Automate Stack 3.0 Demo

User Guide

FPGA-UG-02181-1.1

June 2023

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 associated risk the responsibility entirely of the Buyer. Buyer shall not rely on any data and performance specifications or parameters provided herein. 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. No Lattice products should be used in conjunction with mission- or safety-critical or any other application in which the failure of Lattice's product could create a situation where personal injury, death, severe property or environmental damage may occur. The information provided in this document is proprietary to Lattice Semiconductor, and Lattice reserves the right to make any changes to the information in this document or to any products at any time without notice.

Contents

Figures

Acronyms in This Document

A list of acronyms used in this document.

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

1. Introduction

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

There are two FPGA systems in the Automate Stack: the main system and the node system.

The node system controls a motor and collects the data from the motor for the predictive maintenance neural network to use. The node system can connect to external sensors through three different protocols (SPI, UART/Modbus, and I2C), and has two-way communication with the main system through EtherConnect.

The main system acts as a central hub which passes commands up to 8 different node systems. The main system also contains the AI/ML powered predictive maintenance system, which uses a neural network to analyze the collected data and predict failure ahead of time. In the Automate demo, this collected data is the current signature of the motor, but the PDM IP block can use any quantitative data for its input if the neural network is trained on the same type of data.

The main system has a RISC-V CPU running FreeRTOS, a real time operating system often used in embedded applications. All modules in the main system and the node system FPGA designs are connected to each other using an AXI4 bus interface. This interface has high throughput and can run at a frequency of up to 100 MHz. It allows parallel data transfer, meaning a read and a write between two modules on the AXI bus can happen simultaneously.

The user interacts with the main system through a GUI running on a PC. The GUI and main system are connected with an Ethernet cable and communicate with each other using OPCUA, a machine-to-machine pub-sub communication protocol frequently used in the IIOT (Industrial Internet of Things).

2. Demo System Setup

The Automate demo uses one main system (running on a CertusPro-NX Versa board) and at least one node system (running on a Certus-NX Versa board). It can support up to 16 node systems. It is recommended only to run up to 8 node systems for best performance. Each node system is connected to a motor driver board, which is connected to a brushless DC motor. The main system holds the predictive maintenance CNN co-processor.

The main system connects to the Host PC. The Host PC communicates via OPCUA with a PC running a GUI (Lattice Automate 3.0), through which the user interacts with the motors and receives PDM data.

2.1. System Level Block Diagram

Figure 2.1 shows the system-level block diagram. The Host PC running the Automate 3.0 GUI is connected to the main system over a standard 1G Ethernet cable. The main system running on CertusPro-NX is connected to the node system over EtherConnect.

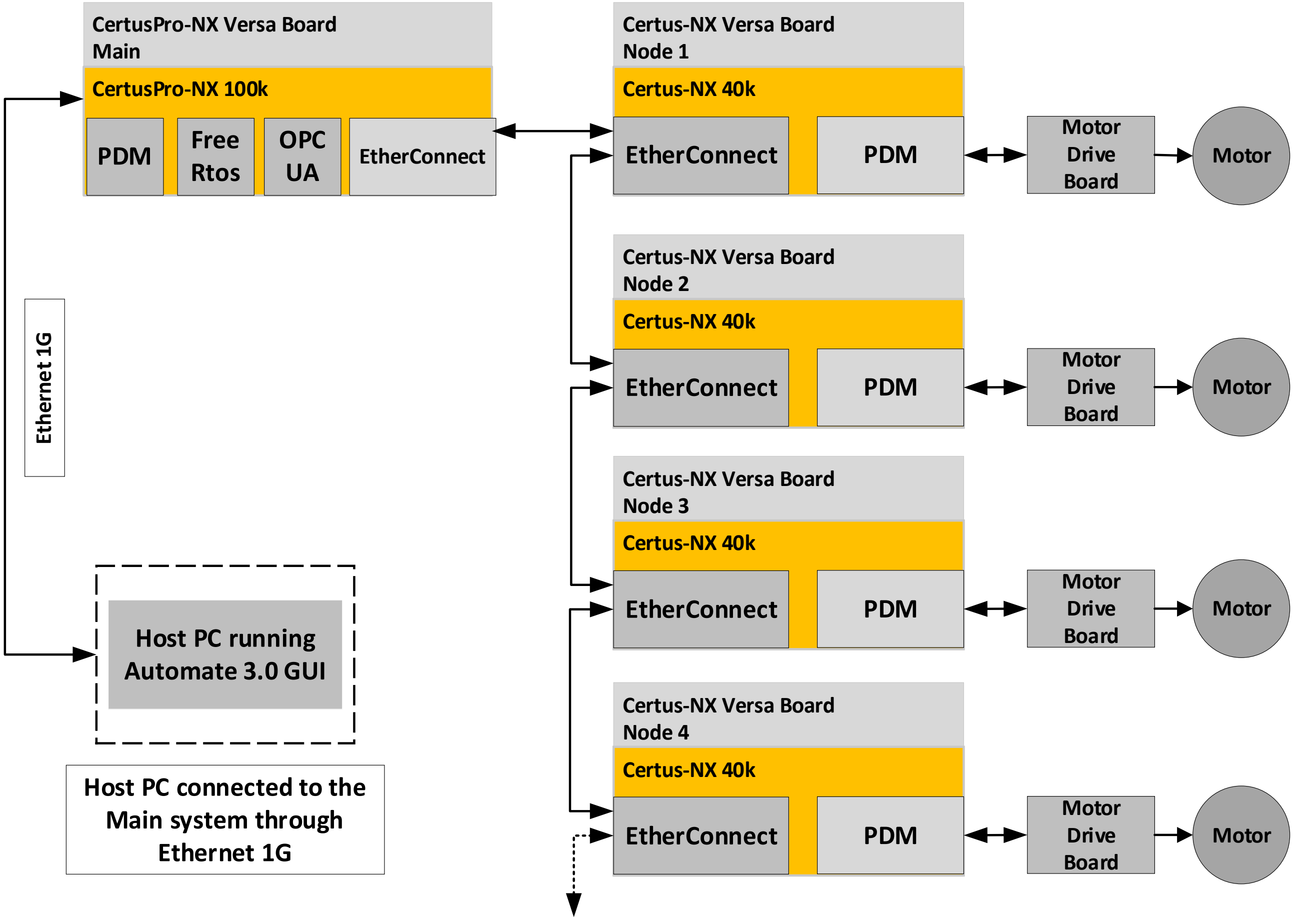


Figure 2.1. System Block Diagram

3. Communication Protocols (OPCUA)

Automate 3.0 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 scaled to be appropriate for many more use cases, including building automation and cloud applications.

OPCUA has two variations, client/server and publish/subscribe. Automate 3.0 uses the publish/subscribe version of OPCUA.

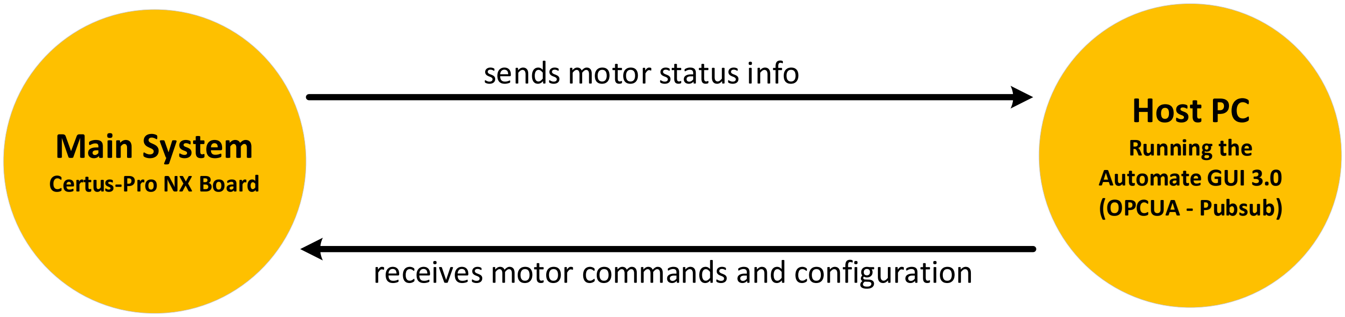


Figure 3.1. OPCUA High-Level Block Diagram

3.2. UDP Communication

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

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

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

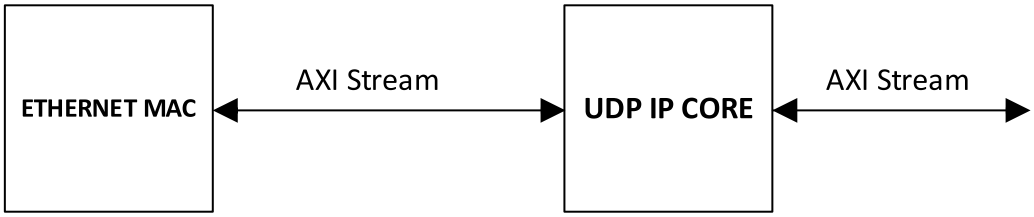


Figure 3.2. UDP IP Core Overview

4. Hardware

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

4.1. Hardware Requirements

This demonstration requires the following hardware components:

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

4.2. Hardware Setup

Details of hardware connections are illustrated in Figure 4.1.

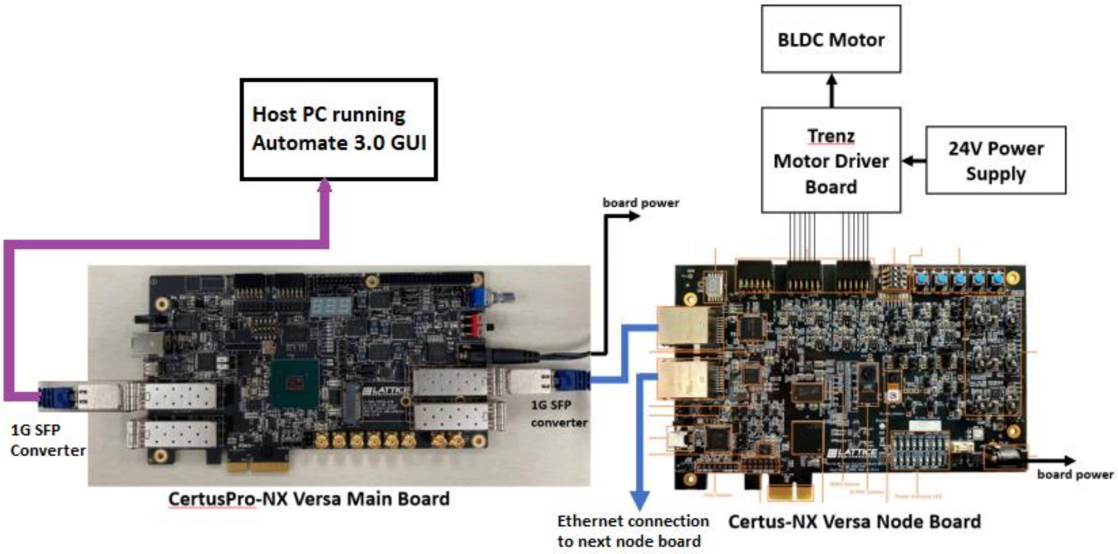

BLDC Motor
Host PC running Automate 3.0 GUI
Trenz Motor Driver Board
24V Power Supply
board power
1G SFP Converter
1G SFP converter
board power
CertusPro-NX Versa Main Board
Ethernet connection to next node board
Certus-NX Versa Node Board

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

4.3. Hardware System Readiness

1. Hardware should be connected properly as shown in Figure 4.1.
2. All boards should be programmed with. bit and .bin files.
   a. If the user using a brand new CertusPro-NX Versa board, follow the Programming the Automate Stack on Respective FLASH section for programming.
   b. Otherwise, for main and node systems, refer to the Programming a Fresh Main System Board section for board programming.
3. Power cycle every board after programming.
4. After the power cycle, reset the main system (press the SW3 button on the main system).
5. Power up sequence.
   a. Power on the Host PC.
   b. Power on the node system, motor driver, and motors.
   c. Power on the main system.
6. Check the main and node system ready LEDs to check connections:
   a. Main System:
      D67: Main System Power On
      D64: Main system ready
      D63: Ethernet connection established
   b. Node System:
      D32 and D34: Link up LEDs
      D30: Illuminates if the next node in the chain is connected. (Not needed for the last node in the chain.)
      D18 - D25: Node system ready
      7-segment LED: Node system ready

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

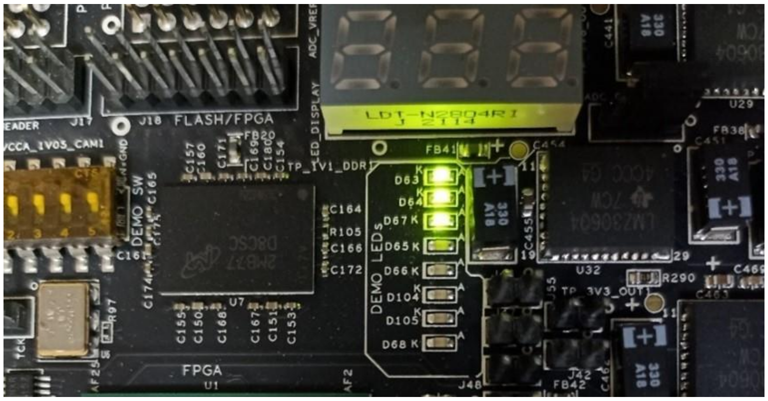


Figure 4.2. Main System Ready LEDs

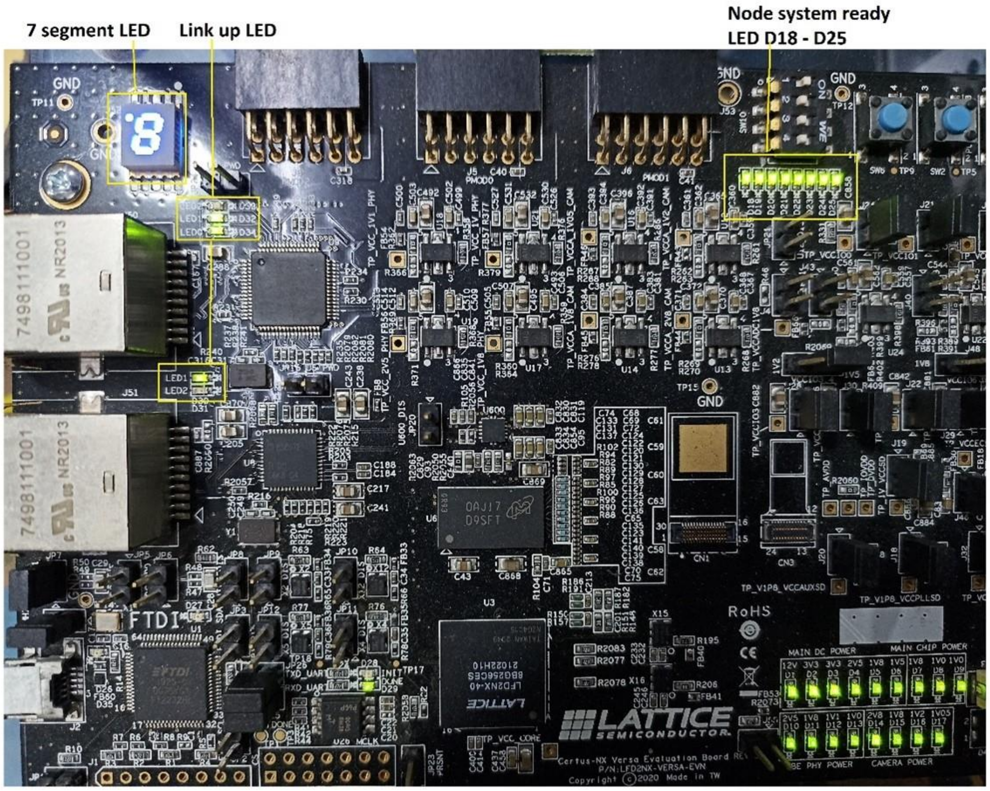


Figure 4.3. Node System Ready LEDs

5. Software Requirements

- Lattice Radiant 2022.1 or later
- Lattice Radiant Programmer 2022.1 or later (part of Radiant install)
  The following required software programs are available here:
  www.latticesemi.com/en/Products/DesignSoftwareAndIP.
  Lattice Automate program is available here:
  https://www.latticesemi.com/en/Solutions/Solutions/SolutionsDetails02/Automate
- Lattice Automate 3.0 Test Application Software
  NOTE: Lattice Automate 3.0 is available in OPCUA to publish/subscribe version.
  The software programs are available for download only if the user log in at www.latticesemi.com.

5.1. Optional Software

- Total Phase Control Center 4.1 or later for testing node peripherals (Purchase may be needed)
- Wireshark 4.0.3 or later (open source)
- Packet Sender 8.1.1 or later (open source)
- Docklight v2.4 or later (open source)

5.2. Software System Readiness

1. Establish the Ethernet connection before connecting to the GUI.
   a. Follow steps a through c in Ethernet Connection section to establish the Ethernet connection.
2. GUI installation and running
   a. Refer to GUI Application Installation (PC) section to install the GUI application on the PC (one time).
   b. Refer to the Running the Motor through Automate GUI 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 from www.latticesemi.com/Solutions/Solutions/SolutionsDetails02/Automate.

6.1. Executables

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

6.1.1. Main System

1. c_main_system_3_0.bin
2. soc_main_system_impl1.bit

6.1.2. Node System

1. c_node_system_2_0.bin
2. soc_node_impl1.bit

6.2. GUI

These files are programmed onto the Host PC.

6.2.1. GUI

Automtae3p0GUI.exe

6.3. Documentation

Automate Stack 3.0 Demo User Guide (this document)

7. Running the Motor through Automate GUI Application (PC)

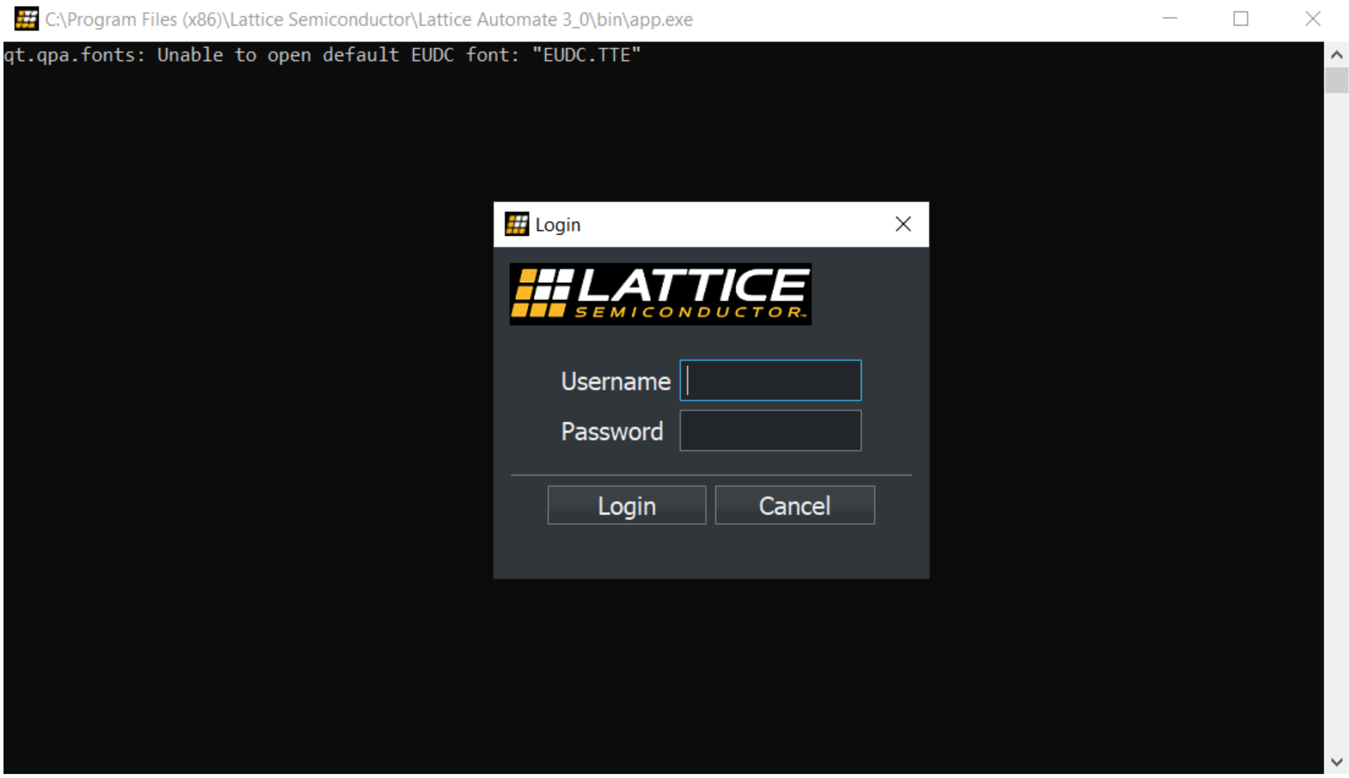
The Lattice Automate 3.0 GUI is developed for users to easily control the motor behavior from the host PC. This GUI allows the user to update motor parameters, and control the speed and directions of all the motors. The motor running procedure is defined in this section through a Graphical User Interface/Test Application.

Note: For Lattice Automate Stack 3.0 Application Installation on the PC, refer to GUI Application Installation (PC) section.

Note: For programming the main system and node system boards, refer to Programming the Automate Stack on Respective FLASH section.

7.1. Start the Application on Host PC

1. Open the Lattice Automate application.


C:\Program Files (x86)\Lattice Semiconductor\Lattice Automate 3_0\bin\app.exe
Login
LATTICE SEMICONDUCTOR
Username
Password
Login
Cancel

```
qt.qpa.fonts: Unable to open default EUDC font: "EUDC.TTE"
```

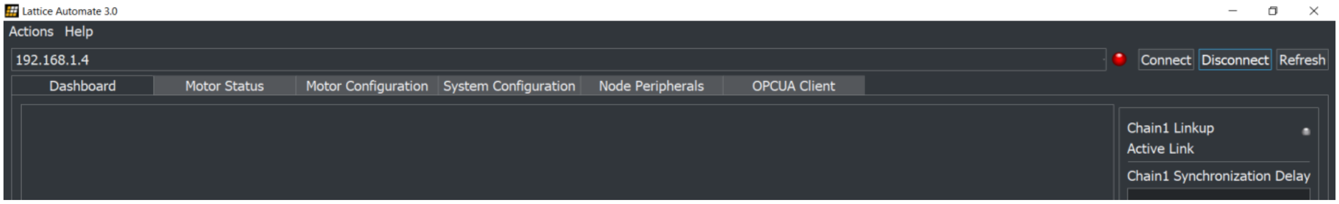
Figure 7.1. GUI Application Login Screen

2. Enter the credentials and click **Login**:

   Username: lattice

   Password: lattice

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

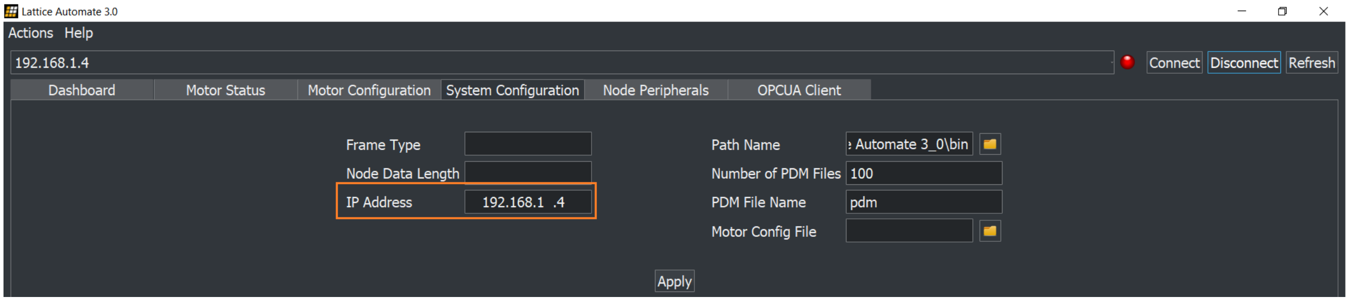


192.168.1.4

Figure 7.2. GUI Application Dashboard Tab

7.2. Connect to the Main System

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



192.168.1.4

Frame Type

Node Data Length

IP Address 192.168.1 .4

Path Name : Automate 3_0\bin

Number of PDM Files 100

PDM File Name pdm

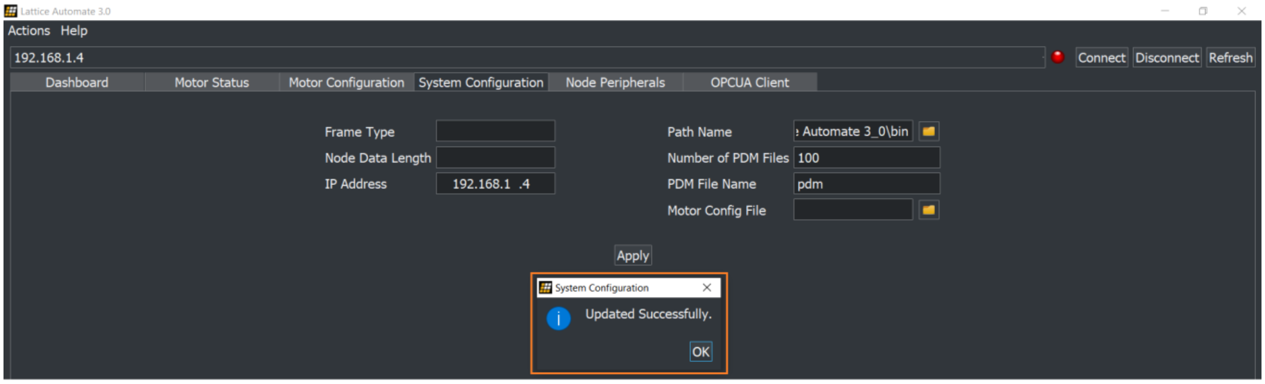
Motor Config File

Figure 7.3. GUI Application System Configuration: IP Address

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

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

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



192.168.1.4

Frame Type

Node Data Length

IP Address 192.168.1 .4

Path Name : Automate 3_0\bin

Number of PDM Files 100

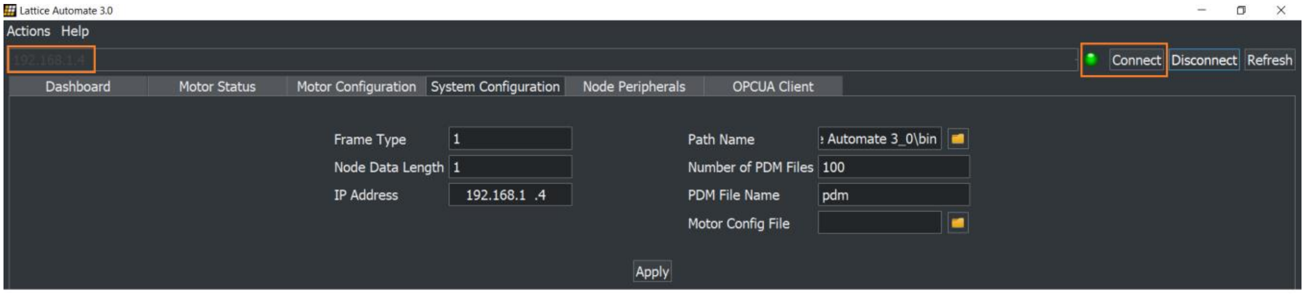
PDM File Name pdm

Motor Config File

Updated Successfully.

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

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



192.168.1.4

Frame Type 1

Node Data Length 1

IP Address 192.168.1 .4

Path Name : Automate 3_0\bin

Number of PDM Files 100

PDM File Name pdm

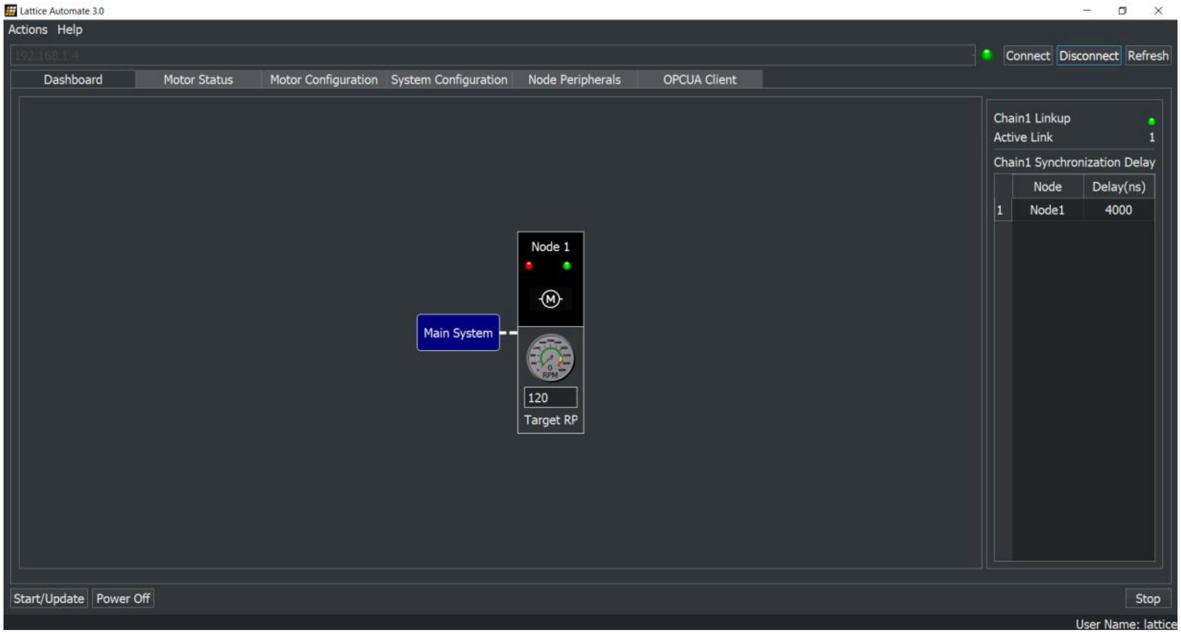
Motor Config File

Figure 7.5. GUI Application System Configuration: IP Address Configured Status

7.3. Dashboard Tab

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

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



Chain1 Linkup ●
Active Link 1

Chain1 Synchronization Delay

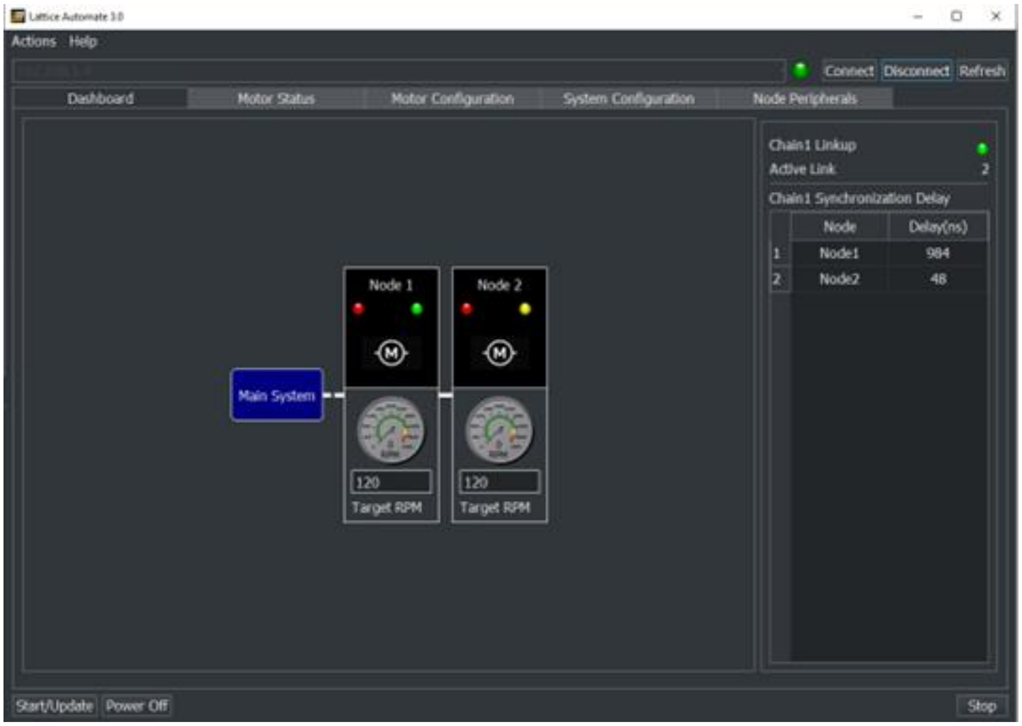
	Node	Delay(ns)
1	Node1	4000

120
Target RP

Figure 7.6. GUI Application Dashboard: List Status

7.3.1. Refresh Button

Figure 7.7 shows the Connected nodes. The Refresh button can be used to update nodes, if the user is already connected and needs to add or remove a node. To test and demonstrate its functionality, follow these steps.



