Industrial Automation
Tech Note 48

Emerson ROC Communications

red lion®

Abstract:

This document describes configuring communications between an approved Red Lion Controls (RLC) product and an Emerson Remote Operations Controller (ROC), using serial and Ethernet communications.

Products:

CR1000 Human Machine Interface (HMI), CR3000 HMI , Data Station Plus (DSP), G3 HMI, G3 Kadet HMI , Graphite® Controller, Graphite HMIs , Modular Controller, and ProducTVity Station™ (PTV)

Use Case: Communicating with an ROC

This document describes the configuration steps/settings needed to properly configure communications between an approved RLC product and an Emerson ROC, using serial and Ethernet communications.

Required Software:

Crimson® 3.0 or 3.1

Required Firmware:

Crimson 3.0 build 693 or higher

Crimson 3.1 all builds

Required Operating System:

Microsoft Windows 2000, or above

Introduction

Approved RLC products previously detailed in this document can be configured to communicate with Emerson ROCs using either serial communications or Ethernet communications. This document describes the steps that must be followed for both communications methods, including: the serial and Ethernet configuration of the ROC and Red Lion device, and creating and mapping Crimson tags to the ROC device.

Serial Configuration

Two serial configuration options can be used to configure an approved RLC device to communicate with an Emerson ROC: ROCLINK serial configuration and Crimson serial configuration. Both options are described in the remainder of this section.

ROCLINK Serial Configuration

Perform the following steps to configure an RLC device to communicate with an Emerson ROC using ROCLINK serial configuration:

1. Referring to Figure 1, go to the menu bar and click *ROC – Comm Ports*.
2. Select the port to configure from the options provided in the Comm Port popup; 1 - Local Port in this example.
3. Click the General tab.
4. Configure the `Baud Rate`, `Parity`, `Data Bits`, and `Stop Bits` parameters, as appropriate for your configuration.
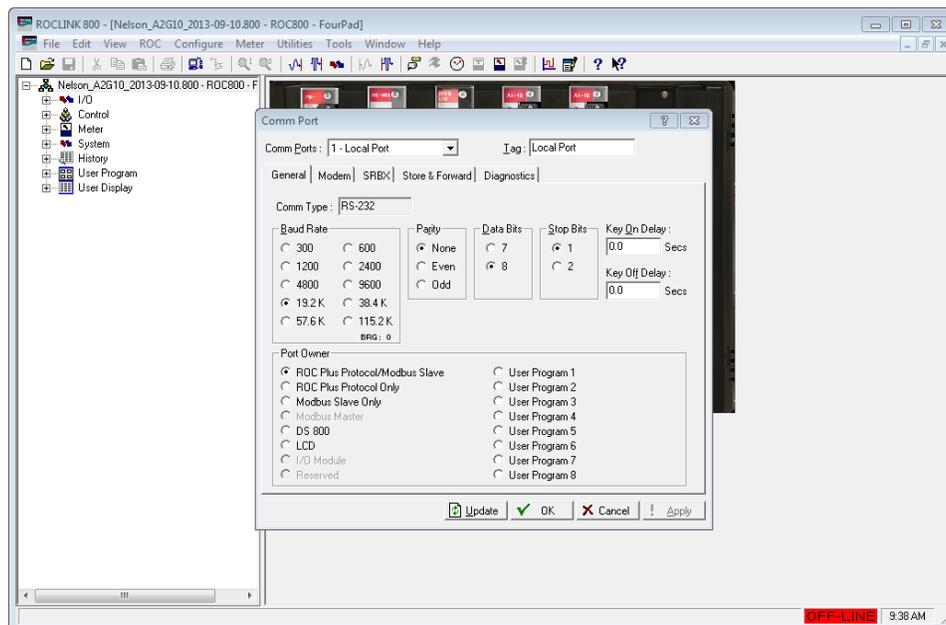5. Set the `Port Owner` parameter to an option that includes *ROC Plus Protocol*.

Figure 1.

6. Referring to Figure 2, click the SRBX tab.

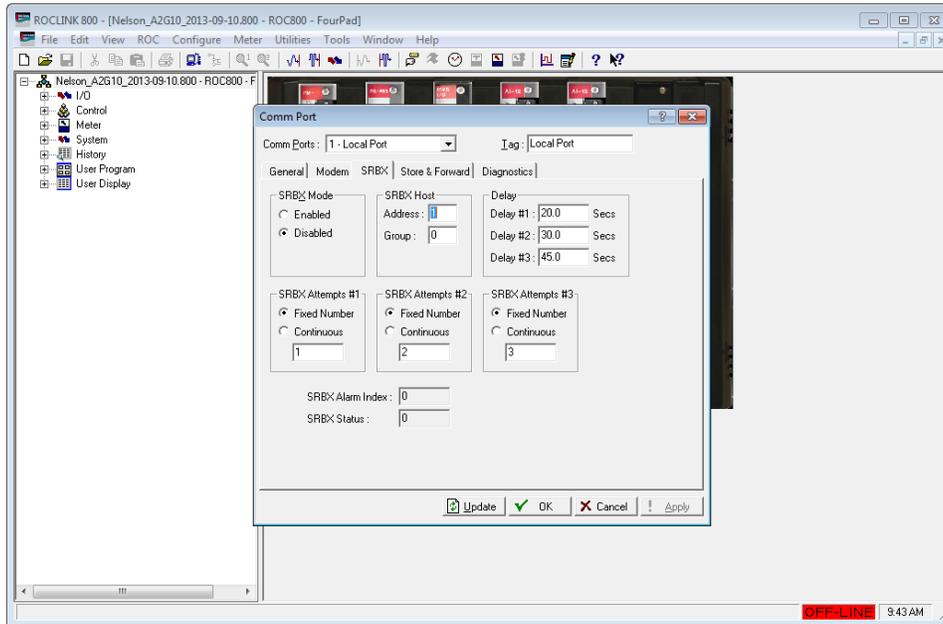7. Configure the `Address` and `Group` parameters in the SRBX Host section, as appropriate.

