Automate Stack 3.1 Demo

User Guide

FPGA-UG-02207-1.1

February 2024

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 REDUDANCIES, 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.

Contents

Figures

Tables

Acronyms in This Document

A list of acronyms used in this document.

Acronym	Definition
BLDC	Brushless Direct Current
FPGA	Field Programmable Gate Array
IIOT	Industrial Internet of Things
IP	Internet Protocol
OPCUA	Open Platform Communications United Architecture
SPI	Serial Peripheral Interface
UART	Universal Asynchronous Receiver Transmitter
UDP	User Datagram Protocol

1. Introduction

The Lattice Automate™ Stack showcases a variety of solutions for industrial automation. The Automate demo uses an end-to- end system for motor control and ML/AI powered predictive maintenance. However, the components of the Automate Stack are flexible enough to be applicable for a wide range of industrial automation use cases.

There are two FPGA systems in the Automate Stack: the Main System and the Node System.

The node system controls a motor and collects the data from the motor for the predictive maintenance neural network to use. The node system can connect to external sensors through three different protocols (SPI, UART/Modbus, and I^2C), and has two-way communication with the main system through Etherconnect.

The main system has a RISC-V CPU running FreeRTOS, a real time operating system often used in embedded applications.

All modules in the main system and the node system FPGA designs are connected to each other using an AXI4 bus interface. This interface has high throughput and can run at a frequency of up to 100 MHz. It allows parallel data transfer, meaning a read and a write between two modules on the AXI bus can happen simultaneously.

The main system acts as a central hub which passes commands to up to eight different node systems. The main system also contains the AI/ML powered predictive maintenance system, which uses a neural network to analyze the collected data and predict failure ahead of time. In the Automate demo, this collected data is the current signature of the motor, but the PDM IP block can use any quantitative data for its input if the neural network is trained on the same type of data.

The Lattice Automate 3.1 solution stack uses the Golden System reference design (GSRD) developed on Lattice CertusPro™-NX FPGA as a base starting template. For Automate 3.1, the key IP blocks like UDP and EtherConnect are easily integrated into the GSRD template to build the entire Automate embedded system.

The Golden Software Reference Design (GSRD) is a comprehensive embedded system created by incorporating GHRD with the necessary device drivers. GSRD is a thoroughly tested known good design showcasing a system using FPGA-based RISC-V soft processor, intended to be used as a baseline project.

The GSRD comprises of the following components:

- GHRD (Hardware components designed by Lattice and can be easily build using the Lattice Propel™ builder environment).
- Reference RISC-V software components:
  - Bootloader
  - FreeRTOS device drivers for all IPs
  - FreeRTOS sample applications

Note: For more details on GHRD and GSRD, refer to the GHRD/GSRD Reference Design web page.

With the addition of these IP the total device usage is about 85% while still providing users with additional resource to implement minor design customization.

Moreover, Automate Stack 3.1 supports OPC-UA based packet exchange between the Main and Node system for various data transfers. OPC-UA, which stands for Open Platform Communications Unified Architecture, is a standardized communication framework designed for industrial automation and the Industrial Internet of Things (IIoT). This platform-independent and service-oriented architecture facilitates secure data exchange among diverse industrial systems. OPC-UA is known for its versatility, supporting various communication models and data types. With features such as platform independence, robust security mechanisms, and extensibility, OPC UA has become a widely embraced protocol in industrial communication. It plays a crucial role in enabling seamless connectivity and information exchange across different devices and systems in a standardized fashion.

Note: For more details on the OPC-UA, refer to Automate Stack 3.1 Reference Design (FPGA-RD-02284).

In the Automate system, you interact with the main system through the user application running on a PC. The user interface and main system are connected with an Ethernet cable and communicate with each other using OPC-UA, a machine-to-machine pub-sub communication protocol frequently used in the IIOT.

2. Demo System Setup

The Automate demo uses one main system (running on a CertusPro-NX Versa board) and at least one node system (running on a Certus™-NX Versa board). It can support up to 16 node systems. It is recommended only to run up to eight node systems for best performance. Each node system is connected to a motor driver board, which is connected to a brushless DC motor. The main system holds the predictive maintenance CNN co-processor.

The main system connects to the Host PC. The Host PC communicates through OPCUA with a PC running the user interface (Lattice Automate 3.1), through which you interact with the motors and receives PDM data.

2.1. System Level Block Diagram

Figure 2.1 shows the system-level block diagram. The Host PC running the Automate 3.1 user interface is connected to the main system over a standard 1G Ethernet cable. The main system running on CertusPro-NX is connected to the node system over EtherConnect.

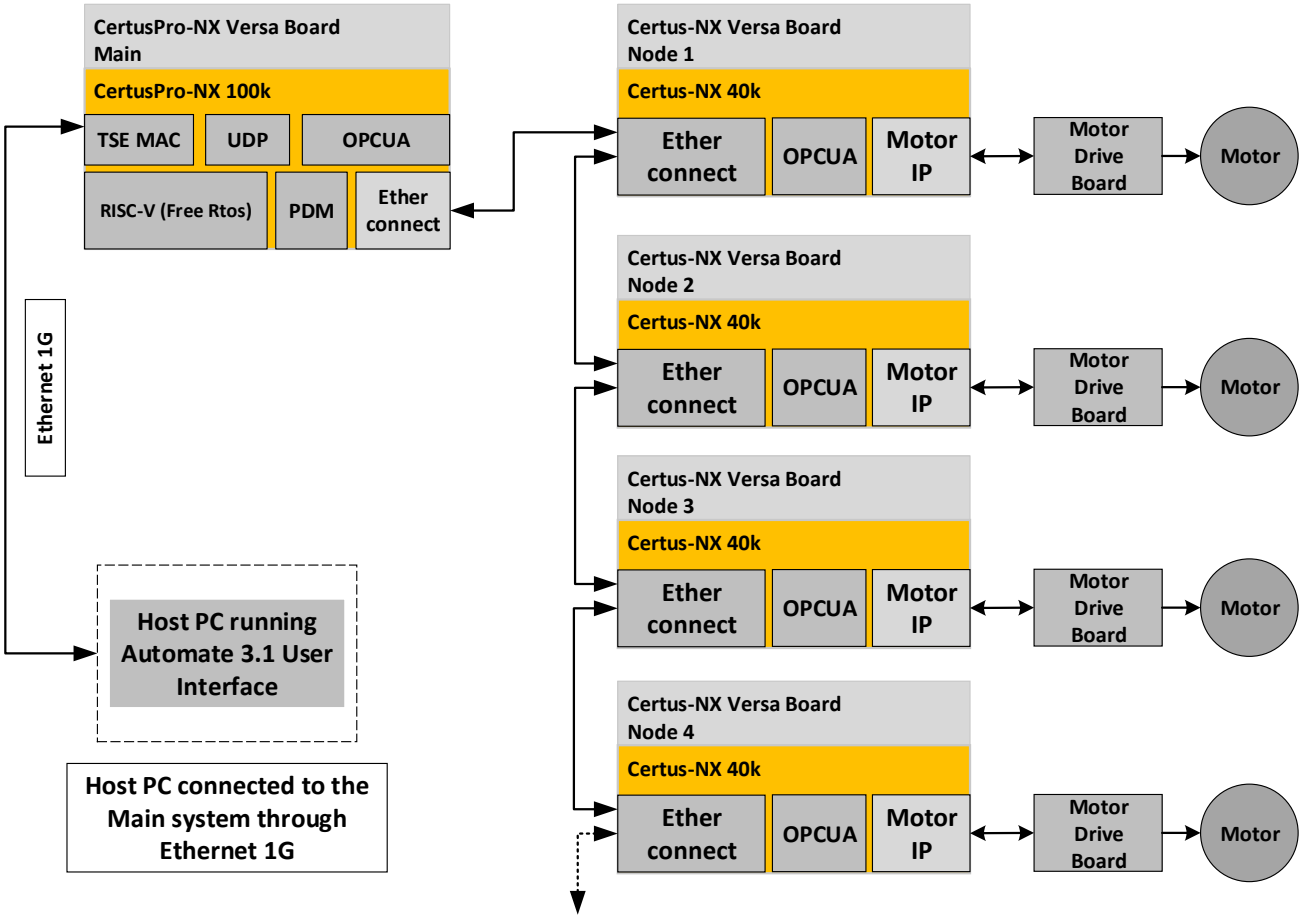

Figure 2.1. System Block Diagram

3. Communication Protocols (OPCUA)

Automate 3.1 uses OPCUA as a data exchange standard for communication between the host PC and the main system FPGA board. This protocol is frequently used in IIOT settings.

3.1. Main System to Host PC Communication - OPCUA

OPCUA is a cross-platform open-source data exchange standard developed by the OPC Foundation. It is frequently used in industrial applications for machine-to-machine communication. The OPCUA standard can be implemented on various platforms and is independent of the communication protocol. It was originally developed for data collection and control for industrial equipment, but it has now scaled to be appropriate for many more use cases, including building automation and cloud applications.

OPCUA has two variations, client/server and publish/subscribe. Automate 3.1 uses the publish/subscribe version of OPCUA.

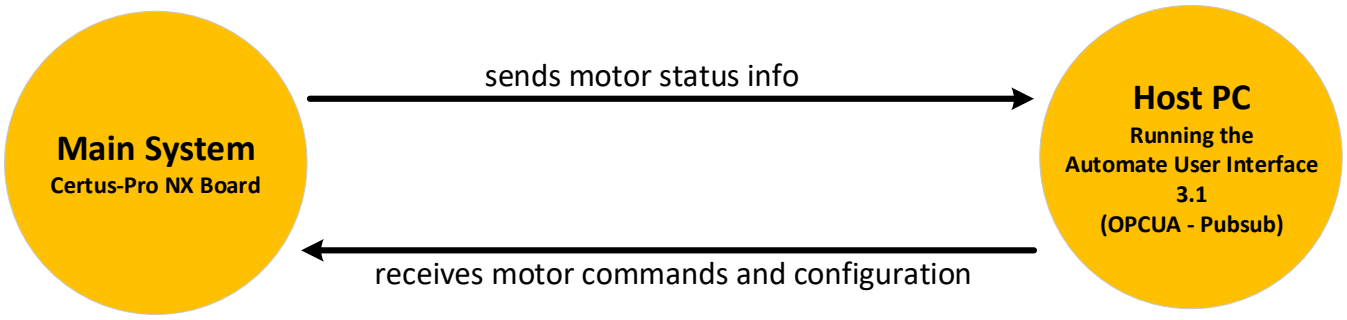

Figure 3.1. OPCUA High-Level Block Diagram

3.2. Main System to Node System Communication – OPCUA

The Main system and Node system communicate using the OPC UA protocol (client server mode). The Main system acts as the central hub for the system, receiving requests from the host PC and routing the relevant ones to the appropriate Node system based on their intended purpose. The Node system is responsible for performing operations based on the requests received from the Main system and sending the results back to the Main system for further processing.

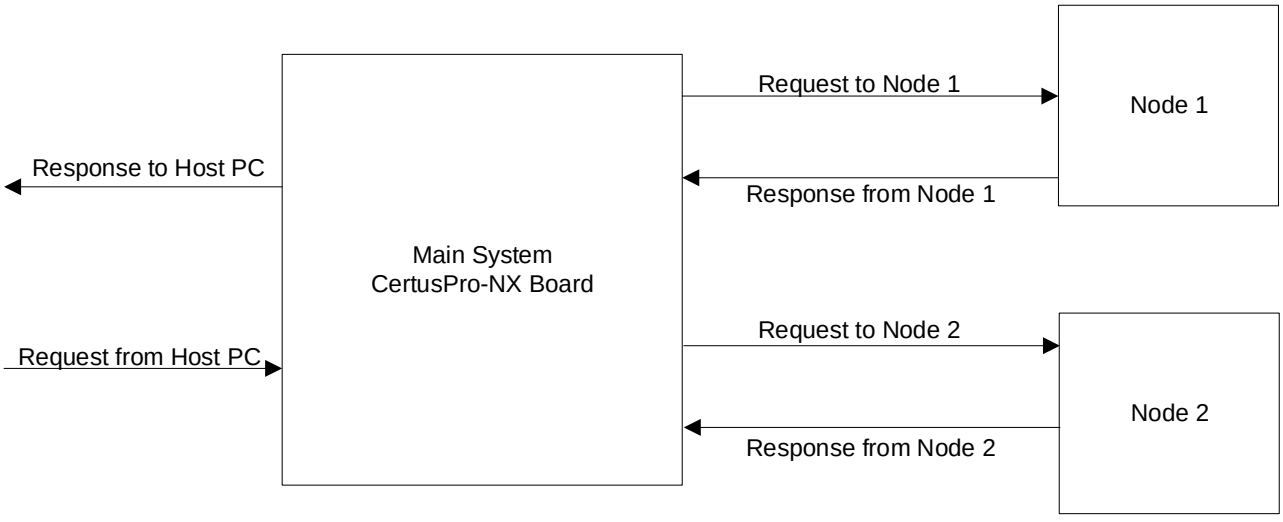

Figure 3.2. Main and Node System Block Diagram

3.3. Node System to Node System Communication

Each node is connected in a chain of sequential nodes. It can communicate with the master using port 0 and can be connected to another Node module using port 1 for the first node. For subsequent nodes, node 0 is connected using port 0, and node 2 is connected using port 1, continuing in this sequential chain of nodes. Synchronization between different nodes is required to drive each node simultaneously.

3.4. UDP Communication

User Datagram Protocol (UDP/IP) is a communications protocol used for establishing connections between applications on the Internet. The UDP Protocol is a transport layer that operates atop the Internet Protocol (IP) Layer and is used for connections where high sustained throughput is a requirement and some data loss is anticipated, such as video and audio streaming. This UDP IP core for FPGAs offers minimal latency and bandwidth overhead since it transmits data packets without verifying reception and it can support up to 1 Gbps for any low-cost FPGA operating at 125 MHz.

A simple AXI streaming interface is all that is required to start sending and receiving UDP datagrams, and only the user data payload is exchanged between the application and the UDP core. The source UDP port, destination UDP port, destination IP address, and MAC address can be configured through the APB interface. These components can be easily enabled or disabled as needed by the user's application.

The core is delivered in a format that allows direct connection to the Lattice 1G TSE Ethernet MAC IP Core.

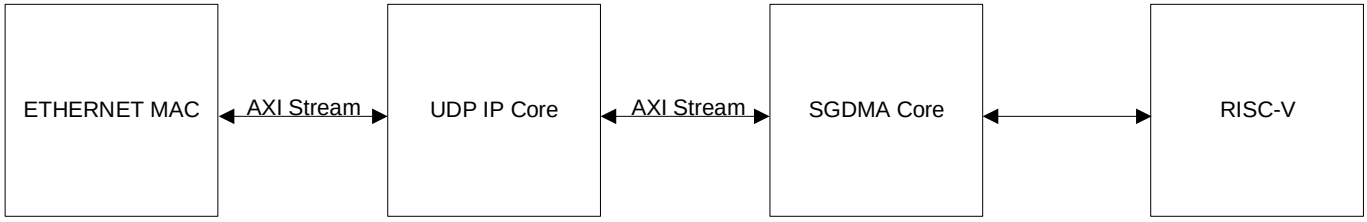

Figure 3.3. UDP IP Core Overview

4. Hardware

To demonstrate the Lattice Automate Solution stack, the below hardware is required. In the current demo a brushless DC (BLDC) motor is used from Anaheim Automation, but user can use any other BLDC motor which has a similar configuration. For the demo to run successfully, the users need to connect at least one node system to the main system.

4.1. Hardware Requirements

This demonstration requires the following hardware components:

- Main system:
  - Lattice CertusPro-NX Versa Evaluation Board
  - USB Type-A (UART) cable for programming the bitstream and binary files
    Note: One cable can be reused to program the main system and each node system, one at a time.
  - Electrical 1G SFP(s) Model FS SFP-GB-GE-T to make an Ethernet connection on the main system board (Insert at J15 ports of CertusPro-NX Versa Board).
  - Ethernet cable to connect the main system to the host PC
  - 12 V power adapter for board power
- Node system(s):
  Note: Demo supports up to 16 node systems. Requirements for 1 node system are listed:
  - Lattice Certus-NX Versa Evaluation Board(s)
  - 12 V power adapter(s) for board power
  - Ethernet cable(s) to connect the node system to the main system, and daisy chain node system boards.
  - Aardvark I2C/SPI Host Adapter to test Node peripheral interfaces (optional)
- Motor(s):
  - Anaheim Automation Brushless DC Motor(s) Model GB-42 BLS 24 V, 5000 RPM
  - Trenz TEP0002 motor control board(s)
  - 24 V-10 Amp DC Power Supply for motor(s)
- User interface (client system):
  - PC running Windows 10 Operating System of 1920 x 1080 resolution, 100% dpi

4.2. Hardware Setup

The details of the hardware connections are illustrated in Figure 4.1.

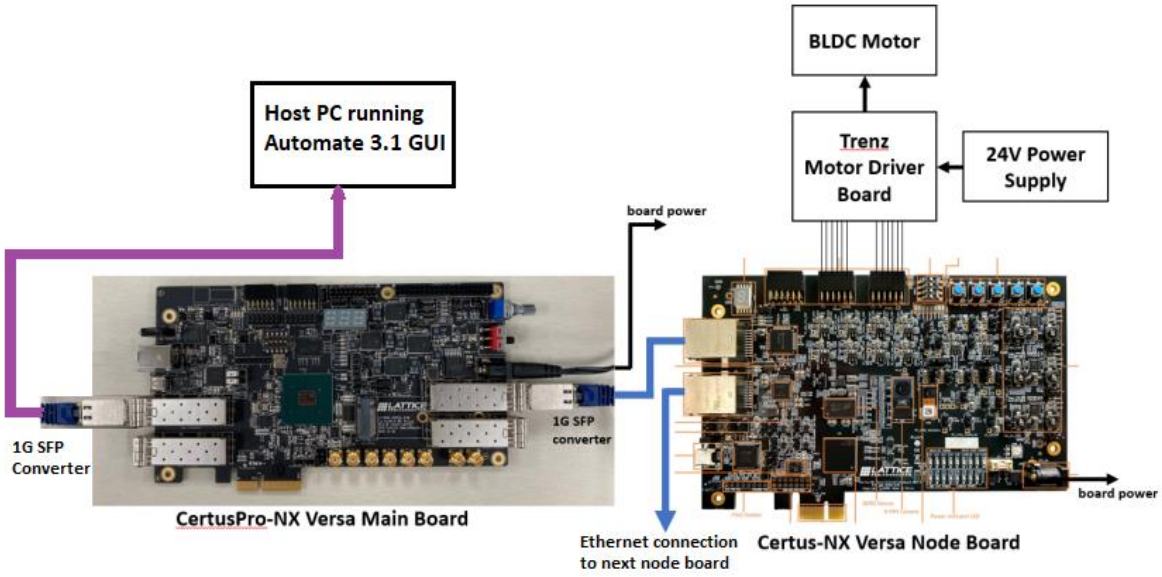

Figure 4.1. System Setup with Host PC (Example with One Node)

4.3. Hardware System Readiness

4.3.1. FPGA Board Programming

To program the FPGA board:

1. Hardware should be connected properly as shown in Figure 4.1.
2. All boards should be programmed with. bit and .bin files.
   a. If the user using a brand new CertusPro-NX Versa board, refer to the Appendix C. Programming a Fresh Main System Board section for programming.
   b. Otherwise, for main and node systems, refer to the Appendix A. User Interface Application Installation (PC) section for board programming.
3. Power cycle every board after programming.
4. After the power cycle, reset the main system (press the SW3 button on the main system).

4.3.2. Power-up Sequence

To start power-up sequence:

1. Power on the Host PC.
2. Power on the node system, motor driver, and motors.
3. Power on the main system.

4.3.3. Status LEDs on Boards

1. Check the main and node system ready LEDs to check connections:
   a. Main System:
      - D67: Main System Power On
      - D64: Main System Ready
      - D63: Ethernet Connection Established
   b. Node System:
      - D32 and D34: Link up LEDs

- D30: Illuminates if the next node in the chain is connected. (Not needed for the last node in the chain.)
- D18 - D25: Node system ready
- 7-segment LED: Node system ready

Note: Wait 60 seconds after power on for these LEDs to illuminate. If the above LEDs are not all illuminated, power cycle all boards and/or refer to the Appendix F. Troubleshooting section.

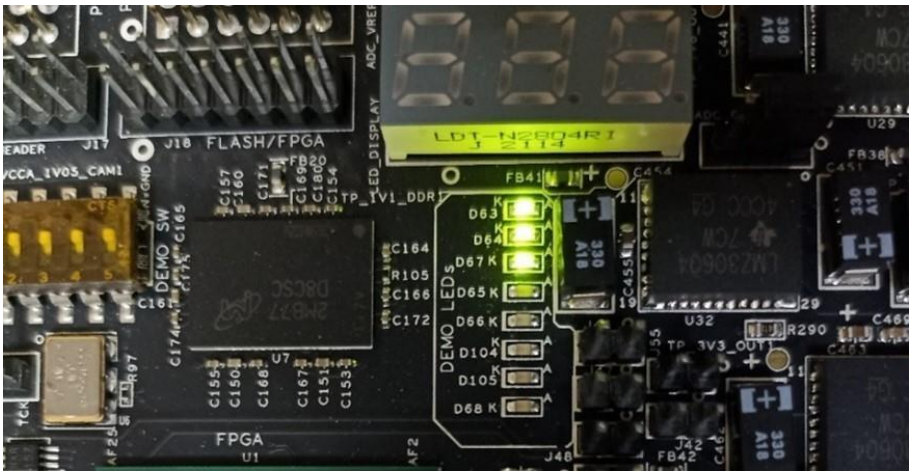

Figure 4.2. Main System Ready LEDs

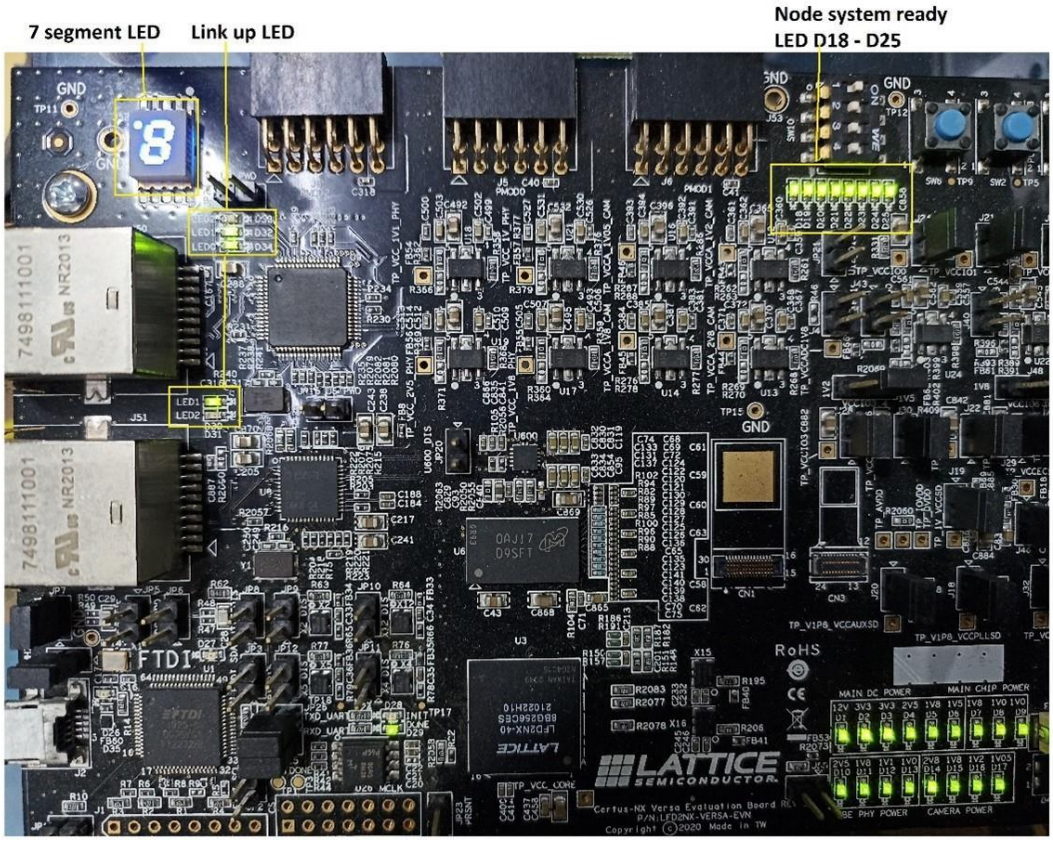

Figure 4.3. Node System Ready LEDs

5. Software Requirements

- Lattice Radiant 2022.1 or later
  Lattice Radiant Programmer 2022.1 or later (part of Radiant install). The following required software programs are available here: www.latticesemi.com/en/Products/DesignSoftwareAndIP.
- Lattice Automate demo bitstreams and binaries are available here under the Design File section:
  https://www.latticesemi.com/en/Solutions/Solutions/SolutionsDetails02/Automate
- Lattice Automate 3.1 Test Application Software
  Note: Lattice Automate 3.1 is available in OPCUA to publish/subscribe version.
  The software programs are available for download only if the user log in at www.latticesemi.com.

5.1. Optional Software

- Total Phase Control Center 4.1 or later for testing node peripherals (purchase may be needed)
- Wireshark 4.2.0 or later (open source)
- Packet Sender 8.5.2 or later (open source)
- Docklight v2.4 or later (open source)

5.2. Software System Readiness

To check the software system readiness:

1. Establish the Ethernet connection before connecting to the user interface.
   a. Follow steps A through C in the Ethernet Connection section to establish the Ethernet connection.
2. User interface installation and running:
   a. Refer to User Interface Application Installation (PC) section to install the user interface application on the PC (one time).
   b. Refer to the Running the Motor through Automate User Interface Application (PC) section to run the user interface application on the PC (every time the user run the demo).

6. Automate Stack File Directory Structure

All files can be downloaded from www.latticesemi.com/Solutions/Solutions/SolutionsDetails02/Automate.

6.1. Executables

These files are programmed onto the main and node FPGA boards.

6.1.1. Main System

The Main System has the following folders.

6.1.1.1. Golden Folder

Table 6.1. Golden Project Files

File Description	File Name	Starting Address in Flash
Golden FW Image without CRC	Golden_App.bin	0x02_000_000
Golden FW Image	Golden_AppCrc.bin	0x02_000_000
Golden FPGA Image	soc_main_system_impl_1.bit	0x00_000_000

6.1.1.2. MCS Folder

The MCS file has both the main system and the golden project files combined into one single file.

- soc_main_system_3_1_impl_1.mcs

6.1.1.3. Primary Folder

Table 6.2. Primary Project Files

File Description	File Name	Starting Address in Flash
Primary FW Image without CRC	c_main_system_3_1_cnn.bin	0x02_0A0_000
Primary FW Image	c_main_system_3_1_cnnCRC.bin	0x02_0A0_000
Primary FPGA Image	soc_main_system_3_1_impl_1.bit	0x00_000_000

6.1.2. Node System

- c_node_system_3_1.bin
- soc_node_impl1.bit

6.2. User Interface

Install the *Automate3p1GUI.exe* user interface to the Host PC.

7. Running the Motor through Automate User Interface Application (PC)

The Lattice Automate 3.1 user interface is developed for users to easily control the motor behavior from the host PC. This user interface allows you to update motor parameters and control the speed and directions of all the motors. The motor running procedure is defined in this section through a Graphical User Interface/Test Application.

Note: For Lattice Automate Stack 3.1 Application Installation on the PC, refer to Appendix A. User Interface Application Installation (PC) section. For programming the main system and node system boards, refer to Appendix B. Programming the Automate Stack on Respective Flash section.

7.1. Starting the Application on Host PC

To start the application on host PC:

1. Open the Lattice Automate application.

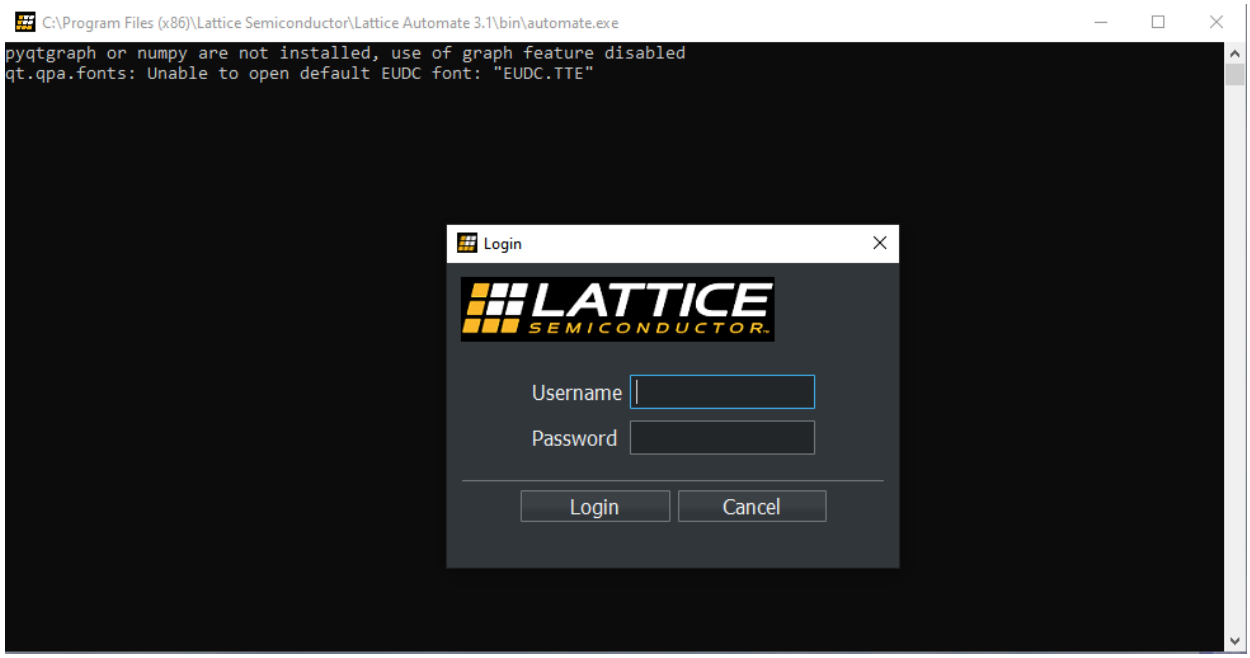

Figure 7.1. User Interface Application Login Screen

2. Enter the credentials and click **Login**:
   - **Username: lattice**
   - **Password: lattice**

   After successful login, the **Dashboard** tab opens.

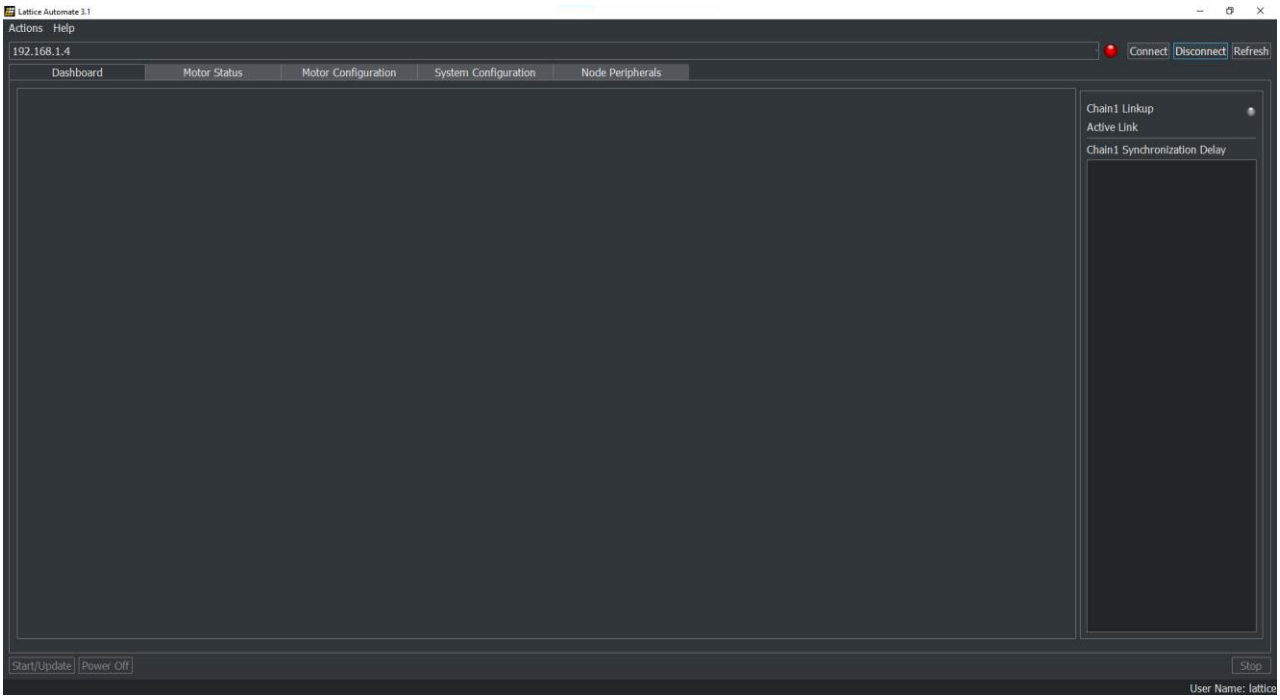

Figure 7.2. User Interface Application Dashboard Tab

7.2. Connect to the Main System

To connect to the main system:

1. Click on the **System Configuration** tab.

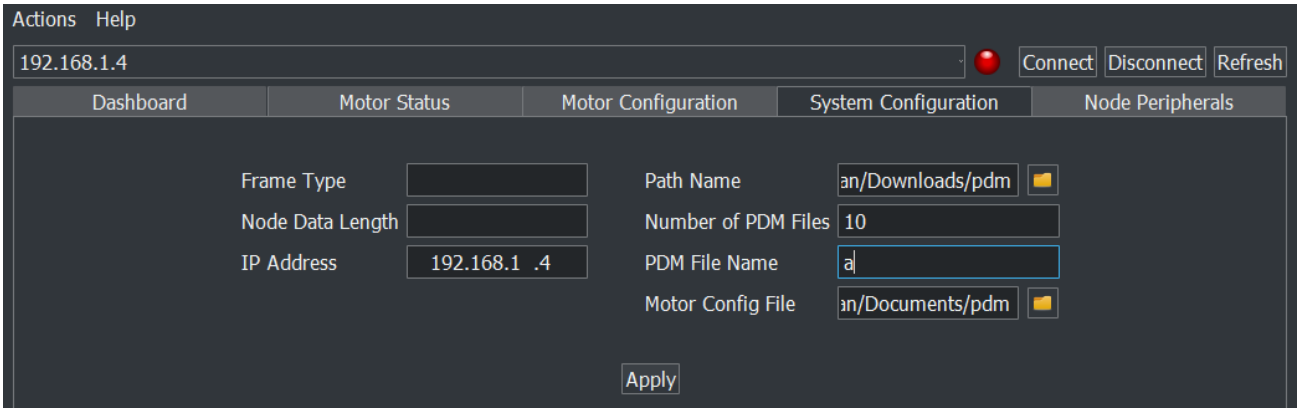

Figure 7.3. User Interface Application System Configuration: IP Address

2. Type the IP Address in the **IP Address** field.

   Note: Do not type leading zeros in the IP address, as shown in Figure 7.3.

3. Click **Apply**. The **Updated Successfully** pop-up displays. The updated IP address is visible on top of the IP Address bar.

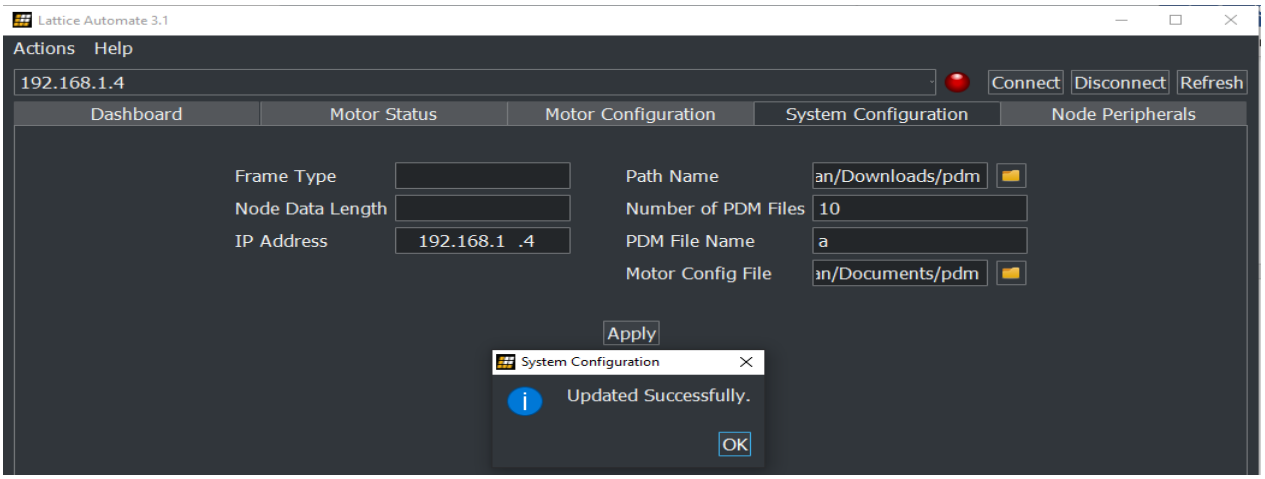

Figure 7.4. User Interface Application System Configuration: Updated Successfully Pop-up

4. Click **Connect**.
5. Once the IP Address is configured, the connection status is green and the IP Address bar disabled.

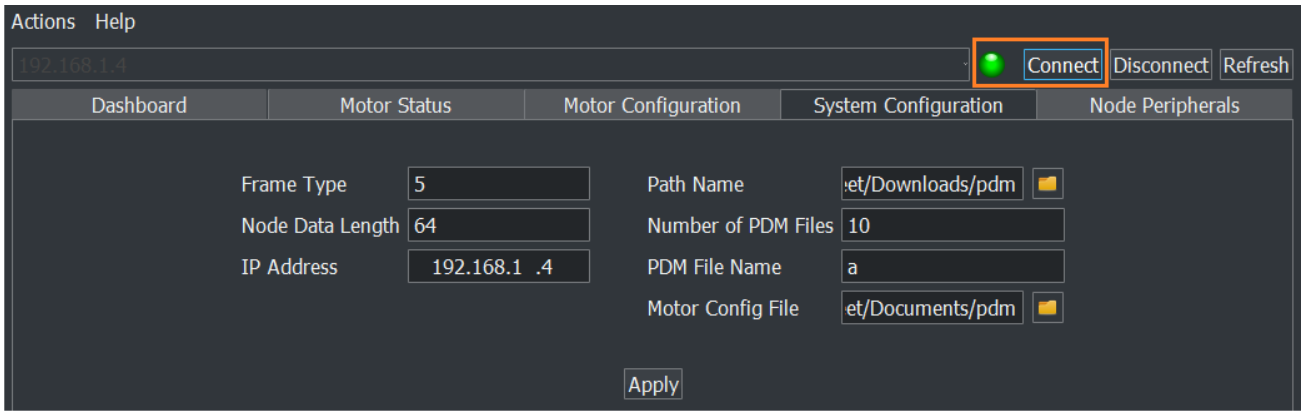

Figure 7.5. User Interface Application System Configuration: IP Address Configured Status

7.3. Dashboard Tab

Click the **Dashboard** tab as shown in Figure 7.6. This displays the following details:

- Chain 1 Linkup status
- Active link status
- Check the Chain1 – nodes synchronization delay

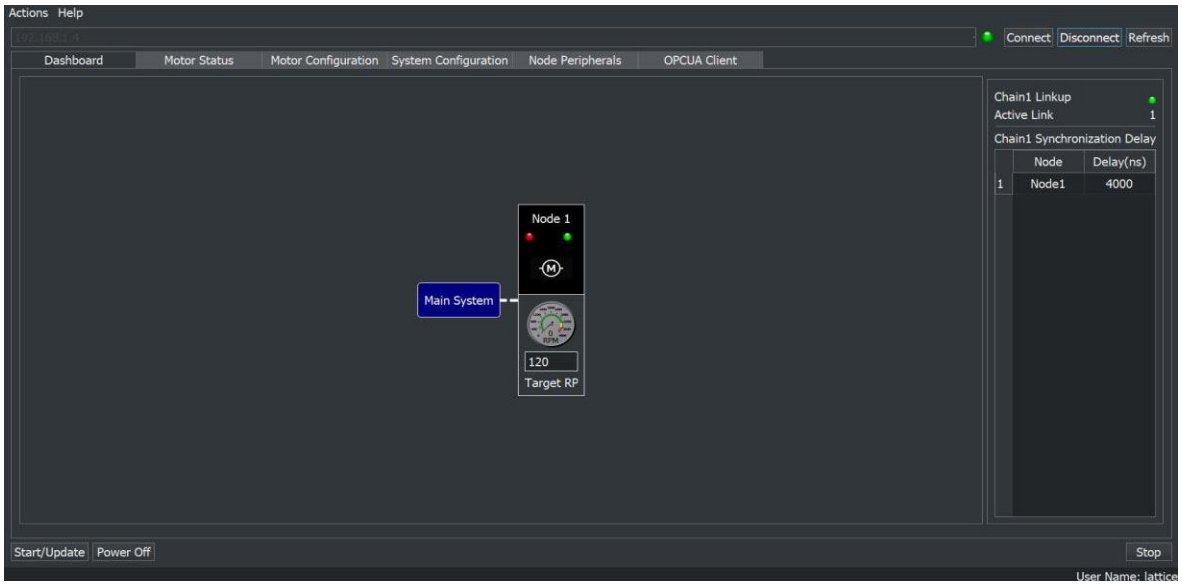

Figure 7.6. User Interface Application Dashboard: List Status

7.4. Refresh Button

Figure 7.7 shows the connected nodes. The Refresh button can be used to update nodes, if you are already connected and need to add or remove a node. To test and demonstrate its functionality, follow the steps below.

