

User's Guide

MCF8329HS Tuning Guide



ABSTRACT

This tuning guide provides step-by-step guidance to setup the MCF8329HS, connect the MCF8329HSEVM to *MotorStudio*, and tune a 3-phase brushless DC motor using the MCF8329HS motor driver.

Table of Contents

1 Introduction	2
2 Spinning the motor	3
2.1 Connecting to the GUI	4
2.2 SmartTune	5
2.3 Manual Tuning	8
2.4 Testing for Successful Startup Into Closed Loop	16
3 Basic Controls	17
3.1 Speed Input Mode	17
3.2 Preventing Back Spin of Rotor During Startup	18
3.3 Faster Startup Timing	19
3.4 Improving Current and Speed Regulation	21
3.5 Limiting and Regulating Supply Power	24
3.6 MTPA Tuning	25
3.7 Motor Studio Optimization Wizards	25
4 Fault Handling	27
4.1 Abnormal BEMF Fault [ABN_BEMF]	28
4.2 Lock Current Limit [LOCK_LIMIT]	29
4.3 Hardware Lock Current Limit [HW_LOCK_LIMIT]	30
4.4 No Motor Fault [NO_MTR]	31
4.5 Abnormal Speed [ABN_SPEED]	32
5 Hardware Design and Setup	33
5.1 Board Design	33

List of Figures

Figure 2-1. Tuning Sequence of Events	3
Figure 2-2. EVM Hardware Setup	4
Figure 2-3. EVM Connected Indicators	4
Figure 2-4. <i>SmartTune</i> Basic Configuration	5
Figure 2-5. Stages of tuning	6
Figure 2-6. <i>SmartTune</i> Success	6
Figure 2-7. <i>SmartTune</i> Performance Tuning	7
Figure 2-8. <i>SmartTune</i> Optional Parameters	7
Figure 2-9. Manual Tuning	9
Figure 2-10. BASE_CURRENT Bit Field	11
Figure 2-11. Current Protection Limits	12
Figure 2-12. OL_ILIMIT, ALIGN_OR_SLOW_CURRENT_ILIMIT, and IPD_CURR_THR Current Limits	12
Figure 2-13. ILIMIT Current Limit	13
Figure 2-14. Voltage Limits	13
Figure 2-15. Motor Max Speed	14
Figure 2-16. Motor Resistance and Inductance	15
Figure 2-17. Closed Loop Spin Test Steps	16
Figure 3-1. Speed Mode Selection	17
Figure 3-2. SW1 Position for I2C Speed Mode	18
Figure 3-3. Optimal Startup Page	19

Figure 3-4. Phase Current, FG and Motor Speed - Faster Startup Time.....	20
Figure 3-5. Current Loop Tuning.....	21
Figure 3-6. Motor Characteristics.....	22
Figure 3-7. Motor Operation Tuning Parameters.....	23
Figure 3-8. Motor Speed Tuning Parameters.....	23
Figure 3-9. Power Control Settings.....	24
Figure 3-10. MTPA Settings.....	25
Figure 3-11. Saliency Register.....	25
Figure 3-12. Optimization Wizards Page.....	26
Figure 4-1. Faults Tab.....	27
Figure 4-2. ABNORMAL_BEMF_THR.....	28
Figure 4-3. LOCK_ILIMIT.....	29
Figure 4-4. HW_LOCK_LIMIT.....	30
Figure 4-5. NO_MTR.....	31
Figure 4-6. ABN_SPEED.....	32

Trademarks

All trademarks are the property of their respective owners.

1 Introduction

The MCF8329HS is a 4.5V to 60V, three-phase brushless-DC (BLDC) gate driver IC with code-free sensorless field oriented control (FOC) for motor drive applications. The device integrates a single shunt current sense amplifier (CSA) along with an external shunt resistance to sense the motor current. This document helps customers to set up the MCF8329HS, enabling them to experience the devices' powerful performance and flexible programmability.

This tuning guide is applicable for both industrial (MCF8329HSEVM) and automotive (MCF8329HSRRYEVM) versions of the MCF8329HS evaluation modules. References to MCF8329EVM markings throughout this document may refer to either the MCF8329HSEVM or MCF8329HSRRYEVM, depending on which evaluation module the user possess.

Note

Before proceeding through this tuning guide make sure to do the following:

1. Read the [MCF8329HS-Q1 Sensorless Field Oriented Control \(FOC\) Three-phase BLDC Gate Driver Data Sheet](#), [MCF8329HSRRYEVM User's Guide](#), and watch the quick start video.
 2. Get a [MCF8329HSRRYEVM](#) board.
 3. Install the [MotorStudio](#) application.
-

2 Spinning the motor

This section provides standardized steps to tune the MCF8329HS's settings so that the motor can successfully spin-up and enter closed loop control.

The general steps to tune the MCF8329HS's registers so the motor can spin-up and enter closed loop control are outlined in [Figure 2-1](#)

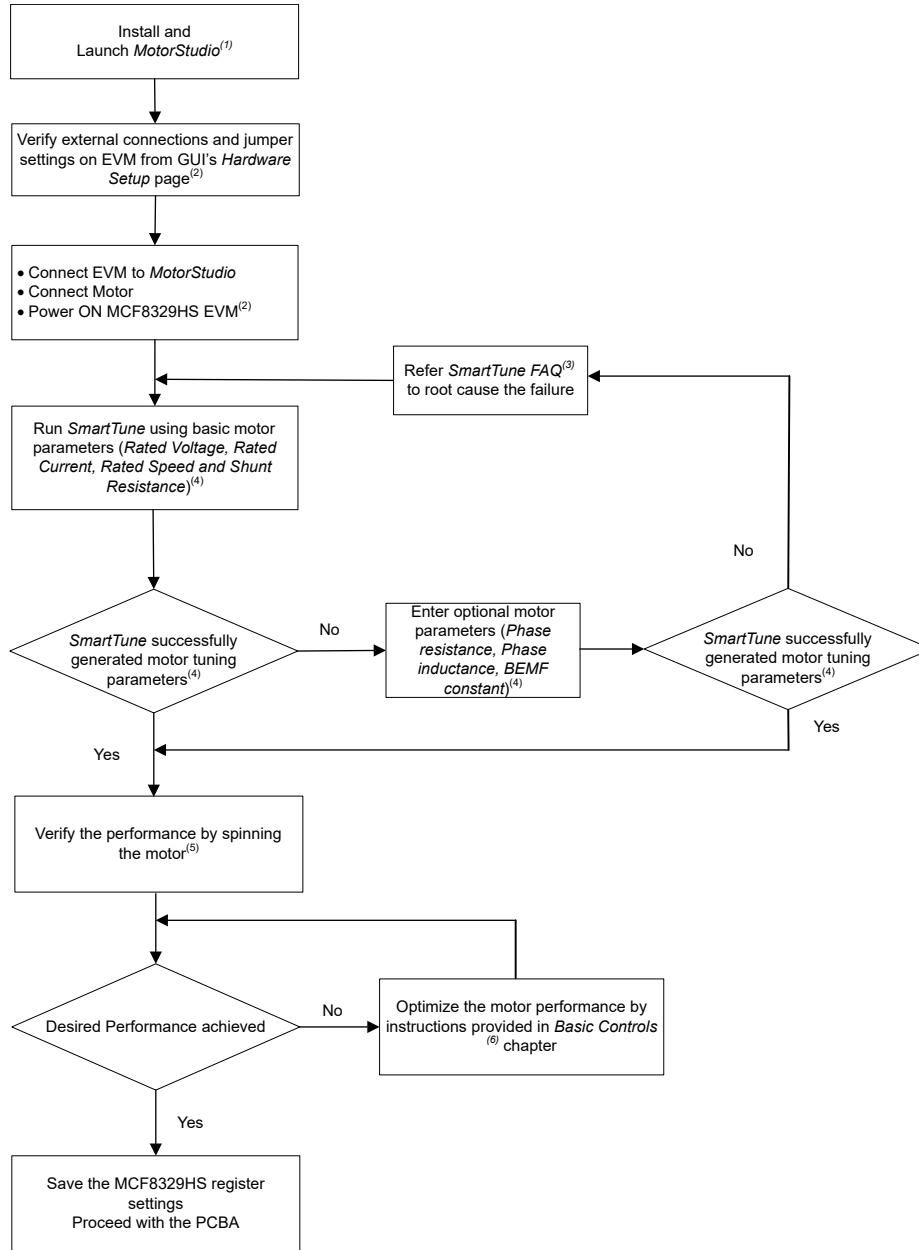


Figure 2-1. Tuning Sequence of Events

Note

1. [MotorStudio](#)
2. [Connecting to the GUI](#)
3. [SmartTune FAQ](#)
4. [SmartTune Execution](#)
5. [Testing for Successful Startup Into Closed Loop](#)
6. [Basic Controls](#)

2.1 Connecting to the GUI

Before connecting the MCF8329HSEVM to the computer, start up the *MotorStudio* application and select MCF8329HS 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.

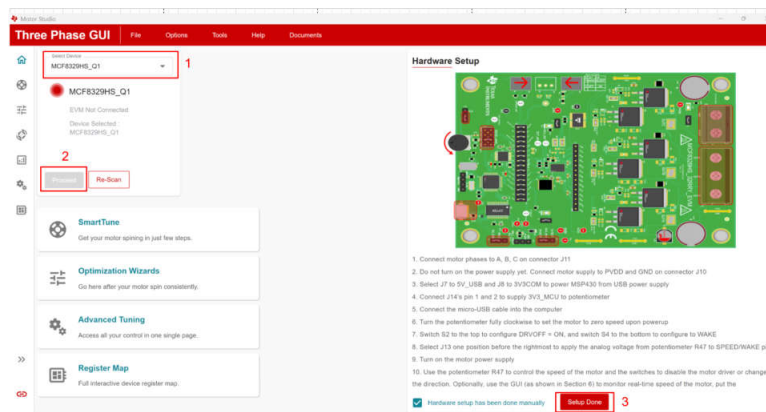


Figure 2-2. EVM Hardware Setup

Once the hardware setup is completed, turn on the power supply connected to the EVM. After the PVDD LED D3 lights up connect a micro-USB to USB cable between the EVM and PC. After a few seconds, *MotorStudio* should connect to the EVM and the two icons outlined in [Figure 2-3](#) will turn green. If the EVM is not connecting click the *Re-Scan* button.

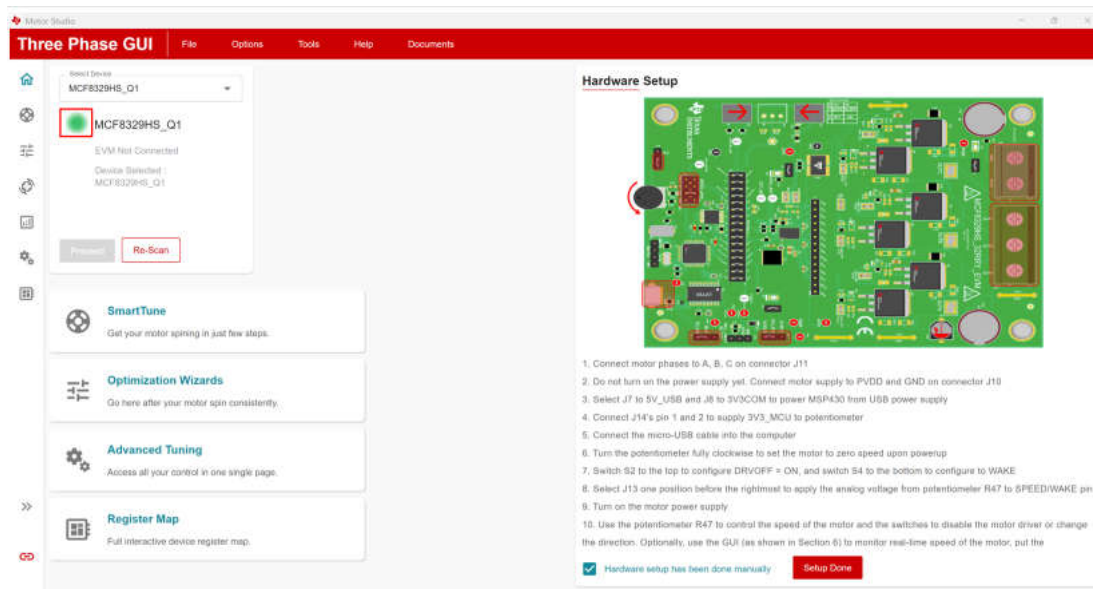


Figure 2-3. EVM Connected Indicators

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.

2.2 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 couple of minutes with only four required inputs - *Rated Voltage*, *Rated Current*, *Rated Speed* and *Shunt Resistance* value.

2.2.1 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.