Figure 2.

8. Click *OK*.

Crimson Serial Configuration

Perform the following steps to configure an RLC device to communicate with an Emerson ROC using Crimson serial configuration:

1. Referring to Figure 3, navigate to the Communications section.

2. Click on the serial port to be configured; *RS-232 Comms Port* in this example. The Communications - RS-232 Comms Port window popup appears.

3. Click *Pick* next to the `Driver Selection` parameter; the Driver Picker for Serial Port popup appears.

4. Under the Manufacturer heading, select *Emerson Process*.

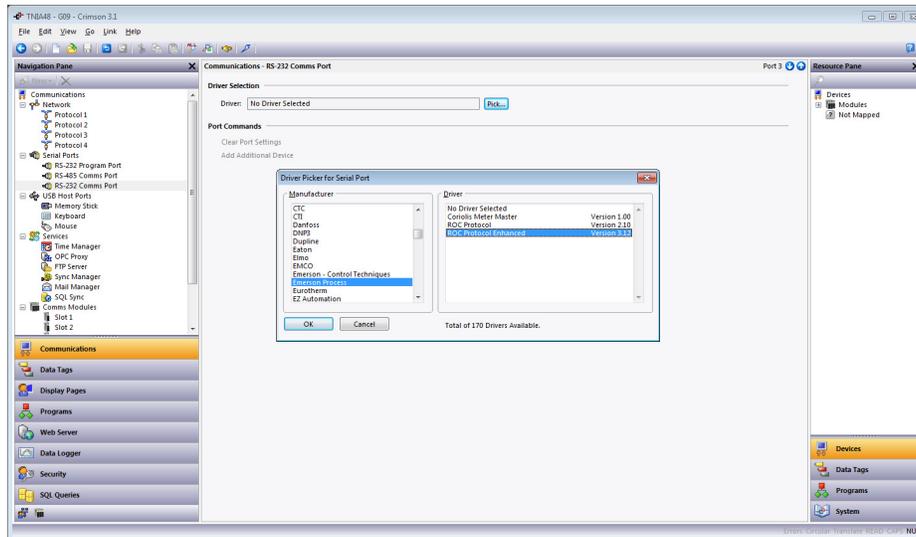5. Under the Driver heading, select *ROC Protocol Enhanced*.

Figure 3.

6. Click *OK*.

7. Referring to Figure 4, configure the Host Identification section with the `Group Number` and `Unit Number` parameters used by the RLC device.

8. Configure the Port Settings to match the ROC's port settings.
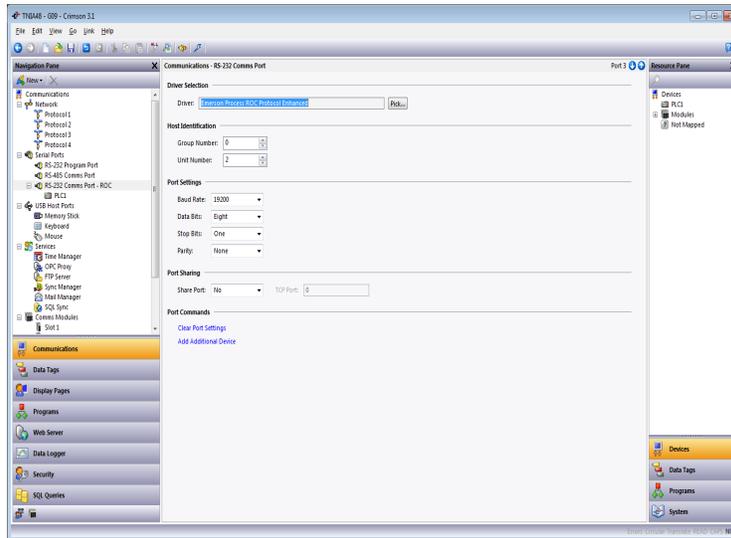

Figure 4.

