ADAPTIVE TechMemo #99-0003

PRIORITY: Normal

DATE: September 24, 1999

TITLE: Using Modbus ASCII protocol to display messages on ALPHA signs

ECO REFERENCE: None

PRODUCT(S) AFFECTED: Adaptive signs with Gateway upgrades

SUMMARY:

This document shows how to format the Modbus ASCII protocol to display text messages on ALPHA signs equipped with the Gateway firmware option. Using the Gateway Messaging software, messages are created, assigned a message number, and then stored in Gateway-equipped ALPHA signs. The Modbus ASCII protocol can be used to display these stored messages by using the message numbers.

1.0 Related documentation

Part #	Title	Description
9703-7004	Gateway Messaging Software User Manual	Describes how to create and store text messages on ALPHA signs that have Gateway firmware.
9708-8046	Network Configurations	Describes how to network ALPHA signs.

2.0 Required materials (minimum)

Part #	Description
Call Adaptive.	Gateway Messaging Software
Call Adaptive.	an ALPHA sign with the Gateway firmware option
1088-1111	Converter Box III
1088-8634	RS232 cable (connects Converter Box III to a PC)
1088-8626	RS485 cable (connects Converter Box III to an ALPHA sign)

Adaptive Micro Systems, Inc. • 7840 North 86th Street • Milwaukee, WI 53224 USA
414-357-2020 (phone) • 414-357-2029 (fax) • www.ams-i.com • techsupport@ams-i.com

3.0 Creating and storing messages

The Gateway Messaging software is used to create and store all the messages you want to appear in ALPHA signs. (For more information, see the **Gateway Messaging Software User Manual**, pn 9703-7004.)

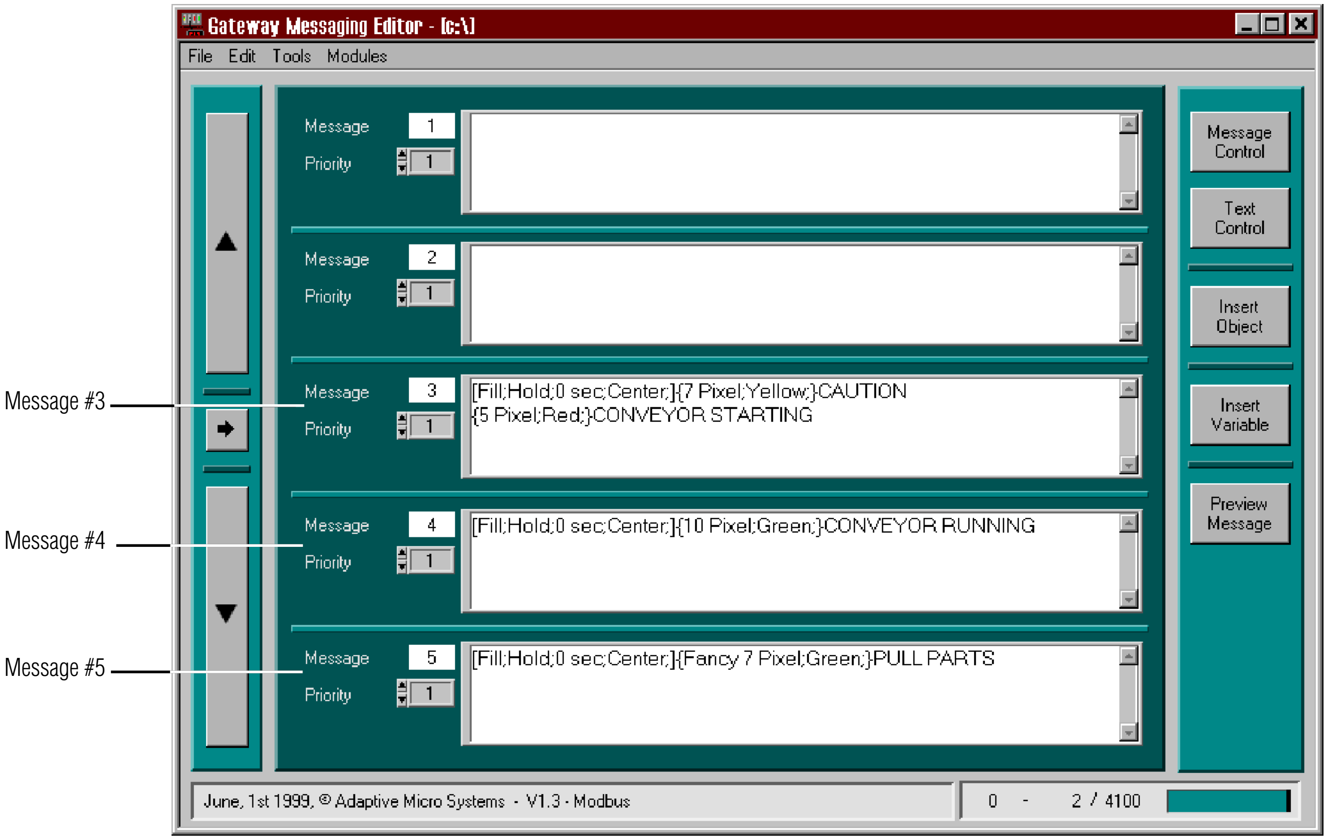
For example, let's say we have a conveyor system that will be started and stopped. When certain events happen in the conveyor process, we want to display one of three different messages on an ALPHA sign:

