SOFTWARE SP798-91

NCMZ-798 MODBUS User's Guide

NLAM-B205

May 19, 1988

Introduction

The SP798-91 firmware allows the NCMZ-798 communication module to function as an interface between Westinghouse PC-700, PC-900, or PC-1100 controllers and a MODBUS network. The module operates as a slave on the network and is able to respond to commands from the network master.

Communication to the PLC is by a cable from port A on the module to the program loader port of the controller using the standard Westinghouse 6-byte protocol. The MODBUS network is connected to port B of the module. Communication protocol from port B is MODBUS ASCII or RTU as defined by a configuration register in the PLC. Communication speed for each port is hardware configurable between 110 and 9600 baud.

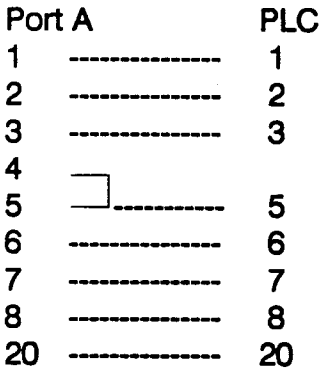
Installation

The module requires two vertical rack positions or one horizontal rack position. No communication via the I/O bus is necessary for operation, but once installed in a rack, it occupies one IR and one OR.

Power requirements:

+5.7 VDC – 500 mA
+12 VDC – 50 mA
–12 VDC – 30 mA

Cable connections between NCMZ port A and the PLC:



Cable connections between NCMZ port B and the Modbus network:

Pin 2 receives data
Pin 3 transmits data
Pin 4 (RTS) requires 12V signal input
Pin 5 (CTS) 12V signal output (only when module transmitting)
Pin 6 (DSR) 12V signal output
Pin 7 Signal ground
Pin 8 12V signal output
Pin 20 (DTR) requires 12V signal input

Module Set-Up

On power up, the module reads the PLC parameter table located between 8200_h and $821F_h$. This table is used to check the validity of addresses received from the MODBUS master.

The module also reads HR5 for configuration information. The high order byte of HR5 contains the MODBUS slave address (01-FF). The MODBUS communication Data Format is in HR5, low order byte as follows:

Hex Code	Format	Mode
01	7B, Even Par, 2 stop bits	ASCII
05	7B, Odd Par, 2 Stop bits	ASCII
09	7B, Even Par, 1 Stop bit	ASCII
0D	7B, Odd Par, 1 Stop bit	ASCII
11	8B, 2 Stop bits	RTU
15	8B, 1 Stop bit	RTU
19	8B, Even Par, 1 Stop bit	RTU
1D	8B, Odd Par, 1 Stop bit	RTU

ALL OTHER CODES ARE INVALID.

During the Power-Up sequence, the status LED will flash (2 Hz) until all data is read successfully. The length of the power up sequence at 9600 baud is less than 2 seconds.

Application Notes

1. The parameter table is only read at power-up. It is necessary to cycle the power to the module if the configuration register information is changed.

2. Early versions of Westinghouse Programmable Controllers did not support block read and block write functions. A block read command is used to read the parameter table at module power up. The controller to be used MUST support the block read function.

   The following Executive Firmware versions support block read/write: PC-700 Version 6.3; PC-900 Version 4.7 or greater; PC-900 EPROM Version 5.7; PC-1100 Version 2.0 or greater. If the PLC to be used does not support block functions it should be upgraded.

3. If a single register is requested or sent via the MODBUS (codes 03 and 06), the module will use the single register read and write commands to the PLC. If multiple registers are required (codes 03 and 16), a block read or write command is issued to the PLC. Use of the block commands greatly improves the communications throughput.

4. The modify coil status function writes to the output status table. This is equivalent to setting a bit in an Output Group. Therefore, a coil which is turned ON over MODBUS may be turned OFF by the ladder program. The coil may also be reset by turning the key switch to stop or by cycling power to the CPU.

5. The module is communicating through the program loader port. The communications is such that all read/writes are performed at end of ladder scan.

6. The status LED reports the operating condition of the module.

Flashing Fast (2Hz)	Power-Up sequence
Flashing Medium (1Hz)	Communication error on Port A
Flashing Slow (.5Hz)	Communication error on Port B
Steady	Normal Operation

7. The handshaking lines for port B may be defeated by jumpering pins 4, 8, and 20.

Introduction to MODBUS

The MODBUS communications network was designed to provide a simple means of connecting slave controllers to a master controller for the purpose of distributed control and data acquisition. The simplicity of the protocol has made it a popular choice among many control and instrumentation vendors seeking a "common" language.

The MODBUS protocol consists of a slave address, function code, data, and error checking. The master controller issues a command to which the appropriate slave responds.

The following describes the message parameters:

Address - The address field contains the address of a specific slave (1-256), as assigned by the user. Each slave must be assigned a unique address. Only the addressed slave will respond to a message. The response message contains the address of the slave which issued it.