9. Navigate to the Resource Pane and click on the device below the Comms Port, *PLC1* in this example; the Communications - RS-232 Comms Port -PLC1 popup shown in Figure 5 appears.
10. Configure the Identification section with the `Group Number` and `Unit Number` of the ROC device.
11. Configure the Security section parameters, as appropriate.
12. Under Settings:
    a. Set the `Ping TLP` parameter so that it will return a UINT8 value.
    b. Set the `User Defined TLP Selection` parameter to:
        1) *Yes,* to enter the TLP numbers manually.
        2) *No,* to choose the TLPs by name.
    c. If the `User Defined TLP Selection` is set to *No*, set the Hardware Series to the model of ROC being used.
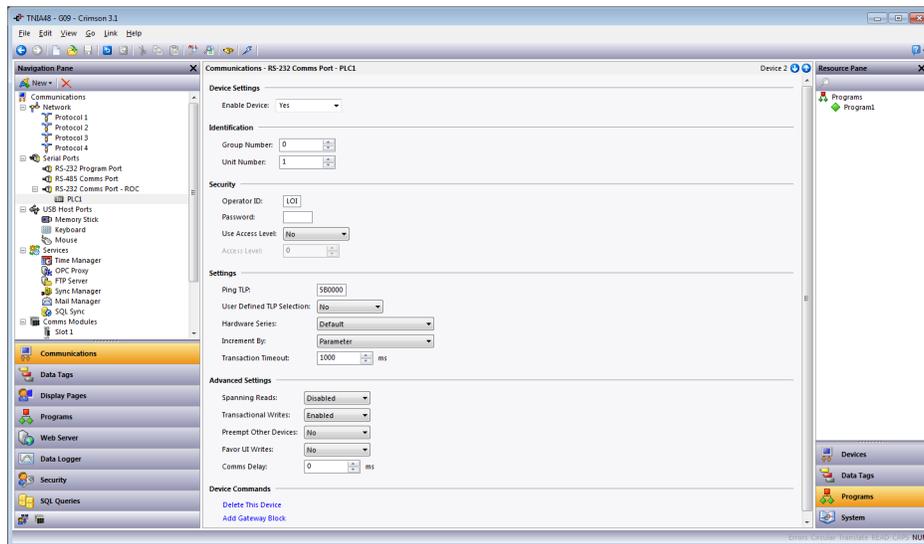13. Under Advanced Settings, set the `Spanning Reads` parameter to *Disabled*.

Figure 5.

ROCLINK Ethernet Configuration

Perform the following steps to configure an RLC device to communicate with an Emerson ROC using ROCLINK Ethernet Configuration:

1. Referring to Figure 6, go to the menu bar and click *ROC – Comm Ports*.
2. Go to the Comm Ports drop down menu and select `2 - Ethernet`.
3. Click the General tab.
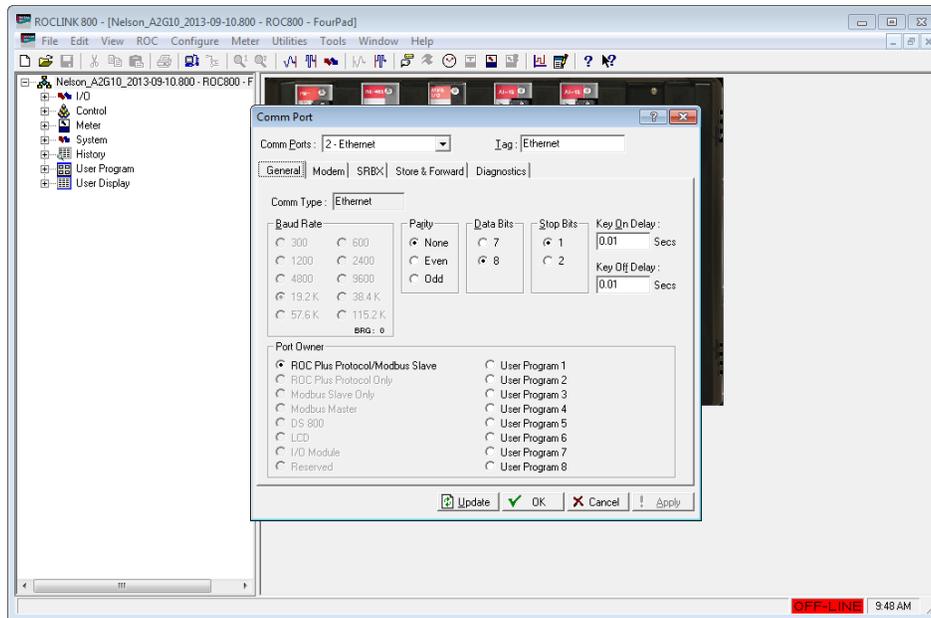4. Set the Port Owner to an option that includes *ROC Plus Protocol*.

Figure 6.

