

**TOTAL ENGINEERING SERVICES TEAM, INC.**

**RTU / SCADA SYSTEMS**

**T Y P E 2 2 0 0 R T U**

**SCADA NODE**

**SYSTEM REFERENCE**

***Document No. 1210-01***

***Aug 19, 1993***

P.O. Drawer 1760  
671 Whitney Ave.  
Gretna, La. 70054  
(504) 368-6792

Contact Arthur Zatarain, PE  
via [www.artzat.com](http://www.artzat.com)  
For information on this document

*This document is (C) copyright 1993 by  
Total Engineering Services Team, Inc., Gretna, La. USA.  
All rights are reserved.*

## *1 - INTRODUCTION*

## *2 - RTU HARDWARE DESIGN*

- 2.1 Basic Components
- 2.2 RTU Channel Types

## *3 - TYPE 2200 RTU COMPONENTS*

- 3.1 CPU / I/O CONTROLLER
- 3.2 Type 2200 System Support Board
- 3.3 System Setup
- 3.4 Analog to Digital Converter

## *4 - POWER SUPPLIES*

- 4.1 Power Supplies

## *5 - COMMUNICATIONS DEVICES*

- 5.1 Basic RTU Communications
- 5.2 Phone Modems
- 5.3 Radio Modems (Packet Controllers)
- 5.4 Multi-Drop systems
- 5.5 Serial Port Cables

## *6 - TYPE 2200 RTU OPERATION*

- 6.1 Power On Sequence
- 6.2 Communications Power Down Sequence

## *7 - TYPE 2200 RTU PROGRAM*

- 7.1 RTU Program Interface
- 7.2 Command Line Format
- 7.3 Command Processing
- 7.4 Command Line Comments

## *9 - COMMAND SUMMARY*

## **1 - INTRODUCTION**

The Type 2200 SCADA Node RTU is a Remote Telemetry Unit designed for integration into an industrial/oilfield SCADA system. The unit is based on a 80C52 industrial microcontroller. The system uses a Intel 80C52 CPU board and a TEST Inc. designed and manufactured system support board. The RTU handles the usual status and analog inputs, and has limited local processing capability via 8 opto isolated control outputs and two 1-5V or 4-20ma analog outputs. The CPU handles all communications and command processing tasks as a subset of the standard TEST SCADA system. This provides a low cost alternative to the larger TYPE 1200 RTU for locations not requiring local processing and display capability.

Applications include stand-alone remote operation as well as distributed processing applications within a single facility. The unit can operate on a wide variety of communications modes for short on long distance network operation in a SCADA or Distributed Control System application.

The unit normally communicates with TEST's SCADA software in a manner similar to the larger Type 1000 or 1200 RTU, and uses the same communications protocol (TSP) . The commands processed by the Type 2200 RTU form a subset of the commands belonging to the larger TEST Inc. SCADA systems. With the Type 2200, all functions are controlled by the Host computer. The RTU serves to execute the commands from the host, and to interface with the field devices at the remote location. The Type 2200 does not have call-out capability, and all communications sessions will originate from the Host.

A MODBUS version of the Type 2200 RTU allows connection to older SCADA systems using that limited protocol. The unit appears as a standard MODICON PLC with 8 inputs, 8 coils, 8 input registers, and 2 holding registers. An additional 8 holding registers are used to access the built-in timers that control the 8 status outputs.

This SCADA software system allows a host computer to call up an RTU and download information, and for the Host to control outputs at the RTU. The Type 2200 is a very low power design with typical total system requirements of 3 watts. Power requirements are kept to a minimum allowing operation with a small solar power system.

The communications system uses standard ASCII serial ports. Any type of phone, 2 way radio, or direct link can be used as the link between the RTU and HOST computer in point-to-point systems. Multi-drop systems can use any simple modem or direct wire link applicable to that type of system. The transmission speed is limited by the communications system, although 1200 baud is adequate for most installation.

## **2 - RTU HARDWARE DESIGN**

### ***2.1 - BASIC COMPONENTS***

The standard TYPE 2200 RTU is packaged in a single enclosure consisting of the following components:

1. Intel 80C52 CPU with serial and parallel ports, Digital and Analog I/O, and

- 1.6 second watchdog timer.
- 2. TEST Inc. TYPE 2200 System Support board with DC/DC converters, field termination points, setup switches, opto-isolators, serial port connector, reset button.
- 3. Hayes compatible modem, Packet Radio Controller, Multi-Drop modem, or other standard serial communications device. Direct wire via RS-232 is built-into the support board.
- 4. NEMA 4X Fiberglass enclosure, 12" x 20" x 8" (typical)

## 2.2 RTU CHANNEL TYPES

The Type 2200 supports 5 different channel types. These are essentially the same as those on the larger TYPE 1200 RTU, with a few restrictions. The types of channels and the maximum number available are listed below:

<u>CHANNEL TYPE</u>	<u>MAX NO.</u>
STATUS INPUT	8
STATUS OUTPUT	8 (7 plus communications control)
TIMER	8
ANALOG INPUT	8
ANALOG OUTPUT	2

The *STATUS INPUTS* are simple off-on type signals that can be read by the RTU and transmitted to the host as either a 0 or 1. Normally, these are used to detect the status of a piece of equipment (off or on), or the status of an alarm condition (active or inactive).

*STATUS OUTPUTS* are off-on control signals that can be manipulated by the RTU under command from the Host computer system. The outputs on the TYPE 2200 are energized for short periods to perform the desired action. If a constantly energized output is needed (as in a Hurricane Timer), a separate latching relay can be provided. Each output has an associated timer as described below.

*TIMERS* are simple down counters that can be set to any number of seconds. The timers start with the preset count, and begin counting down to zero. The only application of these timers is to control output channels, which will be pulsed on for the number of seconds that the timer is running. When the timer is at 0, the output is off. When the timer is non-zero, the associated output is on. Each timer controls an output with the same number. So, T1 controls O1, T5 controls O5, and so on. However, if an output is turned directly on (without the use of the timer), then the timer will not control the output until it makes a 1 to 0 transition.

