MODEL 2000 SERIAL COMMUNICATIONS USER MANUAL

Manual 9, Revision 2

Russelectric Inc.

MODEL 2000 SERIAL COMMUNICATIONS USER MANUAL

Manual 9, Release 1, Rev. 2

Foreword

The following pages describe the Russelectric Model 2000 serial communications procedures and installation, with specific descriptions on how the system operates. The manual is shipped with the most recent date. If changes are made to the manual, pages within the manual are dated to indicate the most recent updated pages. If the entire manual is extensively updated, a revision change occurs, and the manual displays the next revision number.

Revision	Date	Changes Made
Preliminary Release	31 May 96	
Release 1	17 June 96	
Release 1, Rev. 1	27 Feb. 97	Updates to the cable diagrams
Release 1, Rev. 2	27 Oct. 97	Updates to the cable diagrams

This manual consists of the following sections:

Front Matter

- Title Page
- Foreword
- Table of Contents
- List of Figures

Introduction

- About This Manual
- Company and Field Service Information

Section 1. Quick Guide to the Manual

Provides an overview of the manual, ATS communications, and the RTU communication protocol.

Section 2. Model 2000 Serial Communications

This section describes the serial communications for the Model 2000.

Section 3. RTU Protocol

This section contains information regarding the RTU protocol.

Section 4. Building Serial Communication Cables

This section contains necessary information to build serial communication cables.

Section 5. Appendix

The section contains additional relevant information for the serial communications manual such as a glossary (acronyms, abbreviations, and terminology) and a complete ASCII code list.

Table of Contents

List of Figures

RUSSELECTRIC INC.

South Shore Park, Hingham, MA 02043-4387

(617) 749-6000

***

FIELD SERVICE INFORMATION

If a problem arises with your system, please contact a Field Service Representative at (617) 749-6000 or the 24-hour emergency number (800) 654-3020.

Telefax (617) 749-8077

NOTICE

The information in this manual is subject to change. This manual is believed to be complete and accurate at the time of publication and no responsibility is assumed for any errors that may appear. In no event shall Russelectric Inc. be liable for incidental or consequential damages in connection with or arising from the use of the manual and its accompanying related materials.

1.0 Overview

This manual describes how to use the serial communications port (SCP) from both hardware and software points of view. Hardware topics include serial port setup, and constructing and installing the serial communications cables. Software topics include configuring the Model 2000 and the embedded communications protocols.

In addition, this manual describes some of the serial data communication fundamentals that relate directly to the ATS SCADA software available from Russelectric.

ATS Communications

The Model 2000 ATS contains two serial ports:

- One port provides serial interfaces to SCADA nodes and other MMI (man-machine interface) devices. This port supports RS-422 in full-duplex mode.
- The other port provides serial interfaces to optional I/O devices such as power monitoring transducers. This port also supports RS-422 in full-duplex mode. (Future use only.)

RTU Communications Protocol

Remote terminal unit (RTU) protocol is included in the EPROM. The RTU protocol, as implemented in the Model 2000, is a subset of the Modbus remote terminal unit serial communications protocol provided for slave operation only. (The slave unit is the responding device in a master/ slave communication system.) After configuration, the RTU protocol is enabled. The RTU protocol uses the RS-422 electrical standard. Data rates, stop bits, parity, and RTU addresses can be configured manually.

2.0 Serial Communication for the Model 2000

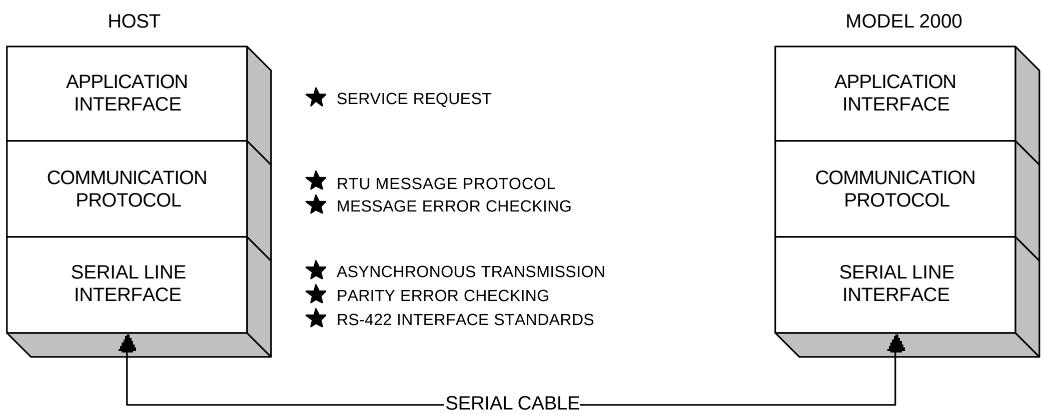
This section provides a brief overview of Model 2000 serial communications. The subject of serial communications covers a wide variety of topics from application programs, to communication protocols, to hardware interfaces. To organize this information, a serial communications model is shown to give an overview of Model 2000 serial communications. Subsequent sections describe this communications model in greater detail.

NOTE:* *The reader should have some familiarity with the binary and hexadecimal numbering systems and a basic understanding of the Model 2000. The information in this section is intended as background information only. Specific information on serial communications and related topics can be found in later sections.

2.0.1 Communications Reference Model

The figure below shows the main functions of serial communications between a host computer and a Model 2000, or between a host computer and multiple Model 2000s. An example of a typical message transfer is included later on to clarify the concepts of the reference model.

Figure 1. The Model 2000 Reference Model



2.0.2 The Serial Line Interface

The serial line interface encodes/decodes messages according to a particular information code, is responsible for the electrical transmission of messages over the serial line, and performs parity checking of each character received over the serial line. The drivers for the transmission line are also part of the serial line interface.

This section covers the following topics:

- Information Codes
- Transmission Errors and Detection
- Asynchronous Transmission
- Serial Line Interface Standards (RS-232 and RS-422).

Information Codes

An information code is a standard by which numbers, letters, symbols, and control characters are presented for serial transmission. In the RTU protocol, message characters and data are sent as 8-bit binary coded data.

There are a number of different coding schemes used today, but the most common and the type used in RTU communications is the American Standard Code for Information Interchange or ASCII code. In the figure below, the RTU protocol uses a message constructed with an 8-bit character code plus two 8-bit error checking codes.

Figure 2. The Model 2000 Protocol for Serial Transmission

		MSB			DATA BITS				LSB	
10	9	8	7	6	5	4	3	2	1	0
STOP	PARITY (OPTIONAL)									START

Transmission Errors and Detection

The RTU protocol uses error checking and detection mechanisms to reduce the number of transmission errors and ensure reliable transfer of data. The error checking methods used by the RTU protocol are described in ***Section 3***, *RTU Protocol Definition*.

Parity Checking

Parity checking can be generally specified as even, odd, or none. The parity bit, derived by the sender and monitored by the receiver, is dependent on the number of 1s occurring in the binary character. If parity is defined as odd, the total number of 1s in the binary character (in addition to the parity bit) must be odd. If the parity is even, the total number of 1s in the character including the parity bit must be even. If the parity is none, no parity checking is performed.

In the example shown below, the ASCII coded 'A' contains two 1s; therefore, the parity bit must be 1 for odd parity. The parity bit would be 0 in the case parity were defined as even. The parity bit is not transmitted when the ‘no parity’ or ‘none’ option is selected. For the RTU protocol, the parity may be odd, even or none. If parity checking is employed, and one of the bits is transmitted incorrectly, the parity bit reflects the error.

Figure 3. The ASCII Sequence with Error Checking

ASCII CHARACTER A RECEIVED CORRECTLY

	DATA BITS								
PARITY BIT (ODD)	8	7	6	5	4	3	2	1	
1	0	1	0	0	0	0	0	1	Received Data Bytes

ASCII CHARACTER A RECEIVED WITH AN ERROR IN THE FIRST BIT

	DATA BITS								
PARITY BIT (ODD)	8	7	6	5	4	3	2	1	
1	0	1	0	0	0	0	0	0	Received Data Bytes

The receiver monitors the parity bit and detects the error in transmission because the received character with parity has an even number of 1s instead of an odd number.

If, on the other hand, an even number of bits in a character is transmitted incorrectly, the parity bit will not reflect the error.

Figure 4. The ASCII Sequence with Error Checking (Two Incorrect Bits)

ASCII CHARACTER A RECEIVED WITH AN ERROR IN THE FIRST TWO BITS

	DATA BITS								
PARITY BIT (ODD)	8	7	6	5	4	3	2	1	
1	0	1	0	0	0	0	1	0	Received Data Bytes

The parity bit does not reflect the error because the received character with parity shows an odd number of 1s as it is supposed to.

Transmission Timing Errors

Timing problems between transmitter and receiver can produce other kinds of errors such as overrun, framing, and time-out errors.

Overrun Errors

Timing problems between the transmitter and receiver cause characters to be sent faster than the receiver can handle them – this produces a situation known as an overrun. In this case, the previous character is overwritten and an error is indicated.

Framing Errors

In asynchronous transmission, this type of error occurs when the receiver mistakes a logic 0 data bit or a noise burst for a start bit. The error is detected because the receiver knows which bit after the start bit must be a logic 1 stop bit. In the case where the start bit is really a data bit, and the expected stop bit is not the stop bit (but a start or data bit) then the framing error is reported.

Time-out Errors

This type of error checking is performed by the communication protocol. Time-outs are used to ensure that timely communications exist between devices. When a source device initiates a communication, the target must respond within a certain amount of time or a time-out condition occurs, causing the communication to be aborted.

Asynchronous Transmission

Asynchronous serial transmission is used in the Model 2000. Although there is no synchronizing clock used, the transmitting and receiving equipment must be operating at the same bit rate or errors occur as mentioned in the previous section. The general format for asynchronous communications includes a start bit, eight data bits, an optional parity bit, and a stop bit.

Figure 5. Serial Data Format (Initially Received Data)

SERIAL DATA FORMAT										
BIT 0	BIT 1	BIT 2	BIT 3	BIT 4	BIT 5	BIT 6	BIT 7	BIT 8	BIT 9	BIT 10
START	LSB	←		ACTIVE DATA BITS			→	MSB	PARITY	STOP
0	←			1	or	0		→		1

When the receiver detects the leading edge of the start bit, which is always logic 0, a timer is triggered to allow sampling to occur in the middle of each bit. After the last data bit (or the parity bit) has been received, the logic state of the line must be a 1 for at least one bit-time before receiving the next character. If no more characters are sent, the line is maintained in the 1 state.

Figure 6. Serial Data Format (Additionally Received Data)

LOGIC 1	START BIT	MSB BIT 1	BIT 2	BIT 3	DATA BITS BIT 4	BIT 5	BIT 6	BIT 7	LSB BIT 8	PARITY BIT (Optional)	STOP BIT
LOGIC 0		FIRST ←			ORDER OF TRANSMISSION				→ LAST		

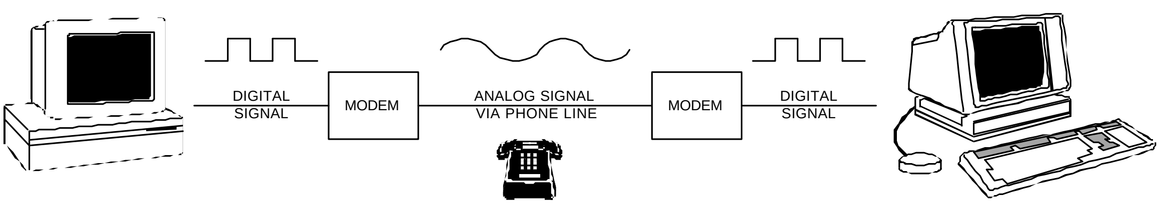
Serial Communications Line

The serial communications line is the physical medium over which messages travel. The line may be a direct connection between devices or a connection through modems for long distance communications. The characteristics of the communications line depend on the requirements of the user and the electrical interface standard to which the line is constructed.