Message #	Message text	Purpose
3	CAUTION CONVEYOR STARTING	Show conveyor system has started
4	CONVEYOR RUNNING	Show conveyor system is running
5	PULL PARTS	Signal operator to start pulling parts

When the conveyor system is starting up, we want to display message 3. After the system has started and the conveyor is running, we want to display message 4. While the conveyor is running, we want to display message 5 to signal an operator to start pulling parts.

3.1 Create the messages

Use the Gateway Messaging software to create the messages. Our example looks like this:



Message #3

Message #4

Message #5

Message 1 Priority 1

Message 2 Priority 1

Message 3 Priority 1
[Fill;Hold;0 sec;Center;]{7 Pixel;Yellow;}CAUTION
{5 Pixel;Red;}CONVEYOR STARTING

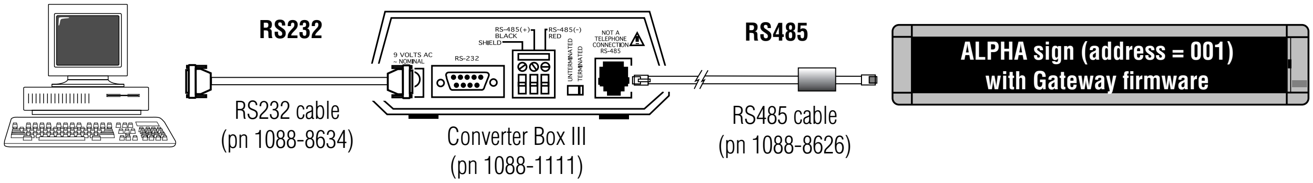
Message 4 Priority 1
[Fill;Hold;0 sec;Center;]{10 Pixel;Green;}CONVEYOR RUNNING

Message 5 Priority 1
[Fill;Hold;0 sec;Center;]{Fancy 7 Pixel;Green;}PULL PARTS

June, 1st 1999, © Adaptive Micro Systems - V1.3 - Modbus 0 - 2 / 4100

3.2 Store the messages on an ALPHA sign

To store the three example messages to the ALPHA sign, the computer and the sign must be connected as shown below. When the hardware is connected, use the Gateway Messaging software to transmit the messages to the ALPHA sign.



NOTE: To connect just *one* sign, an RS232 cable can be used and the Converter Box III would not be necessary. For more information, see “Connecting a single sign to a cabled PC network” in the **Network Configurations** manual (pn 9708-8046.)

