FOC Motor Control

Reference Design

FPGA-RD-02325-1.0

January 2026

Disclaimers

Lattice makes no warranty, representation, or guarantee regarding the accuracy of information contained in this document or the suitability of its products for any particular purpose. All information herein is provided AS IS, with all faults, and all associated risk is the responsibility entirely of the Buyer. The information provided herein is for informational purposes only and may contain technical inaccuracies or omissions, and may be otherwise rendered inaccurate for many reasons, and Lattice assumes no obligation to update or otherwise correct or revise this information. Products sold by Lattice have been subject to limited testing and it is the Buyer's responsibility to independently determine the suitability of any products and to test and verify the same. LATTICE PRODUCTS AND SERVICES ARE NOT DESIGNED, MANUFACTURED, OR TESTED FOR USE IN LIFE OR SAFETY CRITICAL SYSTEMS, HAZARDOUS ENVIRONMENTS, OR ANY OTHER ENVIRONMENTS REQUIRING FAIL-SAFE PERFORMANCE, INCLUDING ANY APPLICATION IN WHICH THE FAILURE OF THE PRODUCT OR SERVICE COULD LEAD TO DEATH, PERSONAL INJURY, SEVERE PROPERTY DAMAGE OR ENVIRONMENTAL HARM (COLLECTIVELY, "HIGH-RISK USES"). FURTHER, BUYER MUST TAKE PRUDENT STEPS TO PROTECT AGAINST PRODUCT AND SERVICE FAILURES, INCLUDING PROVIDING APPROPRIATE REDUNDANCIES, FAIL-SAFE FEATURES, AND/OR SHUT-DOWN MECHANISMS. LATTICE EXPRESSLY DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY OF FITNESS OF THE PRODUCTS OR SERVICES FOR HIGH-RISK USES. The information provided in this document is proprietary to Lattice Semiconductor, and Lattice reserves the right to make any changes to the information in this document or to any products at any time without notice.

Inclusive Language

This document was created consistent with Lattice Semiconductor's inclusive language policy. In some cases, the language in underlying tools and other items may not yet have been updated. Please refer to Lattice's inclusive language FAQ 6878 for a cross reference of terms. Note in some cases such as register names and state names it has been necessary to continue to utilize older terminology for compatibility.

Contents

Figures

Tables

Abbreviations in This Document

A list of abbreviations used in this document.

Abbreviation	Definition
ADC	Analog-to-Digital Converter
AHB	Advanced High-Performance Bus
AHB-Lite	Advanced High-Performance Bus Lite
APB	Advanced Peripheral Bus
BLDC	Brushless Direct Current
COM	Communication
CPU	Central Processing Unit
DC	Direct Current
DIP	Dual In-line Package
FOC	Field-Oriented Control
FPGA	Field Programmable Gate Array
GPIO	General Purpose Input/Output
GUI	Graphical User Interface
HDL	Hardware Description Language
IP	Intellectual Property
IRQ	Interrupt Request
KSPS	Kilo Samples Per Second
MOSFET	Metal-Oxide Semiconductor Field-Effect Transistor
PC	Personal Computer
PDM	Predictive Data Maintenance
PID	Proportional Integral Derivative
PLL	Phase-Locked Loop
PWM	Pulse Width Modulator
RC	Read to Clear
RISC-V	Reduced Instruction Set Computer V
RO	Read Only
RPM	Revolutions Per Minute
RSVD	Reserved
RTL	Register Transfer Level
RW	Read and Write
SCLK	Serial Clock
SPI	Serial Peripheral Interface
SVPWM	Space Vector Pulse Width Modulator
UART	Universal Asynchronous Receiver Transmitter
USB	Universal Serial Bus
WO	Write Only

1. Introduction

The Field-Oriented Control (FOC) Motor Control reference design comprises a complete RISC-V embedded system integrated with the FOC Motor Control IP. A RISC-V MC CPU core is used in the system to configure and initialize the FOC Motor Control IP registers. The FOC Motor Control IP is built with the combination of a custom register transfer level (RTL) IP core and MATLAB-generated modules. This reference design includes the MATLAB model, which allows you to generate hardware description language (HDL) code using the MATLAB HDL coder. The generated MATLAB HDL modules are instantiated in the FOC Motor Control IP. This reference design also includes a graphical user interface (GUI) that allows the host PC to communicate with the Lattice Certus™-NX FPGA device through the universal asynchronous receiver transmitter (UART) interface.

1.1. Quick Facts

Download the reference design files from the Lattice FOC Motor Control Reference Design web page.

Table 1.1. Summary of the Reference Design

General	Target Devices	LFD2NX-40-8BG256C
	Source code format	Verilog, C
Simulation	Functional simulation	Partially performed
	Timing simulation	Not performed
	Test bench	Available
	Test bench format	—
Software Requirements	Software tool and version	Lattice Radiant™ Software 2024.2.1.330.0 Lattice Propel™ Builder 2024.2.2503210206_SP1 MATLAB R2024a
	IP version	SPI Controller IP v2.3.0 UART IP v1.3.0 GPIO IP v1.6.2 RISC-V MC CPU IP v2.5.0 AHB-Lite Interconnect Module v1.3.0 APB Interconnect Module v1.2.0 AHB-Lite to APB Bridge Module v1.1.0 PLL IP v1.8.0 System Memory IP v2.3.0 EnDat2.2 Master IP v1.0.22[1] FOC Motor Control IP v3.0[1]
Hardware Requirements	Board	Certus-NX Versa Evaluation Board RS485 Encoder Transceiver Board Trenz Electronic TEP0002 motor driver board
	Cable	Heidenhein 8-pin M12 encoder cable (1133832-01) USB 2.0 cable 24-V power supply and adapter

Note:

1. This is a third-party vendor IP.

1.2. Features

Key features of the FOC Motor Control reference design include:

- Position control
- Speed control
- Magnetic field current control (Id)
- Torque current control (Iq)
- 25-bit position feedback from encoder

- Single-turn encoder position feedback
- 12-bit analog-to-digital converter (ADC) current feedback for three-phase brushless direct current (BLDC) motor
- Moving average filter for current feedback
- 50-kHz control loop frequency
- Space Vector Pulse Width Modulator (SVPWM) algorithm
- 50-kHz pulse width modulator (PWM) switching frequency
- 5-ns PWM resolution
- 313-KSPS ADC current sampling rate
- MATLAB model configured based on the reference design hardware specifications

1.3. Naming Conventions

1.3.1. Nomenclature

The nomenclature used in this document is based on Verilog HDL.

