

Brooks PetroCount® Inventory Management System

Data Communications Interface User Guidelines

Essential Instructions

Brooks PetroCount Inventory Management System (IMS) is a product of Fisher-Rosemount Petroleum.

Fisher-Rosemount Petroleum (FRP) designs, manufactures, and tests its products to meet many national and international standards. The products sold and distributed by Fisher-Rosemount Petroleum are sophisticated technical instruments that must be properly installed, used, and maintained to ensure they continue to operate within their normal specifications. The following instructions must be followed and integrated into your program when installing, using, and maintaining any of the products purchased from Fisher-Rosemount Petroleum.

- Read all instructions prior to installing, operating, and maintaining your meter. If this manual is not the manual you need, telephone (912) 489-0200, or the local Fisher-Rosemount Petroleum office, and the necessary manual will be mailed to you. Save this manual for future reference.
- If you do not understand the instructions, contact your sales representative for clarification.
- Follow all warnings, cautions, and instructions marked on and supplied with your meter.
- Inform and educate your personnel in the proper installation, operation, and maintenance of your meter.
- Install and maintain all related equipment as specified in the manual instructions and in accordance with local and national codes. Connect all products to the proper electrical and pressure sources.
- To ensure proper performance, use qualified personnel to install, operate, update, program, and maintain the equipment.
- When replacement parts are required, ensure that qualified service personnel use replacement parts specified by Fisher-Rosemount Petroleum. Unauthorized parts and procedures can affect the product's performance and endanger your operation. Look-alike substitutions may result in fire, electrical hazards, or improper operation.
- Except when maintenance is being performed by qualified persons, ensure all equipment doors are closed and protective covers are in place to prevent electrical shock and personal injury.

Warning

This instrument is designed for use in hazardous areas. Do not remove covers before first disconnecting all sources of electrical power. Failure to do so may cause an explosion resulting in serious personal injury or death.

Cautions

It is recommended that this bulletin be read in its entirety before performing any operation. Should this equipment require repair or adjustment, contact the nearest Fisher-Rosemount Petroleum Sales or Service office. It is important that servicing be performed only by trained and qualified service personnel.

This instrument contains electronic components that are susceptible to damage by static electricity. Proper handling procedures must be observed during the removal, installation or other handling of internal circuit boards or devices.

Handling Procedure

1. Power to the unit must be OFF.
2. Personnel must be grounded, via a wrist strap or other safe, suitable means before any printed circuit board or other internal device is installed, removed or adjusted.
3. Printed circuit cards must be transported in a conductive bag or other conductive container. Boards must not be removed from the protective enclosure until immediately before installation. Removed boards must immediately be placed in a protective container for transport, storage, or return to factory.

Comments

This instrument is not unique in its content of ESD (electrostatic discharge) sensitive components. Most modern electronic designs contain components that utilize metal oxide technology (NMOS, CMOS, etc.). Experience has proven that even small amounts of static electricity can damage or destroy these devices. Damaged components, even though they appear to function properly, may exhibit early failure. Care must be taken when handling these parts.

PetroCount[®] IMS Data Communications Interface

Introduction

In this section the means of communication between and with the PetroCount[®] IMS family of products is detailed. This method for transferring data is called the Brooks Protocol. To insure that there is a common frame of reference, there is included a detailed explanation of the following topics:

- Systems Configurations
- Interface Types
- System Wiring
- Serial Data Formats

Overview

Serial communications is one method used to transfer data and information between two or more electronic devices. One of these devices is usually referred to as the Master station. All other devices are referred to as Slave stations. The Master station initiates all data and information transfers, while Slave stations are only able to respond to requests from the Master.

A Master station and a Slave station communicate via a special language known as a protocol. The protocol defines the exact order of all data and information that is passed between the devices. The PetroCount[®] IMS communicates using a protocol called the BROOKS Protocol.

All data is transferred in an asynchronous serial format. The transmitting station transfers data one bit at a time with a specified time period between bits. The receiving station rebuilds the message by reversing the process. The communications terminology for binary 1 is Mark, and for a binary 0 is Space.

Systems Configurations

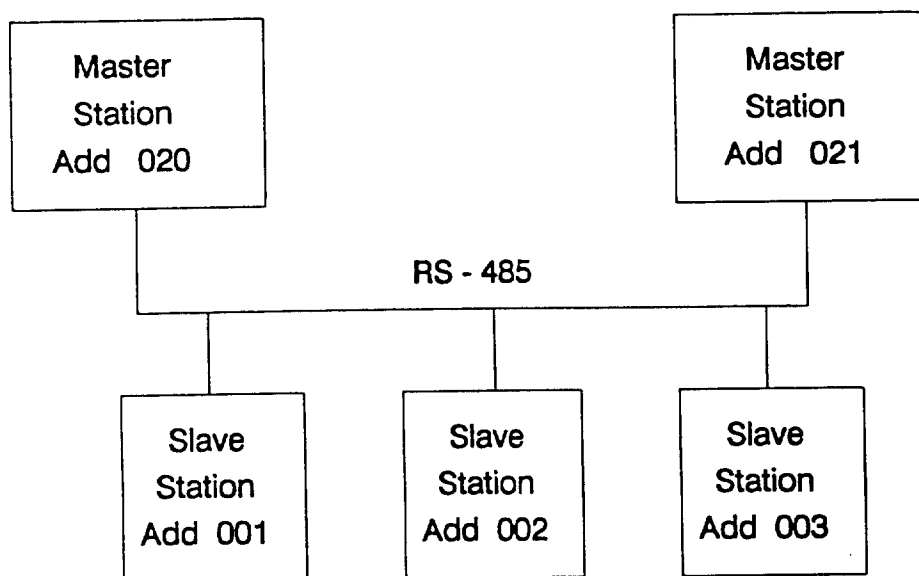
There are two main types of valid system configurations; multi-drop and dedicated. Multi-drop configurations permit a single Master station to communicate with as many as twenty-four Slave stations over a single communications line. Dedicated configurations permit a single Master station to communicate with only one Slave station over a single communications line. Variations of the multi-drop and dedicated configurations can be developed which are considerably more complex.

Multi-drop

The multi-drop configuration is used whenever the Master station must communicate with more than one Slave station. See Figure 5.1. To facilitate communications, the Master station and each Slave station are assigned a unique station address. The electrical interface and wiring details are covered in depth in the sections covering interface types and system wiring.

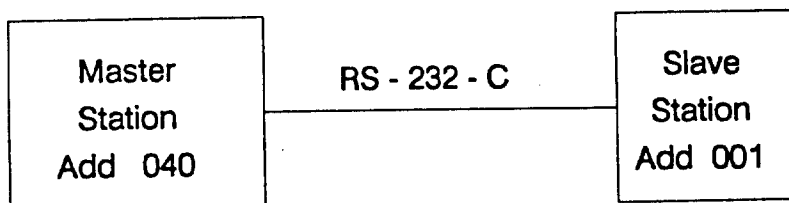
In a practical application of a multi-drop configuration, there are some limitations:

- The number of Slaves on a single multi-drop configuration is limited by the number of valid addresses in the protocol being used, and by the electrical interface.



