Automate Stack 4.0 Demo

User Guide

FPGA-UG-02231-1.0

January 2025

Disclaimers

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

Inclusive Language

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

Contents

Contents 3
Abbreviations in This Document..9
1. Introduction..10
1.1. Key Components and Technologies 10
1.2. Automate 4.0 Demo Overview..10
1.3. User Interface..10
1.4. Demonstration Capabilities...10
2. Demo System Setup..11
2.1. System Level Block Diagram..11
3. Communication Protocols (OPCUA) 12
3.1. Main System to Host PC Communication – OPCUA 12
3.2. Main System to Node System Communication – OPCUA 12
3.3. Node System to Node System Communication 13
3.4. Ethernet Communication with lwIP stack 13
4. Hardware Requirements 14
4.1. Hardware Setup 14
4.2. Hardware System Readiness...15
4.2.1. Programming the FPGA Board 15
4.2.2. Power up Sequence...15
4.2.3. LED Status on Boards...15
5. Software Requirements...17
5.1. Optional Software 17
5.2. Software System Readiness 17
6. Automate Stack File Directory Structure 18
6.1. Executables 18
6.1.1. Main System..18
6.2. User Interface..18
6.3. Automate Propel Patch 18
7. Automate Stack Graphical User Interface Application (PC)..19
7.1. Starting the Application on the Host PC..19
7.2. Connecting to the Main System 20
7.3. Dashboard Tab 21
7.4. Refresh Button 22
7.5. Motor Configurations...23
7.6. Target RPM, Voltage, and Drive Status Using the Dashboard Tab.......................................27
7.7. Testing the Motor Status 29
7.8. Testing the Forward/Reverse Rotation 31
7.9. Testing the Encoder Status Position..33
7.10. PDM Data Collection 33
7.10.1. Collecting the PDM Data 33
7.10.2. Batch Mode 36
7.10.3. PDM Data Collection Process 38
7.10.4. Collecting PDM Data from the Simulated Faulty Motor Using the Switches 40
7.11. Testing the Node Peripherals...41
7.11.1. Connecting in the Total Phase Control Center 41
7.11.2. I2C...44
7.11.3. Modbus 45
8. Known Issues 48
8.1. Motor RPM Shown Zero Intermittently in the Automate 4.0 Graphical User Interface 48
8.2. Node Peripherals I2C Read/Write Operation...48
8.3. Multiboot MCS file 48
8.4. Multiboot Intermittent Failure..48

Appendix A. Installing the Graphical User Interface Application (PC) 49
Appendix B. Programming the Main and Node Systems with Automate Stack Bitstreams 52
B.1. Main System 52
B.2. Node System 58
Appendix C. Debugging the Main System (Avant-E Evaluation Board) 65
C.1. Debugging the Primary Main System Standalone Project 65
C.2. Debugging the Golden Main System Standalone Project 68
Appendix D. Connecting to the Ethernet 69
D1. Connecting Between the Main System and the Laptop/PC 69
D.2. Setting the Ethernet Configuration 69
D.3. Checking the Connection through Ping 72
D.4. UDP Packet Transaction 72
D.5. Packet Sender Commands 76
D.5.1. Motor Config Commands 76
D.5.2. Recommending Command Sequences to Test Basic Functionality 78
Appendix E. Troubleshooting 79
E.1. Troubleshooting the Automate 4.0 User Interface 79
E.1.1. Motor Configuration Values are 0 79
E.1.2. Motor Status Tab Failure (RPM Lock Status, Voltage Status, Drive Status and Encoder Link Status are Red) 82
E.1.3. One of the Selected Motor is Running 83
E.1.4. Response Not Received from the Board Warning 83
E.1.5. Allowing Permission to the Publisher and Subscriber 87
References 90
Technical Support Assistance 91
Revision History 92

Figures

Figure 2.1. System Block Diagram 11
Figure 3.1. Main System and Host PC Communication 12
Figure 3.2. Main System and Node System Communication 12
Figure 3.3. UDP Communication Overview 13
Figure 4.1. System Setup with Host PC (Example with One Node) 14
Figure 4.2. Main System Ready LEDs 16
Figure 4.3. Node System Ready LED 16
Figure 7.1. User Interface Application Login Screen 19
Figure 7.2. User Interface Application Dashboard Tab 20
Figure 7.3. User Interface Application System Configuration: IP Address 20
Figure 7.4. User Interface Application System Configuration: Updated Successfully Window 21
Figure 7.5. User Interface Application System Configuration: IP Address Configured Status 21
Figure 7.6. User Interface Application Dashboard: Number of Nodes Connected 22
Figure 7.7. User Interface Application Dashboard: 2 Nodes Connected 22
Figure 7.8. Refresh Process 23
Figure 7.9. User Interface Application Dashboard: 1 Node Connected after Refresh 23
Figure 7.10. User Interface Application System Configuration: Motor Config File 24
Figure 7.11. User Interface Application System Configuration: Updated Successfully Pop-up 24
Figure 7.12. User Interface Application Motor Configuration: Node Selection 24
Figure 7.13. User Interface Application Motor Configuration: Node Selected 25
Figure 7.14. User Interface Application Motor Configuration: Warning Message 25
Figure 7.15. User Interface Application Motor Configuration: Authentication Window 26
Figure 7.16. User Interface Application Motor Configuration: Update Configuration 26
Figure 7.17. Motor Config File Saved in Host PC 26
Figure 7.18. Motor Config.txt File in Host PC 26
Figure 7.19. User Interface Application Dashboard: Set Target RPM 27
Figure 7.20. User Interface Application Dashboard: Start/Update 27
Figure 7.21. User Interface Application Dashboard: RPM Lock Achieved Status 28
Figure 7.22. User Interface Application Motor Status: RPM Lock, Voltage, and Drive Status 28
Figure 7.23. User Interface Application Dashboard: Stop and Power Off Buttons 29
Figure 7.24. User Interface Application Motor Status: Start/Update 30
Figure 7.25. User Interface Application Motor Status: RPM Lock Status When Target RPM is Achieved 30
Figure 7.26. User Interface Application Motor Status: Forward Rotation Status 31
Figure 7.27. User Interface Application Motor Status: Reverse Rotation Status 32
Figure 7.28. User Interface Application Motor Status: Motor Power Off 32
Figure 7.29. User Interface Application PDM Data: Path Name 33
Figure 7.30. User Interface Application PDM Data: Updated Successfully Window 33
Figure 7.31. User Interface Application PDM Data: Collect PDM Data 34
Figure 7.32. User Interface Application PDM Data: Collecting PDM Data from Node 34
Figure 7.33. User Interface Application PDM Data: Analyzing PDM Data from Node 35
Figure 7.34. User Interface Application PDM Data: PDM Image 35
Figure 7.35. User Interface Application PDM Data: Clear Status and Motor Status 36
Figure 7.36. User Interface Application PDM Data: Browse Button and Zoom Slider 36
Figure 7.37. User Interface Application PDM Batch Mode: Number of PDM files 37
Figure 7.38. User Interface Application PDM Batch Mode: Updated Successfully Pop-up 37
Figure 7.39. User Interface Application PDM Batch Mode: Start 37
Figure 7.40. User Interface Application PDM Batch Mode: Batch Mode 100% Status 38
Figure 7.41. User Interface Application PDM Batch Mode: Collecting Multiple Images 38
Figure 7.42. User Interface Application PDM Data: Normal Mode 39
Figure 7.43. User Interface Application PDM Data: Components Involved in PDM Data Collection Process 39
Figure 7.44. User Interface Application PDM Data: Process Flow for PDM Data Collection 39
Figure 7.45. Three Resistor Switches Tilted Towards the Resistors 40

Figure 7.46. PDM Data Collection: Example PDM Image of a Faulty Motor....41
Figure 7.47. Certus-NX Board – I2C Connection....41
Figure 7.48. Total Phase – Adapter Setting (I2C)....42
Figure 7.49. Total Phase – I2C SPI Selection....42
Figure 7.50. Total Phase – I2C Control (I2C)....43
Figure 7.51. Total Phase – SPI Control: SPI Enable....43
Figure 7.52. Total Phase – I2C and SPI Control Enabled....44
Figure 7.53. Application Software – Node Peripherals: I2C Write....44
Figure 7.54. Total Phase – Transaction Log-I2C....44
Figure 7.55. Total Phase Aardvark I2C Control User Interface....45
Figure 7.56. Total Phase – Transaction Log - I2C....45
Figure 7.57. Application Software – Node Peripherals: I2C-Read....45
Figure 7.58. Application Software - Node Peripherals: Modbus-Write....46
Figure 7.59. Docklight COM Selection....46
Figure 7.60. USB Serial Port Selection....46
Figure 7.61. Docklight – Project Setting....46
Figure 7.62. Docklight – Run....47
Figure 7.63. Application Software – Node Peripheral: Modbus Write....47
Figure 7.64. Docklight Prints....47
Figure A.1. Installer Directory....49
Figure A.2. Automate 4.0 Initial Install Setup....49
Figure A.3. Automate 4.0 Next Step....50
Figure A.4. Automate 4.0 Next Step....50
Figure A.5. Automate 4.0 Installation in Progress....51
Figure A.6. Automate 4.0 Installation Finish....51
Figure B.1. Main System Board....52
Figure B.2. Radiant Programmer – Default Screen....53
Figure B.3. Radiant Programmer – Initial Project Window....53
Figure B.4. Radiant Programmer - Device Selection....54
Figure B.5. Radiant Programmer - Device Operation....54
Figure B.6. Radiant Programmer – Program Primary Main System Binary....55
Figure B.7. Radiant Programmer – Program Primary Main System Bitstream....56
Figure B.8. Radiant Programmer – Program Golden Main System Binary....57
Figure B.9. Radiant Programmer – Program Primary u-Boot Binary....58
Figure B.10. Radiant Programmer - Default Screen (Node System)....59
Figure B.11. Radiant Programmer – Initial Project Window (Node System)....60
Figure B.12. Radiant Programmer - Device Properties (Node System)....60
Figure B.13. Radiant Programmer – Device Operation (Node System)....61
Figure B.14. Radiant Programmer – Erase All (Node System)....62
Figure B.15. Radiant Programmer – Bit Stream Flashing Settings (Node System)....63
Figure B.16. Radiant Programmer – Binary Flashing Settings (Node System)....64
Figure B.17. SW3 Button of the Certus-NX Versa Board....64
Figure C.1. Open Propel Terminal....65
Figure C.2. Propel terminal selection window....66
Figure C.3. Propel terminal – Com and Baud Rate Selection....66
Figure C.4. Primary Main System Propel Prints....67
Figure D.1. Run the ncpa.cpl to Open the Network Connections Control Panel....69
Figure D.2. Ethernet Settings....69
Figure D.3. Ethernet Properties....70
Figure D.4. Double Click on Internet Protocol Version....70
Figure D.5. Configure the Settings....71
Figure D.6. Disable the Setting....71
Figure D.7. Enable the Setting....72
Figure D.8. Ping on cmd Terminal....72

Figure D.9. Downloadable Link of Wireshark 73
Figure D.10. Downloadable Link of Packet Sender 73
Figure D.11. Docklight – Com Selection 73
Figure D.12. Docklight – Com and Baud Rate Selection 74
Figure D.13. Docklight – Run (Start Communication) 74
Figure D.14. Wireshark Tool – Ethernet Selection 74
Figure D.15. Packet Sender Tool – Send 64 Bytes of ASCII Characters 75
Figure D.16. Packet Sender Tool – Send UDP Packet 75
Figure D.17. Packet Sender Tool – Transaction Log 75
Figure D.18. Docklight – ASCII Logs 75
Figure D.19. Wireshark Tool – Write udp.port == 1486 75
Figure D.20. Source and Destination UDP Packet 76
Figure D.21. Wireshark Tool – First UDP Packet 76
Figure D.22. Wireshark Tool – Second UDP Packet 76
Figure E.1. Motor Configuration Values are All 0 79
Figure E.2. Motor Configuration: Node Selection 80
Figure E.3. Motor Configuration: All Nodes Selected 80
Figure E.4. Motor Configuration Value Changes 81
Figure E.5. Application Software – Motor Configuration: Warning Message 81
Figure E.6. Application Software – Motor Configuration: Authentication Window 81
Figure E.7: Application Software – Motor Configuration: Update Configuration 82
Figure E.8. Application Software – Motor Status Tab: Red Status 82
Figure E.9. Motor Running Warning Message 83
Figure E.10. Motor Connection Message 83
Figure E.11. Open Propel Terminal 84
Figure E.12. Propel Terminal Selection Window 84
Figure E.13. Propel terminal – Com and Baud Rate Selection 85
Figure E.14. Main system Print Statements 86
Figure E.15. Automate 4.0 User Interface 87
Figure E.16. Allow Permission: Subscriber 87
Figure E.17. Allow Permission: Publisher 88
Figure E.18. Firewall and Network Protection 88
Figure E.19. Allow an App through Firewall 88
Figure E.20. Scroll Down and Find Publisher 89
Figure E.211. Scroll Down and Find Subscriber 89

Tables

Table 6.1. Primary Main System Project Files....18
Table 6.2. Golden Main System Project Files....18
Table 6.3. Node System Project Files....18

Abbreviations in This Document

A list of abbreviations used in this document.

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

1. Introduction

The Lattice Automate Solution Stack is a comprehensive offering from Lattice Semiconductor, specifically designed to address the needs of the industrial sector. This solution stack integrates hardware and software components to enable key industrial automation technologies such as motor control, predictive maintenance, OPCUA (Open Platform Communications Unified Architecture), Ethernet based communication and seamless interaction with multiple systems. The Automate 4.0 uses the Lattice Avant™-E FPGA for the main system and Lattice Certus™-NX FPGA for the node system to showcase key features enabled for industrial automation

1.1. Key Components and Technologies

- Motor Control – Efficiently manage and control different motors simultaneously. Closed loop and open loop motor control capabilities.
- Predictive Maintenance – Implement advanced algorithms to predict and prevent equipment failures, thereby reducing downtime and maintenance costs.
- OPCUA – Facilitate secure and reliable data exchange between industrial equipment and systems.
- System Interaction – Ensure smooth communication and interoperability between different industrial host systems and devices.

1.2. Automate 4.0 Demo Overview

The Automate 4.0 Demo is a practical demonstration of the capabilities of the Lattice Automate Solution Stack. It comprises of two main components: the main system and the node system.

- Main System – Utilizes the Avant-E FPGA on the Avant-E evaluation board, equipped with an Ethernet FMC daughter card to enable Gigabit Ethernet communication.
- Node System – Utilizes the Certus-NX FPGA on the Certus-NX Versa development board, connected to a Trenz motor driver board for motor control and an encoder board for connecting to an encoder over ENDAT interface.

To operate this demonstration, both the main and node systems must run Automate demo specific bitstreams.

1.3. User Interface

The Automate 4.0 demo features a user-friendly user interface that simplifies the configuration and operation of the motors within the demo system. The user interface provides functionalities to:

- Configure motor settings
- Adjust motor speed and direction
- Monitor encoder position
- Display voltage and current indicators
- Collection and analyzing motor data for Predictive Maintenance

1.4. Demonstration Capabilities

The Automate 4.0 demo effectively showcases how the host system, connected to the main system over OPCUA, can effortlessly interact with the end node system. The main system and node system communicate over Ethernet showing networking capabilities. Through the intuitive Automate 4.0 user interface, you can control a BLDC motor connected to multiple node systems, demonstrating the ease of integration and operation within an industrial automation environment.

This detailed introduction highlights the robust features and practical applications of the Lattice Automate Solution Stack, emphasizing its role in advancing industrial automation technologies.

2. Demo System Setup

The Automate demo uses one main system (running on an Avant-E 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 8 node systems for best performance. Each node system is connected to a Trenz motor driver board, which is connected to a brushless DC motor. The node system is also connected to an RS-485 encoder board, which is connected to Heidenhain rotary encoder over the EnDat interface.

The main system connects to the Host PC. The Host PC communicates through OPCUA with a PC running the user interface (Lattice Automate 4.0), through which you can 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 4.0 user interface is connected to the main system over a standard 1G Ethernet cable. The main system running on the Avant-E device is connected to the node system over EtherConnect.

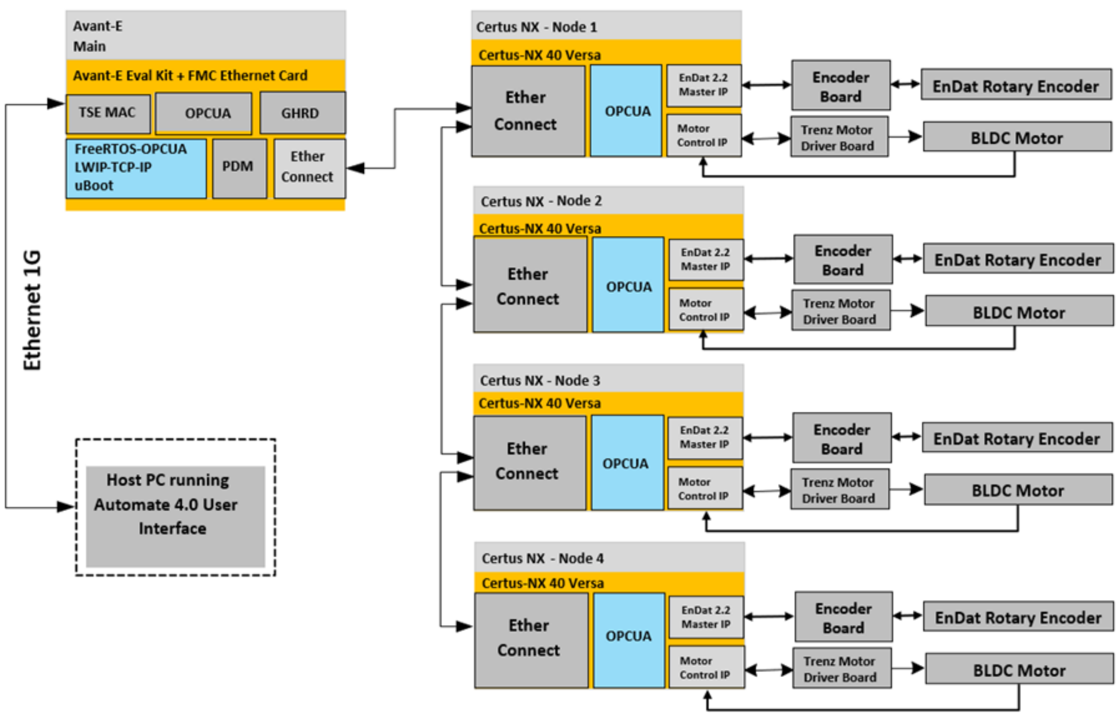

Figure 2.1. System Block Diagram

3. Communication Protocols (OPCUA)

Automate 4.0 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 been 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 4.0 uses the publish/subscribe version of OPCUA.

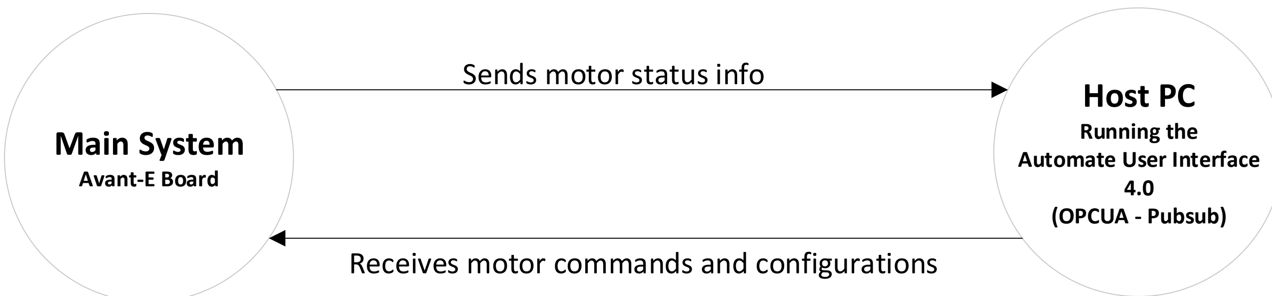

Figure 3.1. Main System and Host PC Communication

3.2. Main System to Node System Communication – OPCUA

The Main system and Node system communicate using the OPCUA 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.

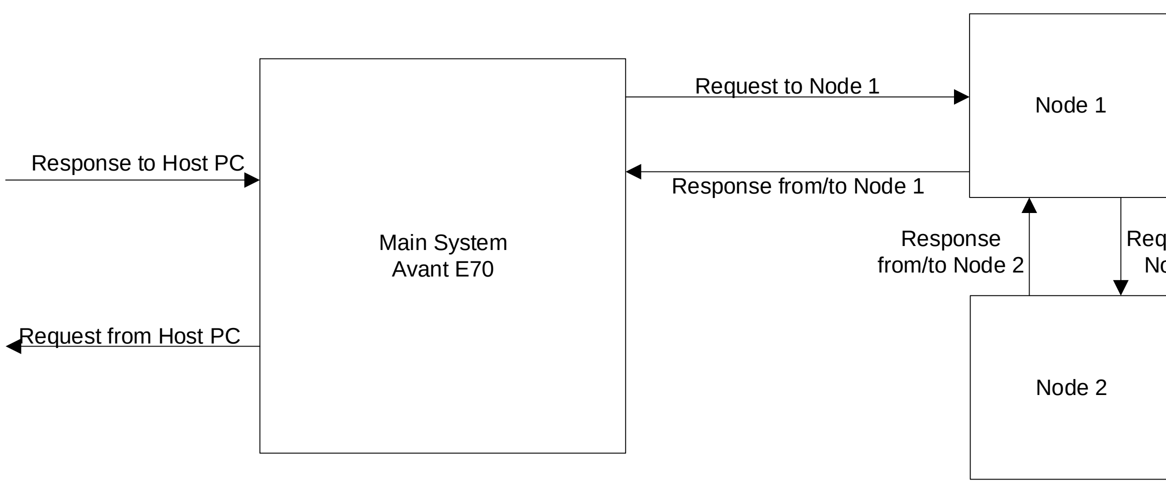

Figure 3.2. Main System and Node System Communication

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. Ethernet Communication with lwIP stack

The Lightweight IP (lwIP), a widely used open-source TCP/IP stack designed for embedded systems, is being used to pack and send UDP datagrams from user application to Lattice 1G TSE Ethernet MAC IP Core and vice versa. The User Datagram Protocol (UDP/IP) is a communications protocol integrated in the lwIP that is 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.

The source UDP port, destination UDP port, destination IP address, and MAC address can be configured through the lwIP interface. The core is delivered in a format that allows direct connection to the Lattice 1G TSE Ethernet MAC IP Core.

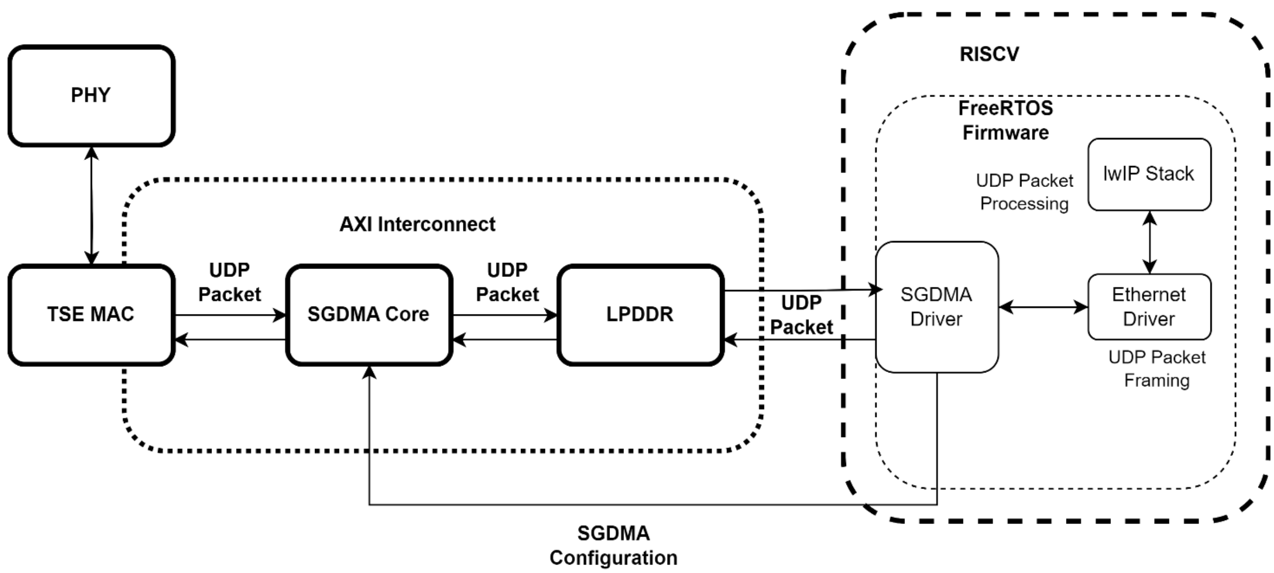

Figure 3.3. UDP Communication Overview

4. Hardware Requirements

This demonstration requires the following hardware components:

- Main System
  - Lattice Avant-E 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.
  - Ethernet FMC daughter card for Ethernet connection over RGMII (insert at FMC1 Connector of the Avant-E Evaluation Board).
  - Ethernet cable to connect the main system to the host PC
  - 12 V power adapter for board power
- Node System(s)
  Note: The demo supports up to 16 node systems. The requirements for one node system are listed below:
  - 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 BLY171D 24V, 4000 RPM
  - Trenz TEP0002 motor driver board(s)
  - 24 V-10 Amp DC Power Supply for motor(s)
  - Encoder RS485 transceiver board
  - Heidenhain ROQ 437 EnDat Rotary Encoder
  - Heidenhain K17 Diaphragm coupling
  - 12 V-3A Amp DC Power Supply for Encoder RS485 transceiver board
- User Interface (Client System)
  - PC running Windows 10 Operating System of 1920 x 1080 resolution, 100% dpi

4.1. Hardware Setup

The details of hardware connections are illustrated in Figure 4.1.

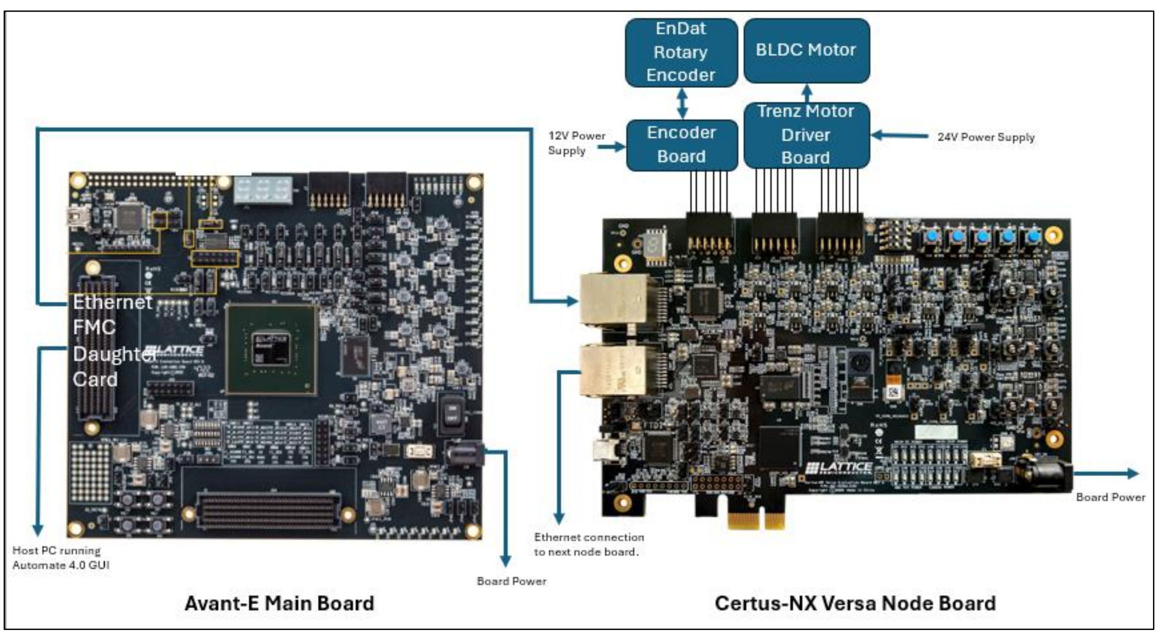

Figure 4.1. System Setup with Host PC (Example with One Node)

4.2. Hardware System Readiness

4.2.1. Programming the FPGA Board

To program the FPGA board, perform the following:

1. Hardware must be connected properly as shown in Figure 4.1.
2. All boards must be programmed with bitstream, binaries and MCS files. Refer to the Appendix B. Programming the Main and Node Systems with Automate Stack Bitstreams section for the steps on programming the board.
3. Power cycle every board after programming.
4. After the power cycle, reset the main system (press the **SW3** button on the main system).

4.2.2. Power up Sequence

To power up the board, perform the following:

1. Power on the Host PC.
2. Power on the node system in the following sequence:
   - **Rotary Encoder, Encoder Board > Motor Driver Board, Motor > Certus-NX Versa Node Board**

   Note: The Encoder Rs485 transceiver board and the Rotary Encoder **MUST** be switched on first before the Certus-NX Versa Node Board is switched on.
3. Power on the main system.
4. Press the **SW3** button on main system to reset the main system.

   Note: The DIPSWTCH4 located SW10 location of Certus-NX Versa Node Board to select between open loop or closed loop system:
   - 1: Close Loop
   - 0: Open Loop
5. You **MUST** power cycle the Node System before attempting to change the switch setting and repeat again the programming sequence from beginning.

4.2.3. LED Status on Boards

To check the LED status on the boards, perform the following:

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 to D25: Node system ready
      - If the 7-segment LED is light up (see Figure 4.3), the Ethernet is connection established.

   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 E. Troubleshooting section.