*ANALOG INPUTS* are 12 bit precision, single ended values that represent a value between 0 and  $\pm 5$  VDC. The TYPE 2200 software scales up the 12 bits to the signed 16 bit representation used in the SCADA software, so that a SCAN from the TYPE 2200 looks exactly like a scan from a TYPE 1200. The restriction here is that the scan must be done in RAW mode, where the RTU returns the A/D conversion value directly. On the larger RTU, an engineering units option is available that converts the raw number into a scaled, final value. The TYPE 2200 does not have this capability, so the conversion to engineering units must be done

at the Host computer after it receives the raw data from the TYPE 2200.

*ANALOG OUTPUTS* The analog outputs are 0-5VDC and can be converted to any other standard output such as 4-20ma by a simple loop isolator. The value for the analog output is derived from a PID channel on a standard TEST Inc. RTU or Host computer and downloaded to the Type 2200. The Type 2200 does not manipulate the analog output in any way other than sending the downloaded value out of the two available ports. Any PID or other calculation must be done on another system connected to the SCADA Node.

### **3 - TYPE 2200 RTU COMPONENTS**

#### ***3.1 CPU/ I/O CONTROLLER***

This board contains the 80C52-BASIC microprocessor, 32K bytes of EPROM, 8K bytes of RAM, a serial port, and an 8255 PIA chip for parallel data. The board also contains 8 opto-isolated inputs, 8 12-bit analog input channels, and 2 12-bit Analog Outputs. The single board also contains a small isolated DC/DC converters for the analog/digital converter.

Although the CPU card uses an Intel 80C52 processor which contains a built-in BASIC interpreter, the TEST software uses the chip as if it were a plain 8031 or 8051 and runs a complex compiled program from non-volatile memory.

#### ***3.2 TEST Inc. TYPE 2200 SYSTEM SUPPORT BOARD***

The Intel microcontroller board mates with the TEST Inc. Type 2200 System Support Board which provides all I/O terminations and power supply functions. The two boards form a complete RTU that does not need any external devices other than a modem if required.

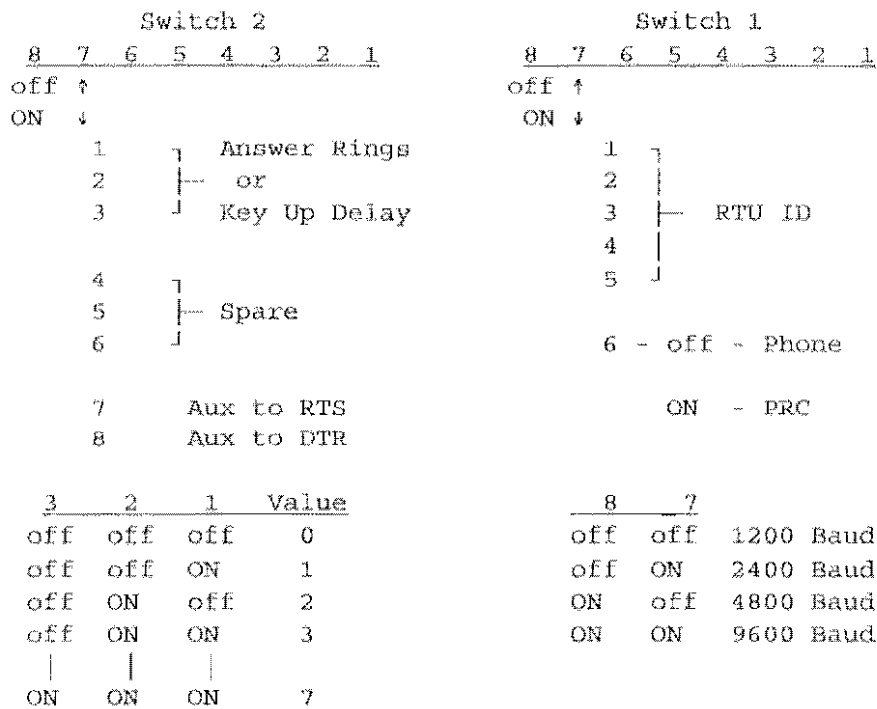
The jumper settings for the CPU board are as follows:

<u>JUMPER</u>	<u>SELECTION</u>	<u>PURPOSE</u>
JP1	IN	Select Compiled Program Operation
JP2	2-3	Watchdog Source on CPU pin T1
JP3	None	Aux reset source
JP4	2-3	Ram size 8K
JP5	1-2	ROM size 32K
JP6	1-2	No battery backup required
JP7	U11 1-2, 2-3, 4-5, 6-7 U10 8-9	ROM in lower 32K RAM in 32-40K
JP8	5-6	A/D Osc for Single Ended A/D
JP9	5-6	0 to +5VDC Analog Input
JP10	5-6	0 to +5VDC Analog Output
JP11	None	Clock interrupt source
JP12	1-2	ULN2803A Output Driver
JP13	1-2	ULN2803A Output Driver
JP14	1-2	ULN2803A Output Driver
JP15	None	
JP16	Built-in	Analog Gnd to Digital Gnd
JP17	IN	Single Ended A/D

### 3.3 SYSTEM SETUP

The System Support board has 2 DIP switches that are labeled SW1 and SW2. These switches are used to setup the specific configuration of an RTU. Setting a switch to "ON" represents a logical "1", while setting it "OFF" represents a logical "0". The following diagram shows the functions of these switches:

#### Type 2200 RTU - SCADA NODE



Rings, Key Up Delay, and RTU ID are set as binary numbers.

Switch 2 Point 6 is not used on Multi-Drop systems.