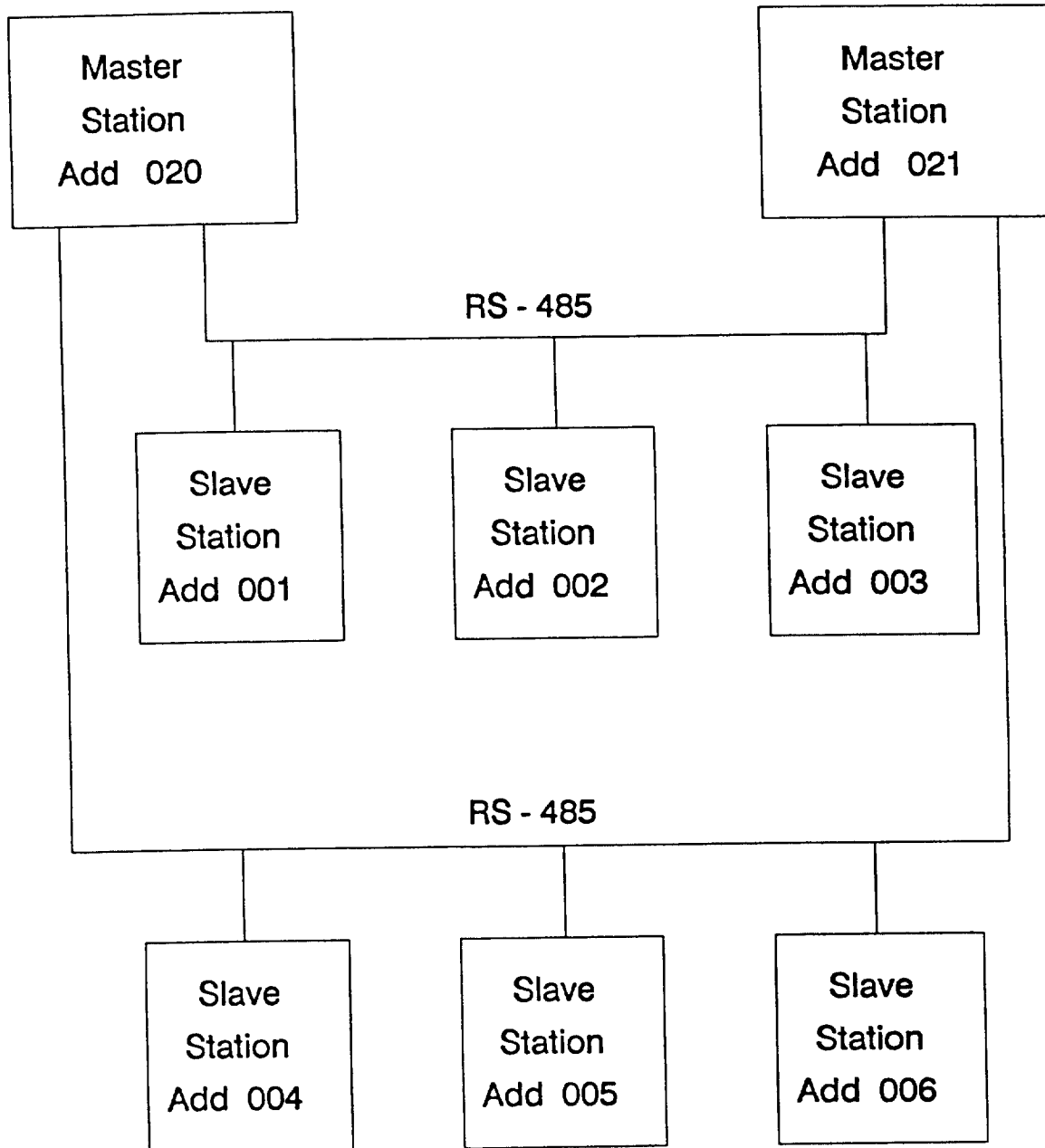
MULTI - DROP CONFIGURATION

FIGURE 5.1



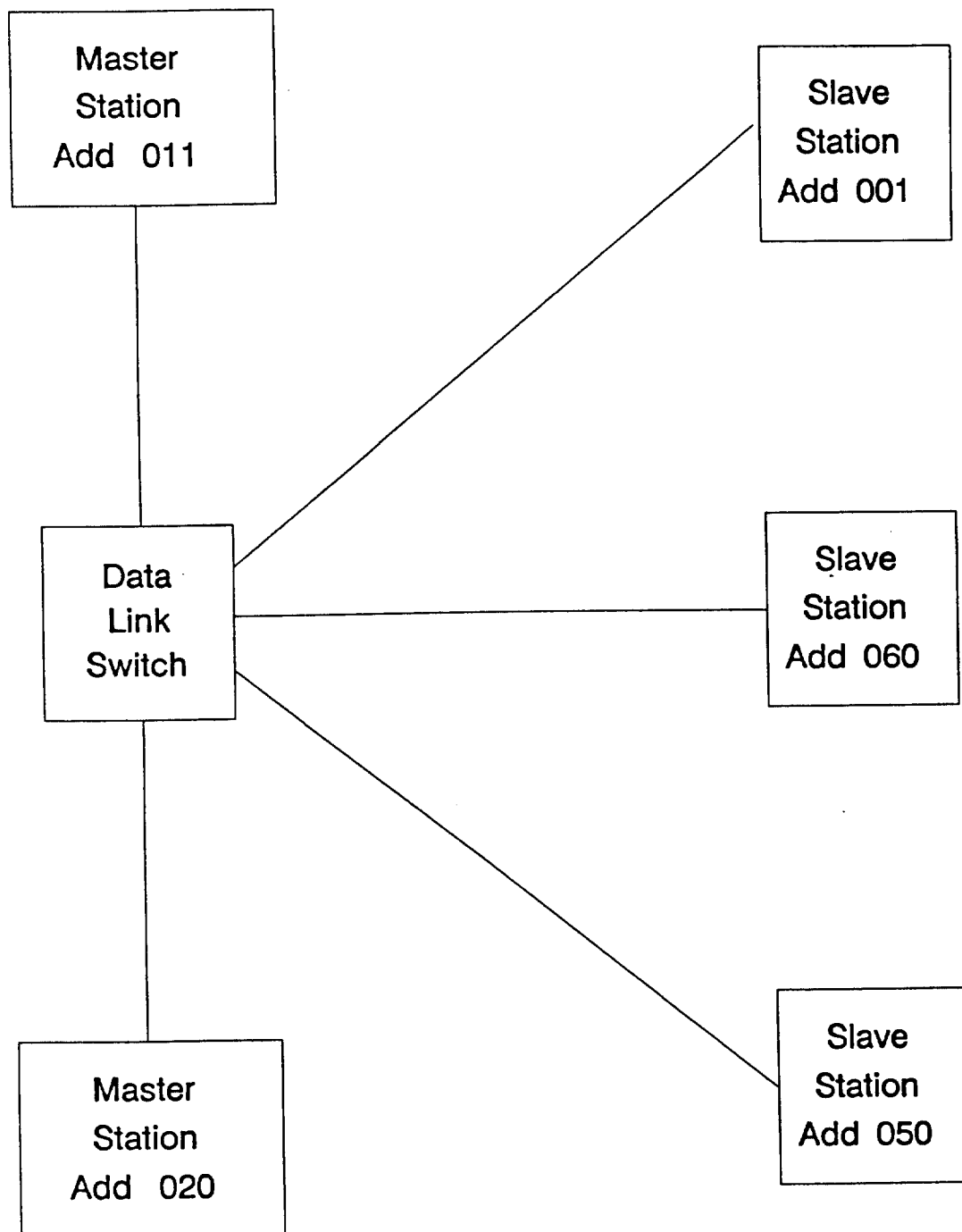
DEDICATED CONFIGURATION

FIGURE 5.2



COMPLEX MULTI - DROP CONFIGURATION

FIGURE 5.3



COMPLEX DEDICATED CONFIGURATION

FIGURE 5.4

Systems Configurations (cont'd)

Multi-drop (cont'd)

- The electrical interface used must have the ability for all line drivers to be placed in a tri-state or high impedance state. All stations must maintain their drivers in the tri-state condition except when that station is transmitting.
- While multiple potential Master stations are permitted in a multi-drop configuration, only one is permitted to be an operating Master station at a time. The others must respond as Slave stations until the system mastership is transferred.

Dedicated

A dedicated system configuration is used to connect a single Master station to a single Slave station. This configuration permits simpler protocols and simpler electrical interfaces due to the one to one relationship. See Figure 5.2

Complex Configurations

Variations of the multi-drop and dedicated configurations are used to form configurations which are more complex than those described previously. For example, a single computer may be used as the Master station on more than one multi-drop system. See Figure 5.3. Also, more than one Slave station can be connected to a single Master station in a dedicated configuration if an extra piece of equipment called a data link switch is used. See Figure 5.4. It is important to remember that the same rules described in the preceding sections must be adhered to, and careful attention paid to the specifications of the electrical interfaces used.

Interface Types

There are many different types of electrical interfaces being used in serial communication, but the following have become the leading choices:

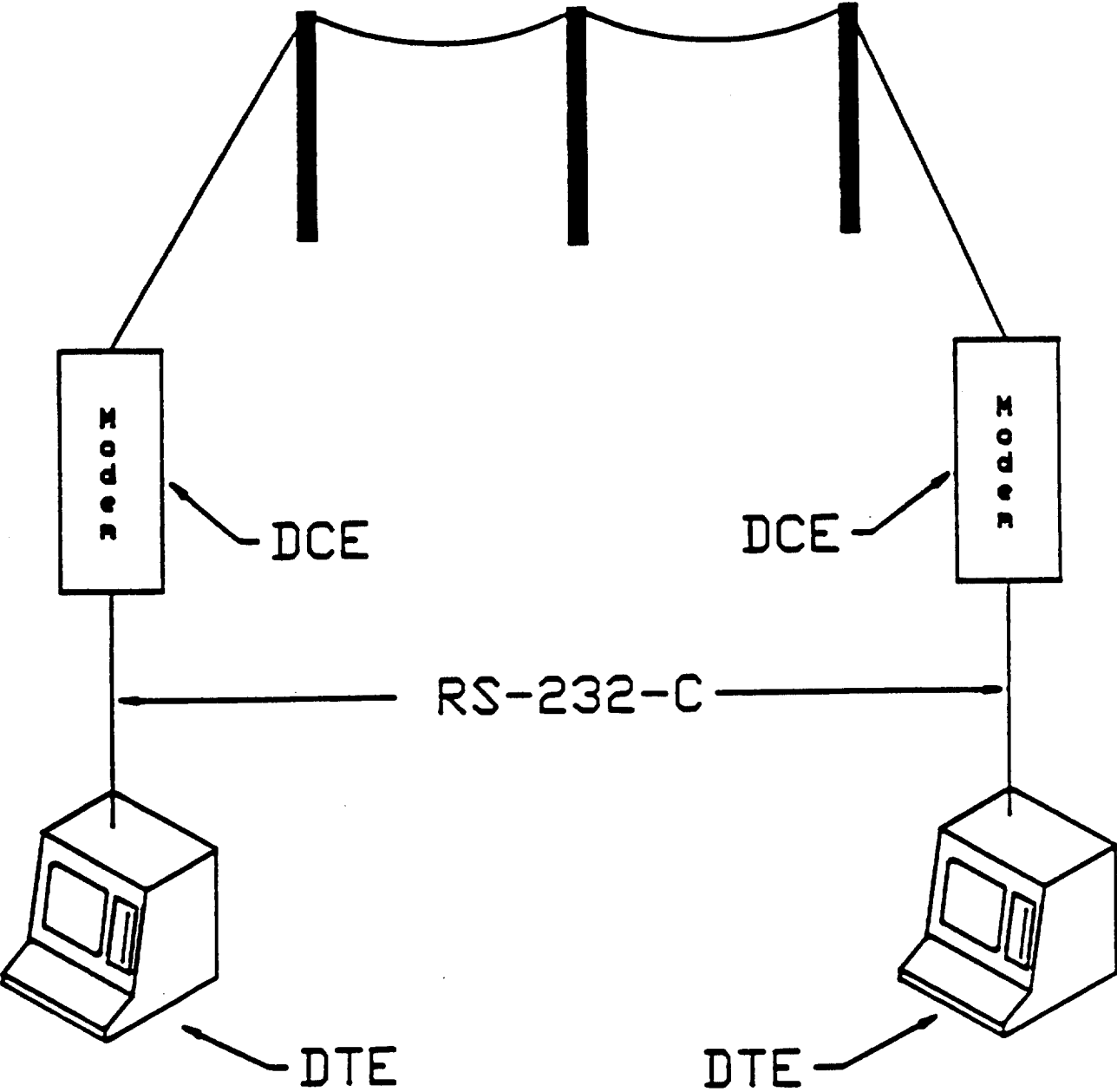
- RS-232-C
- RS-485

The choice of interface in any application is dependent on the details of the application. Some of the common factors are:

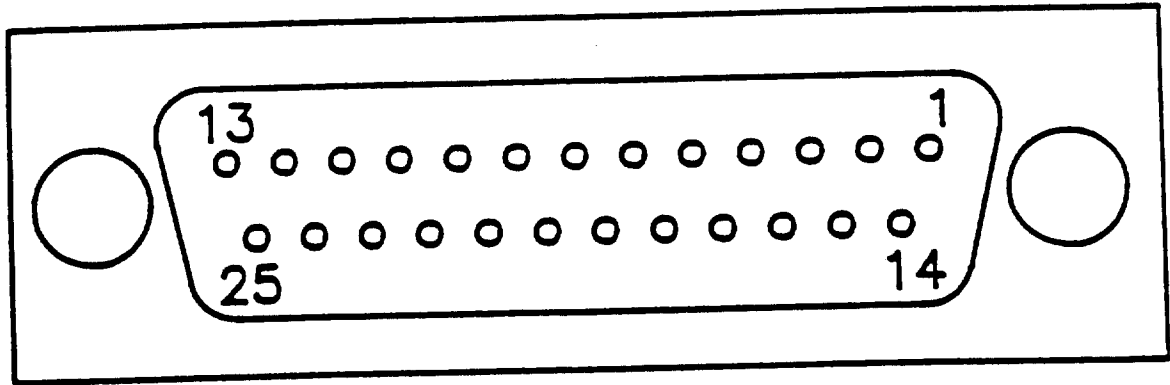
- Configurations type (Dedicated or Multi-drop)
- Distance and cable lengths between stations
- Interfaces available common to all equipment being used

RS-232-C

RS-232-C is the most common electrical interface in use today with serial communications configurations. While originally developed for interfacing between terminal and modem, it is now used to connect almost any type of equipment. The terminology used to describe many of its features reflects this origin.



CONVENTIONAL RS - 232 - C CONFIGURATION
FIGURE 5.5



| <u>Pin #</u> | <u>Signal</u> |
|--------------|---------------------|
| 1 | Protective Ground |
| 2 | Transmitted Data |
| 3 | Received Data |
| 4 | Request to Send |
| 5 | Clear to Send |
| 6 | Data Set Ready |
| 7 | Signal Ground |
| 8 | Carrier Detect |
| 20 | Data Terminal Ready |

RS - 232 - C PINOUT DEFINITION

FIGURE 5.6

Interface Types (cont'd)

RS-232-C (cont'd)

In the ideal RS-232-C interface, one device is defined as the Data Communication Equipment (DCE) and the other as the Data Terminal Equipment (DTE). See Figure 5.5. In this case a standard cable can be connected between the two devices which allows for properly connected signals.

With the many types of equipment using this interface today, the ideal interface is a rarity. Careful analysis of the signal requirements and connector pinouts of both devices is required to determine the cabling requirements.