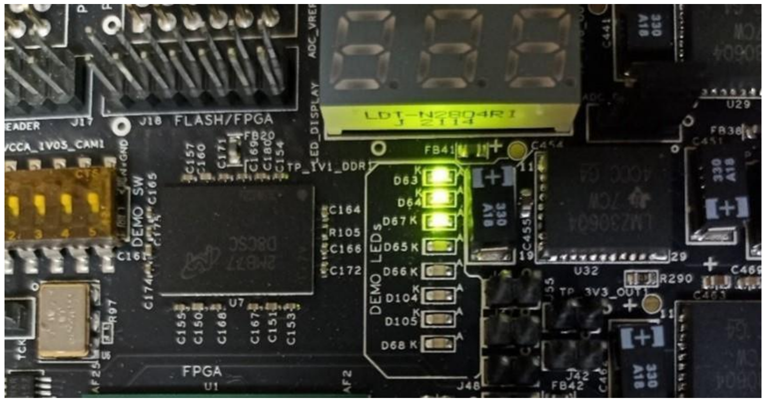

Figure 4.2. Main System Ready LEDs

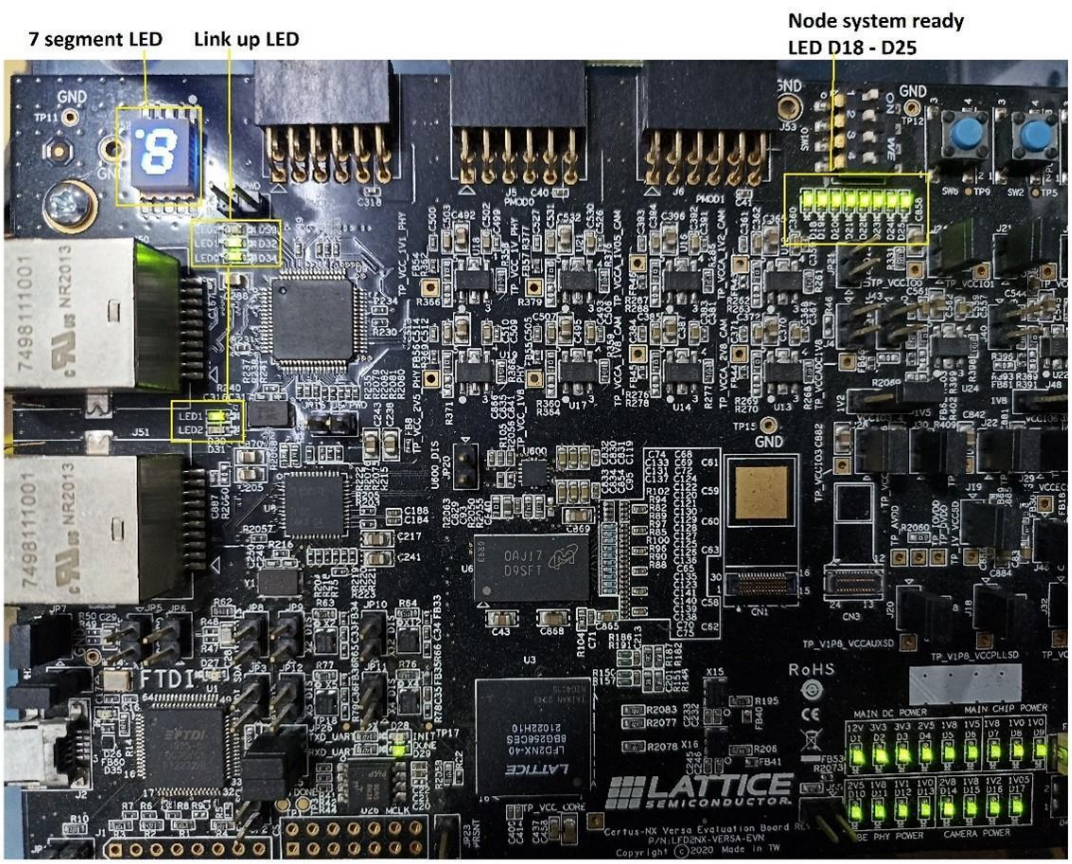

Figure 4.3. Node System Ready LED

5. Software Requirements

- Lattice Radiant 2024.1 or later
- Lattice Radiant Programmer 2024.1 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 4.0 Test Application Software
  Note: Lattice Automate 4.0is available in OPCUA to publish/subscribe version. The software programs are available for download only if the user log in at www.latticesemi.com.
- Lattice Automate Propel Patch for the project can be downloaded from the Downloadable software section: https://www.latticesemi.com/en/Solutions/Solutions/SolutionsDetails02/Automate.

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)

5.2. Software System Readiness

1. Establish the ethernet connection before connecting to the user interface.
   a. Follow steps *a* through *c* in the Appendix D. Connecting to the Ethernet section to establish the ethernet connection.
2. To install and run the user interface, see the following sections:
   a. Refer to Appendix A. Installing the Graphical User Interface Application (PC) section to install the user interface application on the PC (one time).
   b. Refer to the Automate Stack Graphical User Interface Application (PC) section to run the GUI application on the PC (every time the user run the demo).

6. Automate Stack File Directory Structure

All files can be downloaded in 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

6.1.1.1. Primary Main System

Default binaries for Main System.

Table 6.1. Primary Main System Project Files

File Name	Description
c_main_system_4_0_cnnCrc.bin	Primary Main System application binary.
primary_u-boot.bin	Primary uBoot binary to boot and jump to primary application.
soc_primary_main_system_4_0_impl1.bit	Primary FPGA bitstream.

6.1.1.2. Golden Main System

Backup images when primary binaries are corrupted, and CRC fails.

Table 6.2. Golden Main System Project Files

File Name	Description
Golden_AppCrc.bin	Golden Main System application binary.
golden_u-boot.bin	Golden uBoot binary to boot and jump to golden application.
soc_golden_main_system_4_0_impl1.bit	Golden FPGA bitstream.

6.1.1.3. Node System

Table 6.3. Node System Project Files

File Name	Description
c_node_system_4_0.bin	Binary for Node System application.
soc_node_impl1.bit	Bitstream for Certus-NX FPGA

6.2. User Interface

These files are programmed onto the Host PC.

- Automate4p0GUI.exe

6.3. Automate Propel Patch

The Automate Propel Patch can be downloaded from the *Downloadable Software* section in the Lattice Automate web page.

- Propel2024.1_patch_automate_2412182140.exe

7. Automate Stack Graphical User Interface Application (PC)

The Lattice Automate 4.0 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. This section describes the various components of the Automate 4.0 user interface and steps to run the motor.

Notes:

1. For Lattice Automate Stack 4.0 Application Installation on the PC, refer to Appendix A. Installing the Graphical User Interface Application (PC) section.
2. For programming the main system and node system boards, refer to the Appendix B. Programming the Main and Node Systems with Automate Stack Bitstreams section.

7.1. Starting the Application on the Host PC

To start the application on the host PC, perform the following:

1. Open the Lattice Automate application.

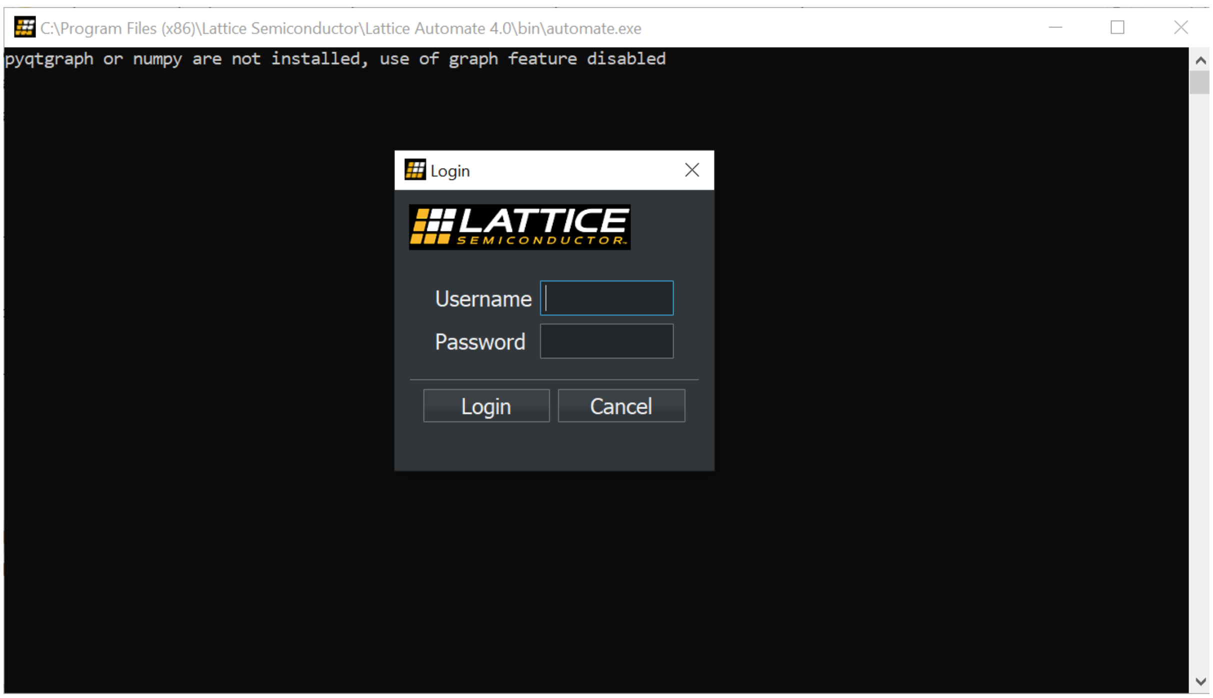

Figure 7.1. User Interface Application Login Screen

2. Enter the following credentials and click **Login**:
    - **Username: lattice**
    - **Password: lattice**
3. After successful login, the Dashboard tab opens.

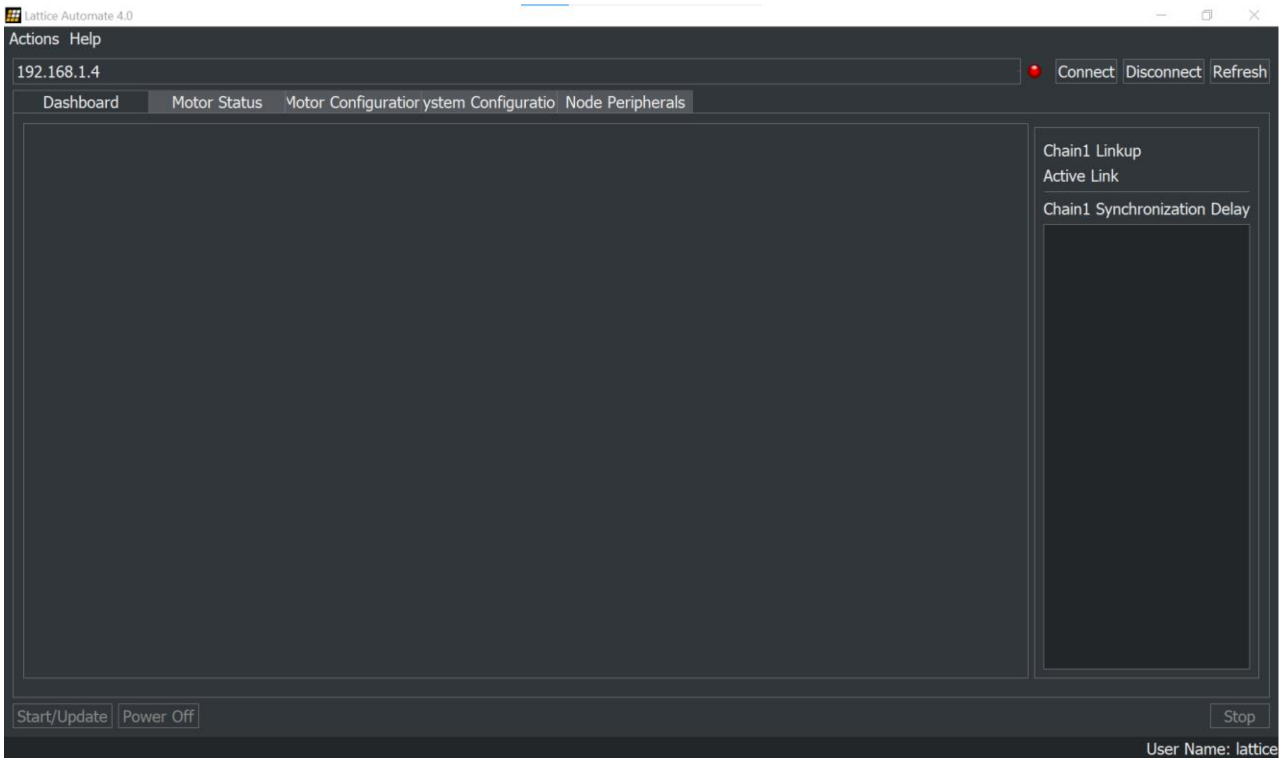

Figure 7.2. User Interface Application Dashboard Tab

7.2. Connecting to the Main System

To connect to the main system, perform the following:

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

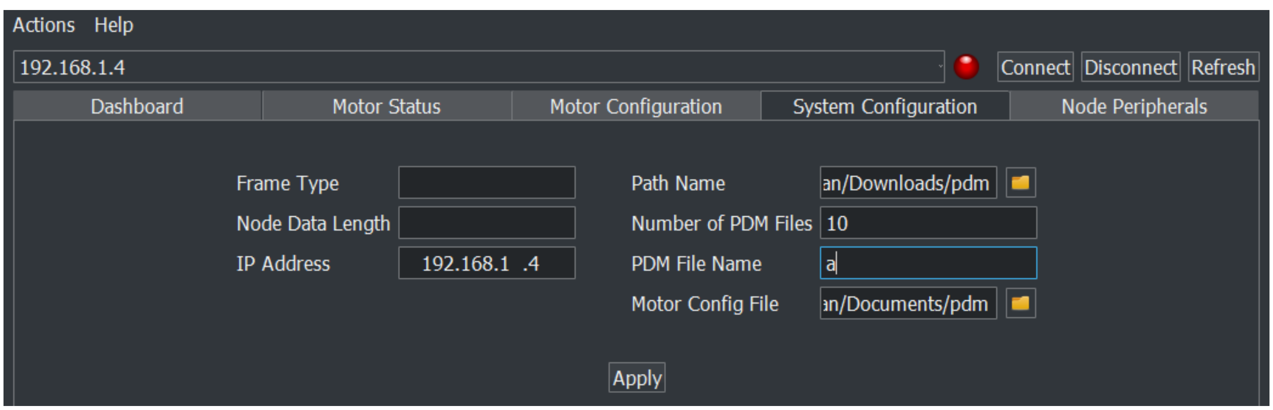

Figure 7.3. User Interface Application System Configuration: IP Address

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**.
4. The *Updated Successfully* pop-up window appears. The updated IP address is visible on top of the IP Address bar.

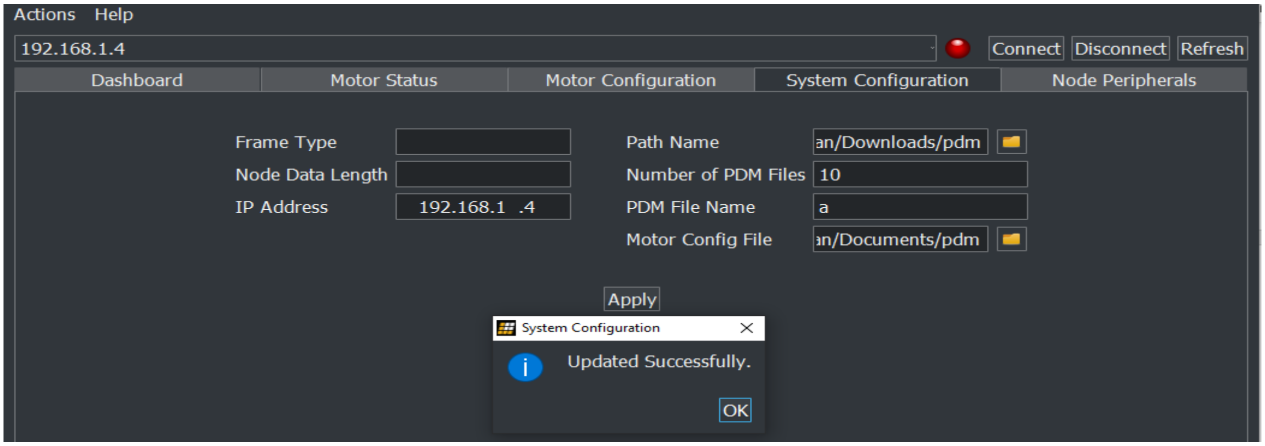

Figure 7.4. User Interface Application System Configuration: Updated Successfully Window

5. Click **Connect**.
6. Once it is connected, the *Connect* status turns green.

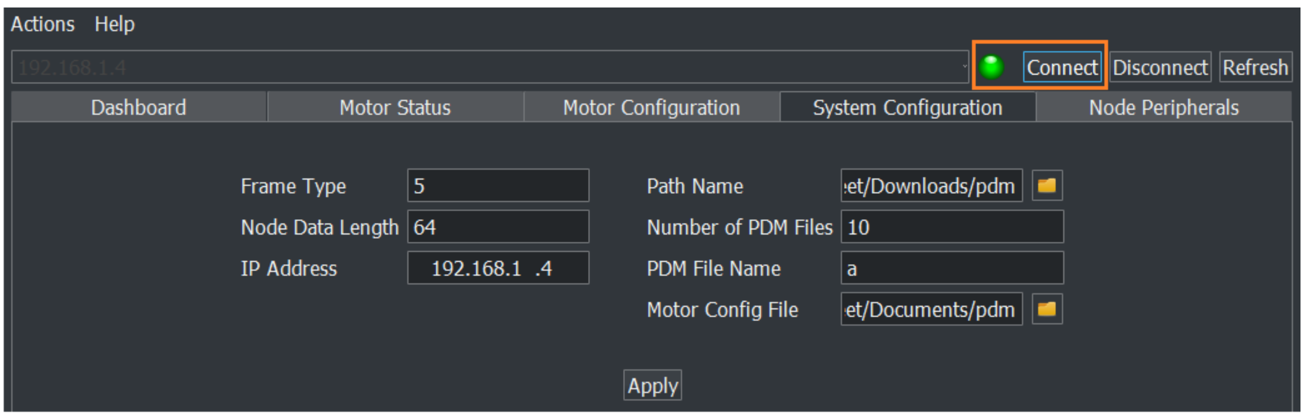

Figure 7.5. User Interface Application System Configuration: IP Address Configured Status

7.3. Dashboard Tab

The Dashboard shows the number of node systems connected to main system. In the below example there are four nodes connected to main system.

To view the dashboard tab:

1. Click the **Dashboard** tab as shown in Figure 7.6.
2. Check the Chain 1 Link up status.
3. Active link status shows the four nodes.
4. Check the Chain1 – nodes synchronization delay.

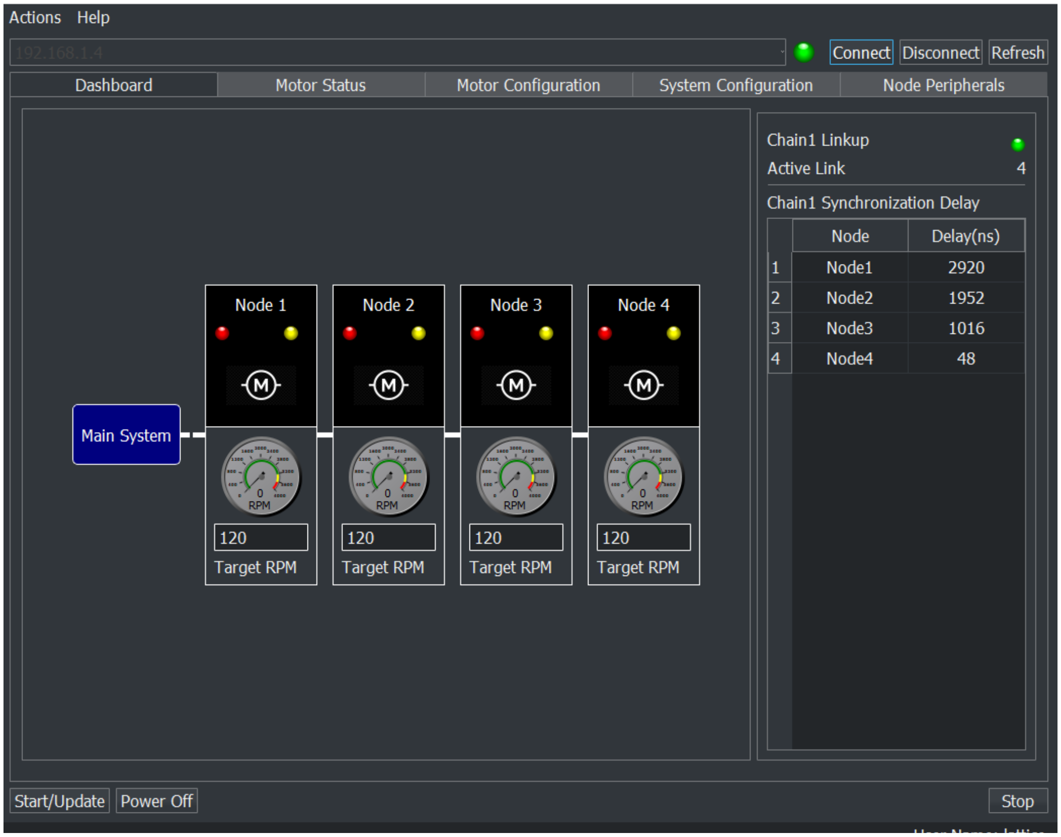

Figure 7.6. User Interface Application Dashboard: Number of Nodes Connected

7.4. Refresh Button

The Refresh button can be used to update nodes, if you are already connected and need to add or remove a node. Figure 7.7 shows the connected nodes.

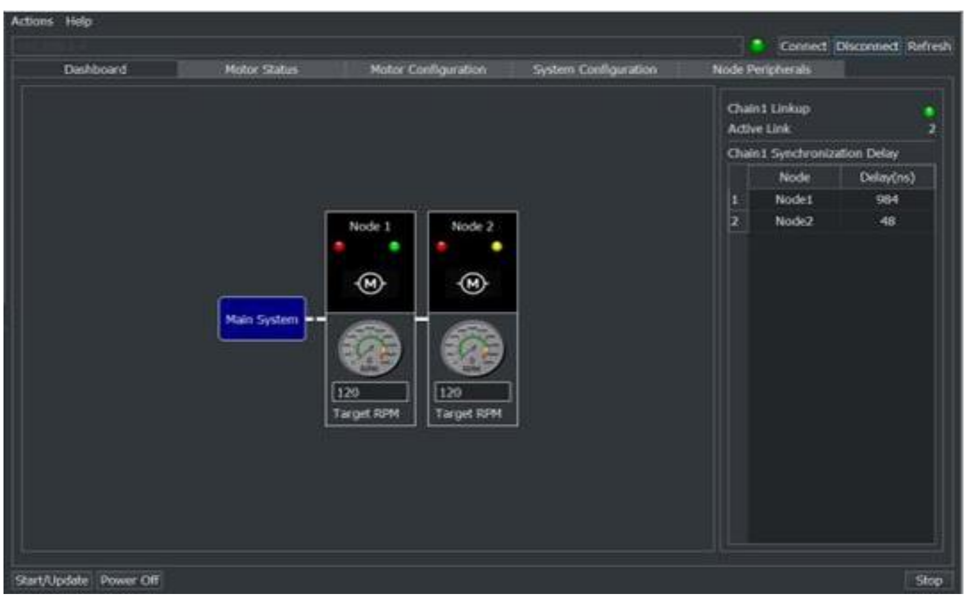

Figure 7.7. User Interface Application Dashboard: 2 Nodes Connected

To test and demonstrate the refresh button, perform the following:

1. Disconnect the last node of the chain.
2. Click the **Refresh** button within 10 seconds.
3. Wait for the refresh process to complete.

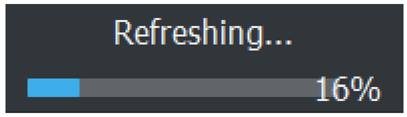

Figure 7.8. Refresh Process

4. After refresh completes, the dashboard must reflect the actual connected node(s).

Figure 7.9. User Interface Application Dashboard: 1 Node Connected after Refresh

7.5. Motor Configurations

To set up the motor configuration, perform the following:

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

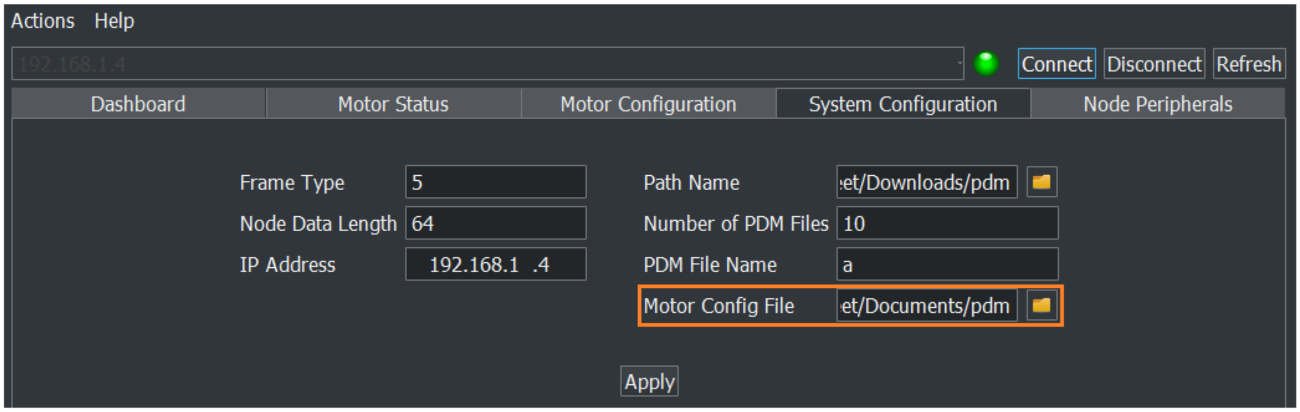

Figure 7.10. User Interface Application System Configuration: Motor Config File

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

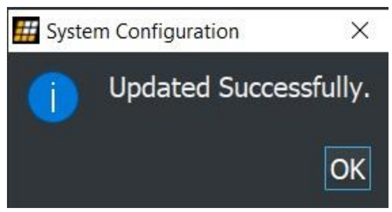

Figure 7.11. User Interface Application System Configuration: Updated Successfully Pop-up

4. Go to the **Motor Configuration** tab and select the number of nodes.

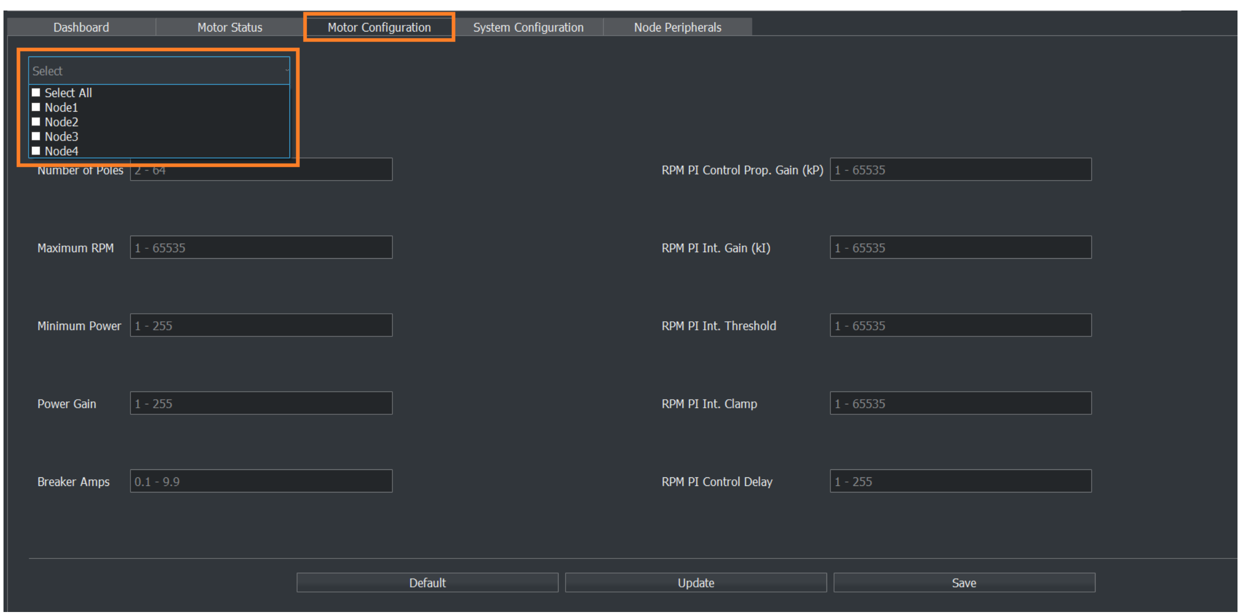

Figure 7.12. User Interface Application Motor Configuration: Node Selection

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

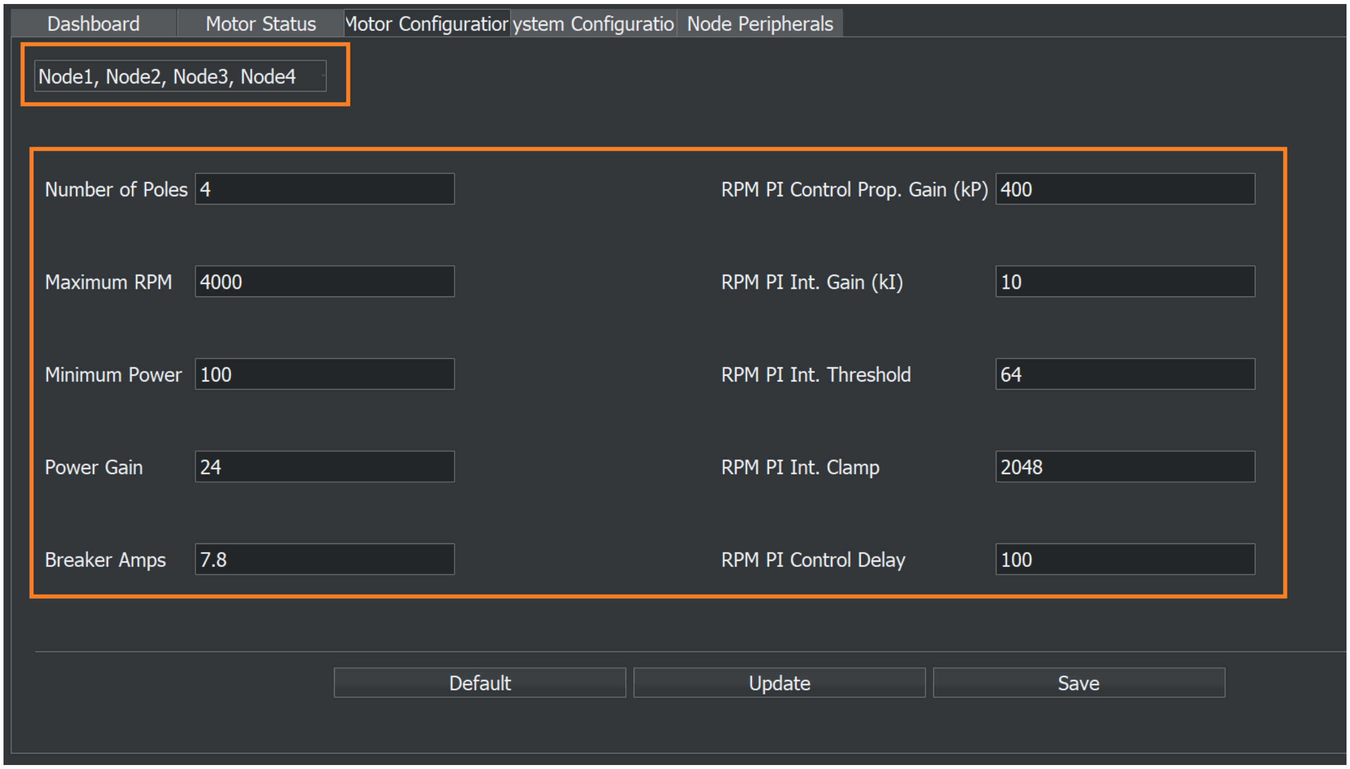

Figure 7.13. User Interface Application Motor Configuration: Node Selected

6. Change the Motor Configuration to the following values:
   - RPI PI Control Prop. Gain (kP): 400
   - RPI PI Int. Gain (kI): 10
   - Minimum Power: 100
   - Power Gain: 24
   - Breaker Amps: 7.8
   - RPI PI Control delay: 100

   Note: These configuration values are for the BLY171D 24V and 4000 RPM Motors only. Modify as needed for a different motor.
7. Click **Save** and **Update**. A pop window appears. Confirm the update action by clicking **Yes**.