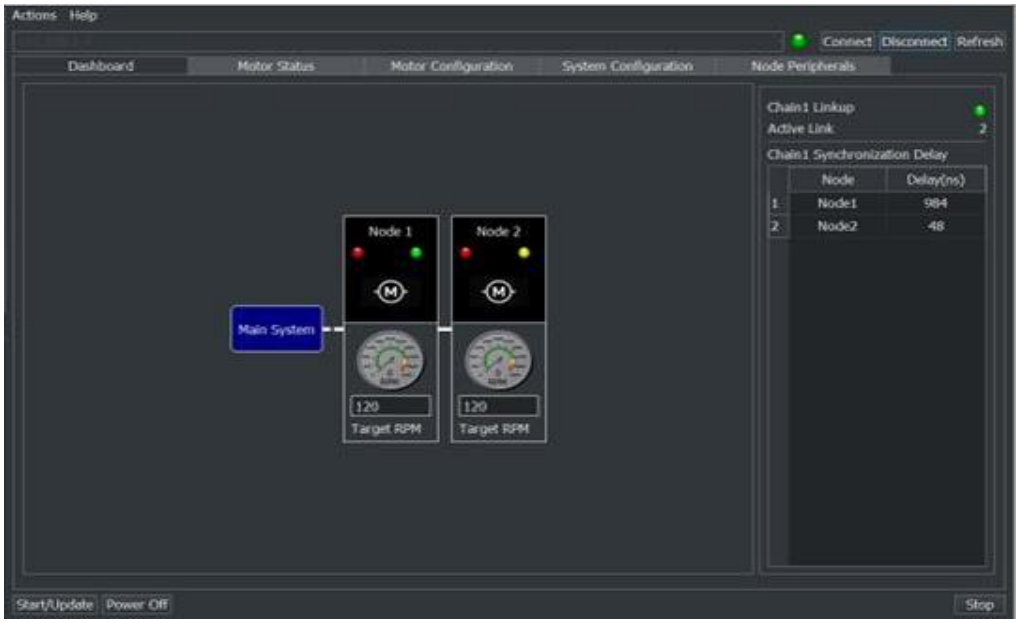

Figure 7.7. User Interface Application Dashboard: 2 Nodes Connected

To use the refresh button:

1. Disconnect the last node of the chain.
2. Wait for 10 seconds, then click **Refresh** button.
3. After Refresh completes, the dashboard reflects the actual connected node(s).

Figure 7.8. User Interface Application Dashboard: 1 Node Connected after Refresh

7.5. Motor Configurations

To configure the motor:

1. Click the **System Configuration** tab.
2. Click on the folder ( ) icon and select or create a location to save the motor config file. Save location can be anywhere in the system except C: drive.

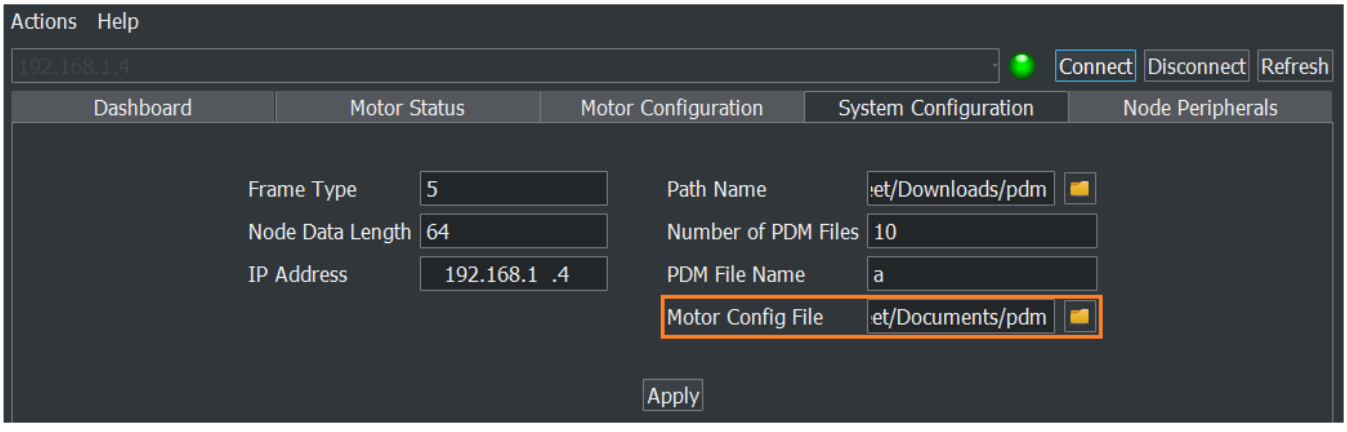

Figure 7.9. User Interface Application System Configuration: Motor Config File

3. Click **Apply**.
4. Click **OK** on the **Updated Successfully** pop-up.

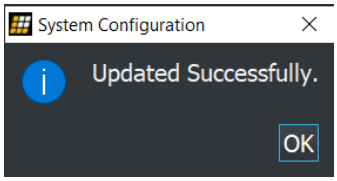

Figure 7.10. User Interface Application System Configuration: Updated Successfully Pop-up

5. Click on the **Motor Configuration** tab.
6. Select the number of nodes.

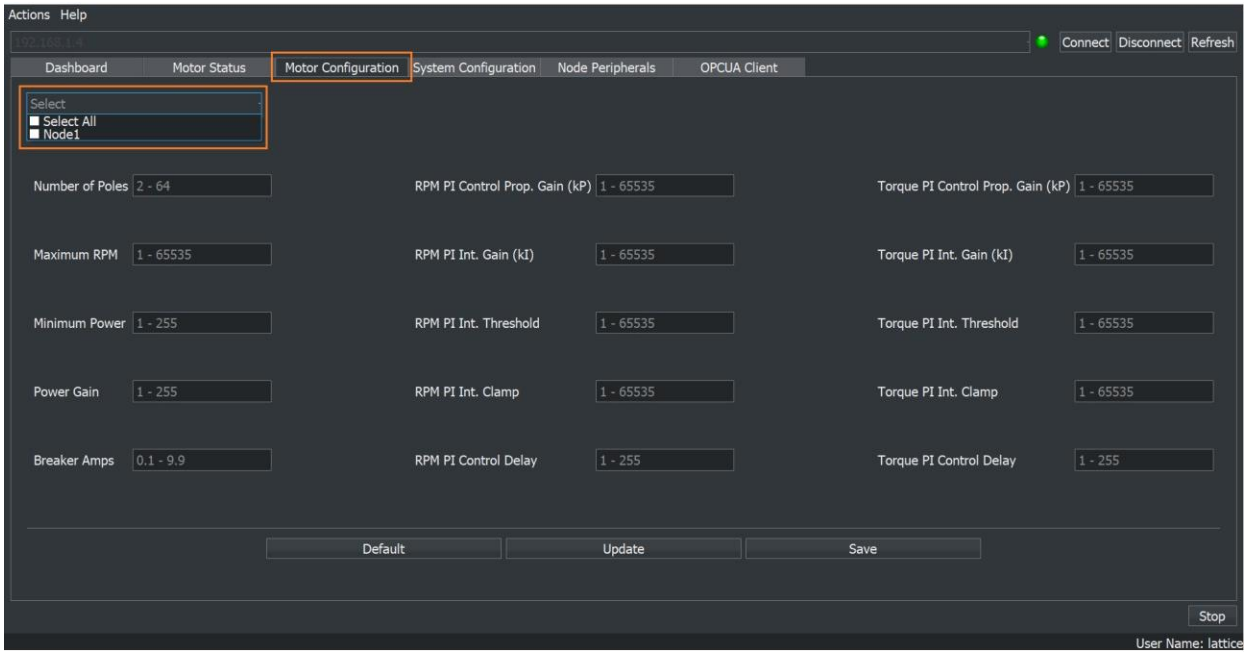

Figure 7.11. User Interface Application Motor Configuration: Node Selection

7. Either Select **All** to configure all the nodes at once or select one node at a time to configure it individually.

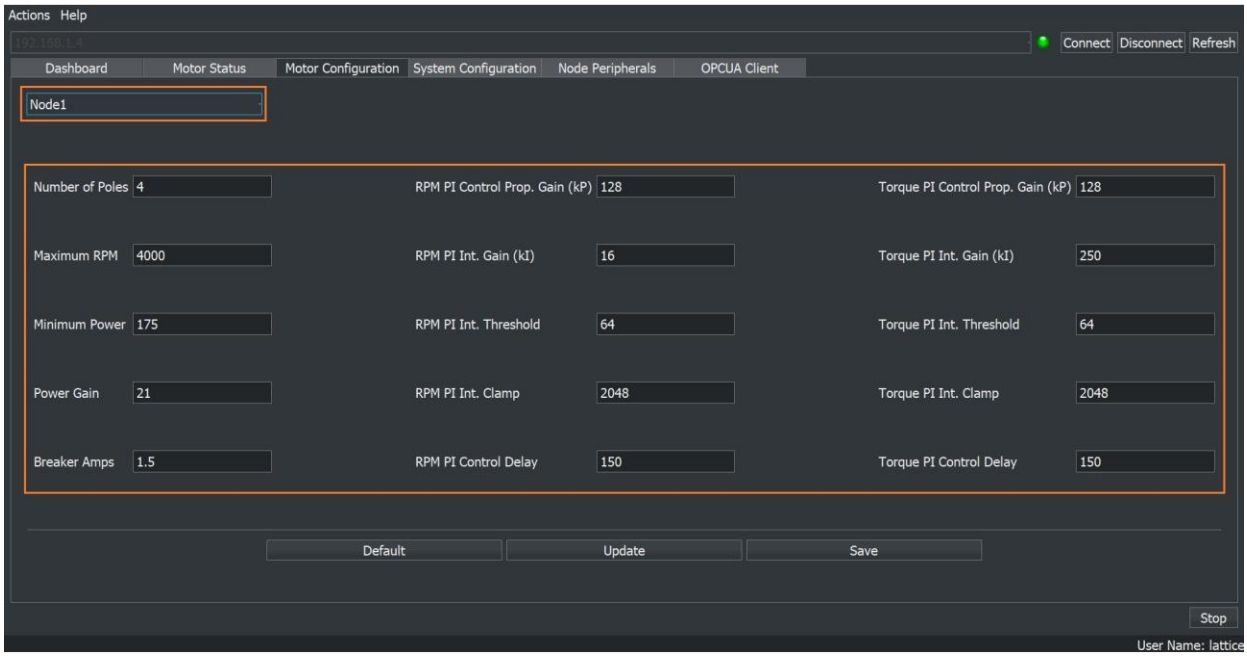

Figure 7.12. User Interface Application Motor Configuration: Node Selected

8. Change the Motor Configuration to the following values:
   - RPI PI Control Prop. Gain(kP): **90**
   - RPI PI Int. Gain(kI): **10**
   - RPI PI Control delay: **200**
   - Minimum Power: **190 - 200**
   - Power gain: **22 – 24**
   - Breaker Amps: **1.8 - 2.0**

   Note: These configuration values are for the GB-42 BLS 24 V, 5000 RPM Motors only. Modify them as needed for a different motor.

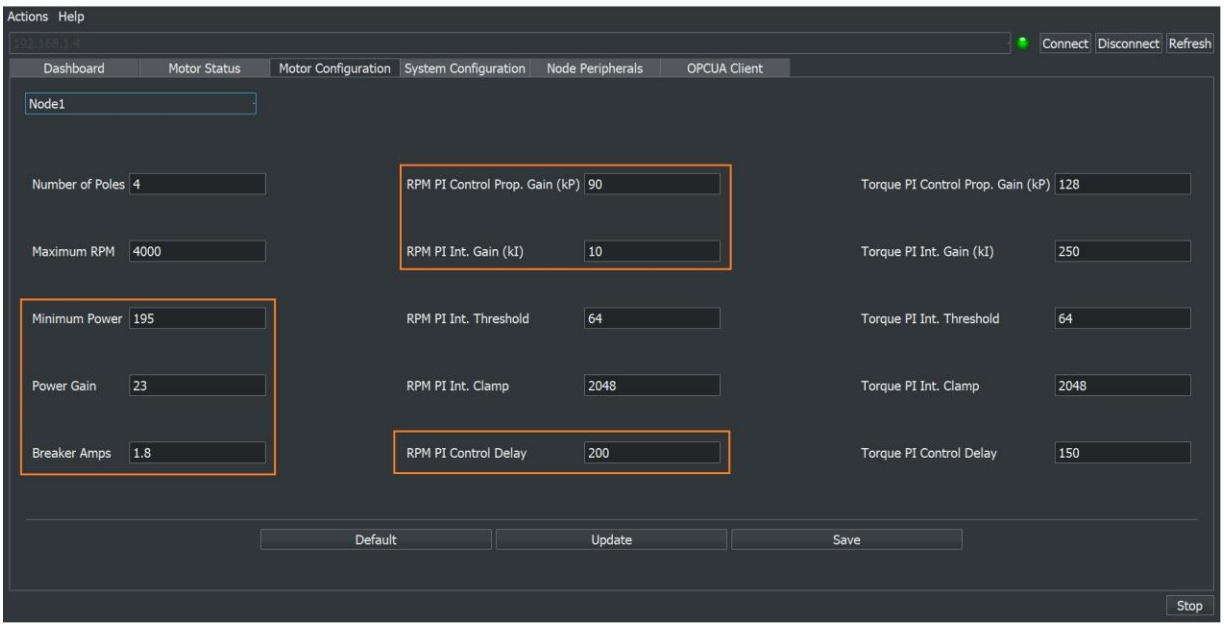

Figure 7.13. User Interface Application Motor Configuration: Configuration Value Changes

9. Click **Save**.
10. Click **Default**.
11. Click **Update**. A pop message appears.
12. Confirm the update action by clicking **Yes**. This displays a Authentication pop-up.

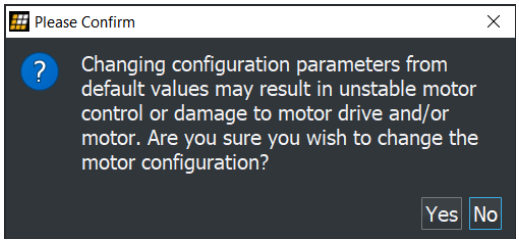

Figure 7.14. User Interface Application Motor Configuration: Warning Message

13. Enter the following credentials and click **Login**:
    - **Username: lattice**
    - **Password: lattice**

Figure 7.15. User Interface Application Motor Configuration: Authentication Pop-up

14. Click **OK** in the **Successfully updated the configuration of the selected node** pop-up.

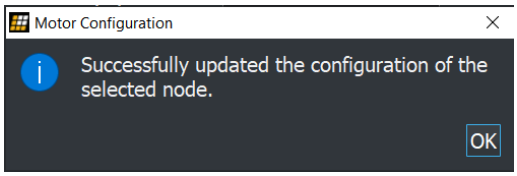

Figure 7.16. User Interface Application Motor Configuration: Update Configuration

15. Locate the motor_config.txt file. It is saved in the location that user selected in step 2.

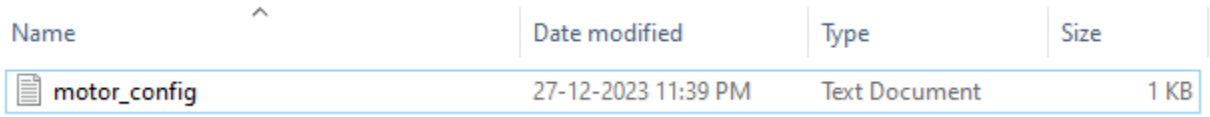

Figure 7.17. Motor Config File Saved in Host PC

16. Open the *motor_config.txt* file and verify that the saved motor configurations match the values that user entered in step 7.

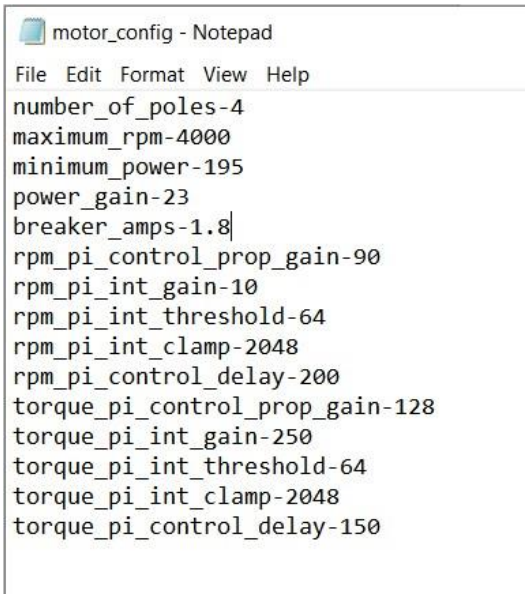

Figure 7.18. Motor Config.txt file in Host PC

7.6. Target RPM, Voltage, and Drive Status Using the Dashboard Tab

To target RPM, voltage, and drive status:

1. Click the **Dashboard** tab.
2. Enter the **Target RPM** Value as **120**.

   Note: You can gradually increase the RPM up to 2000 RPM.

   The following RPM increments are recommended: 120, 400, 800, 1400, 1600, 1800, and 2000.

   Note: The target RPM value should not exceed the Maximum RPM value entered in the **Motor Configuration** tab.

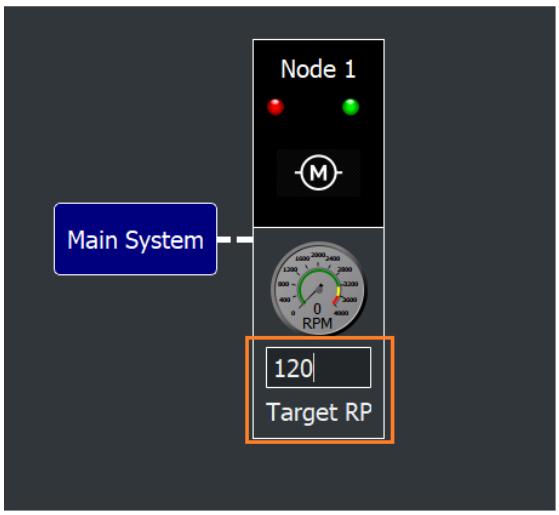

Figure 7.19. User Interface Application Dashboard: Set Target RPM

3. Click **Start/Update**.

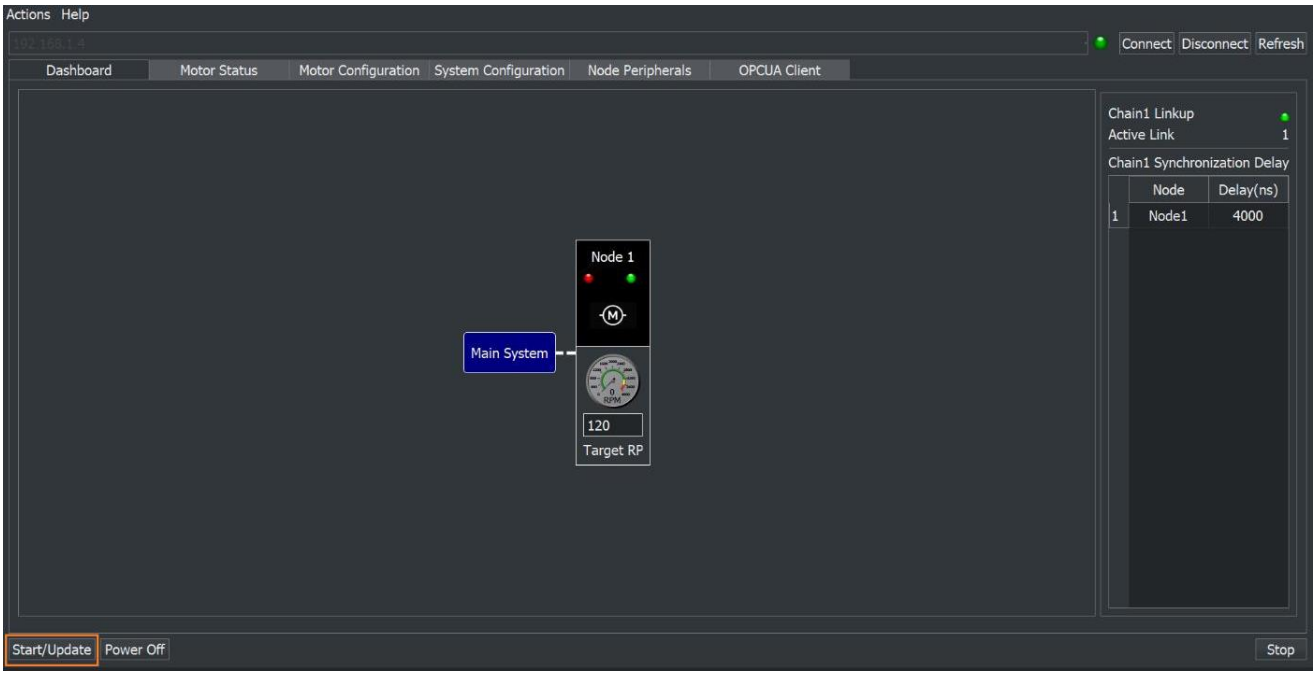

Figure 7.20. User Interface Application Dashboard: Start/Update

4. After the RPM Lock is achieved, the Node LED is green.

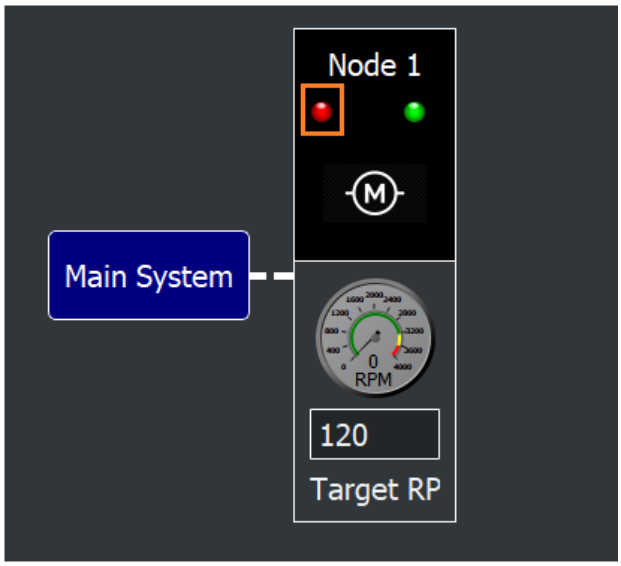

Figure 7.21. User Interface Application Dashboard: RPM Lock Achieved Status

5. Click the **Motor Status** tab to check the RPM, Voltage, and Current values.
6. Select the node that you want to check.

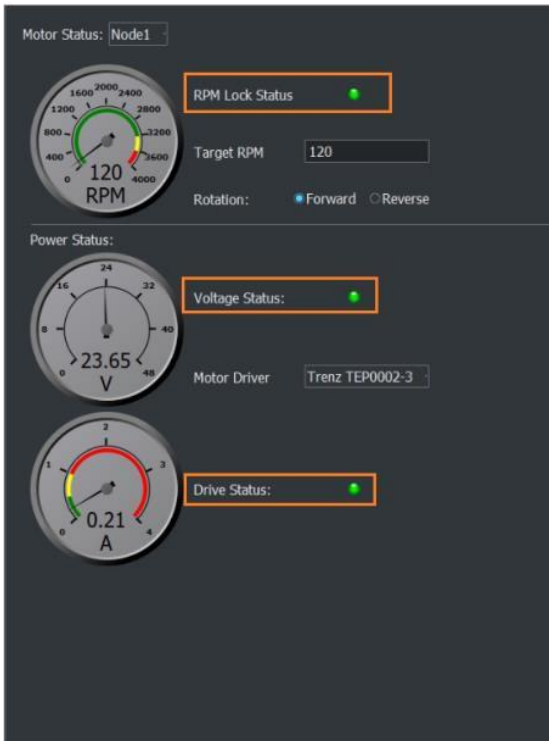

Figure 7.22. User Interface Application Motor Status: RPM Lock, Voltage, and Drive Status

7. To stop the motor, click **Stop** on either the **Motor Status** tab or the **Dashboard** tab.
8. Click **Power Off**.

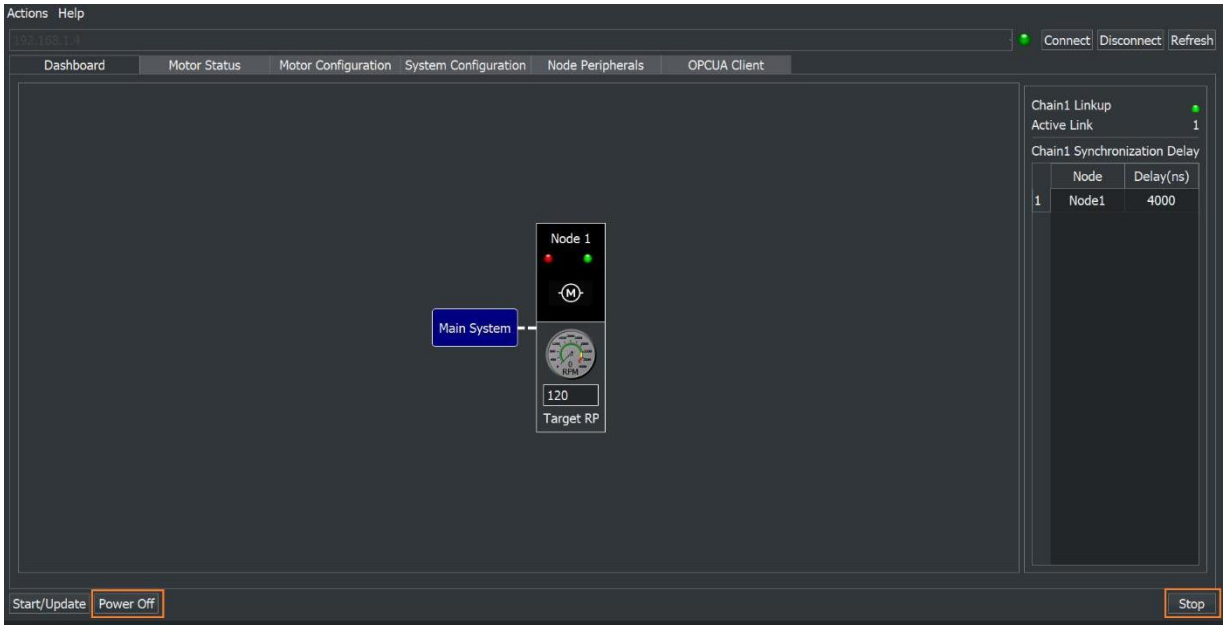

Figure 7.23. User Interface Application Dashboard: Stop and Power Off Buttons

Note: The **Power Off** button disengages the motor power supply completely. This helps to prevent the motor from continuously drawing current from the power supply.

7.7. Motor Status Test

To check the motor status:

1. Click the **Motor Status** tab.
2. Set the Target RPM to **120**.

   Notes:
   - You can gradually increase the RPM up to 2000 RPM.
   - The following RPM increments are recommended: 120, 400, 800, 1400, 1600, 1800, and 2000.
   - Target RPM should not exceed the Maximum RPM value entered in the Motor Configuration tab.
3. Click **Start/Update**.

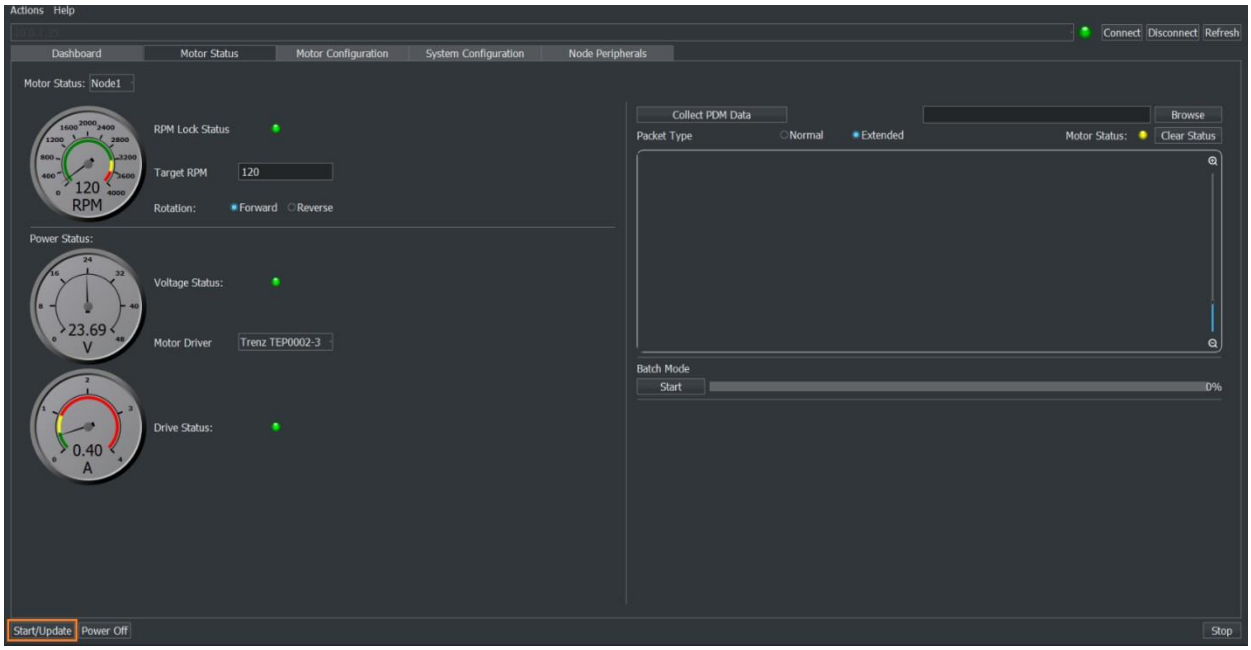

Figure 7.24. User Interface Application Motor Status: Start/Update

4. Once the actual RPM reaches the target RPM, the meter gauge displays 120 and the RPM Lock button becomes green.
5. To update the RPM speed, set the **Target RPM** to 500 RPM.
6. Once the actual RPM reaches the target RPM, the meter gauge displays 500 and the RPM Lock button becomes green.

Figure 7.25. User Interface Application Motor Status: RPM Lock Status When Target RPM is Achieved

7. To stop the motor, click **Stop**.
8. Click **Power Off** to stop the current withdraw.

7.8. Forward/Reverse Rotation Test

To test the forward/reverse rotation:

1. Click the **Motor Status** tab.
2. Select the **Forward** option for the rotation.
3. Enter the **Target RPM** anywhere between 120 RPM to 2000 RPM.
4. Click **Start/Update**. The Motor starts rotating in a clockwise direction.

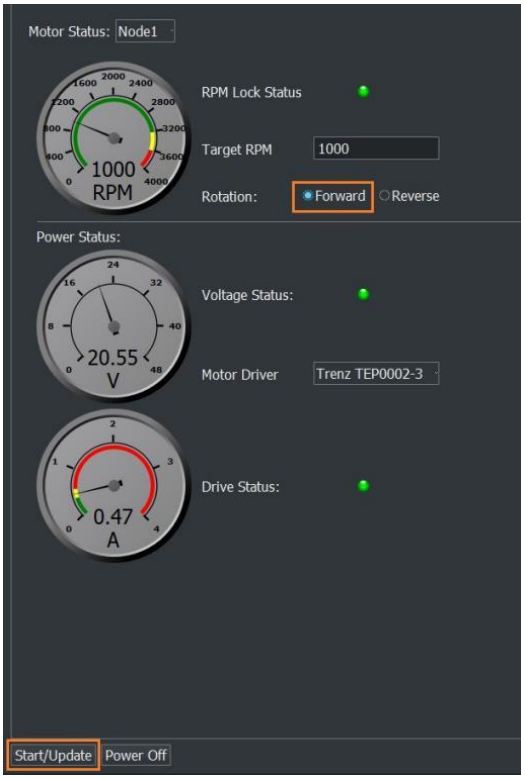

Figure 7.26. User Interface Application Motor Status – Forward Rotation Status

5. Wait until the RPM Lock status becomes green.
6. Select the **Reverse** option to rotate the motor in the opposite direction.

Figure 7.27. User Interface Application Motor Status – Reverse Rotation Status

7. Click **Start/Update**. The Motor changes the direction to rotate in a counterclockwise direction.
8. Wait until the RPM Lock status becomes green.
9. To stop the motor, click **Stop**, and then **Power Off**.

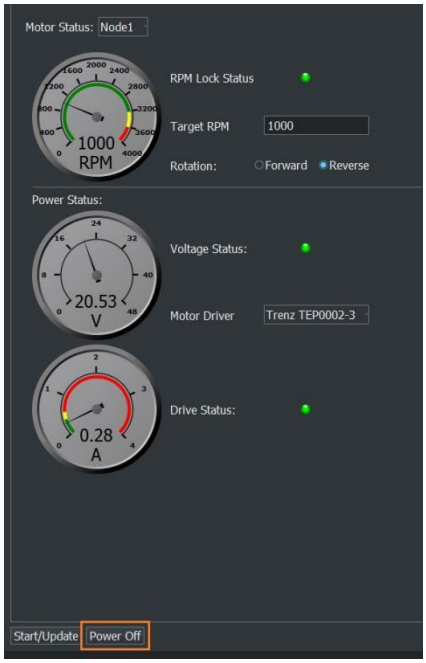

Figure 7.28. User Interface Application Motor Status – Motor Power Off

7.9. PDM Data Collection

7.9.1. Collect PDM Data

To collect PDM data:

1. Click the **System Configuration** tab.
2. Click on the folder ( ) icon and select or create a location to save the PDM images. Save location can be anywhere in the system except C: drive.

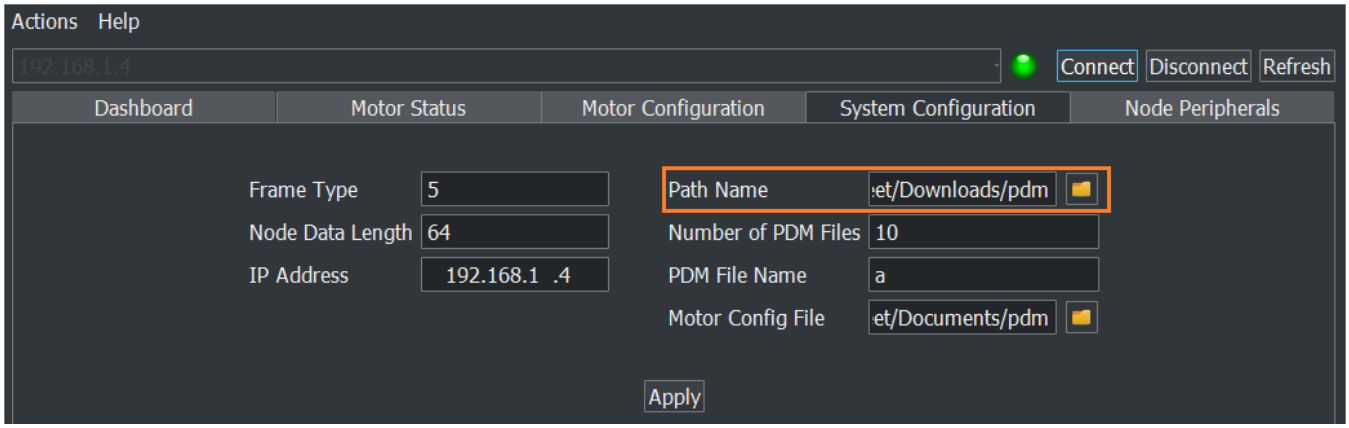

Figure 7.29. User Interface Application PDM Data: Path Name

3. Click **Apply**. Click **OK** on the **Updated Successfully** pop-up.

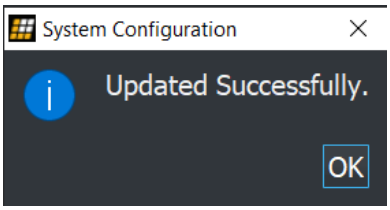

Figure 7.30. User Interface Application PDM Data: Updated Successfully Pop-up

4. Click the **Motor Status** tab.
5. Set the **Target RPM** initially to 120, then gradually increase the RPM as described in Motor Status Test section.
6. Click **Start/Update**. Wait for the RPM lock.
7. Click **Collect PDM Data**. Wait for the PDM data process to complete. It may take a few minutes.

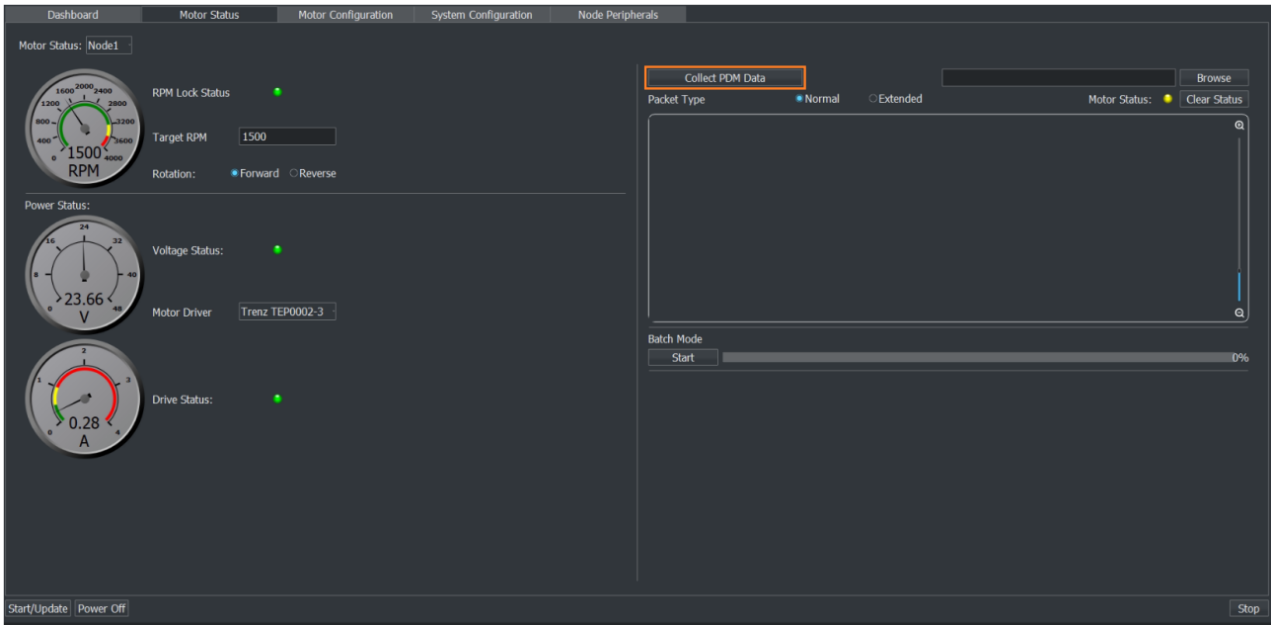

Figure 7.31. User Interface Application PDM Data: Collect PDM Data

8. Choose Normal or Extended for the Packet type. Both Normal mode and Extended mode appear the same from a user's perspective, but Extended mode is more efficient if there are many nodes in the chain.