A broadcast message uses a slave address of 0. All slaves on the network interpret the instruction and take action but do not issue a response message. Only function codes 5, 6 and 16 can be used with this feature.

Function Code - This field defines the action to be performed. The following table lists the function codes supported by the NCMZ-798 module and how they are interpreted.

01 Read Coil Status

Obtains the current status (ON/OFF) of a group of logic coils. Validity of coil start address and quantity of coils is checked with the number shown in the parameter table. Unrequested coils in the response message are zeroed. Coil CR1 is MODBUS address 0000_h. Coil CR512 is address $01FF_h$.

02 Read Input Status

Obtains the current status (ON/OFF) of a group of discrete inputs. Validity of requested addresses are checked against the parameter table. Input IN1 is MODBUS address 0000_h. Input IN256 is address $00FF_h$.

03 Read Output (Holding) Registers

Obtains the current value in one or more holding registers. A maximum quantity of 64 registers is allowed per request.

The MODBUS specification does not differentiate between Output Registers and Holding Registers as Westinghouse does. This function code allows the user to read any memory location in the PLC. The following table lists the Westinghouse PLC memory locations and their corresponding MODBUS addresses.

Westinghouse	MODBUS
HR1 - HR1792	0000_h - $06FF_h$
OR1 - OR32	read address $820D_h$ in PLC for location of OR1 ($80A0_h$ - $80BF_h$ in PC-700)
Fault Register	8206_h

04 Read Input Registers

Obtains current value in one or more input registers. Validity is checked with parameter table.

05 Modify Coil Status

Set logic coil to a state of ON or OFF. Validity of coil number is checked with parameter table. The change is made to the output status table (NOT THE FORCE TABLE!).

06 Modify Register Contents

Write information to one register. The user should use extreme caution when writing to any memory location other than holding registers. See Read Registers (03) above.

07 Read Exception Status

The low order byte of HR6 is read. The contents of HR6 may be set by the Ladder Program.

08 Loop Back Diagnostic

Diagnostic test message which can be used to test communications. The response message is identical to received message. The test message must be the full length of a normal command (8 bytes for RTU mode, 17 characters for ASCII mode).

16 Preset Multiple Registers

Information can be written to multiple holding registers (up to 64) with one command.

Data - The data field contains information needed to perform the specific function. This may be values, addresses, or limits. For example, the function code Read Coils (01), requires data of starting coil number and quantity.

Error Check - This field contains the error checksum which is used to determine if a transmission has been received correctly. Two forms of error checking are used by the protocol. The error check for ASCII mode is Longitudinal Redundancy Check (LRC) and is an eight bit binary number represented and transmitted as two ASCII hex characters. The error check

for RTU mode is Cyclical Redundancy Check (CRC) and is appended to the message as two CRC check bytes.

MODBUS Protocol Modes

The MODBUS protocol allows two modes of transmission. One mode is ASCII and the other is RTU. Only one mode may be used per network, they can not be mixed.

ASCII mode is a hexadecimal coded system using 7-bit ASCII characters (0-9, A-F). The message must begin with a colon (:) and end with a line feed (LF). Advantages of ASCII mode include the ASCII characters are easy to view when troubleshooting and the ease of implementation in higher level languages such as BASIC or FORTRAN.

RTU mode transmits information in 8-bit binary format. Message start is defined as data received following 3 consecutive character lengths with no transmission and ends with 3 consecutive character lengths with no transmission. RTU mode uses half as many characters as ASCII mode but is more difficult to program and troubleshoot. In the RTU mode, characters must be sent continuously where ASCII mode allows up to 1 second between characters to allow for a relatively slower master.

MODBUS commands involving illegal data or communication failures will result in an exception code response from the module. When the module detects one of these errors, it sends a response to the master consisting of slave address, function code, error code and error check fields. To indicate that the response is a notification of an error, the high order bit of the function code is set to 1.

Error Response Codes

Code	Meaning
01	The specified function is invalid.
02	The start-point specification is invalid.
03	The specified data or quantity of points is invalid.
04	Erroneous communication between module and PLC, without loss of synchronization. If synchronization is lost, the module will fall back into the Power-Up sequence, and no response message is generated.

Error messages only occur after a correctly received and formatted message. For this reason, they are most likely due to programming errors at the master.