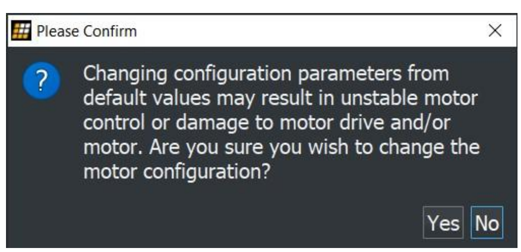

Figure 7.14. User Interface Application Motor Configuration: Warning Message

8. The authentication window appears. Enter the credentials and click **Login**:
   - **Username: lattice**
   - **Password: lattice**

Figure 7.15. User Interface Application Motor Configuration: Authentication Window

9. The *Successfully updated the configuration of the selected node* message appears. Click **OK**.

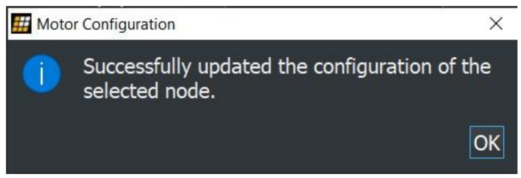

Figure 7.16. User Interface Application Motor Configuration: Update Configuration

10. Locate the config.txt file. It is saved in the location that user selected in step 2.

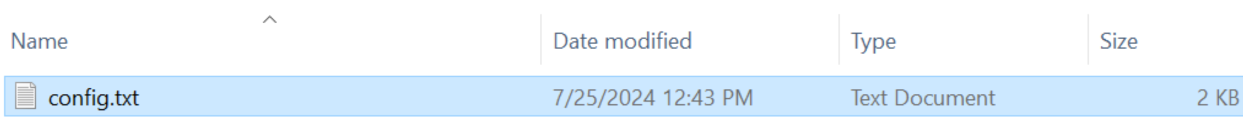

Figure 7.17. Motor Config File Saved in Host PC

11. Open the *config.txt* file and verify the saved motor configurations matches the values you entered in step 6.

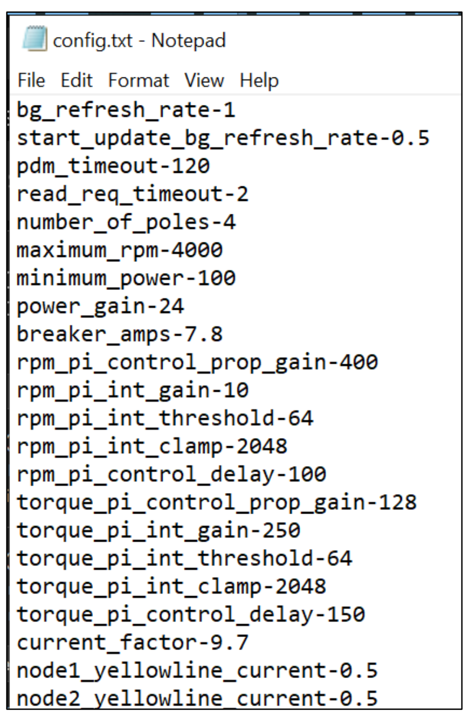

Figure 7.18. Motor Config.txt File in Host PC

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

To target the RPM, voltage and drive status, perform the following:

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

   Note: You can gradually increase the RPM up to 1800 RPM (2000 RPM for close loop motor). The following RPM increments are recommended: 120, 400, 800, 1400, 1600, 1800, and 2000 (for close loop motor). The target RPM value must 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**.

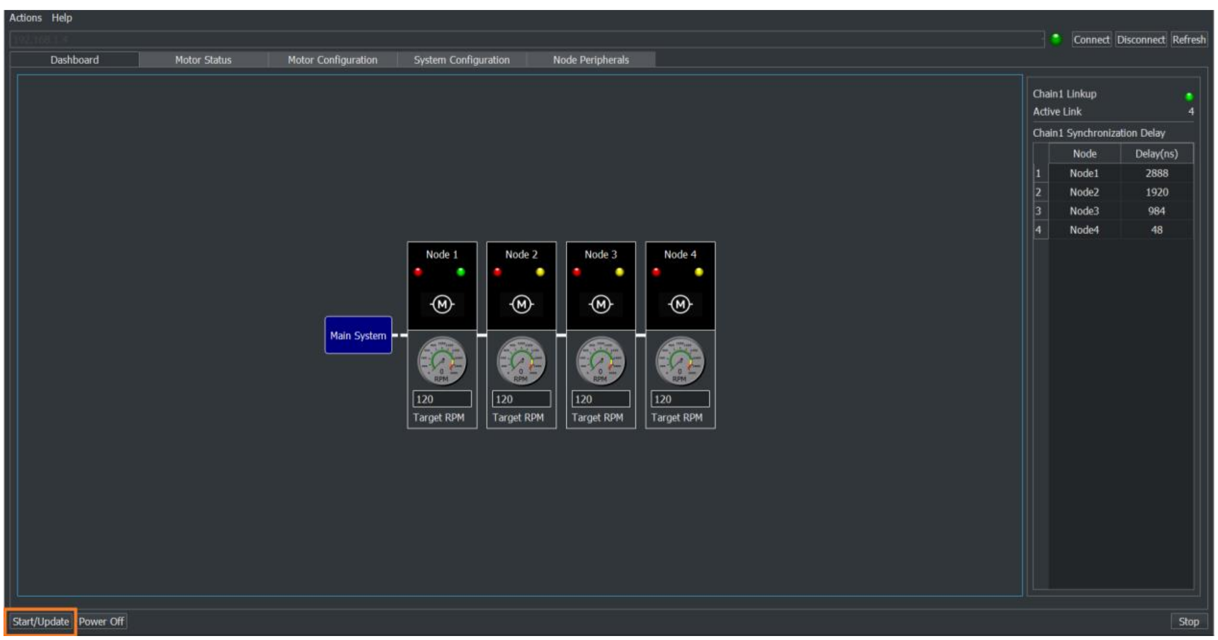

Figure 7.20. User Interface Application Dashboard: Start/Update

4. After the RPM Lock is achieved, the Node LED turns 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**.

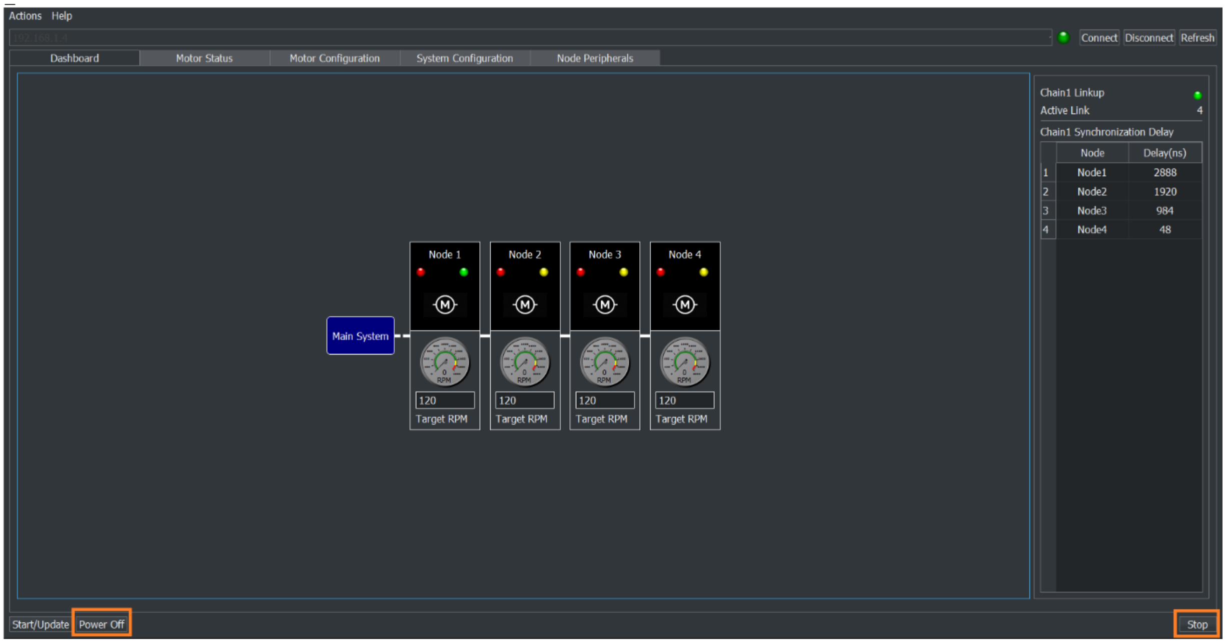

Figure 7.23. User Interface Application Dashboard: Stop and Power Off Buttons

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. Testing the Motor Status

To test the motor status, perform the following:

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

   Note: You can gradually increase the RPM up to 1800 RPM (2000 RPM for close loop). The following RPM increments are recommended: 120, 400, 800, 1400, 1600, 1800, and 2000 (for close loop motor). Target RPM must 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 **400 RPM**.
6. Once the actual RPM reaches the target RPM, the meter gauge displays 400 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. Testing the Forward/Reverse Rotation

To test the forward/reverse rotation, perform the following:

1. Click the **Motor Status** tab.
2. Select the Forward option for the rotation.
3. Enter the **Target RPM** anywhere between 120 RPM to 1800 RPM (2000 RPM for close loop motor).
4. Click **Start/Update**.

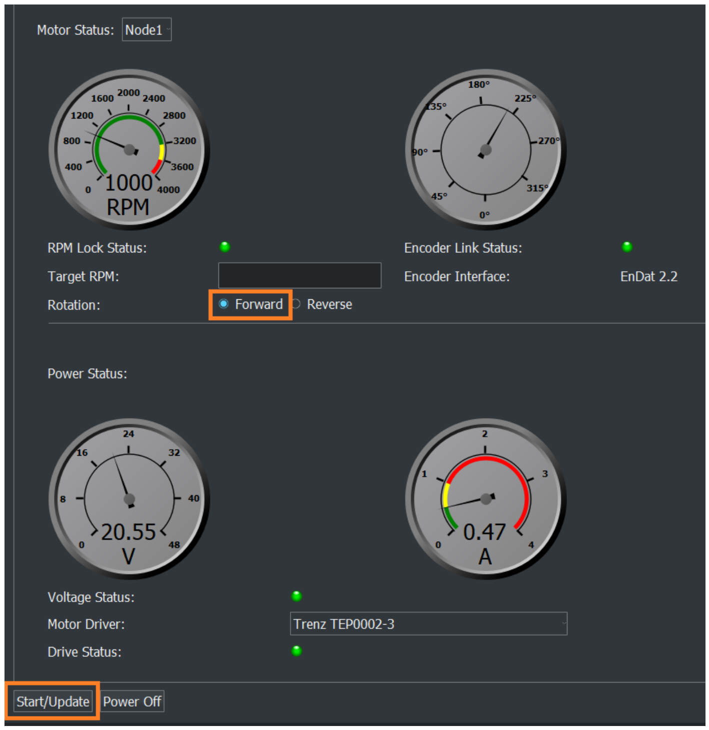

Figure 7.26. User Interface Application Motor Status: Forward Rotation Status

5. Motor starts rotating in a clockwise direction.
6. Wait until the RPM Lock status becomes green.
7. Select the Reverse option to rotate the motor in the opposite direction.

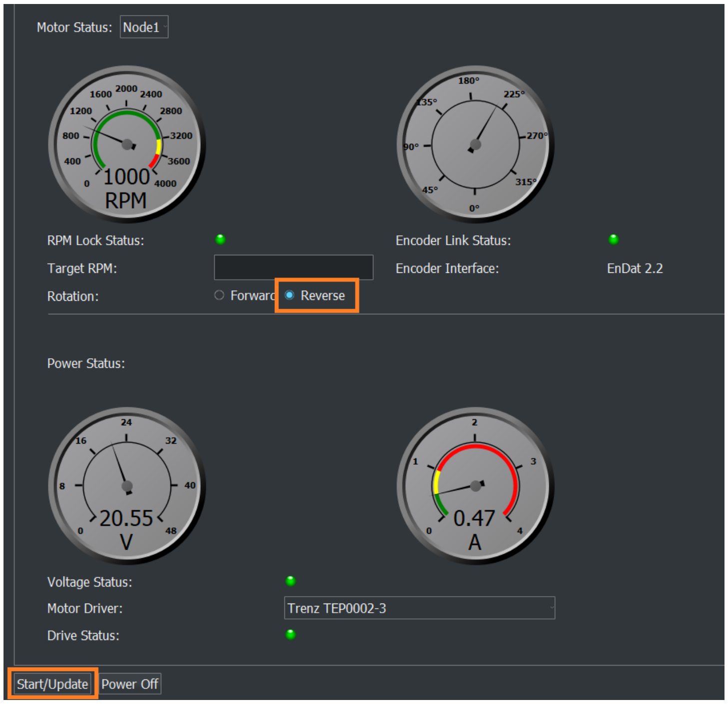

Figure 7.27. User Interface Application Motor Status: Reverse Rotation Status

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

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

7.9. Testing the Encoder Status Position

Ensure that the DIPSWTCH4 at SW10 location of Certus-NX Versa Node Board is set to 1 to run in the close loop system before proceeding to following steps.

1. Click the **Motor Status** tab.
2. Select Forward/Reverse option for the rotation.
3. Enter the **Target RPM** anywhere between 120 RPM to 1800 RPM (2000 RPM for close loop motor).
4. Click **Start/Update**.
5. The degree value of the meter gauge displays should be changing continuously as the motor is turning.

7.10. PDM Data Collection

7.10.1. Collecting the PDM Data

To collect the PDM data, perform the following:

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

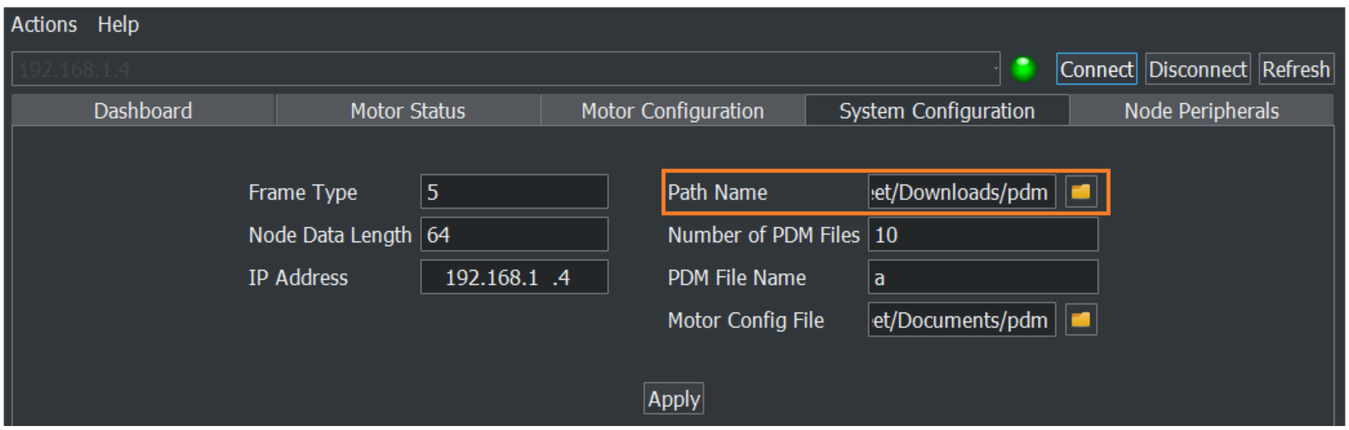

Figure 7.29. User Interface Application PDM Data: Path Name

3. Click **Apply** and **OK**. The *Updated Successfully* message appears as shown in Figure 7.30.

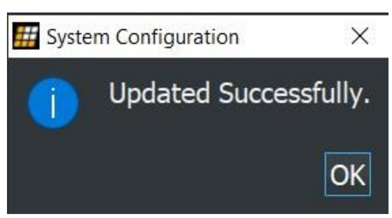

Figure 7.30. User Interface Application PDM Data: Updated Successfully Window

4. Click the **Motor Status** tab.
5. Set the Target RPM initially to 800, then gradually increase the RPM with maximum up to 1500 RPM.
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.

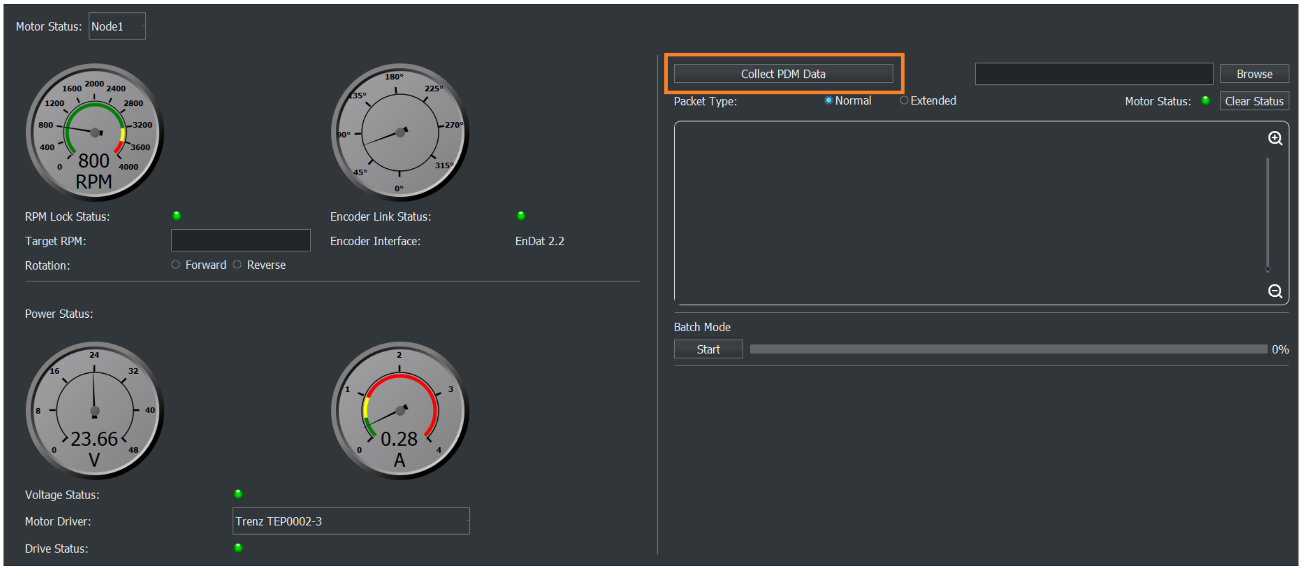

Figure 7.31. User Interface Application PDM Data: Collect PDM Data

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

   Note: See 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.

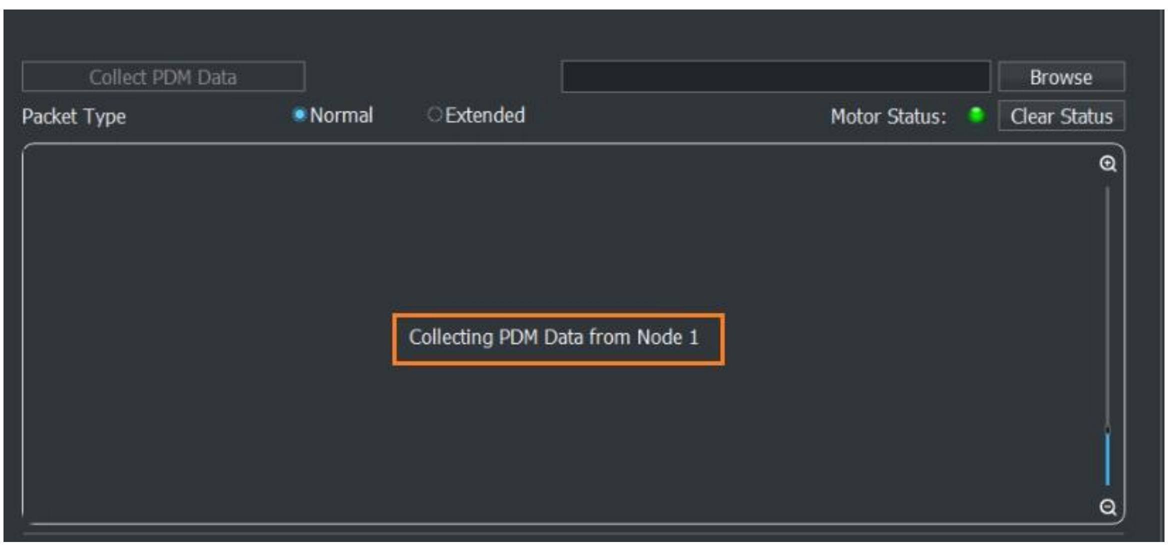

Figure 7.32. User Interface Application PDM Data: Collecting PDM Data from Node

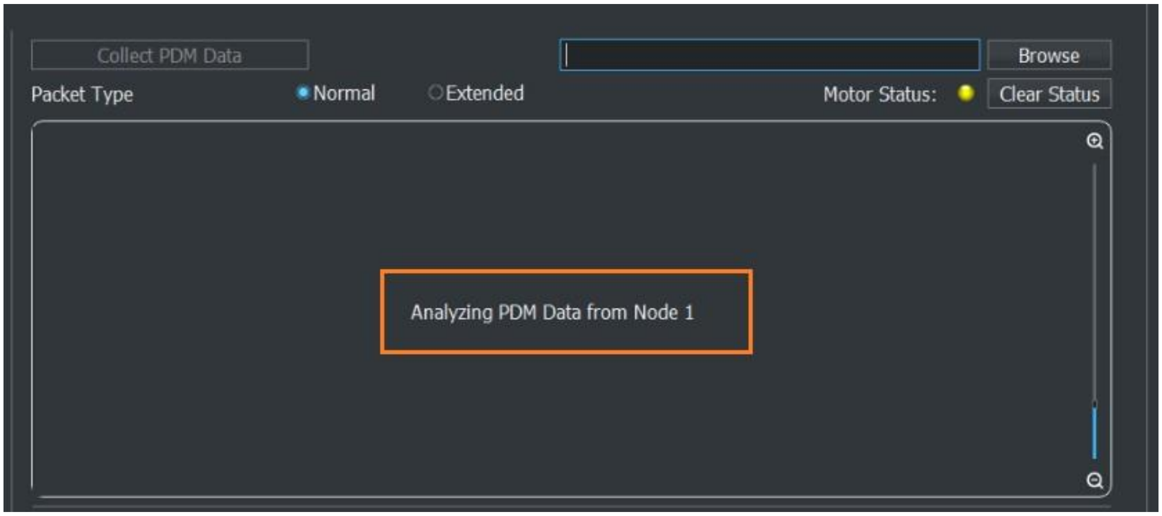

Figure 7.33. User Interface Application PDM Data: Analyzing PDM Data from Node

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

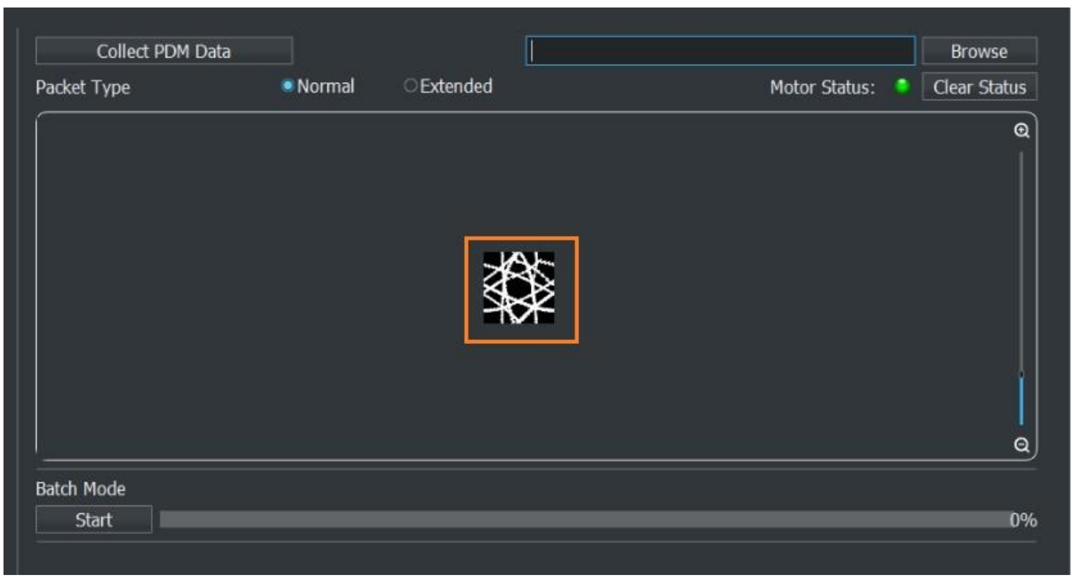

Figure 7.34. User Interface Application PDM Data: PDM Image

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

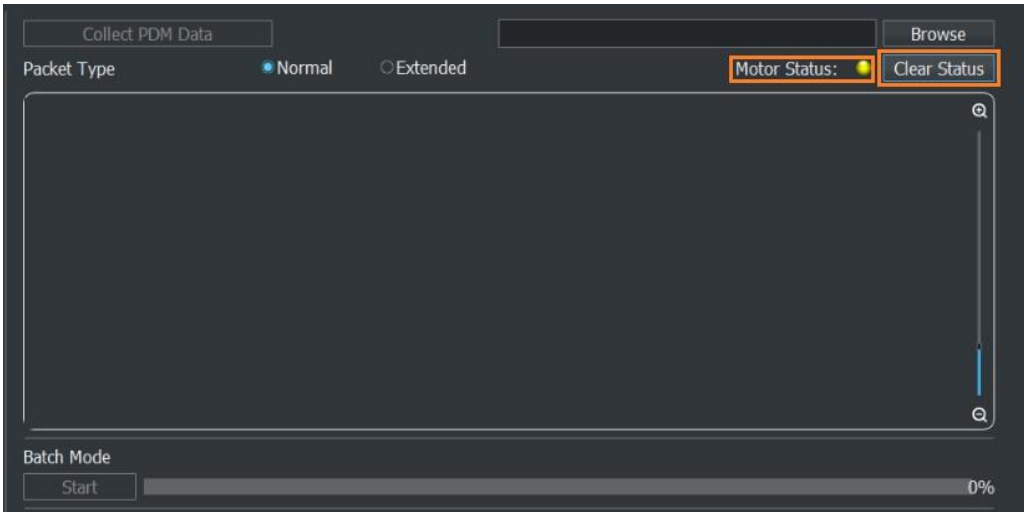

Figure 7.35. User Interface Application PDM Data: Clear Status and Motor Status

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.

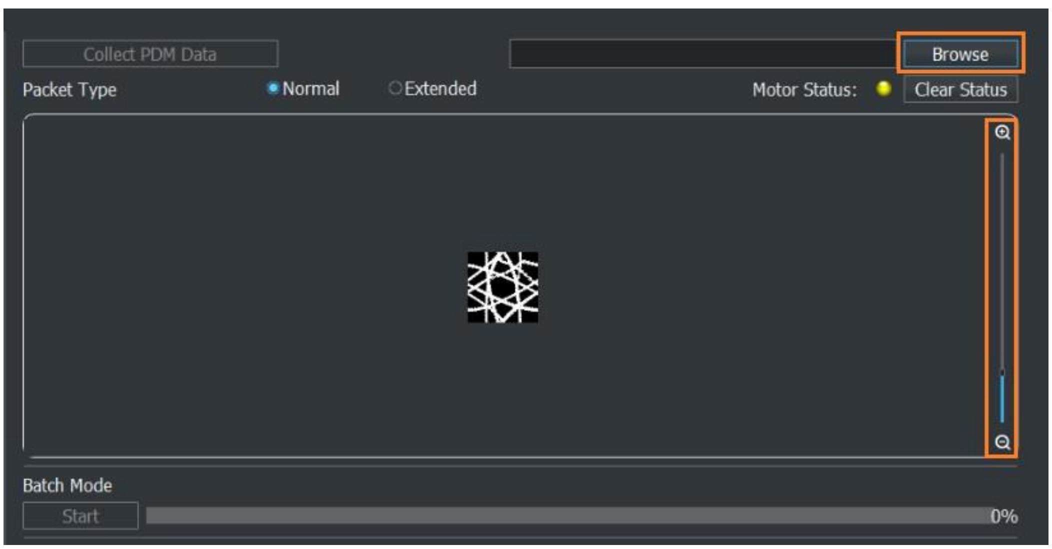

Figure 7.36. User Interface Application PDM Data: Browse Button and Zoom Slider

7.10.2. Batch Mode

Batch mode allows you to collect multiple PDM images, one after another. First, you must choose how many PDM files you wish to collect. Afterwards, the steps below are similar to the standard Collect PDM Data process.

1. Click the **System configuration** tab and enter the number of PDM files that user wants 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 the path name and browse for a location, as described in the Collecting the PDM Data section, Step 2.

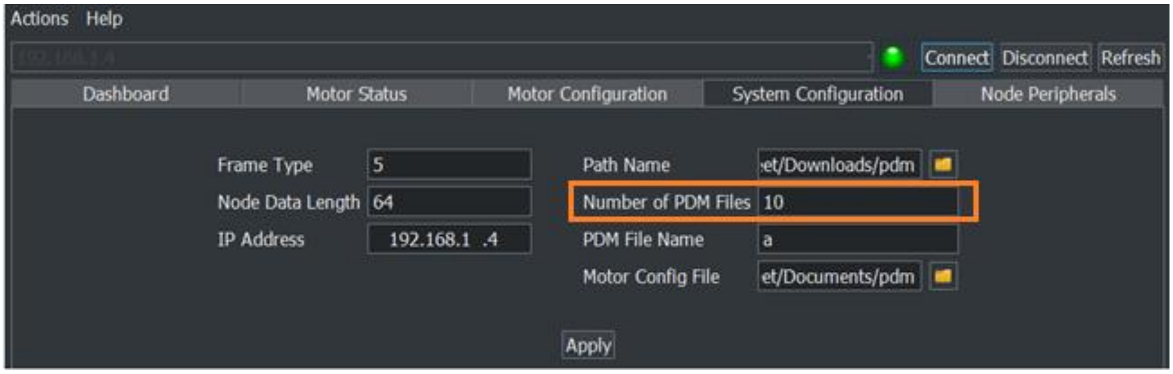

Figure 7.37. User Interface Application PDM Batch Mode: Number of PDM files

3. Click **Apply** and **OK** on the *Updated Successfully* window.

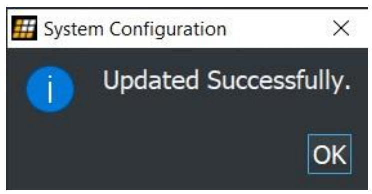

Figure 7.38. User Interface Application PDM Batch Mode: Updated Successfully Pop-up

4. Click the **Motor Status** tab.
5. Click **Start** under *Batch Mode*.

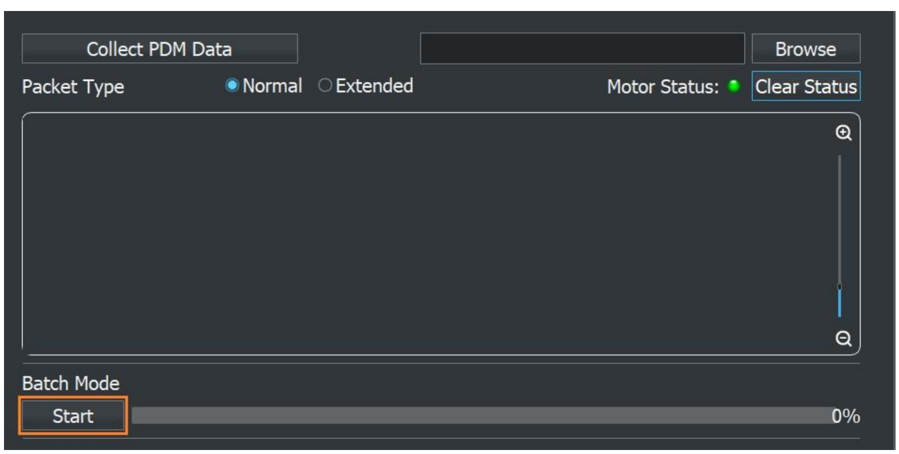

Figure 7.39. User Interface Application PDM Batch Mode: Start

6. 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.