   Note: See Details about PDM Data Collection Process section for a description of Normal mode and Extended mode.

9. **Collecting PDM Data from Node** and **Analyzing PDM Data from Node** messages are displayed while the image is captured.

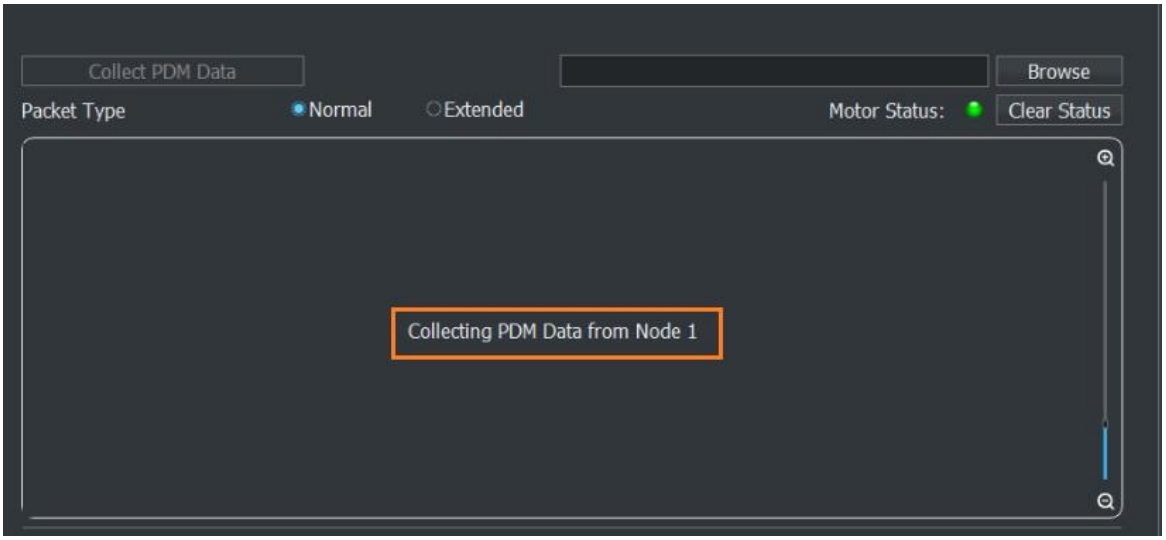

Figure 7.32. User Interface Application PDM Data: Collecting PDM Data from Node

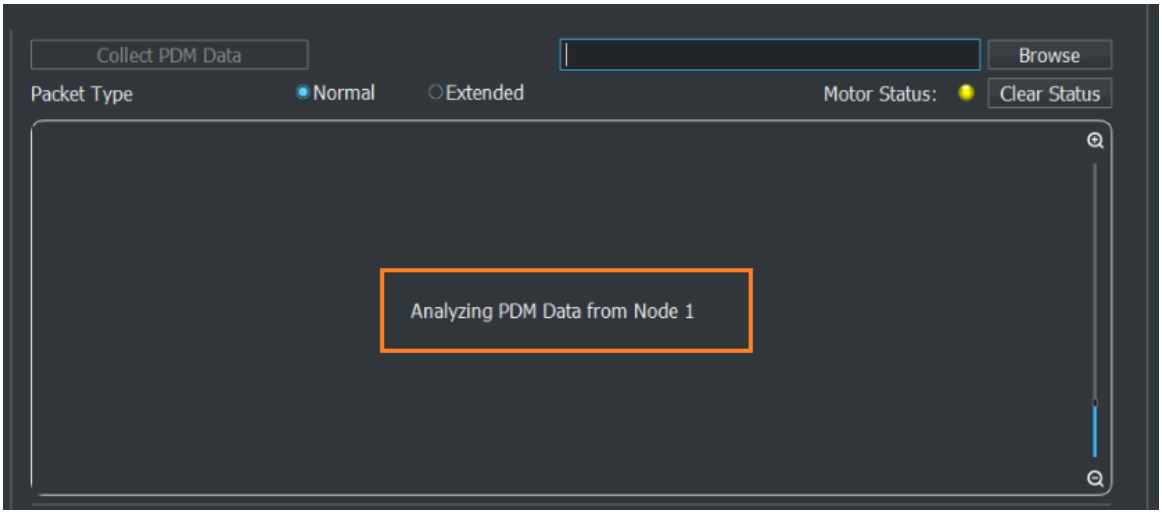

Figure 7.33. User Interface Application PDM Data: Analyzing PDM Data from Node

10. Once the Collect PDM Data process is complete, the PDM image appears on the screen.

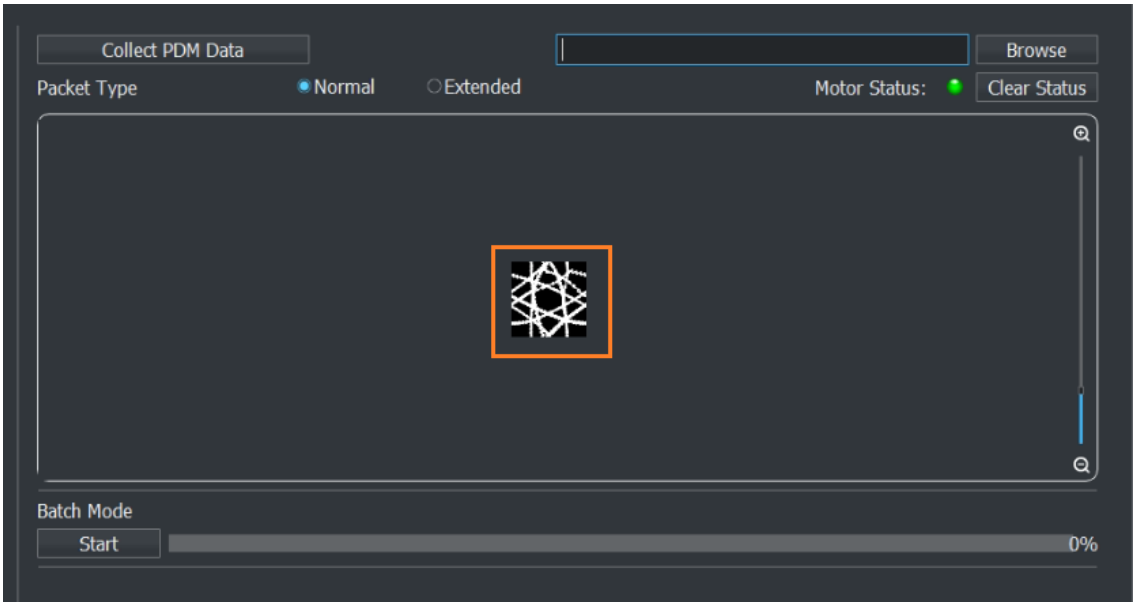

Figure 7.34. User Interface Application PDM Data: PDM Image

11. To remove the image and clear motor status, click **Clear Status**. Motor Status becomes yellow.

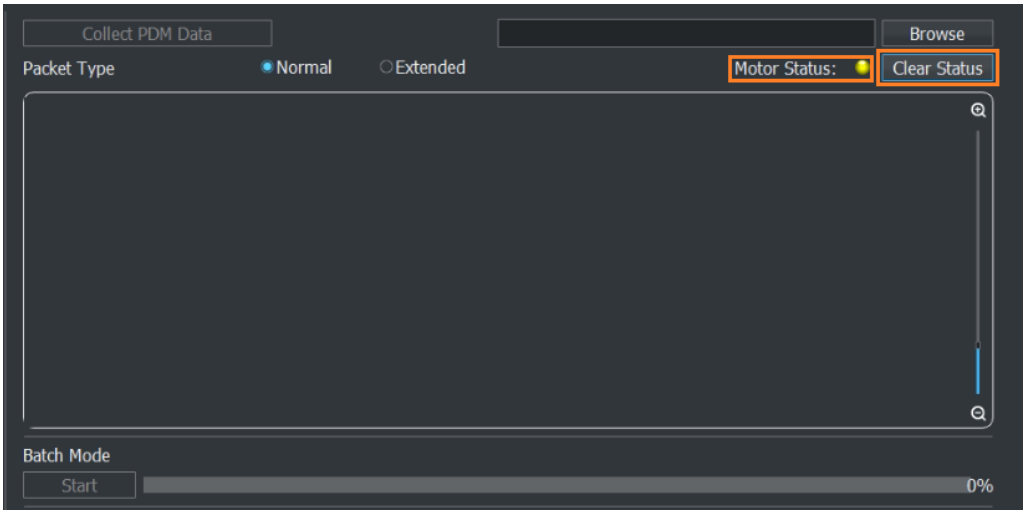

Figure 7.35. User Interface Application PDM Data: Clear Status and Motor Status

12. To fetch the previous images, click **Browse**.
13. To zoom in or zoom out the PDM image, move the cursor up or down on the zoom slider.

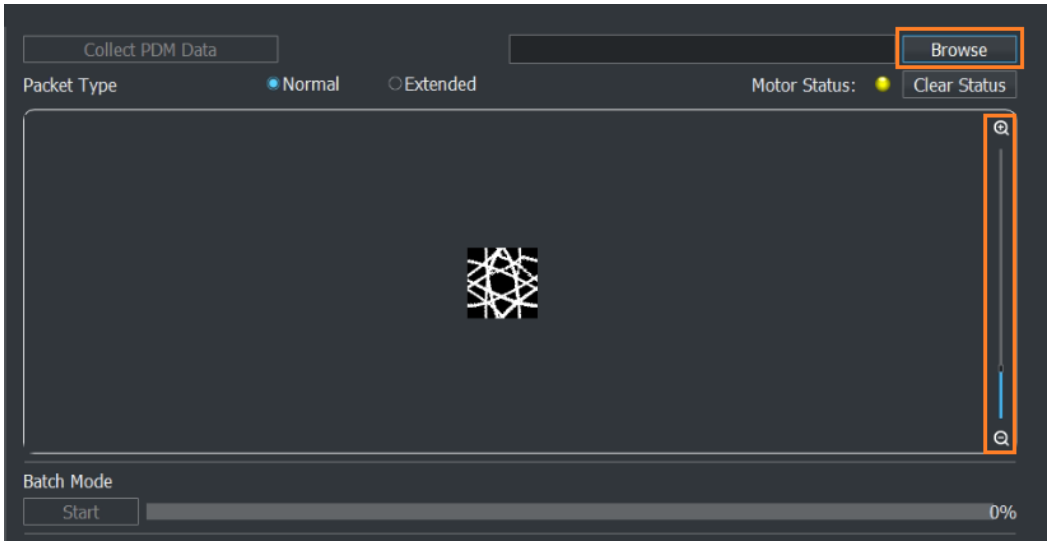

Figure 7.36. User Interface Application PDM Data: Browse Button and Zoom Slider

7.9.2. Batch Mode

Batch mode allows you to collect multiple PDM images, one after another. First, you must choose how many PDM files you want to collect. After that, the steps are like the standard Collect PDM Data process.

1. Click the **System Configuration** tab and enter the number of PDM files that you want to collect. The system supports a range of 1-20 PDM files.
2. If you have not yet selected a location to save the images, click the folder image next to Path Name and browse for a location, as described in the Collect PDM Data section, Step 2.

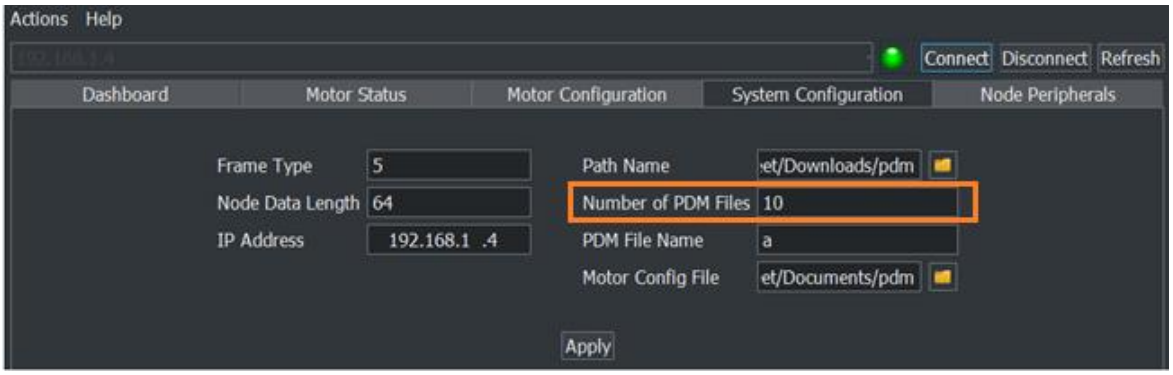

Figure 7.37. User Interface Application PDM Batch Mode: Number of PDM files

3. Click **Apply**.
4. Click **OK** on the **Updated Successfully** pop-up.

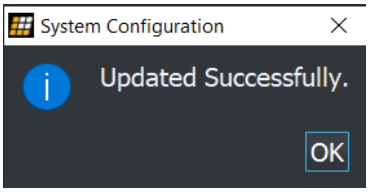

Figure 7.38. User Interface Application PDM Batch Mode: Updated Successfully Pop-up

5. Click the **Motor Status** tab.
6. Click on the **Start** button under the words Batch Mode.

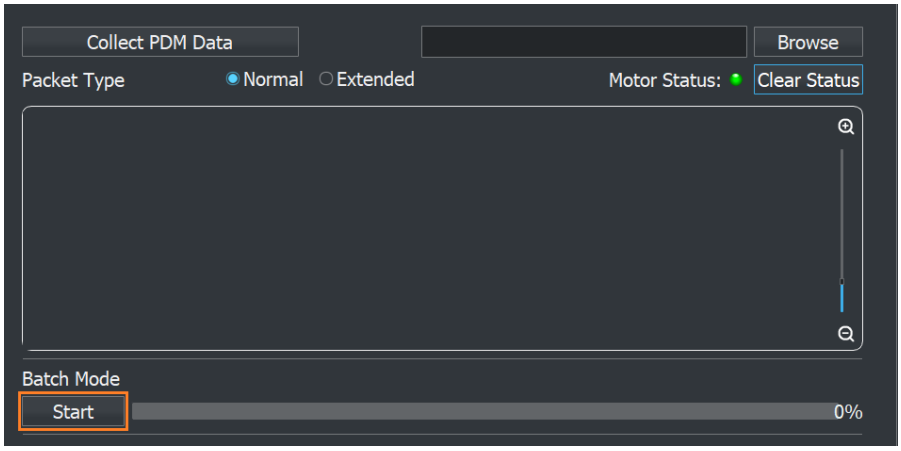

Figure 7.39. User Interface Application PDM Batch Mode: Start

7. Wait for some time to collect multiple images until the status bar reaches 100%. This takes a few minutes. The more images that you are collecting, the longer it takes.

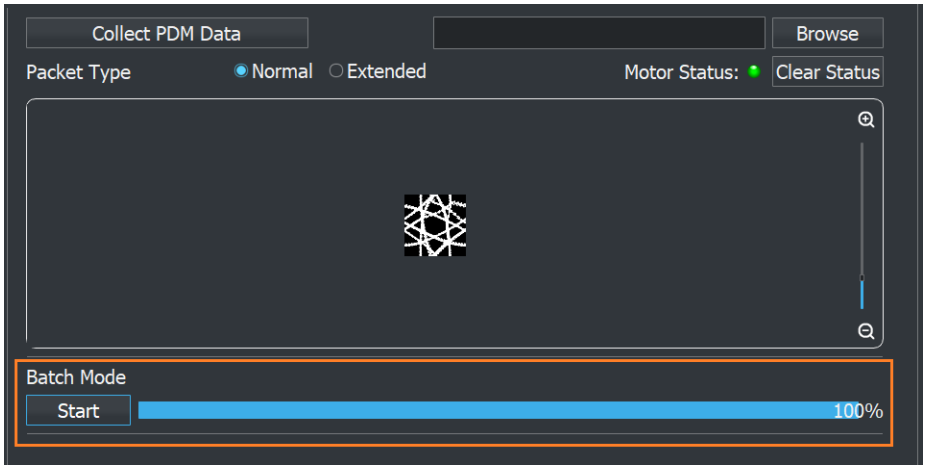

Figure 7.40. User Interface Application PDM Batch Mode: Batch Mode 100% Status

8. If you do not want to capture all the images, click **Stop** to stop the image collecting.

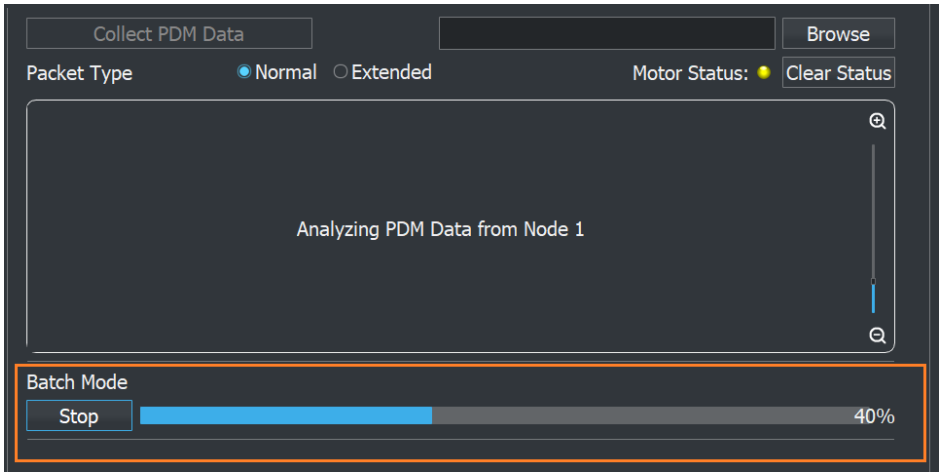

Figure 7.41. User Interface Application PDM Batch Mode: Collecting Multiple Images

9. Click **Stop** to stop the motor.
10. Click **Power Off**.

7.9.3. Details about PDM Data Collection Process

When you do a PDM data request in the user interface, the request is sent over OPCUA and a PDM Data command is sent through Etherconnect to the main system's RISC-V CPU. The main system first confirms that the motor is running, RPM is locked, and calibration is done.

The main system creates a frame and sends a packet through the control/PHY to the node system.

When the node system receives the frame, the node system's RISC-V CPU uses the PDM Data Collector to fill the FIFO DMA with data collected from the motor. When the FIFO DMA is full, an Ethercontrol packet is returned to the main system, using an interrupt to alert the main system that data is ready.

The ethernet packet continues making round trips. If the packet ID matches the request ID, then it contains valid data and is processed. Otherwise, the packet is discarded.

In both normal and extended mode, PDM data is collected for one node at a time. The node to collect data on must be selected in the user interface. In Extended mode, the number of transactions depends on the active nodes connected in the chain. As the number of active nodes increases, the number of transactions decreases, hence the time to collect PDM data also decreases.

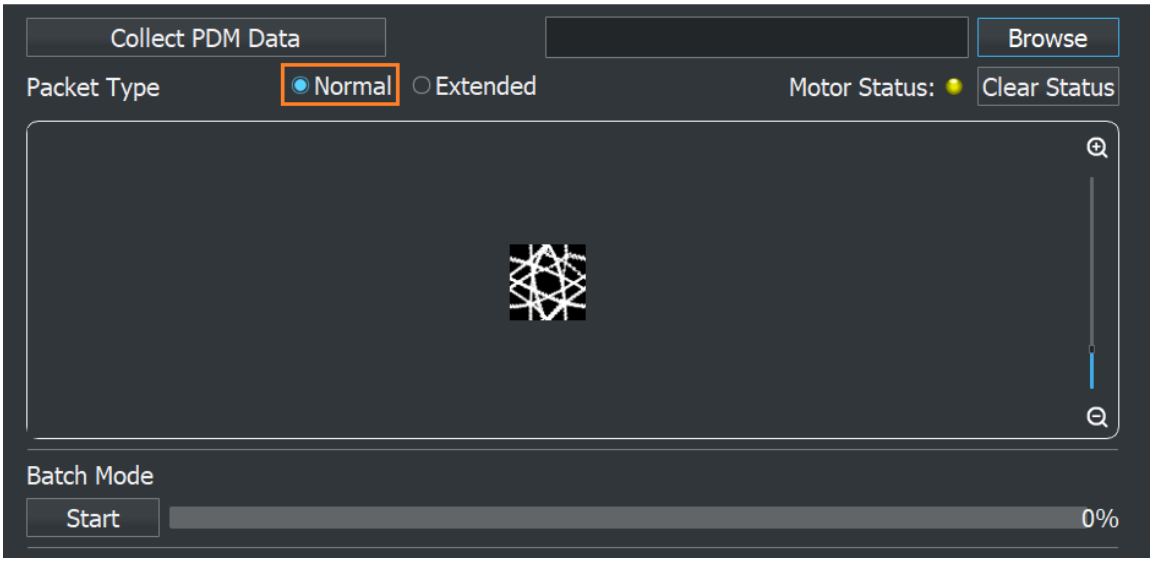

Figure 7.42. User Interface Application PDM Data: Normal Mode

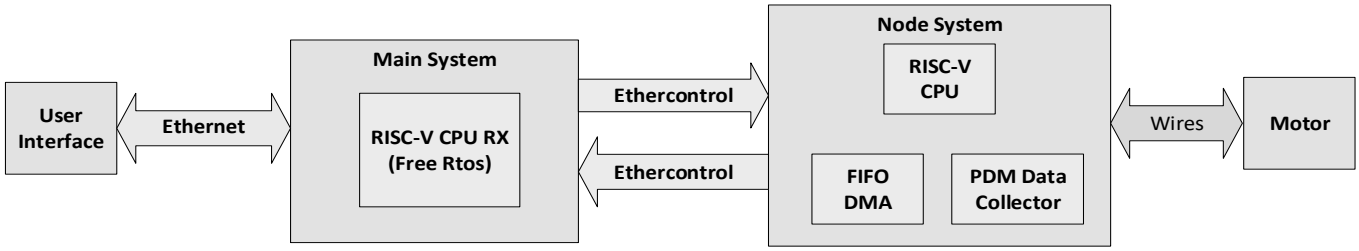

Figure 7.43. User Interface Application PDM Data: Components Involved in PDM Data Collection Process

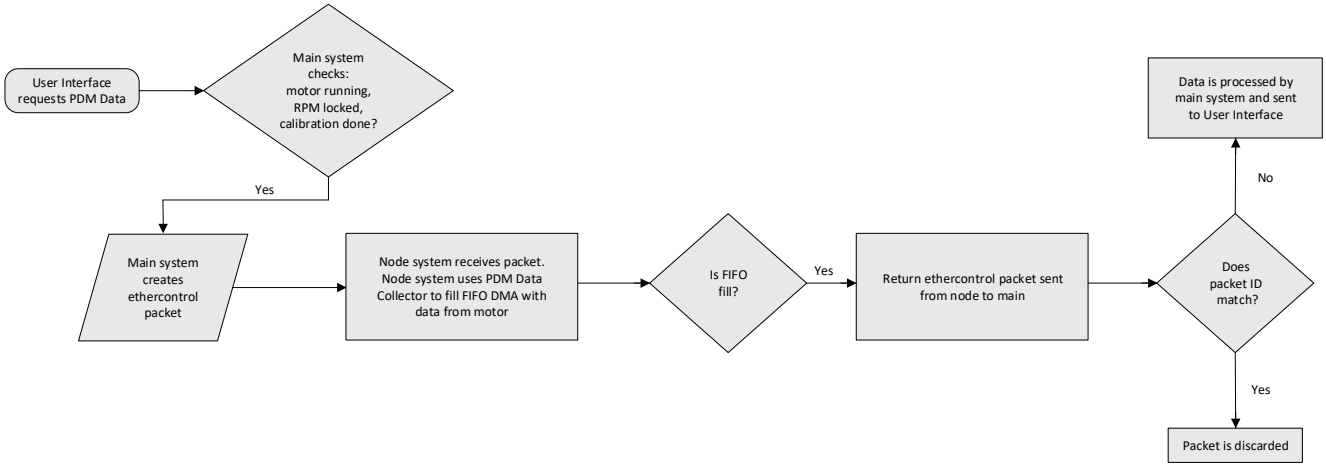

Figure 7.44. User Interface Application PDM Data: Process Flow for PDM Data Collection

7.9.4. Collecting PDM Data from a Simulated Faulty Motor Using the Switches

7.9.4.1. Introduction and Purpose of the Switches

Normally, all the motors in the Automate demo should be working well and not failing. To show the benefits of the PDM data collection, the Automate demo setup includes three switches that control the power resistors connected to the motor. By toggling these switches, the user can disable the resistors and simulate what the motor would look like if it were failing or faulty.

Each switch is for one phase of the motor. When the switches are tilted towards the power resistors of the motor, this means the power resistors are included in the circuit and the motor should behave normally. If any single switch is tilted away from the power resistors of the motor, the resistor is disconnected, and the motor exhibits faulty behavior. This faulty behavior may not be observable to a human, which is where the power of PDM data collection comes in.

PDM data collection collects the current signature of the motor. Inside the PDM module, there is a convolutional neural network that has been trained on the current signatures of good and faulty motors. This setup can be extended to collect data at regular intervals and send an alert if one of the motors starts to show signs of failing. It is easy to see how early detection of a problem can save time and expense in an industrial automation scenario.

Safety Note: Only one resistor should be disabled at a time. In other words, only tilt one switch away from the motor at a time. If two or three switches are tilted away at the same time, the motor does not rotate, and the motor can be damaged.

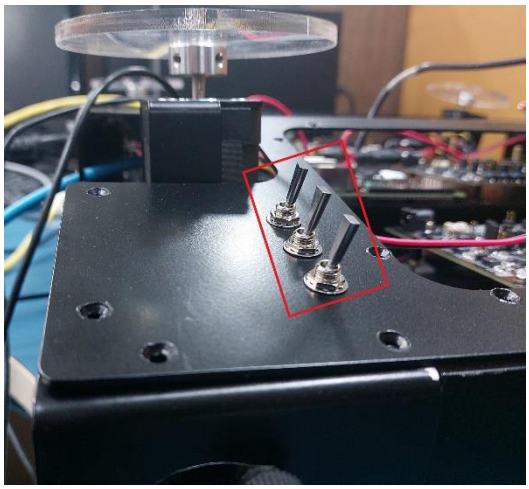

Figure 7.45. Three Resistor Switches Tilted Towards the Resistors (Meaning All Three Resistors are Enabled and the Motor is “good”)

7.9.4.2. Example Workflow to Collect Faulty PDM Data with the Switches

To collect faulty PDM data with switches:

1. Set the target initially to 120 RPM and start the motor.
2. Gradually increase the RPM as described in Motor Status Test section.
3. Continue increasing the RPM up to 1500.
4. Disable any one switch by tilting it away from the power resistors of the motor.
5. Follow the instructions in the Collect PDM Data section to collect PDM data in either Normal or Extended mode.
6. When the PDM image appears, the lines in the image should look different than the image collected when all the power resistors are enabled and the Motor Status icon should be red.

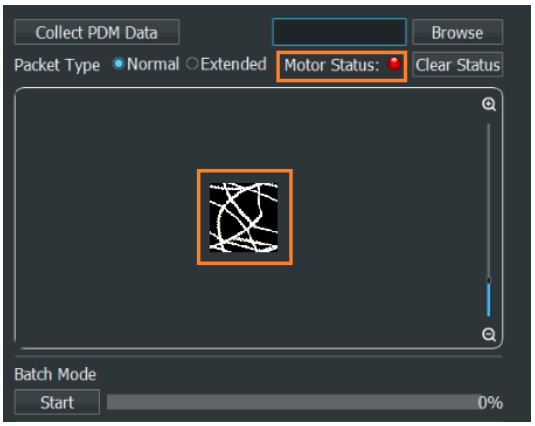

Figure 7.46. PDM Data collection: Example PDM Image of a Faulty Motor

7. Stop the motor and power it off.
8. Tilt the switch back towards the motor to re-enable the power resistor.

7.10. Testing Node Peripherals

To test the node peripherals:

1. Connect the I²C/SPI Host Adapter between the Lattice Certus NX board and Host PC-1.
2. For I²C connection, connect the SDA to pin 1 of JP3 and SCK pin to pin 2 of JP2 on the Lattice Certus NX board.
   a. The arrow printed on the board indicates pin 1, so the other pin in the jumper pair is pin 2.

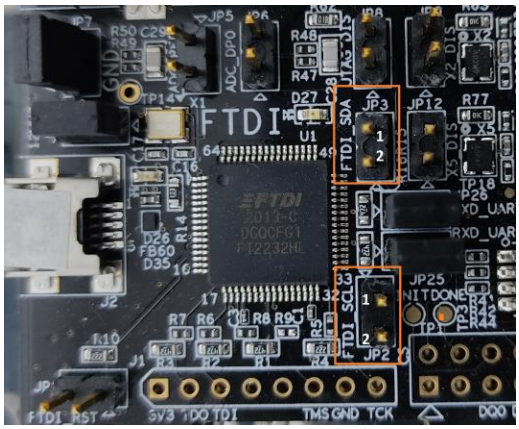

Figure 7.47. Certus NX Board – I²C Connection

3. For SPI Connection, connect the MISO, MOSI, SCK, and CS into J8 PMOD2 pins in the upper slot of the Certus NX board. Starts from the arrow:
   a. Pin 1. MISO (PMOD 1)
   b. Pin 2. MOSI (PMOD2)
   c. Pin 3. SCLK (PMOD 3)
   d. Pin 4. CS/SS (PMOD 4)
   e. Connect the GND pin to the GND port of the Certus NX Board as shown in Figure 7.48.

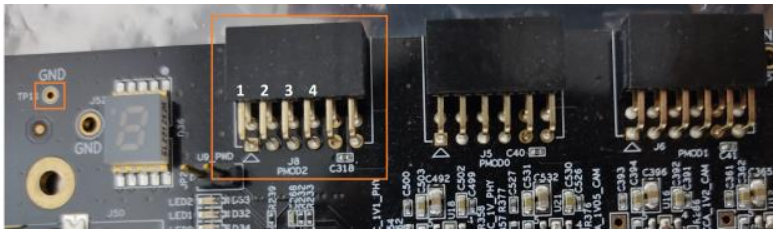

Figure 7.48. Certus NX Board - SPI Connection

7.10.1. Making the Connection in Total Phase Control Center

To make the connection in total phase control center:

1. Open **Total Phase Control Center**.
2. Click on the **Adapter** tab, and then Click **Connect**.

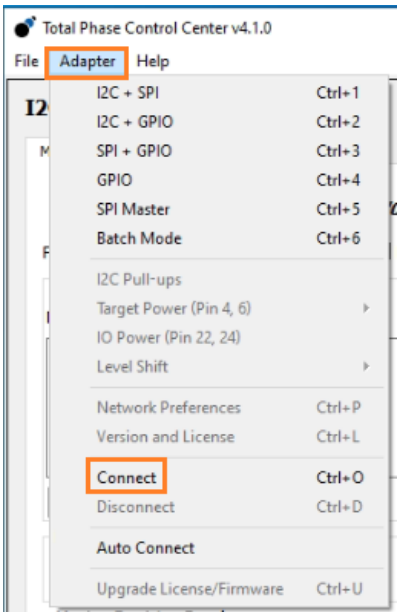

Figure 7.49. Total Phase - Adapter Setting (I²C)

3. Configure the adapter page opens. Select a Mode: **I²C-SPI**.
4. Click **OK**.

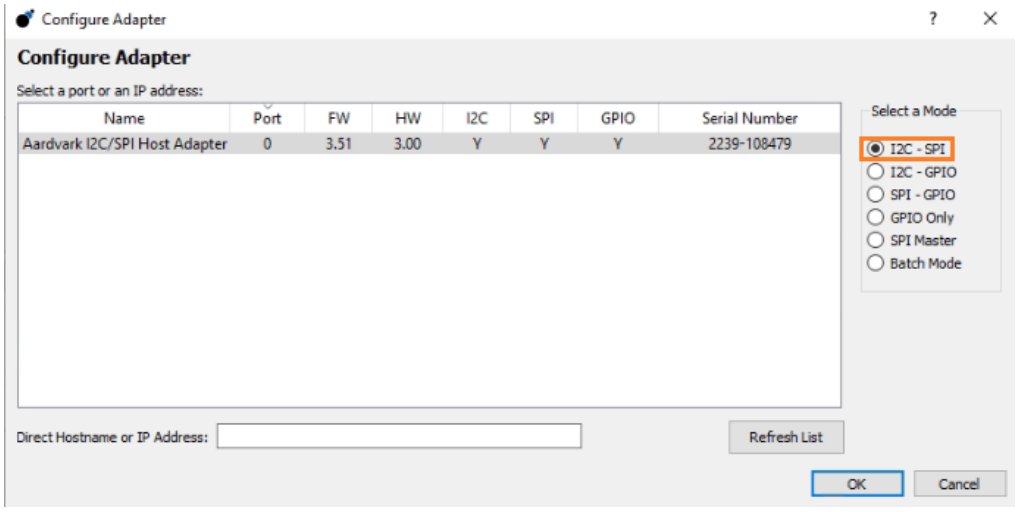

Figure 7.50. Total Phase - I²C-SPI Selection

5. Select the **Slave** tab on the **I²C Control** page.
6. Click **Enable**.

   Note: Make sure that the physical connection between the Aardvark and PMOD of the node board is correct.

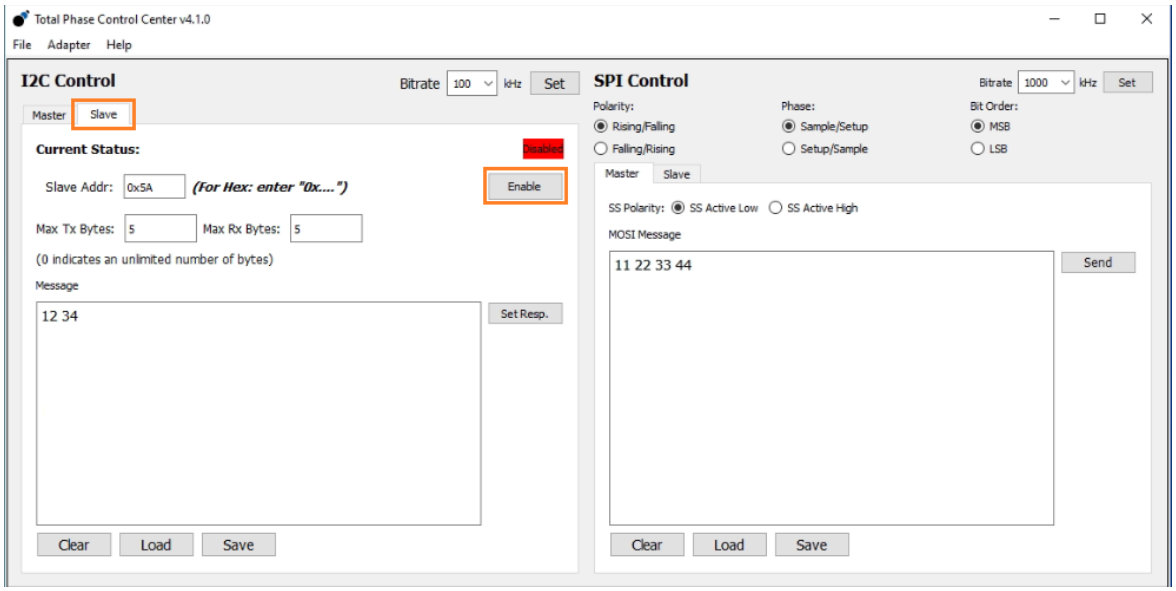

Figure 7.51. Total Phase - I²C Control (I²C)

7. Select the **Slave** tab on **SPI Control** page.
8. Click **Enable**.

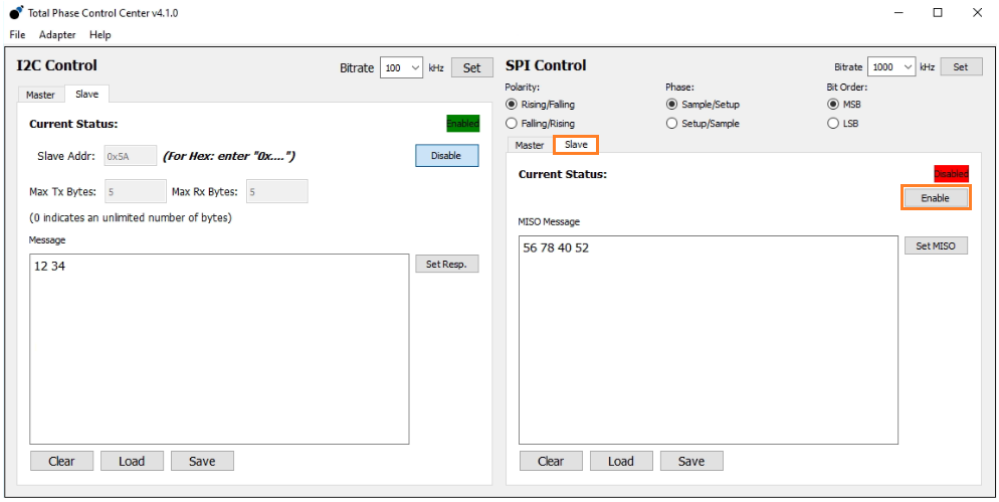

Figure 7.52. Total Phase - SPI Control: SPI Enable

9. Once Enabled both I²C and SPI control, the display looks like Figure 7.53 and the connection status shows in the transaction log.

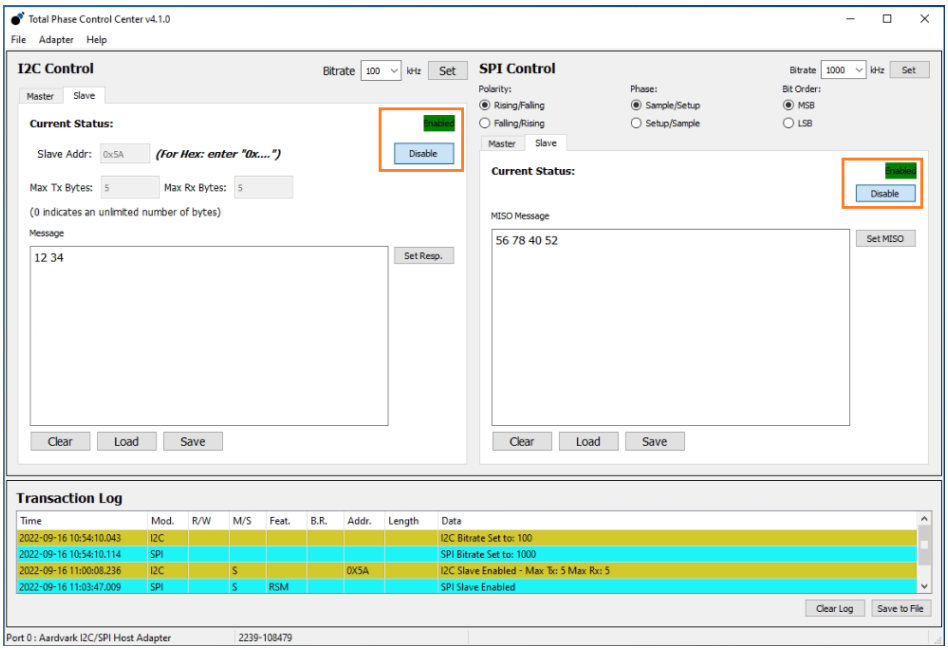

Figure 7.53. Total Phase - I2C and SPI Control Enabled

7.10.2. I²C

For write operation:

1. In the Automate 3.1 user interface, click the **Node Peripherals** tab.
2. Select the **Node** which is connected to the Aardvark I²C/SPI Host Adapter.
3. Select the protocol: **I²C**
4. Select the operation: **Write**
5. Enter the Slave Address 2 bit: **5A**
6. Enter the Data 2 bit: 25 (User can write any data from 0x00 to 0xFF.)
7. Click **Write**.

