Application Note

Getting Started with MOTORSTUDIO

TEXAS INSTRUMENTS

Dhinesh Kumar Shanmugam

ABSTRACT

MCF83xx is TI's code-free sensorless FOC algorithm-based BLDC motor driver product that comes with numerous features catering to different BLDC motor-driven application requirements. *MotorStudio* is the graphical user interface tool that assists users in configuring device features for their specific applications. This document provides a guide for users evaluating a given BLDC motor with TI's MCF83xx portfolio using *MotorStudio*.

The document is applicable for all the following devices (referred to as MCF83xx devices in this document):

- MCF8315A
- MCF8315C
- MCF8315C-Q1
- MCF8316A
- MCF8316C
- MCF8316C-Q1
- MCF8315D
- MCF8316D
- MCF8329A
- MCF8329A-Q1
- MCF8329HS
- MCF8329HS-Q1

Table of Contents

1 Introduction....3
2 System Requirements....3
3 *MOTORSTUDIO* GUI Installation....3
4 Prerequisites to Run the Motor with *MOTORSTUDIO*....3
5 Getting Started with *MOTORSTUDIO*....4
5.1 Home....5
5.2 SmartTune....6
5.3 Optimization Wizard....6
5.4 Motor Parameter Extraction Tool (MPET)....7
5.5 Closed Loop Tuning....7
5.6 Advanced Tuning....8
5.7 Register Map....9
5.7.1 Register read....11
5.8 Control Utilities....11
5.8.1 Control....11
5.8.2 Faults....13
5.8.3 Charts....14
5.8.4 Logs....15
5.9 Other Utilities....16
5.9.1 Scripting Tool....16
5.9.2 Compare Tuning File Tool....18
6 Spinning the motor....19
6.1 Connecting MCF83xx to the GUI....19
6.2 SmartTune Execution....20

6.3 Improving Current and Speed Regulation........ 23
6.4 Testing for Successful Startup Into Closed Loop........ 27
7 Summary........ 28
8 References........ 28
9 Revision History........ 29

Trademarks

All trademarks are the property of their respective owners.

1 Introduction

MotorStudio is an intuitive graphical user interface (GUI) designed to simplify the tuning process for brushless DC (BLDC) motor drivers, thereby reducing device evaluation and product development time. The GUI enables users to efficiently initialize their motors and optimize various stages of motor operation to achieve optimal motor performance, stable motor startup, and minimal noise. This guide provides instructions on utilizing the features offered by the *MotorStudio* GUI to tune BLDC motors using MCF83xx BLDC controllers.

Note

This document serves exclusively as an introductory guide for getting started with *MotorStudio*. It does not include guidance for tuning MCF83xx parameters.

2 System Requirements

The following list shows the minimum recommended system requirements for the *MotorStudio* GUI application software:

- PC with Intel® Core i3 processor (or equivalent) or higher
- Windows™ 10 (64-bit) or higher
- 4 GB RAM
- Minimum display resolution - SVGA (800 × 600), Recommended - XGA (1024 × 768) or higher
- 500 MB of free hard-disk space
- USB port

3 *MOTORSTUDIO* GUI Installation

Download the *MotorStudio* GUI PC software installation setup from www.ti.com to the PC's local drive. Execute the *MotorStudio* installation file and follow the on-screen instructions to complete the software installation process.

Note

If an "Error while installing VCP driver" error message appears during *MotorStudio* installation, please refer to the instructions provided in the E2E FAQ for resolution steps.

4 Prerequisites to Run the Motor with *MOTORSTUDIO*

Table 4-1. Prerequisites to Work With *MOTORSTUDIO*

Sl No	Requirements
1	Laptop installed with MOTORSTUDIO
3	MCF8329HS/29A/16A/16C/16D/15A/15C/15D EVM
4	BLDC Motor
5	DC power supply
6	USB Cable
7	Connecting wires

Table 4-2 are the specific parameters related to the motor that the user needs to document before proceeding to the next steps in the tuning process.

Table 4-2. Motor Parameters Required

Sl No	Required Parameters
1	Rated input DC voltage (V)
2	Maximum motor speed (RPM/Hz)
3	Rated motor phase peak current (A)

5 Getting Started with *MOTORSTUDIO*

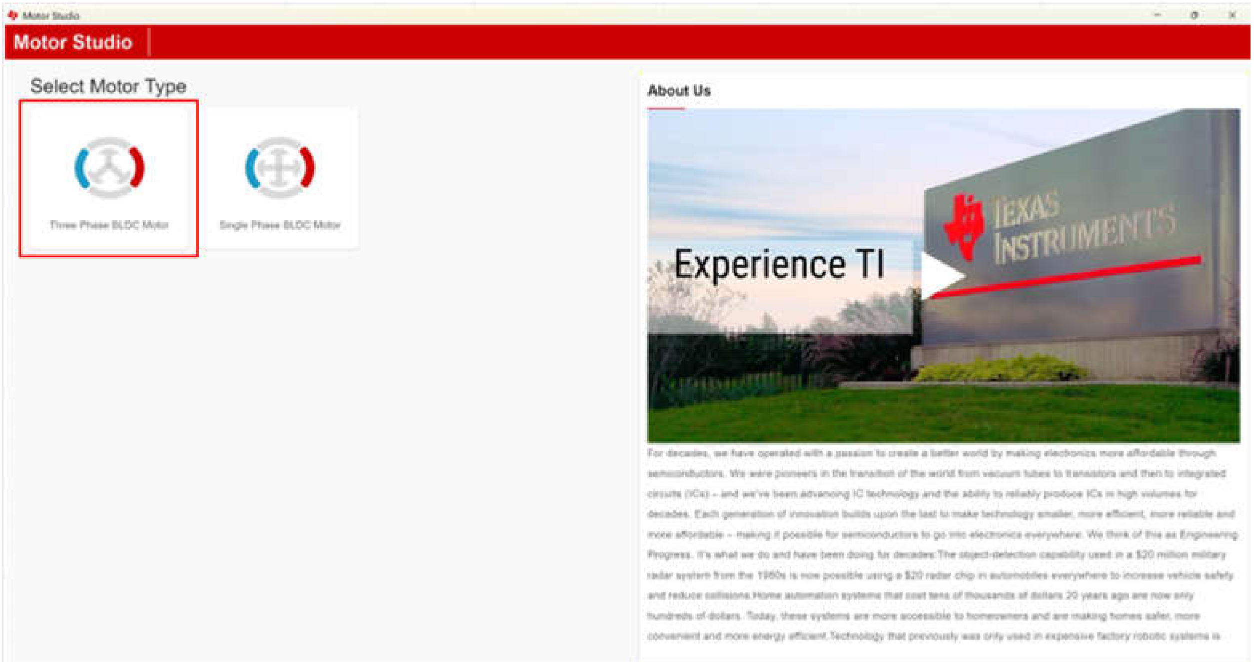

Figure 5-1. MotorStudio Landing Page

MotorStudio GUI is designed to support TI's three-phase integrated sensorless Field-Oriented Control (FOC) BLDC motor driver and single-phase motor driver products. In the *MotorStudio* landing page, select "*Three-phase BLDC motor*" for MCF83xx product configuration and "*Single-phase BLDC motor*" for MCxx product configuration.

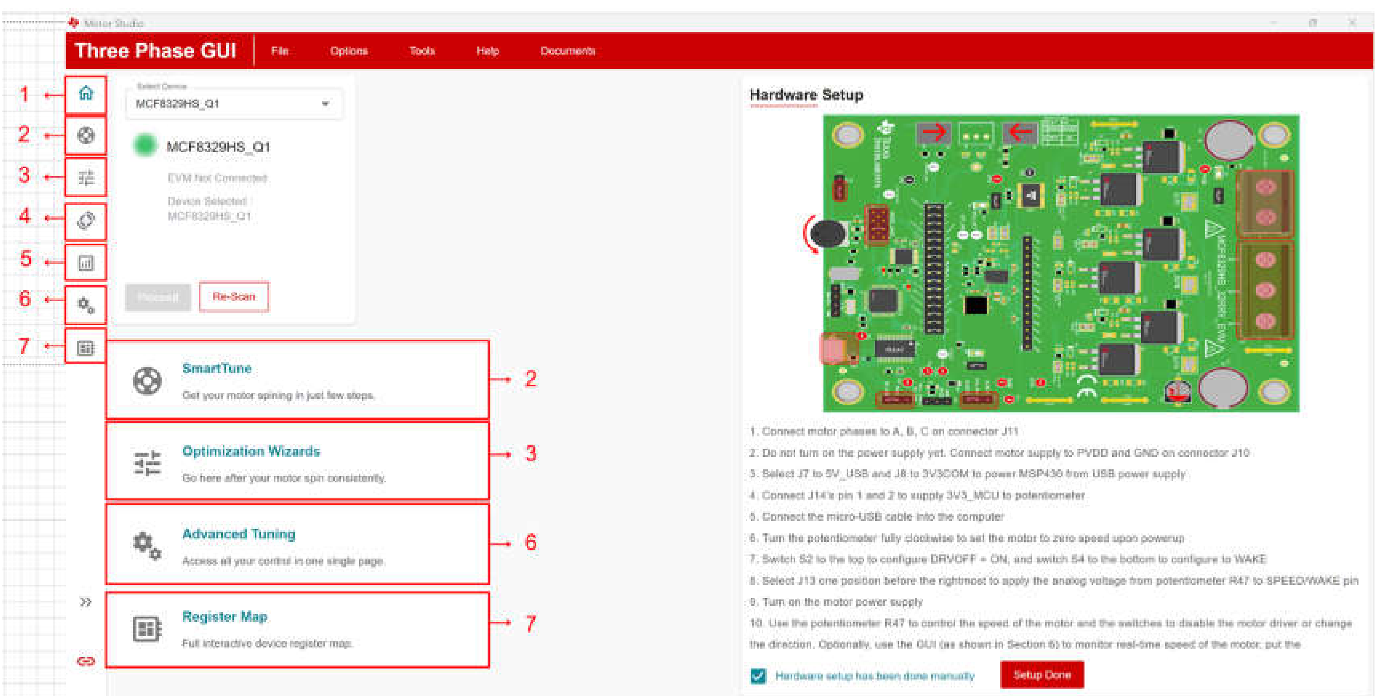

Figure 5-2. Home Page

MotorStudio GUI provides multiple utilities and options for accessing device register configurations. A brief description of each roption is provided in the following sections.

Note

1. Home Page
2. *SmartTune*
3. Optimization Wizards
4. Motor Parameter Extraction Tool (MPET)
5. Close loop Tuning
6. Advanced Tuning
7. Register Map

5.1 Home

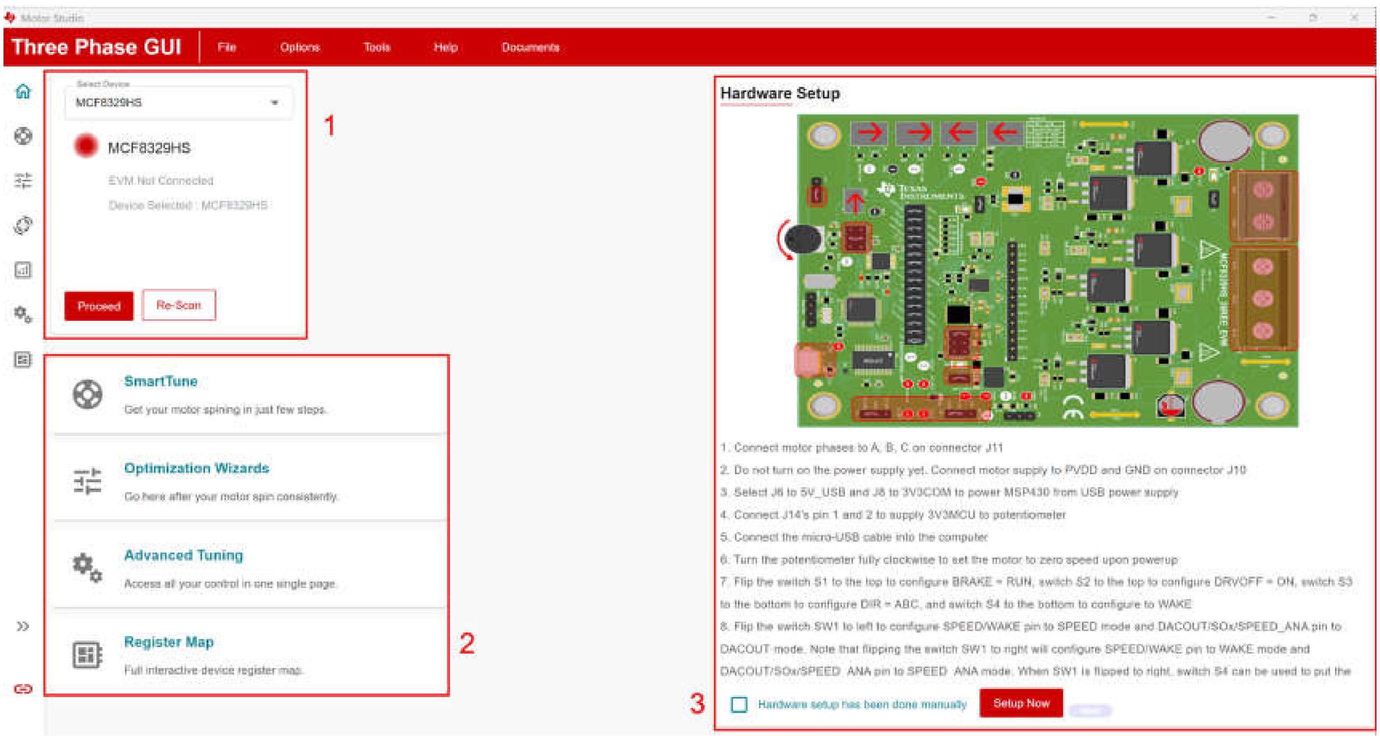

Figure 5-3. Home

Home page covers the following funcitonalities:

1. Displays the connected device part number and connection status.

   Connection status indicators:

   - Green: Successful device connection with the GUI

   - Red: Connection failure (select the *Rescan* button to refresh the connection)

   • Select the *Proceed* button to confirm device connection with the GUI

   • Users may also select a device from the drop-down menu and choose the *Proceed* button to view the device's register settings offline
2. Quick access buttons to navigate different tools available in the GUI.
3. Display the EVM hardware setup and connection details. This serves as a quick reference for users to ensure the EVM's jumper settings and connections are correct.

