ATC-300+ Modbus Communications Guide

66A7787 rev 1

This page is intentionally left blank.

Eaton Corp.
1000 Cherrington Parkway
Moon Township, PA 15108

TABLE OF CONTENTS

1. INTRODUCTION

This document is to be used as a reference to communicate with the ATC-300+ Automatic Transfer Switch Controller using the Modbus protocol. .

1.1. Overview

A typical Modbus network is shown in Figure 1. The network communicates using a master-slave technique. A single master device initiates all transactions, called queries, on the network. Slave devices respond to the master's queries, either by returning data or performing an action requested by the query. A query is addressed to an individual slave or broadcast to all slaves. Slave devices do not respond to a broadcast query.

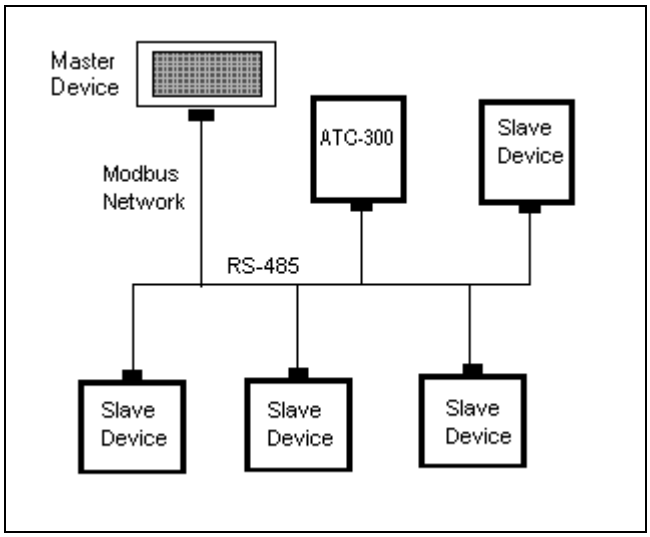


Figure 1: A Typical Modbus Network

A multi-slave device Modbus network may be implemented using a 2-wire half-duplex RS-485 implementation. Various slave devices from Eaton or other Modbus compliant devices may be connected to the Modbus network. A maximum of 32 slave devices may be connected to the network at distances up to 4000 feet.

A 121 ohm terminating resistor can be added as an end of line terminator. The ATC-300+ has a DIP switch on the back of the unit to switch the resistor in or out of the RS-485 receiver/transmitter circuit by the user as needed. Termination resistors are typically not needed for baud rates of 19200 and lower at distances up to 4000 feet.

The Modbus protocol specifies two transmission modes: ASCII (American Standard Code for Information Interchange) and RTU (Remote Terminal Unit). The ATC-300+ will support the RTU mode of the Modbus protocol.

1.2. Definitions

Character Time -	The time for one character (11 bits) to be transmitted over the Modbus network at the prevailing baud rate.
CRC -	Cyclical Redundancy Check.
Frame Packet -	Is interchangeable with Character time.
Message Packet -	A complete Modbus message made up of frame packets containing an address, function code, data field and error-checking field.
Modbus Coil -	Information contained as a 1-bit quantity.
Modbus Register -	Information contained as a 16-bit quantity.
Query -	A Modbus message from the Modbus master to the product.
Response -	A Modbus message from the product to the Modbus master.

1.3. RS-485 Connections

A 4-pin connector (J10) is provided for wiring to the RS-485 network. The following chart shows the ATC-300+ J10 connector pin-out assignment.

J10	Signal
1	B (+)
2	A (-)
3	Common
4	Shield

The polarity of the A (-) and B (+) signals is very important. In the Modbus network, A (-) terminals must connect to other A (-) terminals and B (+) terminals must connect to other B (+) terminals.

Use a shielded twisted pair cable 22 AWG (0.33 mm^2) or thicker and ground the shield only once at the Master device. If there is more than one Slave device cabled to the Modbus Master, tie the cable shields together but do not connect to ground at any point other than at the Master device.

1.4. References

Modbus® is a registered trademark of Schneider Automation, Inc.

The following documents are referenced by this specification and may be necessary to properly understand this material.

IL 17384	“IMPACC System Communications”,
02-PMP-01	“Modbus RTU Products Specification”, Eaton Corp., Rev 1.02, November 2004.
PI-MBUS-300	"Modicon Modbus Protocol Reference Guide", MODICON, Inc., Industrial Automation Systems, Rev. J, June 1996. -

2. Modbus RTU Message Protocol

2.1. Modbus RTU Message Protocol

The Modbus RTU protocol is based on a technique in which a single master initiates a transaction (called a query) on the network. Every slave device connected to the network receives the Modbus query. A query is broadcast to all slaves or addressed to an individual slave. Slave devices do not respond to a broadcast query. An individually addressed slave device responses to the master query by either (1) returning data requested by the query, (2) performing an action requested by the query and returning status of that action, or (3) returning an error code.

The Query–Response Cycle between a Modbus master and slave is shown in Figure 2.

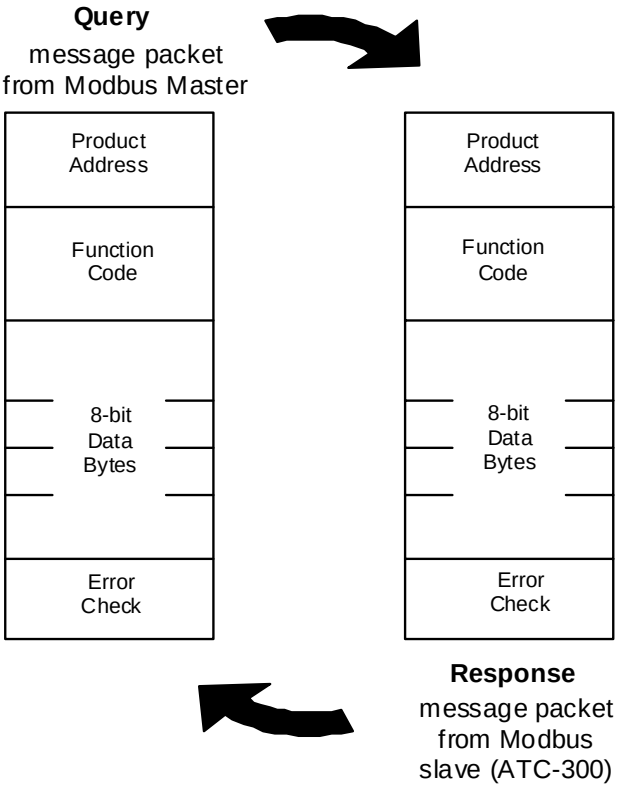


Figure2: Modbus Master-Slave Query-Response Cycle

The address is the first byte of each Modbus transmission. Only the addressed slave device responds to a query beginning with its individual address.

The function code in the query tells the addressed slave what kind of action to perform. The data bytes contain additional information that the slave needs in order to perform the function.

For example, function code 04 queries the slave to read actual value registers and respond with the contents of those registers. The “data field” must contain the information that specifies to the slave which register to begin reading and the number of registers to read.

The “error check” field provides a method for the slave to validate the integrity of the query message contents.

The function code in a normal response from the slave is an echo of the function code from the query. The data bytes contain the information requested; i.e., register contents.