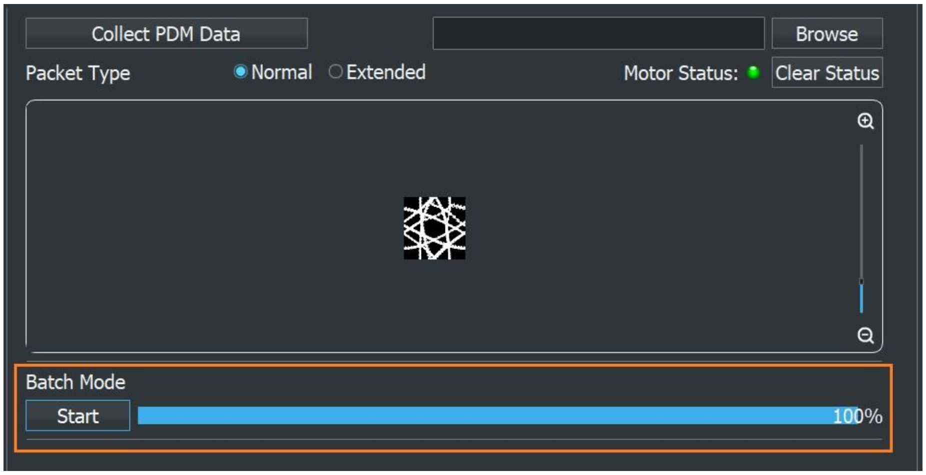

Figure 7.40. User Interface Application PDM Batch Mode: Batch Mode 100% Status

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

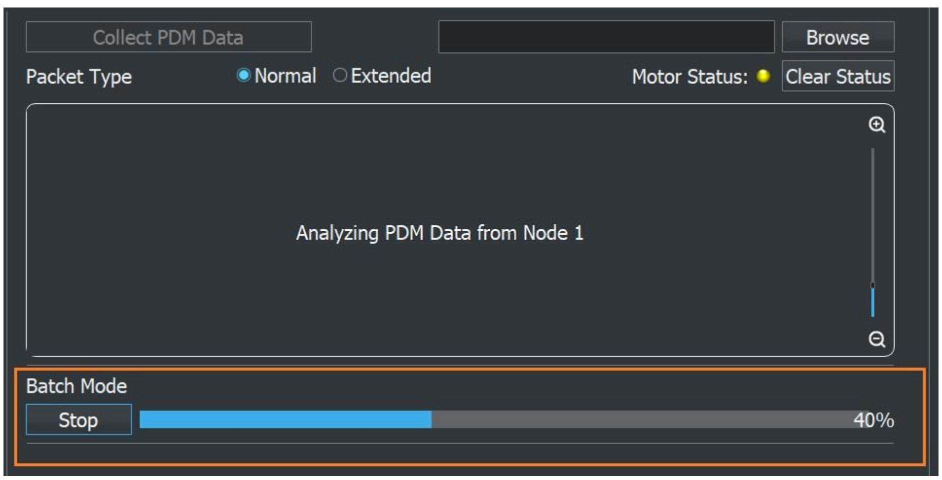

Figure 7.41. User Interface Application PDM Batch Mode: Collecting Multiple Images

8. Click **Stop** to stop the motor, and then click **Power Off**.

7.10.3. PDM Data Collection Process

When you create a PDM data request in the user interface, the request is sent over OPCUA and a PDM Data command is sent through the Etherconnect to the RISC-V CPU of the main system. 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 RISC-V CPU of the node system 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.

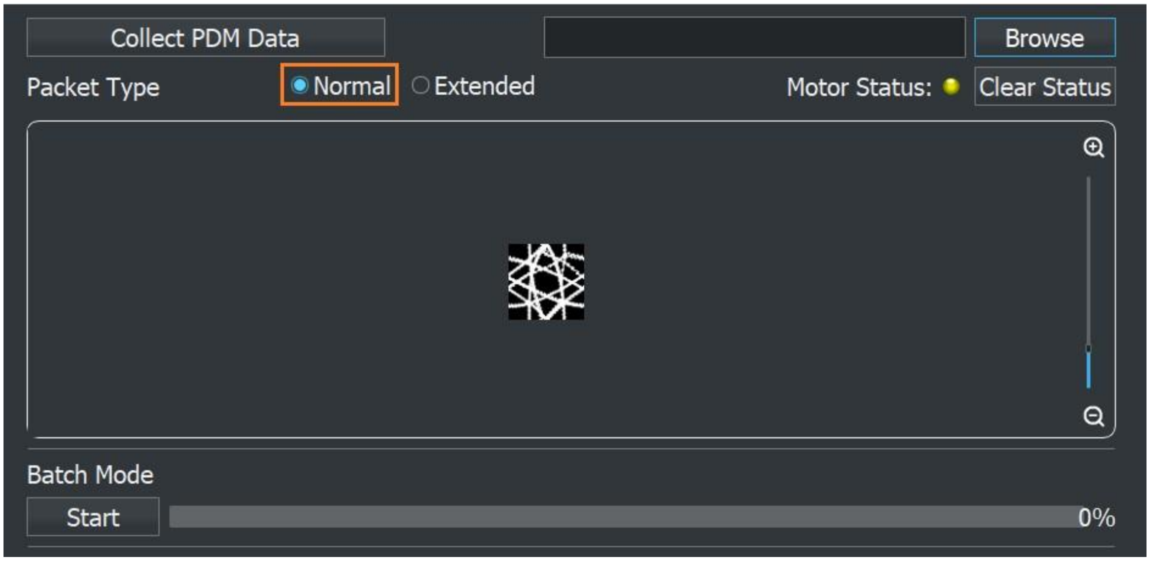

Figure 7.42. User Interface Application PDM Data: Normal Mode

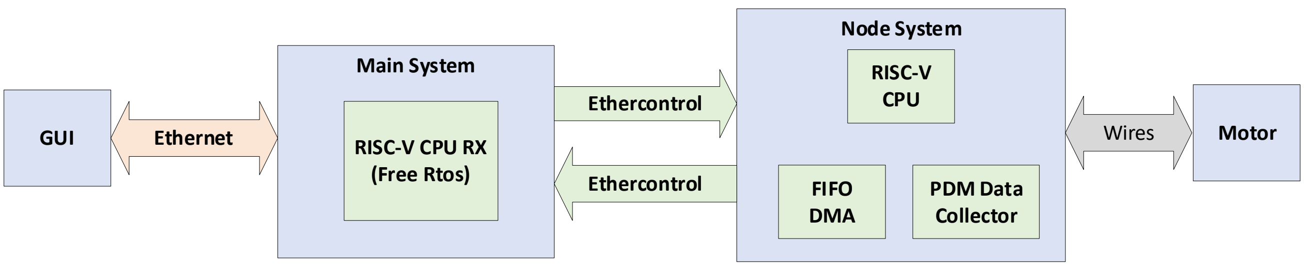

Figure 7.43. User Interface Application PDM Data: Components Involved in PDM Data Collection Process

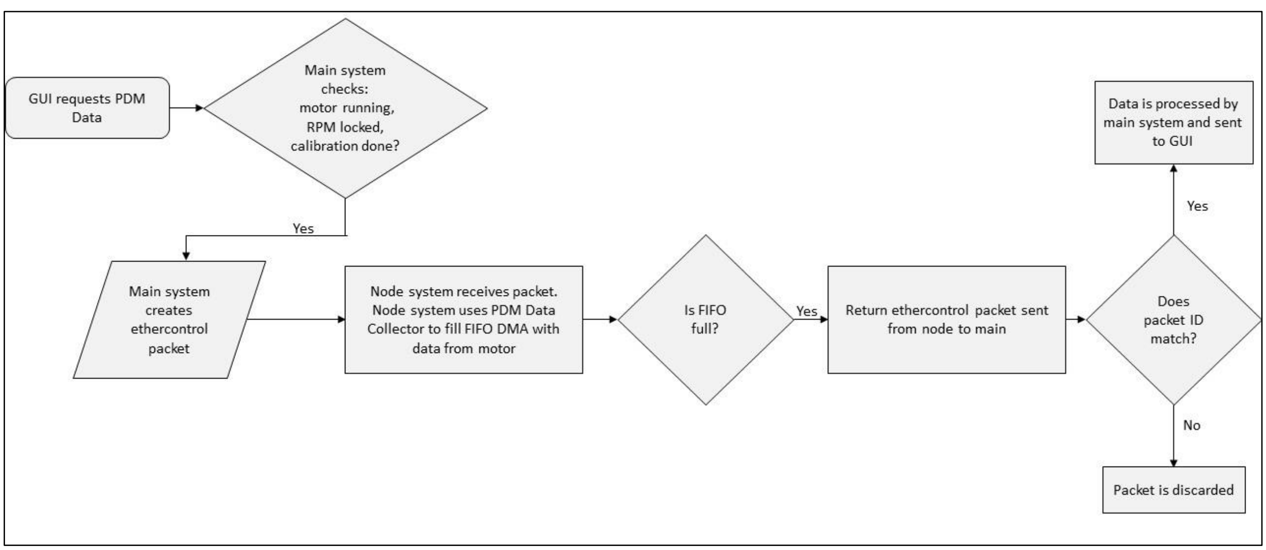

Figure 7.44. User Interface Application PDM Data: Process Flow for PDM Data Collection

7.10.4. Collecting PDM Data from the Simulated Faulty Motor Using the Switches

7.10.4.1. Introduction and Purpose of the Switches

To show the benefits of the PDM data collection, the Automate demo setup includes three switches that disable/enable the resistors which placed in series with each phase of the 3-phase BLDC motor. The resistor on each phase of the motor is disabled when the switch is tilted away from the resistor.

If any switch is tilted away from the other switch, this setup emulates imbalance load on the motor and hence exhibits faulty motor behavior. This faulty behavior may not be observable to a human, which is where the benefits of PDM data collect comes in.

PDM data collection collects the current data of the motor. In the Automate Stack main system, there is a convolutional neural network IP that has been trained on the current data 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.

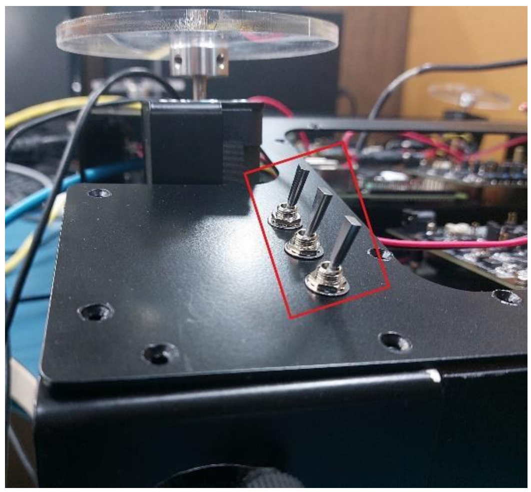

Figure 7.45. Three Resistor Switches Tilted Towards the Resistors

7.10.4.2. Sample Workflow to Collect Faulty PDM Data with the Switches

To collect the faulty PDM data with the switches, follow the sample workflow below:

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

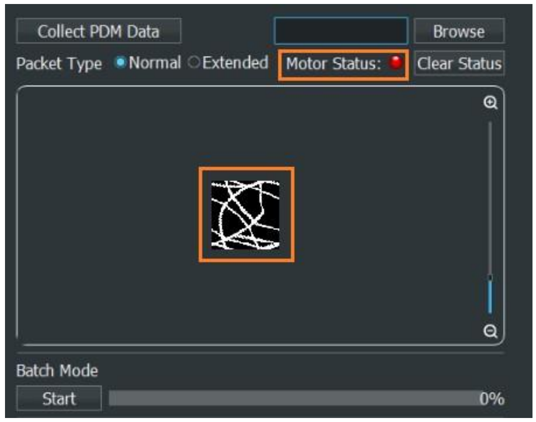

Figure 7.46. PDM Data Collection: Example PDM Image of a Faulty Motor

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

7.11. Testing the Node Peripherals

To test the node peripheral, perform the following:

1. Connect the I2C 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.

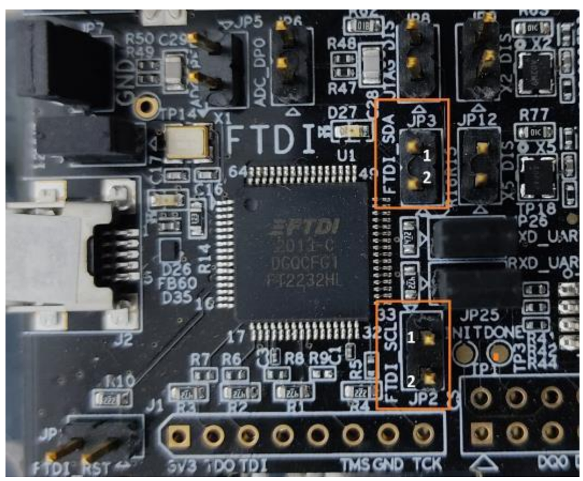

Figure 7.47. Certus-NX Board – I2C Connection

7.11.1. Connecting in the Total Phase Control Center

To connect to the Total Phase Control Center, perform the following:

1. Open Total Phase Control Center.
2. Select the **Adapter** tab and click **Connect**.

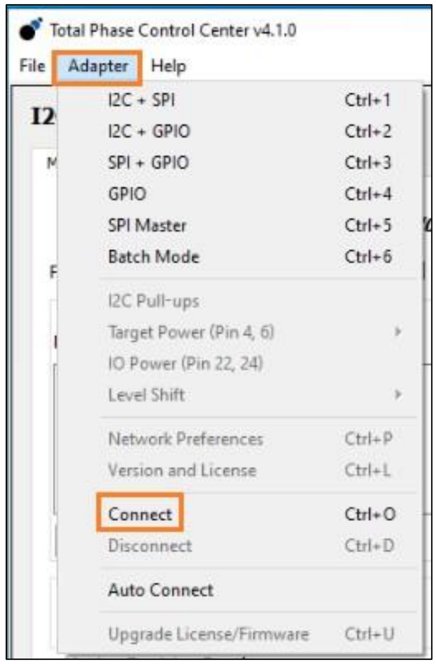

Figure 7.48. Total Phase – Adapter Setting (I2C)

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

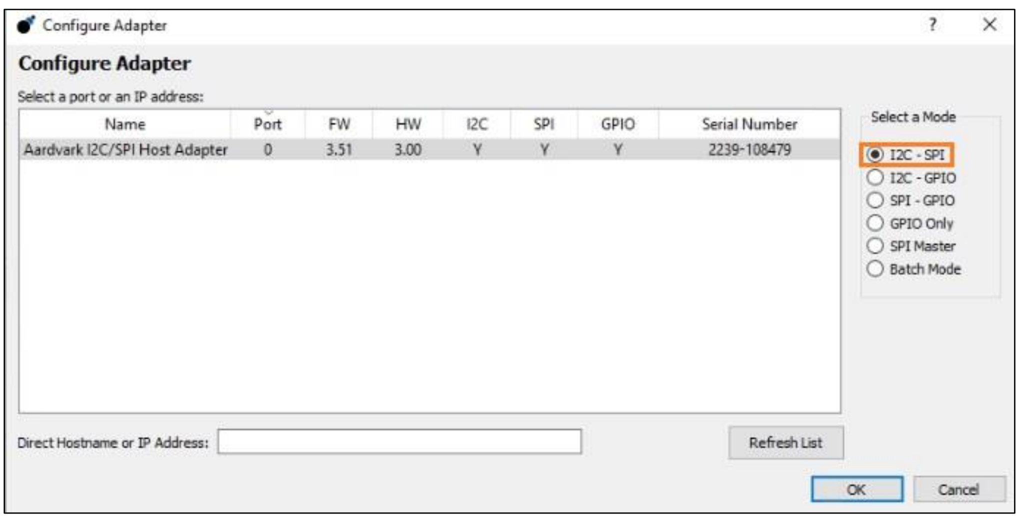

Figure 7.49. Total Phase – I2C SPI Selection

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.

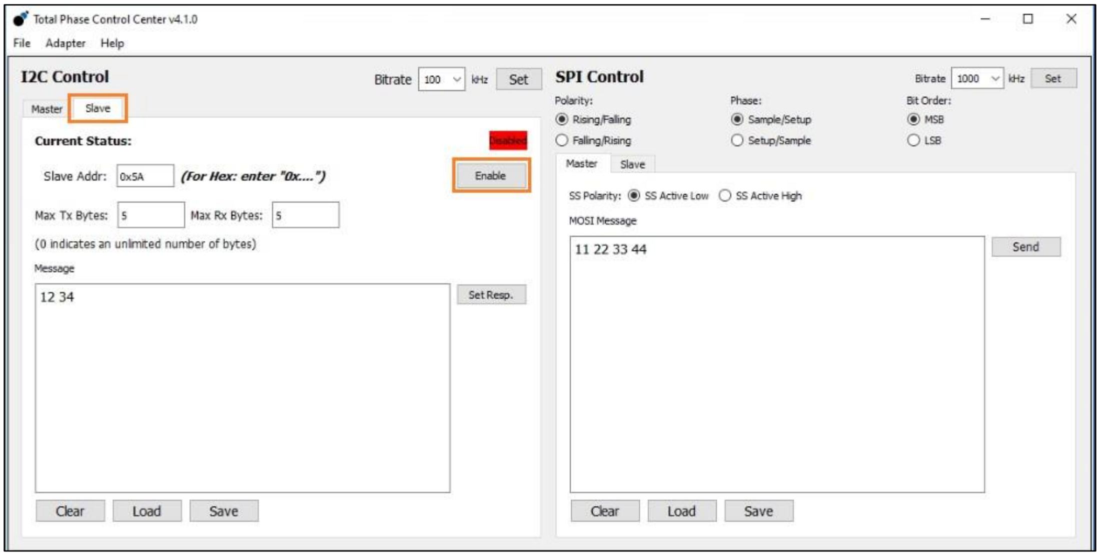

Figure 7.50. Total Phase – I2C Control (I2C)

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

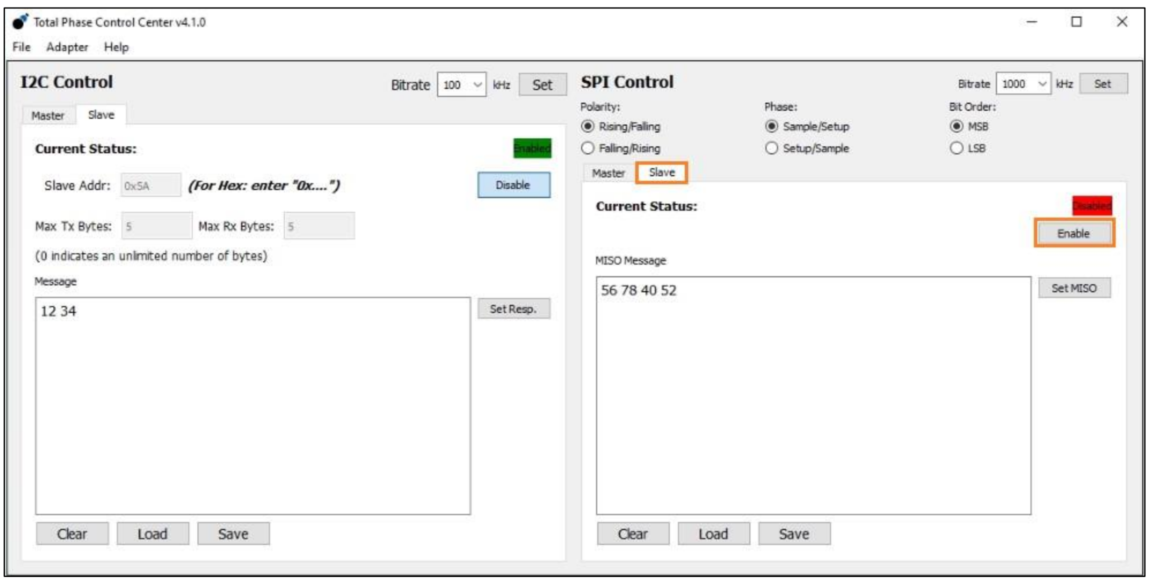

Figure 7.51. Total Phase – SPI Control: SPI Enable

9. Once you enabled both the I2C and SPI control, the display looks similar to Figure 7.52 and the connection status shows in the transaction log.
10. Update the Message in the I2C control window and click **Set Resp**. Update the MISO Message in the SPI control window and click **Set MISO** as shown in Figure 7.52.

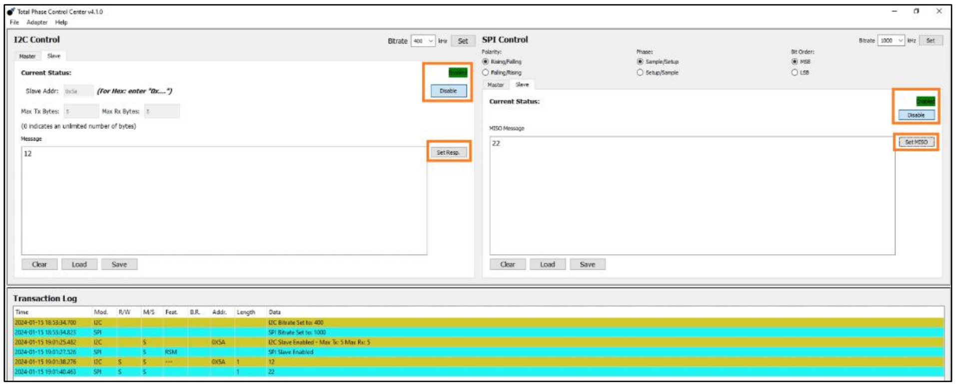

Figure 7.52. Total Phase – I2C and SPI Control Enabled

7.11.2. I2C

To perform the Write Operation in I2C, follow the steps below:

1. In the Automate 4.0 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 (you can write any data from 0x00 to 0xFF).
7. Click **Write**.

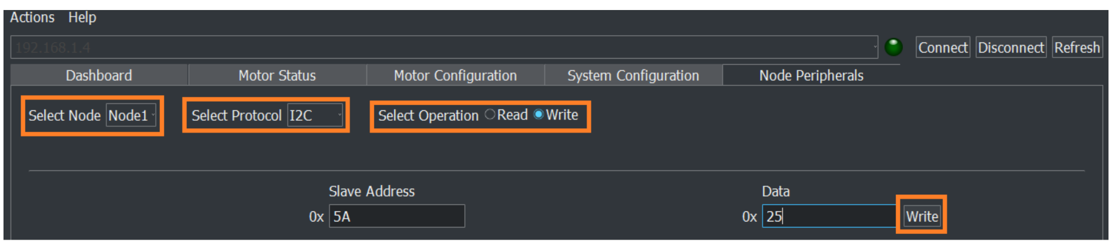

Figure 7.53. Application Software – Node Peripherals: I2C Write

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

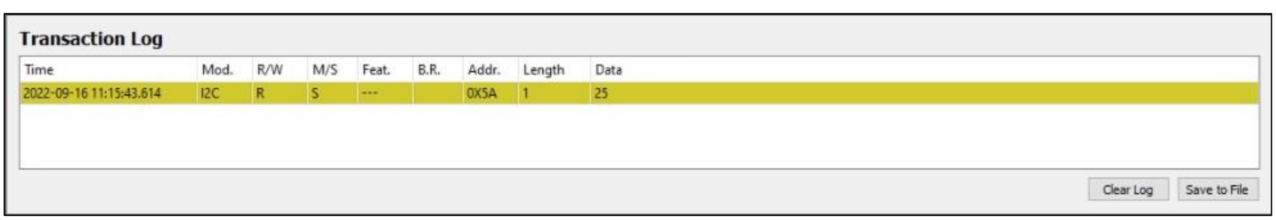

Figure 7.54. Total Phase – Transaction Log-I2C

To perform the Read Operation in I2C, follow the steps below:

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

Figure 7.55. Total Phase Aardvark I2C Control User Interface

2. Check the transaction log.

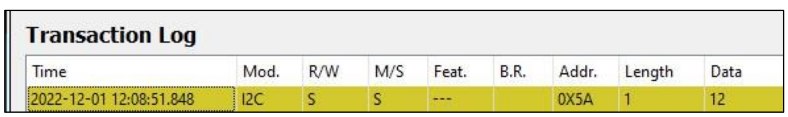

Figure 7.56. Total Phase – Transaction Log - I2C

3. In the Node Peripherals tab of the Automate 4.0 user interface, select **Read** as the **Operation**.
4. Click **Read**. The data value read must be 12.

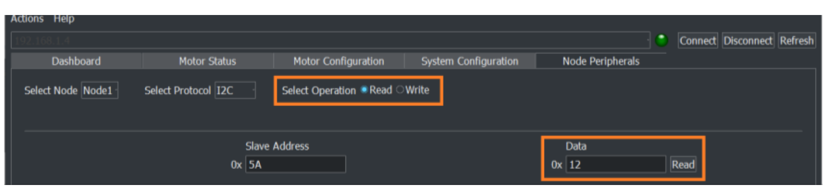

Figure 7.57. Application Software – Node Peripherals: I2C-Read

7.11.3. Modbus

Notes:

- The UART cable from the UART port of the Node system must be connected to the PC/Laptop.
- Required tool: Docklight v2.4 or later

To configure the Modbus, perform the following:

1. In the Automate 4.0 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).

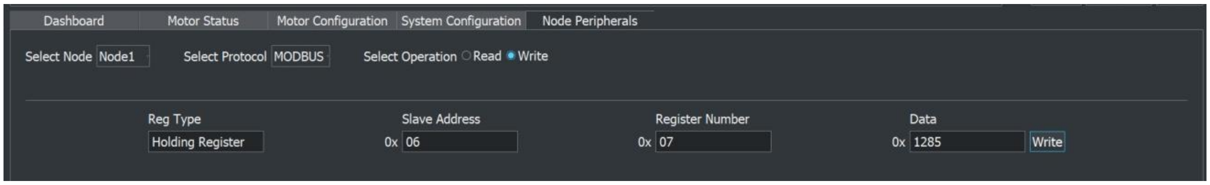

Figure 7.58. Application Software - Node Peripherals: Modbus-Write

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

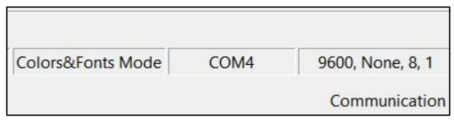

Figure 7.59. Docklight COM Selection

7. Select the last USB serial port in the list, as shown in Figure 7.60.

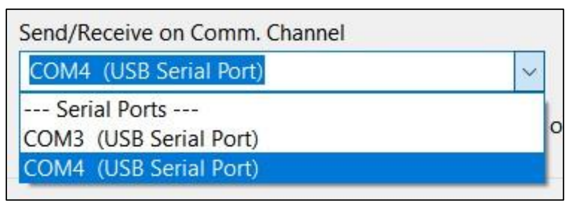

Figure 7.60. USB Serial Port Selection

8. Select the Baud Rate: **115200**.

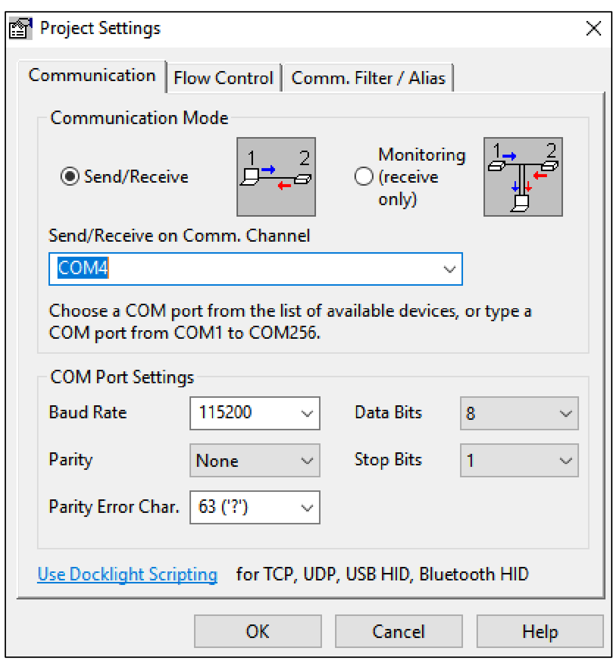

Figure 7.61. Docklight – Project Setting

9. Click **Run**.

Figure 7.62. Docklight – Run

10. In the Automate 4.0 user interface, click **Write**.

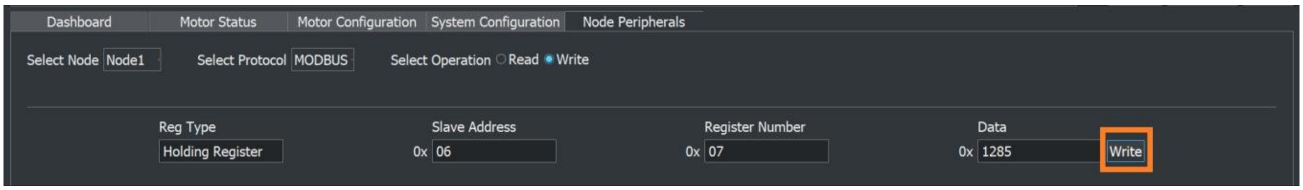

Figure 7.63. Application Software – Node Peripheral: Modbus Write

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

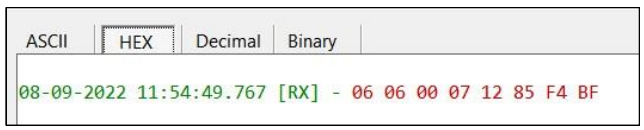

Figure 7.64. Docklight Prints

8. Known Issues

8.1. Motor RPM Shown Zero Intermittently in the Automate 4.0 Graphical User Interface

When the motor is running, the Motor RPM of some nodes is shown zero and back to the correct RPM value intermittently on Automate 4.0 user interface.

- **Workaround:** There is no workaround for this issue, but this will be fixed in future patch release.
- **Impact:** No impact to the system functionality. The impacted nodes are still running, and the motor RPM returns to the correct RPM value.

8.2. Node Peripherals I2C Read/Write Operation

I2C read and write operation under Automate 4.0 User Interface Node Peripherals is not configuring nodes I2C correctly.

- **Workaround:** There is no workaround for this issue but will be fixed in future patch release.
- **Impact:** You will not be able to use the I2C interface on the node.

8.3. Multiboot MCS file

The deployment tool is not able to generate multiboot MCS file with Primary and Golden bitstream and Primary and Golden U-Boot.bin correctly.

- **Workaround:** Program the Primary and Golden U-Boot.bin files after programming the multiboot MCS file.
- **Impact:** You need to program two U-Boot.bin after the multiboot MCS file is programmed.

8.4. Multiboot Intermittent Failure

Intermittent Failure happens where multiboot jumps to Primary pattern after pressing SW button where it supposed to jump to Golden pattern.

- **Workaround:** No workaround. You need to press the SW2 button again.
- **Impact:** You may not be able to jump to the Golden pattern after pressing the SW2 button.