The detailed procedure for connecting the EVM to *MotorStudio* is explained in Connecting MCF83xx to GUI

5.2 SmartTune

SmartTune is an intelligent auto-configuration tool integrated into TI's *MotorStudio* GUI. *SmartTune* simplifies and automates the tuning process for BLDC motors utilizing TI's 3-phase sensorless FOC (Field-Oriented Control) motor drivers. Users can configure and operate a motor within a couple of minutes with only four required inputs: motor rated voltage, rated current, rated speed, and shunt resistor value. The detailed SmartTune execution procedure is explained in *SmartTune Execution* chapter.

5.3 Optimization Wizard

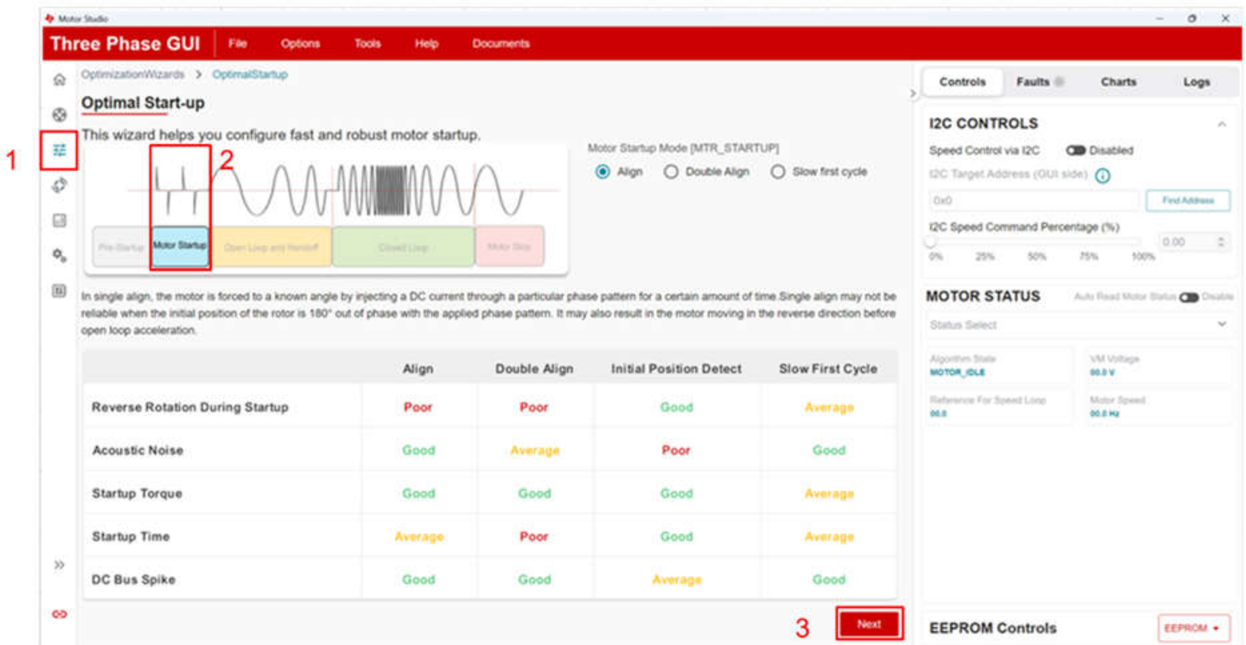

Figure 5-4. Optimization Wizards

This page provides users with access to device configurations based on the motor stages of operation, complete with detailed descriptions. Different stages of motor sensorless FOC operation are depicted in a diagram at the top of the page. Users can select any of these stages to access parameters related to the specific stage selected.

For example,When a user selects the Motor Startup section in the diagram, motor startup-related settings are displayed for convenient access and configuration.

5.4 Motor Parameter Extraction Tool (MPET)

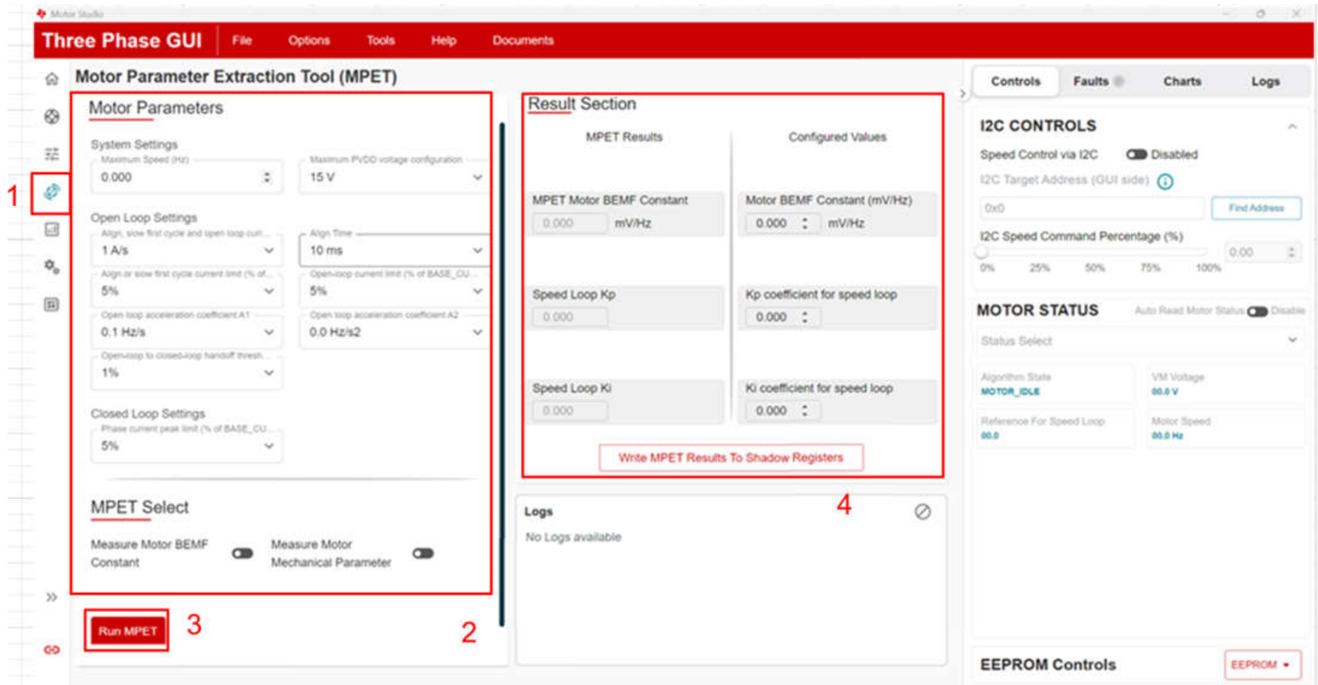

Figure 5-5. Motor Parameter Extraction Tool (MPET)

The Motor Parameter Extraction Tool (MPET) is a device algorithm feature that assists in identifying motor electrical parameters, including phase resistance, phase inductance, BEMF constant, and speed loop coefficients (Kp and Ki).

To execute MPET, users must configure the motor startup parameters listed under the *Motor Parameters* section on the MPET page.

For MCF8329x devices, the following configurations must be established before running MPET:

i) Base current and CSA gain parameters based on the shunt resistor value

ii) Motor phase resistance and phase inductance

Users can select the parameters to be measured during MPET operation by choosing options under *MPET Select*.

Click the "*Run MPET*" command button to initiate motor parameter measurement.

Upon successful execution of MPET, the measured parameters are displayed in the MPET Result section. Users can select the "*Write MPET Result To Shadow Registers*" button to transfer the measured motor parameters into the device registers.

Note

MPET is a subset of the SmartTune feature. TI recommends using the SmartTune tool to identify motor parameters.

5.5 Closed Loop Tuning

SmartTune provides basic tuning configuration, which includes current and speed loop Kp and Ki coefficients. For users who prefer to tune these coefficients manually, the Closed Loop Tuning page offers an option to adjust these loop coefficients. The detailed procedure for tuning speed and current loop coefficients is explained in the Improving Current and Speed Regulation

5.6 Advanced Tuning

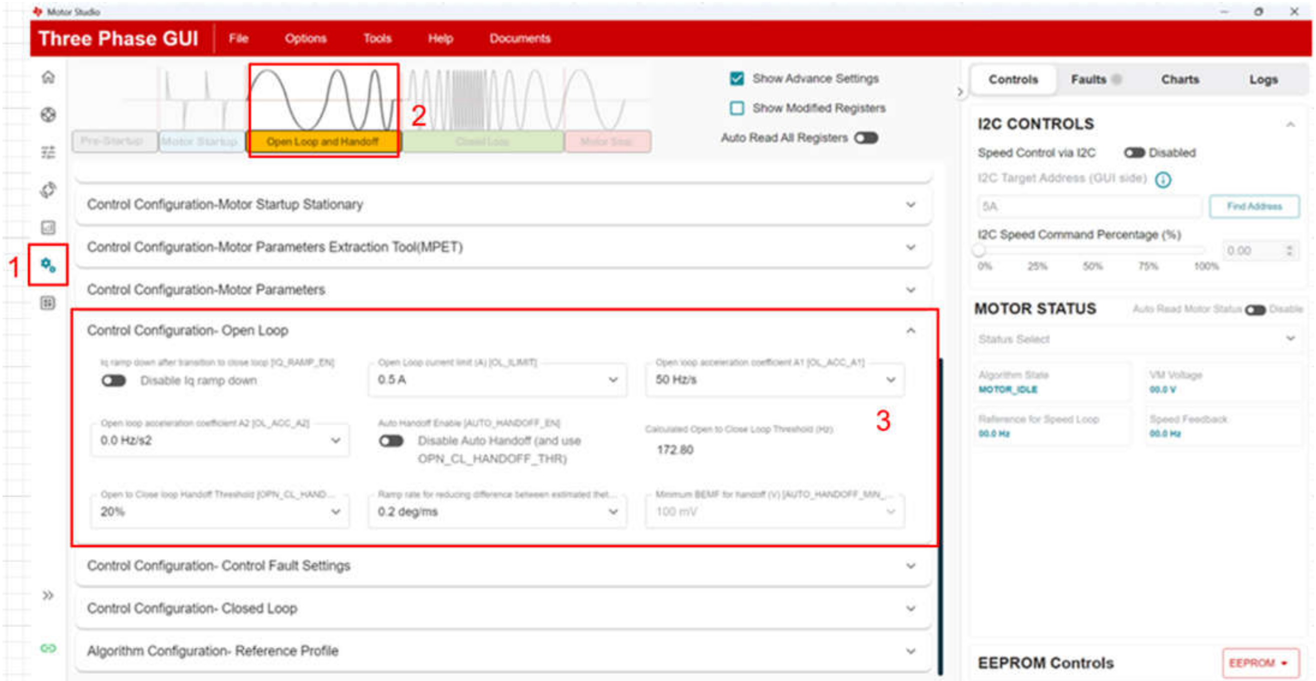

Figure 5-6. Advanced Tuning

This page provides users with the option to navigate different settings organized by motor operating stages, similar to the optimization wizard. Users can select the motor operating stages in the diagram at the top, and the GUI will open the corresponding configuration tab for access.

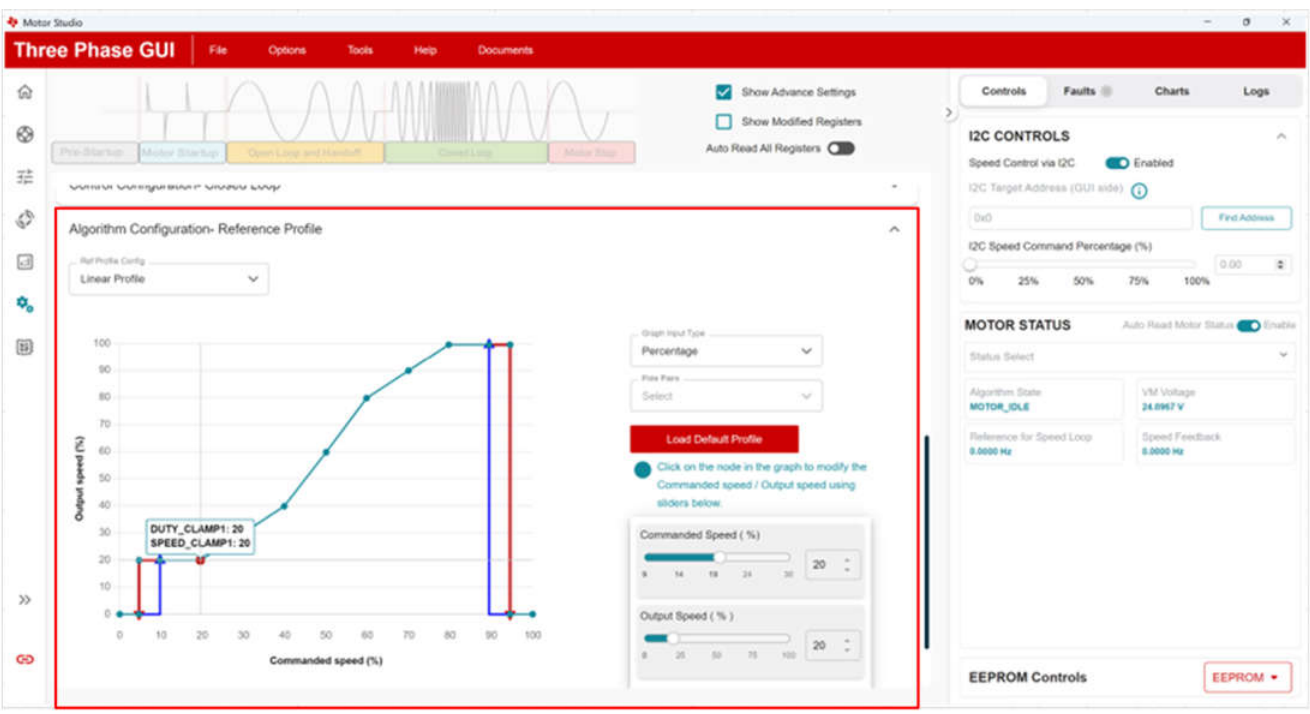

Figure 5-7. Reference Profile

The Advanced Tuning page also provides users with the option to configure the reference profile graphically.

1. Users can choose one of the reference profile options (Linear mode, Staircase mode, and Forward/Reverse mode) from the "*Ref Profile Config*" drop-down menu.
2. The reference profile graph can be plotted with different units (Percentage, RPM, and Hz). The motor's number of poles needs to be selected for reference profile plotting with RPM.
3. "*Load Default Profile*" will load the default graph. Users can modify the profile coordinates using the drag-and-drop method.
4. Users can also alter the profile by selecting one of the coordinates and adjusting the value using the *Command Speed (%)* and *Output Speed (%)* sliders.
5. Users can adjust the speed-duty coordinates by selecting the desired coordinate and moving and dropping it to the desired location.