If the slave receives a query message that is in error, the function code is modified to indicate the response message is an error response. The data bytes of the response contain an exception code that describes the error.

The error check field of the response allows the master to confirm the response message contents are valid.

2.2. Modbus Message Types and Framing

The Modbus protocol defines two data exchange modes - ASCII (American Standard Code for Information Interchange) and RTU (Remote Terminal Unit). All devices (master and slaves) on a single Modbus network must communicate using the same exchange mode. ASCII transfers provide each eight-bit byte of information encoded in two ASCII characters. RTU transfers provide each eight-bit byte of information as two binary encoded four-bit hexadecimal characters.

The Eaton ATC-300+ supports the RTU mode. The main advantage of the RTU mode is its greater character density[1], which provides for better data throughput at the same baud rate.

A RTU query or response is placed by the transmitting device into a Modbus message packet, which has a known beginning and ending point. The message packet is made up of multiple frame packets. This allows receiving devices to begin at the start of the message packet, read the address portion to determine which device is addressed[2] and to know when the message is completed. Partial messages can be detected and errors can be identified as a result.

Each RTU frame packet contains a start bit, eight data bits[3], and if parity is used, a bit for even / odd parity and one stop bit. If parity is not used, another stop bit is generally used in its place[4], thus resulting in two stop bits. Each frame packet, therefore, contains a total of 11 bits for each eight-bits of data exchanged. Each eight-bit data byte is defined as two binary encoded four-bit hexadecimal characters 0 … 9, A … F.

RTU message packets start with a silent interval of at least 3.5 frame packet times. This is most easily implemented as a multiple of frame packet times at the baud rate being used on the network. The silent interval between message packets is:

(3.5 frame packets) x (11 bits / frame packet) x (1 sec / baud rate).

The silent intervals for each selectable baud rate is shown in Table 1. Networked devices monitor the network bus continuously, including silent intervals.

Baud Rate (bits / sec)	Silent Interval (milliseconds)
9600	4.01
19200	2.01

Table1: Silent Interval Times

[1] Nearly twice as dense as the Modbus ASCII mode message protocol.

[2] Or if all devices are addressed in the case of a broadcast message.

[3] The least significant bit is sent first.

[4] To accommodate systems which do not incorporate a second stop bit when no parity is selected, an ideal device could be set to receive no parity and one stop bit while transmitting no parity and two stop bits.

Upon detecting an appropriate silent interval, all Modbus products prepare to recognize the next received byte as the address field. If the received address is the same as the address assigned to the slave, the slave receives the rest of the query from the master and responds appropriately. The slave always responds with its assigned address to the master.

The entire message packet must be transmitted as a continuous stream. If a silent interval of more than 3.5 frame times occurs before completion of the message packet, the receiving device flushes the incomplete message and assumes the next frame packet begins a new message.

If a new message begins earlier than 3.5 frame times following a previous message, the receiving device considers it a continuation of the previous message. This causes an error, as the value in the final CRC error checking field is not valid for the combined messages.

A slave device will not respond to message packets in which a computed CRC doesn't match the received CRC.

A typical message packet is shown in Figure 3.

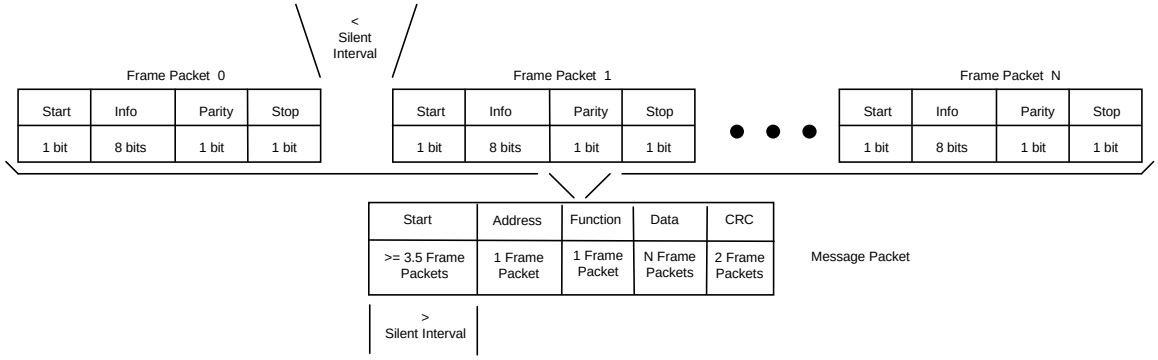


Figure 3: RTU Message Packet

2.3. Device Addressing

The first frame packet of a message contains the eight-bit address field. Valid device addresses are in the range of 1 ... 247[5] decimal.

A master addresses a slave by placing the slave address in the address field of the message packet. When the slave sends its response, it places its own address in the address field of the response to verify to the master the correct slave is responding.

2.4. Register Addressing

All data addresses of the registers, which are transmitted in a data field's 16-bit address contents of a Modbus message, are referenced from 0 through $FFFF_{16}$ ($65{,}535_{10}$). Therefore, the address of a register is one count less than the register number.

[5] A Modbus protocol limited range of addresses.

By convention, this document will present the register number in decimal and the register address in hexadecimal. Thus, Setpoint register 3001_{10} is register address $BB8_{16}$ (i.e., 3000_{10}).

2.5. Function Codes

The frame packet following the address in a message packet contains the eight-bit function code field. When sent from a master to the ATC-300+, the function code field tells the ATC-300+ what action to perform. Examples include reading the ON / OFF states of a group of inputs, reading the data contents of a group of registers, reading the diagnostic status of the slave or writing to designated outputs or registers. Valid function codes from the master are 1 ... 127 decimal. The ATC-300+ supports the function codes listed in Table 2.

When the ATC-300+ responds to the master, it uses the function code field to indicate either a normal (error-free) response or an error condition occurred (called an exception response). For a normal response, the ATC-300+ performs the requested function and simply echoes the original function code in the response message.

Function Code	Action	Modbus definition	ATC-300+ Register Group
01	Read	Coil status	Discrete outputs
02	Read	Input status	Discrete inputs
03	Read	Holding registers	Setpoints
04	Read	Input registers	Actual values
05	Write	Force single coil	Operation command
16	Write	Preset multiple registers	Write Setpoints

Table2: Function Codes

When the ATC-300+ does not perform the action associated with the function code of the message packet, it returns an exception response. For an exception response, the slave returns a code that is equivalent to the original function code with its most significant bit set to logic 1, i.e., it is defined to have a value greater than 127. For example, a message from master to ATC-300+ to read a group of registers would have the following function code:

```
0000 0011              (Hexadecimal 03)
```

If the ATC-300+ takes the requested action without error, it returns the same function code in its response. If an exception occurs the requested action is not performed by the ATC-300+ and it returns:

```
1000 0011              (Hexadecimal 83)
```

In addition to modifying the function code for an exception response, the ATC-300+ places a unique exception code into a single byte data field of the response message. This tells the master what kind of error occurred, or the reason for the exception. Exception codes are defined in Table 24.

2.6. Data Format

Each Modbus register is defined in the Modbus protocol as a 16-bit (two byte) entity. Modbus protocol defines register information to be transmitted with the high-byte first, followed by the low-byte.

