

875 Electrochemical Analyzer Remote Communications

Introduction

Remote communications for 875 Electrochemical Analyzers should be achieved via a PC-Based software program, RCOM, available from Invensys Foxboro. However, those users who choose not to use this program, can create their own program using the information contained in this instruction.

Connecting the Analyzer and PC

Attach cable BS809WH to your computer serial port (COM1 or COM2). If the computer serial port has a 25-pin connector, use a 25- to 9-pin adapter. Attach the other end of the cable to the RS-232-C connection on your analyzer.

Programming Information

Operating Modes

The remote control operates in one of several operating modes, matching those on the front panel. The modes are selected by a mode request message with either an acknowledgment or the mode data returned.

Devices

The Remote Control port can connect to virtually any RS232 compatible device. Only certain operating modes apply to each device.

Terminal: Continuous Measurement Data mode only

Printer: Continuous Measurement Data mode only

Modem: All modes

PC: All modes

Physical Interface

The Remote Control Port uses the industry standard RS232. Three signal lines are used: TxD, RxD, and Ground wired as DTE using a nine pin D connector. The character format is start bits {1}, data bits {7 or 8}, parity {none, even, odd}, stop bits {1 or 2}. Baud rates from 300 to 19,200 are supported. These settings must be configured in the analyzer.

Character Set

The character set used by the Remote Control port is shown below. All characters transferred are ASCII (not binary).

```
a ... z
A ... Z
0 ... 9
` ! @ # $ % ^ & * ( ) _ - + = \ | [ ] { } : ; " ' < > , . ? /
<CR> Carriage Return (0D hex)
<LF> Line Feed (0A hex)
<STX> Start of Text (02 hex)
<ETX> End of Text (03 hex)
<ACK> Positive Acknowledgement (06 hex)
<NAK> Negative Acknowledgement (15 hex)
<ESC> Escape character (1B hex)
```

Message Structure

Every outgoing data message starts with the <STX> character then an ASCII number for the number of characters (length) that follow including the <ETX> and CRC. Next, the mode and operation command of the message is output. Then one of several different menu data from “Menu Data Types” on page 22 follows. The end of the message data is marked with the <ETX> character. Finally, a four character Cyclic Redundancy Check (CRC 16) follows that includes all characters in the message from and including the <STX>, to and including the <ETX> as shown below.

```
<STX>
length <CR>
MODE: mode <CR>
OP: operation <CR>
( menu data )
<ETX>
(crc)
```

where: length is the number of characters that follow including <ETX> and crc (4 hexadecimal characters, "0000" to "FFFF")

mode is a string { "MEASURE", "STATUS", "HOLD", "CONFIG", "CAL", "DIAG", "CONNECT", "DISCONNECT", "UPLOAD_MAIN", "DOWNLOAD_MAIN", or "MAINCODE" }

operation is a string { "REQUEST", "READY", "NOTREADY", "DATA", "MENU", "RESPONSE", "EXIT", "END", "DONE" }

crc are the Cyclic Redundancy Check characters (4 characters for 16-bit hex, "0000" to "FFFF")

For ease of reading, spaces are shown but are not included in the protocol.

CRC Function

The Cyclic Redundancy Check (CRC) calculates a 16-bit check value from any data block in accordance with the following code:

```
/******
```

```
RS232_ADD_CRC
```

```
*****
```

DESCRIPTION: This routine computes the crc for the given string.

INPUT ARGS: buffer to compute crc on length of buffer

OUTPUT ARGS: none

RETURNS: crc value

NOTES: none

```
*****/
```

```
uint16
```

```
RS232_CRC(char* buffer,
```

```
uint16 length)
```

```
{
```

```
/*----- literals -----*/
```

```
#define PID 16
```

```
#define FEEDBACK 0x8408 // for CRC-CCITT
```

```
/*----- variables -----*/
```

```
char c;
```

```
uint16 crc;
```

```
uint16 bits;
```

```
uint16 index, jdex;
```

```
/******
```

```
    crc = 0xFFFF;
```

```
    // for each character in the buffer
```

```
    for (index = 0; index < length; ++index)
```

```
        {
```

```
            c = buffer[index]; // look at each bit in each character
```

```
            for (jdex = 0; jdex < 8; ++jdex)
```

```
                {
```

```

        bits = (c & 1) ^ (crc & 1);
        c >>= 1; // do an exclusive-OR between this
        // bit and the least significant bit
        // of the fcs
        crc >>= 1;
        if (bits != 0)
            crc ^= FEEDBACK; // now, do an exclusive-OR between
            // this bit and bits 1, 14, and 15 of
            // the fcs, without disturbing the
            // other bits
        }
    }
    crc = ~crc; // lastly, take the ones complement
    return(crc);
}

#undef PID
/*****

```

Flow Control Protocol

Flow control prevents data loss when the recipient's data buffer is nearly full and requires some time to empty the buffer. Flow control from either side is by XON-XOFF. When the transmitting side receives <XOFF> (data flow stop [13 hex]), the transmitter temporarily suspends data transfer. Later, when the transmitting side receives <XON> (data flow on [11 hex]), it resumes data transfer. The Flow Control characters are not included in message protocol nor the CRC.

Message Transport Protocol

For every outgoing data message, the recipient returns a data transfer acknowledgment. The <ACK> character is returned if the character parity and CRC are okay, the <NAK> character is returned otherwise. Should the sender receive a <NAK>, the message is retransmitted, up to the retry count. The 875 reports an fault if it receives the retry limit of <NAK>'s or sends the retry count of <NAK>'s without a successful reception or transmission. The 875 then returns to outputting Continuous Measurement Messages if so configured.

Message Content Faulty

Message Transport Protocol (ACK and NAK) verifies the integrity of the message transmission, not the message content. Should the data content of a message received by the 875 be faulty (but the message was transported without error), the 875 returns an error message including <STX>, <ETX>, etc.

Message Interchange

Initially, the 875 sends Continuous Measurement Messages, if so configured, believing that a printer or terminal is connected to the Remote port. When the 875's Remote port is connected to a computer, the Remote (computer) begins the interchange with the Connect Request message. The 875 discontinues the Continuous Measurement Messages and responds with the Connect Response message. At this point, the Remote requests any particular mode by sending a "mode" request message with the 875 returning a response. When the Remote wishes to discontinue the connection, it sends the Disconnect Request message. The 875 then returns to sending out Continuous Measurement Messages, if so configured. A communications inactivity of more than the configured inactivity time causes the 875 in some modes to automatically disconnect. Refer to "Message Interchange Example" immediately below. Should this occur, the remote computer re-issues the Connect Request message.