1.3.2. Signal Names

Signal names that end with:

- _n are active low signals (asserted when value is logic 0)
- _i are input signals
- _o are output signals

2. Directory Structure and Files

Figure 2.1 shows the directory structure.

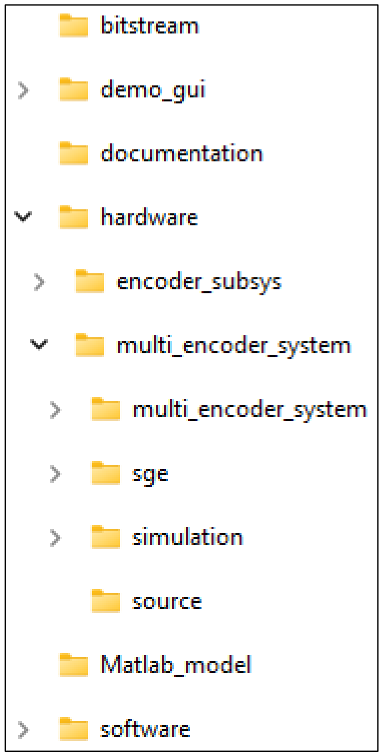

Figure 2.1. Directory Structure

The FOC Motor Control reference design package includes the hardware design files, software project, MATLAB model, and GUI for communication between the host PC and Lattice FPGA device.

Table 2.1 shows the list of directories included in the reference design package.

Table 2.1. Directory List

Directory	Description
bitstream	Contains the Radiant project bitstream compiled from the *hardware* folder.
demo_gui	Contains the .exe file to run the motor control solution kit on the host PC by connecting to the FPGA device through the UART interface.
documentation	Contains the user guide.
hardware	Contains the Radiant project and Propel Builder RISC-V embedded system.
hardware/simulation	Contains the simulation testbench.
hardware/multi_encoder_system/ multi_encoder_system	Contains the RTL HDL files generated from Propel Builder.
hardware/multi_encoder_system/source	Contains the RTL HDL files generated from the MATLAB model located in /Matlab_model.
hardware/multi_encoder_system/simulation	Contain a simple testbench and the QuestaSim .do file.
Matlab_model	Contains the FOC motor control MATLAB model .slx and .m files.
software	Contains the C code to run the software for this reference design. The compiled .mem file is loaded into the system memory module of the project.

3. Functional Description

The FOC Motor Control reference design reuses the RISC-V embedded system design of the Closed Loop BLDC Motion Control reference design but with the Motor Control IP replaced with the FOC Motor Control IP to enable field-oriented control features.

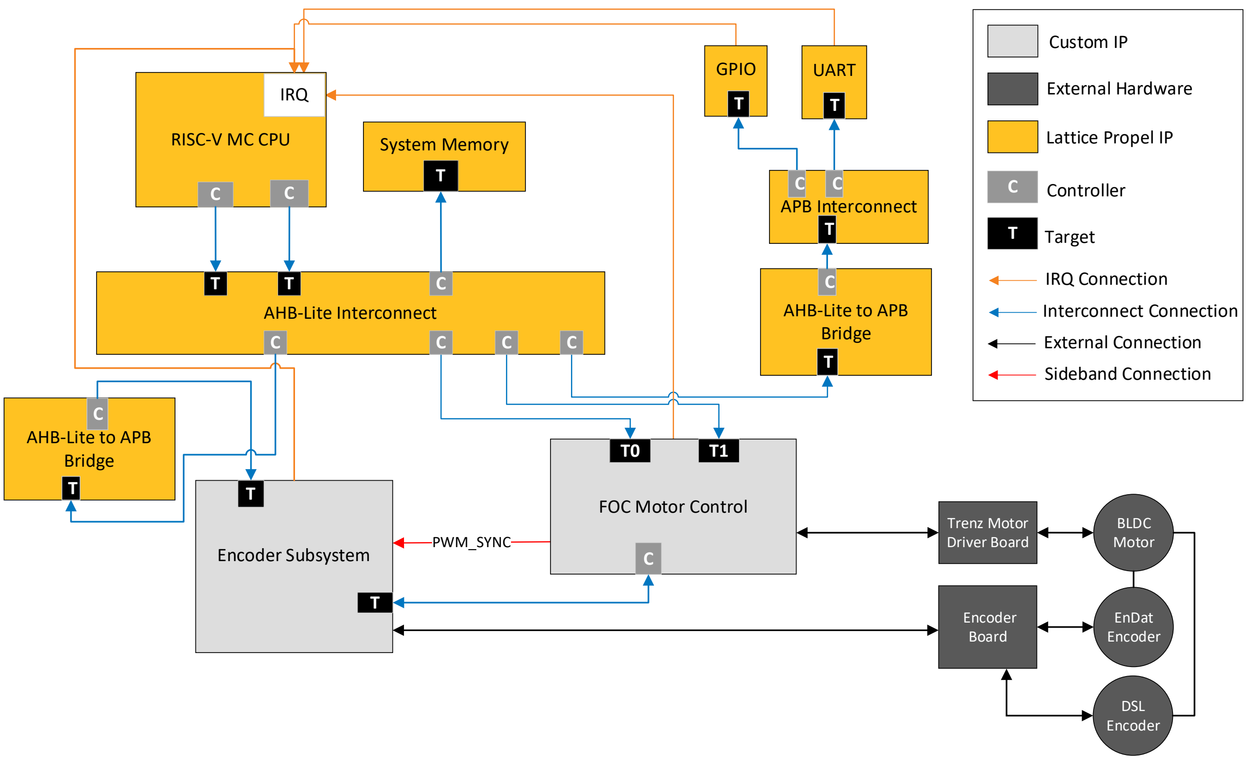

Figure 3.1. Reference Design Block Diagram

3.1. Design Components

The FOC Motor Control reference design includes the following components:

- UART Controller
- General-Purpose Input/Output (GPIO)
- RISC-V MC CPU
- Phase-locked loop (PLL)
- FOC Motor Control
- Advanced High-Performance Bus (AHB)-Lite Interconnect
- Advanced Peripheral Bus (APB) Interconnect
- AHB-Lite to APB Bridge
- System Memory
- Encoder Subsystem

This document primarily describes the FOC Motor Control IP. For more information on the other components, refer to the Closed Loop Brushless Direct Current (BLDC) Reference Design User Guide (FPGA-RD-02308).

3.1.1. FOC Motor Control

The FOC Motor Control IP is built with a combination of custom RTL IPs and MATLAB-generated modules. The blue blocks represent modules configured and generated using MATLAB. The yellow blocks represent newly created custom IPs. The grey block represents the existing custom IP reused from the BLDC motor control IP. The blocks shaded both yellow and blue represent MATLAB-generated modules that require additional enhancements (or customization) before integration into the system.