Appendix A. Installing the Graphical User Interface Application (PC)

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

To install the user interface application, perform the following:

1. Download or locate the Lattice Automate 4.0 Installer.
2. Double-click on the installer to install the application.

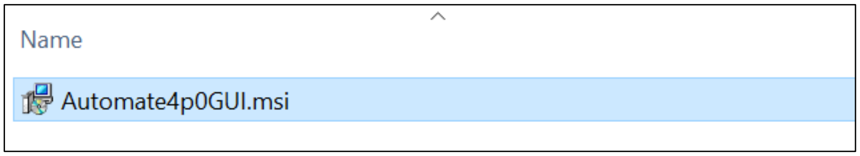

Figure A.1. Installer Directory

3. Wait for the installation to complete.

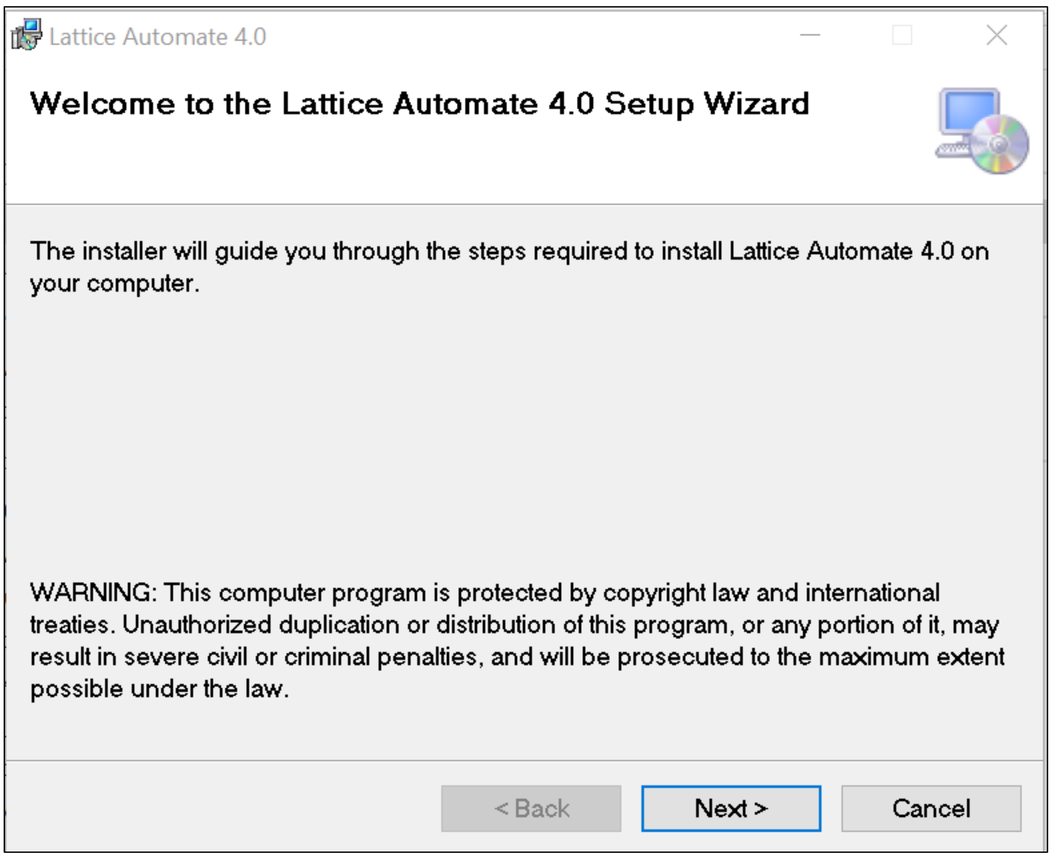

Figure A.2. Automate 4.0 Initial Install Setup

4. Click **Next**.

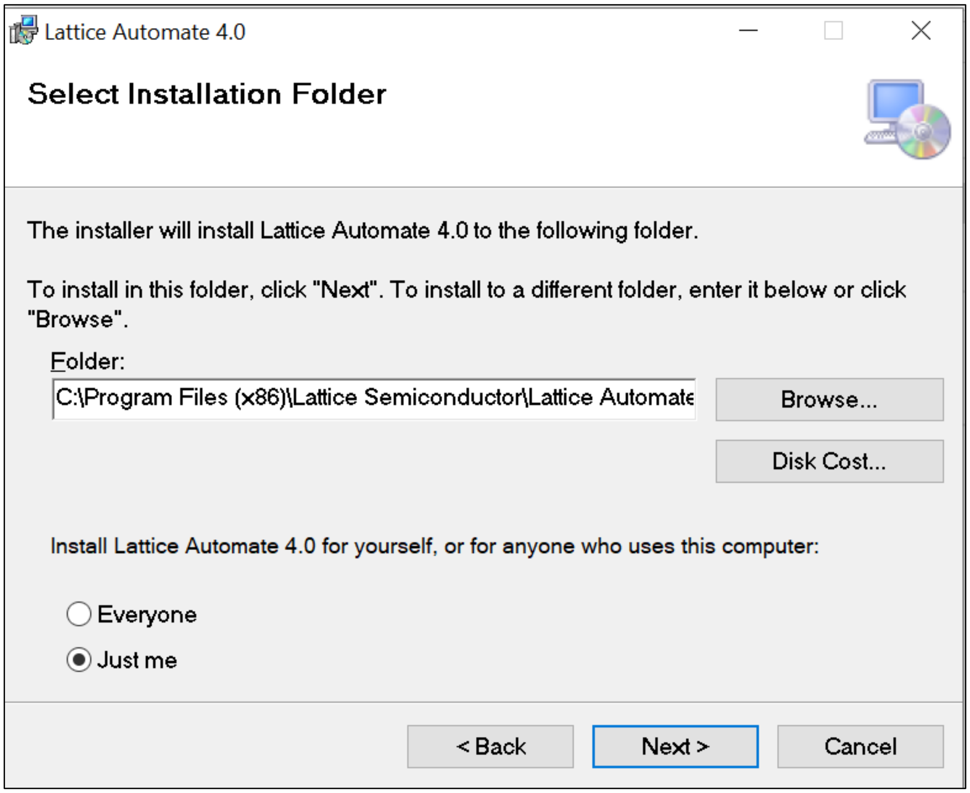

Figure A.3. Automate 4.0 Next Step

5. Click **Next**.

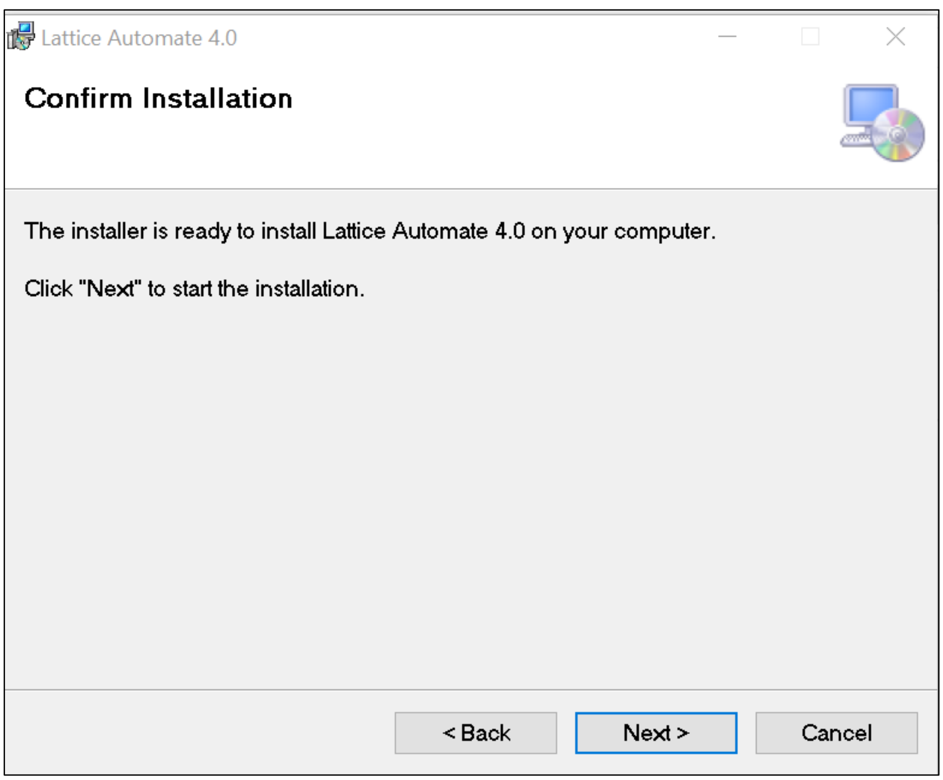

Figure A.4. Automate 4.0 Next Step

6. Wait for the installation to complete.

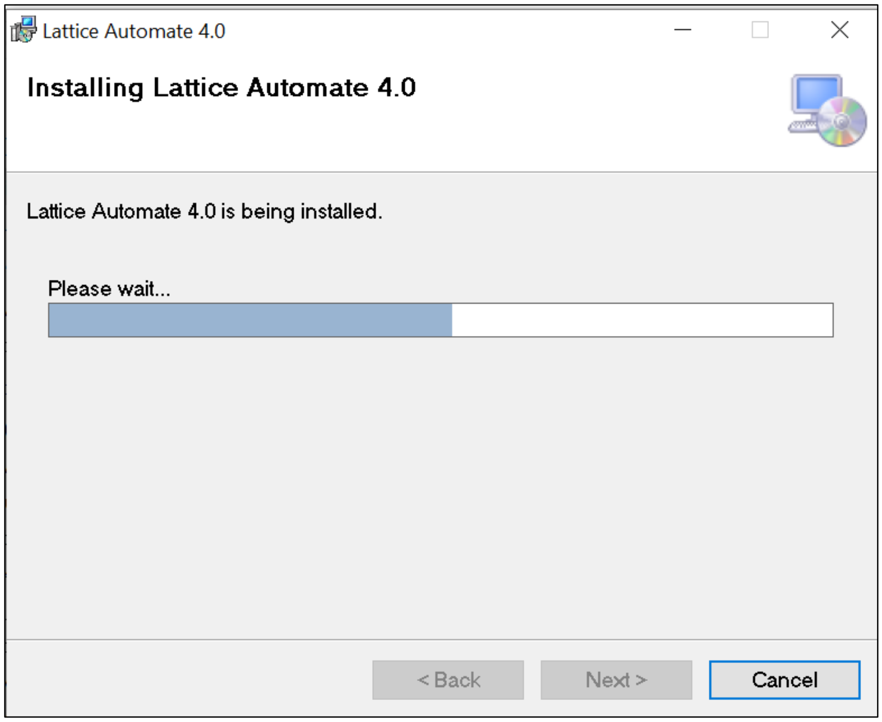

Figure A.5. Automate 4.0 Installation in Progress

7. Click **Finish**.

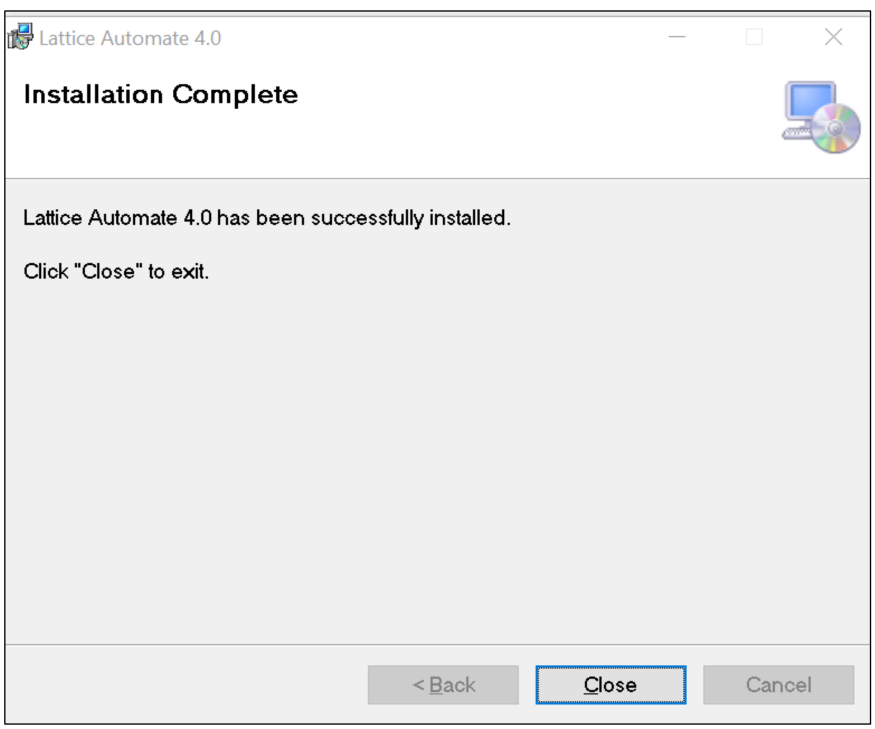

Figure A.6. Automate 4.0 Installation Finish

Appendix B. Programming the Main and Node Systems with Automate Stack Bitstreams

B.1. Main System

This section provides the procedure for programming the SPI Flash on the Avant 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 to the board:

- Pin 1 and Pin 2 on both JP1 and JP2 switches to enable UART.
- For FPGA executables programming, use SW7 in JTAG mode. Once programmed, switch to MSPI.

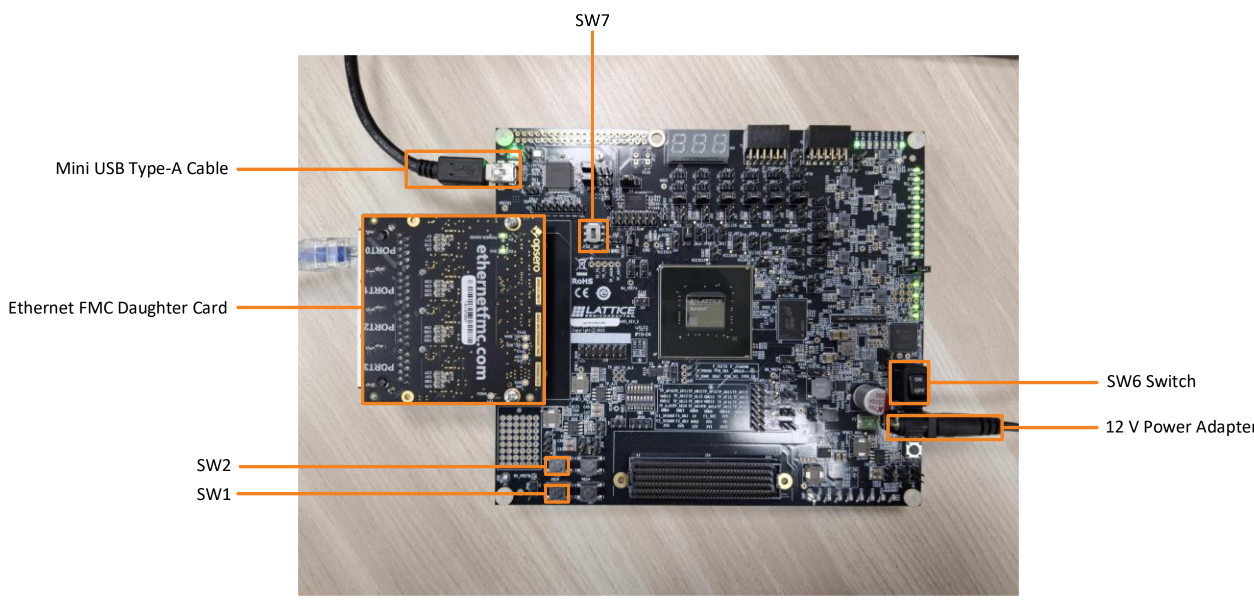

Figure B.1. Main System Board

If you are programming the Main System board for the first time, perform the following one-time procedure to avoid the firmware booting issue. When the boards are delivered from the foundry, these are programmed in quad mode. But when the SPI Flash Controller starts with serial mode (default) for commands, it selects the quad mode for the read operation.

To program the SPI Flash in Quad mode:

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

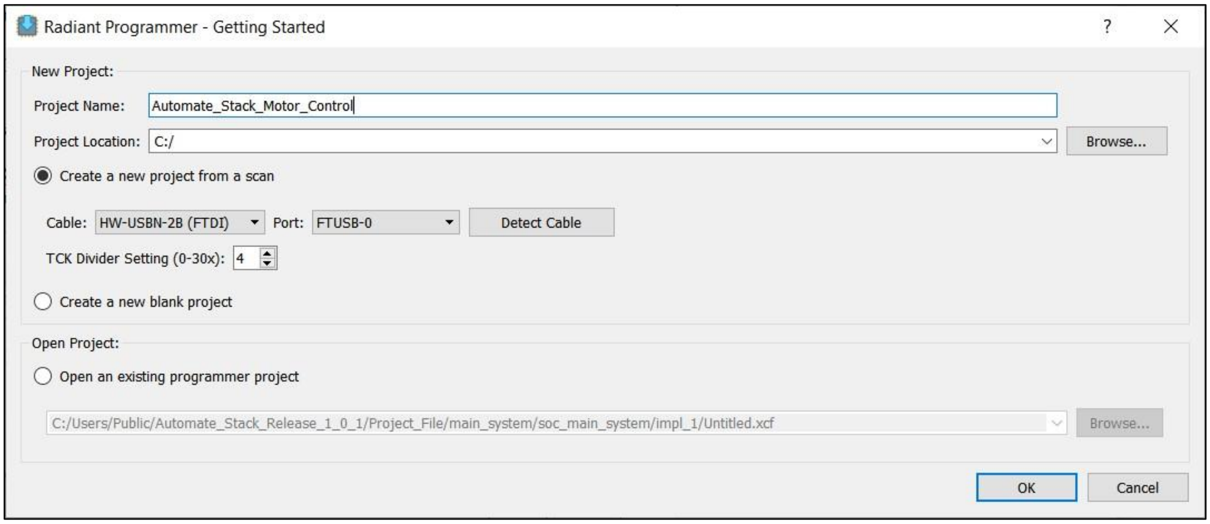

Figure B.2. Radiant Programmer – Default Screen

3. Click **OK**.

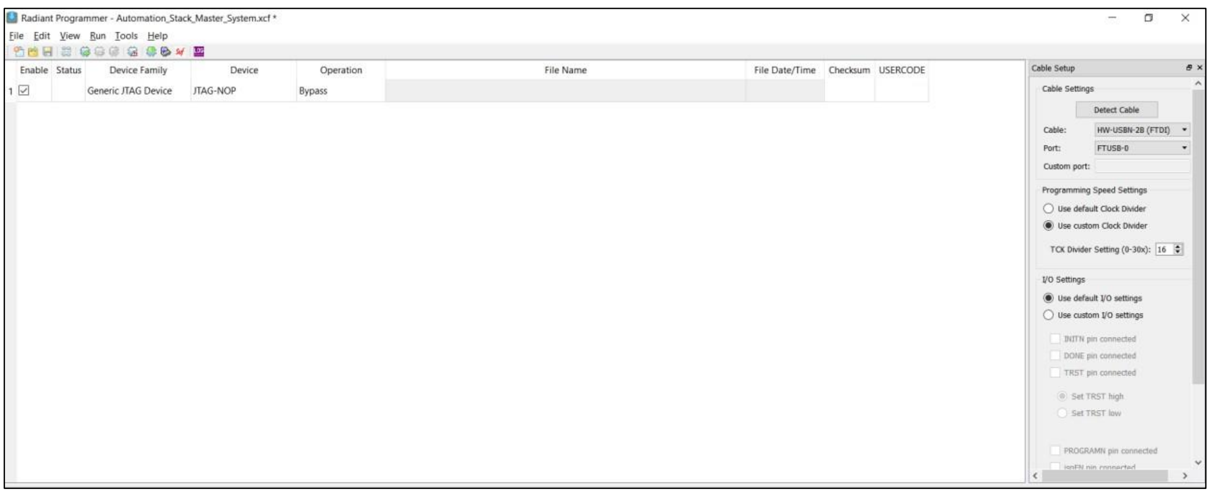

Figure B.3. Radiant Programmer – Initial Project Window

4. In the Radiant Programmer, select **LAV-AT_ENG** for **Device Family** and **LAV-AT-E70ES1** for **Device** or detect automatically as shown in Figure B.4.

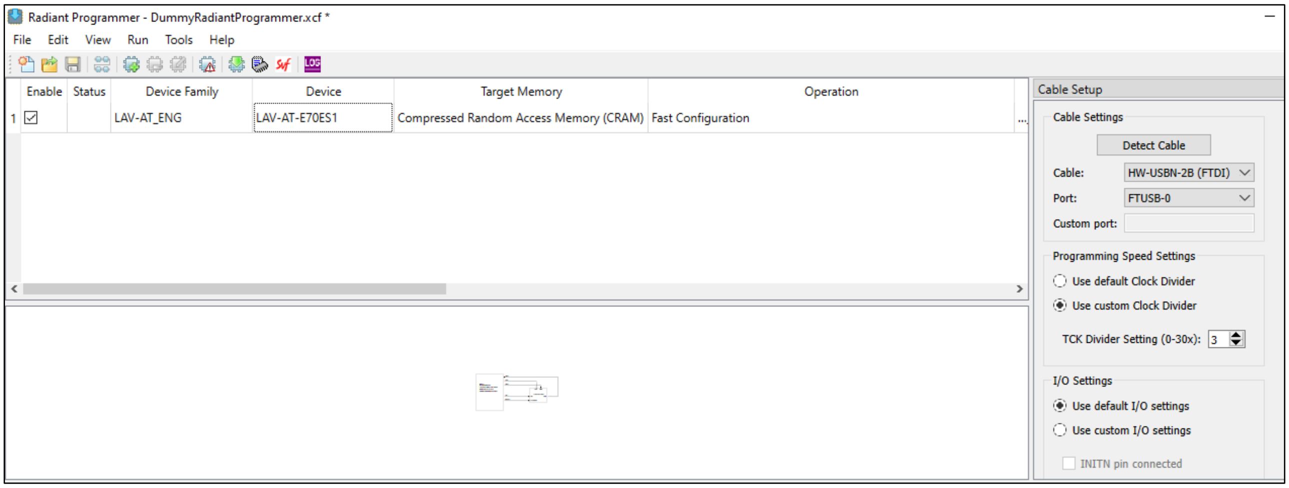

Figure B.4. Radiant Programmer - Device Selection

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

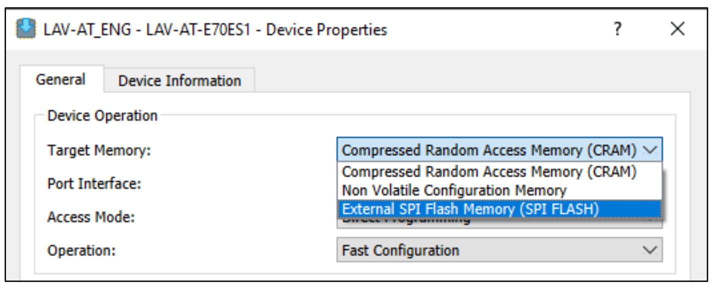

Figure B.5. Radiant Programmer - Device Operation

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**
   b. Under SPI Flash Options, select the options below:
      - Family – **SPI Serial Flash**
      - Vendor – **Winbond**
      - Device – **W25Q512JV**
      - Package – **16-pin SOIC**

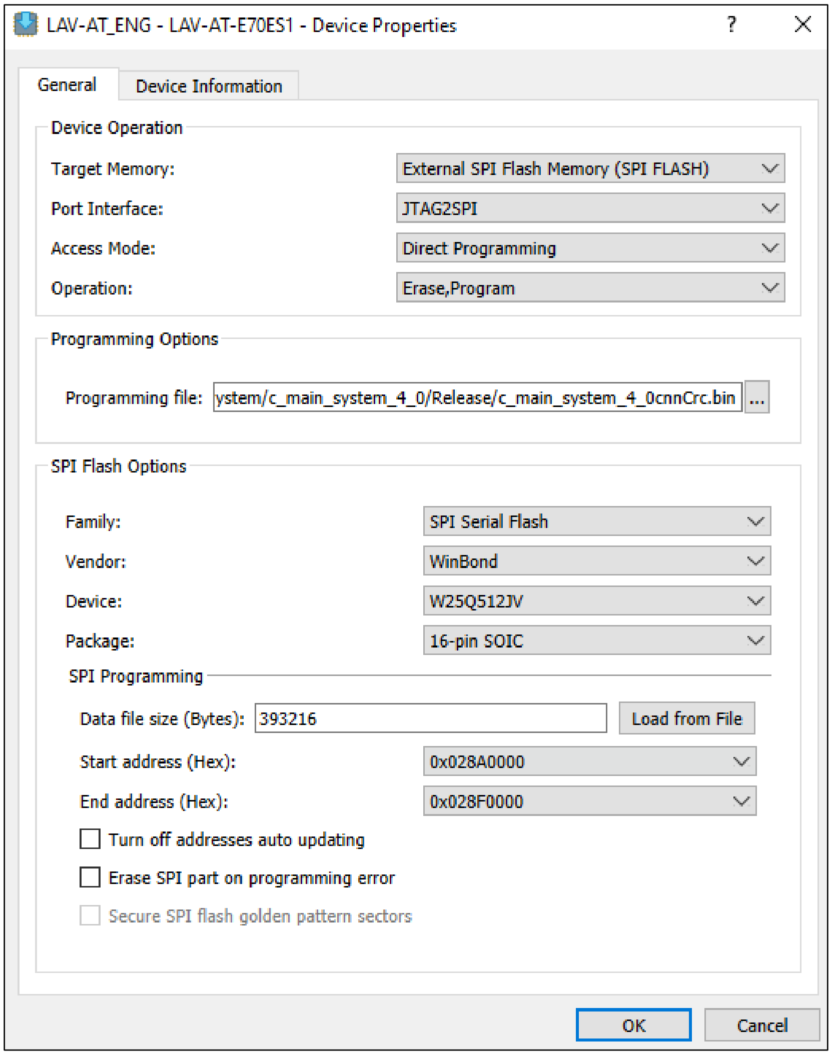

Figure B.6. Radiant Programmer – Program Primary Main System Binary

7. In the Programming Options, select the *c_main_system_4_0cnnCrc.bin binary* file.
   a. Make sure that the following addresses are correct:
      - Start Address (Hex) – **0x028A0000**
      - End Address (Hex) – **0x028F0000**
8. Click **OK** and then click the Program Device icon or the menu item **Run ->Program Device**.
9. To program the bitstream file, select the options as shown in Figure B.7.

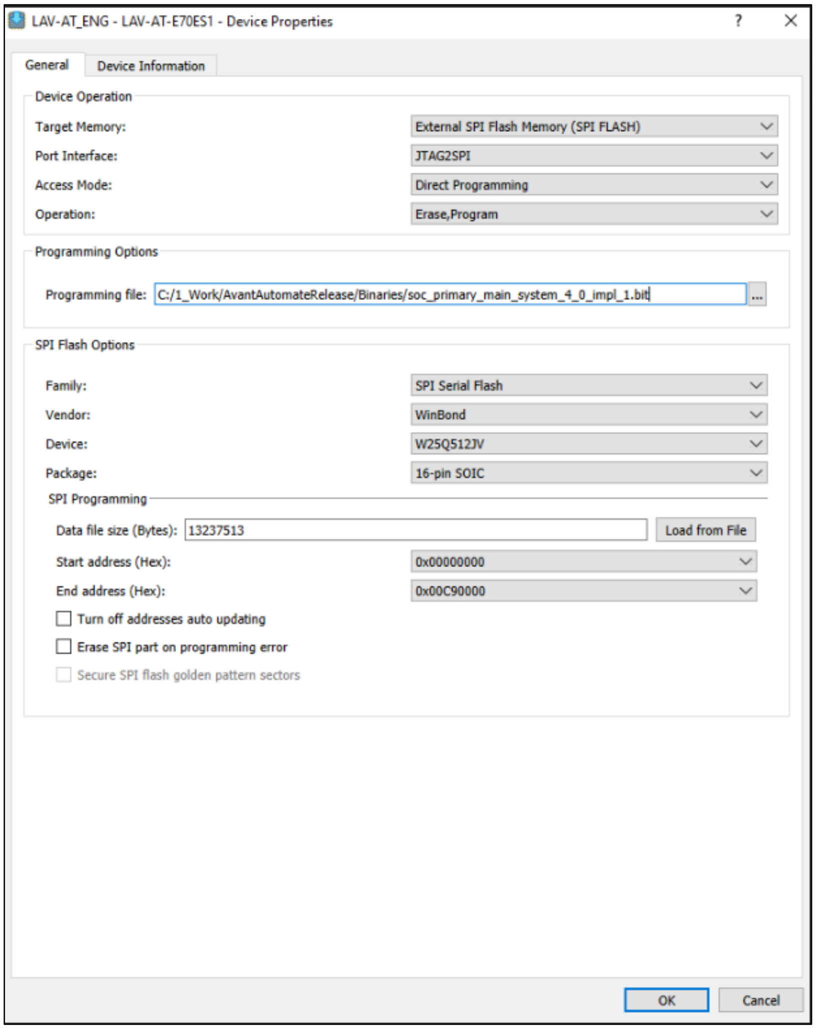

Figure B.7. Radiant Programmer – Program Primary Main System Bitstream

10. In the Programming Options, select the *soc_primary_main_system_4_0_impl_1.bit* bitstream file in the Programming file. Make sure that the following addresses are correct:
    - Start Address (Hex) – **0x00000000**
    - End Address (Hex) – **0x00C90000**
11. Click **OK** and then click the Program Device icon or the menu item **Run > Program Device**.
12. To program the Golden Main System binary, select the Golden_AppCrc.bin binary file under the Programming Options. Refer to Figure B.8. Make sure that the following addresses are correct:
    - Start Address (Hex) – **0x02800000**
    - End Address (Hex) – **0x02850000**

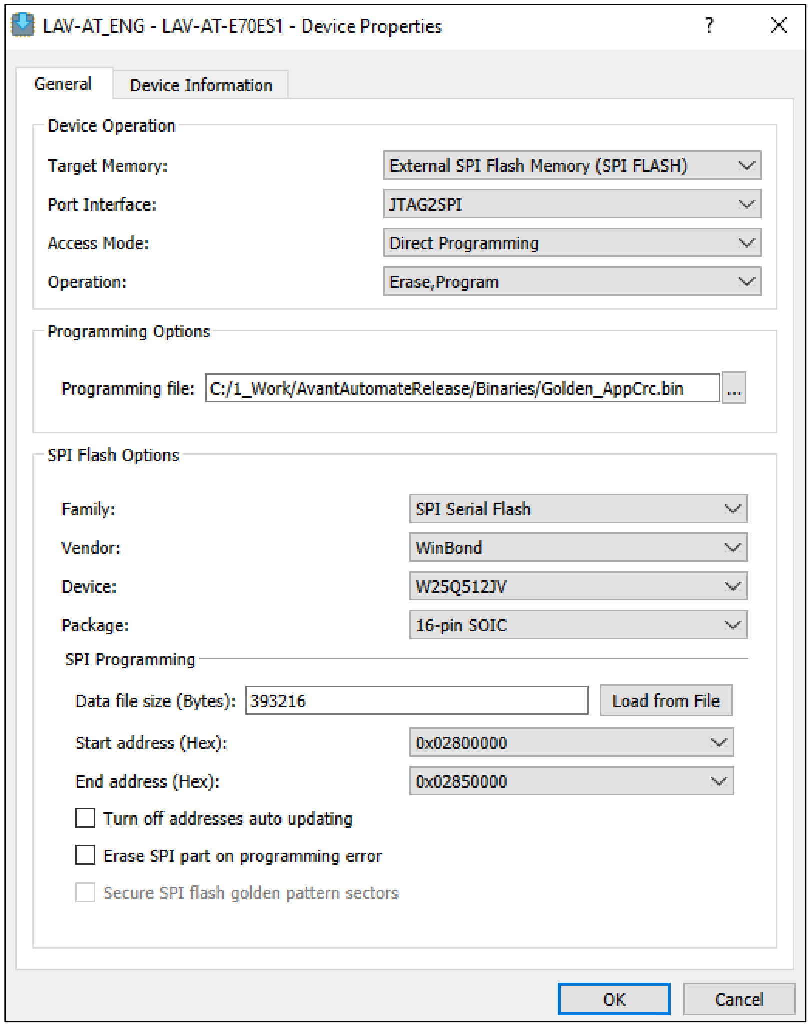

Figure B.8. Radiant Programmer – Program Golden Main System Binary

13. Click the Program Device icon  or the menu item **Run > Program Device**.
14. To program the uBoot binary, select the primary_u-boot.bin binary file under the Programming Options. Refer to Figure B.9. Make sure that the following addresses are correct:
    - Start Address (Hex) – **0x021A0000**
    - End Address (Hex) – **0x02200000**