4.0 Formatting Modbus ASCII strings

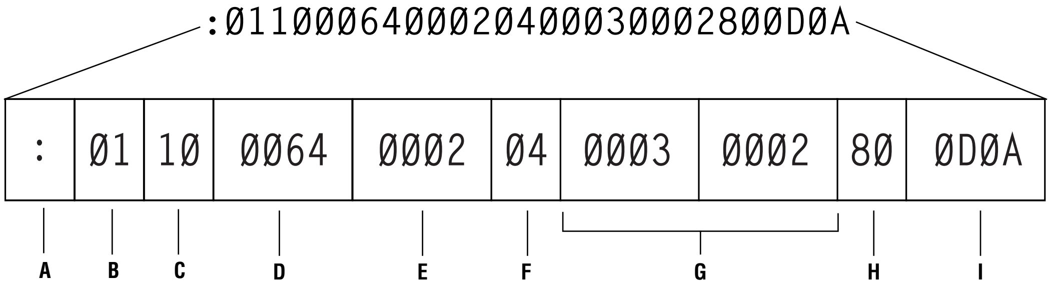
Modbus ASCII strings can be formatted to act like switches that turn sign messages on and off.

In our example, we want to turn one message on and the previous message off. For instance, to turn Message #2 off (previous message) and Message #3 on, the Modbus ASCII code string would look like the following:

NOTE: *Remember that a message's **number**, not the actual message text itself, is used in a Modbus ASCII string.*

NOTE: For more information, see “6.2 Generic Modbus ASCII string format” on page 5.

Modbus ASCII code string to turn Message #3 on and Message #2 off

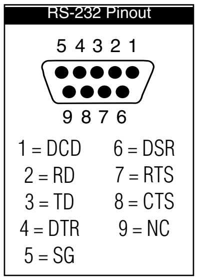


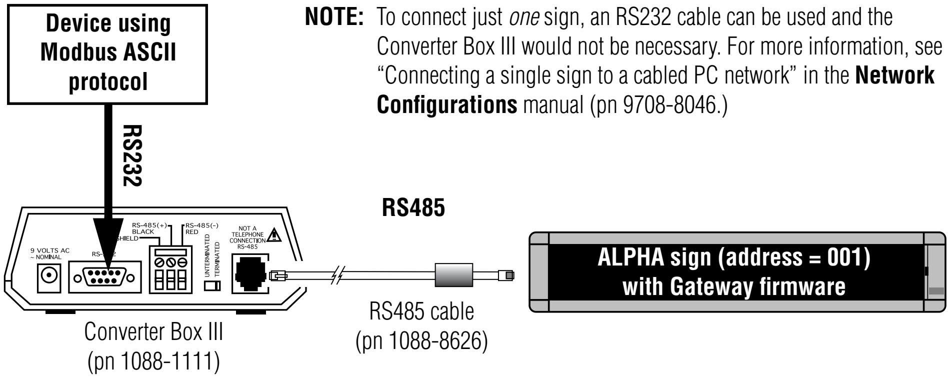
Item	Name	Number of bytes	Value (hex)	Description
A	Start of frame delimiter	1	:	ASCII “:” (colon) character
B	Sign Address	1	Ø1	ALPHA sign address.
C	Function	1	1Ø	Write a binary value into two or more registers
D	Start Address	2	ØØ64	Activation Register address (65h) **minus 1**.
E	Number of Registers	2	ØØØ2	Two registers (Activation and De-activation) will be written to.
F	Number of Bytes	1	Ø4	# of bytes in Data Register #1 (0003) + # of bytes in Data Register #2 (0002) = 4 bytes
G	Data Register #1	2	ØØØ3	The message *number* (ØØØ3) is the data.
	Data Register #2	2	ØØØ2	The message *number* (ØØØ2) is the data.
H	LRC	1	8Ø	(((01 + 10 + 00 + 64 + 00 + 02 + 04 + 00 + 03 + 00 + 02) XOR FF) = 7F) + 1 = 80h
I	End of frame delimiters	2	ØD + ØA	Carriage return + Line feed

5.0 Displaying stored messages using Modbus ASCII strings

5.1 Connect the Modbus ASCII device

NOTE: To connect just *one* sign, an RS232 cable can be used and the Converter Box III would not be necessary. For more information, see “Connecting a single sign to a cabled PC network” in the **Network Configurations** manual (pn 9708-8046.)




