

Emerson Process Coriolis Meter Master Information Sheet for Crimson v3.0+

Compatible Devices

- RFT9739 Multivariable Field Mount Transmitter
- 9739 MVD Field and Truck Mount Multivariable Transmitters
- CDM Analog
- Model 1500 / LF-Series LFT2
- Model 1500 with the Filling and Dosing Application
- Model 1700 / LF-Series LFT1
- Model 2500 / LF-Series LFT5
- Model 2700 / LF-Series LFT3
- Series 3000 MVD
- Series 3000 MVD NOC(4-Wire)
- Series 3000 NOC(9-Wire)
- Filling Mass Transmitter(FMT)
- HFVM Analog
- FDM Analog
- FVM Analog
- SGM Analog
- GDM Analog
- MVD Direct Connect with Standard Core Processor
- MVD Direct Connect with Enhanced Core Processor
- Model 5700 Config I/O

Verified Device

- Model 5700 Config I/O (RS-485)

Accessible Data

Prefix	Description	Element Size	Access
IC	Input Coils	Bit	R
OC	Output Coils	Bit	R/W
IR	Input Register	16-Bit Word 32-Bit Real 32-Bit Long 64-Bit Real 64-Bit Long	R
HR	Holding Register	16-Bit Word 32-Bit Real 32-Bit Long 64-Bit Real 64-Bit Long	R/W

Access to the Emerson Coriolis string registers can be attained by mapping to a string data tag in Crimson 3.0+. The string length should be set to the maximum number of characters allowed in the Emerson Coriolis. For example, if a string register begins at address 68 and ends at address 71, it consists of 4 registers. Since each register consists of 2 characters, the maximum length is considered to be 8. In this case the string length should be set to a value of 8. Note – It is the programmer’s responsibility to set the string length. If the string length is not set correctly unintended behavior can occur. Set the Packing to “ASCII Big-Endian” so Crimson can know how to convert the data value to a string format.

Some registers in the MMI Driver are meant to be accessed for their individual bit values. For example, “Modbus1”, a register that is in the “Emerson 5700 Config” device is one such register. To access the individual bits of this register, create a Flag Tag in Crimson and pick the desired register, “Modbus1” in this example. In the “Treat As:” dropdown, select Bit Array Big Endian, then choose the applicable bit in the “Bit Number:” field.

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.

Cable Information

From Emerson Device(RS485)		Connector Pinout(8-pin)	HMI
		RJ45	RS485 Comms Port
(A)		1	TxB
(B)		2	TxA
		3	RxA
		4	RxB
		5	TxEN
(C)		6	COMM
		7	TxB
		8	TxA

Revision History

9/19/16 – Created

10/14/16- Added 64-bit access note

2/21/17- Added notes on accessing Strings and Bit Access