

Form A6137

Part Number D301183X012

January 2003

Modbus Host User Program

(for FloBoss 103 Flow Managers)

User Manual

Flow Computer Division

Website: www.EmersonProcess.com/flow



Modbus Host User Program User Manual

Revision Tracking Sheet

January 2003

This manual is periodically altered to incorporate new or updated information. The date revision level of each page is indicated at the bottom of the page opposite the page number. A major change in the content of the manual also changes the date of the manual, which appears on the front cover. Listed below is the date revision level of each page.

Page	Revision
All Pages	01/03

FloBoss and ROCLINK are marks of one of the Emerson Process Management companies. The Emerson logo is a trademark and service mark of Emerson Electric Co. All other marks are the property of their respective owners.

© Fisher Controls International, LLC. 2003. All rights reserved.

Printed in the U.S.A.

While this information is presented in good faith and believed to be accurate, Fisher Controls does not guarantee satisfactory results from reliance upon such information. *Nothing contained herein is to be construed as a warranty or guarantee, express or implied, regarding the performance, merchantability, fitness or any other matter with respect to the products, nor as a recommendation to use any product or process in conflict with any patent.* Fisher Controls reserves the right, without notice, to alter or improve the designs or specifications of the products described herein.

Modbus Host User Program User Manual

TABLE OF CONTENTS

Table of Contents	iii
Section 1 – Getting Started.....	1-1
1.1 Overview.....	1-1
1.2 Additional Information	1-1
1.3 Organization of Manual.....	1-1
1.4 Modbus Protocol Host User Program Overview	1-2
1.5 User Program Names and Versions.....	1-4
Section 2 – Modbus Configuration	2-1
2.1 Modbus Modem.....	2-1
2.2 Modbus Configuration.....	2-3
2.3 Modbus Functions.....	2-6
2.4 Modbus Host.....	2-10
2.5 Sending a Message.....	2-16
Section 3 – Problem Solving	3-1
3.1 FloBoss 103 or Slave Error Messages	3-1
3.2 FloBoss or Slave Responds with Wrong Data in Response	3-1
3.3 FloBoss Appends a Character to Message.....	3-1
3.4 FloBoss Does Not Respond in Point to Multi-Point Comm System	3-1
3.5 Integer to Float Conversion	3-2
3.6 FloBoss Response Preceded By Modem Commands	3-2
Appendix A – Point Types	A-1
A.1 Point Types and Data Types	A-1
Appendix B – Configuration Worksheets	B-1
Index	I-1

Modbus Host User Program User Manual

Modbus Host User Program User Manual

SECTION 1 – GETTING STARTED

1.1 Overview






The Modbus Protocol Host Emulation User Program is designed to allow the FloBoss™ 103 Flow Managers, version 1.10 or greater, to emulate the communications protocol used by Modbus Host devices. This makes it possible to integrate the FloBoss 103 units and Modbus devices into the same Host/Slave system.

- ❖ **NOTE:** Modbus Slave capabilities are standard in the firmware, while the use of this user program allows host capabilities.

This manual describes how to configure and use the Modbus Protocol Host Emulation Program, which is configured by using the ROCLINK™ 800 Configuration Software version 1.10 or greater.

1.2 Additional Information

This document is intended to be used in conjunction with the following manuals and Internet Web sites to assist you. The physical aspects of the FloBoss units are contained in their respective instruction manuals.

-  *ROCLINK 800 Configuration Software User Manual (Form A6121)*
-  *FloBoss 103 Flow Manager Instruction Manual (Form A6114)*
-  *ROC Protocol User Manual (Form A4199)*
-  *Modicon Modbus Protocol Reference Guide (January 1985), form PI-MBUS-300 Revision B available from Modicon/AEG Schneider Automation*
-  www.modicon.com Web site.

1.3 Organization of Manual

Section 2 – Modbus Configuration describes configuration of the Modbus Modem, Configuration, Function, and the Host.

Section 3 – Problem Solving provides information detailing response failures, error messages, responses containing invalid data, multi-point communication errors, and other conditions.

Appendix A – FloBoss Point Types provides information detailing Point Types, Data Types, Point Type Parameters, Opcodes, and User Defined Point Types.

Appendix B – Configuration Worksheets provides worksheets to help you configure and maintain a record of your Modbus Host/Slave network.

Topical Index –Lists alphabetically the items contained in this manual, including the section and page number.

Modbus Host User Program User Manual

1.4 Modbus Protocol Host User Program Overview

The Modbus Protocol Host User Program allows a FloBoss 103 to emulate the Modbus Protocol Host. This makes it possible to enable a FloBoss to be a Host Modbus device. FloBoss units may communicate directly using the Local Operator Interface (LOI) Local Port or remotely via modems using the Comm ports. The Slave is built-in to the FloBoss 103 firmware. Refer to the *ROCLINK 800 Configuration Software User Manual (Form A6121)*.

Modbus allows a FloBoss 103 Host to access and request information from the Slave device. You may also tell the FloBoss to perform an action such as changing the Status of a Discrete Output (coil) from OFF to ON. Depending on the Host request, the Slave (FloBoss or other device) may or may not return a message. Modbus is capable of sending a request to all devices attached to the communications network. This type of mass request is called a Broadcast. Only one Host device may poll for data while the other Slave devices respond to the message sent by the Host.

If the Modbus Host is running, it cannot function as a Slave device at the same time on the same communication port. The Modbus Host must be turned off before it can act as a Slave device. However, the Slave operation will work on any other communications port on which the Modbus Host is not active.

The Modbus **Host** requires the following to perform a request or action:

- ◆ **Address** – Location Address of the Modbus Slave device, such as a FloBoss, to be polled (ROC > Information). Each Modbus Slave has a unique Address.
- ◆ **Function Code** – Informs the FloBoss of the action or request to perform.
- ◆ **Data** – Any additional data required by the FloBoss to perform the action or request specified by the Function Code.
- ◆ **Error Checking** – Informs the Host if the action or request was successful.
- ◆ **Reply Message** – After the message is sent and the Slave performs the action or request, the Host receives a reply message containing the information it requested.

The **Slave** device receives the message and performs the following:

- ◆ Recognizes that a message has been addressed to the Slave device's ROC Address.
- ◆ Determines the action or request sent by the Host. This includes the Function Codes and any necessary Data.
- ◆ Acquires the data (information) or performs the action as stated in the message.
- ◆ Creates a reply message.
- ◆ Sends the reply message to the Host containing any information (data) required by the initial message or sends an error code if the action or request failed. Replies are not sent when a Broadcast message is requested.

The Modbus Host User Program is configured and downloaded to Flash memory by ROCLINK 800. The FloBoss uses physical addressing of I/O, while Modbus applications use logical I/O addressing to retain compatibility with Modbus.

The Modbus messages are configured and input / output parameters are referenced in the FloBoss in the same manner as standard Point Type parameters by TLP: Point Type (Type), Logical Number (Lgl#),

Modbus Host User Program User Manual

and Parameter (Param) to define point locations. Each input/output (I/O) Point Type must be individually configured. Refer to Appendix A.

The Modbus Protocol Emulation program Function Codes 1, 2, 3, 4, 5, 6, 15, and 16 use the same command and response format as the functions listed in the *Modicon Modbus Protocol Reference Guide (January 1985)*, form PI-MBUS-300 Rev B. Depending upon the Parameter configured (Byte, Word, Double Word, or Floating Point); Function Codes 3, 4, and 6 can contain more Data bytes than the other Function Codes. The Data byte upper limit is 240 bytes.

Table 1-1 provides details of the Modbus Function Codes supported by the FloBoss.

Table 1-1. Modbus Function Codes

Slave Function Codes	Meaning	Request Data from Slave Action	Valid TLP
01	Read Logic Coil Status	Obtain current Status (ON/OFF) of a group of outputs.	Type and LGL#
02	Read Discrete Input Status	Obtain current Status (ON/OFF) of a group of inputs.	Type and LGL#
03	Read Output Holding Registers	Obtain current binary value in one or more output holding registers.	Type, LGL#, and Param
04	Read Input Registers	Obtain current binary value in one or more input registers.	Type, LGL#, and Param
Host Function Code	Meaning	Transmit Data to Slave Action	Valid TLP
05	Force Single Logic Coil	Force logic output coil to a Status of ON or OFF.	Type and LGL#
06	Preset Single Holding Register	Place a specific binary value into an output holding register.	Type, LGL#, and Param
15	Force Multiple Logic Coils	Force a series of consecutive logic output coils to a defined ON or OFF Status.	Type and LGL#
16	Preset Multiple Holding Registers	Place specific binary values into a series of consecutive holding registers.	Type, LGL#, and Param

1.4.1 Data Link

The data link between the Host device and the FloBoss include:

- ◆ Direct Link: LOI – Local Port.
- ◆ Direct Link: EIA-485 (RS-485) – Comm Port 1.
- ◆ EIA-232 (RS-232) Serial Communications Card – Comm Port 2.
- ◆ EIA-485 (RS-485) Serial Communications Card – Comm Port 2.
- ◆ Dial-Up Modem Communications Card – Comm Port 2.