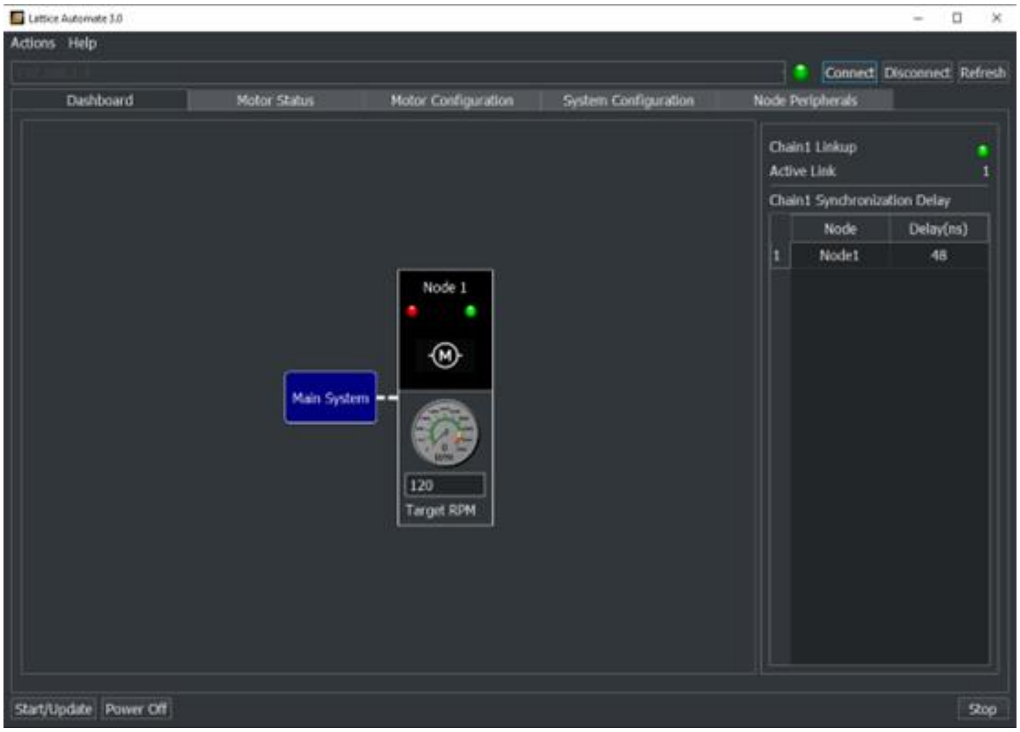
Chain1 Linkup ●
Active Link 2

Chain1 Synchronization Delay

	Node	Delay(ns)
1	Node1	984
2	Node2	48

Figure 7.7. GUI Application Dashboard: 2 Nodes Connected

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



Chain1 Linkup

Active Link 1

Chain1 Synchronization Delay

	Node	Delay(ns)
1	Node1	48

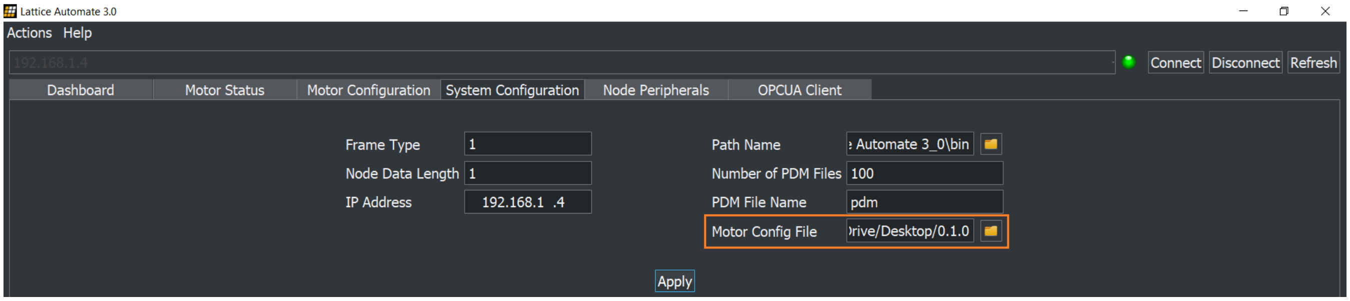
120

Target RPM

Figure 7.8. GUI Application Dashboard: 1 Node Connected after Refresh

7.4. Motor Configurations

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



Frame Type 1

Node Data Length 1

IP Address 192.168.1 .4

Path Name ꞉ Automate 3_0\bin

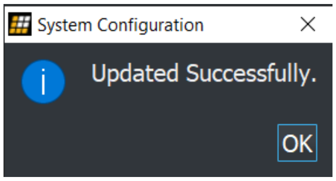
Number of PDM Files 100

PDM File Name pdm

Motor Config File ›rive/Desktop/0.1.0

Figure 7.9. GUI Application System Configuration: Motor Config File

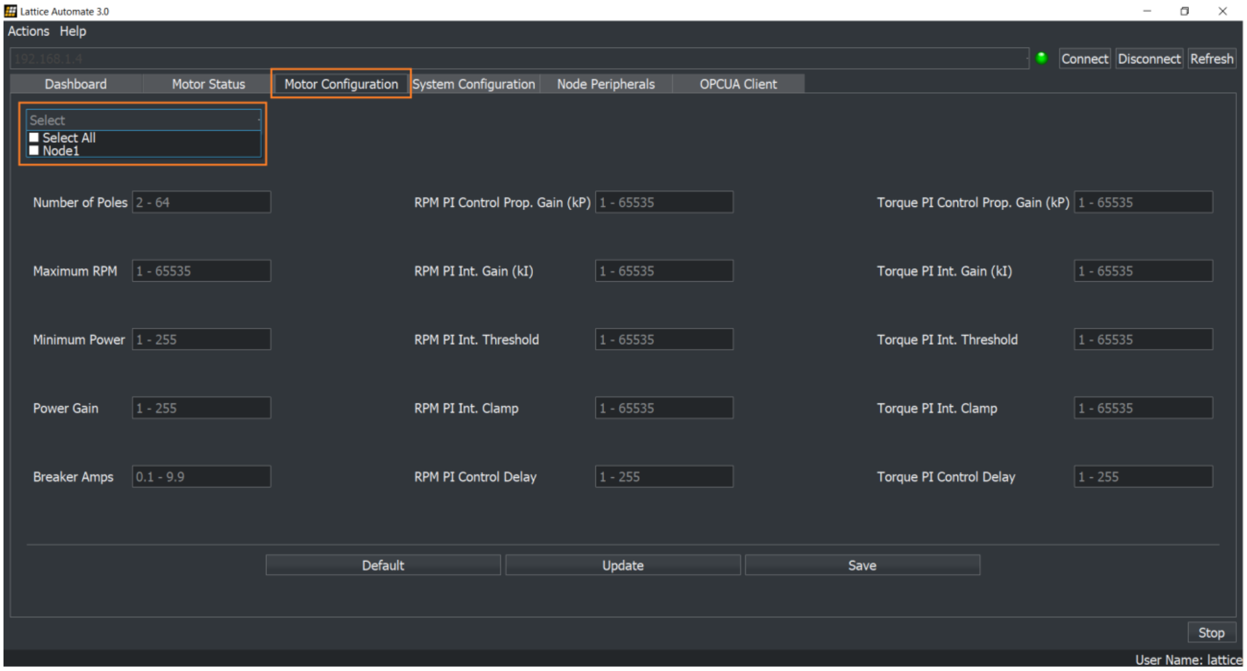
3. Click **Apply**.
4. Click **OK** on the **Updated Successfully** pop-up.



Updated Successfully.

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

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

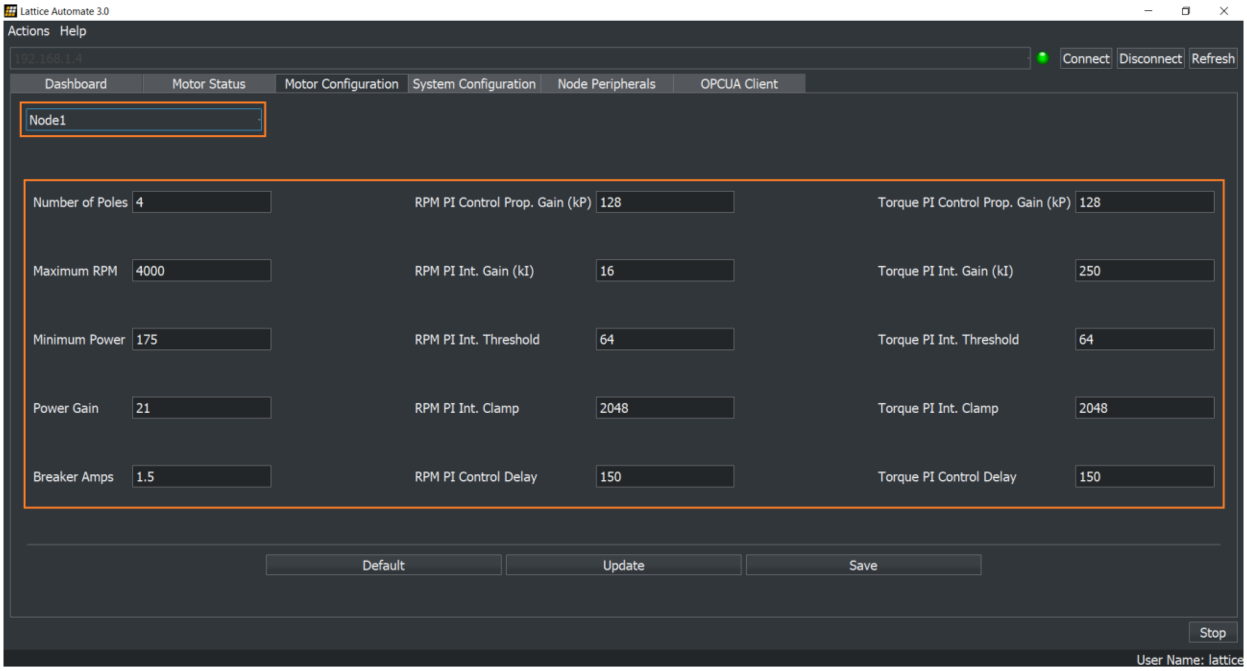


□ Select All
□ Node1

Number of Poles 2 - 64
Maximum RPM 1 - 65535
Minimum Power 1 - 255
Power Gain 1 - 255
Breaker Amps 0.1 - 9.9
RPM PI Control Prop. Gain (kP) 1 - 65535
RPM PI Int. Gain (kI) 1 - 65535
RPM PI Int. Threshold 1 - 65535
RPM PI Int. Clamp 1 - 65535
RPM PI Control Delay 1 - 255
Torque PI Control Prop. Gain (kP) 1 - 65535
Torque PI Int. Gain (kI) 1 - 65535
Torque PI Int. Threshold 1 - 65535
Torque PI Int. Clamp 1 - 65535
Torque PI Control Delay 1 - 255

Figure 7.11. GUI Application Motor Configuration: Node Selection

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



Node1

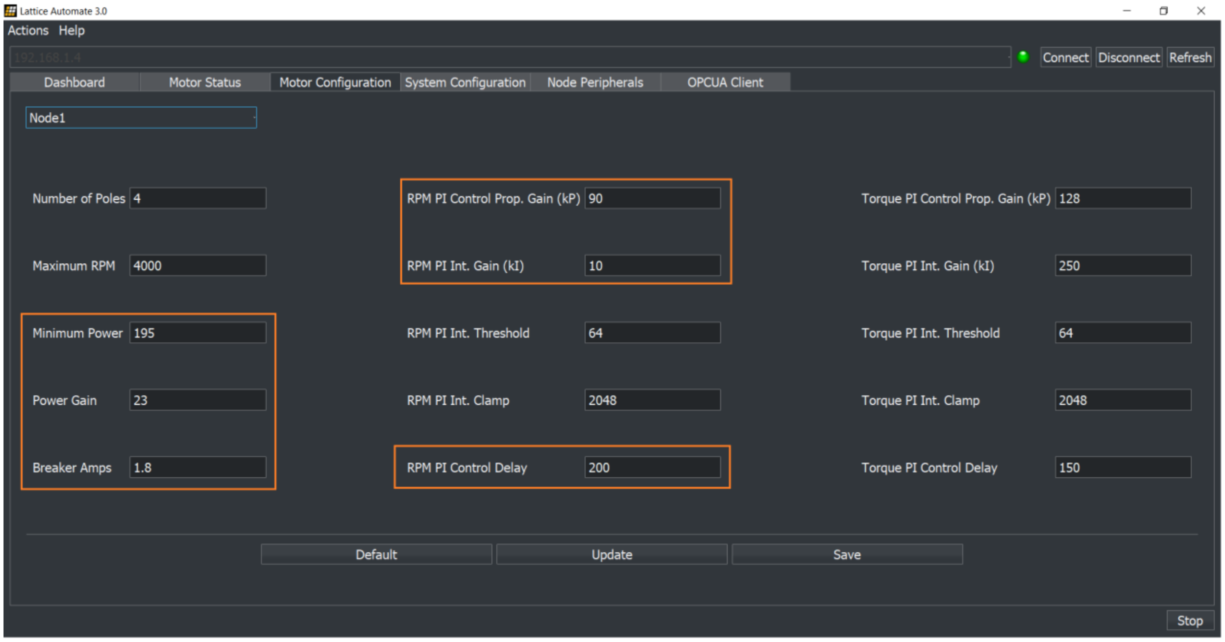
Number of Poles 4
Maximum RPM 4000
Minimum Power 175
Power Gain 21
Breaker Amps 1.5
RPM PI Control Prop. Gain (kP) 128
RPM PI Int. Gain (kI) 16
RPM PI Int. Threshold 64
RPM PI Int. Clamp 2048
RPM PI Control Delay 150
Torque PI Control Prop. Gain (kP) 128
Torque PI Int. Gain (kI) 250
Torque PI Int. Threshold 64
Torque PI Int. Clamp 2048
Torque PI Control Delay 150

Figure 7.12. GUI Application Motor Configuration: Node Selected

8. Change the Motor Configuration to the following values:

- RPI PI Control Prop. Gain(kP): **90**
- RPI PI Int. Gain(kI): **10**
- RPI PI Control delay: **200**
- Minimum Power: **190 - 200**
- Power gain: **22 – 24**
- Breaker Amps: **1.8 - 2.0**

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

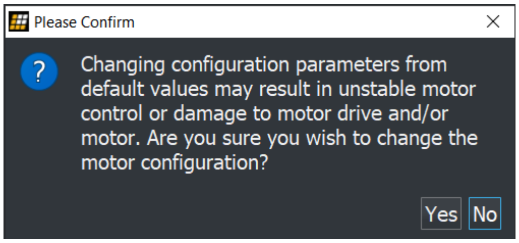


Node1

Number of Poles	4	RPM PI Control Prop. Gain (kP)	90	Torque PI Control Prop. Gain (kP)	128
Maximum RPM	4000	RPM PI Int. Gain (kI)	10	Torque PI Int. Gain (kI)	250
Minimum Power	195	RPM PI Int. Threshold	64	Torque PI Int. Threshold	64
Power Gain	23	RPM PI Int. Clamp	2048	Torque PI Int. Clamp	2048
Breaker Amps	1.8	RPM PI Control Delay	200	Torque PI Control Delay	150

Figure 7.13. GUI Application Motor Configuration: Configuration Value Changes

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



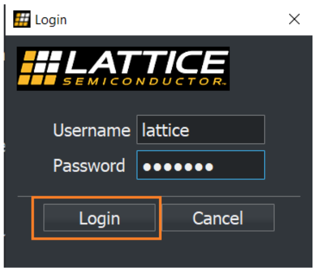
Changing configuration parameters from default values may result in unstable motor control or damage to motor drive and/or motor. Are you sure you wish to change the motor configuration?

Figure 7.14. GUI Application Motor Configuration: Warning Message

13. Enter the credentials and click Login:

Username**: lattice**

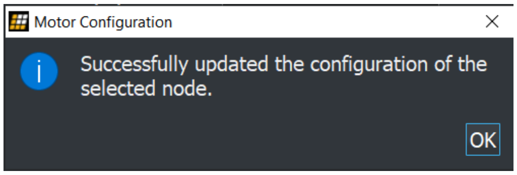
Password**: lattice**



Username lattice
Password •••••••

Figure 7.15. GUI Application Motor Configuration: Authentication Pop-up

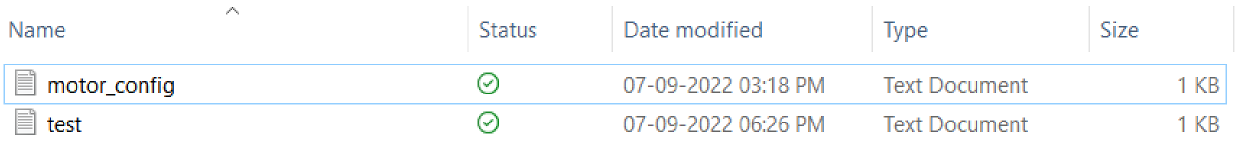
14. Click **OK** in the **Successfully updated the configuration of the selected node** pop-up.



Successfully updated the configuration of the selected node.

Figure 7.16. GUI Application Motor Configuration: Update Configuration

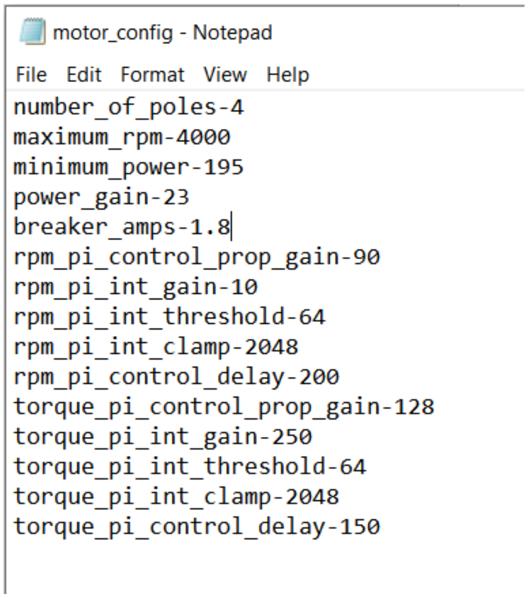
15. Locate the motor_config.txt file. It is saved in the location that user selected in step 2.



Name	Status	Date modified	Type	Size
motor_config	✓	07-09-2022 03:18 PM	Text Document	1 KB
test	✓	07-09-2022 06:26 PM	Text Document	1 KB

Figure 7.17. Motor Config File Saved in Host PC

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



```
number_of_poles-4
maximum_rpm-4000
minimum_power-195
power_gain-23
breaker_amps-1.8
rpm_pi_control_prop_gain-90
rpm_pi_int_gain-10
rpm_pi_int_threshold-64
rpm_pi_int_clamp-2048
rpm_pi_control_delay-200
torque_pi_control_prop_gain-128
torque_pi_int_gain-250
torque_pi_int_threshold-64
torque_pi_int_clamp-2048
torque_pi_control_delay-150
```

Figure 7.18. Motor Config.txt file in Host PC

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

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

Note: The user can gradually increase the RPM up to 2000 RPM

The following RPM increments are recommended: 120, 400, 800, 1400, 1600, 1800, and 2000.
Note: The target RPM value should not exceed the Maximum RPM value entered in the **Motor Configuration** tab.

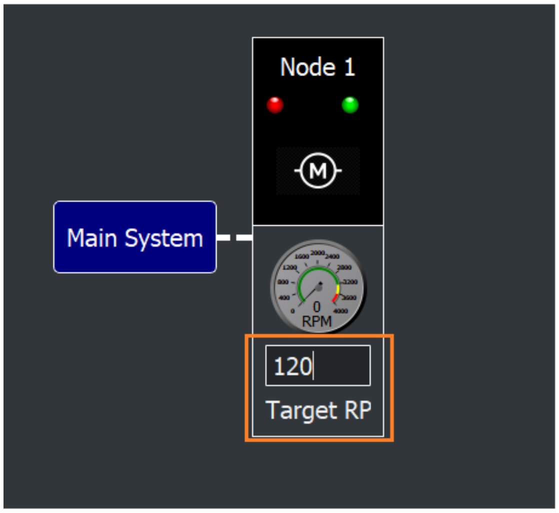
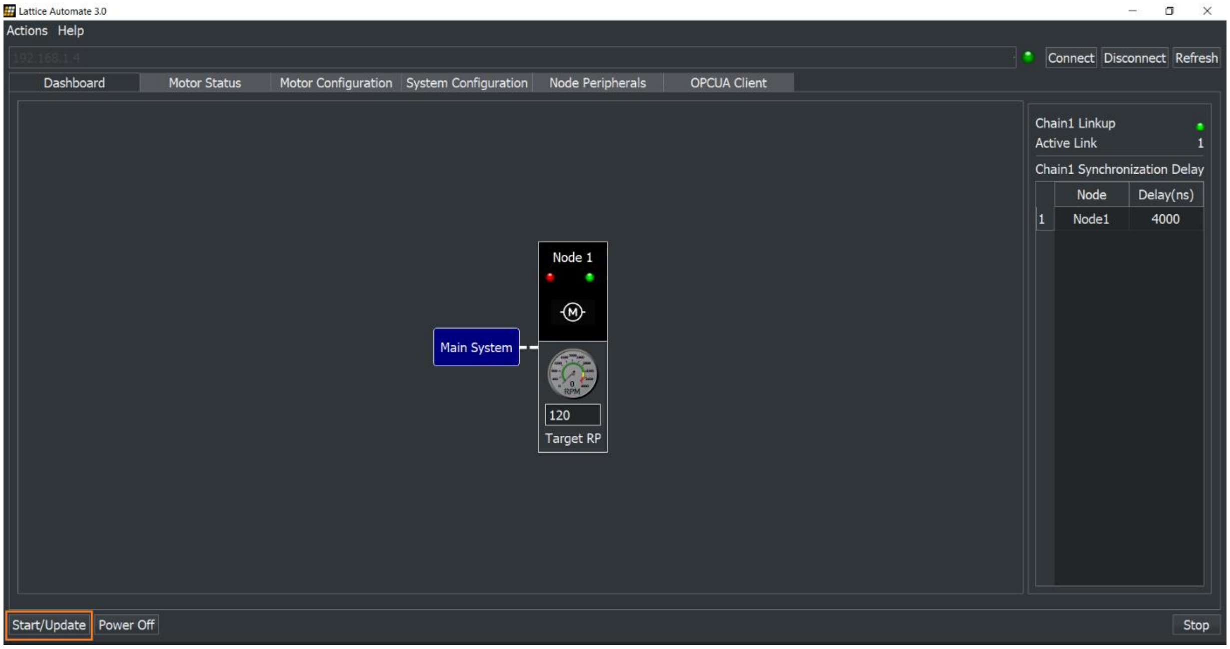


Figure 7.19. GUI Application Dashboard: Set Target RPM

3. Click **Start/Update**.



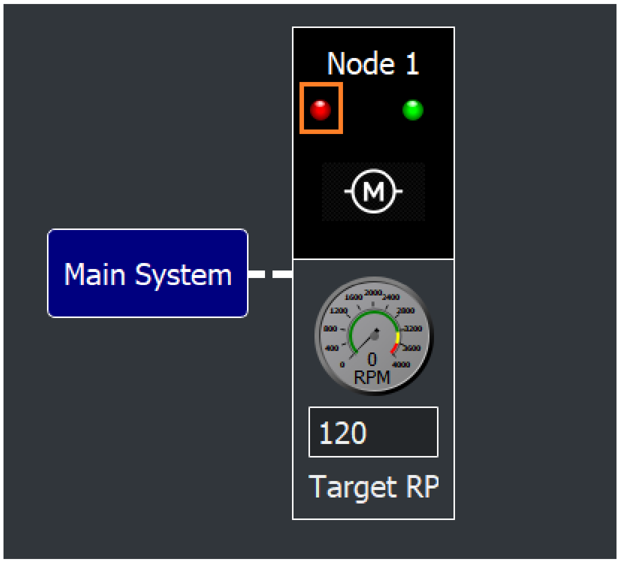
Chain1 Linkup
Active Link 1
Chain1 Synchronization Delay

	Node	Delay(ns)
1	Node1	4000

Target RP: 120

Figure 7.20. GUI Application Dashboard: Start/Update

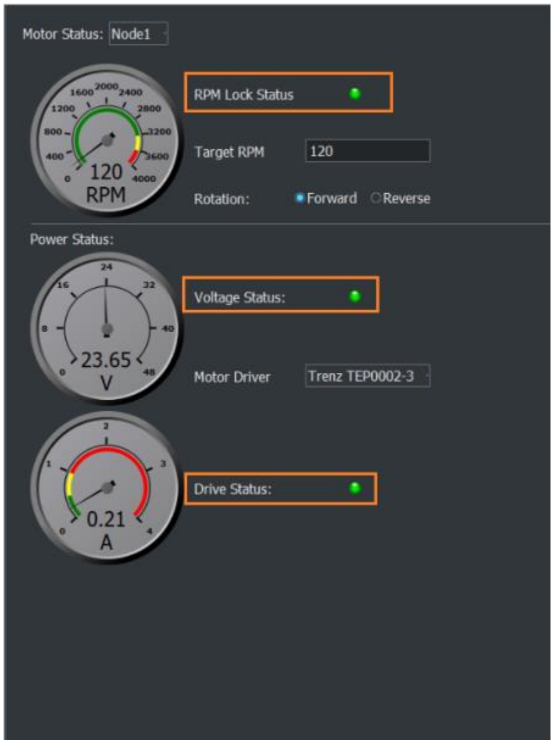
4. After the RPM Lock is achieved, the Node LED is green.



120

Figure 7.21. GUI 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 user wants to check.


Motor Status: Node1

RPM Lock Status ●

Target RPM 120

Rotation: ● Forward ○ Reverse

Power Status:

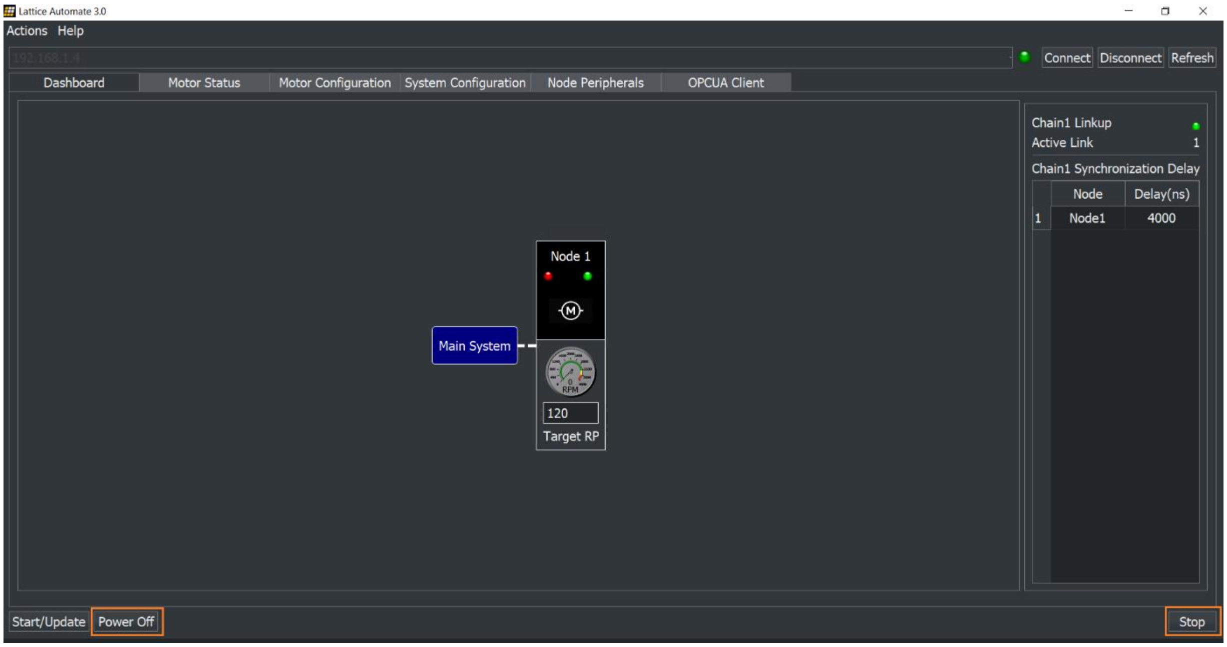
Voltage Status: ●

Motor Driver Trenz TEP0002-3

Drive Status: ●

Figure 7.22. GUI 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**.



Chain1 Linkup ●
Active Link 1
Chain1 Synchronization Delay

	Node	Delay(ns)
1	Node1	4000

120

Figure 7.23. GUI 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.

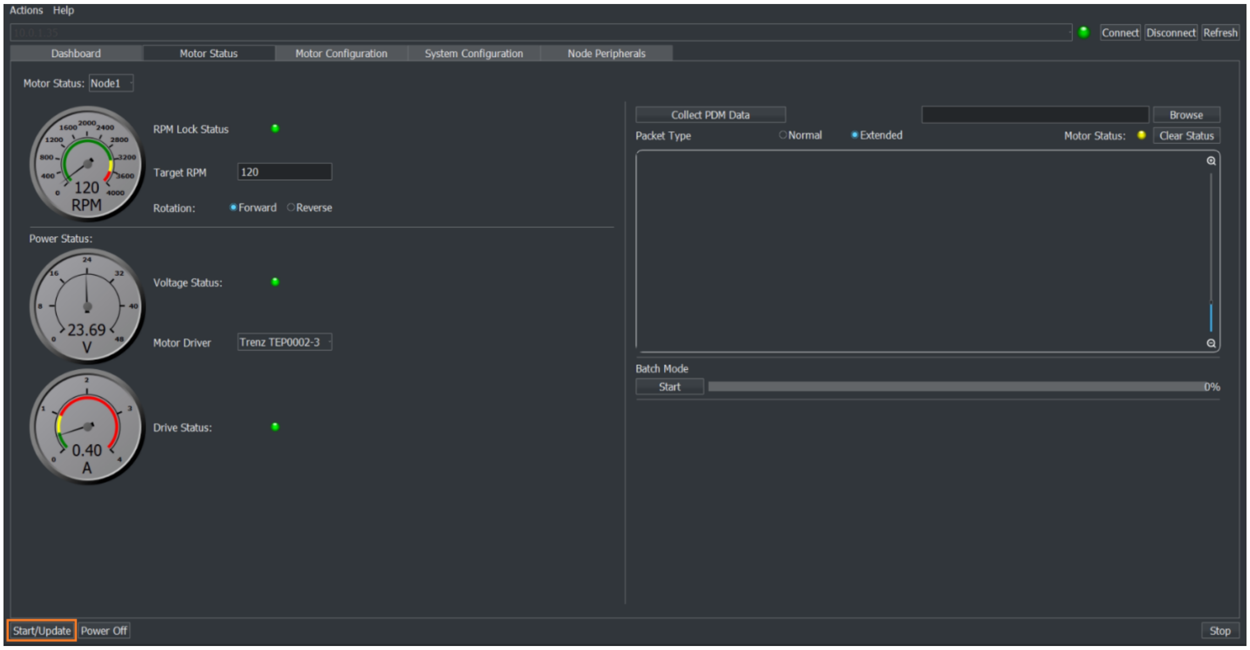
7.6. Motor Status Test

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

Notes:

- Users can gradually increase the RPM up to 2000 RPM
- The following RPM increments are recommended: 120, 400, 800, 1400, 1600, 1800, and 2000.
- Target RPM should not exceed the Maximum RPM value entered in the Motor Configuration tab.

3. Click **Start/Update**.



Motor Status: Node1

120 RPM

Target RPM: 120

Rotation: ● Forward ○ Reverse

23.69 V

Motor Driver: Trenz TEP0002-3

0.40 A

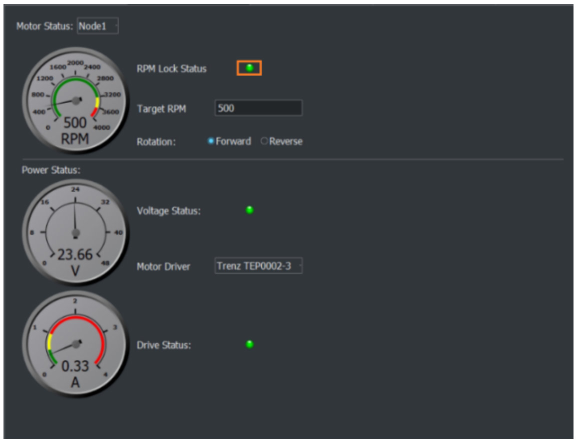
Packet Type: ○ Normal ● Extended

Motor Status: ●

0%

Figure 7.24. GUI Application Motor Status: Start/Update

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



Motor Status: Node1

500 RPM

Target RPM: 500

Rotation: ● Forward ○ Reverse

23.66 V

Motor Driver: Trenz TEP0002-3

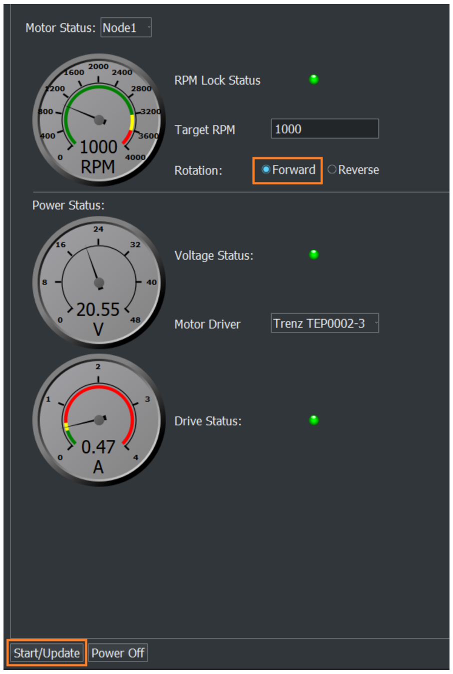
0.33 A

Figure 7.25. GUI 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.

7.7. Forward/Reverse Rotation Test

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



Motor Status: Node1

RPM Lock Status: ●

Target RPM: 1000

Rotation: ● Forward ○ Reverse

Power Status:

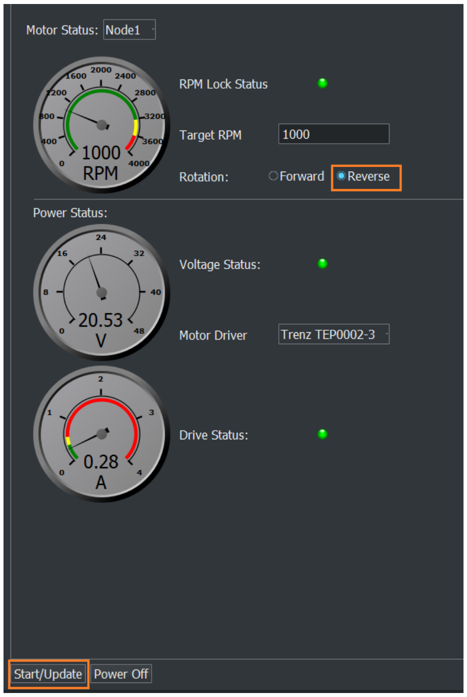
Voltage Status: ●

Motor Driver: Trenz TEP0002-3

Drive Status: ●

Figure 7.26. GUI Application Motor Status: Forward Rotation Status

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



Motor Status: Node1

RPM Lock Status: ●

Target RPM: 1000

Rotation: ○ Forward ● Reverse

Power Status:

Voltage Status: ●

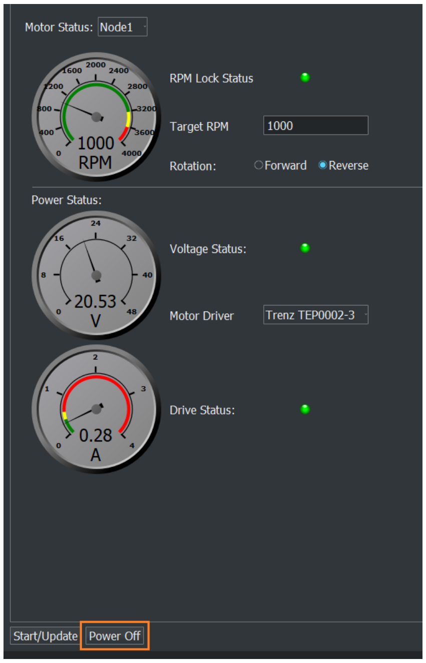
Motor Driver: Trenz TEP0002-3

Drive Status: ●

Figure 7.27. GUI Application Motor Status: Reverse Rotation Status

7. Click **Start/Update**. The Motor changes the direction to rotate in a counterclockwise direction.
8. Wait until the RPM Lock status becomes green.