Modems

The word modem is an acronym for MOdulator/DEModulator. A modem is a device that converts data from digital to analog signals (for transmitting) and from analog to digital signals (for receiving) over telephone communications lines. Some modems utilize other methods of transmission, such as radio or microwave signals.

Figure 7. Computer Connections via Modems



Modems are generally classified as to the type of telephone line facility that can be connected, half-duplex or full-duplex, synchronous or asynchronous, modulation technique for the analog signal, and the maximum data rate in bits per second (bps). Modems were originally designed for, and are most frequently used with the RS-232D interface standard.

Communications Modes

There are three modes of communication:

- **Simplex**

  Information is sent over a communications line in one direction only.

- **Half-duplex**

  Information is sent in both directions over a communications line, but only one direction at a time.

- **Full-duplex**

  Information is sent over a communications line in both directions at the same time.

Interface Standards

An interface standard is a set of rules that define the signal characteristics, cable and connection characteristics, connector pin assignments, and control sequences for a physical link between devices. The communications used by the Model 2000, personal computers (PCs), and modems are based on the interface standards explained below.

RS-232 Standard

This standard was developed for interconnecting data terminal equipment (DTE), such as printers, CRTs, or computers, to data communications equipment (DCE), such as a modems, for transmission over a telephone line or network. It can, however, be used over short distances without a modem. Electrically, RS-232 can be described as an unbalanced or single-ended voltage interface. This means that all the interchange signals share a common electrical ground.

The basic characteristics of the RS-232 standard are:

- Maximum cable length: 50 feet (15 meters)
- Maximum data rate: 20 kilobits/second (20 kB/sec.)
- Logic assignments referenced to signal ground:

  Space or logic 0: +3 V to +25 V

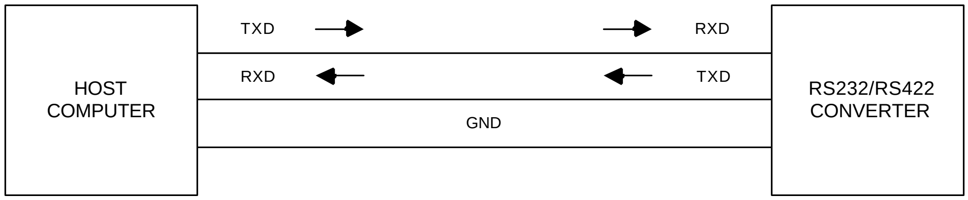
  Mark or logic 1: -3 V to -25 V
- Uses 25-pin D-type connector
- Includes 21 interchange circuits including data transmit and receive, data control, and timing. The most commonly used circuits are shown in the figure below.

Figure 8. Standard RS-232 Communication Interface Signals

PIN #	FUNCTION	ABBREV.	TYPE	DIRECTION
1	PROTECTIVE GROUND	PROT GND	-	-
2	TRANSMIT DATA	TXD	DATA	FROM DTE
3	RECEIVE DATA	RXD	DATA	TO DTE
4	REQUEST TO SEND	RTS	CONTROL	FROM DTE
5	CLEAR TO SEND	CTS	CONTROL	TO DTE
6	DATA SET READY	DSR	CONTROL	TO DTE
7	SIGNAL GROUND	GND	-	-
8	RECEIVED LINE SIGNAL DETECT OR DATA CARRIER DETECT	RLSD or DCD	CONTROL	TO DTE
20	PROTECTIVE GROUND	DTR	CONTROL	FROM DTE

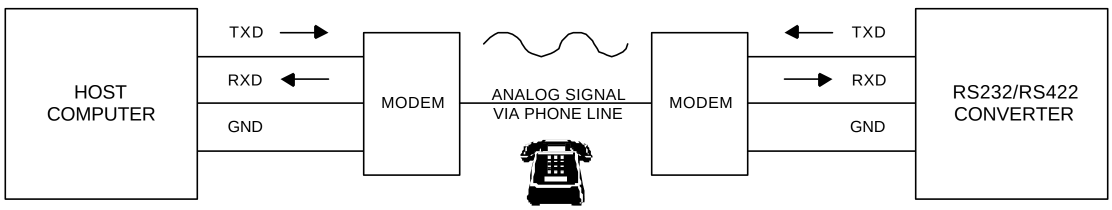
The RS-232 interface can be used for direct connections not exceeding 50 feet (15 meters). The following illustration shows the lines required for both devices to transmit and receive.

Figure 9. RS-232 Direct Connection Without Flow Control



In the above case there is no data flow control – that is, both devices can transmit at any time and there is no check of the communications line before transmission. When modems are used, without data flow control, both devices can transmit at any time and there is no check of the transmission line or that a carrier is present.

Figure 10. RS-232 Modem Connection Without Flow Control



When flow control is desired, the RTS and CTS control circuits is used to permit the following:

- RTS (Requested To Be Sent) – The transmitting device can signal the transmitting modem that data is requested to be sent.
- CTS (Clear To Send) – The transmitting modem can signal back to the transmitting device that it is clear to send the data.

Refer to ***Section 4*** for information on interconnecting host computers to one or more Model 2000s using modems. For a complete explanation of control signal usage with modems, as well as the electrical and mechanical characteristics of the interface, see electrical interface standard (EIA) RS-232D and the user's manual of the modem used in the communication configuration.

RS-449, RS-422, and RS-485 Standards

RS-449, RS-422, and RS-485 comprise a family of standards reflecting advances in integrated circuit technology. The new standards permit greater distance between equipment and a higher maximum data rate.

RS-422 and RS-485 are standards which define electrical interface characteristics. RS-449 is a standard that is used in conjunction with RS-422 and RS-485, that defines the connector pin assignments, cable and connector characteristics, and control signal sequences.

In addition, the RS-422 and RS-485 standards use balanced (differential voltage) interfaces to isolate the signal lines from the ground. One of the interface options that can be used in Model 2000 serial communications is based on the RS-422, RS-485, and RS-449 standards. The basic characteristics of RS-422, RS-485, and RS-449 (referenced as RS-485 in this manual) are:

- Maximum cable length: 4000 feet (1200 meters).

- Maximum data rate: 1 Megabits/second (1 MB/sec.) at 4000 feet and 10 Megabits/second (10 MB/sec.) at 40 feet (12 meters).
- Logic assignments; differential inputs not referenced to ground:

  Space or logic 0: Circuit A is +200 mV to +6V with respect to circuit B.

  Mark or logic 1: Circuit A is -200 mV to -6V with respect to circuit B.
- 25-pin D-type connector.
- 30 interchange circuits.

For a complete explanation of the electrical and mechanical characteristics of these interfaces, see EIA Standards RS-449, RS-422, and RS-485.

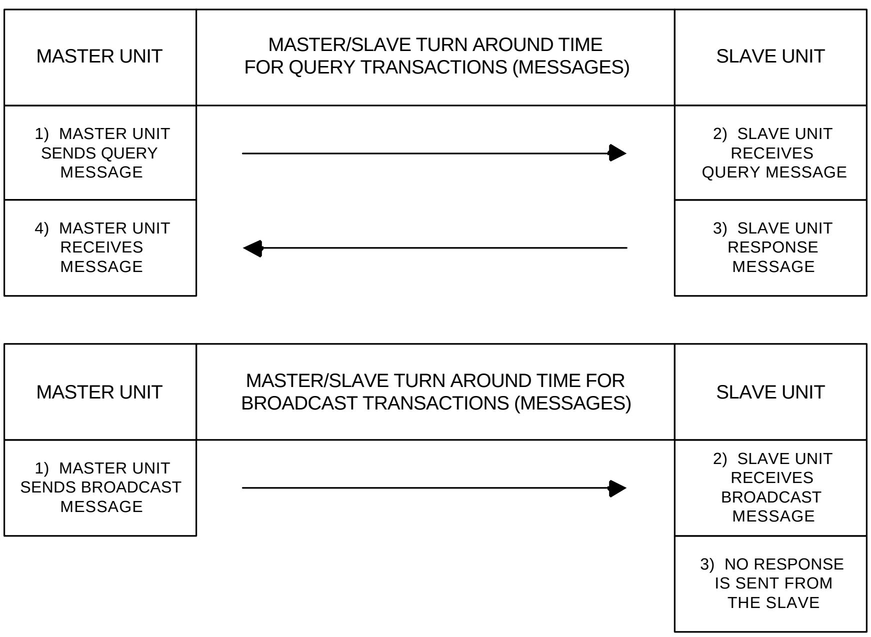
2.0.3 The Communication Protocol

The communication protocol defines the sequence for establishing and relinquishing message connections and for processing messages according to a set of rules or a protocol. For the Model 2000, the only protocol available is RTU. (This protocol is discussed in detail in ***Section 3***). The communication protocol is also responsible for error checking at the message level.

RTU Communication Protocol

The Modbus remote terminal unit (RTU) protocol is a master/slave (query/response) protocol. It is used to link the Model 2000 with a computer, or other intelligent device that supports the RTU protocol. Only the master device can initiate a communications request when RTU mode is used. The Model 2000 can be configured only as a slave RTU device.

Figure 11. RTU Message Protocol



A distinction is made between the two communicating devices. The device that initiates a data transfer is called the master and the other device is called the slave. The Model 2000 can only be an RTU slave.

The master device begins a data transfer by sending a query or broadcast request message. A slave completes that data transfer by sending a response message if the master sent a query message addressed to it. No response message is sent when the master sends a broadcast request. The time between the end of a query and the beginning of the response to that query is called the slave turnaround time. The message fields for a typical RTU message are shown on the following page.

Figure 12. RTU Message Transfers

MESSAGE FRAME			
STATION ADDRESS	FUNCTION CODE	INFORMATION	ERROR CHECK

Cyclic Redundancy Check

The error check field on an RTU message contains a cyclic redundancy check (CRC-16) code. The details of generating this code are found in ***Section 3***, *RTU Protocol*, ***Subsection 3.0.8***, *Cyclic Redundancy Check (CRC)*.

2.0.4 Communication Networks

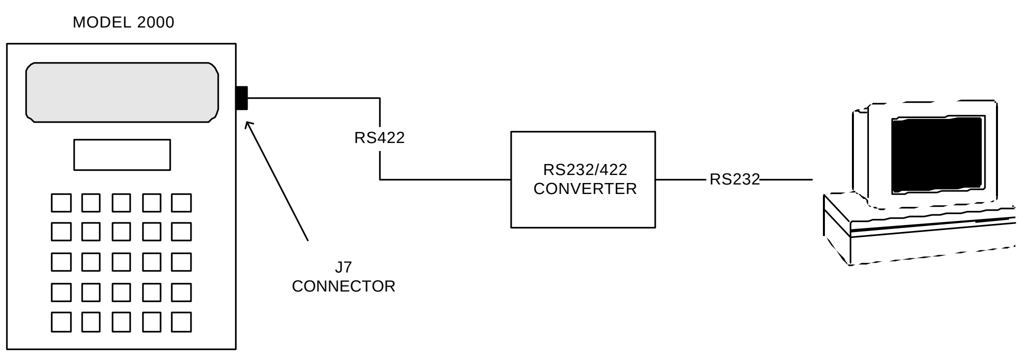
The term network (system) configuration refers to the way in which computers, terminals, and communication equipment are interconnected. The following data communications system configurations are possible with the Model 2000:

- Point-to-Point (Master/Slave)
- Multidrop (Single Master/Multiple Slaves)
- Modem Transmission.