5.7 Register Map

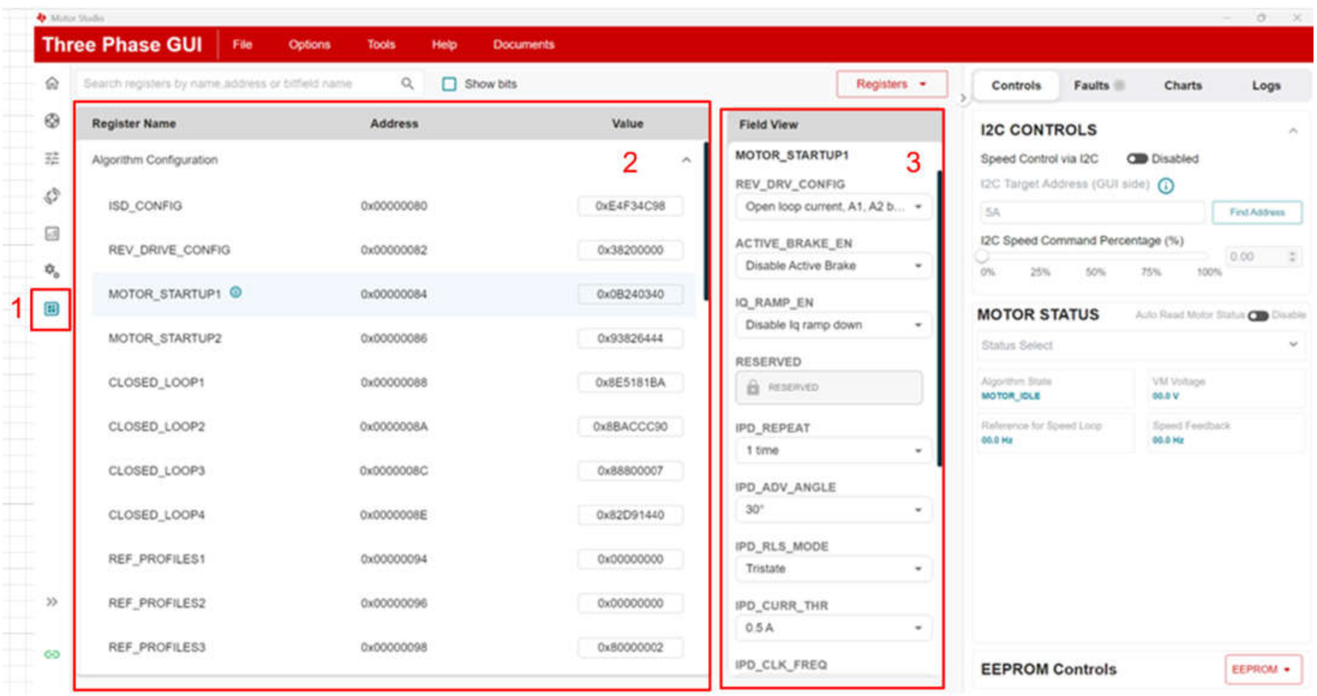

Figure 5-8. Register Map

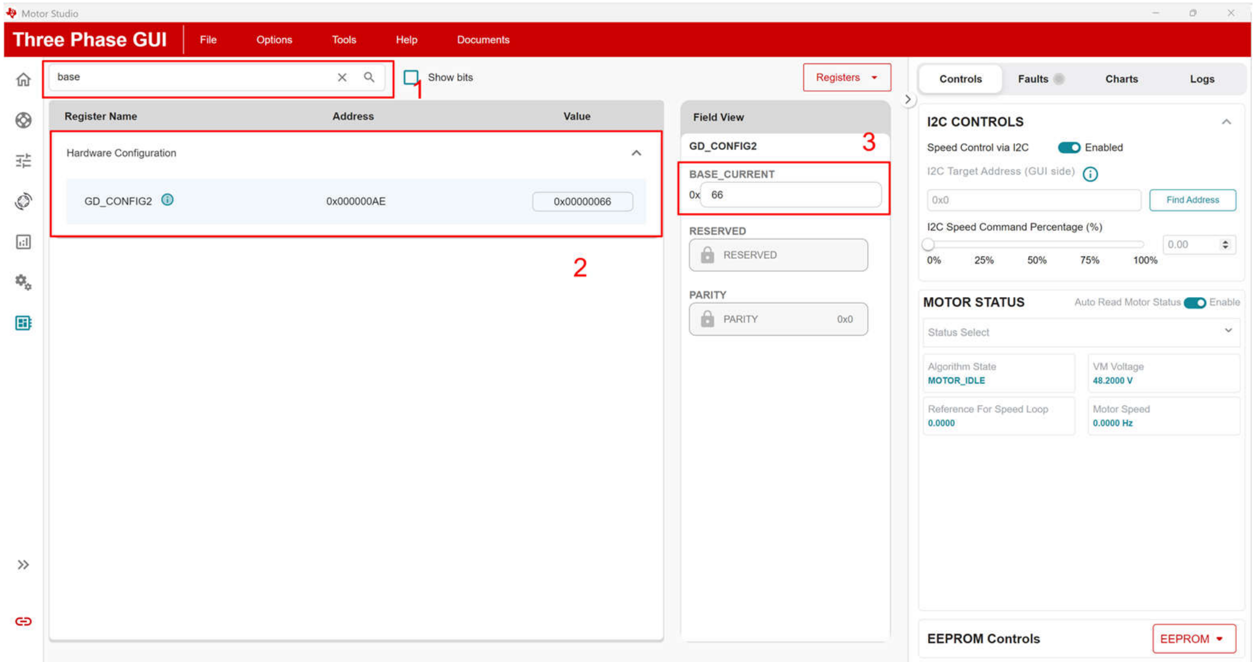

Figure 5-9. Register Map - Search Window

On this page, all device register settings are listed with register name, address, and data. The right-side field view displays the bit fields corresponding to the selected register.

Users can locate registers or register bit fields by typing the register name, register address, or register bit field name directly in the search text box provided at the top.

Example: Searching for the keyword "base" lists all registers and bit field names that contain the "base" keyword.

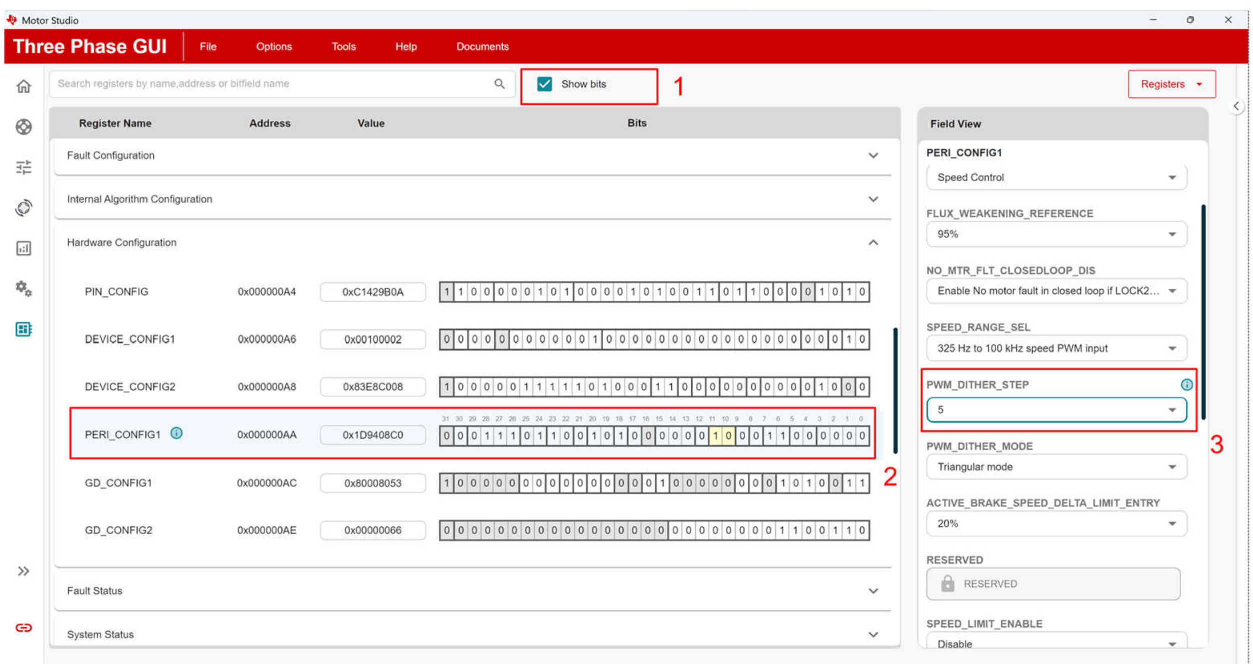

Figure 5-10. Register Map - Show bits

Show bits feature allows user to easily navigate the bit field present in the register.

Specific bit field get highlighted in the bit map window when user select the specific bit field on the *Field View* tab. Otherway, if user select the specific bit field in the bit map then corresponding bitfield get highlighted in the *Field View*.

5.7.1 Register read

Select "*Read All*" or "*Read Selected*" to update *MotorStudio* with the latest register values from the device.

"*Read All*" will read all register content from the device shadow memory and update those values on the *MotorStudio* screen.

"*Read Selected Register*" will only update the register that was selected while executing the read command.

5.8 Control Utilities

5.8.1 Control

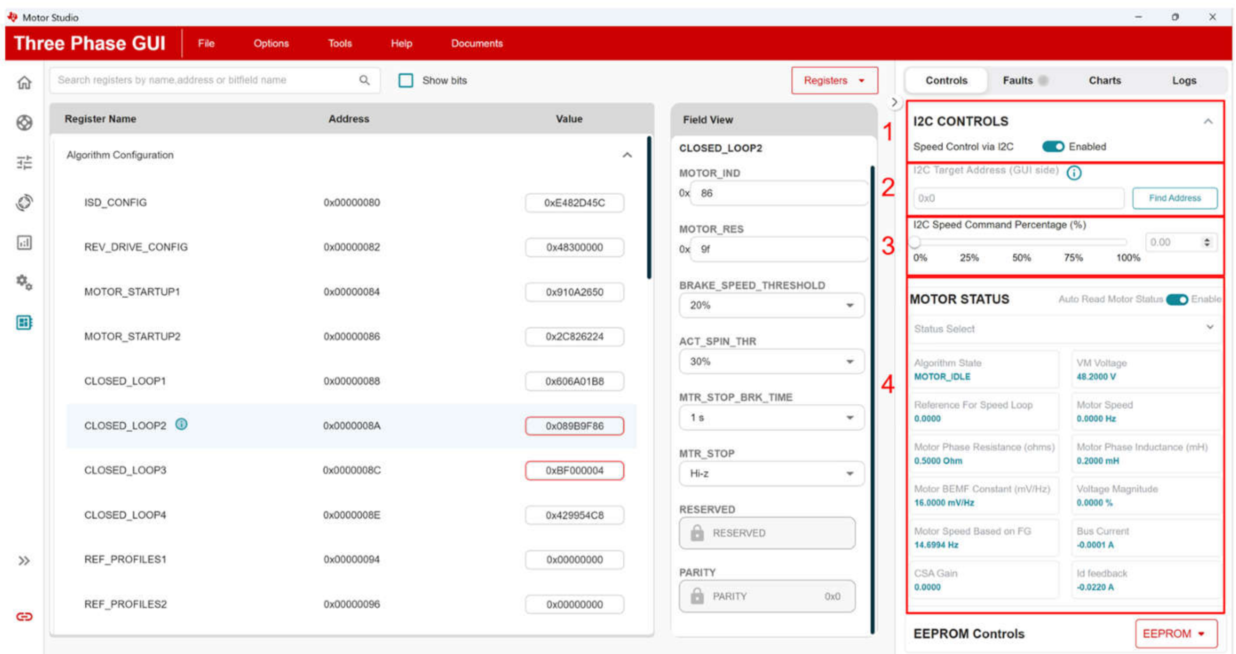

Figure 5-11. Controls

1. *Speed control via I2C*: Enable this option to command the motor speed through the I2C interface.
2. Display the connected device I2C address.
3. Users can enter the speed command percentage through the *I2C speed command percentage (%)* slider or type the speed percentage number directly in the text box next to the slider.
4. Enable "*Auto read Motor Status*" to view the status variables. Users can select the status variables through the dropdown button available in the *Motor Status* section.

5.8.1.1 EEPROM READ/WRITE

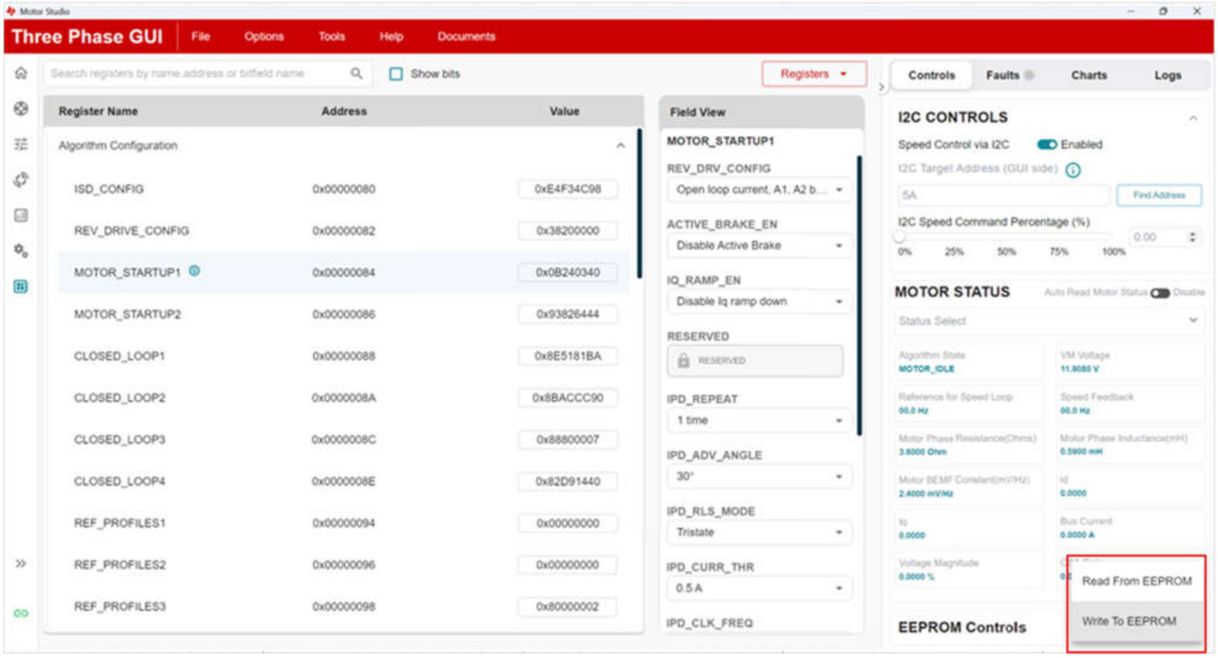

Figure 5-12. Controls - EEPROM Read/Write

1. The GUI displays register data read from the device's shadow memory. The *Read From EEPROM*option moves register data from the device's EEPROM to the shadow memory. Users need to issue a "*Read All*" command from the *Registers* drop-down button to display this EEPROM data on the GUI.
2. The *Write To EEPROM* command allows users to write the updated shadow memory data to the device EEPROM memory.