Message Interchange Example

The following figure shows a connection, configuration mode, and disconnection sequence.

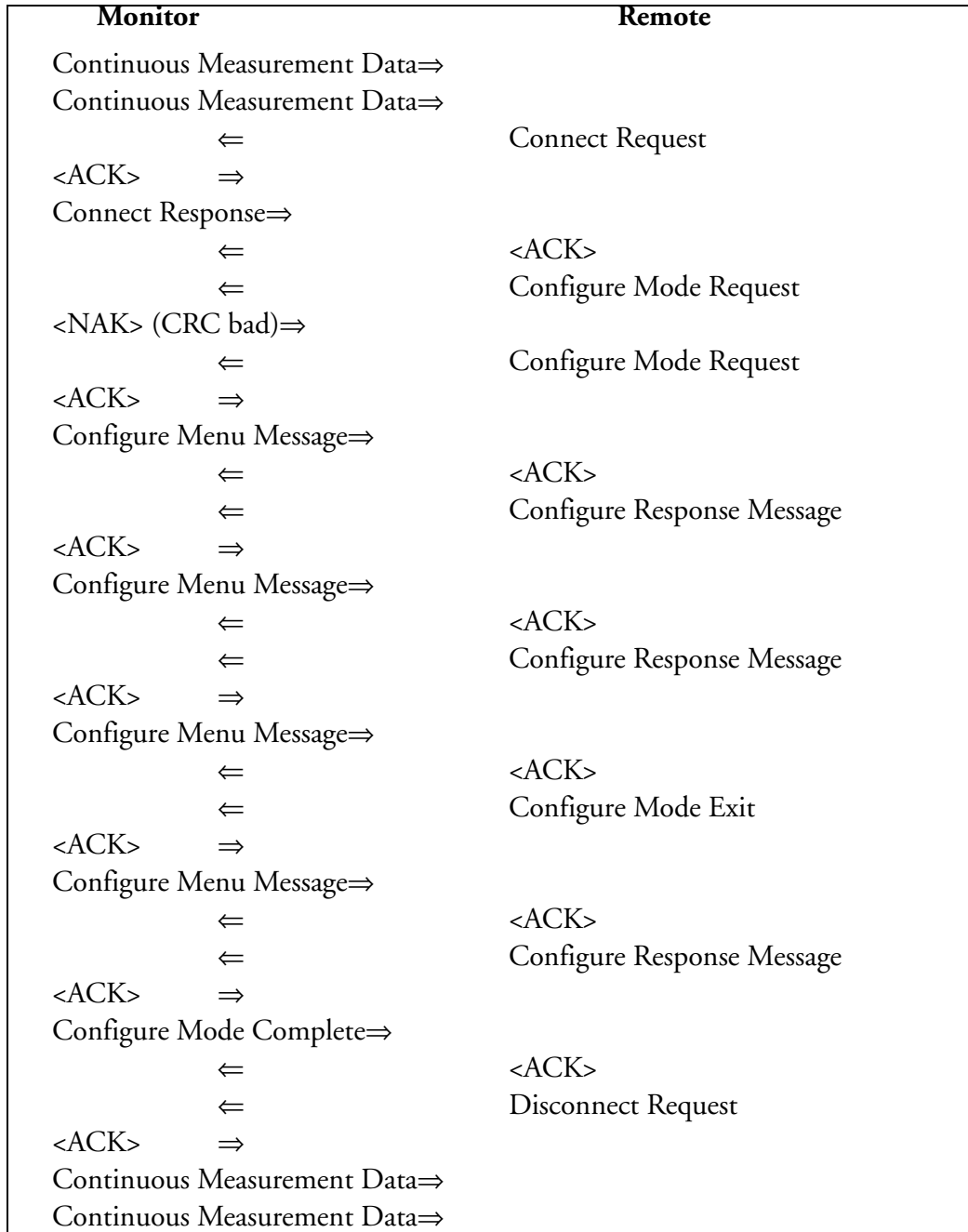


Figure 1. Message Interchange Example

Message Formats

Continuous Measurement Data

— NOTE —

This message is intended for a printer, and spaces are included as shown.

After power-up but before the Connection Message is sent or after a Connection Message followed by a Disconnection Message is sent, the 875 sends out continuous measurement data messages at a regular intervals when so configured. The interval is configurable from 5 to 3600 seconds.

Single Cell Measurement Data (875xx --> Terminal)

```
<CR><LF>
DATE: mm/dd/yy  TIME: hh:mm:ss  HOLD: hold  DEVS: dev_status <CR><LF>
MEASUREMENT: mm.mm units <CR><LF>
UNCERTAINTY: uu.uu units <CR><LF>
MVSTATUS: mv_status <CR><LF>
TEMPERATURE: tt.tt units <CR><LF>
ABSOLUTE: aa.aa units <CR><LF>
```

Dual Cell Measurement Data (875xx --> Terminal)

```
<CR><LF>
DATE: mm/dd/yy  TIME: hh:mm:ss  HOLD: hold  DEVS: dev_status <CR><LF>
MEASUREMENT 1: mm.mm units <CR><LF>
UNCERTAINTY1: uu.uu units <CR><LF>
MVSTATUS1: mv_status <CR><LF>
TEMPERATURE 1: tt.t units <CR><LF>
ABSOLUTE 1: aa.aa units <CR><LF>
MEASUREMENT 2: mm.mm units <CR><LF>
UNCERTAINTY2: uu.uu units <CR><LF>
MVSTATUS2: mv_status <CR><LF>
TEMPERATURE 2: tt.tt units <CR><LF>
ABSOLUTE 2: aa.aa units <CR><LF>
```

where: mm/dd/yy is the month, day, and year (fixed at 8 characters of text)

hh:mm:ss is the hour (24 format), minutes, and seconds (fixed at 8 characters of text)

hold is the hold status { ON or OFF } (1 to 3 characters of text)

dev_status is the device status

mm.mm is compensated measurement value (10 characters (xxxxx.xxxx) for 32-bit real)

uu.uu is the validated uncertainty (future)

mv_status is the measurement value status

tt.tt is temperature (10 characters (xxxxx.xxxx) for 32-bit real)

aa.aa is absolute measurement (10 characters (xxxxx.xxxx) for 32-bit real)

units is the value's units (1 to 6 characters of text)

A spreadsheet comma separated value (CSV) format is also available.

