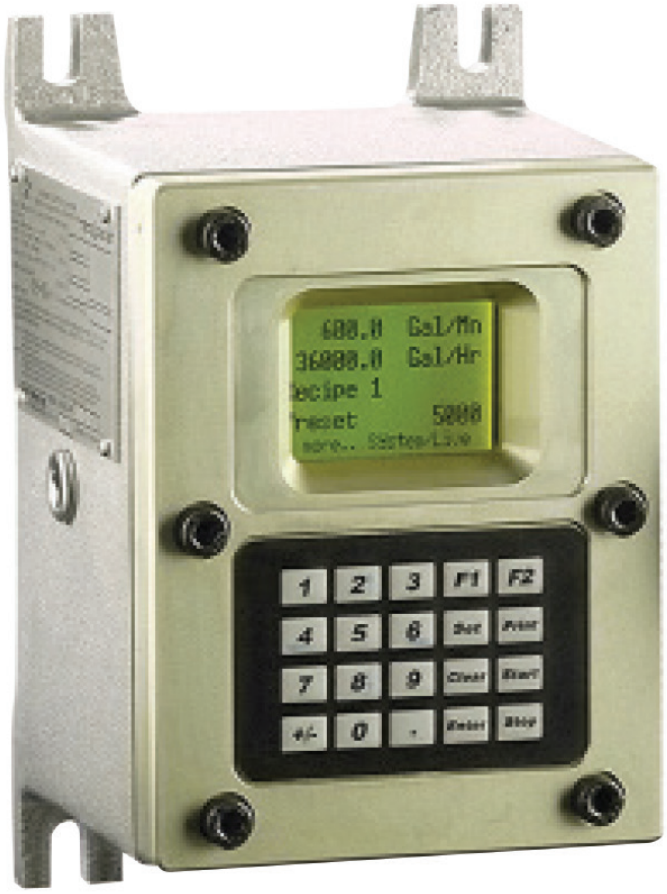
GUIDANT

Electronic Flow Computer

Smith Meter® microFlow.net™ Liquid

Communications Manual

Bulletin MNFL002 Issue/Rev 0.0 (2/12)



Caution

The default or operating values used in this manual and in the program of the microFlow.net are for factory testing only and should not be construed as default or operating values for your metering system. Each metering system is unique and each program parameter must be reviewed and programmed for that specific metering system application.

Disclaimer

Guidant hereby disclaims any and all responsibility for damages, including but not limited to consequential damages, arising out of or related to the inputting of incorrect or improper program or default values entered in connection with the microFlow.net.

Proprietary Notice

Table of Contents

Table of Contents

Section I – Introduction

This manual fully describes how the Smith Meter® microFlow.net Electronic Preset communicates with other computing devices from a simple dumb terminal to a large computer.

Incorporated within the microFlow.net is the ability to communicate directly (i.e., without a modem or multiplexer) with an EIA 232C, EIA 485, and/or Ethernet compatible remote terminal or minicomputer. Depending on the communication type, certain key information from multiple microFlow.nets can be requested (polled). If the microFlow.nets are so programmed, the meter position desired may be remotely authorized and released for the operator's use. The operator may also authorize specific additives, remotely set the batch volume, reset alarms, and remotely program certain microFlow.net program codes.

To communicate with a particular microFlow.net unit, the following communication Program Codes involving type, mode, address, and configuration must be specified for that particular unit as shown in the chart below.

For Serial Port Communications:

System Program Codes 701 - 718			
Port 1	**Port 2**	**Port 3**	
701	707	713	Function
702	708	714	Baud
703	709	715	Data/Parity
704	710	716	Control
705	711	717	Time-out
706	712	718	Mode (RS232/485)

For Ethernet and SLIP Communications:

System Codes	
722	Netmask
723	Gateway
724	Ethernet Host Control
726	Ethernet Time-out

For Both Types of Communication:

System Codes	
721	microFlow.net unit Address (x.x.x.1 – x.x.x.99-serial; valid IP address-Ethernet)
725	Comm Link Programming (Level of Access)

For Modbus Communications:

Communications	
727	Modbus Endian

Serial Communications

Samuel F.B. Morse's dot-dash telegraph code is the earliest example of a practical, time sequential, data-coding scheme for transmission of information by communication equipment. This code is considered the predecessor of the ones and zeros modern digital communication codes now used for serial data transmission of time sequenced information over a pair of wires.

Similar to Morse Code, digital codes provide a means of representing numbers, letters of the alphabet, or other special characters in a digital information system. A digital code is a pattern of binary digits or bits, zeros and ones arranged in a particular fashion. The most familiar code used for arithmetic computations in digital systems is the Binary Coded Decimal, commonly known as BCD code. The BCD code is a weighted code in that a numerical weight is assigned to each bit position in the code. Using a four-bit BCD code for an example, the left-most bit has a numeric weighted value of 8, the next bit has a numeric weighted value of 4, the next to the last bit a weight value of 2 and the last bit, a value of 1. The total value of the coded number is equal to the sum of the numerical weights of the bits represented by the binary digit 1. Four-bit BCD codes are valid only for numbers between 0 and 9. For example, the number 3 is represented by a BCD code of "0011," and the number 9 is "1001." To represent 39, the respective BCD code is "0011 1001."

There are many different codes used to perform specific tasks in digital systems, but the one code most widely used in digital communications systems is the American Standard Code for Information Interchange, or simply ASCII code. Like other binary codes, the ASCII code is a weighted code.

The ASCII code is a more complex code than BCD since it uses patterns of seven bits to represent 128 characters consisting of either upper or lowercase letters of the alphabet, punctuation characters, and control characters in addition to numbers. For example, the ASCII code representation of the number 39 is "0110011 0111001." A complete ASCII code character table is shown in *Table 1*.

ASCII Code Table

ASCII CHARACTER	DECIMAL	HEX	BINARY
NUL	0	0	000 0000
STX	2	2	000 0010
ETX	3	3	000 0011
LF	10	A	000 1010
CR	13	D	000 1101
SP	32	20	010 0000
!	33	21	010 0001
"	34	22	010 0010
#	35	23	010 0011
$	36	24	010 0100
%	37	25	010 0101
&	38	26	010 0110
'	39	27	010 0111
(	40	28	010 1000
)	41	29	010 1001
*	42	2A	010 1010
+	43	2B	010 1011
,	44	2C	010 1100
-	45	2D	010 1101
.	46	2E	010 1110
/	47	2F	010 1111
0	48	30	011 0000
1	49	31	011 0001
2	50	32	011 0010
3	51	33	011 0011
4	52	34	011 0100
5	53	35	011 0101
6	54	36	011 0110
7	55	37	011 0111
8	56	38	011 1000
9	57	39	011 1001
:	58	3A	011 1010
;	59	3B	011 1011
<	60	3C	011 1100
=	61	3D	011 1101
>	62	3E	011 1110

ASCII CHARACTER	DECIMAL	HEX	BINARY
?	63	3F	011 1111
@	64	40	100 0000
A	65	41	100 0001
B	66	42	100 0010
C	67	43	100 0011
D	68	44	100 0100
E	69	45	100 0101
F	70	46	100 0110
G	71	47	100 0111
H	72	48	100 1000
I	73	49	100 1001
J	74	4A	100 1010
K	75	4B	100 1011
L	76	4C	100 1100
M	77	4D	100 1101
N	78	4E	100 1110
O	79	4F	100 1111
P	80	50	101 0000
Q	81	51	101 0001
R	82	52	101 0010
S	83	53	101 0011
T	84	54	101 0100
U	85	55	101 0101
V	86	56	101 0110
W	87	57	101 0111
X	88	58	101 1000
Y	89	59	101 1001
Z	90	5A	101 1010
[	91	5B	101 1011
\	92	5C	101 1100
^	94	5E	101 1101
_	95	5F	101 1111
`	96	60	110 0000
A	97	61	110 0001
B	98	62	110 0010
C	99	63	110 0011
D	100	64	110 0100
E	101	65	110 0101

ASCII CHARACTER	DECIMAL	HEX	BINARY
F	102	66	110 0110
G	103	67	110 0111
H	104	68	110 1000
I	105	69	110 1001
J	106	6A	110 1010
K	107	6B	110 1011
L	108	6C	110 1100
M	109	6D	110 1101
N	110	6E	110 1110
O	111	6F	110 1111
P	112	70	111 0000
Q	113	71	111 0001
R	114	72	111 0010
S	115	73	111 0011
T	116	74	111 0100
U	117	75	111 0101
V	118	76	111 0110
W	119	77	111 0111
X	120	78	111 1000
Y	121	79	111 1001
Z	122	7A	111 1010
{	123	7B	111 1011
∠	124	7C	111 1100
}	125	7D	111 1101
•	126	7E	111 1110
DEL	127	7F	111 1111

Table 1

A computer system always requires some digital data transmission between its various parts: CPU to peripherals, CPU to memory, or memory to peripherals. Data transmission to and from these devices must conform to some accepted standard. To date, the only widely used transmission standards deal with serial digital data. There are essentially three organizations that issue standards that define serial digital communication interface circuits, their electrical and timing characteristics, the manner in which they operate, and the mechanical details of the appropriate connectors. These organizations are the Electronics Industries Association (EIA), the International Consultative Committee for Telephony and Telegraphy (CCITT), and the International Standards Organization (ISO).

EIA Standard, EIA 232 (formerly known as RS 232) is the most popular serial interface standard. This standard is extensively used by terminals, data sets, measuring instruments, and controllers for data transmission rates up to 20,000 bits per seconds for transmission cables up to 50 feet in length. EIA 232 is a single-ended voltage mode transmission system standard that defines data communication between equipment using alternating pulses which can be in one of two states – either high (logic 1) or low (logical 0). These states are often called “mark” (logic 1) or “space” (logic 0). According to EIA, the logical 1 level must be within +3.75 to +25 volts DC, while the logical 0 level must be within 3.75 to 25 volts DC. Any other voltage levels are unacceptable according to EIA standards.

EIA 232 is not the only serial interface standard or system. EIA 422, 485 and 20mA current loop are among the newer long-distance current mode digital communication standards. The current mode standards are better suited for longer distance, higher speed communications than its voltage mode predecessors. Although not a revolutionary concept, the current mode system dates back to the oldest form of binary serial transmission: the telegraph. In this system a current, usually 20mA, flows through a single loop to represent a logic level one, and turns off, “open key” to represent logic zero.

Serial data is typically transmitted among or between devices in an asynchronous fashion. In asynchronous data transmission, each transmitted character is formed by using a start bit which signals the beginning of the character before the ASCII code pattern, and one or two stop bits after the code pattern signaling the end of the character. The ASCII character is described fully by seven bits with an optional parity bit in the eighth position for error control. Therefore each transmitted ASCII character requires at least ten bits for complete definition. As the communication equipment receives the asynchronously transmitted characters, the start and stop bits are stripped off, parity is checked, and the character itself is interpreted and treated according to whether it is alphanumeric data or control information.

“Baud rate” and “bit rate” are two distinct terms used to describe the speed of data transmission. These terms are often used synonymously and cause much confusion if not completely understood. The baud is a measurement unit dating back to the days of Morse Code, and it is defined as the shortest signaling element. In modern telecommunications language, the data rate is more often specified in bits per second (bps), because a single change of state in a signal can represent a group of two or more data bits. If each signal event represents only one bit condition, baud rate equals bps. Typical asynchronous serial baud rates are 1200, 2400, 4800, 9600, 19200, and 38400 bps. To have an interactive session between two computing devices, both of the devices must be transmitting and receiving at the same baud rate, or there must be an intermediate memory device, called a buffer, that accommodates the differences in speed. Refer to the microFlow.net installation manual to determine the appropriate baud rate based on cable length for each unique installation site.

Ethernet and TCP/IP Communications

The proliferation of personal computing beginning in the 1970s gave rise to the need to interconnect groups of computers for the purpose of sharing data, peripheral devices (printers, modems…) and now instruments. The most popular of these groups are known as Local Area Networks (LANs). These networks consist of nodes, where computers, peripherals and instruments are connected to the network, and interconnecting wire or fiber optic cable to interconnect the nodes. A LAN can consist of a few nodes up to several hundred but will be confined to a few buildings within a few thousand meters of one another. Technologies were developed to establish standard interface hardware as well as secure control of the flow of data on the LAN. Ethernet emerged as the primary medium for LANs. The Ethernet technology equipment; interface cards, hubs, switches, and cabling have become commodity items. Software protocols were developed to standardize sharing and transfer of files, mail messages, access to peripherals, and access to the internet. Again a primary standard has emerged in the TCP/IP protocol. The acronym TCP/IP comes from two protocols developed for the internet; Transmission Control Protocol and Internet Protocol.

The microFlow.net can be connected to a TCP/IP LAN using the Ethernet port; or it can be networked in a point-to-point configuration via one of the serial ports using the SLIP protocol.

Communication Types

The type of communicating device that is being used in the system with a microFlow.net is programmable and can be defined in the communications directory of the microFlow.net. Communicating devices can be used with any of the serial communications ports on the microFlow.net.