There are three cascaded control loops in the system: the outer loop is the position control loop, followed by the speed control loop, and finally the current control loop. The current control loop drives the PWM module, which outputs three-phase pulse signals to switch MOSFETs on the Trenz Electronic motor driver board. These MOSFETs generate three-phase voltages that drive the motor.

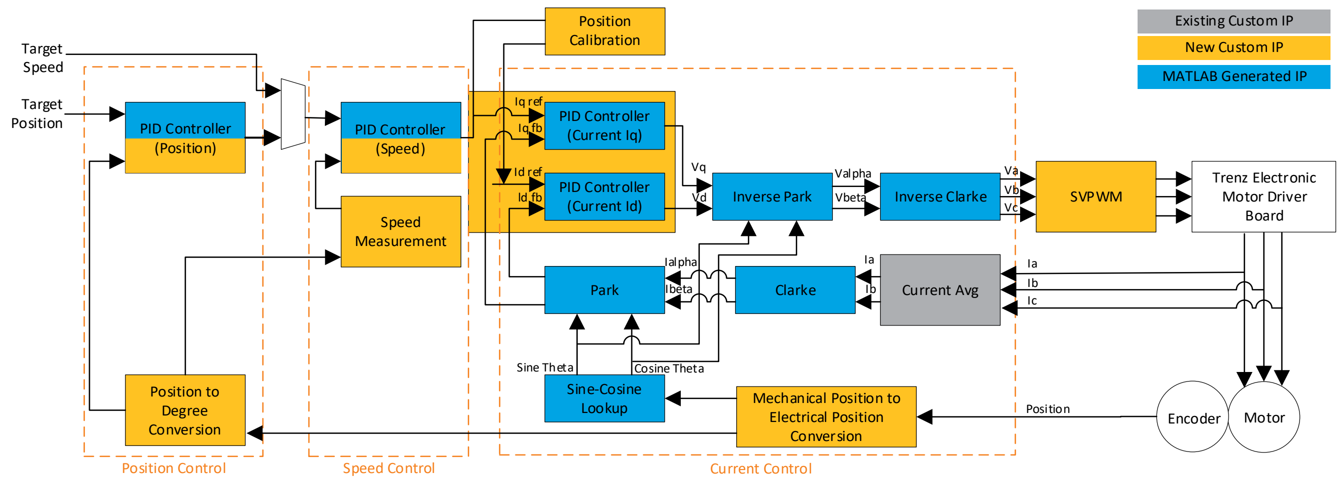

Figure 3.2. FOC Motor Control IP

3.1.1.1. Control Loop Parameters

Table 3.1 shows the default values for the parameters of the FOC motor control loops.

Table 3.1. Default Values for FOC Motor Control Loop Parameters

Parameter	Position PID Controller (RPM/°)	Speed PID Controller (A/RPM)	Id PID Controller (V/A)	Iq PID Controller (V/A)
Proportional gain (Kp)	800 (fixed point)[1] 0.782 (floating point)	30 (fixed point)[2] 0.0018 (floating point)	50 (fixed point)[3] 0.781 (floating point)	50 (fixed point)[3] 0.781 (floating point)
Integral gain (Ki)	1000 (fixed point)[3] 15.625 (floating point)	2048 (fixed point)[1] 2 (floating point)	128 (fixed point)[3] 2 (floating point)	128 (fixed point)[3] 2 (floating point)

Notes:

1. Fixed-point number format is sfix16_En10 (signed 16-bit fixed-point number with 10 fractional bits).
2. Fixed-point number format is sfix16_En14 (signed 16-bit fixed-point number with 14 fractional bits).
3. Fixed-point number format is sfix16_En6 (signed 16-bit fixed-point number with 6 fractional bits).

3.1.1.2. Motor Control IP Parameters

Table 3.2 shows the motor control IP parameters.

Table 3.2. Motor Control IP Parameters

Parameter	Default Value	Description
Device Family	LFD2NX	Specifies the Lattice FPGA device family used.
Enable Absolute Encoder Finite State Machine Logic	1	Enables or disables the absolute encoder finite state machine logic. 0 – Disable; when disabled, you must create an encoder master module that generates encoder position outputs, which need to be connected to enc_pos_i and enc_pos_valid_i of the motor control IP. 1 – Enable; when EnDat encoder master module is used in your reference design.
Enable ADC Current samples averaging logic	1	Enables or disables averaging logic for ADC current samples. 0 – Disable; current samples from the ADC output are fed directly into the FOC motor control algorithm. 1 – Enable; current samples from the ADC output undergo a 16-tap moving-average filtering process before being fed into the FOC motor control algorithm.
Enable Dualport RAM to store 3-phase current values	0	Enables or disables the Dualport RAM for storing 3-phase current values. 0 – Disable 1 – Enable **Note:** Values stored are intended for use in the predictive maintenance feature.
MatLab Model Connection Data Width	16	Sets the MATLAB motor control model interface control and status bus width.
Encoder bus data width	25	Sets the encoder position width.
Width of Pulse Width Modulator counter	14	Sets the PWM counter width.
Motor voltage (v)	24	Sets the BLDC motor voltage.
Motor Control Loop frequency (kHz)	50	Sets the sampling frequency of the motor control current, speed, and position loops.
Ratio of control loop frequency vs PWM Switching frequency	0	Sets the ratio of the control loop frequency to the PWM switching frequency. 0 – Ratio of 1:1 1 – Ratio of 1:2 2 – Ratio of 1:5 3 – Ratio of 1:10
Pulse Width Modulator counter clock frequency (MHz)	200	Sets the frequency of the PWM counter.
Analog-to-Digital converter pin compatible bit	12	Sets the width of the current samples from the ADC output.

3.1.1.3. Control and Status Registers

All registers are accessed through the AHBL_S0 interface (base address 0x11800).