Point-To-Point

Point-to-point connection is the simplest type of system configuration. When using this method, only two devices are connected via master/slave mode, but only the master device can initiate communications. The following figure illustrates point-to-point configuration:

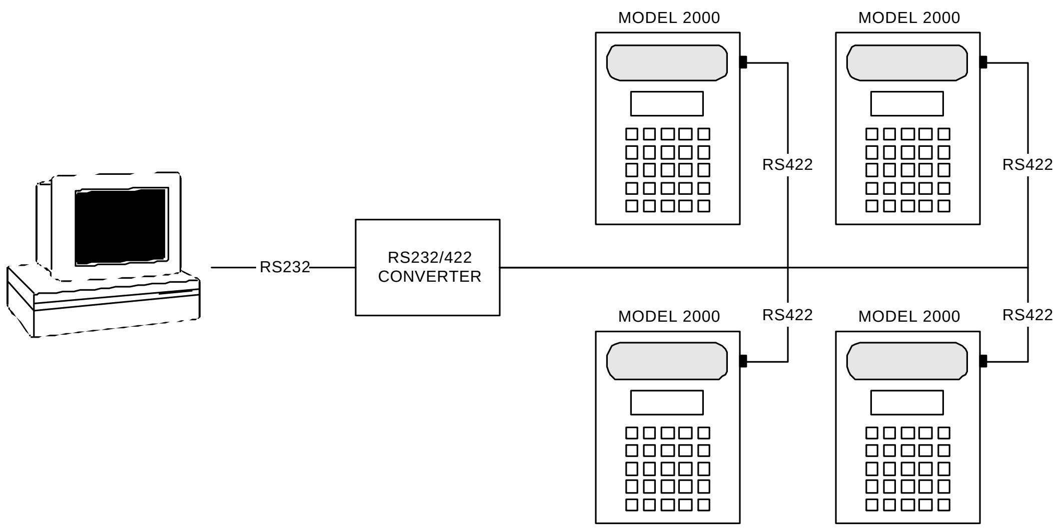
Figure 13. Example Point-To-Point Configuration



Multidrop

The multi-drop configuration is a 'party-line' structure in which several devices share the same communication line. For a hard-wired, multi-drop network, all devices must use RS-422. If converters or modems are used to connect devices to the network, RS-232 or RS-422 may be used. The next figure illustrates the multi-drop configuration:

Figure 14. Example Multidrop System Configuration



In the multi-drop configuration, for RTU operation, a host device capable of emulating RTU protocol is the master and one or more Model 2000s using RTU mode are slaves. Idle slave devices continuously monitor the communication link to determine if the line is busy or idle.

Since there is typically more than one slave device sharing the multi-drop line, each slave only recognizes inquiry sequences containing its own RTU address number. For the RTU protocol, the slaves look for a new request. Since there is typically more than one slave device sharing the multi-drop line, each slave processes only the requests that contain its own station address, or the broadcast address that is sent to all slaves.

3.0 RTU Protocol

3.0.1 Introduction

RTU protocol is a query/response protocol used for communication between the RTU device and a host computer which is capable of communicating using RTU protocol. The host computer is the master device and it transmits a query to a RTU slave which responds to the master. The RTU slave device cannot query – it can only respond to the master.

The RTU data transferred consists of 8-bit binary characters with an optional parity bit. No control characters are added to the data block; however, an error check (cyclic redundancy check) included as the final field of each query and response to ensure accurate transmission of data.

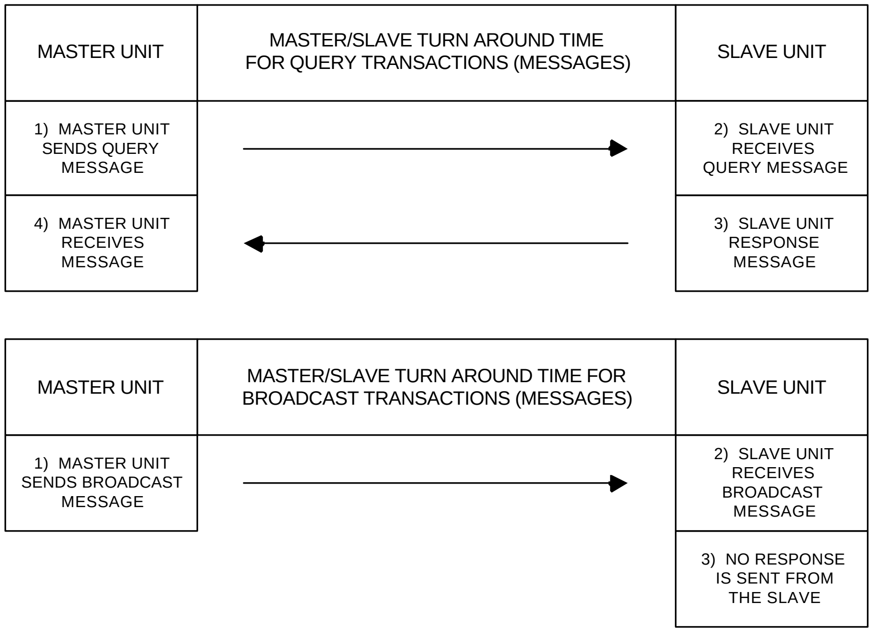
3.0.2 Message Format

A distinction is made between two communicating devices. The device which initiates a data transfer is called the master and the other device is called the slave. The Model 2000 can only be a RTU slave. The master device begins a data transfer by sending a query or broadcast request message. A slave completes that data transfer by sending a response message if the master sent a query message addressed to it. No response message is sent when the master sends a broadcast request. The time between the end of a query and the beginning of the response to that query is called the slave turn-around time.

The turn-around time illustrated below varies based on the query and the activity on the other port. A value of 500 ms can be used as a reasonable worst-case estimate.

The general formats for RTU message transfers are shown below:

Figure 15. RTU Message Transfers



3.0.3 Message Types

The RTU protocol has four message types: query, normal response, error response, and broadcast:

- **Query**

  The master sends a message addressed to a single slave.

- **Normal Response**

  After the slave performs the function requested by the query, it sends back a normal response for that function. This indicates that the request was successful.

- **Error Response**

  The slave receives the query, but for some reason it cannot perform the requested function. The slave sends back an error response which indicates the reason the request could not be processed. (No error messages are sent for certain types of errors. For more information see ***Section 3.0.10***, *Communication Errors*).

- **Broadcast**

  The master sends a message addressed to all slaves by using address 0. All slaves that receive the broadcast message perform the requested function. This transaction is ended by a time-out within the master.

3.0.4 Message Fields

The message fields for a typical message are shown below:

Figure 16. RTU Message Transfers

← MESSAGE FRAME →			
STATION ADDRESS	FUNCTION CODE	INFORMATION	ERROR CHECK

Station Address

The station address is the address of the slave station selected for this data transfer. It is one byte in length and has a value from 0 to 247 inclusive. An address of 0 selects all slave stations, and indicates that this is a broadcast message. An address from 1 to 247 selects a slave station with that station address.

Function Code

The function code identifies the command being issued to the station. It is one byte in length and is defined for the values 0 to 256 as shown in the figure below:

Figure 17. Function Code Definitions

FUNCTION CODE	DESCRIPTION
0 1 2	ILLEGAL FUNCTION READ OUTPUT TABLE UNSUPPORTED FUNCTION
3 4 5	READ REGISTERS UNSUPPORTED FUNCTION FORCE SINGLE OUTPUT
6 7-14 15	PRESET SINGLE REGISTER UNSUPPORTED FUNCTIONS FORCE MULTIPLE OUTPUTS
16 17-127 128-255	PRESET MULTIPLE REGISTERS UNSUPPORTED FUNCTIONS RESERVED FOR EXCEPTION RESPONSES

Information Field

The information field contains additional information required to specify or respond to a requested function. Detailed specifications for the contents of the information field (for each message type – broadcast, query, normal response, and error response) and each function code are found in ***Section 3.0.9,*** *Message Descriptions*. The information field includes entries for the range of data to be accessed in the RTU Slave.

Register and Discrete Output Definition

The figure below represents the Model 2000 slave memory types, unit lengths, and valid ranges:

NOTE: The maximum addressable ranges for each memory type depend upon the CPU model and memory configuration.

Figure 18. Partial List of Slave Memory Types, Unit Lengths, and Valid Ranges

MODEL 2000 MEMORY TYPE	UNIT LENGTH	VALID RANGE
REGISTERS (%R)	WORD	0 - 79
DISCRETE OUTPUTS (%Q)	BIT	0 - 33

NOTE: For a complete list of register and discrete output definitions, refer to Appendix C, in **Section 5.**

Error Check Field

The error check field is two bytes in length and contains a cyclic redundancy check (CRC-16) code. Its value is a function of the contents of the station address, function code, and information field. The details of generating the CRC-16 code are found in ***Section 3.0.8,*** *Cyclic Redundancy Check (CRC)*. Note that the information field is variable in length. In order to properly generate the CRC-16 code, the length of frame must be determined. See ***Section 3.0.8***, *Cyclic Redundancy Check (CRC)***,** subsection *Calculating the Length of Frame*, to calculate the length of a frame for each of the defined function codes.

3.0.5 Character Format

Each byte in a message is transmitted as a character. A message is sent as a series of characters. The illustration below shows the character format. A character consists of a start bit (0), eight data bits, an optional parity bit, and one stop bit (1). Between characters the line is held in the 1 state.

Figure 19. Character Format

		MSB DATA BITS LSB	
10	9	8 7 6 5 4 3 2 1	0
STOP	PARITY (OPTIONAL)		START

3.0.6 Message Termination

Each station monitors the time between characters. When a period of three character times elapses without the reception of a character, the end of a message is assumed.

The reception of the next character is assumed to be the beginning of a new message. The end of a frame occurs when the first of the following two events occurs:

- The number of characters received for the frame is equal to the calculated length of the frame.
- A length of 3 character times elapses without the reception of a character.

3.0.7 Timeout Usage

Time-outs are used on the serial link for error detection, error recovery, and to prevent the missing of the end of messages and message sequences. Note that although the module allows up to three character transmission times between each character in a message received, there is no more than half a character time between each character in a message that the module transmits.

After sending a query message, the master should wait approximately 500 milliseconds before assuming that the slave did not respond to its request.

3.0.8 Cyclic Redundancy Check (CRC)

The cyclic redundancy check (CRC) is one of the most effective systems for checking errors. The CRC consists of 2 check characters generated at the transmitter and added at the end of the transmitted data characters. Using the same method, the receiver generates its own CRC for the incoming data and compares it to the CRC sent by the transmitter to ensure proper transmission.

A complete mathematic derivation for the CRC is not given in this section. This information can be found in a number of texts on data communications. The essential steps to understand when calculating the CRC are:

- The data bits that make up the message are multiplied by the number of bits in the CRC.
- The resulting product is then divided by the generating polynomial (using module 2 with no carries). The CRC is the remainder of this division.
- Disregard the quotient and add the remainder (CRC) to the data bits and transmit the message with CRC.
- The receiver then divides the message plus CRC by the generating the polynomial and if the remainder is 0, the transmission was transmitted without any errors.

A generating polynomial is expressed algebraically as a string of terms in powers of X – such as [$X^3 + X^2 + X^0$ (or 1)] – which can in turn be expressed as the binary number 1101. A generating polynomial could be any length and contain any pattern of 1s and 0s as long as both the transmitter and receiver use the same value.

For optimum error detection, however, certain standard generating polynomials have been developed. RTU protocol uses the polynomial [$X^{16} + X^{15} + X^2 + 1$] which in binary is:

1 1000 0000 0000 0101

The CRC this polynomial generates is known as CRC-16.

The discussion above can be implemented in hardware or software. One hardware implementation involves constructing a multi-section shift register based on the generating polynomial.

Figure 20. Cyclic Redundancy Check (CRC) Register

		MSB DATA BITS LSB	
10	9	8 7 6 5 4 3 2 1	0
STOP	PARITY (OPTIONAL)		START

To generate the CRC, the message data bits are fed to the shift register one at a time. The CRC register contains a preset value. As each data bit is presented to the shift register, the bits are shifted to the right. The LSB is XORed with the data bit and the result is:

- XORed with the old contents of bit 1 (the result placed in bit 0)
- XORed with the old contents of bit 14 (and the result placed in bit 13)
- and finally, the LSB is shifted into bit 15.

This process is repeated until all data bits in a message have been processed. Software implementation of the CRC-16 is explained in the next section.