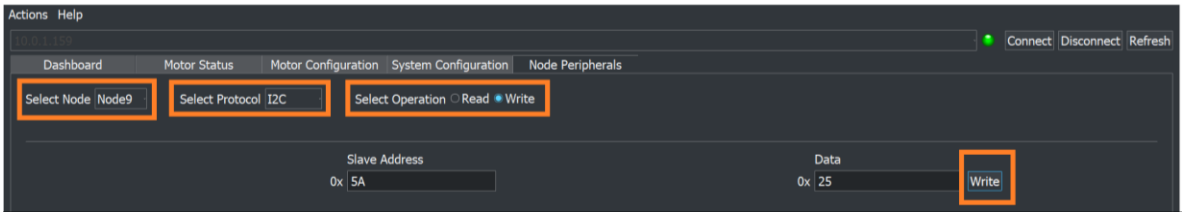

Figure 7.54. Application Software - Node Peripherals: I²C-Write

8. In the Total Phase tool, check the transaction log.

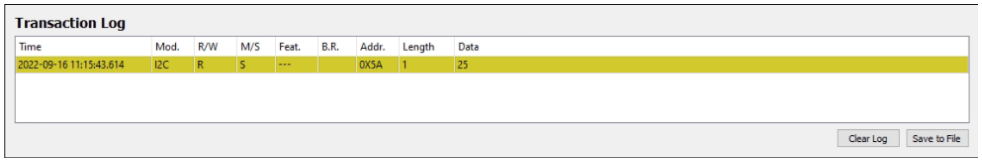

Figure 7.55. Total Phase - Transaction Log-I²C

For read operation:

1. Using the Aardvark user interface, write the value 12 in the message box and click **Set Resp**.

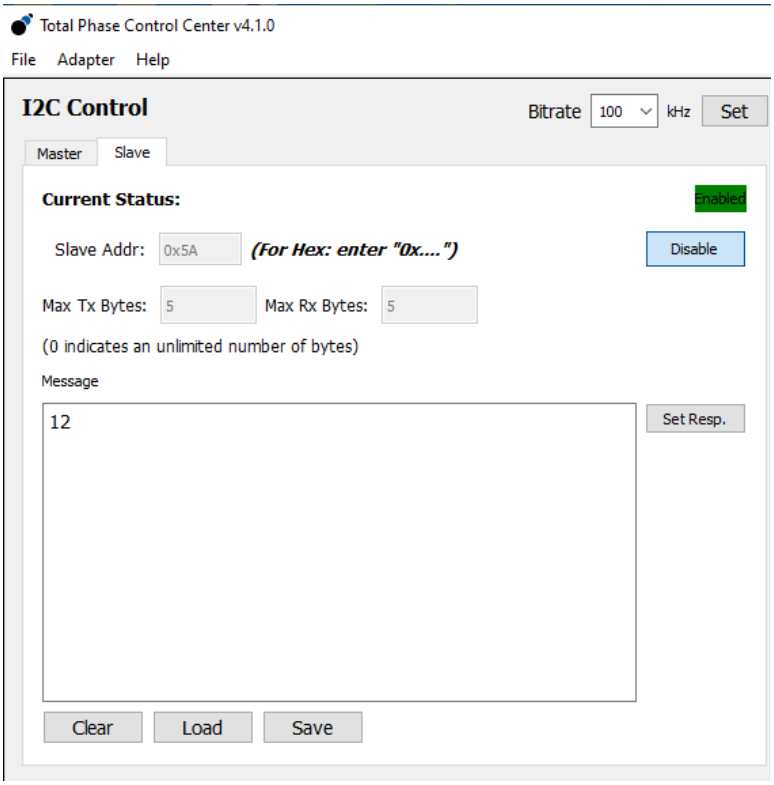

Figure 7.56. Total Phase Aardvark I²C Control User Interface

2. Check the transaction log.

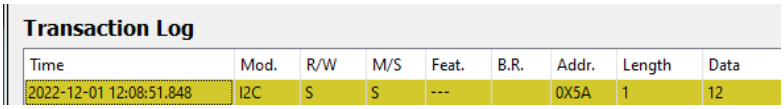

Figure 7.57. Total Phase - Transaction Log - I²C

3. In the Node Peripherals tab of the Automate 3.1 user interface, select the Operation: **Read.**
4. Click **Read**. The data value read should be 12.

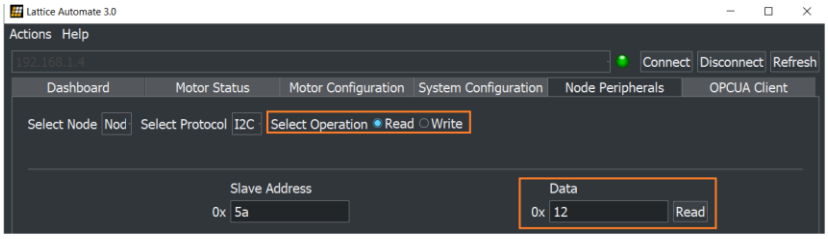

Figure 7.58. Application Software – Node Peripherals: I²C-Read

7.10.3. SPI

For the SPI write and read operation:

1. In the Automate 3.1 user interface, click the **Node Peripherals** tab.
2. Select the **Node** which is connected to the Aardvark I²C/SPI Host Adapter
3. Select the protocol: **SPI**
4. Select the operation: **Write**
5. Slave Address 2 bit: **Disabled**
6. Enter the Data 2 bit: 25 (you can write any data from 0x00 to 0xFF.)
7. Click **Write**.

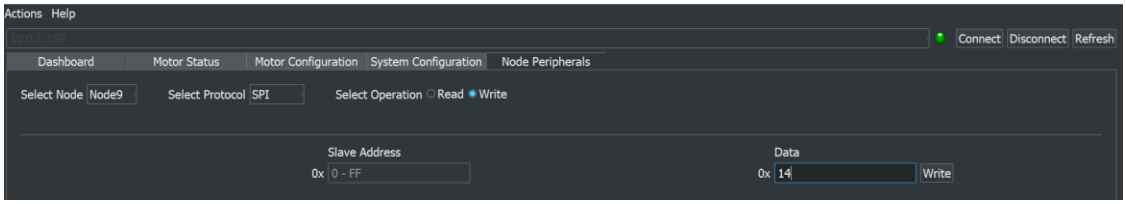

Figure 7.59. Application Software - Node Peripherals: SPI-Write

8. In the Total Phase tool, check the transaction log.
   Note: Make sure that the physical connection between the Aardvark and PMOD of the node board is correct.

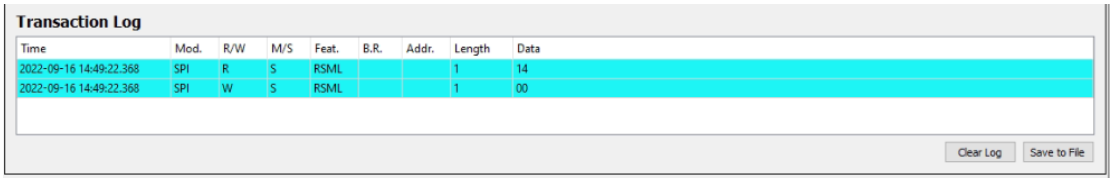

Figure 7.60. Total Phase - Adapter Setting: Transaction Log -SPI

9. Write the value 22 in the MISO message box and click **Set MISO**.

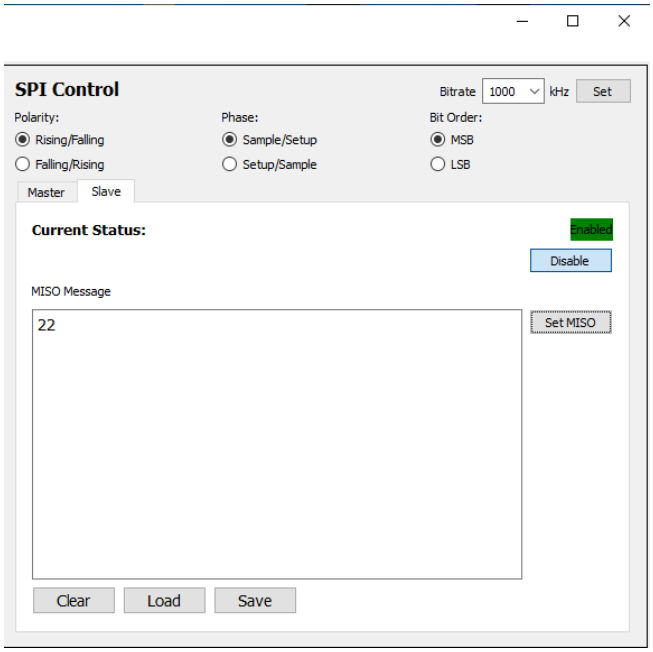

Figure 7.61. Total Phase - Adapter Setting: SPI Control

10. Check the transaction log.

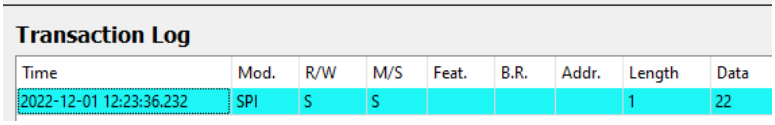

Figure 7.62. Total Phase - Adapter Setting: Transaction Log-SPI

11. In the **Node Peripherals** tab of the Automate 3.1 user interface, select the Operation: **Read,**
12. Click **Read**. The value in the field Data should match the value that was entered in step 9.

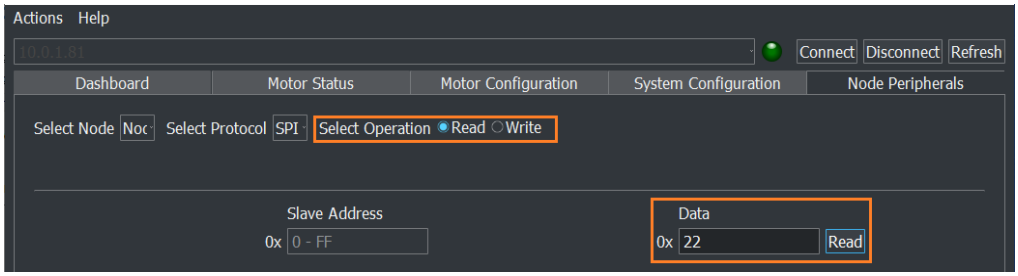

Figure 7.63. Application Software - Node Peripherals: SPI-Read

7.10.4. Modbus

Note: Connect the UART cable from the UART port of the Node system to the PC/Laptop.

Note: Required tool: Docklight v2.4 or later.

For the Modbus write operation:

1. In the Automate 3.1 user interface, click the **Node Peripherals** tab.
2. Enter the Slave Address: 06 (you can enter 1-10).

3. Enter the Register Number: 07 (you can enter 0-9).
4. Enter the Data: 1285 (you can enter 0-FFFF).

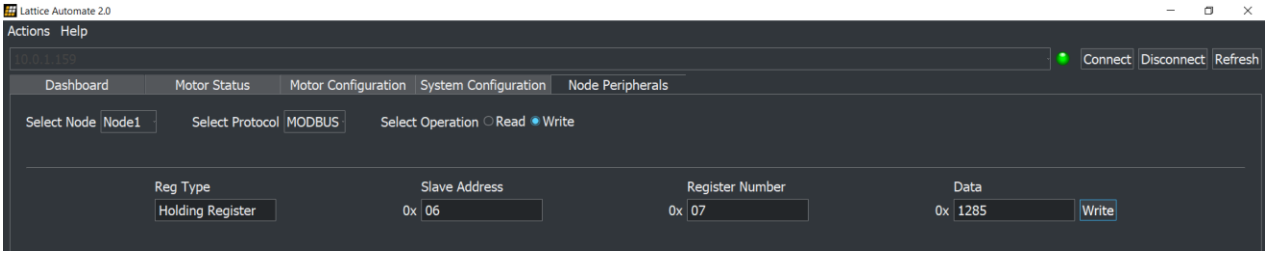

Figure 7.64. Application Software - Node Peripherals: Modbus-Write

5. Open Docklight.
6. Double-click **COM**.

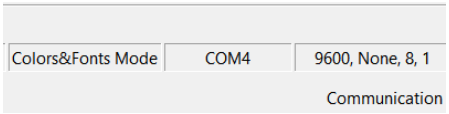

Figure 7.65. Docklight COM Selection

7. Select the last USB serial port in the list, as shown in Figure 7.66.

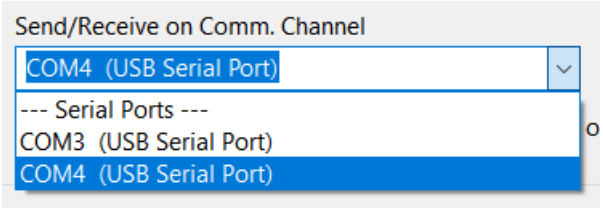

Figure 7.66. USB Serial Port Selection

8. Select the Baud Rate: **9600**.

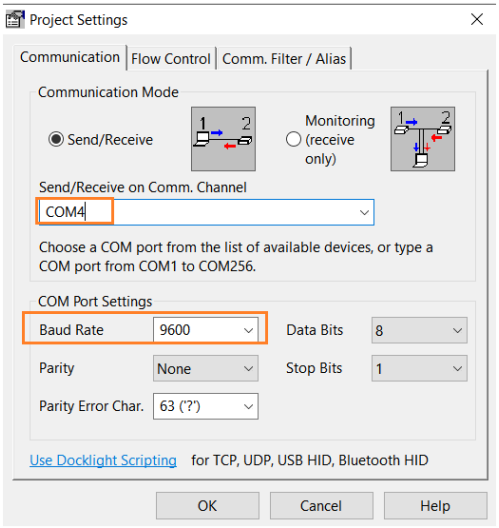

Figure 7.67. Docklight - Project Settings

9. Click **Run**.

Figure 7.68. Docklight - Run

10. In the Automate 3.1 user interface, click **Write**.

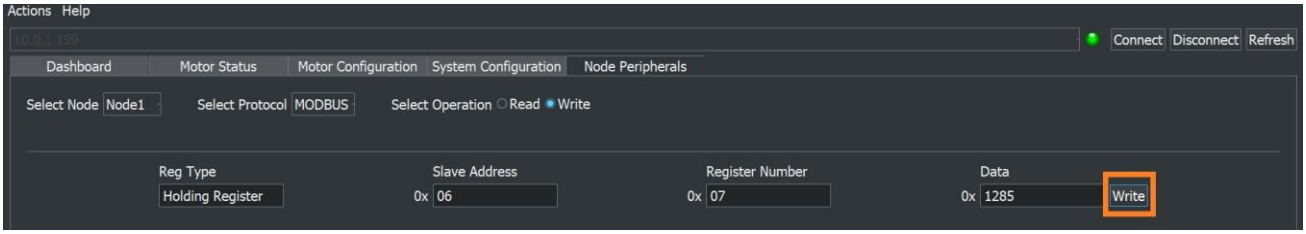

Figure 7.69. Application Software- Node Peripheral: Modbus-Write

11. In Docklight, click the **HEX** tab. You can see the transmitted message printed in red.

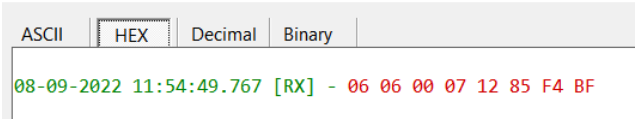

Figure 7.70. Docklight Prints

Appendix A. User Interface Application Installation (PC)

Note: This installation process is done on the PC to run the user application, such as a laptop or desktop PC.

To install the user interface application on a PC:

1. Download or locate the Lattice Automate 3.1 Installer.
2. Double-click on the installer to install the application.

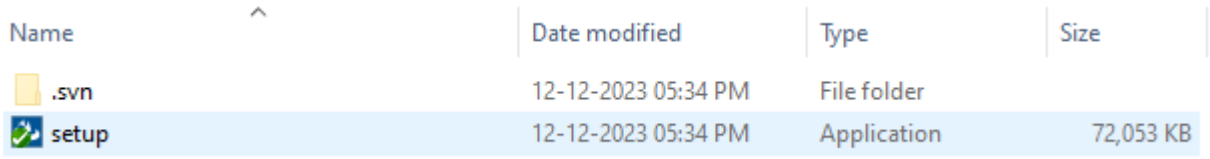

Figure A.1. Installer Directory

3. Wait for the installation to complete.

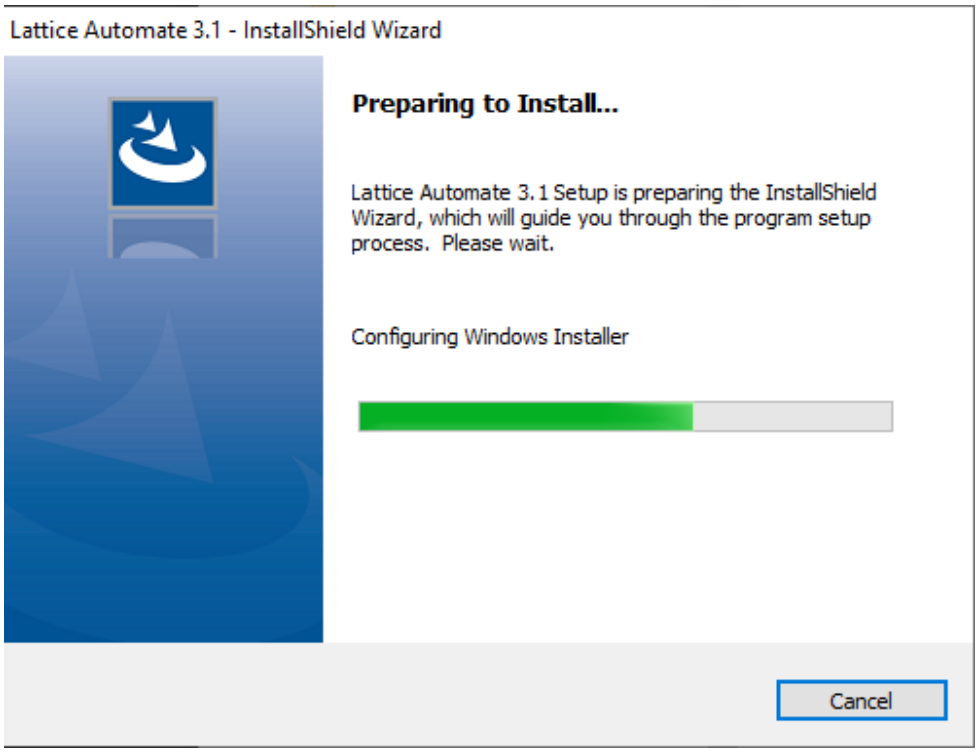

Figure A.2. Automate 3.1 Initial Installation Setup

4. Click **Next**.

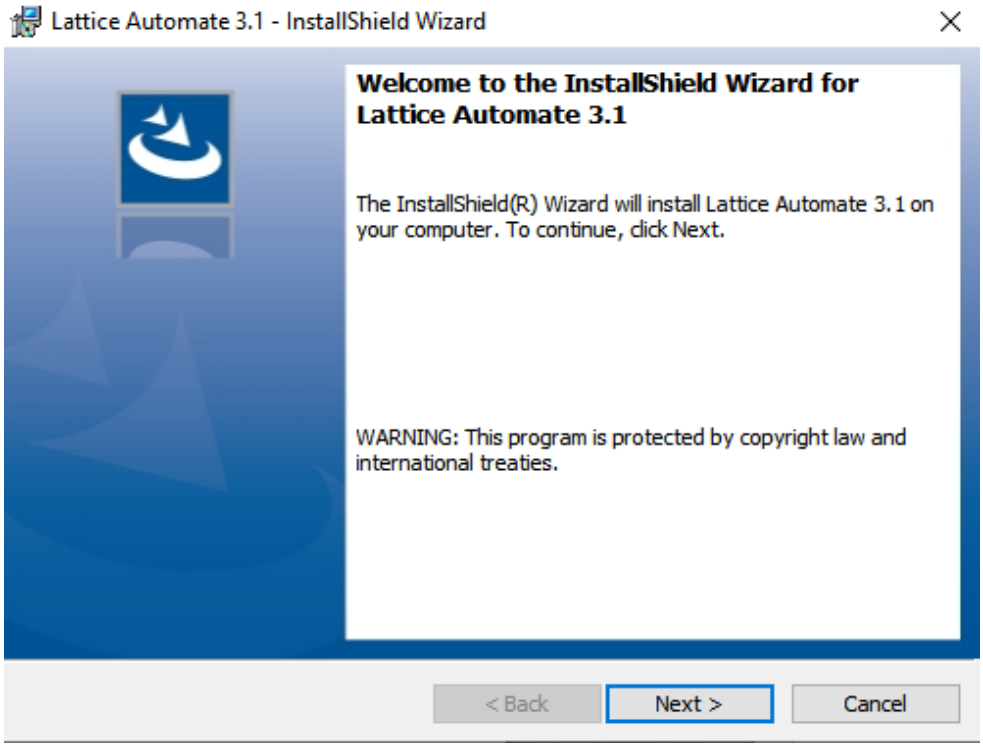

Figure A.3 Automate 3.1 Next Step

5. Click **Next**.

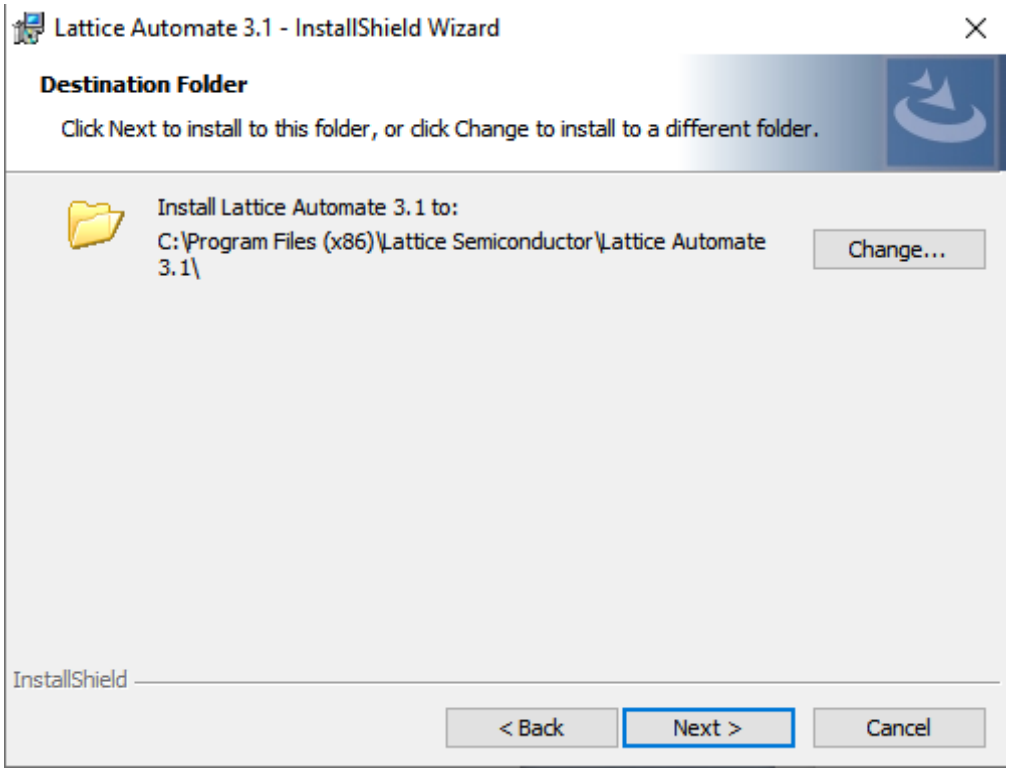

Figure A.4. Automate 3.1 Next Step

6. Click **Install**.

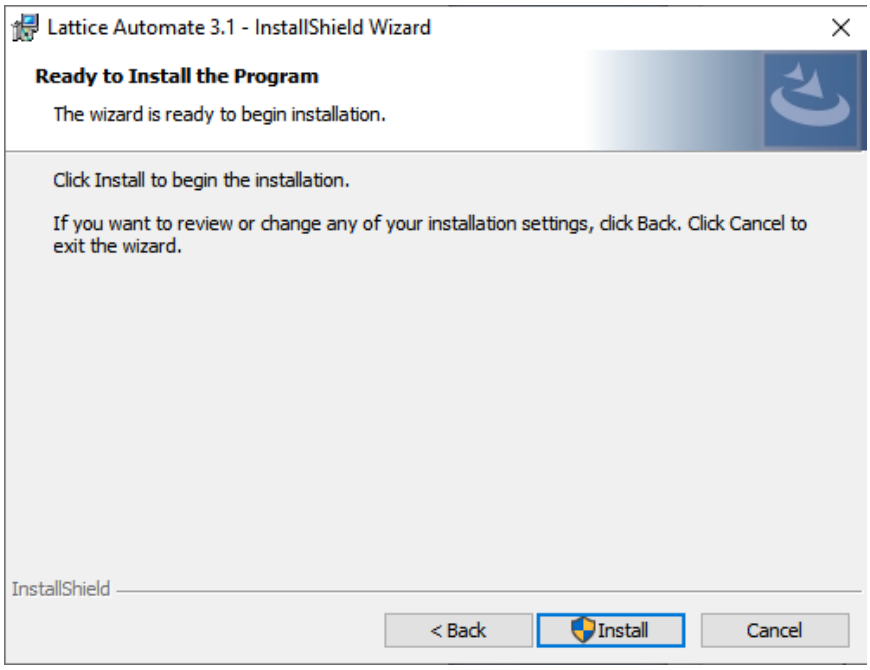

Figure A.5. Automate 3.1 Click Install

7. Wait for the installation to complete. Click **Yes** to allow permission.

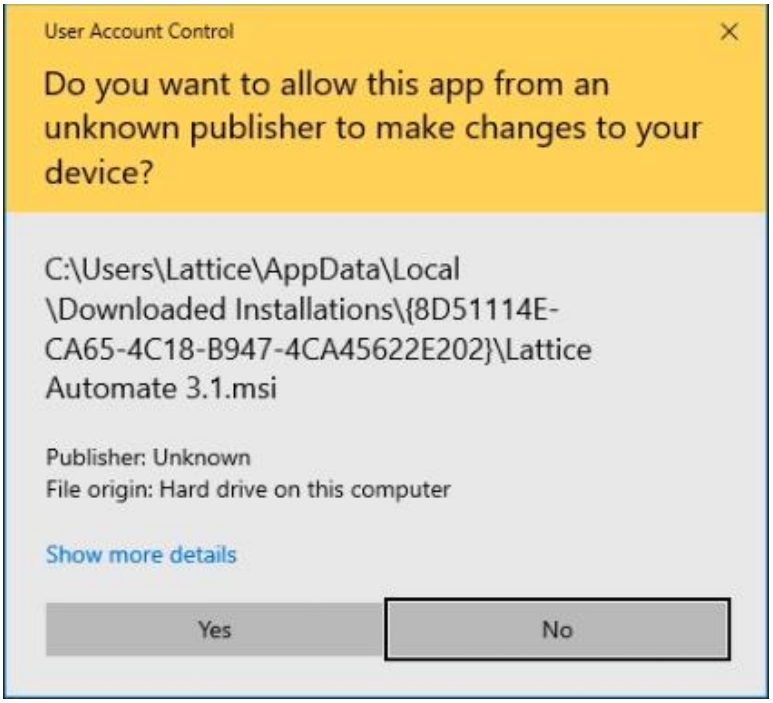

Figure A.6. Allow Permission Window

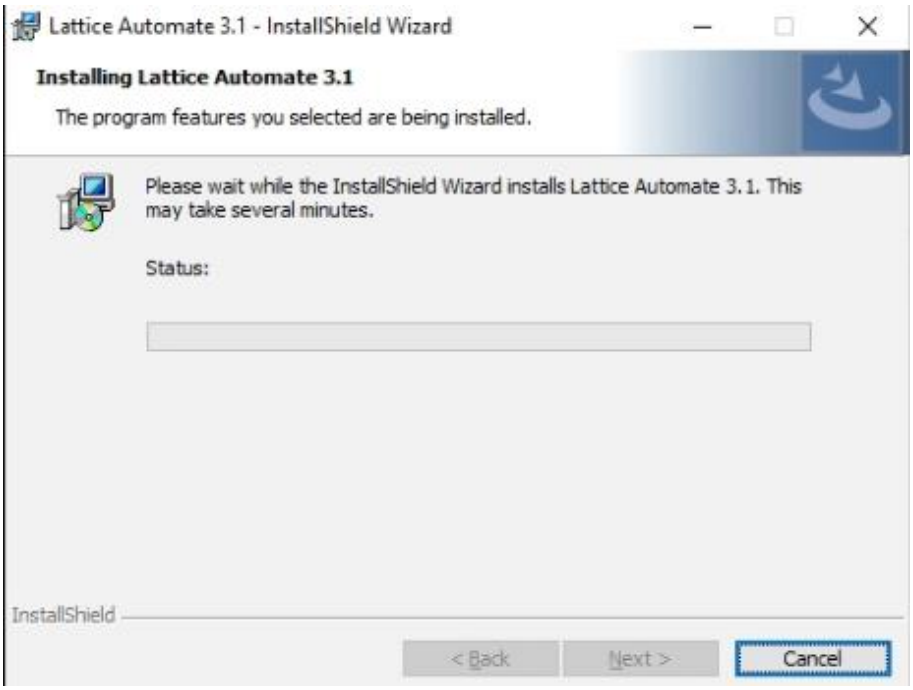

Figure A.7. Automate 3.1 Installation in Progress

8. Click **Finish**.

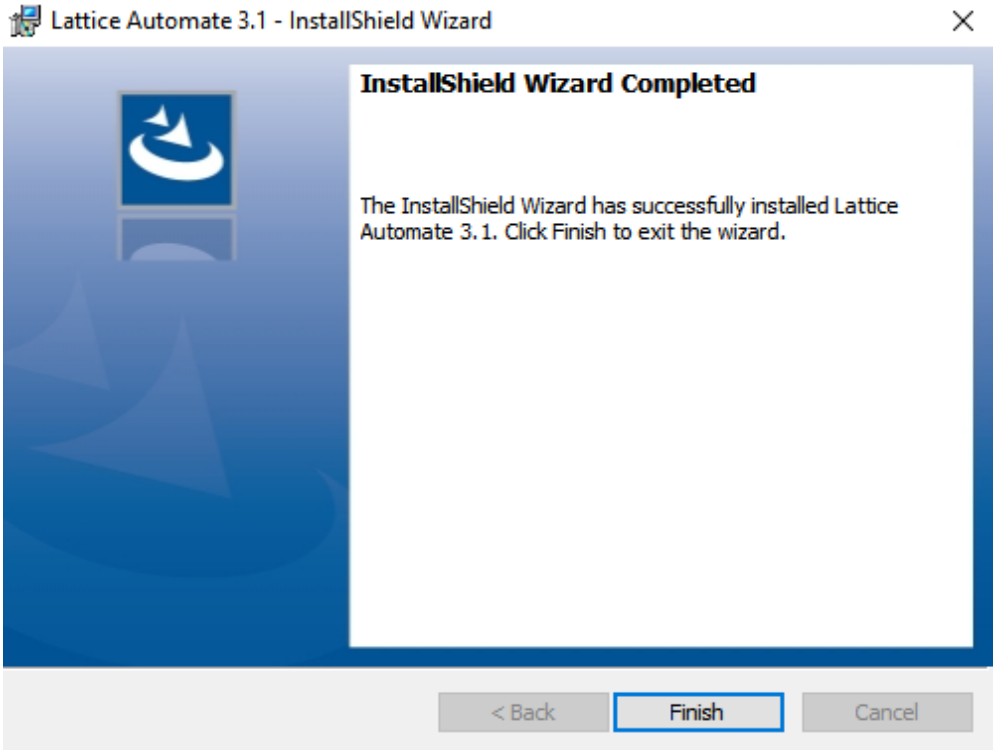

Figure A.8. Automate 3.1 Installation Finish

Appendix B. Programming the Automate Stack on Respective Flash

B.1. Main System

This section provides the procedure for programming the SPI Flash on the CertusPro-NX Versa board for the main system.

Two different files should be programmed into the SPI Flash. These files are programmed to the same SPI Flash, but at different addresses:

- Bitstream (FPGA SoC Design)
- Binary (RISC-V Firmware)

Board Jumper Connections

The following jumpers must be connected on the board:

- Pin 1 and 2 of J32 and J33 must be shorted to select UART.
- Pin 1 and 2 of J58 should be shorted to select the 3.3 V as Flash I/O.

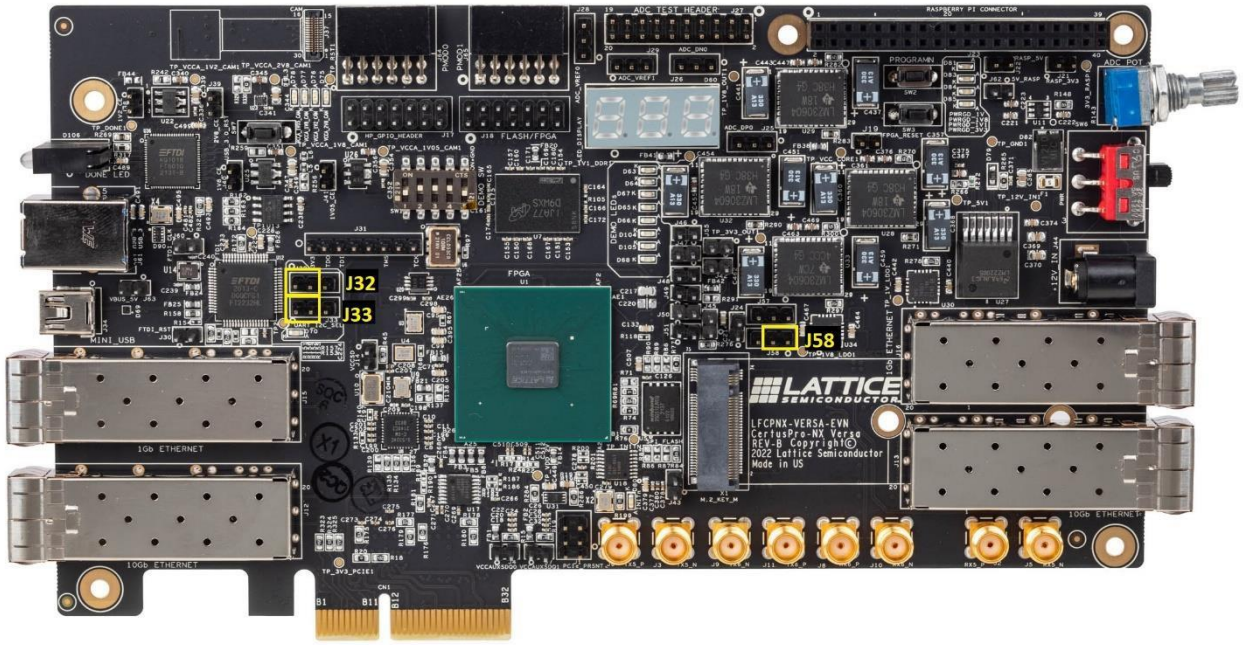

Figure B.1. Main System Jumper Connection

If the user programming the main system board for the first time, refer to the Appendix C. Programming a Fresh Main System Board section for more details. Once done, go back to this section to follow further steps.

B.1.1. Programming NV Register – Enabling 32-bit SPI Addressing

To program the NV register:

1. Apply the settings as shown in Figure B.2. This is a one-time step required to configure the NV Register 1 to enable 32-bit SPI addressing. Click **OK**.

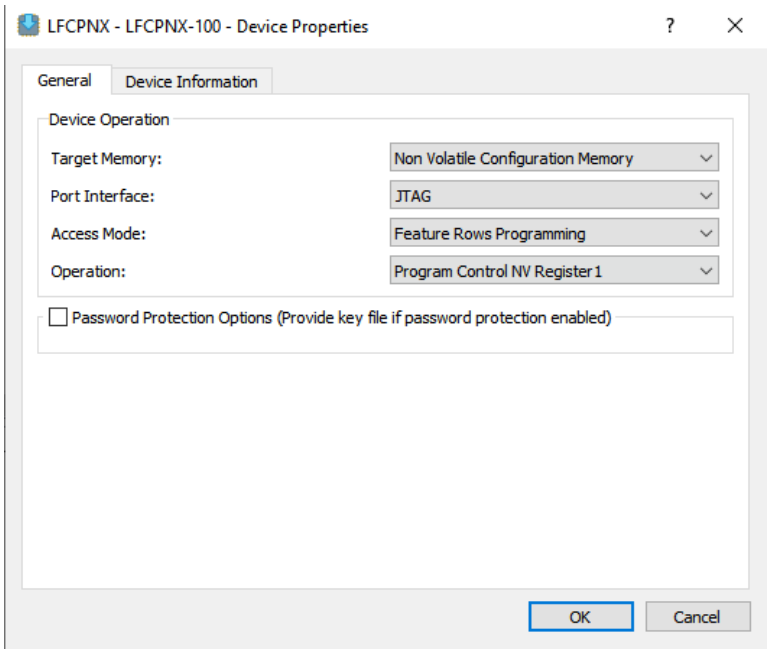

Figure B.2. Radiant Programmer – Configure NV Register 1 Setting

2. Click the **Program Device** icon or go to **Run > Program Device**.
3. Change the bit 0 to 1 as shown in Figure B.3.

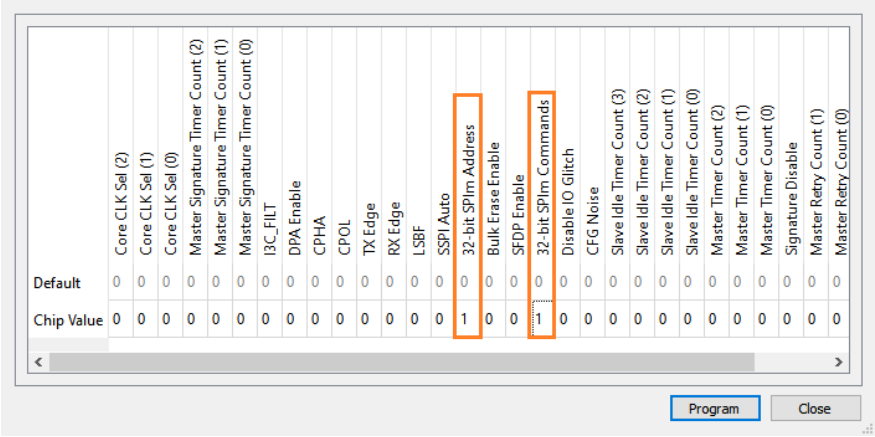

Figure B.3. Radiant Programmer – Configure NV Register 1 Setting: Change Chip Value

4. After changing the bit, click **Program**.
5. Power cycle the CertusPro NX versa board.

B.1.2. Primary Main System

B.1.2.1. Programming the FPGA Bitstream

To program the FPGA bitstream in external SPI Flash using Radiant Programmer:

1. Connect the CertusPro-NX Versa board to the PC using a USB cable.
2. Start Radiant Programmer. In the Getting Started dialog box, select **Create a new blank project**.

