memory Usage : 28 Bytes of 128 Bytes
Task1 - Number of Instruction(s) : 0, Number of Variable(s) : 0, Data Memory Usage : 0 Bytes of 128 Bytes
```

Figure 3 Script Editor: Output window

7.2.3 Search Results

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



```
const int VoltScale = 8;
const int MinSpeedValue =100;
/*********************************************************************  ****/
Script_Task0_init()          /*Task0 init function*/
{
  int16_t Max_Limit;        /*Local variable definition */
  int32_t Volt;

  Max_Limit = 4095;

  VF_Const  = 150;
  VF_Offset = 150;

  EnableCoherentUpdate();
  APP_MOTOR0.AngleSelect    = 0;        /*Set to open loop mode*/
  DoCoherentUpdate();
```

Text to Find: VF_Const

Look in: Active Project

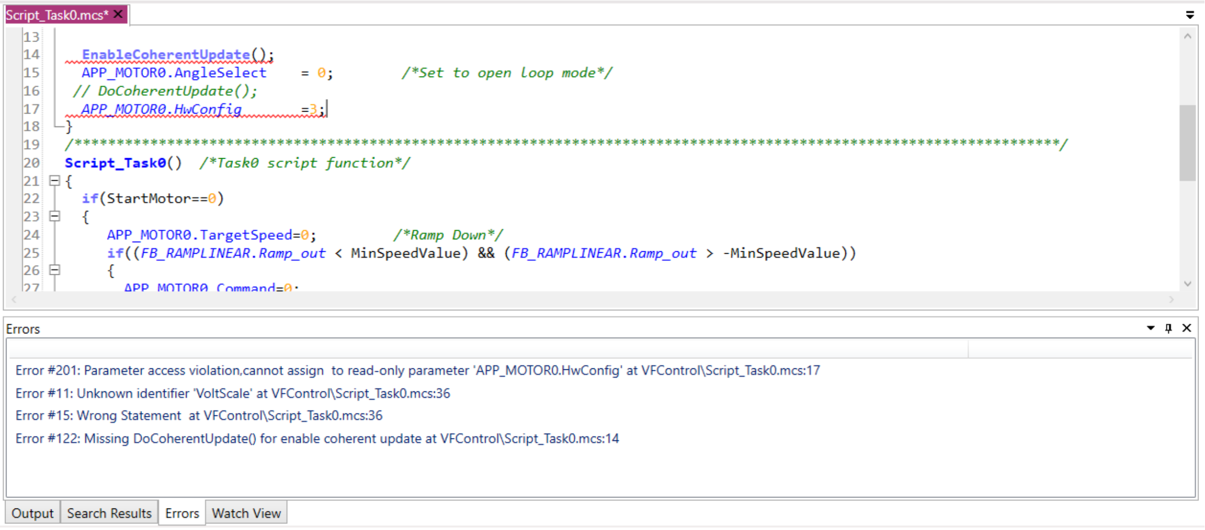
□ Match case
□ Match whole word
□ Regular Expression
□ Wildcards

```
\vfcontrol\globals.mcs(4,7): int             VF_Const /*V/F Constant value*/, VF_Offset/*Offset value*/;
\vfcontrol\script_task0.mcs(11,3):  VF_Const  = 150;
\vfcontrol\script_task0.mcs(35,11):   Volt = ((VF_Const* FB_RAMPLINEAR.Ramp_out) + VF_Offset)>>VoltScale;
```

Figure 4 Script Editor: Search Result Window

7.2.4 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.



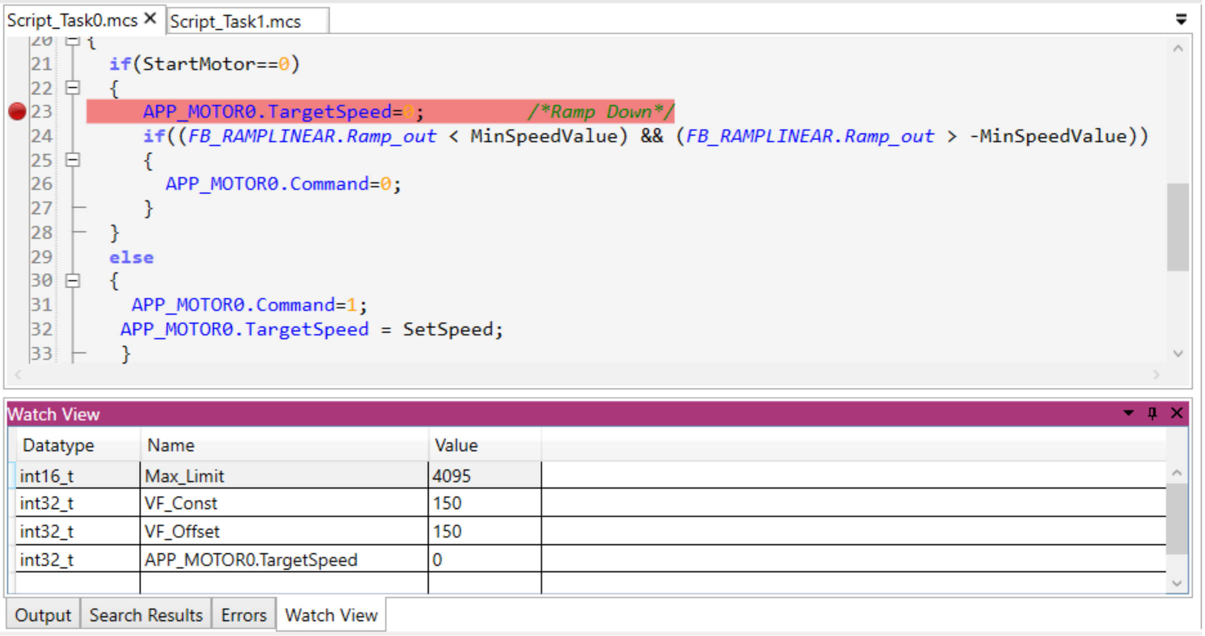
```