5. Referring to Figure 7, click the SRBX tab.

6. Configure the `Address` and `Group` parameters in the SRBX Host section, as appropriate.
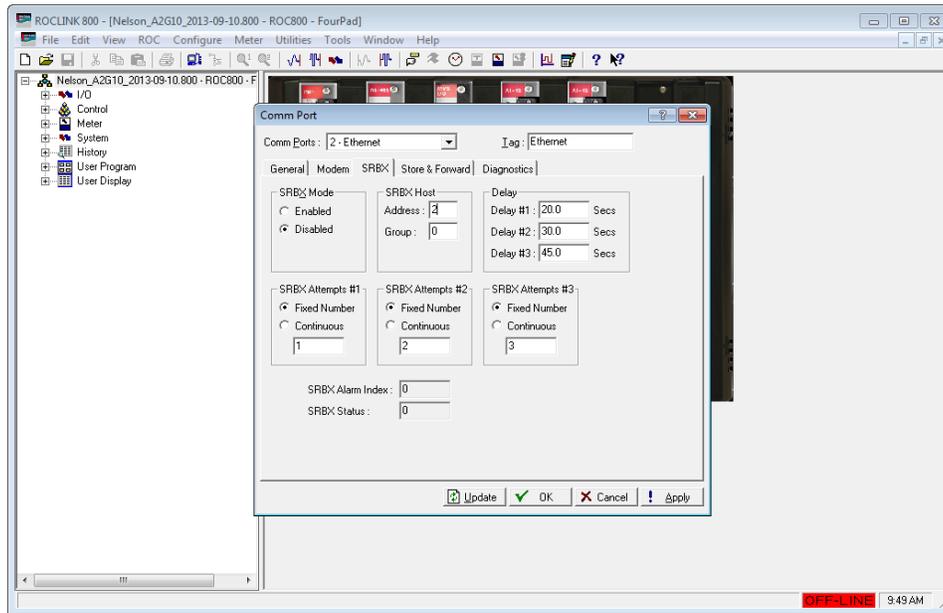

Figure 7.

7. Click *OK*.

8. Referring to Figure 8, go to the menu bar and click *ROC – Information*.

9. Click on the Internet tab.

10. Configure the Addressing section, as appropriate.

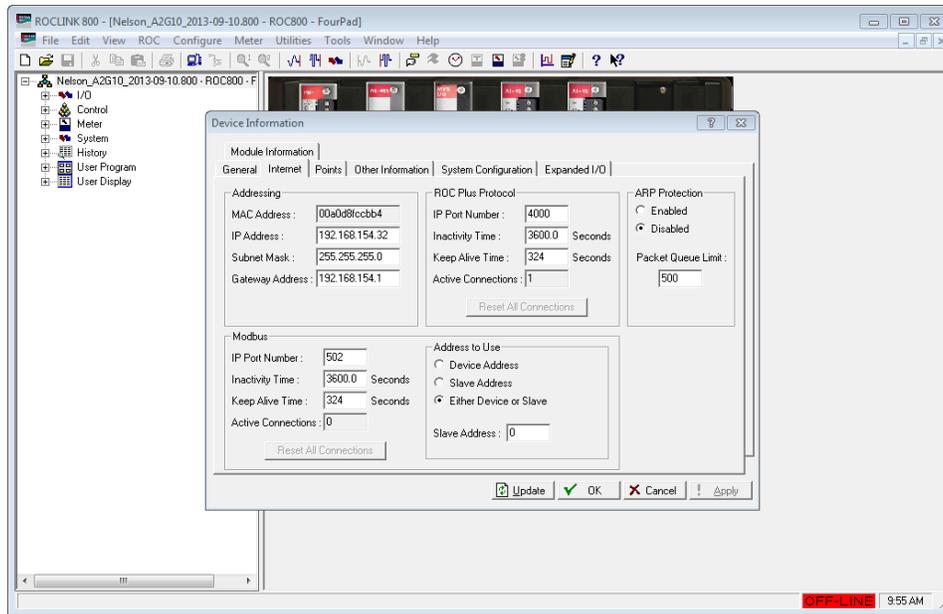11. Configure the `IP Port Number` under the ROC Plus Protocol heading, as appropriate.

Figure 8.

12. Click *OK*.

Crimson Ethernet Configuration

Perform the following steps to configure an RLC device to communicate with an Emerson ROC using Crimson Ethernet Configuration:

1. Referring to Figure 9, navigate to the Communications section.
2. Click on *Network*.
3. Click on the tab of the port to be configured; *Ethernet 1* in this example.
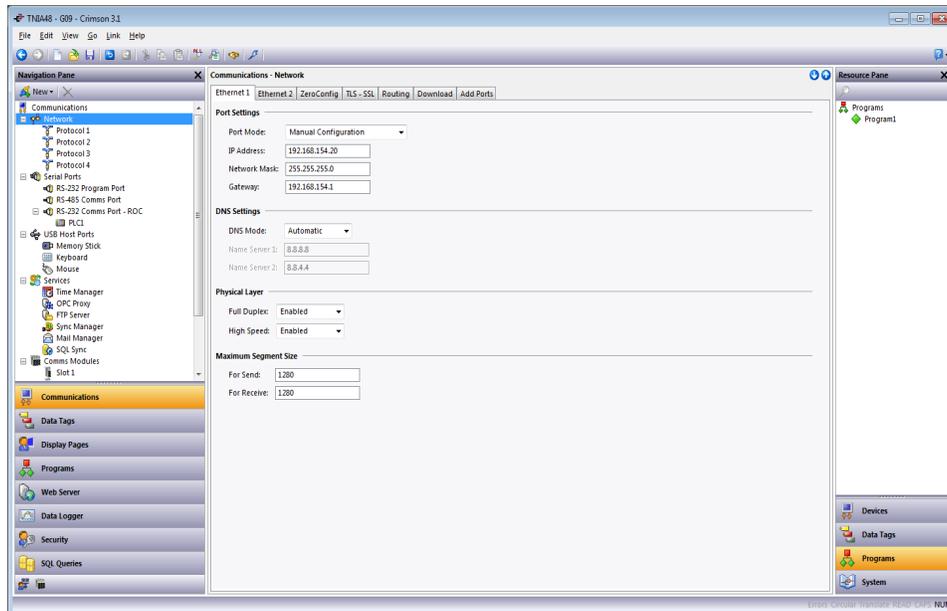4. Configure the Port Settings parameters, as needed for the application.