Modbus Host User Program User Manual

1.5 User Program Names and Versions

Program versions and names display in Table 1-2. Only one user program is required.

Table 1-2. Versions and Program Names

Program Name	FloBoss Type	User Task	Archive Point	Code Blocks	Data Blocks
Mb103_0.bin	FloBoss 103 Version 1.10 or greater	LOI	26, 27, 28, 29	7D0000 - 7DFFFF	474000 - 476FFF
Mb103_1.bin	FloBoss 103 Version 1.10 or greater	COM1	33, 37, 38, 39	7E0000 - 7EFFFF	478000 - 47AFFF
Mb103_2.bin	FloBoss 103 Version 1.10 or greater	COM2	32, 34, 35, 36	7F0000 - 7FFFFFFF	47C000 - 47EFFF

❖ **NOTE:** Mb103_2.bin is required for modem communications.

1.5.1 Downloading the User Program

To download the user program:

1. **Connect** the FloBoss 103 to ROCLINK 800 software through the **LOI** port.
2. Select Utilities > **User Program Administrator**.

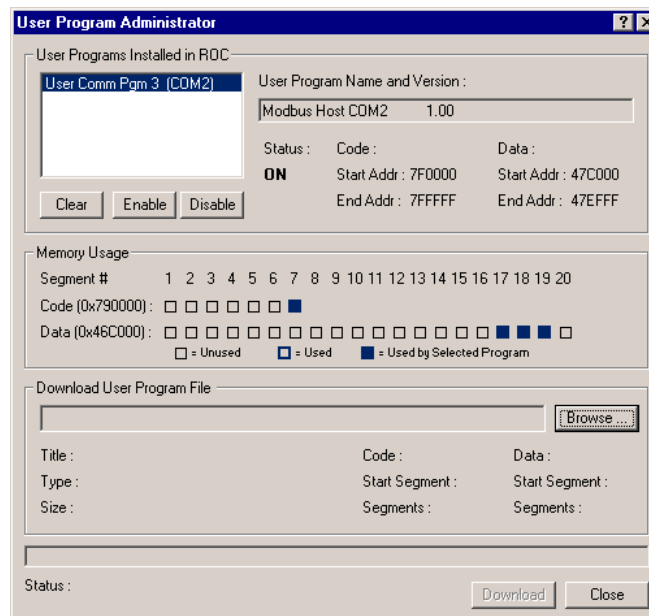


Figure 1-1. User Programs

3. Determine the **Memory Usage**. The memory location must be available for the specific user program that you are loading.

Modbus Host User Program User Manual

4. Click **Browse** and select the user program file to download from the CD-ROM. Program files are typically located in the \Program Files folder. The Open File dialog displays the names of all the files that have the .BIN extension.
5. Select the **File name** you desire to load and click **Open**. Refer to Table 1-2 on page 1-4.
6. Click **Download** to download the selected user program. Only install one user program.
7. Select the user program in the User Program Installed in ROC field, click **Enable**. The Status field displays ON or OFF indicating the Status of the user program.
8. Select System > **Flags**. Refer to Figure 1-2.

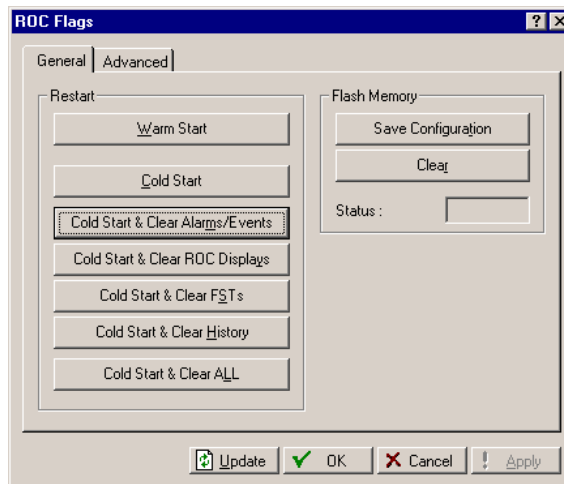


Figure 1-2. ROC Flags

10. Click Flash Memory **Save Configuration** and click **Yes**. This ensures that the program automatically restarts after a Cold Start. Refer to Figure 1-2.
11. Click **OK**.
 - ❖ **NOTE:** You can enable security for the communications port using the ROC > **Security** options.

Modbus Host User Program User Manual

SECTION 2 – MODBUS CONFIGURATION

The Modbus Protocol Emulation Program functions are configured using ROCLINK 800 configuration software. Use the following screens to configure the Modbus Host User Program:

- ◆ **Modbus Modem** – Modem configuration parameters for the Host communication port (Comm Port 2 only).
- ◆ **Modbus Configuration** – Configures the Modbus parameters. Configure both Host and Slave devices to have similarly configured parameters.
- ◆ **Modbus Function** – Corresponds to the same communications port being configured in the Modbus Host screen of the Host that is to poll the Slave device.
- ◆ **Modbus Host** – Sets up the Modbus configuration parameters to allow the FloBoss to act as a Host. This corresponds to the same communications port being configured in the Modbus Functions screen of the Slave to be polled.

2.1 Modbus Modem

Use the Modbus Modem (Comm Port 2 only) screen to configure modem parameters for the Host's communication port. The Modbus program initializes the modem upon power-up, Host modem dial-up, and modem disconnect. Upon initialization, commands are sent to the modem to configure it for proper operation.

1. Select Configure > User Data > **Modbus Modem**. Refer to Figure 2-1.

The screenshot shows a software window titled "Modbus Modem CDM2". At the top left, there is a "Point Number" dropdown menu with "1" selected. Below it are two input fields: "Status" and "ModType 1=Ext 2=Int", both containing the value "0". The main area contains six rows of configuration fields, each with an "RTU Addr" field and a "Connect Command" field, all currently set to "0". At the bottom of the window, there are four buttons: "Update" (with a refresh icon), "OK" (with a green checkmark), "Cancel" (with a red X), and "Apply" (with a downward arrow icon).

Figure 2-1. Modbus Modem

Point Number indicates the Host communications port being configured and is dependent upon which user program is installed.

Modbus Host User Program User Manual

Status reflects codes returned from the modem:

- ◆ 0 = OK
- ◆ 1 = Connect
- ◆ 2 = Ring

The **ModType 1=Ext 2=Int** field indicates the location of the modem as externally (1) or internally (2). Use **ROC > Comm Ports** to configure the modem.

2. Enter the **RTU Address** of the Slave device to be associated with the Connect Command. Up to six different Modbus Slaves can be dialed up through one Communication Port. The RTU (Remote Terminal Unit) Address can have values between 1 and 255.

When attempting to dial, the RTU Address specified in the Modbus Host request is compared to the RTU Address in Modem control screen and determines the Connect Command to execute. The RTU Address is then matched with the FloBoss Address (**ROC > Information**). If the Host RTU Address matches the Modem RTU Address, the Connect Command is used to connect to the modem at the FloBoss Address that is the same as the RTU Addresses. If the Host RTU Address does not match any configured Modem RTU Address fields, the first Connect Command is used.

3. Enter the **Connect Command** (telephone number) to be sent to the Slave device. Modem configuration is based upon the Hayes AT command set.

❖ **NOTE:** Consult your modem's user manual for the valid AT command set; otherwise, the modem may not operate as intended.

Modbus Host User Program User Manual

2.2 Modbus Configuration

Use the Modbus Configuration screen to set up Modbus parameters. Configure both Host and Slave devices to have similarly configured parameters.

1. Select Configure > User Data > **Modbus Config**. Refer to Figure 2-2.

Figure 2-2. Modbus Configuration

Point Number indicates the Host communications port being configured and is dependent upon which user program is installed.

2. Enter the transmission format as **ASCII (0)** or **RTU (1)**. The data link can operate in the RTU (Remote Terminal Unit) mode or the ASCII (American Standard Code for Information Interchange) mode.

❖ **NOTE: All devices in the same communications network must be configured with the same mode of transmission.**

RTU mode allows for greater character density and better data throughput than ASCII for the same baud rate. Each message is transmitted in a continuous stream. Data is sent in 8-bit binary characters. RTU mode uses Cyclic Redundancy Check (CRC) error checking. By default, RTU is enabled.