9. To stop the motor, click **Stop**, and then **Power Off**.



Motor Status: Node1

RPM Lock Status ●

Target RPM 1000

Rotation: ○ Forward ● Reverse

1000 RPM

Power Status:

Voltage Status: ●

Motor Driver Trenz TEP0002-3

20.53 V

Drive Status: ●

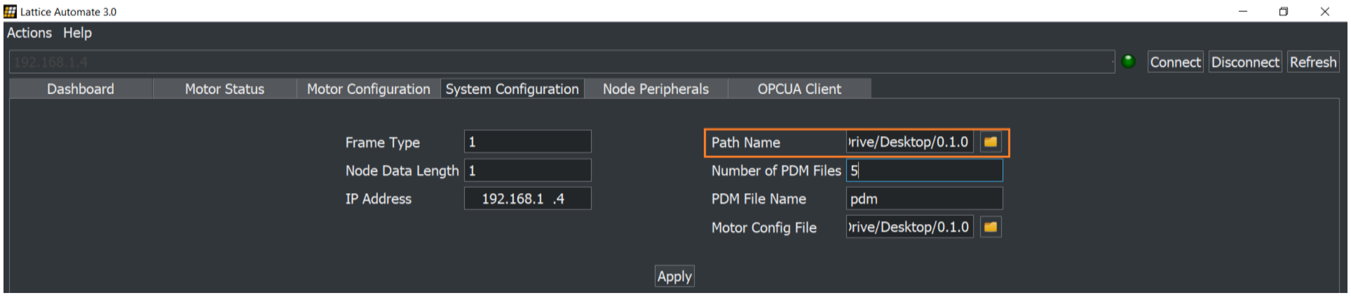
0.28 A

Figure 7.28. GUI Application Motor Status: Motor Power Off

7.8. PDM Data Collection

7.8.1. Collect PDM Data

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



Lattice Automate 3.0

192.168.1.4

Frame Type 1

Node Data Length 1

IP Address 192.168.1 .4

Path Name ›rive/Desktop/0.1.0

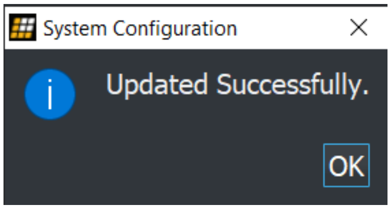
Number of PDM Files 5

PDM File Name pdm

Motor Config File ›rive/Desktop/0.1.0

Figure 7.29. GUI Application PDM Data: Path Name

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

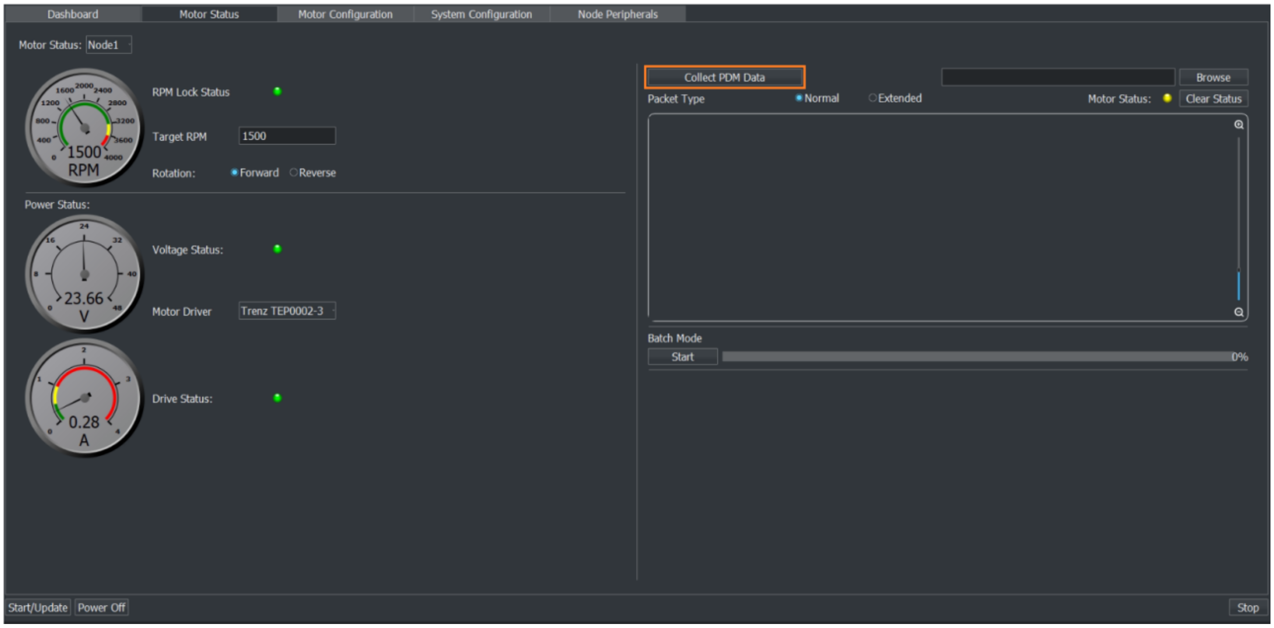


System Configuration

Updated Successfully.

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

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



Motor Status: Node1

Target RPM 1500

Rotation: ● Forward ○ Reverse

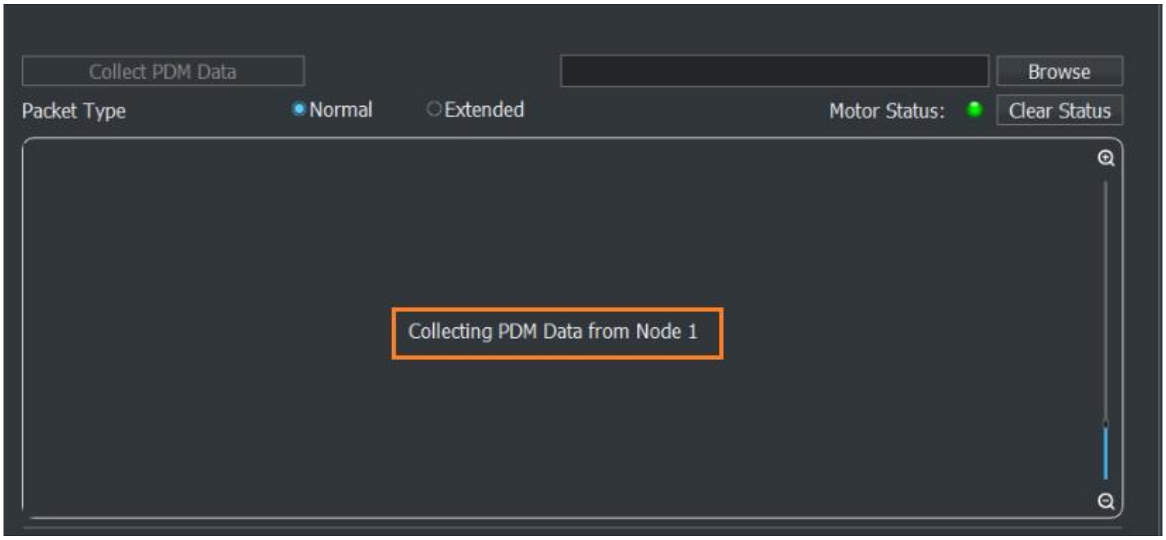
Motor Driver Trenz TEP0002-3

Packet Type ● Normal ○ Extended

Figure 7.31. GUI Application PDM Data: Collect PDM Data

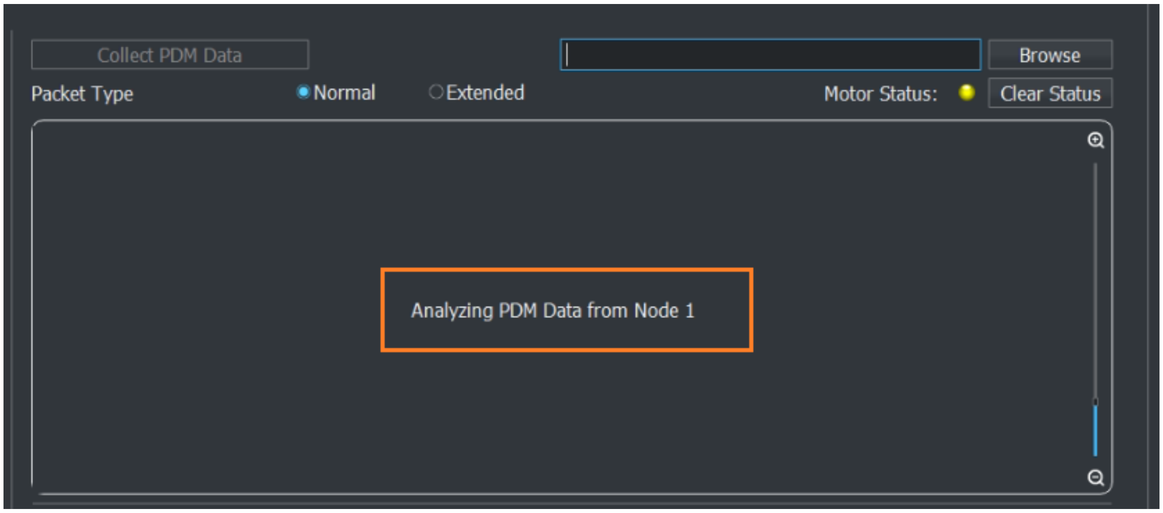
8. Choose Normal or Extended for the Packet type. Both Normal mode and Extended mode appear the same from a user's perspective, but Extended mode is more efficient if there are many nodes in the chain.

   NOTE: See Details about PDM Data Collection Process section for a description of Normal mode and Extended mode.
9. **Collecting PDM Data from Node** and **Analyzing PDM Data from Node** messages are displayed while the image is captured.



Collecting PDM Data from Node 1

Figure 7.32. GUI Application PDM Data: Collecting PDM Data from Node



Analyzing PDM Data from Node 1

Figure 7.33. GUI Application PDM Data: Analyzing PDM Data from Node

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

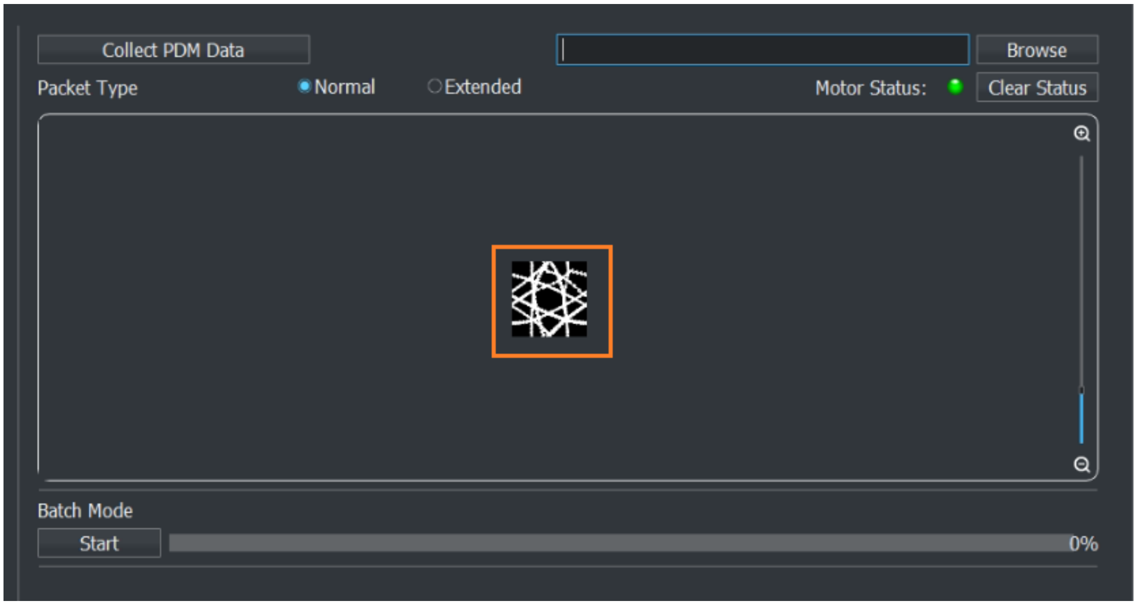


Figure 7.34. GUI Application PDM Data: PDM Image

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

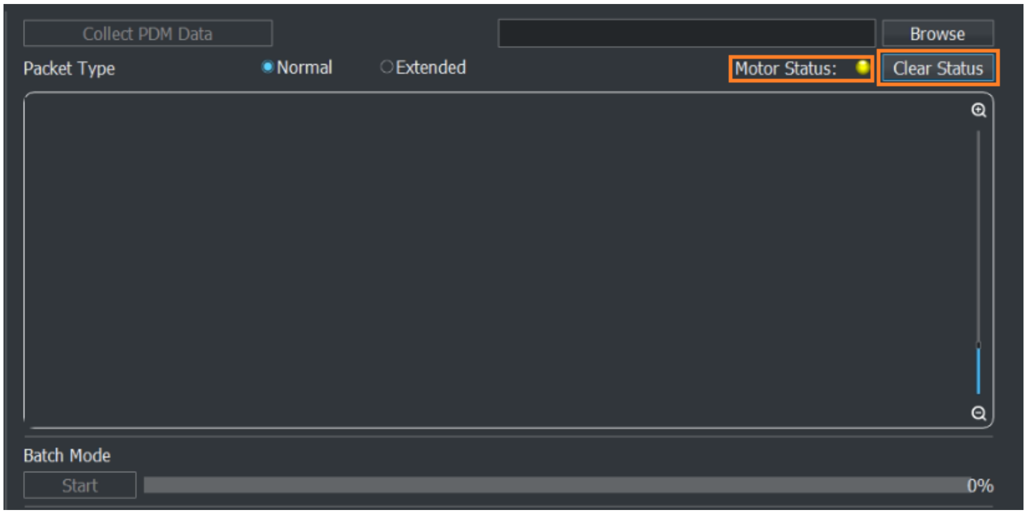
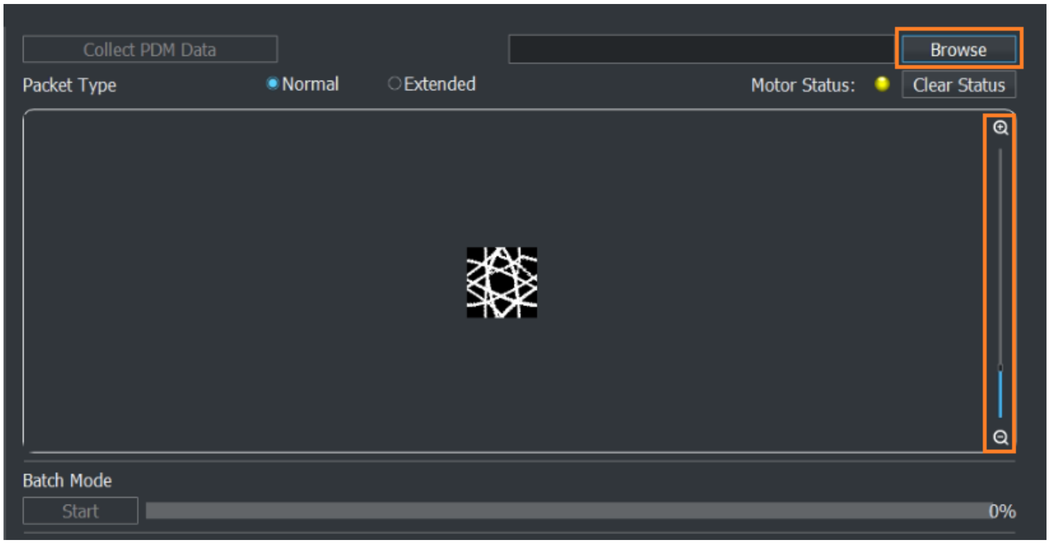


Figure 7.35. GUI Application PDM Data: Clear Status and Motor Status

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.



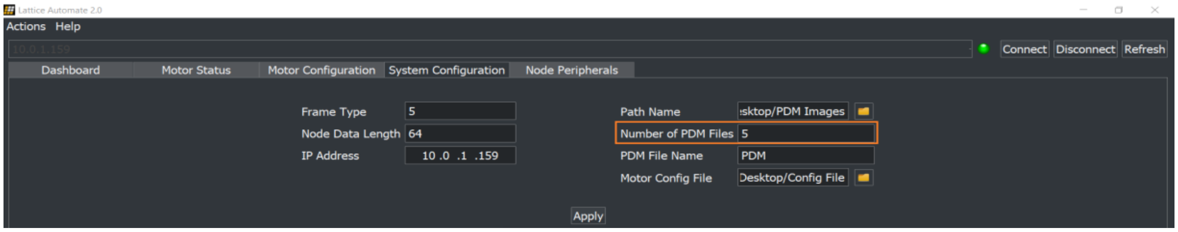
Packet Type ● Normal ○ Extended

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

7.8.2. Batch Mode

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

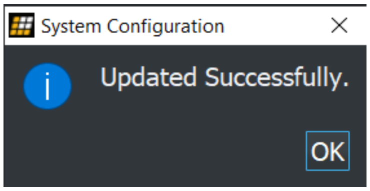
1. Click the **System configuration** tab and enter the number of PDM files that user wants to collect. The system supports a range of 1-20 PDM files.
2. If the users have not yet selected a location to save the images, click the folder image next to Path Name and browse for a location, as described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



Frame Type: 5
Node Data Length: 64
IP Address: 10 .0 .1 .159
Path Name: :sktop/PDM Images
Number of PDM Files: 5
PDM File Name: PDM
Motor Config File: Desktop/Config File

Figure 7.37. GUI Application PDM Batch Mode: Number of PDM files

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



Updated Successfully.

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

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

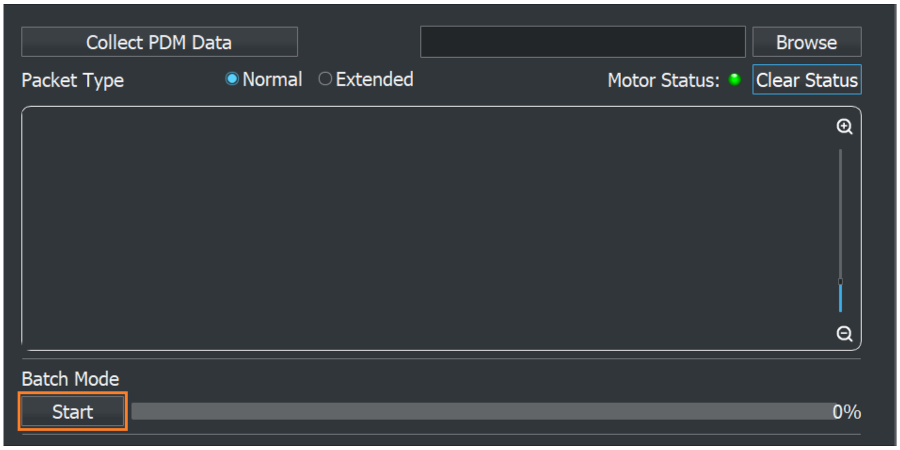


Figure 7.39. GUI Application PDM Batch Mode: Start

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

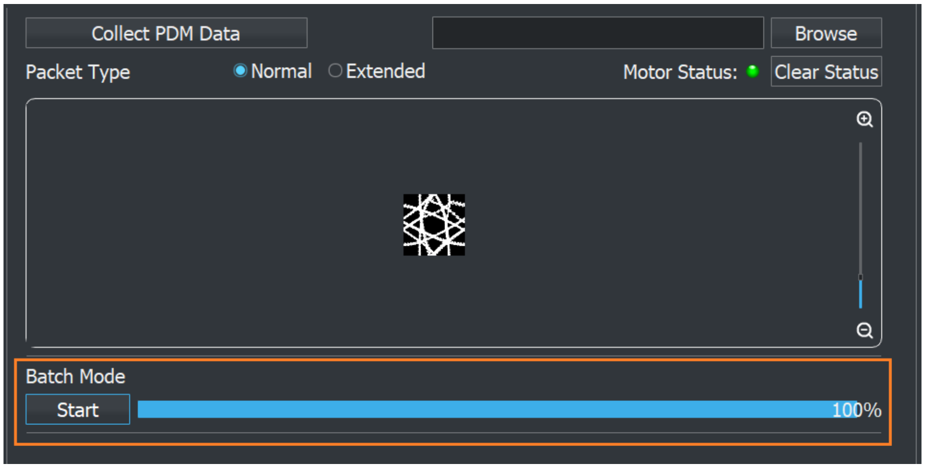


Figure 7.40. GUI Application PDM Batch Mode: Batch Mode 100% Status

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

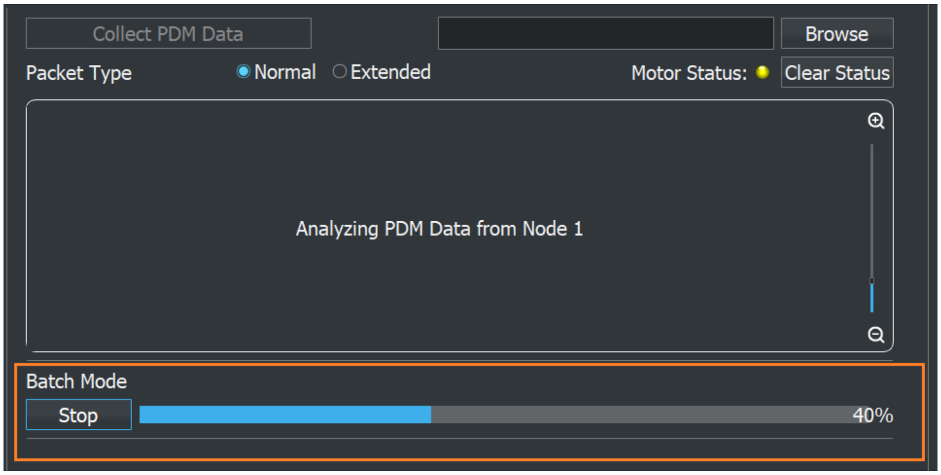


Figure 7.41. GUI Application PDM Batch Mode: Collecting Multiple Images

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

7.8.3. Details about PDM Data Collection Process

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

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

When the node system receives the frame, the node system's RISC-V CPU uses the PDM Data Collector to fill the FIFO DMA with data collected from the motor. When the FIFO DMA is full, 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 GUI. 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.

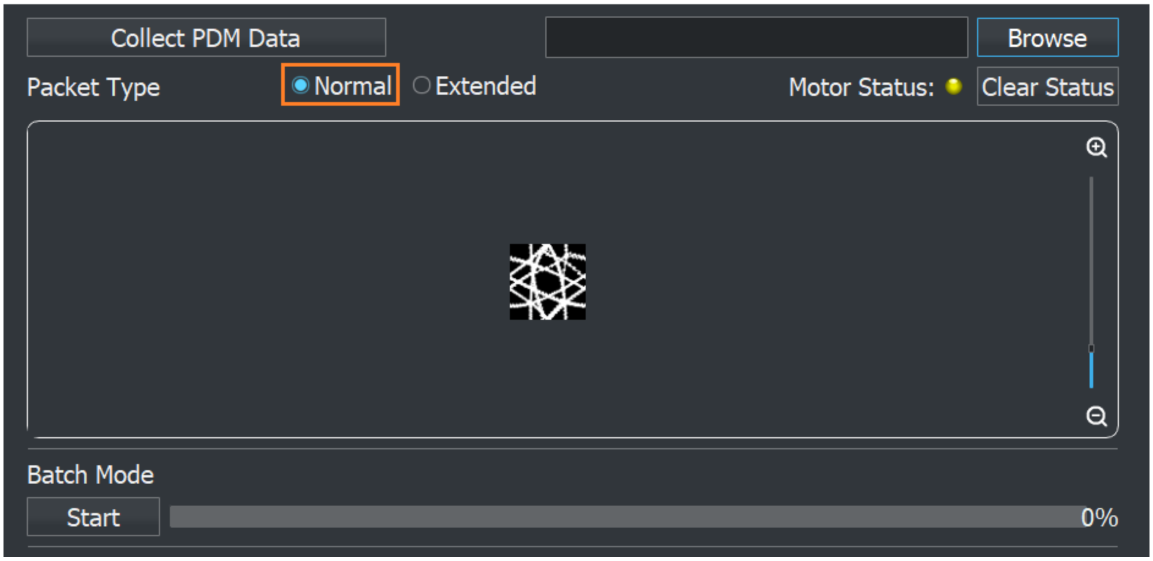


Figure 7.42. GUI Application PDM Data: Normal Mode

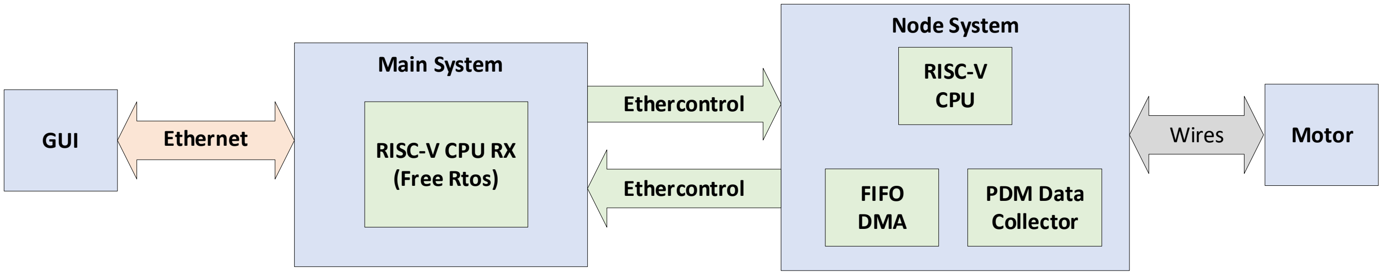


Figure 7.43. GUI Application PDM Data: Components Involved in PDM Data Collection Process

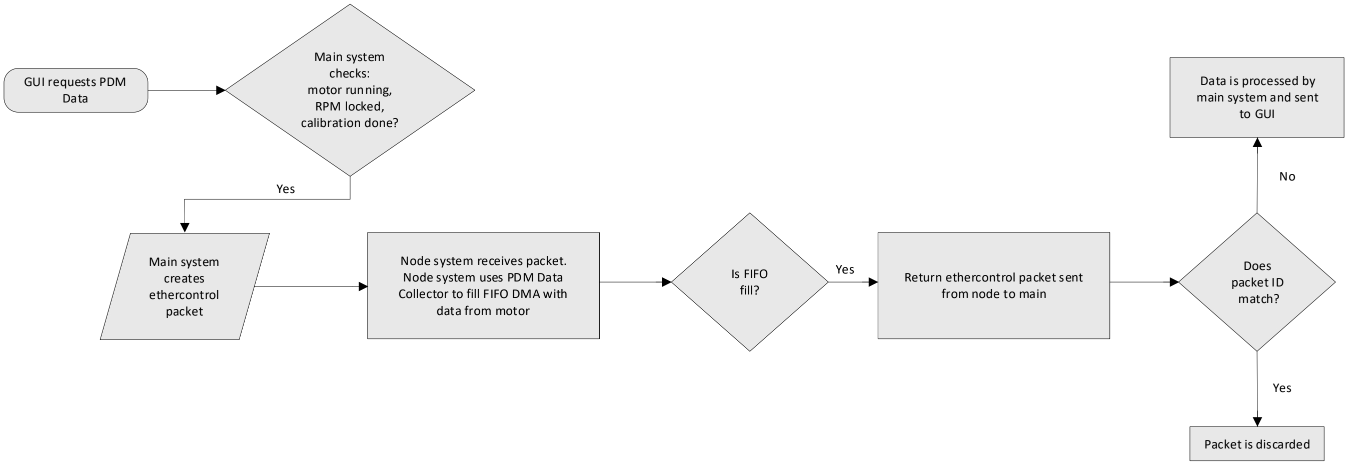

GUI requests PDM Data
Main system checks: motor running, RPM locked, calibration done?
Yes
Main system creates ethercontrol packet
Node system receives packet. Node system uses PDM Data Collector to fill FIFO DMA with data from motor
Is FIFO fill?
Yes
Return ethercontrol packet sent from node to main
Does packet ID match?
No
Data is processed by main system and sent to GUI
Yes
Packet is discarded

Figure 7.44. GUI Application PDM Data: Process Flow for PDM Data Collection

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

7.8.4.1. Introduction and Purpose of the Switches

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

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

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

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

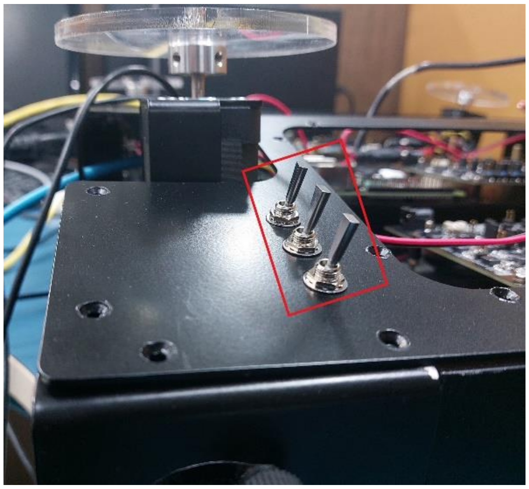


Figure 7.45. Three Resistor Switches Tilted Towards the Resistors (Meaning All Three Resistors are Enabled and the Motor is "good")

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

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

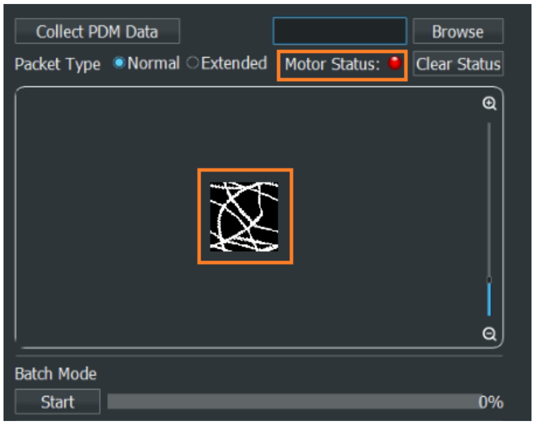


Figure 7.46. PDM Data collection: Example PDM Image of a Faulty Motor

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

7.9. Testing Node Peripherals

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

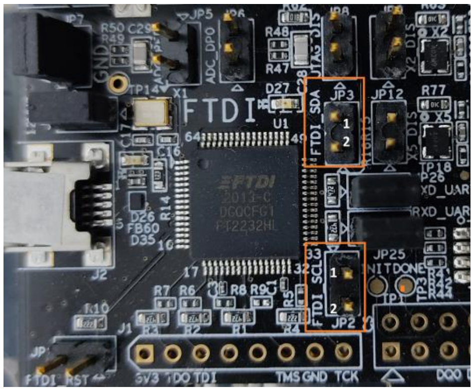


Figure 7.47. Certus NX Board - I2C Connection

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

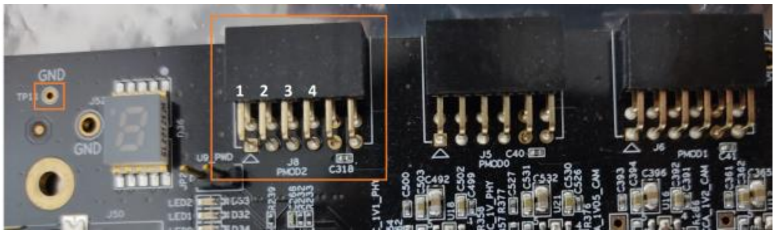


Figure 7.48. Certus NX Board - SPI Connection

7.9.1. Make the Connection in Total Phase Control Center

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

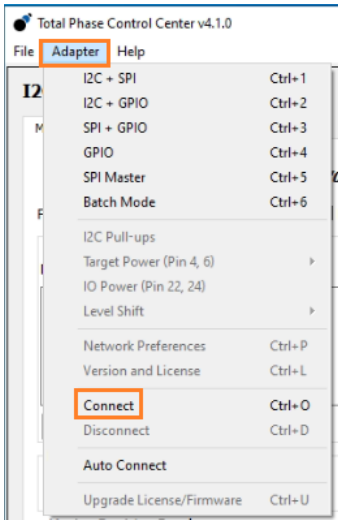
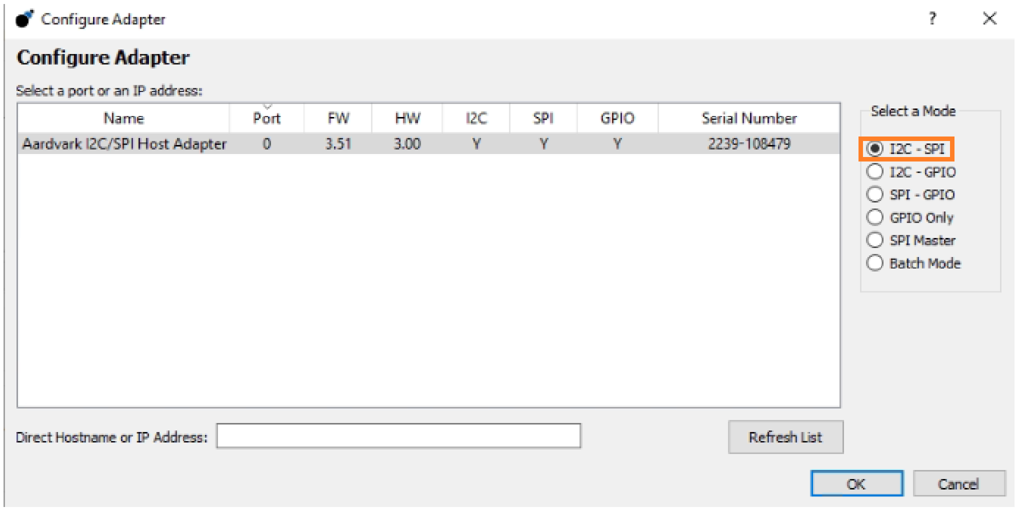


Figure 7.49. Total Phase - Adapter Setting (I2C)

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



Configure Adapter

Select a port or an IP address:

Name	Port	FW	HW	I2C	SPI	GPIO	Serial Number
Aardvark I2C/SPI Host Adapter	0	3.51	3.00	Y	Y	Y	2239-108479