Signals

The Electronic Industries Association (EIA) RS-232-C specification defines 20 different signals, but rarely are more than 7 used. The definitions of those 7 signals are given below:

| | |
|---------------------|---|
| Received Data | - Data that is received by the DTE and transmitted by DCE. |
| Transmitted Data | - Data that is transmitted by the DTE and received by the DCE. |
| Request to Send | - Originated by the DTE and is used to notify the DCE that the DTE wants to transmit data. |
| Clear to Send | - Originated by the DCE and is used as a response to a Request to Send from the DTE. |
| Data Terminal Ready | - Originated by the DTE and is used as an indication that the DTE is ready to receive data. |
| Signal Ground | - A logic ground so that the electrical signals have a common reference. |
| Protective Ground | - Earth ground connection and generally connected to the cable shield. |

The type of connector and pinouts of the signals are rigidly defined by the EIA RS-232-C specification. The type of connector used is a sub-miniature "D" connector with 25 pins. See Figure 5.6.

For permanently mounted instruments that are installed using conduit, it is more practical to install the communications signals at terminal block positions. Consult the corresponding PetroCount® IMS product user's manual for detailed hook-up instructions.

The electrical interface is a single ended bi-polar signal. The minimum voltage permitted by a driver is ± 3 Volts, however, most drivers use ± 12 Volts. A Mark (Logic 1) is defined to be the negative voltage and Space (Logic 0) is defined to be a positive voltage.

Interface Types (cont'd)

RS-485

The RS-485 type interface is used when long cable runs and multi-drop configurations are required. Differential type signals and line drivers with tri-state outputs are utilized to provide these features.

Only the electrical signals are specified by the EIA RS-485 specification. The electrical signals must be of differential type and the magnitude of output voltage at the transmitter, measured across a 100 ohm load, may not exceed 6 volts, nor be less than 2 volts. The receiver requires a minimum of 200 millivolts between the differential inputs. See Figure 5.7.

A mark is defined to be the conditions where the non-inverted output of the differential is negative with respect to the inverted output. Conversely, a space is defined to be the condition where the non-inverted output of the differential driver is positive with respect to the inverted output.

All Brooks equipment that supports the RS-485 type interface and uses a sub-miniature "D" connector with 25 pins, uses the following pinout:

| <u>Pin Number</u> | <u>Signal</u> | |
|-------------------|--------------------|--------------------|
| | <u>4 Wire</u> | <u>2 Wire</u> |
| 1 | Logic Gnd | Logic Gnd |
| 2 | Tx(A) | Tx/Rx(A) |
| 3 | Rx(A) | Not Used |
| 7 | Logic Gnd | Logic Gnd |
| 14 | Tx(B) | Tx/Rx(B) |
| 16 | Rx(B) | Not Used |
| 25 | 100 ohms to Gnd | 100 ohms to gnd |

System Wiring

Proper system wiring is critical to the reliable operation of all types of serial communications interfaces. Improper system wiring can cause high data error rates and therefore reduce data throughput on a communications link. While wiring requirements vary depending on the type of interface used, each of the following items is important to the overall operation of the communications system:

- Cable lengths and types
- Shielding
- Twisted Pair Wiring
- Cabling Stubs

You should consult the user's manual for each product for the precise wiring connection points.

RS-232 Wiring

The wiring in a system with an RS-232-C interface is relatively simple due to the limitation to dedicated system configurations.

The cable used must be shielded 24 AWG stranded, without twisted pairs, and insulation which meets UL style 2464. Recommended cable capacitance is 30 picofarads per foot.

System Wiring (cont'd)

RS-232 Wiring (cont'd)

The maximum recommended cable length is 50 feet. However, longer cable lengths are permissible at data rates of 1200 baud or less. Although it is not recommended, some installations have been successful with cable lengths up to 1000 feet.

RS-485 Wiring

Since RS-485 interfaces are normally used in multi-drop configurations, system wiring can become very complex. The system wiring can be performed in a 2 wire or 4 wire mode. In the 2 wire mode, receive and transmit share the same conductor pair within the cable. See Figure 5.7. In 4 wire mode, receive and transmit have separate conductor pairs. See Figure 5.8. In either case, the pairs **MUST** be a twisted pair due to the differential signal type provided by RS-485.

Wiring must be performed in a daisy chain manner. Cable stubs from the daisy chain are permitted, but must be limited to 15 feet. See Figure 5.9.

In order to minimize data errors, each conductor pair must be terminated with a 100 ohm resistor at each extreme end of the daisy chain as shown in Figure 5.7. This provides the proper line impedances which results in better signal reception.

The cable used must be shielded 24 AWG stranded, with twisted pairs, and insulation which meets UL style 2464. Recommended cable capacitance is 20 picofarads per foot. Using the recommended cable, a RS-485 interface can support up to 24 devices over a wire length of a maximum of 3600 feet.

Serial Data Formats

Serial Data is transmitted, asynchronously, one bit at a time with a specified time between data bits. The time between the data bits is known as the data rate. Most computers or instrumentation that receives and transmits the serial data prefer to work with more than one bit at a time. Therefore, they group bits together into frames. Each frame consists of synchronizing bits, data bits, and error detection bits.

Data Rates

The data rate is the speed at which the data is being sent serially and is measured in bits per second (baud). The most commonly used data rates are 300, 600, 1200, 2400, 4800 and 9600 baud.

Frames

Each frame includes the following bits: See Figure 5.10.

- 1 Start bit
- 7 or 8 data bits
- 0 or 1 parity bit
- 1 or 2 stops bits

The Start Bit is used to allow the receiving device to synchronize the bit timing at the beginning of each frame. The start bit is always a space.

Serial Data Formats (cont'd)

Frames (cont'd)

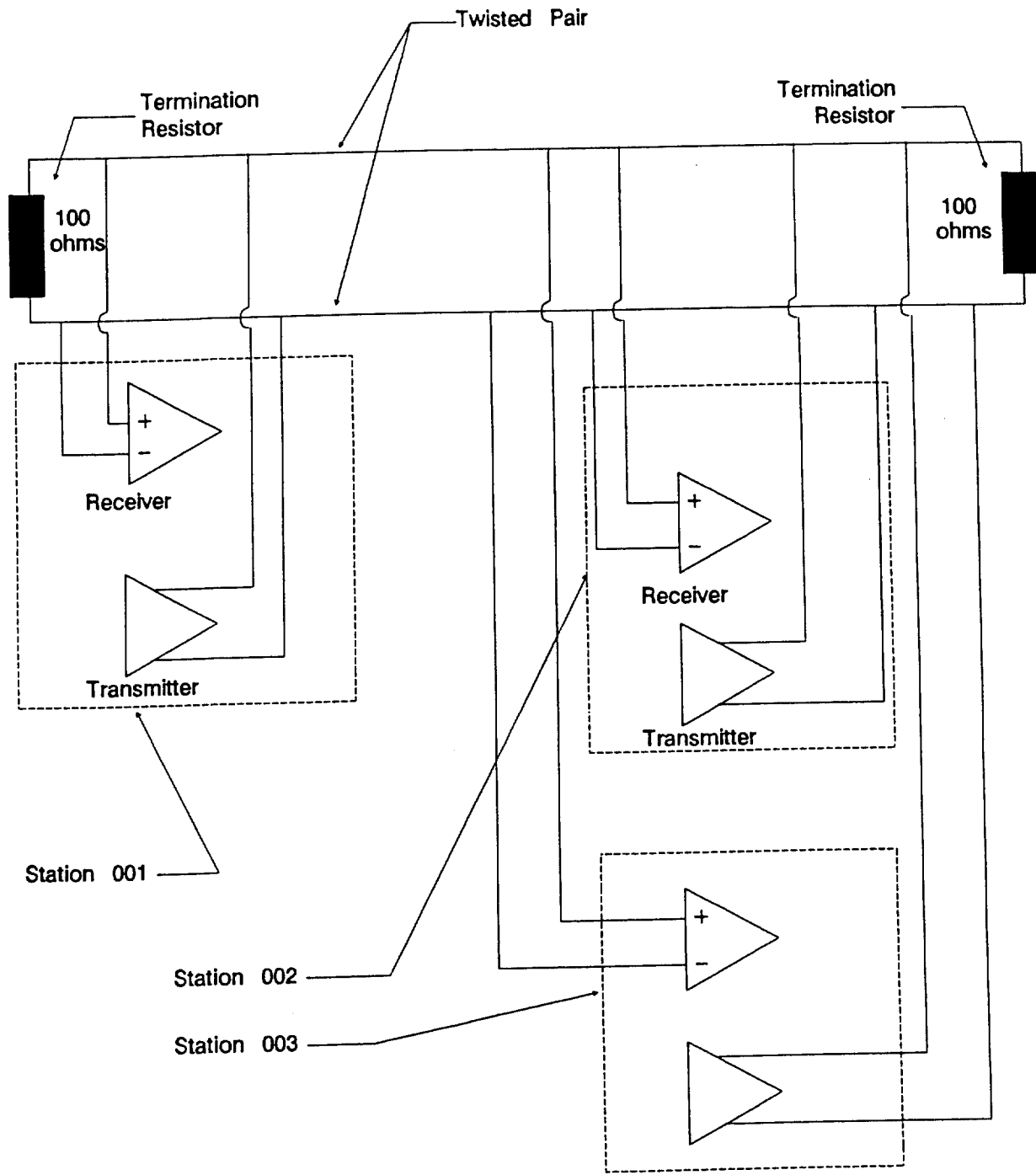
The Data Bits can be in either binary or ASCII format. Binary format is normally 8 bits and represents any code from 0 to 255.

ASCII is a 7 bit encoding system which provides special control characters as well as most characters found on the average typewriter. Figure 5.11 is a chart of all the available ASCII characters and their corresponding codes. Codes 00 - 1F (hex) represent control characters which are used to provide special functions to be discussed later. These codes are also known as unprintable characters because most computer terminals do not display them. Codes 20 - 7F are the printable characters and include the entire upper and lower case alphabet, the numbers 0 to 9, and other punctuation type characters.

Parity is a method used to detect data errors. Two techniques can be used, odd parity and even parity. If odd parity is used, the parity bit is set such that the number of data bits and the parity bit which are ones, is an odd number. If even parity is used, the parity bit is set such that the number of data bits and the parity bit which are ones, is an even number. The parity bit is optional. If it is not used, each frame is one bit shorter.