ASCII mode represents each 8-bit byte of data as two ASCII characters that are the hexadecimal representation of the value. This allows the messages to be read with the use of a dumb terminal, but uses twice as many characters as the RTU mode. Each character sent is composed of a Start bit, 7 or 8 Data bits, and one or two Stop bits with Even, Odd, or No parity. ASCII mode uses Longitudinal Redundancy Checking (LRC) error checking.

In either mode, ASCII or RTU, a Modbus message is placed by the transmitting device into a frame that has a known beginning and ending point.

Modbus Host User Program User Manual

ASCII Message Framing

Begin of Frame	Address	Function	Data	LRC Error Check	End
:	2 Characters	2 Characters	N Characters	2 Characters	CRLF

RTU Message Framing

Begin of Frame	Address	Function	Data	CRC Error Check	End
T1-T2-T3-T4	1 Byte	1 Byte	N * 1 Byte	2 Bytes	T1-T2-T3-T4

3. Enter the **Byte Order** of data bytes in a message transmission or request. This only affects the Data field of a Modbus message and has no effect on data for Function Codes 01, 02, and 05.
 - ◆ **Least Significant Byte First (0)** – Places the Least Significant Byte first.
 - ◆ **Most Significant Byte First (1)** – Places the Most Significant Byte first.
4. Enter 1 in the **Log Enable** field to log all parameter changes made via Modbus to the Event Log.
5. Enter 1 in the **Host Active** field to initiate a transmission. The Host Active Flag remains set while the transmission is in progress and then clears (0). If the Continuous Poll is set, the Host Active flag remains set (1).
6. Enter 1 in the **Continuous Poll** field to activate polling of the Slave devices. If the Continuous Poll is set, the Host Active flag remains set (1).
7. Enter a value in seconds in the **Poll Timer in Sec** field to configure for how many seconds the Slaves are polled.
8. Enter a value for the **Table Start** for the Modbus Address Table. Certain Modbus Hosts request the Address Table from the FloBoss for configuration of the Host. Normally, during initialization, the Host reads the Address Table to determine where the Input and Output data tables are located, and to determine the 100% and 0% values for all analog data.

Function Code 03 allows the transmission of the Input Data Start register, Output Data Start register, HI Integer Scale, and LO Integer Scale values. The Address Table normally resides at the start of the Modbus registers, either at location 0 or 40000.

9. Enter the **Input Data Start** for the starting register address for the block of registers of the input data table.
10. Enter the **Output Data Start** for the starting register address for the block of registers of the output data table.
11. Enter the **HI Integer Scale** values for the 100% (High Reading EU) value for all analog data registers.
12. Enter the **LO Integer Scale** values for the 0% (Low Reading EU) value for all analog data registers.

In the FloBoss, the endpoints of the Analog Inputs and the Analog Outputs are used to scale or calibrate the range of the input or output signal. Because each I/O point can have different scaling, the Raw Values from the Analog I/O points are normalized to the values defined by the HI Integer Scale and LO Integer Scale fields.

Modbus Host User Program User Manual

These values are signed integers ranging from zero to 32767. The default signal range is 800 to 4000, which represents a 1 to 5 volt or 4 to 20 milliAmp signal. These data fields can also be used to scale the Analog I/O to integer values with an implied decimal point.

For example, all Analog I/O Raw values can be transmitted with 0 to 1000 values (0 to 100.0, decimal point implied) by setting the values in this field to 0 for the LO Integer Scale and 1000 for the HI Integer Scale. The scaling is used only on Analog I/O specified by I/O Point Type 3 (AIN), Parameter 17 (Raw A/D Input), and I/O Point Type 4 (AO), Parameter 9 (Raw D/A Output).

13. Enter the **HI Float Scale** and **LO Float Scale** values. The HI Float Scale and LO Float Scale fields are normally used when the Host is not able to process floating point numbers. Eight sets of floating point ranges for values can be specified. This allows floating point values, such as PID Setpoints, to be read and set by the Host as integer values. The FloBoss units floating point values are converted to integers by configuring a register or range of registers with the Conv (conversion) fields set in the Modbus Function configuration screens. The equations used to convert floating point to integer values are:

$$\begin{aligned}\text{Float Range} &= \text{HI Float Scale} - \text{LO Float Scale} \\ \text{Integer Range} &= \text{HI Integer Scale} - \text{LO Integer Scale} \\ \text{Adjusted Reading} &= \text{Float Reading} - \text{LO Float Scale} \\ \text{Integer} &= \frac{\text{Integer Range} \times \text{Adjusted Reading}}{\text{Float Range}} + \text{LO Integer Scale}\end{aligned}$$

The equations used to convert integers to floating point values are:

$$\begin{aligned}\text{Float Range} &= \text{HI Float Scale} - \text{LO Float Scale} \\ \text{Integer Range} &= \text{HI Integer Scale} - \text{LO Integer Scale} \\ \text{Adjusted Integer} &= \text{Integer Sent} - \text{LO Integer Scale} \\ \text{Float Value} &= \frac{\text{Adjusted Integer} \times \text{Float Range}}{\text{Integer Range}} + \text{LO Float Scale}\end{aligned}$$

The HI Float Scale and LO Float Scale values are also used as high and low deadband limits when Convert Code 49 through 56 is selected. Refer to the Modbus Functions on page 2-6.

14. Enter 1 in the **Enable Debug** field to enable debugging of Modbus communications. When an LCD (liquid crystal display) is installed, the debug messages appear on the LCD.

The top line of the LCD indicates which Modbus RTU Address is currently being polled. The bottom line indicates the action that the user program is performing:

- ◆ **SEND** – FloBoss is sending a Modbus message to the device RTU Address displayed on the top line.
- ◆ **RCVED** – FloBoss has received a valid response from the device RTU Address displayed on the top line.
- ◆ **TOMSG** – FloBoss timed out waiting for a response from the device RTU Address displayed on the top line.

Modbus Host User Program User Manual

The bottom line also indicates connection states when using an External or Internal modem.

- ◆ **CONC** – FloBoss is attempting to establish a connection with the remote device.
- ◆ **WAITC** – FloBoss is waiting for a connection.
- ◆ **TMOCT** – FloBoss has timed out waiting for a connection.
- ◆ **DISCC** – FloBoss is disconnecting from a remote device.

2.3 Modbus Functions

Modbus Functions correspond to the same communications port being configured in the Modbus Host screen of the Host that is to poll the Slave device.

When a message is received from the Host, the Slave device:

1. Reads the RTU Addr from the Host Config screen sent from the Host and matches it with the FloBoss Address (ROC > Information screen) of the Slave.
2. If the message is for that Slave device or is a Broadcast message, the Slave reads the Function Code (Fnc Num) so that it knows what action to perform. If the RTU Addr does not match the FloBoss Address, the message is ignored.
3. Next, the Slave searches the Function configuration table (Figure 2-3) for the Modbus register numbers requested (Reg# through #Regs), starting with the first table (first Tag ID) entry down to the last (last Tag ID).
4. If a register number match is found ($Start \geq Reg\#$ through $End\ Addr \leq \#Regs$), it builds a response based on the TLP configured in the table. If no register number match is located, an error message (Exception Response) returns. The User Program locates a register number as long as it matches the beginning register number (Start), the ending register number (End Addr), or any number in between for that particular entry in the table.
 - ❖ **NOTE:** TLP is the Point Type (Type), Logical Number (Lgl#), or Parameter (Param)
5. A response message containing the data requested or an error message is then sent back to the Host. When the data response message is returned, the data collected from the Slave is stored beginning at the register defined in the Save# field of the Modbus Host configuration screen.

When configuring the table, register numbers (Start through End Addr) and table names (Tag ID) should be unique. Register numbers may be duplicated as long as they are located in a separate Modbus Function configuration table. For example, The Start register 500 may appear in Function table “Tag ID - Funct 1” and Function table “Tag ID - Funct 2.” If a register number is duplicated within the same Modbus Function table, the first occurrence will be used. It is best to number the tables from the smallest register number to largest especially when using two table entries to configure a continuous group of register numbers.

Up to thirty different requests can be configured for Modbus Functions 3, 4, and 16 (three displays of ten register number ranges). By making the register numbers continuous, meaning the Start address of a new line is one greater than the End Addr (end address) of the previous line, a continuous data table can be created up to the limit of 240 bytes. This type of data table allows access to all data with one request.

Modbus Host User Program User Manual

Up to ten different requests can be configured for Modbus Function Codes 1, 2, 5, 6, and 15. For Function Codes 1, 2, 5, and 15, the parameter specified should be a single-byte parameter type, preferably a Status parameter. Each **address span** (Start through End Addr) must be unique within the Function Code for proper operation. If not, the first valid address is used. A 0 in the Type field disables that Function Code.