  EnableCoherentUpdate();
  APP_MOTOR0.AngleSelect    = 0;        /*Set to open loop mode*/
 // DoCoherentUpdate();
  APP_MOTOR0.HwConfig         =3;
}
/***************************************************************************************************/
Script_Task0()  /*Task0 script function*/
{
  if(StartMotor==0)
  {
     APP_MOTOR0.TargetSpeed=0;          /*Ramp Down*/
     if((FB_RAMPLINEAR.Ramp_out < MinSpeedValue) && (FB_RAMPLINEAR.Ramp_out > -MinSpeedValue))
     {
       APP_MOTOR0.Command=0;
```

```
Error #201: Parameter access violation,cannot assign  to read-only parameter 'APP_MOTOR0.HwConfig' at VFControl\Script_Task0.mcs:17
Error #11: Unknown identifier 'VoltScale' at VFControl\Script_Task0.mcs:36
Error #15: Wrong Statement  at VFControl\Script_Task0.mcs:36
Error #122: Missing DoCoherentUpdate() for enable coherent update at VFControl\Script_Task0.mcs:14
```

Figure 5 Script Editor: Error Window

7.2.5 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.



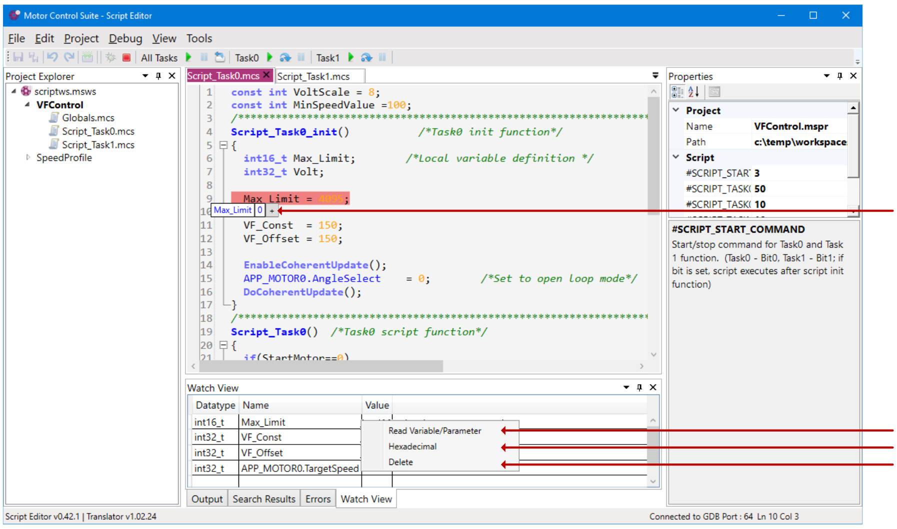
```
{
  if(StartMotor==0)
  {
    APP_MOTOR0.TargetSpeed=0;             /*Ramp Down*/
    if((FB_RAMPLINEAR.Ramp_out < MinSpeedValue) && (FB_RAMPLINEAR.Ramp_out > -MinSpeedValue))
    {
      APP_MOTOR0.Command=0;
    }
  }
  else
  {
    APP_MOTOR0.Command=1;
   APP_MOTOR0.TargetSpeed = SetSpeed;
  }
```

Watch View

Datatype	Name	Value
int16_t	Max_Limit	4095
int32_t	VF_Const	150
int32_t	VF_Offset	150
int32_t	APP_MOTOR0.TargetSpeed	0

Figure 6 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.



```
const int VoltScale = 8;
const int MinSpeedValue =100;
/*********************************************************************
Script_Task0_init()          /*Task0 init function*/
{
  int16_t Max_Limit;        /*Local variable definition */
  int32_t Volt;

  Max_Limit = 4095;

  VF_Const  = 150;
  VF_Offset = 150;