2.7. Error-Checking Field

The error-checking field contains a 16-bit value implemented as two 8-bit bytes. The error-check value is the result of a Cyclical Redundancy Check (CRC) calculation performed on the entire

Bits 15……8	Bits 7……0	Bits 31……24	Bits 23……16	Bits 47……40	Bits 39……32	Bits 63……56	Bits 55……48
1st byte	0th byte	3rd byte	2nd byte	5th byte	4th byte	7th byte	6th byte
Register x		Register x+1		Register x+2		Register x+3	

Table 3: Default Multi-Register Fixed Point Transmission Order

contents of the message packet. Only the eight-bits of data in each frame packet is applied to the CRC calculation. The start bit, parity bit and stop bits do not apply to the CRC.

The error-checking field is appended to the message packet as the last field. Opposite to data field information, the low-order byte of the CRC calculation is transmitted first, followed by the high-order byte. Thus, the high-order byte is the last byte to be sent in the message packet.

If the ATC-300+ detects a CRC error, the entire message packet must be discarded. An ATC-300+ detecting a CRC error in a received Modbus message does not respond to the master device.

3. Function Code Descriptions

3.1. Function Code 01 – Read ATS Status bits

Function code 01 reads the ON / OFF status of the 13 discrete outputs in the ATC-300+.

Name	Register Number (decimal)	Register Address (hex)
Source 1 Available	1000	3E7
Source 2 Available	1001	3E8
Source 1 Connected	1002	3E9
Source 2 Connected	1003	3EA
ATS In Test	1004	3EB
ATS Waiting for Sync	1005	3EC
K1 Relay	1006	3ED
K2 Relay	1007	3EE
K3 Relay	1008	3EF
K4 Relay	1009	3F0
Pretransfer Relay	1010	3F1
Alarm Relay	1011	3F2
Gen Start Relay	1012	3F3

Table 4: Function Code 01 Definitions

The query message format in given in Table 5. The query specifies the starting status bit address and the quantity of status bits to be read. This example requests the Source 1 and Source 2 Available and Connected status bits.

Query Field Name	Example
Slave Address	08_{16}
Function Code	01_{16}
Starting Address High Byte	03_{16}
Starting Address Low Byte	$E7_{16}$
Number of Points High Byte	00_{16}
Number Of Points Low Byte	$0D_{16}$
Error Check Low Byte	CRC Low
Error Check High Byte	CRC High

Table 5: Read ATS Status (01) Query

The response message format is given in Table 6. Each status bit requested is contained in one bit of the data field. The least significant bit of the first data byte contains the status of the starting addressed status bit. Each successive status bit corresponds to the next significant bit in the data field. If the number of status bits to be returned is not a byte (8-bit) multiple, the remaining unused bits in the last data byte are set to logical zeros. The Byte Count field

contains the number of data bytes being returned. A logical one indicates the ON condition while a logical zero indicates the OFF condition.

Response Field Name	Example
Slave Address	08_{16}
Function Code	01_{16}
Byte Count	02_{16}
Data from Status Bits at X (e.g., 1000_{10} through 1003_{10})	03_{16}
Data from Status Bits at X+8	01_{16}
Error Check Low Byte	CRC Low
Error Check High Byte	CRC High

Table 6: Read ATS Status (01) Response

3.2. Function Code 02 – Read Input Status

Function code 02 reads the ON / OFF status of the 5 discrete inputs in the ATC-300+. "ON" means that the particular input feature is activated. "OFF" means that it is not.

Name	Modbus Address	
	Register Number (decimal)	Register Address (hex)
Lockout	2000	7CF
Go To Source 2	2001	7D0
Monitor Mode	2002	7D1
Manual Retransfer	2003	7D2
Emergency Inhibit	2004	7D3

Table 7: Function Code 02 Definitions

The query message format in given in Table 8. The query specifies the starting address (which is always one less than the starting register number) and the quantity of binary inputs to be read.

Query Field Name	Example
Slave Address	34_{16}
Function Code	02_{16}
Starting Address High Byte	07_{16}
Starting Address Low Byte	$D0_{16}$
Number of Points High Byte	00_{16}
Number Of Points Low Byte	03_{16}
Error Check Low Byte	CRC Low
Error Check High Byte	CRC High

Table 8: Read Input Status (02) Query

The response message format is given in Table 9. Each binary input status requested is contained in one bit of the data field. The least significant bit of the first data byte contains the input status of the starting addressed input. Each successive input status bit corresponds to the next significant bit in the data field. If the number of inputs to be returned is not a byte (8-bit) multiple, the remaining unused bits in the last data byte are set to logical zeros. The Byte Count field contains the number of data bytes being returned. A logical one indicates the ON condition while a logical zero indicates the OFF condition.

Response Field Name	Example
Slave Address	34_{16}
Function Code	02_{16}
Byte Count	01_{16}
Data from Binary Inputs at X (e.g., 2000_{10} through 2003_{10})	01_{16}
Error Check Low Byte	CRC Low
Error Check High Byte	CRC High

Table 9: Read Input Status (02) Response

3.3. Function Code 03 - Read Setpoints

Function code 03 reads the setpoints registers.

Setpoints registers have been reserved to hold configuration information parameters that are programmable. Setpoints information starts at register number 3001 (i.e., holding register address $BB8_{16}$). Setpoints are written using function code 16 (10_{16}): Write Setpoints.

The query message format in given in Table 10. The query specifies the starting register address (which is always one less than the starting register number) and the quantity of registers to be read.

Query Field Name	Example
Slave Address	21_{16}
Function Code	03_{16}
Starting Address High Byte	$0B_{16}$
Starting Address Low Byte	$B9_{16}$
Number of Registers High Byte	00_{16}
Number Of Registers Low Byte	02_{16}
Error Check Low Byte	CRC Low
Error Check High Byte	CRC High

Table 10: Read Setpoints Registers (03) Query

The response message format is given in Table 11. The contents of each 16-bit register are returned as two bytes, with the high order byte returned first. The Byte Count field contains the number of data bytes being returned, which is calculated as two times the number of registers requested.

Response Field Name	Example
Slave Address	21_{16}
Function Code	03_{16}
Byte Count	04_{16}
Data from High Byte of Register X (e.g., $0BB9_{16}$)	00_{16}
Data from Low Byte of Register X (e.g., $0BB9_{16}$)	03_{16}
Data from High Byte of Register X+1 (e.g., $0BBA_{16}$)	00_{16}
Data from Low Byte of Register X+1 (e.g., $0BBA_{16}$)	05_{16}
Error Check Low Byte	CRC Low
Error Check High Byte	CRC High

Table 11: Read Setpoints Registers (03) Response

There are 32 Setpoints registers as shown in Table 12.