When setting the DIP switches, each switch can be set to either On or Off. The On and Off settings are interpreted by the RTU computer as a 1 and a 0, respectively. By using binary notation, where a pattern of 0's and 1's are used to represent a unique number, a group of switches can be used to represent a single number other than 0 and 1. For example, the RTU ID is determined by a group of 5 switches. With 5 switches, each of which can represent a 0 or a 1, it is possible to create 32 unique bit patterns of 0's and 1's. Therefore, an RTU can contain an RTU ID in the range 0 - 31.

<u>16</u>	<u>8</u>	<u>4</u>	<u>2</u>	<u>1</u>		<— Digital Value
<u>5</u>	<u>4</u>	<u>3</u>	<u>2</u>	<u>1</u>		<— Bit Locations
0	0	0	0	0	=	0
0	0	0	0	1	=	1
0	0	0	1	0	=	2
0	0	0	1	1	=	3
0	0	1	0	0	=	4
0	0	1	0	1	=	5
1	1	1	1	0	=	30
1	1	1	1	1	=	31

On single node or point-to-point systems, Switch 6 of SW1 is used to configure the RTU for phone or radio communications. Switch SW1-6 is off for phone, on for radio. If using a phone, switches 1-4 of SW2 are used to determine the number of rings that the modem will wait before answering. If using a Packet controller, switches 1-4 of SW2 are used to determine the key-up delay for the PRC. For PRC communications, this setting represents multiples of 100 milliseconds. Therefore, to set the key-up delay to 300 milliseconds, set switches 1 & 2 of SW2 to On and switches 3 & 4 of SW2 to Off.

Switches 1-5 of SW1 must be set to determine the RTU ID number in situations where a radio is being used. The RTU ID is used to set the RTU's call sign. The RTU call sign consists of the prefix "RTU" followed by the ID number. So, if switches 1-5 of SW1 represent the number 3, the RTU's call sign would be "RTU3". This is the call sign used by the host computer to call up the RTU.

Switches 7 and 8 of SW1 are used to set the baud rate of the modem or PRC. The diagram above shows the settings for different baud rates.

### **3.4 ANALOG TO DIGITAL CONVERTER**

This 12-bit analog-to-digital converter board provides the ability to monitor 8 single-ended (0 to  $\pm 5$  volts) analog signals in unipolar operation. When an analog channel is read, the Type 2200 calculates a 12-bit number in the range plus and minus 0-4095. However, the RTU program converts the 12-bit number into a 16-bit number in the range  $\pm 0$ -32760. The 16-bit number is the number returned to the host computer when an analog channel is scanned.

The A/D converter is located on the CPU board. All scaling resistors and field termination points are located on the Type 2200 Support Board. This board also has an optional loop power supply to provide power to field transmitters.

## **4 - POWER SUPPLIES**

### **4.1 POWER SUPPLIES**

A number of regulated DC voltages are required in order for the Type 2200 SCADA system to operate properly. These are all derived internally by various high efficiency power supplies that allow the entire unit to operate from a single DC supply. Normally, the power supply is 12VDC from a battery backup, although the unit can be configured to operate up to 30VDC. All other voltages, including isolated loop and switch voltages, are generated by on-board DC/DC converters.

## **5 - COMMUNICATIONS DEVICES**

### **5.1 BASIC RTU COMMUNICATIONS**

The Type 2200 RTU requires the use of some type of communications system to send data from the RTU to the host computer. The serial port on the CPU card is used for all communications in and out of the RTU. The Type 2200 can use either telephone or radio communications on a "dial up" basis to minimize connect time and its associated costs. A local connection to a small PC and communications program can also be used for local operation.

A modem will normally be used to connect the Type 2200 to a long distance communications line or network. The modem changes the digital computer data into audio tones or digital pulses that can be transmitted much as any other sound is carried on the line. A similar modem at each end of the link converts the tones back into digital data for use by the other computer. This effectively connects the two computers such they can communicate as if they were connected, close to each other, by simple wires.

Several types of intelligent modems are used by this system depending on the type of communications line being used as described below. All types are actually micro-processor based devices that are themselves small computer systems. Their processing capabilities are specifically designed for communications tasks, and they take care of many dialing, timing, call sign, and other low level tasks.

Multi-drop systems uses less intelligent modems that transform RS-232 into some other media such as audio or RS-485. These modems all use a line from the CPU to control the transmitter of the modem. All devices listen to the line but only transmit when it is their turn to send a reply. The actual mechanics of the modem is transparent to the RTU. The unit works the same regardless of the type of multi-drop modem.

### **5.2 PHONE MODEMS**

Dial-up telephone systems require the use of a Hayes compatible phone modem connected to the serial port of the CPU. The modem will be programmed by the RTU using the standard Hayes "AT" command set, and the modem must retain these settings after power is removed. This modem type is used to connect a computer to a phone line. The TEST SCADA systems can use almost any compatible modem, although some have specific advantages (like 12 VDC operation) that promote their use in certain applications.

Telephone systems used by the SCADA system can be almost any type that provides a standard Bell RJ11 (modular phone) jack. The communications rate is limited by the quality and reliability of the channel, although the software is designed to handle poor communications. Generally speaking, the telephone line is the weakest part of most remote SCADA system installations.



### **5.3 RADIO MODEMS (PACKET CONTROLLERS)**

Many TEST RTU systems use PACKET RADIO CONTROLLERS as the means of sending digital data over all radio systems. The PRC is a special modem device that allows connection of the computer to a voice type radio. The unit takes care of message routing and error checking so that many units can share a common radio frequency.

Radio systems can be any type that can provide a 3Khz bandwidth for the modem. FCC limitations may restrict the amount of data transmitted at any one time. Radio systems, once installed properly, generally provide less trouble than phone systems due to the consistency of the equipment and signal levels.

