

Emerson Process ROC Protocol Enhanced Master Driver v3.0 Information Sheet for Crimson v3.0+

Compatible Devices

All Emerson Process devices supporting ROC or ROC Plus protocols.

Verified Device

ROC827L (Please note that driver is not limited to ROC800 devices.)

Overview

The Emerson Process ROC Protocol Enhanced Master Driver provides a simple UI with few limitations. Data types are user selectable to fully support many User Defined Parameters.

This "enhanced" ROC driver has been written to be much more versatile than the existing driver and allow the user to define any point type supported by any Emerson Process device that itself supports the ROC protocol. Existing predefined point types from the previous driver have been merged with the enhanced driver, however. A list of these predefined point types is provided below in this document.

A dedicated Ethernet version of the driver is provided; virtual ports are no longer required for Ethernet communications.

Please review the following information to achieve successful communications.

Device Configuration

A security section is provided within the device configuration area in Crimson. Please specify the Operator ID and Password that the Red Lion device should use when communicating with the Emerson device. Also, indicate the Access Level if this security feature should be used.

The Settings group within the device configuration should be reviewed for each device. Note that when communicating with a remote ROC device using Network Radio Modules (NRM), the transaction timeout will likely need to be increased from the default for reliable communications.

The first field provides the ability to select a Ping TLP. This TLP should contain only 1 byte of data (UINT8) and should always be available in the target device. This value should be entered in the format of TTLLPP, where TT is the Point Type, LL is the Logical Number and PP is the parameter in hexadecimal form.

The Increment By field in the settings section allows the user to choose whether to increment 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 will be useful for a single application, it is possible to create 2 identical devices for the Emerson Process ROC Protocol Enhanced driver with only the Increment By parameter set uniquely.

When using the Ethernet version of the driver, be sure to review the settings in the Protocol Options group. In particular, the IP address will need to be configured to match the address configured for the ROC device on the network. Also verify that the correct TCP port is configured.

A "Purge, Sort and Rebuild" action has been provided in driver v3.10+. This action is most useful when changing the Increment By field at mid development or to optimize communications.

Disabling "Spanning Reads" in the Advanced Settings may be useful as the system will always attempt to optimize communications.

Accessing Data

A TLP selection dialog box is available in Crimson which allows simple input of the Point Type, Logical Number, Parameter Number and data type for each value. If a predefined TLP selection is desired, select "No" for the User Defined TLP Selection field in the device options of Crimson—see below for list of currently available predefined point types. Please review the following notes for successful data access.

Data types of type "TLP" should be assigned to a numeric data tag and should be interpreted as follows: XXTLLPP, where XXTLLPP is an eight

digit hexadecimal number. TT is the Point Type, LL is the Logical Number and PP is the Parameter Number.

When accessing items with a data type of "String", the item should be assigned to a string tag and the tags Packing should be set to ASCII Big-Endian. 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.

Access to 64-bit double values is possible by using data arrays in conjunction with user functions provided in Crimson 3.0+. 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.

```
cstring AsTextR64(Data)
```

Where Data is the first element in the array of the double value that will be passed as a string.

```
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.

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

Currently Available Predefined Point Types

0-Configurable Opcode	46-Meter Configuration Parameters	108-Multivariable Sensor
1-Discrete Inputs	47-Meter Flow Values	109-System Analog Inputs
2-Discrete Outputs	48-PID Control Parameters	110-PID Control Parameters
3-Analog Inputs	56-Analog Input Calibration Parameters	113-Orifice Meter Run Configuration
4-Analog Outputs	58-Revision Information	114-Orifice Meter Run Values
5-Pulse Inputs	86-Extended History Parameters	115-Turbine Meter Run Configuration
7-AGA Flow Parameters	91-System Variables	116-Turbine Meter Run Values
8-History Parameters	92-Logon Parameters	122-DS800 Configuration
10-AGA Flow Calculation Values	93-License Key Information	123-Security Group Configuration
12-Clock	94-User C Configuration	124-History Segment Configuration
13-Flags	95-ROC Comm Ports	125-History Segment 0 Point Configuration
15-System Variables	96-FST Parameters	136-ROC Clock
16-FST Registers	97-FST Register Tags	137-Internet Configuration Parameters
17-Soft Point Parameters	98-Soft Point Parameters	138-User Program Configuration
19-Database Parameters	99-Configurable Opcode	139-Smart IO Module Information
21-Information for User Defined Points 1	100-Power Control Parameters	140-AC Input Output Module
41-Run Parameters	101-Discrete Inputs	200-Liquid Preferences
42-Extra Run Parameters	102-Discrete Outputs	201-Liquid Products
43-User List Parameters	103-Analog Inputs	202-Density Interface
44-Power Control Parameters	104-Analog Outputs	203-Liquid Station
45-Meter Calibration and Sampler	105-Pulse Inputs	204-Liquid Meters

Cable Information

G3 RS232 Port	FloBoss 103 LOI Port
2 - Rx	Tx
5 - Tx	Rx
3,4 - Common	5 - Common

G3 RS232 Port	ROC800 CPU RS232C COM2 Port
2 - Rx	2 - Tx
5 - Tx	1 - Rx
3,4 - Common	5 - Common

Revision History

- 08/16/12 – Created
- 08/27/12 – Updated string data access notes.
- 09/11/12 – Added User Defined TLP selection.
- 10/12/12 – Added list of predefined point types.
- 10/18/12 – Clarified “Compatible Devices” and “Verified Device” sections.
- 07/01/13 – Updated string data access notes.
- 01/27/14 – Updated list of predefined point types.
- 01/30/15 – Added note on timeout when using NRMs.
- 02/03/15 – Added “Spanning Reads” and “Purge, Sort and Rebuild” notes.

07/25/16 – Added Ethernet driver.