Terminal (Term Host) – The microFlow.net communications ports communicate with a terminal type device using a simplified communications protocol.

Minicomputer (Minicomp Host) – The microFlow.net communications ports communicate with a minicomputer type device using a sophisticated and secure communications protocol.

SLIP (Serial Line Internet Protocol) – The microFlow.net comunications ports communicate with a minicomputer type device using TCP/IP over a serial communications line. Note that host communications over TCP/IP (either via SLIP or Ethernet) follows the Terminal mode protocol and uses port 7734.

Printer – The microFlow.net will automatically print a report at the end of each transaction. Each microFlow.net may be connected to a printer or shared printing can also be used (several microFlows utilizing one printer).

Smart Injector – Permits the microFlow.net to communicate with smart additive injector systems. (Examples are Gate City Blend-Pak and Mini-Pak, and Titan Pak3).

Card Reader – Permits the microFlow.net to communicate with the Smith Card Reader.

Mass Meter – Permits the microFlow.net to communicate with the Smith SMass or Apollo mass meters.

Modbus Host – The microFlow.net communicates with other computer systems using the Modbus protocol. (Available in Rev 0.07 and above).

Communications for Terminal Mode of Operation

The microFlow.net System Program Code Communications Port Function must be set to Terminal Host. This character-oriented protocol uses the ASCII character "*" to define the start of a message and Carriage Return – Line Feed (CR-LF) characters to terminate the message. No error checking other than parity on each character is performed.

The message format is:

*	A1 A2	text	CR	LF

for an instruction to microFlow.net, or

*	A1 A2	text	CR	LF

for a response from microFlow.net

Where:

*	=	Asterisk Hex “2A”
Text	=	Character string containing instructional or response information
CR	=	Carriage return Hex “0D”
LF	=	Line feed Hex “0A”
A1 A2	=	microFlow.net Address (01 to 99)

The universal or global address “00” is an invalid address and must not be assigned to any microFlow.net. The address, A1 A2, always consists of two ASCII characters.

Data is formatted using ASCII characters and each character frame consists of 1 start bit, 7 or 8 data bits, none, even or odd parity, and 1 or 2 stop bits. A maximum communication rate of 38,400 baud is supported. There is no echo back of received characters by the microFlow.net in the Terminal Mode of operation.

Note: *Via an established Ethernet or SLIP connection, this protocol is always available via port 7734.*

Communications for Minicomputer Mode of Operation

This character-oriented protocol uses the transmission control character STX to define the start of a message, and ETX to terminate the message. A Longitudinal Redundancy Check (LRC) character follows the ETX character for additional message error detection beyond the traditional parity check done on each transmitted character.

The message format is:

STX	A1 A2	text	ETX	LRC

for an instruction to microFlow.net, or

NL	STX	A1 A2	text	ETX	LRC	PAD

for a response from microFlow.net

Where:

NL	=	Null character Hex “00”
STX	=	Start of Text Hex “02”
Text	=	Character string containing instructional or response information
ETX	=	End of Text Hex “03”
LRC	=	Longitudinal Redundancy Check
PAD	=	Pad character Hex “7F”
A1 A2	=	microFlow.net Address (01 to 99)

The LRC is an ASCII character computed as the exclusive OR (XOR) sum of all characters following the STX and including the ETX transmission control characters.

The universal or global address “00” is an invalid address and must not be assigned to any microFlow.net. The address, A1 A2, always consists of two ASCII characters.

Data is formatted using ASCII characters and each character frame consists of 1 start bit, 7 or 8 data bits, none, even or odd parity, and 1 or 2 stop bits. There is no echo back of received characters by the microFlow.net in the Minicomputer Mode of communications.

Text Format

Command and Response text will be shown enclosed in single quotes. Embedded spaces are represented by an underscore character (_). Any other character representation will be described where used.

An “OK” is used in response to any action type command that has been successfully carried out. For request only commands, a good response will report the data requested in the format shown for that command.

A “NOXX” (XX represents a two character code) is used to show that the command has been rejected. The two-character code represents the condition causing the rejection. For an expanded description of these codes, see “Appendix II.”

Time-out, or no response received from the microFlow.net, occurs when the command string has been entered incorrectly. The communicating program should set an upper limit on the amount of time it will wait for a response from any microFlow.net, and register a time-out when that time has elapsed, to prevent a bad command from locking up the communications. Commands must be formatted exactly as stated. Invalid addresses, incomplete data, and excess data are all causes for this to occur. A more detailed explanation follows:

Invalid Address – The microFlow.net will ignore a command whose address does not match its own. The communication address is programmed into the microFlow.net System program code 721. For serial communications, the last octet of the four octet IP address is used.

Incomplete Data – The code format for each communication command is stated in the Command Reference Guide section. If any portion of the command is left out, a time-out will occur.

Excess Data – Commands must be formatted exactly as stated. No excess data may be inserted or added.

Communication Control Selections

The amount of control that the communicating device has over the microFlow.net is programmable for various degrees of control.

Poll and Program – Identical to “Polling Only” and adds programming privilege, but excludes authorizing privilege.

Host Control – Permits the EIA-232, EIA-485 or Ethernet communication device to request information and to authorize operation, or to have complete control over all operations.

XON/XOFF – Printer security protocol, designed to keep the printer buffer from overflowing. The printer sends an XOFF(13 hex) when the print buffer is nearly full. The microFlow.net stops sending data until the printer sends an XON (11 hex) signifying that it is ready for more data.

PTB-FX – Printer security protocol, designed to guarantee the printer received and printed each line it is sent. Primarily used in European markets.

PTB-LQ – Printer security protocol, designed to guarantee the printer received and printed each line it is sent. Primarily used in European markets.

Each command listed in the Command Reference Guide section of this manual indicates the supporting communication modes.

Section IV – Communications with Smart Additive Injectors

The microFlow.net communicates with the Titan, Gate City Smart Additive, and Smith Smart Additive Systems. After three tries, if there is no response from the additive injector system, an alarm will be set and the microFlow.net will respond to the alarm as it has been instructed to do in the programming.

The microFlow.net controls the Additive Subsystem totally through communications. The additive pacing is by communications, the additive system receives communication commands at the same interval at which the piston injectors would receive a signal to inject. With smart injectors, no incoming pulses are required by the additive injector. Certain parameters must be established in the Additive Subsystem prior to each batch, commands that will have to be issued during the batch and the end of the batch. In all cases, the parameters and commands are only issued to those additive injector systems that have been authorized for use for the current batch.

Passby Communications Mode

The Passby Communications Mode is designed to allow supervisory computers to communicate with smart additive injectors under the control of the microFlow.net. Commands from the supervisory computer will be received from the microFlow.net on one communications line, and forwarded to the Additive Injection Subsystem on another communications line. Queries and control commands will only be permitted while the microFlow.net is in the Ready Mode. No queries or commands will be accepted by the microFlow.net while it is in the Run Mode or the Programming Mode. “IC” and “IR” commands are used to implement Passby Communications Mode.

Communications from a supervisory computer to a smart additive injector will be a three-step process. First, the supervisory computer will issue a passby command to the microFlow.net. The microFlow.net will acknowledge receipt of the command to the supervisory computer. Next, the microFlow.net will add framing characters as required and issue the command to the smart additive injector. The additive injector will return an appropriate response to the microFlow.net. The final step occurs when the supervisory computer requests the additive’s response from the microFlow.net.

The communications scheme was developed to maintain a response time consistent with other commands sent by the supervisory computer.

Command Code AR – Alarm Reset

This command resets one of the alarms currently registered at the microFlow.net.

Command:

"AR" Reset all alarms in all tables
"AR_XX_DD" Reset specific alarm for a specified directory
"AR_XX_IN_YY" Reset specific alarm for a specified injector

Where: XX = Two-character alarm code
DD = Directory, where:
SY = System
YY = Injector (01 – 04)
IN = Constant (indicates additive injector alarm)

Responses:

"OK" Good response to the Alarm Reset Command

or...

"NOXX" The alarm was not reset.

Remarks: The two-character alarm code must be one of those alarms that is allowed to be reset through the communication channel. All alarms except "DA" can be cleared through communications.

Constraints: The alarm code must be able to be reset through communications. If it is allowed, it must be pending or a "NO" will be returned.

Special Case: A special code, "AA," may be used to reset all resettable alarms that are pending in the directory specified.

Comm. Modes: Host Control.

Alarm Status Codes That Can Be Reset Through Communications – System

Code	Condition
BP	Back-Pressure – Back pressure per the entries set cannot be maintained during flow enough to maintain entry set in product program code.
CM	Communications Alarm – Communications failure on one of the communications channels.
DR	Density Transducer – Density transducer failure or out-of-range condition.
HB	The S&W alarm limit has been exceeded.
HD	High Density – Density transducer is out of range of the high alarm setting.
HF	High Flow – Flow rate has exceeded limit set by Excess High Flow program parameter for more than 4 seconds.
HP	High Pressure – Pressure transducer is out of range of the high alarm setting.
HT	High Temperature – Temperature probe or transducer is out of range of the high alarm setting.
LD	Low Density – Density transducer is out of range of the low alarm setting.
LF	Low Flow Alarm – Flow rate was at or below the minimum flow rate established by Low Flow Limit program code for longer that 8 seconds.
LP	Low Pressure – Pressure transducer is out of range of the low alarm setting.
LT	Low Temperature – Temperature probe or transducer is out of range of the low alarm setting.
MF	Mass Meter Communications Failure – This alarm is set when any command sent to a mass meter fails both the first and second attempt. The normal polling sequence to each of the mass meters is not interrupted by the occurrence of a mass meter communications alarm. The mass meter in alarm will be skipped in the polling loop until this alarm is cleared.
MO	Mass Meter Overdrive – This alarm is set when a mass meter reports a status indicating a tube imbalance condition exists. (This alarm is valid only for S-Mass.)
MT	Mass Meter Tube – This alarm is set when a mass meter reports a status indicating a tube imbalance condition exists. (This alarm is valid only for S-Mass.)
PA	Power-Fail Alarm – The unit either had a power failure or a hardware reset occurred.
PP	Reprint Report – Initiates a reprint of the requested batch report.
PR	Pressure Transducer – Pressure transducer failure or out-of-range condition.
PS	Pulse Security – Used only with the security pulse option. Indicates that an excessive number of out-of-sequence errors in the A-B pulse stream have been detected.
SA	Indicates that excessive flow rate has caused the sampler to miss a sample.
SF	Batch storage has been completely filled.
SP	Shared Printer – An output was attempted to the shared printer but was unsuccessful because the shared printer remained busy longer than the programmed communications time-out.
SR	S&W Transducer failure or out-of-range condition.
TP	Temperature Probe – Short or open condition in the temperature probe circuit.
U1	User Alarm #1
U2	User Alarm #2

Code	Condition
U3	User Alarm #3
U4	User Alarm #4
U5	User Alarm #5
UC	Ultrasonic communications has failed.
UM	Ultrasonic meter alarm.
VF	Valve Fault – Indicates that the meter was still registering flow when the valve fault timer expired after the microFlow.net commanded the valve to close.
ZF	Zero Flow – The microFlow.net did not see flow through the system before the zero flow timer expired.

Resettable Alarms – System

Alarm Status Codes That Can Be Reset Through Communications – Injector

Code	Condition
AC	Additive Communications Error – Indicates a failure on the master/slave communications line between the microFlow.net and the Additive Injector Subsystem.
CR	Command Refused Error – Indicates that the command from the microFlow.net to the injector was rejected.
FA	Additive Feedback Error – Indicates that the additive feedback has exceeded the programmed number of errors.
GA	General Additive Error – Indicates that there is an additive injector error.
KA	Low Additive Error – Indicates that not enough additive was injected during one cycle or an average of several cycles.
MA	Excessive Additive Pulses Error – Indicates that too many additive flow meter pulses were detected.
NA	No Additive Pulses Error – Indicates that the additive flow meter's pulses were not detected.
OR	Overspeed Injector – Indicates that the meter on the metered injector has exceeded its specified maximum frequency.
RA	Additive Frequency Error – Indicates that the additive volume is too high for the rate selected; a second dose of additive is being requested before delivery of the first dose completes.

Resettable Alarms – Injector

Command Code AT – Request Additive Volumes by Transaction

This command retrieves additive batch volumes from microFlow.net.

Command:

Current Batch

"AT_X" Cumulative batch additive volume
"AT_X_Z"

Local Storage Batch

"AT_X_NNN" Total additive in an historic batch
"AT_X_Z_NNN"

Where: X = 1 through 4 (Additive #)
NNN = number of transactions back into local storage to retrieve the data
Z = Volume type
R = IV
G = GV
N = GST
P = GSV
M = Mass

Responses:

Good Response:

Current Batch

"AT_X_OI_VVVVV.VVV" For "AT_X"

Local Storage Batch

"AT_X_OI_VVVVV.VVV_NNN" For "AT_X_OI_NN"

Where: X = 1 through 4 (Additive #)
VVVVV.VVV = Total Additive Batch Volume
NNN = # of Batches back into Local Storage to retrieve the data
or...