Calculating the CRC-16

The pseudo code for calculation of the CRC-16 is given below.

Figure 21. CRC-16 Pseudo Code

```
            Preset byte count for data to be sent.
            Initialize the 16-bit remainder (CRC) register to all ones.
            XOR the first 8-bit data byte with the high order byte of the 16-bit CRC register. The result
            is the current CRC.
INIT SHIFT: Initialize the shift counter to 0.
SHIFT:      Shift the current CRC register 1 bit to the right.
            Increment shift count.
            Is the bit shifted out to the right (flag) a 1 or a 0?
               If it is a 1, XOR the generating polynomial with the current CRC.
               If it is a 0, continue.
            Is shift counter equal to 8?
               If NO, return to SHIFT.
               If YES, increment byte count.
            Is byte count greater than the data length?
               If NO, XOR the next 8-bit data byte with the current CRC and go to INIT SHIFT.
               If YES, add current CRC to end of data message for transmission and exit.
```

When the message is transmitted, the receiver performs the same CRC operation on all the data bits and the transmitted CRC. If the information is received correctly the resulting remainder (receiver CRC) is 0.

Example CRC-16 Calculation

The RTU device transmits the right-most byte (of registers or discrete data) first. The first bit of the CRC-16 transmitted is the MSB. Therefore, in the example, the MSB of the CRC polynomial is to the extreme right. The x^{16} term is dropped because it affects only the quotient (which is discarded) and not the remainder (the CRC characters). The generating polynomial is therefore 1010 0000 0000 0001. The remainder is therefore initialized to all 1s.

The example below calculates the CRC-16 for RTU Message, Read Exception Status (07). The message format is shown in the figure below.

Figure 22. CRC-16 RTU Message: Read Exception Status (07)

ADDRESS	FUNCTION	CRC-16	
01	07	?	?

In this example we are querying device number 1 (address 01). We need to know the amount of data to be transmitted and this information can be found for every message type in ***Section 3.0.8***, *Cyclic Redundancy Check (CRC)*, subsection *Calculating the Length of Frame* (this is found on the next page). For this message the data length is 2 bytes.

Figure 23. CRC-16 Complete Solution

TRANSMITTER CRC-16 ALGORITHM	MSB[1]			LASB[2]	FLAG
INITIAL REMAINDER	1111	1111	1111	1111	
XOR 1ST DATA BYTE	0000	0000	0000	0000	
CURRENT CRC	1111	1111	1111	1110	
SHIFT 1	0111	1111	1111	1111	0
SHIFT 2	0011	1111	1111	1111	1
XOR GEN. POLYNOMIAL	1010	0000	0000	0001	
CURRENT CRC	1001	1111	1111	1110	
SHIFT 3	0100	1111	1111	1111	0
SHIFT 4	0010	0111	1111	1111	1
XOR GEN. POLYNOMIAL	1010	0000	0000	0001	
CURRENT CRC	1000	0111	1111	1110	
SHIFT 5	0100	0011	1111	1111	0
SHIFT 6	0010	0001	1111	1111	1
XOR GEN. POLYNOMIAL	1010	0000	0000	0001	
SHIFT 7	0100	0000	1111	1111	0
SHIFT 8	0010	0000	0111	1111	1
XOR GEN. POLYNOMIAL	1010	0000	0000	0001	
CURRENT CRC	1000	0000	0111	1110	
XOR 2ND DATA BYTE	0000	0000	0000	0111	
CURRENT CRC	1000	0000	0111	1001	
SHIFT 1	0100	0000	0011	1100	1
XOR GEN. POLYNOMIAL	1010	0000	0000	0001	
CURRENT CRC	1110	0000	0011	1101	
SHIFT 2	0111	0000	0001	1110	1
XOR GEN. POLYNOMIAL	1010	0000	0000	0001	
CURRENT CRC	1101	0000	0001	1111	
SHIFT 3	0110	1000	0000	1111	1
XOR GEN. POLYNOMIAL	1010	0000	0000	0001	
CURRENT CRC	1100	1000	0000	1110	
SHIFT 4	0110	0100	0000	0111	0
SHIFT 5	0011	0010	0000	0011	1
XOR GEN. POLYNOMIAL	1010	0000	0000	0001	
CURRENT CRC	1001	0010	0000	0010	
SHIFT 6	0100	1001	0000	0001	0
SHIFT 7	0010	0100	1000	0000	1
XOR GEN. POLYNOMIAL	1010	0000	0000	0001	
CURRENT CRC	1000	0100	1000	0001	
SHIFT 8	0100	0010	0100	0000	1
XOR GEN. POLYNOMIAL	1010	0000	0000	0001	
TRANSMITTED CRC	1110	0010	0100	0001	
	E	2	4	1	

RECEIVER CRC-16 ALGORITHM	MSB[1]			LASB[2]	FLAG
RCVR CRC AFTER DATA	1110	0010	0100	0001	
XOR 1ST BYTE TRANS CRC	0000	0000	0100	0001	
CURRENT CRC	1110	0010	0000	0000	
SHIFT 1	0111	0001	0000	0000	0
SHIFT 2	0011	1000	1000	0000	0
SHIFT 3	0001	1100	0100	0000	0
SHIFT 4	0000	1110	0010	0000	0
SHIFT 5	0000	0111	0001	0000	0
SHIFT 6	0000	0011	1000	1000	0
SHIFT 7	0000	0001	1100	0100	0
SHIFT 8	0000	0000	1110	0010	0
XOR 2ND DATA BYTE	0000	0000	1110	0010	
CURRENT CRC	0000	0000	0000	0000	
SHIFT 1-8 YIELDS	0000	0000	0000	0000	

ALL ZEROES FOR THE RECEIVER'S FINAL CRC-16 INDICATE A CORRECT TRANSMISSION.

[1] AS STATED BEFORE, THE RECEIVER PROCESSES INCOMING DATA THROUGH THE SAME CRC ALGORITHM AS THE TRANSMITTER. THE EXAMPLE FOR THE RECEIVER STARTS AT THE POINT AFTER ALL THE DATA BITS, BUT NOT ALL OF THE TRANSMITTED CRCs HAVE BEEN RECEIVED CORRECTLY. THEREFORE, THE RECEIVER'S CRC SHOULD BE EQUAL TO THE TRANSMITTED CRC AT THIS POINT. WHEN THIS OCCURS, THE OUTPUT OF THE CRC ALGORITHM WILL BE ZERO (THIS INDICATES THAT THE TRANSMISSION IS CORRECT).

THE TRANSMITTED MESSAGE WITH THE CRC WOULD BE:

ADDRESS	FUNCTION	CRC-16	
01	07	42	E2

[2] THE MSB AND LSB REFERENCES ARE TO THE DATA BYTES ONLY, AND NOT THE CRC BYTES. THE CRC MSB AND LSB ORDER IS THE REVERSE ORDER OF THE DATA BYTE ORDER.

Calculating the Length of Frame

To generate the CRC-16 for any message, the message length must be known. The length for all types of messages can be determined from the table below:

Figure 24. CRC-16 Message Lengths By Type

FUNCTION CODE	FUNCTION NAME	QUERY OR BROADCAST MESSAGE LENGTH LESS CRC CODE	RESPONSE MESSAGE LENGTH LESS CRC CODE
0		NOT DEFINED	NOT DEFINED
1	READ OUTPUT TABLE	6	3 + 3^{RD} BYTE[1]
2	READ INPUT TABLE	6	3 + 3^{RD} BYTE[1]
3	READ REGISTERS	6	3 + 3^{RD} BYTE[1]
4	READ ANALOG INPUT	6	3 + 3^{RD} BYTE[1]
5	FORCE SINGLE OUTPUT	6	6
6	PRESET SINGLE REGISTER	6	6
7	LOOPBACK/MAINTENANCE	2	3
8		6	6
9-14	FORCE MULTIPLE OUTPUTS	NOT DEFINED	NOT DEFINED
15	PRESET MULTILE REGISTERS	7 + 7^{TH} BYTE[1]	6
16	REPORT DEVICE TYPE	7 + 7^{TH} BYTE[1]	6
17		2	8
18-66	READ SCRATCH PAD	NOT DEFINED	NOT DEFINED
67		6	3 + 3^{RD} BYTE[1]
68-127		NOT DEFINED	NOT DEFINED
128-255		NOT DEFINED	3

[1] The value of this byte is the number of bytes contained in the data being transmitted.

3.0.9 Message Descriptions

The following pages explain the format and fields for several RTU message types.

Message (01): Read Output Table

Figure 25. RTU Message 01: Read Output Table

QUERY

ADDRESS	FUNC 01	STARTING POINT NO.		NUMBER OF POINTS		ERROR CHECK	
		HI	LO	HI	LO		

NORMAL RESPONSE

ADDRESS	FUNC 01	BYTE COUNT	DATA				ERROR CHECK	

Query:

- An address of 0 is not allowed since this cannot be a broadcast request.
- The function code is 01.
- The starting point number is two bytes in length and may be any value less than the highest output point number available in the attached Model 2000. The starting point number is equal to one less than the number of the first output point returned in the normal response to this request.
- The number of points value is two bytes in length. It specifies the number of output points returned in the normal response. The sum of the starting point value and the number of points value must be less than or equal to the highest output point number available in the attached Model 2000. The high order byte of the starting point number and number of bytes fields is sent as the first byte. The low order byte is the second byte in each of these fields.

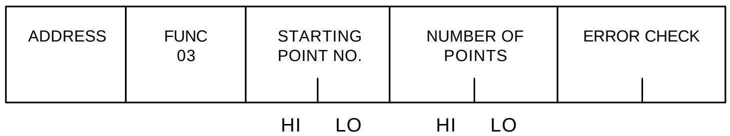
Response:

- The byte count is a binary number from 1 to 256 (0 = 256). It is the number of bytes in the normal response following the byte count and preceding the error check.
- The data field of the normal response is packed output status data. Each byte contains 8 output point values. The least significant bit (LSB) of the first byte contains the value of the output point whose number is equal to the starting point number plus one. The values of the output points are ordered by number starting with the LSB of the first byte of the data field and ending with the most significant bit (MSB) of the last byte of the data field. If the number of points is not a multiple of 8, then the last data byte contains zeros in one to seven of its highest order bits.

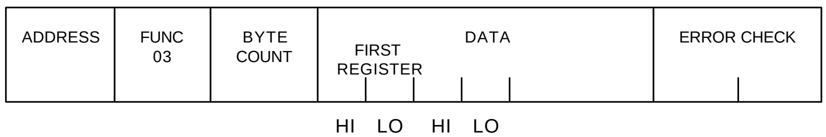
Message (03): Read Registers

Figure 26. RTU Message 03: Read Registers

QUERY



NORMAL RESPONSE



Query:

- An address of 0 is not allowed since this request cannot be a broadcast request.
- The function code is equal to 3.
- The starting register number is two bytes in length. The starting register number may be any value less than the highest register number available in the attached Model 2000. It is equal to one less than the number of the first register returned in the normal response to this request.
- The number of registers value is two bytes in length. It must contain a value from 1 to 125 inclusive. The sum of the starting register value and the number of registers value must be less than or equal to the highest register number available in the attached Model 2000. The high order byte of the starting register number and number of registers fields is sent as the first byte in each of these fields. The low order byte is the second byte in each of these fields.

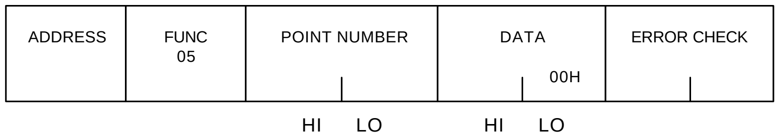
Response:

- The byte count is a binary number from 2 to 250 inclusive. It is the number of bytes in the normal response following the byte count and preceding the error check. Note that the byte count is equal to two times the number of registers returned in the response. A maximum of 250 bytes (125) registers is set so that the entire response can fit into one 256 byte data block.
- The registers are returned in the data field in order of number with the lowest number register in the first two bytes and the highest number register in the last two bytes of the data field. The number of the first register in the data field is equal to the starting register number plus one. The high order byte is sent before the low order byte of each register.