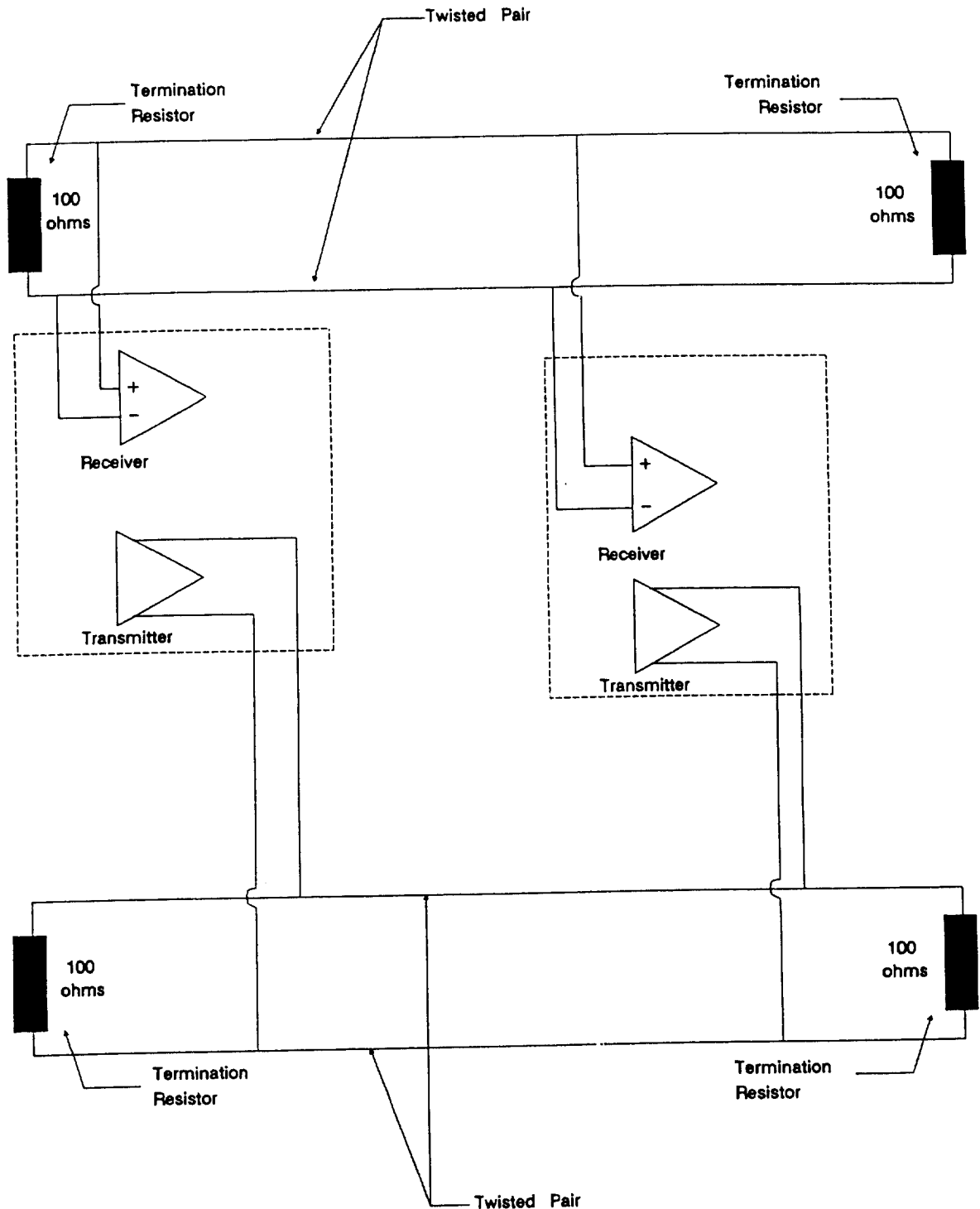
The Stop Bit is used to guarantee the one to zero transition that occurs at the start bit. Thus, the Stop Bit is always a Mark.

For successful data communications to take place, the number of START bits, data bits, parity bits and STOP bits must be the same for both transmitter and receiver. Likewise, the transmission data rate (Baud) must be the same on both ends.



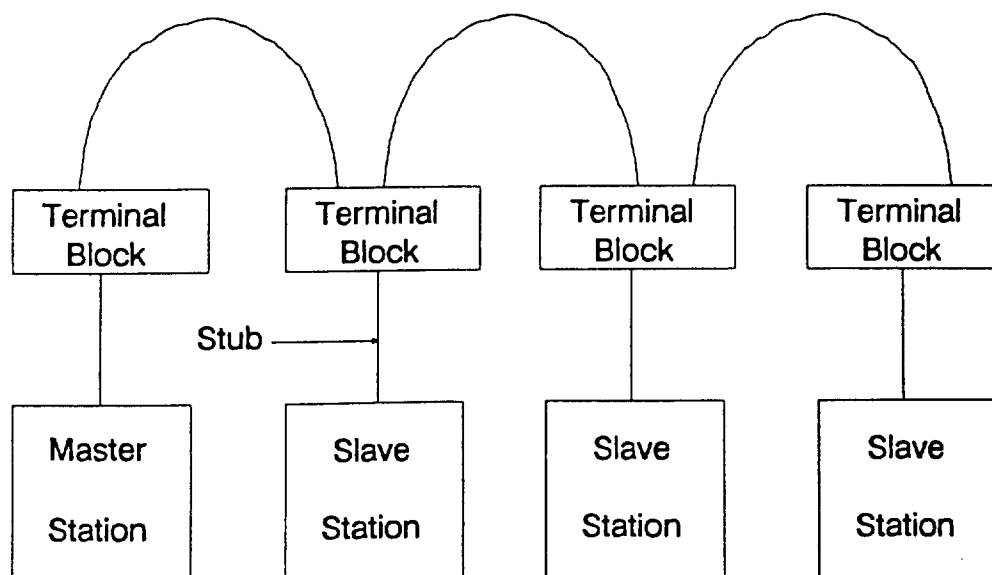
TYPICAL TWO - WIRE RS - 485 INTERFACE SCHEMATIC

FIGURE 5.7

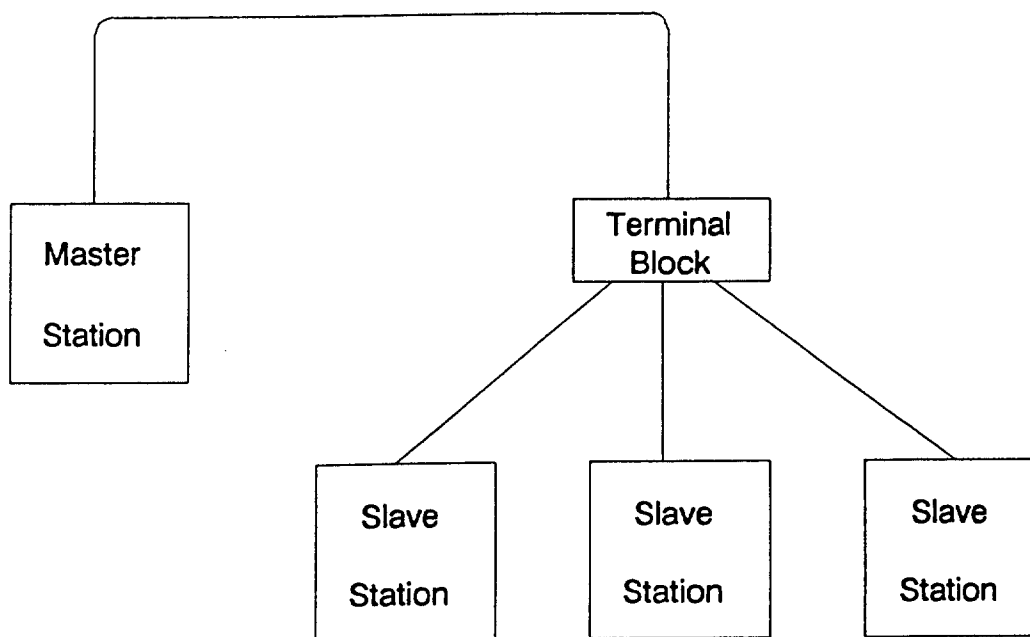


TYPICAL FOUR - WIRE RS - 485 INTERFACE SCHEMATIC

FIGURE 5.8



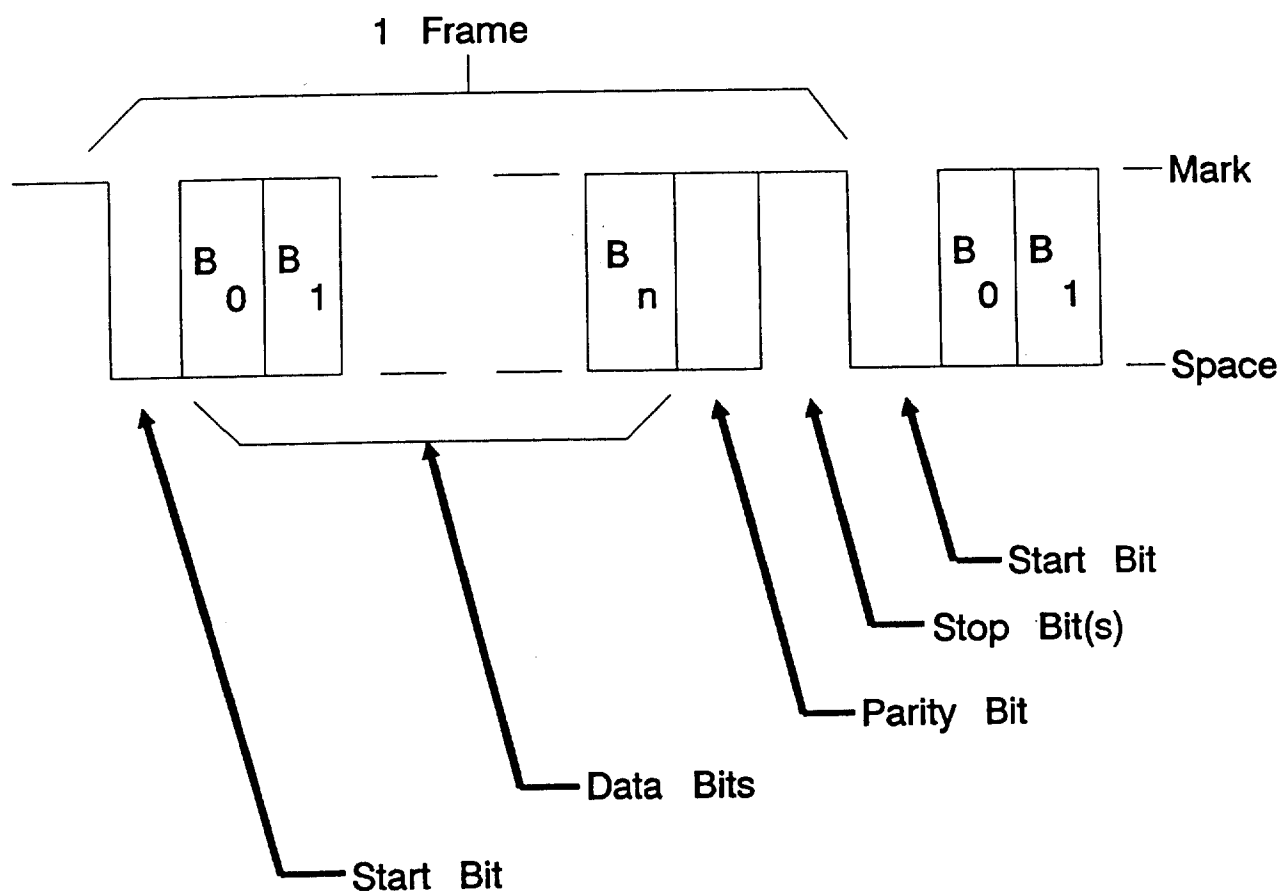
Right



Wrong

DAISY CHAIN WIRING - RS - 485

FIGURE 5..9



SERIAL FRAME

FIGURE 5.10

American National Standard Code for Information Interchange

| Bits | | | | | 0 0 0 | 0 0 1 | 0 1 0 | 0 1 1 | 1 0 0 | 1 0 1 | 1 1 0 | 1 1 1 |
|----------------|----------------|----------------|----------------|-----|-------|-------|-------|-------|-------|-------|-------|-------|
| b ₄ | b ₃ | b ₂ | b ₁ | ROW | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 0 | 0 | 0 | 0 | 0 | NUL | DLE | SP | 0 | • | P | \ | p |
| 0 | 0 | 0 | 1 | 1 | SOH | DC1 | ! | 1 | A | Q | a | q |
| 0 | 0 | 1 | 0 | 2 | STX | DC2 | " | 2 | B | R | b | r |
| 0 | 0 | 1 | 1 | 3 | ETX | DC3 | # | 3 | C | S | c | s |
| 0 | 1 | 0 | 0 | 4 | EOT | DC4 | \$ | 4 | D | T | d | t |
| 0 | 1 | 0 | 1 | 5 | ENQ | NAK | % | 5 | E | U | e | u |
| 0 | 1 | 1 | 0 | 6 | ACK | SYN | & | 6 | F | V | f | v |
| 0 | 1 | 1 | 1 | 7 | BEL | ETB | / | 7 | G | W | g | w |
| 1 | 0 | 0 | 0 | 8 | BS | CAN | (| 8 | H | X | h | x |
| 1 | 0 | 0 | 1 | 9 | HT | EM |) | 9 | I | Y | i | y |
| 1 | 0 | 1 | 0 | 10 | LF | SUB | * | : | J | Z | j | z |
| 1 | 0 | 1 | 1 | 11 | VT | ESC | + | ; | K | [| k | { |
| 1 | 1 | 0 | 0 | 12 | FF | FS | , | < | L | \ | l | |
| 1 | 1 | 0 | 1 | 13 | CR | GS | - | = | M |] | m | } |
| 1 | 1 | 1 | 0 | 14 | SO | RS | . | > | N | ^ | n | ~ |
| 1 | 1 | 1 | 1 | 15 | SI | US | / | ? | O | — | o | DEL |

FIGURE 5.11

The Brooks Protocol

In order for two devices to communicate, they must be able to understand a common language. A language used for communication between devices is called a protocol. Unlike a human language, a protocol requires very exact grammar. The protocol defines the grammar and the order of all frames that are passed between devices. A protocol has been defined that is used by all Brooks products for communications. This protocol is known as the Brooks Protocol. The Brooks Protocol defines the message structure as well as the commands that can be used to pass information between the devices.