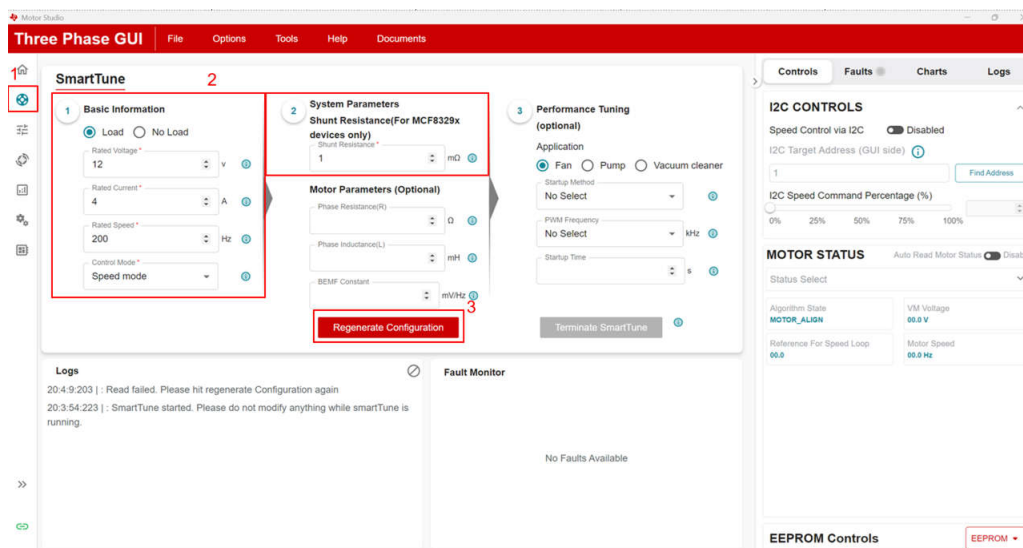


Figure 2-4. SmartTune Basic Configuration

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 2-5](#).

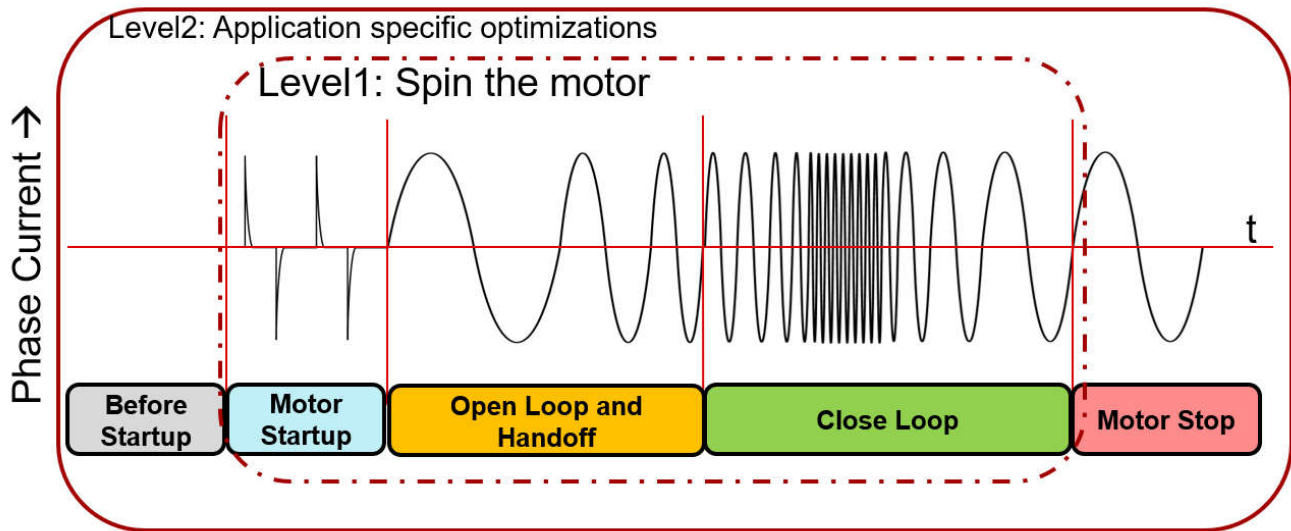


Figure 2-5. Stages of tuning

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 [Section 2.4](#).

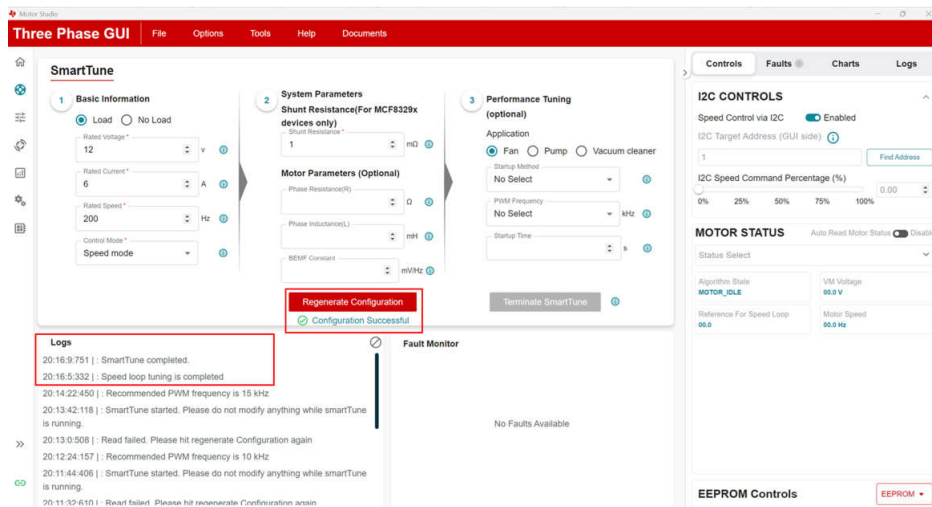


Figure 2-6. *SmartTune* Success

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.

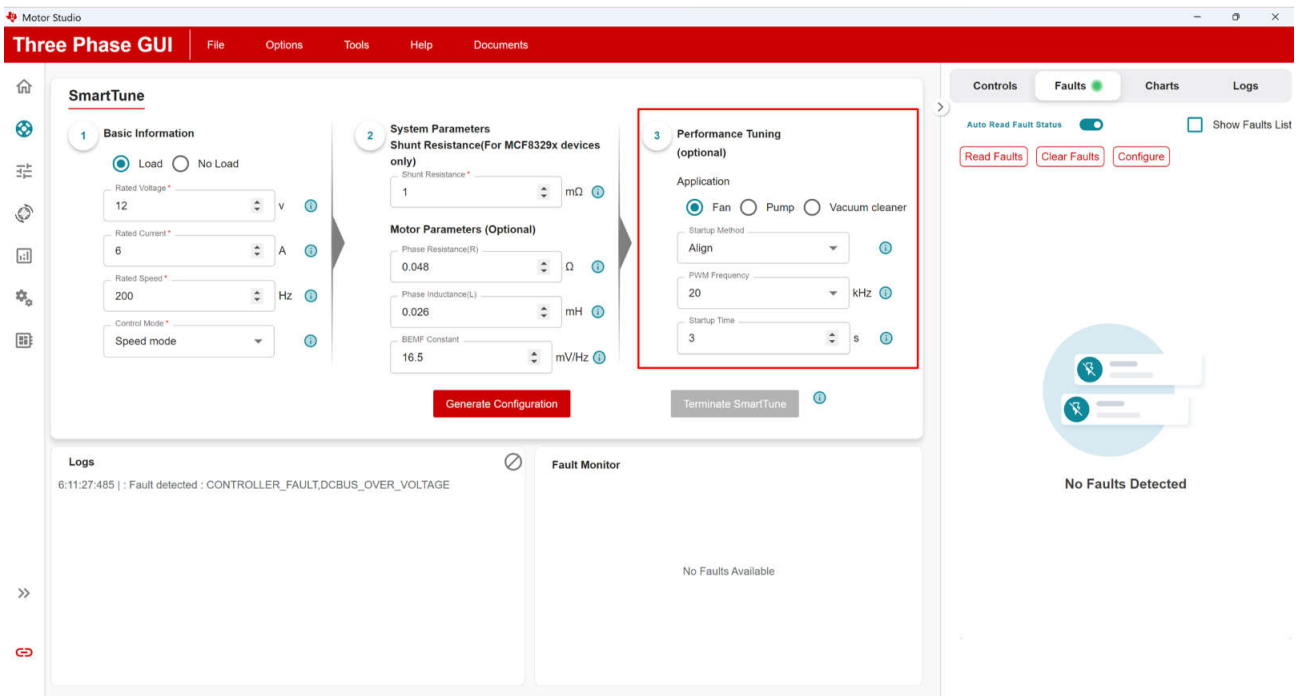


Figure 2-7. SmartTune Performance Tuning

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.

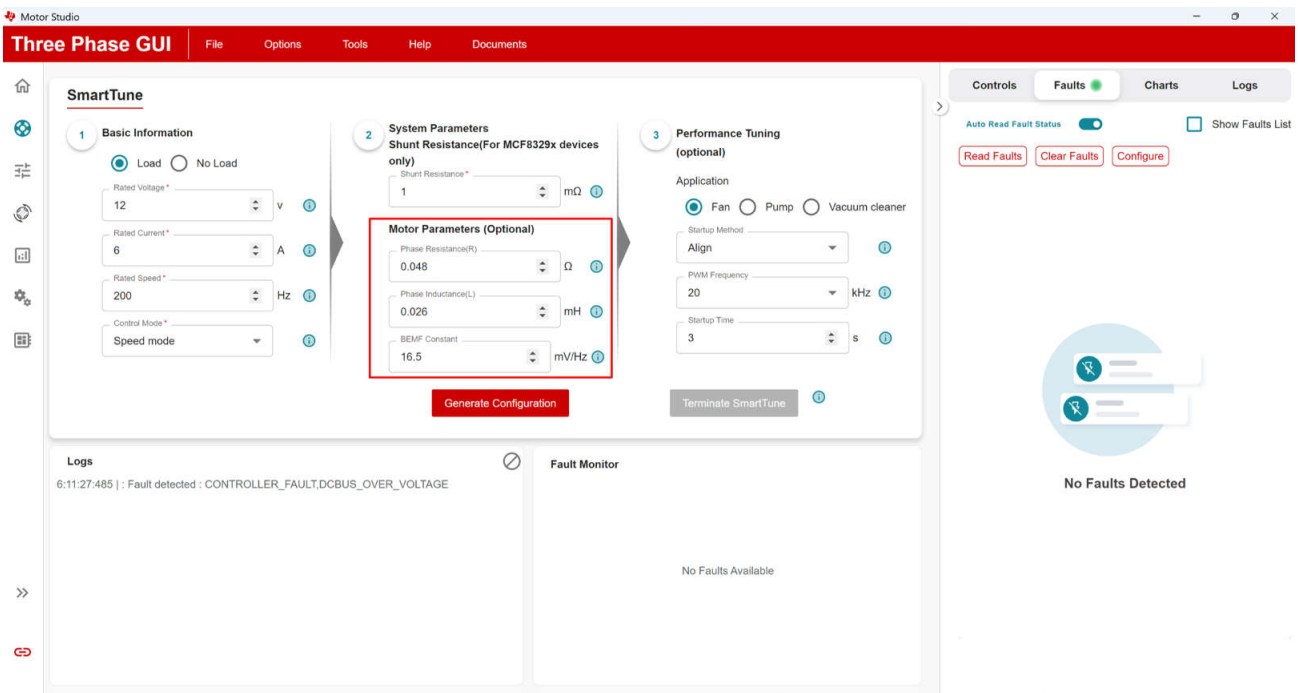


Figure 2-8. SmartTune Optional Parameters

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 advised to proceed with manual motor parameter configuration according to the procedure specified in the [Section 2.3](#).

SmartTune provides basic tuning configuration required to spin the motor reliably. User may want to further tune the configuration parameters to achieve shorter start up time, smooth transition from open to close loop control and steady closed loop control without speed overshoot during motor acceleration. Some of these tuning processes are explained in [Section 3](#).

Note

The MCF8329HS includes many advanced features such as dry-run detection, PWM dithering, speed/power limiting, overmodulation, and external watchdog functionality, which are not enabled by *SmartTune*. Users may enable these features manually based on their specific application requirements. By default, *SmartTune* configures all the device faults for retry mode.

2.3 Manual Tuning

SmartTune can occasionally fail to generate the final tuning file for a given motor under certain exceptional conditions. However, it continues to provide numerous essential configurations required for motor operation, including motor startup parameters, operating current thresholds, and fault current thresholds based on the entered motor current, voltage, speed, and shunt resistor values. Users can leverage these configurations to bypass the current and voltage configuration procedures outlined in subsequent sections and proceed directly to motor electrical parameter measurement and control loop tuning. Users also have the option to override the *SmartTune* generated motor startup parameters by manually configuring the device's voltage and current settings according to the procedures detailed in the following sections.