  EnableCoherentUpdate();
  APP_MOTOR0.AngleSelect    = 0;         /*Set to open loop mode*/
  DoCoherentUpdate();
}
/*********************************************************************
Script_Task0()  /*Task0 script function*/
{
  if(StartMotor==0)
```

Project	
Name	VFControl.mspr
Path	c:\temp\workspace
Script	
#SCRIPT_STAR	3
#SCRIPT_TASK(	50
#SCRIPT_TASK(	10

#SCRIPT_START_COMMAND
Start/stop command for Task0 and Task 1 function. (Task0 - Bit0, Task1 - Bit1; if bit is set, script executes after script init function)

Watch View

Datatype	Name	Value
int16_t	Max_Limit	
int32_t	VF_Const	
int32_t	VF_Offset	
int32_t	APP_MOTOR0.TargetSpeed	

Script Editor v0.42.1 | Translator v1.02.24
Connected to GDB Port : 64 Ln 10 Col 3

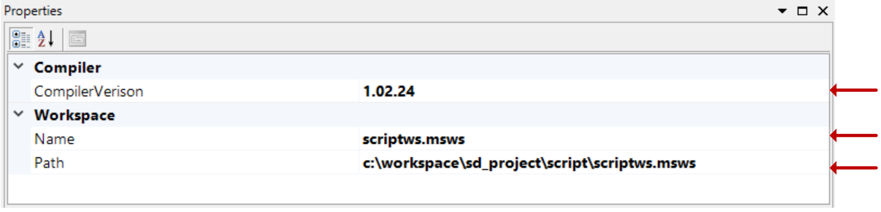
Press '+' to add the variable into Watch View Window

Enable or Disable read Operation
Change display format (HEX or DEC)
Remove the variable for Watch window

Figure 7 Script Editor: Watch View Window

7.2.6 Property Window

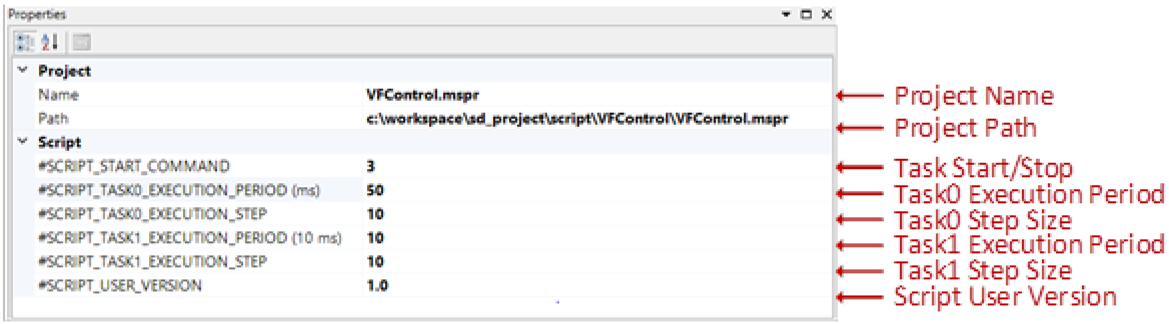
The property window is used to view and modify the property of selected items of the active project, Work Space, Project file, or Script file.



Compiler	
CompilerVerison	1.02.24
Workspace	
Name	scriptws.msws
Path	c:\workspace\sd_project\script\scriptws.msws

Compiler Version
Workspace Name
Workspace Path

Figure 8 Script Editor: Workspace Property window



Properties

Project	
Name	VFControl.mspr
Path	c:\workspace\sd_project\script\VFControl\VFControl.mspr
Script	
#SCRIPT_START_COMMAND	3
#SCRIPT_TASK0_EXECUTION_PERIOD (ms)	50
#SCRIPT_TASK0_EXECUTION_STEP	10
#SCRIPT_TASK1_EXECUTION_PERIOD (10 ms)	10
#SCRIPT_TASK1_EXECUTION_STEP	10
#SCRIPT_USER_VERSION	1.0

Project Name
Project Path
Task Start/Stop
Task0 Execution Period
Task0 Step Size
Task1 Execution Period
Task1 Step Size
Script User Version

Figure 9 Script Editor: Project Property window

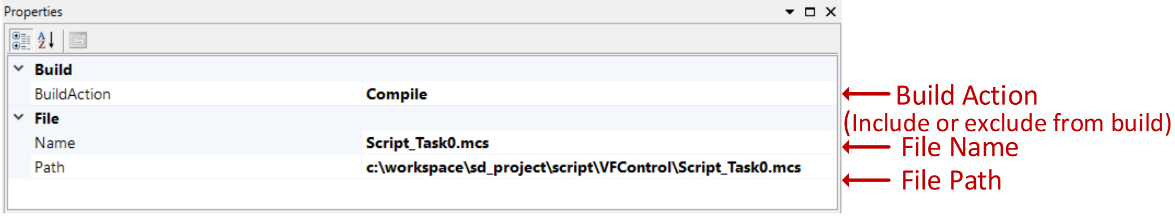
#SCRIPT_START_COMMAD: Provides configuration to start or stop the script execution after script initialization. Bit0 value of this configuration controls Task0 execution and Bit1 value controls Task1 execution. If the bit value is 1, then Task function will start executing after initialization. **APP_Scripting.Command**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

#SCRIT_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.

#SCRIT_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.

#SCRIT_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 $50 \cdot 1ms = 50ms$ for Task0

#SCRIT_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 $10 \cdot 10ms = 100ms$ for Task 1



Properties

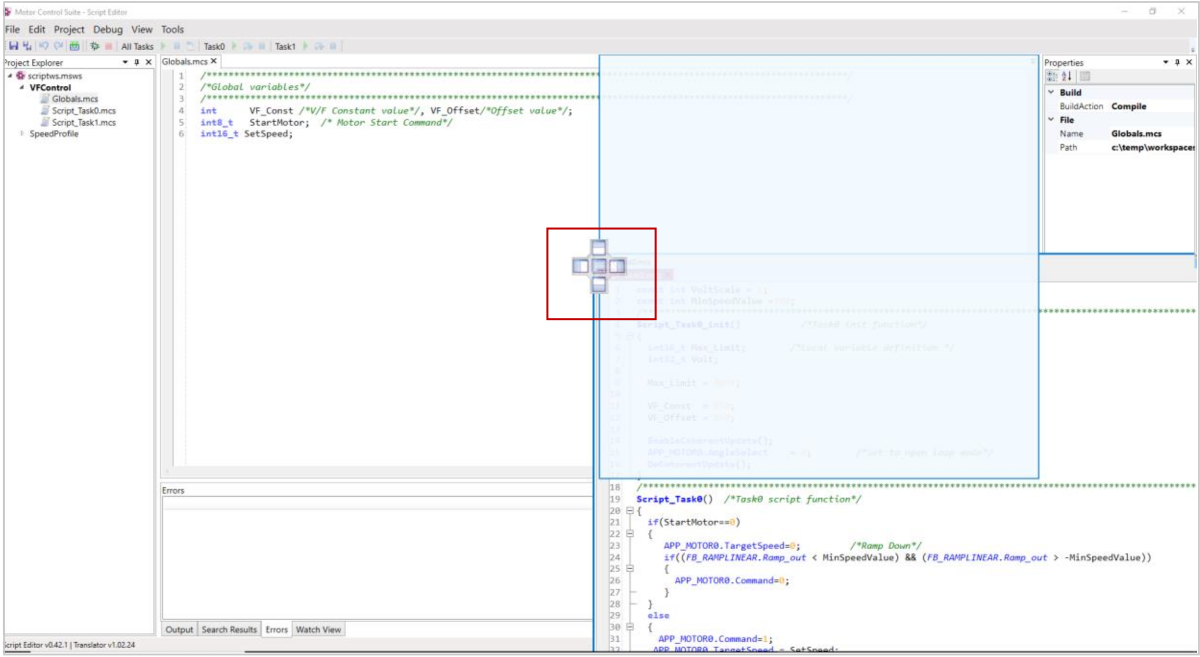
Build	
BuildAction	Compile
File	
Name	Script_Task0.mcs
Path	c:\workspace\sd_project\script\VFControl\Script_Task0.mcs

Build Action
(Include or exclude from build)
File Name
File Path

Figure 10 Script Editor: File Property window

7.2.7 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.