"NOXX" Additive volume can not be reported

Remarks: None.

Constraints: Volume units are assumed as the type programmed into microFlow.net. Additive must be assigned.

Special Case: None.

Comm. Modes: No Control, Host Control, Poll and Program.

Command Code BR – Boolean/Algebraic Variable Read

This command retrieves the Boolean/Algebraic from microFlow.net.

Command:

"BR_X_YYY" Reads the Boolean / Algebraic Variables.

Where: X = F – Algebraic Variable (single precision, floating point)
B – Boolean Variable
T – Timer Variable
S – String Variable

YYY = Variable number; 1-50 for Float and Boolean types, 1-8 for timers and strings.

Responses:

Good Response
"BR_X_YYY_D...D"

Where: X = F – Algebraic Variable (single precision, floating point)
B – Boolean Variable
T – Timer Variable
S – String Variable

YYY = Variable number
D...D = Data; 0 to 255 for Boolean variables and numeric string for algebraic and timer variables. Character string for string variables.

or...

"NOXX"

Remarks: None.

Constraints: None.

Special Case: None.

Comm. Modes: No Control, Host Control, Poll and Program.

Command Code BW – Boolean/Algebraic Variable Write

This command writes to the Boolean/Algebraic variables in microFlow.net.

Command:

"BW_X_YYY_D...D" Writes to the Boolean / Algebraic Variables.

Where: X = F – Algebraic Variable (single precision, floating point)
B – Boolean Variable
T – Timer Variable
A – Set User Alarm (Leave off _D...D)
S – String Variable

YYY = Variable number – 1 to 50 for Boolean and Float types, 1 to 8 for Timers, 1 to 5 for User Alarms

D...D = The data; 0 to 255 for Boolean variables and numeric floating point string for algebraic and timer variables (no radix point allowed for timer variables)

Responses:

Good Response
"BR_X_YYY_D...D"

Where: X = F – Algebraic Variable (single precision, floating point)
B – Boolean Variable
T – Timer Variable
S – String Variable

YYY = Variable number

D...D = The data; 0 or 1 for Boolean variables and numeric string for algebraic and timer variables. 1 to 32 characters for string variables. Numeric data for float and timer requests.

or...

"NOXX"

Remarks: None.

Constraints: None.

Special Case: None.

Comm. Modes: No Control, Host Control, Poll and Program.

Command Code DY – Request Dynamic Display Values

This command retrieves a current dynamic display value from the microFlow.net. Information available includes current and load average batch data.

Command:

"DY_ddxx"

Where: dd = dynamic display type
- SY = System
- IN = Injector
- Bz = Batch, where "z" is 1...9 for batches 1-9, and "A" for batch 10
- CB = Current Batch
- TR = Batch
- rr = Recipe where "rr" is 01...12 for recipes 1-12

xx = variable number to access
- System (00 - 12)
- Injector (00 - 07)
- Batch (00 - 19)
- Batch (00 - 15)
- Recipe (00 - 05)

Responses:

Good Response:

"DY_V..V_D..D"

Where: V..V = the value of the variable
D..D = description of the variable (may contain spaces)

or...

"NOXX" Requested display not returned.

Remarks: No response exceeds 31 characters.

Constraints: None.

Special Case: None.

Comm. Modes: No Control, Host Control, Poll and Program.

Batch "DY_BNXX" Dynamic Display Values

Index Number ("XX")	Description	Response	
00	Recipe Name and Number	DY Recipe XX	XXXXXXXXX
01	Raw Batch Volume	DY IV Batch	XXXXXXX.XX Gal
02	Gross Batch Volume	DY GV Batch	XXXXXXX.XX Gal
03	GST Batch Volume	DY GST Batch	XXXXXXX.XX Gal
04	GSV Batch Volume	DY GSV Batch	XXXXXXX.XX Gal
05	Mass Batch Total	DY Mass Batch	XXXXXXX.XX Gal
06	Batch Average Meter Factor	DY Batch Avg Mtr Factor	X.XXXXX
07	Batch Average Temperature	DY Batch Avg Temp	SXXXX.X F
08	Batch Average Density	DY Batch Avg Dens	XXXX.X Lb/F3
09	Batch Average API	DY API	SXXX.X
10	Batch Average Reference Density	DY Ref Dens	XXXX.X Lb/F3
11	Batch Average Relative Density	DY Rel Dens	X.XXXX
12	Batch Average Pressure	DY Batch Avg Press	XXXX.X PSI
13	Batch Average Vapor Pressure	DY Avg Vapor Press	XXXX.X PSI
14	Batch Average CTL	DY Batch Avg CTL	X.XXXXX
15	Batch Average CPL	DY Batch Avg CPL	X.XXXXX
16	Additive 1 Batch Total	DY Add 1 Batch	XXXXXXXX.XXX
17	Additive 2 Batch Total	DY Add 2 Batch	XXXXXXXX.XXX
18	Additive 3 Batch Volume	DY Add 3 Batch	XXXXXXXX.XXX
19	Additive 4 Batch Volume	DY Add 4 Batch	XXXXXXXX.XXX
20	Batch Average CTPL	DY Batch Avg CTPL	X.XXXXX
21*	Batch Average Rel Dens @ 60F	DY Avg Rel Dens @ 60F	X.XXXX

* E tables only

Recipe "DY_RRXX" Dynamic Display Values

Index Number ("XX")	Description	Response	
00	Recipe Name and Number	DY Recipe XX	XXXXXXXXX
01	Transaction Raw Volume	DY R#XX Raw Trans	XXXXXXX Gal
02	Transaction Gross Volume	DY R#XX Gross Trans	XXXXXXX Gal
03	Transaction GST Volume	DY R#XX GST Trans	XXXXXXX Gal
04	Transaction GSV Volume	DY R#XX GSV Trans	XXXXXXX Gal
05	Transaction Mass Volume	DY R#XX Mass Trans	XXXXXXX Gal

System "DY_SYXX" Dynamic Display Values

Index Number ("XX")	Description	Response
00	Current Flow Rate Units/Min	DY Flow XXXXX.X Gal/Min
01	Current Flow Rate Units/Hr	DY Flow XXXXXXX.X Gal/Hr
02	Current Recipe Name	DY Recipe XXXXXXXXX
03	Not Used	
04	Not Used	
05	Not Used	
06	Current Meter Factor	DY Current Meter Fact. X.XXXXX
07	Current Temperature	DY Current Temp. SXXXX.X F
08	Current Density	DY Current Density SXXXX.X Kg/M3
09	Current Pressure	DY Pressure XXXX.X PSI
10	Vapor Pressure	DY Avg. Vapor Press. XXXX.X PSI
11	Current Valve Requested Position	DY Valve Requested Closed
12	Time of Last Power Fail	DY PowerFail DD-MM-YY HH:MM:SS

Injector "DY_INXX" Dynamic Display Values

Index Number ("XX")	Description	Response
00	Injector 1 Current Pulse Rate	DY Inj 1 Cal XXXX.XX
01	Injector 2 Current Pulse Rate	DY Inj 2 Cal XXXX.XX
02	Injector 3 Current Pulse Rate	DY Inj 3 Cal XXXX.XX
03	Injector 4 Current Pulse Rate	DY Inj 4 Cal XXXX.XX
04	Injector 1 Programmed Pulse Rate	DY Inj 1 Prg XXXX.XX
05	Injector 2 Programmed Pulse Rate	DY Inj 2 Prg XXXX.XX
06	Injector 3 Programmed Pulse Rate	DY Inj 3 Prg XXXX.XX
07	Injector 4 Programmed Pulse Rate	DY Inj 4 Prg XXXX.XX

TR (Batch) "DY_TRXX" Dynamic Display Values

Index Number ("XX")	Description	Response	
00	Recipe Name and Number	DY Multiple Recipes	
01	Raw Trans Volume	DY Raw Trans	XXXXXXX.XX Gal
02	Gross Trans Volume	DY Gross Trans	XXXXXXX.XX Gal
03	GST Trans Volume	DY GST Trans	XXXXXXX.XX Gal
04	GSV Trans Volume	DY GSV Trans	XXXXXXX.XX Gal
05	Mass Trans Total	DY Mass Trans	XXXXXXX.XX Gal
06	Trans Average Meter Factor	DY Trans Avg Mtr Factor	X.XXXXX
07	Trans Average Temperature	DY Trans Avg Temp	SXXXX.X F
08	Trans Average Density	DY Trans Avg Dens	XXXX.X Lb/F3
09	Trans Average Pressure	DY Trans Avg Pres	XXXX.X PSI
10	Trans Average CTL	DY Trans Avg CTL	X.XXXX
11	Trans Average CPL	DY Trans Avg CPL	X.XXXX
12	Additive 1 Trans Volume	DY Add 1 Trans	XXXXXXXX.XXX
13	Additive 2 Trans Volume	DY Add 2 Trans	XXXXXXXX.XXX
14	Additive 3 Trans Volume	DY Add 3 Trans	XXXXXXXX.XXX
15	Additive 4 Trans Volume	DY Add 4 Trans	XXXXXXXX.XXX

Command Code EA – Enquire Alarms

This command requests the alarm status from the microFlow.net. Data is returned in a bit-mapped format.

Command:

"EA_DD" (Request alarm status of directory)

Where DD = Alarm type
SY = System
IN = Injector

Responses:

"A1A2A3 ... A9" Good Response. Nine characters for system
"A1A2A3 ... A10" Good Response. Ten characters for injectors

or...

"NOXX" Alarm status cannot be reported.

Remarks: Allow for additional characters to be added to the end when alarms are added in the future.

Constraints: None.

Special Case: None.

Comm. Modes: No Control, Host Control, Poll and Program.

Response to System Enquire Alarms Command, Character A1

Character		System Alarm Requests			
Char	Hex	RAM Corrupt (DA)	Flash Error (DA)	RAM Bad (DA)	ROM Bad (DA)
0	30				
1	31				X
2	32			X	
3	33			X	X
4	34		X		
5	35		X		X
6	36		X	X	
7	37		X	X	X
8	38	X			
9	39	X			X
:	3A	X		X	
;	3B	X		X	X
<	3C	X	X		
=	3D	X	X		X
>	3E	X	X	X	
?	3F	X	X	X	X

Response to Command Code EA – Character A1 (System)

Response to System Enquire Alarms Command, Character A2

Character		System Alarm Requests			
Char	Hex	Passcode Reset (DA)	System Program Error (DA)	Watchdog (DA)	Finish Backup Bad (DA)
0	30				
1	31				X
2	32			X	
3	33			X	X
4	34		X		
5	35		X		X
6	36		X	X	
7	37		X	X	X
8	38	X			
9	39	X			X
:	3A	X		X	
;	3B	X		X	X
<	3C	X	X		
=	3D	X	X		X
>	3E	X	X	X	
?	3F	X	X	X	X

Response to Command Code EA – Character A2 (System)

Response to System Enquire Alarms Command, Character A3

Character		System Alarm Requests			
Char	Hex	User Alarm 3 (U3)	User Alarm 2 (U2)	User Alarm 1 (U1)	Power-Fail Alarm (PA)
0	30				
1	31				X
2	32			X	
3	33			X	X
4	34		X		
5	35		X		X
6	36		X	X	
7	37		X	X	X
8	38	X			
9	39	X			X
:	3A	X		X	
;	3B	X		X	X
<	3C	X	X		
=	3D	X	X		X
>	3E	X	X	X	
?	3F	X	X	X	X

Response to Command Code EA – Character A3 (System)

Response to System Enquire Alarms Command, Character A4

Character		System Alarm Requests			
Char	Hex	Ticket Alarm (TK)	Communications (CM)	User Alarm 5 (U5)	User Alarm 4 (U4)
0	30				
1	31				X
2	32			X	
3	33			X	X
4	34		X		
5	35		X		X
6	36		X	X	
7	37		X	X	X
8	38	X			
9	39	X			X
:	3A	X		X	
;	3B	X		X	X
<	3C	X	X		
=	3D	X	X		X
>	3E	X	X	X	
?	3F	X	X	X	X

Response to Command Code EA – Character A4 (System)

Response to System Enquire Alarms Command, Character A5

Character		System Alarm Requests			
Char	Hex	Pulse Security (PS)	Add Clean Line (CA)	Overrun Alarm (OA)	Zero Flow Alarm (ZF)
0	30				
1	31				X
2	32			X	
3	33			X	X
4	34		X		
5	35		X		X
6	36		X	X	
7	37		X	X	X
8	38	X			
9	39	X			X
:	3A	X		X	
;	3B	X		X	X
<	3C	X	X		
=	3D	X	X		X
>	3E	X	X	X	
?	3F	X	X	X	X

Response to Command Code EA – Character A5 (System)