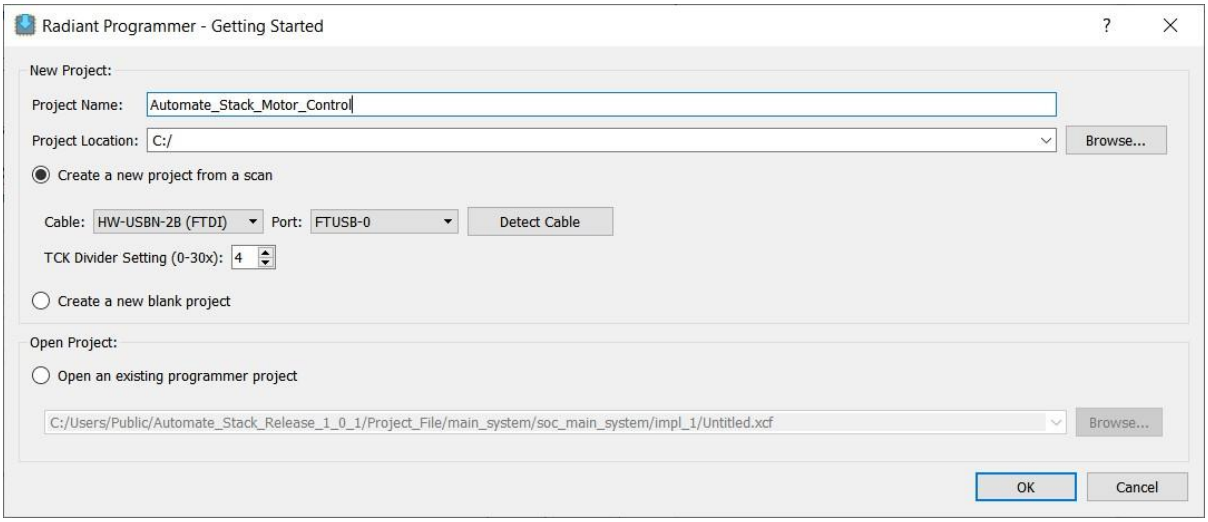

Figure B.4. Radiant Programmer - Default Screen (Main System)

3. Click **OK**.

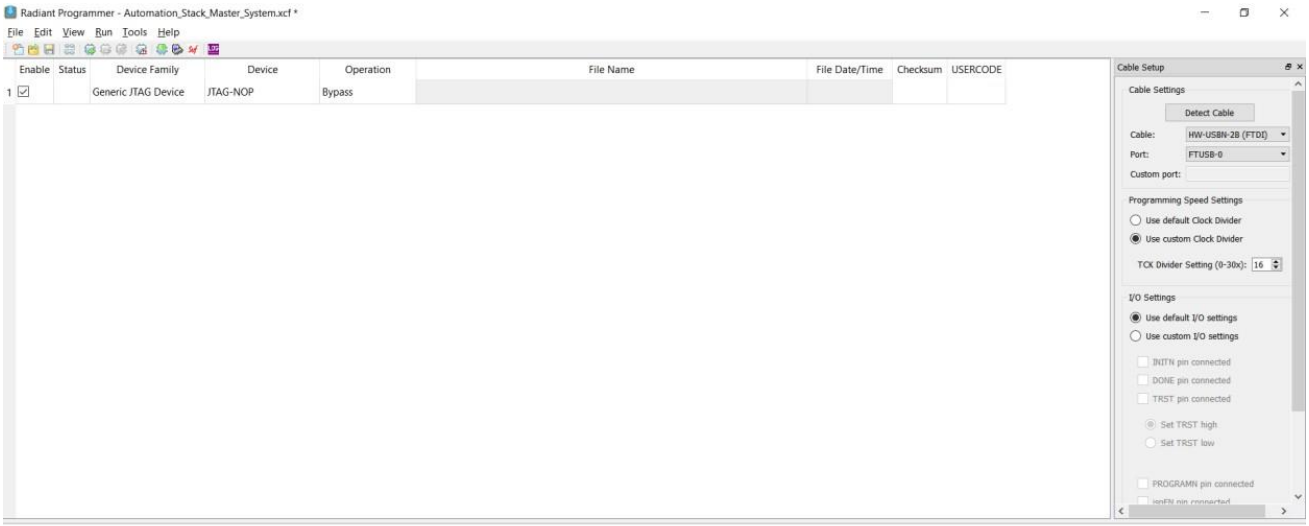

Figure B.5. Radiant Programmer - Initial Project Window (Main System)

4. In the Radiant Programmer main interface, select **LFCPNX** for **Device Family** and **LFCPNX-100** for **Device** or detect automatically as shown in Figure B.6.

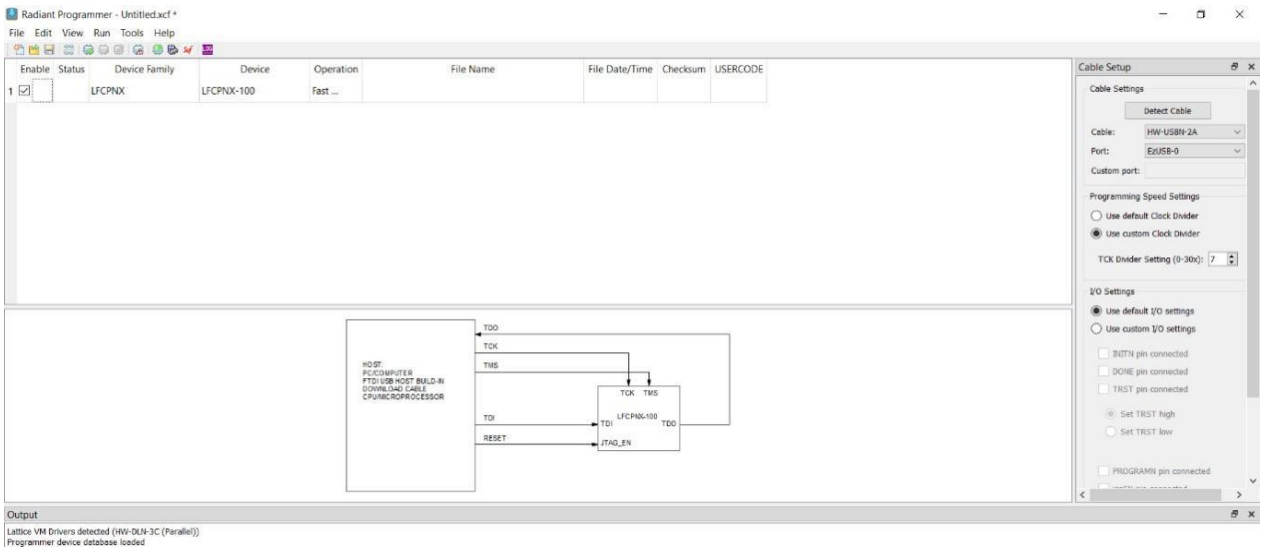

Figure B.6. Radiant Programmer - Device Selection (Main System)

5. Right-click and select **Device Properties**.

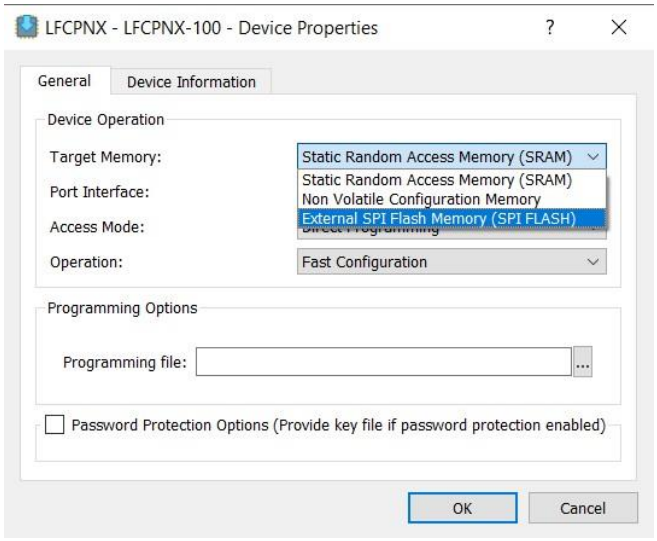

Figure B.7. Radiant Programmer - Device Operation (Main System)

6. Before programming, apply the settings below to erase the flash memory:
   a. Under Device Operation, select the options below:
      - Target Memory – **External SPI Flash Memory (SPI FLASH)**
      - Port Interface – **JTAG2SPI**
      - Access Mode – **Direct Programming**
      - Operation – **Erase all**
   b. Under SPI Flash Options, select the options below:
      - Family – **SPI Serial Flash**
      - Vendor – **Macronix**
      - Device – **MX25L51245G**
      - Package – **8-land WSON**

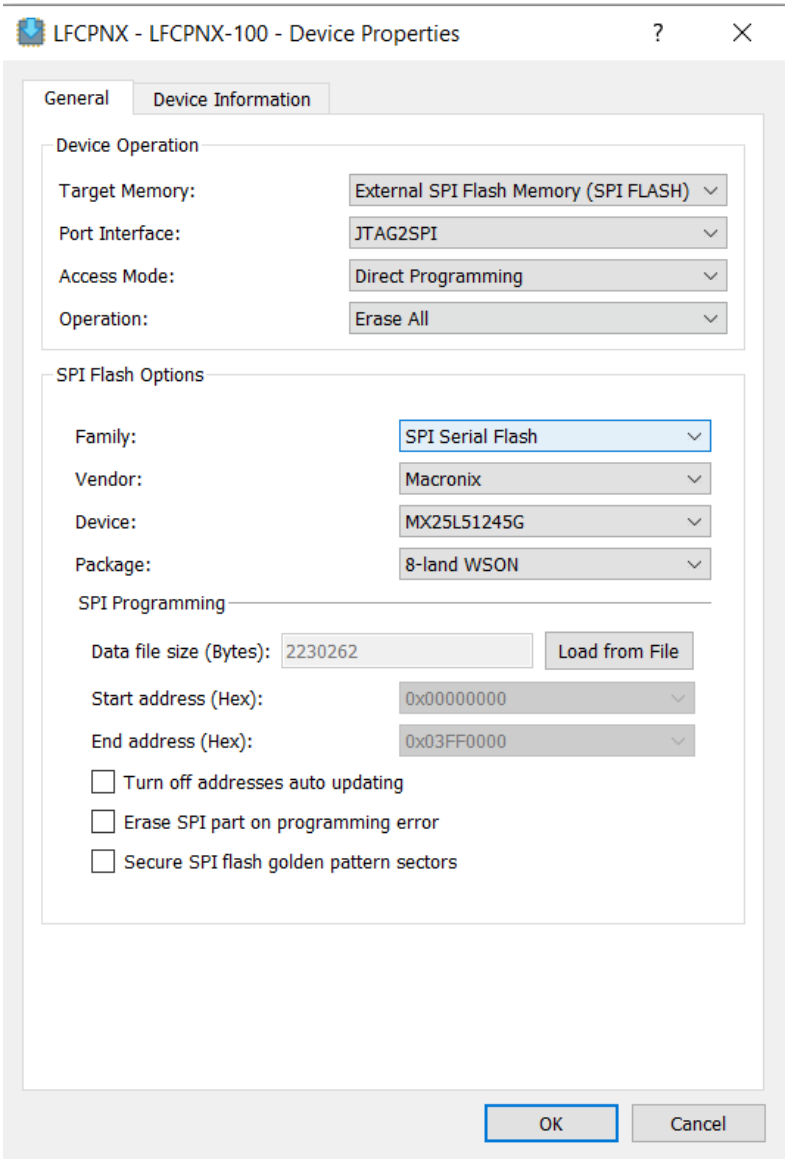

Figure B.8. Radiant Programmer - Erase All (Main System)

7. Click **OK** and then click the Program Device icon or go to **Run > Program Device**. This erases the flash memory if any other data is already present in it.
8. After erasing the flash, power cycle the board and apply the settings below:
   a. Under Device Operation, select the options below:
      - Target Memory – **External SPI Flash Memory (SPI FLASH)**
      - Port Interface – **JTAG2SPI**
      - Access Mode – **Direct Programming**
      - Operation – **Erase, Program, Verify**
   b. Under SPI Flash Options, select the options below:
      - Family – **SPI Serial Flash**
      - Vendor – **Macronix**
      - Device – **MX25L51245G**
      - Package – **8-land WSON**

9. To program the **bitstream file**, select the options as shown in Figure B.9.

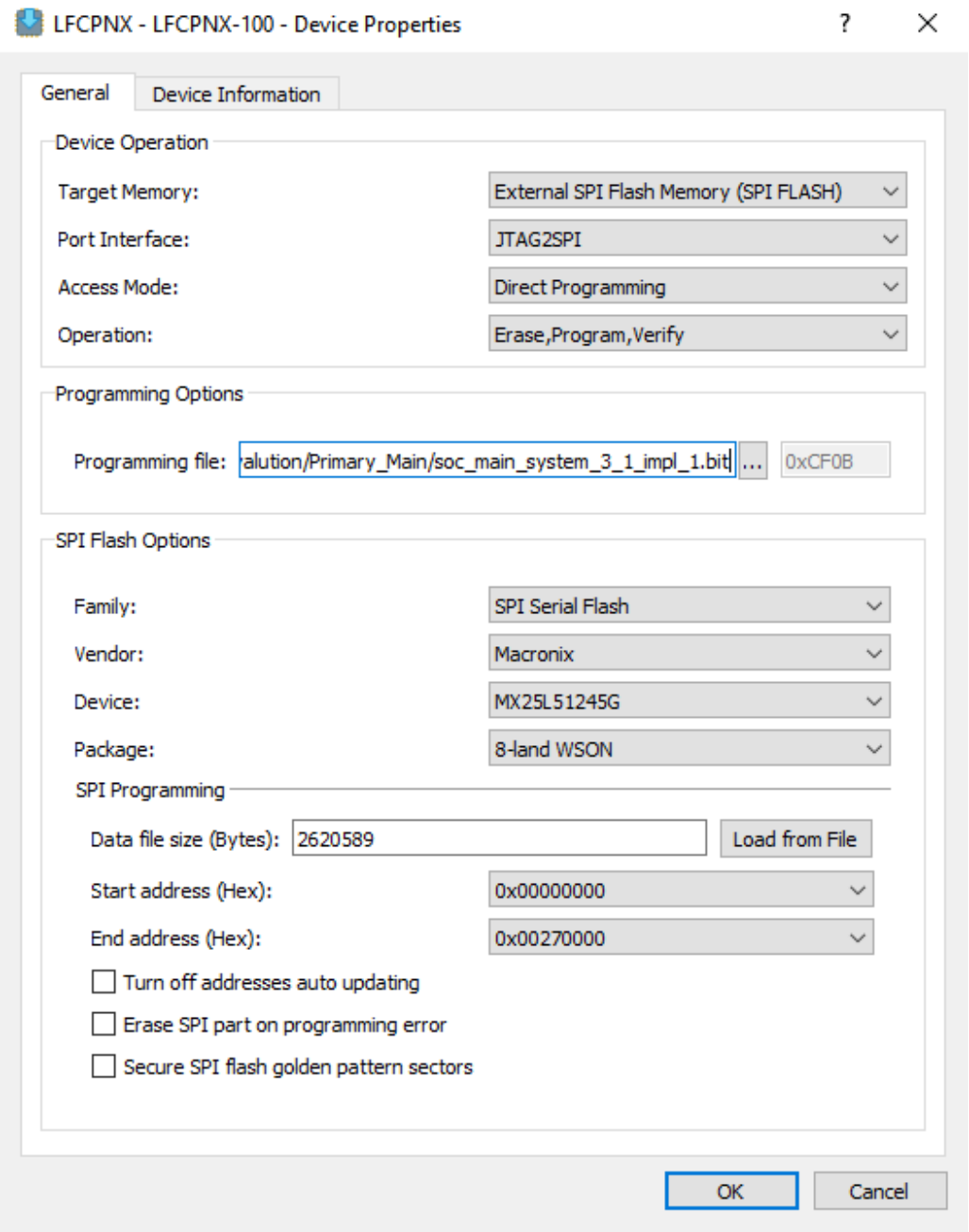

Figure B.9. Radiant Programmer - Bitstream Flashing Settings (Main System)

a. Under **Programming** Options, select the **soc_main_system_impl_1.bit** bitstream file in the Programming file.
b. Click **Load from File** to update the Data file size (Bytes) value.
c. Make sure that the following addresses are correct:
   - Start Address (Hex) – **0x00000000**
   - End Address (Hex) – **0x00280000**

10. Click the Program Device icon or go to **Run > Program Device**.

B.1.2.2. Primary Main System Firmware

To program the firmware, perform the following steps:

1. Select the options as shown in Figure B.10.
   a. Under Programming Options, select the **c_main_system_3_1_cnnCrc.bin** binary file.
   b. Make sure that the following addresses are correct:
      - **Start Address (Hex) – 0x20A0000**
      - **End Address (Hex) – 0x020D0000**

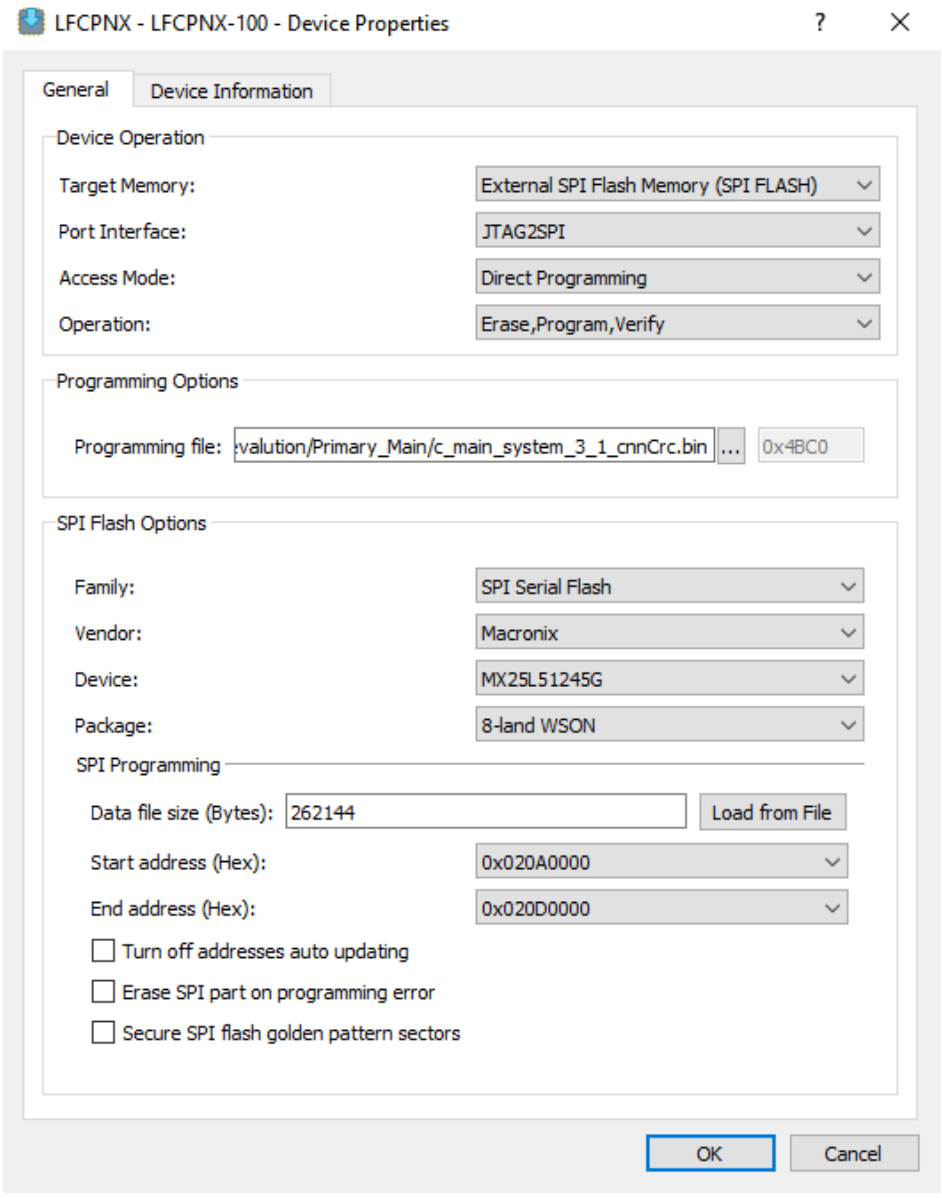

Figure B.10. Radiant Programmer - Binary Flashing Settings

2. Click the Program Device icon or go to **Run > Program Device.**
3. Power cycle the CertusPro NX versa board.

B.1.3. Golden Main System

To program the firmware in the SPI Flash in Radiant Programmer:

1. Select the options as shown in Figure B.11.
   a. Under Programming Options, select the **Golden_AppCrc.bin** binary file.
   b. Make sure that the following addresses are correct:
      - **Start Address (Hex) – 0x02000000**
      - **End Address (Hex) – 0x02030000**

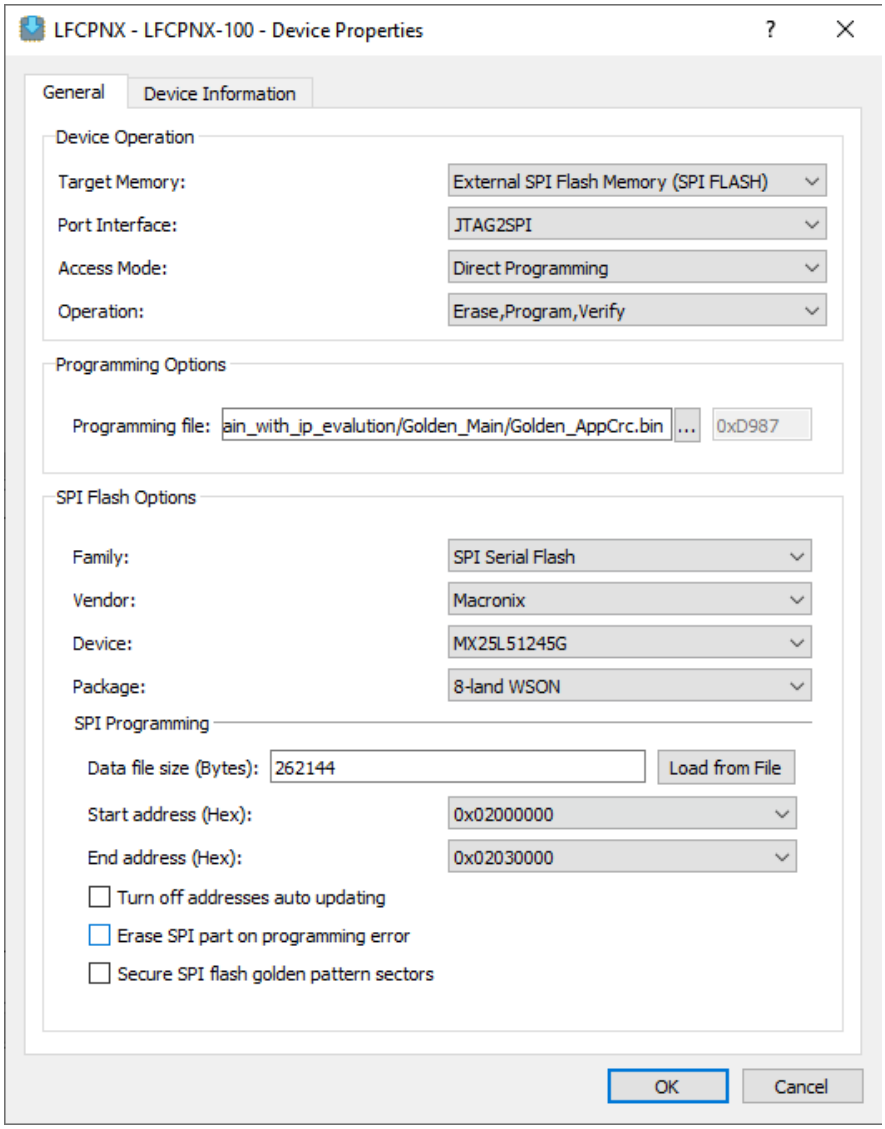

Figure B.11. Radiant Programmer - Binary Flashing Settings

2. Click the Program Device icon or go to **Run > Program Device**

B.1.3.1. Golden Main System Bitstream

Note: The bitstream steps below are only valid for standalone projects. No need to program the bitstream while programming the MCS file.

After erasing the flash, perform the steps below:

1. Power cycle the board and apply the setting s below:
   a. Under Programming Options, select the **soc_main_system_impl_1.bit** bitsream file.
   b. Click **Load** from File to update the Data file size (Bytes) value.
   c. Make sure that the following addresses are correct:
      - **Start Address (Hex) – 0x00000000**
      - **End Address (Hex) – 0x00240000**
   d. To program the bitstream, select the options as shown in Figure B.12

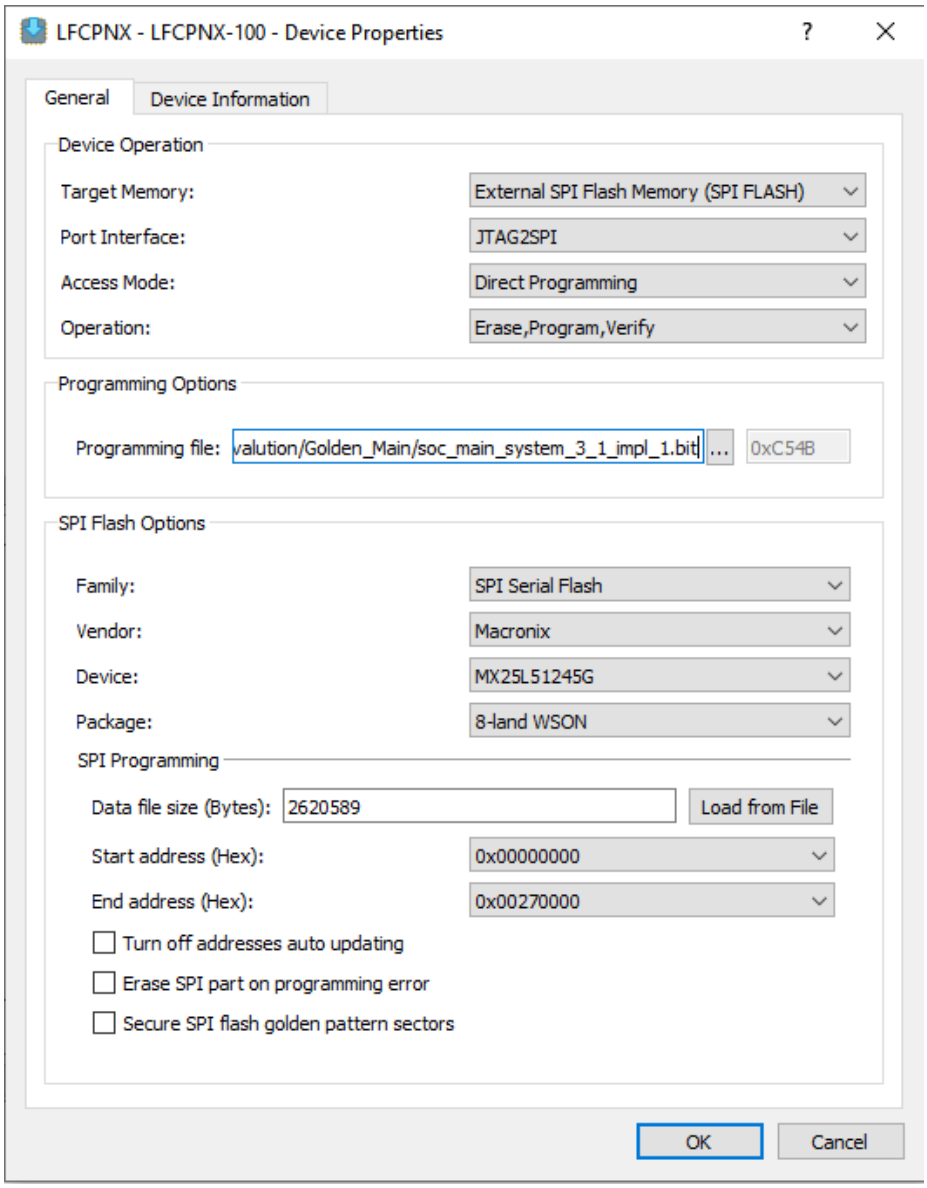

Figure B.12. Radiant Programmer - Bitstream Flashing Settings

2. Click the Program Device icon or go to **Run > Program Device**.
3. Power cycle the CertusPro NX versa board.

B.1.4. Programming the MCS File

To program MCS file in the SPI Flash in Radiant Programmer:

1. To program the MCS file, select the options as shown in Figure B.13.
   a. Under Programming Options, select the **soc_golden_system_impl_1.mcs** mcs file.
   b. Make sure that the following addresses are correct:
      - **Start Address (Hex) – 0x0000000**
      - **End Address (Hex) – 0x007F0000**

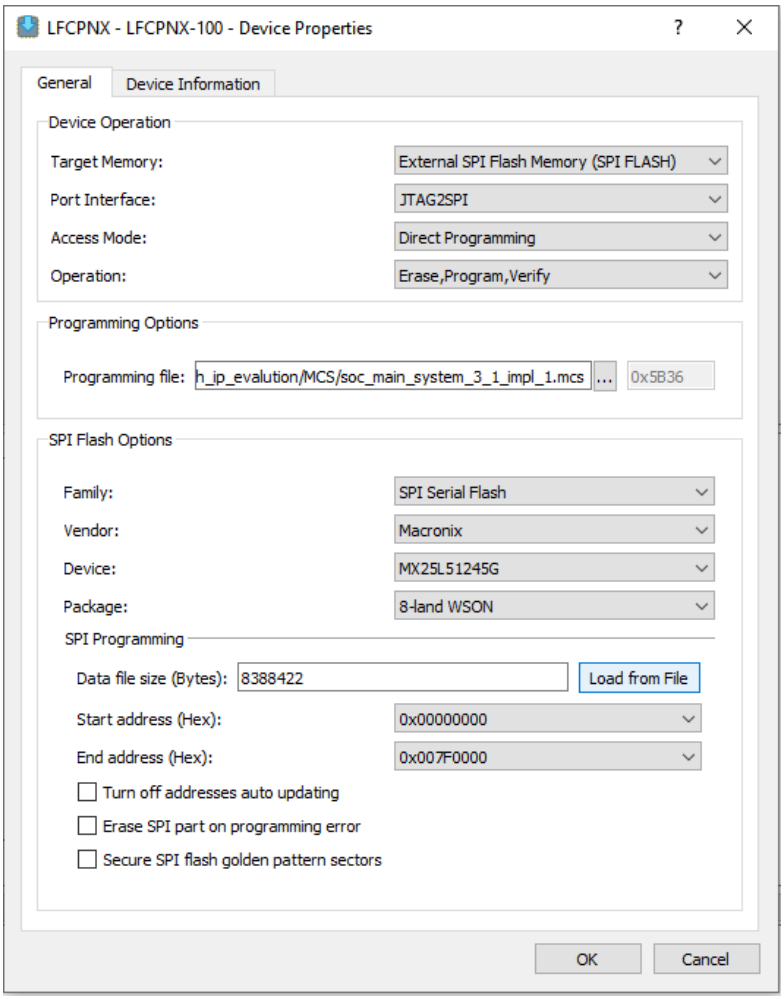

Figure B.13. Radiant Programmer - MCS Flashing Settings

2. Click the Program Device icon or go to **Run > Program Device.**
3. Power cycle the board the CertusPro NX versa board.

B.2. Node System

This section provides the procedure for programming the SPI Flash on the Certus-NX Versa board for node. Two different files should be programmed into the SPI Flash. These files are programmed to the same SPI Flash, but at different addresses:

- Bitstream
- Binary

Board Jumper Connections

Make sure that the following jumpers are connected on board:

- Pin 1 and 2 of JP25 and JP26 should be shorted to select UART.
- Pin 1 and 2 of J47 should be shorted to select the 1.8 V as Flash I/O.

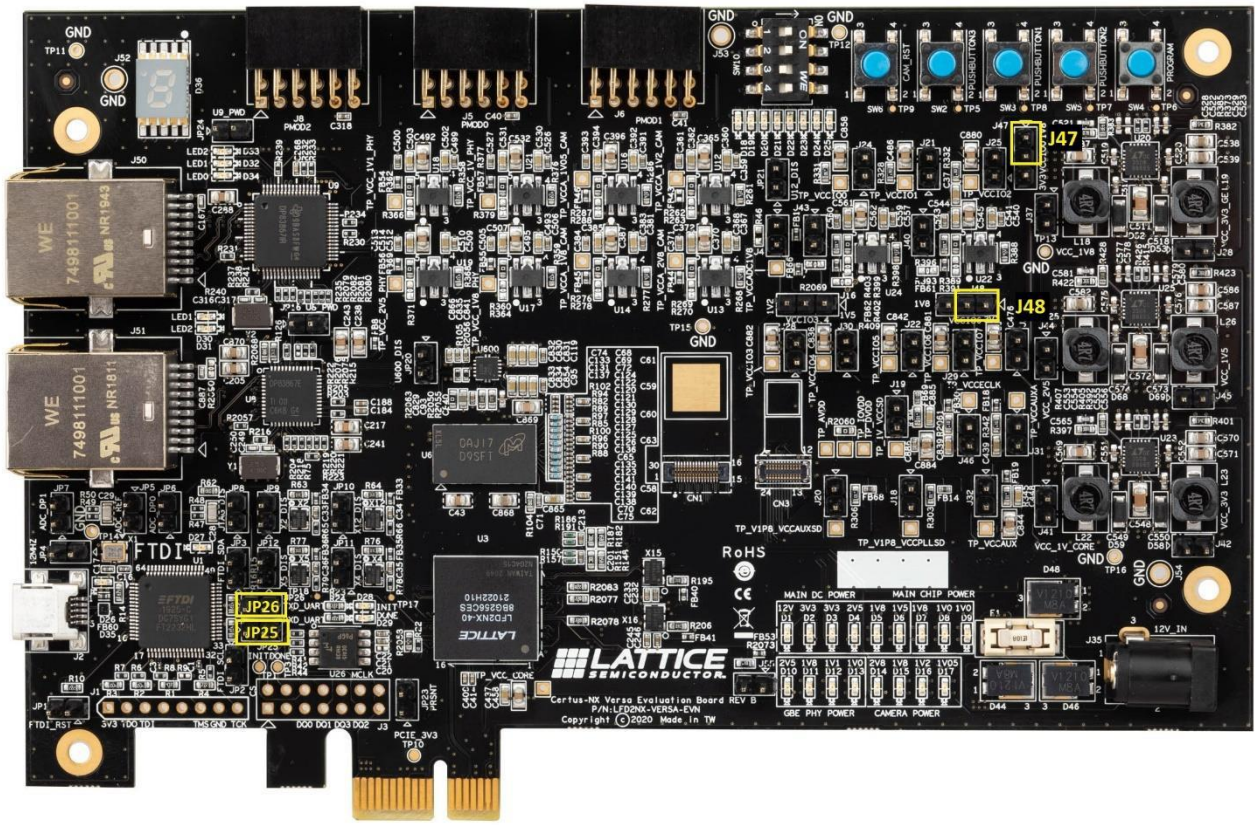

Figure B.14. Node System Jumper Connection

To program the SPI Flash in Radiant Programmer:

1. Connect the Certus-NX Versa board to the PC using a USB cable.
2. Start Radiant Programmer. In the **Getting Started** dialog box, select **Create a new blank project**.

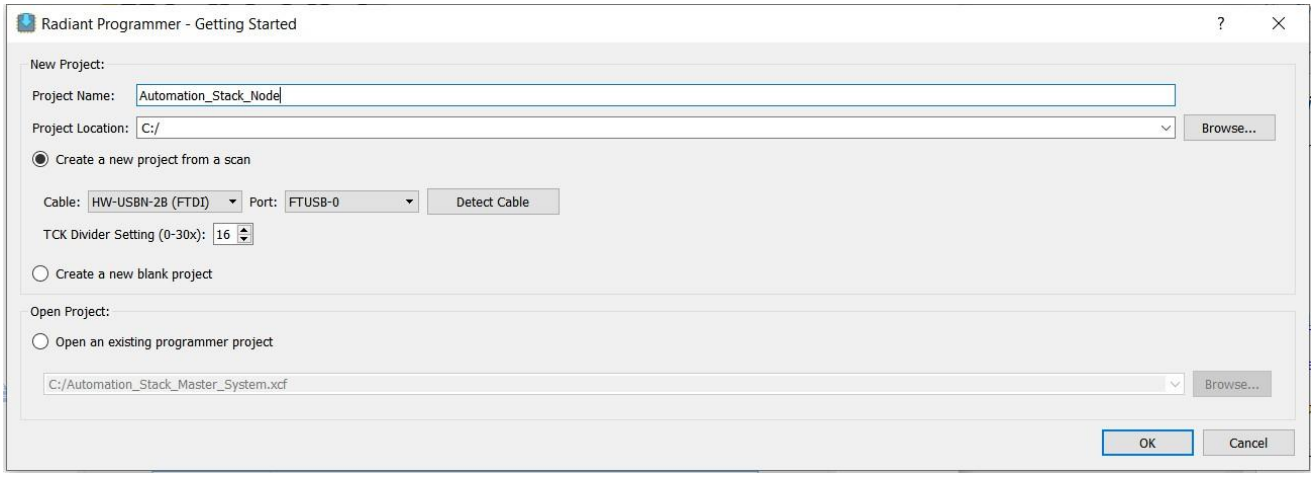

Figure B.15. Radiant Programmer - Default Screen (Node System)

3. Click **OK**.

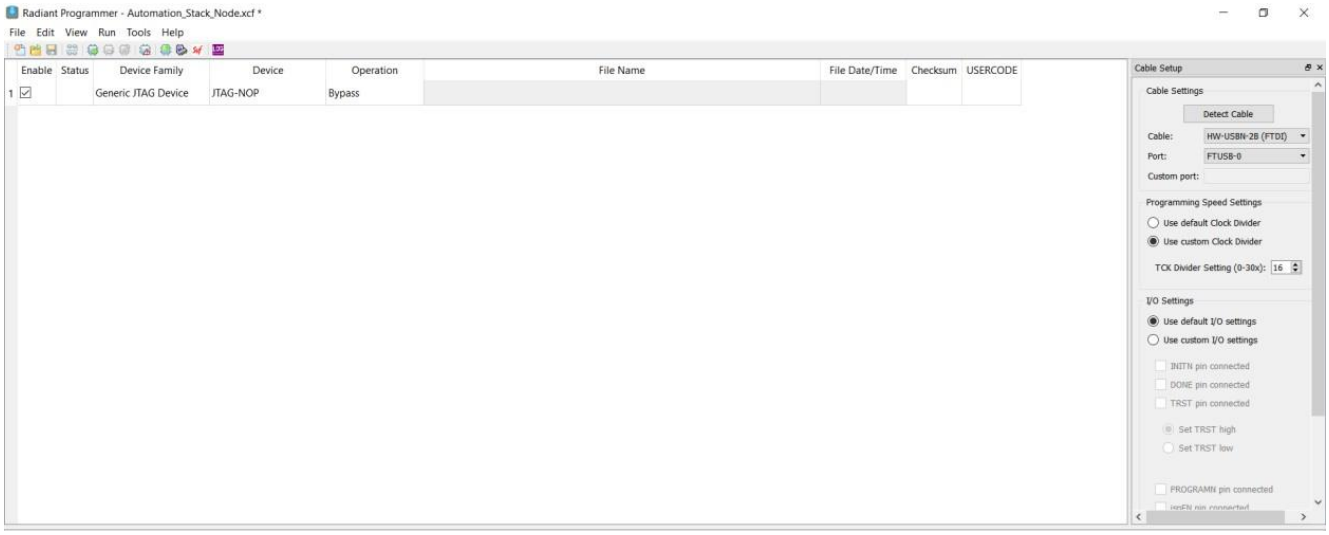

Figure B.16. Radiant Programmer - Initial Project Window (Node System)

4. In the Radiant Programmer main interface, select **LFD2NX** for **Device Family** and **LFD2NX-40** for **Device** as shown in Figure B.17.