```
/****************************************************************************
/*Global variables*/
/****************************************************************************
int      VF_Const /*V/F Constant value*/, VF_Offset/*Offset value*/;
int8_t   StartMotor;  /* Motor Start Command*/
int16_t SetSpeed;
```

BuildAction Compile
Name Globals.mcs
Path c:\temp\workspace

```
/****************************************************************************
Script_Task0()  /*Task0 script function*/
{
  if(StartMotor==0)
  {
    APP_MOTOR0.TargetSpeed=0;         /*Ramp Down*/
    if((FB_RAMPLINEAR.Ramp_out < MinSpeedValue) && (FB_RAMPLINEAR.Ramp_out > -MinSpeedValue))
    {
      APP_MOTOR0.Command=0;
    }
  }
  else
  {
   APP_MOTOR0.Command=1;
  APP_MOTOR0.TargetSpeed = SetSpeed;
```

Figure 11 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.

7.3 Script Editor Menu

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

7.3.1 Project Menu

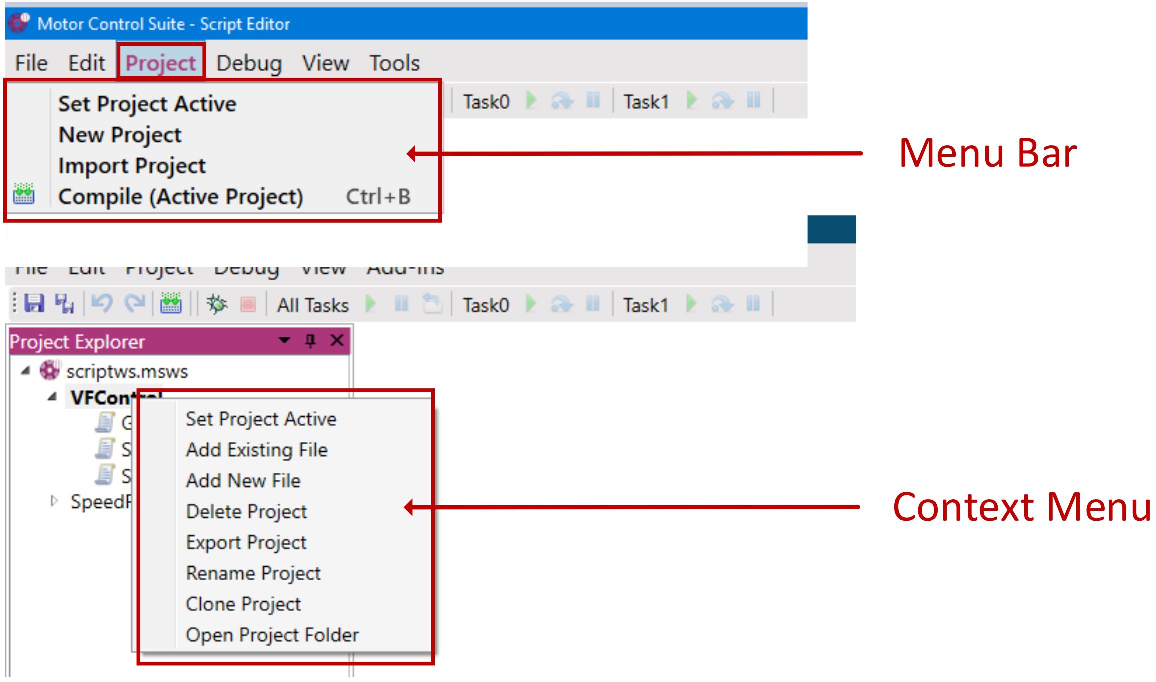


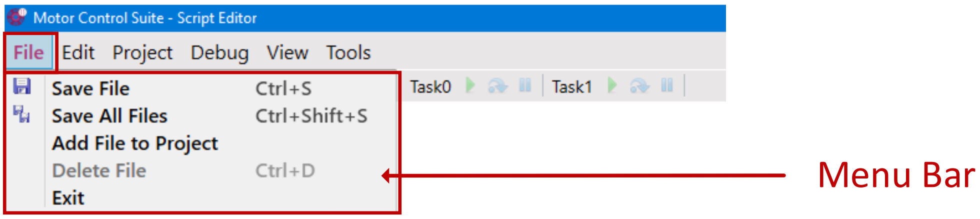
Table 1 **Script Editor: Project Menu**

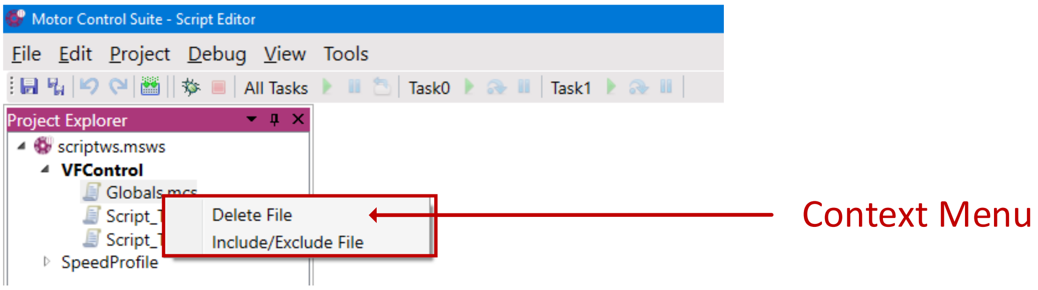
Menu	Shortcut	Description
Set Project Active		Set the selected project as active project
New Project		Create new project
Import Project		Import an existing project into current workspace
Compile (Active Project)	Ctrl+B	Compile the active project
Add Existing File		Add existing script file in the selected 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 fodler 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, won’t change project file name (<ProjectName>.mspr) inside the zip folder.

7.3.2 File Menu


Motor Control Suite - Script Editor
File Edit Project Debug View Tools
Save File Ctrl+S
Save All Files Ctrl+Shift+S
Add File to Project
Delete File Ctrl+D
Exit
Task0
Task1
Menu Bar



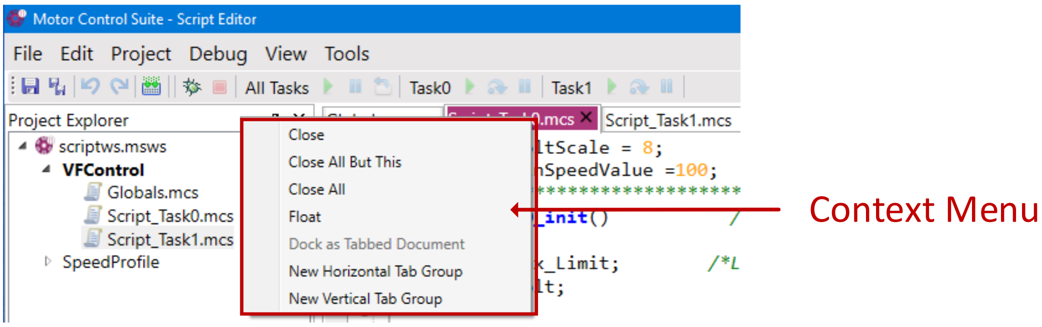
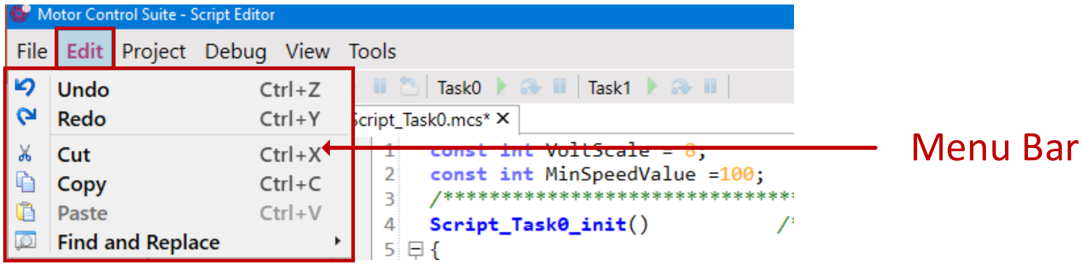


Table 2 Script Editor: File Menu

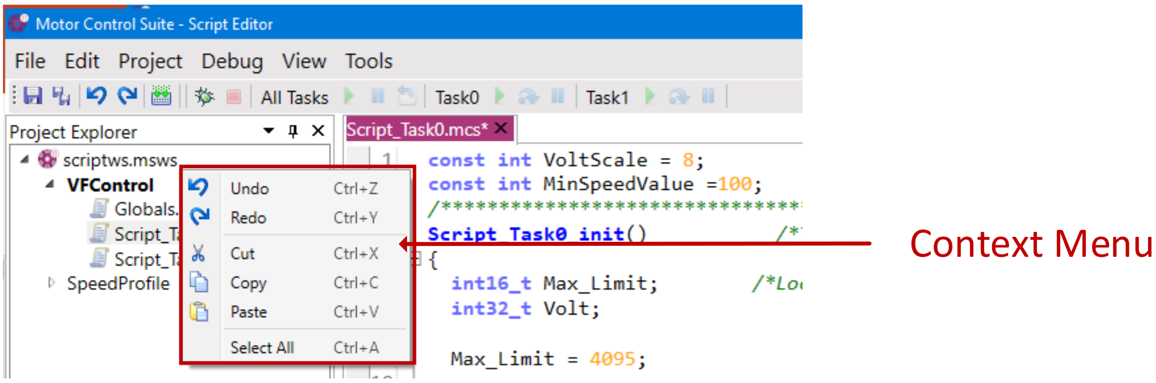
Menu	Shortcut	Description
Save File	Ctrl+S	Save- the active file
Save All Files	Ctrl+Shift+S	Save- all the open file/s
Add File to Project		Add a new file into the active project
Delete File	Ctrl+D	Delete- the active file
Exit		Close Script Editor
Include/Exclude File		Include or exclude the active file 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+SpaceBar shortcut key provides complete list of all the available MCE parameters and variables inside editor area.

7.3.3 Edit Menu