Figure 9.

5. Referring to Figure 10, go to the Navigation Pane and click on one of the available protocols under Network; *Protocol 1* in this example. The Communications - Network - Protocol 1 popup appears.

6. Click the *Pick* button under the Driver Selection heading; the Driver Picker for Ethernet Port popup appears.

7. Under the Manufacturer heading, select *Emerson Process*.

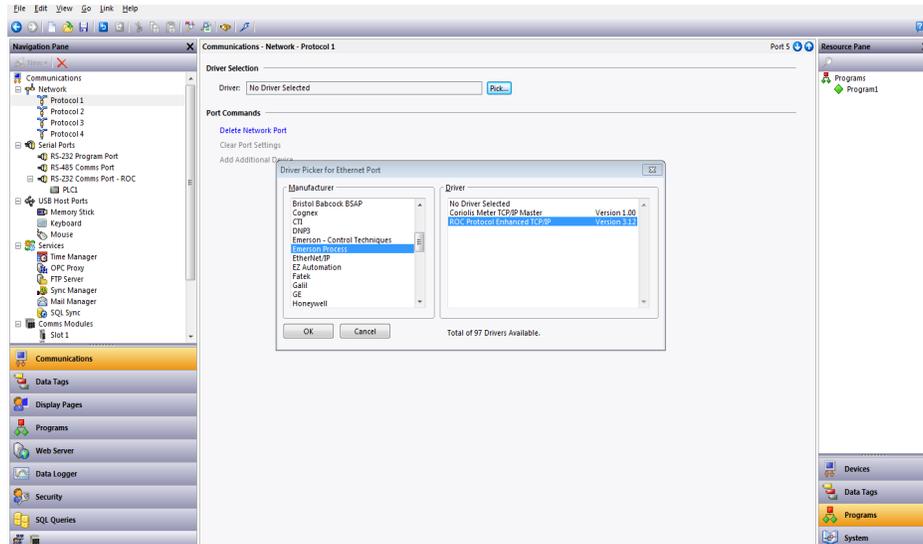8. Under Driver, select *ROC Protocol Enhanced TCP/IP*.

Figure 10.

9. Click *OK*; the Communications - Network - Protocol 1 popup shown in Figure 11 appears.

10. Referring to Figure 11, configure the Host Identification section with the `Group Number` and `Unit Number` that the RLC device will use.
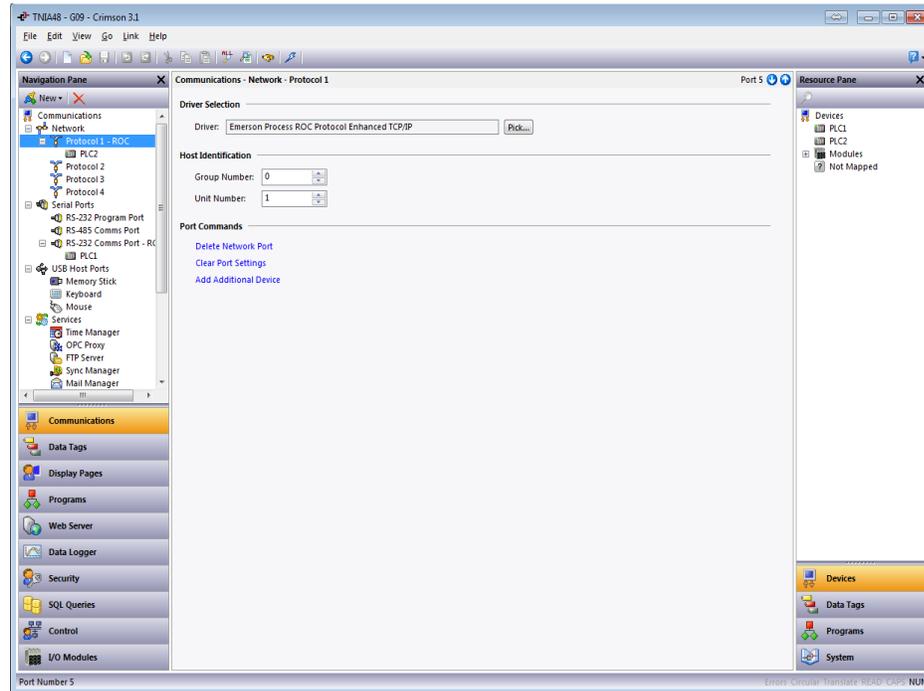

Figure 11.

11. Referring to Figure 12, go to the Navigation Pane and click on the device below the Protocol1 - ROC heading; *PLC2* in this example. The Communications - Network - Protocol 1 - PLC2 popup appears.