Name	Register Number (decimal)	Register Address (hex)	Data Range	scale factor	Units
TDES timer	3001	BB8	0 to 120	1	sec
TDNE timer	3002	BB9	0 to 1800	1	sec
TDEN timer	3003	BBA	0 to 1800	1	sec
TDEC timer	3004	BBB	0 to 1800	1	sec
Nominal Frequency	3005	BBC	50 or 60	10	Hz
Nominal Voltage	3006	BBD	120 to 600	1	V
S1 Undervoltage Dropout	3007	BBE	Breaker ATS: 50 to 97% Contactor ATS: 78 to 97%	1	V
S2 Undervoltage Dropout	3008	BBF	Breaker ATS: 50 to 97% Contactor ATS: 78 to 97%	1	V
S1 Undervoltage Pickup	3009	BC0	Breaker ATS: (dropout + 2%) to 99% Contactor ATS: (dropout + 2%) to 99%	1	V
S2 Undervoltage Pickup	3010	BC1	Breaker ATS: (dropout + 2%) to 99% Contactor ATS: (dropout + 2%) to 99%	1	V
S1 Overvoltage Dropout	3011	BC2	Breaker ATS: 105 to 120% Contactor ATS: 105 to 110%	1	V
S2 Overvoltage Dropout	3012	BC3	Breaker ATS: 105 to 120% Contactor ATS: 105 to 110%	1	V
S1 Overvoltage Pickup	3013	BC4	Breaker ATS: 103% to (dropout - 2%) Contactor ATS: 103% to (dropout - 2%)	1	V
S2 Overvoltage Pickup	3014	BC5	Breaker ATS: 103% to (dropout - 2%) Contactor ATS: 103% to (dropout - 2%)	1	V
S1 Underfrequency Dropout	3015	BC6	Breaker ATS: 90 to 97% Contactor ATS: 90 to 97%	10	Hz
S2 Underfrequency Dropout	3016	BC7	Breaker ATS: 90 to 97% Contactor ATS: 90 to 97%	10	Hz
S1 Underfrequency Pickup	3017	BC8	Breaker ATS: (dropout + 1Hz) to 99% Contactor ATS: (dropout + 1Hz) to 99%	10	Hz
S2 Underfrequency Pickup	3018	BC9	Breaker ATS: (dropout + 1Hz) to 99% Contactor ATS: (dropout + 1Hz) to 99%	10	Hz
S1 Overfrequency Dropout	3019	BCA	Breaker ATS: 103 to 110% Contactor ATS: 103 to 105%	10	Hz
S2 Overfrequency Dropout	3020	BCB	Breaker ATS: 103 to 110% Contactor ATS: 103 to 105%	10	Hz
S1 Overfrequency Pickup	3021	BCC	Breaker ATS: 101% to (dropout - 1Hz) Contactor ATS: 101% to (dropout - 1Hz)	10	Hz
S2 Overfrequency Pickup	3022	BCD	Breaker ATS: 101% to (dropout - 1Hz) Contactor ATS: 101% to (dropout - 1Hz)	10	Hz
Reserved	3023	BCE	-	-	-
TDN timer	3024	BCF	0 to 120	1	sec
Modbus Baud Rate	3025	BD0	0 = 9600	1	-
Modbus Address	3026	BD1	1 to 247	1	-

Table 12: Read Setpoints Registers (03) – continued on next page

Name	Register Number (decimal)	Register Address (hex)	Data Range	scale factor	Units
Plant Exerciser Interval	3027	BD2	0 = disabled, 1 = daily, 2 = 7-day, 3 = 14-day, 4 = 28-day	1	-
Plant Exerciser Load Transfer	3028	BD3	0 = disabled, 1 = enabled	1	-
Plant Exerciser Day	3029	BD4	1 = Sunday, 2 = Monday, etc	1	-
Plant Exerciser Hour	3030	BD5	0 to 23	1	hour
Plant Exerciser Minute	3031	BD6	0 to 59	1	min
Test Mode	3032	BD7	0 = no load transfer, 1 = load transfer, 2 = disable test	1	-
Engine Run Time	3033	BD8	0 to 600	1	min
Pretransfer timer	3034	BD9	0 to 120	1	sec
Number of Generators	3035	BDA	0 to 1	1	-
Three phase / Single Phase	3036	BDB	1 or 3	1	-
Voltage Unbalance On/Off	3037	BDC	0 = disabled, 1 = enabled	1	-
Voltage Unbalance Dropout	3038	BDD	5 to 20	1	%
Voltage Unbalance Pickup	3039	BDE	3 to (dropout – 2%)	1	%
Voltage Unbalance Delay	3040	BDF	10 to 30	1	sec
TDEF timer	3041	BE0	0 to 6	1	sec
In-phase Transition On/Off	3042	BE1	0 = disabled, 1 = enabled	1	-
In-phase Transition Freq Difference	3043	BE2	0 to 3	10	Hz
Synchronization timer	3044	BE3	1 to 60	1	min
Phase Reversal On/Off	3045	BE4	0 = off, 1 = ABC, 2 = CBA	1	-
Daylight Savings Time Auto Adjust	3046	BE5	0 = disabled, 1 = enabled	1	-
Manual Re-transfer On/Off	3047	BE6	0 = disabled, 1 = enabled	1	-
Display Language	3048	BE7	0 = English, 1 = French, 2 = Spanish	1	-

Table 12: Read Setpoints Registers (03)

3.4. Function Code 04 – Read Actual Values

Actual Value registers contain dynamic information such as device status and metered values, like voltages and frequencies. Actual value registers are read-only and are accessed using function code 04. Each register is two bytes.

The query message format in given in Table 13. The query specifies the starting register address (which is always one less than the starting register number) and the quantity of registers to be read.

Query Field Name	Example
Slave Address	21_{16}
Function Code	04_{16}
Starting Address High Byte	20_{16}
Starting Address Low Byte	02_{16}
Number of Registers High Byte	00_{16}
Number Of Registers Low Byte	02_{16}
Error Check Low Byte	CRC Low
Error Check High Byte	CRC High

Table 13: Read Actual Value Registers (04) Query

The response message format is given in Table 14. The contents of each 16-bit register are returned as two bytes, with the high order byte returned first. The Byte Count field contains the number of data bytes being returned, which is calculated as two times the number of registers requested.

Response Field Name	Example
Slave Address	21_{16}
Function Code	04_{16}
Byte Count	04_{16}
Data from High Byte of Register X (e.g., 2002_{16})	02_{16}
Data from Low Byte of Register X (e.g., 2002_{16})	58_{16}
Data from High Byte of Register X+1 (e.g., 2003_{16})	01_{16}
Data from Low Byte of Register X+1 (e.g., 2003_{16})	$2C_{16}$
Error Check Low Byte	CRC Low
Error Check High Byte	CRC High

Table 14: Read Actual Values Registers (04) Response

Objects currently assigned to the Actual Value registers are listed in Table 15. There are 32 Actual Value registers. The ATC-300+ only supports fixed point values.