```
const int VoltScale = 8;
const int MinSpeedValue =100;
/*******************************
Script_Task0_init()            /*
{
```



```
const int VoltScale = 8;
const int MinSpeedValue =100;
/*******************************
Script_Task0_init()            /*
{
    int16_t Max_Limit;        /*Lo
    int32_t Volt;

    Max_Limit = 4095;
```

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.
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

7.3.4 View Menu

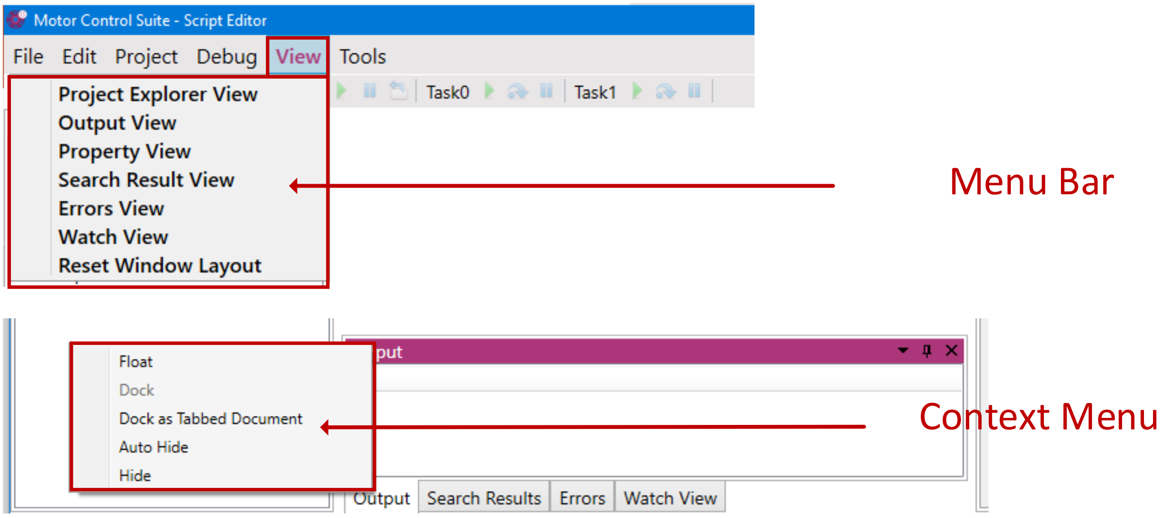


Table 4 Script Editor: View Menu

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
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

7.3.5 Debug Menu

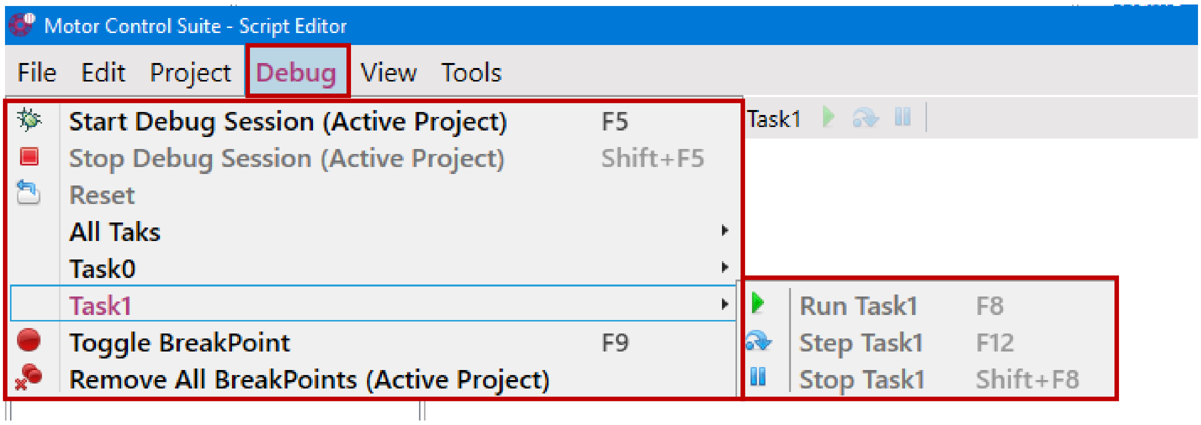

Motor Control Suite - Script Editor
File
Edit
Project
Debug
View
Tools
Task1
Start Debug Session (Active Project) F5
Stop Debug Session (Active Project) Shift+F5
Reset
All Taks
Task0
Task1
Toggle BreakPoint F9
Remove All BreakPoints (Active Project)
Run Task1 F8
Step Task1 F12
Stop Task1 Shift+F8

Table 5 Script Editor: Debug Menu

Menu	Shortcut	Description
Start Debug Session	F5	Start script debug session
Stop Debug Session	Shitf+F5	Stop script debug session
All Tasks ->Run All Tasks	F6	Run both the script tasks, task0 and task1