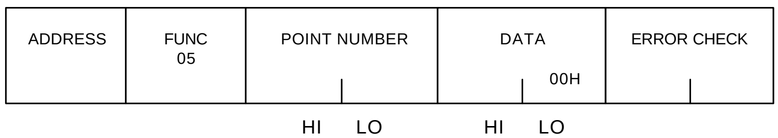
Message (05): Force Single Output

Figure 27. RTU Message 05: Force Single Output

QUERY



NORMAL RESPONSE



Query:

- An address of 0 indicates a broadcast request. All slave stations process a broadcast request and no response is sent.
- The function code is equal to 05.
- The point number field is two bytes in length. It may be any value less than the highest output point number available in the attached Model 2000. It is equal to one less than the number of the output point to be forced on or off.
- The first byte of the data field is equal to either 0 or 255 (FFH). The output point specified in the point number field is to be forced off if the first data field byte is equal to 0. It is to be forced on if the first data field byte is equal to 255 (FFH). The second byte of the data field is always equal to zero.

Response:

- The normal response to a force single output query is identical to the query.

Message (06): Preset Single Register

Figure 28. RTU Message 06: Preset Single Register

QUERY

ADDRESS	FUNC 06	POINT NUMBER		DATA		ERROR CHECK	
		HI	LO	HI	LO		

NORMAL RESPONSE

ADDRESS	FUNC 06	POINT NUMBER		DATA		ERROR CHECK	
		HI	LO	HI	LO		

Query:

- An address 0 indicates a broadcast request. All slave stations process a broadcast request and no response is sent.
- The function code is equal to 06.
- The register number field is two bytes in length. It may be any value less than the highest register available in the attached Model 2000. It is equal to one less than the number of the register to be preset.
- The data field is two bytes in length and contains the value that the register specified by the register number field is to be preset to. The first byte in the data field contains the high order byte of the preset value. The second byte in the data field contains the low order byte.

Response:

- The normal response to a force single output query is identical to the query.

Message (15): Force Multiple Outputs

Figure 29. RTU Message 15: Force Multiple Outputs

QUERY

ADDRESS	FUNC 15	STARTING POINT NUMBER	BYTE COUNT	DATA	ERROR CHECK

NORMAL RESPONSE

ADDRESS	FUNC 15	STARTING POINT NUMBER	NUMBER OF POINTS	ERROR CHECK

Query:

- An address of 0 indicates a broadcast request. All slave stations process a broadcast request and no response is sent.
- The value of the function code is 15.
- The starting point number is two bytes in length and may be any value less than the highest output point number available in the attached Model 2000. The starting point number is equal to one less than the number of the first output point forced by this request.
- The number of points value is two bytes in length. The sum of the starting point number and the number of points value must be less than or equal to the highest output point number available in the attached Model 2000. The high order byte of the starting point number and number of bytes fields is sent as the first byte in each of these fields. The low order byte is the second byte in each of these fields.
- The byte count is a binary number from 1 to 256 (0 = 256). It is the number of bytes in the data field of the force multiple outputs request.
- The data field is packed data containing the values that the outputs specified by the starting point number and the number of points fields are to be forced to. Each byte in the data field contains the values that eight output points are to be forced to. The least significant bit (LSB) of the first byte contains the value that the output point whose number is equal to the starting point number plus one is to be forced to. The values for the output points are ordered by number starting with the LSB of the first byte of the data field and ending with the most significant bit (MSB) of the last byte of the data field. If the number of points is not a multiple of 8, then the last data byte contains zeros in one to seven of its highest order bits.

Response:

- The description of the fields in the response are covered in the query description.

Message (16): Preset Multiple Registers

Figure 30. RTU Message 16: Preset Multiple Registers

QUERY

ADDRESS	FUNC 16	STARTING REGISTER NO.	NUMBER OF REGISTERS	BYTE COUNT	DATA	ERROR CHECK

NORMAL RESPONSE

ADDRESS	FUNC 16	STARTING REGISTER NO.	NUMBER OF REGISTERS	ERROR CHECK

Query:

- An address of 0 indicates a broadcast request. All slave stations process a broadcast request and no response is sent.
- The value of the function code is 16.
- The starting register number is two bytes in length. The starting register number may be any value less than the highest register number available in the attached Model 2000. It is equal to one less than the number of the first register preset by this request.
- The number of registers value is two bytes in length. It must contain a value from 1 to 125 inclusive. The sum of the starting register number and the number of registers value must be less than or equal to the highest register number available in the attached Model 2000. The high order byte of the starting register number and number of registers fields is sent as the first byte in each of these fields. The low order byte is the second byte in each of these fields.
- The byte count field is one byte in length. It is a binary number from 2 to 250 inclusive. It is equal to the number of bytes in the data field of the preset multiple registers request. Note that the byte count is equal to twice the value of the number of registers.
- The registers are returned in the data field in order of number with the lowest number register in the first two bytes and the highest number register in the last two bytes of the data field. The number of the first register in the data field is equal to the starting register number plus one. The high order byte is sent before the low order byte of each register.

Response:

- The description of the fields in the response are covered in the query description.

3.0.10 Communication Errors

Serial link communication errors are divided into three groups: invalid transactions, invalid query messages, and serial link time outs.

Invalid Transactions

If an error occurs during transmission that does not fall into the category of an invalid query message or a serial link time-out, it is known as an invalid transaction. Types of errors causing an invalid transaction include:

- Bad CRC
- The data length specified by the memory address field is longer than the data received
- Framing or overrun errors
- Parity errors.

If an error in this category occurs when a message is received by the Model 2000, the RTU slave does not return an error message. The RTU slave treats the incoming message as though it was not intended.

Invalid Query Messages

When the communications module receives a query addressed to itself, but cannot process the query, it sends one of the following error responses:

	Subcode
Invalid Function Code	1
Invalid Address Field	2
Invalid Data Field	3
Query Processing Failure	4

The format for an error response to a query is as follows:

Figure 31. RTU Invalid Query Message (Error Response)

ADDRESS	EXCEPTION FUNCTION	ERROR SUBCODE	ERROR CHECK

The address reflects the address provided on the original request. The exception function code is equal to the sum of the function code of the query plus 124. The error subcode is equal to 1, 2, 3, or 4. The value of the subcode indicates the reason the query could not be processed.

Invalid Function Code Error Response (1)

An error response with a subcode of 1 is called an invalid function code error response. This response is sent by a slave if it receives a query whose function code is not equal to 1 through 8, 15, 16, 17, or 67.

Invalid Address Error Response (2)

An error response with a subcode of 2 is called an invalid address error response. This error response is sent in the following cases:

- The starting point number and number of points fields specify output points or input points that are not available in the attached Model 2000 (returned for function codes 1, 2, 15).
- The starting register number and number of registers fields specify registers that are not available in the attached Model 2000 (returned for function codes 4, 16).
- The starting analog input number and analog input number fields specify analog inputs that are not available in the attached Model 2000 (returned for function code 3).
- The point number field specifies an output point not available in the attached Model 2000 (returned for function code 5).
- The register number field specifies a register not available in the attached Model 2000 (returned for function code 6).
- The analog input number field specifies an analog input number not available in the attached Model 2000 (returned for function code 3).
- The diagnostic code is not equal to 0, 1, or 4 (returned for function code 8).
- The starting byte number and number of bytes fields specify a scratch pad memory address that is not available in the attached Model 2000 (returned for function code 67).

Invalid Data Value Error Response (3)

An error response with a subcode of 3 is called an invalid data value error response. This response is sent in the following cases:

- The first byte of the data field is not equal to 0 or 255 (FFH) or the second byte of the data field is not equal to 0 for the Force Single Output Request (Function Code 5) or the initiate communication restart request (function code 8, diagnostic code 1). The two bytes of the data field are not both equal to 0 for the Force Listen-Only request (Function Code 8, Diagnostic Code 4).
- This response is also sent when the data length specified by the memory address field is longer than the data received.

Serial Link Timeouts

The only cause for a RTU device to time-out is if an interruption to a data stream of 3 character times occurs while a message is being received. If this occurs the message is considered to have terminated and no response is sent to the master. There are certain timing considerations due to the characteristics of the slave that should be taken into account by the master. After sending a query message, the master should wait approximately 500 milliseconds before assuming that the slave did not respond to its request.

4.0 Building Serial Communications Cables

This section provides the information necessary for you to build serial communications cables from the Model 2000 to another device.

4.0.1 Cable Assembly Specifications

The cable assembly presents one of the most common causes of communication failure. For best performance, construct the cable assemblies according to the recommended connector parts and specifications:

- Cable connector to Model 2000 port, female pluggable type, BEAU Vernitron part #860506, or equivalent.
- The Model 2000 board has a serial port connector labeled J7.
- Maximum length is 4000 ft. (1200 meters) for RS-422 with isolation at the remote end.
- Overall shield.
- 24 AWG (minimum).
- Connector to external device as specified by the external device manufacturer. The following cables provide acceptable operation at data rates up to 15.2 kbps and distances up to 4000 ft. for RS-422:

Figure 32. Cable Specifications

PRODUCT	MODEL TYPE
BELDEN	9505
BELDEN	9184
BELDEN	9302
NEC	222P1SLCBT
BELDEN	8723

When using RS-422, the twisted pairs should be matched so that both transmit signals make up one twisted pair and both receive signals make up the other twisted pair. If this is ignored, crosstalk resulting from the mismatching will affect the performance of the communications system. When routing communication cables outdoors, transient suppression devices can be used to reduce the possibility of damage due to lightning or static discharge.

CAUTION: Care should be exercised to ensure that both the Model 2000 module and the device to which it is connected are grounded to a common point. Failure to do so could result in damage to the equipment.

4.0.2 RS-422 Interface, Cabling Information, and Diagrams

The RS-422 Interface

The Model 2000 is compatible with EIA RS-422 specifications. RS-422 drivers and receivers are utilized to accomplish communications between several system components using multiple driver/ receiver combinations on a single cable with four twisted pairs. The total cable length cannot exceed 4000 feet.

A multi-drop system of a driver and 8 receivers can be configured. The maximum common mode voltage between each additional drop is the RS-422 standard of +12 V to -7 V. The driver output must be capable of ±1.5 V minimum into 60 Ohms. The driver output impedance must be at least 120 kOhms in the high impedance state. The receiver input resistance is 12 kOhms or greater. Receiver sensitivity is ±200 millivolts.

CAUTION: Care must be taken to meet the common mode voltage specifications. Common mode conditions that exceed those specified will result in errors in transmission and/or damage to Model 2000 components.

Constructing RS-422 Cables

When connecting the Model 2000 to a host system (using the RS-422 standard) the host device's line receiver must contain fail safe capability. This means that in an idle, open, or shorted line condition, the output of the line receiver chip must assume the marking state. When using RS-422, the twisted pairs should both be matched so that both transmit signals make up one twisted pair and both receive signals make up the other twisted pair.