Table 3.3. Register Access Types

Access Type	Access Type Abbreviation	Behavior on Read Access	Behavior on Write Access
Read only	RO	Returns register value	Ignores write access
Write only	WO	Returns 0	Updates register value
Read and write	RW	Returns register value	Updates register value

Access Type	Access Type Abbreviation	Behavior on Read Access	Behavior on Write Access
Read to clear	RC	Returns register value	Reads to clear or reset the register bit to the default value

Table 3.4. FOC Motor Control IP Registers

Register Offset	Register Name	Description	Access Point
0x00	MTRCR0	Sets motor pole and position control derivative gain values.	AHBL_S0
0x04	MTRCR1	Sets maximum RPM.	AHBL_S0
0x08	MTRCR2	Sets current control gain value for Id (magnetic field current).	AHBL_S0
0x0C	MTRCR3	Sets current control gain value for Iq (torque current).	AHBL_S0
0x10	MTRCR4	Sets number of rotations and angle for position control.	AHBL_S0
0x14	MTRCR5	Sets speed control gain value.	AHBL_S0
0x18	MTRCR6	Motor control register	AHBL_S0
0x1C	MTRCR7	Sets speed control derivative gain value and target RPM.	AHBL_S0
0x20	MTRCR8	Sets position control gain value.	AHBL_S0
0x24	RSVD	Reserved	—
0x28	Status 0	Indicates the current motor speed and position value of the motor rotor.	AHBL_S0
0x2C	Status 1	Indicates the motor control status.	AHBL_S0
0x30–0x34	RSVD	Reserved	—
0x38	Current Sampling Status	Indicates the current sampling calibration done status.	AHBL_S0
0x3C–0x5C	RSVD	Reserved	—
0x60	Encoder Raw Position	Indicates the encoder raw position value.	AHBL_S0

Table 3.5. MTRCR0 Register

Field	Name	Access	Default	Description
[31:24]	RSVD	RO	‘d0	Reserved.
[23:16]	MTRPOLES	RW	‘d4	Number of motor stator pole pairs. This value must be configured according to the data sheet for the specific motor.
[15:0]	PARAMPOSCONTROL_D	RW	‘d1024	Derivative gain (Kd) value used in position PID controller.

Table 3.6. MTRCR1 Register

Field	Name	Access	Default	Description
[31:16]	RSVD	RO	‘d0	Reserved.
[15:0]	MAXRPM	RW	‘d2000	Maximum RPM is the upper limit RPM. Valid values are dependent on the maximum RPM supported by the motor.

Table 3.7. MTRCR2 Register

Field	Name	Access	Default	Description
[31:16]	PARAMCURRENTCONTROL_P_D	RW	‘d50	Kp value used in magnetic field current Id PID controller.
[15:0]	PARAMCURRENTCONTROL_I_D	RW	‘d128	Ki value used in magnetic field current Id PID controller.

Table 3.8. MTRCR3 Register

Field	Name	Access	Default	Description
[31:16]	PARAMCURRENTCONTROL_P_Q	RW	‘d50	Kp value used in torque current Iq PID controller.
[15:0]	PARAMCURRENTCONTROL_I_Q	RW	‘d128	Ki value used in torque current Iq PID controller.

Table 3.9. MTRCR4 Register

Field	Name	Access	Default	Description
[31:16]	NUM_ROTATION	RW	‘d1	Number of full rotations.
[15:0]	ANGLE_DEGREE	RW	‘d180	Angle in degrees. current position angle (in degrees) + angle_degree = final position angle (in degrees)

Table 3.10. MTRCR5 Register

Field	Name	Access	Default	Description
[31:16]	PARAMSPEEDCONTROL_P	RW	‘d30	Kp value used in speed PID controller.
[15:0]	PARAMSPEEDCONTROL_I	RW	‘d2048	Ki value used in speed PID controller.

Table 3.11. MTRCR6 Register

Field	Name	Access	Default	Description
[31]	ENGAGE	RW	‘d0	Sync signal to latch all control registers from AHB-L clock domain (100 MHz) to motor clock domain (20 MHz). 0 – No updates to motor or predictive data maintenance (PDM) control registers 1 – Transfers all control register from AHB-L holding registers to motor active registers
[30]	DIRECTION	RW	‘d1	Indicates the direction of motor. 0 – Counter clockwise 1 – Clockwise
[29]	POS_CALIBRATION	RW	‘d0	Enables motor rotor position calibration to zero degrees. 0 – Disable 1 – Enable
[28]	MOTOR_ENABLE	RW	‘d0	Enables motor drivers. 0 – Disable 1 – Enable
[27]	MOTOR_ESTOP	RW	‘d0	Emergency stop. 0 – Normal operation 1 – Engages E-brakes without sync delay or ENGAGE
[26]	RSVD	RO	‘d0	Reserved.

Field	Name	Access	Default	Description
[25]	STOP	RW	'd0	Holds the motor in position. 0 – Normal operation 1 – Stops motor rotation
[24]	POS_CONTROL_EN	RW	'd0	Enables position control mode. 0 – Disable 1 – Enable
[23:0]	RSVD	RO	'd0	Reserved.

Table 3.12. MTRCR7 Register

Field	Name	Access	Default	Description
[31:16]	PARAMSPEEDCONTROL_D	RW	'd0	Kd value used in speed PID controller.
[15:0]	TARGET_RPM	RW	'd120	Speed value in RPM.

Table 3.13. MTRCR8 Register

Field	Name	Access	Default	Description
[31:16]	PARAMPOSCONTROL_P	RW	'd800	Kp value used in position PID controller.
[15:0]	PARAMPOSCONTROL_I	RW	'd1000	Ki value used in position PID controller.

Table 3.14. Status 0 Register

Field	Name	Access	Default	Description
[31:16]	CURRENT_POS	RO	'd0	Indicates the current arrow position on the motor wheel. 0 – Min ⋮ 360 – Max **Note:** The actual value shown on the Position gauge of the demo GUI is (360 – CURRENT_POS).
[15:0]	CURRENT_SPEED	RO	'd0	Indicates the current motor speed in RPM. 0 – Min ⋮ 2000 – Max

Table 3.15. Status 1 Register

Field	Name	Access	Default	Description
[31]	ENC_LINK_STAT	RO	'd0	0 – Encoder link is not established 1 – Encoder link is established
[30]	RSVD	RO	'd0	Reserved.
[29]	APB_IDLE	RO	'd1	0 – APB controller is active 1 – APB controller is inactive
[28:8]	RSVD	RO	'd0	Reserved.
[7]	DRIVE_FAULT	RO	'd0	0 – Drive fault has not occurred 1 – Drive fault has occurred This bit, from the motor driver board driving the actual motor, is set when an overcurrent fault is detected by the protection circuit.
[6:5]	RSVD	RO	'd0	Reserved.
[4]	STOP	RO	'd0	0 – Motor is at non-zero RPM 1 – Motor is at zero RPM

Field	Name	Access	Default	Description
[3]	RPM_LOCK	RO	‘d0	Indicates whether motor is at target RPM. 0 – Motor is not at target RPM 1 – Motor is at target RPM
[2]	RSVD	RO	‘d0	Reserved.
[1]	RSVD	RO	‘d0	Reserved.
[0]	MOV	RO	‘d0	Indicates whether motor is moving. 0 – Motor stops or is coasting. 1 – Motor is moving under control

Table 3.16. Current Sampling Status Register

Field	Name	Access	Default	Description
[31:3]	RSVD	RO	‘d0	Reserved.
[2]	CAL_DONE	RO	‘d0	ADC offset calibration status. 0 – Offset calibration is not complete 1 – Offset calibration is complete
[1:0]	RSVD	RO	‘d0	Reserved.