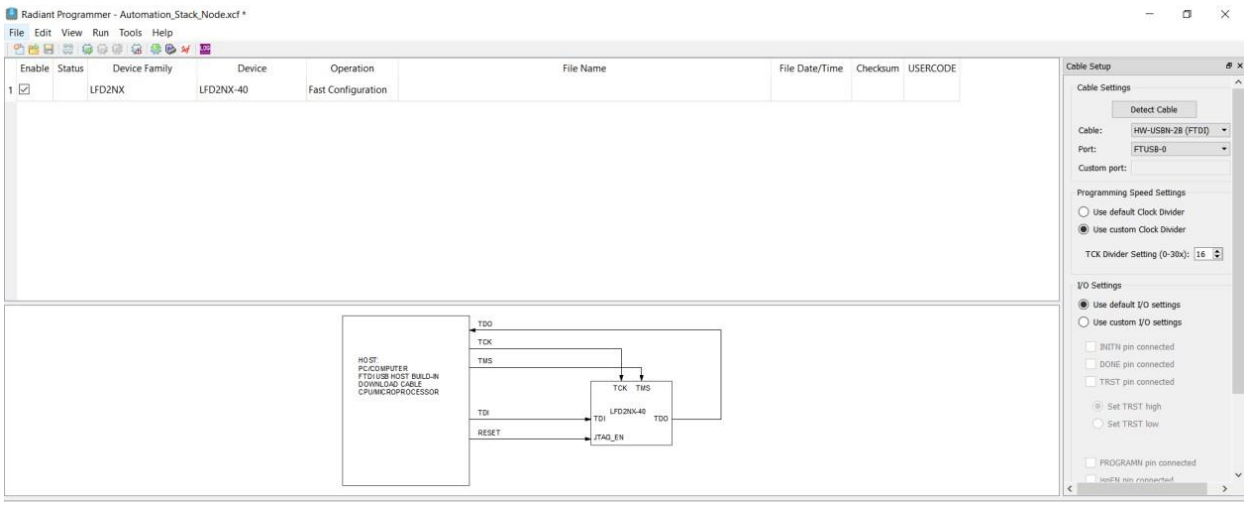

Figure B.17. Radiant Programmer - Device Properties (Node System)

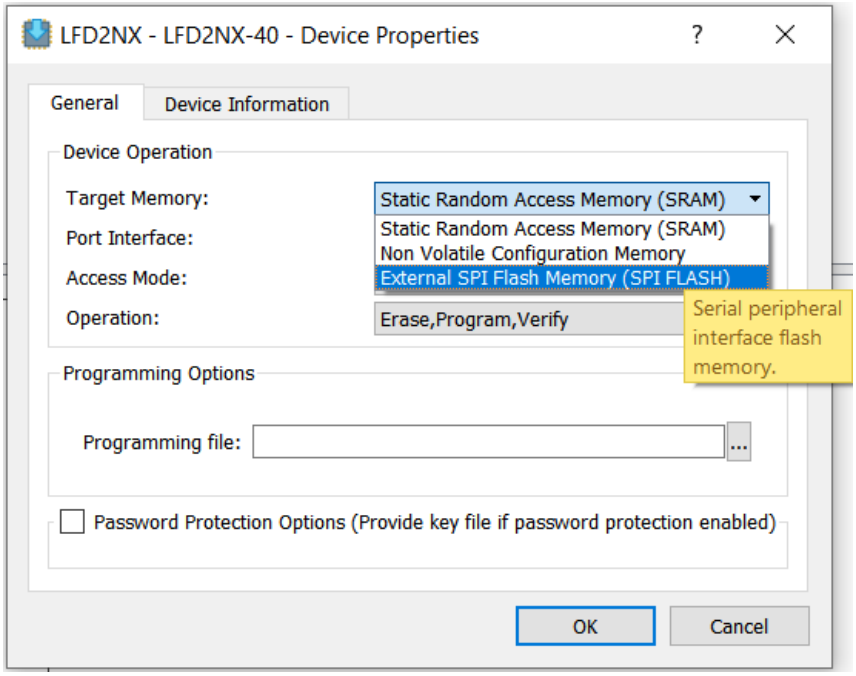

Figure B.18. Radiant Programmer - Device Operation (Node System)

5. Before programming, apply the settings below to erase the flash memory:
   a. Under Device Operation, select the options below:
      - Target Memory – **External SPI Flash Memory (SPI FLASH)**
      - Port Interface – **JTAG2SPI**
      - Access Mode – **Direct Programming**
      - Operation – **Erase all**
   b. Under SPI Flash Options, select the options below:
      - Family – **SPI Serial Flash**
      - Vendor – **Micron**
      - Device – **MT25QU128**
      - Package – **8-pin SOP2**

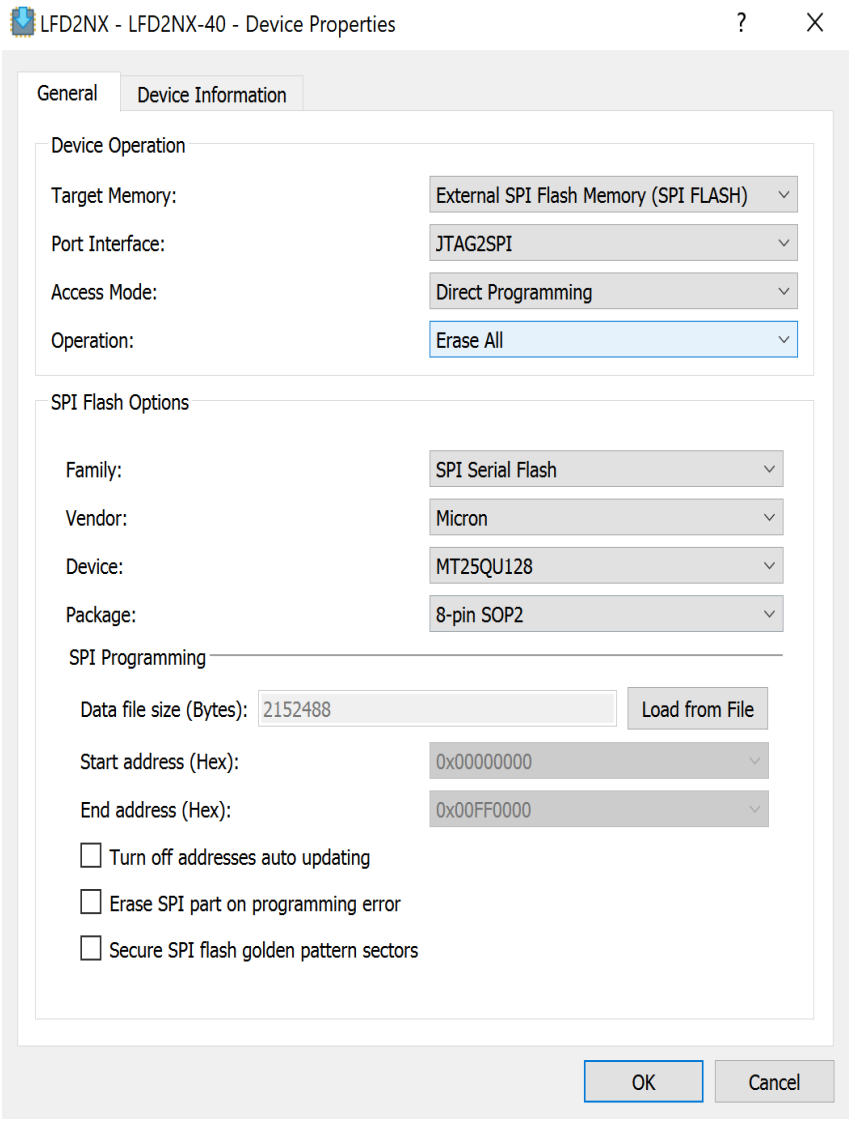

Figure B.19. Radiant Programmer - Erase All (Node System)

6. Click **OK** and then click the Program Device icon or go to **Run > Program Device**. This erases the flash memory if any other data is already present in it.
7. After erasing the flash, power cycle the board and apply the settings below:
   a. Under Device Operation, select the options below:
      - Target Memory – **External SPI Flash Memory**
      - Port Interface – **SPI**
      - Access Mode – **Direct Programming**
      - Operation – **Erase, Program, Verify**
   b. Under SPI Flash Options, select the options below:
      - Family – **SPI Serial Flash**
      - Vendor – **Micron**
      - Device – **MT25QU128**
      - Package – **8-pin SOP2**

8. To program the bitstream file, select the options as shown in Figure B.20.

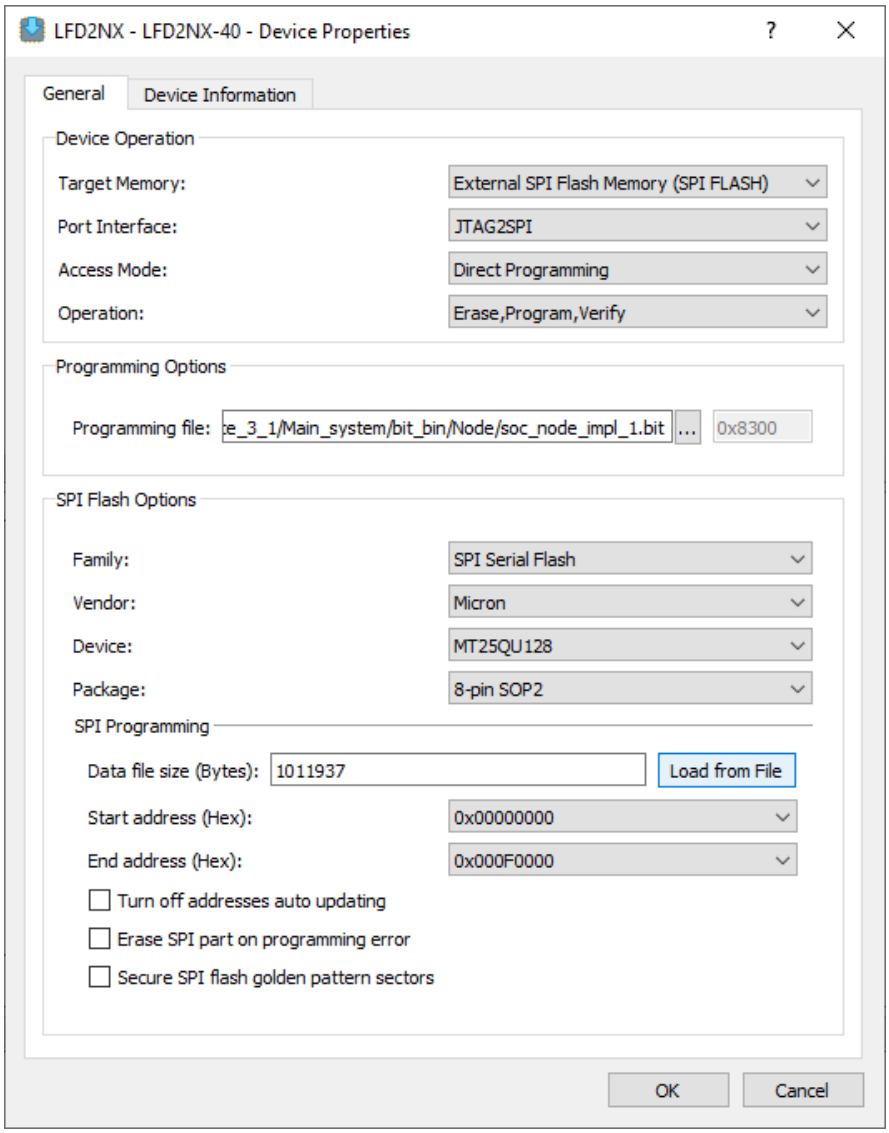

Figure B.20. Radiant Programmer - Bit Stream Flashing Settings (Node System)

   a. Under Programming Options, select the **soc_node_impl_1.bit** bitstream file in the programming file.
   b. Click Load from File to update the Data file size (Bytes) value.
   c. Make sure that the following addresses are correct:
      - Start Address (Hex) – **0x00000000**
      - End Address (Hex) – **0x00100000**

9. Click the Program Device icon or go to **Run > Program Device**.
10. To program the firmware, select the options as shown in Figure B.21.
    a. Under Programming Options, select the **c_node_system_2_0.bin** binary file.
    b. Make sure that the following addresses are correct:
       - Start Address (Hex) – **0x00140000**
       - End Address (Hex) – **0x00220000**

11. Click the **Program Device** icon or go to Run > Program Device.

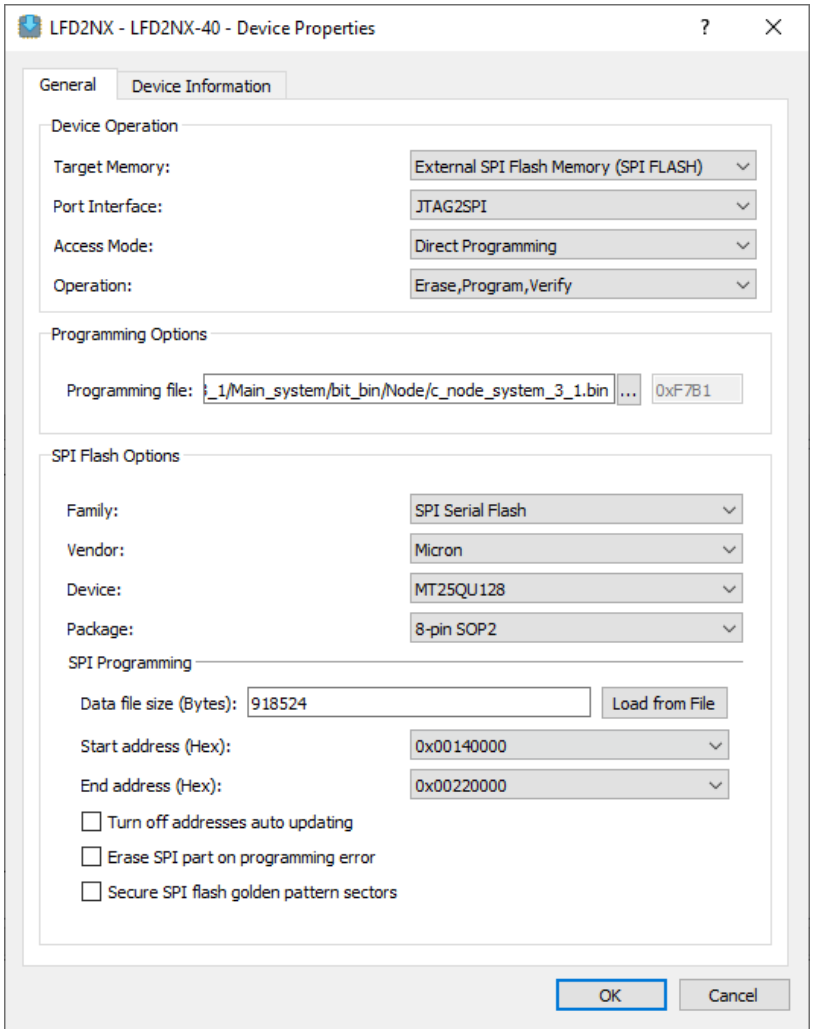

Figure B.21. Radiant Programmer - Binary Flashing Settings (Node System)

Note: After programming the Boards, power cycle the CertusPro NX Versa board as shown in Figure B.22 and each Certus NX Versa board as shown in Figure B.23, and press the system **Reset** button **SW3**.

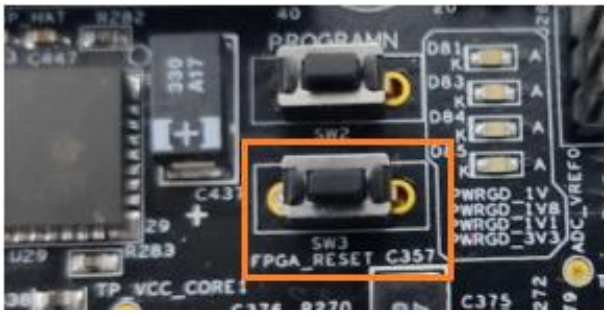

Figure B.22. Reset Button SW3 of CertusPro NX Versa Board

Figure B.23. Button SW3 of Certus NX Versa Board

Appendix C. Programming a Fresh Main System Board

If the user programming the Main System board for the first time, use the following one-time procedure to avoid the firmware booting issue. When the boards are delivered from the foundry, they are programmed in *Quad* mode. But the SPI Flash Controller starts with *Serial* mode (default) for commands and then it selects the quad mode for the read operation.

To program the SPI Flash in Quad mode:

1. Connect the CertusPro-NX Versa board to the PC using a USB cable.
2. Start Radiant Programmer. In the **Getting Started** dialog box, select **Create a new blank project**.

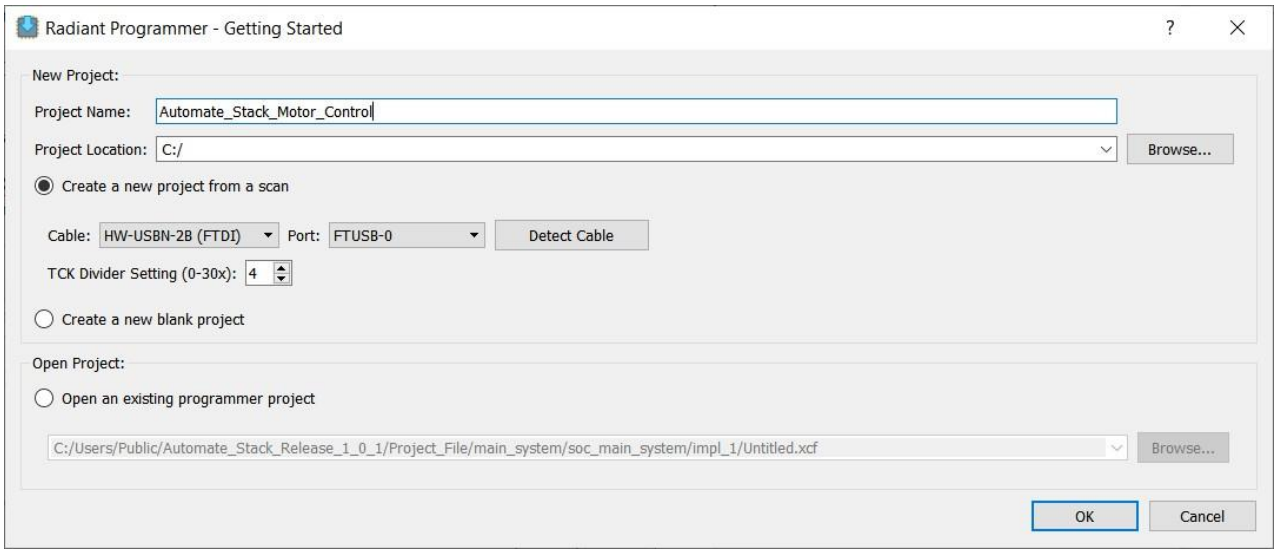

Figure C.1. Radiant Programmer - Default Screen

3. Click **OK**.

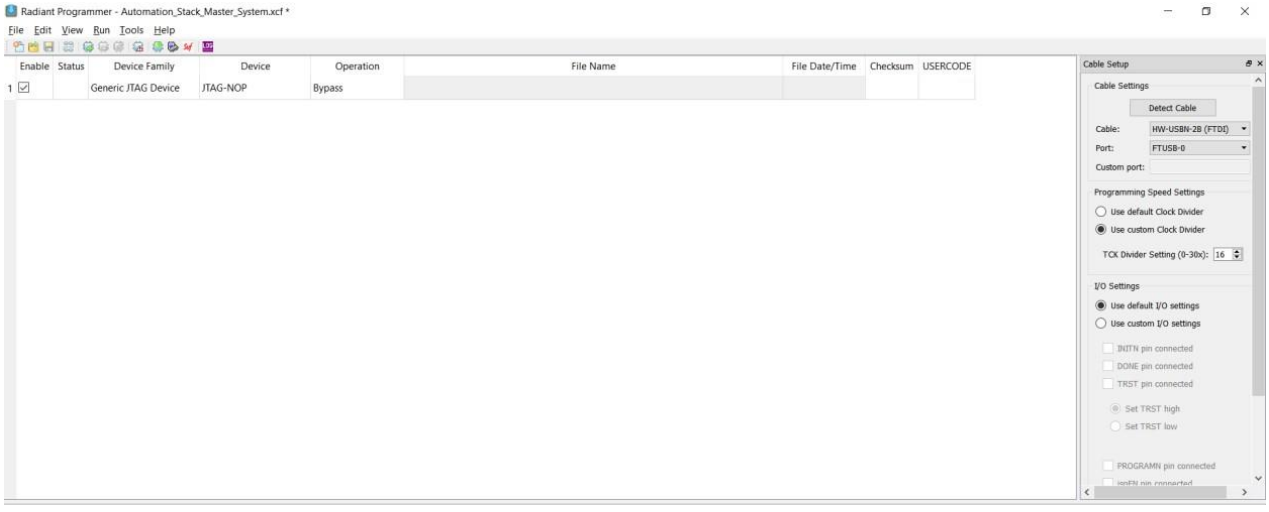

Figure C.2. Radiant Programmer - Initial Project Window

4. In the Radiant Programmer, select **LFCPNX** for **Device Family** and **LFCPNX-100** for **Device** or detect automatically as shown in Figure C.3.

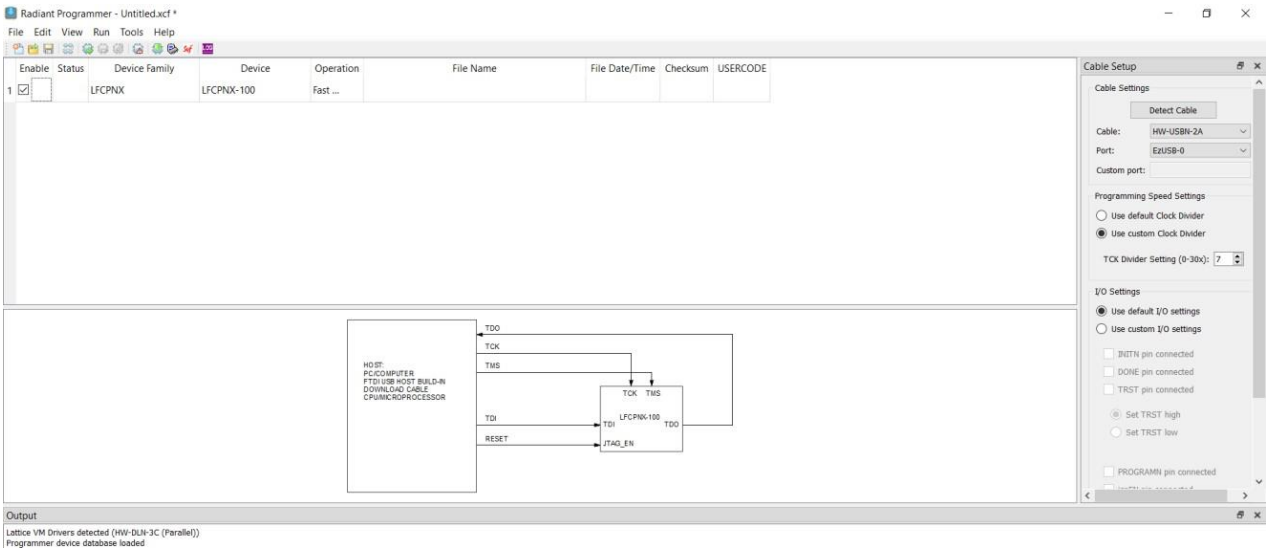

Figure C.3. Radiant Programmer - Device Selection

5. Right-click and select **Device Properties**.

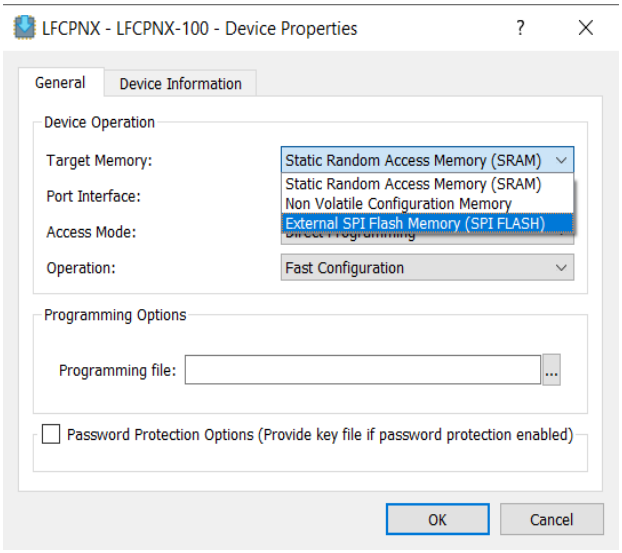

Figure C.4. Radiant Programmer - Device Operation

6. To program the binary file, apply the settings below:
   a. Under Device Operation, select the options below:
      - Target Memory – **External SPI Flash Memory (SPI FLASH)**
      - Port Interface – **JTAG2SPI**
      - Access Mode – **Direct Programming**
      - Operation – **Erase, Program, Verify Quad 1**
   b. Under SPI Flash Options, select the options below:
      - Family – **SPI Serial Flash**
      - Vendor – **Macronix**
      - Device – **MX25L51245G**
      - Package – **8-land WSON**

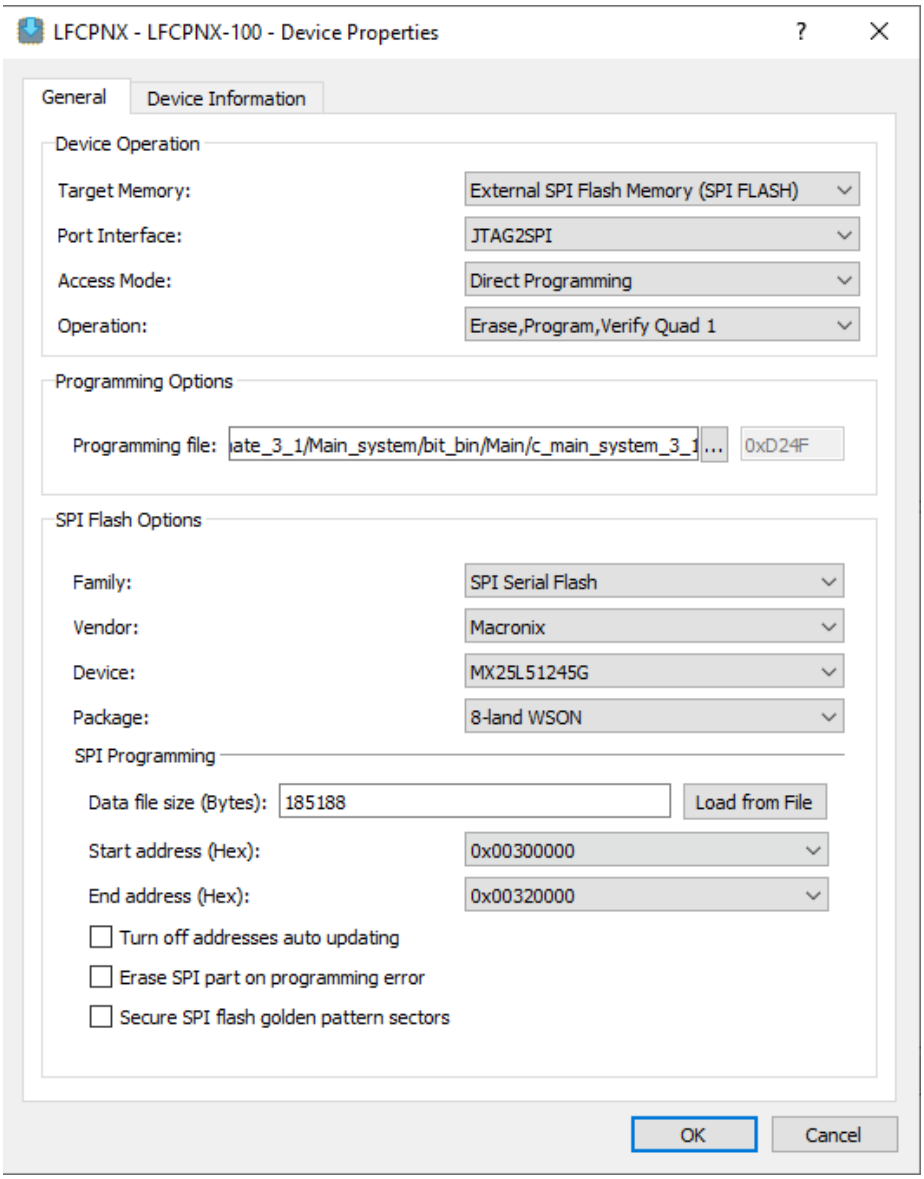

Figure C.5. Radiant Programmer - Quad Mode Programming

7. Under Programming Options, select the *c_main_system_3_1.bin* binary file.
   a. Make sure that the following addresses are correct:
      - Start Address (Hex): **0x00300000**
      - End Address (Hex): **0x00380000**
8. Click **OK** and then click the Program Device icon or the menu item **Run ->Program Device**. After that power cycle the board.
9. To program the bitstream file, select the options as shown in Figure C.5.
10. Under Programming Options, select the *soc_main_system_impl_1.bit* bitstream file in the Programming file.

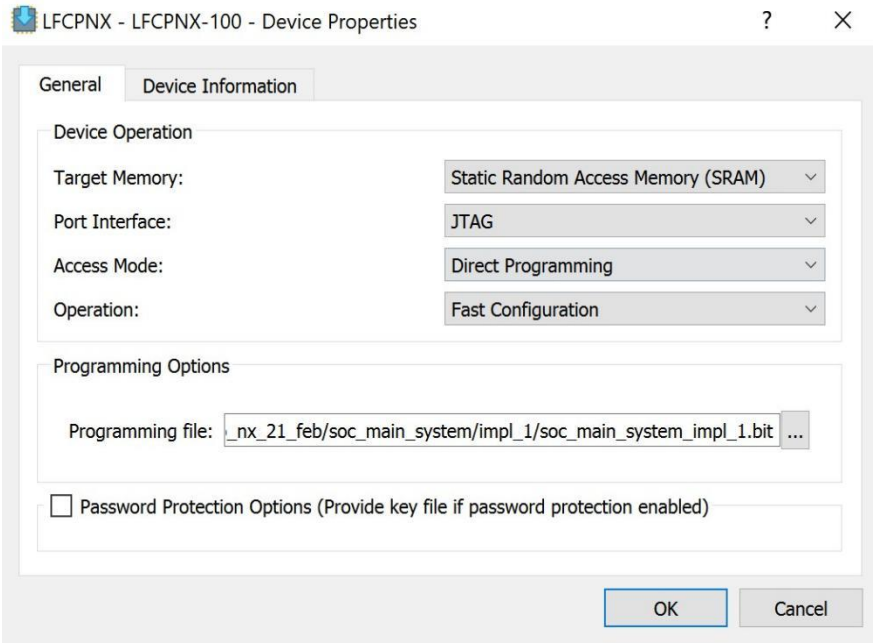

Figure C.6. Radiant Programmer - Device Properties

11. Click the Program Device icon or go to **Run > Program Device**.
12. Apply the settings below to erase the SPI Flash:
    a. Under Device Operation, select the options below:
       - Target Memory – **External SPI Flash Memory (SPI FLASH)**
       - Port Interface – **JTAG2SPI**
       - Access Mode – **Direct Programming**
       - Operation – **Erase All**
    b. Under SPI Flash Options, select the options below:
       - Family – **SPI Serial Flash**
       - Vendor – **Macronix**
       - Device – **MX25L51245G**
       - Package – **8-land WSON**

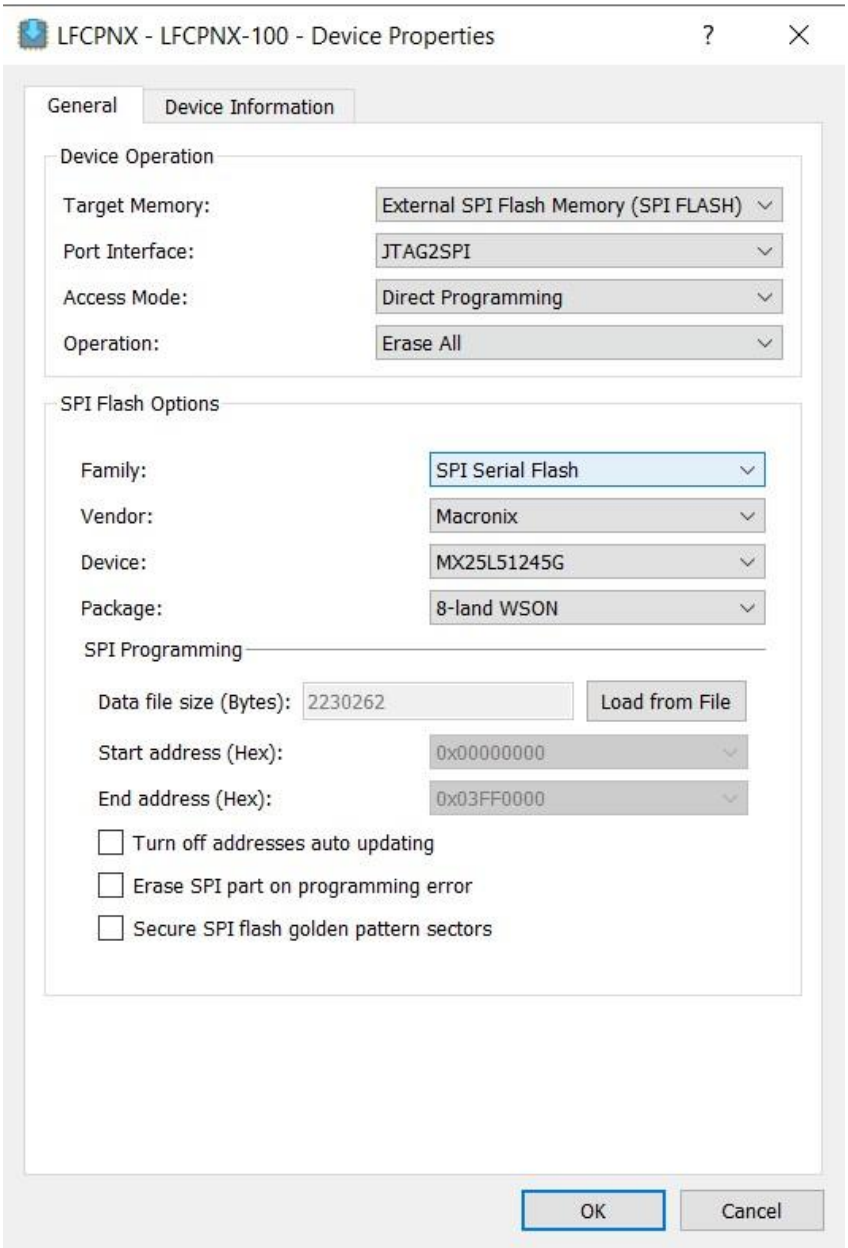

Figure C.7. Radiant Programmer - Erase All

13. Click the Program Device icon or go to **Run > Program Device**.
14. Power cycle the CertusPro NX Versa Board.

    Note: This is a one-time process that must be followed for every new board.

Appendix D. Checking Prints with MCS Manual and Automatic Booting

D.1. Checking the Primary Main System Standalone Project Prints

To check the primary main system standalone project prints:

1. Connect the UART Cable between Host PC to CertusPro-NX Board.
2. Check the Status of UART Cable in Device Manager.
3. Go to the Port section of Device Manager.
4. Check Connection status is showing or not.
5. Program the following bitstream and binary files in the main system. To program the files, refer to Appendix B. Programming the Automate Stack on Respective Flash.
   a. Program the Primary Main System CRC binary file: **c_main_system_3_1_cnnCrc.bin**.
   b. Program the Primary Main System bitstream file: **soc_main_system_3_1_impl_1.bit**.
6. After programming the Primary binary and bitstream file, check the log messages on the serial terminal.
7. Open the Propel terminal.
8. Double-click terminal button in the bottom of Propel SDK window as shown in Figure D.1. The Launch Terminal window appears as shown in Figure D.2.

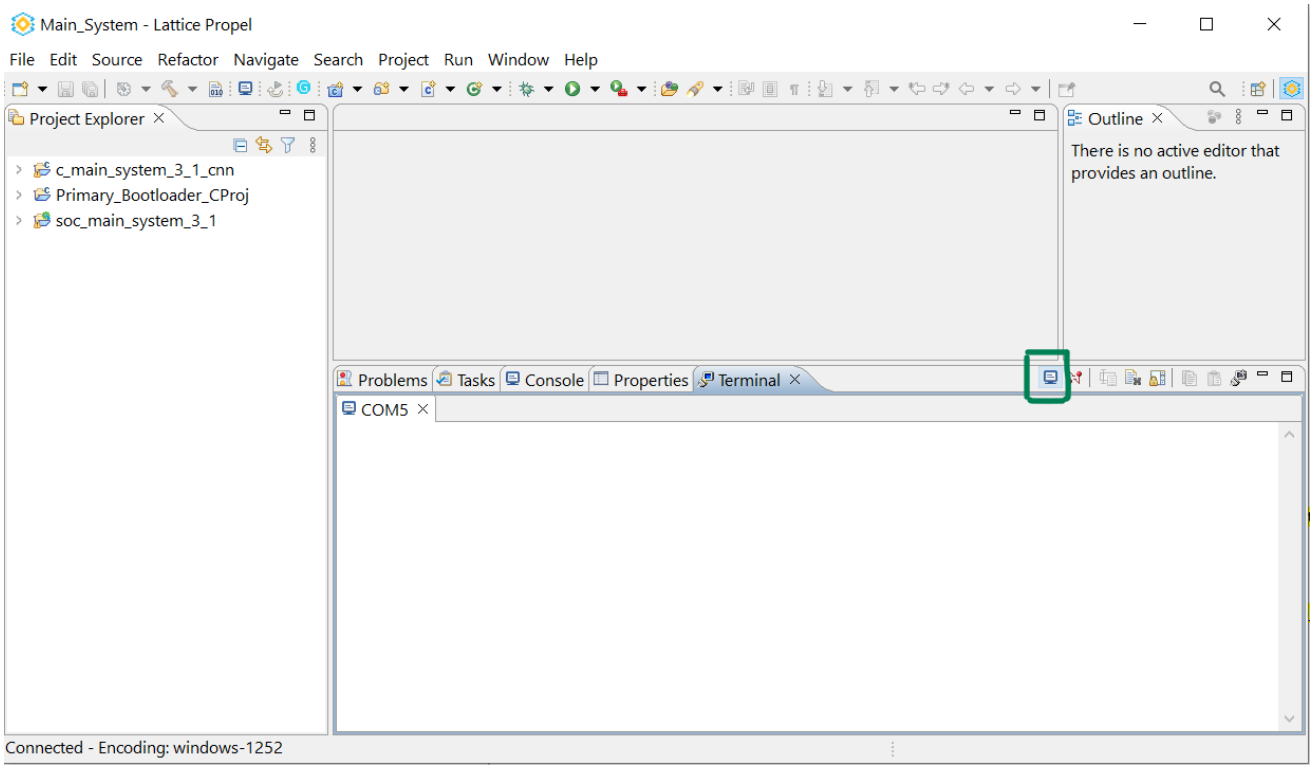

Figure D.1. Open Propel Terminal

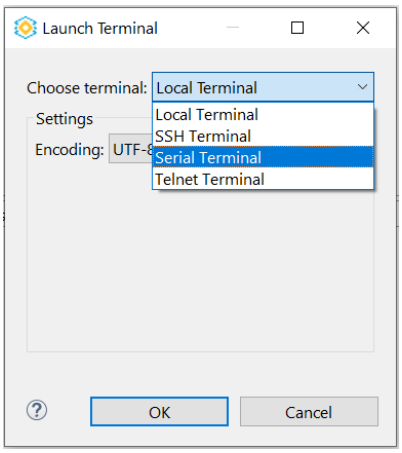

Figure D.2. Propel Terminal Selection Window

9. Choose terminal: Serial terminal as shown in Figure D.3.
10. Select the right serial port, drop-down the serial port, and select the last COM.
11. Select the Baud rate **115200**.
12. Click **OK**.

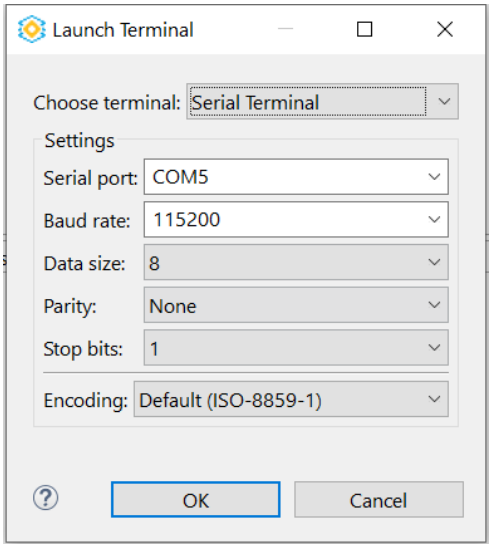

Figure D.3. Propel Terminal - Com and Baud Rate Selection

13. If you are not receiving any packets, press the **SW3** reset button on the board or follow the below steps:
    a. Press the Windows + R button simultaneously.
    b. Type the command *ncpa.cpl* and press **Enter**.
    c. Right-click on **Ethernet** and click **Disable**.
    d. Right-click again and click **Enable**.
14. Wait for 40 seconds. The print is now available in the log section.