The Model 2000 is supplied with a 120 Ohm terminating resistor in each RS-422 receiver circuit. If the module is at either end of an RS-422 multi-drop or point-to-point link, the DIP switch must be configured according to ***Figure 33a*** below. If the module is an intermediate drop in the multi-drop link, this DIP switch must be configured according to ***Figure 33b*** below:

Figure 33a. Model 2000 (RS-422 Multidrop or Point-To-Point Link) DIP Switch Positions

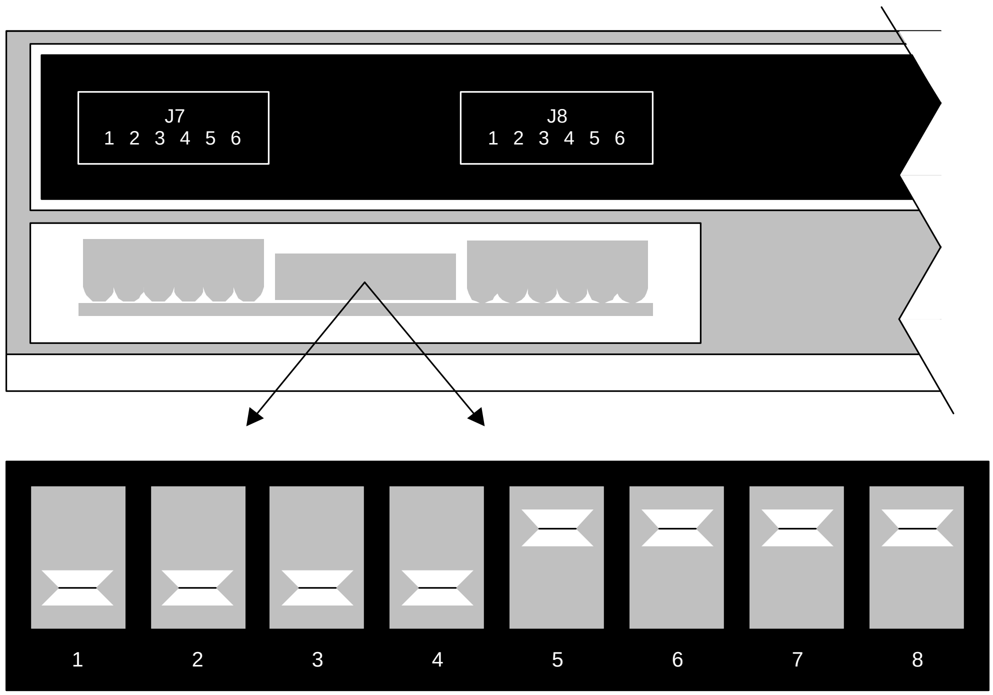
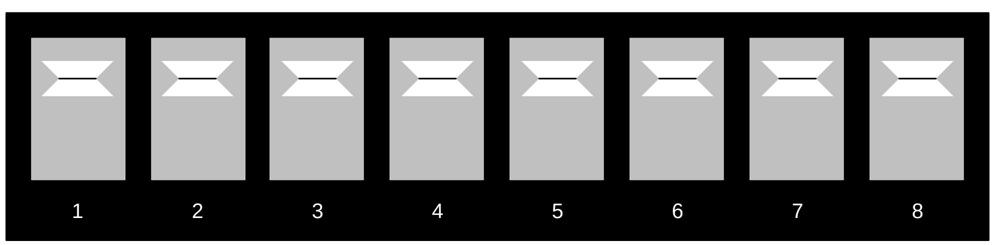


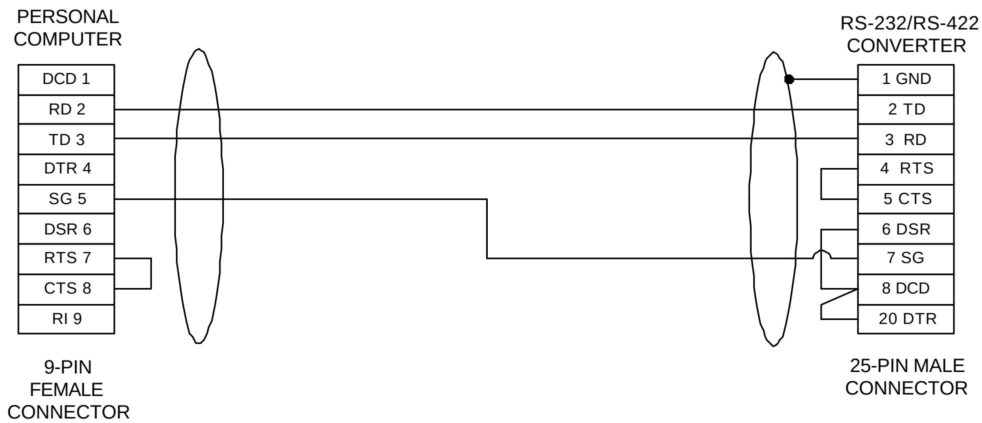
Figure 33b. Model 2000 (Intermediate Drop) DIP Switch Positions



Cable Diagrams

The cable diagrams below are referred to as 'cables' from the system configurations in the previous figures. These diagrams show the principles for constructing cables and can be modified to fit specific applications.

Figure 34. Cable A – RS-232 Host to Converter



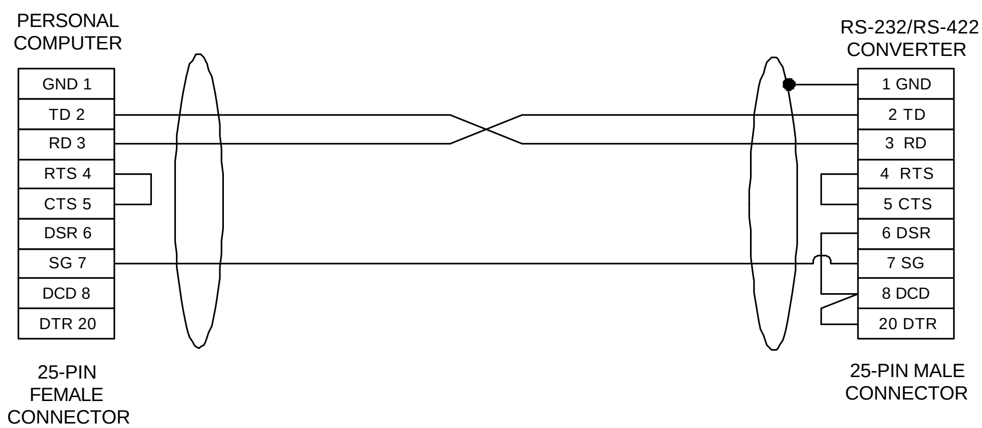


Figure 35.   Cable B – RS-422 Model 2000 to Converter

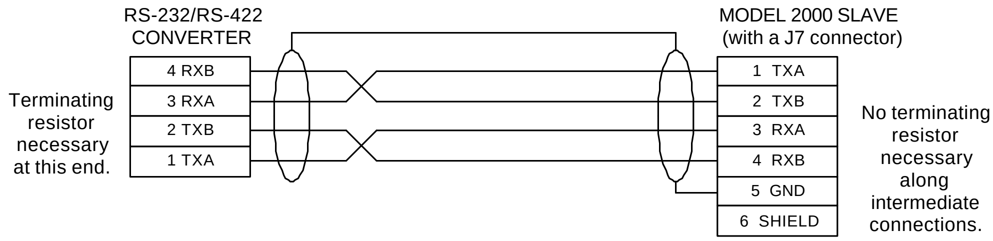
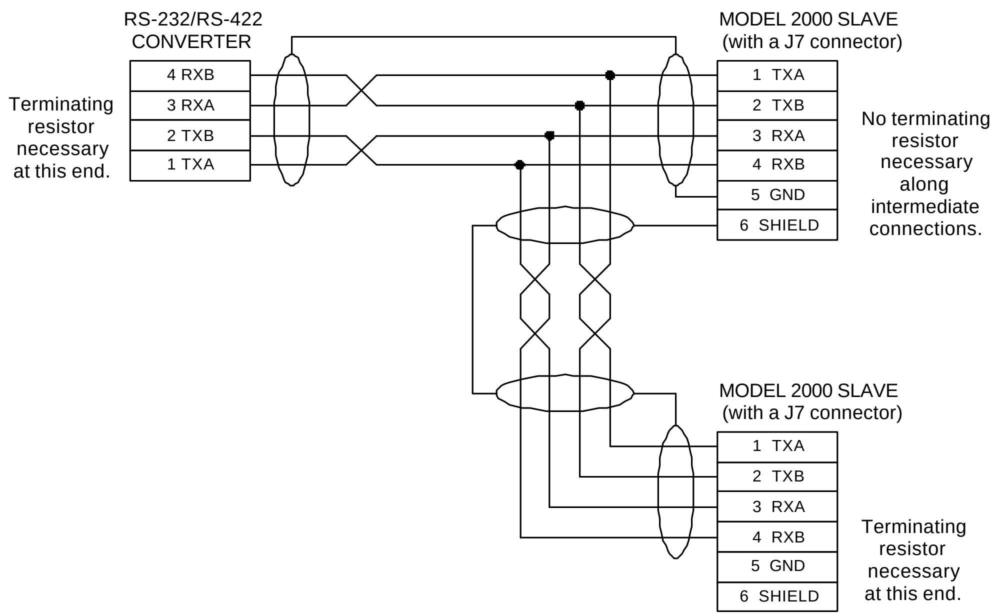


Figure 36.   Cable C – RS422 Twisted Pair (Multidrop Function)



4.0.3 The RS-232/RS-422 Converter (IC107A)

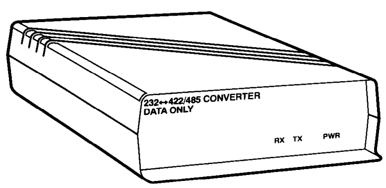
This section describes how to use the RS232/RS422 converter. The section covers the following topics:

- Description of the converter
- User elements
- Converter jumper and switch settings.

Description of the Converter

The isolated repeater (IC107A) is used to convert RS232 protocol to RS422 protocol. This unit can be purchased from Russelectric. Please contact any Russelectric sales office or field service representative.

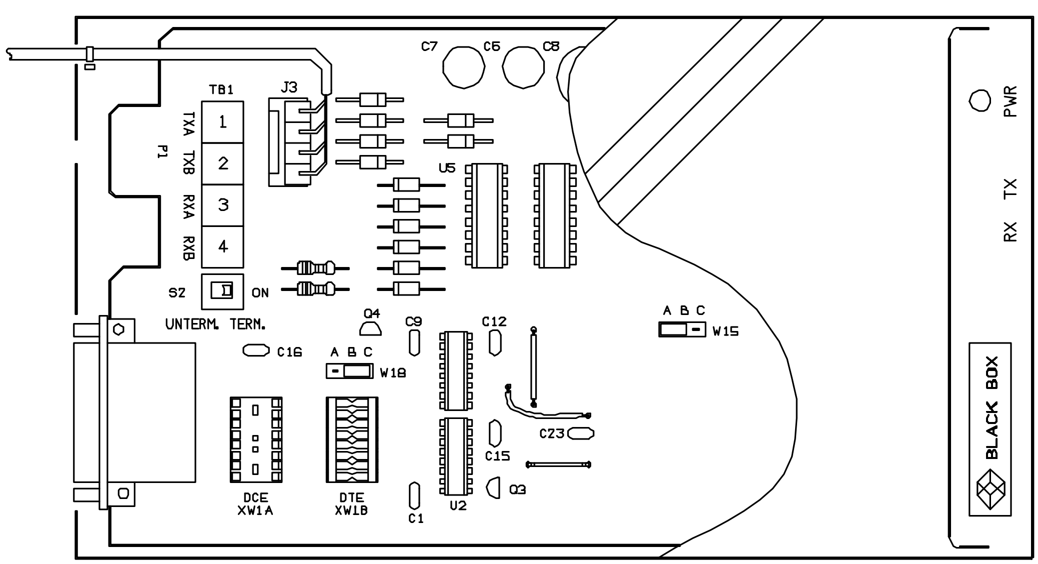
Figure 37. The RS-232/RS-422 Converter



The user elements of the converter are described below:

- One 4-wire terminal blocks and one DB25 female connector
- 115/230 Vac power adapter
- Power ON (green) indicator LED, and transmit and receive LEDs.

Figure 38. The RS-232/RS-422 Converter Jumper and Switch Settings



5.0 Appendices

5.0.1 Appendix A – Glossary of Terms

In serial communications, a number of special terms are used, and are usually referenced by acronyms. For example, a Model 2000, computer or other device that connects to a network is called by the general name station. This appendix contains a concise, alphabetized listing of conventional communications terms and (where applicable) their associated acronyms.