Category	Name	Units	Register Address (decimal)	Register Address (hex)	scale factor	format
Measured Values	S1 V_{AB}	V	6145	1800	1	unsigned
	S1 V_{BC}	V	6146	1801	1	unsigned
	S1 V_{CA}	V	6147	1802	1	unsigned
	S1 Freq	Hz	6148	1803	10	unsigned
	S2 V_{AB}	V	6149	1804	1	unsigned
	S2 V_{BC}	V	6150	1805	1	unsigned
	S2 V_{CA}	V	6151	1806	1	unsigned
	S2 Freq	Hz	6152	1807	10	unsigned
Timers	TDES Timer	seconds	6153	1808	1	unsigned
	TDNE Timer	seconds	6154	1809	1	unsigned
	TDEN Timer	seconds	6155	180A	1	unsigned
	TDEC Timer	seconds	6156	180B	1	unsigned
	TDN Timer	seconds	6157	180C	1	unsigned
	TDEF Timer	seconds	6158	180D	1	unsigned
	Pretransfer Timer	seconds	6159	180E	1	unsigned
	Engine Run Timer	minutes	6160	180F	1	unsigned
	Sync Timer	minutes	6161	1810	1	unsigned
System Counters	S2 Engine Run Time	hours	6162	1811		
	S1 Connect Time	hours	6163	1812		
	S2 Connect Time	hours	6164	1813		
	S1 Available Time	hours	6165	1814		
	S2 Available Time	hours	6166	1815		
	Load Energized Time	hours	6167	1816		
ATS Info	Primary Status	-	6168	1817	1	unsigned
	Number of Transfers	-	6169	1818	1	unsigned
	Cause of Latest Event	-	6170	1819	1	unsigned
Controller Info	Product ID	-	6171	181A	-	encoded
	Hardware Revision	-	6172	181B	-	unsigned
	Firmware Version	-	6173	181C	-	unsigned
	Firmware Revision	-	6174	181D	-	unsigned
	Serial Number - high	-	6175	181E	-	unsigned
	Serial Number – low	-	6176	181F	-	unsigned

Table 15: Function Code 04 Register Map

The Primary Status is contained in the high byte of the register. Decoding is shown in Table 16.

Code	Definition
4	Alarmed
8	Starting
12	Transferred
27	On Good Source

Table 16: Decoding for Primary Status register

The Cause of Latest Event register decoding is shown in Table 17.

Code	Definition
1	Preferred Source became Available
2	Overvoltage
3	Undervoltage
4	Overfrequency
5	Underfrequency
6	Plant Exerciser
7	Engine Test
9	Remote Engine Test
11	Voltage Unbalance
12	Phase Reversal
14	Go To Emergency
15	Lockout
16	Failed to sync (Phase angle)
17	Failed to sync (Freq difference)
18	Monitor Mode
19	Engine Test or Plant Exercise Aborted
20	Source 1 Breaker/Contactor Error
21	Source 2 Breaker/Contactor Error

Table 17: Decoding for Cause of Latest Event Register

3.5. Function Code 05 - Operation Commands

Function code 05 executes an Operation Command by sending the Execute Command data (FF00) to the appropriate register address.

Name	Register Address (decimal)	Register Address (hex)
Reset Number of Transfers	5000	1387
Reset S1 Available Time	5001	1388
Reset S1 Connect Time	5002	1389
Reset S2 Available Time	5003	138A
Reset S2 Connect Time	5004	138B
Reset S2 Engine Run Time	5005	138C
Reset Load Energized Time	5006	138D
Reset Transfer Status	5007	138E
Initiate ATS Test	5008	138F
Cancel ATS Test	5009	1390
Bypass TDNE/TDEN	500A	1391
Manual Retransfer	500B	1392
Go To Emergency	500C	1393
Cancel Go To Emergency	500D	1394

Table 18: Function Code 05 Definitions

The command message format is given in Table 19. This example is for initiating an ATS Test.

Query Field Name	Example
Slave Address	34_{16}
Function Code	05_{16}
Operation Register Address High Byte	13_{16}
Operation Register Address Low Byte	$8F_{16}$
Execute Command High Byte	FF_{16}
Execute Command Low Byte	00_{16}
Error Check Low Byte	CRC Low
Error Check High Byte	CRC High

Table 19: Operation Command (05) Query

The response is an echo to the query as shown in Table 20.

Response Field Name	Example
Slave Address	34_{16}
Function Code	05_{16}
Operation Address Register High Byte	13_{16}
Operation Address Register Low Byte	$8F_{16}$
Execute Command High Byte	FF_{16}
Execute Command Low Byte	00_{16}
Error Check Low Byte	CRC Low
Error Check High Byte	CRC High

Table 20: Operation Command (05) Response

3.6. Function Code 16 - Write Setpoints

Function code 16 provides the capability to write setpoints to ATC-300+. The entire setpoints buffer (48 registers) must be written.

The query message format in shown in Table 21. The query specifies the starting register address ($BB8_{16}$), the number of registers to be written to (30_{16}), the number of data bytes to follow (60_{16}) and the setpoint data values of the registers. The setpoint values for each 16-bit register are transmitted as two bytes, with the high order byte transmitted first. .

Query Field Name	Example
Slave Address	42_{16}
Function Code	10_{16} (16_{10})
Starting Register Address High Byte (TDES Setpoint)	$0B_{16}$
Starting Register Address Low Byte (TDES Setpoint)	$B8_{16}$
Number of Registers High Byte	00_{16}
Number Of Registers Low Byte	30_{16}
Byte Count	60_{16}
TDES Setpoint Data High Byte	00_{16}
TDES Setpoint Data Low Byte	03_{16}
TDNE Setpoint Data High Byte	00_{16}
TDNE Setpoint Data Low Byte	05_{16}
: :	
Display Language Data High Byte	00_{16}
Display Language Data Low Byte	00_{16}
Error Check Low Byte	CRC Low
Error Check High Byte	CRC High

Table 21: Write Setpoints (16) Query

The response message format in given in Table 22. The response echoes the starting register address and the number of setpoint registers from the query message.

Response Field Name	Example
Slave Address	42_{16}
Function Code	10_{16} (16_{10})
Starting Register (TDES) Address High Byte	$0B_{16}$
Starting Register (TDES) Address Low Byte	$B8_{16}$
Number of Registers High Byte	00_{16}
Number Of Registers Low Byte	30_{16}
Error Check Low Byte	CRC Low
Error Check High Byte	CRC High

Table 22: Write Setpoints (16) Response

The setpoints registers and data ranges are shown in Table 23. There are 48 setpoints registers. Each one is two bytes.