Select a Mode

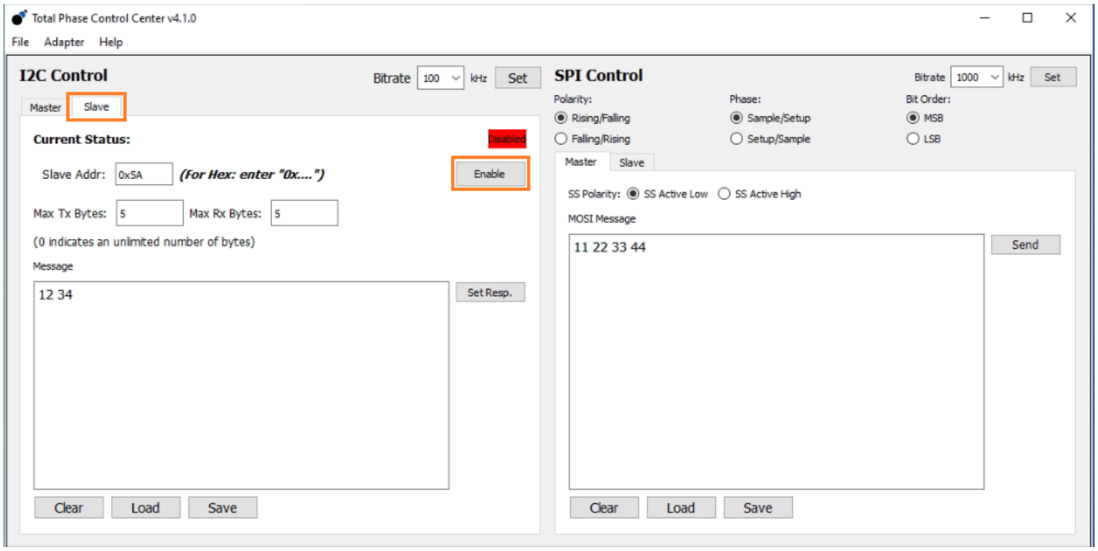
● I2C - SPI
○ I2C - GPIO
○ SPI - GPIO
○ GPIO Only
○ SPI Master
○ Batch Mode

Direct Hostname or IP Address:

Figure 7.50. Total Phase - I2C-SPI Selection

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

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

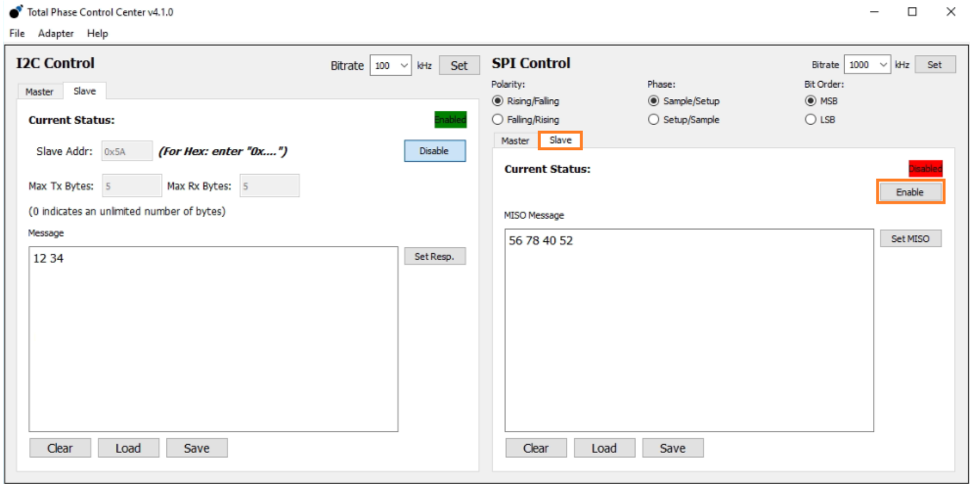

Total Phase Control Center v4.1.0

I2C Control — Bitrate 100 kHz — Slave tab
Current Status: Disabled
Slave Addr: 0x5A (For Hex: enter "0x....")
Max Tx Bytes: 5 Max Rx Bytes: 5
(0 indicates an unlimited number of bytes)
Message: 12 34

SPI Control — Bitrate 1000 kHz
Polarity: ● Rising/Falling ○ Falling/Rising
Phase: ● Sample/Setup ○ Setup/Sample
Bit Order: ● MSB ○ LSB
Master tab — SS Polarity: ● SS Active Low ○ SS Active High
MOSI Message: 11 22 33 44

Figure 7.51. Total Phase - I2C Control (I2C)

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

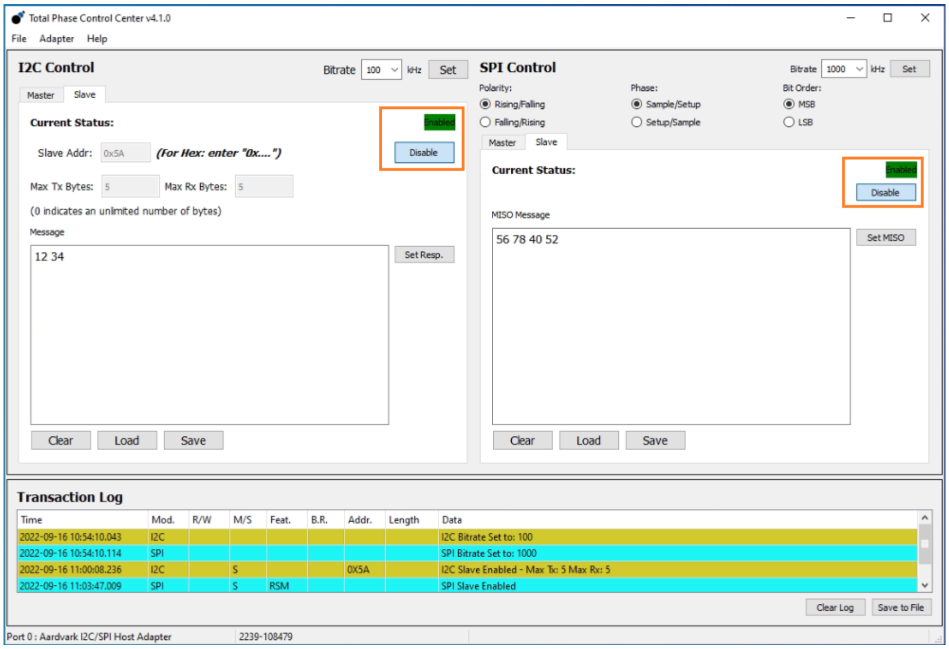

Total Phase Control Center v4.1.0

I2C Control — Bitrate 100 kHz — Slave tab
Current Status: Enabled
Slave Addr: 0x5A (For Hex: enter "0x....")
Max Tx Bytes: 5 Max Rx Bytes: 5
(0 indicates an unlimited number of bytes)
Message: 12 34

SPI Control — Bitrate 1000 kHz
Polarity: ● Rising/Falling ○ Falling/Rising
Phase: ● Sample/Setup ○ Setup/Sample
Bit Order: ● MSB ○ LSB
Slave tab — Current Status: Disabled
MISO Message: 56 78 40 52

Figure 7.52. Total Phase - SPI Control: SPI Enable

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


Total Phase Control Center v4.1.0

I2C Control

Bitrate: 100 kHz

Current Status: Enabled

Slave Addr: 0x5A *(For Hex: enter "0x....")*

Max Tx Bytes: 5 Max Rx Bytes: 5

(0 indicates an unlimited number of bytes)

Message: 12 34

SPI Control

Bitrate: 1000 kHz

Polarity: ● Rising/Falling ○ Falling/Rising

Phase: ● Sample/Setup ○ Setup/Sample

Bit Order: ● MSB ○ LSB

Current Status: Enabled

MISO Message: 56 78 40 52

Transaction Log

Time	Mod.	R/W	M/S	Feat.	B.R.	Addr.	Length	Data
2022-09-16 10:54:10.043	I2C							I2C Bitrate Set to: 100
2022-09-16 10:54:10.114	SPI							SPI Bitrate Set to: 1000
2022-09-16 11:00:08.236	I2C		S			0X5A		I2C Slave Enabled - Max Tx: 5 Max Rx: 5
2022-09-16 11:03:47.009	SPI		S	RSM				SPI Slave Enabled

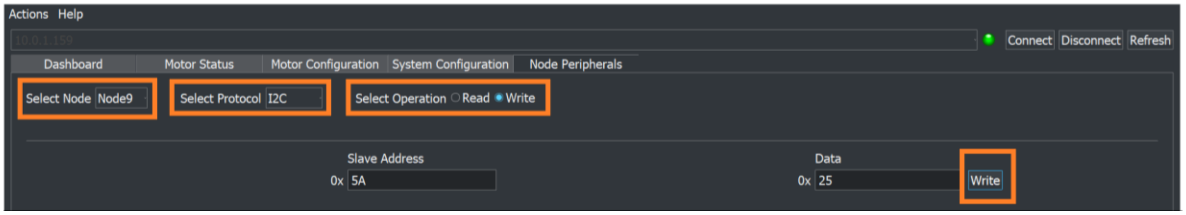
Port 0 : Aardvark I2C/SPI Host Adapter 2239-108479

Figure 7.53. Total Phase - I2C and SPI Control Enabled

7.9.2. I2C

Write Operation:

1. In the Automate 3.0 GUI, click the **Node Peripherals** tab.
2. Select the **Node** which is connected to the Aardvark I2C/SPI Host Adapter.
3. Select the protocol: **I^{2}C**
4. Select the operation: **Write**
5. Enter the Slave Address 2 bit: **5A**
6. Enter the Data 2 bit: 25 (User can write any data from 0x00 to 0xFF.)
7. Click **Write**.



Select Node: Node9

Select Protocol: I2C

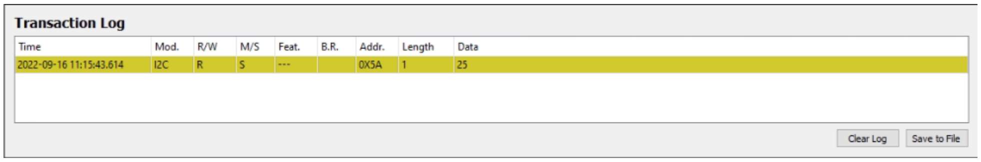
Select Operation: ○ Read ● Write

Slave Address: 0x 5A

Data: 0x 25

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

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

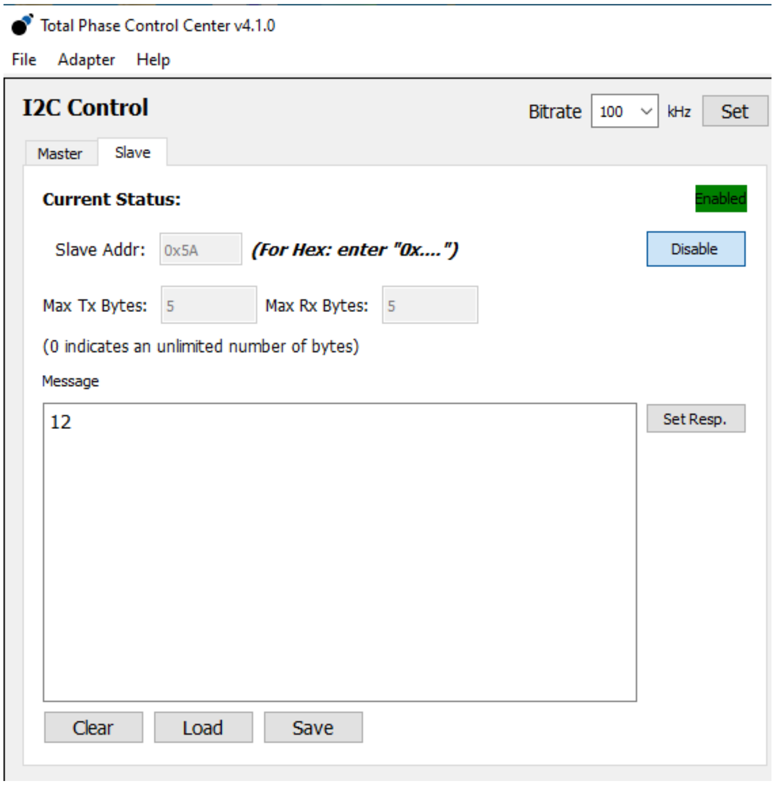

Transaction Log

Time	Mod.	R/W	M/S	Feat.	B.R.	Addr.	Length	Data
2022-09-16 11:15:43.614	I2C	R	S	---		0X5A	1	25

Figure 7.55. Total Phase - Transaction Log-I2C

Read Operation:

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



Total Phase Control Center v4.1.0

I2C Control

Bitrate: 100 kHz

Current Status: Enabled

Slave Addr: 0x5A (For Hex: enter "0x....")

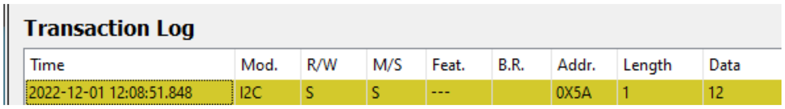
Max Tx Bytes: 5 Max Rx Bytes: 5

(0 indicates an unlimited number of bytes)

Message: 12

Figure 7.56. Total Phase Aardvark I2C Control GUI

2. Check the transaction log.



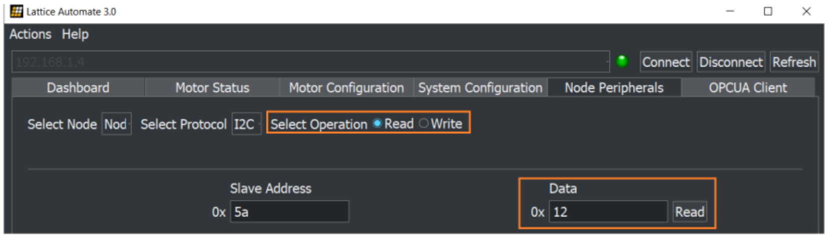
Transaction Log

Time	Mod.	R/W	M/S	Feat.	B.R.	Addr.	Length	Data
2022-12-01 12:08:51.848	I2C	S	S	---		0X5A	1	12

Figure 7.57. Total Phase - Transaction Log - I2C

3. In the Node Peripherals tab of the Automate 3.0 GUI, select the Operation: **Read**

4. Click **Read**. The data value read should be 12.



Lattice Automate 3.0

Select Node: Nod Select Protocol: I2C Select Operation: ● Read ○ Write

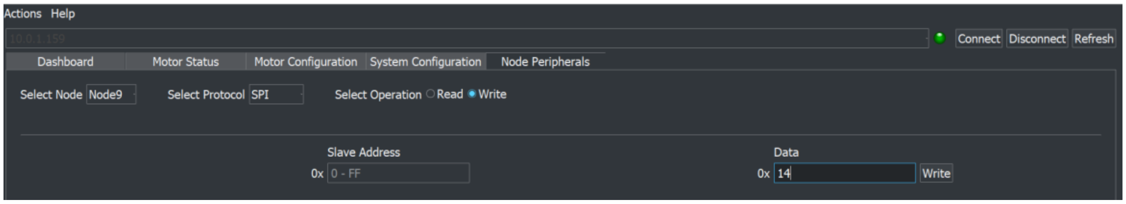
Slave Address: 0x 5a

Data: 0x 12

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

7.9.3. SPI

1. In the Automate 3.0 GUI, click the Node Peripherals tab.
2. Select the **Node** which is connected to the Aardvark I2C/SPI Host Adapter
3. Select the protocol: **SPI**
4. Select the operation: **Write**
5. Slave Address 2 bit: **Disabled**
6. Enter the Data 2 bit: 25 (User can write any data from 0x00 to 0xFF.)
7. Click **Write**.



Select Node Node9

Select Protocol SPI

Select Operation ○ Read ● Write

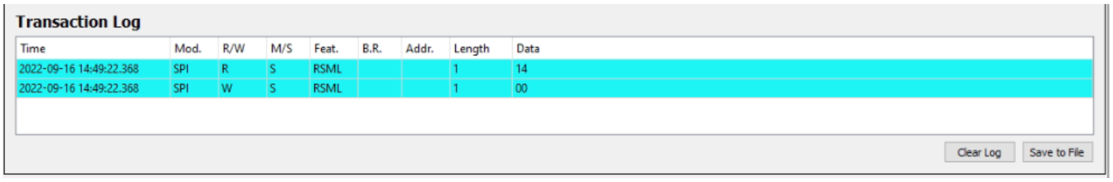
Slave Address 0x 0 - FF

Data 0x 14

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

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

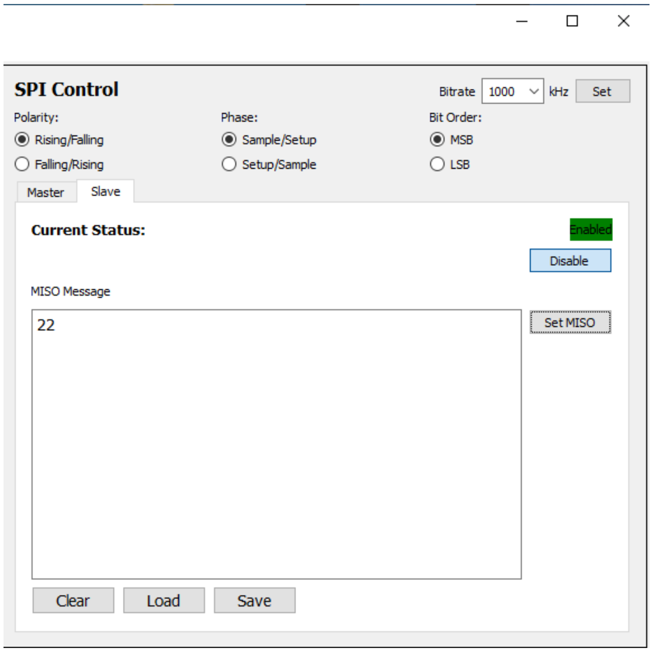
   Note: Make sure that the physical connection between the Aardvark and PMOD of the node board is correct.



Time	Mod.	R/W	M/S	Feat.	B.R.	Addr.	Length	Data
2022-09-16 14:49:22.368	SPI	R	S	RSML			1	14
2022-09-16 14:49:22.368	SPI	W	S	RSML			1	00

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

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



SPI Control

Bitrate 1000 kHz

Polarity: ● Rising/Falling ○ Falling/Rising

Phase: ● Sample/Setup ○ Setup/Sample

Bit Order: ● MSB ○ LSB

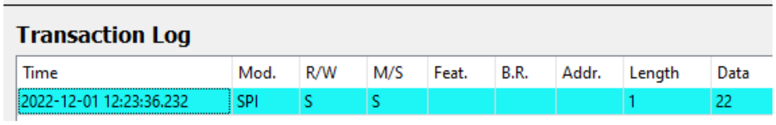
Current Status: Enabled

MISO Message

22

Figure 7.61. Total Phase - Adapter Setting: SPI Control

10. Check the transaction log.

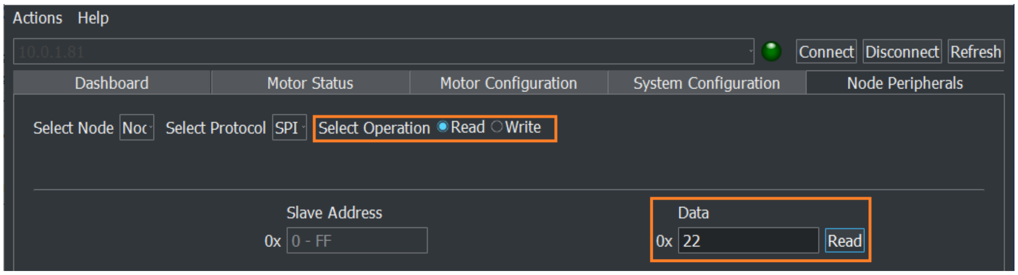


Transaction Log

Time	Mod.	R/W	M/S	Feat.	B.R.	Addr.	Length	Data
2022-12-01 12:23:36.232	SPI	S	S				1	22

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

11. In the **Node Peripherals** tab of the Automate 3.0 GUI, select the Operation: **Read**
12. Click **Read**. The value in the field Data should match the value that was entered in step 9.



Select Node Noc

Select Protocol SPI

Select Operation ● Read ○ Write

Slave Address 0x 0 - FF

Data 0x 22

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

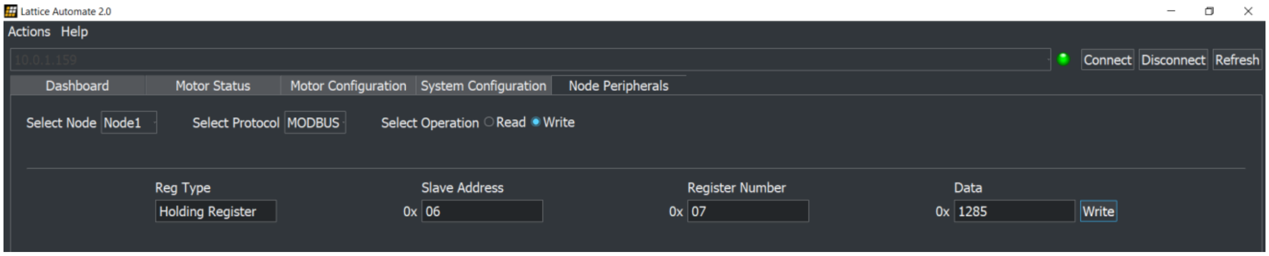
7.9.4. Modbus

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

Note: Required tool: Docklight v2.4 or later

1. In the Automate 3.0 GUI, click the **Node Peripherals** tab.
2. Enter the Slave Address: 06 (User can enter 1-10)
3. Enter the Register Number: 07 (User can enter 0-9)

4. Enter the Data: 1285 (User can enter 0-FFFF)



Select Node Node1

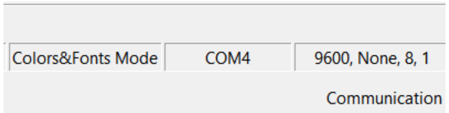
Select Protocol MODBUS

Select Operation ○ Read ● Write

Reg Type	Slave Address	Register Number	Data
Holding Register	0x 06	0x 07	0x 1285

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

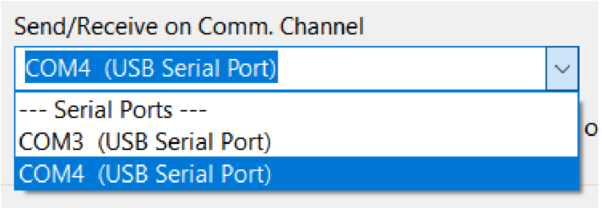
5. Open Docklight.
6. Double-click **COM**.



Colors&Fonts Mode	COM4	9600, None, 8, 1
		Communication

Figure 7.65. Docklight COM Selection

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



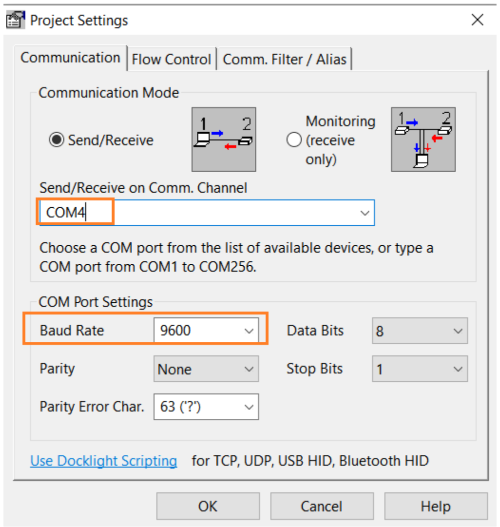
Send/Receive on Comm. Channel

COM4 (USB Serial Port)

--- Serial Ports ---
COM3 (USB Serial Port)
COM4 (USB Serial Port)

Figure 7.66. USB Serial Port Selection

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



Communication Mode

● Send/Receive ○ Monitoring (receive only)

Send/Receive on Comm. Channel: COM4

Choose a COM port from the list of available devices, or type a COM port from COM1 to COM256.

COM Port Settings

Baud Rate	9600	Data Bits	8
Parity	None	Stop Bits	1
Parity Error Char.	63 ('?')		

Use Docklight Scripting for TCP, UDP, USB HID, Bluetooth HID

Figure 7.67. Docklight - Project Settings

9. Click **Run**.

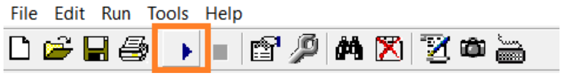
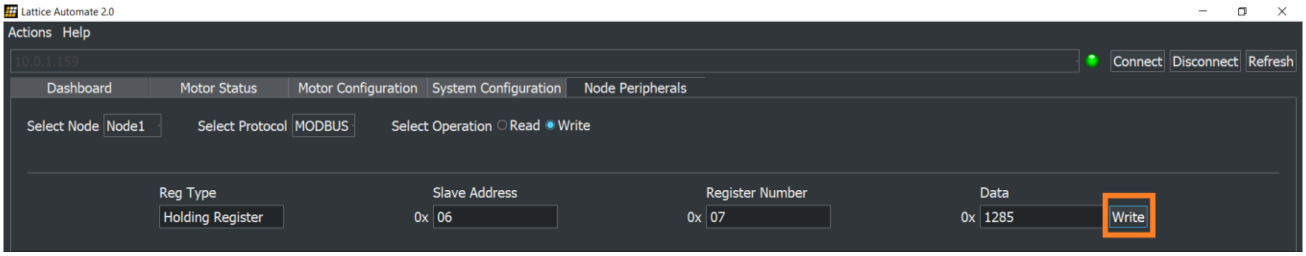


Figure 7.68. Docklight - Run

10. In the Automate 3.0 GUI, click **Write**.



10.0.1.159

Select Node Node1

Select Protocol MODBUS

Select Operation ○ Read ● Write

Reg Type: Holding Register

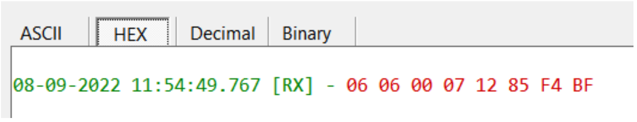
Slave Address: 0x 06

Register Number: 0x 07

Data: 0x 1285

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

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



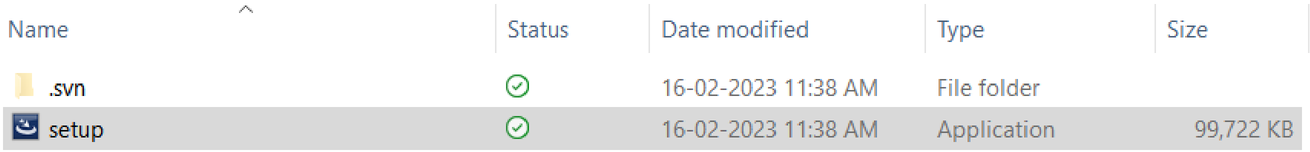
```
08-09-2022 11:54:49.767 [RX] - 06 06 00 07 12 85 F4 BF
```

Figure 7.70. Docklight Prints

Appendix A. GUI Application Installation (PC)

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

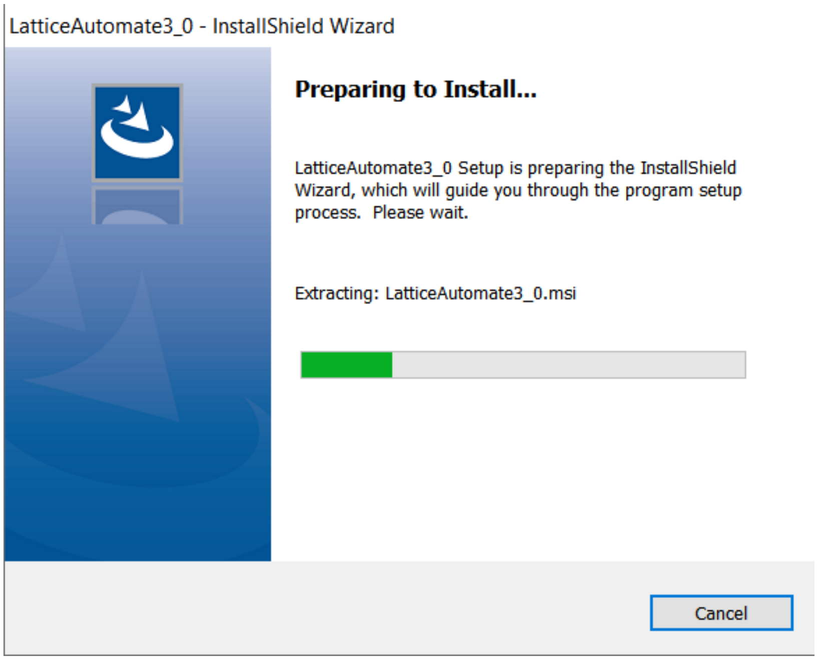
1. Download or locate the Lattice Automate 3.0 Installer.
2. Double-click on the installer to install the application.



Name	Status	Date modified	Type	Size
.svn	✓	16-02-2023 11:38 AM	File folder	
setup	✓	16-02-2023 11:38 AM	Application	99,722 KB

Figure A.1. Installer Directory

3. Wait for the installation to complete.



LatticeAutomate3_0 - InstallShield Wizard

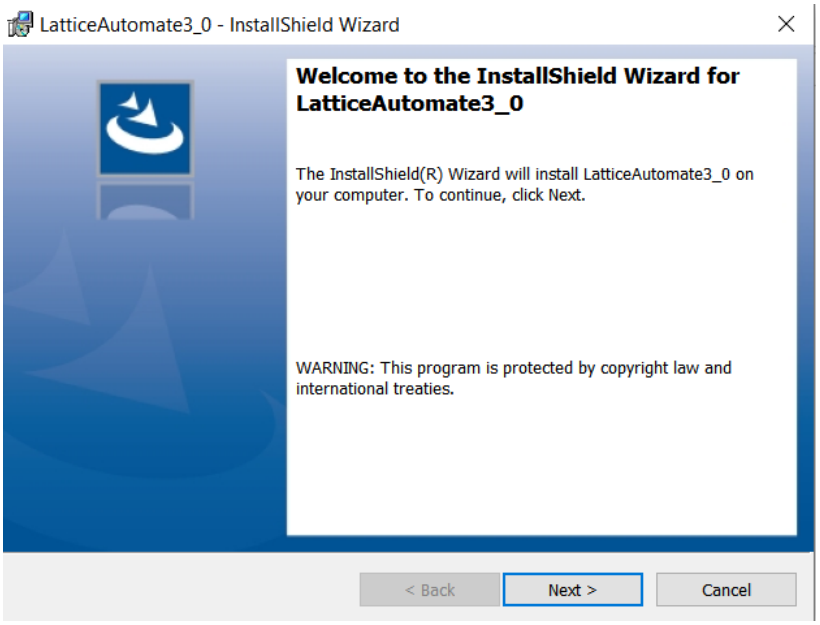
Preparing to Install...

LatticeAutomate3_0 Setup is preparing the InstallShield Wizard, which will guide you through the program setup process. Please wait.

Extracting: LatticeAutomate3_0.msi

Figure A.2. Automate 3.0 Initial Install Setup

4. Click **Next**.


LatticeAutomate3_0 - InstallShield Wizard

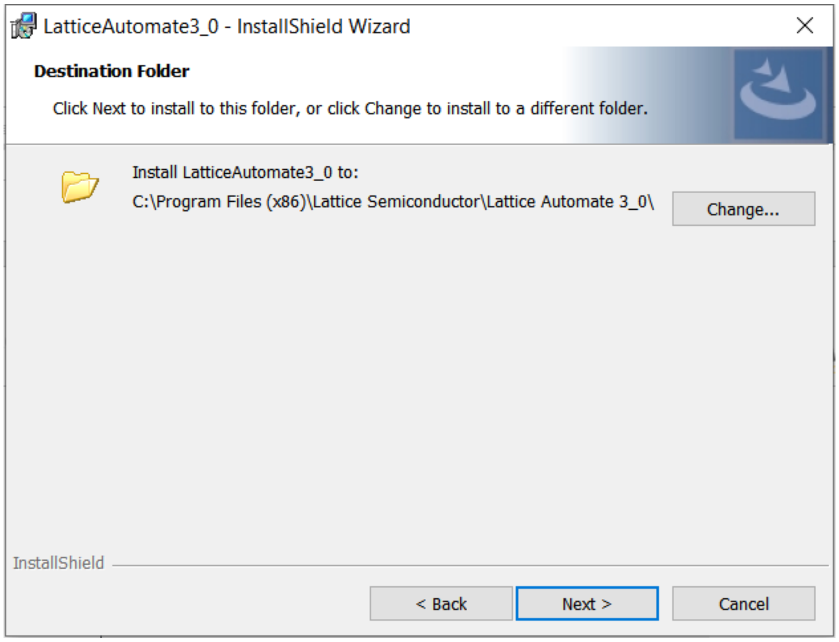
Welcome to the InstallShield Wizard for LatticeAutomate3_0

The InstallShield(R) Wizard will install LatticeAutomate3_0 on your computer. To continue, click Next.

WARNING: This program is protected by copyright law and international treaties.

Figure A.3 Automate 3.0 Next Step

5. Click **Next**.


LatticeAutomate3_0 - InstallShield Wizard

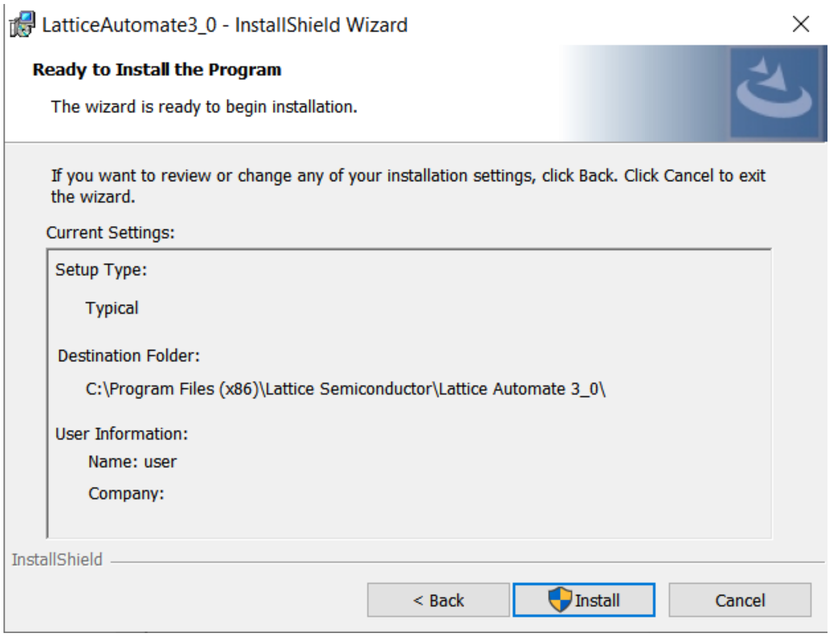
Destination Folder

Click Next to install to this folder, or click Change to install to a different folder.