### **5.4 MULTI-DROP SYSTEMS**

A separate program ROM is available to allow multi-drop operation of several Type 2200 RTU's on a single communications line. This is typically either a hard wired link or a simple radio based system. When operating in this mode, the RTU will monitor all transmissions and respond only to those addressed for its own ID. Other messages will be ignored. The only exception is messages addressed to RTU0. Any message broadcast to RTU0 will be executed by all systems. However, none will send a reply.

Refer to TEST Document 1170 for more information on Multi-Drop communications operation for all TEST Inc. SCADA systems.

### **5.5 SERIAL PORT CABLES**

All communications is from the serial port of the RTU. The 9 pin D Male connector on the Type 2200 Support board provides a standard PC compatible serial port connection. The pinout is as follows:

1	DCD (not connected)
2	RD Receive Data into CPU
3	TD Transmitted Data out of CPU
4	DTR Data Terminal Ready out of CPU
5	SG Common Signal Ground
6	DSR Data Set Ready (not connected)
7	RTS Request to Send out of CPU
8	CTS Clear to send (not connected)
9	RI Ring Indicator (not connected)

Pins 4 and 7 are Auxiliary handshake lines used by the CPU to control modems, radios, and multi-drop drivers. The system will keep these pins active when on-line for phone/PRC systems, dropping them only during a modem reset.

In multi-drop systems, the CPU will toggle these lines active when it wants to transmit. The delay between handshake active and transmit can be set by Switch 2 on the Type 2200 Support board. Two switch positions on Switch 2 determine which RS-232 pins are connected to the auxiliary control line. Normal operation will have both RTS and DTR active by closing positions 7 and 8 on Switch 2.

## **6 - TYPE 2200 RTU OPERATION**

### **6.1 POWER ON SEQUENCE**

Every time the Type 2200 System is powered it goes through a brief initialization sequence. This sequence is also executed when the unit is reset, either by a manual keypress or due to a watchdog timer failure. The computer starts execution by reading DIP switches SW1 and SW2 on the Type 2200 Support board. This allows the computer to set the baud rate of the serial port. On a "cold start" from a non-powered condition, the system will clear all timers and output values. If the system is being re-started due to a watchdog timer reset, the status of timers and outputs will be retained and reset quickly. However, there may be a one second blip on the output lines while the system is being reset.

Also determined from the DIP switches is whether a phone or PRC is used for communications. If using a phone, the modem is programmed to answer on the number of rings specified by SW2. If using a PRC or a multi-drop setup, the unit ID is read from the switches.

### **6.2 COMMUNICATIONS POWER DOWN SEQUENCE**

Unlike the older TEST Type 2000 RTU, the Type 2200 SCADA Node does routinely turn itself off and on. However, the system does provide automatic support for communications reset on every BYE and also once per hour of inactivity. Output point 8 is used to control external modems, cellular phones, and other devices that may require periodic power cycling for proper operation. The system will turn point 8 off for two seconds whenever one of the following occurs:

1. Programmed BYE command is executed.
2. One minute has passed since last command was processed.
3. One hour has passed without any commands.

The RTU will stay on line as long as commands are processed at least once per minute. If a minute goes by without a command, the program goes "off-line" and hangs up the modem with the RTS/DTR line as well as a two second toggle on Output 8. Normally, the modem and/or phone will be wired to output 8 so that they will definitely power down and go to a known state at the end of each call.

Once the unit is off-line, it waits for another command. If none comes in one hour, the program repeats the power cycle on Output 8. This prevents a locked up modem or phone from hanging the system indefinitely.

## **7 - TYPE 2200 RTU PROGRAM**

### **7.1 RTU PROGRAM INTERFACE**

There are two modes in which the RTU program can operate when processing commands. In RTU (computer) mode, message numbers and a checksum are transmitted with each command. In terminal (human) mode, only the command to process is transmitted to the RTU. As the RTU receives commands, they are processed immediately and the results are sent back over the

communications device to the unit that initiated the command.

The interface is very similar to that of the standard TEST TYPE 1000 or 1200 RTU, except that only a small subset of the commands are supported. The commands are all related to scanning data channels or controlling the communications link.

## **7.2 COMMAND LINE FORMAT**

Command line formats differ depending on whether the program is in TERMINAL mode or RTU mode. Normally, RTU mode is never used manually as the error checking procedures are awkward for a person to use. The computer, however, handles the problems easily and does the processing necessary in RTU mode. Upon powering up, the Type 2200 RTU program begins in RTU mode. The only way to get out of RTU mode and into TERMINAL mode is to enter the 4 character key sequence [CTRL-A]\*T [ENTER] (T must be upper case). Note that the [CTRL-A] is a single keystroke. Once placed into TERMINAL mode, the program will operate in that mode until powered down.

Multi-drop systems must use a different protocol to access the RTU and have it process the \*TERMINAL command. The proper address must be sent ahead of the \*TERMINAL code in order for the RTU to recognize it as a message for that unit. The syntax is similar to the normal protocol except that the "to and from" header appears after the control-A code. The multi-drop ID of the RTU comes first, followed by a slash and the name of the sending unit, followed by another slash. The sending unit ID is irrelevant when switching to TERMINAL mode because the multi-drop protocol will be suspended anyway. So, any "From ID" can be used. For example, an RTU with the ID of 2 would be accessed as follows:

```
[CTRL-A] RTU2/x/*T [ENTER]
```

The heading RTU2/x/ is processed as a message destined for "RTU2" that is sent by unit "x". Obviously the "x" is a dummy ID that is simply a place holder that will allow proper processing of the \*TERMINAL command.

Once in TERMINAL mode, commands are simply entered as typed text and ended with a carriage return (ENTER key). The backspace key can be used to back over and erase typing mistakes, and the standard DOS cancel (control-C) key can be used to terminate an entire line.