Note

The device must be in idle mode while performing EEPROM write operations. *Speed control via I2C* option should be enabled with the *I2C speed command percentage (%)* set to 0 to maintain the device in idle state.

5.8.2 Faults

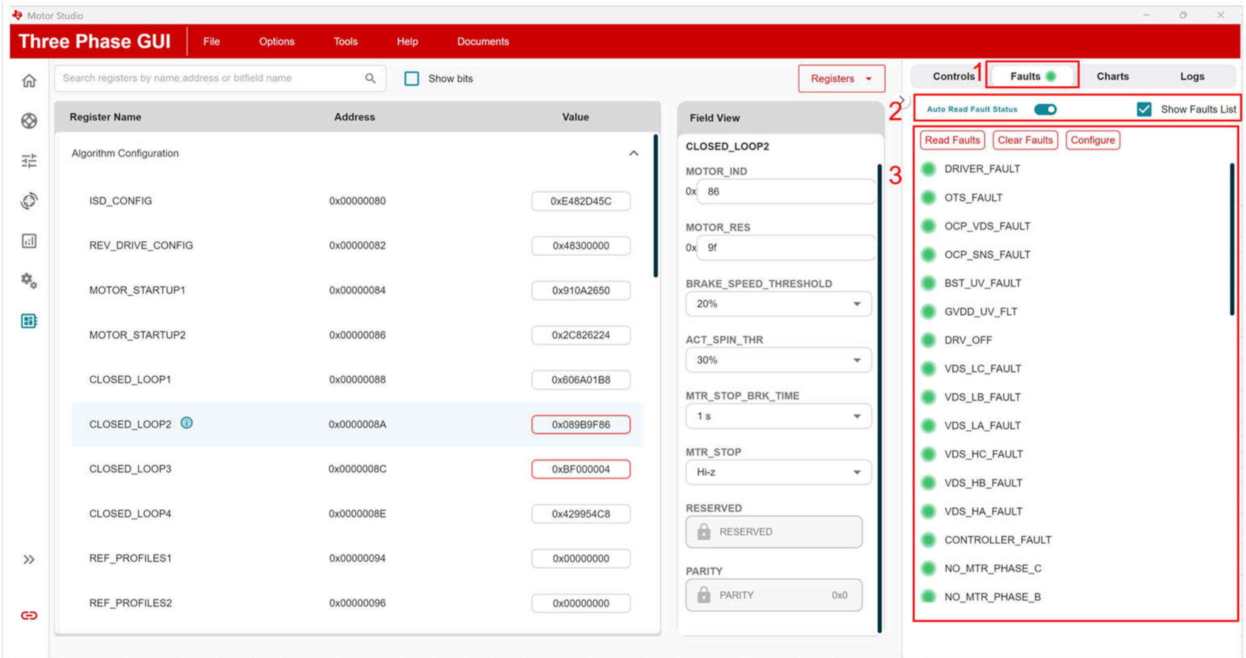

Figure 5-13. Faults

Display the device's fault status. Keep the "*Auto Read Fault Status*" option selected to keep the device fault status updated on the screen.

The "*Clear Faults*" button clears any latched faults that occurred during motor operation and allows the motor to spin.

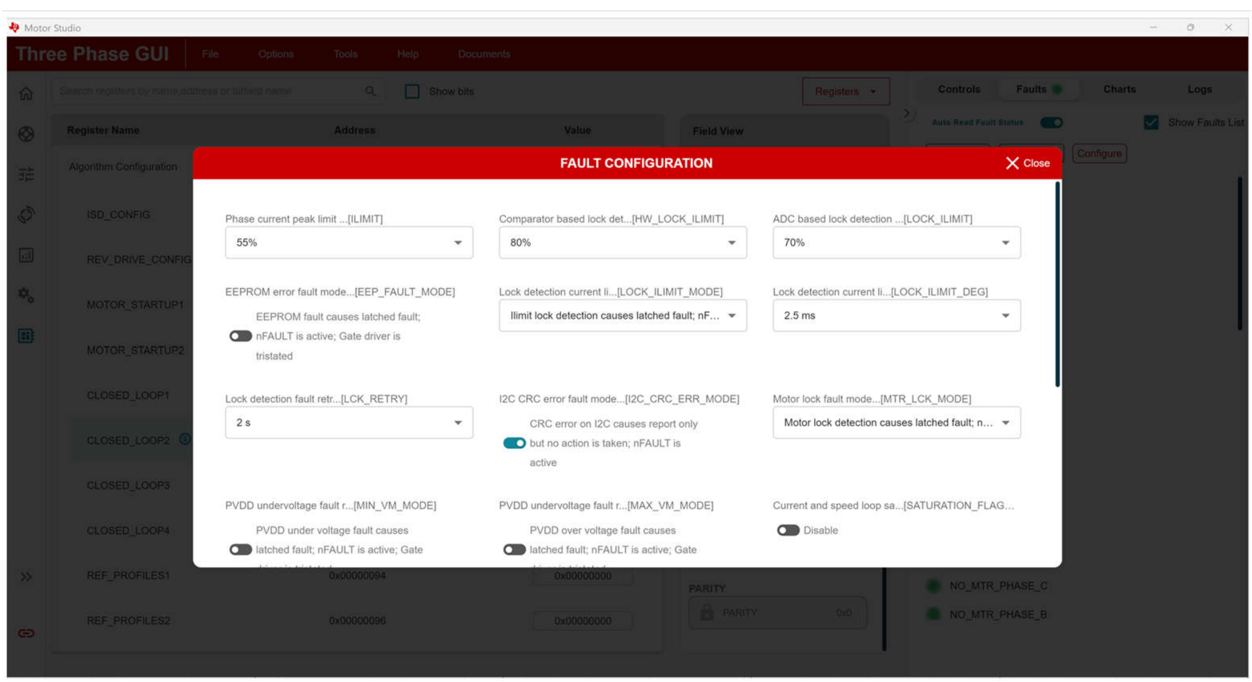

Figure 5-14. Faults - FAULT CONFIGURATION

The *Configure* button helps users navigate different fault configurations that the device supports.

Upon a fault, users can open the fault configuration window and adjust the parameters based on the application requirements.

5.8.3 Charts

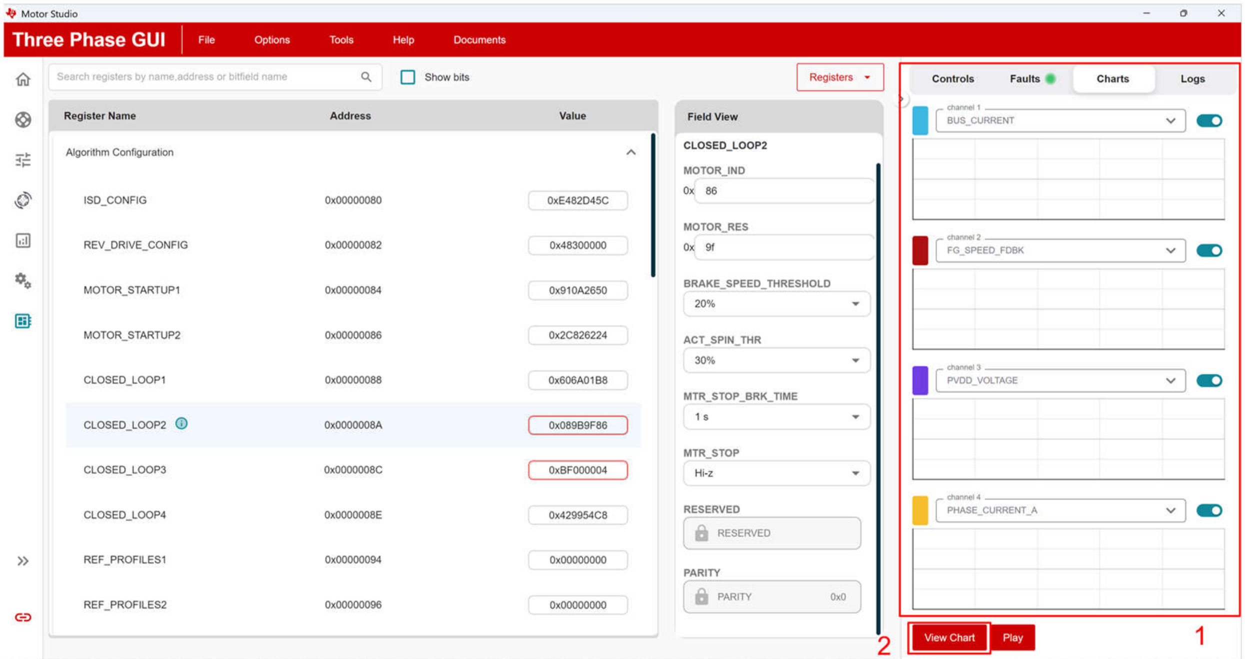

Figure 5-15. Charts

All motor status variables can be selected for viewing in the charts window. Four motor status variables can be plotted on the chart simultaneously.

Select the desired motor status variables for all four channels and select the Play button to view the variable status on the chart window.

The chart screen can be maximized by selecting the "*View chart*" option.

Figure 5-16. Charts - Oscilloscope

1. Users can adjust the speed command through the *I2C Speed Command Percentage (%)* slider.
2. Charts can be viewed individually or in a unified way. In the unified view, two motor status variables are displayed together for easy correlation.
3. Graph resolution can be selected by adjusting the *Time/Samples* value.
4. Graph data can be downloaded in CSV format for users to analyze the data manually by selecting *Download* button.
5. Selecting the *Play* button will start the graph plotting on the screen.
6. The maximum time scale of the graph can be adjusted by selecting the desired number in "*Grid Max*."

5.8.4 Logs

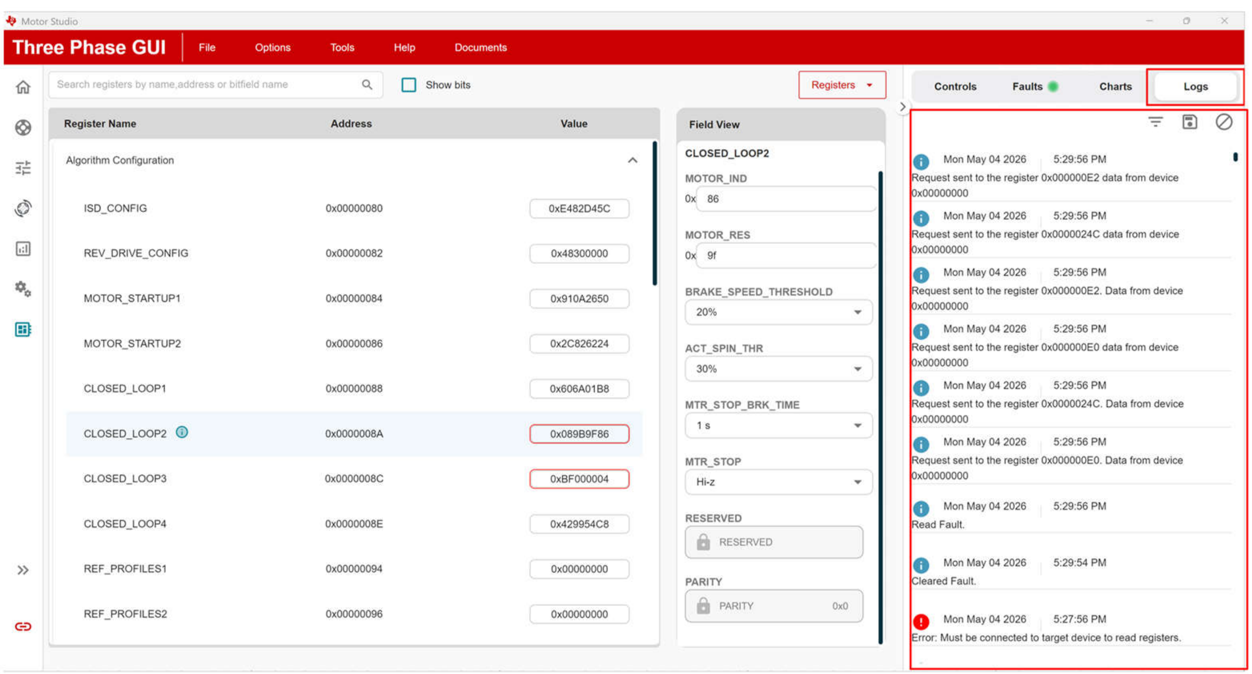

Figure 5-17. Logs

All *MotorStudio* activities are captured in the log window with a standard timestamp.

The *Filter* option enables users to select specific categories of logs.

Users can *Save* the logs to the local PC drive for offline review.

The *Clear* option helps to clear the logs on the screen.