Table 3.17. Encoder Raw Position Register

Field	Name	Access	Default	Description
[31:25]	RSVD	RO	‘d0	Reserved.
[24:0]	ENCODER_POS	RO	—	25-bit EnDat encoder position value.

3.1.1.4. Hardware Specifications

The control algorithm is configured based on the following hardware specifications.

Table 3.18. Hardware Specifications

Hardware	Parameter	Specification
Motor	Brand	Anaheim
	Name (Model)	Brushless DC motor (BLY174D-24V-4000)
	Rated Voltage (V)	24
	Rated Power (W)	104
	Rated Torque (oz-in)	35.4
	Rated Speed (RPM)	4000
	Winding Type	4 pole pairs
Encoder	Brand	Heidenhain
	Name (Model)	Rotary encoder (ROQ 437)
	Rotation	Single rotation
	Position Values per Revolution	25 bits
ADC Chip[1]	Brand	Texas Instruments
	Name (Model)	Precision ADC (ADS7947)
	Sampling Rate (KSPS)	313
	Number of Bits	12
	SPI Interface SCLK Frequency (MHz)	20
	Dual Channel	Channel 0 – Current Channel 1 – Voltage

Note:

1. Located on the Trenz Electronic motor driver board.

3.2. Clocking Scheme

The clocking scheme implemented in this reference design is similar to the Closed Loop BLDC Motion Control reference design. For more information, refer to the Closed Loop Brushless Direct Current (BLDC) Reference Design User Guide (FPGA-RD-02308).

3.3. Reset Scheme

The reset scheme implemented in this reference design is similar to the Closed Loop BLDC Motion Control reference design. For more information, refer to the Closed Loop Brushless Direct Current (BLDC) Reference Design User Guide (FPGA-RD-02308).

4. Running the Reference Design

This section describes how to run the FOC Motor Control reference design using the Lattice Radiant software. For more details on the Lattice Radiant software, refer to the Lattice Radiant Software User Guide. Once the hardware is set up and the FPGA device is configured, launch the demo GUI on the host PC to enable communication with the FPGA device through the UART interface.

4.1. Compiling the Reference Design

The reference design has been pre-compiled.

4.2. Generating the Bitstream File

The bitstream file of this reference design is pre-generated.

4.3. Programming the Bitstream into the FPGA Device

To program the generated bitstream into the FPGA device using the Lattice Radiant software, perform the following:

1. Click **Tools > Programmer**. The **Lattice Radiant Programmer** window opens.
2. Click **Detect Cable** to ensure the FPGA device is connected and detected properly**.**

Figure 4.1. Lattice Radiant Programmer – Detecting Cable

3. Select *LFD2NX* and *LFD2NX-40* for the **Device Family** and **Device** fields, respectively.
4. Select the .bit file in /bitstream.
5. Click **Program Device**. The output console displays the operation successful message when the bitstream is programmed successfully.

5. Simulating the Reference Design

A simple testbench is provided in the FOC Motor Control reference design to showcase the functionality of the motor control algorithm. Three sinusoidal phase current values are generated and assigned to the FOC Motor Control IP internal signals o_phase_current_a, o_phase_current_b, and o_phase_current_c to emulate the three-phase feedback currents from the motor. In addition, the testbench generates encoder feedback position values from 0 to 360 degrees and provides the values as feedback to the FOC motor control IP.

To simulate the design, perform the following:

1. In the Radiant software, select **Tools > QuestaSim Lattice Edition**. The **Questa Lattice OEM Edition** window opens.