Example Message Exchanges

EXAMPLE #1 - Loop Back Diagnostic Test, Slave 2

QUERY MESSAGE

	RTU		ASCII	
Header	None		:	
Address	0000	0010	0	2
Function	0000	1000	0	8
Diagnostic Codes H.O.	0000	0000	0	0
L.O.	0000	0000	0	0
Sample Data H.O.	1010	0101	A	5
L.O.	0011	0111	3	7
Error Check	1101	1110	1	A
	1011	1110		
Trailor	none		CR	LF

RESPONSE MESSAGE

	RTU		ASCII	
Header	None		:	
Address	0000	0010	0	2
Function	0000	1000	0	8
Diagnostic Codes H.O.	0000	0000	0	0
L.O.	0000	0000	0	0
Sample Data H.O.	1010	0101	A	5
L.O.	0011	0111	3	7
Error Check	1101	1110	1	A
	1011	1110		
Trailor	none		CR	LF

EXAMPLE #2 - Read Coils CR20 to CR56 from Slave Number 2

QUERY MESSAGE

	RTU		ASCII	
Header	None		:	
Address	0000	0010	0	2
Function	0000	0001	0	1
Starting Address H.O.	0000	0000	0	0
L.O.	0001	0011	1	3
Quantity of Points H.O.	0000	0000	0	0
L.O.	0010	0101	2	5
Error Check	0000	1110	C	5
	1000	0100		
Trailor	none		CR	LF

RESPONSE MESSAGE

	RTU		ASCII	
Header	None		:	
Address	0000	0010	0	2
Function	0000	0001	0	1
Byte Count	0000	0101	0	5
Data	1100	1101	C	D
	0110	1011	6	B
	1011	0010	B	2
	0000	1110	0	E
	0001	1011	1	B
Error Check	0100	0101	E	5
	1110	0110		
Trailor	none		CR	LF

Shows following coils ON:

20, 22, 23, 26, 27, 28, 29, 31, 33, 34, 37, 40, 41, 43, 45, 46, 47, 52, 53, 55, 56

Coils 57-59 included in data but shown as 0 since not requested.

EXAMPLE #3 - Incorrect Request and Error Response

QUERY MESSAGE

(Read coil status of CR1245 in slave 10)

	RTU		ASCII	
Header	None		:	
Address	0000	1010	0	A
Function	0000	0001	0	1
Starting Address H.O.	0000	0100	0	4
L.O.	1010	0001	A	1
Quantity of Points H.O.	0000	0000	0	0
L.O.	0000	0001	0	1
Error Check	1010	1100	4	F
	0110	0011		
Trailor	none		CR	LF

RESPONSE MESSAGE

(Error code 02, Illegal Data Address)

	RTU		ASCII	
Header	None		:	
Address	0000	1010	0	A
Function	1000	0001	8	1
Exception Code	0000	0010	0	2
Error Check			7	3
Trailor	none		CR	LF

Test Program

The following BASIC program is intended for use on an IBM-PC with a serial communication port to test the NCMZ-798 in ASCII mode.

Use the NLC-3PL cable between IBM COM port and NCMZ port B.

```
1     REM IBM PERSONAL COMPUTER
2     REM KNOWN BUGS:   THERE IS NO TIME-OUT ON INPUT (LINE 290)
3     REM               USE BREAK IF NO REPLY FROM MODULE
4     REM               TOTAL MESSAGE LENGTH MUST NOT EXCEED 255 CHAR
10    CLS
20    PRINT "MODBUS ASCII Test Program"
30    PRINT
40    PRINT "The module must be in ASCII mode (set HR5)"
50    PRINT: PRINT "ADAPT LINE 80 TO APPROPRIATE FORMAT"
60    PRINT
70    GOSUB 330
80    OPEN "COM1:9600,E,7,1,CS" AS #1
85    REM CS IGNORES CLEAR TO SEND HANDSHAKING LINE
90    CLS
100   PRINT "Colon and Checksum will be added"
110   INPUT "Enter string to transmit: ";A$
120   CHECKSUM=0
130   IF (LEN(A$) AND 1)0 THEN PRINT "Odd number of characters not permitted"
      :GOTO 110
140   IF LEN(A$)12 THEN PRINT "***WARNING*** Number of Characters ";LEN(A$)
150   FOR I=1 TO LEN(A$) STEP 2
160   CHECKSUM=VAL("&h"+MID$(A$,I,2))+CHECKSUM
170   IF CHECKSUM255 THEN CHECKSUM=CHECKSUM-256
180   NEXT I
190   CHECKSUM=256-CHECKSUM
200   IF CHECKSUM THEN CC$="0" ELSE CC$=""
210   CC$=CC$+HEX$(CHECKSUM)
220   PRINT
230   PRINT "STRING CONVERTED TO... : ";A$;CC$
240   PRINT:PRINT
250   PRINT "START TRANSMISSION AT ";TIME$
260   PRINT #1, ":";A$;CC$;CHR$(13);CHR$(10);
270   INPUT #1, REPLY$
280   PRINT "RESPONSE RECEIVED AT ";TIME$
290   PRINT:PRINT:PRINT "RECEIVED MESSAGE ";REPLY$
310   GOSUB 330
320   GOTO 90
330   PRINT:PRINT "STRIKE ANY KEY TO CONTINUE"
340   IF INKEY$="" THEN 340 ELSE RETURN
```