5.9 Other Utilities

5.9.1 Scripting Tool

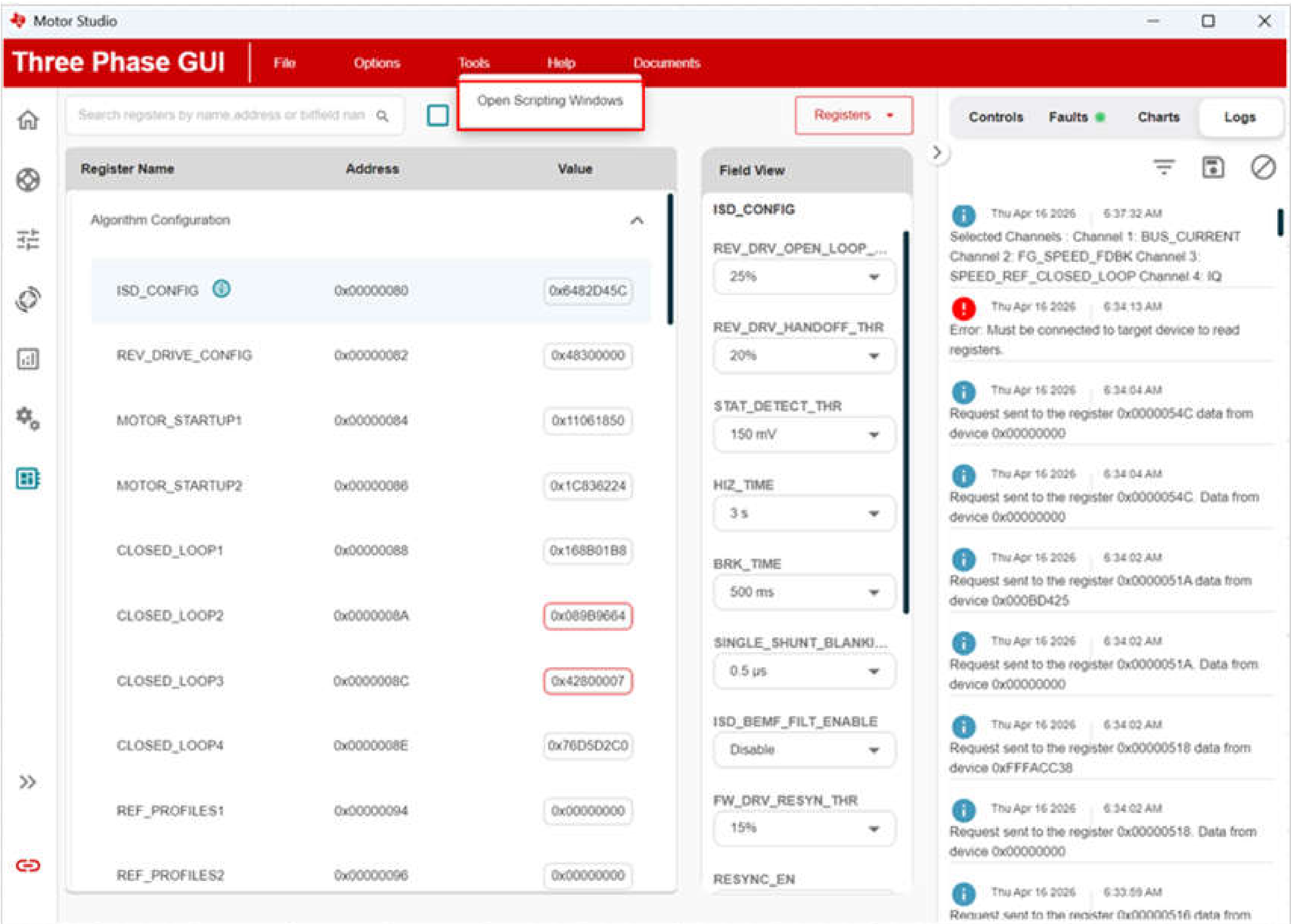

Figure 5-18. Open Scripting Windows

This is *MotorStudio's* built-in scripting feature that allows users to automate device register read/write operations with all programming logic to perform tasks such as motor start/stop operations and periodically reading motor fault status.

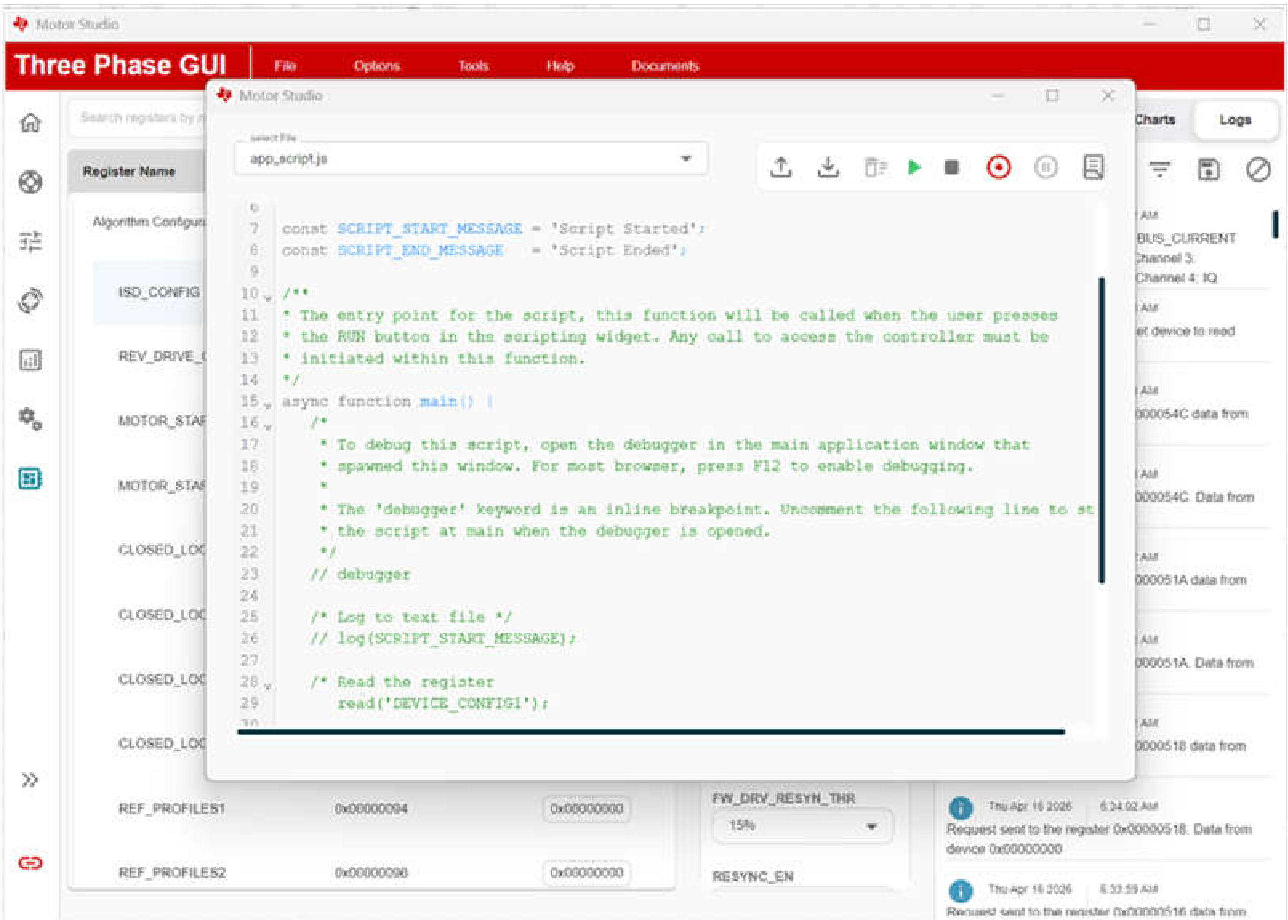

Figure 5-19. Scripting Window

Select "*Open Scripting Windows*" under the *Tools* menu to open the scripting tool. The scripting tool will open with a sample script.

Note that the existing demo script file cannot be modified. Users need to create a new .js file to write a new script and select the "*Open Script*" option to bring the new script into the scripting tool.

Select "*Save Script*" if you make any changes to the script while it is open in the scripting tool.

Users can select the script file from the "*Select File*" dropdown box and select the *Play* button to run the script.

Script execution can be terminated by selecting the *Stop* button.

5.9.2 Compare Tuning File Tool

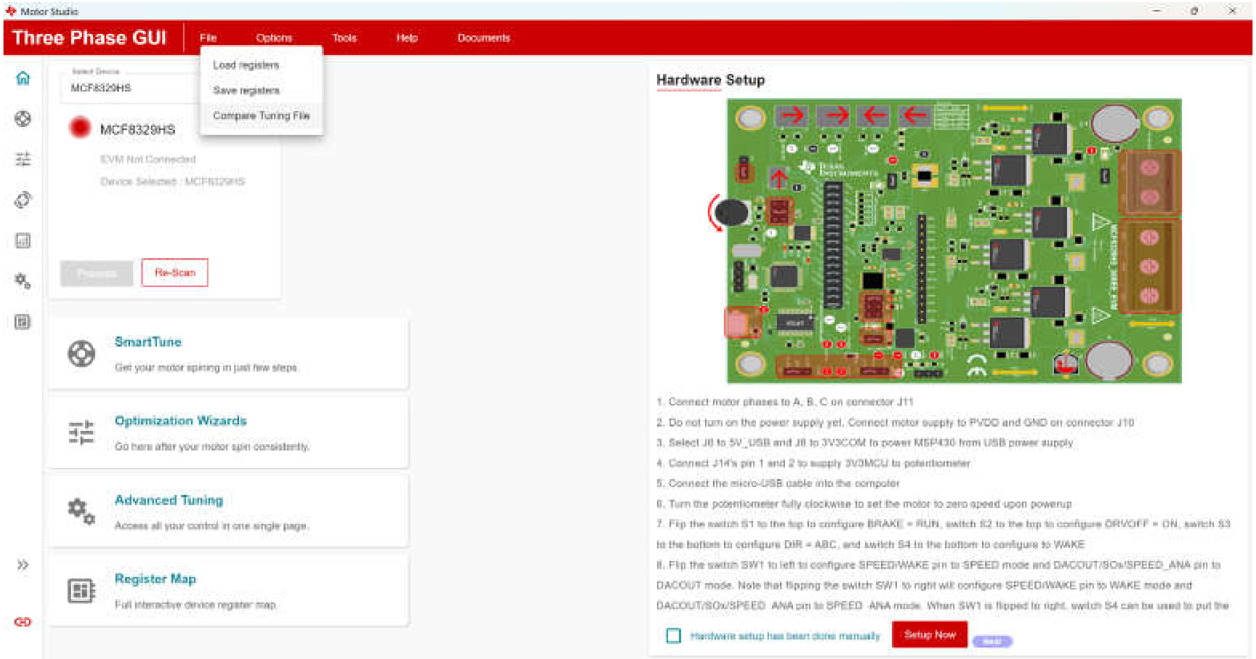

Figure 5-20. Open Compare Tuning File

This tool helps users to compare the configuration present in two different tuning files.

The tool can be opened by selecting "*Compare Tuning File*" option from the *File* menu.

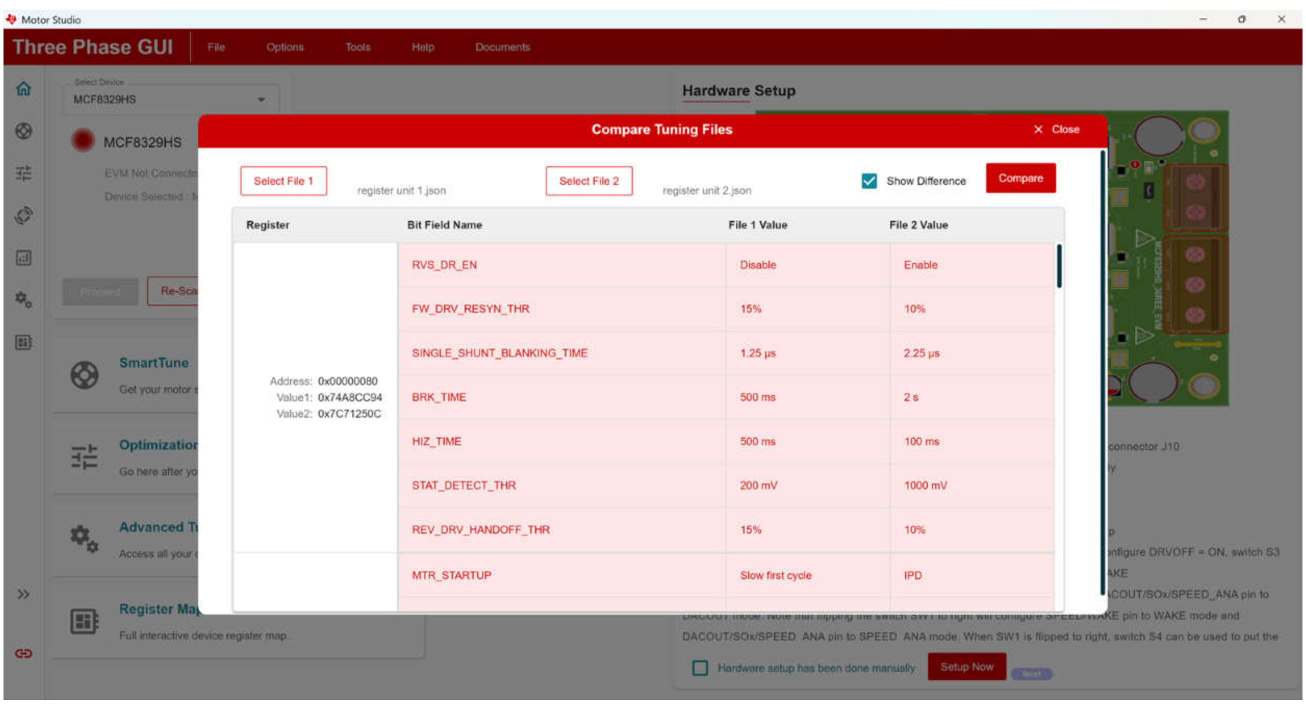

Figure 5-21. Comparing Tuning Files

Select two files that need to be compared by selecting the "*Select file 1*" and "*Select file 2*" buttons, then select the "*Compare*" button to view the configuration differences between the selected tuning files.

6 Spinning the motor

6.1 Connecting MCF83xx to the GUI

Before connecting the MCF83xx EVM to the computer, start up the *MotorStudio* application and select specifc product from the drop down. Click on *Proceed* and then click on the *Setup Now* button for instructions on how to connect power, connect a motor, and configure the jumpers and switches on the EVM.