The Brooks Protocol requires that each master station and each slave station be assigned a unique station address. In order to send information to a slave station, the master station must send a message in the format defined in the section entitled Message Structure. All slave stations receive the message. The slave station whose address is specified in the message responds in one of four ways:

- Return data if required
- Return an acknowledge
- Return a negative acknowledge
- No response

The type of response depends on the message received. Some messages require data to be returned by the slave station. Other messages only require an acknowledgement of successful receipt of the message. One special type of message requires the slave station not respond to the message. If the slave station detects an error in the message, a negative acknowledge is returned.

Two modes of operation are supported in the Brooks Protocol:

- Computer Mode
- Terminal Mode

The Computer Mode has added error detection capabilities while the Terminal Mode simplifies data transmission directly from a dumb terminal.

Message Structure

The message structure in the Brooks Protocol is dependent on the mode of operation. (See Figures 5.12 and 5.13)

Computer Mode

The message structure in the computer Mode consists of three sections:

- Heading
- Text
- Block check characters

The heading is made up of the Start of Header (SOH) character, the source address, and the destination address. The SOH is an ASCII character with a value of 01 which tells the receiving station that the characters to follow are the destination and source addresses. The source address is the station address of the station transmitting the message. It must be three ASCII characters long and in the range of 001 to 999. The destination address is the address of the station to which the message is being sent. The destination address must be three ASCII characters long and in the range of 001 to 999.

The Brooks Protocol (cont'd)

Computer Mode (cont'd)

(Note: Address 000 is reserved for Broadcast messages and cannot be assigned to a master or slave station.)

The text section of the message consists of Start of Text (STX) character, the information to be passed to the destination station, and the End of Text (ETX) character. The STX is an ASCII character with a value of 02 which tells the destination station that the characters to follow are the text of the message. The text of the message must be in ASCII type. The ETX is an ASCII character with a value of 03 that tells the destination station that the end of the text portion of the message has been reached.

In computer mode, the Block Check Characters (BCC) always follow the ETX character. These characters are used by the destination station to verify the accuracy of the data received. These two characters are the ASCII representation of the binary sum of all the data in the message from the SOH character to the ETX character, inclusive. In simpler terms, all the data starting at SOH and ending at, and including ETX are added together in hexadecimal format. The resulting two hexadecimal digits are then translated to their ASCII representation. These two ASCII characters are the Block Check Characters. See Figure 5.14. The transmitting station calculates the Block Check Characters for each and transmits them following the ETX. The receiving station then calculates the Block Check Characters for each message it receives and then compares the calculated BCC's to the received BCC's. If they are different, a data error has occurred.

Terminal Mode

The message structure in the terminal mode consists of two sections:

- Heading
- Text

The only difference in the message structure between the two modes is that terminal mode does not include the Block Check Characters. The format of both the heading and the text sections are identical to the message structure used in the computer mode. However, in order to make direct keyboard entry of messages easier, the control characters used for SOH, STX, AND ETX are different. The following table shows the ASCII characters that must be used in place of the control characters in terminal mode.

| | | |
|-----|---|---------------------|
| SOH | ! | (exclamation point) |
| STX | " | (quotation mark) |
| ETX | | (carriage return) |

Message Type

The text portion of the message may contain the following message types:

- Read character data
- Write character data with return data
- Write character data with return acknowledge
- Write character data with no response
- Write character data upon confirmation
- Confirm write character data
- Execute a task

| | | | | | | | |
|-------------|-----------------|-------------------|-------------|-----------------|-------------|------------------|------------------|
| S O H | Dest Address | Source Address | S T X | Message Text | E T X | B C C 1 | B C C 2 |
|-------------|-----------------|-------------------|-------------|-----------------|-------------|------------------|------------------|

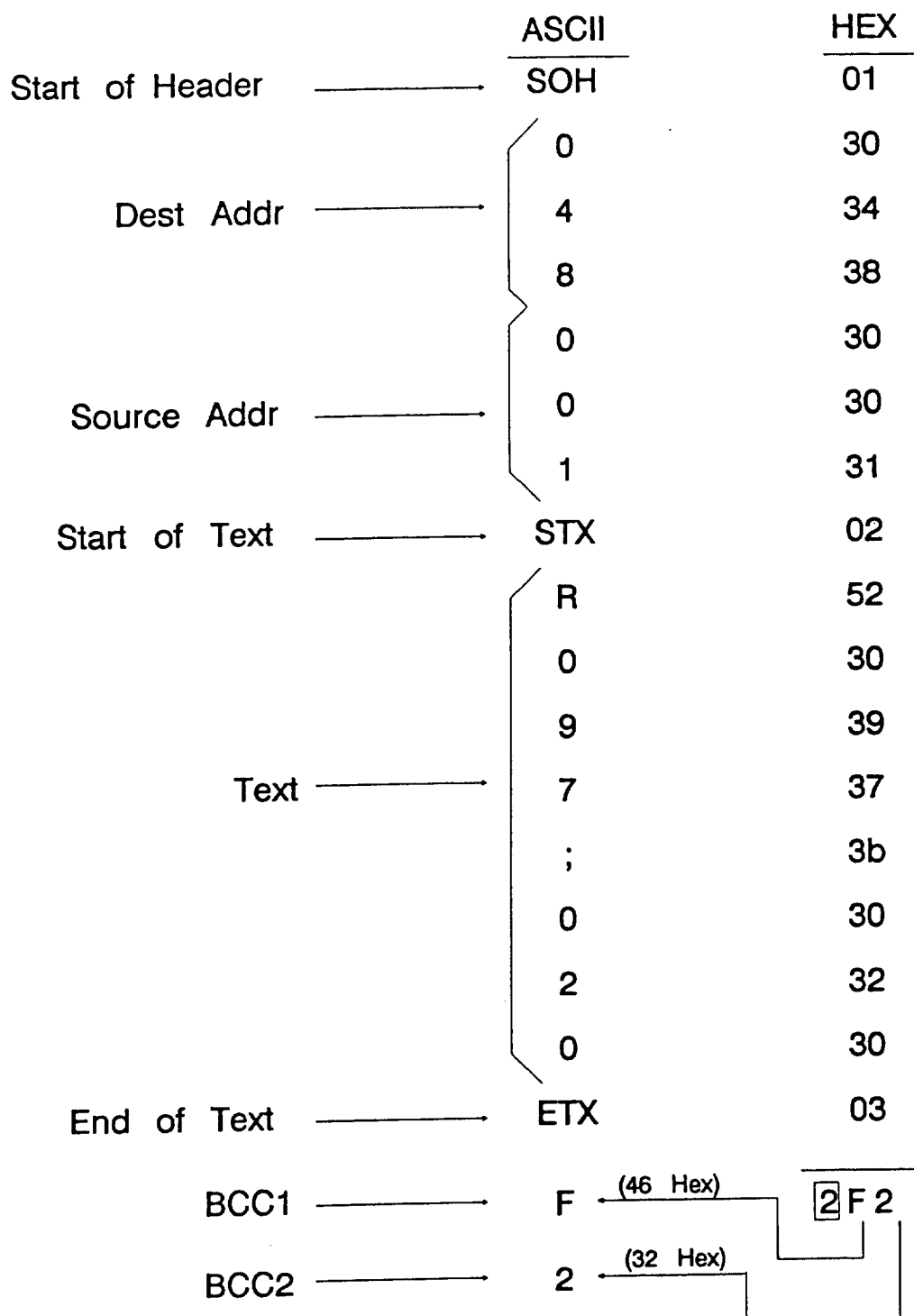
MESSAGE STRUCTURE - COMPUTER MODE

FIGURE 5.12

| | | | | | |
|---|-----------------|-------------------|---|-----------------|----------------|
| ! | Dest Address | Source Address | „ | Message Text | C _R |
|---|-----------------|-------------------|---|-----------------|----------------|

MESSAGE STRUCTURE - TERMINAL MODE

FIGURE 5.13



BCC CALCULATION

FIGURE 5.14

The Brooks Protocol (cont'd)

Message Types (cont'd)

Read Character Data

The read character message type is used to inspect or interrogate a particular parameter value within the receiving station. This message type is identified by the ASCII code for the symbol R (52 hex).

Write Character Data with Return Data

The write character data with return data message type is used to store or change a parameter value in the destination station. If no error is detected in the received message, the destination station stores the value received in the specified parameter. The text portion of the message is returned exactly as received to acknowledge receipt of the message. This message type is identified by the ASCII code for the symbol W (57 hex).

Write Character Data With Return Acknowledge

The write character data with return acknowledge message type is also used to store or change a parameter value in the destination station. If no error is detected in the received message, the destination station stores the value received in the specified parameter. The destination station replies to the master station with an acknowledge. This message type is identified by the ASCII code for the symbol A (41 hex).

Write Character Data with No Response

The write character data with no response message type is also used to store or change a parameter value in the destination station. If no error is detected in the received message, the destination station stores the value received in the specified parameter. The destination station does not reply to the master station. This message type is identified by the ASCII code for the symbol B (42 hex). Note, this message type may only be used in a broadcast message.

Write character Data Upon Confirmation

The write character data upon confirmation message type is also used to store or change a parameter value in the destination station. If no error is detected in the received message, the destination station echos the received message text back to the master station, but does not change the specified parameter at this time. A confirmation message type must be received before any parameter values are received. The write character data upon confirmation message type is identified by the ASCII code for the symbol C (43 hex).

Confirm Write Character Data

The confirm write character data is used to confirm a write character data upon confirmation message. The master station transmits this message after receiving the destination station's reply to the write character data upon confirmation message. The destination station stores or changes the parameters specified in the previous write character data upon confirmation message only after receiving an error free confirm write character data message. The destination station must then reply to the master station with

The Brooks Protocol (cont'd)

Message Types (cont'd)

an ACK. The confirm write character data message type is identified by the ASCII code for the symbol K (48 hex).