All Tasks ->Stop All Tasks	Shitf+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

7.3.6 Script Editor Tool Bar

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

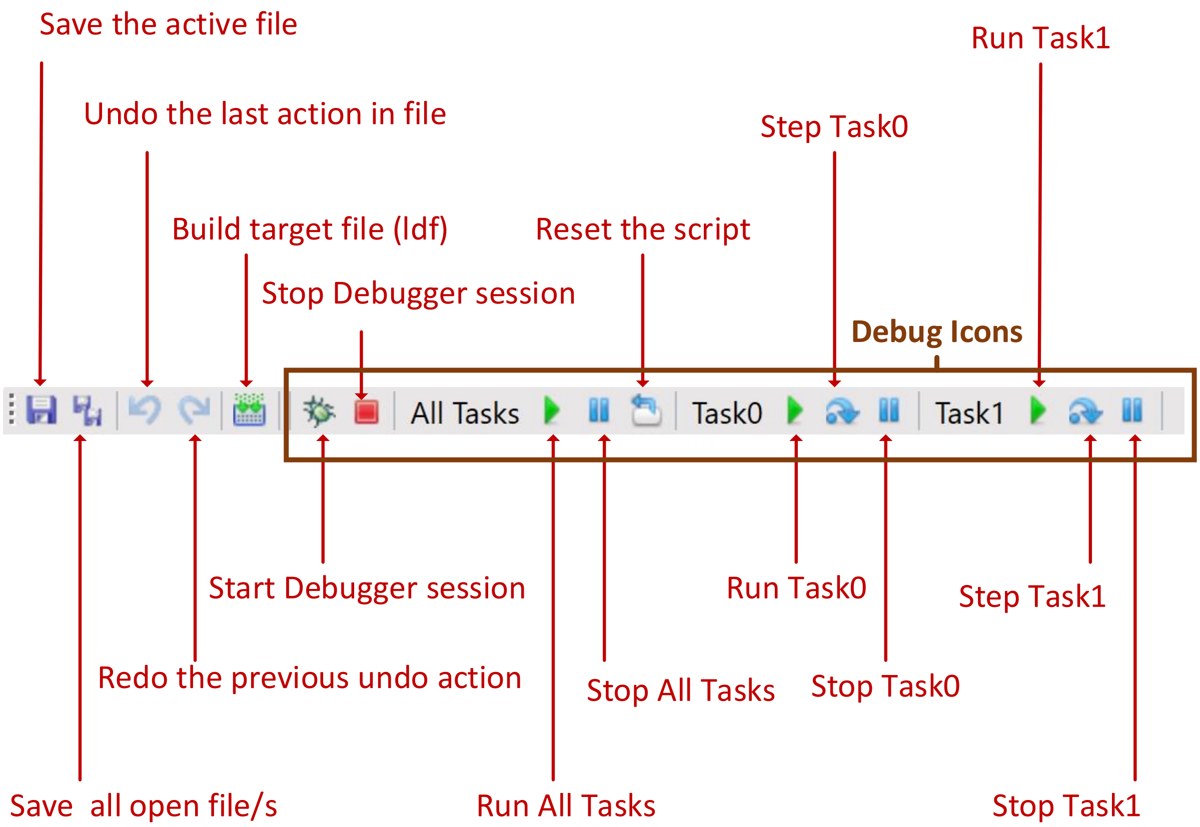
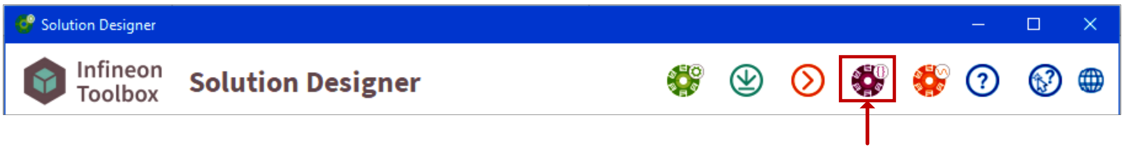


Figure 12 Script Editor: Tool Bar

7.4 Working with a Script Project

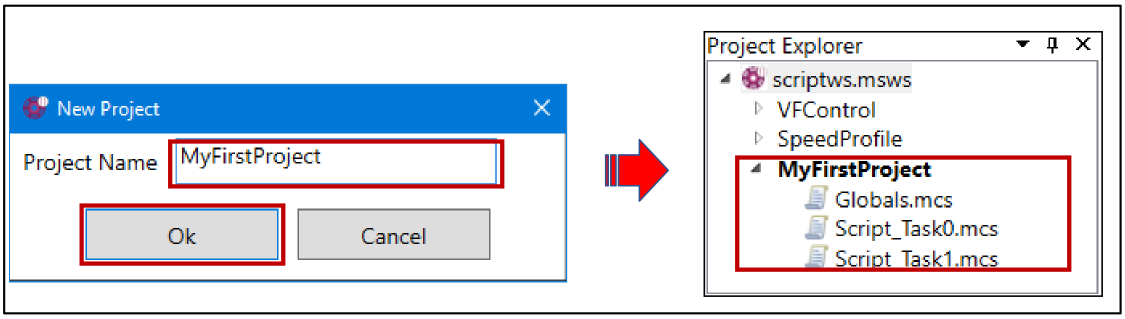
7.4.1 Creating a Script Project

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

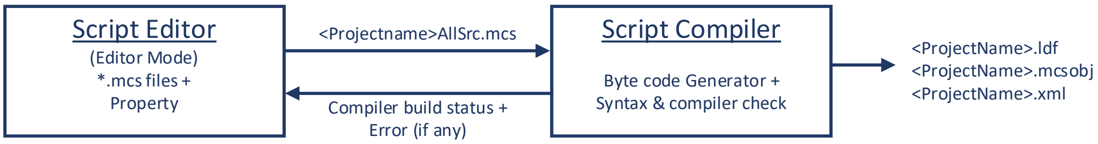


Open Script Editor

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



- 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.



Script Editor (Editor Mode) *.mcs files + Property

<Projectname>AllSrc.mcs

Compiler build status + Error (if any)

Script Compiler

Byte code Generator + Syntax & compiler check

<ProjectName>.ldf
<ProjectName>.mcsobj
<ProjectName>.xml

Note:

<Projectname>AllSrc.mcs	- Combined file of all script mcs files (build enabled) and project property
<ProjectName>.ldf	- Script byte code
<ProjectName>.mcsobj	- Script object file
<ProjectName>.xml	- Script variables definition

Figure 13 Script editor: Editor mode

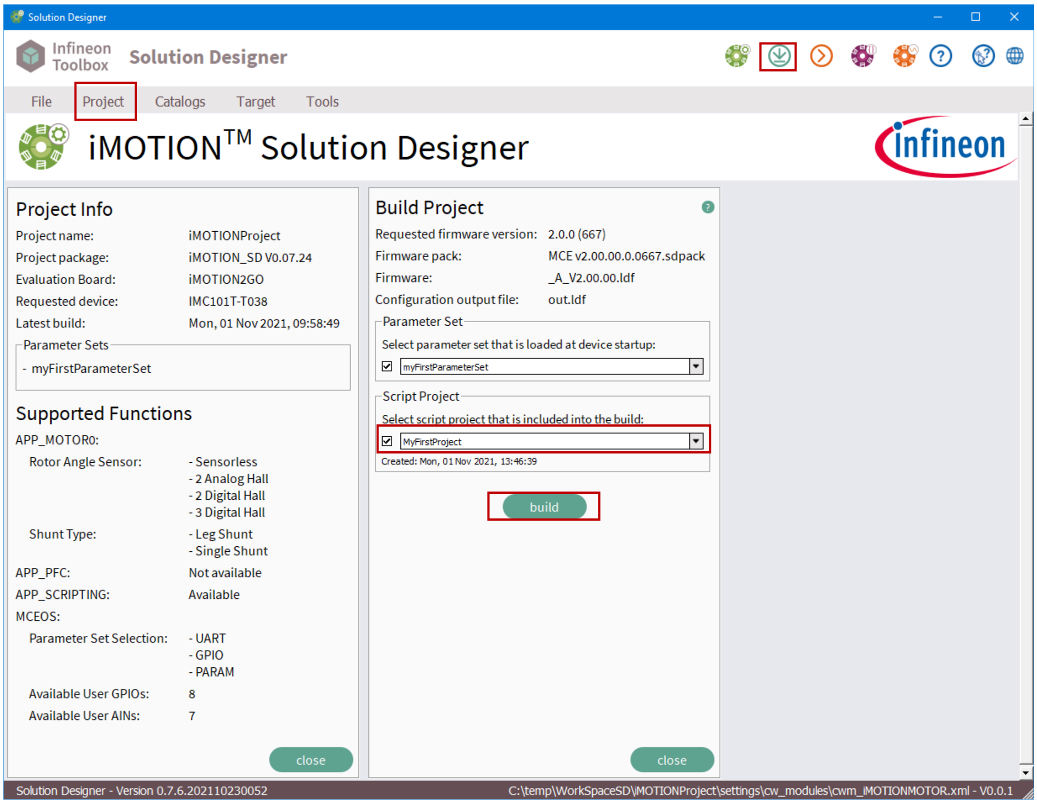
Each script project contains following files

- <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)