Connection Messages

The Connect Message allows a computer to connect to the 875's RS232 port and exchange information. After receiving a Connect Message, the 875 stops the Continuous Measurement Messages above, if any, and returns the Connect Response. The Connect Message may be issued at any time with any pass-code.

Connect Request (Remote --> 875xx)

```
<STX>
length <CR>
MODE: CONNECT <CR>
OP: REQUEST <CR>
PASSCODE: pass-code <CR>
<ETX>
(crc)
```

where:pass-code is the Level 1, 2, or 3 pass-code { "0000" to "9999" }.

For ease of reading, spaces are shown but are not included in the protocol.

Connect Response (875xx --> Remote)

```
<STX>
length <CR>
MODE: CONNECT <CR>
OP: DONE
TYPE: DATA <CR>
MODEL: model <CR>
LANG: language <CR>
HW REV: hw_rev <CR>
FW REV: fw_rev <CR>
CONFIG DATE: date <CR>
CONFIG TIME: time <CR>
LEVEL: level <CR>
<ETX>
(crc)
```

In the case that connection was not made or allowed:

Connect Response (875xx --> Remote)

```
<STX>
length <CR>
MODE: CONNECT <CR>
OP: REJECTED
<ETX>
(crc)
```

where:model is the 875's model number ("875CR", "875EC", "875PH", etc.)

language is one of {"ENGLISH", "SPANISH", "GERMAN", "FRENCH"}

hw_rev is the 875's electronics revision

fw_rev is the 875's firmware revision

date is the date of the last configuration (in the form mm/dd/yyyy where mm is the month, dd is the day of month, yyyy is the year)

time is the time of the last configuration (in the form hh:mm:ss where hh is the hour, mm is minutes, ss is seconds)

level is the security level inferred from the given pass-code { "0", "1", "2", or "3" },
(1 character of text)

For ease of reading, spaces are shown but are not included in the protocol.

The Disconnect Request Message indicates the remote computer wishes to discontinue the exchange of information with the 875. The 875 finishes any dialogues required by the current mode, (e.g. validity) and finally returns the Disconnect Response. After the 875 returns the Disconnect Response, the Continuous Measurement Messages, if so configured, resume within 5 seconds and then at the designated rate. On power-up, the Disconnect Response is output just in case the 875 was interrupted when previously connected to the Remote.

Disconnect Request (Remote --> 875xx)

```
<STX>
length <CR>
MODE: DISCONNECT <CR>
OP: REQUEST <CR>
<ETX>
(crc)
```

Disconnect Response (875xx --> Remote)

```
<STX>
length <CR>
MODE: DISCONNECT <CR>
OP: DONE <CR>
<ETX>
(crc)
```

Measurement Messages

To view measurement data, the Remote sends a Measure Mode Request message. The 875 responds with the Measure Data message (either single or dual cell). For every Measure Mode Request message the 875 returns a Measure Data message.

Measure Mode Request (Remote --> 875xx)

```
<STX>
length <CR>
MODE: MEASURE <CR>
OP: REQUEST <CR>
<ETX>
(crc)
```

Measure Data for Single Cell (875xx --> Remote)

```
<STX>
length <CR>
MODE: MEASURE <CR>
OP: DATA <CR>
TYPE: SINGLE <CR>
DATE: mm/dd/yy <CR>
TIME: hh:mm:ss <CR>
```

HOLD: hold <CR>
DEVS: dev_status <CR>
PROBE: 1 <CR>
MEASUREMENT: mm.mm units <CR>
UNCERTAINTY: uu.uu units <CR>
MVSTATUS: mv_status <CR>
TEMPERATURE: tt.t units <CR>
ABSOLUTE: aa.aa units <CR>
<ETX>
(crc)

Measure Data for Dual Cell (875xx --> Remote)

<STX>
length <CR>
MODE: MEASURE <CR>
OP: DATA <CR>
TYPE: DUAL <CR>
DATE: mm/dd/yy <CR>
TIME: hh:mm:ss <CR>
HOLD: hold <CR>
DEVS: dev_status <CR>
PROBE: 1 <CR>
MEASUREMENT: mm.mm units <CR>
UNCERTAINTY: uu.uu units <CR>
MVSTATUS: mv_status <CR>
TEMPERATURE: tt.t units <CR>
ABSOLUTE: aa.aa units <CR>
PROBE: 2 <CR>
MEASUREMENT: mm.mm units <CR>
UNCERTAINTY: uu.uu units <CR>
MVSTATUS: mv_status <CR>
TEMPERATURE: tt.t units <CR>
ABSOLUTE: aa.aa units <CR>
<ETX>
(crc)

where:mm/dd/yy is the month, day, and year (fixed at 8 characters of text)

hh:mm:ss is the hour (24 format), minutes, and seconds (fixed at 8 characters of text)

hh is the hold status { "ON" or "OFF" } (1 to 3 characters of text)

dev_status is the device status

mm.mm is compensated measurement value (10 characters (xxxxx.xxxx) for 32-bit real)

uu.uu is the validated uncertainty (future)

mv_status is the measurement value status)

tt.t is temperature (10 characters (xxxxx.xxxx) for 32-bit real)

aa.aa is absolute measurement (10 characters (xxxxx.xxxx) for 32-bit real)

units is the value's units; a space separates the value and units

For ease of reading, spaces are shown but are not included in the protocol.

Status Message

To view status data, the Remote sends a Status Mode Request message. The 875 responds with one or more Status Info Data messages in series. The data items are sent in the same order and format as they appear in the front panel Status Mode menus. For every Status Mode Request message the 875 returns a sequence of Status Info Data messages.

Status Mode Request (Remote --> 875xx)

```
<STX>
length <CR>
MODE: STATUS <CR>
OP: REQUEST <CR>
<ETX>
(crc)
```

Status Info Data (875xx --> Remote)

```
<STX>
length <CR>
MODE: STATUS <CR>
OP: DATA <CR>
( info data )
<ETX>
(crc)
```

```
<STX>
length <CR>
MODE: STATUS <CR>
OP: DATA <CR>
( info data )
<ETX>
(crc)
```

```
<STX>
length <CR>
MODE: STATUS <CR>
OP: END <CR>
( info data )
<ETX>
(crc)
```

where:info data is defined in “Menu Data Types” on page 22.

For ease of reading, spaces are shown but are not included in the protocol.