Name	Register Number (decimal)	Register Address (hex)	Data Range	scale factor	Units
TDES timer	3001	BB8	0 to 120	1	sec
TDNE timer	3002	BB9	0 to 1800	1	sec
TDEN timer	3003	BBA	0 to 1800	1	sec
TDEC timer	3004	BBB	0 to 1800	1	sec
Nominal Frequency	3005	BBC	50 or 60	10	Hz
Nominal Voltage	3006	BBD	120 to 600	1	V
S1 Undervoltage Dropout	3007	BBE	Breaker ATS: 50 to 97% Contactor ATS: 78 to 97%	1	V
S2 Undervoltage Dropout	3008	BBF	Breaker ATS: 50 to 97% Contactor ATS: 78 to 97%	1	V
S1 Undervoltage Pickup	3009	BC0	Breaker ATS: (dropout + 2%) to 99% Contactor ATS: (dropout + 2%) to 99%	1	V
S2 Undervoltage Pickup	3010	BC1	Breaker ATS: (dropout + 2%) to 99% Contactor ATS: (dropout + 2%) to 99%	1	V
S1 Overvoltage Dropout	3011	BC2	Breaker ATS: 105 to 120% Contactor ATS: 105 to 110%	1	V
S2 Overvoltage Dropout	3012	BC3	Breaker ATS: 105 to 120% Contactor ATS: 105 to 110%	1	V
S1 Overvoltage Pickup	3013	BC4	Breaker ATS: 103% to (dropout - 2%) Contactor ATS: 103% to (dropout - 2%)	1	V
S2 Overvoltage Pickup	3014	BC5	Breaker ATS: 103% to (dropout - 2%) Contactor ATS: 103% to (dropout - 2%)	1	V
S1 Underfrequency Dropout	3015	BC6	Breaker ATS: 90 to 97% Contactor ATS: 90 to 97%	10	Hz
S2 Underfrequency Dropout	3016	BC7	Breaker ATS: 90 to 97% Contactor ATS: 90 to 97%	10	Hz
S1 Underfrequency Pickup	3017	BC8	Breaker ATS: (dropout + 1Hz) to 99% Contactor ATS: (dropout + 1Hz) to 99%	10	Hz
S2 Underfrequency Pickup	3018	BC9	Breaker ATS: (dropout + 1Hz) to 99% Contactor ATS: (dropout + 1Hz) to 99%	10	Hz
S1 Overfrequency Dropout	3019	BCA	Breaker ATS: 103 to 110% Contactor ATS: 103 to 105%	10	Hz
S2 Overfrequency Dropout	3020	BCB	Breaker ATS: 103 to 110% Contactor ATS: 103 to 105%	10	Hz
S1 Overfrequency Pickup	3021	BCC	Breaker ATS: 101% to (dropout - 1Hz) Contactor ATS: 101% to (dropout - 1Hz)	10	Hz
S2 Overfrequency Pickup	3022	BCD	Breaker ATS: 101% to (dropout - 1Hz) Contactor ATS: 101% to (dropout - 1Hz)	10	Hz
Reserved	3023	BCE	-	-	-
TDN timer	3024	BCF	0 to 120	1	sec

Table 23: Write Setpoints Registers (16) – continued on next page

Name	Register Number (decimal)	Register Address (hex)	Data Range	scale factor	Units
Modbus Baud Rate	3025	BD0	0 = 9600	1	-
Modbus Address	3026	BD1	1 to 247	1	-
Plant Exerciser Interval	3027	BD2	0 = disabled, 1 = daily, 2 = 7-day, 3 = 14-day, 4 = 28-day	1	-
Plant Exerciser Load Transfer	3028	BD3	0 = disabled, 1 = enabled	1	-
Plant Exerciser Day	3029	BD4	1 = Sunday, 2 = Monday, etc	1	-
Plant Exerciser Hour	3030	BD5	0 to 23	1	hour
Plant Exerciser Minute	3031	BD6	0 to 59	1	min
Test Mode	3032	BD7	0 = no load transfer, 1 = load transfer, 2 = disable test	1	-
Engine Run Time	3033	BD8	0 to 600	1	min
Pretransfer timer	3034	BD9	0 to 120	1	sec
Number of Generators	3035	BDA	0 to 1	1	-
Three phase / Single Phase	3036	BDB	1 or 3	1	-
Voltage Unbalance On/Off	3037	BDC	0 = disabled, 1 = enabled	1	-
Voltage Unbalance Dropout	3038	BDD	5 to 20	1	%
Voltage Unbalance Pickup	3039	BDE	3 to (dropout – 2%)	1	%
Voltage Unbalance Delay	3040	BDF	10 to 30	1	sec
TDEF timer	3041	BE0	0 to 6	1	sec
In-phase Transition On/Off	3042	BE1	0 = disabled, 1 = enabled	1	-
In-phase Transition Freq Difference	3043	BE2	0 to 3	10	Hz
Synchronization timer	3044	BE3	1 to 60	1	min
Phase Reversal On/Off	3045	BE4	0 = off, 1 = ABC, 2 = CBA	1	-
Daylight Savings Time Auto Adjust	3046	BE5	0 = disabled, 1 = enabled	1	-
Manual Re-transfer On/Off	3047	BE6	0 = disabled, 1 = enabled	1	-
Display Language	3048	BE7	0 = English, 1 = French, 2 = Spanish	1	-

Table 23: Write Setpoints Registers (16)

3.7. Exception Codes

Under certain circumstances, the ATC-300+ will return an exception code. The exception codes are shown in Table 24.

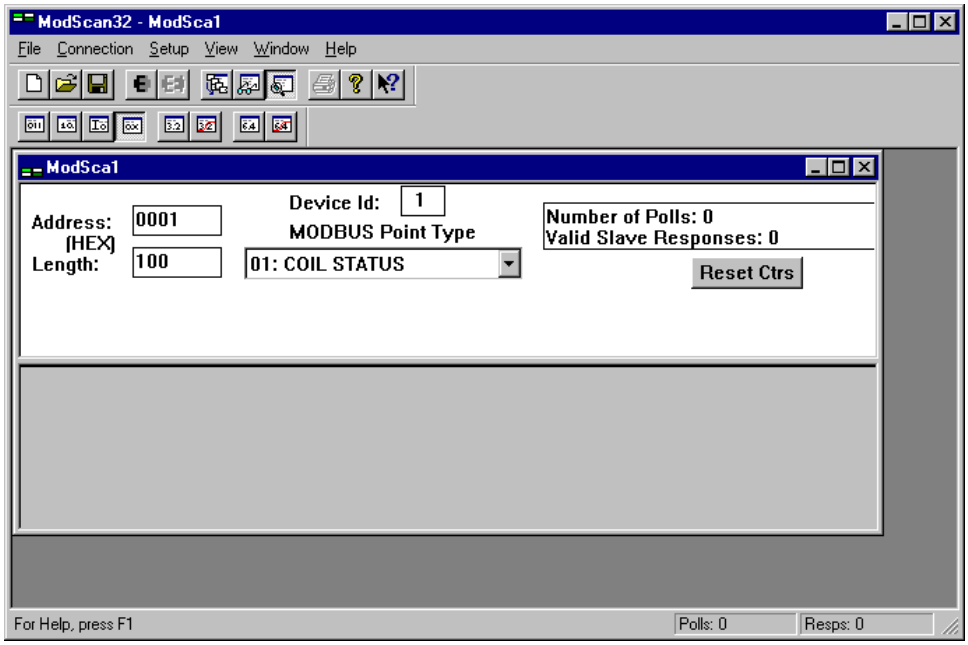
Exception Code (hex)	Description
01	Invalid Function
02	Invalid Register
03	Invalid Data
84	Partial Register Access Error

Table 24: Exception Codes

Appendix A

Using WinTECH ModScan32 Software with the ATC-300+

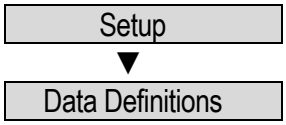
The software displays the ModScan32 window with the Toolbar, Status Bar and Display Bar all selected (in the View menu). A sub-window (ModSca1) allows the construction of an outgoing message and a data transmission/reception box.