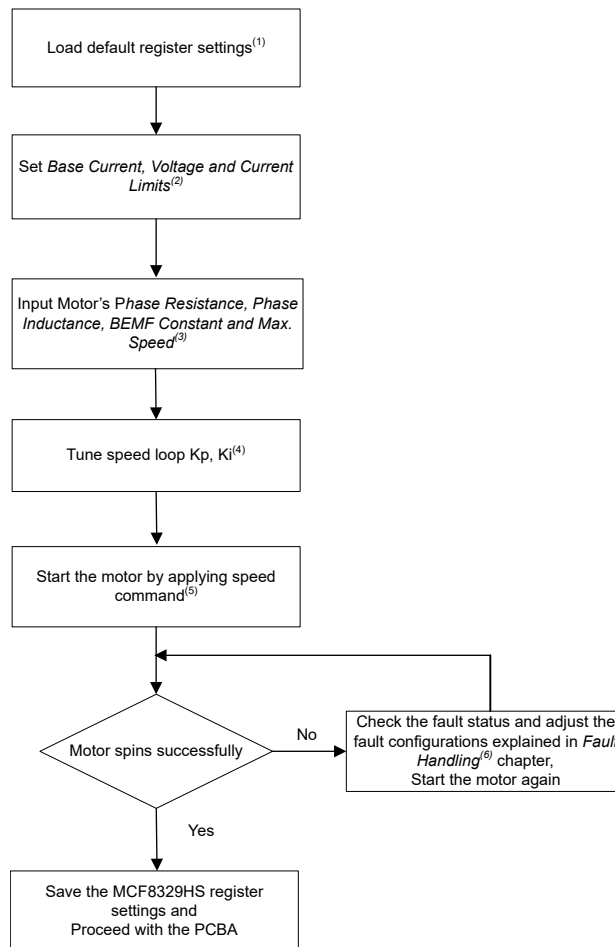


Figure 2-9. Manual Tuning

Note

1. [Loading Recommended Default Values](#)
2. [Setting Base Current, Setting Current Limits and Setting Voltage Limits](#)
3. [Input the Motor's Phase Resistance, Inductance and BEMF Constant and Maximum Electrical Speed](#)
4. [Speed loop Manual Tuning](#)
5. [Testing for Successful Startup Into Closed Loop](#)
6. [Fault Handling](#)

2.3.1 Loading Recommended Default Values

The device incorporates numerous tuning parameters that provide users the capability to configure the system based on specific application requirements. This document covers the essential tuning parameters required for proper motor operation.

Additionally, it is recommended that the device's default configuration be loaded into the device memory before proceeding with manual tuning. The MCF8329HS comes with default settings pre-programmed in the device EEPROM. If these settings have been modified and user wish to restore them to their original state, load the recommended default values listed in [Recommended Default Values](#) table.

Table 2-1. Recommended Default Values

Register Name	Address	Recommended Value
ISD_CONFIG	0x00000080	0x44638C20

Table 2-1. Recommended Default Values (continued)

REV_DRIVE_CONFIG	0x00000082	0x283AF064
MOTOR_STARTUP1	0x00000084	0x0B6807D0
MOTOR_STARTUP2	0x00000086	0x23066000
CLOSED_LOOP1	0x00000088	0x0C3181B0
CLOSED_LOOP2	0x0000008A	0x1AAD0000
CLOSED_LOOP3	0x0000008C	0x00000000
CLOSED_LOOP4	0x0000008E	0x0000012C
SPEED_PROFILES1	0x00000094	0x00000000
SPEED_PROFILES2	0x00000096	0x00000000
SPEED_PROFILES3	0x00000098	0x00000000
SPEED_PROFILES4	0x0000009A	0x00000000
SPEED_PROFILES5	0x0000009C	0x00000000
SPEED_PROFILES6	0x0000009E	0x00000000
FAULT_CONFIG1	0x00000090	0x5FE80206
FAULT_CONFIG2	0x00000092	0x74000000
PIN_CONFIG	0x000000A4	0x00000000
DEVICE_CONFIG1	0x000000A6	0x00000000
DEVICE_CONFIG2	0x000000A8	0x0000B000
PERI_CONFIG1	0x000000AA	0x40000000
GD_CONFIG1	0x000000AC	0x00000100
GD_CONFIG2	0x000000AE	0x00200000
ALGO_CTRL1	0x000000EA	0x00000000
INT_ALGO_1	0x000000A0	0x00B3407D
INT_ALGO_2	0x000000A2	0x000001A7

2.3.2 Setting Base Current

Using the shunt resistor value and CSA gain value determined in [Section 5.1.4](#), use [Equation 1](#) to calculate the max measurable current of the internal CSA.

$$\frac{1.5}{R_{SENSE} \times CSA_GAIN} \times \frac{32768}{1200} \tag{1}$$

After converting the result from [Equation 1](#) into a hexadecimal value, input the result into the BASE_CURRENT bit field in register GD_CONFIG2 using the *Register Map* page.

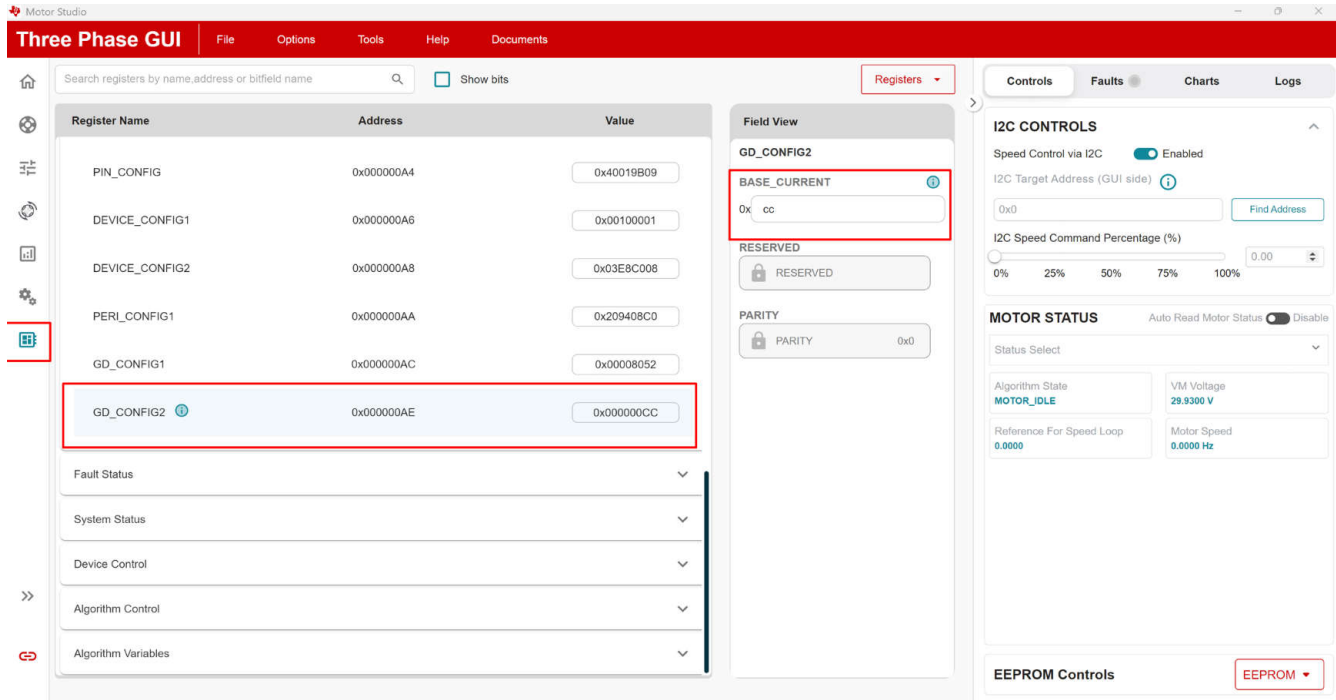


Figure 2-10. BASE_CURRENT Bit Field

2.3.3 Setting Current Limits

All the current limits within the MCF8329HS are set as a percent of the value programmed into the BASE_CURRENT bit field. For example, if BASE_CURRENT is set to 37.5A and ILIMIT is set to 50%, then the current limit set by ILIMIT will be 18.75A.

HW_LOCK_ILIMIT and LOCK_ILIMIT are configurable current limits intended to protect the system from damage. It is recommended to set these limits to two times higher than the motors rated peak phase current. If the motors rated peak phase current falls between two adjacent limit settings in the configuration, choose the higher of the two settings.

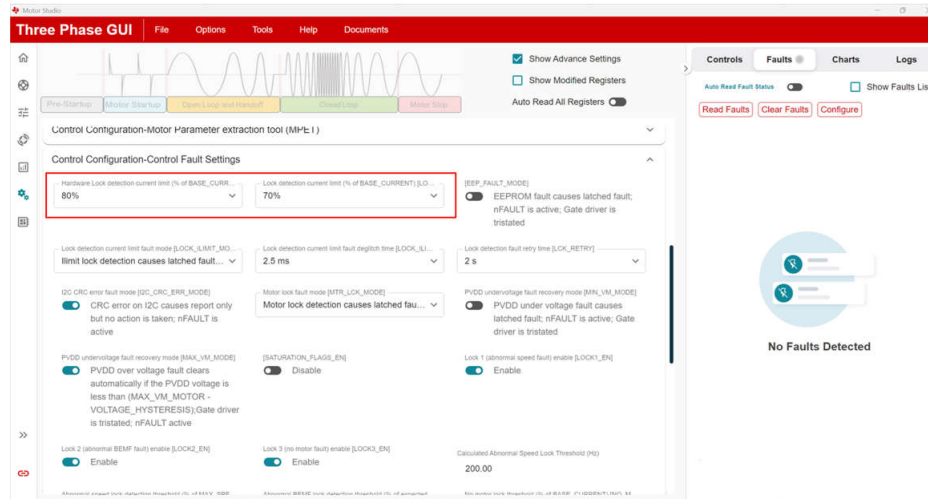


Figure 2-11. Current Protection Limits

ILIMIT, OL_ILIMIT, ALIGN_OR_SLOW_CURRENT_ILIMIT, and IPD_CURR_THR are the max current that are used by the motor driver during the various stages of motor operation. It is recommended to set these values to less than or equal to the rated max phase current of the motor.

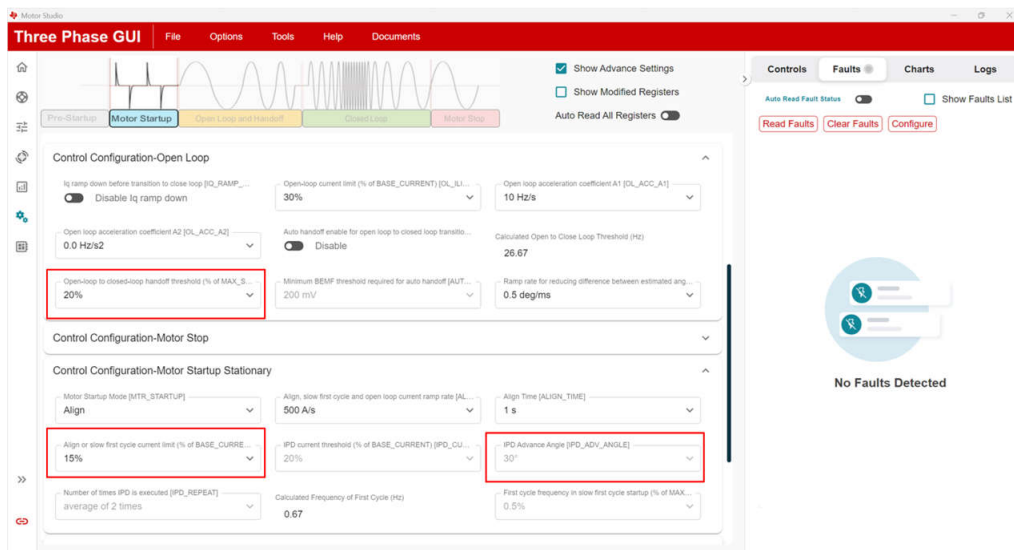


Figure 2-12. OL_ILIMIT, ALIGN_OR_SLOW_CURRENT_ILIMIT, and IPD_CURR_THR Current Limits

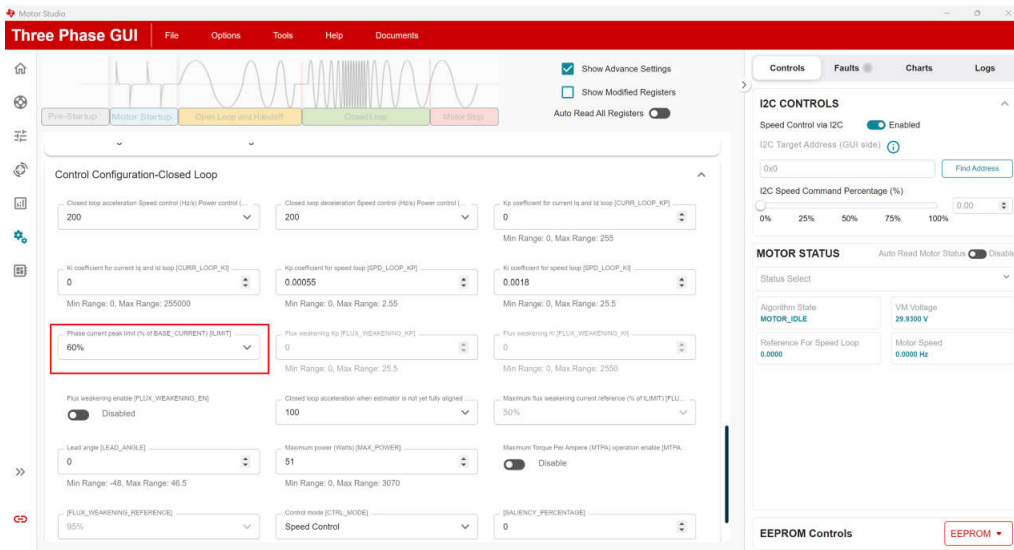


Figure 2-13. ILIMIT Current Limit

2.3.4 Setting Voltage Limits

BUS_VOLT is used to normalize the modulation algorithm that will be used by the MCF8329HS. To improve the resolution of the modulation algorithm at lower motor voltages set the BUS_VOLT to the closest value that is still greater than the expected DC bus voltage or phase voltage. Make sure to account for voltage spikes on the phase node when determining the max expected voltage value.