In RTU mode, however, the line format is more complicated to allow for the automatic checking and correction of various communication problems. From the PC based host, all messages sent to the Type 2200 RTU are numbered. Each message must be acknowledged by the Type 2200 RTU before file processing at the host will continue. The Type 2200 examines the sequence number of each incoming message and ignores any messages that have already been processed (although it will return a dummy acknowledgement). This is helpful in systems with poor communications because it prevents out-of-sequence conditions in cases where the delays inherent in Packet Radio or Satellite communications build up over time.

When a message is received from the host, the Type 2200 will either respond with a message or a simple acknowledgement. In either case, the format

of the response is the same. To return a simple acknowledgement, the message returned is "@CCC," followed by a number indicating the result of the message. Included in either response is the number of the message being acknowledged.

All messages sent from the Type 2200 RTU back to the host have the same format as the messages being sent from the host. As mentioned above, each message is accompanied by its outgoing sequence number. Transmitted with each message is also the sequence number of the last incoming message. This is the number which acknowledges a received message.

The last part of every message is a checksum which is used to make sure that the message received by a unit matches the message that was sent. The receiving computer will use the checksum to detect lost or incorrect characters on the line. If no errors are detected the receiver will acknowledge the message. If an error is detected, a negative acknowledgement (ASCII NAK) code is returned to the sender. If the sender receives a NAK, it will send the same line again.

If a host unit does not receive a response (an ACK or a NAK) to a message within a short period of time after a message is sent, it will resend the message. This is not the case with the Type 2200 RTU. Although the Type 2200 does respond to a NAK, it does not require any outgoing messages to be acknowledged. In fact, when the host returns an ACK to the Type 2200, it is simply ignored. Since the host is responsible for initiating a call to the Type 2200 RTU and requesting information from it, it is up to the host to keep asking until it either gets a proper response or gives up. The Type 2200 RTU does nothing to ensure that the host has received what it was sent.

To summarize, when transmitting in RTU mode the computers will automatically format each line with error checking characters. All lines begin with a special START OF HEADER code, SOH, which is the CONTROL-A (also written ^A). In RTU mode, nothing is processed until the SOH is seen. This is why the CONTROL-A must be entered when switching from RTU mode to TERMINAL mode as described above.

Following the SOH character is the sequence number of the outgoing message from the sending unit. A ^F is used as a separator which is followed by the sequence number of the message being acknowledged. The text portion of the command is next which begins with the START OF TEXT (STX) code, ^B. Following the last character of the text message is the line terminator character, ^C, followed by the checksum. The maximum length of this entire formatted line is 250 characters. If the line is too long, the Type 2200 will simply truncate the line and not be able to process it.

### **7.3 COMMAND PROCESSING**

All command lines begin with a keyword indicating the nature of the command. All keywords are descriptive of the command purpose. The command processor is case insensitive, meaning that any combination of upper and lower case characters can be entered. Some commands are just a simple keyword. Other commands take the form of a keyword followed by a number of parameters. The number and form of the parameters is dependent on the nature of the command. The method used to "parse" a command line will recognize both a comma and a space as a separator between parameters. When processing a command, the type 2200 RTU program only checks the first 2 characters of the keyword.

In normal operation, the PC based host computer will contain a link file with all of the commands necessary to perform a download from the Type 2200 RTU. With other TEST Inc. RTU/SCADA systems, the RTU system usually contains a DOWNLOAD.RTU file which gets read when a host system calls it up. The download file contains the commands that get processed to perform the download. The Type 2200, however, does not contain a download file. Therefore, the host computer must have another way to process a file to perform the download. This is where the link file on the host is used.

When a link at the host is activated and a connection is made with an RTU, the host computer will look for and execute a LINKx.RTU file if it is found. The x in the file name is replaced by the link number. This means that there should be a separate LINKx.RTU file on the host computer for each Type 2200 RTU it must call. The link file contains the commands necessary to download information from the RTU.

Normally a SCAN command is used to request a range of data values from an RTU. However, the SCAN command returns values for the unit processing the command. In the situation of the Type 2200, the host computer is processing the commands in the link file. Since we are interested in downloading a range of data values from the remote RTU rather than receiving a range of data values from the host, we must use a BLOCK SCAN command. The BLOCK command is used to transmit the rest of the line to the other unit for processing. Therefore, using a BLOCK SCAN command will cause the Host to send the line to the RTU which will process the SCAN command and return a range of data values.

In the host link file the BLOCK command is not used only for SCAN commands. The BLOCK command is used before any command that is to be processed by the Type 2200 RTU. If the BLOCK command is not used the commands will be processed locally by the host computer. An example of a typical LINKx.RTU file used to download values from a Type 2200 RTU might look like this:

```
BLOCK DATA T1 E 30           ; Set timer chan 1 at RTU for 30 seconds
BLOCK SCAN S1:S16 R#         ; Scan status chans and clear time/date
BLOCK SCAN A1:A8 R           ; Scan analog channels
BLOCK BYE                     ; Force RTU to process BYE command
BYE                            ; Process BYE command at host
```

#### **7.4 COMMAND LINE COMMENTS**

When creating a file on the host PC that will be used to download information from the Type 2200, the semicolon (;) character is used as a comment character. When transmitting a command line, everything on the line following a semicolon is ignored. This allows for non-effective comments to be placed in command files. This is good programming practice that assists in future modification of the various program files because descriptive comments help explain the purpose of program lines.

## **8 - COMMAND SUMMARY**

This section provides a detailed description all commands supported by the Type 2200 RTU/SCADA system. Only the first 2 letters of any command are

significant. Since all of these commands will be generated from the host computer, the keyword BLOCK must precede all of these commands in order to have the Type 2200 RTU process the commands. The BLOCK command is used to transmit the rest of a line from the host unit to the RTU for processing.