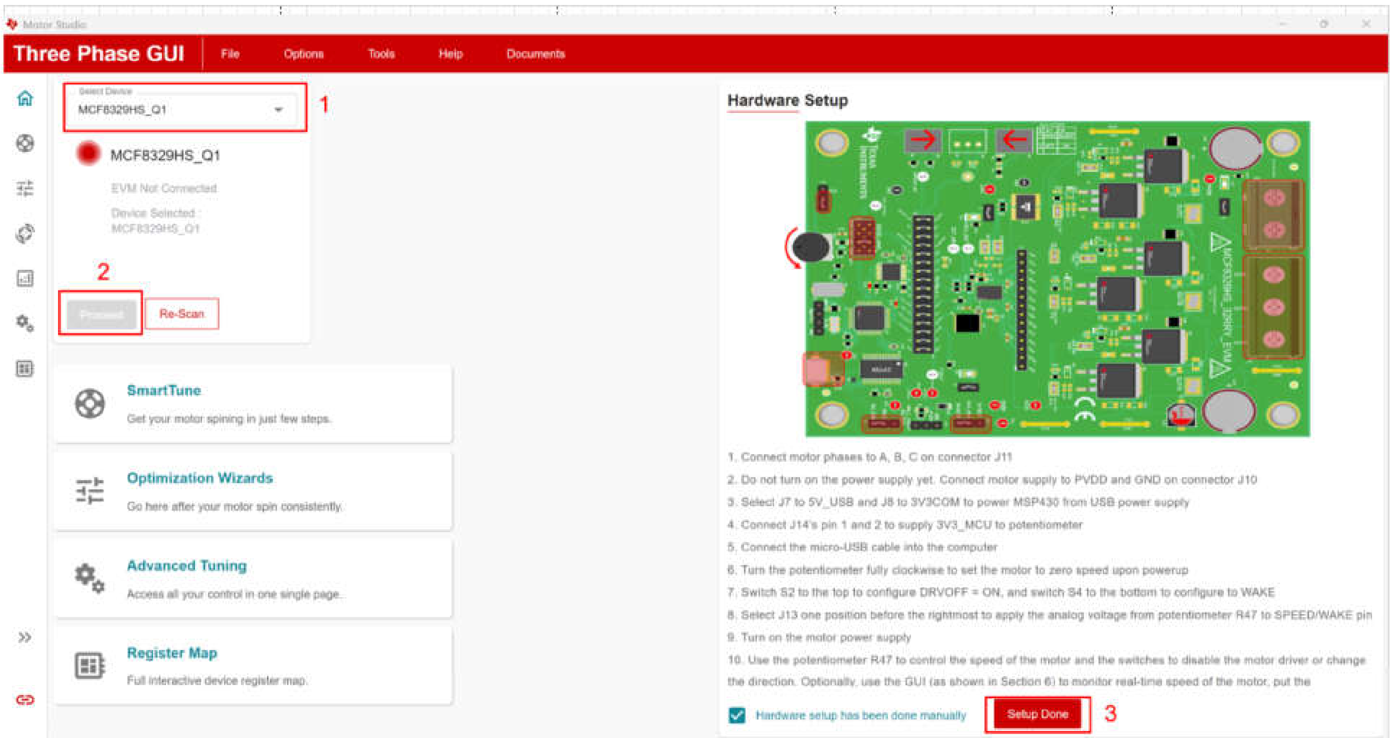

Figure 6-1. EVM Hardware Setup

Once the hardware setup is completed, turn on the power supply connected to the EVM. After a few seconds, *MotorStudio* should connect to the EVM and the two icons outlined in Figure 6-2 will turn green. If the EVM is not connecting click the *Re-Scan* button.

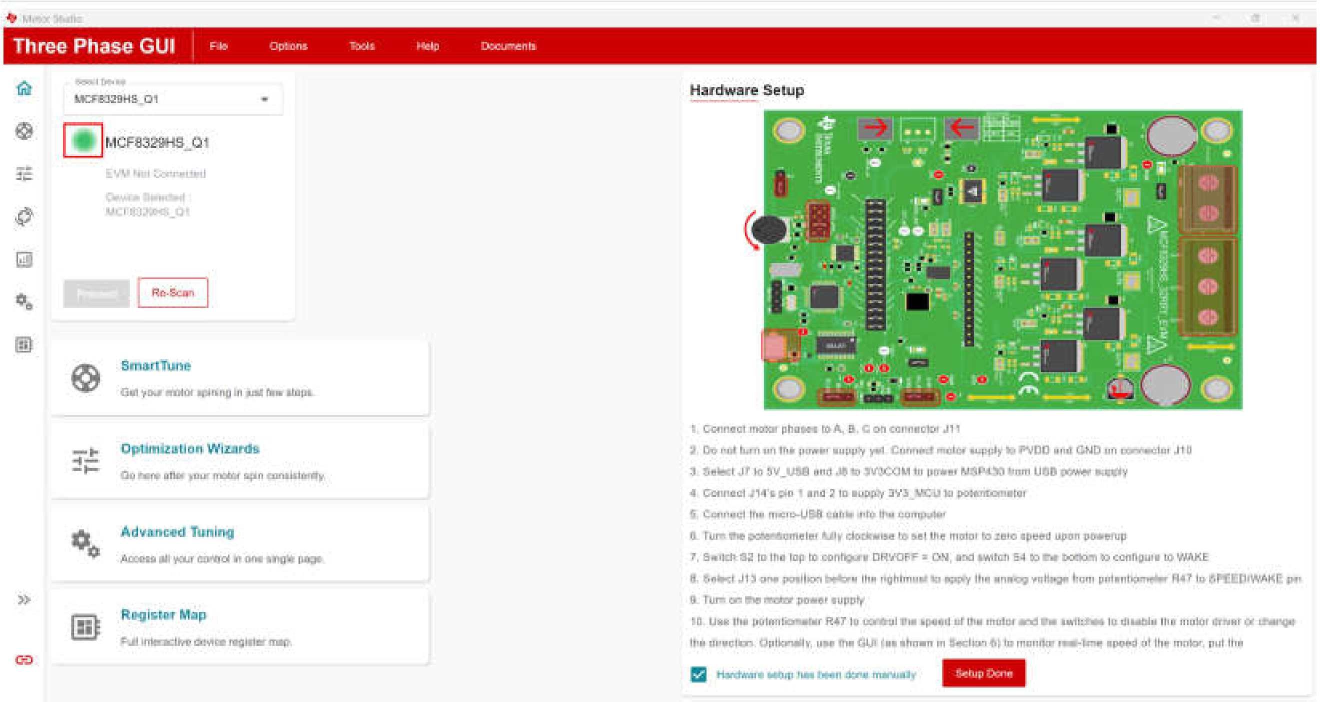

Figure 6-2. EVM Connected Indicators

Note

If the GUI is not able to connect to the GUI after a minute, disconnect the EVM from the PC, restart the *MotorStudio* GUI. After *MotorStudio* has started again, reconnect the EVM to the PC.

6.2 SmartTune Execution

To initiate the configuration process, select *SmartTune* via the *MotorStudio* home page or the left-side navigation menu. Input the required motor parameters, including *Rated Voltage*, *Rated Current*, *Rated Speed*, and EVM *Shunt Resistance* value (5mΩ).

Select the appropriate load condition from *Basic Information* section in *SmartTune* page, *Load* for motors with mechanical load or *No Load* for motors without shaft load.

Choose the *Control Mode* that best suits your application requirements from the available device options, then click *Generate Configuration* button to start *SmartTune*.

Select the ⓘ adjacent to each text box to access additional information about the respective parameters.

Note

Shunt Resistance must be configured only when using MCF8329x products. This specification is not necessary for MCF831x products

---

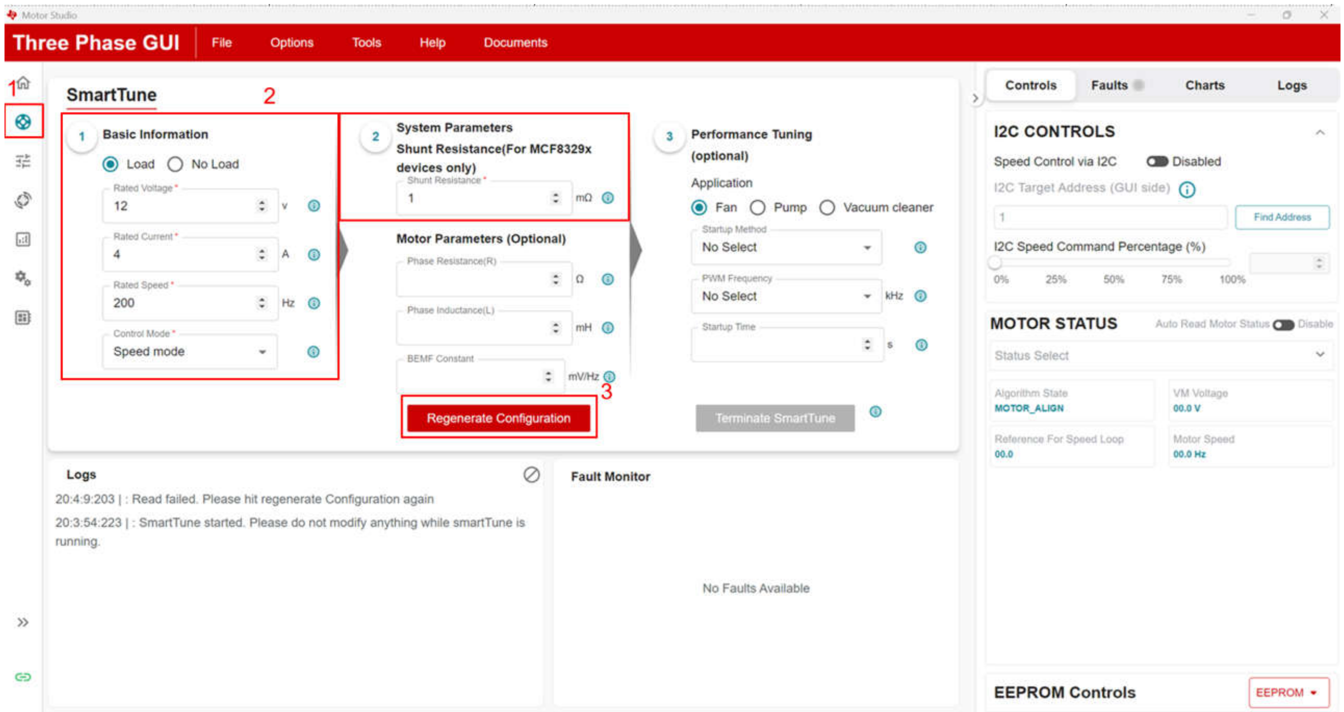

Figure 6-3. *SmartTune* Basic Configuration

SmartTune executes an algorithm that measures the motor's electrical parameters, including motor phase resistance, inductance, BEMF constant, and control loop coefficients based on the user-selected control mode. Additionally, it configures numerous parameters such as startup mode, open-loop current and acceleration rate, and various fault configurations necessary to achieve Level 1: Spin the motor functionality as highlighted in Figure 6-4.

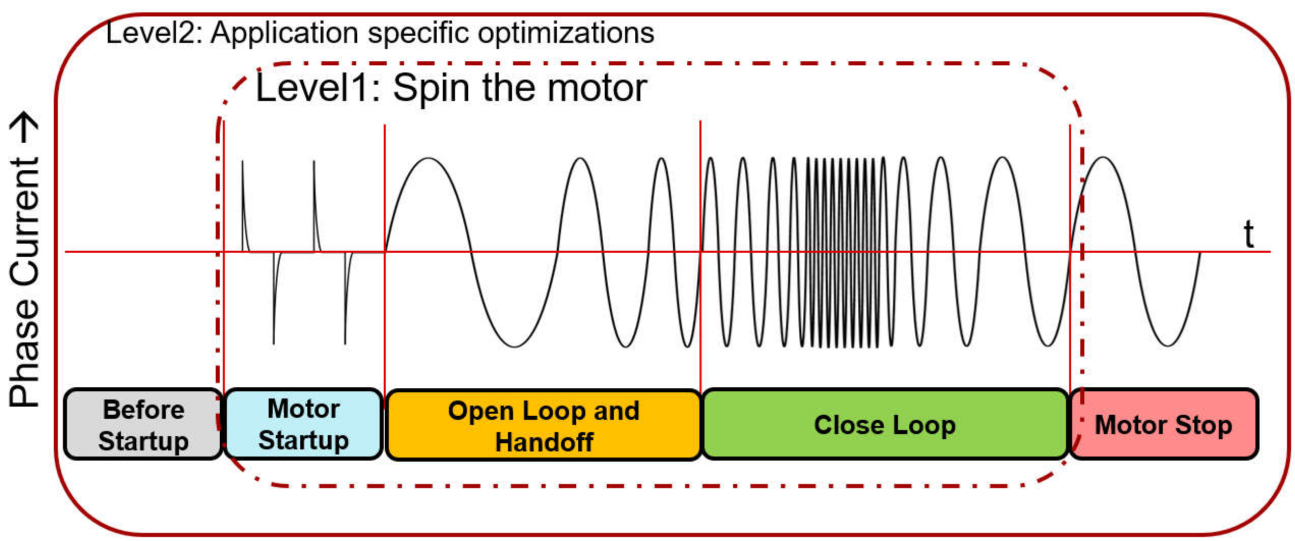

Figure 6-4. Stages of Tuning

Upon successful completion of *SmartTune*, a *Configuration Successful* message appears below the *Regenerate Configuration* button and in the logs window. Users can then operate the motor by following the instructions provided in the *Testing for Successful Startup into Closed Loop*.

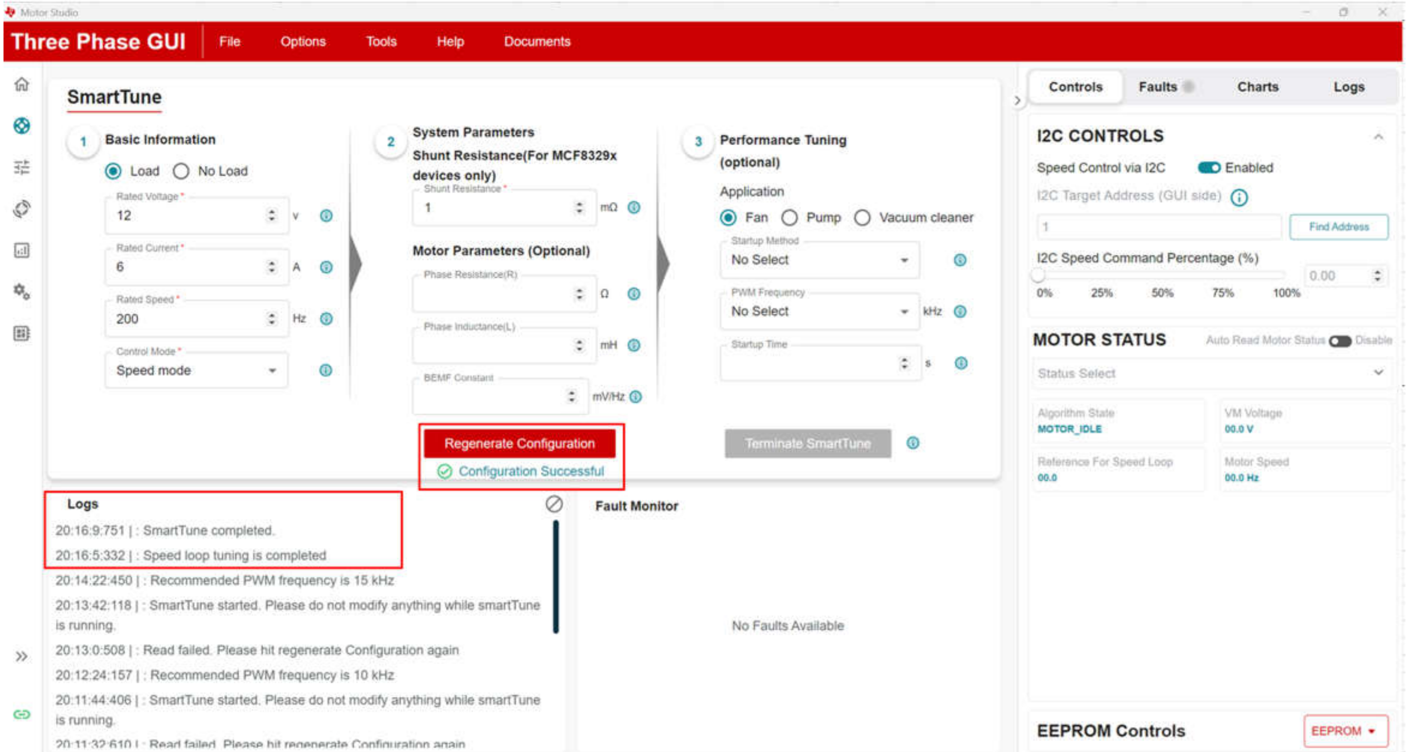

Figure 6-5. *SmartTune* Success

Users can terminate the *SmartTune* execution process at any time by selecting the *Terminate SmartTune* button.