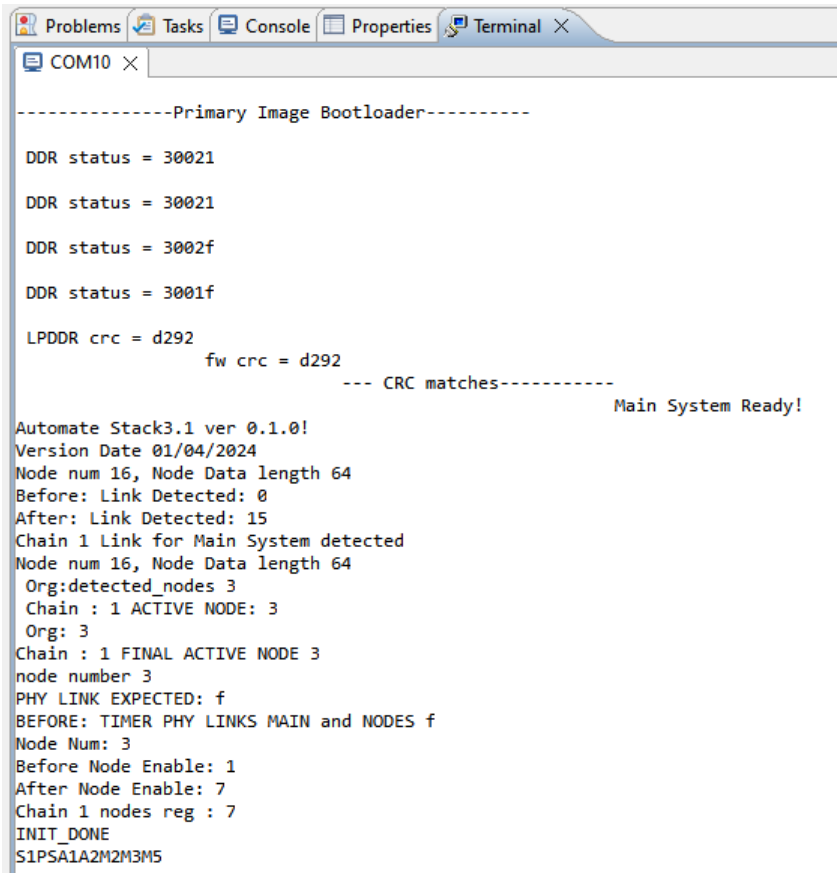

Figure D.4. Primary Main System Propel Prints

15. Press the Reset SW3 button on the CertusPro NX Board to restart the prints.

D.2. Checking the Golden Main System Standalone Project Prints

To check the golden main system standalone project prints:

1. Connect the UART Cable between Host PC to CertusPro-NX board.
2. Check the Status of UART Cable in Device Manager.
3. Go to the Port section of Device Manager.
4. Check Connection status is showing or not.
5. Program the Golden bitstream and binary files in the main system. To program the files, refer to Appendix B. Programming the Automate Stack on Respective Flash.
   a. Program the Golden Main System CRC binary file: **Golden_AppCrc*.bin***
   b. Program the Golden Main System bitstream file: **soc_main_system_impl_1.bit**
6. After programming the Golden binary and bitstream file, follow the below steps to check the prints on propel terminal.
   a. Open the Propel terminal. Refer to steps 8 to 16 of the D.1. Checking the Primary Main System Standalone Project Prints section. The print is available in the log section.
7. While waiting for ethernet packet, you need to send the packet sender udp_rx command.
   a. Open the packet sender tool.
   b. In the top section, create a UDP packet.

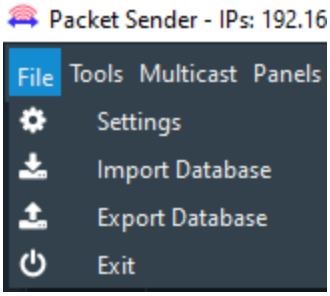

Figure D.5. Packet Sender Application: Load file

c. Provide name of packet, ASCII characters of the packet to be send, Address is 192.168.1.4 (main system address), port is 1486 as shown in Figure D.6.

d. Click the **Send** button to send udp_rx packet.

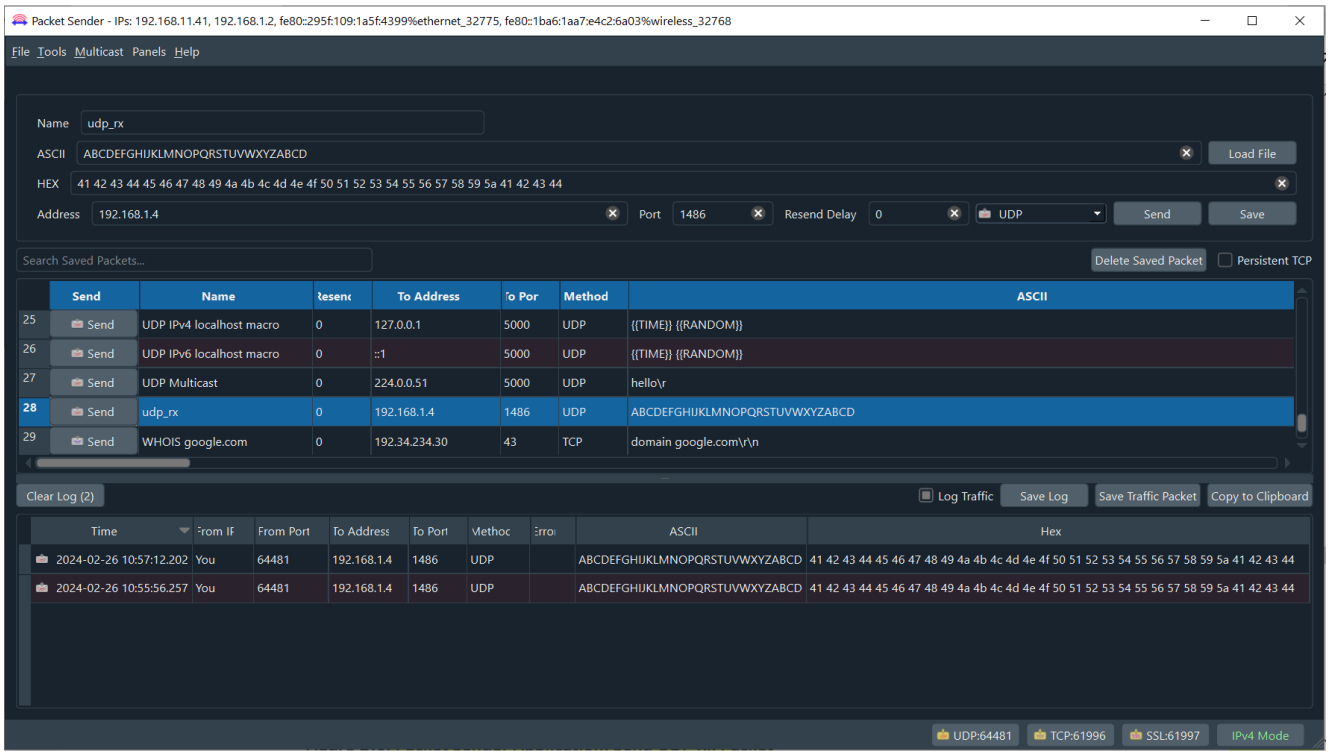

Figure D.6. Packet Sender Application: Send UDP Rx Packet

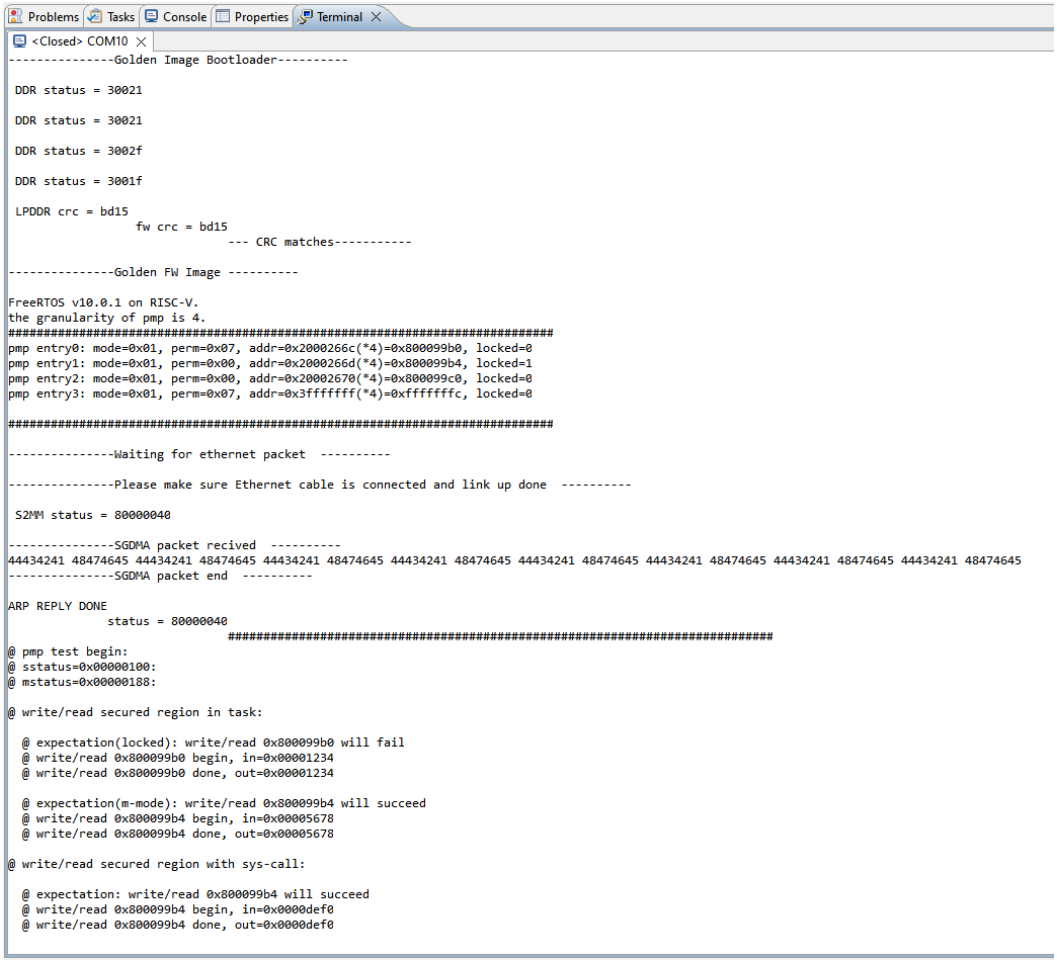

Figure D.7. Golden Main System Propel Prints

8. Press the **Reset** button on the CertusPro NX board to restart the prints.

D.3. Checking the Main System Primary_Golden MCS with Manual Booting Prints

1. Connect the UART Cable between Host PC to CertusPro NX board.
2. Check the Status of UART Cable in Device Manager.
3. Go to the **Port** section of Device Manager.
4. Check the connection status if it is showing or not.
5. The following binaries and MCS are needed to be program, to program the files, refer to Appendix B. Programming the Automate Stack on Respective Flash:
   a. Program the Primary Main System CRC binary file: **Primary_AppCrc*.bin***
   b. Program the Golden Main System CRC binary file: **Golden_AppCrc*.bin***
   c. Program the MCS file. To generate the MCS file, refer to the Appendix D of the Automate Stack 3.1 Reference Design (FPGA-RD-02284).
6. After programming the primary binary, golden binary, and MCS file, follow the steps below to check the prints on the Propel terminal.

a. Open the Propel terminal. Refer to steps 8 to 16 of the D.1. Checking the Primary Main System Standalone Project Prints section.

b. Wait for 40 seconds for the primary system. The print is available in the log section.

7. To send the udp_rx command for Golden System, refer to step 9 of the D.2. Checking the Golden Main System Standalone Project Prints section.

Figure D.8.Primary CRC Match Main System Propel Prints

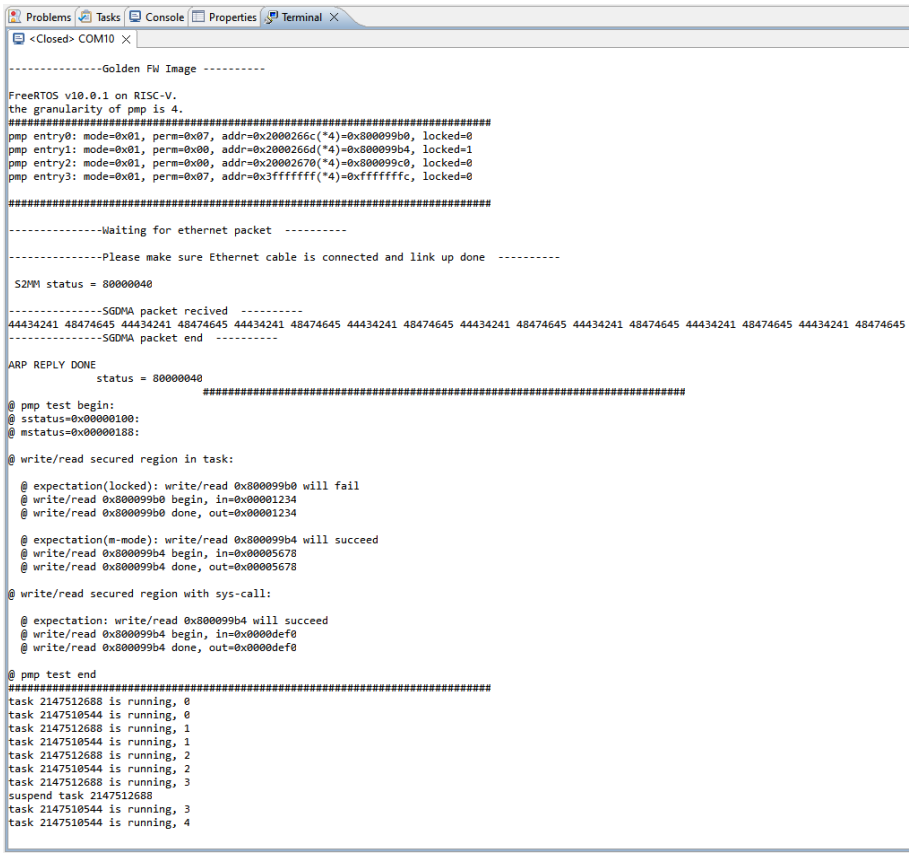

Figure D.9.Golden CRC Match Main System Propel Prints

8. Press the Reset button on the CertusPro NX Board to restart the prints.
9. Press the SW2(program) button on the CertusPro NX Board to switch the prints.

D.4. Check the Main System Primary_Golden MCS with Automatic Booting Prints

1. Connect the UART Cable between Host PC to CertusPro NX Board.
2. Check the Status of UART Cable in Device Manager.
3. Go to the **Port** section of Device Manager.
4. Check the connection status if it is showing or not.
5. After programming the primary binary, golden binary, and MCS file, follow the below steps to check the prints on propel terminal.
   a. The binaries and MCS files need to be programmed. To program the files, refer to Appendix B. Programming the Automate Stack on Respective Flash.
   b. Program the Primary Main System binary file: **Primary_App*.bin***
   c. Program the Golden Main System CRC binary file: **Golden_AppCrc*.bin***
   d. Program the MCS file. To generate the MCS file, refer to Appendix D of the Automate Stack 3.1 Reference Design (FPGA-RD-02284).
6. Open the Propel terminal. Refer to steps 8 to 16 of the D.1. Checking the Primary Main System Standalone Project Prints section.
7. The print is available in the log section.

8. To send the udp_rx command for the Golden system, refer to step 9 of the D.2. Checking the Golden Main System Standalone Project Prints section.

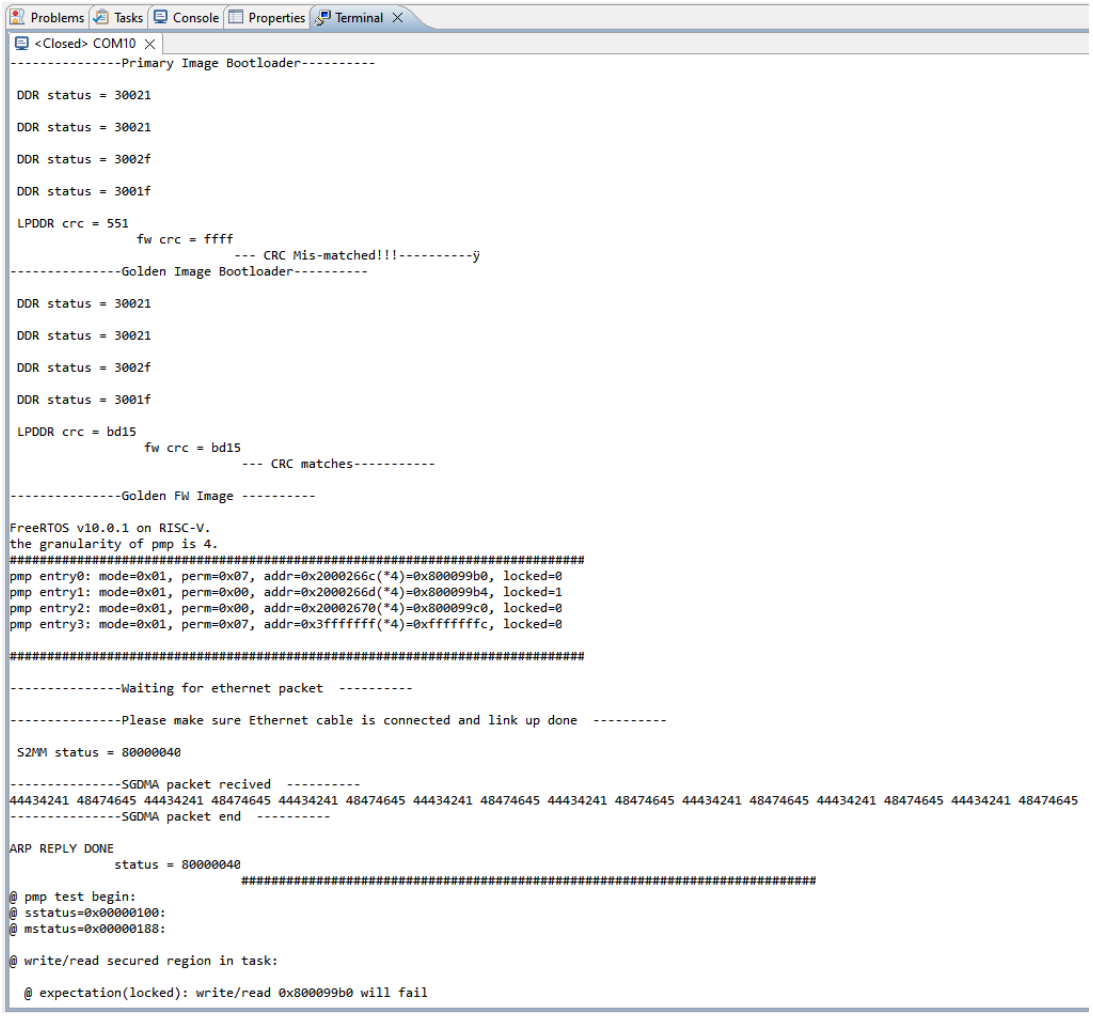

Figure D.10. Primary CRC Mismatch Main System Propel Prints

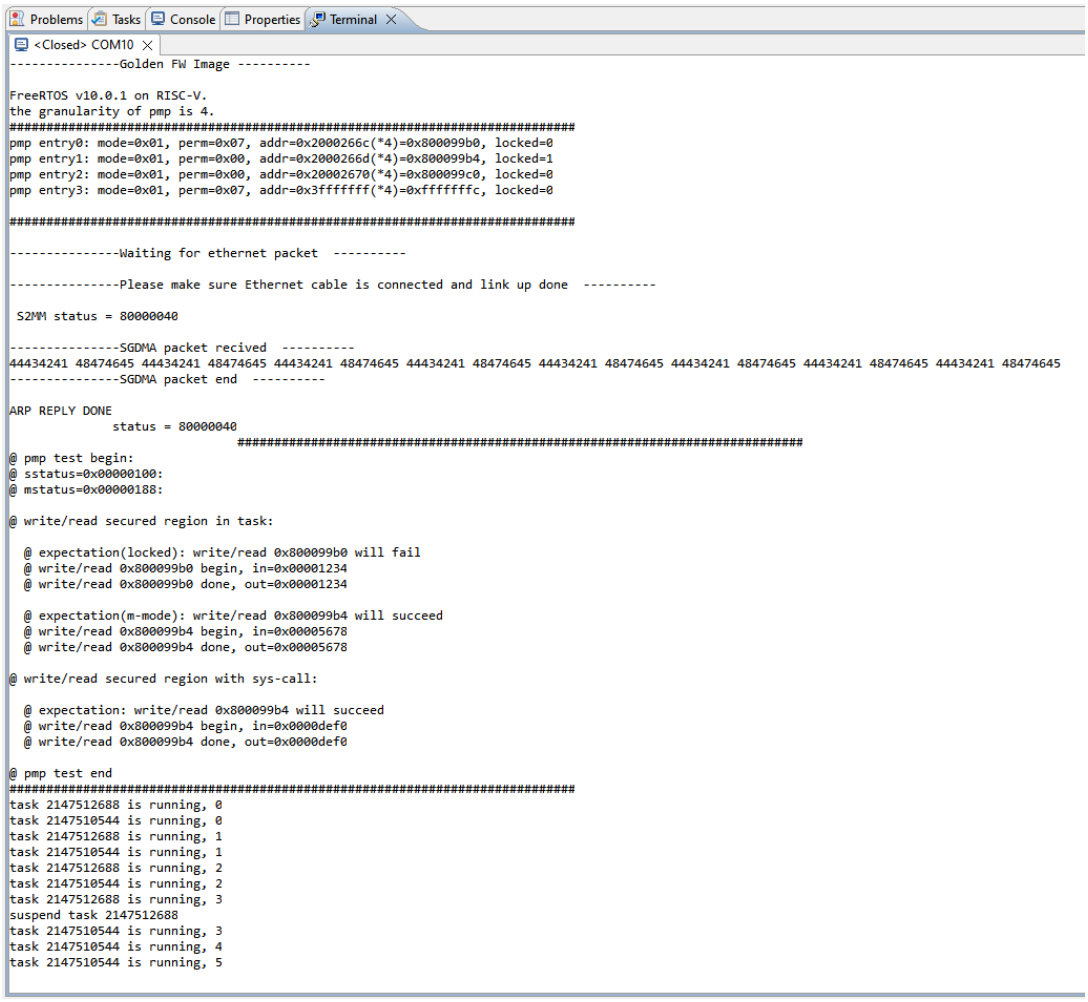

Figure D.11.Golden CRC Match Main System Propel Prints

9. Press the Reset button on the CertusPro NX Board to restart the prints.
10. Press the **SW2** (program) button on the CertusPro NX Board to switch the prints.

D.5 Checking the Main System Primary_Golden MCS with Automatic Booting Prints

1. Connect the UART Cable between Host PC to CertusPro NX board.
2. Check the Status of UART Cable in Device Manager.
3. Go to the Port section of Device Manager.
4. Check Connection status is showing or not.
5. After programming the Primary binary, golden binary, and MCS file, follow the below steps to check the prints on propel terminal.
   a. The binaries and MCS files need to be programmed. To program the files, refer to Appendix B. Programming the Automate Stack on Respective Flash.
   b. Program the Primary Main System CRC binary file: **Primary_AppCrc.bin**
   c. Program the Golden Main System binary file: **Golden_App.bin**
   d. Program the MCS file. To generate the MCS file, refer to Appendix D of the Automate Stack 3.1 Reference Design (FPGA-RD-02284).

6. Open the Propel terminal. Refer to steps 8 to 16 of the D.1. Checking the Primary Main System Standalone Project Prints section.
7. Wait for 40 seconds for the primary system. The print is available in the log section.

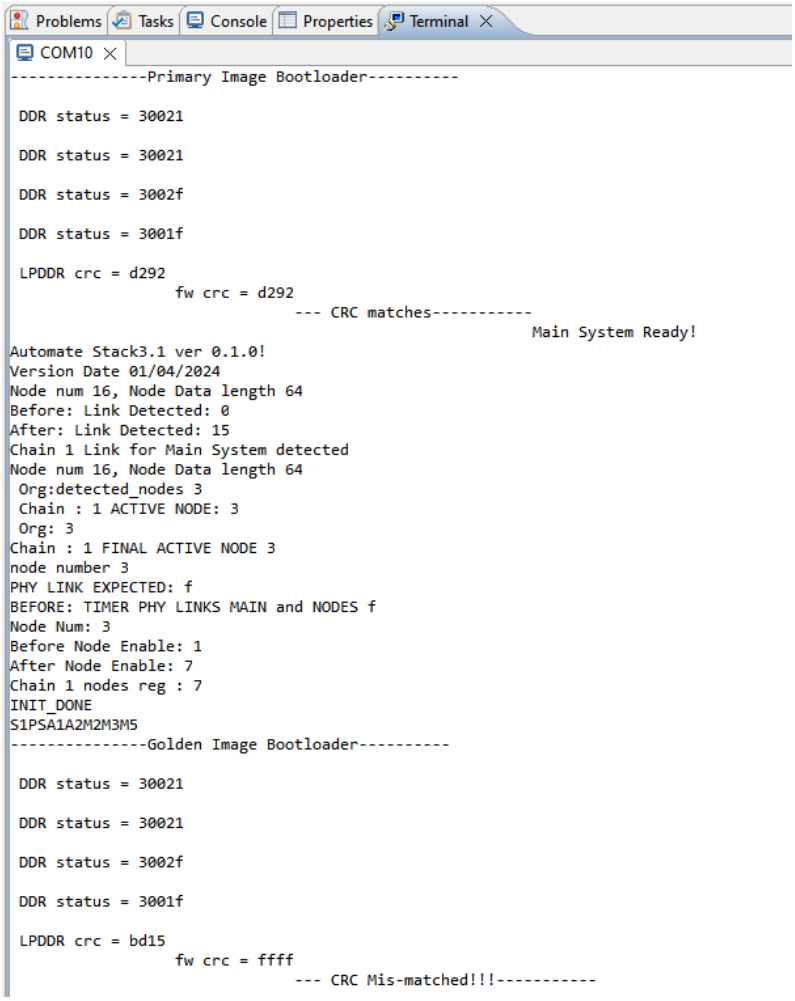

Figure D.12.Primary CRC Match Main System Propel Prints

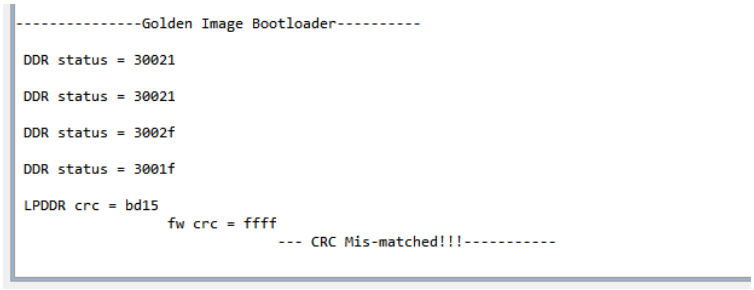

Figure D.13.Golden CRC Mismatch Main System Propel Prints

8. Press the **Reset** button on the CertusPro NX Board to restart the prints.
9. Press the **SW2** (program) button on the CertusPro NX board to switch the prints.

Appendix E. Ethernet Connection

E.1. Connection Between Main system and Laptop/PC

To connect the main system and PC:

1. Set up the following hardware connections, as shown in Figure E.1:
   a. Connect the Ethernet cable between the CertusPro-NX board and the Laptop/PC in the Ethernet port of the laptop and main system.
   b. Connect the 12 V power adapter to the main system board.
2. Connect the UART cable to the main board and laptop or Docklight debugging.

Notes:

1. The firewall must be disabled while doing this activity.
2. .bit and .bin must be programmed in the main system and node systems. Refer to Programming the Automate Stack on Respective FLASH and Programming a Fresh Main System Board sections.

E.2. Ethernet Configuration Setting

1. Open the Network Connections Control Panel from the Start menu or by pressing Windows+R and typing ncpa.cpl.

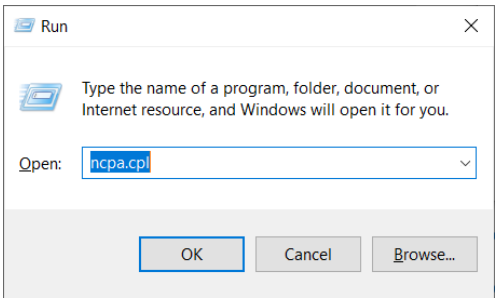

Figure E.1. Run the ncpa.cpl to Open the Network Connections Control Panel

2. Click **Ethernet**.

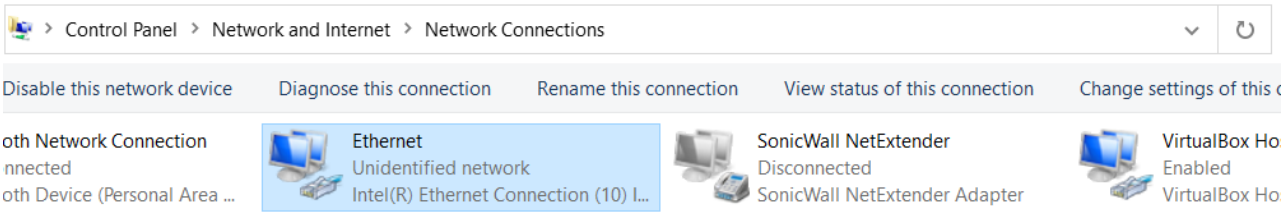

Figure E.2. Ethernet Settings

3. Click **Properties**.

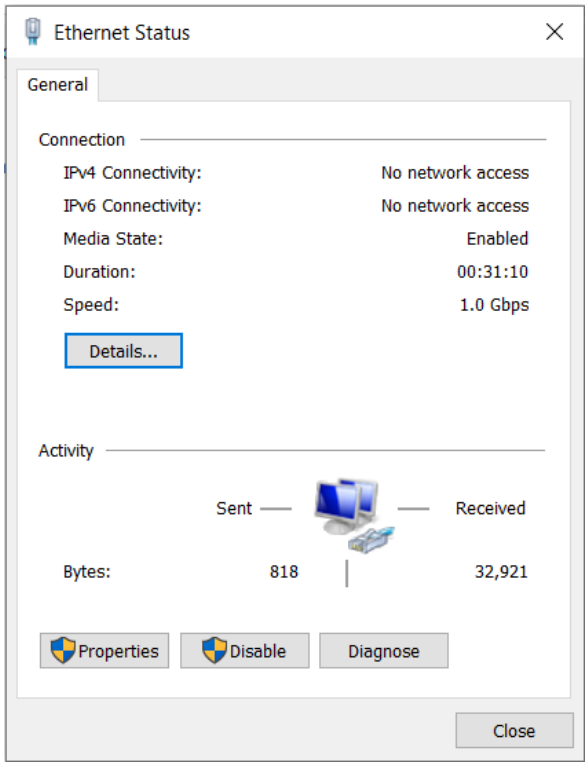

Figure E.3. Ethernet Status – Properties Window

4. Double-click on **Internet Protocol Version**.

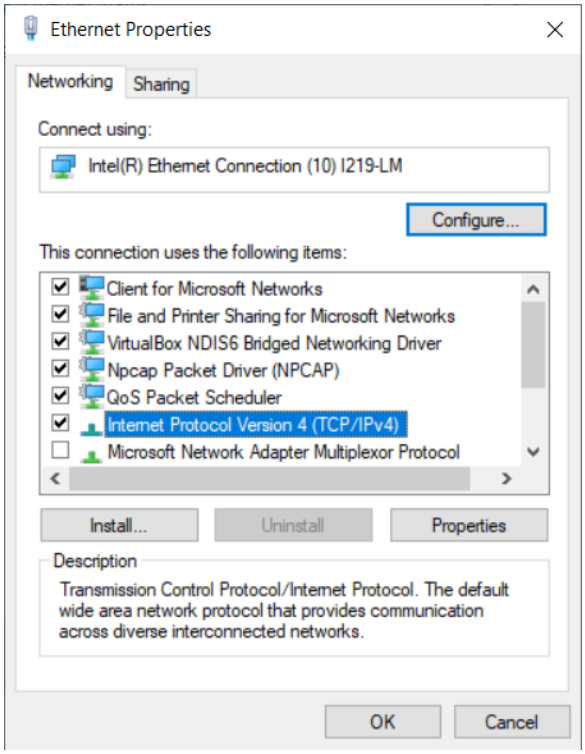

Figure E.4. Ethernet Properties – Internet Protocol Version Window

5. Configure the IP address, subnet mask, and default gateway settings as shown in Figure E.5. Click **OK**.

   Note: If you change the IP address of the PC using Propel Builder, make sure to enter the changed IP address in TCP/IPv4 settings. For more information on how to change the IP addresses of the PC, see section 2.6 of the Automate Stack 3.1 Reference Design (FPGA-RD-02284).

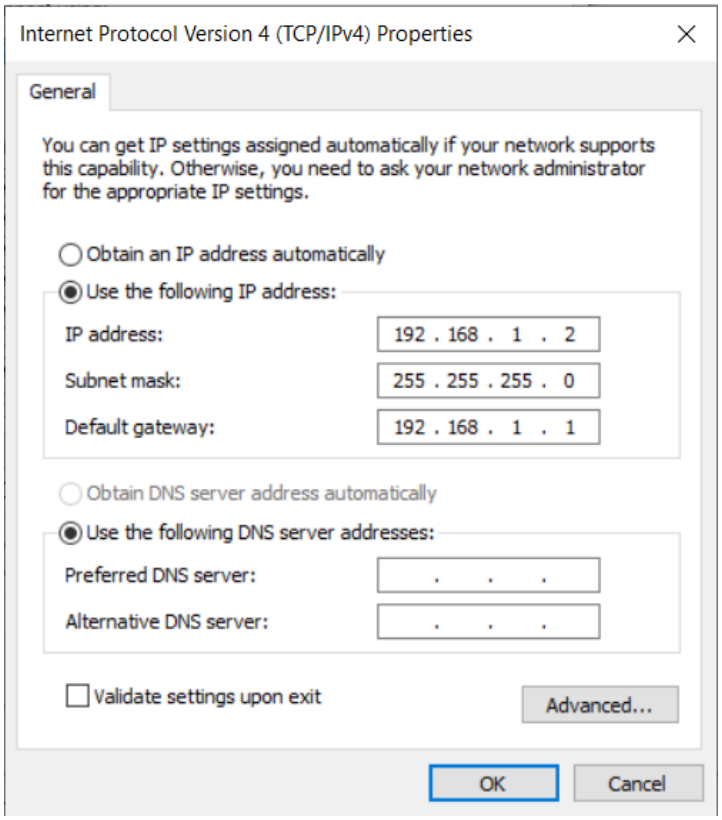

Figure E.5. Configure the Settings

6. Check if Ethernet says *Network cable unplugged*.
7. Right-click on **Ethernet** and select **Disable**.

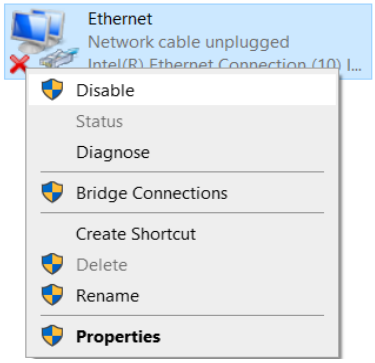

Figure E.6. Disable the Setting

8. Right-click on the **Ethernet** and click **Enable**.

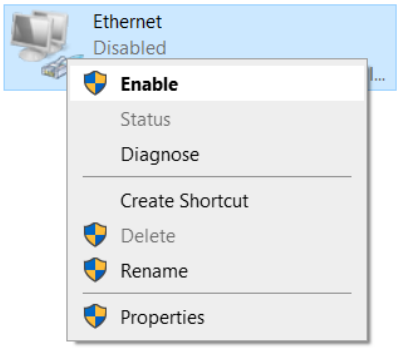

Figure E.7. Enable the Setting

Note: Steps 8 and 9 are required every time the Ethernet cable is disconnected or inserted, or when the main system is power cycled.

E.3. Check the Connection with a Ping

1. Open the command prompt on the PC.
2. Ping the main board using the command: **ping 192.168.1.4**.
3. Make sure the ping is successful. Otherwise, follow the steps in the Connection Between Main system and Laptop/PC and Ethernet Configuration Setting sections again.

```
C:\Users\Pawan>ping 192.168.1.4

Pinging 192.168.1.4 with 32 bytes of data:
Request timed out.
Reply from 192.168.1.4: bytes=32 time<1ms TTL=128
Reply from 192.168.1.4: bytes=32 time<1ms TTL=128
Reply from 192.168.1.4: bytes=32 time<1ms TTL=128

Ping statistics for 192.168.1.4:
    Packets: Sent = 4, Received = 3, Lost = 1 (25% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms

C:\Users\Pawan>
```

Figure E.8. Ping on cmd Terminal

E.4. UDP Packet Transaction

Note: To download the wireshark tool: https://www.wireshark.org/download.html

Wireshark and packet sender both are used for debugging. The packet Sender creates its own payload packets and transmits. Wireshark receives packets and the user can check the data (payload), IP address, and port number.

Packet Sender is also used separately for debugging where you can send multiple commands and check the functionality of the main system.

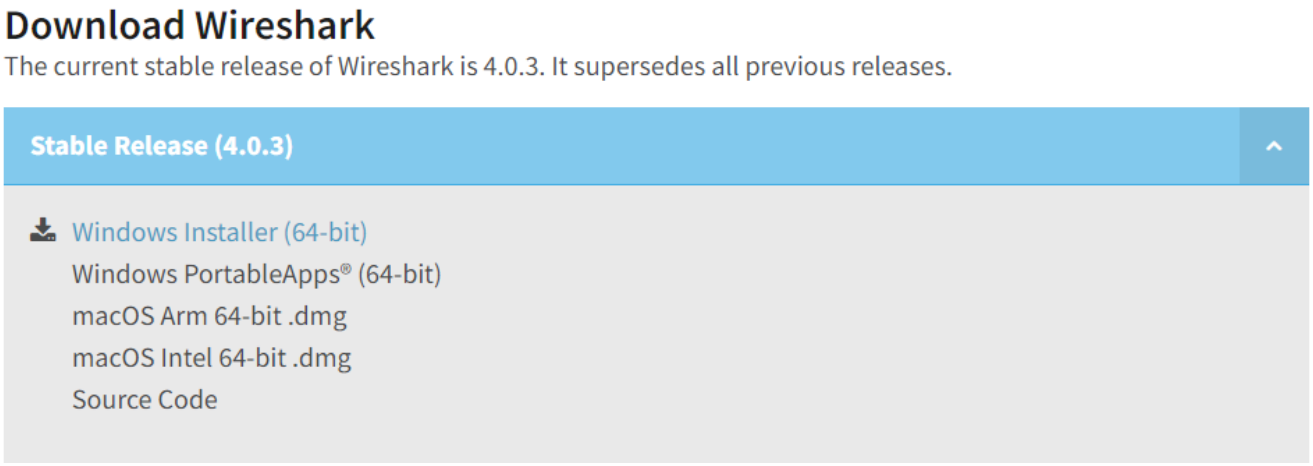

Figure E.9. Wireshark Downloadable Link

Note: To download the Packet Sender tool: https://packetsender.com/download

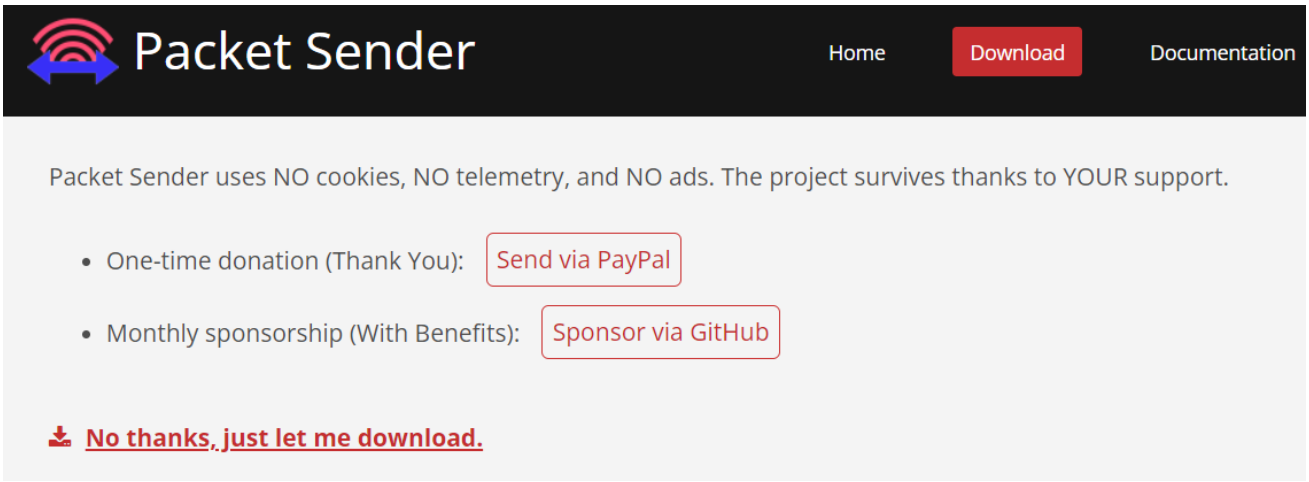

Figure E.10. Packet Sender Downloadable Link

To start the UDP packet transaction using Docklight:

1. Open **Docklight**.
2. Go to **COM**.

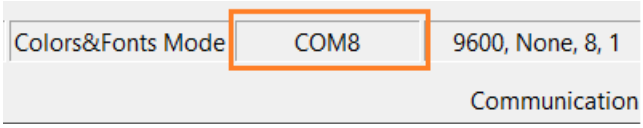

Figure E.11. Docklight - Com Selection

3. Select the right COM port, drop-down the COM and select the last COM.
4. Set the baud rate: **9600**.
5. Click **OK**.