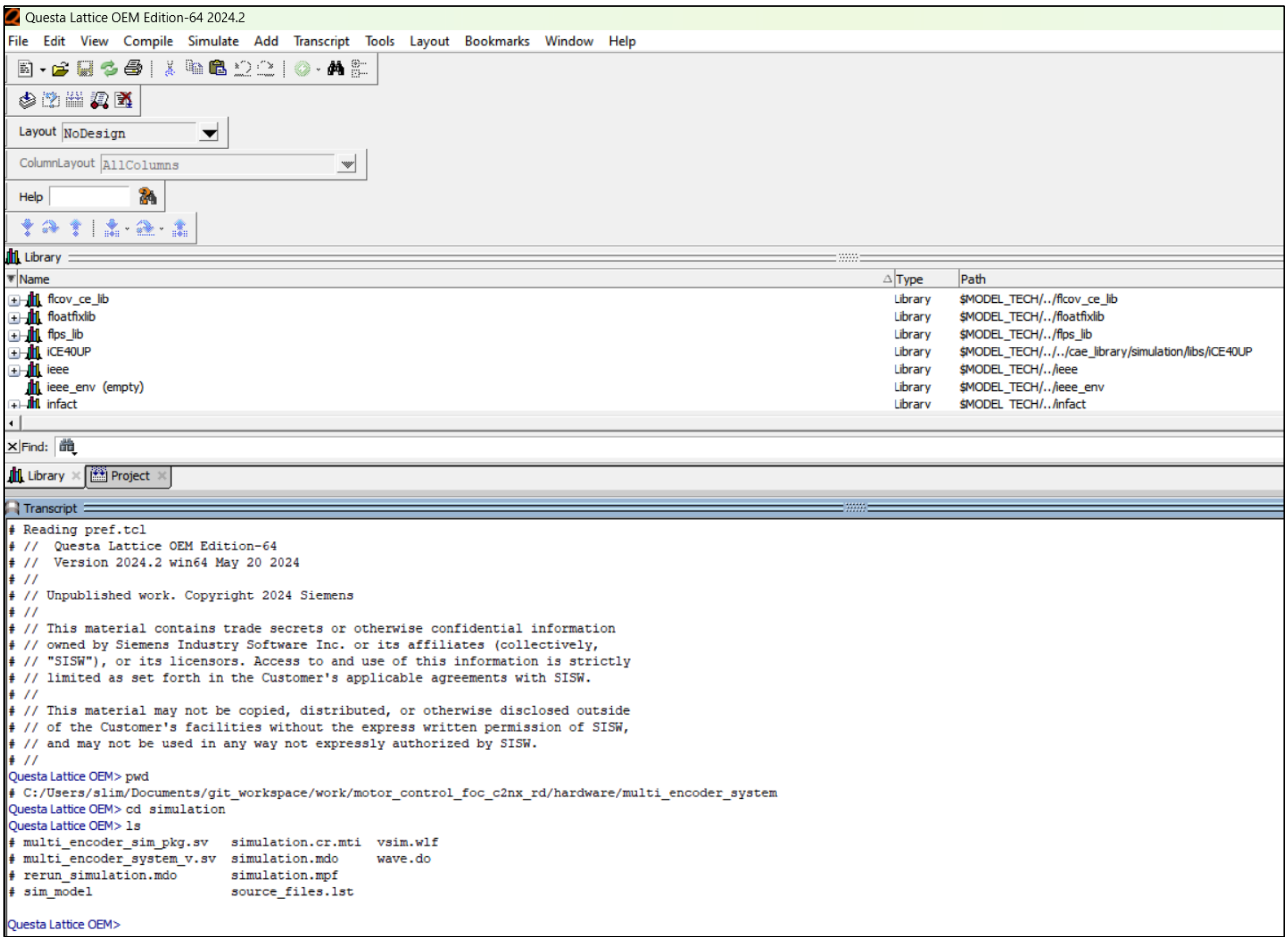

Figure 5.1. Questa Lattice OEM Edition Window

2. Type *cd simulation* in the **Transcript** console.
3. Type *do simulation.mdo* in the **Transcript** console.
4. Type *do wave.do* in the **Transcript** console. The **Wave** window opens.

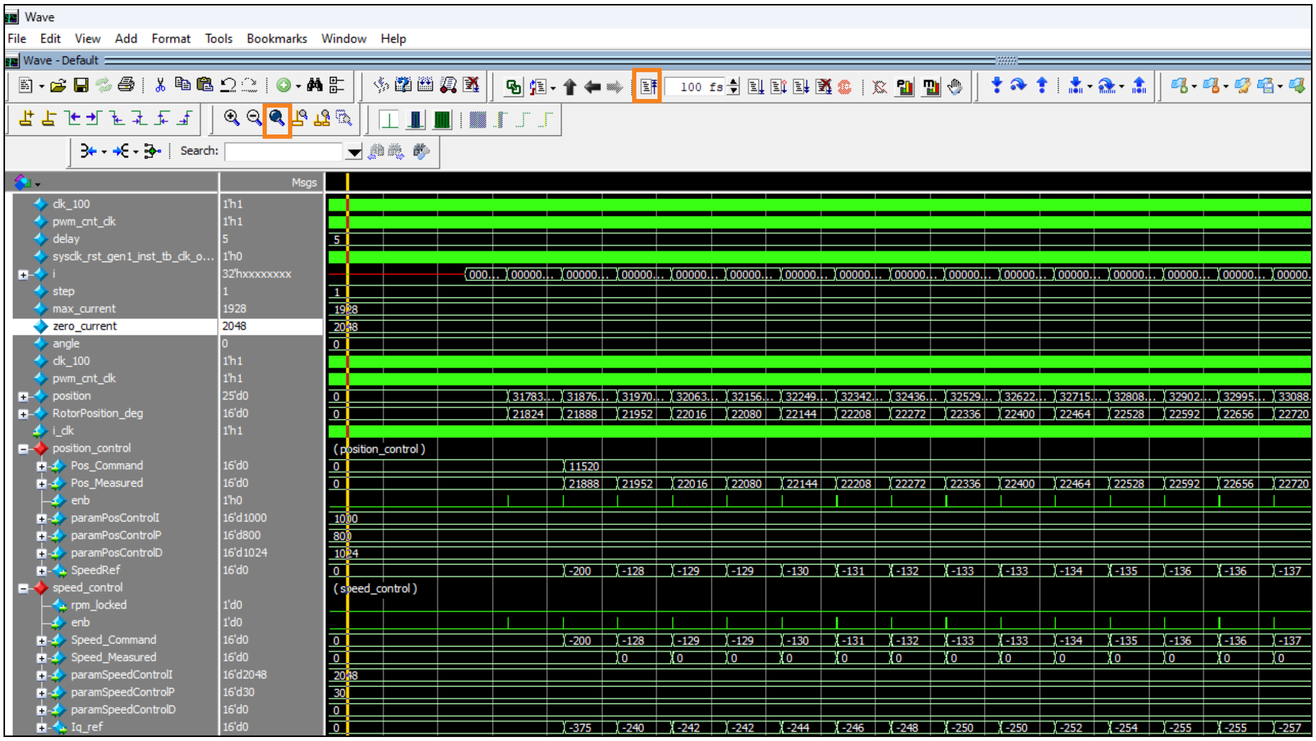

Figure 5.2. Wave Window

5. Click the **Restart** button to restart the simulation.
6. Click the **Zoom Full** button to view the waveforms at full zoom.

6. Implementing the Reference Design on Board

Make sure the DIP switch on the Certus-NX Versa Evaluation Board is set correctly as indicated in the following figure.

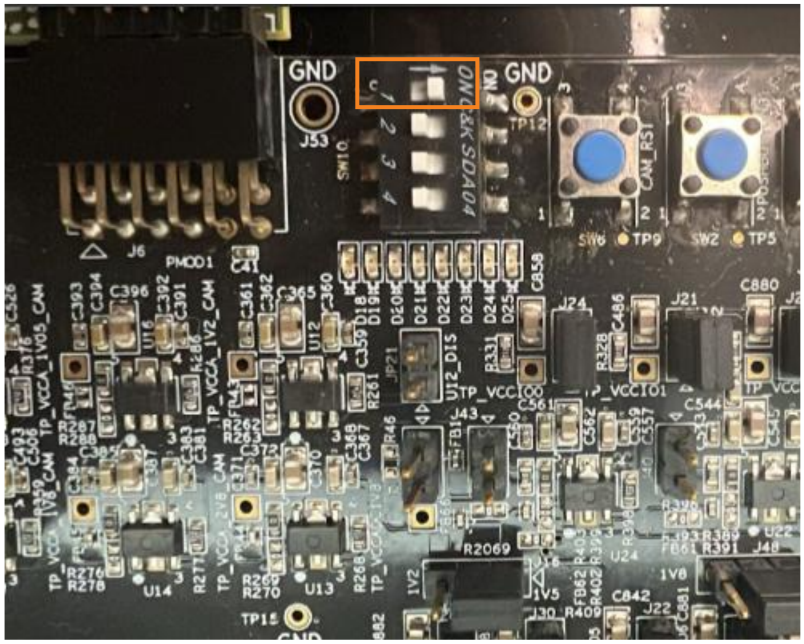

Figure 6.1. Setting Board DIP Switch

7. Using the Motor Control Solution Kit

The demo GUI allows you to run the motor control solution kit.