12. Configure the Identification section with the `Group Number` and `Unit Number` of the ROC device.

13. Configure the Security section, as needed.

14. Under Settings:
    a. Set the `Ping TLP` parameter to a TLP that will return a UINT8 value.
    b. Set the `User Defined TLP Selection` parameter to:
       1) *Yes,* to enter the TLP numbers manually.
       2) *No,* to choose the TLPs by name.
    c. If the `User Defined TLP Selection` parameter is set to *No*, set the `Hardware Series` parameter to the model of ROC being used.
    d. Specify whether the `Increment By` parameter increments by *Parameter* or *Logical Number*. The correct setting here is most useful when using the smart duplicate tag function or data arrays. When using data arrays, it is most useful to set this parameter to *Logical Number* as all array elements will be guaranteed to be of the same data type. It is recommended that this setting be configured before tags are mapped to this device. In the event that both increment methods apply within a single application, it is possible to create two identical devices for the Emerson Process ROC Protocol Enhanced driver; with only the `Increment By` parameter set uniquely.

15. Under Protocol options, set the `IP Address` and `TCP Port` parameters to match the ROC's configuration.

16. Set the `Link Type` parameter to *Use Dedicated Socket*, unless more than four devices will be configured under this protocol.

17. Under Advanced Settings, set the `Spanning Reads` parameter to *Disabled*.
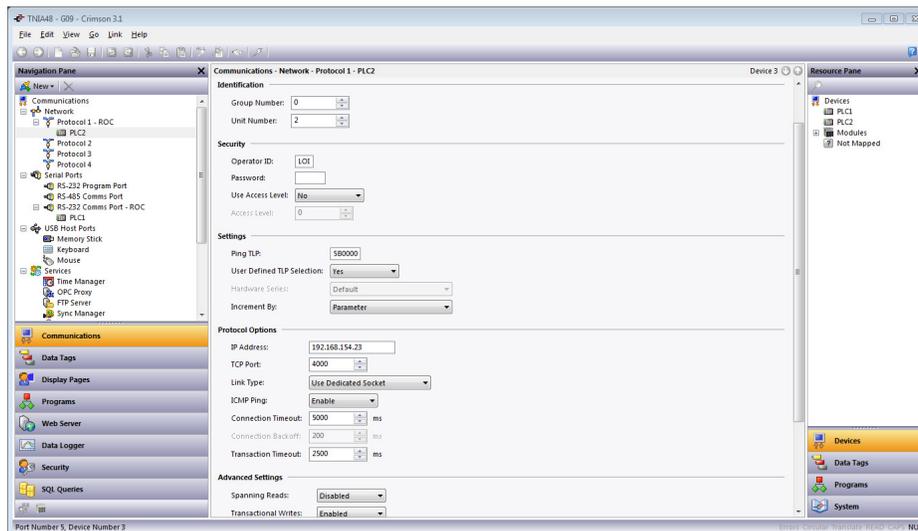

Figure 12.

Mapping Crimson Tags
<u>**User Defined TLPs**</u>

1. Referring to Figure 13, navigate to the Data Tags section.

2. Create a new data tag of the appropriate type; *Numeric* in this example.
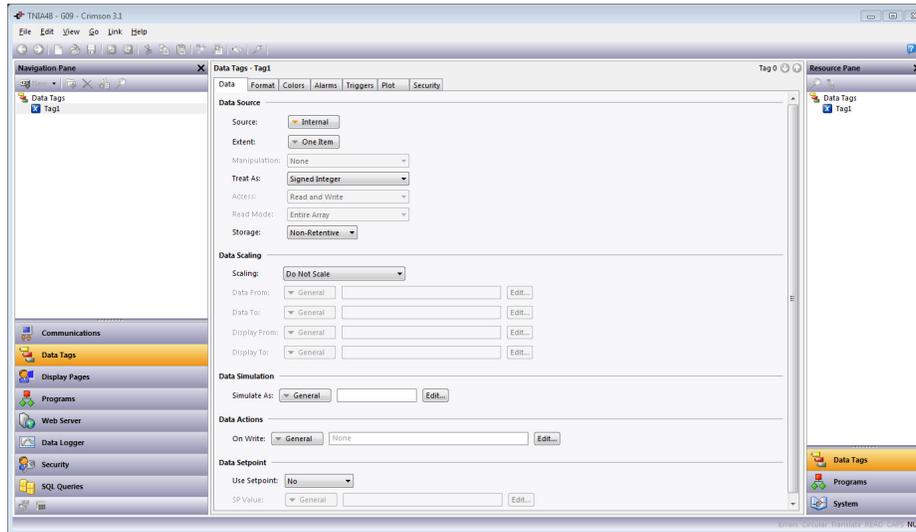
3. Click on the newly created tag.

Figure 13.