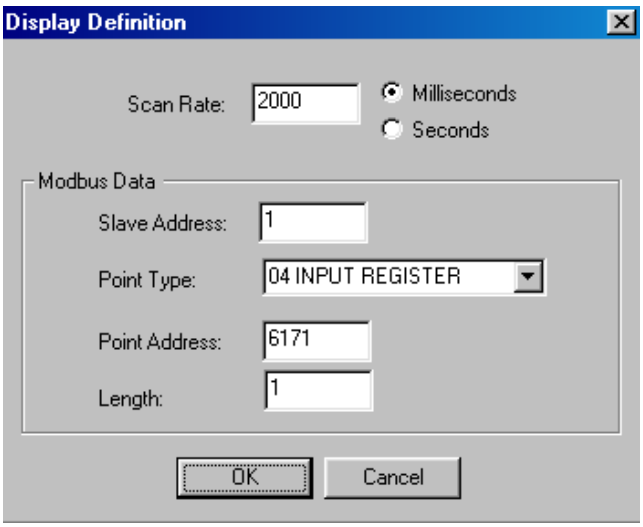
The main menu and its submenu items are:

File	Connection	Setup	View	Window	Help
New	Connect	Data Definitions	Toolbar	Cascade	Help Topics
Custom Form ▶	Disconnect	Display Options ▶	Status Bar	Tile	About ModScan32…
Open…	Auto-Start ▶	Extended ▶	Display Bar	Arrange Icons	
Close	QuickConnect	Text Capture	Config		
Save		Dbase Capture			
Save As…		Capture Off			
Print…		Reset Ctrs			
Print Setup…					
Recent File					
Exit					

In the Setup main menu item, select the Data Definitions submenu item to setup an outgoing message:



Configure this Display Definition window as shown to read the Product ID Actual Value register and click OK:

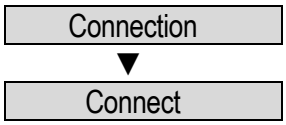


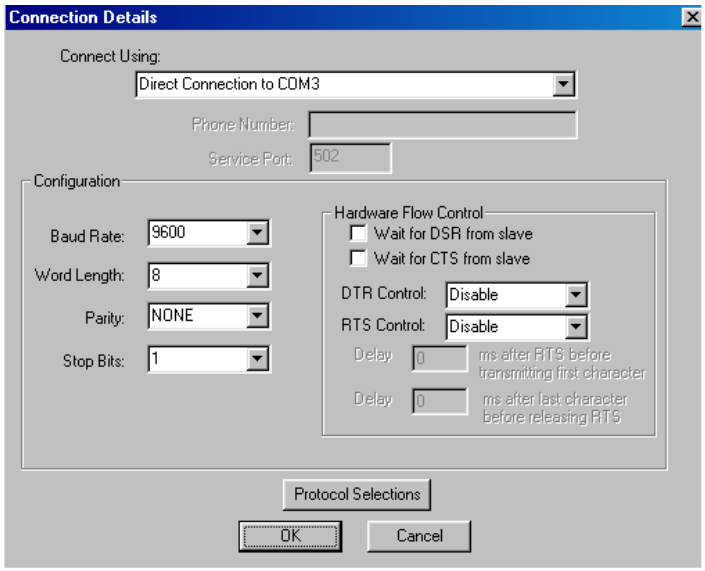
Again, in the Setup main menu item, configure the Display Options submenu item as follows:

Setup	
Display Options ►	Show Data
	[x] Show Traffic
	Binary
	[x] Hex
	Unsigned Decimal
	Integer
	Floating Pt
	Swapped FP
	Dbl Float
	Swapped Dbl
	[x] Hex Addresses

where [x] - denotes a checked (enabled) menu selection.

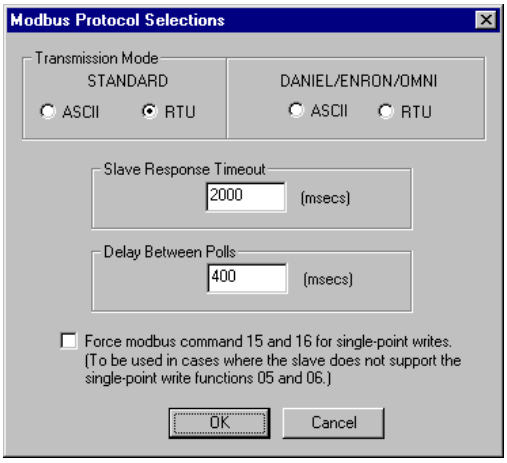
In the Connection main menu item, select the Connect submenu item to configure the serial connection:





Configure this Connection Details dialog box as shown, selecting the proper serial (COM) port for your PC:

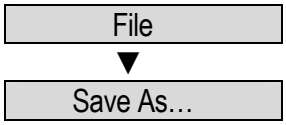
Next click the Protocol Selections button and configure the Modbus Protocol Selections dialog box as shown:



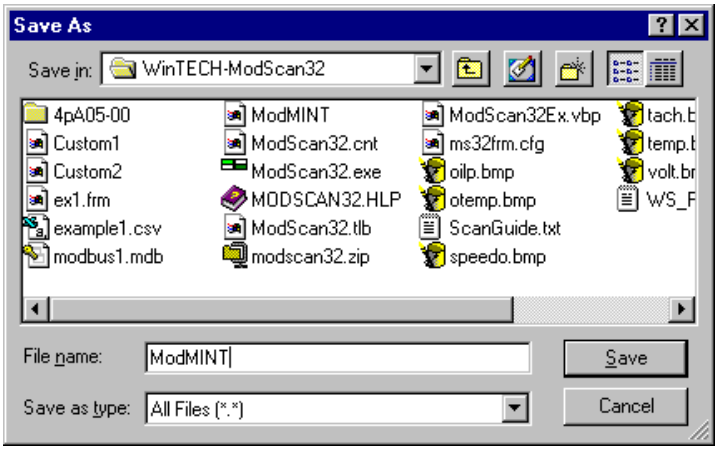
Note: The Slave Response Timeout needs increased to 8000 msec if the Write Setpoints function is going to be used.

Click OK for each dialog box.

Finally, in the File main menu item, select the Save submenu item:



Enter the File name ATC300+ then click Save to store the newly configured information:



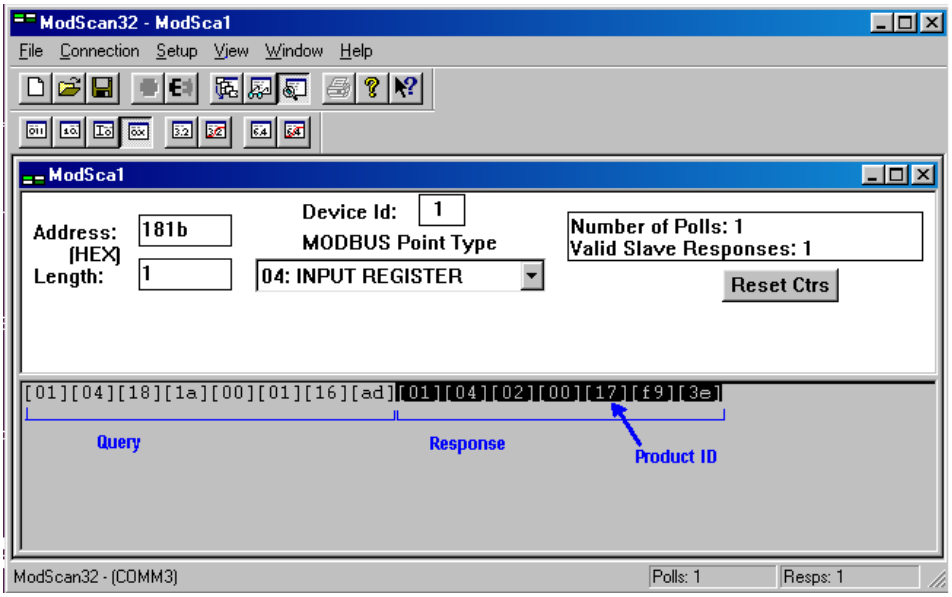
Appendix B

Modscan Examples for Function Codes 01, 02, 03, 04

The following screenshot shows a query/response for reading the ATC-300+ Product ID from the Actual Value registers (Function Code 04).