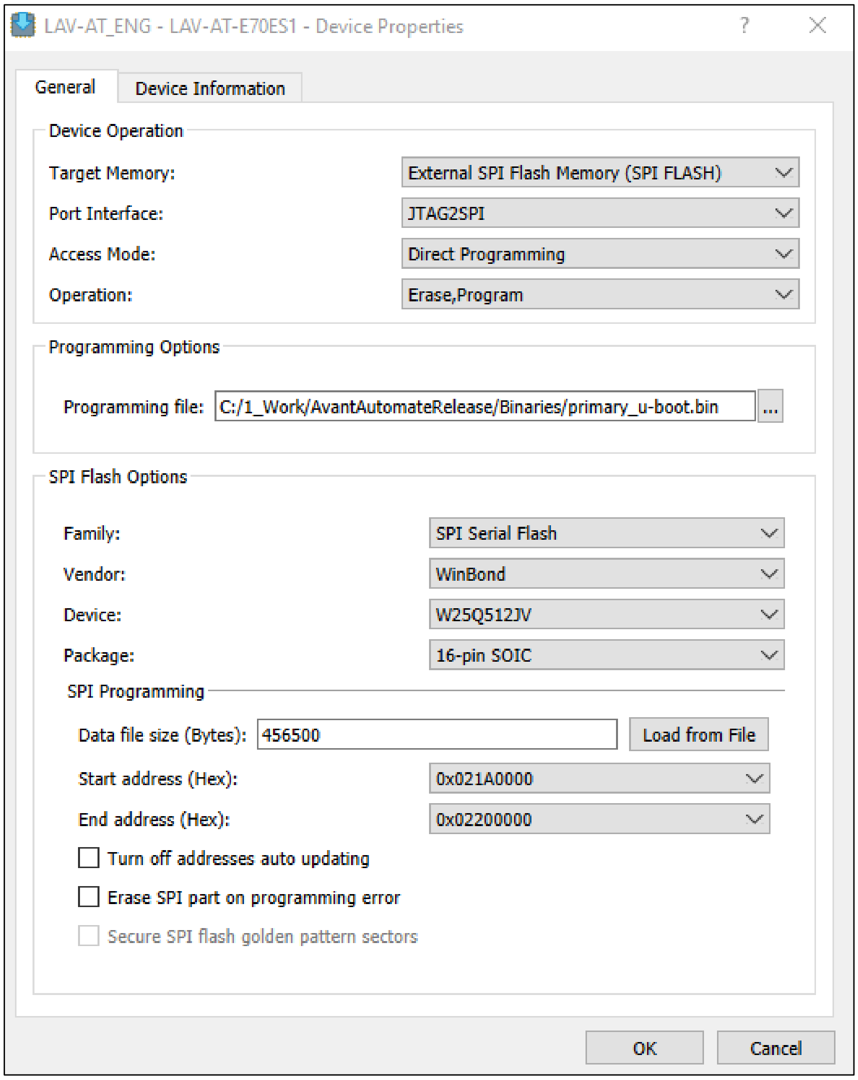

Figure B.9. Radiant Programmer – Program Primary u-Boot Binary

15. Click **OK** and then click the Program Device icon or the menu item **Run >Program Device**. After that power cycle the board.
16. Power cycle the Avant 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 must be programmed into the SPI Flash. These files are programmed to the same SPI Flash, but in different addresses:

- Bitstream
- Binary

Board Jumper Connections

The following jumpers must be connected to the board:

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

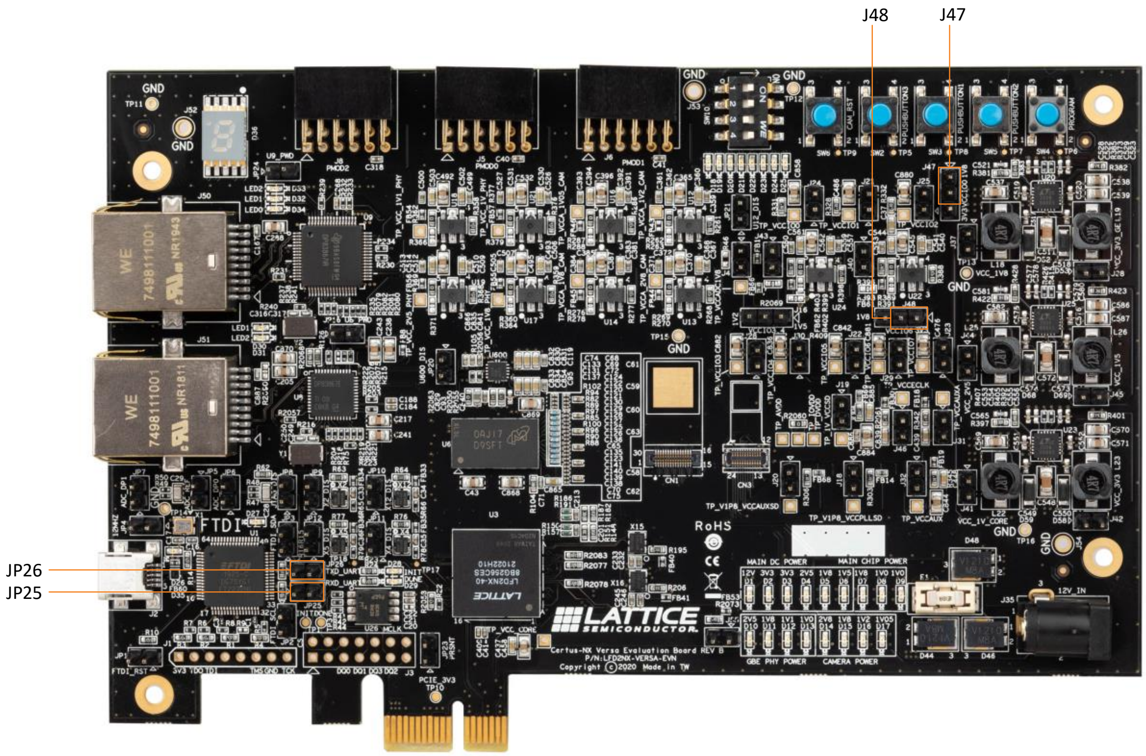

Figure B.10. Node System Jumper Connection

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**.

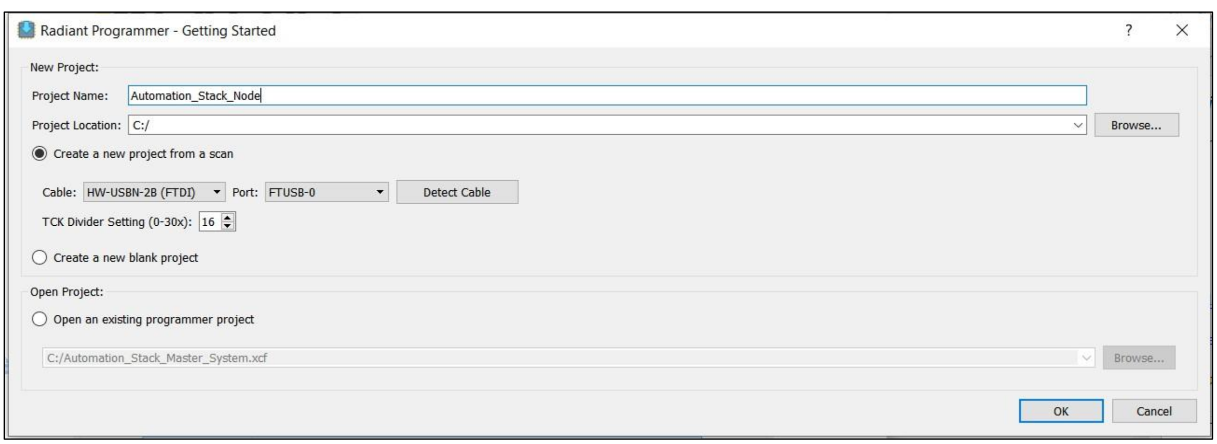

Figure B.10. Radiant Programmer - Default Screen (Node System)

3. Click **OK**.

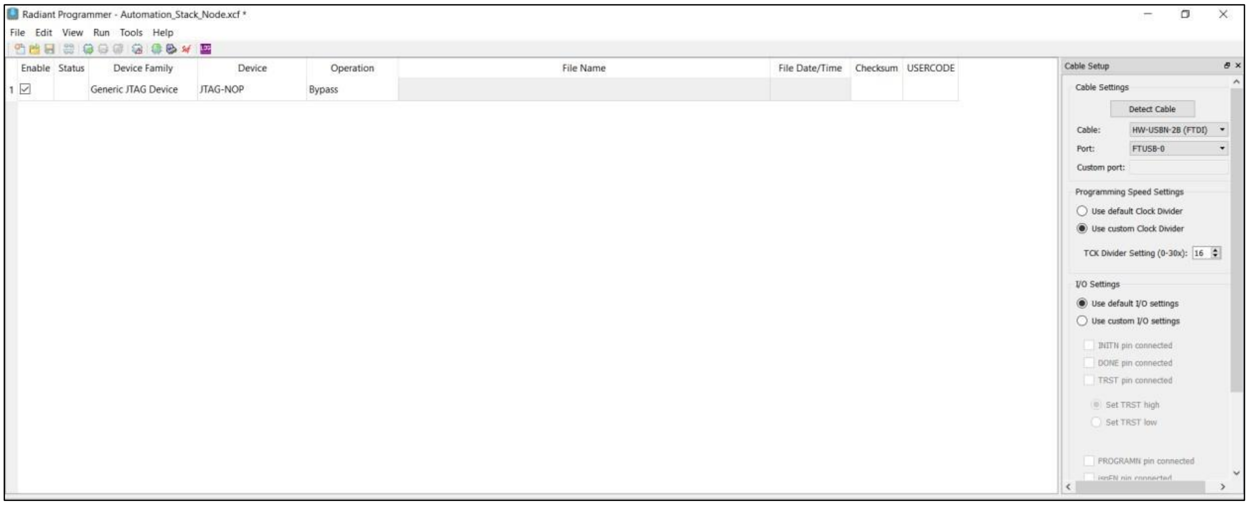

Figure B.11. Radiant Programmer – Initial Project Window (Node System)

4. In the Radiant Programmer main interface, select **LFD2NX** for **Device Family** and **LFD2NX-40** for **Device** as shown in Figure B.12.

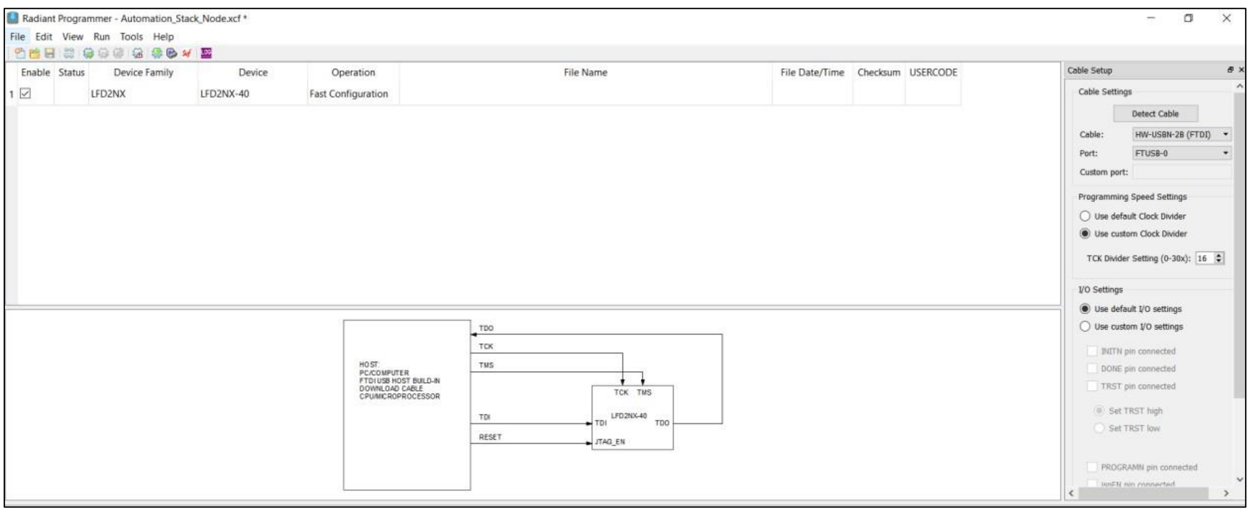

Figure B.12. Radiant Programmer - Device Properties (Node System)

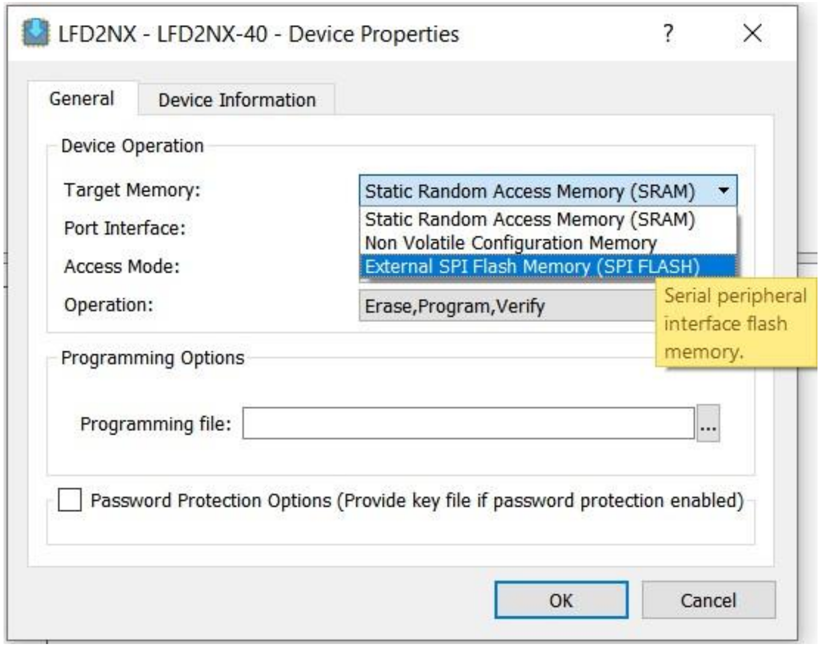

Figure B.13. Radiant Programmer – Device Operation (Node System)

5. Before programming, you must erase the flash memory. 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 all**
   b. Under SPI Flash Options, select the options below:
      - Family – **SPI Serial Flash**
      - Vendor – **Micron**
      - Device – **MT25QU128**
      - Package – **8-pin SOP2**

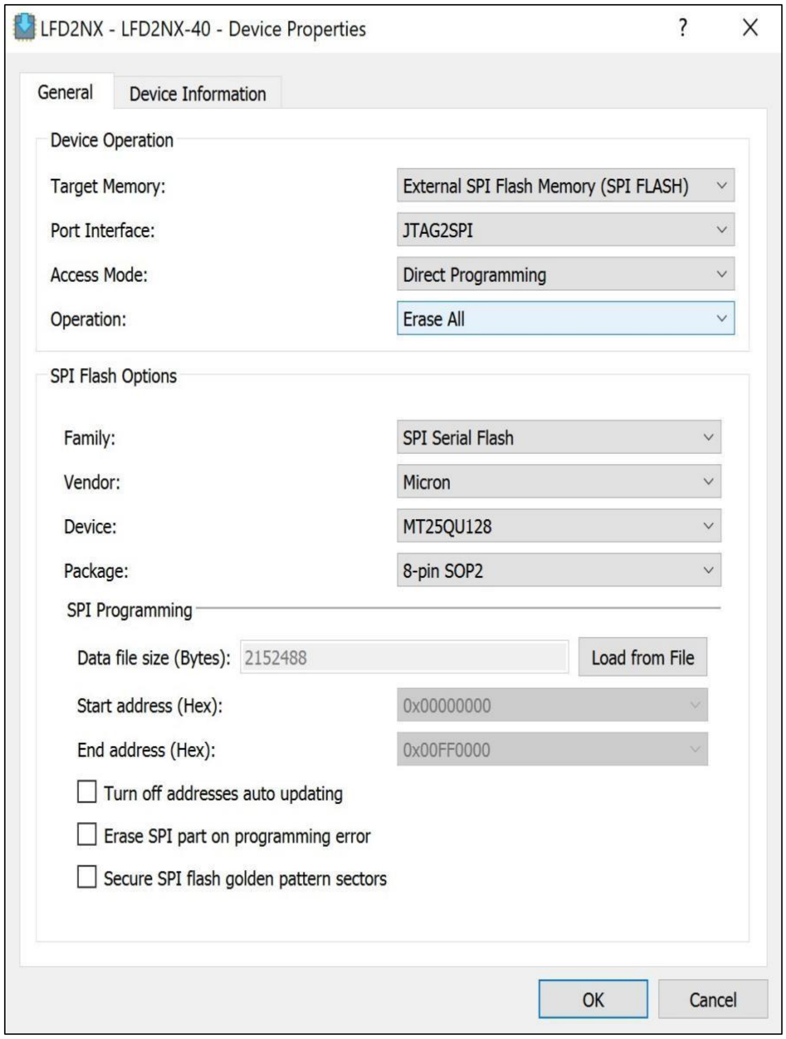

Figure B.14. Radiant Programmer – Erase All (Node System)

6. Click **OK** and then click the Program Device icon or the menu item **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.15.

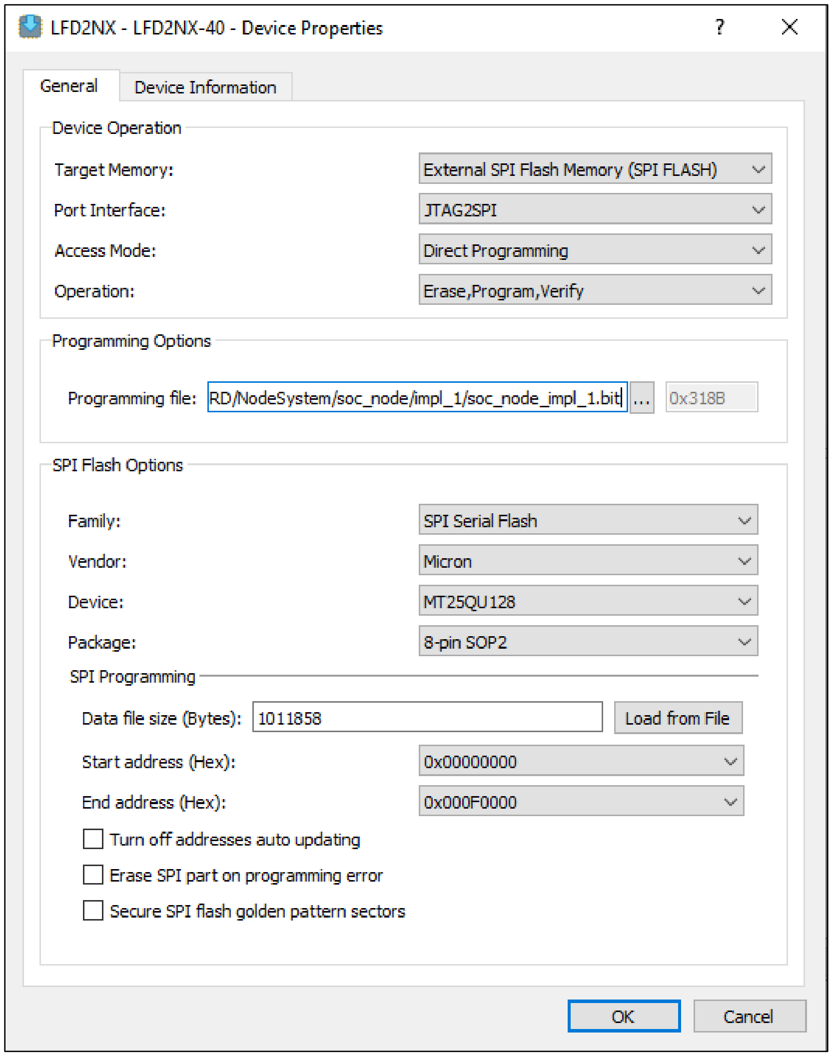

Figure B.15. Radiant Programmer – Bit Stream Flashing Settings (Node System)

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) – **0x000F0000**

9. Click the Program Device icon or the menu item **Run ->Program Device**.
10. To program the firmware, select the options as shown in Figure B.16 below.
    a. Under Programming Options, select the *c_node_system_4_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 the menu item **Run ->Program Device**.

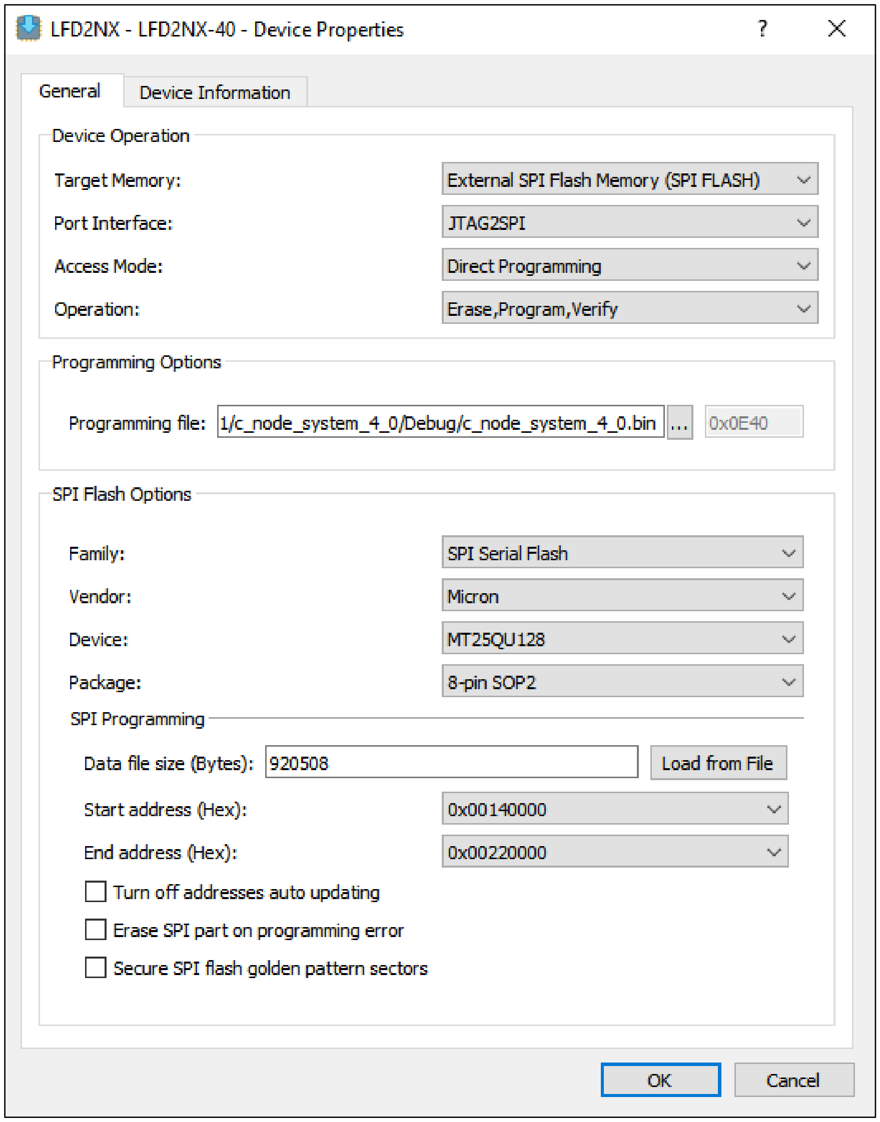

Figure B.16. Radiant Programmer – Binary Flashing Settings (Node System)

Note: After programming the Boards, power cycle each Certus-NX Versa board as shown in Figure B.17, and then press the system reset button SW3.

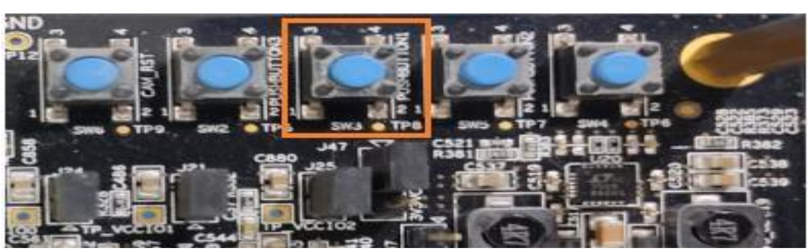

Figure B.17. SW3 Button of the Certus-NX Versa Board

Appendix C. Debugging the Main System (Avant-E Evaluation Board)

C.1. Debugging the Primary Main System Standalone Project

To debug the Primary Main System standalone project:

1. Connect the UART Cable between Host PC to Avant-E Board.
2. Check the Status of UART Cable in Device Manger.
3. Go to the Port section of Device Manager.
4. Check Connection status is showing or not.
5. After programming the Primary binary and bitstream file, refer to Appendix B. Programming the Main and Node Systems with Automate Stack Bitstreams and follow the below steps to check the prints on propel terminal.
6. Following executable are needed to be program. To program the files, refer to Appendix B. Programming the Main and Node Systems with Automate Stack Bitstreams.
    a. Program Primary main system CRC binary file: **c_main_system_4_0_cnnCrc.bin**
    b. Program Primary main system bitstream file: **soc_main_system_4_0_impl_1.bit**
7. Open the Propel terminal.
8. Double-click the terminal button (see Figure C.1) in the bottom of Propel SDK window.

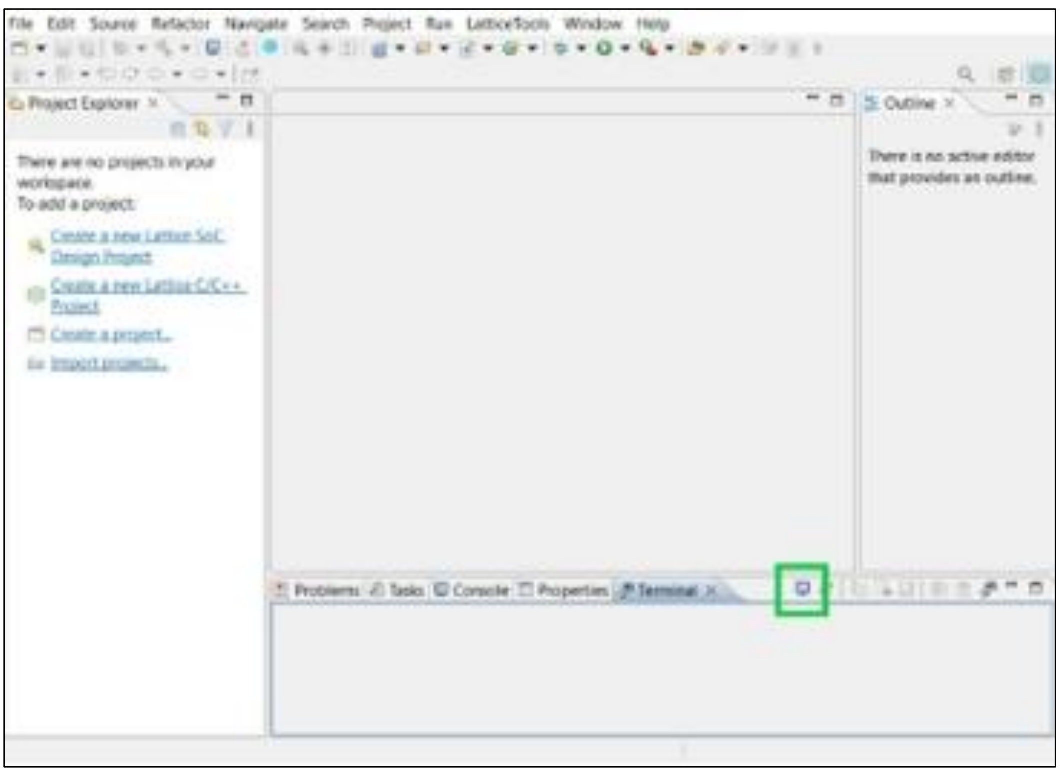

Figure C.1. Open Propel Terminal

9. The Propel terminal window appears as shown in Figure C.2.

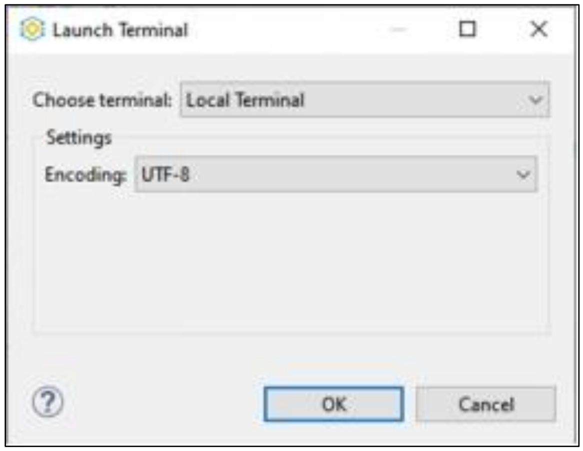

Figure C.2. Propel terminal selection window

10. Choose **Serial Terminal** for the terminal as shown in Figure C.3.
11. Select the right serial port, drop down the serial port, and select the last COM.
12. Select the Baud rate as **115200**.
13. Click **OK**.