- ❖ **NOTE:** When you specify a reference to a Parameter within another Point Type, you must interpret the information returned on the display (for example, SPT, 3, DATA2) in the following manner. The first part is a three-character mnemonic (in this example, “SPT” means Softpoint) that indicates the Point Type. The second part (such as “3”) indicates the Point Number, but keep in mind that for all point types except I/O points, you need to add “1” to the displayed Point Number to get the actual Point Number (for example, 3 + 1 = 4). The third part is a mnemonic indicating the selected Parameter (such as DATA2).

To configure Modbus Functions:

1. Select Configure > User Data > **Modbus Funct.** Refer to Figure 2-3.

	Start	End Addr	Type	Lg#	Param	Conv
1	0	0	0	0	0	0
2	0	0	0	0	0	0
3	0	0	0	0	0	0
4	0	0	0	0	0	0
5	0	0	0	0	0	0
6	0	0	0	0	0	0
7	0	0	0	0	0	0
8	0	0	0	0	0	0
9	0	0	0	0	0	0
10	0	0	0	0	0	0

Figure 2-3. Modbus Functions

Point Number indicates the Host communications port being configured and is dependent upon which user program is installed. **Tag ID** displays the tag identification indicates the Point Number.

2. Enter the **Start** address that represents the first data register in the address span. In certain Modbus Host devices, the register 40101 is actually transmitted as “100.” The value “100” should be placed in the Start address field since the FloBoss uses the actual number sent by the Host. A Start address of 0 is a valid address. Any number from 0 to 65535 is valid. Register numbers can be duplicated as long as they are in separate Modbus Function configuration tables. The tables should be numbered from smallest to largest.

Modbus Host User Program User Manual

For example, the Host device requests the Start address (Reg#) 500 through End Addr (#Regs) 700. The Slave Start address is 400 and the End Addr is 700. All register numbers requested by the Host (500 through 700) are valid and would be responded to because the Reg# through #Regs numbers match or fall in between the Start and End Addr register numbers (400 through 700). The Reg# and #Regs fields are defined in the Modbus Host configuration screen.

The Modbus Host register numbers must be valid requests or an error will result. Refer to Table 2-1.

Table 2-1. Valid Host Register Numbers

Modbus Functions	Modbus Host
Start	$\geq \text{Reg\#}$
End Addr	$\leq \text{\#Regs}$

3. Enter the **End Addr** (End Address) that represents the last register number of a Point Type's data. The value for this number is computed by:

$$\text{End Address} = (\text{Start address} + \text{Number of Functions}) - 1$$

4. Enter the Point **Type** to set or acquire data. Point Type data is associated with an address (Start through End Addr). When the Host requests a valid range of register numbers, the Function Code tells the Slave what to do and between which registers (Start through End Addr). The Type, Logical Number (Lgl#), and Param fields define what data is collected or to what value to set the Point Type register numbers. The Type field relies on the Logical Number (Lgl#) field to determine if the Point Type increments (horizontal indexing) as data is gathered or set.

For example, Point Type 47 is used to indicate the Meter Flow Values. To read the Energyrate per Day, you would select Type 47 and Parameter # 1 (Param 1).

If the Type field is set to **0**, the line in the configuration table is considered invalid and skips when processing a request. Appendix A has a complete list of parameters for each of the ROC Point Types.

5. Enter the **Lgl#** (Logical Number) to specify where to start collecting the Point Type data and, if necessary, whether to increment the Point Type or the Parameter. The Logical Number field specifies the Logical Number to be assigned to the Start address. When the Host requests a valid range of register numbers, the Function Code tells the Slave what to do and between which registers (Start through End Addr). The Type, Logical Number (Lgl#), and Param fields define what data is collected or to what value to set the register numbers. **The Logical Number is used to increment the Point Type number or the Parameter.**
 - ◆ Point indexing is used to increment the Point Type number. When using point indexing, Logical Number (Lgl#) < 128.
 - ◆ Parameter indexing is used to increment the Parameter within the Point Type. When using parameter indexing, Logical Number (Lgl#) \geq 128.

Modbus Host User Program User Manual

Point Indexing

Point indexing is used to increment the Point Type number. If the range of addresses is greater than 0, the next address (Start address + 1) corresponds to the value specified by Logical Number + 1 (Lgl# + 1). When using point indexing, Logical Number (Lgl#) < 128.

For example, when using horizontal indexing the configuration of:

Start: 100 End Addr: 106 Type: 3 Lgl#: 0 Param: 17 Conv: 0

Specifies seven values (100, 101, 102, 103, 104, 105, and 106) for Parameter 17 of Analog Inputs (Type 3) starting at the first Analog Input (Lgl# 0+1). So register 100 = Parameter 17 of AI Point Number 1, register 101 = Parameter 17 of AI Point Number 2, ..., and register 106 = Parameter 17 of AI Point Number 7.

Another example of point indexing uses the configuration of:

Start: 120 End Addr: 120 Type: 17 Lgl#: 14 Param: 5 Conv: 0

Specifies one value (Start 120 through End Addr 120) for Parameter 5 of a Soft Point (Type 17) starting at the fifteenth Soft Point (Lgl# 14+1). So register 120 = Parameter 5 of Soft Point #15.

Parameter Indexing

Parameter indexing is used to increment the Parameter within the Point Type. Indexing of parameters can also be selected by offsetting the Logical Number by 128. When using parameter indexing, Logical Number (Lgl#) ≥ 128.

For example, when using vertical indexing the configuration of:

Start: 109 End Addr: 114 Type: 16 Lgl#: 128 Param: 2 Conv: 1

Specifies six parameters (109 through 114) for FST number 1 (LGL# 0 + 128) starting at Parameter 2. So register 109 = Parameter 2 of FST Point Number 1, register 110 = Parameter 3 of FST Point Number 1, register 111 = Parameter 4 of FST Point Number 1, ..., and register 114 = Parameter 7 of FST Point Number 1. Conversion (Conv 1) Code 1 (Float to integer, Float Scale 1) is used to convert the floating point value to an integer before the response message is returned to the Host.

For another example of parameter indexing, the configuration of:

Start: 57428 End Addr: 57437 Type: 17 Lgl#: 137 Param: 2 Conv: 0

Specifies ten (57428 through 57437) parameters for Soft Point number 10 (Lgl# 9 + 128) starting at Parameter #2. So register 57428 = Parameter 2 of Soft Point #10, register 57429 = Parameter 3 of Soft Point #10, ..., and register 57437 = Parameter 11 of Soft Point #10.

Another way to explain the Logical Number (Lgl#) field is that when the Logical Number is 128 or greater, the Point Type number remains the same, and the Parameter within that point increments. When the Logical Number is less than 128, the Point Type number increments, and the Parameter number remains the same.

Logical Numbers are zero based. For example, there are 32 Soft Points, with the first being logical 0, and the last being logical 31 for a total of 32. To get the correct Logical Number for the point you wish to access, subtract 1 from the number. Soft Point 5 is Logical Number 4, and Soft Point 27 is number logical 26, Analog Input 3 is number logical 2, and so on.

Modbus Host User Program User Manual

Modbus protocol supports 32 Soft points, so the valid Soft Point Logical Numbers are 0-31, and 128-159.

6. Each Type has multiple Parameters associated with the Point Type. Enter a value in the **Param** field to specify the starting Parameter of the Point Type and relies on the Logical Number (Lgl#) field to determine if the Parameter is incremented using vertical indexing. When the Host requests a valid range of register numbers, the Function Code tells the Slave what to do and between which registers (Start through End Addr). The Type, Logical Number (Lgl#), and Param fields define what data is collected or to what value to set the register numbers.

For example, Point Type 14 represents the Communication Ports. To set the Retry Counter to a higher value, you would select Type 14 and Parameter # 13 (Param 13). Be aware of the different data types (Character, Integer, Long, Float) and the size of the data types. See Appendix A for the complete list of the ROC Point Types and Parameters.

7. Enter the **Conv** field to specify the type of conversion required, if any, on the data before it is sent to the Host or before it is written to the FloBoss. **The conversions are used to allow integer values and floating point values to be converted, transmitted and received.** Convert Codes reorganize the data bits into a specific order the Host, Slave, or ROC can understand. Refer to the *ROCLINK 800 Configuration Software User Manual (Form A6121)* concerning Convert Codes used with the Modbus Protocol user program. The Function Codes supported display under the Slave and Host columns.