The monitor sends multiple info messages sequentially. There is no reply from the remote host. The sequence is as follows:

Application number as numeric info (if more than one configured)

Temperature 1 as numeric info

Absolute 1 as numeric info

ATC 1 as numeric info

Cal Person 1 as string info

Cal date 1 as date info

Cal type 1 as string info

If second probe in use, include the following:

Temperature 2 as numeric info

Absolute 2 as numeric info

ATC 2 as numeric info

Cal Person 2 as string info

Cal date 2 as date info

Cal type 2 as string info

Analog output 1 as numeric info

Analog output 2 as numeric info

Analog output 3 as numeric info

Device name as string info

Device location as string info

Device sales order as string info

Device MS code as string info

Device serial number as string info

Firmware version as string info

Device type as string info

Comm type as picklist info

Device Tag as string info

Date as date info

Time as time info

From 1 to 10 history log entries as text info

Hold Messages

To view or change hold data, the Remote sends a Hold Mode Request message. If the pass-code level is insufficient, the 875 replies with the Hold Reject message. Otherwise, the 875 responds with the contents described in “Menu Data Types” on page 22. After deciding the appropriate response and/or changes, the Remote returns a Hold Response message. This results in the 875

returning the next Hold Menu message in the sequence. This interchange continues until the Remote wishes to exit hold mode by sending the Hold Mode Exit message. At this point the hold mode is exited and the 875 returns the Hold Mode Complete message, followed by the Connect Response message.

Alternatively, if the Remote's returned Hold Response Data message is in error, meaning the returned data is faulty, the 875 returns the Hold Mode Error message and exits hold mode, issuing the Hold Mode Complete and Connect Response messages.

Hold Mode Request (Remote --> 875xx)

```
<STX>
length <CR>
MODE: HOLD <CR>
OP: REQUEST <CR>
<ETX>
(crc)
```

Hold Reject (875xx --> Remote)

```
<STX>
length <CR>
MODE: HOLD <CR>
OP: REJECTED <CR>
<ETX>
(crc)
```

Hold Menu Message (875xx --> Remote)

```
<STX>
length <CR>
MODE: HOLD <CR>
OP: MENU <CR>
( menu data )
<ETX>
(crc)
```

Hold Response Message (Remote --> 875xx)

```
<STX>
length <CR>
MODE: HOLD <CR>
OP: RESPONSE <CR>
( response data )
<ETX>
(crc)
```

Hold Mode Exit (Remote --> 875xx)

```
<STX>
length <CR>
MODE: HOLD <CR>
```

```
OP: EXIT <CR>
<ETX>
(crc)
```

Hold Mode Complete (875xx --> Remote)

```
<STX>
length <CR>
MODE: HOLD <CR>
OP: DONE <CR>
<ETX>
(crc)
```

Hold Mode Error (875xx --> Remote)

```
<STX>
length <CR>
MODE: HOLD <CR>
OP: ERROR <CR>
<ETX>
(crc)
```

where:menu data is defined. See “Menu Data Types” on page 22.

response data is defined. See “Menu Data Types” on page 22.

For ease of reading, spaces are shown but are not included in the protocol.

Configuration Messages

To view or change configuration data, the Remote sends a Configure Mode Request message. The 875 responds with the Configure Menu message. After deciding the appropriate response and/or changes, the Remote returns a Configure Response message. This results in the 875 returning the next Configure Menu message in the sequence. This interchange continues until the Remote wishes to exit configuration mode by sending the Configure Mode Exit message. At this point, if no changes were made, the configuration mode is exited when the 875 returns the Configure Mode Complete message. If changes were made, the Remote is returned Configure Menu message that asks if the changes are to be kept or aborted. After deciding the appropriate response, the Remote returns a Configure Response message. The 875 sends a Configure Mode Complete message and the configuration mode interchange is finished. See the applicable analyzer instruction for the Configuration Exit Flow.

Alternatively, if the Remote's returned Configure Response message is in error, meaning the returned data is faulty, the 875 returns the Configure Mode Error message, exits configuration mode, and aborts pending changes.

Configure Mode Request (Remote --> 875xx)

```
<STX>
length <CR>
MODE: CONFIG <CR>
OP: REQUEST <CR>
```

<ETX>
(crc)

Configure Menu message (875xx --> Remote)

<STX>
length <CR>
MODE: CONFIG <CR>
OP: MENU <CR>
(menu data)
<ETX>
(crc)

Configure Response message (Remote --> 875xx)

<STX>
length <CR>
MODE: CONFIG <CR>
OP: RESPONSE <CR>
(response data)
<ETX>
(crc)

Configure Mode Exit (Remote --> 875xx)

<STX>
length <CR>
MODE: CONFIG <CR>
OP: EXIT <CR>
<ETX>
(crc)

Configure Mode Complete (875xx --> Remote)

<STX>
length <CR>
MODE: CONFIG <CR>
OP: DONE <CR>
<ETX>
(crc)

Configure Mode Error (875xx --> Remote)

<STX>
length <CR>
MODE: CONFIG <CR>
OP: ERROR <CR>
<ETX>
(crc)

where:menu data is defined. See “Menu Data Types” on page 22.

response data is defined. See “Menu Data Types” on page 22.

For ease of reading, spaces are shown but are not included in the protocol.

Calibration Messages

To perform calibration, the Remote sends a Calibrate Mode Request message. If the pass-code is insufficient, the 875 replies with the Calibrate Rejected message. Otherwise, the 875 responds with the Calibrate Menu message. After deciding the appropriate response, the Remote returns a Calibrate Response message. This results in the 875 returning the next Calibrate Menu message in the sequence. This interchange continues until the calibration procedure is complete wherein the 875 returns the Calibrate Mode Complete message. The Remote can abort calibration at any time by sending the Calibrate Mode Exit message wherein the 875 returns the Calibrate Mode Complete message and the prior calibration is retained.

Alternatively, if the Remote's returned Calibrate Response Data message is in error, meaning the returned data is faulty, the 875 returns the Calibrate Mode Error message and exits calibration mode, issuing the Calibrate Mode Complete message,

Calibrate Mode Request (Remote --> 875xx)

```
<STX>
length <CR>
MODE: CAL <CR>
OP: REQUEST <CR>
<ETX>
(crc)
```

Calibrate Rejected (875xx --> Remote)

```
<STX>
length <CR>
MODE: CAL <CR>
OP: REJECTED <CR>
<ETX>
(crc)
```

Calibrate Menu message (875xx --=> Remote)

```
<STX>
length <CR>
MODE: CAL <CR>
OP: MENU <CR>

( menu data )

<ETX>
(crc)
```

Calibrate Response message (Remote --> 875xx)

```
<STX>
length <CR>
MODE: CAL <CR>
OP: RESPONSE <CR>

( response data )

<ETX>
(crc)
```


Calibrate Mode Exit (Remote --> 875xx)

```
<STX>
length <CR>
MODE: CAL <CR>
OP: EXIT <CR>
<ETX>
(crc)
```

Calibrate Mode Complete (875xx --> Remote)

```
<STX>
length <CR>
MODE: CAL <CR>
OP: DONE <CR>
<ETX>
(crc)
```

Calibrate Mode Error (875xx --> Remote)

```
<STX>
length <CR>
MODE: CAL <CR>
OP: ERROR <CR>
<ETX>
(crc)
```

where:menu data is defined. See “Menu Data Types” on page 22.

response data is defined. See “Menu Data Types” on page 22.