In applications where the motor voltage cannot exceed a certain range, MAX_VM_MOTOR and MIN_VM_MOTOR can be used to set the desired voltage limits.

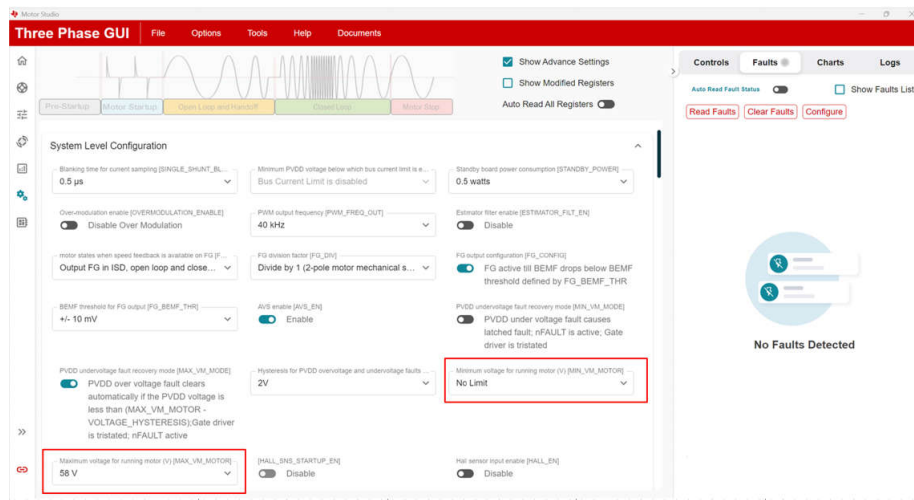


Figure 2-14. Voltage Limits

2.3.5 Maximum Electrical Speed (Hz)

Go to the *Control Configuration-Motor Parameters* Section of the GUI on the *Advanced tuning* tab and enter the motor's max speed in Hz.

If the speed specified in the motor's datasheet is given in revolutions per minute (RPM), then utilize the following formula to convert the RPM to frequency in Hertz (Hz):

$$\text{Motor speed in Hz} = (\text{Number of poles} \times \text{RPM}) \div 120$$

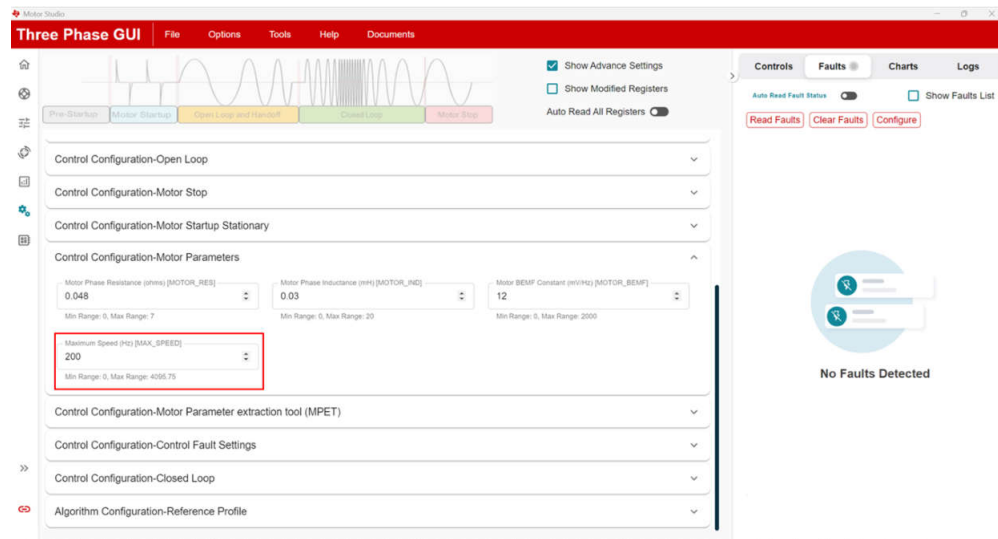


Figure 2-15. Motor Max Speed

Note

Determining number of motor poles without a motor data sheet:

1. Use a lab power supply and make sure it's current limit is set to less than the motor rated current. Do not turn on the supply.
2. Connect V+ of the supply to phase A and V- of the supply to phase B of the motor. Any 2 of the 3 phases can be chosen at random if they are not labeled.
3. Turn on supply. The rotor should have settled at one position with the injecting current.
4. Manually rotate the rotor until rotor snaps to another settle position. It will have several settle-down positions around one mechanical cycle.
5. Count the number of settle-down positions for one fully mechanical cycle, which is the number of pole pairs. Multiplying by two calculates the number of poles.

Be careful of gearing systems within a motor. The gear ratio determines how many rotor revolutions correlate to the shaft's mechanical revolution.

2.3.6 Input the Motor's Phase Resistance, Inductance and BEMF Constant

Using the instructions in the [motor parameters FAQ](#), find the motors phase resistance, inductance and BEMF constant. Once these values are found, input phase resistance into the *Motor Phase Resistance* box, phase inductance into the *Motor Phase Inductance* box and BEMF constant into the *BEMF constant* box available under *Motor Parameters* tab on the *Advanced Tuning* page.

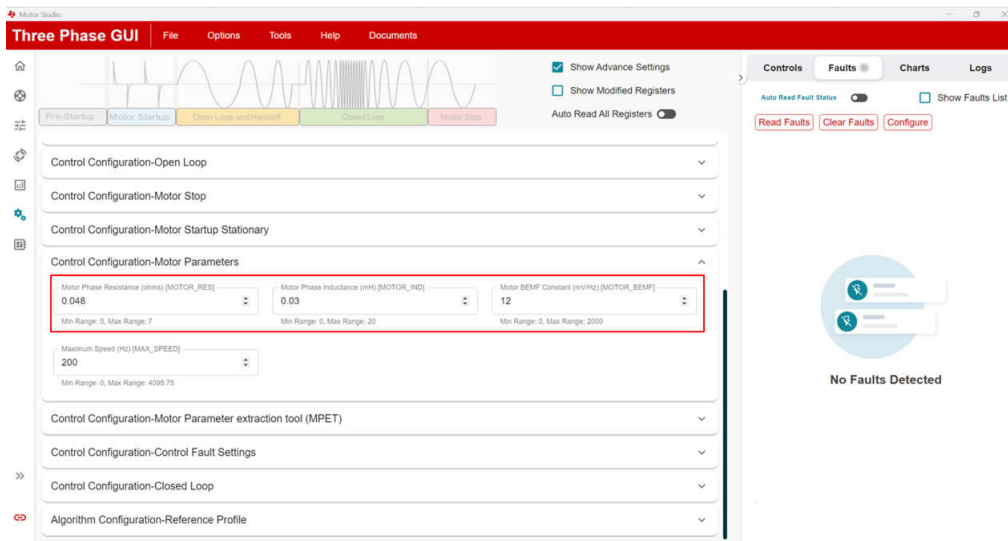


Figure 2-16. Motor Resistance and Inductance

2.3.7 Speed loop Manual Tuning

For applications that require better speed regulation, it is recommended to tune Speed loop PI controllers [SPD_LOOP_KP] and [SPD_LOOP_KI]. Kp coefficient of speed loop [SPD_LOOP_KP] controls the settling time and speed overshoots. Ki coefficient of Speed loop [SPD_LOOP_KI] controls speed overshoot and ensures regulation of speed at set value and drives the error to zero. Speed loop PI controller gains can either be auto-tuned by MCF8329HS or tuned manually.

Manual Tuning: Use the following steps to tune Speed loop PI controller gains manually:

1. Set the control mode [CTRL_MODE] to modulation index control (11b).
2. Issue non-zero speed command to start the motor (refer to [Section 2.4](#), step 1 on how to issue non-zero speed command). Motor will spin in open loop.
3. Allow the open loop current to settle down and then measure the peak open loop current.
4. Stop the motor and set the control mode [CTRL_MODE] to current control.
5. Slowly increase the speed command until the motor speed reaches the max speed. Note down the Iq_ref value being reported in the IQ_REF_CLOSED_LOOP register.
6. Speed loop Kp [SPD_LOOP_KP] is calculated using:

$$\text{Speed loop } K_p = \frac{\text{Iq reference at maximum speed}}{\text{Maximum Electrical Speed in Hz}} \quad (2)$$

7. Speed loop Ki [SPD_LOOP_KI] is calculated using:

$$\text{Speed loop } K_i = 0.1 \times \text{Speed loop } K_p \quad (3)$$

8. Stop the motor and set the control mode [CTRL_MODE] to speed control.

Note

Tuning speed loop Kp and Ki is experimental. If the above recommendation doesn't work, then we recommend to manually tune Speed loop Kp and Ki till the desired results are achieved.

2.4 Testing for Successful Startup Into Closed Loop

1. 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.
2. 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.
3. Check for any faults if the *Faults* tab shows a red circle. If a fault has been reported, go to [Section 4](#) and follow the debug steps to correct the fault.
4. 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.
5. To have configuration for the registers covered in [section 7 of the MCF8329HS data sheet](#) 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 Motor Studio and select the *Write To EEPROM* option. Click the *Yes* button in the window that pops up.

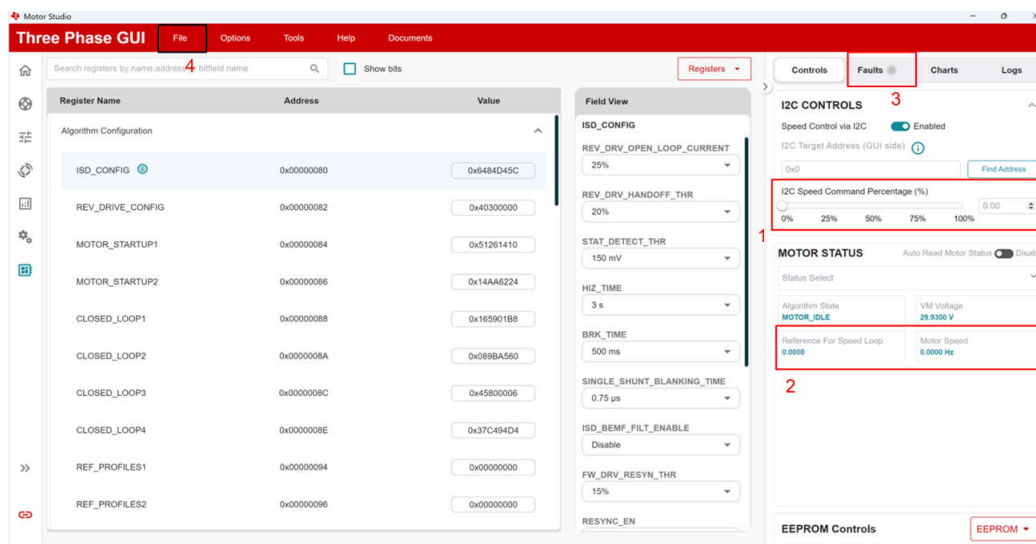


Figure 2-17. Closed Loop Spin Test Steps

3 Basic Controls

This section provides tuning guidance for optimizing the motors performance for many use-case needs.

Note

It s expected to skip the subsection use-cases and scenarios that do not apply to the system or end equipment.

3.1 Speed Input Mode

The MCF8329HS offers four options to control the speed of the motor: PWM, frequency side, Analog, and I2C. The desired speed mode can be set by changing the value of the SPEED_MODE register on the *Advanced Tuning* page. A description of how to configure these control methods is provided in the *Motor Control Input Options* section of the [MCF8329HS Sensorless Field Oriented Control \(FOC\) Three-phase BLDC Gate Driver Data Sheet](#).