2.4 Modbus Host

Modbus Host mode of operation allows the FloBoss to simulate a Host device that can poll other Slave devices for data and to store the data for parameter updates, for use in FST program registers, and for user programs. The FloBoss can also send commands to set outputs and to write data to a Slave device. The Modbus Host mode is enabled by the **Host Active** field on the Modbus Config configuration display as in Figure 2-2. By default, Host Active is always enabled. The Host Active field (1) is what initiates a host transmission. Refer to Sending a Message on page 2-16.

Each Modbus Host configuration screen has a direct correlation to each Modbus Function screen. For example, Modbus Host COM1 corresponds to Modbus Funct COM1 of the Slave and Modbus Host COM2 corresponds to Modbus Funct COM2 of the Slave.

Up to forty different Host requests or commands can be configured for a communications (COM) port. Each request or command can transmit or receive up to 240 bytes of data.

- ❖ **NOTE:** Each Host request configured must have a corresponding entry in the Modbus Functions configuration table.

Modbus Host User Program User Manual

When a Host polls a FloBoss:

1. The Host sends the message to the FloBoss Address (ROC > Information) screen with the corresponding RTU Address.
2. The Host tells the FloBoss to perform the Function Code request or command entered in the Fnc Num field.
3. The Function Code request or command is performed on the data register starting at the data register entered in the Reg# field.
4. If a Slave with a matching address (RTU Addr = FloBoss Address) and the register numbers are valid (Start \geq Reg# through End Addr \leq #Regs), the Slave device performs the request or command for all data registers between the Reg# (starting register) and #Regs (total registers to act upon) values.
5. The Slave responds by sending a message with the results of the request or command or an error if the request fails.
6. The request or command results data bytes are stored in the register beginning with the value entered in the Save# field. The Status field displays the current status of the Host request or command.

Care must be given to insure that there are enough points in the Host to support the points being written to or read from the Slaves. Utilization of Soft Point parameters, unused I/O module parameters, and FST parameters may be required to store all of the necessary data. Use the Modbus Host Configuration worksheet in Appendix B of this manual to assist in documenting configurations.

To configure the Modbus Host:

1. Select Configure > User Data > **Modbus Host**. Refer to Figure 2-3.

	RTU Addr	Fnc Num	Reg#	Save#	#Regs	Status
0.	0	0	0	0	0	0
1.	0	0	0	0	0	0
2.	0	0	0	0	0	0
3.	0	0	0	0	0	0
4.	0	0	0	0	0	0
5.	0	0	0	0	0	0
6.	0	0	0	0	0	0
7.	0	0	0	0	0	0
8.	0	0	0	0	0	0
9.	0	0	0	0	0	0

Figure 2-4. Modbus Host

Modbus Host User Program User Manual

Point Number indicates the Host communications port being configured and is dependent upon which user program is installed. **Tag ID** displays the tag identification indicates the Host table number for the selected Point Number.

2. Enter the **RTU Addr** (Remote Terminal Unit Address) of the Slave device which to poll. The RTU Address can be a number from 0 to 255. This is the FloBoss (Slave) Address set in the ROC > Information screen of ROCLINK 800. The RTU Addr must match a FloBoss (Slave) Address in the same communications network or an error will occur. **RTU Address 0 is used to Broadcast a command to all devices connected to the communications port.** When the RTU Addr is 0 (Broadcast), every Slave device receives the same Function Code and performs the same action, but the Slave devices do not send a response message.
3. Enter the **Fnc Num** to specify the Modbus **Function Code** to be sent to the Slave device. Function Codes tell the Slave device what kind of action to perform. Refer to Table 1-1, Modbus Function Codes.

Function Numbers are entered as decimal values. For example, to set a single coil, enter 05 on the Fnc Num field or the Host desires to send Function Code 02 to a FloBoss. The Host sets the FloBoss Address in the RTU Addr field. The Host sets the Function Code value in the Fnc Num field of the Modbus Host screen.

A Function Number of 0 disables this request or command. Any other Function Numbers not described previously result in an error (Status = 145) and abort the request or command.

4. Enter the **Reg#** starting data register (for this request) in the Slave device. This corresponds to a data register matching or between the Start and End Addr parameters in the Modbus Functions screen. This is the first data byte sent to the Slave and tells the Slave at which register to begin the action associated with the Function Code. A register number is valid as long as it matches the beginning register number (Start), the ending register number (End Addr), or any number in between for that particular entry in the table (Start \geq Reg# through End Addr \leq #Regs). If no register number match is located, an error message (Exception Response) returns.

For example, the Host desires to send Function Code 02 (Read Discrete Input Status) to a FloBoss. The Host sets the FloBoss Address in the RTU Addr field. The Host sets the Function Code 02 value in the Fnc Num field. The Host specifies the starting register in the Reg# field and how many registers to read through in the #Regs field.

5. Enter the **Save#** value. When a message is returned from the Slave, this is the starting register where the data collected from the Slave is stored in the Host. This register number does not need to be the same as the device register number (Reg#). The sequence of data registers following the Save# must be large enough to hold the total number of data bytes sent back to the Host after the action requested by the Function Code has been performed.

Care must be given to insure that the Host has enough points to support the points being written to or read from the Slaves. Utilization of Soft Point parameters, unused I/O module parameters, and FST parameters may be required to store all of the necessary data. Use the Modbus Host Configuration worksheet in Appendix B of this manual to assist in documenting configurations.

Modbus Host User Program User Manual

The Host Save# must be present in the Modbus Functions table for the same communications port you are configuring. For example, if you set the Save# field to 10 in the Modbus Host COM1 screen, the Modbus Funct COM1 screen must have a 10 present in or between the Start and End Addr fields.

The response message is stored beginning at the register entered in the Save# field. To view the returned message, navigate to the TLP location where the data is stored.

6. Enter the **#Regs** to specify the number of registers (data bytes) for the Host to request or set. The Modbus Functions table for the specified number of data bytes must be sized the same as, or greater than, the #Regs field. The values of the Start and End Addr fields in the Slave Modbus Function table must accommodate the number of data bytes set in the #Regs. For example, you cannot request registers Reg# 300 through #Regs 500 if the Start value is 100 and the End Addr value is 200 because the registers are not configured in the Slave Modbus Function table.

A register number is valid as long as it matches the beginning register number (Start), the ending register number (End Addr), or any number in between for that particular entry in the table ($\text{Start} \geq \text{Reg\#}$ through $\text{End Addr} \leq \text{\#Regs}$). If no register number match is located, an error message (Exception Response) returns. Refer to Table 2-2 and Section 4, Exception Responses.

Status displays the status of the Host request or command. Table 2-2 lists the possible status values.

Modbus Host User Program User Manual

Table 2-2. Status of Host Request or Command

Status	Description
0	Inactive or start of transmission. This line in the Host table is inactive or beginning a transmission.
1	Receive timeout error. A slave response was not received. This indicates that the slave did not receive the request message, or the RTU Address within the request message was incorrect. Make sure that the RTU Address of the Host and Slave Address match. Check the cabling between the Host and Slave and make sure that pinouts are correct.
2	Received address check. A byte of the response has been received and the check of the RTU Address is in progress. If the status remains at 2, the wrong RTU Address was received and the program will check the next byte received until a match is found. May indicate that the first byte was missed or there are extra bytes being sent.
3	Received Function Number check. The RTU Address check has passed, the second byte of the response has been received, and the check of the Modbus Function Number is in progress. If the Status remains at 3, the configured Host table Function Number did not match the received Function Number.
4	Number of expected bytes check. The expected number of data bytes in the response message is being checked. If the Status remains at 4. The wrong number of data bytes was received.
5	Receiving Slave response. The RTU Address check, Function Number check, and number of data bytes check have all passed. The data is now being received. If the Status remains at 5, an error occurred reading the data bytes of the response.
6	CRC or LRC check. In RTU mode, the LSB of the CRC has been received. In the ASCII mode of operation, the LRC has been received and the calculated LRC did not match the received LRC,
7	CRC or LRC check. In RTU mode, the MSB of the CRC has been received. If the Status remains at 7, the calculated CRC did not match the received CRC. Not used in ASCII mode of operation.
8	Valid Slave response. The Host request was sent, and a valid Slave response was received and processed.
128	Write ROC date error. An error occurred when writing the data into the Point Type, Logical Number, and Parameter Number (TLP) database.
129	Access FloBoss data error. An error occurred when reading date from the FloBoss TLP database.
130	Host function table error. The line in the Host table request does not match the corresponding function table configuration.
131	Transmit timeout error. The FloBoss was unable to transmit the Host request message. This would indicate a missing communication card, a communication card failure, or a lockup problem, which requires a power cycle to clear.
144	Transmit or receive buffer overflow. The message to be sent or received is too long. Either too many registers are configured in the request message, or the resulting response is too large.
145	Invalid Function Number in request. The line in the Host table is configured with an invalid Modbus Function Number. Valid Function Numbers are 1, 2, 3, 4, 5, 6, 15, and 16.