Install LatticeAutomate3_0 to:
C:\Program Files (x86)\Lattice Semiconductor\Lattice Automate 3_0\

Figure A.4. Automate 3.0 Next Step

6. Click **Install**.



LatticeAutomate3_0 - InstallShield Wizard

Ready to Install the Program

The wizard is ready to begin installation.

If you want to review or change any of your installation settings, click Back. Click Cancel to exit the wizard.

Current Settings:

Setup Type:

Typical

Destination Folder:

C:\Program Files (x86)\Lattice Semiconductor\Lattice Automate 3_0\

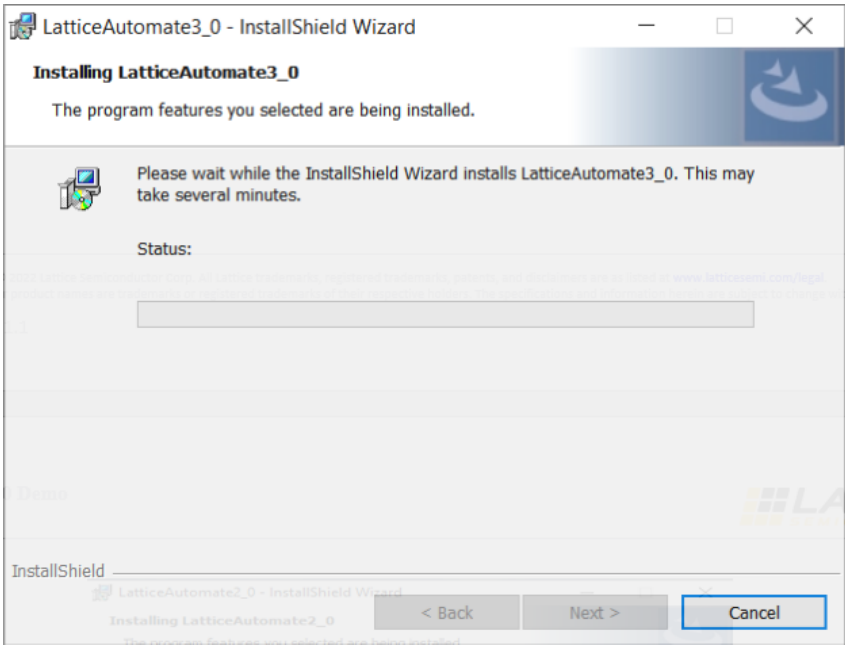
User Information:

Name: user

Company:

Figure A.5. Automate 3.0 Click Install

7. Wait for the installation to complete.



LatticeAutomate3_0 - InstallShield Wizard

Installing LatticeAutomate3_0

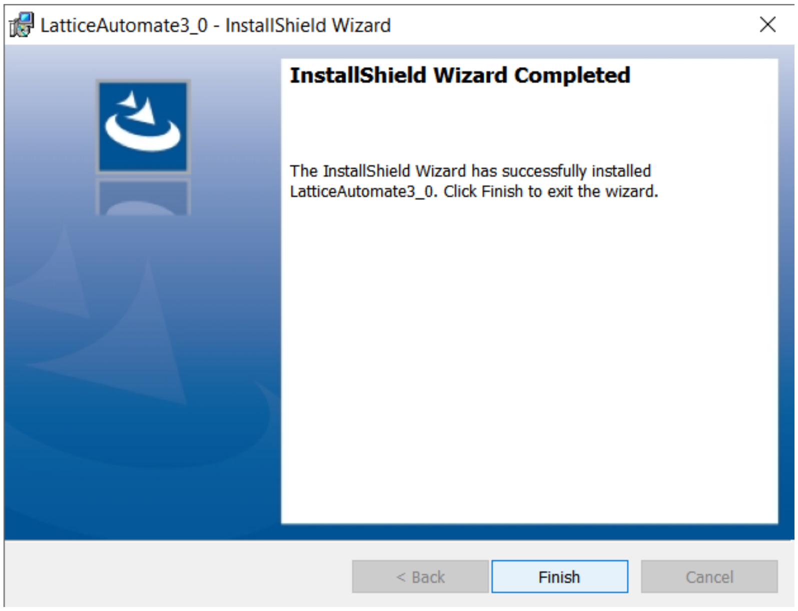
The program features you selected are being installed.

Please wait while the InstallShield Wizard installs LatticeAutomate3_0. This may take several minutes.

Status:

Figure A.6. Automate 3.0 Installation in Progress

8. Click **Finish**.



LatticeAutomate3_0 - InstallShield Wizard

InstallShield Wizard Completed

The InstallShield Wizard has successfully installed LatticeAutomate3_0. Click Finish to exit the wizard.

Figure A.7. Automate 3.0 Installation Finish

Appendix B. Programming the Automate Stack on Respective FLASH

B.1. Main System

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

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

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

Board Jumper Connections

Make sure that the following jumpers are connected on board:

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

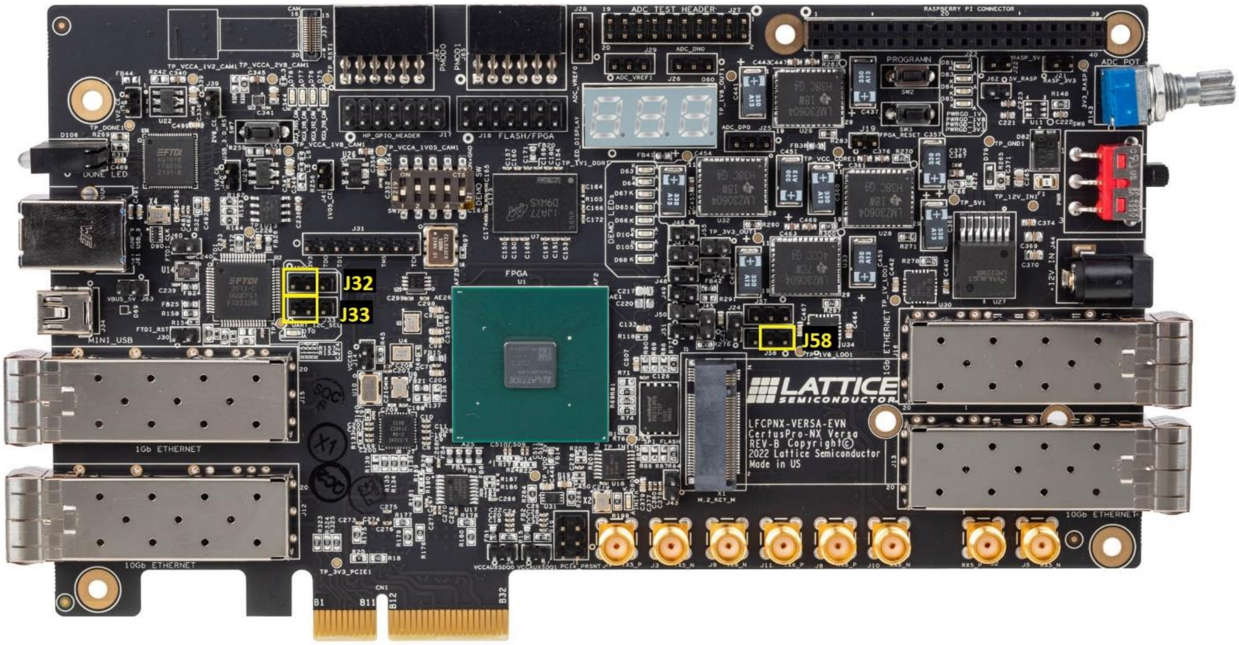
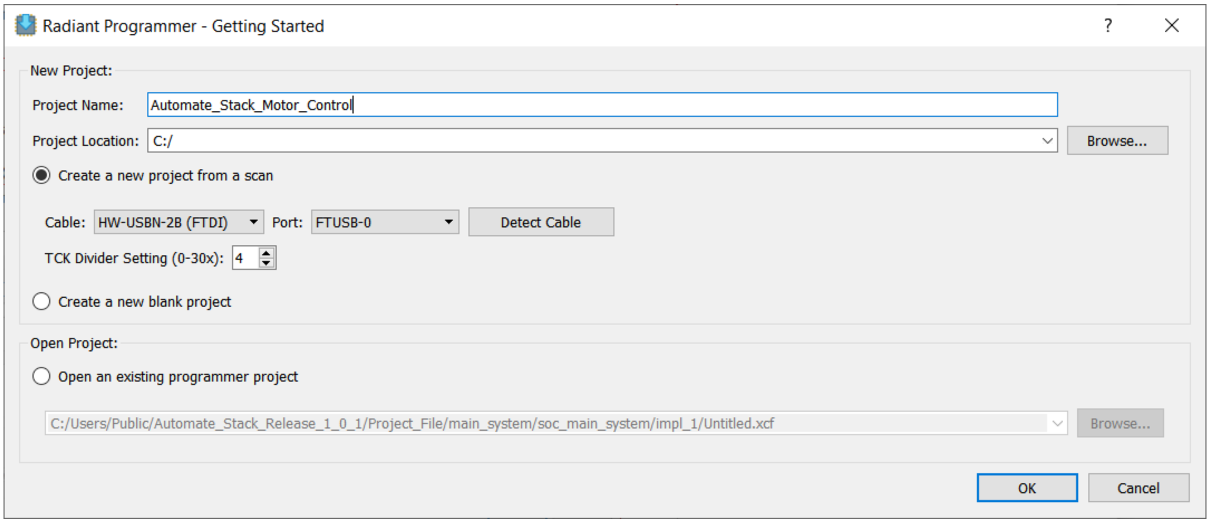


Figure B.1. Main System Jumper Connection

If the user programming the main system board for the first time, please refer to the Appendix C. Programming a Fresh Main System Board section and then come back to this section to follow further steps.

To program the SPI Flash in Radiant Programmer:

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



Radiant Programmer - Getting Started

New Project:

Project Name: Automate_Stack_Motor_Control

Project Location: C:/

● Create a new project from a scan

Cable: HW-USBN-2B (FTDI) Port: FTUSB-0

TCK Divider Setting (0-30x): 4

○ Create a new blank project

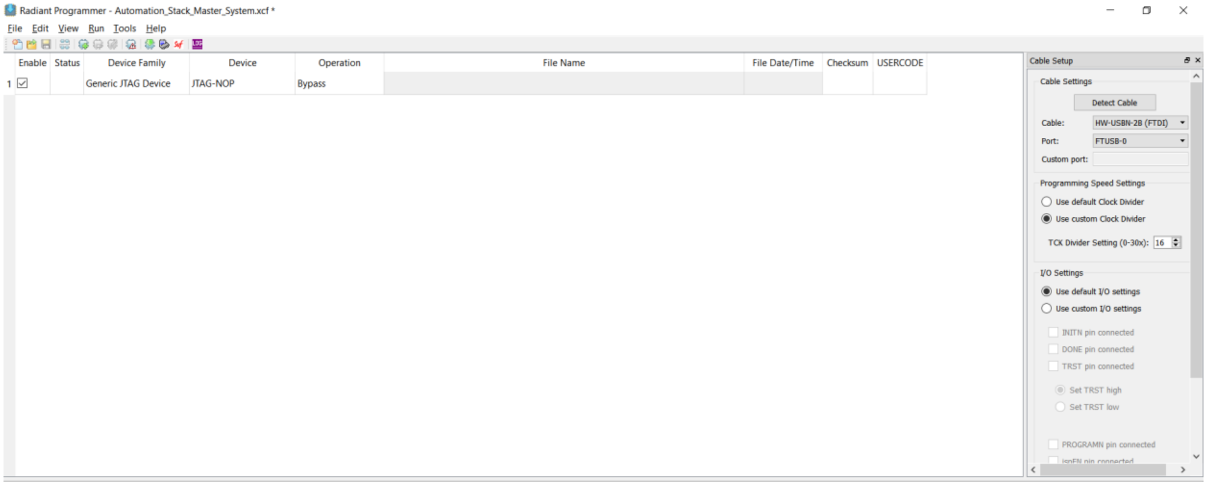
Open Project:

○ Open an existing programmer project

C:/Users/Public/Automate_Stack_Release_1_0_1/Project_File/main_system/soc_main_system/impl_1/Untitled.xcf

Figure B.2. Radiant Programmer - Default Screen (Main System)

3. Click **OK**.



Radiant Programmer - Automation_Stack_Master_System.xcf *

	Enable	Status	Device Family	Device	Operation	File Name	File Date/Time	Checksum	USERCODE
1	☑		Generic JTAG Device	JTAG-NOP	Bypass				

Cable Setup

Cable Settings

Cable: HW-USBN-2B (FTDI)

Port: FTUSB-0

Custom port:

Programming Speed Settings

○ Use default Clock Divider

● Use custom Clock Divider

TCK Divider Setting (0-30x): 16

I/O Settings

● Use default I/O settings

○ Use custom I/O settings

□ INITN pin connected

□ DONE pin connected

□ TRST pin connected

● Set TRST high

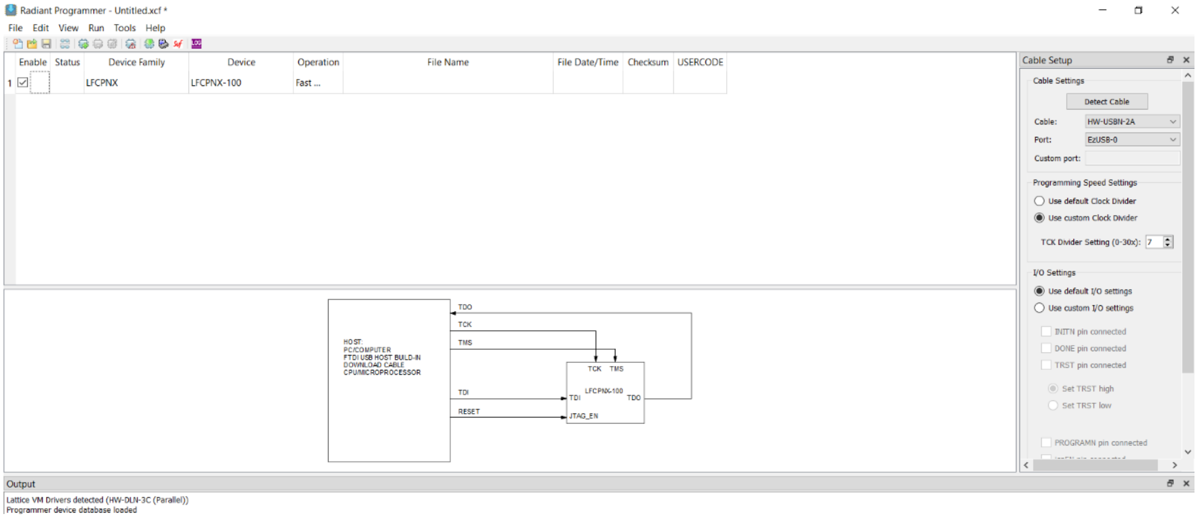
○ Set TRST low

□ PROGRAMN pin connected

□ ispEN pin connected

Figure B.3. Radiant Programmer - Initial Project Window (Main System)

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



Radiant Programmer - Untitled.xcf *

File Edit View Run Tools Help

	Enable	Status	Device Family	Device	Operation	File Name	File Date/Time	Checksum	USERCODE
1	☑		LFCPNX	LFCPNX-100	Fast ...				

Cable Setup

Cable Settings

Detect Cable

Cable: HW-USBN-2A

Port: EzUSB-0

Custom port:

Programming Speed Settings

○ Use default Clock Divider

● Use custom Clock Divider

TCK Divider Setting (0-30x): 7

I/O Settings

● Use default I/O settings

○ Use custom I/O settings

□ INITN pin connected

□ DONE pin connected

□ TRST pin connected

● Set TRST high

○ Set TRST low

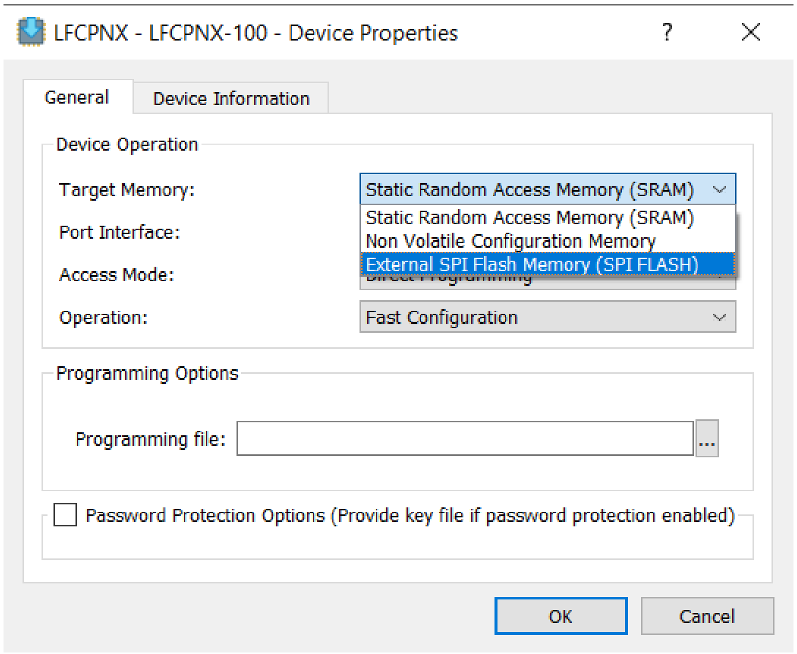
□ PROGRAMN pin connected

Output

Lattice VM Drivers detected (HW-DLN-3C (Parallel))
Programmer device database loaded

Figure B.4. Radiant Programmer - Device Selection (Main System)

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



LFCPNX - LFCPNX-100 - Device Properties

Device Operation

Target Memory: Static Random Access Memory (SRAM)

- Static Random Access Memory (SRAM)
- Non Volatile Configuration Memory
- External SPI Flash Memory (SPI FLASH)

Port Interface:

Access Mode: Direct Programming

Operation: Fast Configuration

Programming Options

Programming file:

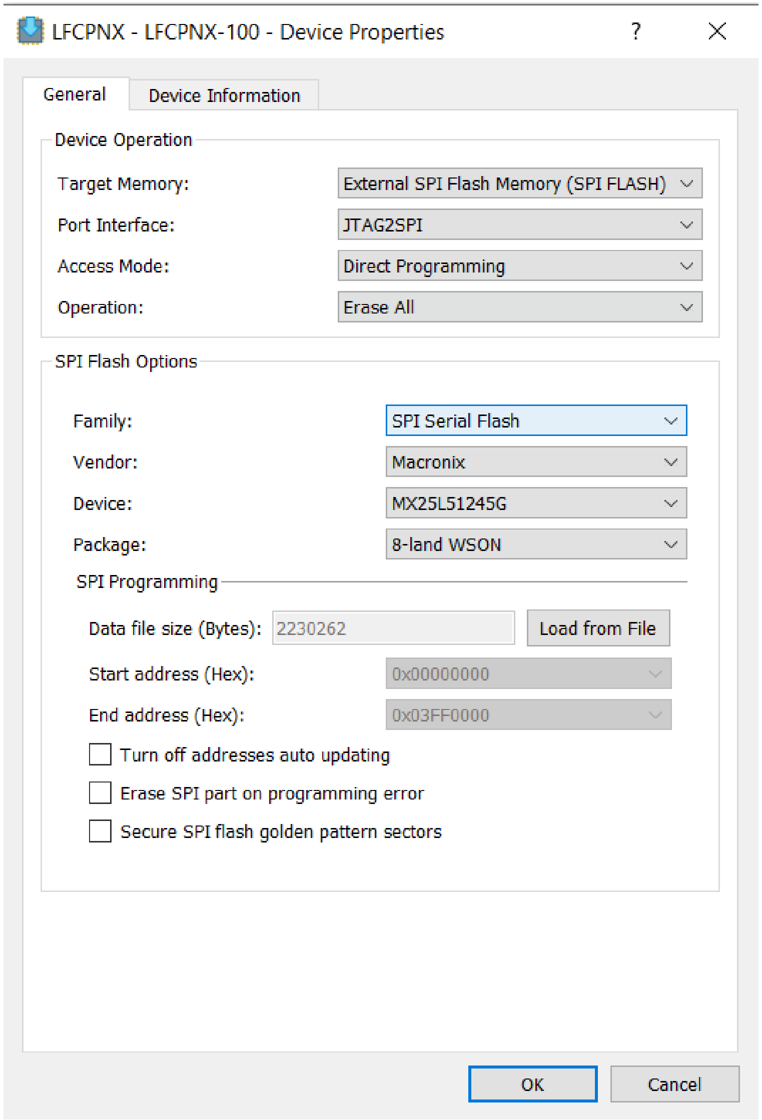
□ Password Protection Options (Provide key file if password protection enabled)

Figure B.5. Radiant Programmer - Device Operation (Main System)

6. Before programming, it is necessary to erase the flash memory. For this, 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: **Macronix**

- Device: **MX25L51245G**
- Package: **8-land WSON**



LFCPNX - LFCPNX-100 - Device Properties

Device Operation

Target Memory: External SPI Flash Memory (SPI FLASH)
Port Interface: JTAG2SPI
Access Mode: Direct Programming
Operation: Erase All

SPI Flash Options

Family: SPI Serial Flash
Vendor: Macronix
Device: MX25L51245G
Package: 8-land WSON

SPI Programming

Data file size (Bytes): 2230262
Start address (Hex): 0x00000000
End address (Hex): 0x03FF0000

☐ Turn off addresses auto updating
☐ Erase SPI part on programming error
☐ Secure SPI flash golden pattern sectors

Figure B.6. Radiant Programmer - Erase All (Main System)

7. 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.
8. After erasing the flash, power cycle the board and apply the settings below:

   a. Under Device Operation, select the options below:

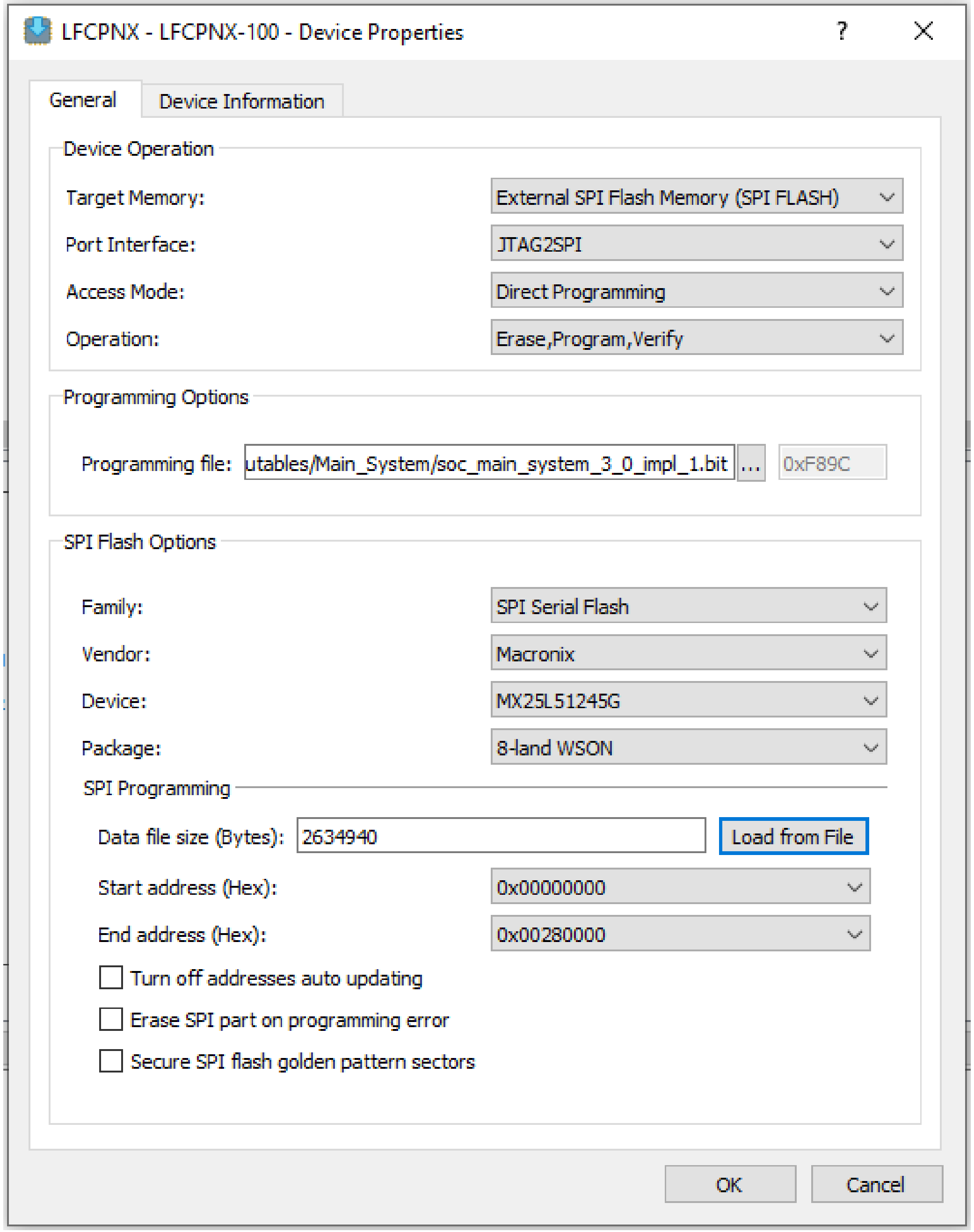
   - Target Memory: **External SPI Flash Memory (SPI FLASH)**
   - Port Interface: **JTAG2SPI**
   - Access Mode: **Direct Programming**
   - Operation: **Erase, Program, Verify**

   b. Under SPI Flash Options, select the options below:

   - Family: **SPI Serial Flash**
   - Vendor: **Macronix**

- Device: **MX25L51245G**
- Package: **8-land WSON**

9. To program the **bitstream file**, select the options as shown in Figure B.7.



LFCPNX - LFCPNX-100 - Device Properties

Device Operation

Target Memory: External SPI Flash Memory (SPI FLASH)

Port Interface: JTAG2SPI

Access Mode: Direct Programming

Operation: Erase,Program,Verify

Programming Options

Programming file: utables/Main_System/soc_main_system_3_0_impl_1.bit ... 0xF89C

SPI Flash Options

Family: SPI Serial Flash

Vendor: Macronix

Device: MX25L51245G

Package: 8-land WSON

SPI Programming

Data file size (Bytes): 2634940

Start address (Hex): 0x00000000

End address (Hex): 0x00280000

□ Turn off addresses auto updating

□ Erase SPI part on programming error

□ Secure SPI flash golden pattern sectors

Figure B.7. Radiant Programmer - Bitstream Flashing Settings (Main System)

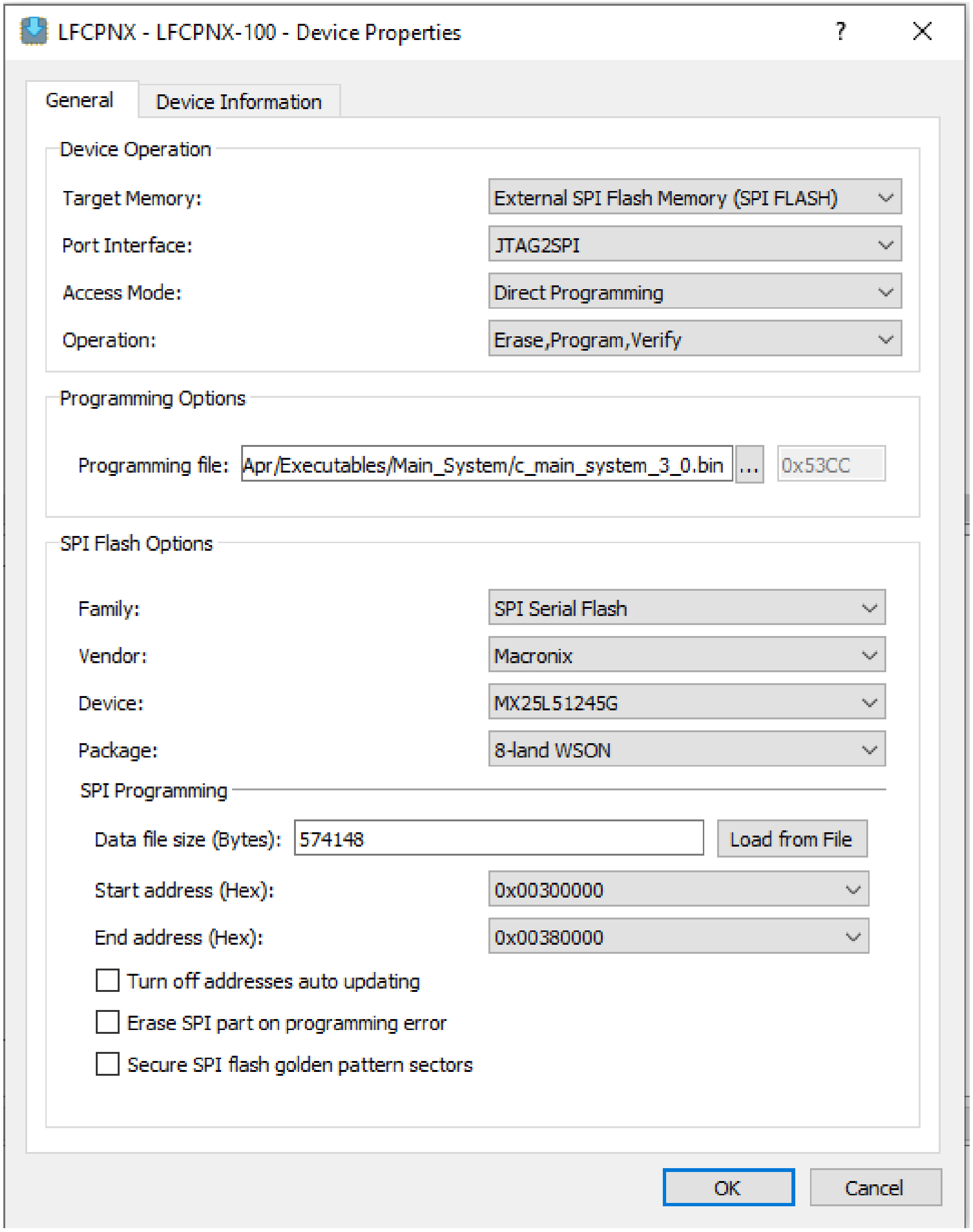
a. Under **Programming** Options, select the **soc_main_system_impl_1.bit bitstream** file in the Programming file.

b. Click **Load** from File to update the Data file size (Bytes) value.

c. Make sure that the following addresses are correct:

- Start Address (Hex): **0x00000000**
- End Address (Hex): **0x00280000**

10. Then click the Program Device icon or the menu item **Run ->Program Device**.
11. Now Power cycle the CertusPro NX Versa Board.
12. To program the firmware, select the options as shown in Figure B.8.
    a. Under Programming Options, select the c_main_system_3_0.bin binary file.
    b. Make sure that the following addresses are correct:
    - Start Address (Hex): **0x00300000**
    - End Address (Hex): **0x00380000**



LFCPNX - LFCPNX-100 - Device Properties

Device Operation

Target Memory: External SPI Flash Memory (SPI FLASH)

Port Interface: JTAG2SPI

Access Mode: Direct Programming

Operation: Erase,Program,Verify

Programming Options

Programming file: Apr/Executables/Main_System/c_main_system_3_0.bin ... 0x53CC

SPI Flash Options

Family: SPI Serial Flash

Vendor: Macronix

Device: MX25L51245G

Package: 8-land WSON

SPI Programming

Data file size (Bytes): 574148 Load from File

Start address (Hex): 0x00300000

End address (Hex): 0x00380000

□ Turn off addresses auto updating

□ Erase SPI part on programming error

□ Secure SPI flash golden pattern sectors

Figure B.8. Radiant Programmer - Binary Flashing Settings (Main System)

13. Click on the Program Device icon or the menu item **Run ->Program Device**.

B.2. Node System

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

- Bitstream
- Binary

Board Jumper Connections

Make sure that the following jumpers are connected on board:

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

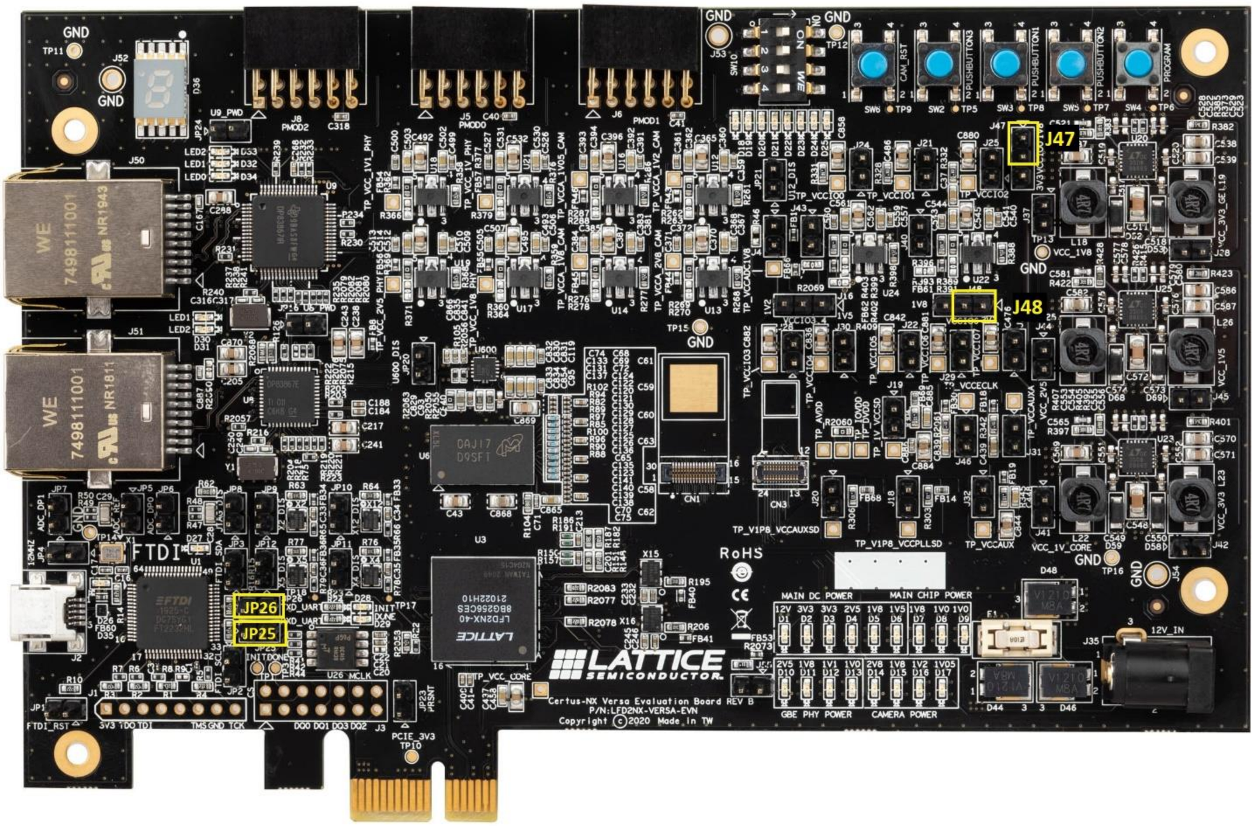
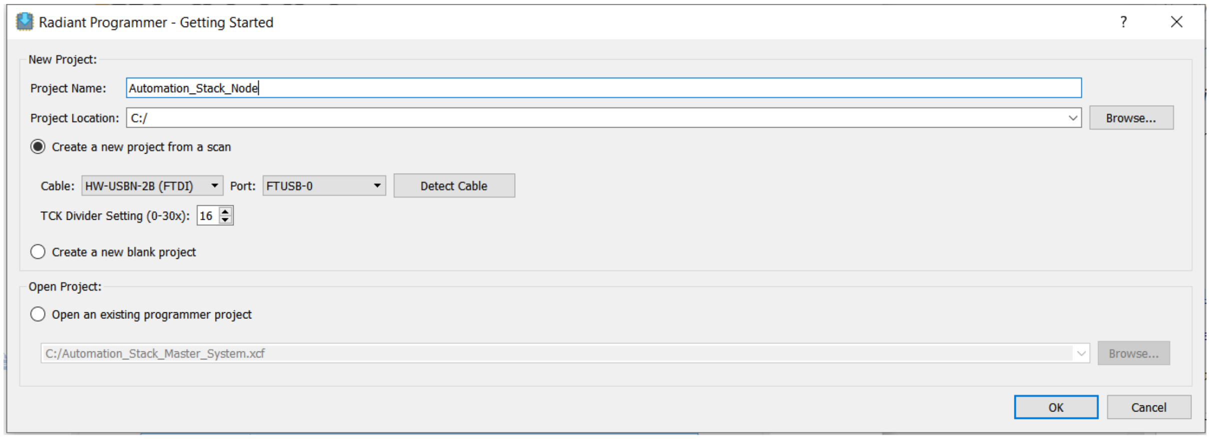


Figure B.9. Node System Jumper Connection

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**.