Response to System Enquire Alarms Command, Character A6

Character		System Alarm Requests			
Char	Hex	Density Trans (DR)	Temp Probe (TP)	Back Pressure (BP)	Valve Fault (VF)
0	30				
1	31				X
2	32			X	
3	33			X	X
4	34		X		
5	35		X		X
6	36		X	X	
7	37		X	X	X
8	38	X			
9	39	X			X
:	3A	X		X	
;	3B	X		X	X
<	3C	X	X		
=	3D	X	X		X
>	3E	X	X	X	
?	3F	X	X	X	X

Response to Command Code EA – Character A6 (System)

Response to System Enquire Alarms Command, Character A7

Character		System Alarm Requests			
Char	Hex	High Density (HD)	High Temp (HT)	High Flow (HF)	Pressure Trans (PR)
0	30				
1	31				X
2	32			X	
3	33			X	X
4	34		X		
5	35		X		X
6	36		X	X	
7	37		X	X	X
8	38	X			
9	39	X			X
:	3A	X		X	
;	3B	X		X	X
<	3C	X	X		
=	3D	X	X		X
>	3E	X	X	X	
?	3F	X	X	X	X

Response to Command Code EA – Character A7 (System)

Response to System Enquire Alarms Command, Character A8

Character		System Alarm Requests			
Char	Hex	Low Density (LD)	Low Temp (LT)	Low Flow (LF)	High Pressure (HP)
0	30				
1	31				X
2	32			X	
3	33			X	X
4	34		X		
5	35		X		X
6	36		X	X	
7	37		X	X	X
8	38	X			
9	39	X			X
:	3A	X		X	
;	3B	X		X	X
<	3C	X	X		
=	3D	X	X		X
>	3E	X	X	X	
?	3F	X	X	X	X

Response to Command Code EA – Character A8 (System)

Response to System Enquire Alarms Command, Character A9

Character		System Alarm Requests			
Char	**Hex**	**Mass Meter Tube (MT)**	**Mass Meter Overdrive (MO)**	**Mass Meter Comm Fail (MC)**	**Low Pressure (LP)**
0	30				
1	31				X
2	32			X	
3	33			X	X
4	34		X		
5	35		X		X
6	36		X	X	
7	37		X	X	X
8	38	X			
9	39	X			X
:	3A	X		X	
;	3B	X		X	X
<	3C	X	X		
=	3D	X	X		X
>	3E	X	X	X	
?	3F	X	X	X	X

Response to Command Code EA – Character A9 (System)

Response to System Enquire Alarms Command, Character A10

Character		System Alarm Requests			
Char	Hex	Not Used	Not Used	Shared Printer (SP)	PTB Printer Failure (PP)
0	30				
1	31				X
2	32			X	
3	33			X	X
4	34		X		
5	35		X		X
6	36		X	X	
7	37		X	X	X
8	38	X			
9	39	X			X
:	3A	X		X	
;	3B	X		X	X
<	3C	X	X		
=	3D	X	X		X
>	3E	X	X	X	
?	3F	X	X	X	X

Response to Command Code EA – Character A10 (System)

Response to Injector Enquire Alarms Command, Character A1

Character		Additive Alarm Requests			
Char	Hex	Injector 4 Comm Error	Injector 3 Comm Error	Injector 2 Comm Error	Injector 1 Comm Error
0	30				
1	31				X
2	32			X	
3	33			X	X
4	34		X		
5	35		X		X
6	36		X	X	
7	37		X	X	X
8	38	X			
9	39	X			X
:	3A	X		X	
;	3B	X		X	X
<	3C	X	X		
=	3D	X	X		X
>	3E	X	X	X	
?	3F	X	X	X	X

Response to Command Code EA – Character A1 (Injector)

Response to Injector Enquire Alarms Command, Character A2

Character		Additive Alarm Requests			
Char	Hex	Injector 4 Cmd Refused	Injector 3 Cmd Refused	Injector 2 Cmd Refused	Injector 1 Cmd Refused
0	30				
1	31				X
2	32			X	
3	33			X	X
4	34		X		
5	35		X		X
6	36		X	X	
7	37		X	X	X
8	38	X			
9	39	X			X
:	3A	X		X	
;	3B	X		X	X
<	3C	X	X		
=	3D	X	X		X
>	3E	X	X	X	
?	3F	X	X	X	X

Response to Command Code EA – Character A2 (Injector)

Response to Injector Enquire Alarms Command, Character A3

Character		Additive Alarm Requests			
Char	Hex	Injector 4 Feedback Error	Injector 3 Feedback Error	Injector 2 Feedback Error	Injector 1 Feedback Error
0	30				
1	31				X
2	32			X	
3	33			X	X
4	34		X		
5	35		X		X
6	36		X	X	
7	37		X	X	X
8	38	X			
9	39	X			X
:	3A	X		X	
;	3B	X		X	X
<	3C	X	X		
=	3D	X	X		X
>	3E	X	X	X	
?	3F	X	X	X	X

Response to Command Code EA – Character A3 (Injector)

Response to Injector Enquire Alarms Command, Character A4

Character		Additive Alarm Requests			
Char	Hex	Injector 4 General Error	Injector 3 General Error	Injector 2 General Error	Injector 1 General Error
0	30				
1	31				X
2	32			X	
3	33			X	X
4	34		X		
5	35		X		X
6	36		X	X	
7	37		X	X	X
8	38	X			
9	39	X			X
:	3A	X		X	
;	3B	X		X	X
<	3C	X	X		
=	3D	X	X		X
>	3E	X	X	X	
?	3F	X	X	X	X

Response to Command Code EA – Character A4 (Injector)

Response to Injector Enquire Alarms Command, Character A5

Character		Additive Alarm Requests			
Char	Hex	Injector 4 Low Additive	Injector 3 Low Additive	Injector 2 Low Additive	Injector 1 Low Additive
0	30				
1	31				X
2	32			X	
3	33			X	X
4	34		X		
5	35		X		X
6	36		X	X	
7	37		X	X	X
8	38	X			
9	39	X			X
:	3A	X		X	
;	3B	X		X	X
<	3C	X	X		
=	3D	X	X		X
>	3E	X	X	X	
?	3F	X	X	X	X

Response to Command Code EA – Character A5 (Injector)

Response to Injector Enquire Alarms Command, Character A6

Character		Additive Alarm Requests			
Char	Hex	Injector 4 Excess Pulses	Injector 3 Excess Pulses	Injector 2 Excess Pulses	Injector 1 Excess Pulses
0	30				
1	31				X
2	32			X	
3	33			X	X
4	34		X		
5	35		X		X
6	36		X	X	
7	37		X	X	X
8	38	X			
9	39	X			X
:	3A	X		X	
;	3B	X		X	X
<	3C	X	X		
=	3D	X	X		X
>	3E	X	X	X	
?	3F	X	X	X	X

Response to Command Code EA – Character A6 (Injector)

Response to Injector Enquire Alarms Command, Character A7

Character		Additive Alarm Requests			
Char	Hex	Injector 4 No Pulses	Injector 3 No Pulses	Injector 2 No Pulses	Injector 1 No Pulses
0	30				
1	31				X
2	32			X	
3	33			X	X
4	34		X		
5	35		X		X
6	36		X	X	
7	37		X	X	X
8	38	X			
9	39	X			X
:	3A	X		X	
;	3B	X		X	X
<	3C	X	X		
=	3D	X	X		X
>	3E	X	X	X	
?	3F	X	X	X	X

Response to Command Code EA – Character A7 (Injector)

Response to Injector Enquire Alarms Command, Character A8

Character		Additive Alarm Requests			
Char	Hex	Injector 4 Over Rev	Injector 3 Over Rev	Injector 2 Over Rev	Injector 1 Over Rev
0	30				
1	31				X
2	32			X	
3	33			X	X
4	34		X		
5	35		X		X
6	36		X	X	
7	37		X	X	X
8	38	X			
9	39	X			X
:	3A	X		X	
;	3B	X		X	X
<	3C	X	X		
=	3D	X	X		X
>	3E	X	X	X	
?	3F	X	X	X	X

Response to Command Code EA – Character A8 (Injector)

Response to Injector Enquire Alarms Command, Character A9

Character		Additive Alarm Requests			
Char	Hex	Injector 4 Frequency	Injector 3 Frequency	Injector 2 Frequency	Injector 1 Frequency
0	30				
1	31				X
2	32			X	
3	33			X	X
4	34		X		
5	35		X		X
6	36		X	X	
7	37		X	X	X
8	38	X			
9	39	X			X
:	3A	X		X	
;	3B	X		X	X
<	3C	X	X		
=	3D	X	X		X
>	3E	X	X	X	
?	3F	X	X	X	X

Response to Command Code EA – Character A9 (Injector)

Response to Injector Enquire Alarms Command, Character A10

Character		Additive Alarm Requests			
Char	Hex	Injector 4 Unauth Failed	Injector 3 Unauth Failed	Injector 2 Unauth Failed	Injector 1 Unauth Failed
0	30				
1	31				X
2	32			X	
3	33			X	X
4	34		X		
5	35		X		X
6	36		X	X	
7	37		X	X	X
8	38	X			
9	39	X			X
:	3A	X		X	
;	3B	X		X	X
<	3C	X	X		
=	3D	X	X		X
>	3E	X	X	X	
?	3F	X	X	X	X

Response to Command Code EA – Character A10 (Injector)

Command Code EQ – Enquire Status

This command retrieves the operational status of microFlow.net. Data is returned in a bit-mapped format.

Command:

"EQ" Request Status

Responses:

"A1A2A3A4A5" **Good Response.** 5 Characters. For descriptions of each of the characters, see the following pages.

Where each "Ax" is a quasi hex value;
"0 1 2 3 4 5 6 7 8 9 : ; < > ?".

Remarks: Allow for additional characters to be added on the end for future status indicators.

Constraints: None.

Special Case: See notes under tables. Character A6 implemented in revision 00.13.

Comm. Modes: No Control, Host Control, Poll and Program.

Response to "EQ" Enquire, Character A1

Character		Condition			
Char	**Hex**	**Program Mode**	**Released**	**Flowing**	**Authorized**
0	30				
1	31				X
2	32			X	
3	33			X	X
4	34		X		
5	35		X		X
6	36		X	X	
7	37		X	X	X
8	38	X			
9	39	X			X
:	3A	X		X	
;	3B	X		X	X
<	3C	X	X		
=	3D	X	X		X
>	3E	X	X	X	
?	3F	X	X	X	X

Response to Command Code EQ – Character A1

Note: *X shows an asserted condition. The microFlow.net is considered released whenever the valve is opened and has not been commanded to close.*

Response to "EQ" Enquire, Character A2

Character		Condition			
Char	**Hex**	**Transaction in Progress**	**Transaction Done**	**Batch Done**	**Keypad Data Pending**
0	30				
1	31				X
2	32			X	
3	33			X	X
4	34		X		
5	35		X		X
6	36		X	X	
7	37		X	X	X
8	38	X			
9	39	X			X
:	3A	X		X	
;	3B	X		X	X
<	3C	X	X		
=	3D	X	X		X
>	3E	X	X	X	
?	3F	X	X	X	X

Response to Command Code EQ – Character A2

Note: *X shows an asserted condition. Some alarm conditions cannot be reset through the communications channel. (See Alarm Reset command "AR.")*

Response to "EQ" Enquire, Character A3

Character		Condition			
Char	**Hex**	**Printing in Progress**	**Premissive Delay**	**New Card Data Available**	**Alarm**
0	30				
1	31				X
2	32			X	
3	33			X	X
4	34		X		
5	35		X		X
6	36		X	X	
7	37		X	X	X
8	38	X			
9	39	X			X
:	3A	X		X	
;	3B	X		X	X
<	3C	X	X		
=	3D	X	X		X
>	3E	X	X	X	
?	3F	X	X	X	X

Response to Command Code EQ – Character A3

Note: *X shows an asserted condition. Some alarm conditions cannot be reset through the communication channel. (See Alarm Reset command "AR.")*

Response to "EQ" Enquire, Character A4

Character		Condition			
Char	**Hex**	**Program Value Changed**	**Delayed Prompt in Effect**	**Display Message Time-Out**	**Power-Fail Occurred**
0	30				
1	31				X
2	32			X	
3	33			X	X
4	34		X		
5	35		X		X
6	36		X	X	
7	37		X	X	X
8	38	X			
9	39	X			X
:	3A	X		X	
;	3B	X		X	X
<	3C	X	X		
=	3D	X	X		X
>	3E	X	X	X	
?	3F	X	X	X	X

Response to Command Code EQ – Character A4

Note: *X shows an asserted condition.*

Response to "EQ" Enquire, Character A5

Character		Condition			
Char	**Hex**	**Checking Entries**	**Input #1**	**Input #1**	**Input #1**
0	30				
1	31				X
2	32			X	
3	33			X	X
4	34		X		
5	35		X		X
6	36		X	X	
7	37		X	X	X
8	38	X			
9	39	X			X
:	3A	X		X	
;	3B	X		X	X
<	3C	X	X		
=	3D	X	X		X
>	3E	X	X	X	
?	3F	X	X	X	X

Response to Command Code EQ – Character A5

Note: *X shows an asserted condition (= AC input at contact).*

Response to "EQ" Enquire, Character A6

Character		Condition			
Char	**Hex**	**Pending Reports**	**Pending Repeat Storage Full**	**Printer Standby**	**Presetting in Progress**
1	31				X
2	32			X	
3	33		X		
4	34	X			

Response to Command Code EQ – Character A6

Note: *X shows an asserted condition (Rev. 00.13 and above).*

Command Code ER – Event Recall

This command retrieves historical data using the sequence number of the data.