Broadcast Messages

Broadcast messages are used to send common information to all slave stations with one transmission. For example, if the slave stations all have a parameter for the time of day, that parameter could be set in all the slave stations with one transmission by using the broadcast message type. A broadcast message is performed by sending a write character data message type with a destination station address of 000. All slave stations receive and process the message, but none of the slave stations reply.

Execute Tasks

The Execute Tasks type of message is used to cause the PetroCount® product to perform a function like START, STOP or RESET. This message type is identified by the ASCII code for the symbol X (58 hex).

Summary of Message Types and Identifiers

| Identifier | Message Type |
|------------|--|
| R (52 hex) | Read character data |
| W (57 hex) | Write character data with return data |
| A (41 hex) | Write character data with return acknowledge |
| B (42 hex) | Write character data with no response |
| C (43 hex) | Write character data upon confirmation |
| K (48 hex) | Confirm write character data |
| X (58 hex) | Execute a task |

Text Formats

The general format to perform a read and write is show below:

R054

W077=23.6

For a read message type, the message type symbol must be used followed by a three digit parameter number. For a write message type, the message type symbol must be followed by a three digit parameter number, then an '=', and followed by the new parameter value.

More than one parameter may be read or written with read and write character data message types within one message. The parameter numbers must be separated by semicolons and the message type symbol must not be repeated. Only one message type is allowed per message. The following are examples of text formats with more than one parameter.

The Brooks Protocol (cont'd)

Text Formats (cont'd)

R001;002;030;040

W033=600;035=5000

A060=5;020=1

Special Message Responses

In some instances, the destination station does not reply with the expected response. Whenever the destination station detects an error in the message or for any reason cannot perform the specified action, the destination replies with a special message response.

The following conditions cause the destination station to reply with a negative acknowledge. In the computer mode, a negative acknowledge is represented by the ASCII code for NAK (15 hex). In the terminal mode, a negative acknowledge is represented by the ASCII code for the symbol N (4E hex).

- If any parameter specified by the master station does not exist within the destination station.
- If an error is detected in the message structure.
- If a value specified to be written to a parameter is not in the allowable range for that parameter.

If after receiving a read message type, the destination station determines that any of the specified parameters have invalid values with the destination station, the values for those parameters are replaced by asterisks.

If the BCC is incorrect, the slave does not respond to the message.

Summary of Computer and Terminal Modes

Two modes of operation are available in the Brooks Protocol, computer and Terminal. Computer mode is intended to be used when a computer acts as the master station. Increased data security is provided and standard ASCII control characters are used. Terminal mode is intended to be used when an operator at a terminal acts as the master station. Block Check Characters have been omitted and normal keyboard characters are used as control characters in order to simplify keyboard entry of messages.

The following table summarizes the control character usage in both computer and terminal mode.

| Control Character | Computer | Terminal |
|-------------------|----------|-----------------|
| Start of Header | SOH | ! |
| Start of Text | STX | " |
| End of Text | ETX | Carriage Return |
| Acknowledge | ACK | Y |
| Neg. Acknowledge | NAK | N |

In either mode the maximum message length is 256 bytes.

The Brooks Protocol (cont'd)

Message Structure Examples

In order to tie together the information presented on the Brooks Protocol, this section shows examples of typical messages and replies between master and slave stations.

The first example shows a typical read message type sent from a master station with an address of 001 to a destination station with an address of 045. Both stations are in the computer mode.

Request

| | | | | | |
|---|--------|---|------|---|---|
| S | | S | | E | B |
| O | 045001 | T | R033 | T | C |
| H | | X | | X | C |

Reply

| | | | | | |
|---|--------|---|-----------------|---|---|
| S | | S | | E | B |
| O | 001045 | T | 033=00000009999 | T | C |
| H | | X | | X | C |

If both devices are in the terminal mode, the message would appear as shown below:

Request

!045001"R033<CR>

Reply

!001045"033=00000009999<CR>

The Brooks Protocol (cont'd)

Message Structure Examples (cont'd)

The second example shows a typical write message type sent from a master station with an address of 789 to a destination station with an address of 333. Both stations are in computer mode.

Request

| | | | | | |
|---|--------|---|------------|---|---|
| S | | S | | E | B |
| O | 333789 | T | W008=040.0 | T | C |
| H | | X | | X | C |

Reply

| | | | | | |
|---|--------|---|------------|---|---|
| S | | S | | E | B |
| O | 789333 | T | W008=040.0 | T | C |
| H | | X | | X | C |

If both devices are in the terminal mode, the message would appear as show below:

Request

!333789"W008=040.0<CR>

Reply

!789333"W008=040.0<CR>

The Brooks Protocol (cont'd)

Message Structure Examples (cont'd)

The third example shows the same write message using the "A" type command in computer mode.

Request

| | | | | | |
|---|--------|---|------------|---|---|
| S | | S | | E | B |
| O | 333789 | T | A008=040.0 | T | C |
| H | | X | | X | C |

Reply

| | | | | | |
|---|--------|---|---|---|---|
| S | | S | A | E | B |
| O | 789333 | T | C | T | C |
| H | | X | K | X | C |

The fourth example shows the same write message using the "C" type command in computer mode.

Request

| | | | | | |
|---|--------|---|------------|---|---|
| S | | S | | E | B |
| O | 333789 | T | C008=040.0 | T | C |
| H | | X | | X | C |

Reply

| | | | | | |
|---|--------|---|------------|---|---|
| S | | S | | E | B |
| O | 789333 | T | C008=040.0 | T | C |
| H | | X | | X | C |

The Brooks Protocol (cont'd)

Message Structure Examples (cont'd)

The value in parameter 008 has not been changed. The value is changed only if the next message is the confirm (K) message.

Request

| | | | | | |
|---|--------|--|---|---|---|
| S | | | S | E | B |
| O | 333789 | | T | K | T |
| H | | | X | X | C |

Reply

| | | | | | | |
|---|--------|--|---|---|---|---|
| S | | | S | A | E | B |
| O | 789333 | | T | C | T | C |
| H | | | X | K | X | C |

Data Communications Tasks

Both the Control Unit and the Remote Access Unit have a variety of executable task functions which are available through the data communications interface. These executable tasks are not available through the operator interface and are in addition to the programmable parameters.

The executable task functions are selected with an X (58 hex) message type identifier. The general format to perform an executable task is:

X036

Each task is identified with a 3 digit code. The message response rules are the same as for the read and write message types. Only one task can be requested per message. No multiple task messages are permitted.

The executable task functions have been provided to give increased control capability throughout the data communications interface. Functions such as stopping, starting, resetting and display control are provided. The available tasks for each IMS product are described below.

The Brooks Protocol (cont'd)

Control Unit Tasks

There are nineteen executable tasks for the Control Unit. These are listed below with an explanation for each and with the three digit identifying code.

1. Display Lock (021 Line 1)
(022 Line 2)

The Display Lock is used to provide total control of the front panel alphanumeric display to the remote computer. The alphanumeric display is blanked when the Display Lock is enabled. The Display String task (023) can be used to present messages on the display. The Control Unit electronics cannot present messages on the display while the Display Lock is on. The Display Lock is removed by the Display Unlock task (025).

2. Display String (023)

The Display String task is used to present text on the front panel alphanumeric display. A total of 32 characters can be displayed per screen (2 x 16). The text of the message automatically wraps around from the first to the second line if there is more than 16 characters in the message. The message format is as follows:

```
SOH DEST SOURCE STX X023
ABCDEFGHIJKLMN OPQRSTUVWXYZ123456
ETX CS1 CS2
```

The text from this message would be displayed on the screen as follows:

```
ABCDEFGHIHJKLMN OP
QRSTUVWXYZ123456
```

There are various display control codes supported such as:

*ESC"C = Clear Display

*ESC"F = One line of text flashes

*ESC"1 = Indicates message to be output to Line 1

*ESC"2 = Indicates message to be output to Line 2

In the last two display control codes a maximum of 16 characters can be displayed. Any further text is ignored. The ASCII code for "ESC" is 1B hex.

The display screen is overwritten by normal display messages unless the Display Lock is enabled.

The Brooks Protocol (cont'd)

Control Unit Tasks (cont'd)

3. Display Unlock (025)

The display unlock is used to release control of the alphanumeric front panel back to the Control Unit electronics. If a display lock was not enabled, this task has no effect.

4. Start (036)

This Start task is used to begin a batch sequence. It is equivalent to pushing the front panel "Start" pushbutton. A preset quantity and blend selection (if enabled) must be entered prior to the request for a start. A NAK is returned if the Control Unit is not in operator mode.

5. Abort Blend Affirmative Acknowledge (037)

This task is used to instruct the Control Unit to continue with a requested blend abort. Use this task only when the Control Status is 18 or 25.

6. Abort Blend Negative Acknowledge (038)

This task is used to instruct the Control Unit to cancel a blend abort and therefore to continue with the blend sequence. A start task must follow this task before the blend is actually restarted. Use this task only when the Control Status is 28 or 25.

7. Keypad Unlock (070)

The Keypad Unlock task is used to enable the front panel pushbuttons after any Keypad Lock task (071 - 078) has been executed. Execution of this task without a keypad lock enabled has no effect.

8. Keypad Remote: Start (071)

This task is used when the remote computer starts the batch via the Start command (036). The START pushbutton has no affect when on the "Start" screen. Normal preset value entry and blend recipe selection by the operator are permitted.

9. Keypad Remote: Preset (072)

The Keypad Remote Preset task is used when the remote computer loads the preset. The PRESET screens are not shown in this configuration. Before the batch is started, the Preset can be viewed on the START screen. All pushbuttons are still valid.

The Brooks Protocol (cont'd)

Control Unit Tasks (cont'd)

10. Keypad Remote: Preset, Start (073)

The Keypad Remote Preset, Start task is used when the remote computer loads the preset. The PRESET screens are not shown in this configuration. The keypad START pushbutton has no effect when on the START screen. The START pushbutton can be used as the decimal point while the computer is in Program mode. The batch is started by the Remote Start task (036). All other pushbuttons are still valid.

11. Keypad Remote: Blend Recipe (074)

The Keypad Remote Blend Recipe task is used when the remote computer loads the blend recipe. The Blend Recipe screen is not shown in this configuration. Before the batch is started, the blend recipe can be viewed on the START screen. All other pushbuttons are still valid.

12. Keypad Remote: Blend Recipe, Start (075)

The Keypad Remote Blend Recipe, Start task is used when the remote computer loads the blend recipe. The Blend screen is not shown in this configuration. The keypad START pushbutton has no effect when on the Start screen. The START pushbutton can be used as the decimal point while the remote computer is in Program mode. The batch is started by the Remote Start task (036). All other pushbuttons are still valid.

13. Keypad Remote: Blend Recipe, Preset (076)

The Keypad Remote Blend Recipe and Preset task is used when the remote computer loads both the blend recipe and the preset. The Blend Recipe and Preset screens are not shown in this configuration. Before the batch is started, the blend recipe and the preset can be viewed on the START screen. All pushbuttons are still valid.

14. Keypad Remote: Blend Recipe, Preset and Start (077)

The Keypad Remote Blend Recipe, Preset and Start task is used for complete remote computer control of the batch. The Blend Recipe and Preset screens are not shown in this configuration. The keypad START pushbutton has no effect when on the Start screen. The START pushbutton can be used as the decimal point while the remote computer is in Program mode. The batch is started by the Remote Start task (036). All other pushbuttons are still valid.

The Brooks Protocol (cont'd)

Control Unit Tasks

15. Keypad Lock (078)

The Keypad Lock task disables the function of all the front panel pushbuttons except the STOP pushbutton. Control of the device is only possible via the two-way data communications links. The keypad is enabled by using the Keypad Unlock task (070).