Project Name: Automation_Stack_Node

Project Location: C:/

● Create a new project from a scan

Cable: HW-USBN-2B (FTDI) Port: FTUSB-0

TCK Divider Setting (0-30x): 16

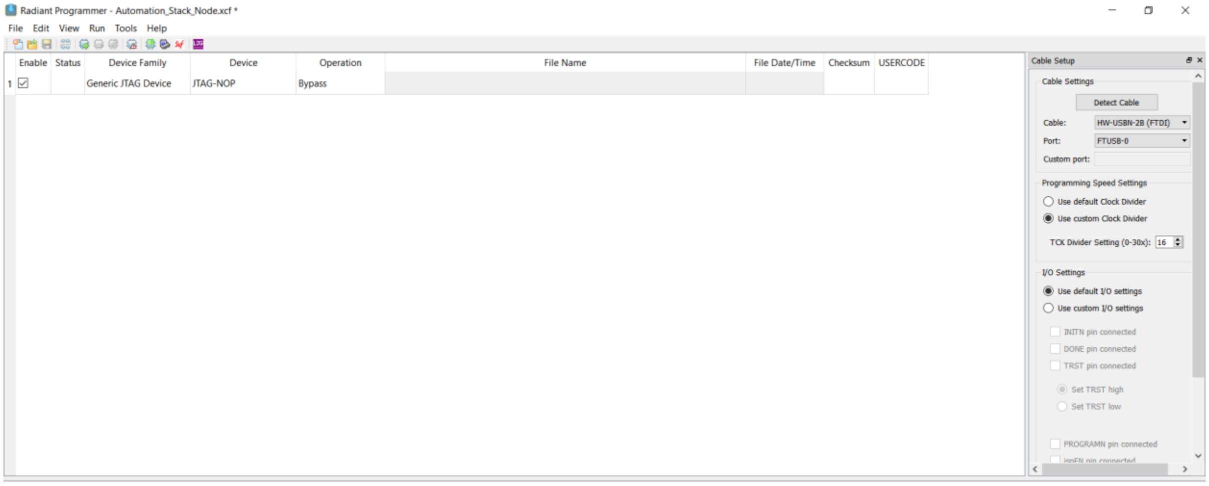
○ Create a new blank project

○ Open an existing programmer project

C:/Automation_Stack_Master_System.xcf

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

3. Click **OK**.


Radiant Programmer - Automation_Stack_Node.xcf *
File
Edit
View
Run
Tools
Help
Cable Setup
Cable Settings
Detect Cable
Programming Speed Settings
I/O Settings

	Enable	Status	Device Family	Device	Operation	File Name	File Date/Time	Checksum	USERCODE
1	☑		Generic JTAG Device	JTAG-NOP	Bypass				

Cable: HW-USBN-2B (FTDI)

Port: FTUSB-0

Custom port:

○ Use default Clock Divider

● Use custom Clock Divider

TCK Divider Setting (0-30x): 16

● Use default I/O settings

○ Use custom I/O settings

□ INITN pin connected

□ DONE pin connected

□ TRST pin connected

● Set TRST high

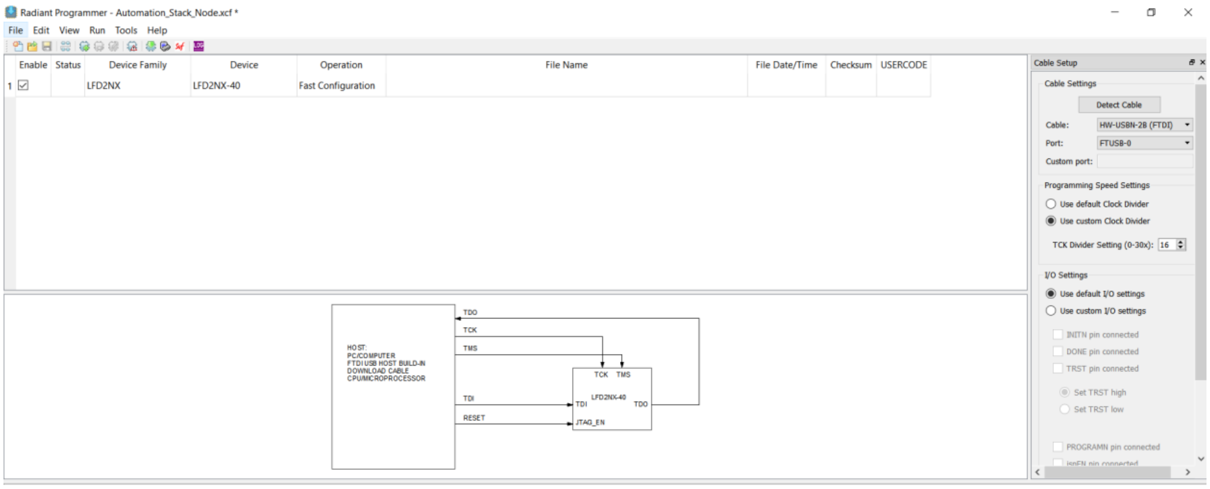
○ Set TRST low

□ PROGRAMN pin connected

□ ispEN pin connected

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.



	Enable	Status	Device Family	Device	Operation	File Name	File Date/Time	Checksum	USERCODE
1	☑		LFD2NX	LFD2NX-40	Fast Configuration				

Cable Settings

Cable: HW-USBN-2B (FTDI)

Port: FTUSB-0

Custom port:

Programming Speed Settings

○ Use default Clock Divider

● Use custom Clock Divider

TCK Divider Setting (0-30x): 16

I/O Settings

● Use default I/O settings

○ Use custom I/O settings

□ INITN pin connected

□ DONE pin connected

□ TRST pin connected

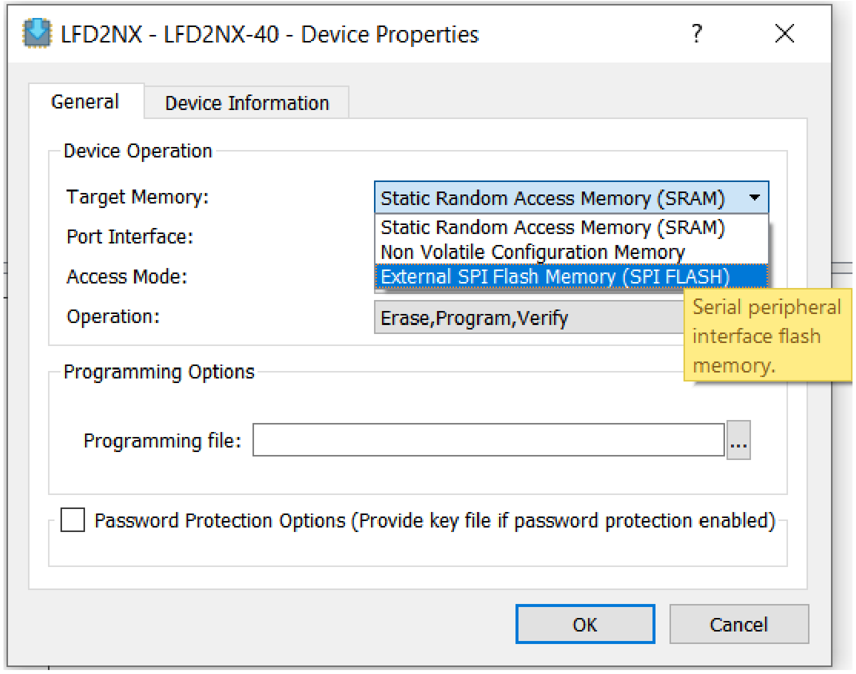
● Set TRST high

○ Set TRST low

□ PROGRAMN pin connected

□ IspEN pin connected

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



Device Operation

Target Memory: Static Random Access Memory (SRAM)

Static Random Access Memory (SRAM)
Non Volatile Configuration Memory
External SPI Flash Memory (SPI FLASH)

Port Interface:

Access Mode:

Operation: Erase,Program,Verify

Serial peripheral interface flash memory.

Programming Options

Programming file:

□ Password Protection Options (Provide key file if password protection enabled)

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

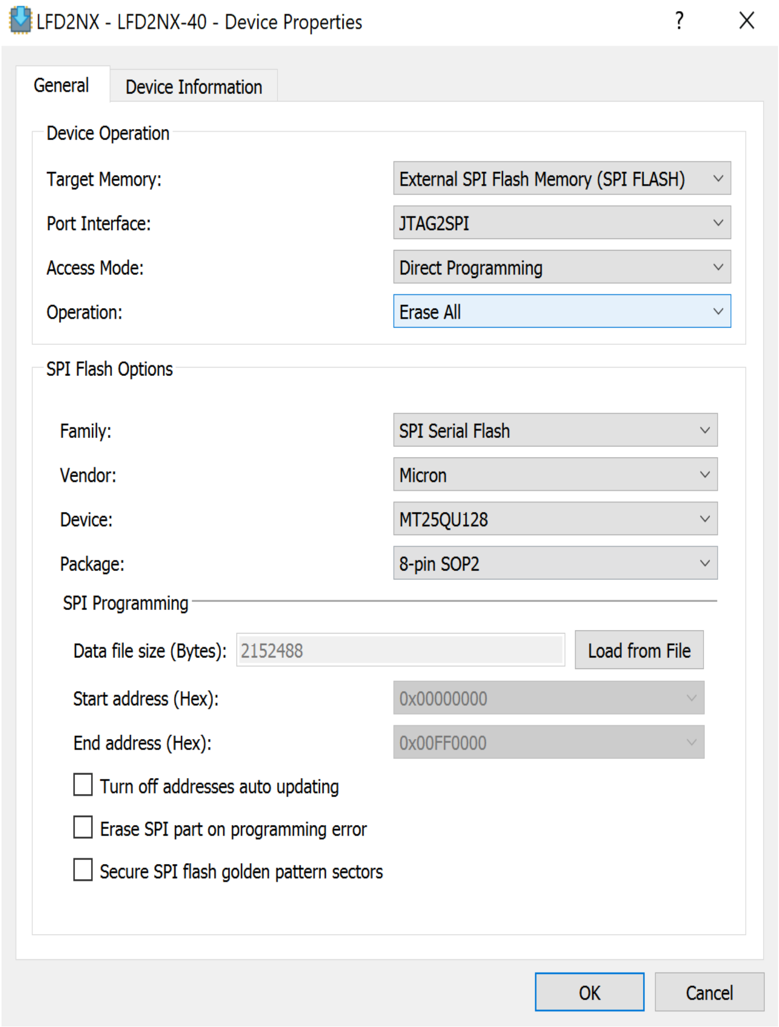
5. Before programming, it is necessary to erase the flash memory. For this, Apply the settings below:

   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**



LFD2NX - LFD2NX-40 - Device Properties

Device Operation

Target Memory: External SPI Flash Memory (SPI FLASH)

Port Interface: JTAG2SPI

Access Mode: Direct Programming

Operation: Erase All

SPI Flash Options

Family: SPI Serial Flash

Vendor: Micron

Device: MT25QU128

Package: 8-pin SOP2

SPI Programming

Data file size (Bytes): 2152488

Start address (Hex): 0x00000000

End address (Hex): 0x00FF0000

☐ Turn off addresses auto updating

☐ Erase SPI part on programming error

☐ Secure SPI flash golden pattern sectors

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:

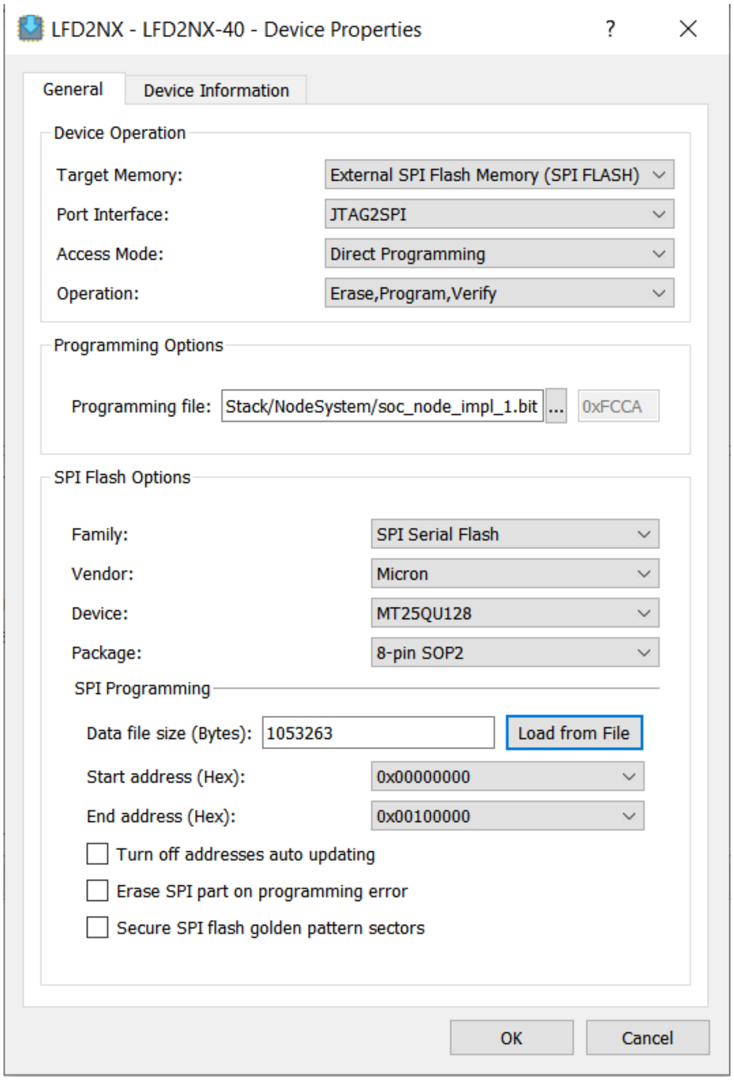
   a. Under Device Operation, select the options below:

   - 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.



LFD2NX - LFD2NX-40 - Device Properties

General | Device Information

Device Operation

- Target Memory: External SPI Flash Memory (SPI FLASH)
- Port Interface: JTAG2SPI
- Access Mode: Direct Programming
- Operation: Erase,Program,Verify

Programming Options

- Programming file: Stack/NodeSystem/soc_node_impl_1.bit ... 0xFCCA

SPI Flash Options

- Family: SPI Serial Flash
- Vendor: Micron
- Device: MT25QU128
- Package: 8-pin SOP2

SPI Programming

- Data file size (Bytes): 1053263 Load from File
- Start address (Hex): 0x00000000
- End address (Hex): 0x00100000
- □ Turn off addresses auto updating
- □ Erase SPI part on programming error
- □ Secure SPI flash golden pattern sectors

OK Cancel

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): **0x00100000**

9. Then click the Program Device icon or the menu item **Run ->Program Device**.

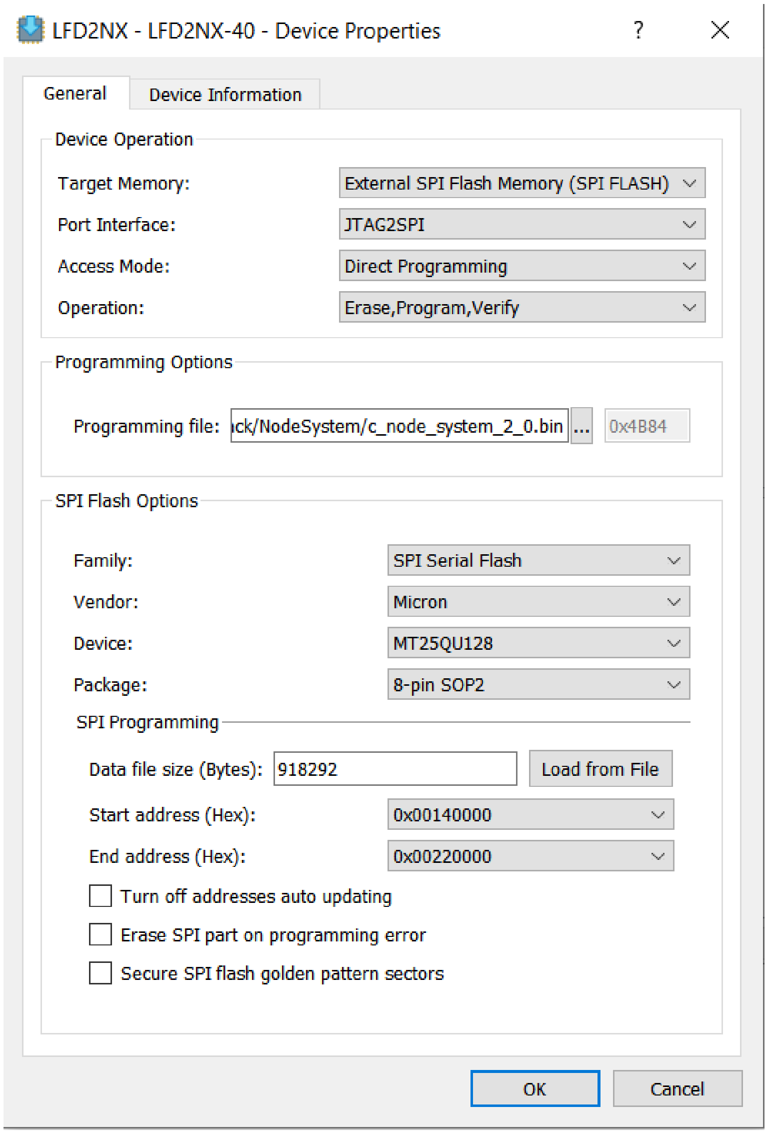
10. To program the firmware, select the options as shown in Figure B.16.

a. Under Programming Options, select the c_node_system_2_0.bin binary file.

b. Make sure that the following addresses are correct:

- Start Address (Hex): **0x00140000**
- End Address (Hex): **0x00220000**

11. Then click the Program Device icon or the menu item **Run ->Program Device**.



LFD2NX - LFD2NX-40 - Device Properties

General | Device Information

Device Operation

Target Memory: External SPI Flash Memory (SPI FLASH)

Port Interface: JTAG2SPI

Access Mode: Direct Programming

Operation: Erase,Program,Verify

Programming Options

Programming file: ıck/NodeSystem/c_node_system_2_0.bin ... 0x4B84

SPI Flash Options

Family: SPI Serial Flash

Vendor: Micron

Device: MT25QU128

Package: 8-pin SOP2

SPI Programming

Data file size (Bytes): 918292 Load from File

Start address (Hex): 0x00140000

End address (Hex): 0x00220000

□ Turn off addresses auto updating

□ Erase SPI part on programming error

□ Secure SPI flash golden pattern sectors

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

Note: After programming the Boards, power cycle the CertusPro NX Versa board as shown in Figure B.17 and each Certus NX Versa board as shown in Figure B.18, and then press the system Reset button SW 3.

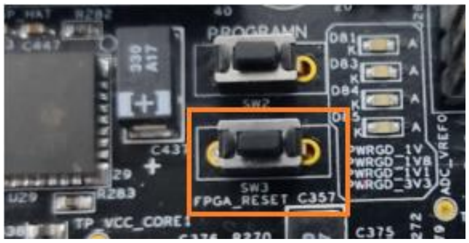


Figure B.17. Reset Button SW3 of CertusPro NX Versa Board

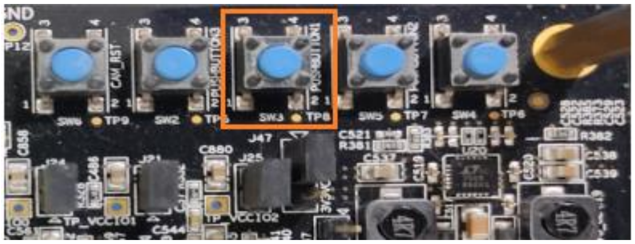


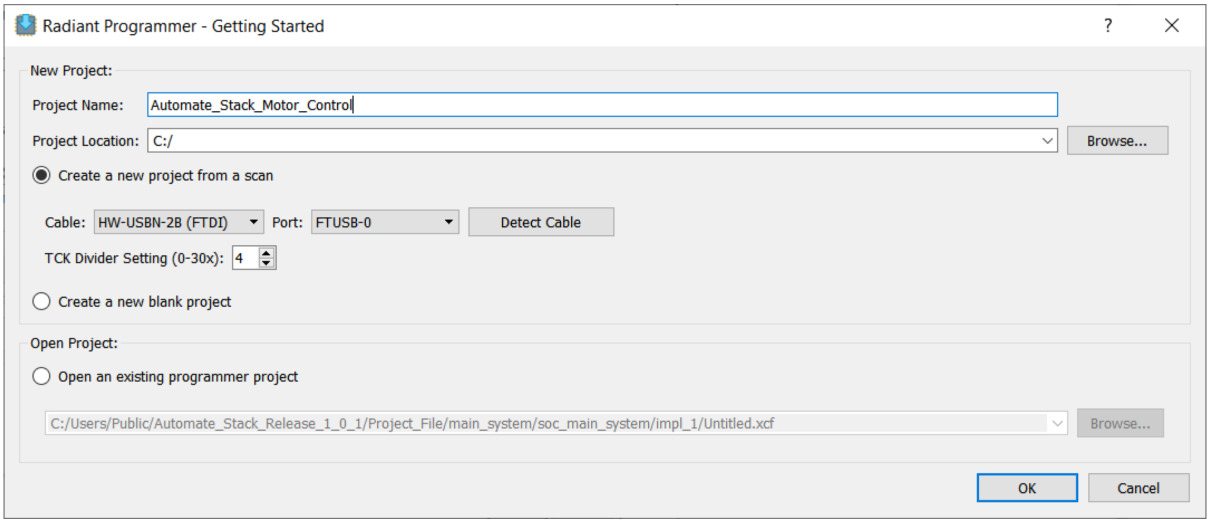
Figure B.18. Button SW3 of Certus NX Versa Board

Appendix C. Programming a Fresh Main System Board

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

To program the SPI Flash in Quad mode:

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



Radiant Programmer - Getting Started

New Project:

Project Name: Automate_Stack_Motor_Control

Project Location: C:/

● Create a new project from a scan

Cable: HW-USBN-2B (FTDI) Port: FTUSB-0 Detect Cable

TCK Divider Setting (0-30x): 4

○ Create a new blank project

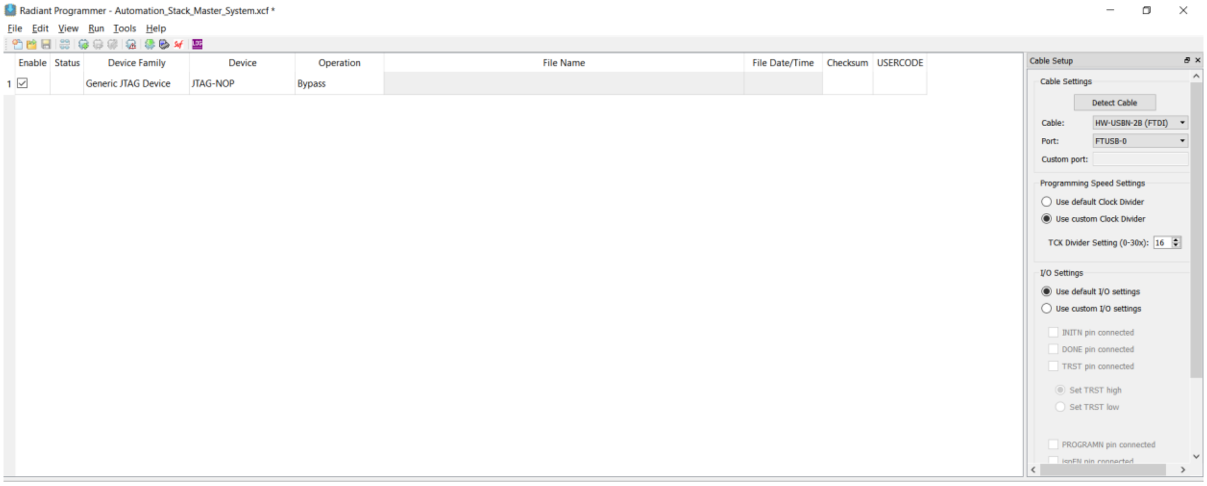
Open Project:

○ Open an existing programmer project

C:/Users/Public/Automate_Stack_Release_1_0_1/Project_File/main_system/soc_main_system/impl_1/Untitled.xcf

Figure C.1. Radiant Programmer - Default Screen

3. Click **OK**.



Radiant Programmer - Automation_Stack_Master_System.xcf *

	Enable	Status	Device Family	Device	Operation	File Name	File Date/Time	Checksum	USERCODE
1	☑		Generic JTAG Device	JTAG-NOP	Bypass				

Cable Setup

Cable Settings

Cable: HW-USBN-2B (FTDI)

Port: FTUSB-0

Custom port:

Programming Speed Settings

○ Use default Clock Divider

● Use custom Clock Divider

TCK Divider Setting (0-30x): 16

I/O Settings

● Use default I/O settings

○ Use custom I/O settings

□ INITN pin connected

□ DONE pin connected

□ TRST pin connected

● Set TRST high

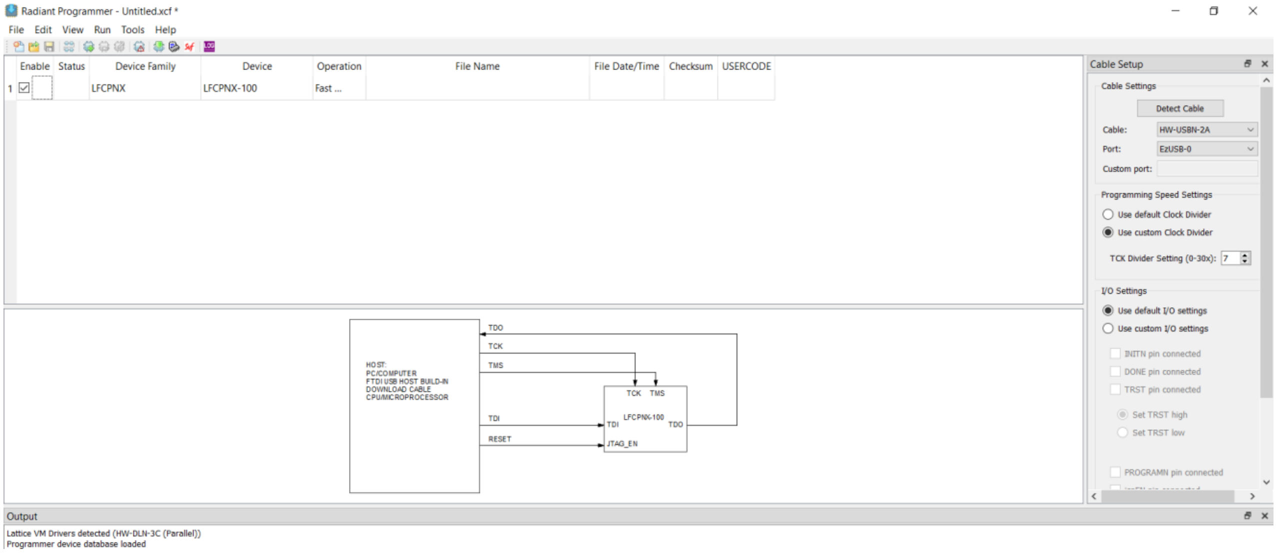
○ Set TRST low

□ PROGRAMN pin connected

□ ispEN pin connected

Figure C.2. Radiant Programmer - Initial Project Window

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



Radiant Programmer - Untitled.xcf *

	Enable	Status	Device Family	Device	Operation	File Name	File Date/Time	Checksum	USERCODE
1	☑		LFCPNX	LFCPNX-100	Fast ...				