Command:

"ER_S...S"

Where S...S is the sequence number.

Responses:

Good Response
"ER_SSSSSSSSSS_DDDDDD_HHNN_X_EEEEE_A...A"
Where:

SSSSSSSSSS	= Sequence number
DDDDDDDD	= Standard Time 'MMDDYYYY' or Military Time "DDMMYYYY"
MM	= Month
DD	= Day
YYYY	= Year
HH	= Hours
NN	= Minutes
X	= A (Standard Time – AM), P (Standard Time – PM), M (Military Time)
EEEEE	= Type Number (Returned, but not currently used)
A....A	= Data Variable length string (may contain "tab" characters)

or...

"NOXX" Data not retrieved.

Remarks: None.

Constraints: None.

Special Case: None.

Comm. Modes: Host Control, No Control, Poll and Program.

Command Code FL – Read Flow Count

This command retrieves raw pulse counts from the microFlow.net.

Command:

"FL" Read flow count for the arm.

Responses:

"FL_VVVVVVVVV" Good Response. Flow count for arm.

Remarks: None.

Constraints: VVVVVVVVV is unfactored raw pulse count. This value is reset to zero at start and end of each batch.

Special Case: None.

Comm. Modes: No Control, Host Control, Poll and Program.

Command Code GD – Get Date and Time

This command retrieves the current date and time from the microFlow.net.

Command:

"GD"

Responses:

"GD_DDDDDDDD_HHNN_X" Good Response.

Where: DDDDDDDD = MMDDYYYY (Standard Time)
= DDMMYYYY (Military Time)

HH	= hours
NN	= minutes
MM	= month
DD	= day
YYYY	= year
X	= A (Standard Time – A.M.)
	= P (Standard Time – P.M.)
	= M (Military Time)

or...

"NOXX" The date and time were not read from the microFlow.net.

Remarks: None.

Constraints: None.

Special Case: None.

Comm. Modes: No Control, Host Control, Poll and Program.

Command Code GP – Get Firmware Signatures

This command retrieves the computed CRC-32 for the firmware currently installed in the microFlow.net.

Command:

"GP"

Responses:

"GP_SSSSSSSS" Good Response.

Where: SSSSSSSS = eight hexadecimal digit signature

or...

"NOXX" Did not read the CRC signatures.

Remarks: CRC signatures may be used to determine firmware revision number. Contact the factory with inquiries.

Constraints: None.

Special Case: None.

Comm. Modes: No Control, Host Control, Poll and Program.

Command Code IC – Injector Command

This command is used to specify the command to be sent to a smart additive injector.

Command:

"IC_c..c"

Where **IC = Injector Command literal**
c..c = command text (as shown below)

c..c = **AAAY...Y**
where: AAA **=** Injector comm address (ASCII)
Y...Y = Injector command (ASCII)

(Gate City Protocol I and Titan PAC-3 Protocol)

Note: *The microFlow.net installs an STX before c..c, an ETX and calculated LRC after c..c, and then sends this out the injector comm port.*

Responses:

"OK" Good Response.

or...

"NOXX"

Where X = "01" In the Programming Mode
= "02" microFlow.net Released
= "06" Operation Not Allowed
= "19" Option Not Installed

When the microFlow.net receives an "IC" command, a "NO06" response will be moved into the appropriate response buffer. This "NO06" response will indicate that no response has yet been received from the Additive Injector Subsystem. This will prevent the supervisory computer from issuing an immediate "IR" command and reading an old response from a previous command that may have been issued to a different additive subsystem.

If communications with an additive subsystem has not been selected in the programming mode, this command will return a "NO" response.

If the specified command is valid for the microFlow.net's current mode of operation, the query or command is copied into the additive command buffer. If the address is invalid, i.e., there is no additive injector with that address connected to the microFlow.net, no response will ever be received. If the command is improperly constructed or not valid for the type of additive injector selected, the additive injector will respond appropriately and the supervisory computer can decode the response given.

STX and ETX characters are added to the command. The LRC is calculated and added onto the end of the command.

Command Code IR – Injector Response

This command is used to retrieve the response from a passby command to an Additive Injection Subsystem.

Command:

"IR"

Responses:

"IR_r..r" **Good Response**

Where IR = Injector Response command
r..r = response text (see below)

r..r of AAAY..Y
AAA = Injector comm address (ASCII)
Y..Y = Injector response (ASCII)

(Gate City Protocol I and Titan PAC-3 Protocol)

Note: *The microFlow.net receives the response from the injector, then strips off the STX, ETX, and LRC, and what remains is "r..r" above.*

or...

"NOXX"

Remarks: None.

Constraints: None.

Special Case: None.

Comm. Modes: Host Control, No Control, Poll and Program.

Command Code LD – Request Load Average Density

This command requests load average density from the microFlow.net.

Command:

Current Batch

"LD _R" Current batch average density

"LD _OI_NNN" Batch average density of a complete batch

Where R = Constant (Current Recipe)
NNN = Number of batches back into Local Storage to retrieve data

Responses:

Current Batch

"LD _OI_RR_SVVVV.V" Good Response. For commands LD_R, LO_OI

"LD _OI_RR_SVVVV.V_NNN" Good Response. For commands LD_OI_NNN

Where:
VVVV.V = Average Value
NNN = Number of batches back into Local Storage to retrieve data
RR = Recipe Number (01-12)
S = Sign (+ or -)

or...

"NOXX" The load average density was not read.

Remarks: Response field padded with leading spaces. If value is negative, minus sign will immediately precede most significant digit.

Constraints: Density units are as programmed in the microFlow.net.

Special Case: None.

Comm. Modes: No Control, Host Control, Poll and Program.

Command Code LO – Request Logout of Program Mode

This command forces an immediate logout of Program Mode.

Command:

“LO”

Responses:

OK

or...

“NOXX”

Remarks: Ten seconds after issuing a PC command, the microFlow.net begins the logout process. The “LO” command starts it immediately. All changes made by the PC command are not available (made active) until the logout process is complete.

Constraints: Logout may only be forced if the comm port is the one logged in (i.e., Port #1 cannot logout Port #2).

Special Case: None.

Comm. Modes: Host Control, Poll and Program.

Command Code LP – Request Load Average Pressure

This command requests the value of the load average pressure from the microFlow.net.

Command:

Current Batch

"LP _R" Current load average pressure for the current recipe.

"LP_OI_NNN" Load average pressure for the batch indicated in local storage.

Where R = Constant (Current Recipe)
NNN = Number of batches back into Local Storage to retrieve data

Responses:

Current Batch

"LP_OI_RR_VVVV.V" For "LP_R" and "LP_OI"

"LP_OI_RR_VVVV.V_NNN" For "LP_OI_NNN"

Where:
RR = Recipe (01-12)
VVVV.V = average value
NNN = number of batches back into Local Storage to retrieve data

or...

"NOXX" The load average density was not read.

Remarks: None.

Constraints: Pressure units are as programmed for the microFlow.net.

Special Case: None.

Comm. Modes: No Control, Host Control, Poll and Program.

Command Code LT – Request Load Average Temperature

This command requests the value of the load average temperature from the microFlow.net.

Command:

Current Batch

"LT _R" Current batch average
"LT_OI_NNN" Batch average of completed batch

Where R = Constant (Current Recipe)
YY = Batch Number (01-50)
NNN = Number of batches back into Local Storage to retrieve data

Responses:

Current Batch

"LT _OI_RR_VVVV.V" For command(s) "LT_R" and "LT_OI"
"LT _OI_RR_VVVV.V_NNN" For "LT_OI_NNN"

Where:
RR = Recipe (01-12)
VVVV.V = average value
NNN = number of transactions back into Local Storage to retrieve data

or...

"NOXX" The load average temperature was not read.

Remarks: None.

Constraints: Temperature units are as programmed for the microFlow.net. Negative temperature is possible.

Special Case: None.

Comm. Modes: No Control, Host Control, Poll and Program.

Command Code MC – Mass Meter Send

This command allows a host system to send commands to a mass meter connected to an microFlow.net.

Command:

"MC_m...m"

Where m...m = Mass Meter command text as defined in MN0M015LX or MN0M016LX for the S-Mass, or MN0M012LX for the Apollo.

Responses:

"OK" Good Response.

or...

"NOXX" The command was not sent to the Mass Meter.

Where

XX = "01" if the microFlow.net is in Program Mode
= "02" if the microFlow.net is released
= "19" if no Mass Meter comm port is defined

Remarks: See "MR" for retrieval of the response from the mass meter.

Constraints: None.

Special Case: None.

Comm. Modes: No Control, Host Control, Poll and Program.

Command Code MR – Mass Meter Receive

This command allows a host system to receive commands from a mass meter connected to a microFlow.net.

Command:

"MR"

Responses:

"MR_r...r" **Good Response.**

Where

r...r = Mass Meter response text as defined in MN0M015LX or MN0M016LX for the S-Mass, or for MN0M012LX for the Apollo.

or...

"NOXX" No response is available from the mass meter.

Where

XX = "06" when no data was received from the Mass Meter

Remarks: See "MC" for a description of how to send a command to the mass meter.

Constraints: None.

Special Case: None.

Comm. Modes: No Control, Host Control, Poll and Program.

Command Code OR – Output Relay

This command will activate or deactivate a general purpose relay output.

Command:

"OR_XX_Y"

Where XX = the output number (01-06)
Y = desired state (1 = on, 0 = off)

Responses:

"OK" **Good Response.** The command was accepted and the desired state was output to the selected contact.

or...

"NOXX" The command was rejected. The microFlow.net did not request a state change at the selected output.

Remarks: "NO03" will be returned if XX or Y is out of range. "NO06" will be returned if the output is not assigned as a general purpose output.

Constraints: This command will not be allowed if the corresponding relay is not configured as a general purpose relay.

Special Case: None.

Comm. Modes: No Control, Host Control, Poll and Program.

Command Code PC – Change Program Code Values

This command instructs the microFlow.net to change the value of one of the programmable entries (001 through 999) in the directory specified.

Command:

"PC_DD_XXX_V...V"

Where DD = Major Directory
= CF – Configuration
= SY – System
= 01-12 – Recipe Number
XXX = Program Parameter Number
V...V = New Value, content depends on parameter

Responses:

"OK" Good Response. Parameter value has been changed.

or...

"NOXX" The program value was not changed.

Remarks: The number of digits or alpha characters entered for the new program code must be EXACTLY equal to the number of digits or alpha characters required for that particular program code, except for codes requiring text strings.

Due to the varying lengths of the programmable display messages, the number of digits or alpha characters entered for the new program code can number up to a maximum of 30. However, the number of digits or alpha characters stored will depend on the maximum length of that particular message being changed.

Constraints: Refer to the Reference section in the Operators Manual for a complete list of parameters in each directory.

Special Case: None.

Comm. Modes: Host Control, Poll and Program.

Note: *The "+" argument appended to the PC command string affects the number of significant digits returned for floating point numbers. For the "+" version of the command, additional decimal digits may be included in the response beyond the specified format for the program code if they are non-zero (up to a maximum of six total digits to the right of the decimal point).*

Examples

01PV 01 011	PV 01 011 0010.000 Inj #1 Vol
01PC 01 011 23.3604	PC 01 011 0023.360 Inj #1 Vol
01PV 01 011	PV 01 011 0023.360 Inj #1 Vol
01PV 01 011+	PV 01 011 23.360400 Inj #1 Vol

Command Code PF – Request Time of Power-Fail

This command requests the time and date the last power-fail occurred.

Command:

"PF"

Responses:

"PF_DDDDDDDD_HHNN_X" Good Response.

Where DDDDDDDD = Power-Fail Date
= (MMDDYYYY for Standard Time)
= (DDMMYYYY for Military Time)
HH = Power-Fail Time, Hours
NN = Power-Fail Time, Minutes
X = A (Standard Time – A.M.)
= P (Standard Time – P.M.)
= M (Military Time)

Remarks: None.

Constraints: None.

Special Case: None.

Comm. Modes: No Control, Host Control, Poll and Program.

Command Code PP – Print Report to Printer

This command initiates a reprint of the requested transaction at the printer for the arm.

Command:

"PP"	for the most recently completed transaction
"PP NNN"	for NNN transactions back in local storage
"PP ST"	for PTB standby to start re-printing all outstanding reports

Responses:

Good Response:

"OK"

or...

"NOXX"	No pending repeats to print

Remarks: None.

Constraints: A printer port must be configured.

Special Case: None.

Comm. Modes: No Control, Host Control, Poll and Program.

Command Code PR – Program Change Recall

This command retrieves program mode changes from the microFlow.net's internal log using the sequence number of the data.