### **BYE Terminate the RTU communications session.**

The proper way to terminate an RTU session is with the BYE command. Although the Type 2200 RTU program will time out after 45 seconds of inactivity, using the BYE command will provide an immediate shutdown of the session. When the BYE is processed the RTU will first send an ACK back to the host unit so that the host computer can continue file processing. The RTU will then disconnect from the communications line and a flag will be set to power down. If no timers are currently active the RTU will be powered down. If any timers are active, the system will not be powered down until the last timer has reached 0.

Example:   BLOCK BYE     ; force RTU to disconnect and power down  
          BYE            ; process BYE command at host

### **CAL Calibrate a channel with rapid display.**

When calibrating an analog channel, it is often desirable to have a quick-response display to assist in the adjustments. The CAL command provides a rapid display of a single channel. Normally, analog inputs are used although any channel type can be specified. This command is only used with the RTU is in local TERMINAL mode, and the output is normally viewed with a PC operating a communications program such as PROCOMM.

The parameters on the line include the required channel identifier and an optional wait count. The wait count is used to control how long the program will wait between displays. If none is specified a value of 50 is used which provides a display that is updated about once per second. Wait counts can range from 0 to 655.

Examples:   CAL A1         ; accept default update rate  
          CAL A1 10       ; very fast display

### **DATA Specify data values for a range of channels.**

When an RTU is uploading channel data to a host computer, it sends each line of information in the form of a DATA line. Each line indicates a range of channels followed by the appropriate data points. The SCAN command requests data for a range of channels, and the DATA command represents a response with that range. The values are listed one after another on the line in ascending channel order.

DATA commands are also used in command files at the host to allow channels at the RTU to be set to specific values. This is how Outputs, Timers, and Counters are set at the RTU. A parameter must be given in the DATA command which specifies the format of the setting values. For Timer channels, the parameter must be an "E" indicating that the setting values are in engineering units. For Counter channels, the parameter must be an "R" indicating that the set values are raw units rather than processed engineering units. For Output channels, the parameter can be either an "E" or an "R".

When the DATA command is used to set Timers and Counters a single integer must be specified for each channel in the range. This is also the case when setting Output channels using the "E" option for engineering units. However, when the DATA command is used to set Output channels in raw units, a single integer is used to set 8 channels. Each integer represents the decimal equivalent of 8 bit values which are used to turn 8 output channels on or off.

Multiple channel types can be present in the same DATA line. Each group (except the first) begins with the forward slash (/) character. This multiple grouping allows for faster transfers in half-duplex radio systems where transmission overhead is a significant portion of the download time. Combining channel types cuts down the number of transmissions required to complete a download.

An RTU name can be specified in "dot" notation for the first channel range in a DATA command. This RTU name will simply be ignored by the Type 2200. However, specifying an RTU name may be helpful when using the SCAN command. See the explanation of the SCAN command for more details.

Examples: DATA T1 E 30 ; Set timer 1 for 30 seconds  
DATA 01:08 E 1 1 1 1 0 0 0 0 ; Turn on outputs 1-4  
DATA 01:08 R 15 ; Turn on outputs 1-4  
DATA RTU1.01:08 R 15 ; Same as above, RTU name is ignored

### **RESYNC Reset incoming and outgoing block counters.**

When the program is operation in RTU mode (computer to computer operations), the command lines are automatically formatted to allow for communication error checking. As part of the error checking system, each message sent to another unit is numbered. Each message received from another unit is also numbered. If the sequence number of an incoming message has already been processed then the duplicated message will be ignored. This type of error checking is useful because it prevents messages from getting out-of-sequence due to poor communications.

In situations of continuous communication the counters for these incoming and outgoing messages may exceed their limits. If this happens, the counters will roll over and start counting again from 0. This will cause the next message that is transmitted to have a sequence number less than the previous message. Subsequently, the message will be ignored. To keep this undesirable situation from occurring the RESYNC command can be used.

The RESYNC command is used to reset the incoming and outgoing block counters. In order for the counters on both units to be reset and stay synchronized the RESYNC command must be used in two steps. First process a BLOCK RESYNC command to reset the message counters at the RTU. Then process a RESYNC command that will reset the message counters on the host unit.

Example: BLOCK RESYNC ; reset block counters at RTU  
RESYNC ; reset block counters at host

NOTE: Most TEST SCADA System Software installations support auto-resync operation that eliminates the need to send an explicit RESYNC command to the Type 2200 RTU. The RTU will ignore old messages unless the message is number

1 which indicates the start of a new command sequence. Therefore, the host system can execute the RESYNC locally to restart its own message counter and continue sending commands to the RTU. The RTU will sense the resync because the next message will be number 1. This capability eliminates the need to send a RESYNC and also execute one locally at the host.

### **SCAN Request a range of data values from the RTU.**

The SCAN command is used to request a range of channel values from an RTU. The result of the request is a DATA command returned by the RTU. The resulting DATA command returns the same range of channels along with their current values.

Along with the requested channel range, the SCAN command uses one of two possible code letters to indicate the format and content of the data to be scanned. These code letters go on the SCAN command line just after the channel range specification and are returned in the resulting DATA line that the SCAN command generates. The two code letters are "E" and "R". The following table shows which options are available for which channel types. An "x" indicates that the option for that column is available for the channel type of that row.

CHANNEL TYPE	CODES	
	E	R
Status	x	x
Outputs	x	x
Timers	x	
Analogs		x
Analog Out		x

The code letter "R" is used to request raw (i.e. direct I/O) data and the code letter "E" is used to request processed (engineering unit) data from the RTU. A SCAN of Analog and Counter channels will generate raw DATA lines that contain an number for each channel scanned. A SCAN of Timer channels will generate the same type of line except that the values returned will be in engineering units rather than raw. The scanning of Status or Output channels works differently depending on whether the SCAN is done for engineering units or raw data.