Commonly used Acronyms and Abbreviations

ACK:	Acknowledge control character
ASCII:	American National Standard Code for Information Interchange
BCC:	Block Check Code
BCD:	Binary Coded Decimal
BPS:	Bits per Second
CPU:	Central Processing Unit
CRC:	Cyclic Redundancy Check
CTS:	Clear to Send control signal
DCD:	Data Carrier Detect control signal
DCE:	Data Communications Equipment
DMA:	Direct Memory Access
DOS:	Disk Operating System
DTE:	Data Terminal Equipment
DTR:	Data Terminal Ready control signal
EEPROM:	Electronically Erasable Programmable Read Only Memory
EIA:	Electronics Industries Association
ENQ:	Inquiry control character
EOT:	End of Transmission control character
EPROM:	Erasable Programmable Read Only Memory
ETB:	End of Transmission Block control character
ETX:	End of Text control character
FCC:	Federal Communications Commission
H:	Hexadecimal
I/O:	Input/Output
IEEE:	Institute of Electrical and Electronic Engineers
ISO:	International Standards Organization
K:	1024
KB:	Kilobyte (1024 bytes)
LAN:	Local Area Network
LED:	Light Emitting Diode
LRC:	Longitudinal Redundancy Check
MAP:	Manufacturing Automation Protocol
MB:	Megabyte (1,048,576 bytes)
MODEM:	MOdulator/DEModulator

NAK:	Negative Acknowledge control character
PC:	Personal Computer (IBM-compatible)
PCM:	Programmable Coprocessor Module
PROM:	Programmable Read Only Memory
RAM:	Random Access Memory
RD, RXD:	Receive Data Signal
RTS:	Ready to Send control signal
RTU:	Remote Terminal Unit
SD, TXD:	Transmit Data Signal
SOH:	Start of Header control character
STX:	Start of Text control character
VME:	Versa Module European

Glossary of Terms

Address:	A series of decimal numbers assigned to specific program memory locations and used to access those locations.
Analog:	A numerical expression of physical variables such as rotation, distance, or voltage to represent a quantity.
Application:	An application is the program that executes in a Model 2000, or in a user's program in a computer.
ASCII:	The American Standard Code for Information Interchange (ASCII) is an 8-level code (7 bits plus 1 parity bit) commonly used for the exchange of data.
Asynchronous:	A type of data transmission in which the time intervals between transmitted characters are of unequal length. This mode is controlled by start and stop bits at the beginning and end of each character.
Backplane:	A group of connectors physically mounted at the back of a rack so that printed circuit boards can be mated to them.
Baud:	A unit of data transmission speed equal to the number of code elements per second.
Binary:	This base 2 numbering system only uses the digits 0 and 1.
Bit:	The smallest unit of memory. Can be used to store only one piece of information that has two states (for example, a one/zero, on/off, yes/no, etc.). Data that requires more than two states (e.g. numerical values 0-999) require multiple bits.
Broadband:	A broadband network handles medium-to-large size applications with up to several hundred attached stations. Broadband technology is used in larger networking systems and requires a headend remodulator.
Bus:	An electrical path for transmitting and receiving data.
Byte:	A group of binary digits operated on as a single unit. A byte is made up of 8 bits.
Carrierband:	A carrierband network is designed to handle small to medium-size applications with 6-20 attached stations. This technology is often used in Local Area Networks (LANs).
CPU:	The central processing unit (CPU) is the central device in a computer that interprets user instructions, makes decisions and executes the functions based on a stored program. (This program specifies actions to be taken to all possible inputs.)
Data Link:	The equipment (interface modules, cables, etc.) for the transmission of information.
Ethernet:	A network technology utilizing Carrier Source Multiple Access with Collision Detection (CSMA/CD) often used in local area networks (LANs).
Firmware:	A series of instructions contained in ROM (Read Only Memory) chips that are used for internal processing functions only. These instructions are transparent to the user.
Hardware:	All of the mechanical, electrical and electronic devices that comprise a device.
Hexadecimal:	A base 16 numbering system represented by the digits 0 through 9, followed by letters A through F (A=11, B=12, etc.).

Initiating Station:	The station from which communication originates.
Input:	A signal, typically ***On*** or ***Off***, that provides information to the Model 2000. Inputs are usually generated by devices such as limit switches and push buttons.
I/O (Input/Output):	That portion of the Model 2000 to which field devices are connected.
I/O Scan:	A method by which the CPU monitors all inputs and controls all outputs within a prescribed time.
ISO Terms	1) ISO Standards – The International Standards Organization (ISO) for Open System Interconnection (OSI). 2) ISO Reference Model for Open System Interconnection – An international network architecture standard that defines a seven-layer model. The intent is to provide a design framework to allow equipment from different vendors to communicate.
Isolation:	A method of separating field wiring from logic level circuitry. Typically accomplished through the use of optical isolation devices.
k:	An abbreviation for the metric term kilo (1000) or exactly 1024 in the world of computers. Usually related to 1024 words of memory.
LED:	A light emitting diode (LED) is a solid state device commonly used as a visual indicator in electronic equipment.
LAN:	A local area network (LAN) is a communication network that covers a limited physical space, and having intermediate data transport capability.
Master/Slave:	Communication between stations where one station always initiates requests and the other station always responds to requests.
Memory:	A grouping of physical circuit elements that have data entry, storage and retrieval capability. (See EPROM, EEPROM, Floppy Disk Drive, Hard Disk Drive, RAM, and ROM.)
Microprocessor:	An electronic processor consisting of integrated circuit chips that contain arithmetic, logic, register, control, and memory functions.
Microsecond:	(µs or µsec) is equal to one millionth of a second (1×10^{-6} or 0.000001 second).
Millisecond:	(ms or msec) is equal to one thousandth of a second (1×10^{-3} or 0.001 second).
Mnemonic:	An abbreviation given to an instruction, usually an acronym, formed by combining initial letters or parts of words.
Module:	A replaceable electronic subassembly usually plugged in and secured in place, but easily removable in case of fault or system redesign. In the Model 2000, a combination of a printed circuit boards and associated face plates form a complete assembly.
Multidrop:	A serial wiring configuration that connects more than two devices. all devices on this multidrop must be uniquely addressable.
Nanosecond:	(ns or nsec) is equal to one billionth of a second (1×10^{-9} or 0.000000001 second).
Noise:	Undesirable electrical disturbances, compared to normal signals, that are generally of high frequency content.
Non-Volatile Mem.:	A memory type that retains its stored information under non-powered conditions (power removed or turned off) such as ROM.
Off-Line:	Equipment or devices that are not connected to a communications line.
On-Line:	Equipment or devices connected to a communications line.
Optical Isolation:	Use of a solid state device to isolate the user input and output devices from internal circuitry of the I/O and the CPU.
Output:	Information transferred from the CPU, through a module for level conversion, for controlling an external device or process.
Output Devices:	Physical devices such as motor starters, solenoids, etc. that receive data from the Model 2000.
Outputs:	A signal originating from the Model 2000, typically ***On*** or ***Off***, that controls external devices based upon commands from the CPU.
Parity:	The anticipated state, either odd or even, of a set of binary digits.
Parity Bit:	A bit added to a memory word to make the sum of the bits in a word always even (even parity), or always odd (odd parity).
Parity Check:	A check that determines whether the total number of ones in a word is odd or even.
Parity Error:	Occurs when a computed parity check does not agree with the parity bit.

Peer-to-Peer:	A communication mode where either station can initiate or respond to requests.
Peripherals:	External units that can communicate with a Model 2000. As an example modems, PCs, etc.
Point-to-Point:	A serial wiring configuration which connects only two devices.
Program:	A sequence of functions entered into a controller that are executed by the processor for the purpose of controlling a machine or process.
PROM:	A Programmable Read Only Memory is a retentive digital device programmed at the factory and not easily altered by the user.
Protocol:	A set of rules for exchanging messages between two communicating processes.
RAM:	Random Access Memory is a solid-state device that allows individual bits to be stored and accessed. A battery backup is required because this type of memory is volatile (stored data is lost during no power conditions). The Model 2000 uses a lithium manganese dioxide battery or an optional external back-up battery for this purpose.
Read:	To retrieve data from a storage device.
RTU:	Remote Terminal Unit (RTU) protocol is a query/response mode of operation used for communication between the RTU device and host computer. The host computer transmits the query to the RTU slave that only responds to the master.
RS-232:	A standard specified by the Electronics Industries Association (EIA) for the mechanical and electrical characteristics of the interface for connecting Data Communications Equipment (DCE) and Data Terminal Equipment (DTE).
RS-422:	A recommended standard defining electrical interface characteristics for Data Terminal Equipment (DTE) or Data Circuit-Transmitting Equipment (DCE). The RS-422 standard permits longer ranges and faster transmission rates than the RS-232D standard.
RS-485:	Similar to RS-422. Contains additional protection for receiver circuits. Also, receivers have greater sensitivity to provide the capability for longer distances and more drops.
Serial Comm:	Serial communication data transfer handles the bits sequentially rather than simultaneously (as in parallel transmission).
Significant Bit:	A bit that contributes to the precision of a number. The number of significant bits is counted beginning with the bit contributing the most value, referred to as the Most Significant Bit (MSB), and ending with the bit contributing the least value, referred to as the Least Significant Bit (LSB).
Storage:	Used synonymously with memory. (Typically non-volatile media.)
Synchronous:	A transmission mode in which data bits are transmitted at a fixed rate, with the transmitter and receiver synchronized by a clock (eliminates the need for start and stop bits).
Terminator:	A device or load connected to the output end of a transmission line to terminate or end the signals on that line.
Volatile Memory:	A memory media that loses the stored information if power is removed from the memory circuit devices. (e.g. RAM)
Word:	A measurement of memory length, usually 16 bits long.
Write:	To transfer, record, or copy data from one storage device to another.

5.0.2 Appendix B – ASCII Code List

Characters 0 – 127

Dec.	Char.	Dec.	Char.	Dec.	Char.	Dec.	Char.
0	NUL	32	[space]	64	@	96	`
1	SOH	33	!	65	A	97	a
2	STX	34	"	66	B	98	b
3	ETX	35	#	67	C	99	c
4	EOT	36	$	68	D	100	d
5	ENQ	37	%	69	E	101	e
6	ACK	38	&	70	F	102	f
7	BEL	39	'	71	G	103	g
8	BS	40	(	72	H	104	h
9	TAB	41	)	73	I	105	i
10	LF	42	*	74	J	106	j
11	VT	43	+	75	K	107	k
12	FF	44	,	76	L	108	l
13	CR	45	-	77	M	109	m
14	SO	45	.	78	N	110	n
15	SI	47	/	79	O	111	o
16	DLE	48	0	80	P	112	p
17	DC1	49	1	81	Q	113	q
18	DC2	50	2	82	R	114	r
19	DC3	51	3	83	S	115	s
20	DC4	52	4	84	T	116	t
21	NAK	53	5	85	U	117	u
22	SYN	54	6	86	V	118	v
23	ETB	55	7	87	W	119	w
24	CAN	56	8	88	X	120	x
25	EM	57	9	89	Y	121	y
26	SUB	58	:	90	Z	122	z
27	ESC	59	;	91	[	123	{
28	FS	60	<	92	\	124	|
29	GS	61	=	93	]	125	}
30	RS	62	>	94	^	126	~
31	US	63	?	95	_	127	“