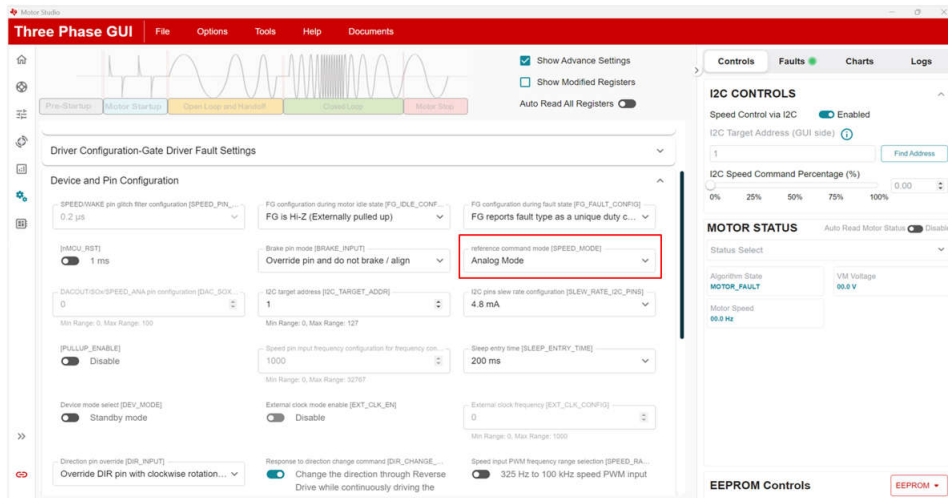


Figure 3-1. Speed Mode Selection

If I2C speed input is chosen, flip SW1 away from the other switches, see [Figure 3-2](#), this will provide a the wake switch signal to the SPEED/WAKE pin to keep the MCF8329HS out of sleep/standby mode. If a speed mode other than I2C is being used, flip the switch to the opposite as what is shown in [Figure 3-2](#) to connect the speed pin to J13. For information on how to set J13, see the *Description of User-Selectable Settings on MCF8329HSEVM (Default in Bold)* table in the [MCF8329HSEVM User's Guide](#).

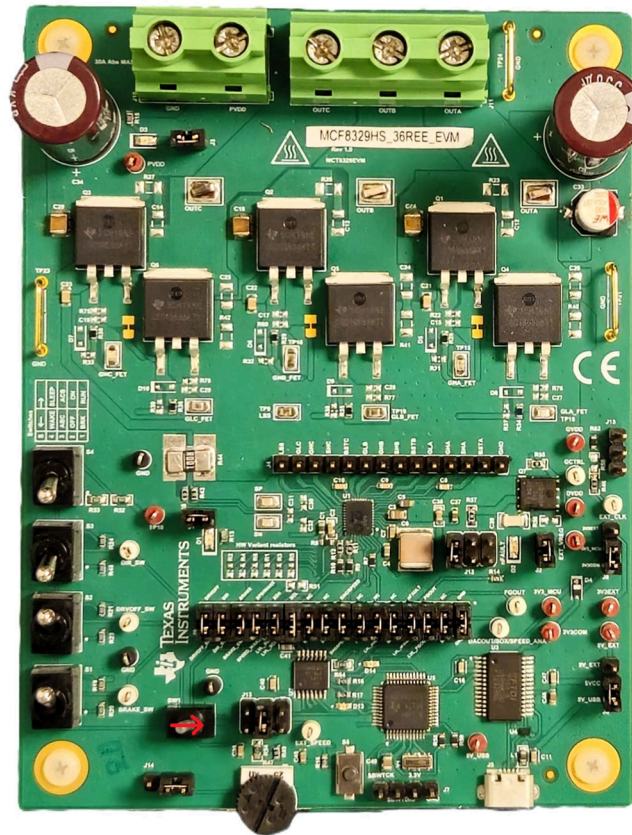


Figure 3-2. SW1 Position for I2C Speed Mode

3.2 Preventing Back Spin of Rotor During Startup

Option 1: Initial Position Detection (IPD):

1. Go to *Optimal Startup* in the *Optimization Wizards* page, select IPD, and click the *Next* button for instructions to set up the IPD start up method.
2. Set the IPD Advance Angle [IPD_ADV_ANGLE] to 90° to get maximum startup torque. If there is sudden jerk observed during startup, then it is recommended to reduce the angle to 60° or 30° for a smoother startup.

Note

Device triggers IPD timeout fault [IPD_T1_FAULT] for motors with very high inductance, or if the motor is not connected. If this fault gets triggered, it is recommended to check if motor is connected to the device.

Device triggers IPD Frequency fault [IPD_FREQ_FAULT] if the IPD clock frequency is set too high. If this fault gets triggered, it is recommended to decrease the IPD Clock value [IPD_CLK_FREQ].

Option 2: Slow First Cycle:

1. Go to *Optimal Startup* in the *Optimization Wizards* page, select slow first cycle, and click the *Next* button for instructions to set up the slow first cycle startup method.

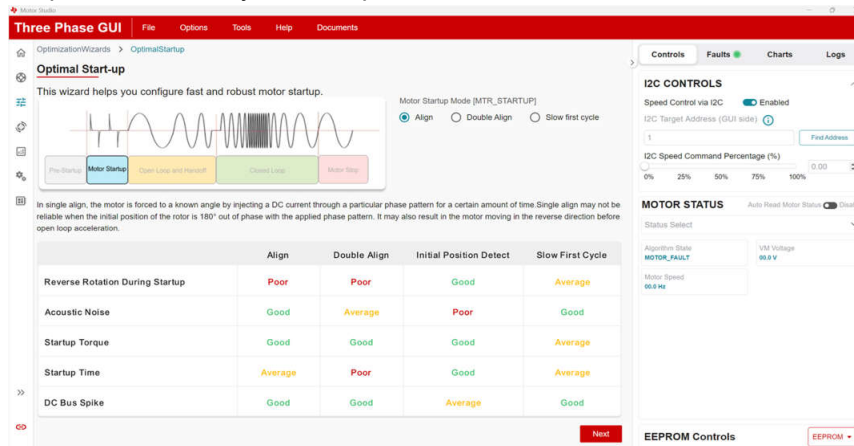


Figure 3-3. Optimal Startup Page

3.3 Faster Startup Timing

Option 1: Initial Position Detection (IPD):

1. Go to *Optimal Startup* in the *Optimization Wizards* page, select IPD, and click the *Next* button.
2. Increase IPD current threshold (A) [IPD_CURR_THR] to the rated current of the motor.
3. Increase IPD clock value [IPD_CLK_FREQ] to a higher frequency up to a value where the device does not trigger IPD frequency fault.
4. Set IPD repeating times [IPD_REPEAT] to 1 time.
5. Set Open loop current limit configuration [OL_ILIMIT_CONFIG] to Open loop current limit defined by ILIMIT.
6. Increase Open loop acceleration coefficient A1 [OL_ACC_A1] and Open loop acceleration coefficient A2 [OL_ACC_A2].

Note

A1 and A2 can be increased until open loop current reaches Lock detection current threshold [LOCK_ILIMIT]. Open loop current can be measured using oscilloscope.

Increasing Open loop acceleration coefficient A1 [OL_ACC_A1] and Open loop acceleration coefficient A2 [OL_ACC_A2] might trigger LOCK_LIMIT fault. If this happens, reduce A1 and A2 until LOCK_LIMIT fault no longer triggers.

7. For ultra-fast startup time (less than 100 ms) it is recommended to follow below steps.
 - a. Disable auto-handoff [AUTO_HANDOFF].
 - b. Configure open to closed loop handoff threshold [OPN_CL_HANDOFF_THR] to a value lesser than or equal to 20 Hz.
8. For startup times above 100ms, it is recommended to follow below steps:
 - a. Enable auto-handoff [AUTO_HANDOFF].

Note

If Abnormal speed fault [ABN_SPEED] gets triggered, it is recommended to decrease open loop acceleration constants [OL_ACC_A1] and [OL_ACC_A2] and also retune IPD by increasing the IPD current threshold [IPD_CURR_THR] and IPD repeat times [IPD_REPEAT].

9. Increase Closed loop acceleration rate [CL_ACC].

Note

LOCK_LIMIT fault handling:

Closed loop acceleration rate [CL_ACC] can be increased until closed loop current reaches Lock detection current threshold [LOCK_ILIMIT]. Closed loop current can be measured using oscilloscope. Increasing closed loop acceleration rate [CL_ACC] might trigger LOCK_LIMIT. If this happens, reduce closed loop acceleration rate [CL_ACC] until no longer triggers.

Option 2: Align

1. Go to *Optimal Startup* in the *Optimization Wizards* page, select Align, and click the *Next* button for instructions to set up the Align startup method.
2. Configure align time [ALIGN_TIME] to 10 ms.
3. Follow Step 6 to Step 9 in Option 1.

Figure 3-4 shows FG, phase current and motor electrical speed waveform. Motor takes 50 ms to reach target speed from zero speed.

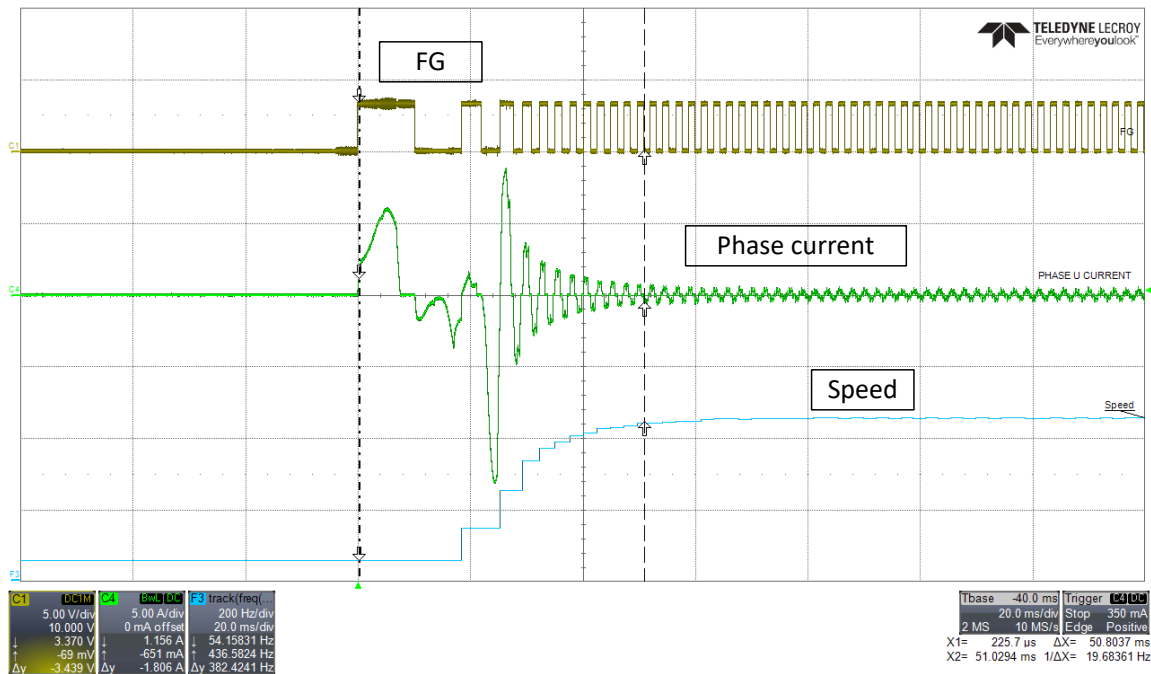


Figure 3-4. Phase Current, FG and Motor Speed - Faster Startup Time

Note

If Abnormal speed fault [ABN_SPEED] or Loss of sync [LOSS_OF_SYNC] fault gets triggered, it is recommended to follow below debug steps:

1. Select Double align as the motor startup method in [MTR_STARTUP].
2. Increase align time [ALIGN_TIME].
3. Configure align current threshold [ALIGN_OR_SLOW_CURRENT_ILIMIT] to 50% of ILIMIT.
4. Configure First cycle frequency select [FIRST_CYCLE_FREQ_SEL] to 0.

Note

Refer “[MCF83xx - Open Loop to Closed Loop Handoff Tuning](#)” application note for detailed procedure in tuning the open loop acceleration rate and open to close loop handoff parameters

3.4 Improving Current and Speed Regulation

The MCF8329HS 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.