7.1. Setting Up the UART Interface

To set up the UART interface, perform the following:

1. In the **UART Configuration** group box, select the COM port associated with your universal serial bus (USB) programming cable from the **COM Port** drop down list.
2. Click **Connect**.

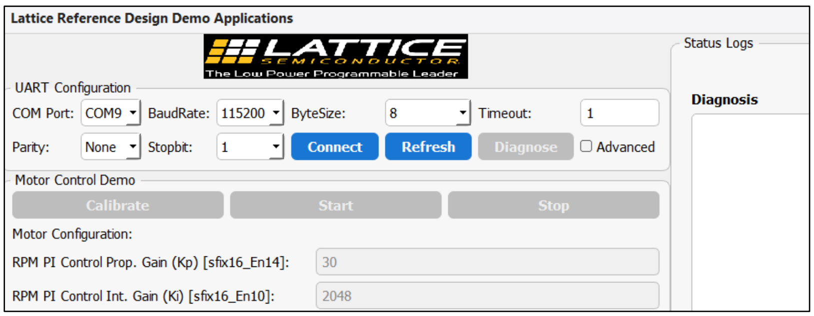

Figure 7.1. Setting Up the UART Interface

7.2. Calibrating the Motor Rotor

To calibrate the motor rotor to position 0 for the first time, perform the following:

1. Adjust the motor wheel arrow so that it points to the 12 o'clock position. This position is assumed to represent 0 degrees.
2. In the **Motor Control Demo** group box, click **Calibrate**. The **Wheel Adjustment Required** dialog box opens.

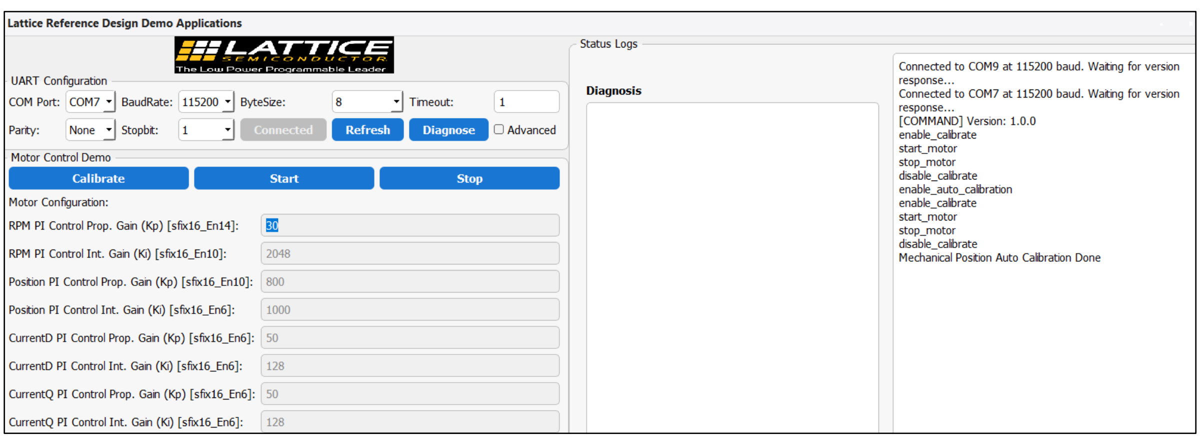

Figure 7.2. Calibrating the Motor Rotor

3. Click **OK** in the dialog box to continue. Calibration begins.
4. Confirm that calibration is complete by verifying that the message *Mechanical Position Auto Calibration Done* appears in the **Status Logs** pane.

7.3. Running in Speed Control Mode

To run in speed control mode, perform the following:

1. In the **Motor Control Demo** group box, check the **Speed** checkbox under the **Configure** button.

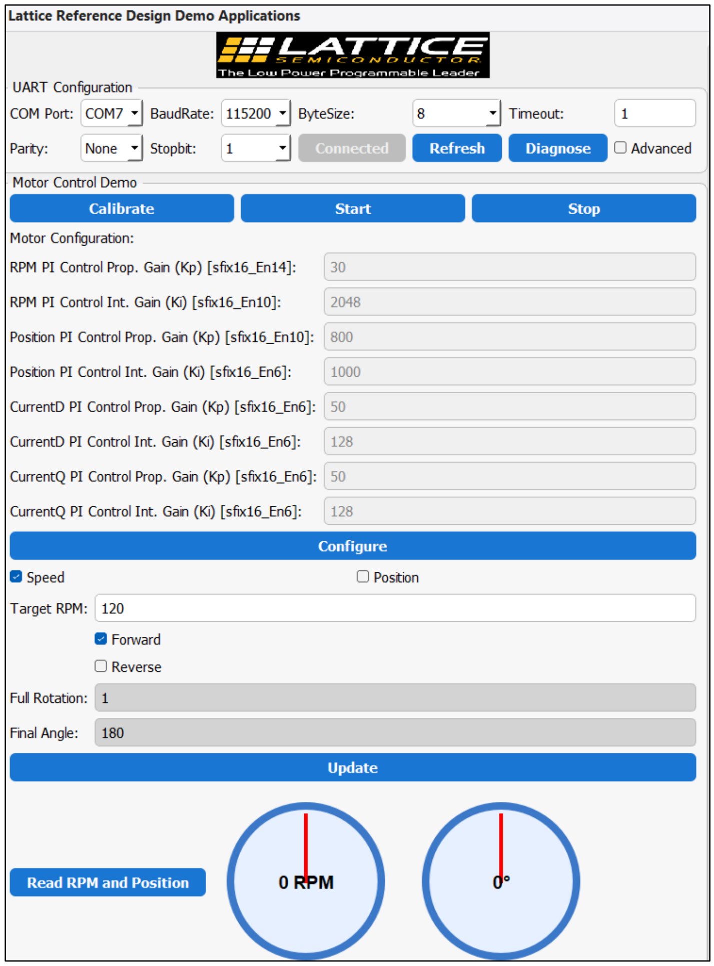

Figure 7.3. Running in Speed Control Mode

2. Enter the target RPM value (from 10 to 2000) in the **Target RPM** field.
3. Check either the **Forward** or **Reverse** checkbox.
4. Click **Update**.
5. Click **Start**. The motor begins turning.
6. Click **Read RPM and Position** to check the RPM status.
7. To stop the motor, click **Stop**.
8. (Optional) To change the RPM value and direction of the motor, repeat Steps 2, 3, and 4.
   Note: You can change the RPM value and direction while the motor is running.