Under Performance Tuning, *SmartTune* allows users to select their *Application* and automatically generates optimal motor parameters. Users can also manually enter specific parameters such as *Startup Method*, *PWM Frequency*, and *Startup Time* as part of the performance tuning options. These manual configurations serve to override the *SmartTune* generated settings.

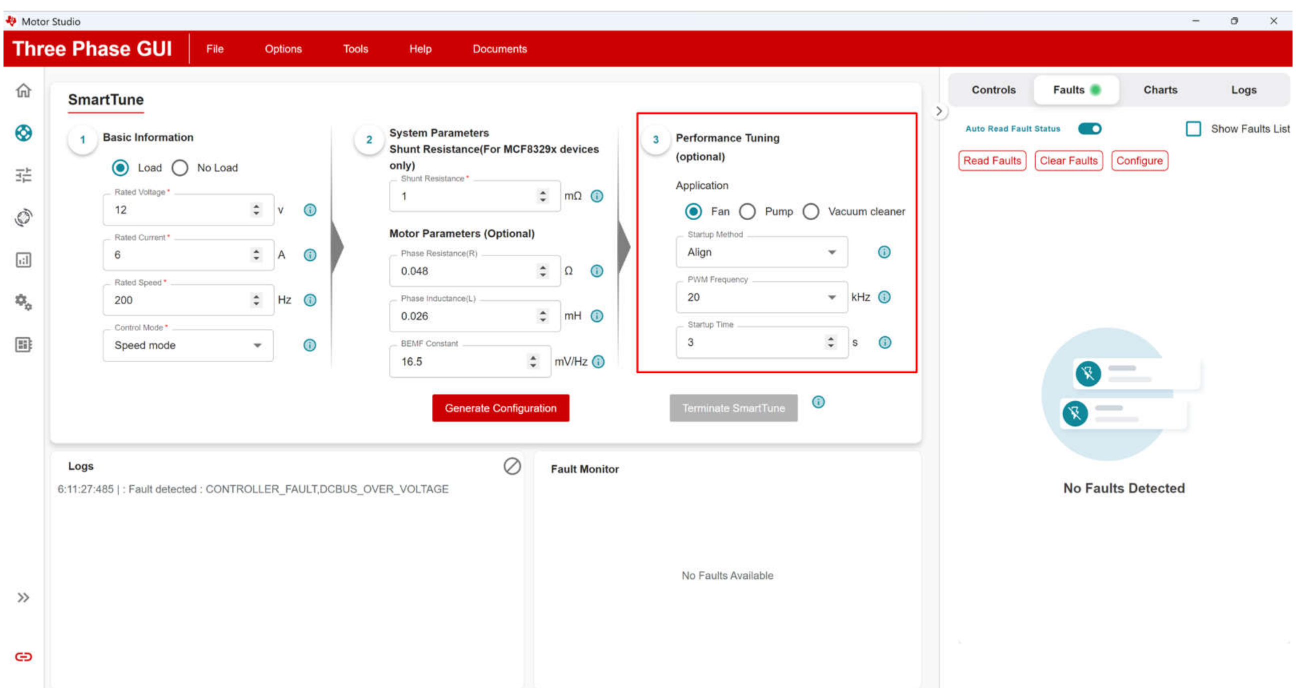

Figure 6-6. *SmartTune* Performace Tuning

If *SmartTune* fails to generate the motor tuning parameters, users may attempt to run the tool again by entering the motor parameters such as *Phase Resistance (R)*, *Phase Inductance (L)*, and *BEMF Constant* in the *Motor Parameters (Optional)* section.

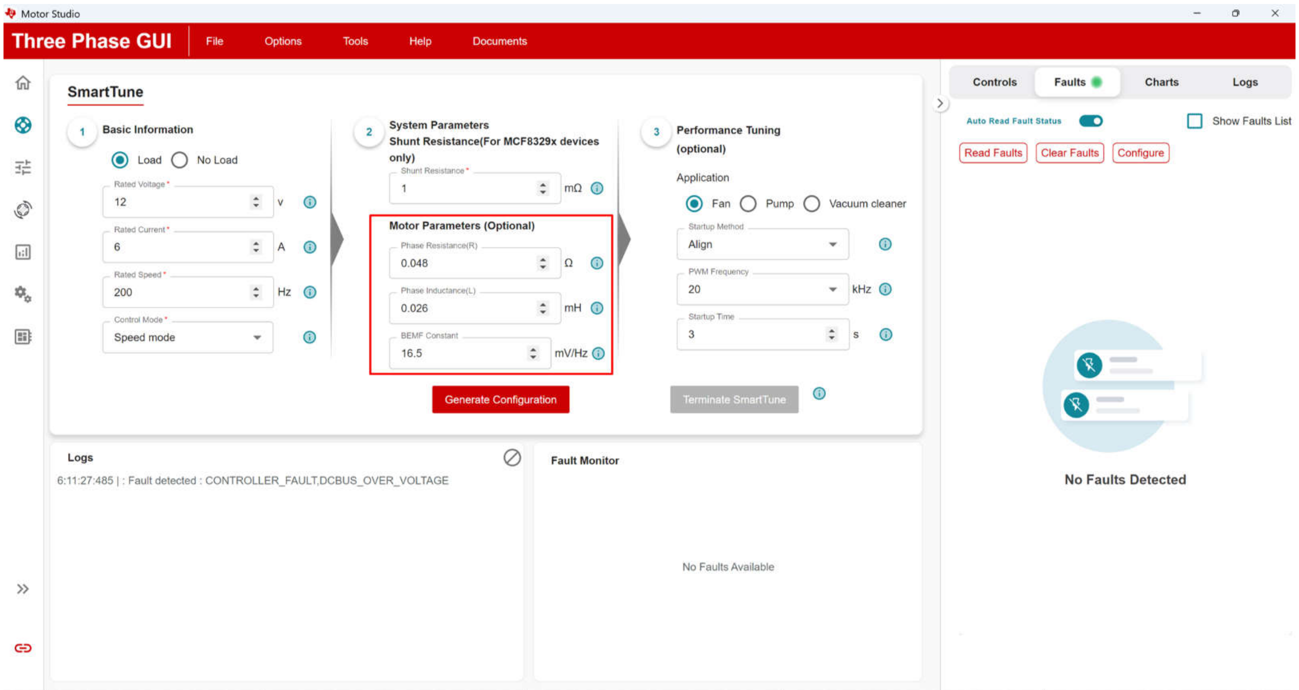

Figure 6-7. *SmartTune* Optional Parameters

If *SmartTune* fails to complete successfully, review the SmartTune FAQ page, which provides details on common failure conditions and their solutions.

In the event that *SmartTune* repeatedly fails to generate motor tuning parameters following the input of motor electrical parameters, users are recommended to configure motor parameters manually by following the applicable product specific tuning guide.

6.3 Improving Current and Speed Regulation

The MCF83xx products operates the motor utilizing Field Oriented Control (FOC), with current and speed regulation accomplished through Proportional-Integral (PI) controllers. The device automatically calculates the current PI control loop coefficients based on the motor electrical parameters, specifically resistance and inductance. These calculated values are displayed in the CURRENT_PI register.

To change the current loop gains, users can copy the device calculated [CURRENT_LOOP_KP] and [CURRENT_LOOP_KI] values from the CURRENT_PI register and input the modified values into the corresponding [CURR_LOOP_KP] and [CURR_LOOP_KI] settings within the CLOSED_LOOP3 register.

MotorStudio Close loop tuning page provides functionality to adjust current loop coefficients through the *"Time Domain Graph"*. User can adjust the loop coefficients *"Kp coefficient for current Iq and Id loop"* and *"Ki coefficient for current Iq and Id loop"*, while monitoring *Current(Amps) vs Time (sec)* through the *"Time Domain Graph"* by selecting the *"Plot Current Response"* button.

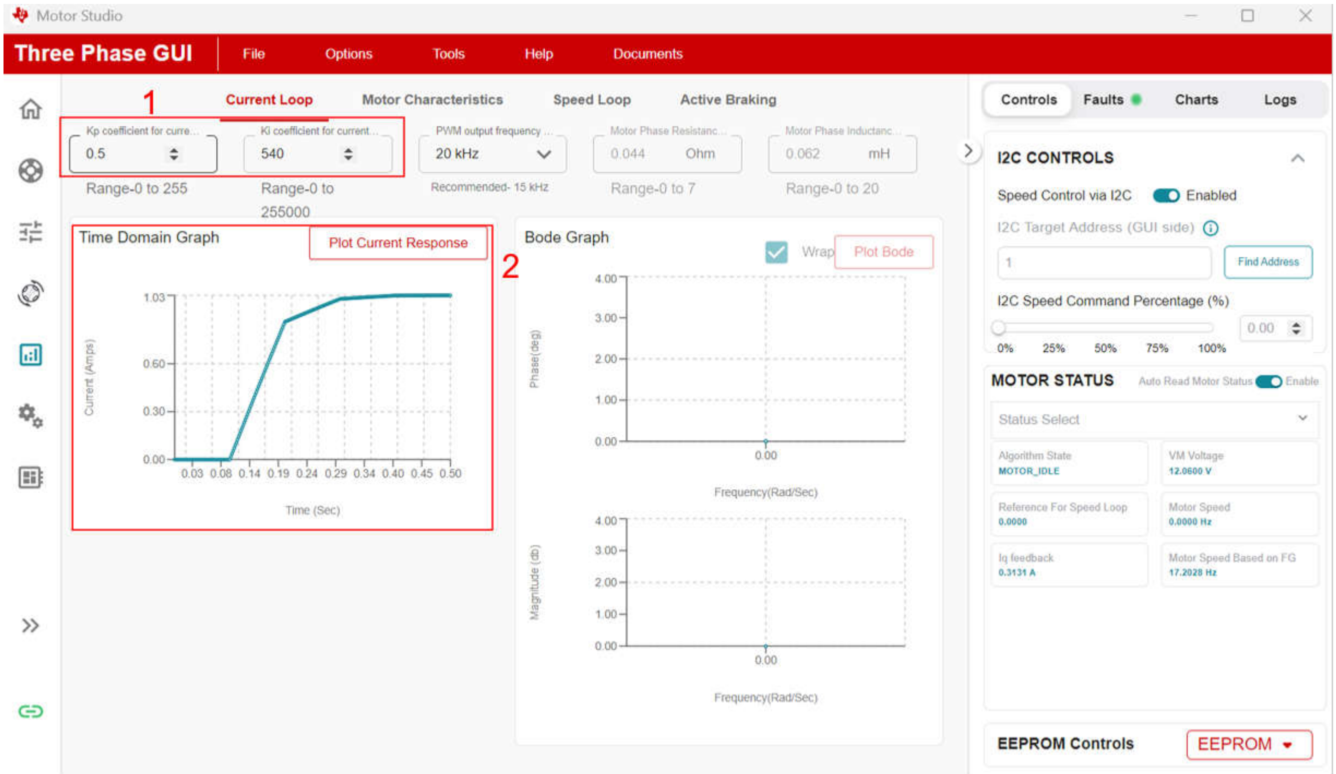

Figure 6-8. Current Loop Tuning

Speed PI loop coefficients are calculated based on motor mechanical parameters, including motor inertia and friction coefficients. In cases where users lack access to these motor mechanical parameters, the *Closed Loop Tuning* page offers functionality to determine the speed loop Kp [SPD_LOOP_KP] and Ki [SPD_LOOP_KI] coefficients for a specified motor. Follow the procedure outlined below to calculate the speed loop coefficients.

1. Set the motor to current control mode using PERI_CONFIG1 -> [CTRL_MODE] = "Current Control" configuration.
2. Complete the *"Current (Ipeak-Amps)"* vs *"Speed Feedback(Hz)"* table provided in the *Motor Characteristics* tab by issuing I2C speed commands that span the entire motor operating speed range. Both the *"Current (Ipeak-Amps)"* and *"Speed Feedback (Hz)"* data can be obtained from the *"Iq feedback"* and *"Motor Speed Based on FG"* parameter within the MOTOR STATUS window.
3. While stopping the motor, record the motor coasting *"Time (sec)"* along with the *"Initial Speed(hz)"* and *"Final Speed (Hz)"*, then enter these captured values into the designated text fields.
4. Select the *"Measure Motor Characteristics"* button, the software will calculate the motor mechanical characteristics and display these parameters in the *Motor Characteristics* window.