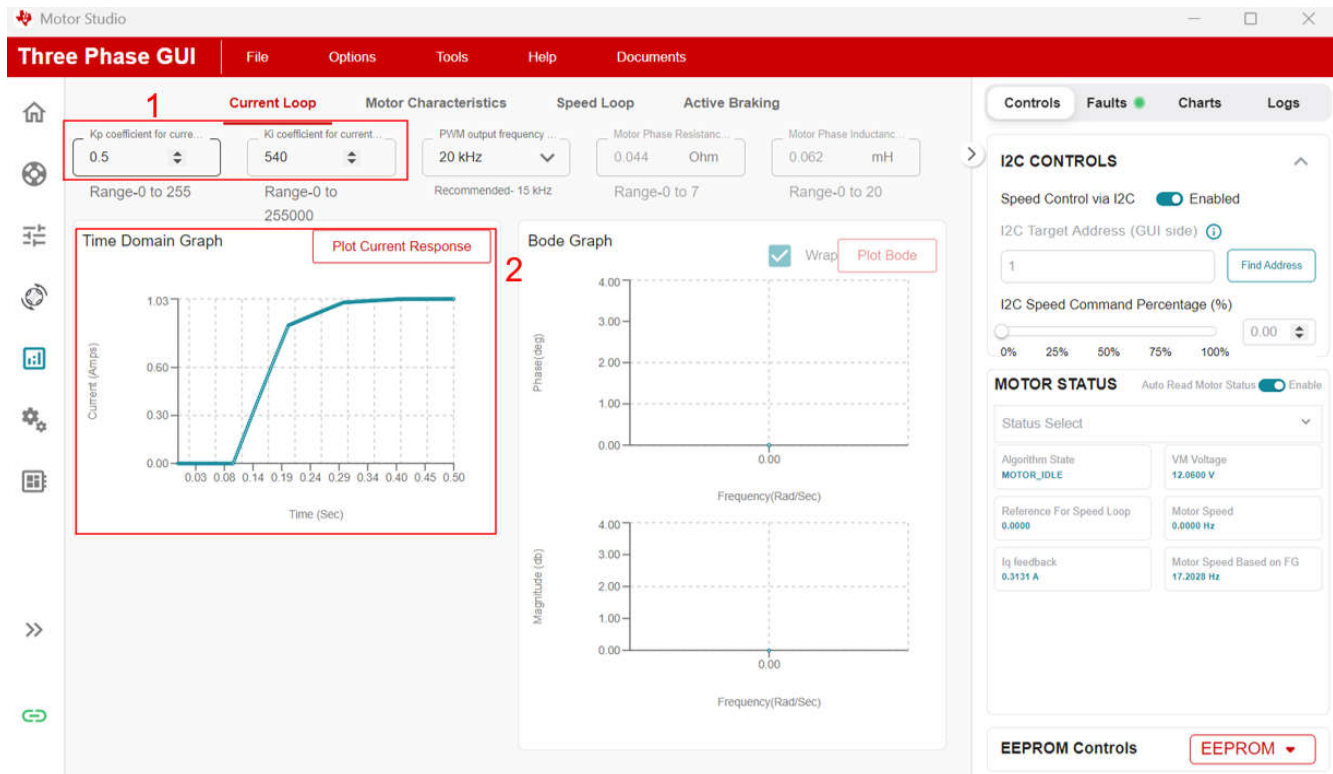


Figure 3-5. Current Loop Tuning

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.

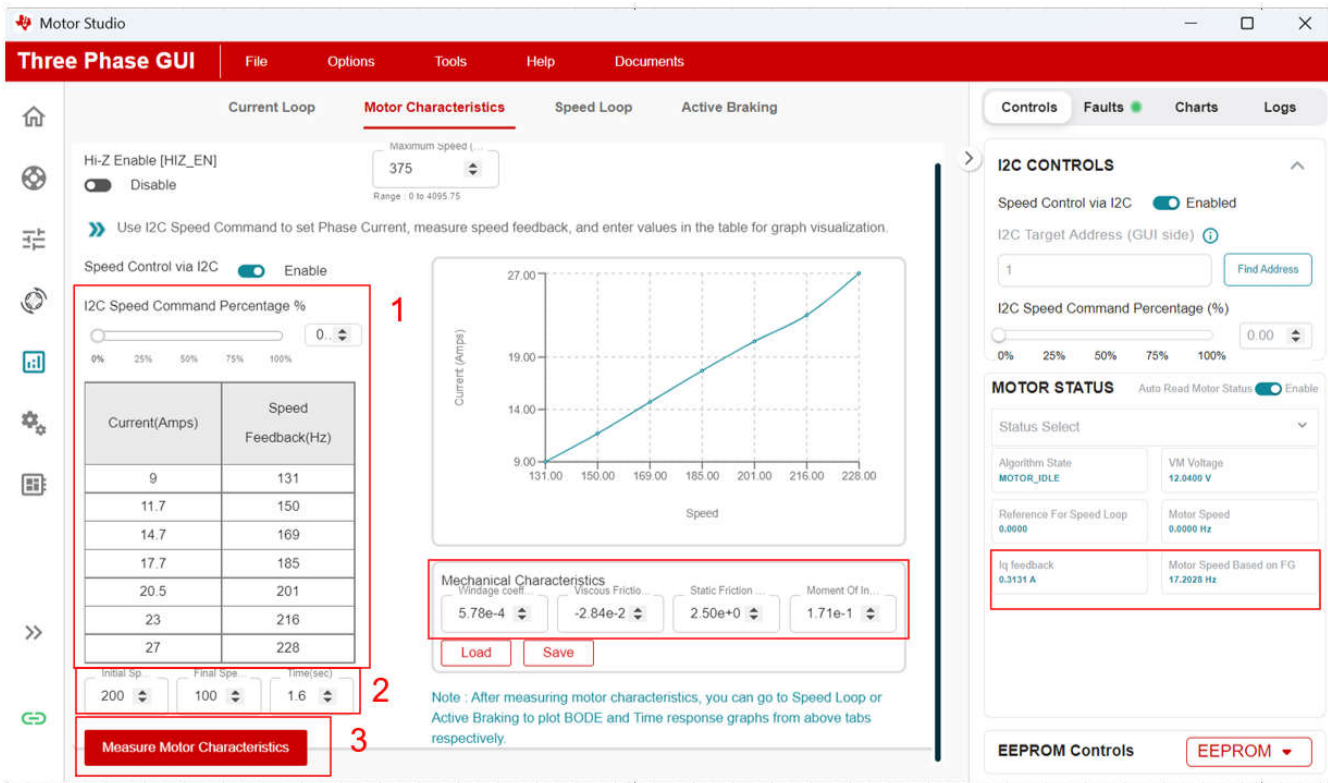


Figure 3-6. Motor Characteristics

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.

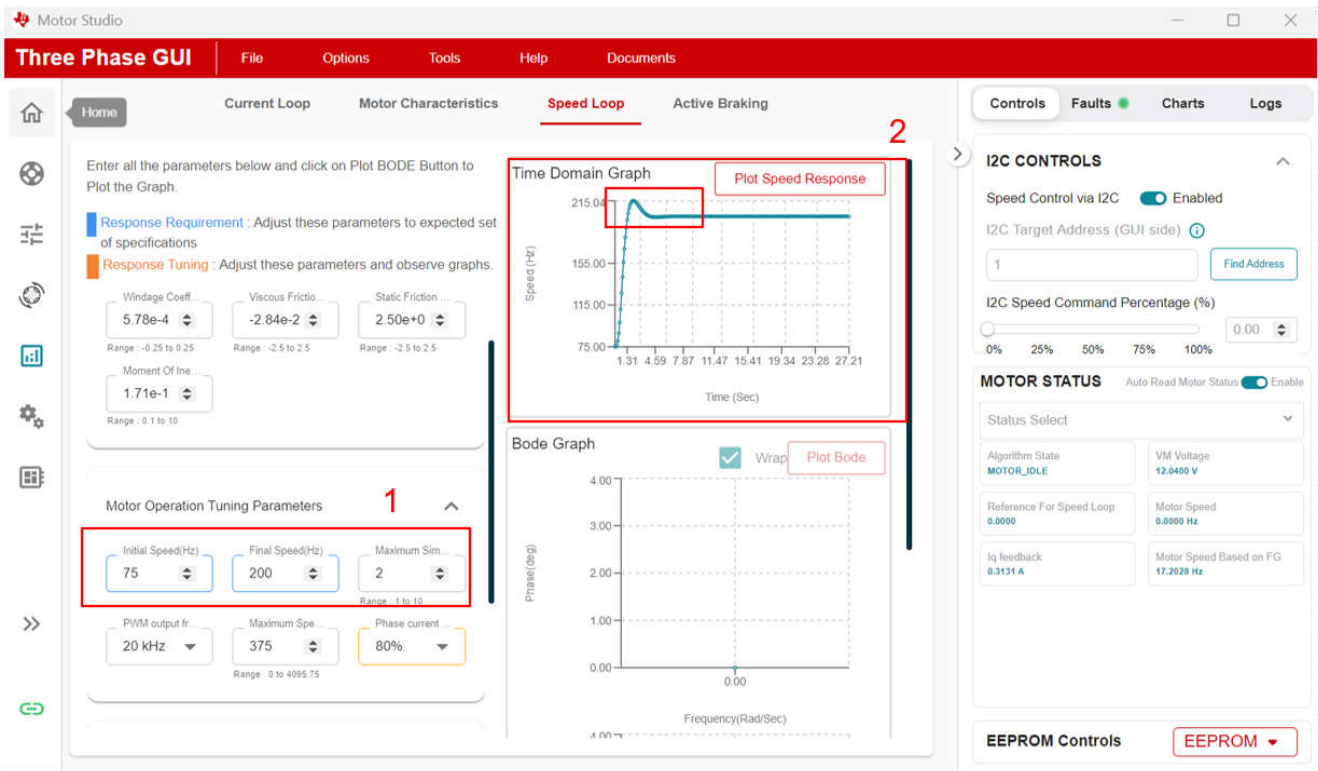


Figure 3-7. Motor Operation Tuning Parameters

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

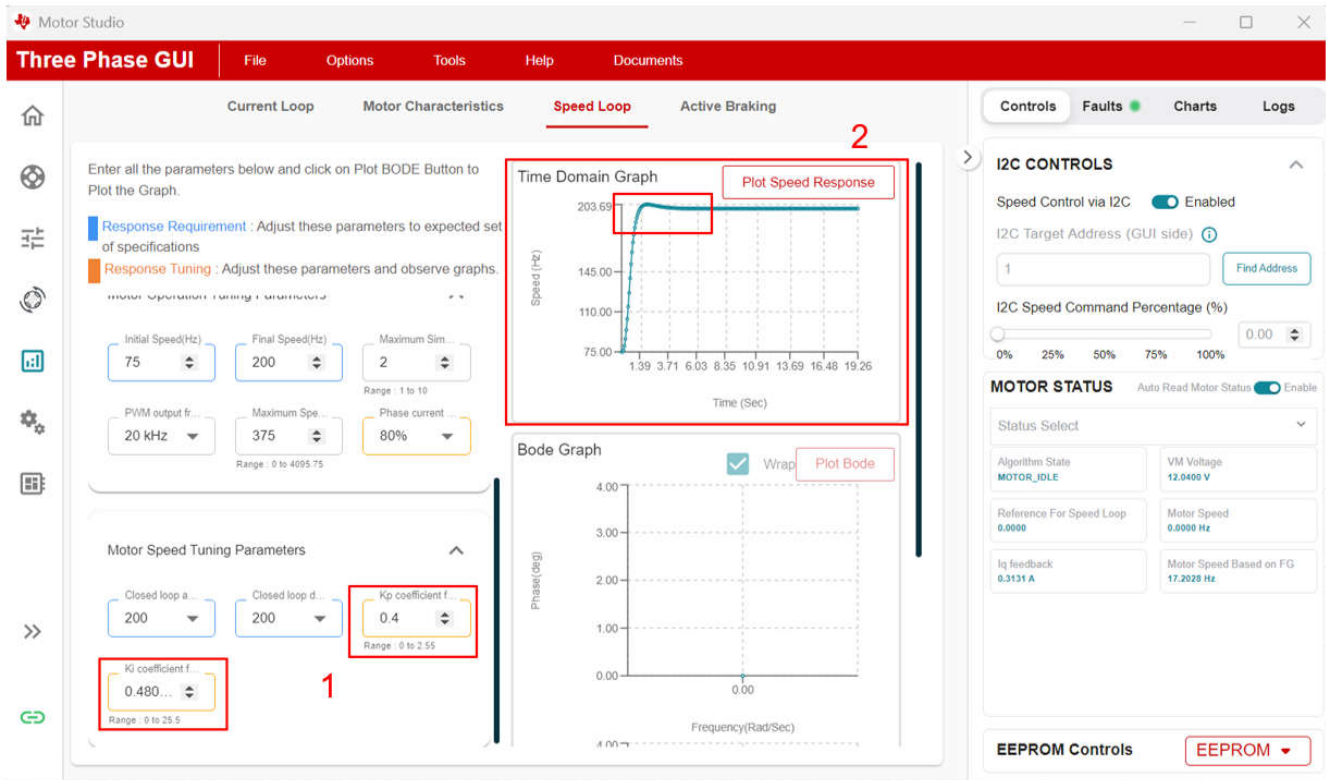


Figure 3-8. Motor Speed Tuning Parameters

3.5 Limiting and Regulating Supply Power

MCF8329HS provides options to limit and regulate supply power. This feature can be utilized in battery powered motor driver applications such as cordless vacuum cleaners, power tools etc.

Use the following steps to limit supply power. In this mode, supply power is only limited to reference power and not actively regulated.