Command:

"PR_S...S"

Where S...S is the sequence number.

Responses:

Good Response:
"PR_SSSSSSSSSS_DDDDDD_HHNN_X_A...A"
Where:

SSSSSSSSSS	= Sequence number
DDDDDDDD	= Standard Time "MMDDYYYY" or Military Time "DDMMYYYY"
MM	= Month
DD	= Day
YYYY	= Year
HH	= Hours
NN	= Minutes
X	= A (Standard Time – A.M.), P (Standard Time – P.M.) M (Military Time)

A...A = WW_ZZ <Tab> Old Data <Tab> New Data
WW = Directory Designation
CF = Configuration
SY = System
Rnn = Where nn = recipe number (01-12)
ZZZ = Program Code Number

or...

"NOXX" Data not retrieved.

Remarks: None.

Constraints: None.

Special Case: None.

Comm. Modes: Host Control, No Control, Poll and Program.

Command Code PS – Last Program Code Change Recall

This command retrieves the last changed program codes sequence number.

Command:

"PS"

Responses:

"PS_SSSSSSSSSS" Good Response.

Where:
SSSSSSSSSS = Sequence number

or...

"NOXX" Sequence number not retrieved.

Remarks: None.

Constraints: None.

Special Case: None.

Comm. Modes: Host Control, No Control, Poll and Program.

Command Code PT – Print Transaction to Host

This command allows a transaction report to be generated directly to the host over the existing communications line. The microFlow.net first responds with an OK response (framed normally according to the current host protocol) followed by the report text. No additional framing characters appear before, during, or after the report text other than those returned with the normal OK response.

Command:

"PT"	for the most recently completed transaction
"PT NNN"	for NNN transactions back in local storage

Responses:

Good Response:

"OK" (followed by the report text)

or...

"NOXX"

Remarks: None.

Constraints: None.

Special Case: None.

Comm. Modes: No Control, Host Control, Poll and Program.

Command Code PV – Request Program Code Values

This command requests program values from the microFlow.net.

Command:

"PV_DD_XXX"

Where DD = Major Directory
CF = Configuration
SY = System
01-12 = Recipe Number

XXX = Program Code Number

Responses:

"PV_DD_XXX_A...A" Good Response for three-digit codes

Where DD = Directory
CF = Configuration
SY = System
01-12 = Recipe Number

XXX = Parameter Number
A...A = Value of the parameter

or...

"NOXX" Program value not read.

Remarks: None.

Constraints: None.

Special Case: None.

Comm. Modes: No Control, Host Control, Poll and Program.

Note: *The "+" argument appended to the PC command string affects the number of significant digits returned for floating point numbers. For the "+" version of the command, additional decimal digits may be included in the response beyond the specified format for the program code if they are non-zero (up to a maximum of six total digits to the right of the decimal point).*

Examples

01PV 01 011	PV 01 011 0010.000 Inj #1 Vol
01PC 01 011 23.3604	PC 01 011 0023.360 Inj #1 Vol
01PV 01 011	PV 01 011 0023.360 Inj #1 Vol
01PV 01 011+	PV 01 011 23.360400 Inj #1 Vol

Command Code RA – Request Alarm Status

This command requests program values from the microFlow.net.

Command:

"RA_DD"

Where DD = Directory
SY = System
IN = Injector

Responses:

"A1 A2 A3 A4 A5" Good Response.

"OK" OK is returned if there are no alarms for that directory

"NOXX" Bad Response

Remarks: The good response is a character string consisting from 1 to 5 status codes separated by a single space. Each status code is two characters, see AR for Alarm Mnemonics.

If alarms exist for any injector, the two-character alarm code will be included in the response string. To determine the specific injector experiencing the alarm condition, the EA command must be used.

Constraints: None.

Special Case: If no alarm condition is set, an "OK" response is issued..

Comm. Modes: No Control, Host Control, Poll and Program.

Command Code RD – Request Analog Input Value

This command requests the value of one of the analog inputs installed and wired to the microFlow.net.

Command:

"RD_X" Request transducer value.

Where X = T (current temperature)
= P (current pressure)
= D (current density)

Responses:

"RD_X_VVVV.V" **Good Response.** Pressure or density, straight product.
"RD_X_SVVVV.V" **Good Response.** Temperature, straight product.

Where X = The number of the analog input
O = RTD
1 = 4 – 20mA
VVVV.V = Currrent value of the analog input
S = Sign (+ or -)

or...

"NOXX" The value was not read.

Remarks: None.

Constraints: None.

Special Case: None.

Comm. Modes: No Control, Host Control, Poll and Program.

Command Code RE – Reset Status Conditions

This command resets or acknowledges pending status conditions of the microFlow.net.

Command:

"RE_XX"

Where XX = the status condition to be reset.
PF = Power Fail
PC = Program code value has changed

Responses:

"OK" **Good Response.** Status condition has been reset. Status condition will not longer appear in response to status requests (EQ and RS).

or...

"NOXX" The status condition has not been reset.

Remarks: None.

Constraints: None.

Special Case: None.

Comm. Modes: No Control, Host Control, Poll and Program.

Note: *If the status code is already reset, a "NO06" will be returned.*

Command Code RL – Show Recipes Loaded

This command requests a bit map of recipes that have been loaded in the current or previous transaction.

Command:

Current Transaction
"RL" (Requests recipes loaded in the transaction)

Local Storage Transaction
"RL_NNN" (Requests recipes loaded in an historical transaction)
Where NNN = number of transactions back into local storage

Responses:

Current Transaction
"RL_C1C2C3" **Good Response.** Bit map of the loaded recipes

Local Storage Transaction
"RL_C1C2C3_NNN" **Good Response.** Bit map of the loaded recipes

Where C1...C3 are bit mapped characters indicating recipes 01-12 (See command AB for tables.)
NNN = number of transaction back into local storage.

or...

"NOXX" The bit map has not been returned.

Remarks: The recipes loaded bitmap will be cleared on authorization of a transaction (if Host Control) or when the microFlow.net is released for delivery for the first batch of a transaction (Polling Only).

Constraints: NO05 will be returned if no transaction has ever been completed.
NO06 will be returned for a "RL" request of current transaction if microFlow.net is authorized prior to transaction being started and also while presetting first batch of transaction.

Special Case: None.

Comm. Modes: No Control, Host Control, Poll and Program.

Command Code RQ – Request Current Flow Rate

This command retrieves the current flow rate(s) from the microFlow.net.

Command:

"RQ" Flow rate for the current product or system flow rate.

Responses:

"RQ_XXXXX" **Good Response.** Current flow rate

where: XXXX = current flow rate
or...
"NOXX" Flow rate was not returned.

Remarks: None.

Constraints: NO31 will be returned if the command format is inconsistent with the currently configured mode of operations.

Special Case: None.

Comm. Modes: No Control, Host Control, Poll and Program.

Command Code RR – Request Recipe

This command requests the current recipe number from microFlow.net.

Command:

"RR"

Responses:

"RR_NN" Good Response.

Where NN = Recipe Number (01-12)

"NOXX" Recipe number not returned.

Remarks: The recipe number returned by the RR will remain in effect until another recipe is selected, either by allocating a single recipe (AB command) or by selection at the microFlow.net keypad. Neither batch done nor transaction done clears the recipe number from the "RR" response.

Constraints: NO05 will be returned if no transaction has ever been completed.

Special Case: None.

Comm. Modes: No Control, Host Control, Poll and Program.

Command Code RS – Request Status

This command requests the operational status of the microFlow.net. Data is returned as two-character mnemonics for each status reported. A maximum of twenty status codes will be reported.

Command:

"RS"	Request Status.

Responses:

"RS_XX_XX_XX_XX ...XX"	**Good Response.** A character string consisting of from 1 to 20 status codes separated by a single space. Each status code is two characters. See table on the following page for more information about status codes.

Remarks: The microFlow.net is considered released whenever the valve is open and has not been commanded to close. Some alarm conditions cannot be reset through the Communication channel. (See Alarm Reset command.)

Constraints: None.

Special Case: A trailing space is returned after the final status code.

Comm. Modes: No Control, Host Control, Poll and Program.

Request Status Codes

Code	Condition
AL	Alarm active
CE	Checking entries
FL	Flowing
I1	Input 1 on
I2	Input 2 on
I3	Input 3 on
LR	Pending (locked) reports
PC	Program parameter changed
PD	Permissive delay active
PF	Power fail occurred
PP	Printing in progress
PW	In program mode
RL	Reserved
TP	Batch in progress

Command Code RT – Request Batch Volume

This command requests the batch data from the microFlow.net.

Command:

Current Batch

"RT_X" Total volume of the current batch.

Local Storage Batch

"RT_X_NNN" Total volume of a complete batch.

Where:

X = R for raw total (Indicated Volume – IV)
= G for gross volume (GV)
= N for gross @ standard temperature volume (GST)
= P for gross @ standard temperature and pressure volume (GSV)
= M for mass total
= D for NSV

RR = Recipe 01 – 04

NNN = Number of batches back into local storage to retrieve data

Responses:

Current Batch

"RT_Z_YY_RR_VVVVVVVV" Good Response. Batch volume, recipe

Local Storage Batch

"RT_Z_YY_RR_VVVVVVVV_NNN" Good Response. Batch volume, recipe, historical

Where:

X = R for raw total (Indicated Volume – IV)
= G for gross volume (GV)
= N for gross @ standard temperature volume (GST)
= P for gross @ standard temperature and pressure volume (GSV)
= M for mass total
= D for NSV

RR = Recipe 01 – 12 or "MR" to indicate multiple recipes

VVVVVVVV = Total batch volume

NNN = Number of batches back into local storage

or...

"NOXX" No batch data was returned.

Remarks: For Total Volume Requests; (RT_Z and RT_Z_NNN) "MR" as the recipe number on the response indicates a multiple recipe transaction. Recipes delivered in batches can be determined via batch volume requests RB or by using the RL or RN commands.

Constraints: Transaction Volume Units are assumed to be as programmed into microFlow.net. NO03 will be returned for a recipe request if the recipe requested was not delivered in the transaction. NO30 will be returned if the recipe specified is not currently configured to the microFlow.net to which the command was directed.

Special Case: None.

Comm. Modes: No Control, Host Control, Poll and Program.

Command Code SB – Set Batch

This command authorizes a batch and presets Batch Volumes for a transaction.

Command:

"SB_VVVVVV"

"SB_A1_VVVVVV"

Where:

A1 = Additive selection code

VVVVVV = Volume to preset

Responses:

"OK" Good Response. Batch volume has been accepted.

or...

"NOXX" The batch volume has not been set.

Remarks: If the additive qualifier (A1) is not used, the additive selection will operate as currently programmed in the microFlow.net. For authorization with additives, only one recipe may be allotted. Additives selected in the qualifier must be programmed for use. If more than one recipe is enabled, the "Select Recipe" prompt will be displayed.

See also Remarks for "AB" – Allocate Blend Recipes command. (*)

See Command Codes AP and AU for a description of A1 characters.

Constraints: Batch volume must not exceed programmed maximum batch size and must not be below the programmed minimum batch size. Units value must correspond to what is programmed into microFlow.net for units of measurement.

Special Case: An authorization command with batch size of 0 allows the driver to select batch size. Driver may clear any preset batch size and enter a new batch volume providing that it is less than the preset batch size. A batch amount of zero while in the Auto Preset Mode will result in the programmed auto preset amount being displayed; a non-zero set batch amount will override the programmed auto preset amount.

Comm. Modes: Host Control.

Command Code SD – Set Date and Time

This command sets the date and time in the microFlow.net.

Command:

"SD_DDDDDDDD_HHNN_X"

Where	DDDDDDDD	= MMDDYYYY (Standard Time)
	DDDDDDDD	= DDMMYYYY (Military Time)
	MM	= month
	DD	= day
	YYYY	= year
	HH	= hours
	NN	= minutes
	X	= A (Standard Time – A.M.)
		= P (Standard Time – P.M.)
		= M (Military Time)

Responses:

"OK" **Good Response.** Time and date value accepted and seconds reset to zero.

or...

"NOXX" The time and date were not accepted.

Remarks: None.

Constraints: Time value must be within range programmed into microFlow.net – 0000 to 2359 for Military, 0000 to 1259 for Standard; month must be within the range of 1 to 12; day must be in the range valid for the month chosen.

Special Case: Leading spaces may be used in place of leading zeros for month, day, year, hours, and minutes. However, this is not recommended.

Comm. Modes: No Control, Host Control, Poll and Program.

Command Code TN – Show Batch Stop Date and Time

This command retrieves the batch stop date and time and internal batch number.

Command:

Current Batch
"TN"

Local Storage Batch
"TI_NNN"

Where: NNN = The number of transactions back into local storage to retrieve data.