Device using Modbus ASCII protocol
RS232
9 VOLTS AC ~ NOMINAL
RS-232
RS-485(+) BLACK
SHIELD
RS-485(-) RED
UNTERMINATED
TERMINATED
NOT A TELEPHONE CONNECTION RS-485
Converter Box III (pn 1088-1111)
RS485
RS485 cable (pn 1088-8626)
ALPHA sign (address = 001) with Gateway firmware

5.2 Set the communication parameters on the device

Set up the device sending the Modbus ASCII protocol to use one of the following formats:

	Format 1	Format 2
Mode:	ASCII	ASCII
Baud rate:	9600	9600
Data bits:	7	7
Parity:	Even	Odd
Stop bits:	2	2
RTS control:	None	None

5.3 Transmit the Modbus ASCII strings to an ALPHA sign

NOTE: To use the Modbus ASCII protocol, an ALPHA sign must be equipped with Gateway firmware.

In this example, we would transmit the following Modbus ASCII strings to display the three example messages we created:

Modbus ASCII string (If there is more than a 3-second delay between strings, “No Network Activity” will appear on the sign.)	**ALPHA sign memory locations** **101 (65h)** **Activation Register**	 **102 (66h)** **De-activation Register**	**Result**	**ALPHA sign**
:011000640002040003000280	3	2	Turn on Message #3. Turn off Message #2	CAUTION CONVEYOR STARTING
:01100064000204000400037E	4	3	Turn on Message #4. Turn off Message #3	CONVEYOR RUNNING
:01100064000204000500047C	5	4	Turn on Message #5. Turn off Message #4	

6.0 Reference

6.1 ALPHA sign memory map

ALPHA signs store data in registers as shown in the “map” below. This document focuses only on the activation (101) and de-activation (102) registers.

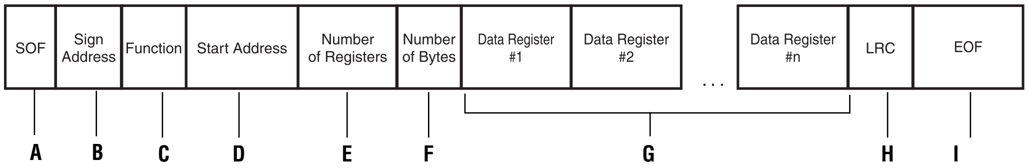
When a message number is written into the activation register, that message is displayed on the sign. When a message number is written into the de-activation register, that message is no longer displayed.

Memory Map of ALPHA Sign with Gateway Option

Memory register (decimal with hex in parentheses)	Description
01 (01h) 02 (02h) 03 (03h) . . . 100 (64h)	Variable 1 Variable 2 Variable 3 . . . Variable 100
101 (65h) 102 (66h)	Activation Register De-activation Register
103 (67h) 104 (68h) 105 (69h) . . 167 (A7h)	Active Message Queue - read only - (Used to show all the messages that can be displayed on a sign. A PLC can read this information from the sign to verify what message have been triggered.)

6.2 Generic Modbus ASCII string format

All frame values below are represented in ASCII hexadecimal characters.
(For example, 06h would be represented by 30h + 36h, the ASCII characters “06”)



Item	Name	# of bytes	Description
A	Start of frame delimiter	1	ASCII “:” (colon) character
B	Sign Address	1	Value from 00 to FFh which represents the address of an ALPHA sign. This address is used to identify a sign in a network of signs. When address 00h is used, a message will be sent to all the signs on a network.
C	Function	1	One of four values to either read or write information on a sign: 03h = Read a binary value from one or more holding registers 04h = Read a binary value from one or more input registers 06h = Write a binary value into one holding register 10h = Write a binary value into two or more holding registers
D	Start Address	2	Starting address **minus 1** (in HO-LO format) of the first register where information will be written or read. For a sign, this is for either the Activation or De-activation register. (The reason for subtracting 1 is that the sign adds 1 to get the true address.)
E	Number of Registers	2	Total number of sequential registers (in HO-LO format) that will be read or written. Maximum number of registers is 128 (80h).
F	Number of Bytes	1	Total number of bytes of data in all data registers to be sent. (Not used if the Function is 03h or 04h.)
G	Data Register #1	2	Data (in HO-LO format) that will be sent or received.
	Data Register #2	2	
	Data Register #n	2	
H	LRC	1	= ((Sum of all preceding values in this frame excluding the initial colon) XOR FFh) + 1
I	End of frame delimiters	2	Carriage return (ØDh) + Line feed (ØAh)