Cable Setup

Cable Settings

Cable: HW-USBN-2A

Port: EzUSB-0

Custom port:

Programming Speed Settings

○ Use default Clock Divider

● Use custom Clock Divider

TCK Divider Setting (0-30x): 7

I/O Settings

● Use default I/O settings

○ Use custom I/O settings

□ INITN pin connected

□ DONE pin connected

□ TRST pin connected

○ Set TRST high

○ Set TRST low

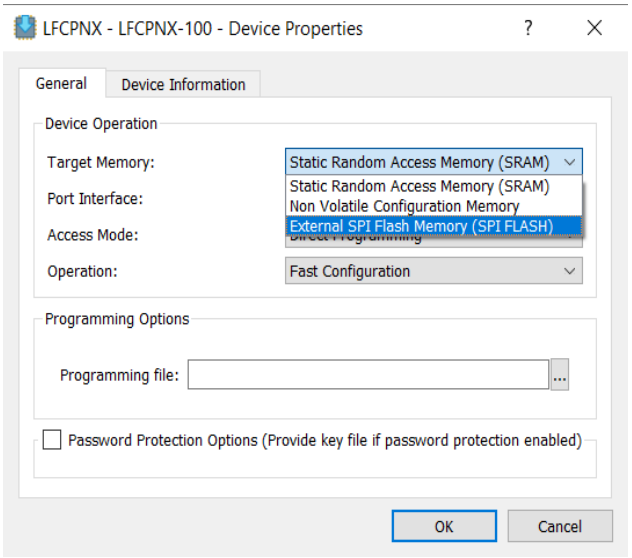
□ PROGRAMN pin connected

Output

```
Lattice VM Drivers detected (HW-DLN-3C (Parallel))
Programmer device database loaded
```

Figure C.3. Radiant Programmer - Device Selection

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



LFCPNX - LFCPNX-100 - Device Properties

Device Operation

Target Memory: Static Random Access Memory (SRAM)

Static Random Access Memory (SRAM)
Non Volatile Configuration Memory
External SPI Flash Memory (SPI FLASH)

Port Interface:

Access Mode: Direct Programming

Operation: Fast Configuration

Programming Options

Programming file:

□ Password Protection Options (Provide key file if password protection enabled)

Figure C.4. Radiant Programmer - Device Operation

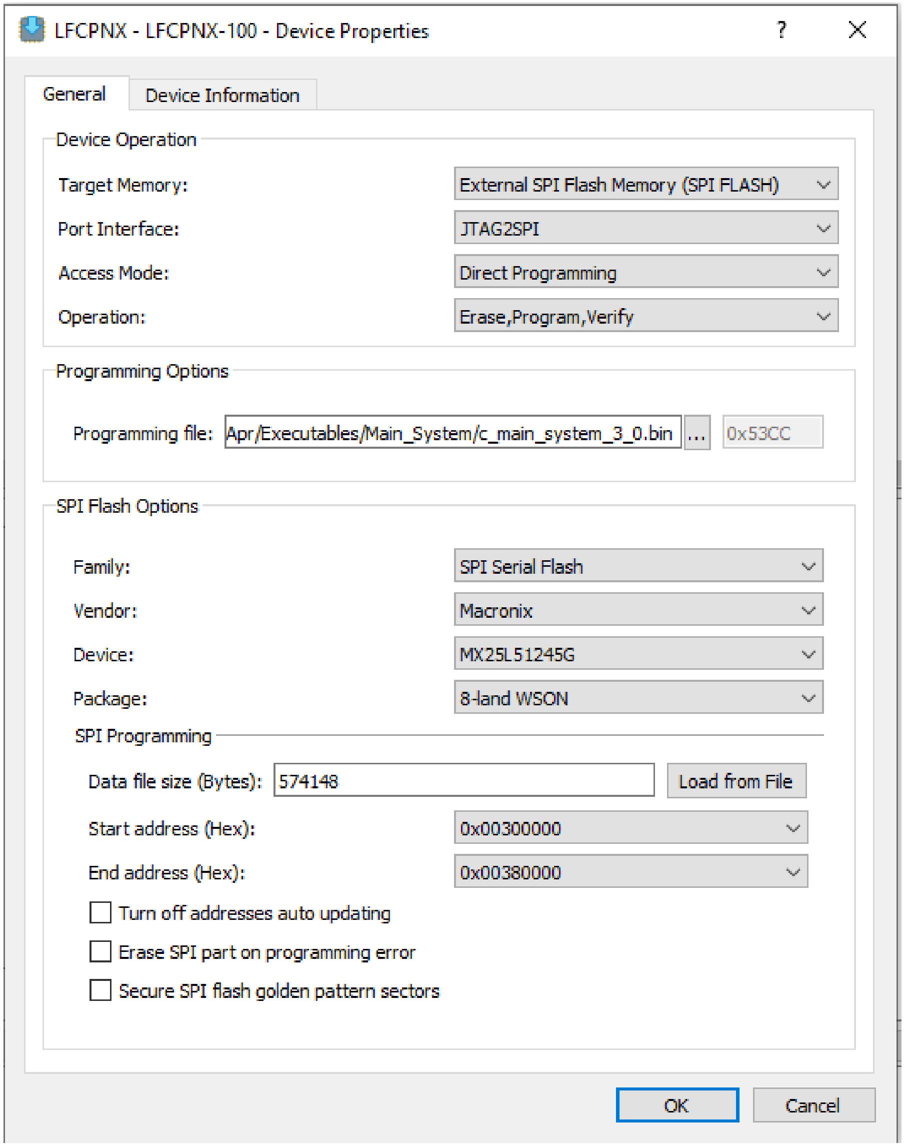
6. To program the binary file, apply the settings below:

   a. Under Device Operation, select the options below:

   - Target Memory: **External SPI Flash Memory (SPI FLASH)**
   - Port Interface: **JTAG2SPI**
   - Access Mode: **Direct Programming**
   - Operation: **Erase, Program, Verify Quad 1**

   b. Under SPI Flash Options, select the options below:

   - Family: **SPI Serial Flash**
   - Vendor: **Macronix**
   - Device: **MX25L51245G**
   - Package: **8-land WSON**



LFCPNX - LFCPNX-100 - Device Properties

General | Device Information

Device Operation

Target Memory: External SPI Flash Memory (SPI FLASH)

Port Interface: JTAG2SPI

Access Mode: Direct Programming

Operation: Erase,Program,Verify

Programming Options

Programming file: Apr/Executables/Main_System/c_main_system_3_0.bin ... 0x53CC

SPI Flash Options

Family: SPI Serial Flash

Vendor: Macronix

Device: MX25L51245G

Package: 8-land WSON

SPI Programming

Data file size (Bytes): 574148 Load from File

Start address (Hex): 0x00300000

End address (Hex): 0x00380000

□ Turn off addresses auto updating

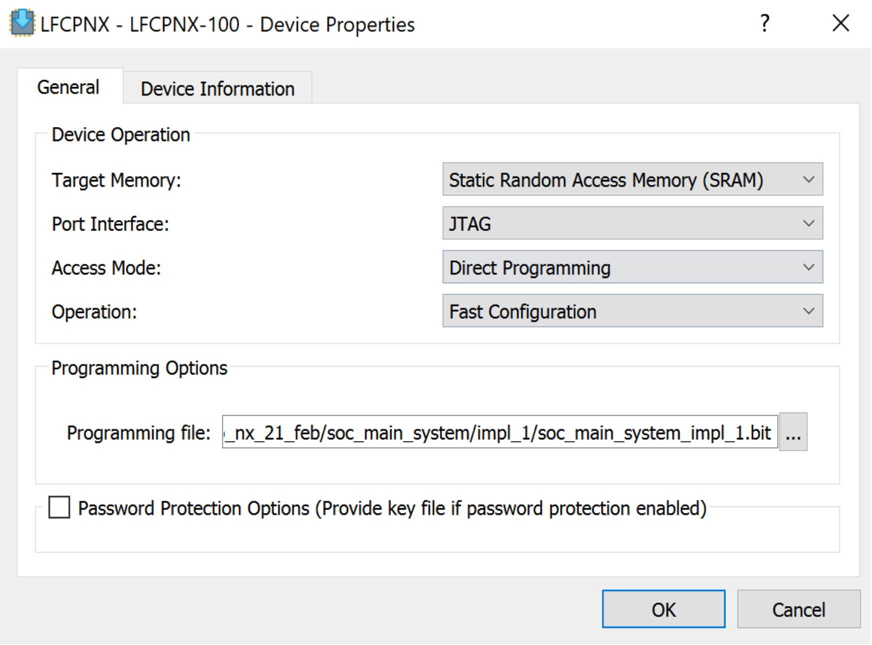
□ Erase SPI part on programming error

□ Secure SPI flash golden pattern sectors

OK Cancel

Figure C.5. Radiant Programmer - Quad Mode Programming

7. Under Programming Options, select the c_main_system_3_0.bin binary file.
   a. Make sure that the following addresses are correct:
   - Start Address (Hex): **0x00300000**
   - End Address (Hex): **0x00380000**
8. Click **OK** and then click the Program Device icon or the menu item **Run ->Program Device**. After that power cycle the board.
9. To program the bitstream file, select the options as shown in Figure C.5.
10. Under Programming Options, select the soc_main_system_impl_1.bit bitstream file in the Programming file.


LFCPNX - LFCPNX-100 - Device Properties

General | Device Information

Device Operation

Target Memory: Static Random Access Memory (SRAM)

Port Interface: JTAG

Access Mode: Direct Programming

Operation: Fast Configuration

Programming Options

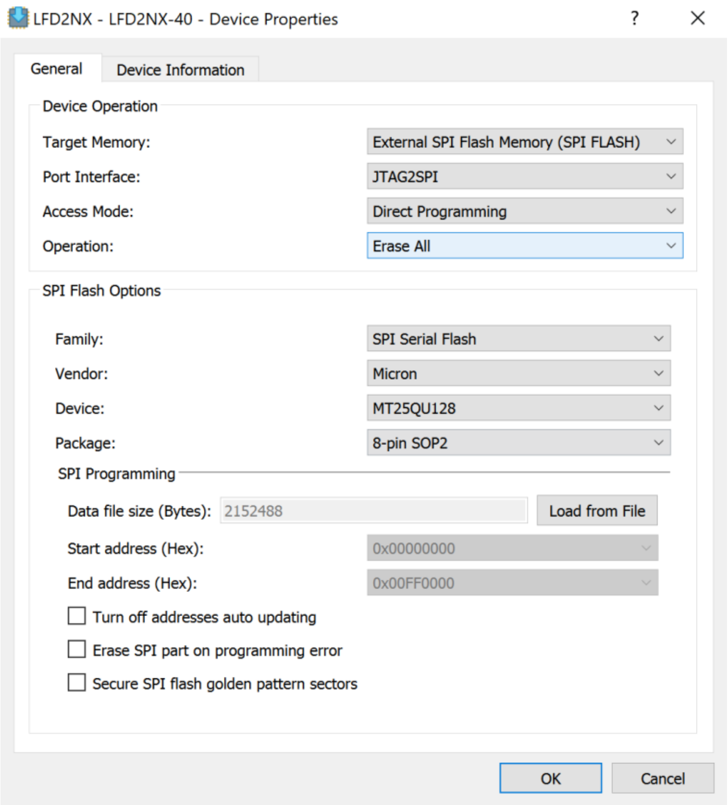
Programming file: _nx_21_feb/soc_main_system/impl_1/soc_main_system_impl_1.bit

□ Password Protection Options (Provide key file if password protection enabled)

Figure C.6. Radiant Programmer - Device Properties

11. Click the Program Device icon or use the menu tab **Run -> Program Device**.
12. Now the SPI flash must be erased. For that, 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: **Macronix**
    - Device: **MX25L51245G**
    - Package: **8-land WSON**



LFD2NX - LFD2NX-40 - Device Properties

Device Operation

Target Memory: External SPI Flash Memory (SPI FLASH)

Port Interface: JTAG2SPI

Access Mode: Direct Programming

Operation: Erase All

SPI Flash Options

Family: SPI Serial Flash

Vendor: Micron

Device: MT25QU128

Package: 8-pin SOP2

SPI Programming

Data file size (Bytes): 2152488

Start address (Hex): 0x00000000

End address (Hex): 0x00FF0000

□ Turn off addresses auto updating

□ Erase SPI part on programming error

□ Secure SPI flash golden pattern sectors

Figure C.7. Radiant Programmer - Erase All

13. Click the Program Device icon or the menu item **Run -> Program Device**.
14. Power cycle the CertusPro NX Versa Board.

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

Appendix D. Ethernet Connection

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

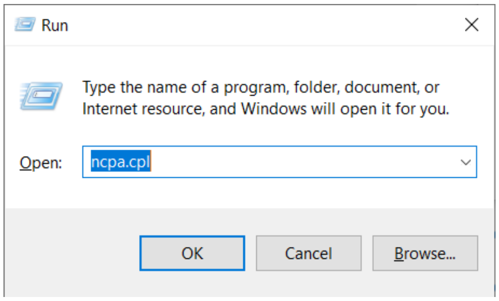
1. Set up the following hardware connections, as shown in Figure 4.1:
   a. Connect the Ethernet cable between the CertusPro-NX board and the Laptop/PC in the Ethernet port of the laptop and main system.
   b. Connect the 12 V power adapter to the main system board.
2. Connect the UART cable to the main board and laptop or Docklight debugging.

Note: The firewall should be disabled while doing this activity.

Note: .bit and .bin should be programmed in the main system and node systems. Refer to Programming the Automate Stack on Respective FLASH and Programming a Fresh Main System Board sections.

D.2. Ethernet Configuration Setting

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



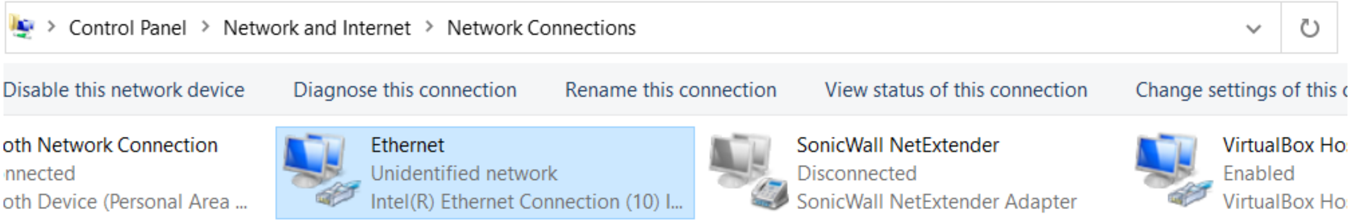
Run

Type the name of a program, folder, document, or Internet resource, and Windows will open it for you.

Open: ncpa.cpl

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

2. Click **Ethernet**.



Control Panel > Network and Internet > Network Connections
Disable this network device
Diagnose this connection
Rename this connection
View status of this connection
Change settings of this c

oth Network Connection
nnected
oth Device (Personal Area ...

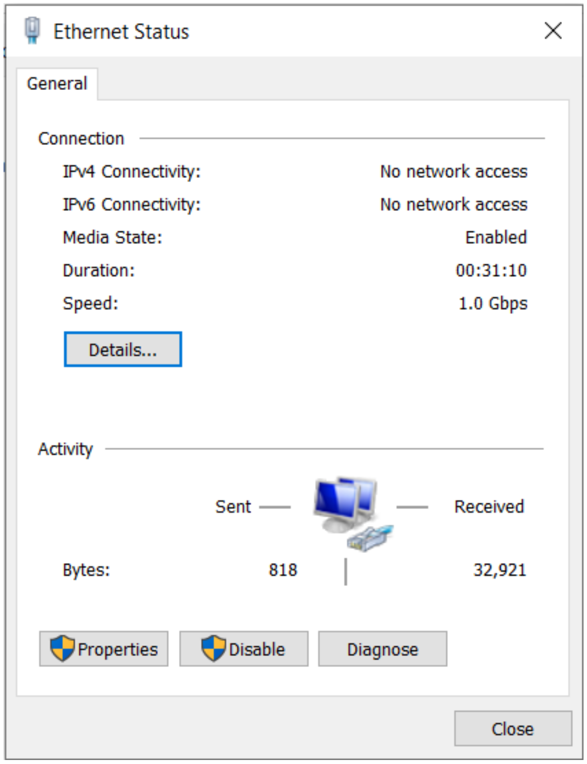
Ethernet
Unidentified network
Intel(R) Ethernet Connection (10) I...

SonicWall NetExtender
Disconnected
SonicWall NetExtender Adapter

VirtualBox Ho
Enabled
VirtualBox Ho

Figure D.2. Ethernet Settings

3. Click **Properties**.



Connection

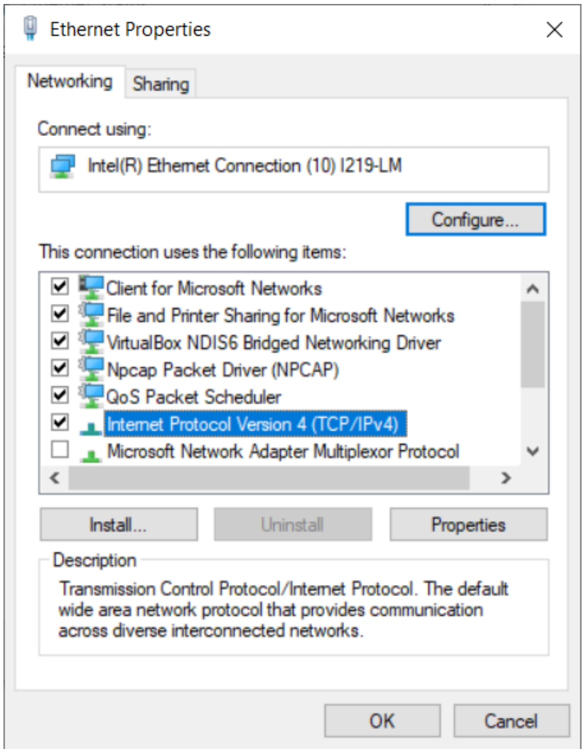
IPv4 Connectivity:	No network access
IPv6 Connectivity:	No network access
Media State:	Enabled
Duration:	00:31:10
Speed:	1.0 Gbps

Activity

	Sent	Received
Bytes:	818	32,921

Figure D.3. Click on Properties

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



Connect using:

Intel(R) Ethernet Connection (10) I219-LM

This connection uses the following items:

- ☑ Client for Microsoft Networks
- ☑ File and Printer Sharing for Microsoft Networks
- ☑ VirtualBox NDIS6 Bridged Networking Driver
- ☑ Npcap Packet Driver (NPCAP)
- ☑ QoS Packet Scheduler
- ☑ Internet Protocol Version 4 (TCP/IPv4)
- □ Microsoft Network Adapter Multiplexor Protocol

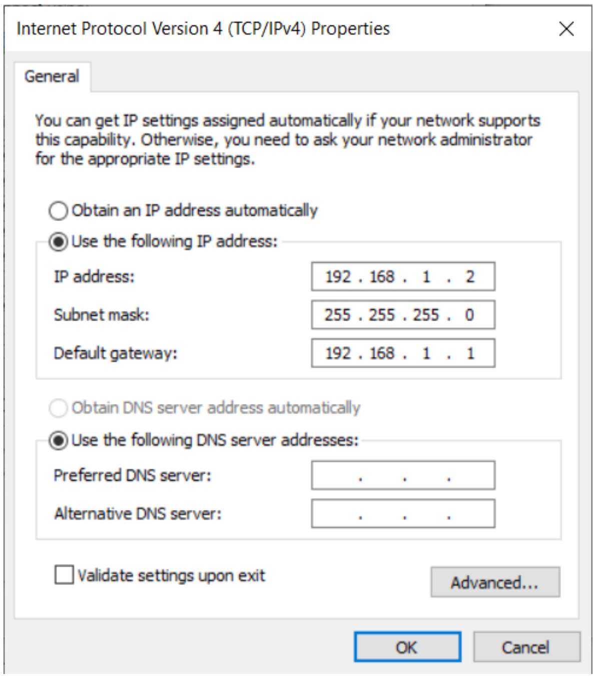
Description

Transmission Control Protocol/Internet Protocol. The default wide area network protocol that provides communication across diverse interconnected networks.

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 and then click **OK**.

Note: If users changed the IP address of the PC using Propel Builder, make sure that enter the changed IP address in TCP/IPv4 settings also. Please see the Reference Design Document Section 2.6 for more information about how to change the IP addresses of the PC.



Internet Protocol Version 4 (TCP/IPv4) Properties

General

You can get IP settings assigned automatically if your network supports this capability. Otherwise, you need to ask your network administrator for the appropriate IP settings.

○ Obtain an IP address automatically
● Use the following IP address:

IP address:	192 . 168 . 1 . 2
Subnet mask:	255 . 255 . 255 . 0
Default gateway:	192 . 168 . 1 . 1

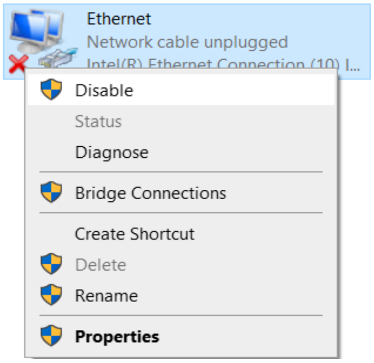
○ Obtain DNS server address automatically
● Use the following DNS server addresses:

Preferred DNS server:	. . .
Alternative DNS server:	. . .

□ Validate settings upon exit

Figure D.5. Configure the Settings

6. Check if Ethernet says “Network cable unplugged”.
7. Right-click on **Ethernet** and, then click on **Disable**.



Ethernet
Network cable unplugged
Intel(R) Ethernet Connection (10) I...

Figure D.6. Disable the Setting

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

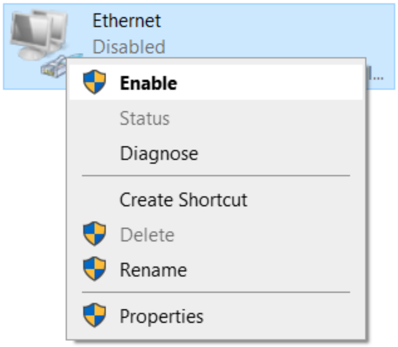
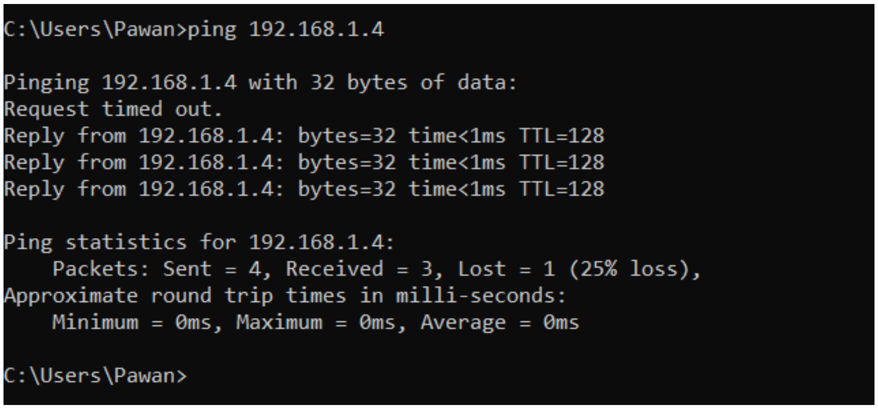


Figure D.7. Enable the Setting

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. Check the Connection with a Ping

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



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

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

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

C:\Users\Pawan>
```

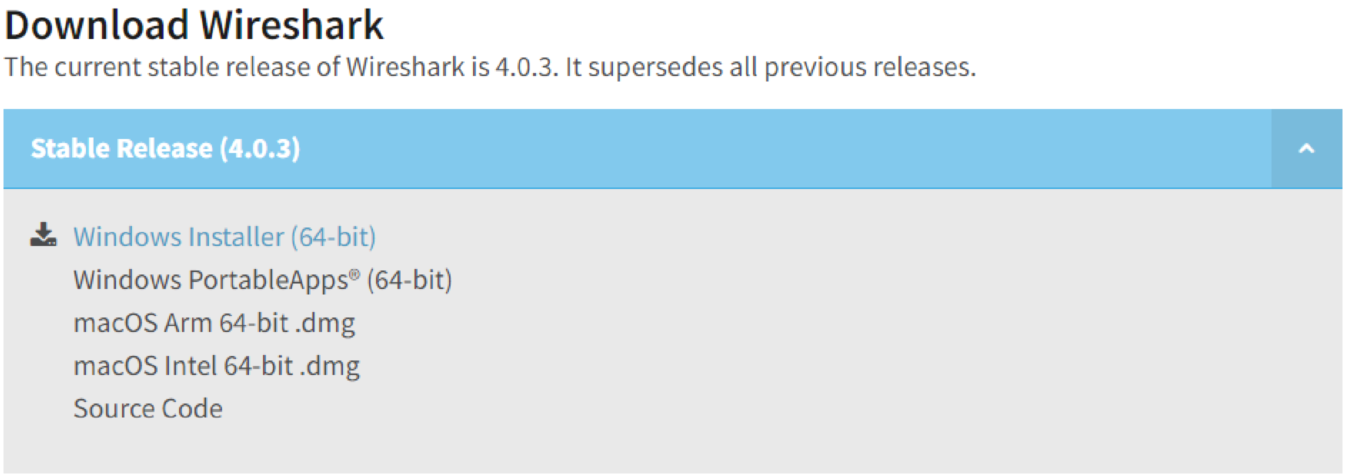
Figure D.8. Ping on cmd Terminal

D.4. UDP Packet Transaction

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

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

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



Download Wireshark

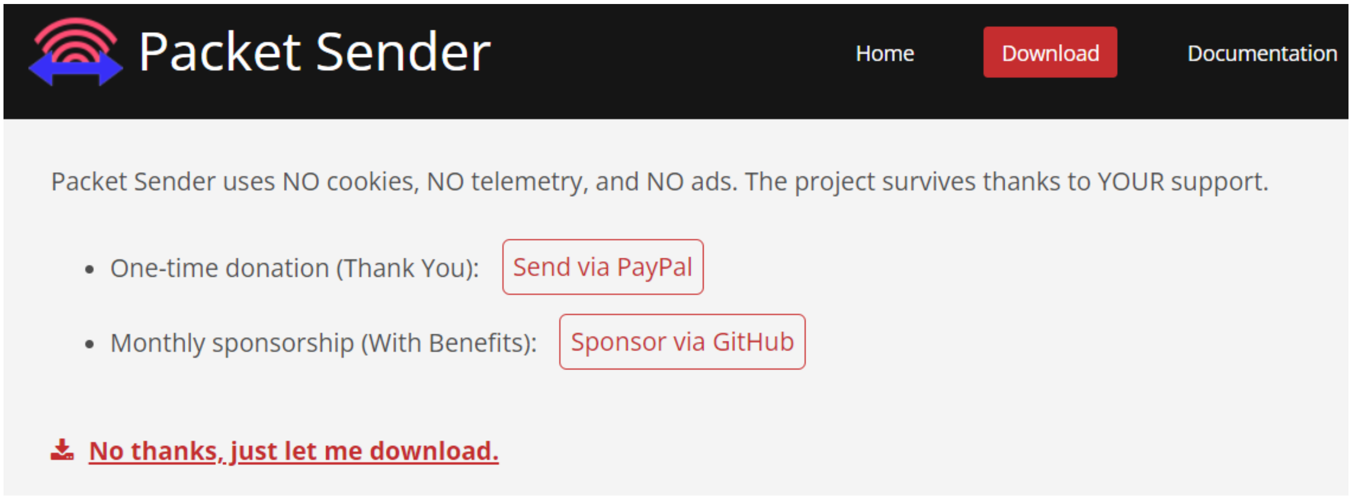
The current stable release of Wireshark is 4.0.3. It supersedes all previous releases.

Stable Release (4.0.3)

- Windows Installer (64-bit)
- Windows PortableApps® (64-bit)
- macOS Arm 64-bit .dmg
- macOS Intel 64-bit .dmg
- Source Code

Figure D.9. Downloadable Link of Wireshark

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



Packet Sender uses NO cookies, NO telemetry, and NO ads. The project survives thanks to YOUR support.

- One-time donation (Thank You): Send via PayPal
- Monthly sponsorship (With Benefits): Sponsor via GitHub

No thanks, just let me download.

Figure D.10. Downloadable Link of Packet Sender

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

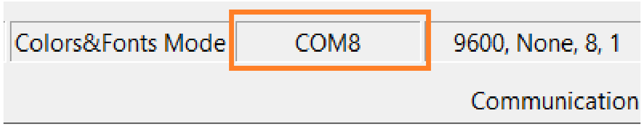
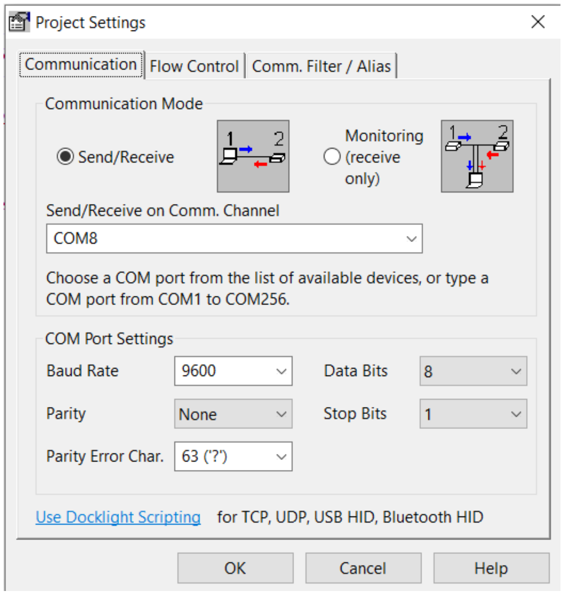


Figure D.11. Docklight - Com Selection

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



Project Settings

Communication Mode

● Send/Receive

○ Monitoring (receive only)

Send/Receive on Comm. Channel: COM8

Choose a COM port from the list of available devices, or type a COM port from COM1 to COM256.

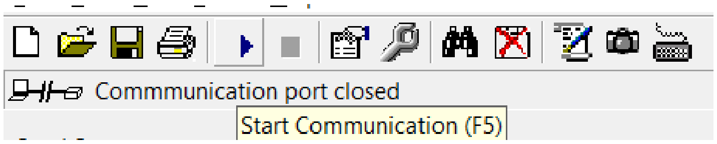
COM Port Settings

Baud Rate	9600	Data Bits	8
Parity	None	Stop Bits	1
Parity Error Char.	63 ('?')		

Use Docklight Scripting for TCP, UDP, USB HID, Bluetooth HID

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

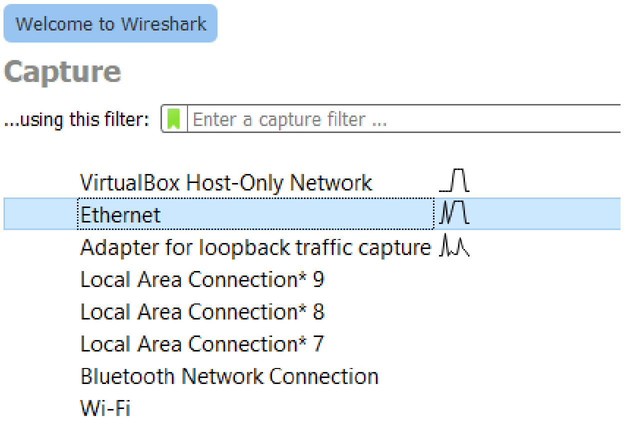
6. Click **Run**.



Commmunication port closed

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

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

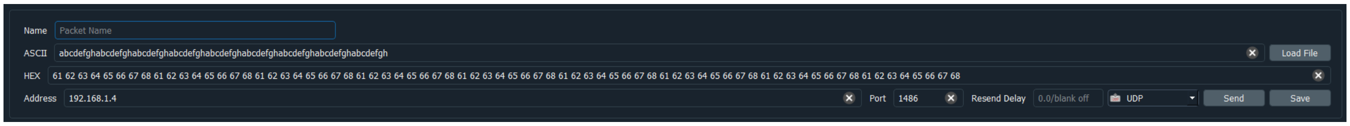


VirtualBox Host-Only Network
Ethernet
Adapter for loopback traffic capture
Local Area Connection* 9
Local Area Connection* 8
Local Area Connection* 7
Bluetooth Network Connection
Wi-Fi

Figure D.14. Wireshark Tool - Ethernet Selection

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

12. Click **Save**.



Name: Packet Name

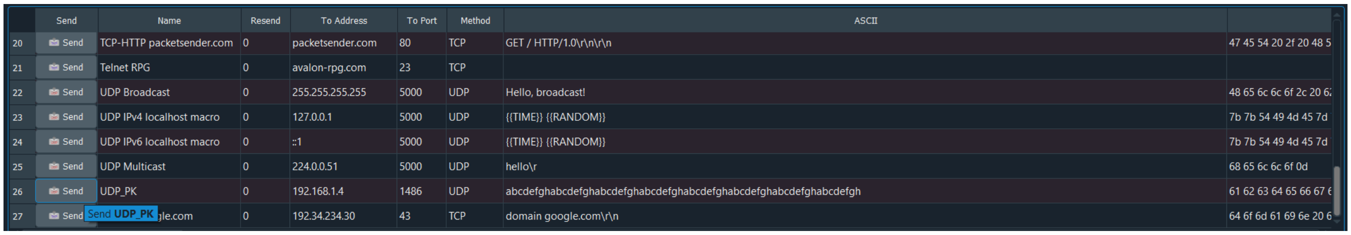
ASCII: abcdefghabcdefghabcdefghabcdefghabcdefghabcdefghabcdefghabcdefgh

HEX: 61 62 63 64 65 66 67 68 61 62 63 64 65 66 67 68 61 62 63 64 65 66 67 68 61 62 63 64 65 66 67 68 61 62 63 64 65 66 67 68 61 62 63 64 65 66 67 68 61 62 63 64 65 66 67 68 61 62 63 64 65 66 67 68

Address: 192.168.1.4 Port: 1486 Resend Delay: 0.0/blank off UDP

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

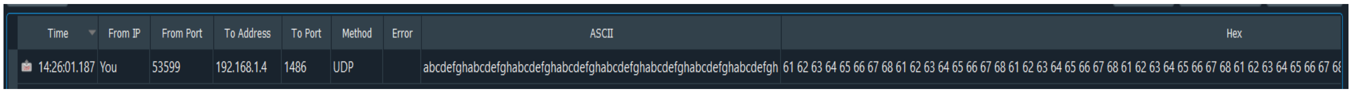
13. Click **Send** to send the UDP_PK packet.



	Send	Name	Resend	To Address	To Port	Method	ASCII	
20	Send	TCP-HTTP packetsender.com	0	packetsender.com	80	TCP	GET / HTTP/1.0\r\n\r\n	47 45 54 20 2f 20 48 5
21	Send	Telnet RPG	0	avalon-rpg.com	23	TCP		
22	Send	UDP Broadcast	0	255.255.255.255	5000	UDP	Hello, broadcast!	48 65 6c 6c 6f 2c 20 6
23	Send	UDP IPv4 localhost macro	0	127.0.0.1	5000	UDP	{{TIME}} {{RANDOM}}	7b 7b 54 49 4d 45 7d
24	Send	UDP IPv6 localhost macro	0	::1	5000	UDP	{{TIME}} {{RANDOM}}	7b 7b 54 49 4d 45 7d
25	Send	UDP Multicast	0	224.0.0.51	5000	UDP	hello\r	68 65 6c 6c 6f 0d
26	Send	UDP_PK	0	192.168.1.4	1486	UDP	abcdefghabcdefghabcdefghabcdefghabcdefghabcdefghabcdefghabcdefgh	61 62 63 64 65 66 67 6
27	Send	Send UDP_PK gle.com	0	192.34.234.30	43	TCP	domain google.com\r\n	64 6f 6d 61 69 6e 20 6

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

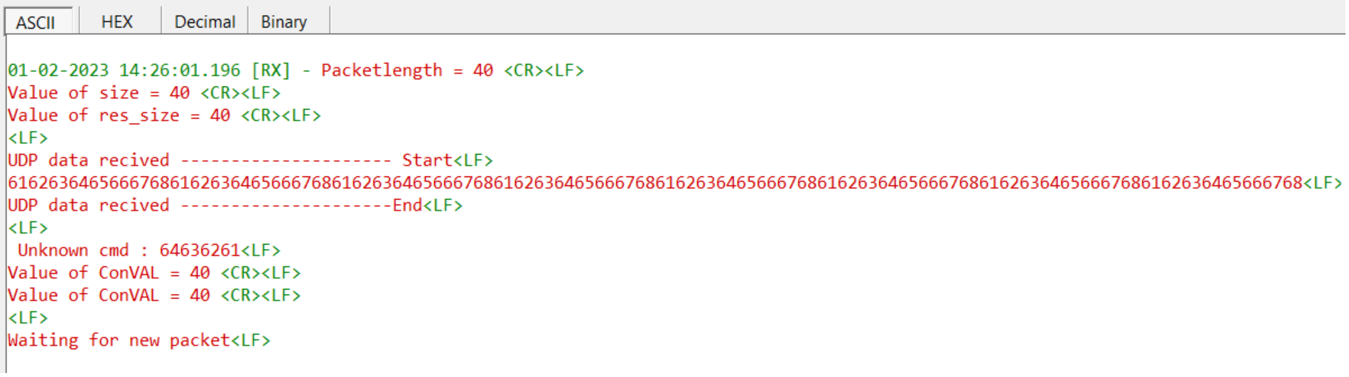
14. Check the log.



Time	From IP	From Port	To Address	To Port	Method	Error	ASCII	Hex
14:26:01.187	You	53599	192.168.1.4	1486	UDP		abcdefghabcdefghabcdefghabcdefghabcdefghabcdefghabcdefghabcdefgh	61 62 63 64 65 66 67 68 61 62 63 64 65 66 67 68 61 62 63 64 65 66 67 68 61 62 63 64 65 66 67 68 61 62 63 64 65 66 67 6

Figure D.17. Packet Sender Tool - Transaction Log

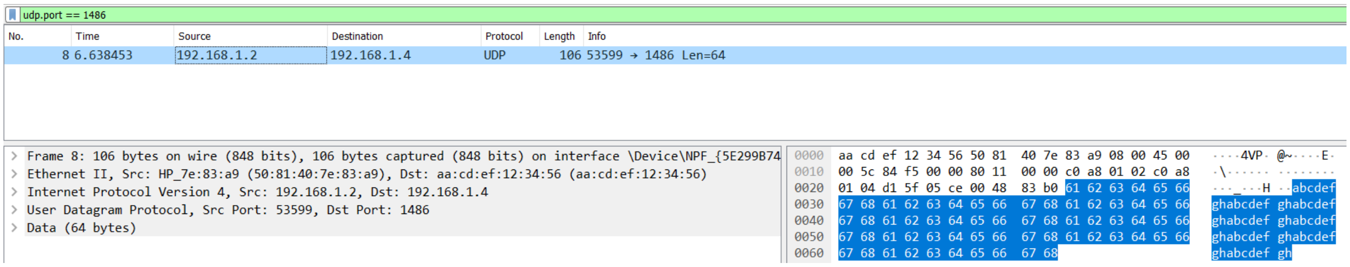
15. Open Docklight to check the data received.
16. Check that the UDP data was received.