Scanning Status or Output channels for engineering units will generate a typical DATA line that contains an integer for each channel scanned. However, scanning Status or Output channels for raw unit data will generate a DATA line where each number represents the on-off status of 8 channels. This packed format allows each bit of a number to represent the status of a single channel. For Status and Output channels, if a code letter is not specified in the SCAN command or if the code letter is invalid, the data returned will be in raw format.

For example, the command `SCAN S1:S32 R` will cause the system to respond with `DATA S1:S32 R A B C D` where A B C and D will be the decimal equivalents of 8 bit values. Status channels 1-8 will be set from bits 1-8 of A, channels 9-16 will be set from bits 1-8 of B, and so on. Using the Raw mode allows for faster data



transfers than the engineering mode.

Although there are only two codes available that will be recognized by the Type 2200 RTU, other codes can be included in the SCAN command along with these codes. If other codes are included, they cannot precede the required valid code. They must immediately follow the required code letter and not be separated by any blanks or commas. All codes will be received by the RTU in the form of the SCAN command and be returned to the host computer in the resulting DATA command. The @ and # characters are two codes that you might want to consider doing this with.

Although @ and # are ignored by the Type 2200 RTU, they have significant effects on the host system that processes the resulting DATA line. Including the # in the first SCAN line of the download file will cause the system processing the DATA line (host system) to clear out its update time and date. This allows transfers that are terminated abnormally to be indicated by a blank time and date for that RTU. Including the @ in the last SCAN line of the download file will cause the host system to update its time and date for that RTU.

Data for multiple channel types can be transferred with a single SCAN command. To set up multiple requests, simply enter the additional ones on the line after the first one. The SCAN command is not repeated, but each channel range and unit type code letter must be present for each group. Also, each group (except the first) begins with the forward slash (/) character. This multiple grouping allows for faster transfers in half-duplex radio systems where transmission overhead is a significant portion of the download time. Combining channel types cuts down the number of transmissions required to complete a download.

For example, the following could be used on a small RTU to transfer all points in one command:

```
SCAN S1:S8 R O1:O8 R A1:A6 R
```

Note that this SCAN command requests data for 3 types of channels in a single line. The response might look like this:

```
DATA S1:S8,R,12,/O1:O8,R,2,/A1:A6,R,12.3,4.33,245.0,0,12.45,22.67
```

Multiple scans must be set up so that the resulting line will not exceed 200 characters. This is the maximum line size allowed in all transmissions. In noisy radio environments, shorter lines may produce better overall results due to the difficulty in sending long transmissions.

An RTU name can be specified in "dot" notation for the first channel range in a SCAN command. This RTU name will simply be ignored by the Type 2200 but included in the resulting DATA line. This can be useful in situations where the host computer must poll more than one RTU. By including the RTU name in the resulting DATA line, the host computer will know which RTU to assign the new channel values to. Otherwise, the host computer will assign the incoming channel values to the currently selected RTU. Chances are the RTU to get updated will not be the correct RTU.

```
SCAN S1:S8 R  
SCAN S1:S8 E O1:O7 R A1:A6 R
```

**\*TERMINAL Put program in terminal (human interface) mode.**

This is a very special command because it is immune from the normal error checking requirements that are in effect when the program is in RTU mode. This command is to be used manually rather than being processed from within a command file. After the command is entered, beginning with the start character ^A (control-A), the program will drop into terminal mode. This allows entry of commands without the checksum requirements normally associated with RTU data transfers.

Example:    ^A\*TERMINAL

**VER Get RTU program version information.**

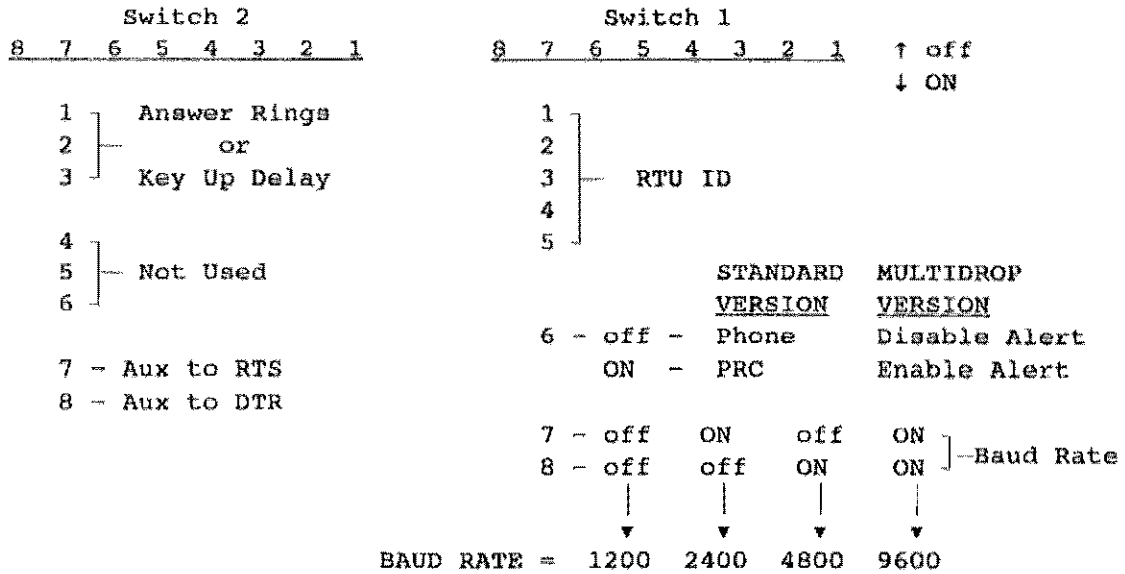
This will display the program version date as well as the current settings of SW1 and SW2 on the TEST Inc. 2200 Control board.