Responses:

Current Batch
"TN_IIII_DDDDDDDD_HHNN_X" **Good Response.**

Local Storage Batch
"TN_IIII_DDDDDDDD_AAAA_X_NNN" **Good Response.**

Where:

IIII	= Internal Batch Number
DDDDDDDD	= Batch Stop Date
	= (MMDDYYYY for Standard Time)
	= (DDMMYYYY for Military Time)
HHNN	= Transaction Stop Time (HHMM)
X	= A (Standard Time – A.M.)
	= P (Standard Time – P.M.)
	= M (Military Time)
NNN	= Number of batches back

or...

"NOXX" The batch stop date and time were not retrieved.

Remarks: None.

Constraints: None.

Special Case: None.

Comm. Modes: No Control, Host Control, Poll and Program.

Command Code TR – Transaction Summary Recall

This command retrieves the historical transaction summary data from microFlow.net using the sequence number of the transaction.

Command:

"TR_S...S"

Responses:

"TR_SSSSSSSSSS" Good Response.

Where:
SSSSSSSSSS = Sequence Number
And [transaction data] is a comma-delimited text record
with the following fields:
Transaction Start Date/Time;
Transaction #,
Raw Card Data,
5 Numeric Prompt Responses,
5 Alphanumeric Prompt Responses,
Total Number of batches included in the transaction,
5 Volume Totals (IV, GV, GST, GSV, Mass),
4 Additive Totals (A1, A2, A3, A4),
6 Transaction Average Values (meter factor, temperature, density, pressure, CTL, CPL),
5 Nonresettable Totalizer Values (IV, GV, GST, GSV, Mass),
Driver Database User field 1 value associated with card,
Driver Database User field 2 value associated with card,
Driver Database User field 3 value associated with card,
HID Factory Code associated with card,
HID Number on card,
Number of Alarms occurring during the transaction
Alarm Codes for alarms occurring during transaction (text field)
Transaction End Time

or...

"NOXX" The transaction summary data was not retrieved.

Remarks: Some fields may be empty, i.e. if no card reader is in use, the raw card data field will not contain any data. The comma delimiter will still be present, even if a field is blank.

Constraints: Sequence number must be valid or a "NO" response will be returned.

Special Case: None.

Comm. Modes: Host Control, Poll and Program.

Command Code TS – Transaction Log Latest Sequence Number

This command retrieves the sequence number of the most recent transaction stored by microFlow.net.

Command:

"TS"

Responses:

"TS_SSSSSSSSSS" **Good Response.** The sequence number has been retrieved.

or...

"NOXX" No data retrieved.

Remarks: None.

Constraints: A "NO" response will be returned if the data log is not readable (corrupt or damaged).

Special Case: None.

Comm. Modes: Host Control, Poll and Program.

Command Code TU – Transaction Log Archived User Data

This command retrieves the historical transaction archived user data from microFlow.net using the sequence number of the transaction.

Command:

"TU_S...S"

Responses:

"TU_SSSSSSSSSS" Good Response.

Where:

SSSSSSSSSS	= Sequence Number
And	[transaction user data] is a comma-delimited text record with the following fields: 5 Integer Values (0-255) representing the values in USERBOOL46-USERBOOL50 at the end of the transaction; 5 User Floating Point Values corresponding to the values in USERFLOAT46-USERFLOAT50 at the end of the transaction,
And	Optionally depending on Program Code System 739, 8 User text fields

or...

"NOXX" The transaction summary data was not retrieved.

Remarks: Some fields may be empty, i.e. if no user text was entered, the user text fields will not contain any data. The comma delimiter will still be present, even if a field is blank.

Constraints: Sequence number must be valid or a "NO" response will be returned.

Special Case: None.

Comm. Modes: Host Control, Poll and Program.

Command Code VT – Request Meter Totalizer Data from the microFlow.net

This command requests a totalizer value from the microFlow.net.

Command:

"VT_X"	Product Non-resettable totals.
"VT_AA"	Additive non-resettable totals.
"VT_X_RR"	Recipe non-resettable totals.
Where:	
AA	= A1 (Additive 1)
	= A2 (Additive 2)
	= A3 (Additive 3)
	= A4 (Additive 4)
X	= R for raw total (Indicated Volume – IV)
	= G for gross volume (GV)
	= N for gross at standard temperature volume (GST)
	= P for gross at standard temperature and pressure volume (GSV)
	= M for mass totals
	= D for NSV
RR	= Recipe Number 01 – 04

Responses:

"VT_XVVVVVVVV"	**Good Response.**
"VT_AA_VVVVVV.VVV"	**Good Response.** Additive Totals.
"VT_X_RR_VVVVVVVVV"	**Good Response.** Recipe Totals.
Where:	
Z	= R for raw total (Indicated Volume – IV)
	= G for gross volume (GV)
	= N for gross volume at standard temperature (GST)
	= P for net volume temperature and pressure (GSV)
	= M for mass totals
	= D for NSV
VVVVVVVVV	= 9-digit totalizer volume
AA	= A1 (Additive 1)
	= A2 (Additive 2)
	= A3 (Additive 3)
	= A4 (Additive 4)
RR	= Recipe Number 01 – 04
or...	
"NOXX"	Totals were not retrieved.

Remarks: Recipes and Additives must be allocated.

Constraints: NO30 will be returned if the additive component, or recipe requested is not currently configured to the microFlow.net to which the request was directed.

Special Case: None.

Comm. Modes: No Control, Host Control, Poll and Program.

Command Code XC – Change Parameter Security Level

This command instructs microFlow.net to modify the security level of a program mode parameter.

Command:

"XC_DD_YYY_Z"

Where DD = Majority directory
CF = Configuration
SY = System
01-12 = Recipe Number

YYY = Parameter Number
Z = New Security level to set (1-3)

Responses:

"XC_DD_YYY_Z_A..A" **Good Response.**

Where DD = Majority Directory
CF = Configuration
SY = System
01-12 = Recipe Number

YYY = Parameter Number
Z = New Security level to set (1-3)
A...A = Programmed Value

or...
"NOXX" Value not changed.

Remarks: None.

Constraints: None.

Special Case: None.

Comm. Modes: Host Control, Poll and Program.

Command Code XV – Read Parameter Security Level

This command requests the current security level for a program mode parameter.

Command:

"XV_DD_YYY"

Where DD = Major directory
CF = Configuration
SY = System
01-12 = Recipe Number

YYY = Parameter Number

Responses:

"XV_DD_YYY_Z_A..A" **Good Response.**

Where DD = Majority Directory
CF = Configuration
SY = System
01-12 = Recipe Number

YYY = Parameter Number
Z = New Security level to set (1-3)
A...A = Programmed Value

or...
"NOXX" Value not read.

Remarks: None.

Constraints: None.

Special Case: None.

Comm. Modes: No Control, Host Control, Poll and Program.

Appendix I – Reference for "NOXX" Responses

XX	Description
00	Invalid Command
01	In Program Mode
02	Released
03	Value Out of Range
04	Flow Active
05	No Batch Ever Done
06	Operation Not Allowed
07	Wrong Control Mode
08	Reserved
09	Alarm Condition
10	Reserved
11	Operation Out Of Sequence
12	Power Fail During Batch
13	Reserved
14	Program Code Not Used
15	Display/Keypad In Use
16	Reserved
17	No Keypad Data Pending
18	No Batch In Progress
19	Option Not Installed
20	Reserved
21	Permissive Delay Active
22	Print Request Pending
23	Reserved
24	Must Be In Program Mode
25	Reserved
26	Volume Type Not Selected
27	Exactly One Recipe Must Be Enabled
28	Reserved
29	Checking Entries
30	Invalid Product/Recipe/Additive
31	Reserved
32	No Key Ever Pressed
33	Reserved
34	Reserved
35	Reserved
36	Reserved
37	Reserved
38	Reserved
41	Reserved
90	Reserved
91	Reserved
92	Reserved
93	Reserved
94	Reserved
95	Reserved
99	Reserved

Appendix II – Alphanumeric Character Set Used By the microFlow.net

The following characters are translated by the microFlow.net to display special characters not found on a typical keyboard: the tilde (~) will display as a degree sign at the microFlow.net; degree signs sent by the microFlow.net in a response will appear as a tilde (~) on your computer. The vertical bar (|) translates to a script lowercase “l”, used to denote liters of volume.

Lowercase letters may not be used to issue any of the two-digit command codes (SB, GD, EA, etc.); a NO00, Command Non-existent, will be returned as the response.

Some special characters (for example, [,], &, +, -, and .) are used in prompting or other data entry; all other special characters and lowercase letters are typically used in prompts and textual descriptions entered at the microFlow.net, such as the product name, etc. The comma (,) may not be used within any prompt (WA-WG, WX, WQ, WP.)

ASCII	DECIMAL	HEX
NUL	0	0
STX	2	2
ETX	3	3
LF	10	A
CR	13	D
SP	32	20
!	33	21
"	34	22
#	35	23
$	36	24
%	37	25
&	38	26
`	39	27
(	40	28
)	41	29
*	42	2A
+	43	2B
'	44	2C
-	45	2D
.	46	2E
/	47	2F
0	48	30
1	49	31
2	50	32
3	51	33
4	52	34
5	53	35
6	54	36
7	55	37
8	56	38
9	57	39
:	58	3A
;	59	3B
<	60	3C
=	61	3D
>	62	3E
?	63	3F

ASCII	DECIMAL	HEX
@	64	40
A	65	41
B	66	42
C	67	43
D	68	44
E	69	45
F	70	46
G	71	47
H	72	48
I	73	49
J	74	4A
K	75	4B
L	76	4C
M	77	4D
N	78	4E
O	79	4F
P	80	50
Q	81	51
R	82	52
S	83	53
T	84	54
U	85	55
V	86	56
W	87	57
X	88	58
Y	89	59
Z	90	5A
[	91	5B
\	92	5C
]	93	5D
^	94	5E
-	95	5F
`	96	60
a	97	61
b	98	62
c	99	63
d	100	64
e	101	65
f	102	66
g	103	67
h	104	68
i	105	69
j	106	6A
k	107	6B
l	108	6C
m	109	6D
n	110	6E
o	111	6F

ASCII	DECIMAL	HEX
p	112	70
q	113	71
r	114	72
s	115	73
t	116	74
u	117	75
v	118	76
w	119	77
x	120	78
y	121	79
z	122	7A
{	123	7B
|	124	7C
}	125	7D
~	126	7E
DEL	127	7F

ASCII Codes

Appendix III – Unauthorized Flow

Unauthorized flow occurs when the microFlow.net picks up and accumulates stray pulses from the meter between transactions. This may be leakage, or it may be product moving back and forth in the meter. These raw pulse counts are accumulated in the flow counter and can be viewed in the response to the “FL” command. These raw pulses can be converted to units of volume (gallons, liters, etc.) by dividing the accumulated pulse count by the input resolution for the meter. For example, if input resolution is set to 50, an accumulated pulse count of 104 reflects unauthorized flow of a little more than 2 units of volume (gallons, liters, etc).

When the microFlow.net is authorized for a transaction, the flow counter is zeroed. This updates the non-resettable total for the product by the amount accumulated in the flow counter.

The presence of unauthorized flow is indicated by a status of Flowing without a corresponding Released Status in the response to the “EQ” or “RS” commands.

Appendix IV – Using the Bit-Map Tables

Many command codes in this manual use bit-mapping to encode information concisely and in as short a form as is possible. Up to four discrete bits of information may be represented by a single ASCII character, both as commands to and responses from the microFlow.net. Most command codes that use bit-mapping consist of two or more such ASCII characters. This appendix describes how to encode or decode a single ASCII character; the process can be repeated for each additional character.

Each option listed across the top of the table carries a binary weighted value associated with it. From right to left, the values are 1, 2, 4, and 8. This is why the table headers may appear to be listed backwards. Special characters are used to represent hexadecimal values A through F, which equate to decimal values 10 through 15, when the bit values for selected options are added together. The “char” column, not the “hex” column, is used to encode and decode ASCII characters.

Encoding a Bit-Mapped Character

An X in the table indicates a selected option. First, determine which of the four column header options will be encoded. Find the row that contains Xs for the options selected. The character listed along the left axis is equal to the value of the options selected.

For example, consider the “AB” command. Suppose we want to enable recipes 1, 3, 6, 7, and 8. Recipes 1 and 3 can be represented in the first ASCII character. The row containing Xs for 1 and 3 only corresponds to the ASCII character “5.” Therefore, the first character of the AB command will be 5. Recipes 6, 7, and 8 can be represented in the second ASCII character. The character corresponding to these values is a “>,” so the second character of the AB command will be >. Because no recipes have been selected that can be encoded in the third, fourth, fifth, or sixth characters of the AB command, these characters will be 0’s.

The complete AB command to enable recipes 1, 3, 6, 7 and 8 is “AB 5>0000.”

Decoding a Bit-Mapped Character

An X in the table indicates an asserted value. Decoding a character is just the opposite of encoding a character. Find the returned ASCII character in the column along the left. For each X in that row, refer to the column header to determine what option or condition is asserted.