For ease of reading, spaces are shown but are not included in the protocol.

Diagnostic Message

To view and possibly suspend faults and to view and clear the history log, the Remote sends a Diagnostic Mode Request message. The 875 responds with the Diagnostic Menu message. After deciding the appropriate response, the Remote returns a Diagnostic Response message. This results in the 875 returning the next Diagnostic Menu message in the sequence. This interchange continues until the Remote wishes to exit diagnostic mode by sending the Diagnostic Mode Exit message. The diagnostic mode is exited when the 875 returns the Diagnostic Mode Complete message. Since the 875 remains connected, the Connect Response message is also issued.

Alternatively, if the Remote's returned Diagnostic Response message is in error, meaning the returned data is faulty, the 875 returns the Diagnostic Mode Error message and exits diagnostic mode, issuing the Diagnostic Mode Complete and the Connect Response messages.

Diagnostic Mode Request (Remote --> 875xx)

```
<STX>
length <CR>
MODE: DIAG <CR>
OP: REQUEST <CR>
<ETX>
(crc)
```

Diagnostic Menu message (875xx --> Remote)

```
<STX>  
length <CR>  
MODE: DIAG <CR>  
OP: MENU <CR>  
  
( menu data )  
  
<ETX>  
(crc)
```

Diagnostic Response message (Remote --> 875xx)

```
<STX>  
length <CR>  
MODE: DIAG <CR>  
OP: RESPONSE <CR>  
  
( response data )  
  
<ETX>  
(crc)
```

Diagnostic Mode Exit (Remote --> 875xx)

```
<STX>  
length <CR>  
MODE: DIAG <CR>  
OP: EXIT <CR>  
  
<ETX>  
(crc)
```

Diagnostic Mode Complete (875xx --> Remote)

```
<STX>  
length <CR>  
MODE: DIAG <CR>  
OP: DONE <CR>  
  
<ETX>  
(crc)
```

Diagnostic Mode Error (875xx --> Remote)

```
<STX>  
length <CR>  
MODE: DIAG <CR>  
OP: ERROR <CR>  
  
<ETX>  
(crc)
```

where: menu data is defined. See “Menu Data Types” on page 22.

response data is defined. See “Menu Data Types” on page 22.

For ease of reading, spaces are shown but are not included in the protocol.

Upload Configuration Message

To upload configuration data from the 875, the Remote sends the Upload Request message. The 875 responds with one or more Upload Data message(s). The final block is identified with an operation code of OP:DONE. Once the Upload Request message is received, the 875 is committed to transmitting all the data.

Upload Request (Remote --> 875xx)

```
<STX>
length <CR>
MODE: UPLOAD_MAIN <CR>
OP: REQUEST <CR>
<ETX>
(crc)
```

Upload Data (875xx --> Remote)

```
<STX>
length <CR>
MODE: UPLOAD_MAIN <CR>
OP: DATA <CR>
( data )
<ETX>
(crc)
```

```
<STX>
length <CR>
MODE: UPLOAD_MAIN <CR>
OP: DATA <CR>
( data )
<ETX>
(crc)
```

```
<STX>
length <CR>
MODE: UPLOAD_MAIN <CR>
OP: DONE <CR>
<ETX>
(crc)
```

where: data is 2 to 512 characters of ASCII data, consisting of name and value for each item.

For ease of reading, spaces are shown but are not included in the protocol.

Download Configuration Message

To download configuration data to the 875, the Remote sends the Download Request message. If the pass-code level is insufficient, or if the database is unavailable, the 875 returns the Download

Rejected message and the Remote retries. Otherwise, the 875 responds with Download Ready message. When Download Ready is received, the Remote then sends the first Download Data message. Download Data messages continue to be sent until the last message shows an operation code of OP:EXIT. Once the Download Ready message is sent, the 875 is committed to receiving all the data. All data blocks within the sequence of Download Data messages are written to temporary storage. After the last data block, the remote issues the Download Exit message. The 875 returns Download Menu message that asks if the changes are to be kept or aborted. After deciding the appropriate response, the Remote returns a Download Response message. The 875 sends a Configure Mode Complete message and the configuration mode interchange is finished. See the applicable analyzer instruction.

Download Request (Remote --> 875xx)

```
<STX>  
length <CR>  
MODE: DOWNLOAD_MAIN <CR>  
OP: REQUEST <CR>  
<ETX>  
(crc)
```

Download Rejected (875xx --> Remote)

```
<STX>  
length <CR>  
MODE: DOWNLOAD_MAIN <CR>  
OP: REJECTED <CR>  
<ETX>  
(crc)
```

Download Ready (875xx --> Remote)

```
<STX>  
length <CR>  
MODE: DOWNLOAD_MAIN <CR>  
OP: READY <CR>  
<ETX>  
(crc)
```

Download Data (Remote --> 875xx)

```
<STX>  
length <CR>  
MODE: DOWNLOAD_MAIN <CR>  
OP: DATA <CR>  
  
( data )  
<ETX>  
(crc)
```

```
<STX>  
length <CR>
```

```

MODE: DOWNLOAD_MAIN <CR>
OP: DATA <CR>
  ( data )
<ETX>
(crc)

```

Download Exit (Remote --> 875xx)

```

<STX>
length <CR>
MODE: DOWNLOAD_MAIN <CR>
OP: EXIT <CR>
<ETX>
(crc)

```

Download Menu message (875xx --> Remote)

```

<STX>
length <CR>
MODE: DOWNLOAD_MAIN <CR>
OP: MENU <CR>
  ( menu data )
<ETX>
(crc)

```

Download Response message (Remote --> 875xx)

```

<STX>
length <CR>
MODE: DOWNLOAD_MAIN <CR>
OP: RESPONSE <CR>
  ( response data )
<ETX>
(crc)

```

Download Mode Complete (875xx --> Remote)

```

<STX>
length <CR>
MODE: DOWNLOAD_MAIN <CR>
OP: DONE <CR>
  ( data )
<ETX>
(crc)

```

where:data is two to 512 characters of ASCII data

For ease of reading, spaces are shown but are not included in the protocol.