4. Click *Internal* next to the `Source` parameter and select the ROC device; PLC1 in this example. The Select TLP popup appears, as shown in Figure 14.
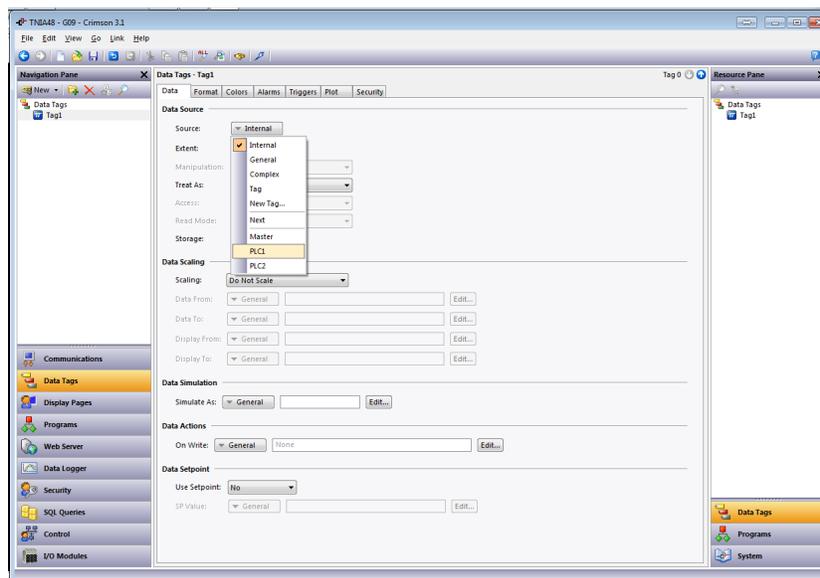

Figure 14.

5. Referring to Figure 15:

    a. Choose the appropriate `Point Type`, `Logical Number`, `Parameter`, and the associated `Data Type` settings; 7, 0, 51, and Float in this example.

    Or

    b. Check the `Use Opcode Table Access` option and specify the `Table Number`, `Location`, and `Data Type` settings.
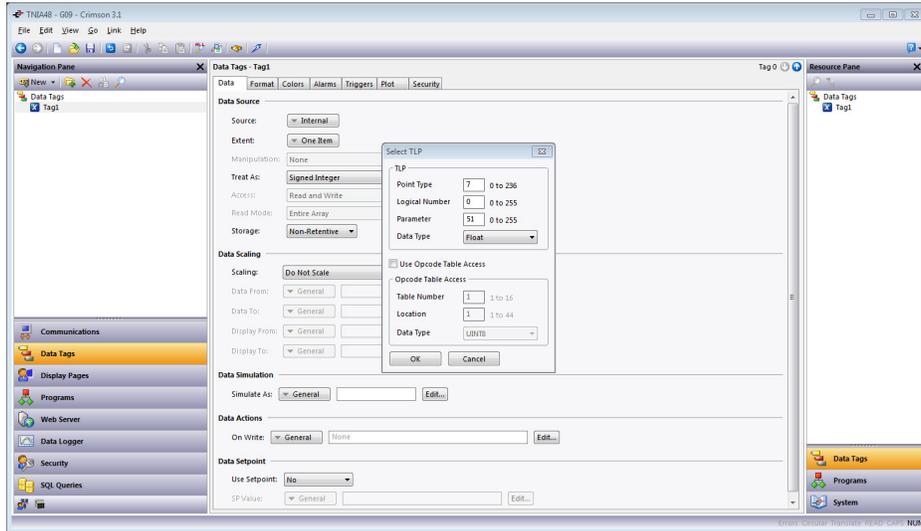

Figure 15.

6. Click *OK*.

Using Predefined TLPs

1. Referring to Figure 16, navigate to the Data Tags section.
2. Create a new data tag of the appropriate type; *Numeric* in this example.
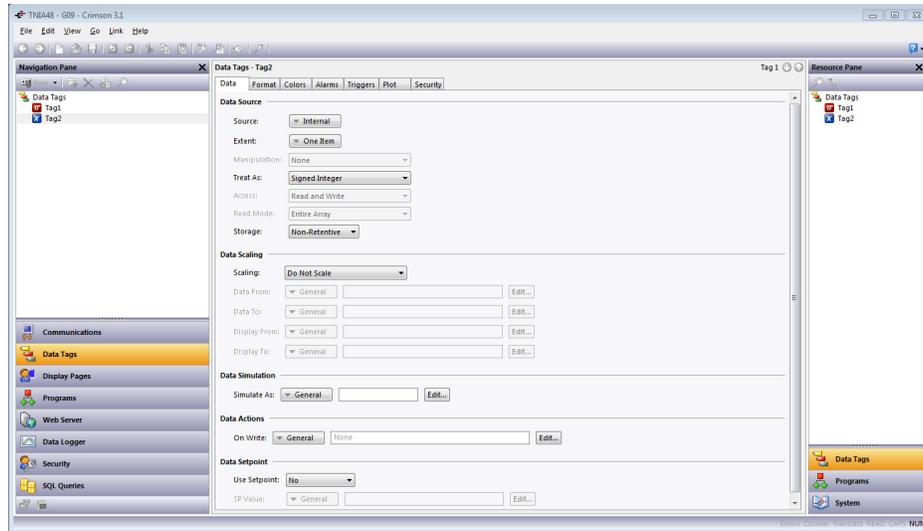3. Click on the newly created tag.

Figure 16.