```
01-02-2023 14:26:01.196 [RX] - Packetlength = 40 <CR><LF>
Value of size = 40 <CR><LF>
Value of res_size = 40 <CR><LF>
<LF>
UDP data recived --------------------- Start<LF>
61626364656667686162636465666768616263646566676861626364656667686162636465666768616263646566676861626364656667686162636465666768<LF>
UDP data recived ---------------------End<LF>
<LF>
 Unknown cmd : 64636261<LF>
Value of ConVAL = 40 <CR><LF>
Value of ConVAL = 40 <CR><LF>
<LF>
Waiting for new packet<LF>
```

Figure D.18. Docklight - Logs in ASCII

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



udp.port == 1486

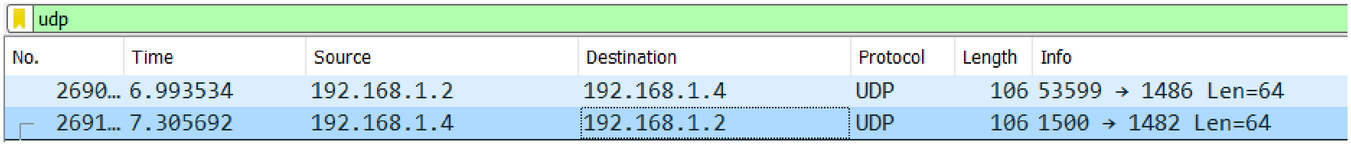
No.	Time	Source	Destination	Protocol	Length	Info
8	6.638453	192.168.1.2	192.168.1.4	UDP	106	53599 → 1486 Len=64

```
> Frame 8: 106 bytes on wire (848 bits), 106 bytes captured (848 bits) on interface \Device\NPF_{5E299B74
> Ethernet II, Src: HP_7e:83:a9 (50:81:40:7e:83:a9), Dst: aa:cd:ef:12:34:56 (aa:cd:ef:12:34:56)
> Internet Protocol Version 4, Src: 192.168.1.2, Dst: 192.168.1.4
> User Datagram Protocol, Src Port: 53599, Dst Port: 1486
> Data (64 bytes)
```

```
0000  aa cd ef 12 34 56 50 81  40 7e 83 a9 08 00 45 00   ····4VP· @~····E·
0010  00 5c 84 f5 00 00 80 11  00 00 c0 a8 01 02 c0 a8   ·\······ ········
0020  01 04 d1 5f 05 ce 00 48  83 b0 61 62 63 64 65 66   ···_···H ··abcdef
0030  67 68 61 62 63 64 65 66  67 68 61 62 63 64 65 66   ghabcdef ghabcdef
0040  67 68 61 62 63 64 65 66  67 68 61 62 63 64 65 66   ghabcdef ghabcdef
0050  67 68 61 62 63 64 65 66  67 68 61 62 63 64 65 66   ghabcdef ghabcdef
0060  67 68 61 62 63 64 65 66  67 68                     ghabcdef gh
```

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

19. Check both the Source and Destination UDP packets.

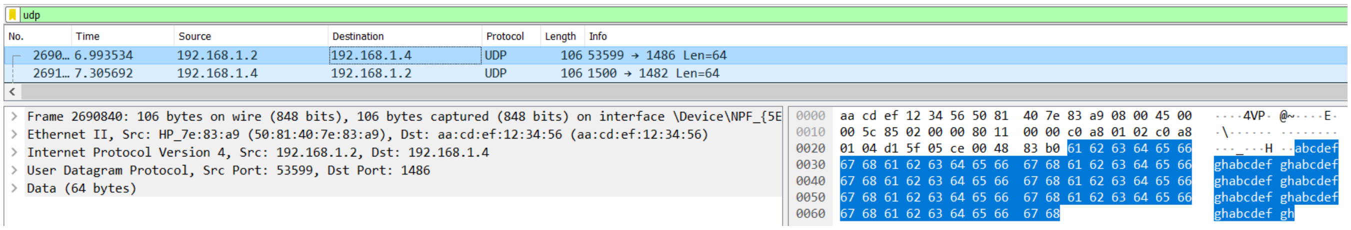


udp

No.	Time	Source	Destination	Protocol	Length	Info
2690…	6.993534	192.168.1.2	192.168.1.4	UDP	106	53599 → 1486 Len=64
2691…	7.305692	192.168.1.4	192.168.1.2	UDP	106	1500 → 1482 Len=64

Figure D.20. Source and Destination UDP Packet

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



udp

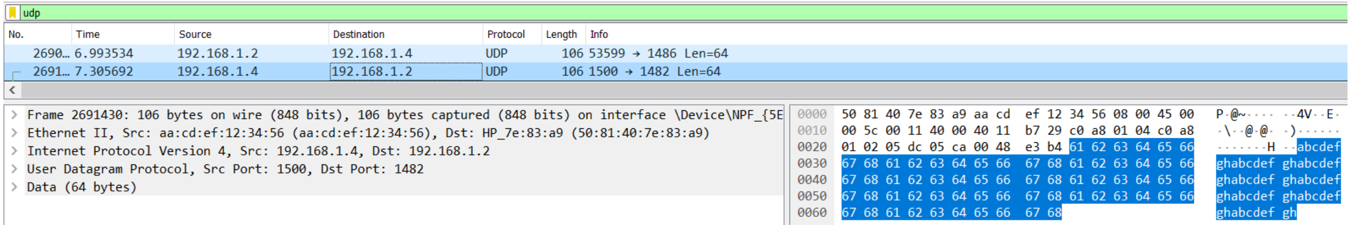
No.	Time	Source	Destination	Protocol	Length	Info
2690…	6.993534	192.168.1.2	192.168.1.4	UDP	106	53599 → 1486 Len=64
2691…	7.305692	192.168.1.4	192.168.1.2	UDP	106	1500 → 1482 Len=64

```
> Frame 2690840: 106 bytes on wire (848 bits), 106 bytes captured (848 bits) on interface \Device\NPF_{5E
> Ethernet II, Src: HP_7e:83:a9 (50:81:40:7e:83:a9), Dst: aa:cd:ef:12:34:56 (aa:cd:ef:12:34:56)
> Internet Protocol Version 4, Src: 192.168.1.2, Dst: 192.168.1.4
> User Datagram Protocol, Src Port: 53599, Dst Port: 1486
> Data (64 bytes)
```

```
0000  aa cd ef 12 34 56 50 81  40 7e 83 a9 08 00 45 00   ····4VP· @~····E·
0010  00 5c 85 02 00 00 80 11  00 00 c0 a8 01 02 c0 a8   ·\······ ········
0020  01 04 d1 5f 05 ce 00 48  83 b0 61 62 63 64 65 66   ···_···H ··abcdef
0030  67 68 61 62 63 64 65 66  67 68 61 62 63 64 65 66   ghabcdef ghabcdef
0040  67 68 61 62 63 64 65 66  67 68 61 62 63 64 65 66   ghabcdef ghabcdef
0050  67 68 61 62 63 64 65 66  67 68 61 62 63 64 65 66   ghabcdef ghabcdef
0060  67 68 61 62 63 64 65 66  67 68                     ghabcdef gh
```

Figure D.21. Wireshark Tool - First UDP Packet

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



udp

No.	Time	Source	Destination	Protocol	Length	Info
2690…	6.993534	192.168.1.2	192.168.1.4	UDP	106	53599 → 1486 Len=64
2691…	7.305692	192.168.1.4	192.168.1.2	UDP	106	1500 → 1482 Len=64

```
> Frame 2691430: 106 bytes on wire (848 bits), 106 bytes captured (848 bits) on interface \Device\NPF_{5E
> Ethernet II, Src: aa:cd:ef:12:34:56 (aa:cd:ef:12:34:56), Dst: HP_7e:83:a9 (50:81:40:7e:83:a9)
> Internet Protocol Version 4, Src: 192.168.1.4, Dst: 192.168.1.2
> User Datagram Protocol, Src Port: 1500, Dst Port: 1482
> Data (64 bytes)
```

```
0000  50 81 40 7e 83 a9 aa cd  ef 12 34 56 08 00 45 00   P·@~···· ··4V··E·
0010  00 5c 00 11 40 00 40 11  b7 29 c0 a8 01 04 c0 a8   ·\··@·@· ·)······
0020  01 02 05 dc 05 ca 00 48  e3 b4 61 62 63 64 65 66   ·······H ··abcdef
0030  67 68 61 62 63 64 65 66  67 68 61 62 63 64 65 66   ghabcdef ghabcdef
0040  67 68 61 62 63 64 65 66  67 68 61 62 63 64 65 66   ghabcdef ghabcdef
0050  67 68 61 62 63 64 65 66  67 68 61 62 63 64 65 66   ghabcdef ghabcdef
0060  67 68 61 62 63 64 65 66  67 68                     ghabcdef gh
```

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
```

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

01 - 1st node

02 - 2nd node

04 - 3rd node and so on

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
```

D.5.2. Recommend Command Sequences to test basic functionality

Basic LED OFF/ON command:

- Send LED OFF Command
- Send LED ON command

Motor Config Test:

- Send the Update config command

Motor start from Dashboard Test:

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

Motor start from Motor Status Test:

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

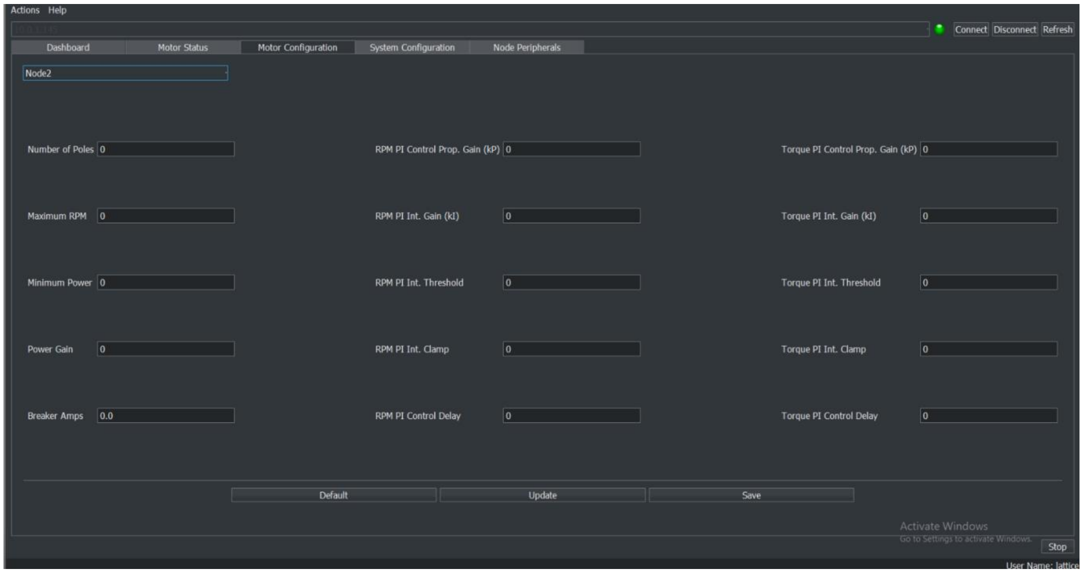
Appendix E. Troubleshooting

E.1. Automate 3.0 GUI Troubleshooting

E.1.1. Motor Configuration Values are 0

Problem Description:

Motor configuration values are all 0.



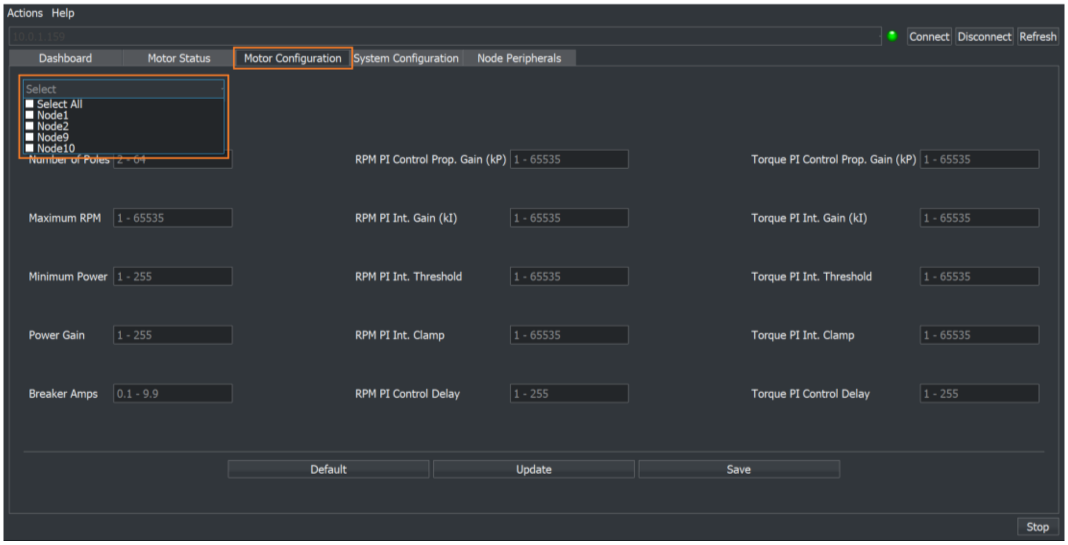
Node2

Number of Poles 0
Maximum RPM 0
Minimum Power 0
Power Gain 0
Breaker Amps 0.0
RPM PI Control Prop. Gain (kP) 0
RPM PI Int. Gain (kI) 0
RPM PI Int. Threshold 0
RPM PI Int. Clamp 0
RPM PI Control Delay 0
Torque PI Control Prop. Gain (kP) 0
Torque PI Int. Gain (kI) 0
Torque PI Int. Threshold 0
Torque PI Int. Clamp 0
Torque PI Control Delay 0

Figure E.1. Motor Configuration Values are All 0

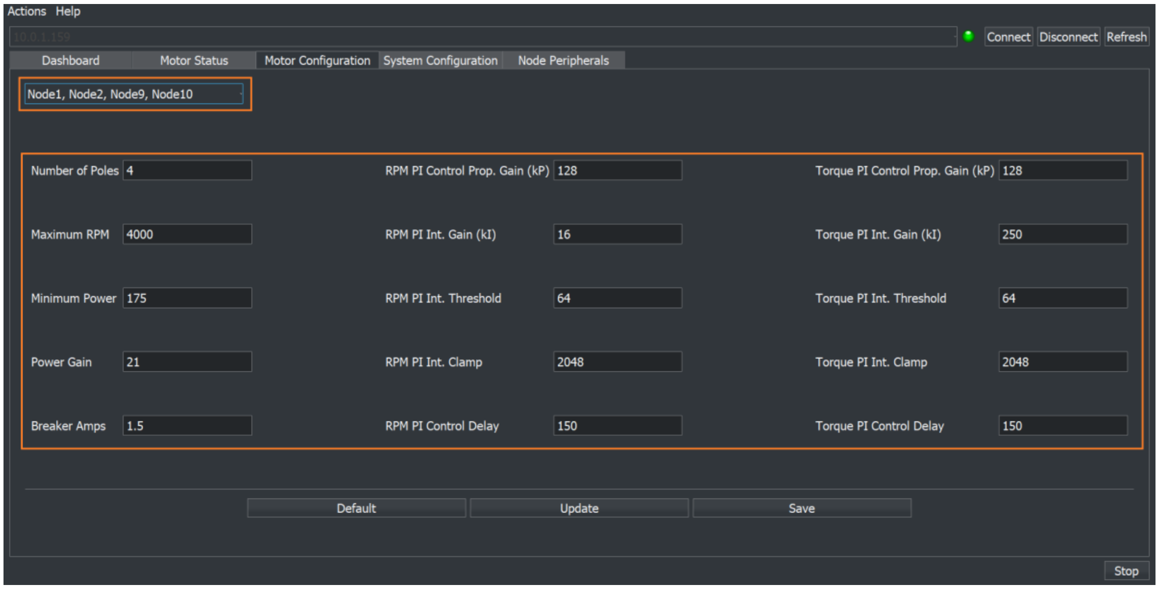
Solution:

1. In the Automate 3.0 GUI, click the **Motor Configuration** tab.
2. From the Select drop-down menu, either Select **All** or select an individual node.



Number of Poles 2 - 64
Maximum RPM 1 - 65535
Minimum Power 1 - 255
Power Gain 1 - 255
Breaker Amps 0.1 - 9.9
RPM PI Control Prop. Gain (kP) 1 - 65535
RPM PI Int. Gain (kI) 1 - 65535
RPM PI Int. Threshold 1 - 65535
RPM PI Int. Clamp 1 - 65535
RPM PI Control Delay 1 - 255
Torque PI Control Prop. Gain (kP) 1 - 65535
Torque PI Int. Gain (kI) 1 - 65535
Torque PI Int. Threshold 1 - 65535
Torque PI Int. Clamp 1 - 65535
Torque PI Control Delay 1 - 255

Figure E.2. Motor Configuration: Node Selection



Node1, Node2, Node9, Node10

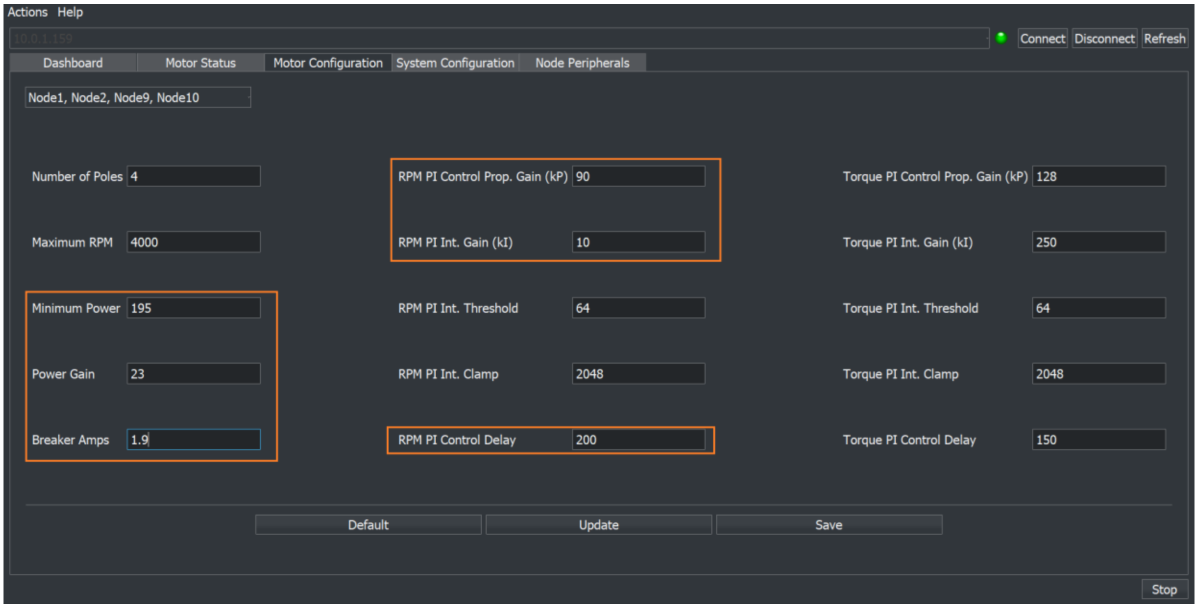
Number of Poles	4	RPM PI Control Prop. Gain (kP)	128	Torque PI Control Prop. Gain (kP)	128
Maximum RPM	4000	RPM PI Int. Gain (kI)	16	Torque PI Int. Gain (kI)	250
Minimum Power	175	RPM PI Int. Threshold	64	Torque PI Int. Threshold	64
Power Gain	21	RPM PI Int. Clamp	2048	Torque PI Int. Clamp	2048
Breaker Amps	1.5	RPM PI Control Delay	150	Torque PI Control Delay	150

Figure E.3. Motor Configuration: All Nodes Selected

3. Change the Motor Configuration fields to the following default values:

- RPI PI Control Prop. Gain(kP): **90**
- RPI PI Int. Gain(kI): **10**
- RPI PI Control delay: **200**
- Minimum Power: 190 - **200**
- Power gain: **22 - 24**
- Breaker Amps: **1.8 - 2.0**

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



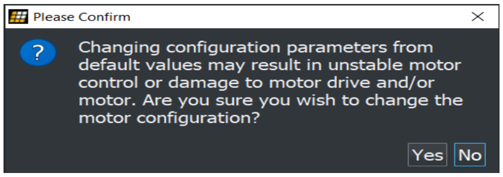
Node1, Node2, Node9, Node10

Number of Poles	4	RPM PI Control Prop. Gain (kP)	90	Torque PI Control Prop. Gain (kP)	128
Maximum RPM	4000	RPM PI Int. Gain (kI)	10	Torque PI Int. Gain (kI)	250
Minimum Power	195	RPM PI Int. Threshold	64	Torque PI Int. Threshold	64
Power Gain	23	RPM PI Int. Clamp	2048	Torque PI Int. Clamp	2048
Breaker Amps	1.9	RPM PI Control Delay	200	Torque PI Control Delay	150

Figure E.4. Motor Configuration Value Changes

4. Click **Default**.

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



Please Confirm

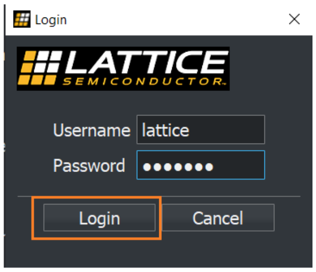
Changing configuration parameters from default values may result in unstable motor control or damage to motor drive and/or motor. Are you sure you wish to change the motor configuration?

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

7. Enter the Credentials in the authentication pop-up and click on **Login**.

   Username**: lattice**

   Password**: lattice**



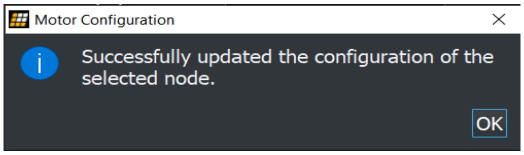
Login

Username: lattice

Password: •••••••

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

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



Motor Configuration

Successfully updated the configuration of the selected node.

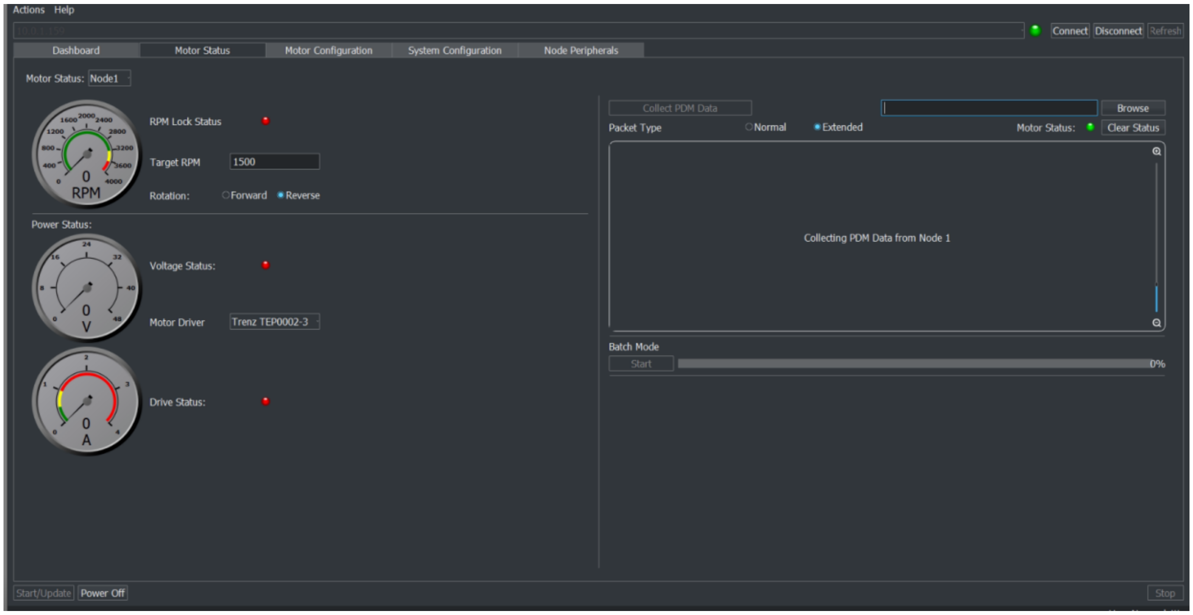
Figure E.7. Application Software - Motor Configuration: Update Configuration

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

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

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

Solution: Power everything off and restart the setup.



Motor Status: Node1

Target RPM 1500

Rotation: ○ Forward ● Reverse

Motor Driver Trenz TEP0002-3

Packet Type ○ Normal ● Extended

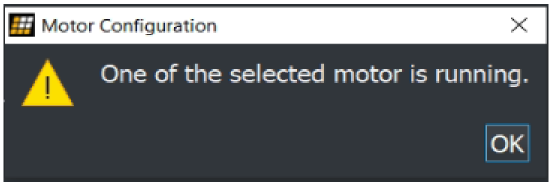
Collecting PDM Data from Node 1

0%

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

E.1.3. One of the Selected Motor Running Warning

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



One of the selected motor is running.

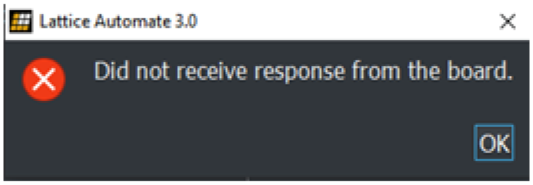
Figure E.9. Motor Running Warning Message

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

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

Problem Description:

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



Did not receive response from the board.

Figure E.10. Motor Connection Message

Solution:

1. Check the Ethernet connection. Follow the steps in Appendix D. Ethernet Connection 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 Docklight to check the print statements from the main board:
   a. Connect a UART cable between the Main Board and a PC with Docklight installed.
   b. Open Docklight on the PC.
   c. Double click COM.

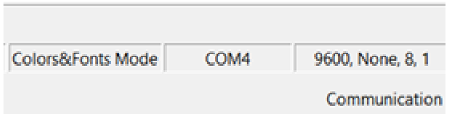


Figure E.11. Motor Connection Message

   d. Select the last USB serial port in the list, as shown below.

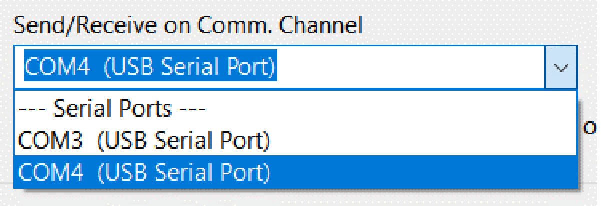
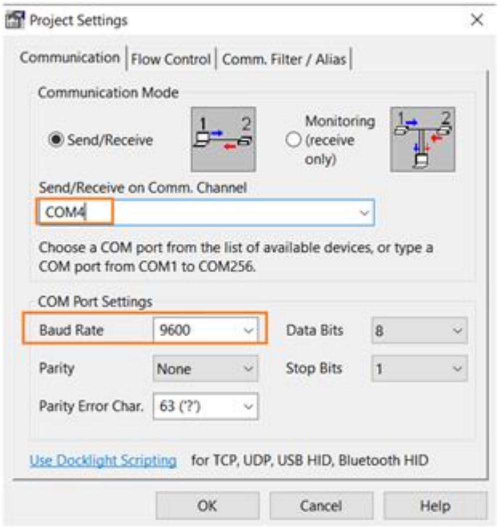


Figure E.12. Send/Receive on Comm. Channel

e. Select baud rate **9600**.



Project Settings
Communication | Flow Control | Comm. Filter / Alias
Communication Mode
● Send/Receive
○ Monitoring (receive only)
1 2
OK
Cancel
Help

Send/Receive on Comm. Channel: COM4

Choose a COM port from the list of available devices, or type a COM port from COM1 to COM256.

COM Port Settings

Baud Rate: 9600
Data Bits: 8
Parity: None
Stop Bits: 1
Parity Error Char.: 63 ('?')

Use Docklight Scripting for TCP, UDP, USB HID, Bluetooth HID

Figure E.13. Project Settings

f. Click **Run.**

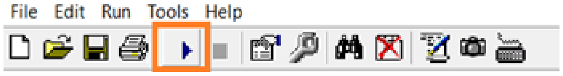
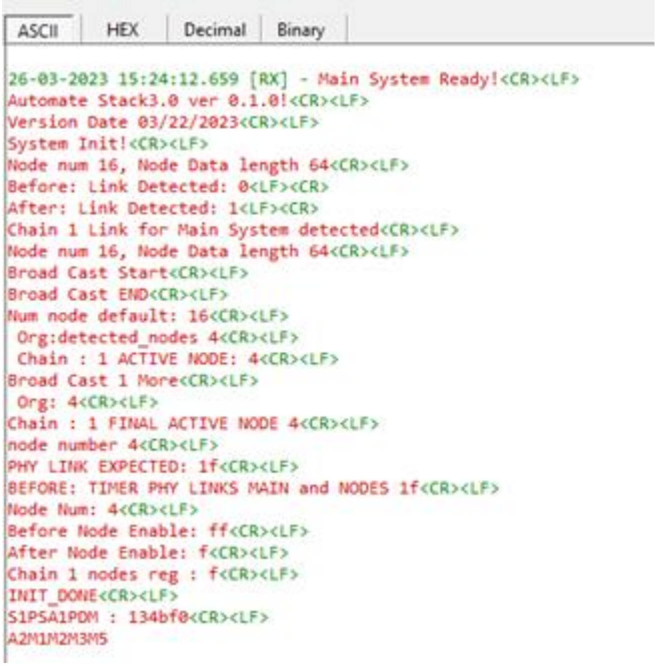


Figure E.14. Run Button

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

h. In Docklight, select the ASCII tab. Wait 60 seconds and check that the print statements are valid, as shown below.



```
26-03-2023 15:24:12.659 [RX] - Main System Ready!<CR><LF>
Automate Stack3.0 ver 0.1.0!<CR><LF>
Version Date 03/22/2023<CR><LF>
System Init!<CR><LF>
Node num 16, Node Data length 64<CR><LF>
Before: Link Detected: 0<LF><CR>
After: Link Detected: 1<LF><CR>
Chain 1 Link for Main System detected<CR><LF>
Node num 16, Node Data length 64<CR><LF>
Broad Cast Start<CR><LF>
Broad Cast END<CR><LF>
Num node default: 16<CR><LF>
 Org:detected_nodes 4<CR><LF>
 Chain : 1 ACTIVE NODE: 4<CR><LF>
Broad Cast 1 More<CR><LF>
 Org: 4<CR><LF>
Chain : 1 FINAL ACTIVE NODE 4<CR><LF>
node number 4<CR><LF>
PHY LINK EXPECTED: 1f<CR><LF>
BEFORE: TIMER PHY LINKS MAIN and NODES 1f<CR><LF>
Node Num: 4<CR><LF>
Before Node Enable: ff<CR><LF>
After Node Enable: f<CR><LF>
Chain 1 nodes reg : f<CR><LF>
INIT_DONE<CR><LF>
S1PSA1PDM : 134bf0<CR><LF>
A2M1M2M3M5
```

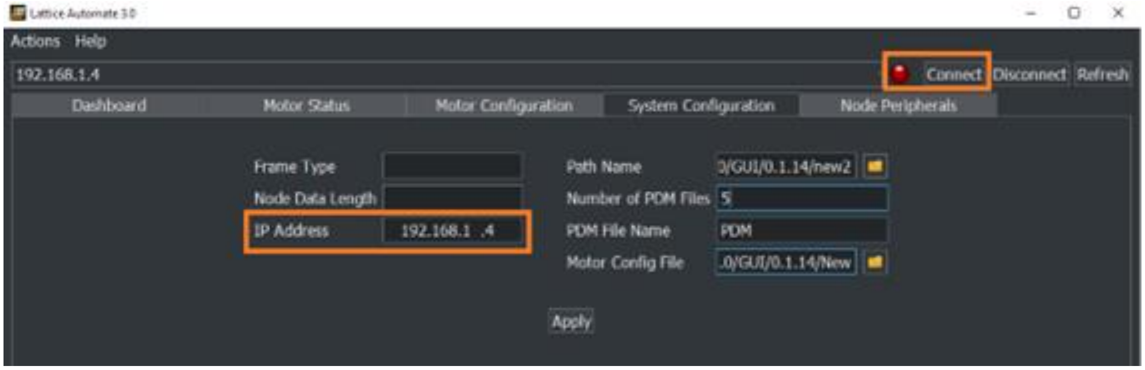
Figure E.15. ASCII Print Statements

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

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

4. Check the IP address on the GUI.

a. Make sure the Automate 3.0 GUI is set to connect to IP address 192.168.1.4.



192.168.1.4

Frame Type

Node Data Length

IP Address 192.168.1 .4

Path Name 0/GUI/0.1.14/new2

Number of PDM Files 5

PDM File Name PDM

Motor Config File .0/GUI/0.1.14/New

Figure E.16. Automate 3.0 GUI

b. If users changed the main system's IP address in Propel Builder, make sure using the changed IP address. See Reference Design Document Section 2.6 for more details about how to change the main system's IP address.

Annexure-A

Cable Description

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

Technical Support Assistance

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

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

Revision History

Revision 1.1, June 2023

Section	Change Summary
Communication Protocols (OPCUA)	Updated Figure 3.1. OPCUA High-Level Block Diagram. Updated Figure 3.2. UDP IP Core Overview to change CARE to CORE.
Hardware	Added Step 6 in the Hardware Setup section. Added Figure 4.2. Main System Ready LEDs. Added Figure 4.3. Node System Ready LEDs.
Running the Motor through Automate GUI Application	Added Refresh Button section. Revised step 8 in Collect PDM Data section. Revised PDM Data Collection Process for normal and extended mode in Details about PDM Data Collection Process section. Updated Figure 7.43. GUI Application PDM Data: Components Involved in PDM Data Collection Process to change from Etherconnect to Ethernet.
Appendix B. Programming the Automate Stack on Respective FLASH	Updated Figure B.7. Radiant Programmer - Bitstream Flashing Settings (Main System).
Appendix C. Programming a Fresh Main System Board	Updated Figure C.5. Radiant Programmer - Quad Mode Programming.
Appendix D. Ethernet Connection	Added D.5. Packet Sender Commands section.
Appendix E. Troubleshooting	Added E.1.4. Did not Receive Response from the Board Warning section.

Revision 1.0, March 2023

Section	Change Summary
All	Initial release.

LATTICE SEMICONDUCTOR®

www.latticesemi.com