5.0.3 Appendix C – Register and Discrete Output Definitions

RTU Discrete Output Address Map

Output. No.	Access	Description	Accy. No.	On State '1'	Off State "0"
1	RO	Position of the Normal Source contacts	NA	closed	open
2	RO	Position of the Emergency Source contacts	NA	closed	open
3	RO	Status of the Normal Source	NPA	available	not-available
4	RO	Status of the Emergency Source	EPA	available	not-available
5	RO	Status of the load test switch	5a	load test	auto
6	RO	Status of the engine start relay	7	stop engine	start engine
7	RO	Status of the block transfer circuit	BTR	transfer	block
8	RO	Status of the load shed circuit	LSR	load shed	off
9	RO	Status of the area protection	17	on	off
10	RO	Status of the exerciser	12a,b,c	on	off
11	RO	Status of Commit/No-commit Switch	12x	commit	no-commit
12	RO	Status of the Maintain Emergency Position Sw.	5d	maintain	auto
13	RW	Bypass TD2 timer (one-shot)	2a	bypass	off
14	RW	Load Test Switch	5a	test	auto
15	RW	Bypass BTR Switch (one-shot)	BTR	bypass	off
16	RW	Maintain Emergency Position Switch	5d	maintain	auto
17	RW	Commit / No-commit Switch	12x	commit	no-commit
18	RW	Peak Shave Switch	PSR	peak shave	auto
19	RW	Daylight Savings Switch	NA	on	off
20	RW	RTU access level	NA	read/write	read-only
21	RO	Peak Shave status	PSR	on	off
22	RW	Historical Data Record Reset Switch	NA	reset	off
23	RW	Exerciser Override Switch (one-shot)	NA	override	off
24	RW	Exerciser On/Off Switch	12d	off	on
25	RW	Exerciser Load/No-Load Switch	12c	no-load	load
26	RO	Exerciser Load/No-Load Switch status	12c	no-load	load
27	RO	Exerciser On/Off Switch status	12d	off	on
28	RO	Position of the Normal Bypass Contacts	NA	closed	open
29	RO	Position of the Emergency Bypass Contacts	NA	closed	open
30	RO	Position of the Bypass Isolation Contacts	NA	closed	open
31	RW	ATS Transfer Mode	12C	Manual	Auto
32	RW	Transfer to Normal Source (one-shot)	NA	Transfer	reset
33	RW	Transfer to Emergency Source (one-shot)	NA	Transfer	reset
34	RO	Controls Not in Automatic	NA	Not-in-Auto	Auto

RTU Register Address Map (01-50)

Reg. No.	Access	Description	Unit Type	Scaling Factor	Lower Limit	Upper Limit
1	RO	Normal Source Phase A-B Volts	Integer	0.1	00000	32767
2	RO	Normal Source Phase B-C Volts	Integer	0.1	00000	32767
3	RO	Normal Source Phase C-A Volts	Integer	0.1	00000	32767
4	RO	Normal Source Frequency	Integer	0.1	00000	32767
5	RO	Emergency Source Phase A-B Volts	Integer	0.1	00000	32767
6	RO	Emergency Source Phase B-C Volts	Integer	0.1	00000	32767
7	RO	Emergency Source Phase C-A Volts	Integer	0.1	00000	32767
8	RO	Emergency Source Frequency	Integer	0.1	00000	32767
9	RW	TD2 timer preset value	Integer	1.0	00000	09999
10	RO	TD2 timer accumulated value	Integer	1.0	00000	09999
11	RW	TDBT timer preset value	Integer	1.0	00000	09999
12	RO	TDBT timer accumulated value	Integer	1.0	00000	09999
13	RW	TNTD timer preset value	Integer	1.0	00000	09999
14	RO	TNTD timer accumulated value	Integer	1.0	00000	09999
15	RW	TDMI timer preset value	Integer	1.0	00000	09999
16	RO	TDMI timer accumulated value	Integer	1.0	00000	09999
17	RW	TD3 timer preset value	Integer	1.0	00000	09999
18	RO	TD3 timer accumulated value	Integer	1.0	00000	09999
19	RW	TETD timer preset value	Integer	1.0	00000	09999
20	RO	TETD timer accumulated value	Integer	1.0	00000	09999
21	RW	TD1 timer preset value	Integer	1.0	00000	09999
22	RO	TD1 timer accumulated value	Integer	1.0	00000	09999
23	RW	AUT timer preset value	Integer	1.0	00000	09999
24	RO	AUT timer accumulated value	Integer	1.0	00000	09999
25	RW	Exerciser event selector switch	Integer	1.0	00001	00034
26	RO	Exerciser start time	Integer	[1]	[1]	[1]
27	RO	Exerciser stop time	Integer	[1]	[1]	[1]
28	RO	Exerciser day or month day	Integer	[2]	[2]	[2]
29	RO	System time lower word	Integer	[3]	[3]	[3]
30	RO	System time upper word	Integer	[3]	[3]	[3]
31	RO	Record reset time lower word	Integer	[3]	[3]	[3]
32	RO	Record reset time upper word	Integer	[3]	[3]	[3]
33	RO	Historical data event [1] time lower word	Integer	[3]	[3]	[3]
34	RO	Historical data event [1] time upper word	Integer	[3]	[3]	[3]
35	RO	Historical data event [1] information field	Integer	[4]	[4]	[4]
36	RO	Historical data event [2] time lower word	Integer	[3]	[3]	[3]
37	RO	Historical data event [2] time upper word	Integer	[3]	[3]	[3]
38	RO	Historical data event [2] information field	Integer	[4]	[4]	[4]
39	RO	Historical data event [3] time lower word	Integer	[3]	[3]	[3]
40	RO	Historical data event [3] time upper word	Integer	[3]	[3]	[3]
41	RO	Historical data event [3] information field	Integer	[4]	[4]	[4]
42	RO	Historical data event [4] time lower word	Integer	[3]	[3]	[3]
43	RO	Historical data event [4] time upper word	Integer	[3]	[3]	[3]
44	RO	Historical data event [4] information field	Integer	[4]	[4]	[4]
45	RO	Historical data event [5] time lower word	Integer	[3]	[3]	[3]
45	RO	Historical data event [5] time upper word	Integer	[3]	[3]	[3]
47	RO	Historical data event [5] information field	Integer	[4]	[4]	[4]
48	RO	Historical data event [6] time lower word	Integer	[3]	[3]	[3]
49	RO	Historical data event [6] time upper word	Integer	[3]	[3]	[3]
50	RO	Historical data event [6] information field	Integer	[4]	[4]	[4]

RTU Register Address Map (51-79)

Reg. No.	Access	Description	Unit Type	Scaling Factor	Lower Limit	Upper Limit
51	RO	Historical data event [7] time lower word	Integer	[3]	[3]	[3]
52	RO	Historical data event [7] time upper word	Integer	[3]	[3]	[3]
53	RO	Historical data event [7] information field	Integer	[4]	[4]	[4]
54	RO	Historical data event [8] time lower word	Integer	[3]	[3]	[3]
55	RO	Historical data event [8] time upper word	Integer	[3]	[3]	[3]
56	RO	Historical data event [8] information field	Integer	[4]	[4]	[4]
57	RO	Historical data event [9] time lower word	Integer	[3]	[3]	[3]
58	RO	Historical data event [9] time upper word	Integer	[3]	[3]	[3]
59	RO	Historical data event [9] information field	Integer	[4]	[4]	[4]
60	RO	Historical data event [10] time lower word	Integer	[3]	[3]	[3]
61	RO	Historical data event [10] time upper word	Integer	[3]	[3]	[3]
62	RO	Historical data event [10] information field	Integer	[4]	[4]	[4]
63	RO	Normal source operations counter	Integer	1.0	00000	32767
64	RO	Normal source availability time lower word	Integer	[5]	[5]	[5]
65	RO	Normal source availability time upper word	Integer	[5]	[5]	[5]
66	RO	Normal source connection time lower word	Integer	[5]	[5]	[5]
67	RO	Normal source connection time upper word	Integer	[5]	[5]	[5]
68	RO	Normal source operations counter	Integer	1.0	00000	32767
69	RO	Normal source availability time lower word	Integer	[5]	[5]	[5]
70	RO	Normal source availability time upper word	Integer	[5]	[5]	[5]
71	RO	Normal source connection time lower word	Integer	[5]	[5]	[5]
72	RO	Normal source connection time upper word	Integer	[5]	[5]	[5]
73	RW	Set system time lower word	Integer	[3]	[3]	[3]
74	RW	Set system time upper word	Integer	[3]	[3]	[3]
75	RW	Set exerciser event selector	Integer	1.0	00001	00034
76	RW	Set exerciser start time	Integer	[1]	[1]	[1]
77	RW	Set exerciser stop time	Integer	[1]	[1]	[1]
78	RW	Set exerciser day or month/day	Integer	[2]	[2]	[2]
79	RW	Set RTU priority for LSR/BTR	Integer	1.0	00000	00099

Special Notes

[1] Special type used for exerciser start and stop time field. Each event has a unique start and stop time in 24-hour format (hh:mm). The first and second positions are the hours (range: 00 – 24). The third and fourth positions are the minutes (range: 00 – 59).

Example: 2314 = 23:14 = 11:14 PM.

[2] Special type used for exerciser day and month/day field. The exerciser has 34 events. The first 10 events are used for 7-day cycles and can be programmed for the following days: Sunday, Monday, Tuesday, Wednesday, Thursday, Friday, Saturday, Weekdays, Everyday and None. Events 11-34 are calendar mode events and can be programmed for specific days of the month, or set to 00/00. The first and second positions are the day (Events 1- 10 range: 00 – 09, Events 11-34 range: 00 – 31). The third and fourth positions are the month (range: 00 – 12).

Example Events 1 – 10 [0 = Sunday, 1 = Monday, 2 = Tuesday, 3 = Wednesday, 4 = Thursday, 5 = Friday, 6 = Saturday, 7 = Weekdays, 8 = Everyday, and 9 = None.]

Example Events 11 – 34 [10/03 = October 3]

[3] Special type used for real time. Real-time is stored as a long integer. RTU protocol does not support long integers, therefore real time is sent as two integers. These two integers must be reconstructed into a long integer. The long integer is the number seconds that have expired since January 1, 1980 (range: 01-01-1980 00:00:00 to 01-01-2050 00:00:00).

[4] Special type used for historical data information field. The controller logs the most recent ten transfer events. Event number one is the most recent, and event number ten is the oldest. Each event contains a date and time stamp along with an information field.

The information field is a numeric expression that is the sum of the values specifying the type and cause of the transfer. The following table illustrates the values used and the meaning of each group of values:

	Numeric Value	Acronym	Description
Group I			
	1	TTN	Transfer To Normal
	2	TTE	Transfer to Emergency
Group II			
	4	UV	Undervoltage
	8	OV	Overvoltage
	16	UF	Underfrequency
	32	OF	Overfrequency
	64	VD	Voltage Differential
	128	PR	Phase Reversal
Group III			
	256	LTH	Load Test Hardware (via physical load test switch)
	512	EX	Exerciser
	1024	RLT	Remote Load Test
	2048	LTS	Load Test Software (via membrane switch or SCADA system)
	4096	PSR	Peak Shave
	8192	AP	Area Protection
	16384	CTT	Commit to Transfer
	32768	LSR	Load Shed

Table Description

The first group (Group I) of numeric values (1, 2) describes the direction of the transfer. The second group (Group II) of numeric values (4, 8, 16, 32, 64, 128) describes normal or emergency voltage sensing failures. The third group (Group III) of numeric values (256, 512, 1024, 2048, 4096, 8192, 16384, 32768) describes user-invoked transfer functions.

To Interpret the Numeric Value in the *Info* Field

The numeric value shown in the ***Info*** field on the display screen, is the sum of one number from Group I, and one or more numbers from Groups II and III. To interpret the numeric value in the ***Info*** field, perform the following calculation:

1. Find the largest number on the table that can be subtracted from the number in the ***Info*** field.

2. Subtract this number and continue to repeat this process until the ***remainder*** is zero.

Example

If the number in the ***Info*** field is 258, then the largest number that can be subtracted from this number on the table is 256. The remainder is two. Next, subtract the remainder 2 from the closest numeric value on the table, and the remainder is zero.

Using the above table, it is obvious that the number 2 indicates that the direction of the transfer was a 'transfer-to-emergency' and the reason for the transfer, 256, was 'load test hardware'.

[5] Special type used for total time. Total time is stored as a long integer. RTU protocol does not support long integers, therefore total time is sent as two integers. These two integers must be reconstructed back into a long integer. The long integer is the total time in seconds.