4. Click *Internal* next to the `Source` parameter and select the ROC device; PLC2 in this example. The Select TLP popup appears, as shown in Figure 17.
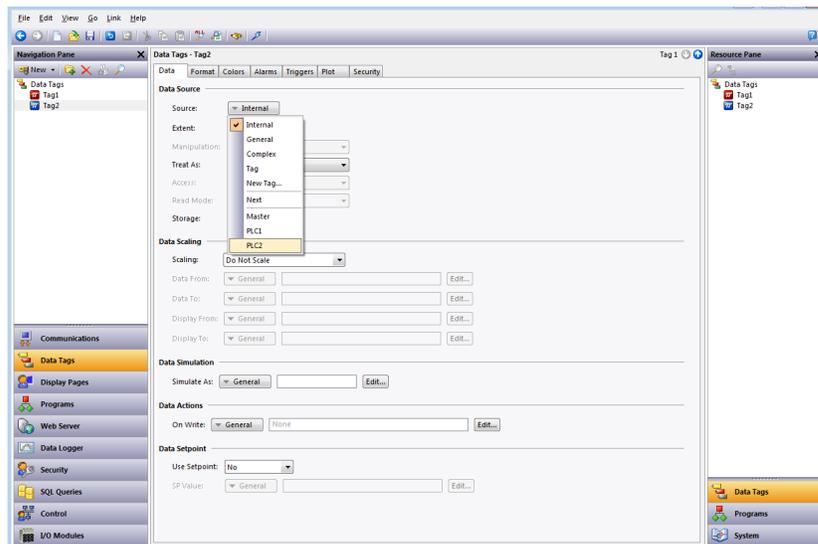

Figure 17.

5. Referring to Figure 18:

    a. Choose the appropriate `Point Type`, `Logical Number`, `Parameter`, and the associated `Data Type` settings; 7-AGA Flow Parameters, 1, and 51-Meter Value in this example.

    <u>Or</u>

    b. Check the Custom TLP option and enter the `Point Type`, `Logical Number`, `Parameter`, and the associated `Data Type` settings.
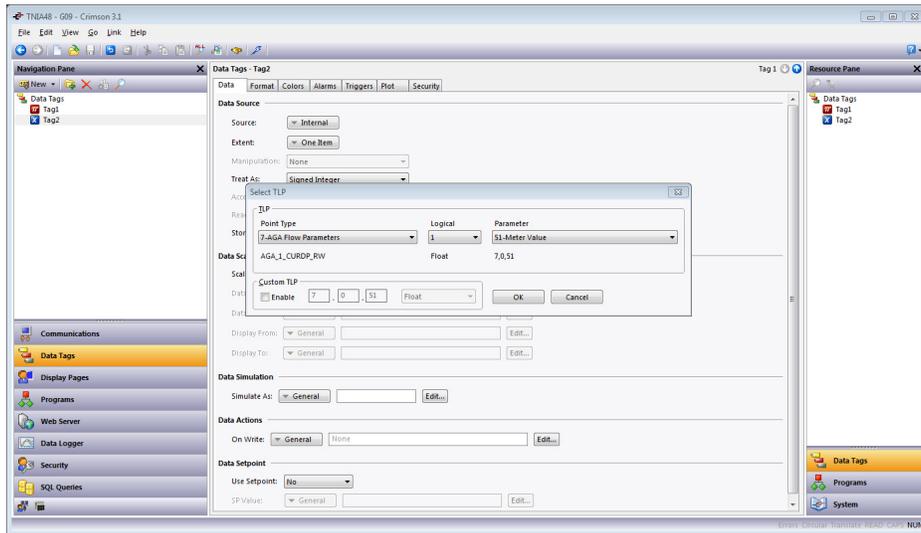

Figure 18.

6. Click *OK*.

Special Tag Mapping Considerations

Strings

When accessing items with a data type of "String", the item should be assigned to a string tag and the tag's Packing should be set to ASCII BigEndian. Since packing is used, the string length must be set to the selected text length or the selected text length + 2 when the selected text length is not divisible by 4. If the string tag is not initially set to the proper length, the communication blocks may need to be rebuilt. This can be accomplished by running the Rebuild Comms Blocks utility available in Crimson's File->Utility menu. Care should be taken when writing to string items. In the event that the current string value contains the same first 4 or more characters as the desired string value, the string should be cleared before writing the desired string.

64-Bit Data

Access to 64-bit double values is possible by using data arrays in conjunction with user functions provided in Crimson; Version 3.0 and later. Simply map all parameters of type double to a numeric tag array. The array selection is available in each tag. Then use the following user functions to get and set double values, respectively:

1) `cstring AsTextR64(Data)` - Where `Data` is the first element in the array of the double value that will be passed as a string.

2) `void TextToR64(Input, Output)` - Where `Input` is a string representing a double value and `Output` is the first element in the array of the double value to be set.

Note: Other 64-bit math functions are also available. Please review the Crimson 3.0 Reference Manual.

Time Parameters

Use Crimson's Time and Date primitive or format type when displaying the target device's Current Time (Parameter 8 of Point Type 12, Parameter 7 of Point Type 136) or any other Time data type.

Disclaimer

It is the customer's responsibility to review the advice provided herein and its applicability to the system. Red Lion makes no representation about specific knowledge of the customer's system or the specific performance of the system. Red Lion is not responsible for any damage to equipment or connected systems. The use of this document is at your own risk. Red Lion standard product warranty applies.

Red Lion Technical Support

If you have any questions or trouble contact Red Lion Technical Support by emailing support@redlion.net or calling 1-877-432-9908.

For more information: http://www.redlion.net/support/policies-statements/warranty-statement