1. Configure CTRL_MODE to power control (1b).
2. Configure MAX_POWER. This sets the maximum power that MCF8329HS can draw from the DC input supply at 100% duty command. For example, if MAX_POWER is configured to 25 W, MCF8329HS draws 12.5 W from power supply at 50% duty command.
3. The power control loop uses the same PI controller parameters as in the speed loop mode. Kp and Ki coefficients are configured through SPD_POWER_KP and SPD_POWER_KI. Tuning SPD_POWER_KP and SPD_POWER_KI is experimental. The recommendation is to manually tune both parameters until the desired results are achieved.

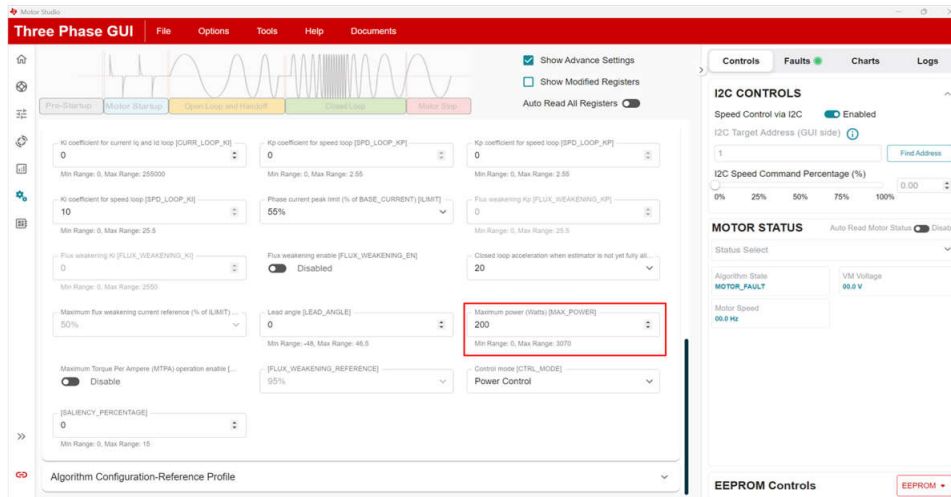


Figure 3-9. Power Control Settings

3.6 MTPA Tuning

Maximum torque per ampere (MTPA) is a feature in the MCF8329HS to maximize the torque generated per ampere of current for salient motors. To enable MTPA, set MTPA_EN to 1b and set the SALIENCY_PERCENTAGE to a non-zero value calculated by the Lq and Ld values provided in the motor's data sheet.

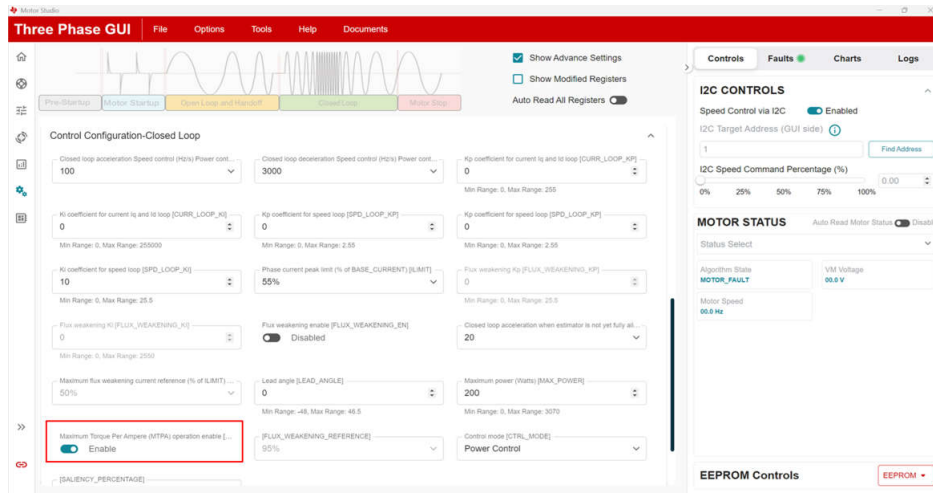


Figure 3-10. MTPA Settings

Note

If the motor's Ld or saliency percentage is not known, the approximate SALIENCY_PERCENTAGE can be determined by following the steps below:

1. Set the SALIENCY_PERCENTAGE to 0x1h
2. Set the CTRL_MODE to Current Control mode
3. Provide a speed command.
4. While the motor is spinning, increment the SALIENCY_PERCENTAGE value by 1h until the motor's speed begins to decrease.

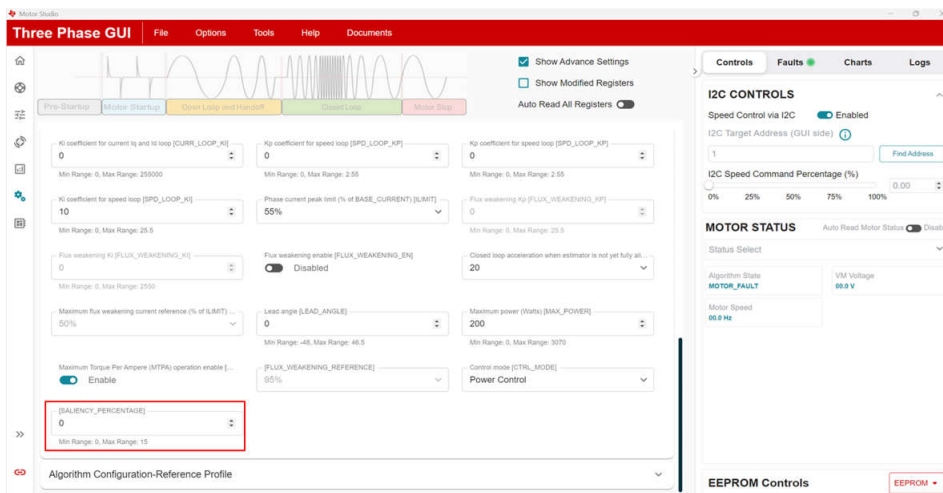


Figure 3-11. Saliency Register

3.7 Motor Studio Optimization Wizards

For step-by-step guidance on configuring the MCF8329HS for additional use-cases and optimization features, see the *Optimization Wizards* page on Motor Studio.

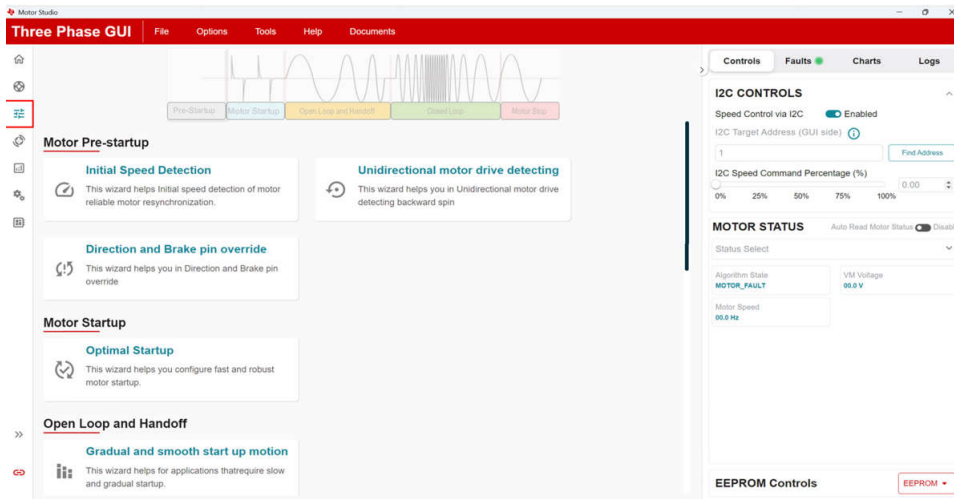


Figure 3-12. Optimization Wizards Page

4 Fault Handling

To see which fault has been reported by the MCF8329HS, go to the *Faults* tab and check if any faults with red circles appear. If a fault is shown in this tab, see the following section that has a title similar to the reported fault.

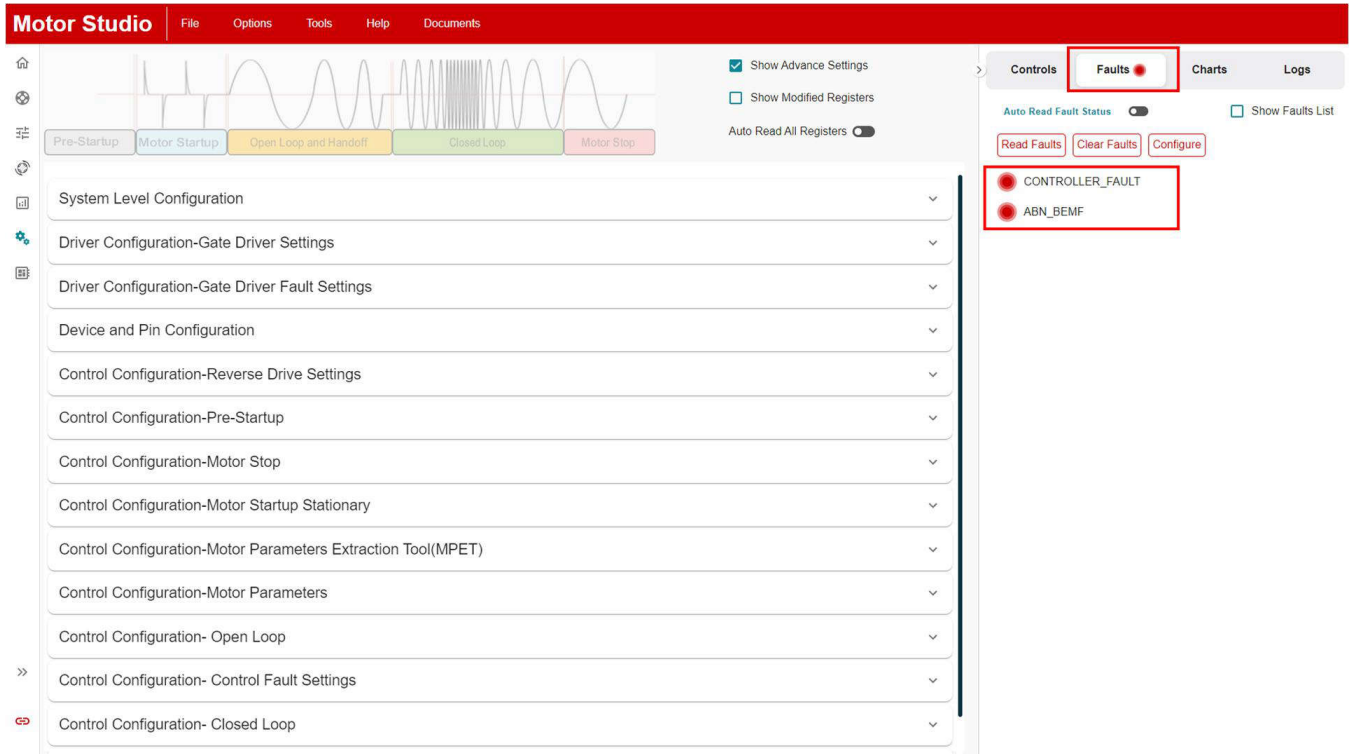


Figure 4-1. Faults Tab

4.1 Abnormal BEMF Fault [ABN_BEMF]

This fault gets triggered when the difference between the estimated BEMF voltage exceeds the threshold set by ABNORMAL_BEMF_THR. If this fault is triggered, then go the *Control Fault Settings* tab in the *Advanced Tuning* page in Motor Studio and follow the below suggestions:

1. For applications with load dynamics (sudden change in load), it is recommended to set the Abnormal BEMF threshold to 70% to avoid triggering this fault.
2. This fault can get triggered if the programmed BEMF constant is inaccurate. Follow steps recommended in [motor parameters FAQ](#) to obtain accurate BEMF constant.

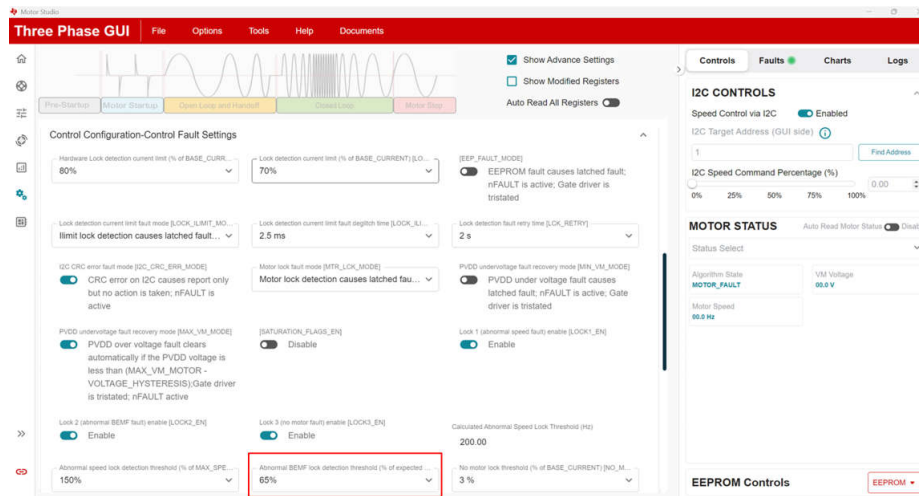


Figure 4-2. ABNORMAL_BEMF_THR

4.2 Lock Current Limit [LOCK_LIMIT]

This fault gets triggered when the phase current exceeds the LOCK_ILIMIT threshold. If this fault is triggered, check the motor data sheet for stall torque and load the motor below the stall torque specified in the data sheet. If the load torque is still within the stall torque, go to the *Control Fault Settings* tab in the *Advanced Tuning* page and increase the value of LOCK_ILIMIT.