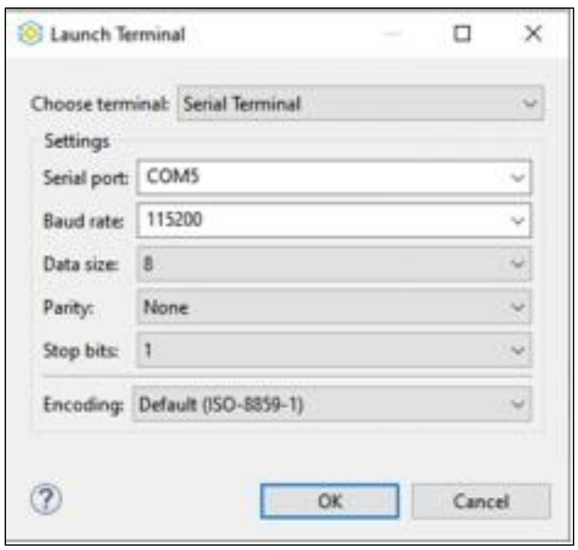

Figure C.3. Propel terminal – Com and Baud Rate Selection

14. If you are not receiving any Ethernet packet, press the **SW3** reset button on the board or apply the settings:
    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**.
15. Wait for 40 seconds for the prints to be available in the log section.

```
U-Boot SPL 2024.01-00035-g3ec17456c6-dirty (Sep 25 2024 - 14:36:14 +0800)
Trying to boot from MTD


U-Boot 2024.01-00035-g3ec17456c6-dirty (Nov 20 2024 - 14:37:09 +0800)

CPU:   rv32imc
Model: lattice,riscv
DRAM:  1 GiB
Core:  11 devices, 8 uclasses, devicetree: separate
Flash: 128 MiB
Loading Environment from nowhere... OK
In:    serial@40090000
Out:   serial@40090000
Err:   serial@40090000
Net:   No ethernet found.
Hit any key to stop autoboot:  0
Reading 393216 byte(s) at offset 0x028a0000
## Checking CRC
CRC pass: 0x22ce
## Starting application at 0x80000000 ...
Main System Ready!
Automate Stack 4.0 ver 0.0.1!
Version Date 11/11/2024
System Init!
Node num 16, Node Data length 64
Before: Link Detected: 1
Polling Phy Link status....................
After: Link Detected: 31
Chain 1 Link for Main System detected
Node num 16, Node Data length 64
Broadcast END
Num node default: 16
 Org:detected_nodes 4
 Chain : 1 ACTIVE NODE: 4
 Org: 4
Chain : 1 FINAL ACTIVE NODE 4
node number 4
PHY LINK EXPECTED: 1f
BEFORE: TIMER PHY LINKS MAIN and NODES 1f
Node Num: 4
Before Node Enable: 1
After Node Enable: f
Chain 1 nodes reg : f
INIT_DONE
Sys Init
In PU Seq
PU Seq Done
StartModelAllocation.
                     StartModelAllocation - Done.
                                                 operators_size 13
                                                                  operators_size 13
                                                                                   Registering op code index 0
builtin_code '3'Registering op code index 1
builtin_code '17'Registering op code index 2
builtin_code '3'Registering op code index 3
builtin_code '3'Registering op code index 4
builtin_code '17'Registering op code index 5
```

```
builtin_code '3'Registering op code index 6
builtin_code '3'Registering op code index 7
builtin_code '17'Registering op code index 8
builtin_code '3'Registering op code index 9
builtin_code '3'Registering op code index 10
builtin_code '17'Registering op code index 11
builtin_code '22'Registering op code index 12
builtin_code '9'Successfully allocating
                                       free up ether ip flag
                                                            Starting lwIP, local interface IP is 192.168.1.4
                                                                                                            End
```

Figure C.4. Primary Main System Propel Prints

16. Press the **Reset SW3** button on the Avant-E Board to restart the prints.

C.2. Debugging the Golden Main System Standalone Project

To debug the Golden Main System standalone project:

1. Connect the UART Cable between Host PC to the Avant-E board.
2. Check the Status of UART Cable in Device Manger.
3. Go to the **Port** section of Device Manager.
4. Check the connection status is showing or not.
5. To program the following Golden binary and bitstream files, refer to Appendix B. Programming the Main and Node Systems with Automate Stack Bitstreams.
    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. Open the Propel terminal. Refer to C.1. Debugging the Primary Main System Standalone Project to perform steps 8 to 14. Once completed, the log section prints should be the same as Figure C.4 in step 15.
7. Press the **Reset** button on the Avant-E Board to restart the prints.

Appendix D. Connecting to the Ethernet

D1. Connecting Between the Main System and the Laptop/PC

To connect the main system and the PC:

1. Set up the following hardware connections, as shown in Figure D.1.
   a. Connect the ethernet cable between the Avant-E 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:
   - The firewall must be disabled while doing this activity. The *.bit* and *.bin* files must be programmed in the main system and node systems. Refer to Appendix B. Programming the Main and Node Systems with Automate Stack Bitstreams and B.1. Main System
   - sections.

D.2. Setting the Ethernet Configuration

To set up the ethernet configuration, perform the following:

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

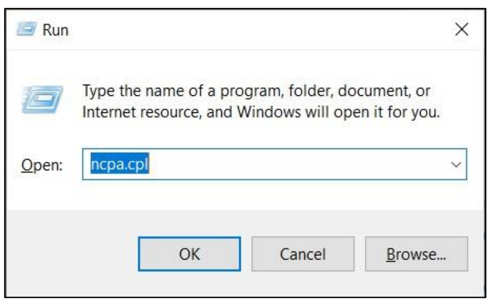

Figure D.1. Run the ncpa.cpl to Open the Network Connections Control Panel

2. Click Ethernet.

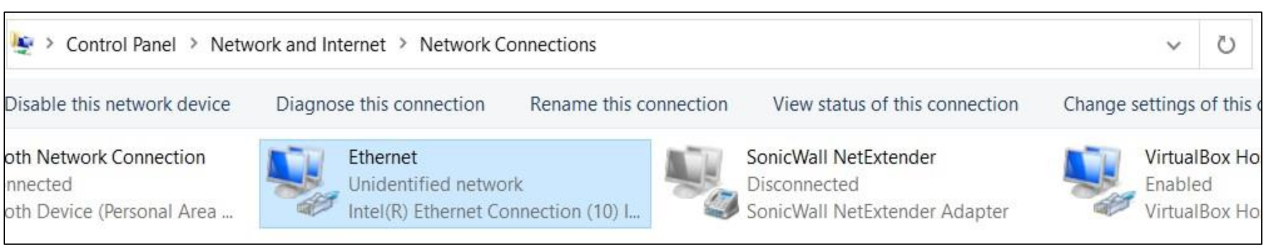

Figure D.2. Ethernet Settings

3. Right-click and go to **Properties**.

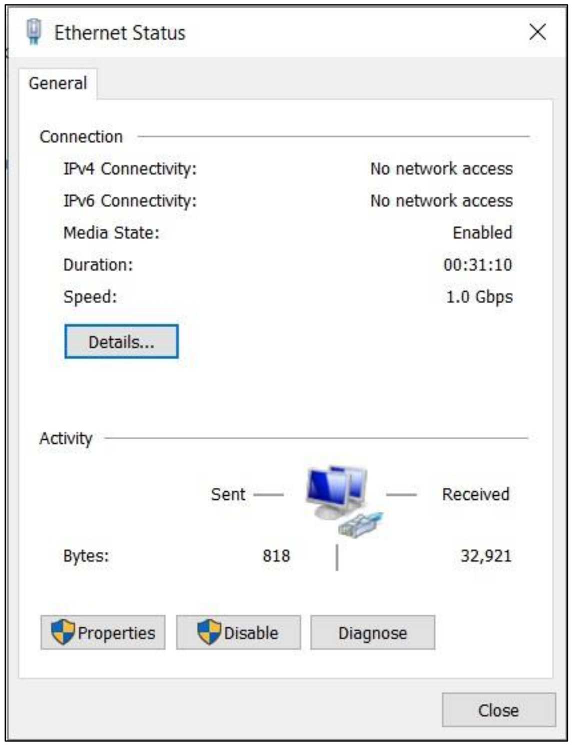

Figure D.3. Ethernet Properties

4. Double-click on Internet Protocol Version.

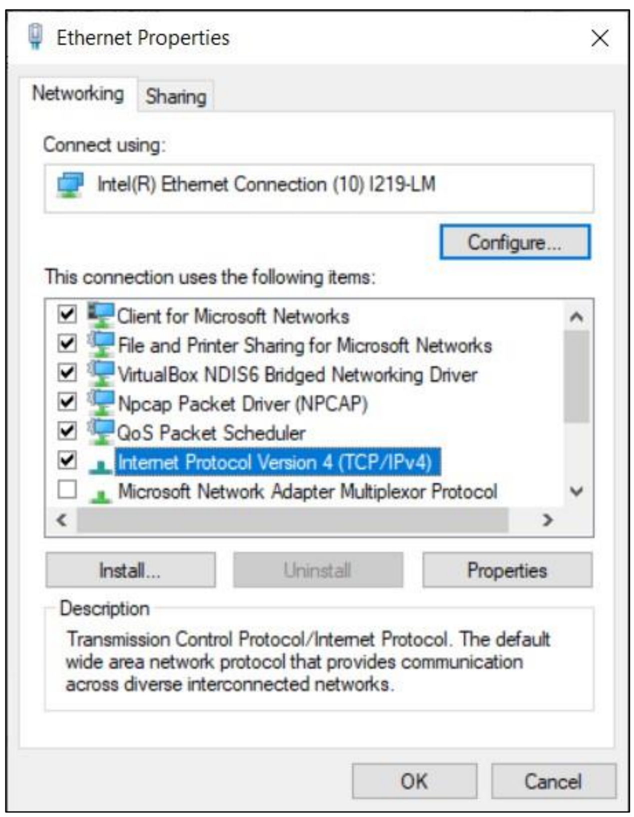

Figure D.4. Double Click on Internet Protocol Version

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

   Note: If you changed the IP address of the PC using the Propel Builder, make sure to enter the changed IP address in TCP/IPv4 settings. Refer to section 2.6 of the *Automate Stack 4.0 Reference Design (FPGA-RD-02302)* document for more information on how to change the IP addresses of the PC.

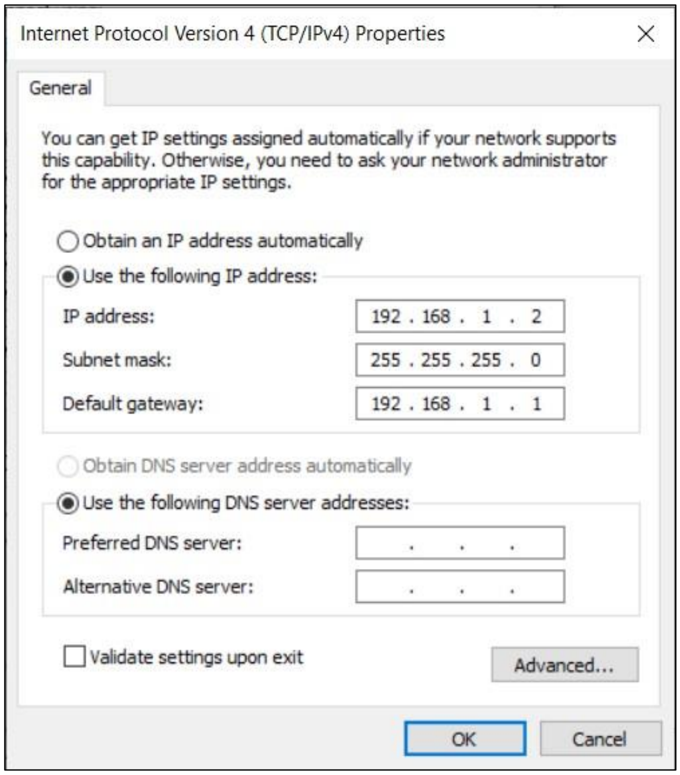

Figure D.5. Configure the Settings

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

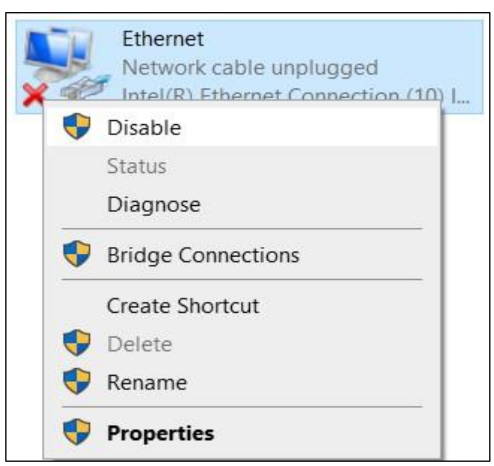

Figure D.6. Disable the Setting

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

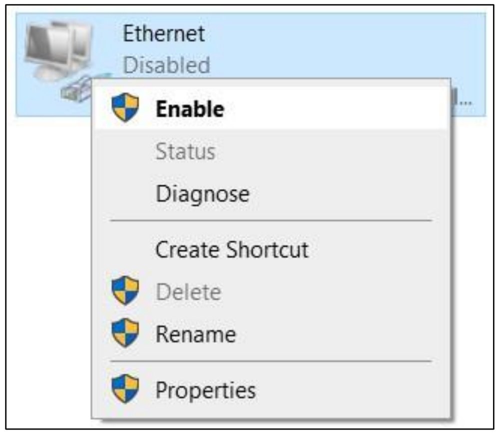

Figure D.7. Enable the Setting

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

D.3. Checking the Connection through Ping

To check the connection through 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 was successful. Otherwise, follow the steps in the D1. Connecting Between the Main System and the Laptop/PC and D.2. Setting the Ethernet Configuration sections.

```
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 D.8. Ping on cmd Terminal

D.4. UDP Packet Transaction

Note: Download the wireshark tool in 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 the packets, and you 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.

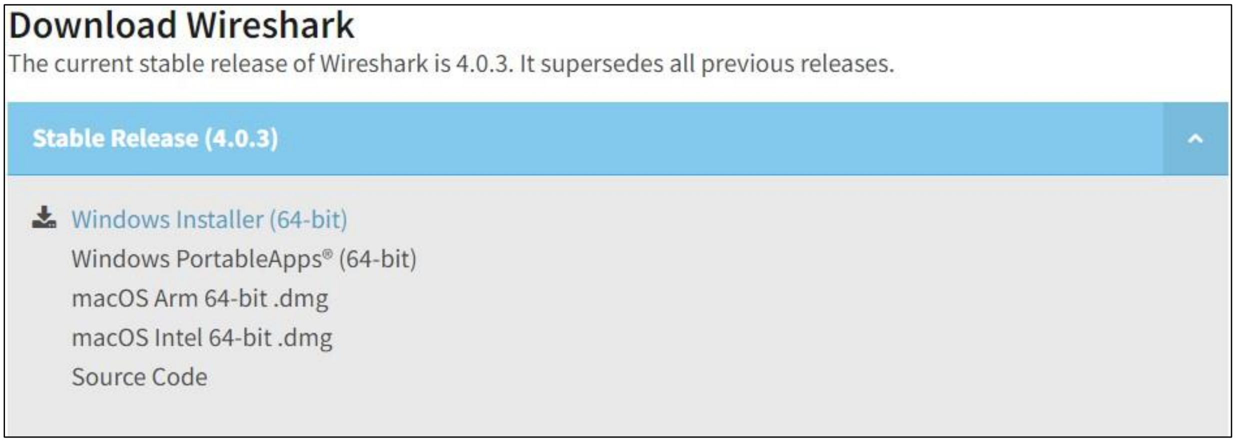

Figure D.9. Downloadable Link of Wireshark

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

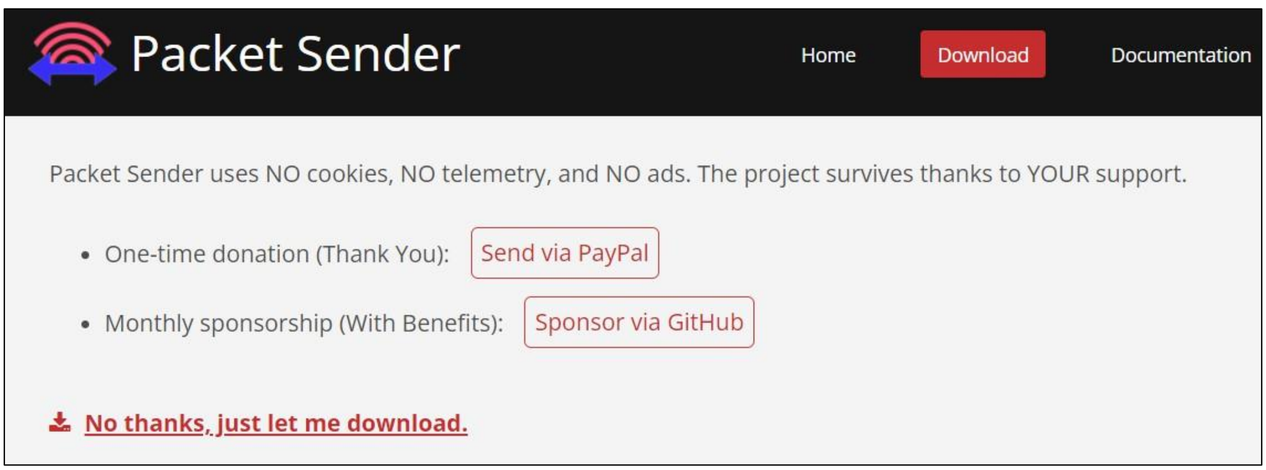

Figure D.10. Downloadable Link of Packet Sender

To configure the Docklight, perform the following:

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

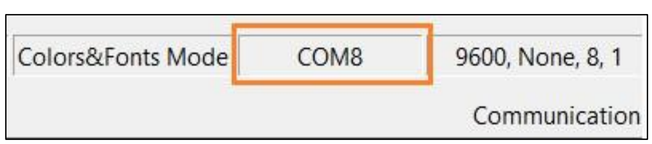

Figure D.11. Docklight – Com Selection

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

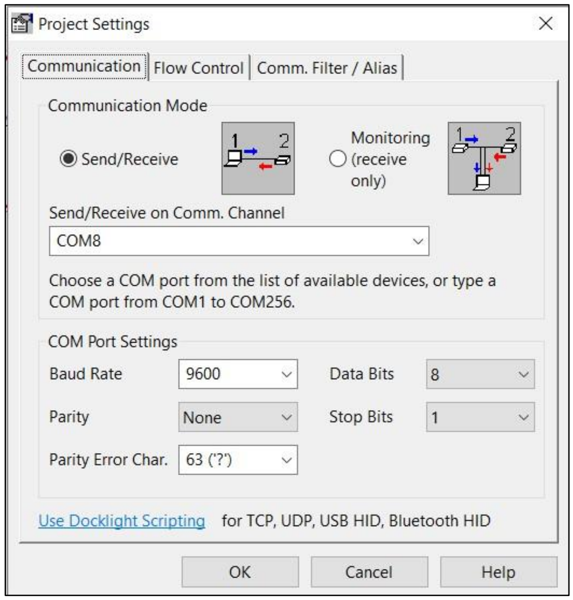

Figure D.12. Docklight – Com and Baud Rate Selection

6. Click **Run**.

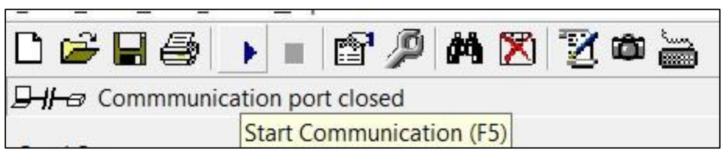

Figure D.13. Docklight – Run (Start Communication)

7. Open the Wireshark tool and select **Ethernet**.

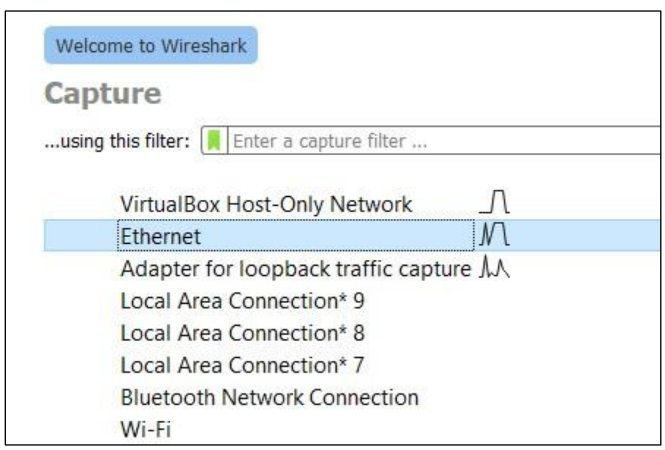

Figure D.14. Wireshark Tool – Ethernet Selection

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

Figure D.15. Packet Sender Tool – Send 64 Bytes of ASCII Characters

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

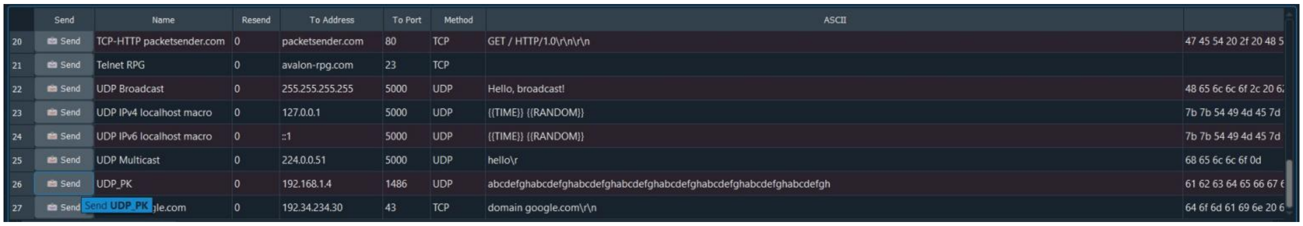

Figure D.16. Packet Sender Tool – Send UDP Packet

12. Check the logs.

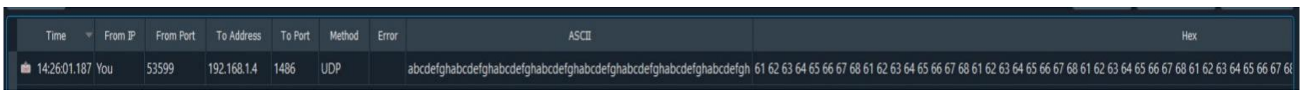

Figure D.17. Packet Sender Tool – Transaction Log

13. Open Docklight to check the data received.
14. Check if the UDP data is received.

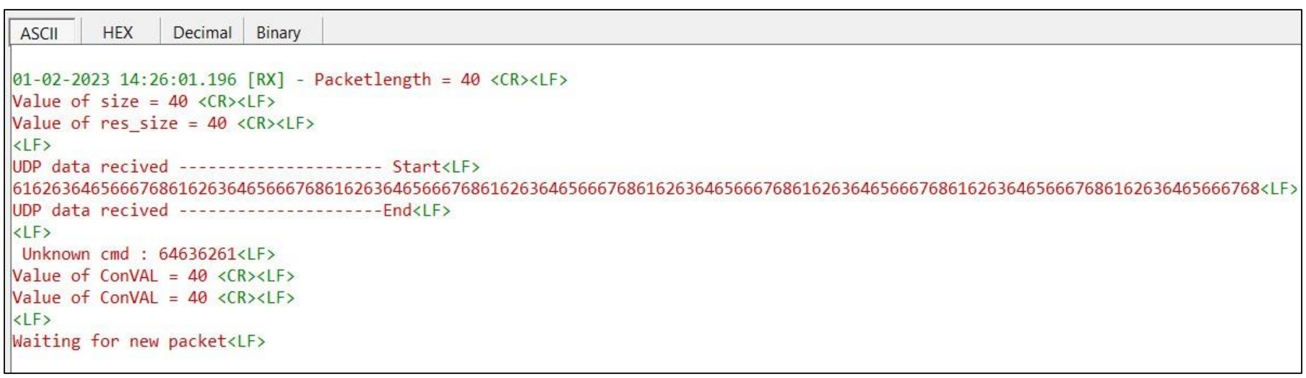

Figure D.18. Docklight – ASCII Logs

15. Go to the Wireshark tool.
16. Type **udp.port == 1486** on top of the bar as shown in Figure D.19.

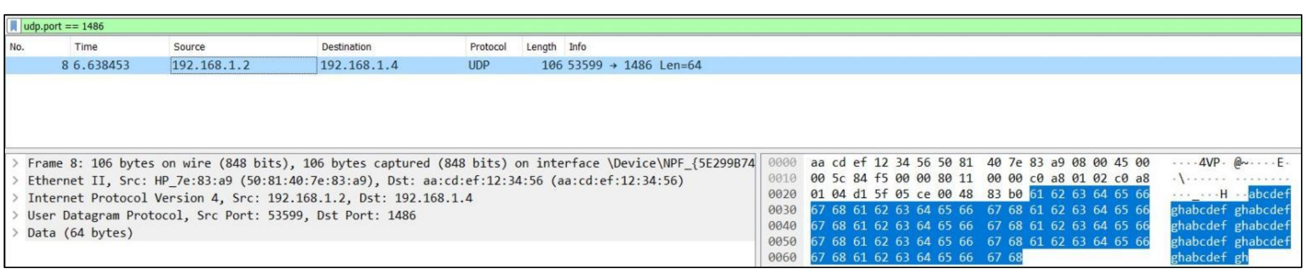

Figure D.19. Wireshark Tool – Write udp.port == 1486

a. Check both the Source and Destination UDP packets.

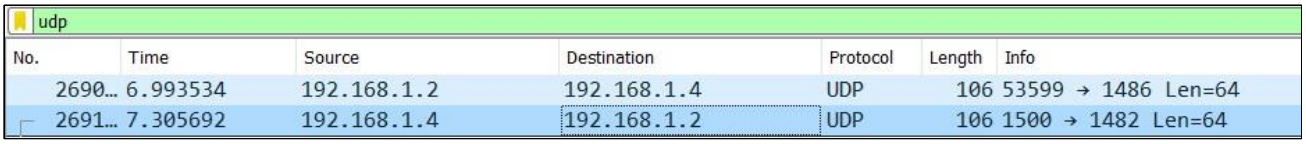

Figure D.20. Source and Destination UDP Packet

b. Click on the first UDP transaction to check in detail.

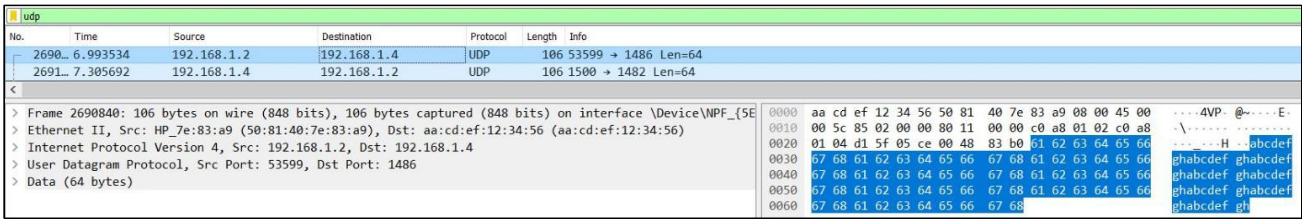

Figure D.21. Wireshark Tool – First UDP Packet

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

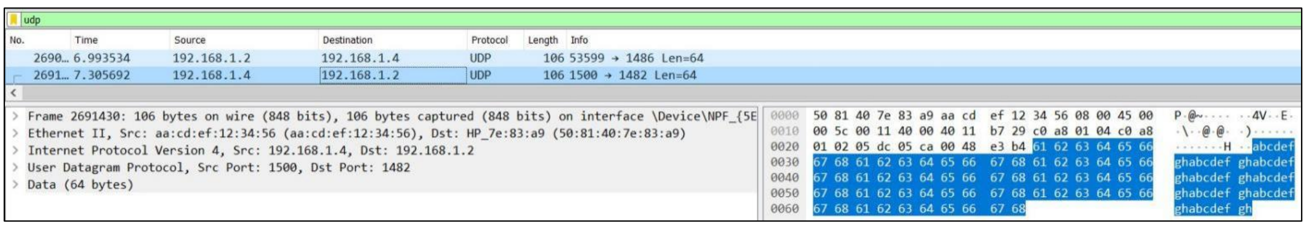

Figure D.22. Wireshark Tool – Second UDP Packet

D.5. Packet Sender Commands

D.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

D.5.2. Recommending 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 E. Troubleshooting

E.1. Troubleshooting the Automate 4.0 User Interface

E.1.1. Motor Configuration Values are 0

Problem Description: Motor configuration values are all 0.

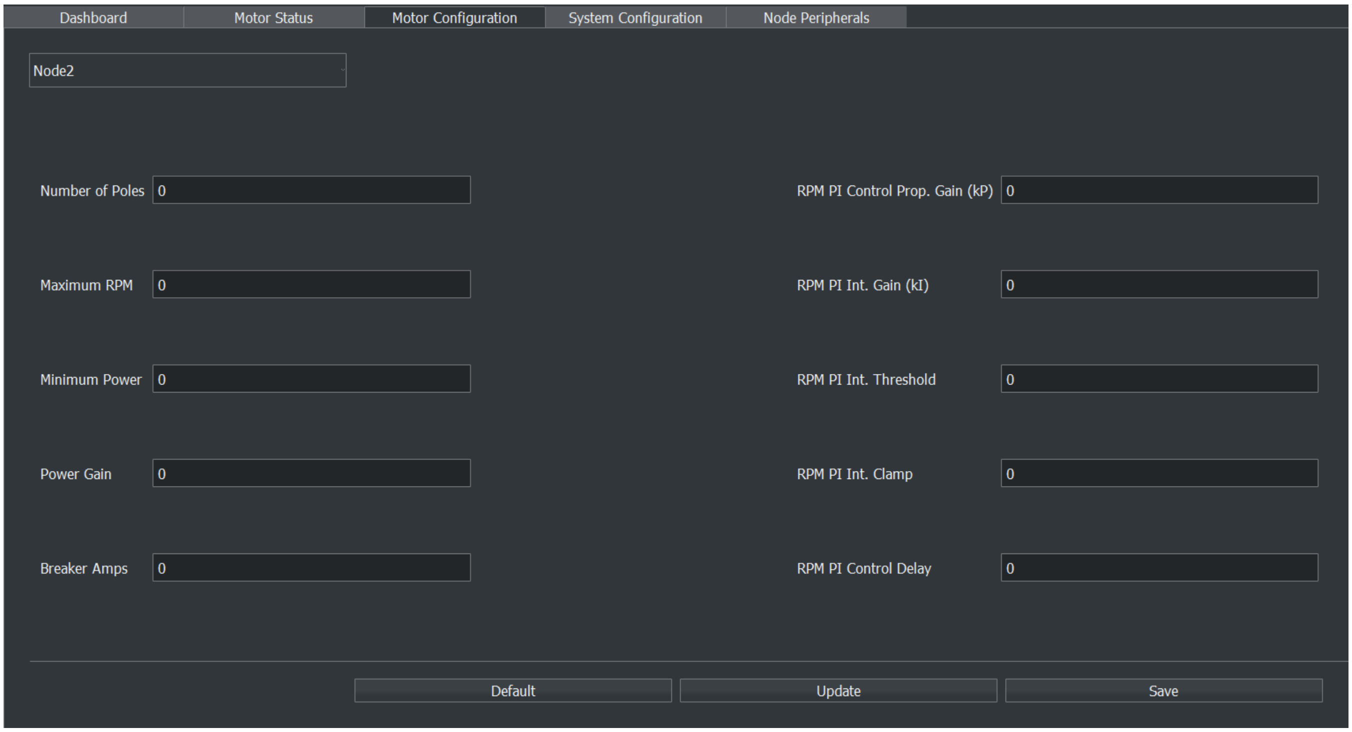

Figure E.1. Motor Configuration Values are All 0

Solution:

1. In the Automate 4.0 user interface, click the **Motor Configuration** tab.
2. From the Select drop-down menu, either **Select All** or select an individual node.

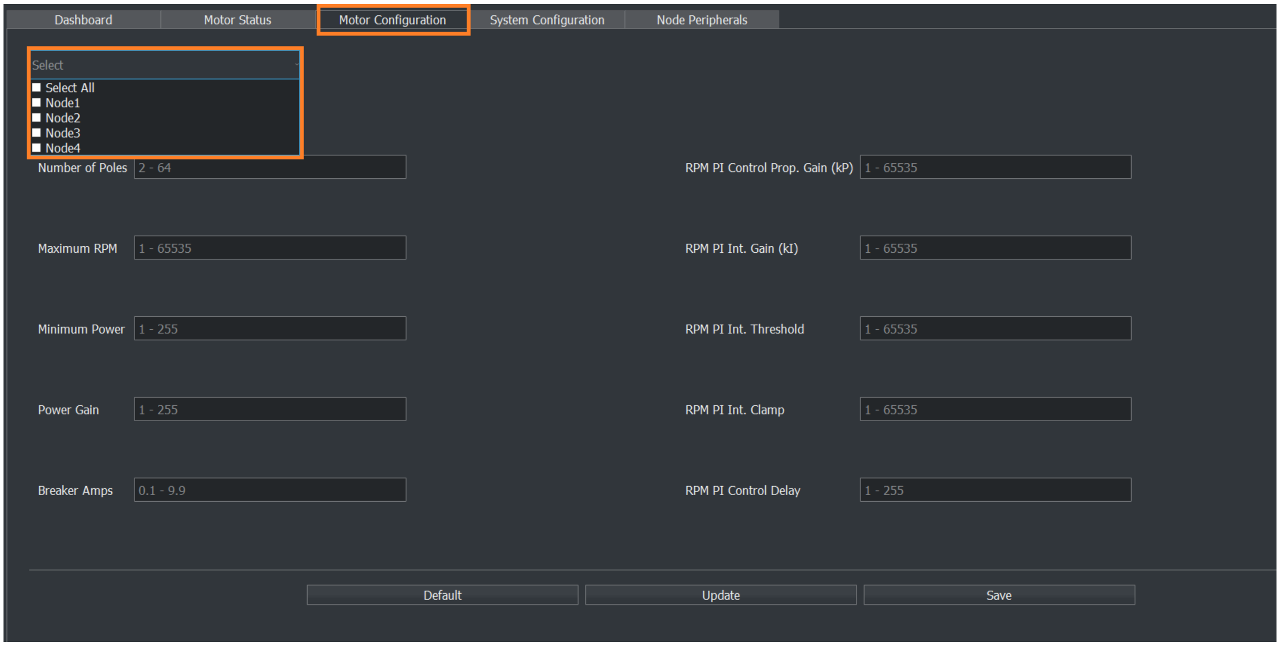

Figure E.2. Motor Configuration: Node Selection

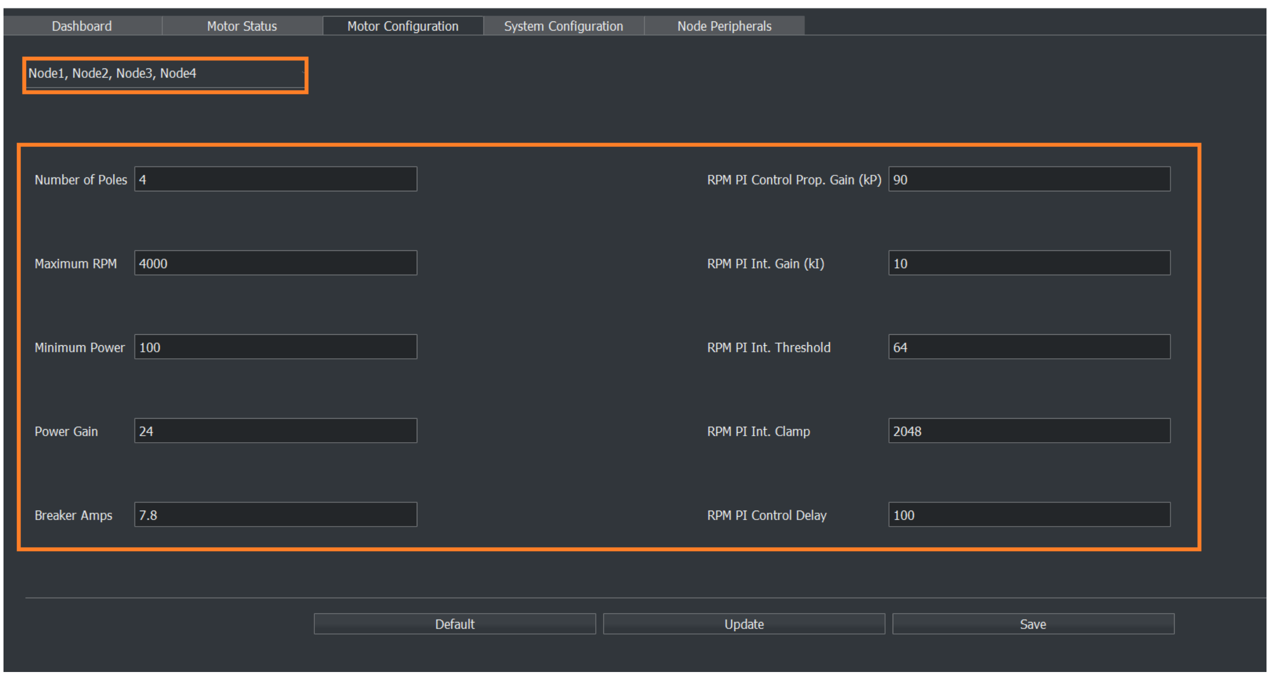

Figure E.3. Motor Configuration: All Nodes Selected

a. Change the Motor Configuration fields to the following default values:
   - RPM PI Control Prop. Gain(kP): 150
   - RPM PI Int. Gain(kI): 10
   - RPM PI Control delay: 100
   - Minimum Power: 100
   - Power gain: 24
   - Breaker Amps: 7.8

Note: These values are for the BLY171D 24 V, 4000 RPM motors only. Select the appropriate values for the motors you are using.

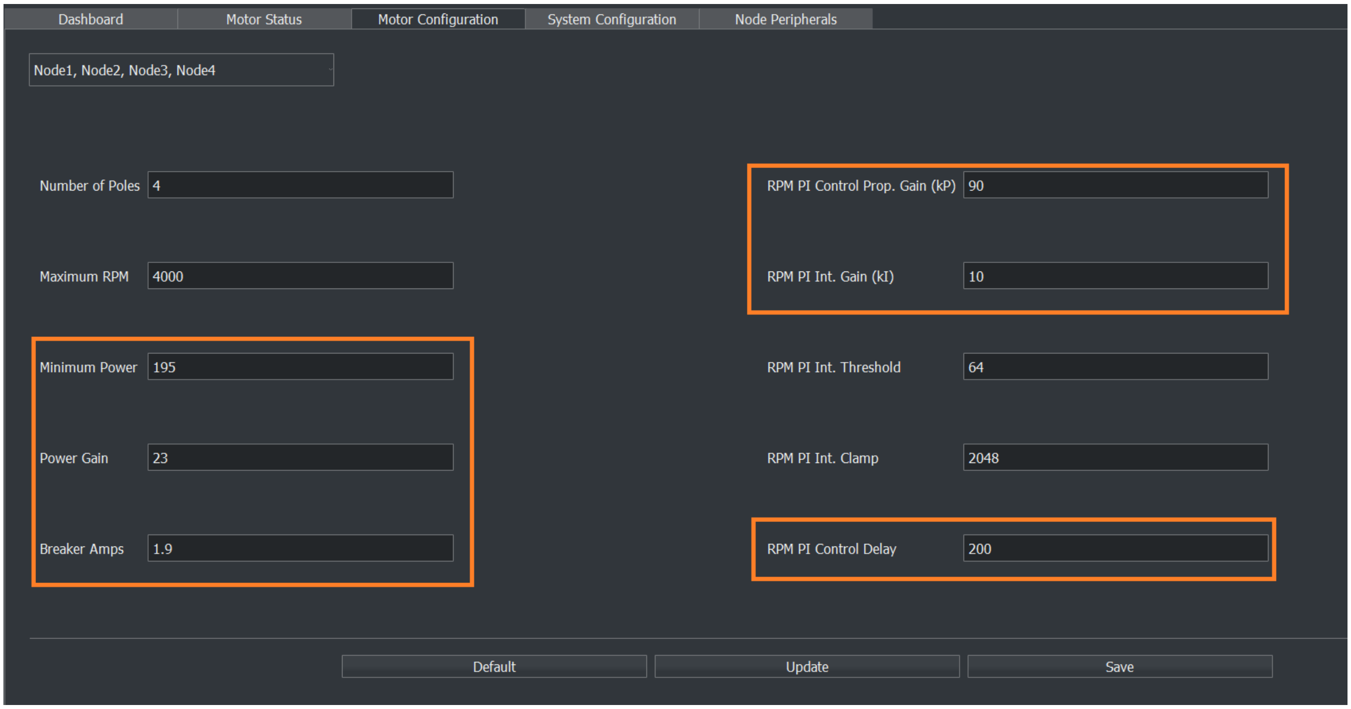

Figure E.4. Motor Configuration Value Changes

3. Click **Default** and **Update**.
4. The confirmation message window appears. Click **Yes**.

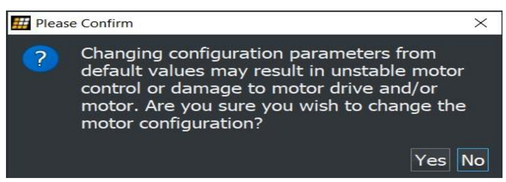

Figure E.5. Application Software – Motor Configuration: Warning Message

5. The authentication window appears. Enter the credentials and click **Login**.
   - **Username: lattice**
   - **Password: lattice**

Figure E.6. Application Software – Motor Configuration: Authentication Window

6. The *Successfully updated the configuration of the selected node* message appears. Click **OK.**

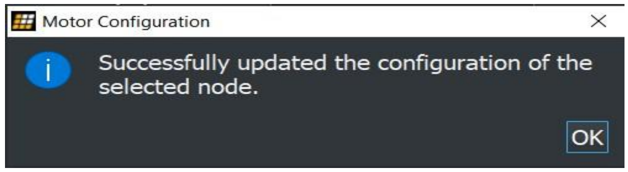

Figure E.7: Application Software – Motor Configuration: Update Configuration

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

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

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

Solution: Power everything off and restart the setup.

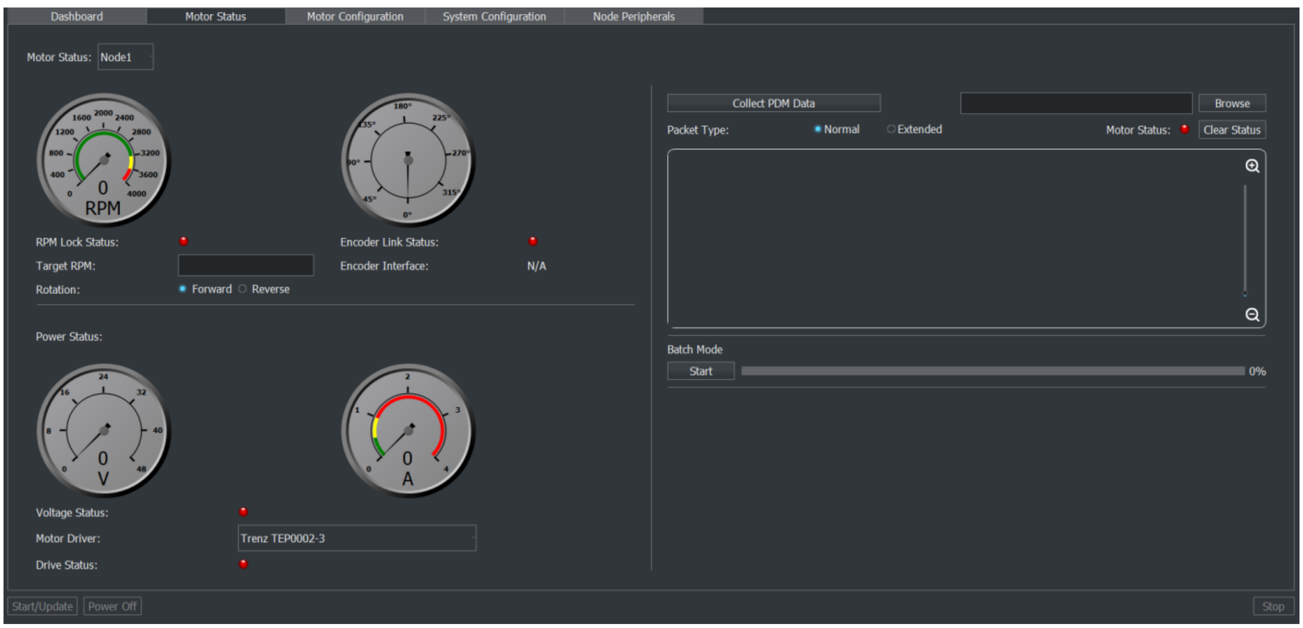

Figure E.8. Application Software – Motor Status Tab: Red Status

Note: For the Encoder Link Status issue, ensure the actual encoder is connected and following the node system initialization sequence as described in the Hardware System Readiness section.

E.1.3. One of the Selected Motor is Running

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

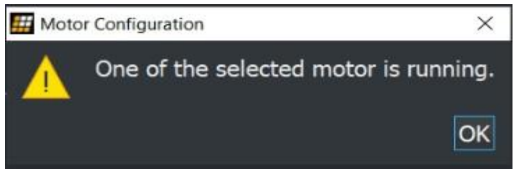

Figure E.9. Motor Running Warning Message

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

E.1.4. Response Not Received from the Board Warning

Problem Description: The *Did not receive response from the board appears* message while connecting to the user interface.

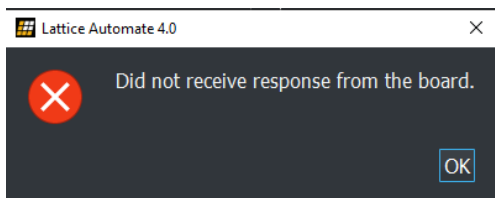

Figure E.10. Motor Connection Message

Solution:

1. Check the Ethernet connection. Follow the steps in Appendix D. Connecting to the Ethernet to ping the main board from the PC.
2. Make sure the Main System Ready and Node System Ready LEDs are illuminated, as described in the Hardware System Readiness section.
3. Use the Propel terminal to check the print statements from the main board:
4. Connect the UART cable between the Main Board and a PC with Propel terminal.
    a. Open the Propel terminal.
    b. Double-click the terminal button at the bottom part of Propel SDK window as shown in Figure E.11. The Propel terminal appears as shown in Figure E.12.

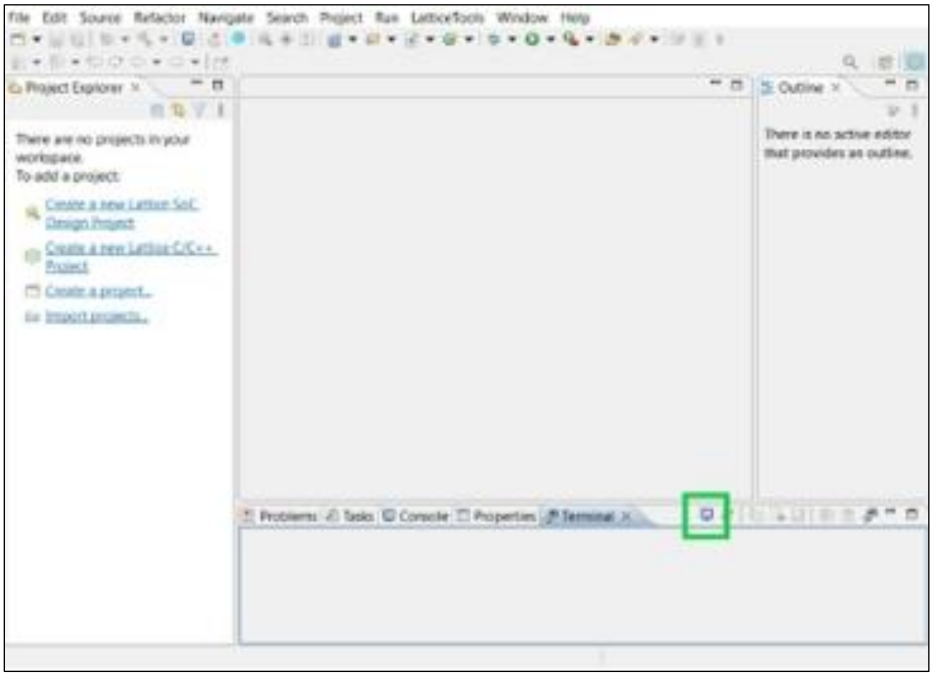

Figure E.11. Open Propel Terminal

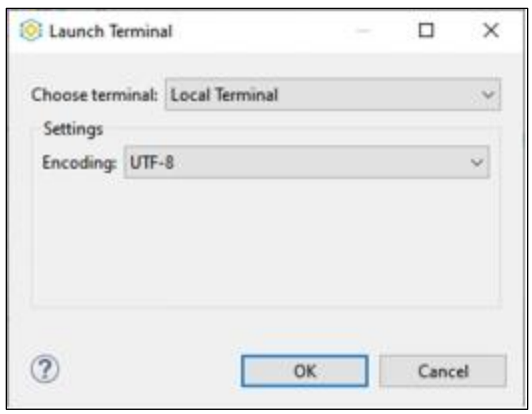

Figure E.12. Propel Terminal Selection Window

c. Choose terminal: Serial terminal as shown in Figure E.13.

d. Select the right serial port, drop down the serial port, and select the last COM.

e. Select **115200** for the Baur Rate. Click **OK**.

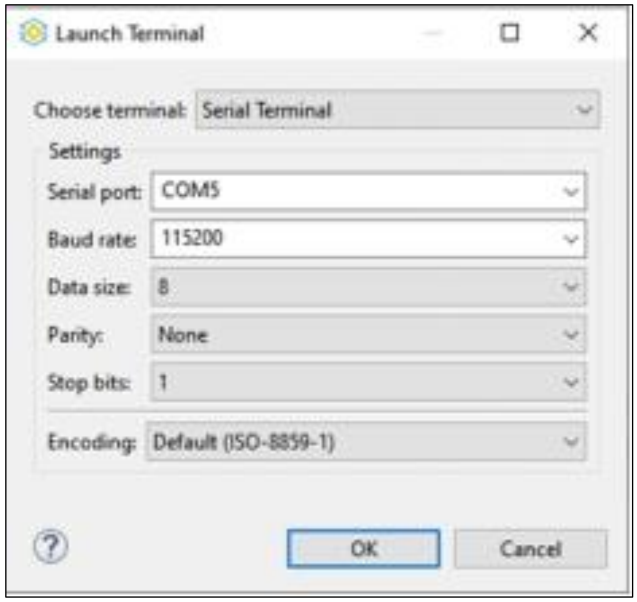

Figure E.13. Propel terminal – Com and Baud Rate Selection

f. If countering waits for Ethernet packet

g. If you are not receiving any packets, press the **SW3** reset button on the board or apply the settings below:

   i. Press the **Windows + R** button simultaneously.

   ii. Type the *ncpa.cpl* command and **Enter**.

   iii. Right-click on Ethernet and click **Disable**.

   iv. Right-click and click Enable.

h. Power cycle the Main Board or press SW3 button on the Main Board.

i. In Docklight, select the ASCII tab. Wait 60 seconds and check that the print statements are valid, as shown in Figure E.14.