16. Stop (080)

The Stop task is equivalent to the use of the front panel STOP pushbutton during a batch operation. The batch is stopped by the execution of this task. This task has no effect during a non-batch situation. A NAK is returned if the Control Unit is not batching.

17. Reset (085)

The Reset task is equivalent to the use of the front panel RESET pushbutton. The actual effect of this task depends upon the state of the Control Unit. This is determined by reading Control Status (Parameter 806). The state of the Control Unit is moved back one level.

18. Master Reset (086)

The Master Reset task returns the Control Unit to Operator mode, archives the transaction and zeros the delivered quantity during any non-batching situation.

19. The Remote Driver ID Acknowledge (090)

The Remote Driver ID Acknowledge task is used in conjunction with the Driver ID Enable Parameter (041) and the Control Status (806). Driver ID Enable must first be set to a "3" (REMOTE). The remote computer constantly polls parameter 806 to check if a driver ID number has been entered. After the remote computer determines that an ID number has been entered, the remote computer reads the ID value (845) and then verifies the number against its data base. If the number is valid the Remote Driver ID Acknowledge task is used to advance Operator mode to the next valid screen.

20. Remote Driver ID Retry (091)

The Remote Driver ID Retry task is used in conjunction with the Remote Driver ID Acknowledge task (090) and the Control Status parameter (806). Task 091 is used only if the Driver ID value is invalid. This task resets the Control Status parameter to a 0 which indicates ID Mode Waiting for Code Entry from a 1 indicating ID Mode Code Available. This reset allows the remote computer to know when a new number is entered.

Control Status Codes

For all PetroCount® IMS products the control and status registers provided for remote computer control of the product are grouped in the parameter range from 800 - 899. Please consult the corresponding product user's manual for the available status registers.

The control status register gives a precise indication of the current operating state of the device. Each possible operating state is assigned a code number. The following describes the control status register codes for each PetroCount® IMS product.

Control Unit Status Codes

The following describes the meaning of each of the possible control status codes found in Parameter 806 of a Control Unit.

| <u>Code</u> | <u>Description</u> |
|-------------|--|
| 0 | Driver ID, Waiting for Input - This code indicates that the Control Unit is presently displaying the Driver ID screen, and that an ID has not been entered. |
| 1 | Driver ID, Code Available - This code indicates that the Control Unit is presently displaying the Driver ID screen, and that an ID code has been entered. |
| 2 | Blend Selection - This code indicates that blending is enabled, and that the Control Unit is presently displaying the Blend Selection Screen. |
| 3 | Preset Selection - This code indicates that the Control Unit is presently displaying the Preset Entry Screen or the Press Start screen. |
| 4 | Batching - This code indicates that the Control Unit has commanded the control valve to open and flow should be occurring. This code is presently only when blending is disabled. |
| 5 | Blending - This code indicates that the Control Unit has commanded the control valve to open and flow should be occurring. This code is presented only when blending is enabled. |
| 6 | Program Mode, Weights and Measures Lock - This code indicates that the Control Unit is in the program mode, but the Weights and Measures security switch is in the secure position and therefore protected parameters may not be altered. |
| 7 | Program Mode, Weights and Measures Unlocked - This code indicates that the Control Unit is in the program mode, and that the Weights and Measures security switch has been pulled out and therefore all writable parameters may be altered. |
| 8 | Security Lock - This code indicates that the Control Unit is in a locked condition due to 3 consecutive unsuccessful attempts to enter an access code. The access code must be successfully entered from the front panel controls to clear this condition. |
| 9 | Operator Error - This code indicates that a data entry error has occurred. A reset must be sent to restore to normal operator mode. |

Control Status Codes (cont'd)

Control Unit Status Codes (cont'd)

- 10 **Start Failed** - This code indicates that the Control Unit has been instructed to start, but was unable to start the batch. Any of the following reasons may be the cause of a start failure: Archive full and halt on full selected; Pre-batch testing detected an error; Weights and Measured switch not in the secured position.
- 11 **Testing** - This code indicates that the Control Unit is performing pre-batch tests. No Operator or Remote Computer intervention is permitted during this state.
- 12 **Waiting for AC Permissive** - This code indicates that the Control Unit is waiting for the presence of the AC Permissive input before continuing with the batch. The Control Unit remains in this state until the AC Permissive is detected, or until a STOP Command is received.
- 13 **Opening Blend Valve** - This code indicates that the Control Unit has commanded a blend valve to open and is waiting for the Component Settling Time to expire. No Operator or Remote Computer intervention is permitted during this state.
- 14 **Blend Valve Open, Waiting for Conformation** - This code indicates that the Control Unit has commanded a blend valve to open and is waiting for conformation from the operator or remote computer. A START Command is required for conformation. A RESET Command causes the valve to be closed and the batch terminated.
- 15 **Valve Open Delay** - This code indicates that the Control Unit has commanded the Pump Contact to close, and is waiting for the specified Valve Open Delay Period before opening the flow control valve. No Operator or Remote Computer intervention is permitted during this state.
- 16 **Closing Blend Valve** - This code indicates that the Control Unit has commanded the Blend Valve to close, and is waiting for the blend valve feedback signals to indicate that all valves are closed. If all valves are not closed within the feedback alarm time an alarm is signaled if enabled, and control exits this state. No Operator or Remote Computer intervention is permitted during this state.
- 17 **Blend Valve Closed, Waiting for Conformation** - This code indicates that the desired blend valve has been closed and is waiting for operator or remote computer conformation. A START Command causes that Control Unit to continue normal operation. A RESET causes the Control Unit to terminate the batch.
- 18 **Waiting for Abort Blend Conformation, Blend Valve Open** - This code indicates that the a blend has been terminated by the operator or remote computer by issuing a RESET during the Blend Valve Open, Waiting for Conformation state. A START Command confirms that the blend is to be aborted. A STOP command indicates that the blend is not to be aborted, and the operator or remote computer can again be prompted to confirm valve open.

Control Status Codes (cont'd)

Control Unit Status Codes (cont'd)

- 19 **Alarm Halt** - This code indicates that the batch has been stopped due to a programmed alarm condition. The operator or remote computer must issue one RESET for each alarm active. When all alarms have been cleared, the operator or remoter computer will be prompted to issue a START.
- 20 **Operator Halt** - This code indicates that the batch has been stopped due to operator or remote computer command. A RESET must be issued to continue.
- 21 **Batch Halted due to AC Permissive Lost** - This code indicates that the batch has been stopped because of an interruption in the AC Permissive input. A RESET must be issued to continue.
- 22 **Batch Stopped, Waiting for Start Command** - This code indicates that stopped batch has been acknowledged by the operator or remote computer. A START Command will now cause the Control Unit to attempt to restart the batch. A RESET will terminate the batch.
- 23 **Blend Valve Not Open** - This code indicates that the blend valve did not open within the specified Settling Time. The operator or remote computer must acknowledge this condition with a RESET.
- 24 **Blend Valve Not Closed** - This code indicates that the blend valve did not close within the programmed feedback alarm time period. The operator or remote computer must acknowledge this condition with a RESET.
- 25 **Waiting for Abort Blend Conformation, Blend Valve Closed** - This code indicates that the a blend has been terminated by the operator or remote computer by issuing a RESET during the Blend Valve Closed, Waiting for Conformation state. A START Command confirms that the blend is to be aborted. A STOP command indicates that the blend is not to be aborted, and the operator or remote computer will again be prompted to confirm valve closed.
- 26 **Closing Blend Valve following Abort Command** - This code indicates that following an abort blend request, the Control Unit has commanded the Blend Valve to close, and is waiting for the blend valve feedback signals to indicate that all valves are closed. If all valves are not closed within the feedback alarm time and alarm is be signaled if enabled, and control will exit this state. No Operator or Remote Computer intervention is permitted during this state.
- 27 **Blend Valve Closed following Abort Command, Waiting for Conformation** - This code indicates that the desired blend valve has been closed and is waiting for operator or remote computer conformation. A START Command will cause the Control Unit to return to operator mode.

Control Status Codes (cont'd)

Control Unit Status Codes (cont'd)

- 28 **Closing Blend Valve at Completion of Batch** - This code indicates that following a successful completion of a batch, the Control Unit has commanded the Blend Valve to close, and is waiting for the blend valve feedback signals to indicate that all valves are closed. If all valves are not closed within the feedback alarm time and alarm is signaled if enabled, and control will exit this state. No Operator or Remote Computer intervention is permitted during this state.
- 29 **Blend Valve Closed, Batch Complete** - This code indicates that the desired blend valve has been closed and is waiting for operator or remote computer conformation. A START Command will cause the Control Unit to return to operator mode.
- 30 **Batch Complete** - This code indicates that a batch has been successfully completed. If any alarms occurred during the batch, the operator or remote computer must issue a RESET for each active alarm. After all alarms have been cleared, a RESET will return the Control Unit to the operator mode.
- 31 **Archiving/Data Logging** - This code indicates that the Control Unit is performing an archive and/or data log function. No operator or remote computer intervention is permitted during this condition.
- 32 **Unit Shutdown** - This code indicates that a serious internal fault condition has been detected. Any delivery operations have been halted by the Control Unit. This state can only be cleared by cycling main power to the Control Unit. Brooks strongly recommends that the unit be removed from service and the on-board diagnostic programs be executed to locate the source of the shutdown.
- 33 **Unauthorized Flow** - This status occurs if the displayed delivered quantity reaches 1 or more between transactions. (After an archive and before START is pressed for the next batch.) One RESET causes the unauthorized transaction to be archived and the delivered quantity to be reset to zero.
- 34 **Batch Stopped, Blend Valve Failure** - This status indicates that the Control Unit stopped the batch due to a failure of blend valve feedback. A RESET must be issued to continue.

Data Communications Interface Examples

The following provide some meaningful examples of using the PetroCount[®] IMS data communications interface to monitor and control the activity of the PetroCount[®] Ims products. For all of the following examples the computer has a communication address of 555 and the PetroCount[®] IMS device has an address of 777. Each example is illustrated using the computer mode of the Brooks Protocol.

Data Communications Interface Examples (cont'd)

Control Unit Examples

Example #1: Driver ID mode with remote computer verification.

This example assumes that the Control Unit is reset and the ID mode is configured for remote verification. (Parameter 041 = 2).

Step 1: Verify that the ID code value has been entered by the operator using the control status register (806).

Request

| | | | | |
|---|--------|---|------|-------|
| S | | S | | E |
| O | 777555 | T | R806 | T 3A |
| H | | X | | X |

Reply

| | | | | |
|---|--------|---|------------------|-------|
| S | | S | | E |
| O | 555777 | T | 806=000000000001 | T 36 |
| H | | X | | X |

Step 2: Retrieve the ID code value from the data register (845).

Request

| | | | | |
|---|--------|---|------|-------|
| S | | S | | E |
| O | 777555 | T | R845 | T 3D |
| H | | X | | X |

Reply

| | | | | |
|---|--------|---|---------------|-------|
| S | | S | | E |
| O | 555777 | T | 845=079321153 | T 96 |
| H | | X | | X |

**Data Communications
Interface Examples
(cont'd)**Control Unit Examples (cont'd)

Step 3a: If the ID code data is valid, acknowledge the entry and proceed with the next operator screen.

Request

| | | | | |
|---|--------|---|------|---|
| S | | S | | E |
| O | 777555 | T | X090 | T |
| H | | X | | X |

3B**Reply**

| | | | | |
|---|--------|---|--------|---|
| S | | S | | E |
| O | 555777 | T | X090=Y | T |
| H | | X | | X |

D1

Step 3b: If the ID code data is invalid, request a retry.