Status values 0 and 2 through 8 are active on the Host transmission. These values appear for a very short period and then step to the next value if the process is without error. If an error occurs in the step, the value is present until the next transmission is requested. A transmission without error has a Status value of 8. Refer to Section 2.4.1, Exception Response, on page 2-15.

Modbus Host User Program User Manual

2.4.1 Exception Response

When a Host sends a message to a Slave, it expects a normal return response message. Broadcast mode is exempt from return response messages. The following may occur:

- ◆ The Slave device receives the message, processes the action without error, and returns a normal return response message.
- ◆ The Slave receives the message but interprets a communications error in Parity, LRC, or CRC and does return a response message. The Host times out.
- ◆ The Slave does not receive the message due to a communications error and does not return a response message. The Host times out.
- ◆ The Slave receives the message without communications errors, but is unable to process the message such as the request to preset a coil that does not exist; the Slave returns a response message called an exception response indicating the cause of the error.

When a normal return response message is returned, the Function Code echoes the Function Code included with the initial Host to Slave message. Function Codes have a most significant bit (MSB) of 0 with values below 80 hexadecimal.

The Exception Response message is transmitted if the FloBoss detects an error in the received message and causes the return response message Function Code field to be Enabled. This causes the Function Code value in an exception response to 80 hexadecimal thus indicating an error. Refer to Table 2-3. The received Function Code returns with the most significant bit set to signify an error. The exception codes listed in Table 2-4 are for reference only.

When a normal return response message is received by the Host, the Slave returns data results or statistics acquired during the poll. When an error occurs, the exception code displays in the Data field. The exception code details why the error occurred.

Table 2-3. ROC Exception Response Example Message

Message Field	Device Address	Function Code	Exception Code	Error Check
Bytes	1	1	1	2
TX Order				LS MS
Value	As Received	FC + 80H	1 to 3	CRC-16

Table 2-4. Modbus Protocol Errors that Elicit an Exception Response Message

Error Code	Error Detected	Usage
01	Illegal Function Code	Implemented
02	Illegal Data Address	Implemented
03	Illegal Data Value	Implemented

Error Code 01 – Function Code not listed. Refer to Table 1-1.

Error Code 02 – Register value not configured. The Start \geq Reg# through End Addr \leq #Regs are not valid.

Error Code 03 – Wrong number of data bytes sent. Refer to Table 2-2.

Modbus Host User Program User Manual

2.5 Sending a Message

To send a message to a Slave device:

1. Configure the parameters in the Modbus Host screen to indicate to which Slave Address (RTU Addr) the message is to be sent, which Function Code the Slave device is to perform, and to which registers. Refer to Section 2.4, Modbus Host, on page 2-10.
2. Configure the parameters in the Modbus Functions screen to indicate which data is required from the Slave device, from which registers the data will be polled, and any conversion required. Refer to Section 2.3, Modbus Functions, on page 2-6.
3. Configure the ROC > Comm Ports parameters. Refer to *ROCLINK 800 Configuration Software User Manual (Form A6121)*.
4. Configure the Modbus Modem parameters. Refer to Section 2.1, Modbus Modem, on page 2-1.
5. Set Host Active to 1 in the Modbus Config screen to enable Host communications. Refer to Section 2.2, Modbus Configuration, on page 2-3.
6. Set Continuous Poll to 1 in the Modbus Config screen.
7. Set the Poll Time value to the desired polling frequency in seconds.
8. Click Apply.

The message is sent beginning with the Request Number set in the Starting Register (ROCLINK 800). Subsequent Request Numbers are transmitted until a Function Number of 0 is encountered or the last Request Number (39) has been transmitted. If Continuous Polling (ROCLINK 800) is Enabled, the Host waits the amount of time specified and starts the transmission sequence over again.

2.5.1 Controlling Modbus Host by FST

An FST (Function Sequence Table) can be used to schedule Modbus Host requests. If an FST is created to schedule Host requests, the FST should examine the Host Active flag of the Modbus Config Point Type and wait until the request is completed before proceeding. Otherwise, Host requests can be skipped and the desired data transfer may not have completed before proceeding. To initiate a Host transmission, the Host Active flag is set to a 1. Upon completion of the request, the Host Active flag is cleared (0). Wait for this Host Active flag to clear (0) before proceeding.

❖ **NOTE:** The Host Active flag does not clear if the Continuous Poll flag is also set (1).

Using an FST, other scenarios can be developed to control the Modbus Hosts ability to dial other Modbus devices on a regular interval. FST Timers can be set to indicate when it is time for the Modbus Host to call other Modbus devices. Comparing values for status change, limit exceeded, or Deadband exceeded can initiate Modbus Host requests to signal warnings to other Modbus devices.

SECTION 3 – PROBLEM SOLVING

3.1 FloBoss 103 or Slave Error Messages

Error messages are usually caused by the FloBoss or Slave configuration. Check to see that:

1. The proper format (ASCII or RTU) is selected.
2. The Modbus Functions are configured correctly. The Start \geq Reg# through End Addr \leq #Regs are not valid.
3. Enough continuous registers are configured.
4. The requested Type, Lgl#, and Param match the configured Type, Lgl#, and Param.
5. The conversions (Conv) are used correctly.

3.2 FloBoss or Slave Responds with Wrong Data in Response

Wrong data is usually caused by the FloBoss or Slave configuration. Check to see if:

1. The data order is wrong. Check MSB and LSB configuration in the Modbus Config screen.
2. The wrong register numbers are configured. Check the Reg#, #Regs, Start, and End Addr fields.
3. The wrong Parameter is selected. Check the Type, Lgl#, and Param fields.
4. Indexing is not proper. See Section 3, Lgl# field.

3.3 FloBoss Appends a Character to Message

Check to see if a Turnaround Delay or Key Off Delay is configured in the Communication Parameters screen (ROC > Comm Ports) when it is not required.

3.4 FloBoss Does Not Respond in Point to Multi-Point Comm System

Check to see if the FloBoss responds in a point-to-point hook-up. If it does, the problem is associated with delays in the system. Check:

1. The FloBoss Key On Delay (ROC > Comm Ports) has been configured correctly.
2. The FloBoss Turnaround Delay or Key Off Delay has been configured correctly.
3. The Host has configurable Key On Delay and Turnaround Delay or Key Off Delay
4. Each FloBoss must have a unique FloBoss Address.

Modbus Host User Program User Manual

3.5 Integer to Float Conversion

Integer to float conversions may not produce a fractional component. To correct, use a float scale range that more closely corresponds to the range of the integer being converted. Refer to Section 3, Conv fields.

3.6 FloBoss Response Preceded By Modem Commands

If the FloBoss sends modem commands before the transmission of a response to a request, connect the FloBoss DSR signal to the modem's DCD signal or a similar pin indicating modem activity.

Modbus Host User Program User Manual

APPENDIX A – POINT TYPES

A.1 Point Types and Data Types

This appendix details Point Types (Table A-1), Data Types (Table A-2), and User-Defined Point Types for the Modbus Host user program display in Table A-3 through Table A-6.

A.1.1 User Defined Point Types

User Defined Point Types (UDP) are available in FloBoss units. UDPs enable configuration data to be available to ROCLINK 800. The Modbus program for reserves UDPs for the following communication ports (ROC > Comm Ports):

- ◆ **Local Port (LOI)** – UDP 26, UDP 27, UDP 28, and UDP 29.
- ◆ **COM1 Port** – UDP 33, UDP 37, UDP 38, and UDP 39.
- ◆ **COM2 Port** – UDP 32, UDP 34, UDP 35, and UDP 36.

For standard Point Types, refer to *ROC Protocol User Manual (Form A4199)*.