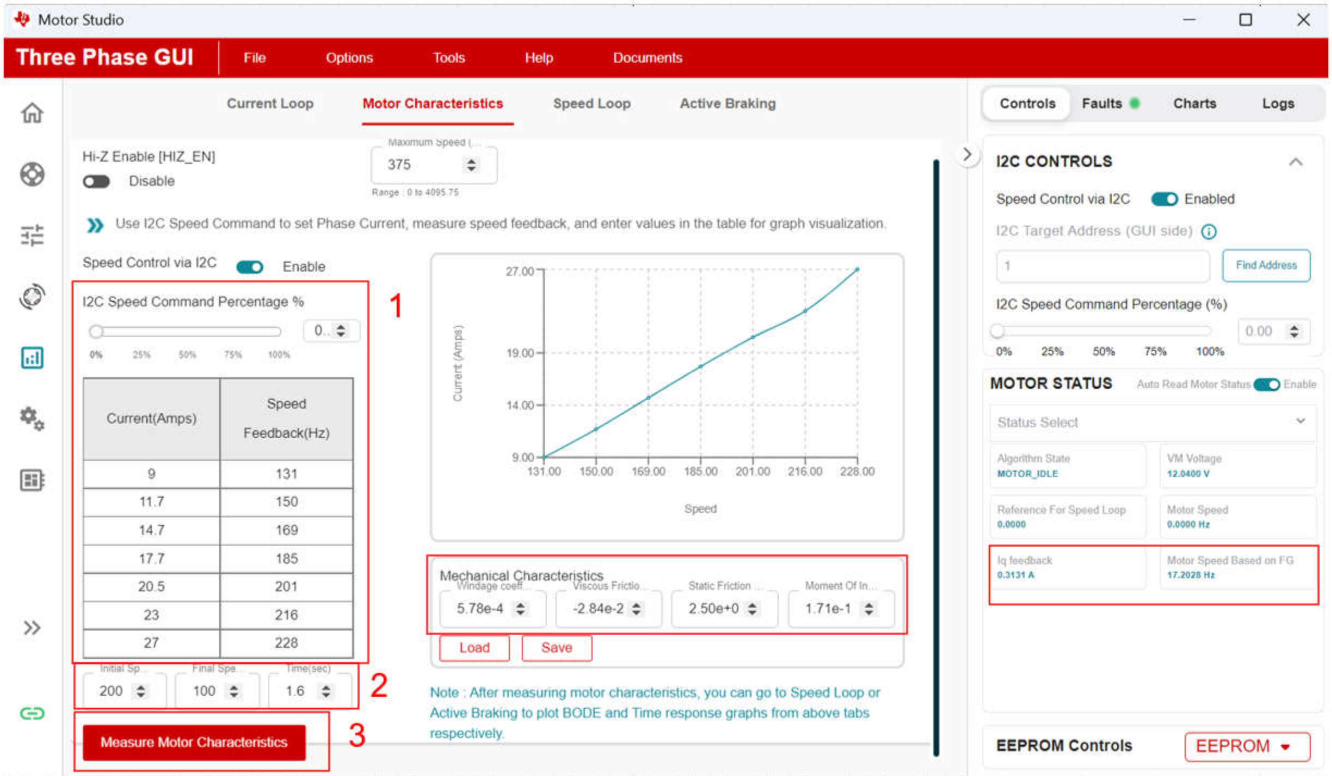

Figure 6-9. Motor Characteristics

Navigate to the *Speed Loop* tab, enter the desired motor *"Initial Speed (Hz)"* and *"Final Speed (Hz)"* with *"Maximum Simulation Time (sec)"* and select the *"Plot Speed Response"* button in the *"Time Domain Graph"* to generate a *Speed (Hz) vs Time(sec)* response plot for the calculated speed loop coefficients.

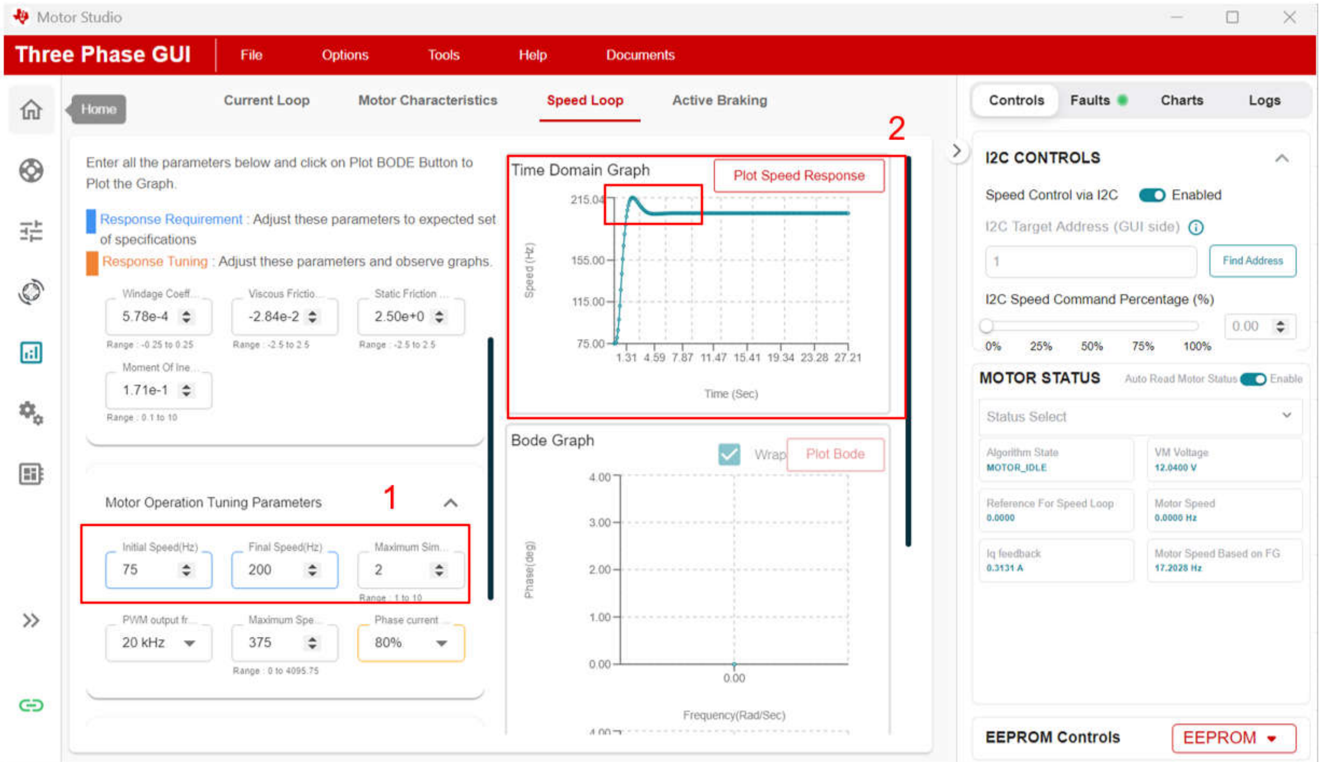

Figure 6-10. Motor Operation Tuning Parameters

User can adjust *"Kp coefficient for speed loop"* and *"Ki coefficient for speed loop"* values and monitor the response again through the *"Time Domain Graph"* for further tuning.

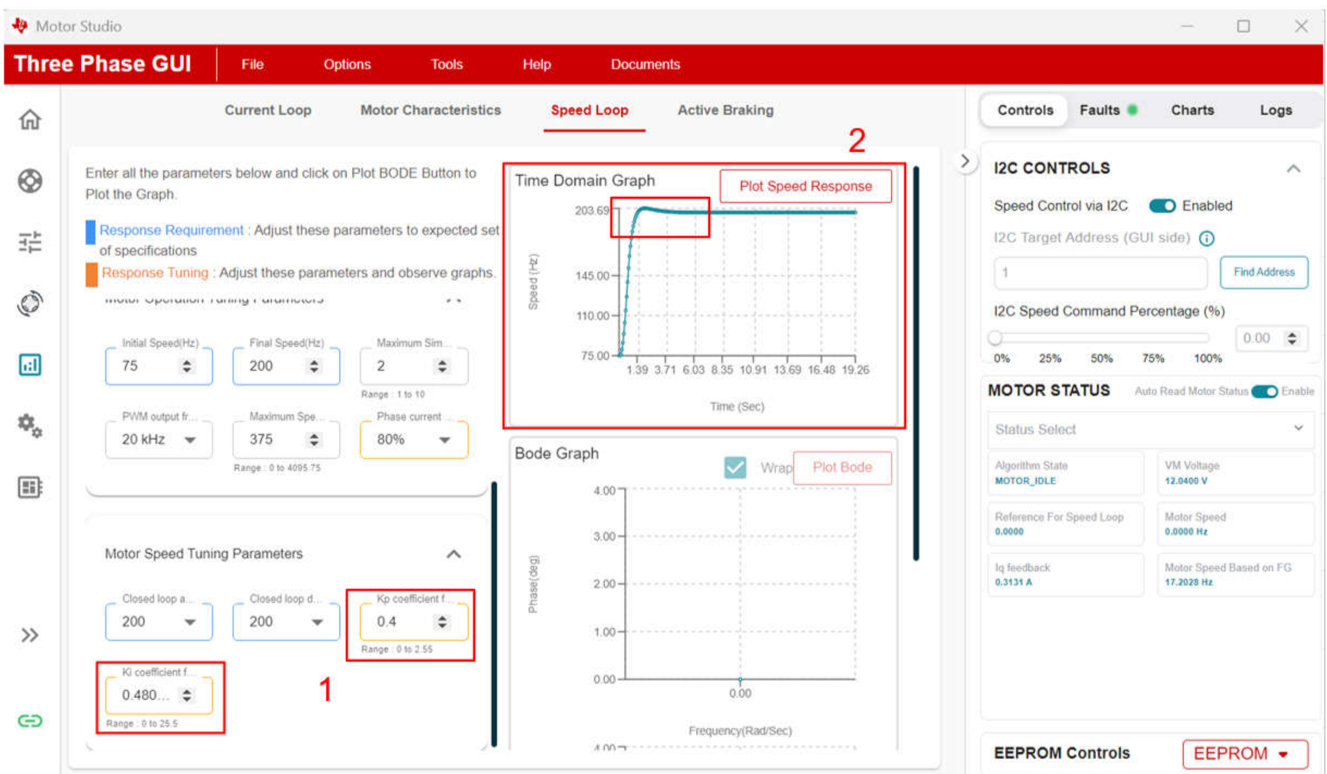

Figure 6-11. Motor Speed Tuning Parameters

6.4 Testing for Successful Startup Into Closed Loop

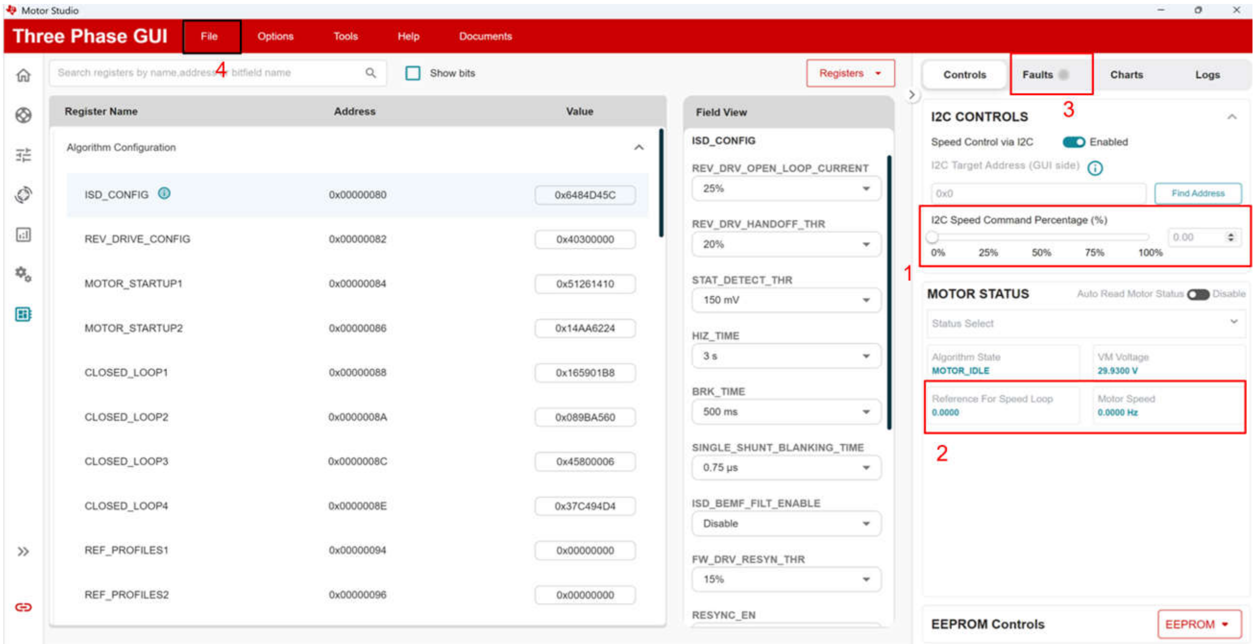

Figure 6-12. Spinning the Motor

1. Enable the *Speed command via I2C* using the slider button.
2. Apply a nonzero speed command using the slider or text box within the *Controls* section. Once a speed command is provided, the motor should begin to spin and accelerate until the motor reaches the target speed.
3. After the motor stops accelerating check that the values in *Reference for Speed Loop* and *Speed FDBK* under the *Motor* Status section are close to the same value.
4. Check for any faults if the *Faults* tab shows a red circle. If a fault has been reported, go to fault handling topic in the product specific tuning guide and follow the debug steps to correct the fault.
5. Once the motor is able to spin into closed loop and not trigger any faults, stop the motor and save the register configuration to a json file by clicking on *File -> Save Registers*. In the window that pops up, select *Json File* and click on the *Save* Button.
6. To have configuration for the registers covered in section 7 of the MCF8329HS datasheet load when the device powers-up these register values can be loaded into EEPROM. To write the configured register values to EEPROM, click the *EEPROM* drop down located at the bottom right of *MotorStudio* and select the *Write To EEPROM* option. Click the *Yes* button in the window that pops up.

7 Summary

MotorStudio is a user-friendly graphical interface tool that streamlines the tuning process for brushless DC (BLDC) motors. It accelerates device evaluation and product development by allowing users to easily initialize motors and optimize parameters. This guide specifically covers how to utilize Motor Studio's features for tuning BLDC motors with MCF83xx controllers.

8 References

- Texas Instruments, *MCF8316D Sensorless Field Oriented Control (FOC) Integrated FET BLDC Driver*, data sheet.
- Texas Instruments, *MCF8329HS-Q1 Sensorless Field Oriented Control (FOC) Three-phase BLDC Gate Driver*, data sheet.

9 Revision History

Changes from Revision * (April 2025) to Revision A (April 2026) **Page**

- Deleted *Motor Runs* section.......... 3
- Added System Requirements section.......... 3
- Added *MOTORSTUDIO* GUI Installation section.......... 3
- Updated Getting Started with *MOTORSTUDIO* section.......... 4
- Added Spinning the motor section.......... 19