7.4.2 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)



Project Info

Project name: iMOTIONProject
Project package: iMOTION_SD V0.07.24
Evaluation Board: iMOTION2GO
Requested device: IMC101T-T038
Latest build: Mon, 01 Nov 2021, 09:58:49

Parameter Sets
- myFirstParameterSet

Supported Functions

APP_MOTOR0:
Rotor Angle Sensor: - Sensorless
- 2 Analog Hall
- 2 Digital Hall
- 3 Digital Hall
Shunt Type: - Leg Shunt
- Single Shunt
APP_PFC: Not available
APP_SCRIPTING: Available
MCEOS:
Parameter Set Selection: - UART
- GPIO
- PARAM
Available User GPIOs: 8
Available User AINs: 7

Build Project

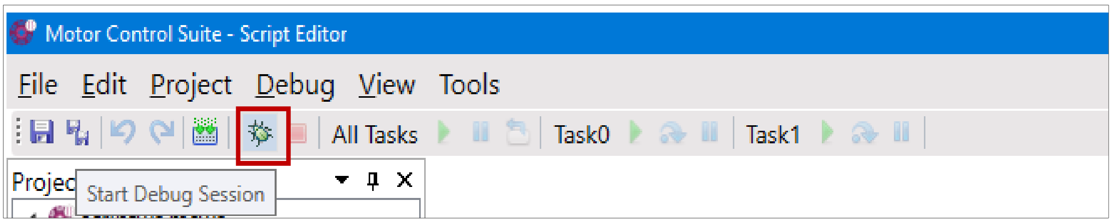
Requested firmware version: 2.0.0 (667)
Firmware pack: MCE v2.00.00.0.0667.sdpack
Firmware: _A_V2.00.00.ldf
Configuration output file: out.ldf

Parameter Set
Select parameter set that is loaded at device startup:
☑ myFirstParameterSet

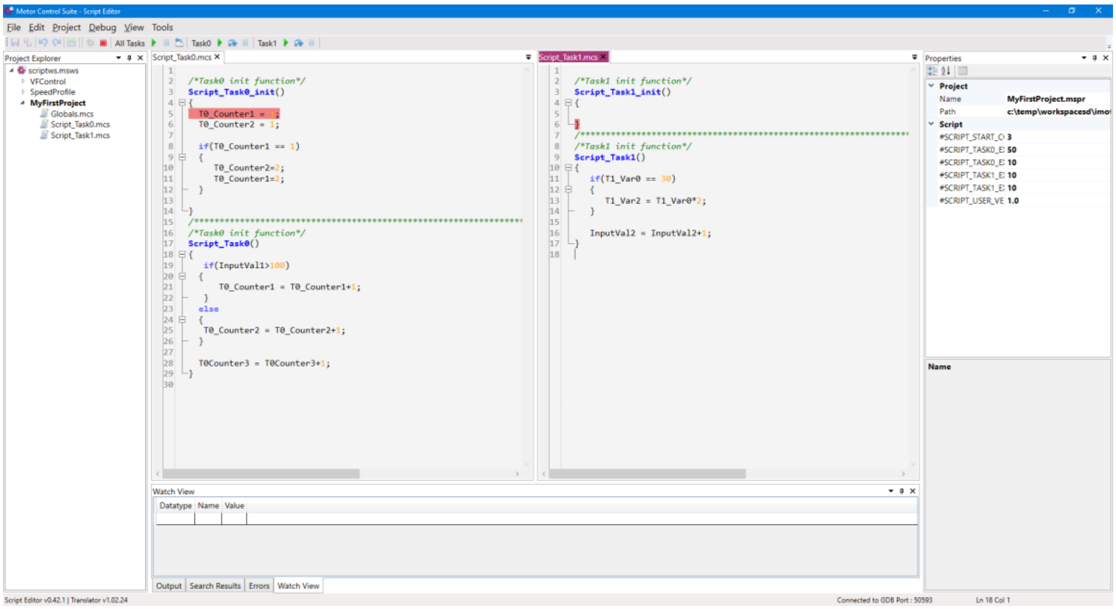
Script Project
Select script project that is included into the build:
☑ MyFirstProject
Created: Mon, 01 Nov 2021, 13:46:39

Solution Designer - Version 0.7.6.202110230052
C:\temp\WorkSpaceSD\iMOTIONProject\settings\cw_modules\cwm_iMOTIONMOTOR.xml - V0.0.1

- 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.



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