Menu Data Types

These menus describe data intended for a human user (not a computer). The following menu data types apply to all modes. In general, there are three types of menu data:

- ◆ Menu Data
- ◆ Response Data
- ◆ Information Data

Menu Data is of type "TEXT", "PICKLIST", "MULTIPLE", "NUMERIC", "STRING", or "DATETIME" as defined below. Menu Data contains everything about a particular menu. Menu Data is used to present the user (or computer) with an option to change a configurable option and its particulars.

Response Data is of type "TEXT", "PICKLIST", "MULTIPLE", "NUMERIC", "STRING", or "DATETIME" as defined below. Text Response returns the users response { "YES", "NO", or "ENTER" } which correspond to the front panel keys. YES and NO are returned when a question is posed. ENTER is returned when a user acknowledgement is requested, or an automatic time-out is specified.

The other menu types return the user's response { PREV, NEXT, or ENTER } which corresponds to the front panel keys and the associated choice, value, or string parameter data. PREV routes to the previous menu, if any, while NEXT routes to the next menu in the series and both without making changes to the associated data. ENTER also routes to the next menu and the associated data "may" have changed.

Information Data is of type "TEXT", "PICKLIST", "MULTIPLE", "NUMERIC", "STRING", or "DATETIME" as defined below. Information Data provides only the bare minimum of the menu data and is used when viewing (not changing) a menu. Information Data is used only by the Status Message defined in "Status Message" on page 11.

Each data type (Menu, Response, or Information Data) is composed of several terms each with an identifier and data. An identifier is capitalized followed by a colon (":"), the term's data, and finally a carriage return (<CR>). For ease of reading, spaces are shown but are not included in the protocol. For example,

```
NAME: name <CR>
```

where NAME: is the identifier and name is the term's data. Note that terms contain all ASCII characters. Also, the term's data, if a word or phase (rather than a number), is in most cases multilingual as configured. The <CR> is present to view messages during debugging.

Adjust Menu Data

Adjust Menu Data provides a fine, medium or coarse adjustment of a numeric value parameter. The Adjust Response Data is returned showing a new or same adjustment value.

```
TYPE: ADJUST <CR>
NAME: name <CR>
HELP: help <CR>
TITLE: title <CR>
ATTRIB: attributes <CR>
TERMS: terms<CR>
```

VALUE: value <CR>
 MIN: min <CR>
 MAX: max <CR>
 FINE: delta_fine <CR>
 MEDIUM: delta_medium <CR>
 COARSE: delta_coarse <CR>
 UNITS: units <CR>

where:name is the name of the data

help is a verbose description of the parameter

title is the parameter's title

attributes of menu (1 character for 4 hex bits, "0" - "9" and "A" - "F"),

bit 0 = viewable(0) or changeable(1)

bit 1 = ok (0) or invalid or out-of-range (1)

bit 2 = unused

bit 3 = unused

terms describes the three granularity positions, generally terms contains "COARSE
MEDIUM FINE"

value is the adjustable parameter's value (1 to 11 characters for 32-bit decimal real,
"-99999.9999" to "99999.9999")

min is the value's minimum range (1 to 11 characters for 32-bit decimal real,
"-99999.9999" to "99999.9999")

max is the value's minimum range (1 to 11 characters for 32-bit decimal real,
"-99999.9999" to "99999.9999")

fine_delta is the amount integer is incremented or decremented by (1 to 11 characters for 32-bit decimal real, "-99999.9999" to "99999.9999")

medium_delta is the amount integer is incremented or decremented by (1 to 11 characters for 32-bit decimal real, "-99999.9999" to "99999.9999")

coarse_delta is the amount integer is incremented or decremented by (1 to 11 characters for 32-bit decimal real, "-99999.9999" to "99999.9999") units is the value's unit of measure

For ease of reading, spaces are shown but are not included in the protocol.

Adjust Response Data

This data is returned in response to the Adjust Menu Data.

TYPE: ADJUST <CR>
 NAME: name <CR>
 RESPONSE: response <CR>
 VALUE: value <CR>

where:name is the name of the data

response is one of the following { "PREV", "NEXT", or "ENTER" }

value is the parameter's value (1 to 11 characters for 32-bit decimal real, "-99999.9999" to "99999.9999")

For ease of reading, spaces are shown but are not included in the protocol.

Adjust Info Data

This data type is not available or necessary.

Date and Time Menu Data

Date and Time Menu Data are separate and each provides three parameters. The Date or Time Response Data is returned showing the new date or time.

Date Data

TYPE: DATE <CR>
NAME: name <CR>
HELP: help <CR>
TITLE: title <CR>
ATTRIB: attributes <CR>
DAY: day <CR>
MONTH: month <CR>
YEAR: year <CR>

Time Data

TYPE: TIME <CR>
NAME: name <CR>
HELP: help <CR>
TITLE: title <CR>
ATTRIB: attributes <CR>
SECONDS: seconds <CR>
MINUTES: minutes <CR>
HOURS: hours <CR>

where:name is the name of the data

help is a verbose description of the parameter

title is the parameter's title

attributes of menu (1 character for 4 hex bits, "0" - "9" and "A" - "F"),

bit 0 = viewable(0) or changeable(1)

bit 1 = unused

bit 2 = unused

bit 3 = unused

seconds is a parameter value (1 to 2 characters of text, "0" to "59")

minutes is a parameter value (1 to 2 characters of text, "0" to "59")

hours is a parameter value (1 to 2 characters of text, "1" to "24")

day is a parameter value (1 to 2 characters of text, "1" to "31")

month is a parameter value (1 to 2 characters of text, "1" to "12")

year is a parameter value (4 characters of text, "0000" to "9999")

For ease of reading, spaces are shown but are not included in the protocol.

Date and Time Response Data

This data is returned in response to the Date or Time Menu Data.

Date Data

TYPE: DATE <CR>
 NAME: name <CR>
 RESPONSE: response <CR>
 DAY: day <CR>
 MONTH: month <CR>
 YEAR: year <CR>

Time Data

TYPE: TIME <CR>
 NAME: name <CR>
 RESPONSE: response <CR>
 SECONDS: seconds <CR>
 MINUTES: minutes <CR>
 HOURS: hours <CR>

where:name is the name of the data

response is one of the following { "PREV", "NEXT", or "ENTER" }

seconds is a parameter value (1 to 2 characters of text, "0" to "59")

minutes is a parameter value (1 to 2 characters of text, "0" to "59")

hours is a parameter value (1 to 2 characters of text, "1" to "24")

day is a parameter value (1 to 2 characters of text, "1" to "31")

month is a parameter value (1 to 2 characters of text, "1" to "12")

year is a parameter value (4 characters of text, "0000" to "9999")

For ease of reading, spaces are shown but are not included in the protocol.