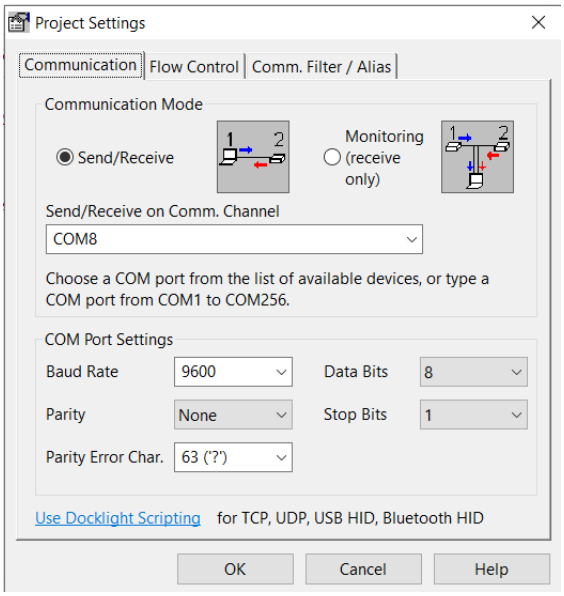

Figure E.12. Docklight - Com and Baud Rate Selection

6. Click **Run**.

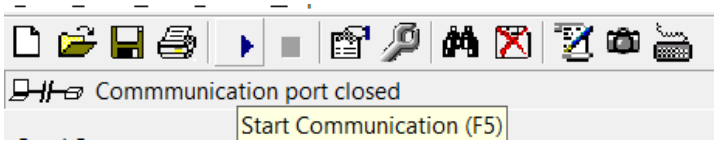

Figure E.13. Docklight - Run (Start Communication)

7. Open the Wireshark tool.
8. Click **Ethernet**.

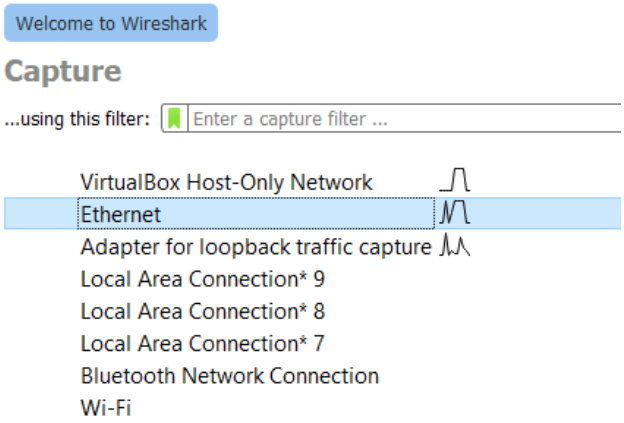

Figure E.14. Wireshark Tool - Ethernet Selection

9. Click on the Run ( ) icon.
10. Open the **Packet Sender** tool.
11. Enter 64 ASCII characters for a total of 64 bytes of data. For example, enter abcdefgh eight times.

12. Click **Save**.

Figure E.15. Packet Sender Tool - Send 64 Bytes of ASCII Characters

13. Click **Send** to send the UDP_PK packet.

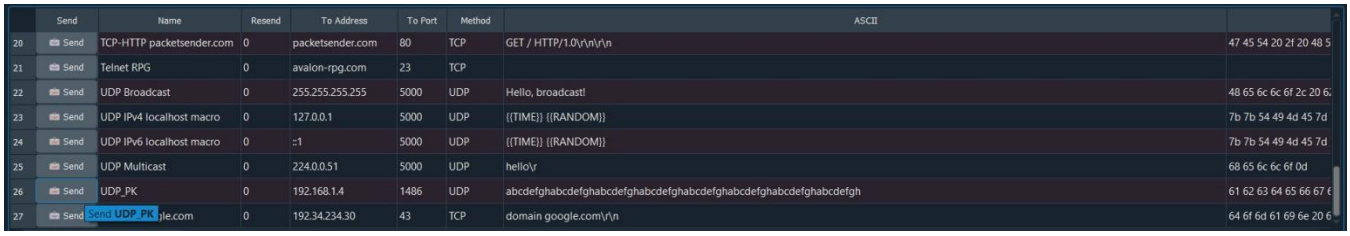

Figure E.16. Packet Sender Tool - Send UDP Packet

14. Check the log.

Figure E.17. Packet Sender Tool - Transaction Log

15. Open Docklight to check the data received.
16. Check if the UDP data is received.

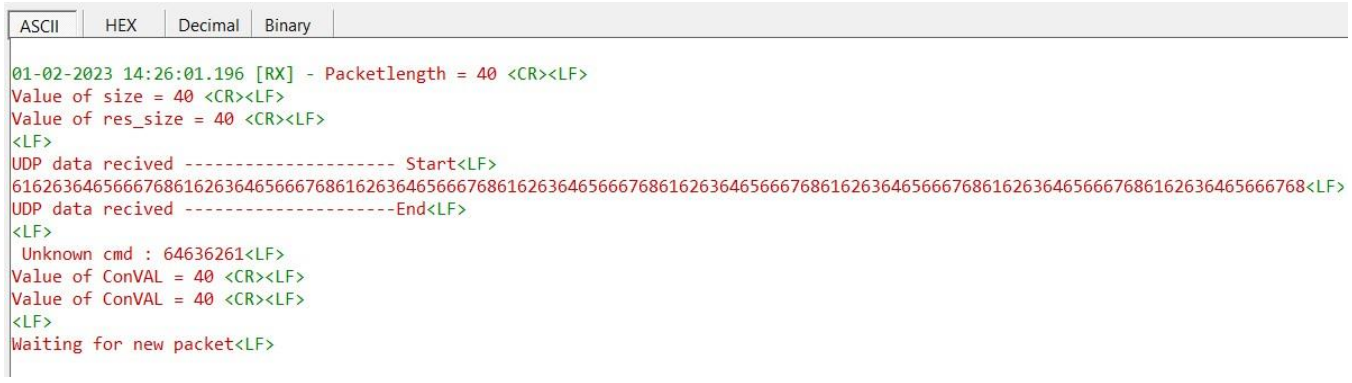

Figure E.18. Docklight - Logs in ASCII

17. Go to the Wireshark tool.
18. Type **udp.port == 1486** on top of the bar, as shown in Figure E.19.

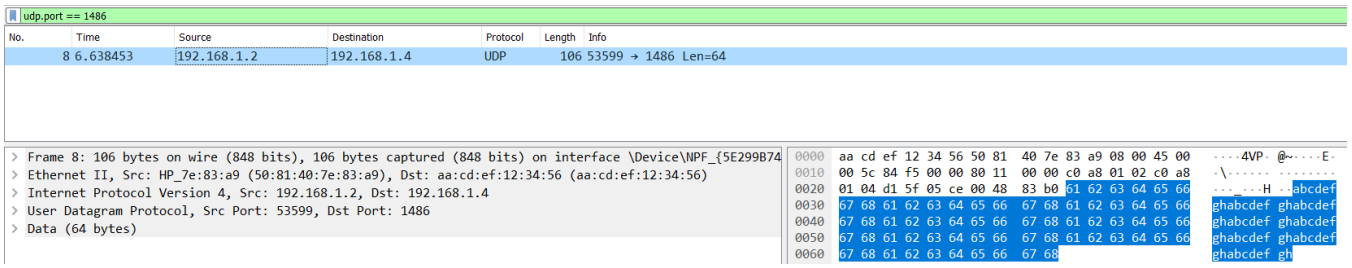

Figure E.19. Wireshark Tool - Write udp.port == 1486

19. Check both the Source and Destination UDP packets.

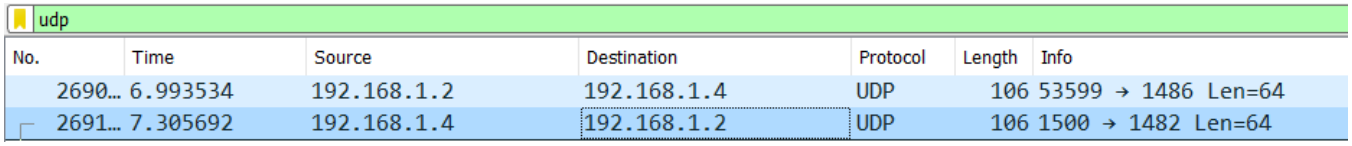

Figure E.20. Source and Destination UDP Packet

20. Click the first UDP transaction to check in detail.

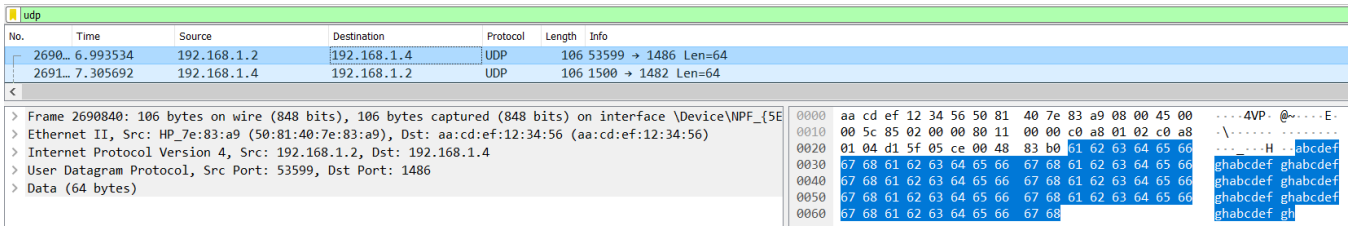

Figure E.21. Wireshark Tool - First UDP Packet

21. Click the second UDP transaction to check in detail.

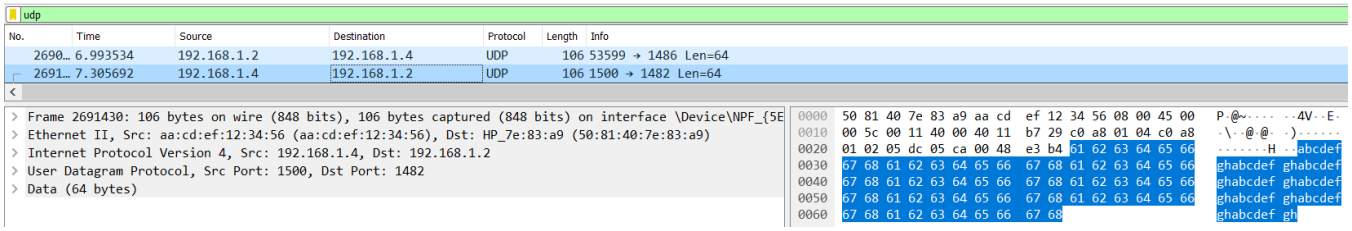

Figure E.22. Wireshark Tool - Second UDP Packet

E.5. Packet Sender Commands

E.5.1. Motor Config Commands

Update Config

```
hexString=f1 01 77 cc 01 64 00 01 4d f4 63 ea 37 18 1e 7c 37 18 1e 00 05 00 00 00 02
00 00 00 07 00 00 00 af 96 04 96 a0 0f 15 26 80 00 40 00 10 00 00 08 80 00 40 00 fa
00 00 08 00 00 00 00 00 00 00 00
```

TestNodeReg_Write_led_OFF

```
hexString=f1 01 77 cc 01 64 00 01 4d f4 e3 10 e4 31 bf 1d e4 2f d9 00 ff ff ff 0f 00
01 00 00 54 40 18 00 ff ff ff ff 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00
```

TestNodeReg_Write_led_ON

```
hexString=f1 01 77 cc 01 64 00 01 4d f4 e3 10 e4 31 bf 1d e4 2f d9 00 ff ff ff 0f 00
01 00 00 54 40 18 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00
```

StartMotor_dashboard

```
hexString=f1 01 77 cc 01 64 00 01 4d f4 63 ea 37 18 1e 7c 37 18 1e 00 12 00 00 00 02
00 00 00 07 00 00 00 78 00 01 00 78 00 01 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00
```

StopMotor_dashboard

```
hexString=f1 01 77 cc 01 64 00 01 4d f4 63 ea 37 18 1e 7c 37 18 1e 00 02 00 00 00 02
00 00 00 07 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00
```

PowerOffMotor_dashboard

```
hexString=f1 01 77 cc 01 64 00 01 4d f4 63 ea 37 18 1e 7c 37 18 1e 00 03 00 00 00 02
00 00 00 07 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00
```

MotorStart_MotorStatusTab

```
hexString=f1 01 77 cc 01 64 00 01 4d f4 63 ea 37 18 1e 7c 37 18 1e 00 01 00 00 00 02
00 00 00 01 00 00 00 78 00 00 00 01 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00
```

MotorStop_MotorStatusTab

```
hexString=f1 01 77 cc 01 64 00 01 4d f4 63 ea 37 18 1e 7c 37 18 1e 00 02 00 00 00 02
00 00 00 01 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00
```

PowerOffMotor_StatusTab

```
hexString=f1 01 77 cc 01 64 00 01 4d f4 63 ea 37 18 1e 7c 37 18 1e 00 03 00 00 00 02
00 00 00 01 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00
```

Motor_rpm_500_MotorStatusTab

```
hexString=f1 01 77 cc 01 64 00 01 4d f4 63 ea 37 18 1e 7c 37 18 1e 00 01 00 00 00 02
00 00 00 01 00 00 00 f4 01 00 00 01 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00
```

Motor_rpm_1000_MotorStatusTab

```
hexString=f1 01 77 cc 01 64 00 01 4d f4 63 ea 37 18 1e 7c 37 18 1e 00 01 00 00 00 02
00 00 00 01 00 00 00 e8 03 00 00 01 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00
```

Motor_rpm_1400_MotorStatusTab

```
hexString=f1 01 77 cc 01 64 00 01 4d f4 63 ea 37 18 1e 7c 37 18 1e 00 01 00 00 00 02
00 00 00 01 00 00 00 78 05 00 00 01 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00
```

Motor_rpm_1800_MotorStatusTab

```
hexString=f1 01 77 cc 01 64 00 01 4d f4 63 ea 37 18 1e 7c 37 18 1e 00 01 00 00 00 02
00 00 00 01 00 00 00 08 07 00 00 01 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00
```

Note: To select a different node, change the highlighted bit in Motor Start Motor status command.

- 01 – First node
- 02 – Second node
- 04 – Third node and so on
- 07 – All Nodes

E.5.2. Recommend Command Sequences to Test Basic Functionality

Basic LED OFF/ON Command

- Send LED OFF Command
- Send LED ON command

Motor Config Test

- Send the Update config command

Motor Start from Dashboard Test

- Send the start motor dashboard command
- Send the Motor RPM 500 Motor Status command
- Send the Motor RPM 1000 Motor Status command
- Send the Motor RPM 1400 Motor Status command
- Send the Motor RPM 1800 Motor Status command
- Send the Stop motor dashboard command
- Send the Power Off motor dashboard command

Motor start from Motor Status Test

- Send the Motor Start Motor status command
- Send the Motor RPM 500 Motor Status command
- Send the Motor RPM 1000 Motor Status command
- Send the Motor RPM 1400 Motor Status command
- Send the Motor RPM 1800 Motor Status command
- Send the Motor Stop Motor status command
- Send the Power Off Motor Status command

Appendix F. Troubleshooting

F.1. Automate 3.1 User Interface Troubleshooting

F.1.1. Motor Configuration Values are 0

Problem Description:

Motor configuration values are all 0.

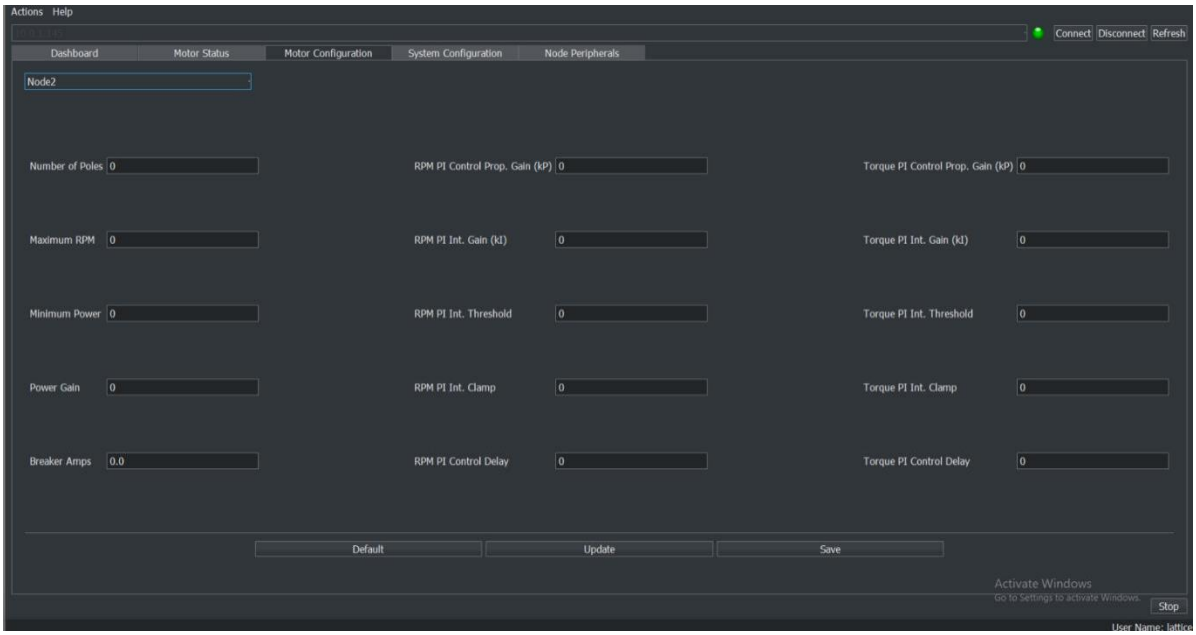

Figure F.1. Motor Configuration Values are All 0

Solution:

1. In the Automate 3.1 user interface, click the **Motor Configuration** tab.
2. From the Select drop-down menu, either Select **All** or select an individual node.

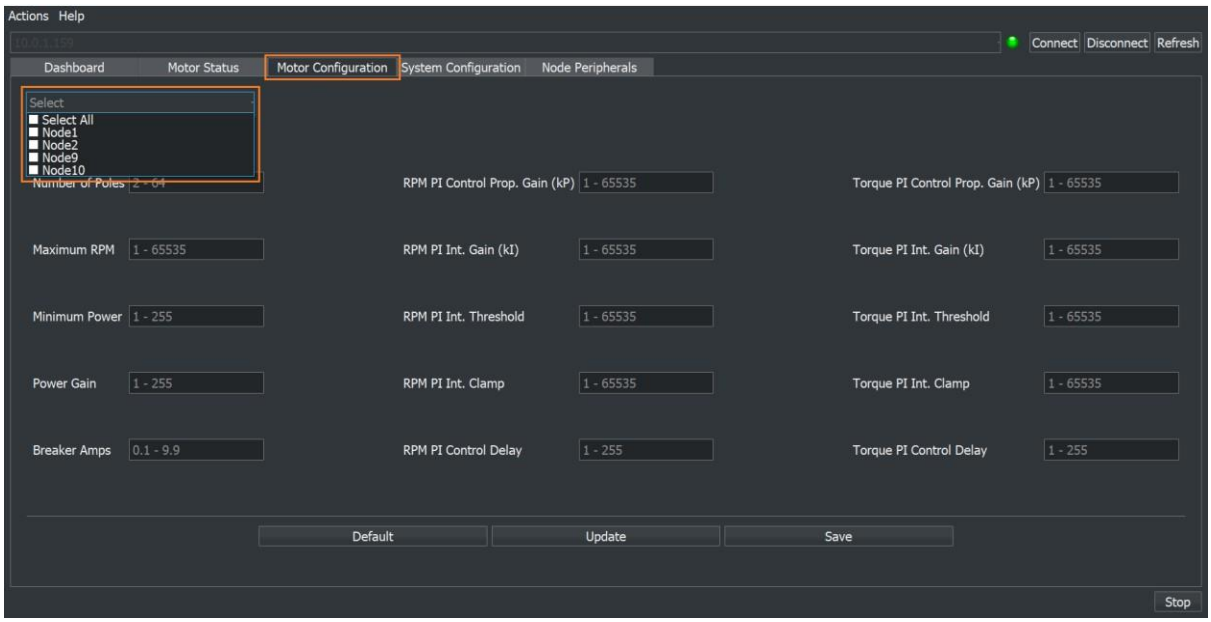

Figure F.2. Motor Configuration: Node Selection

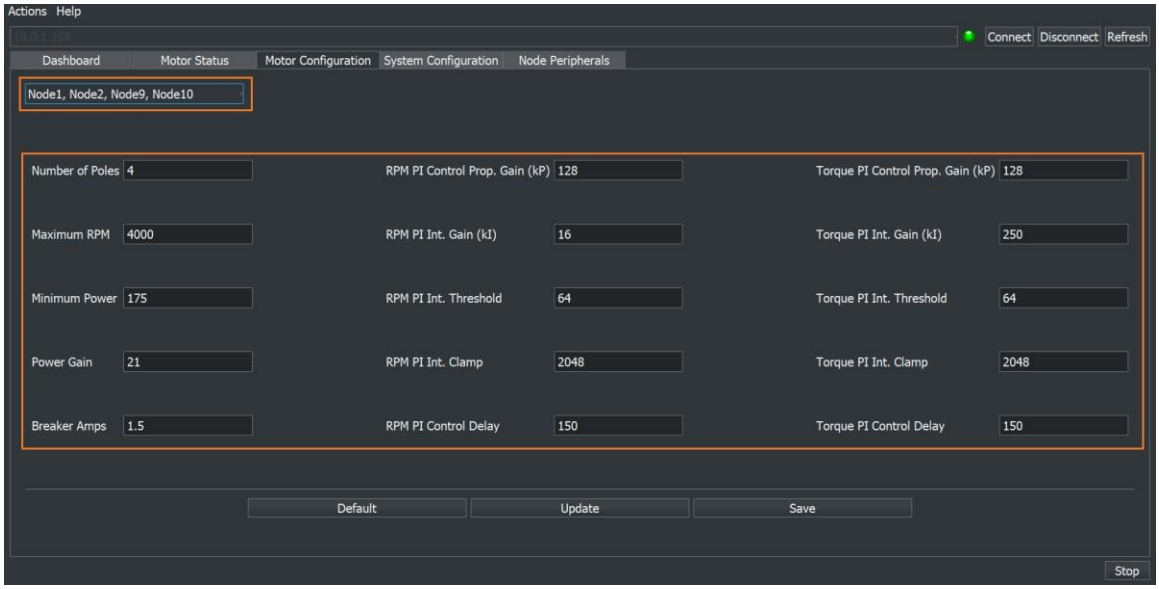

Figure F.3. Motor Configuration: All Nodes Selected

3. Change the Motor Configuration fields to the following default values:
   - RPI PI Control Prop. Gain(kP): **90**
   - RPI PI Int. Gain(kI): **10**
   - RPI PI Control delay: **200**
   - Minimum Power: **190 - 200**
   - Power gain: **22 - 24**
   - Breaker Amps: **1.8 - 2.0**

   Note: These values are for the GB-42 BLS 24 V, 5000 RPM motors only. Choose appropriate values for the motors that you are using.

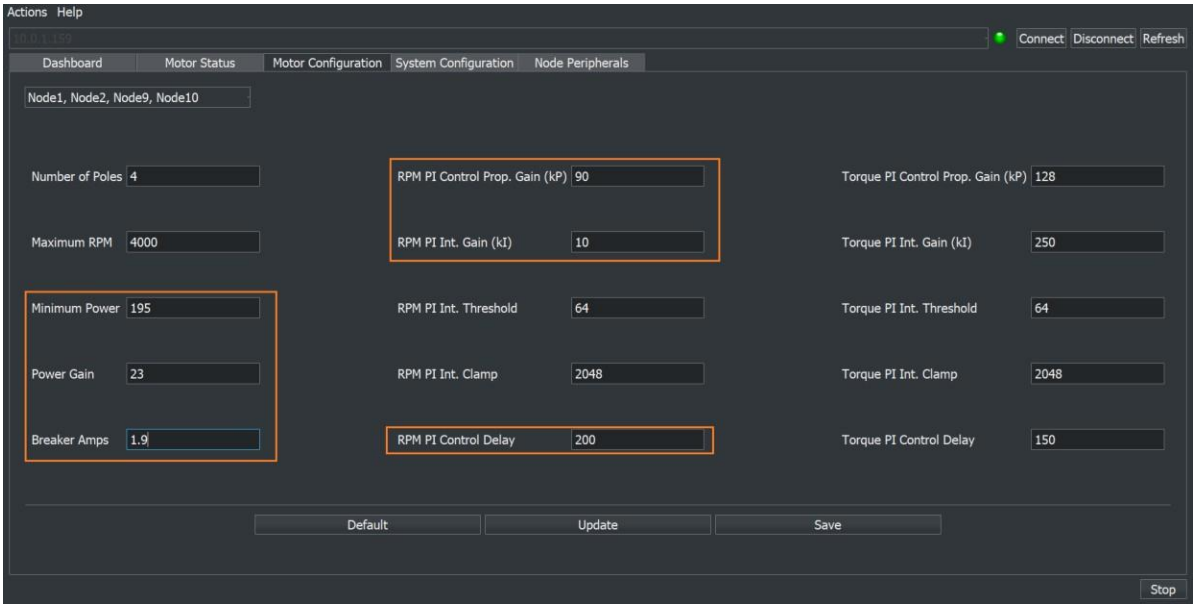

Figure F.4. Motor Configuration Value Changes

4. Click **Default**.
5. Click **Update**.
6. Click **Yes** on the Pop-up confirmation message.

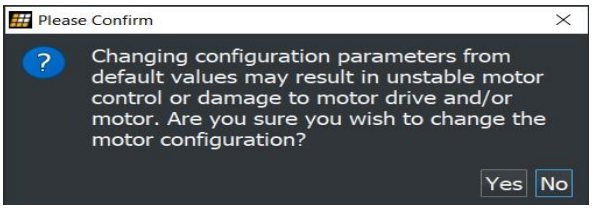

Figure F.5. Application Software - Motor Configuration: Warning Message

7. Enter the Credentials in the authentication pop-up and click on **Login**.
   - **Username: lattice**
   - **Password: lattice**

Figure F.6. Application Software - Motor Configuration: Authentication Pop-up

8. Click **OK** on the pop-up message stating that successfully updated the configuration of the selected node appears.

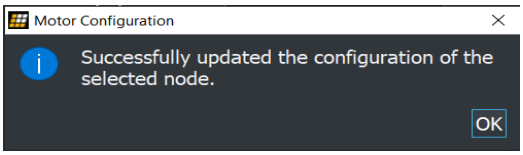

Figure F.7. Application Software - Motor Configuration: Update Configuration

9. If a success pop-up message does not appear, or if a message appears stating that it failed to configure the node, restart the system.

F.1.2. Motor Status Tab Failure (RPM Lock Status, Voltage Status and Drive Status are Red)

Problem Description: The motor cannot be stopped from the user interface, and/or RPM Lock Status, Voltage Status, and Drive Status indicators are red, as shown in Figure F.8.

Solution: Power everything off and restart the setup.

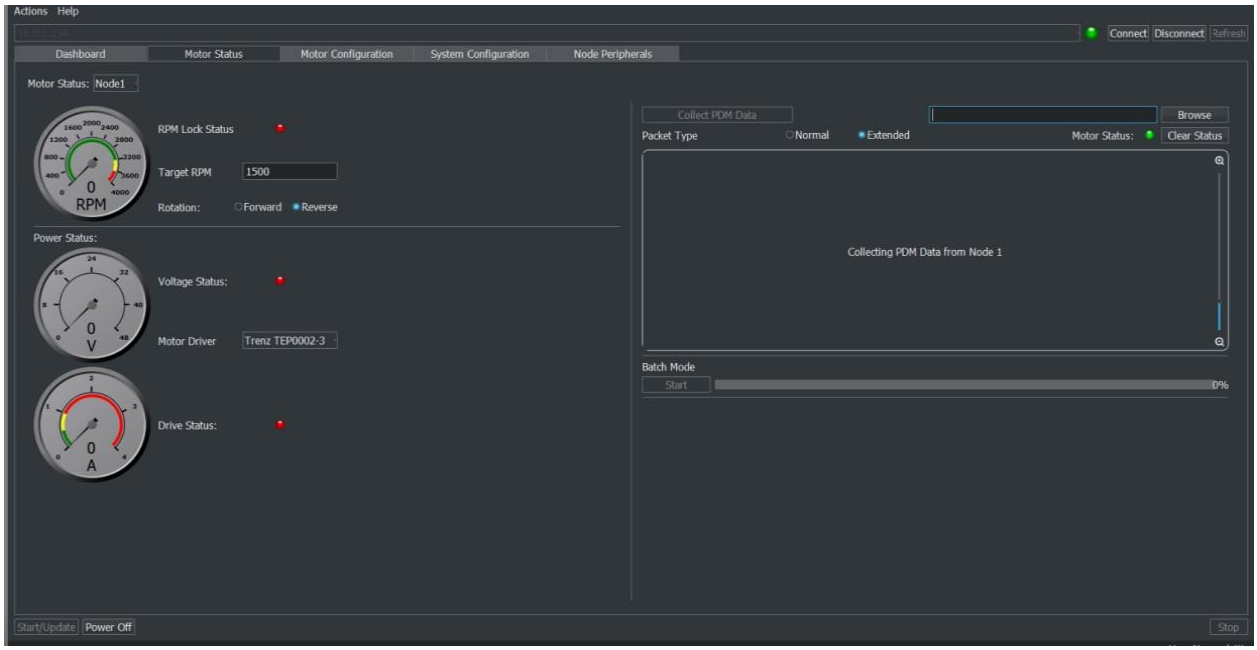

Figure F.8. Application Software- Motor Status Tab: Red Status

F.1.3. One of the Selected Motor Running Warning

Problem Description: The pop-up message *One of the selected motors is running* appears.

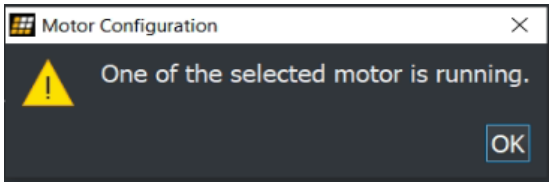

Figure F.9. Motor Running Warning Message

Solution: Stop all the motors, then configure or update the Motor Configuration.

F.1.4. Did not Receive Response from the Board Warning

Problem Description:

The pop-up message **Did not receive response from the board appears** while connecting to the user interface.

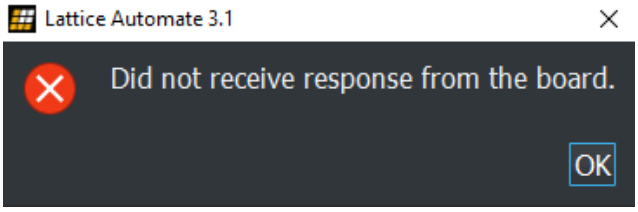

Figure F.10. Motor Connection Message

Solution:

1. Check the Ethernet connection. Follow the steps in Appendix E. Ethernet Connection
2. to ping the main board from the PC.
3. Make sure the Main System Ready and Node System Ready LEDs are illuminated, as described in the Hardware System Readiness section.
4. Use Docklight to check the print statements from the main board:
   a. Connect a UART cable between the Main Board and a PC with Docklight installed.
   b. Open Docklight on the PC.
   c. Double-click **COM**.

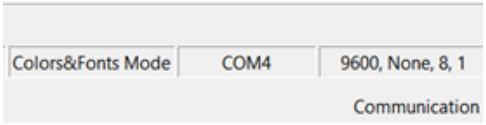

Figure F.11. Motor Connection Message

   d. Select the last USB serial port in the list, as shown in Figure F.12.

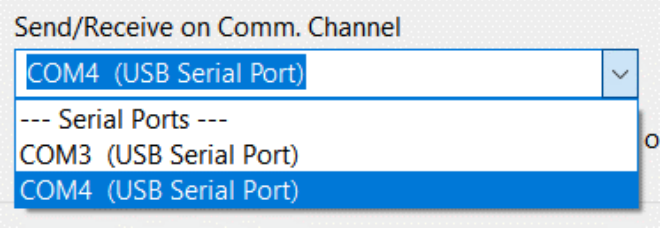

Figure F.12. Send/Receive on Comm. Channel

e. Select baud rate **9600**.

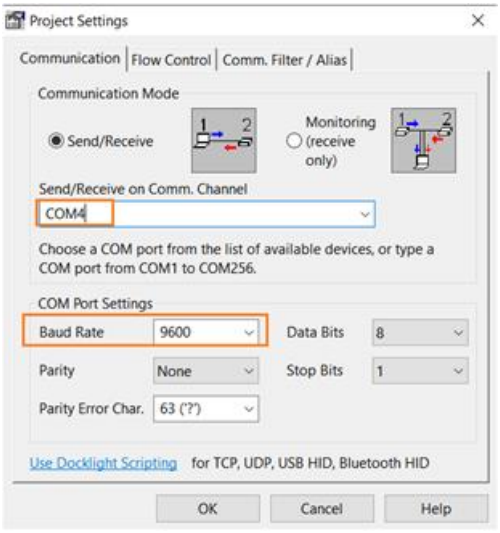

Figure F.13. Project Settings

f. Click **Run.**

Figure F.14. Run Button

g. Power cycle the Main Board or press **SW3** button on the Main Board.

h. In Docklight, select the **ASCII** tab. Wait 60 seconds and check that the print statements are valid, as shown in Figure F.15.

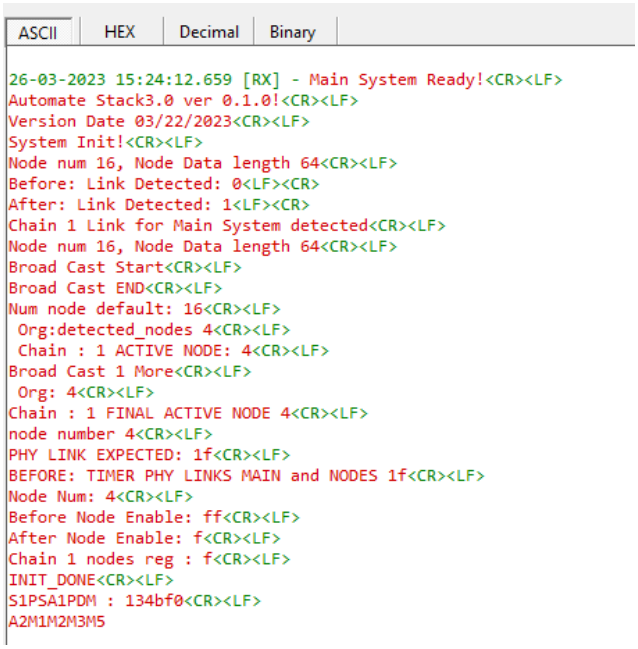

Figure F.15. ASCII Print Statements

i. LED D64 on the Main Board should be illuminated after the Docklight prints have appeared.

j. If LED D64 did not illuminate and/or the Docklight prints are incorrect, power cycle the Main System.

5. Check the IP address on the user interface.

a. Make sure the Automate 3.1 user interface is set to connect to IP address **192.168.1.4**.

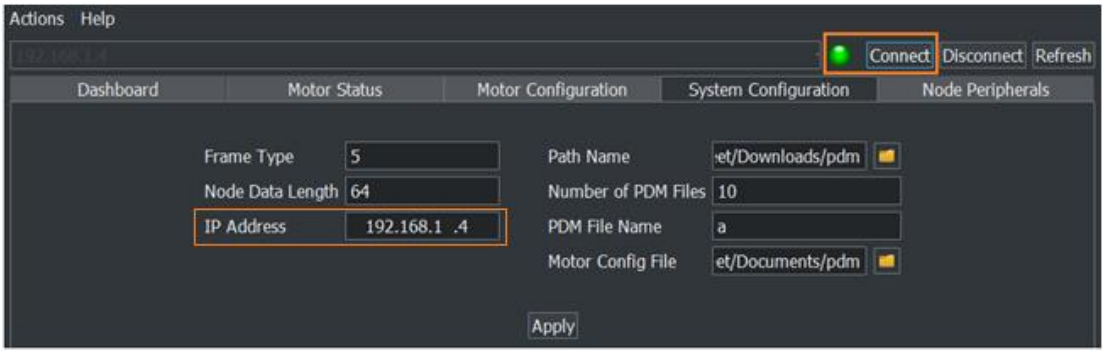

Figure F.16. Automate 3.1 User Interface

b. If you change the main system's IP address in Propel Builder, make sure using the changed IP address. Refer to section 2.6 of the Automate Stack 3.1 Reference Design (FPGA-RD-02284) document for more details on changing the main system's IP address.

F.1.5. Allowing Permission to Publisher and Subscriber

Problem Description:

You skipped the *allow the permission* while installing the Automate 3.1 user application and throw this message on the user interface: *Did not receive response from the board (Not able to load)*.

Solution:

1. Complete the installation steps.
2. Open the user interface.
3. Click **Connect**.
4. Allow permission to the subscriber.

Figure F.17. Allow Permission: Subscriber

5. Allow the permission to publisher. Another way is to changes the directory location.

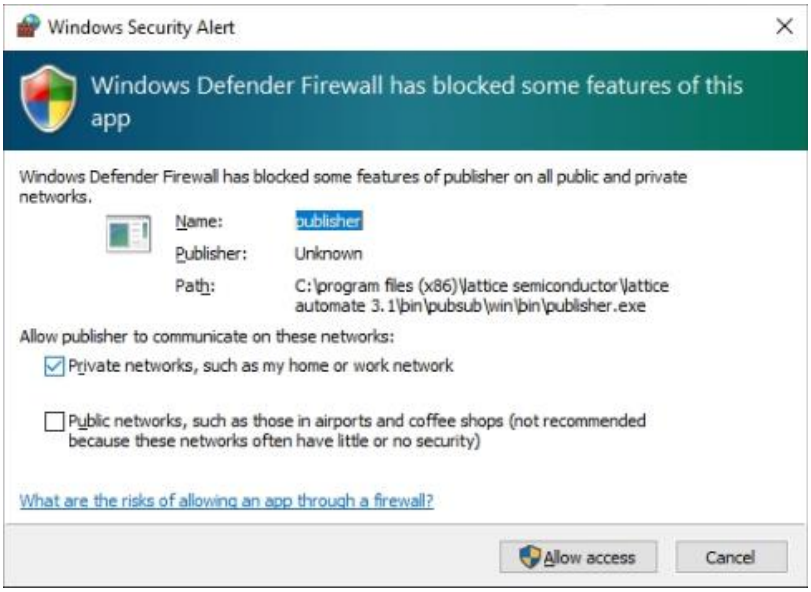

Figure F.18. Allow Permission: Publisher

Annexure-A

Cable Description

Sl. No.	Cable	Description
1	Mini USB 	It is used to interface the Mini USB port of the board to the PC for the purpose of loading .bit file and Modbus.
2	Aardvark I^{2}C/SPI Host Adapter 	It is used to interface between PC and Lattice Certus NX Board through I2C and SPI protocol.
3	FS SFP-GB-GE-T 	It is used to make an Ethernet connection on the main system board

References

- CertusPro-NX web page
- Certus-NX web page
- Lattice Automate

Other references:

- Lattice Insights for Lattice Semiconductor training courses and learning plans
- Lattice Radiant FPGA design software

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 1.1, February 2024

Section	Change Summary
Automate Stack File Directory Structure	Updated the Main System content.
Appendix B. Programming the Automate Stack on Respective Flash	Added the following sections: • B.1.1. Programming NV Register – Enabling 32-bit SPI Addressing • B.1.2. Primary Main System • B.1.3. Golden Main System
Appendix D. Checking Prints with MCS Manual and Automatic Booting	Newly added section.

Revision 1.0, December 2023

Section	Change Summary
All	Initial release.

www.latticesemi.com