```
/*Task0 init function*/
Script_Task0_init()
{
  T0_Counter1 = 1;
  T0_Counter2 = 1;

  if(T0_Counter1 == 1)
  {
     T0_Counter2=2;
     T0_Counter1=2;
  }

}
/*********************************************************************
/*Task0 init function*/
Script_Task0()
{
   if(InputVal1>100)
  {
     T0_Counter1 = T0_Counter1+1;
   }
  else
  {
   T0_Counter2 = T0_Counter2+1;
  }

  T0Counter3 = T0Counter3+1;
}
```

```
/*Task1 init function*/
Script_Task1_init()
{

}
/*********************************************************************
/*Task1 init function*/
Script_Task1()
{
   if(T1_Var0 == 30)
   {
      T1_Var2 = T1_Var0*2;
   }

   InputVal2 = InputVal2+1;
}
```

Project	
Name	MyFirstProject.mspr
Path	c:\temp\workspacesd\imot
Script	
#SCRIPT_START_C	3
#SCRIPT_TASK0_E	50
#SCRIPT_TASK0_E	10
#SCRIPT_TASK1_E	10
#SCRIPT_TASK1_E	10
#SCRIPT_USER_VE	1.0

Name

Datatype	Name	Value

Script Editor v0.42.1 | Translator v1.02.24 Connected to GDB Port : 50593 Ln 18 Col 1

- 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.

8 Revision History

Document version	Date of release	Description of changes
1.0	2022-12-21	Initial document release