Table A-1. Point Types and Descriptions

Point Types	Description	Point Types	Description
0	Configurable Opcode	34	Modbus Host User Program – COM2
1	Discrete Inputs	35	Modbus Host User Program – COM2
2	Discrete Outputs	36	Modbus Host User Program – COM2
3	Analog Inputs	37	Modbus Host User Program – COM1
4	Analog Outputs	38	Modbus Host User Program – COM1
7	AGA Flow Parameters	39	Modbus Host User Program – COM1
8	History Parameters	41	Run Parameters
10	AGA Flow Calculation Values	42	Extra Run Parameters
12	ROC Clock	43	User Lists Parameters
13	ROC Flags	44	Power Control Parameters
14	ROC Communication Ports	45	Meter Calibration and Sampler
15	System Variables	46	Meter Configuration Parameters
16	FST Registers	47	Meter Flow Values
17	Soft Points Parameters	48	PID Control Parameters
19	Database Parameters	53	Modbus Configuration Parameters
26	Modbus Host User Program – LOI	54	Modbus Function Tables
27	Modbus Host User Program – LOI	55	Modbus Special Function Table
28	Modbus Host User Program – LOI	56	Analog Input Calibration Parameters
29	Modbus Host User Program – LOI	57	Logon Securities Parameters
32	Modbus Host User Program – COM2	58	Revision Information
33	Modbus Host User Program – COM1	86	Extended History Parameters

Modbus Host User Program User Manual

Table A-3. Modbus Modem

Modbus Modem Point Types – 26 (LOI), 32 (COM2), and 33 (COM1)				
Parameter #	Read-Write	Data Type	Length	Description
0	R/W	INT 8	1	Status
1	R/O	UINT 8	1	Modem Type
2	R/W	UINT 8	1	1st RTU Addr
3	R/W	UINT 8	1	2nd RTU Addr
4	R/W	UINT 8	1	3rd RTU Addr
5	R/W	UINT 8	1	4th RTU Addr
6	R/W	UINT 8	1	5th RTU Addr
7	R/W	UINT 8	1	6th RTU Addr
8	R/W	AC	30	1st Connect Command
9	R/W	AC	30	2nd Connect Command
10	R/W	AC	30	3rd Connect Command
11	R/W	AC	30	4th Connect Command
12	R/W	AC	30	5th Connect Command
13	R/W	AC	30	6th Connect Command

Table A-4. Modbus Configuration

Modbus Config Point Types – 27 (LOI), 34 (COM2), and 37 (COM1)				
Parameter #	Read-Write	Data Type	Length	Description
0	R/W	UINT 8	1	ASCII = 0, RTU = 1
1	R/W	UINT 8	1	Byte Order 1 = MSB 1st
2	R/W	UINT 8	1	Log Enable 1 = Yes
3	R/W	UINT 8	1	Host Active 1 = Active
4	R/W	UINT 8	1	Continuous Poll 1 = Yes
5	R/W	UINT 16	2	Poll Timer in Sec.
6	R/W	UINT 8	1	Table Start
7	R/W	UINT 16	2	Input Data Start
8	R/W	UINT 16	2	Output Data Start
9	R/W	UINT 16	2	HI Integer Scale
10	R/W	UINT 16	2	LO Integer Scale
11	R/W	INT 16	2	HI Float Scale 1
12	R/W	INT 16	2	LO Float Scale 1
13	R/W	FLP	4	HI Float Scale 2

Modbus Host User Program User Manual

Modbus Config Point Types – 27 (LOI), 34 (COM2), and 37 (COM1)				
Parameter #	Read-Write	Data Type	Length	Description
14	R/W	FLP	4	LO Float Scale 2
15	R/W	FLP	4	HI Float Scale 3
16	R/W	FLP	4	LO Float Scale 3
17	R/W	FLP	4	HI Float Scale 4
18	R/W	FLP	4	LO Float Scale 4
19	R/W	FLP	4	HI Float Scale 5
20	R/W	FLP	4	LO Float Scale 5
21	R/W	FLP	4	HI Float Scale 6
22	R/W	FLP	4	LO Float Scale 6
23	R/W	FLP	4	HI Float Scale 7
24	R/W	FLP	4	LO Float Scale 7
25	R/W	FLP	4	HI Float Scale 8
26	R/W	FLP	4	LO Float Scale 8
27	R/W	UINT 8	1	Enable Debug

Table A-5. Modbus Functions

Modbus Functions Point Types – 28 (LOI), 35 (COM2), and 38 (COM1)				
Parameter #	Read-Write	Data Type	Length	Description
0	R/W	AC	10	Function Tag Identification
1	R/W	UINT 16	2	1 Start
2	R/W	UINT 16	2	End Addr
3	R/W	UINT 8	1	Type
4	R/W	UINT 8	1	Lgl#
5	R/W	UINT 8	1	Param
6	R/W	UINT 8	1	Conv
7	R/W	UINT 16	2	2 Start
8	R/W	UINT 16	2	End Addr
9	R/W	UINT 8	1	Type
10	R/W	UINT 8	1	Lgl#
11	R/W	UINT 8	1	Param
12	R/W	UINT 8	1	Conv
13	R/W	UINT	2	3 Start
14	R/W	UINT	2	End Addr
15	R/W	UINT 8	1	Type
16	R/W	UINT 8	1	Lgl#

Modbus Host User Program User Manual

Modbus Functions				
Point Types – 28 (LOI), 35 (COM2), and 38 (COM1)				
Parameter #	Read-Write	Data Type	Length	Description
17	R/W	UINT 8	1	Param
18	R/W	UINT 8		Conv
19	R/W	UINT 16	2	4 Start
20	R/W	UINT 16	2	End Addr
21	R/W	UINT 8	1	Type
22	R/W	UINT 8	1	Lgl#
23	R/W	UINT 8	1	Param
24	R/W	UINT 8	1	Conv
25	R/W	UINT 16	2	5 Start
26	R/W	UINT 16	2	End Addr
27	R/W	UINT 8	1	Type
28	R/W	UINT 8	1	Lgl#
29	R/W	UINT 8	1	Param
30	R/W	UINT 8	1	Conv
31	R/W	UINT 16	2	6 Start
32	R/W	UINT 16	2	End Addr
33	R/W	UINT 8	1	Type
34	R/W	UINT 8	1	Lgl#
35	R/W	UINT 8	1	Param
36	R/W	UINT 8	1	Conv
37	R/W	UINT 16	2	7 Start
38	R/W	UINT 16	2	End Addr
39	R/W	UINT 8	1	Type
40	R/W	UINT 8	1	Lgl#
41	R/W	UINT 8	1	Param
42	R/W	UINT 8	1	Conv
43	R/W	UINT 16	2	8 Start
44	R/W	UINT 16	2	End Addr
45	R/W	UINT 8	1	Type
46	R/W	UINT 8	1	Lgl#
47	R/W	UINT 8	1	Param
48	R/W	UINT 8	1	Conv
49	R/W	UINT 16	2	9 Start
50	R/W	UINT 16	2	End Addr
51	R/W	UINT 8	1	Type
52	R/W	UINT 8	1	Lgl#

Modbus Host User Program User Manual

Modbus Functions Point Types – 28 (LOI), 35 (COM2), and 38 (COM1)				
Parameter #	Read-Write	Data Type	Length	Description
53	R/W	UINT 8	1	Param
54	R/W	UINT 8	1	Conv
55	R/W	UINT 16	2	10 Start
56	R/W	UINT 16	2	End Addr
57	R/W	UINT 8	1	Type
58	R/W	UINT 8	1	Lgl#
59	R/W	UINT 8	1	Param
60	R/W	UINT 8	1	Conv

Table A-6. Modbus Host

Modbus Host Point Types – 29 (LOI), 36 (COM2), and 39 (COM1)				
Parameter #	Read-Write	Data Type	Length	Description
0	R/W	AC	10	Host Tag Identification
1	R/W	UINT 8	1	0 RTU Addr
2	R/W	UINT 8	1	Fnc Num
3	R/W	UINT 16	2	Reg#
4	R/W	UINT 16	2	Save#
5	R/W	UINT 8	1	#Regs
6	R/W	UINT 8	1	Status
7	R/W	UINT 8	1	1 RTU Addr
8	R/W	UINT 8	1	Fnc Num
9	R/W	UINT 16	2	Reg#
10	R/W	UINT 16	2	Save#
11	R/W	UINT 8	1	#Regs
12	R/W	UINT 8	1	Status
13	R/W	UINT 8	1	2 RTU Addr
14	R/W	UINT 8	1	Fnc Num
15	R/W	UINT 16	2	Reg#
16	R/W	UINT 16	2	Save#
17	R/W	UINT 8	1	#Regs
18	R/W	UINT 8	1	Status
19	R/W	UINT 8	1	3 RTU Addr
20	R/W	UINT 8	1	Fnc Num
21	R/W	UINT 16	2	Reg#
22	R/W	UINT 16	2	Save#