Date and Time Info Data

This data is made available in Status Mode.

Date Data

TYPE: DATE <CR>
 NAME: name <CR>
 HELP: help <CR>
 TITLE: title <CR>
 DAY: day <CR>
 MONTH: month <CR>
 YEAR: year <CR>

Time Data

```
TYPE: TIME <CR>
NAME: name <CR>
HELP: help <CR>
TITLE: title <CR>
SECONDS: seconds <CR>
MINUTES: minutes <CR>
HOURS: hours <CR>
```

where:name is the name of the data

help is a verbose description of the parameter

title is the parameter's title

seconds is a parameter value (1 to 2 characters of text, "0" to "59")

minutes is a parameter value (1 to 2 characters of text, "0" to "59")

hours is a parameter value (1 to 2 characters of text, "1" to "24")

day is a parameter value (1 to 2 characters of text, "1" to "31")

month is a parameter value (1 to 2 characters of text, "1" to "12")

year is a parameter value (4 characters of text, "0000" to "9999")

For ease of reading, spaces are shown but are not included in the protocol.

Live Menu Data

Live menu data conveys text that displays live measurement updates during the calibration process. Live Response Data is returned as in other menu types, but the Live menu timing is quite different from other menus. Live menu is intended as a quick display update. As such, it returns its reply as quickly as possible, but that returned reply is accumulated from any time since the previous Live Menu Data. This contrasts with all other menu types that display data and wait indefinitely for some user response before the reply message is generated.

```
TYPE: LIVE <CR>
NAME: name <CR>
HELP: help <CR>
TITLE: title <CR>
VALUE1: line1 <CR>
UNITS1: units1 <CR>
VALUE2: line2 <CR>
UNITS2: units2 <CR>
```

where:name is the name of the data

help is a verbose description of the data

title is the data's title

line1 is text representing the first display value

unit1 is text representing the units for the first display value

line2 is text representing the second display value

unit2 is text representing the units for the second display value

— NOTE

The first display value is likely to be changed once the measurement is made. It is intended to have the same look as Numeric Menu Data, but without the modification options.

For ease of reading, spaces are shown but are not included in the protocol.

Live Response Data

This data is returned in response to the Text Menu Data.

```
TYPE: LIVE <CR>
NAME: name <CR>
RESPONSE: response <CR>
```

where:name is the name of the data

response is one of the following { "NO", or "ENTER" }

For ease of reading, spaces are shown but are not included in the protocol.

Live Info Data

Live menu info is not implemented or required.

Multiple Menu Data

Multiple Menu Data provides a list of choices from which one or more is selected for the parameter. The Multiple Response Data is returned showing the selected choice(s).

```
TYPE: MULTIPLE <CR>
NAME: name <CR>
HELP: help <CR>
TITLE: title <CR>
ATTRIB: attributes <CR>
VISIBLE: visible_bits <CR>
SELECTION: selection_bits <CR>
CHOICE:choice_1<CR>
```

```
CHOICE:choice_n<CR>
```

```
CHOICE END: <CR>
```

where:name is the name of the data

help is a verbose description of the parameter

title is the parameter's title

attributes of menu (1 character for 4 hex bits, "0" - "9" and "A" - "F"),

bit 0 = viewable(0) or changeable(1)

bit 1 =

bit 2 =

bit 3 =

visible_bits indicates the choices that are visible (8 characters for 32-bit hex, 1 or more bits of 32, "00000000" to "FFFFFFFF")

selection_bits is the currently selected choices (8 characters for 32-bit hex, 1 or more bits of 32, "00000000" to "FFFFFFFF")

choice_1...choice_n is one of several choices text

For ease of reading, spaces are shown but are not included in the protocol.

Multiple Response Data

This data is returned in response to the Multiple Menu Data.

TYPE: MULTIPLE <CR>
NAME: name <CR>
RESPONSE: response <CR>
SELECTION: selection_bits <CR>

where:name is the name of the menu

response is one of the following { "PREV", "NEXT", or "ENTER" }

selection_bits are the new selected choices (8 characters for 32-bit hex, 1 or more bits of 32, "00000000" to "FFFFFFFF")

For ease of reading, spaces are shown but are not included in the protocol.

Multiple Info Data

This data is made available in Status Mode.

TYPE: MULTIPLE <CR>
NAME: name <CR>
HELP: help <CR>
TITLE: title <CR>
CHOICE: choice_1 <CR>

CHOICE: choice_n <CR>

where:name is the name of the data

help is a verbose description of the parameter

title is the parameter's title line

choice_1,..., choice_n is one of several selected choices text

For ease of reading, spaces are shown but are not included in the protocol.

Numeric Menu Data

Numeric Menu Data provides a numeric value parameter. The Numeric Response Data is returned showing new or same value.

TYPE: NUMERIC <CR>
NAME: name <CR>

HELP: help <CR>
 TITLE: title <CR>
 ATTRIB: attributes <CR>
 VALUE: value<CR>
 MIN: min <CR>
 MAX: max <CR>
 DELTA: delta <CR>
 UNITS: units <CR>

where:name is the name of the data

help is a verbose description of the parameter

title is the parameter's title

attributes of menu (1 character for 4 hex bits, "0" - "9" and "A" - "F"),

bit 0 = viewable(0) or changeable(1)

bit 1 = ok (0) or invalid or out-of-range (1)

bit 2 =

bit 3 =

value is the parameter's value (1 to 11 characters for 32-bit decimal real, "-99999.9999" to "99999.9999")

min is the value's minimum range (1 to 11 characters for 32-bit decimal real, "-99999.9999" to "99999.9999")

max is the value's minimum range (1 to 11 characters for 32-bit decimal real, "-99999.9999" to "99999.9999")

delta is the amount integer is incremented or decremented by (1 to 11 characters for 32-bit decimal real, "-99999.9999" to "99999.9999")

units is the value's unit of measure

For ease of reading, spaces are shown but are not included in the protocol.

Numeric Response Data

This data is returned in response to the Numeric Menu Data.