The Product ID for the ATC-300+ is 23 which is 17_{16}.

The register address is 181A but Modbus requires that the register number (which is one more than the register address) is entered in the Address box below. Therefore, 181B is entered and Modbus transmits the address of [18][1A] in the query message.



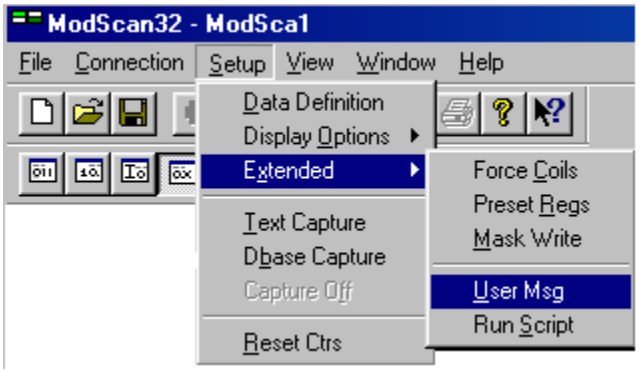
Function codes 01 (Read ATS Status Bits), 02 (Read Input Status), and 03 (Read Setpoints) are implemented the same way with Modscan. Each one can be selected using the “MODBUS Point Type” pulldown box.

Appendix C

Sending Modbus Queries via Modscan User Defined Messages

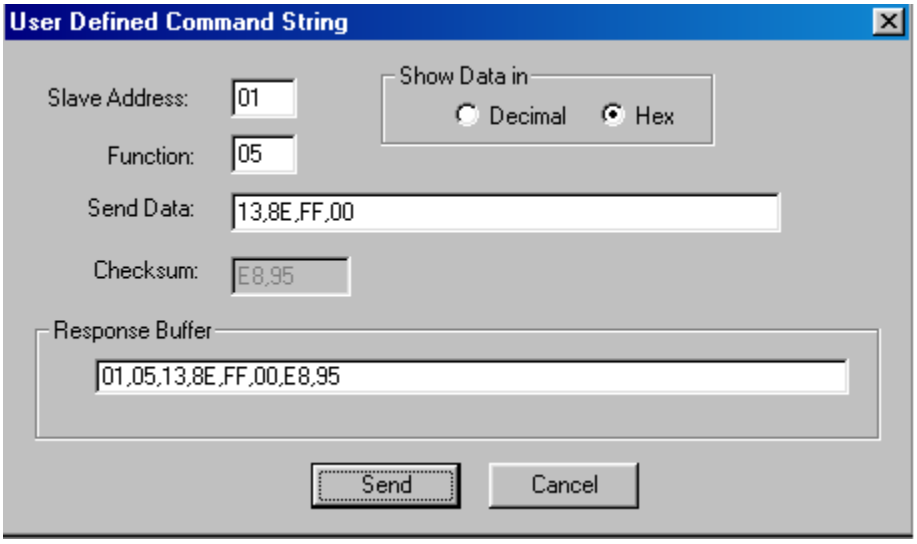
All Modbus queries can also be sent to the ATC-300+ using the User Defined Command String.

The following screenshot shows how to select User Msg for transmitting queries in Modscan.



This screenshot shows an example of how to send the Operation Command (Function Code 05) to reset the Transfer Status. The register address for Reset Transfer Status is 138E and this is what gets entered into the Send Data Field when using the User Defined Command String.

The Checksum box gets automatically filled in by Modscan.

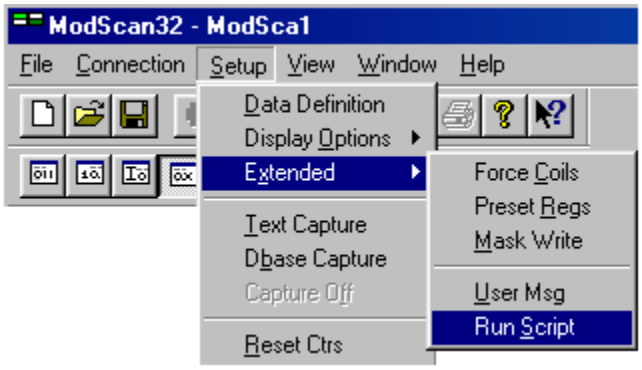


Appendix D

Writing Setpoints (Function Code 16) using a Modscan Test Script

This example shows how to write the Setpoints buffer to the ATC-300+ using the Modscan Test Script method. The Test Script may be written as a .csv file.

The following screenshot shows how to select the Test Script method.



An example script for Writing Setpoint is shown on the next page.

Before executing the Writing Setpoints script, it is necessary to make sure that the The Slave Response Timeout is increased to 8000 msec. This is found under Protocol Selection in the Connection Details menu.

```
// Modbus Test Script for Writing Setpoints to the ATC-300+
//
// Each Script entry consists of the following comma delimited data fields:
// TEST NAME, NODE, FUNCTION, ADDRESS, LENGTH, DATA, CONTROL CODE
//
// Double slashes on the front of a line denote comments. Don't leave a blank line.
//
// The following Control Codes may be used (i.e. last field on each line)
//                  \ -- Continue DATA fields on next line
//                  C -- Generate Bad CRC message to slave
//                  D -- Check response data quantity only, (ignore actual data)
//                  1 -- Expect Exception Response 01
//                  2 -- Expect Exception Response 02
//                  4 -- Expect Exception Response 04
//                  R -- Expect no Response
//                  T (default) -- Verify Response Data
//
//
//          ------------------------------------------------
//          ATC-300 Write Sepoints script (Function Code 16
//          ------------------------------------------------
//
// Test:   TEST NAME, NODE, FUNCTION CODE, STARTING ADDR HI, STARTING ADDRESS LO,
//         NUMBER REGS HI, NUMBER REGS LO, BYTE COUNT,SETPOINT DATA REGISTERS - QTY 48
//
//Setpoint data registers are 16 bits.  Need to enter into script as 32 bits as
//follows:
//                                Setpoint 1 = 0x1234
//                                Setpoint 2 = 0x5678
//
//           Enter into script as 0x56781234 and Modscan will send as 12,34,56,78
//
1. Write ATC-300 Setpoints,1,16,0x0BB9,48,\
,0x00050005,\
,0x00030004,\
,0x00780258,\
,0x00600060,\
,0x00660066,\
,0x00840084,\
,0x00820082,\
,0x023A023A,\
,0x02460246,\
,0x02760276,\
,0x02640264,\
,0x00050000,\
,0x00010000,\
,0x00010002,\
,0x00010001,\
,0x00010000,\
,0x00050001,\
,0x00010001,\
,0x000A0000,\
,0x000A0007,\
,0x00000003,\
,0x0005000A,\
,0x00010000,\
,0x00000000,T
//
// END OF SCRIPT
end
```