Request

| | | | | |
|---|--------|---|------|---|
| S | | S | | E |
| O | 777555 | T | X091 | T |
| H | | X | | X |

3C**Reply**

| | | | | |
|---|--------|---|--------|---|
| S | | S | | E |
| O | 555777 | T | X091=Y | T |
| H | | X | | X |

D2

Data Communications Interface Examples (cont'd)

Control Unit Examples (cont'd)

Example #2: Remote display control

Step 1: Execute a keypad lock-out task to prevent the state of the device from being changed when the display is locked.

Request

| | | | | | |
|---|--------|---|------|---|----|
| S | | S | | E | |
| O | 777555 | T | X078 | T | 41 |
| H | | X | | X | |

Reply

| | | | | | |
|---|--------|---|--------|---|----|
| S | | S | | E | |
| O | 555777 | T | X078=Y | T | D7 |
| H | | X | | X | |

Step 2: Execute a display lock-out task (both lines) to take control of the front panel alphanumeric display.

Request

| | | | | | |
|---|--------|---|------|---|----|
| S | | S | | E | |
| O | 777555 | T | X021 | T | 35 |
| H | | X | | X | |

Reply

| | | | | | |
|---|--------|---|--------|---|----|
| S | | S | | E | |
| O | 555777 | T | X021=Y | T | CB |
| H | | X | | X | |

Data Communications Interface Examples (cont'd)

Control Unit Examples (cont'd)

Step 2: (cont'd)

Request

| | | | | | |
|---|--------|---|------|---|----|
| S | | S | | E | |
| O | 777555 | T | X022 | T | 36 |
| H | | X | | X | |

Reply

| | | | | | |
|---|--------|---|--------|---|----|
| S | | S | | E | |
| O | 555777 | T | X022=Y | T | CC |
| H | | X | | X | |

Step 3: Send the desired message text to the front panel display.

Request

| | | | | | |
|---|--------|---|----------------|---|----|
| S | | S | | E | |
| O | 333789 | T | see text below | T | F6 |
| H | | X | | X | |

X023"ESC"CWELCOME TO OUR TERMINAL

Reply

| | | | | | |
|---|--------|---|--------|---|----|
| S | | S | | E | |
| O | 555777 | T | X023=Y | T | CD |
| H | | X | | X | |

Data Communications Interface Examples (cont'd)

Control Unit Examples (cont'd)

Step 4: Unlock the alphanumeric display for future use.

Request

| | | | | |
|---|--------|---|------|-------|
| S | | S | | E |
| O | 777555 | T | X025 | T 39 |
| H | | X | | X |

Reply

| | | | | |
|---|--------|---|--------|-------|
| S | | S | | E |
| O | 555777 | T | X025=Y | T CF |
| H | | X | | X |

Note: The keypad lock-out is still in affect. The state of the device cannot be changed locally until the keypad is enabled, (Task 070).

Example #3: Archive memory upload and reset

Step 1: Verify that the Control Unit is not batching by using the control status register 806.

Request

| | | | | |
|---|--------|---|------|-------|
| S | | S | | E |
| O | 777555 | T | R806 | T 3A |
| H | | X | | X |

Reply

| | | | | |
|---|--------|---|------------------|-------|
| S | | S | | E |
| O | 555777 | T | 806=000000000003 | T 38 |
| H | | X | | X |

Data Communications Interface Examples (cont'd)

Control Unit Examples (cont'd)

Step 2: Execute the keypad lock-out task to prevent the operator from changing the state of the unit.

Request

| | | | | | |
|---|--------|---|------|---|----|
| S | | S | | E | |
| O | 777555 | T | X078 | T | 41 |
| H | | X | | X | |

Reply

| | | | | | |
|---|--------|---|--------|---|----|
| S | | S | | E | |
| O | 555777 | T | X078=Y | T | D7 |
| H | | X | | X | |

Step 3: Check the total number of transactions that have been stored (930).

Request

| | | | | | |
|---|--------|---|------|---|----|
| S | | S | | E | |
| O | 777555 | T | R930 | T | 38 |
| H | | X | | X | |

Reply

| | | | | | |
|---|--------|---|-----------------|---|----|
| S | | S | | E | |
| O | 555777 | T | 930=00000000002 | T | 35 |
| H | | X | | X | |

Data Communications Interface Examples (cont'd)

Control Unit Examples (cont'd)

Step 4: Set the transaction index (900) to 1 to view the first transaction in the archive memory.

Request

| | | | | | |
|---|--------|---|---------|---|----|
| S | | S | | E | |
| O | 777555 | T | W900=01 | T | D8 |
| H | | X | | X | |

Reply

| | | | | | |
|---|--------|---|---------|---|----|
| S | | S | | E | |
| O | 555777 | T | W900=01 | T | D8 |
| H | | X | | X | |

Step 5: Read the desired archive parameters.

Request

| | | | | | |
|---|--------|---|----------------|---|----|
| S | | S | | E | |
| O | 777555 | T | see text below | T | 94 |
| H | | X | | X | |

R901;902;903;904;905

Reply

| | | | | | |
|---|--------|---|----------------|---|----|
| S | | S | | E | |
| O | 555777 | T | see text below | T | CF |
| H | | X | | X | |

901=00000891532;902=000745639.27;
903=000786532.98;904=10:27:33;
905=10:05:29

Step 6: Set the transaction index (900) to the next value and reread the desired archive parameter values for the next transaction. Continue this process until all of the stored transactions have been read.

Data Communications Interface Examples (cont'd)

Control Unit Examples (cont'd)

Step 7: Clear the archive memory.

Request

| | | | | |
|---|--------|---|--------|---|
| S | | S | | E |
| O | 777555 | T | W930=0 | T |
| H | | X | | X |

AA

Reply

| | | | | |
|---|--------|---|--------|---|
| S | | S | | E |
| O | 555777 | T | W930=0 | T |
| H | | X | | X |

AA

Example #4: A batch delivery sequence with computer control of security access blend recipe selection and preset quantity with local start by the operator.

This example assumes that the Control Unit is reset and ID mode is configured for remote verification (Parameter 041 = 2).

Step 1: Verify that the ID code value has been entered by the operator using the control status register (806).

Request

| | | | | |
|---|--------|---|------|---|
| S | | S | | E |
| O | 777555 | T | R806 | T |
| H | | X | | X |

3A

Reply

| | | | | |
|---|--------|---|------------------|---|
| S | | S | | E |
| O | 555777 | T | 806=000000000001 | T |
| H | | X | | X |

36

**Data Communications
Interface Examples
(cont'd)**

Control Unit Examples (cont'd)

Step 2: Retrieve the ID code value from the data register (845).

Request

| | | | | |
|---|--------|---|------|-------|
| S | | S | | E |
| O | 777555 | T | R845 | T 3D |
| H | | X | | X |

Reply

| | | | | |
|---|--------|---|--------------|-------|
| S | | S | | E |
| O | 555777 | T | 845=56798321 | T D1 |
| H | | X | | X |

Step 3: If the ID code data is invalid, request a retry.

Request

| | | | | |
|---|--------|---|------|-------|
| S | | S | | E |
| O | 777555 | T | X091 | T 3C |
| H | | X | | X |

Reply

| | | | | |
|---|--------|---|--------|-------|
| S | | S | | E |
| O | 555777 | T | X091=Y | T D2 |
| H | | X | | X |

Data Communications Interface Examples (cont'd)

Control Unit Examples (cont'd)

Step 4: If the ID code data is valid, enter the desired blend selection and preset value (blend#3, 1200 units).

Request

| | | | | | |
|---|--------|---|--------------------|---|----|
| S | 777555 | S | W804=03 ; 820=1200 | E | |
| O | | T | | T | B2 |
| H | | X | | X | |

Reply

| | | | | | |
|---|--------|---|--------------------|---|----|
| S | 555777 | S | W804=03 ; 820=1200 | E | |
| O | | T | | T | B2 |
| H | | X | | X | |

Step 5: Execute the keypad remote task to disable local blend recipe and preset entry and permit local start of the batch.

Request

| | | | | | |
|---|--------|---|------|---|----|
| S | 777555 | S | X076 | E | |
| O | | T | | T | 3F |
| H | | X | | X | |

Reply

| | | | | | |
|---|--------|---|--------|---|----|
| S | 555777 | S | X076=Y | E | |
| O | | T | | T | D5 |
| H | | X | | X | |

The Control Unit is now ready to begin the delivery sequence. The delivery is initiated by the operator pushing the START button.

Data Communications Interface Examples (cont'd)

Control Unit Examples (cont'd)

Step 6: Poll the control status register to determine when the batch is complete and the unit is reset and awaiting the next operator.

Request

| | | | | |
|---|--------|---|------|-------|
| S | | S | | E |
| O | 777555 | T | R806 | T 3A |
| H | | X | | X |

Reply

| | | | | |
|---|--------|---|------------------|-------|
| S | | S | | E |
| O | 555777 | T | 806=000000000000 | T 35 |
| H | | X | | X |

Step 7: Upload the desired archive parameters.

Request

| | | | | |
|---|--------|---|----------------|-------|
| S | | S | | E |
| O | 777555 | T | see text below | T 94 |
| H | | X | | X |

R901;902;903;904;905

Reply

| | | | | |
|---|--------|---|----------------|-------|
| S | | S | | E |
| O | 555777 | T | see text below | T CF |
| H | | X | | X |

901=00000891532;902=000745639.27;
 903=000786532.98;904=10:27:33;
 905=10:05:29

**Data Communications
Interface Examples
(cont'd)**

Control Unit Examples (cont'd)

Step 8: Clear the archive memory.

Request

| | | | | |
|---|--------|---|--------|--------|
| S | | S | | E |
| O | 777555 | T | W930=0 | T AA |
| H | | X | | X |

Reply

| | | | | |
|---|--------|---|--------|--------|
| S | | S | | E |
| O | 555777 | T | W930=0 | T AA |
| H | | X | | X |

Manufacturing Operations
Fisher-Rosemount Petroleum
19267 Highway 301 North
PO Box 450
Statesboro, GA 30459-0450
USA
Tel +1 912-489-0200
Fax +1 912-489-0430

Central USA
The Sulton Group
4404 South Maybelle Avenue
PO Box 9186
Tulsa, OK 74107
USA
Tel +1 918 446-1611
Fax +1 918 446-2817

Europe
Fisher-Rosemount Petroleum
Groeneveldselaan 6
3903 AZ Veenendaal
The Netherlands
Tel +31-318-549-549
Fax +31-318-549-559

Far East
Fisher-Rosemount Petroleum
1 Pandan Crescent
Singapore 128461
Republic of Singapore
Tel +65-777-8211
Fax +65-777-0947

Middle East
Fisher-Rosemount Petroleum
Jebel Ali Free Zone
PO Box 17033
Dubai
United Arab Emirates
Tel +971-4 835-235
Fax +971-4 835-312

Division Headquarters
Fisher-Rosemount Petroleum
4910 Wright Road, Suite 100
Stafford, Texas 77477 USA
Tel +1 281-240-0701
Fax +1 281-274-9888

The contents of this publication are presented for informational purposes only, and while every effort has been made to ensure their accuracy, they are not to be construed as warranties or guarantees, express or implied, regarding the products or services described herein or their use or applicability. We reserve the right to modify or improve the designs or specifications of such products at any time without notice.

Fisher-Rosemount Petroleum Division, Emerson Electric Co.

Fisher-Rosemount Petroleum
FISHER-ROSEMOUNT™ Managing The Process Better.