```
U-Boot SPL 2024.01-00035-g3ec17456c6-dirty (Sep 25 2024 - 14:36:14 +0800)
Trying to boot from MTD


U-Boot 2024.01-00035-g3ec17456c6-dirty (Nov 20 2024 - 14:37:09 +0800)

CPU:   rv32imc
Model: lattice,riscv
DRAM:  1 GiB
Core:  11 devices, 8 uclasses, devicetree: separate
Flash: 128 MiB
Loading Environment from nowhere... OK
In:    serial@40090000
Out:   serial@40090000
Err:   serial@40090000
Net:   No ethernet found.
Hit any key to stop autoboot:  0
Reading 393216 byte(s) at offset 0x028a0000
## Checking CRC
CRC pass: 0x22ce
## Starting application at 0x80000000 ...
Main System Ready!
Automate Stack 4.0 ver 0.0.1!
Version Date 11/11/2024
System Init!
Node num 16, Node Data length 64
Before: Link Detected: 1
Polling Phy Link status.....................
After: Link Detected: 31
Chain 1 Link for Main System detected
Node num 16, Node Data length 64
Broadcast END
Num node default: 16
 Org:detected_nodes 4
 Chain : 1 ACTIVE NODE: 4
 Org: 4
Chain : 1 FINAL ACTIVE NODE 4
node number 4
PHY LINK EXPECTED: 1f
BEFORE: TIMER PHY LINKS MAIN and NODES 1f
Node Num: 4
Before Node Enable: 1
After Node Enable: f
Chain 1 nodes reg : f
INIT_DONE
Sys Init
In PU Seq
PU Seq Done
StartModelAllocation.
                     StartModelAllocation - Done.
                                                 operators_size 13
                                                                  operators_size 13
                                                                                   Registering op code index 0
builtin_code '3'Registering op code index 1
builtin_code '17'Registering op code index 2
builtin_code '3'Registering op code index 3
builtin_code '3'Registering op code index 4
builtin_code '17'Registering op code index 5
```

Figure E.14. Main system Print Statements

j. LED D64 on the Main Board must be illuminated after the Docklight prints appeared.

k. 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 4.0 user interface is set to connect to IP address **192.168.1.4**.

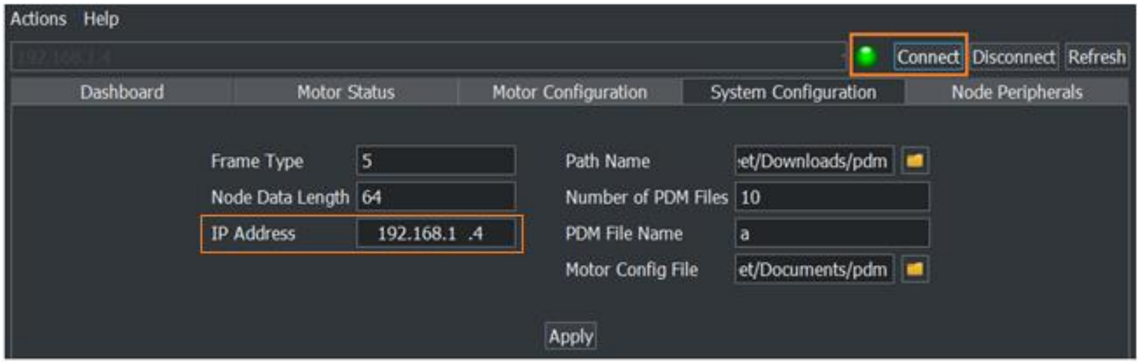

Figure E.15. Automate 4.0 User Interface

b. If you changed the main system IP address in the Propel Builder, make sure you are using the changed IP address. Refer to section 2.6 of the *Automate Stack 4.0 Reference Design (FPGA-RD-02302)* document for more information on how to change the IP address of the main system.

E.1.5. Allowing Permission to the Publisher and Subscriber

Problem Description: If you forgot or skipped the *Allow the permission* process while installing the Automate 4.0 user interface application and this message is shown on the user interface *Did not receive response from the board (Not able to load)*.

Solution 1:

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

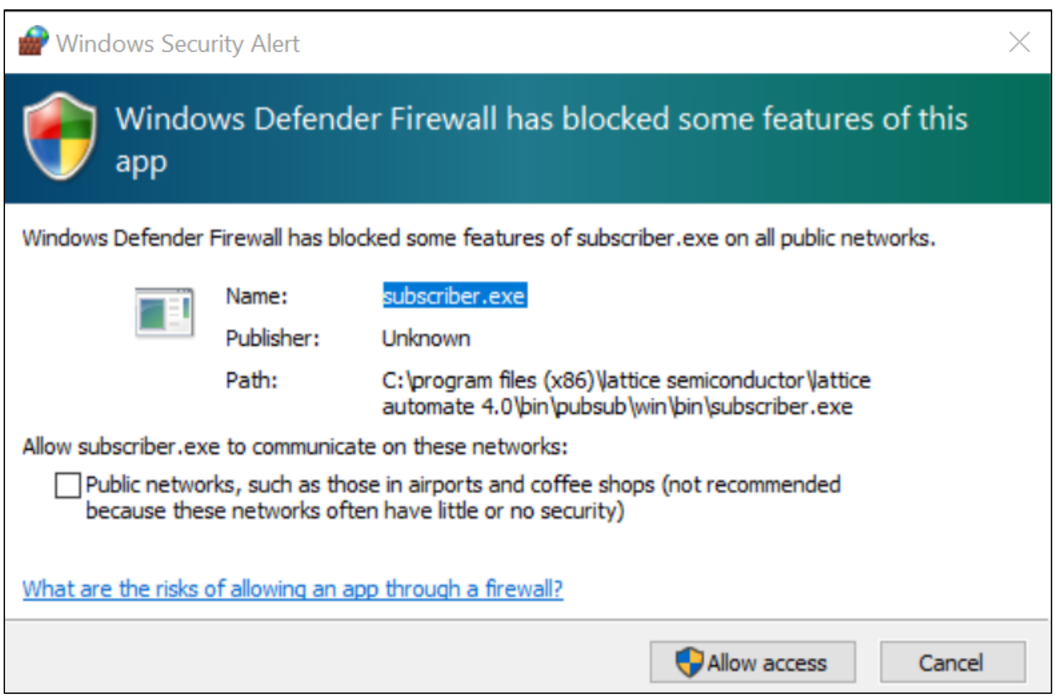

Figure E.16. Allow Permission: Subscriber

5. Click **Allow Access** to allow permission to publisher.

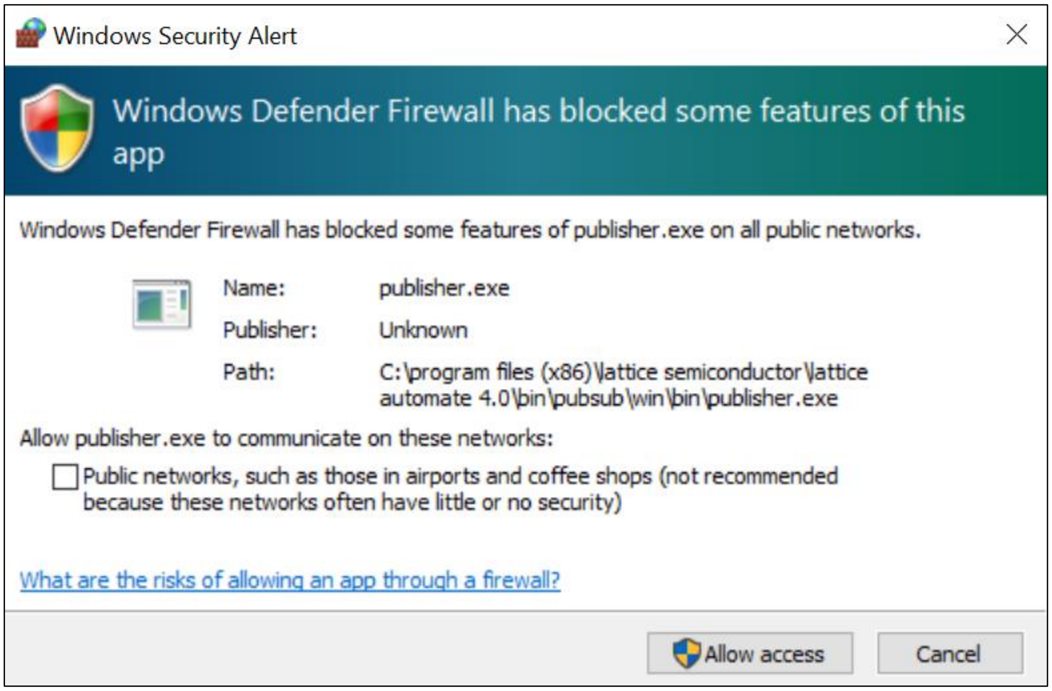

Figure E.17. Allow Permission: Publisher

Another way is to change the directory path or create new folder inside the C directory: **C:\Program Files (x86)\Lattice Semiconductor\Lattice Automate 4.0**.

Solution 2:

1. Open the firewall and network protection from the search. Click **Open**.

Figure E.18. Firewall and Network Protection

2. Go to **Allow an app through firewall**.

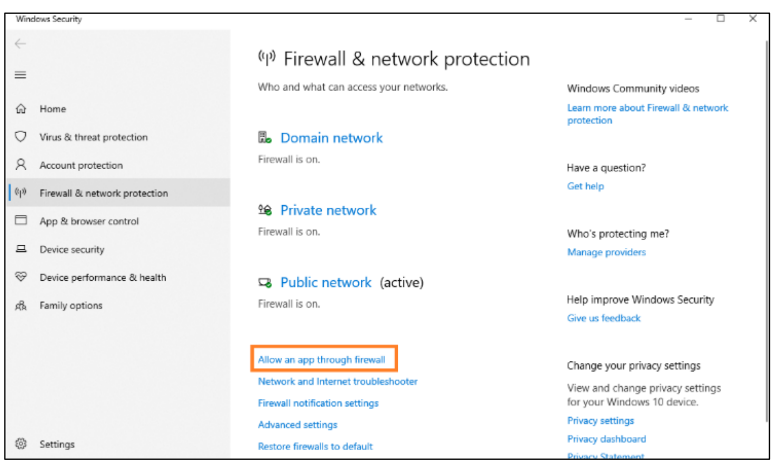

Figure E.19. Allow an App through Firewall

3. Scroll down and find publisher to check the correct path where the installer is installed.
4. Click **Change setting** to check the correct path.

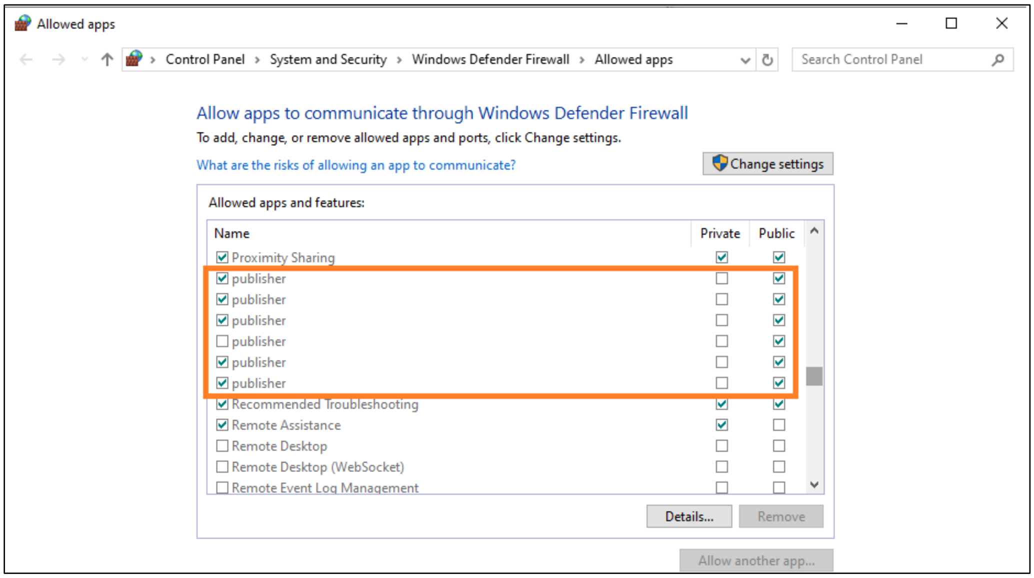

Figure E.20. Scroll Down and Find Publisher

5. Double-click on publisher.
6. The Publisher window opens in edit mode. You can verify it and allow the correct path. Otherwise, no need to allow if it is showing a different path.
7. Check one by one all publishers for the correct path.
8. Scroll down and find subscriber to check the correct path where the installer is installed.
9. Click **Change setting** to check the correct path.

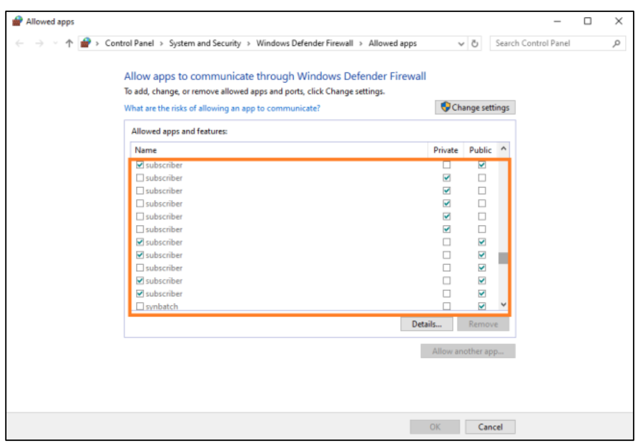

Figure E.211. Scroll Down and Find Subscriber

10. Double-click on subscriber.
11. The Subscriber tab opens in edit mode. You can verify it and allow the correct path. Otherwise, no need to allow if it is showing a different path.

References

- 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, please refer to the Lattice Answer Database at www.latticesemi.com/Support/AnswerDatabase

Revision History

Revision 1.0, January 2025

Section	Change Summary
All	Production release.

www.latticesemi.com