Modbus Host User Program User Manual

Modbus Host Point Types – 29 (LOI), 36 (COM2), and 39 (COM1)				
Parameter #	Read-Write	Data Type	Length	Description
23	R/W	UINT 8	1	#Regs
24	R/W	UINT 8	1	Status
25	R/W	UINT 8	1	4 RTU Addr
26	R/W	UINT 8	1	Fnc Num
27	R/W	UINT 16	2	Reg#
28	R/W	UINT 16	2	Save#
29	R/W	UINT 8	1	#Regs
30	R/W	UINT 8	1	Status
31	R/W	UINT 8	1	5 RTU Addr
32	R/W	UINT 8	1	Fnc Num
33	R/W	UINT 16	2	Reg#
34	R/W	UINT 16	2	Save#
35	R/W	UINT 8	1	#Regs
36	R/W	UINT 8	1	Status
37	R/W	UINT 8	1	6 RTU Addr
38	R/W	UINT 8	1	Fnc Num
39	R/W	UINT 16	2	Reg#
40	R/W	UINT 16	2	Save#
41	R/W	UINT 8	1	#Regs
42	R/W	UINT 8	1	Status
43	R/W	UINT 8	1	7 RTU Addr
44	R/W	UINT 8	1	Fnc Num
45	R/W	UINT 16	2	Reg#
46	R/W	UINT 16	2	Save#
47	R/W	UINT 8	1	#Regs
48	R/W	UINT 8	1	Status
49	R/W	UINT 8	1	8. RTU Addr
50	R/W	UINT 8	1	Fnc Num
51	R/W	UINT 16	2	Reg#
52	R/W	UINT 16	2	Save#
53	R/W	UINT 8	1	#Regs
54	R/W	UINT 8	1	Status
55	R/W	UINT 8	1	9 RTU Addr
56	R/W	UINT 8	1	Fnc Num
57	R/W	UINT 16	2	Reg#
58	R/W	UINT 16	2	Save#

Modbus Host User Program User Manual

Modbus Host Point Types – 29 (LOI), 36 (COM2), and 39 (COM1)				
Parameter #	Read-Write	Data Type	Length	Description
59	R/W	UINT 8	1	#Regs
60	R/W	UINT 8	1	Status

APPENDIX B – CONFIGURATION WORKSHEETS

The Modbus Function and Host Configuration worksheets are provided to assist you in configuring the Modbus program. Make copies as needed.

Modbus Host User Program User Manual

INDEX

#

#Regs..... 2-13

Extensions

.BIN..... 1-5
.H00..... 1-4

A

AC.....A-2
Address..... 2-2, 2-4
 End..... 2-8
Address Span..... 2-7
Address Table..... 2-4
Adjusted Reading..... 2-5
Analog I/O
 Raw Values..... 2-5
ASCII..... 2-3
 Character.....A-2

B

Binary.....A-2
Bit Assignments.....A-2
BN.....A-2
Broadcast..... 1-2, 2-6, 2-12
Byte Order..... 2-4

C

Communications Parameters..... 3-1
Compatibility..... 1-3
CONC..... 2-6
Connect Command..... 2-2
Continuous Poll..... 2-4
CONV..... 2-5
Conv Field..... 2-10
Conversions
 Integer to Float..... 3-2

D

Data
 Link..... 1-3
Data Bytes
 Upper Limit..... 1-3
Data Type.....A-2
Data Types.....A-1
DCD..... 3-2
Description.....A-2
Dial-Up Modem Communications Card..... 1-3

DISCC..... 2-6
Download..... 1-5
Downloading the User Program..... 1-4
DSR..... 3-2

E

EIA-232 Serial Cards..... 1-3
Enable Debug..... 2-5
End Addr..... 2-8
End Address..... 2-8
Error Codes
 01..... 2-15
 02..... 2-15
 03..... 2-15
Errors
 Exception Response..... 2-15
 FloBoss appends character to message..... 3-1
 FloBoss does not respond in point to multi-point
 comm system..... 3-1
 FloBoss responds with wrong data..... 3-1
 Messages..... 3-1
Example
 Worksheets.....B-1
Exception Response..... 2-15
Extensions
 .BIN..... 1-5
 .H00..... 1-4

F

Figure 1-1. User Programs..... 1-4
Figure 1-2. ROC Flags..... 1-5
Figure 2-1. Modbus Modem..... 2-1
Figure 2-2. Modbus Configuration..... 2-3
Figure 2-3. Modbus Functions..... 2-7
Figure 2-4. Modbus Host..... 2-11
Flags..... 1-5
Float Range..... 2-5
Float Scale..... 2-9
Float to Integer..... 2-9
Floating Point.....A-2
FloBoss 103 Flow Manager Instruction Manual..... 1-1
FloBoss Address..... 2-2
FLP.....A-2
Fnc Num..... 2-12
FST..... 2-16
 Program Registers..... 2-10
Function Code 03..... 2-4
Function Codes..... 1-3, 2-12

Modbus Host User Program User Manual

G

Getting Started	1-1
Gould Modbus Protocol Reference Guide	1-1, 1-3

H

Hayes AT Command Set	2-2
HI Float Scale	2-5
HI Integer Scale	2-4
High Reading EU	2-4
Host	1-2
Configuration	2-10
Configuration Requests and Commands	2-10
Host Active	2-4, 2-10
Host Configuration Display	
Fnc Num	2-12
RTU Addr	2-12
Host/Slave	
Data Link	1-3

I, J, K

IEEE Format	A-2
Indexing	
Parameter	2-9
Point	2-9
Input Data Start	2-4
INT	A-2
Integer	2-5
Integer Range	2-5
Integer Scale	2-4
Integer to Float	
Conversions	3-2
Integers	A-2

L

Length	A-2
Lgl#	2-8, 2-9
LO Float Scale	2-5
LO Integer Scale	2-4
Log Enable	2-4
Logical Number	1-2, 2-8, A-2
Low Reading EU	2-4

M, N

Manuals	1-1
Memory Usage	1-4
Message	
Errors	2-15
Milliamp	2-5
Modbus	
Compatibility	1-3
Host	2-10
Modbus Config	2-3
Modbus Configuration	2-1

Modbus Configuration Display	
HI Float Scale	2-5
Low Float Scale	2-5
Modbus Functions	2-6
Modem	
Hayes AT Command Set	2-2
Response preceded by modem commands	3-2
RTU Address	2-2
Status	2-2
Modus Configuration	2-3
Modus Modem	2-1

O

Output	
Data Start	2-4
Overview	1-1, 1-2

P, Q

Param	2-10
Parameter	1-2
Parameter #	A-2
Parameter Indexing	2-9
Parameter Number	A-2
Point Indexing	2-9
Point or Logical Number	A-2
Point Type	2-8
Point Types	1-2, A-1
User Defined	A-1
Poll Timer in Sec	2-4
Program	
Versions and Names	1-4
Protocol User Manual	1-1, A-1

R

Raw Values	2-4
RCVED	2-5
Read-Write	A-2
Reg#	2-12
Register Number	
Address Span	2-7
Requests	2-7
Response	
Exception	2-15
ROC Flags	1-5
ROC Protocol User Manual	1-1, A-1
ROCLINK 800 Configuration Software User Manual	
.....	1-1, 2-10
RTU	2-3
RTU Addr	2-2, 2-12
RTU Address	2-2

S

Save#	2-12
Scope	1-1

Modbus Host User Program User Manual

SEND	2-5
Sending a Message	2-16
Signed Integers	2-5, A-2
Slave	1-2
Host Configuration Display	2-12
Soft Points	2-9, 2-11, 2-12
Start	2-7
Status	2-2, 2-13

T

Table 1-1. Modbus Function Codes	1-3
Table 1-2. Versions and Program Names	1-4
Table 2-1. Valid Host Register Numbers	2-8
Table 2-2. Status of Host Request or Command	2-14
Table 2-3. ROC Exception Response Example Message	2-15
Table 2-4. Modbus Protocol Errors that Elicit an Exception Response Message.....	2-15
Table A-1. Point Types and Descriptions.....	A-1
Table A-2. Data Type	A-2
Table A-3. Modbus Modem	A-3
Table A-4. Modbus Config.....	A-3
Table A-5. Modbus Functions	A-4
Table A-6. Modbus Host	A-6
Table Name	A-2
Table Start	2-4

Tag ID	2-7, 2-11
TLP.....	A-2
TLP Box Selections	1-2
TMOCT	2-6
TOMSG	2-5
Troubleshooting	
Error Codes.....	2-15
Type.....	2-8, A-2

U

UDP	A-1
UNIT	A-2
Unsigned Integer	A-2
User Defined Point Types	A-1
User Programs	1-4
.H00	1-4

V

Versions.....	1-4
---------------	-----

W-Z

WAITC.....	2-6
Worksheets	B-1
www.modicon.com	1-1

Modbus Host User Program User Manual

If you have comments or questions regarding this manual, please direct them to your local sales representative or contact:

Emerson Process Management

Flow Computer Division

Marshalltown, IA 50158 U.S.A.

Houston, TX 77065 U.S.A.

Pickering, North Yorkshire UK Y018 7JA

Website: www.EmersonProcess.com/flow