TYPE: NUMERIC <CR>
 NAME: name <CR>
 RESPONSE: response <CR>
 VALUE: value <CR>

where:name is the name of the data

response is one of the following { "PREV", "NEXT", or "ENTER" }

value is the parameter's value (1 to 11 characters for 32-bit decimal real, "-99999.9999" to "99999.9999")

For ease of reading, spaces are shown but are not included in the protocol.

Numeric Info Data

This data is made available in Status Mode.

TYPE: NUMERIC <CR>
NAME: name <CR>
HELP: help <CR>
TITLE: title <CR>
VALUE: value <CR>
UNITS: units <CR>

where:name is the name of the data

help is a verbose description of the parameter

title is the parameter's title

value is the parameter's value (1 to 11 characters for 32-bit decimal real,
" 99999.9999" to "99999.9999") units is the value's unit of measure

For ease of reading, spaces are shown but are not included in the protocol.

Picklist Menu Data

Picklist Menu Data provides a list of choices from which one is selected for the parameter. The Picklist Response Data is returned showing the selected choice.

TYPE: PICKLIST <CR>
NAME: name <CR>
HELP: help <CR>
TITLE: title <CR>
ATTRIB: attributes <CR>
VISIBLE: visible_bits <CR>
SELECTION: selection_bit <CR>
CHOICE:choice_1<CR>

CHOICE:choice_n<CR>
CHOICE END: <CR>

where:name is the name of the data

help is a verbose description of the parameter

title is the parameter's title

attributes of menu (1 character for 4 hex bits, "0" - "9" and "A" - "F"),

bit 0 = viewable(0) or changeable(1)
bit 1 = regular menu (0) or top-level menu (1)
bit 2 =
bit 3 =

visible_bits indicates the choices that are visible (8 characters for 32-bit hex, 1 or more bits of 32, "00000000" to "FFFFFFFF")

selection_bit is the currently selected choice (8 characters for 32-bit hex, 1 bit of 32, "00000000" to "FFFFFFFF")

choice_1...choice_n is one of several choices text

For ease of reading, spaces are shown but are not included in the protocol.

Picklist Response Data

This data is returned in response to the Picklist Menu Data.

```
TYPE: PICKLIST <CR>
NAME: name <CR>
RESPONSE: response <CR>
SELECTION: selection_bit <CR>
```

where:name is the name of the data

response is one of the following { "PREV", "NEXT", "ENTER" }

selection_bit is the new selected choice (8 characters for 32-bit hex, 1 bit of 32, "00000000" to "FFFFFFFF")

For ease of reading, spaces are shown but are not included in the protocol.

Picklist Info Data

This data is made available in Status Mode.

```
TYPE: PICKLIST <CR>
NAME: name <CR>
HELP: help <CR>
TITLE: title <CR>
CHOICE: choice <CR>
```

where:name is the name of the data

help is a verbose description of the parameter

title is the parameter's title

choice is the selected choice text

For ease of reading, spaces are shown but are not included in the protocol.

String Menu Data

String Menu Data provides a string parameter. The String Response Data is returned showing the new or same string.

```
TYPE: STRING <CR>
NAME: name <CR>
HELP: help <CR>
TITLE: title <CR>
ATTRIB: attributes <CR>
STRING: string <CR>
```

where:name is the name of the data

help is a verbose description of the parameter

title is the parameter's title

attributes of menu (1 character for 4 hex bits, "0" - "9" and "A" - "F"),

bit 0 = viewable(0) or changeable(1)

bit 1 =

bit 2 =

bit 3 =

string is the parameter, padded with spaces to its full length

For ease of reading, spaces are shown but are not included in the protocol.

String Response Data

This data is returned in response to the String Menu Data.

TYPE: STRING <CR>

NAME: name <CR>

RESPONSE: response <CR>

STRING: string <CR>

where:name is the name of the parameter

response is one of the following { "PREV", "NEXT", or "ENTER" }

string is the new string menu parameter. Extra characters are truncated without error.

For ease of reading, spaces are shown but are not included in the protocol.

String Info Data

This data is made available in Status Mode.

TYPE: STRING <CR>

NAME: name <CR>

HELP: help <CR>

TITLE: title <CR>

STRING: string <CR>

where:name is the name of the data

help is a verbose description of the parameter

title is the parameter's title

string is the parameter

For ease of reading, spaces are shown but are not included in the protocol.

Text Menu Data

Text menu data conveys text that is displayed indefinitely or for a brief time. When the pause time expires or a response is readied, whichever comes first, the Text Response Data is returned.

TYPE: TEXT <CR>

NAME: name <CR>

HELP: help <CR>

TITLE: title <CR>

ATTRIB: attributes <CR>

MAJOR: major <CR>
 MINOR: minor <CR>
 PAUSE: pause <CR>

where:name is the name of the data

help is a verbose description of the data

title is the data's title

attributes of menu (1 character for 4 hex bits, "0" - "9" and "A" - "F"),

bit 0 = viewable(0) or changeable(1)
 bit 1 = regular menu (0) or top-level menu (1)
 bit 2 = mode inhibit (0) or mode allowed (1)
 bit 3 =

major is the major text line

minor is the minor text line

pause is the maximum time in seconds that the text is displayed (1 to 5 characters for 16-bit decimal integer, "0" means infinite time and return text menu response from user, real delays from 1 to "65535")

For ease of reading, spaces are shown but are not included in the protocol.

Text Response Data

This data is returned in response to the Text Menu Data.

TYPE: TEXT <CR>
 NAME: name <CR>
 RESPONSE: response <CR>

where:name is the name of the data

response is one of the following { "YES", "NO", "ENTER" }

For ease of reading, spaces are shown but are not included in the protocol.

TEXT Info Data

Text menu data conveys text that is displayed. No response is returned.

TYPE:TEXT<CR>
 NAME:name<CR>
 HELP:help<CR>
 TITLE:title<CR>
 MAJOR:major<CR>
 MINOR:minor<CR>

where:name is the name of the data

help is a verbose description of the data

title is the data's title

major is the major text line

minor is the minor text line

33 Commercial Street
Foxboro, MA 02035-2099
United States of America

<http://www.foxboro.com>
Inside U.S.: 1-866-746-6477
Outside U.S. 1-508-549-2424 or
contact your local Foxboro
Representative.
Facsimile: (508) 549-4999

Invensys and Foxboro are trademarks of Invensys plc, its subsidiaries, and affiliates.
All other brand names may be trademarks of their respective owners.

Copyright 2008 Invensys Systems, Inc.
All right reserved

MB 123

Printed in U.S.A.

0508