————— End of TYPE 2200 Document —————

*AMZ/nt RTU2200.DOC*

**TOTAL ENGINEERING SERVICES TEAM, INC**  
**(TEST INC)**  
**New Orleans, La. USA (504)368-6792**

**Type 2200 RTU - SCADA NODE**

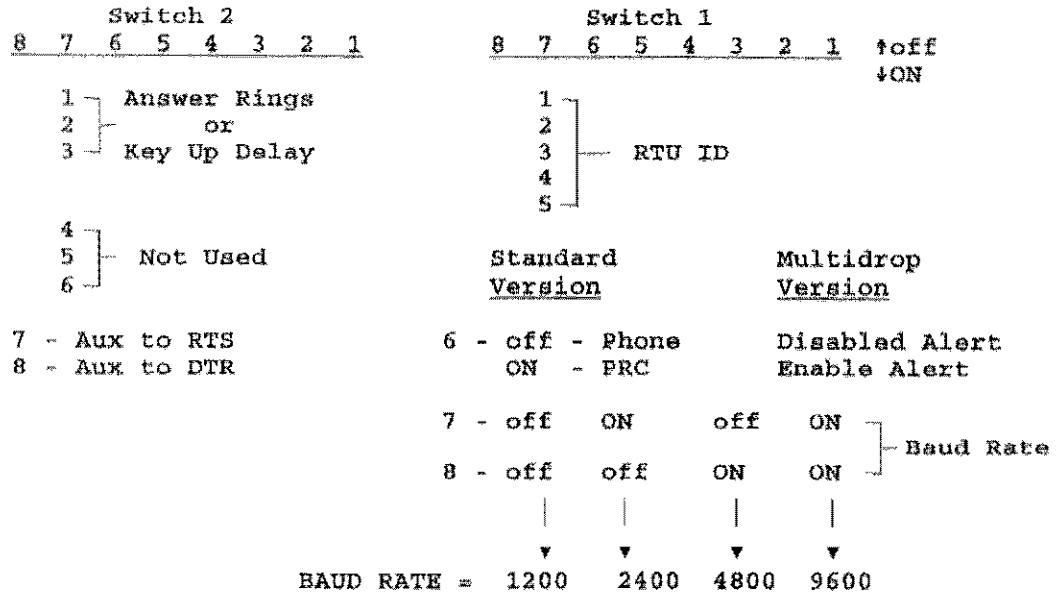


The RTU ID and Key Up Delay are set in binary as shown below:

1 - off	ON	off	ON	off	...	ON
2 - off	off	ON	ON	off	...	ON
3 - off	off	off	off	ON	...	ON
4 - off	off	off	off	off	...	ON
5 - off	off	off	off	off	...	ON
	▼	▼	▼	▼		▼
VALUE =	0	1	2	3	4	31

TOTAL ENGINEERING SERVICES TEAM, INC.  
 (TEST, INC.)  
 HARVEY, LA USA (504) 340-8378

## TYPE 2200 RTU - SCADA



The RTU ID and Key Up Delay are set in binary as shown below:

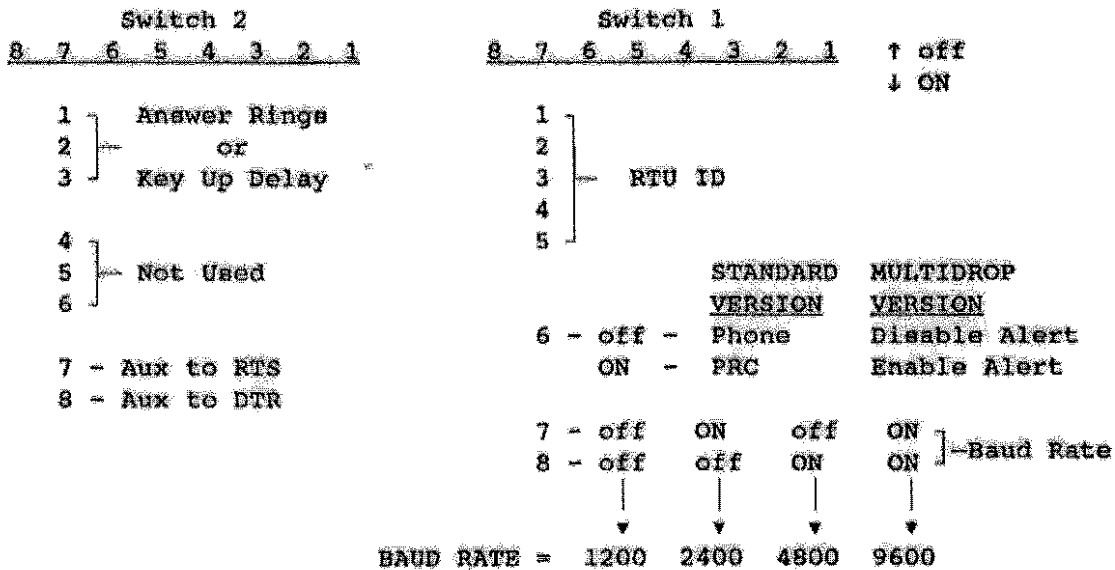
	1	- off	ON	off	ON	off	...	ON
	2	- off	off	ON	ON	off	...	ON
	3	- off	off	off	off	ON	...	ON
	4	- off	off	off	off	off	...	ON
	5	- off	off	off	off	off	...	ON
			▼	▼	▼	▼		▼
VALUE =			0	1	2	3	4	... 31

TOTAL ENGINEERING SERVICES TEAM, INC

(TEST INC)

New Orleans, La. USA (504)368-6792

Type 2200 RTU - SCADA NODE



The RTU ID and Key Up Delay are set in binary as shown below:

	1	-	off	ON	off	ON	off	...	ON
	2	-	off	off	ON	ON	off	...	ON
	3	-	off	off	off	off	ON	...	ON
	4	-	off	off	off	off	off	...	ON
	5	-	off	off	off	off	off	...	ON
			↓	↓	↓	↓	↓		↓
VALUE =			0	1	2	3	4	...	31