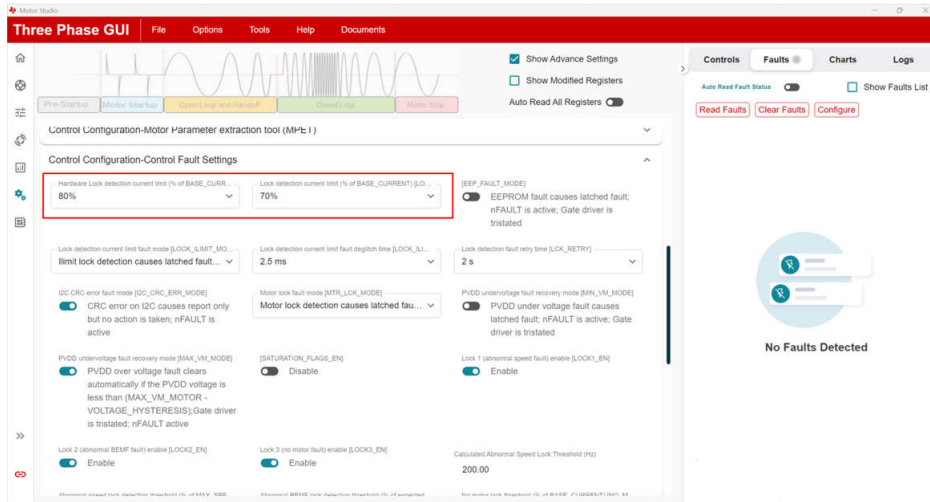


Figure 4-3. LOCK_ILIMIT

4.3 Hardware Lock Current Limit [HW_LOCK_LIMIT]

This fault gets triggered when the phase current exceeds the HW_LOCK_ILIMIT threshold. If this fault is triggered, use the following recommendations:

1. Using the fields circled in [Figure 4-4](#), set SPD_LOOP_KP, SPD_LOOP_KI, CURR_LOOP_KP, and CURR_LOOP_KI to zero. This enables the MCF8329HS to automatically calculate the speed loop and current loop PI controller gains.
2. If the fault still persists, check the continuity across phase-to-phase, phase-to-GND, and PVDD-to-GND to make sure there is no short across these terminals.

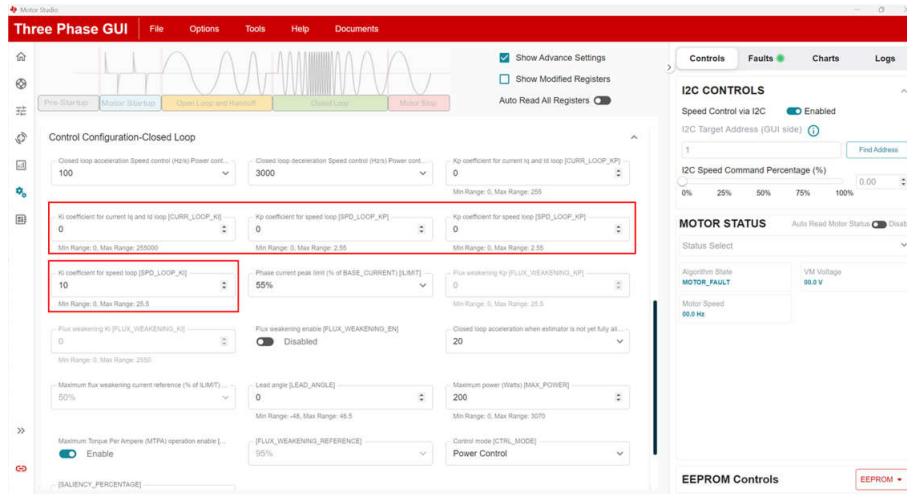


Figure 4-4. HW_LOCK_LIMIT

4.4 No Motor Fault [NO_MTR]

This fault gets triggered when the phase current is below the no motor lock threshold for 500ms during open loop. When this fault gets triggered, use the following recommendations:

1. Make sure the motor phases are securely connected to the OUTA, OUTB, and OUTC test points or the connector block J11.
2. If the fault persists, set the no motor lock current threshold [NO_MTR_THR] to 5%.
3. For low inductance motors, increase PWM switching frequency [PWM_FREQ_OUT].

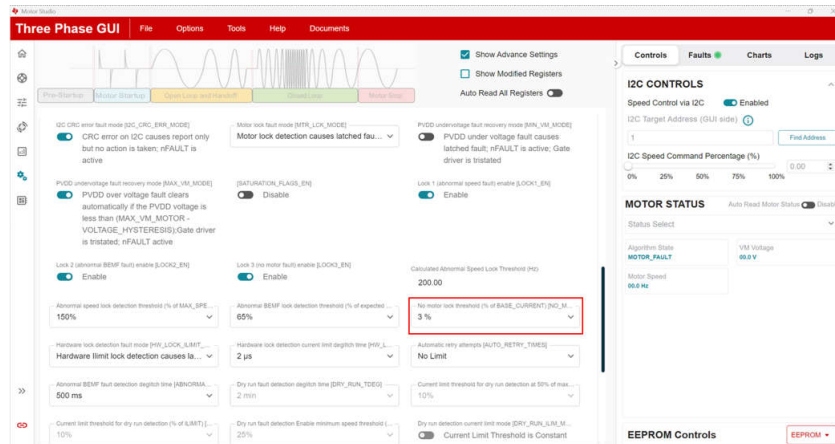


Figure 4-5. NO_MTR

Note

The MCF8329HS might trigger loss of sync [LOSS_OF_SYNC] when motor phases are disconnected while the motor is spinning.

4.5 Abnormal Speed [ABN_SPEED]

This fault gets triggered when motor speed exceeds abnormal speed threshold [LOCK_ABN_SPEED]. When this fault gets triggered, use the following recommendations:

1. Increase align time [ALIGN_TIME], decrease slow first cycle frequency [SLOW_FIRST_CYC_FREQ], or increase the IPD current threshold [IPD_CURR_THR] and IPD repeat times [IPD_REPEAT] depending on the start-up mode selected.
2. Decrease open loop acceleration A1 [OL_ACC_A1] and open loop acceleration A2 [OL_ACC_A2].
3. Decrease closed loop acceleration [CL_ACC].

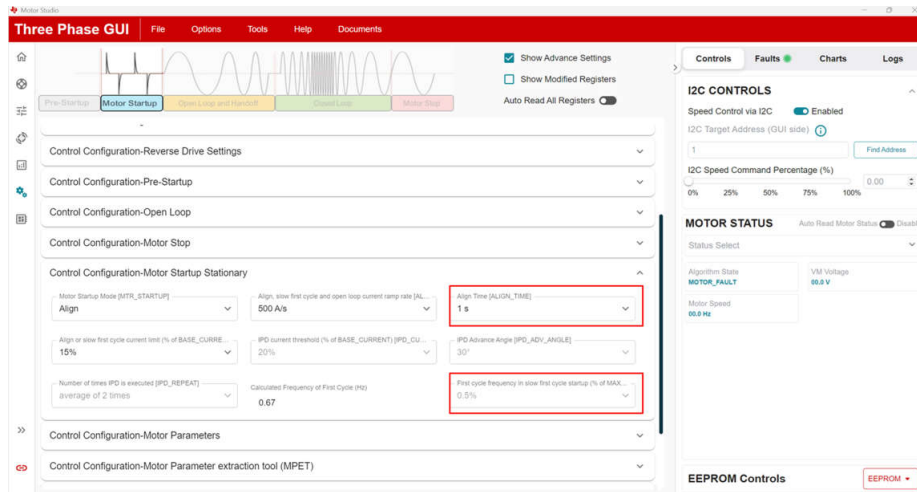


Figure 4-6. ABN_SPEED

5 Hardware Design and Setup

The goal of this section is to help users select appropriate components for the external power stage components of the MCF8329HS and set up the user configurable settings of the MCF8329HSEVM.

5.1 Board Design

The following sections provide equations and guidelines for selecting power stage components to achieve desired performance from the motor driver system.

5.1.1 External MOSFET Selection

The MOSFETs for the external half bridge that can be supported by the MCF8329HS can be determined by inputting the MOSFET gate charge, output PWM switching frequency, and PVDD voltage into the [Max Qg MOSFET Calculator Tool](#) available on ti.com.

5.1.2 Gate Resistor Selection

Selection of an appropriate gate resistance to limit the gate drive current so that the drain-to-source voltage slew rate (VDS) is set to an appropriate level for the external MOSFETS is essential to achieving good system performance. For more information on the importance of and how to select an appropriate gate resistor value, see the *Gate Drive Current and Gate Resistor Selection* sections in the [MCF8329HS Sensorless Field Oriented Control \(FOC\) Three-phase BLDC Gate Driver Data Sheet](#). To simplify the gate resistance selection process, the [Gate Resistor Calculator](#) can be used to estimate the gate resistance required to achieve a desired VDS rise and fall time with an accuracy of $\pm 30\%$.

5.1.3 Bootstrap and GVDD Capacitor Selection

The bootstrap and GVDD capacitors must both be sized appropriately to maintain the bootstrap voltage above the under-voltage lockout threshold during normal operation. For instructions to determine an appropriate capacitance for both the bootstrap capacitors and GVDD capacitor, see the *Bootstrap Capacitor and GVDD Capacitor Selection* sections of the [MCF8329HS Sensorless Field Oriented Control \(FOC\) Three-phase BLDC Gate Driver Data Sheet](#).

5.1.4 Current Shunt Resistor Selection

The internal FOC algorithm uses the output of the internal current sense amplifier (CSA) in its computations. It is recommended to set the max measurable current of the internal CSA to 10% above the motors stall current. To determine an appropriate value for the CSA gain and external low-side shunt resistor, see section 7.3.5 of the [MCF8329HS Sensorless Field Oriented Control \(FOC\) Three-phase BLDC Gate Driver Data Sheet](#).

5.1.5 VREG MOSFET Selection

The GCTRL pin can be used to drive an external MOSFET that can be used as a voltage regulator to provide power to the VREG pin to reduce power dissipation within the MCF8329HS. Instructions on how to select an appropriate MOSFET are provided in section 8.2.1 of the [MCF8329HS Sensorless Field Oriented Control \(FOC\) Three-phase BLDC Gate Driver Data Sheet](#).

5.1.6 Additional External Power Stage Components

For additional considerations for external components of the MCF8329HS and for high power systems, see the *System Considerations in High Power Designs*, *Capacitor Voltage Ratings*, and *External Power Stage Components* sections of the [MCF8329HS Sensorless Field Oriented Control \(FOC\) Three-phase BLDC Gate Driver Data Sheet](#).

IMPORTANT NOTICE AND DISCLAIMER

TI PROVIDES TECHNICAL AND RELIABILITY DATA (INCLUDING DATASHEETS), DESIGN RESOURCES (INCLUDING REFERENCE DESIGNS), APPLICATION OR OTHER DESIGN ADVICE, WEB TOOLS, SAFETY INFORMATION, AND OTHER RESOURCES "AS IS" AND WITH ALL FAULTS, AND DISCLAIMS ALL WARRANTIES, EXPRESS AND IMPLIED, INCLUDING WITHOUT LIMITATION ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT OF THIRD PARTY INTELLECTUAL PROPERTY RIGHTS.

These resources are intended for skilled developers designing with TI products. You are solely responsible for (1) selecting the appropriate TI products for your application, (2) designing, validating and testing your application, and (3) ensuring your application meets applicable standards, and any other safety, security, regulatory or other requirements.

These resources are subject to change without notice. TI grants you permission to use these resources only for development of an application that uses the TI products described in the resource. Other reproduction and display of these resources is prohibited. No license is granted to any other TI intellectual property right or to any third party intellectual property right. TI disclaims responsibility for, and you fully indemnify TI and its representatives against any claims, damages, costs, losses, and liabilities arising out of your use of these resources.

TI's products are provided subject to [TI's Terms of Sale](#), [TI's General Quality Guidelines](#), or other applicable terms available either on ti.com or provided in conjunction with such TI products. TI's provision of these resources does not expand or otherwise alter TI's applicable warranties or warranty disclaimers for TI products. Unless TI explicitly designates a product as custom or customer-specified, TI products are standard, catalog, general purpose devices.

TI objects to and rejects any additional or different terms you may propose.

Copyright © 2026, Texas Instruments Incorporated

Last updated 10/2025