7.4. Running in Position Control Mode

To run in position control mode, perform the following:

1. In the **Motor Control Demo** group box, check the **Position** checkbox under the **Configure** button.

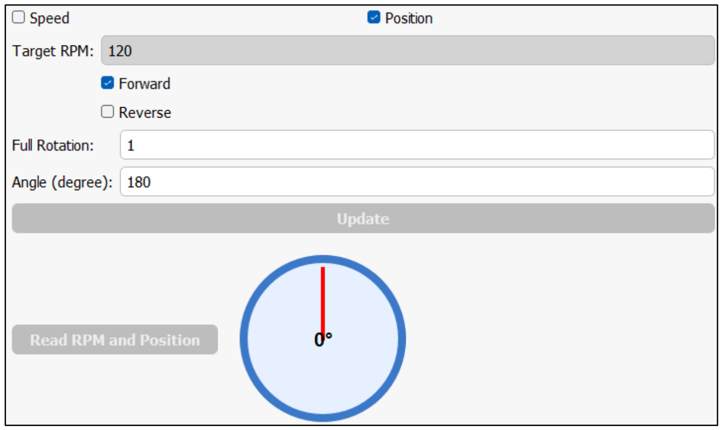

Figure 7.4. Running in Position Control Mode

2. Check either the **Forward** or **Reverse** checkbox.
3. Enter the number of full rotations (from 0 to 10) in the **Full Rotation** field.
4. Enter the angle value (from 1 to 360 degrees) in the **Angle** field.
5. Click **Update**. The motor begins turning. The final position of the motor is determined by adding the angle entered in the **Angle** field to the current rotor position.

7.5. Recalibrating the Motor Rotor

To recalibrate the motor rotor to a new position 0, perform the following:

1. In the **UART Configuration** group box, click **Refresh**.

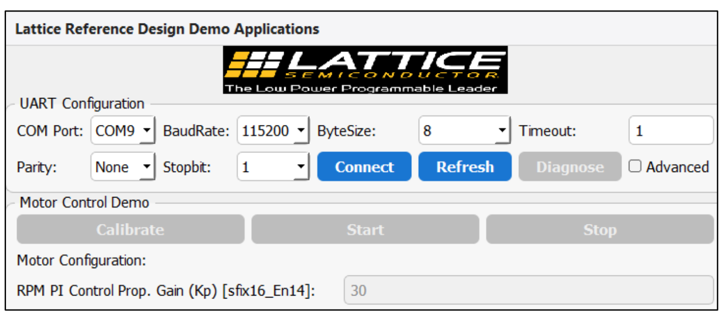

Figure 7.5. Recalibrating the Motor Rotor

2. Select the COM port associated with your USB programming cable from the **COM Port** drop down list.
3. Click **Connect**.
4. Adjust the motor wheel arrow so that it points to the new position. This position is assumed to represent 0 degrees.
5. In the **Motor Control Demo** group box, click **Calibrate**. The **Wheel Adjustment Required** dialog box opens.
6. Click **OK** in the dialog box to continue. Calibration begins.
7. Confirm that calibration is complete by verifying that the message *Mechanical Position Auto Calibration Done* appears in the **Status Logs** pane.

8. Editing the MATLAB Motor Control Model

The FOC Motor Control reference design includes the MATLAB motor control model. This model is used to generate the HDL files of the FOC motor control modules. You can edit the MATLAB motor control model parameters to make changes to the FOC motor control algorithm.

To open the MATLAB motor control model, perform the following:

1. Open MATLAB.
2. Click **Open** on the MATLAB toolbar.
3. Select the reference design ZIP file *hdlcoderFocCurrentFixptHdl.slx* at /Matlab_model. The MATLAB motor control model opens in the **Simulink** window.

To generate the RTL HDL files of the MATLAB motor control model, perform the following:

1. In the **Simulink** window, click **APPS** followed by **HDL Coder**. The **HDL CODE** tab appears.
2. On the **HDL CODE** tab, click **Settings**. The **Configuration Parameters** window opens.
3. Click **HDL Code Generation** on the left pane to access basic HDL code generation settings.
4. Click **Browse** associated with **Code Generation Folder**, then navigate to the location you want to generate the RTL HDL files.
5. Click **Target** under **HDL Code Generation** on the left pane, then set **Workflow** to *Generic ASIC/FPGA* and **Synthesis Tool** to *No synthesis tool specified*.
6. Click **OK** to close the **Configuration Parameters** window.
7. On the **HDL CODE** tab, click **Generate HDL Code**. The **Diagnostic Viewer** shows the HDL code generation status and indicates whether HDL code generation is successful.

9. Debugging

9.1. Auto-Calibration Failed Error Message

Auto-calibration failed! Please manually rotate the motor wheel a few degree to the left then hit calibrate button!!

This error occurs when the motor rotor arrow is positioned between two motor poles (for example, 45°, 135°, 225°, or 315°). At this position, the magnetic force is insufficient to pull the rotor into alignment with the stator pole.

To resolve this issue, follow these guidelines:

- Manually move the motor rotor to avoid the specified angles during calibration.
- Then, click the **Calibrate** button.

9.2. Calibration Timed Out Error Message

Calibration timed out: No response received from UART in 20 seconds.

This error occurs when the system does not receive a response from the UART interface within the 20-second timeout period during calibration.

9.3. Motor Configuration Kp and Ki Values are Grayed Out

The Kp and Ki values are grayed out because the FOC Motor Control IP currently supports only a fixed hardware configuration. The Kp and Ki values cannot be modified.

References

- Closed Loop Brushless Direct Current (BLDC) Reference Design User Guide (FPGA-RD-02308)
- Certus-NX web page
- Certus-NX Versa Evaluation Board web page
- RS485 Encoder Transceiver Board web page
- Closed Loop BLDC Motion Control Reference Design web page
- SPI Controller IP Core web page
- UART IP Core web page
- GPIO IP Core web page
- RISC-V MC CPU IP Core web page
- AHB-Lite Interconnect Module web page
- APB Interconnect Module web page
- AHB-Lite to APB Bridge Module web page
- Trenz Electronic web page
- Anaheim Automation web page
- Heidenhain web page
- Siemens Questa Advanced Simulator web page
- Lattice Radiant Software web page
- Lattice Propel Design Environment web page
- Lattice Solutions Reference Designs web page
- Lattice Solutions IP Cores web page
- Lattice Insights for Lattice Semiconductor training courses and learning plans

Technical Support Assistance

Submit a technical support case through www.latticesemi.com/techsupport.

For frequently asked questions, refer to the Lattice Answer Database at www.latticesemi.com/Support/AnswerDatabase.

Revision History

Revision 1.0, January 2026

Section	Change Summary
All	Initial release.