For example, consider the following response to the “EQ” command: “580027”

“5” represents microFlow.net Authorized and microFlow.net Released

“8” represents Transaction in Progress

“0” represents no conditions met in character 3

“0” represents no conditions met in character 4

“2” represents Input #2 contact

“7” represents Input #5, Input #6, and Input #7

“0” represents no conditions met in character 7

“0” represents no conditions met in character 8.

Appendix V – Interfacing with the microFlow.net via Ethernet (TCP/IP)

Parameters Affecting TCP/IP Communications

Address: Note that the address is in the form of a TCP/IP address – 4 numbers, each from 0-255. Since each of the 4 numbers can be stored in 8 bits of data, they are often referred to by the term “octet”. The last octet in the IP address is the value used for the communications address for the RS232/RS485 ports.

Netmask: This program code allows the entry of the network mask. Internet standards specify that each IP address has two parts – one part is the network address, and the other part is the host machine’s address on the network. Due to the dynamic nature of the Internet, these “parts” are not always divided up in the same place. There are different “classes” of networks, and hence different “masks” defining which bits in the IP address are the network portion of the address. The remaining bits are the host address.

A very common network, the class “C” network, has a netmask of 255.255.255.0. This means that the first 3 octets (24 bits) define the network, and the last octet (8 bits) defines the specific machine on the network. Taking into account the reserved addresses of 0 and 255, this allows for 254 hosts on a class “C” network.

Gateway: This address specifies where the host should send IP packets when the IP address has a different network than the host. This address specifies the address of a switch or router that will pass packets to networks other than the local network out to the Internet. Note that application layer protocols such as FTP or HTTP connecting to the microFlow.net from outside do not require any value to be programmed here; this entry is only used for initiating a connection from the microFlow.net to a point outside the local network.

Using the Smith Meter protocol over TCP/IP

The microFlow.net has a reserved port that supports Smith Terminal communications. This protocol is currently fixed at port 7734. The microFlow expects a single complete command to be contained in each packet received. The microFlow will currently ignore fragmented commands or any additional commands after the first in a packet. Due to the relatively small size of the Smith Meter command frame, this usually poses no problem for the communicating client. However, most Telnet-type programs will attempt to send data as soon as it is entered, so the resulting packets received by the microFlow do not have complete commands and are ignored. The FlowMate has a built in tool – the Terminal Emulator – that sends an entire command in one packet. Custom software written to communicate with the microFlow.net can duplicate this functionality easily by submitting a completely formed communication command along with any required arguments to the TCP transport layer all at once.

Using a Web Browser to view microFlow.net information

By pointing your Web browser at the IP address of the microFlow, you can retrieve various Web pages from the device showing the current state of operation, etc. For example, assuming a microFlow.net programmed with address 192.168.1.13 is on your network. Enter **http://192.168.1.13** in your web browser to display the home page.

Appendix VI – Windows Setup of SLIP Ports

Windows 2000

- From *Control Panel*, select: *Network and Dial-up Connections*
- Select: *Make New Connection* to start the *New Connection* wizard
- Select: *Connect to Another Computer* on the first page of the wizard
- Select: *Guest* on second page of the wizard
- Select the desired comm. port on the third page of the wizard
- Select: *For all Users*
- Name the connection appropriately (i.e. “microFlow SLIP connection”)
- If prompted to login, click *Properties* or return to *Network Connections* folder, find new connection, right click and select: *Properties*
- Under *General* tab verify the device port desired, click on *Configure*, set baud rate appropriately
- Under *Options* tab disable prompt for name and password, etc.
- Under *Networking* tab
  - Select *SLIP: UNIX Connection* in the *Type of Dial-up Server I am Calling* combo box
  - Clear all check boxes except for *Internet Protocol (TCP/IP)*
  - Click *Properties* for the Internet Protocol component
    - In the Properties dialog for the TCP/IP connection, select *Use the following IP Address*, and specify an address that is different but on the same subnet as the microFlow.net (i.e. if your microFlow.net is 192.168.0.1, make the address for the SLIP client 192.168.0.9 or similar.

Windows XP

- From *Control Panel*, select: *Network and Internet Connections*
- Select: *Create a New Connection* to start the *New Connection* wizard
- From the *Network Connection Type* page select: *Set up an advanced connection*
- From the Advanced Connection Options page select: Connect directly to another computer
- From the *Host or Guest?* page select *Guest*
- On the *Connection Name* page, name the connection appropriately (i.e. “microFlow SLIP connection”)
- From the *Select a Device* page select Communications Cable between two Computers (COM _) from the list
- From the *Connection Availability* page select *Anyone’s use*
- On the *Connect* page leave *Save this user name and password for the following users* unchecked
- From Connect page select: *Properties*
- Under *General* tab select *Communications cable between two computers* then, click on *Configure*, set baud rate appropriately
- Under *Options* tab uncheck *Dialing options*
- Under *Networking* tab
  - Select *SLIP: UNIX Connection* in the *Type of Dial-up Server I am Calling* combo box
  - Clear all checkboxes except for *Internet Protocol (TCP/IP)* and *QoS Packet Scheduler*
  - Click *Properties* for the Internet Protocol component
    - In the Properties dialog for the TCP/IP connection, select *Use the following IP Address*, and specify an address that is different but on the same subnet as the microFlow.net (i.e. if your microFlow.net is 192.168.0.1, make the address for the SLIP client 192.168.0.9 or similar.

Acoustic Coupler: A device that converts electrical signals into audio signals, enabling data to be transmitted over the public telephone network via a conventional telephone handset.

Address: A coded representation of the origin or destination of data.

Algorithm: A procedure for solution of a problem in a finite number of steps.

Applications Software: The applications tasks within a system that make the unit conform to the unique circumstances which it must control. Each task within the applications software performs a function corresponding to an external event such as xxx etc.

ASCII (American Standard Code for Information Interchange): This term is pronounced "asky." It is a seven-bit-plus-parity code established by ANSI to achieve compatibility between data services.

Assembly Language: A machine-oriented language designed to be used to write or express statements of an assembly program. The instruction code written in an assembly language is often a mnemonic code for assembling machine language computer instructions.

Asynchronous Transmission: Transmission in which time intervals between transmitted characters may be of unequal length. Transmission is controlled by start and stop bits at the beginning and end of each character.

Attenuation: The decrease in magnitude of a signal.

Bandwidth: The range of frequencies available for signaling; the difference expressed in Hertz between the highest and lowest frequencies of a band.

Baud: Unit of signaling speed. The speed in baud is the number of discrete conditions or signal events per second. If each signal event represents only one bit condition, baud rate equals bps. When each signal event represents other than one bit, e.g., digit, baud rate does not equal bps.

BCC (Block Check Character): The result of a transmission verification algorithm accumulated over a transmission block. It is normally appended at the end; (e.g., CRC, LRC).

Binary Coded Decimal Representation (BCD): A system of representing decimal numbers, in which each decimal digit is represented by a combination of four digits (bits). For example, the decimal value 6 is represented by 0110 in BCD, the decimal value 15 is represented by 0001 0101.

Binary Digit (bit): A numeral in the binary scale of notation. This digit may be zero or one, which is equivalent to an off or an on position value.

Bisynchronous Transmission (BSC): An IBM communications protocol which uses a defined set of control characters for synchronized transmission of binary coded data between stations in a data communications system.

Bit (Binary Digit): Contraction of "binary digit," the smallest unit of information in a binary system. A bit represents the choice between a one or zero condition. Block one or more records considered or transferred as a unit, particularly with reference to input and output.

Block Parity Check: In data transmission, it is an error detection technique, which is used in addition to parity checks. That is, in addition to bits, one or more check characters are added to each message transmitted. When received, if these characters match the one transmitted, the message is assumed correct, otherwise an error is noted.

BPS (Bits Per Second): Unit of data transmission rate.

Buffer: A storage device used to compensate for a difference in rate of data flow or event timing when transmitting data from one device to another.

Buss: One or more conductors used for transmitting signals, data or power. Often a buss acts as a common connection between several locations.

Byte: A binary element string operated upon as a unit and usually shorter than a computer "word." Eight-bit bytes are most common. A byte is also called a "character."

Carriage Return: In a character-by-character printing mechanism, the operation that causes the next character to be printed at the left margin.

Cathode Ray Tube (CRT): A television-like picture tube used in visual display terminals.

CCITT: International Telegraph and Telephone consultative Committee (from the French, Comite Consultatif International Telegraphique et Telephonique). An international consultative committee that sets international communications standards.

Character: The actual or coded representation of a digit, letter or special symbol.

Clock: Shorthand term for the source(s) of timing signals used in synchronous transmission. More generally: the source(s) of timing signals sequencing electronic events.

Code: A system of symbols and rules for use in representing information.

Compiler: A computer program that prepares a machine-language program from instructions or sub-routines written in a high-level language. A compiler usually generates more than one machine instruction for each symbolic instruction.

Computer: A device capable of solving problems by accepting data, performing prescribed operations on the data under direction of a stored program, and supplying the results of these operations.

Conditioning: The addition of equipment to a leased voice grade channel to provide minimum values of line characteristics required for transmission.

Console: The part of a computer that is used for communications between operators or service personnel and the system. The console contains lights, keys, switches, and related circuits for man-machine communication. The console may be used to control the machine manually, correct errors, determine the status of machine circuits, registers, and counters, determine the contents of storage, and manually revise the contents of storage.

Contention: The facility provided by the dial network or a port selector that allows multiple terminals to compete on a first-come-first-served basis for a smaller number of computer ports.

Conversational Mode: A procedure for communication between a terminal and the computer in which each entry from the terminal elicits a response from the computer and vice versa.

CPU (Central Processing Unit): Portion of a computer which directs the sequence of operations and initiates the proper commands to the computer for execution.

CR (Carriage Return): A formatting tool that moves the active position to the first character position of the same line.

CRC (Cyclic Redundancy Check): An error detection scheme in which the check character is generated by taking the remainder after dividing all the serialized bits in a block by a predetermined binary number.

CTS (Clear To Send): Physical modern interface control signal from data communications equipment (DCE) that indicates to the data terminal equipment (DTE) that it may begin data transmission.

Current Loop: Method of interconnecting terminals and transmitting signals, whereby a mark (binary 1) is represented by current on the line and a space (binary 0) is represented by the absence of current.

Data Integrity: A performance measure based on the rate of undetected errors.

Data Set: A device that converts the signals of a business machine to signals suitable for transmission over communication lines and vice versa. It may also perform other related functions.

DC (Device Control): A category of control characters primarily intended for turning on or off a subordinate device. Samples of DC characters are as follows: DC1, DC2, etc. (See X-ON and X-OFF).

DCE (Data Communications Equipment): The equipment that provides the functions required to establish, maintain and terminate a data transmission connection; e.g., a modem.

Debugging: The process of identifying and correcting mistakes in a computer program.

DIP (Dual In-Line Package): An electronic component package characterized by two rows of external connecting pins which are inserted into the holes of the printed circuit board.

Diskette: A small magnetic disk (resembles a 45-rpm record), which is sealed in a square plastic jacket and weighs less than 2 ounces.

DTE (Data Terminal Equipment): The equipment acting as data source, data sink or both.

EIA (Electronic Industries Association): A standards organization in the U.S.A. specializing in the electrical and functional characteristics of interface equipment.

EIA-232C: Interface between data terminal equipment and data communication equipment employing unbalanced voltage digital interface circuits.

EIA-422: Electrical characteristics of balanced-voltage digital interface circuits.

Emulate: To imitate a computer system by a combination of hardware and software that allows programs written for one computer to run on another.

Ethernet: Networking technology popularly used for Local Area Networks (LANs)

ETX (End of Text): A transmission control character which terminates a text.

File Maintenance: The activity of keeping a file up-to-date by adding, changing or deleting data.

Firmware: A computer program or software stored permanently in PROM or ROM or semi-permanently in EPROM.

FTP: File Transfer Protocol, an application layer protocol used on TCP/IP networks especially for moving large files between hosts on the internet.

Full-duplex: Simultaneous, two-way, independent transmission in both directions.

Half-duplex: Transmission in either direction, but not both directions simultaneously.

Handshaking: Exchange of predetermined signals between two devices for purposes of control.

Hardcopy: A printed copy of machine output in readable form, for example, reports, listings, documents, summaries.

HDLC (High Level Data Link Control): The international standard communication protocol defined by ISO.

Header: The control information prefixed in a message text, e.g., source or destination address, sequence number or message length or type.

Hertz (Hz): A measure of frequency or bandwidth. The same as cycles per second.

Hexadecimal Number System: The number system with the base of sixteen. In hexadecimal, the first ten digits are 0 9 and the last six digits are represented by the letters A-F.

HTTP: Hypertext Transfer Protocol; an application-level protocol used widely on the World Wide Web

Impact Printer: A printer forms characters by the use of print hammers that press the paper and ribbon against selected type characters as they pass in front of the paper. Type characters are commonly mounted on a moving chain or are engraved on the face of a rotating drum. Typical speeds range from 500 to 2,000 lines per minute.

ISO: International Standards Organization.

KSR: Keyboard Send/Receive. A combination teleprinter transmitter and receiver with transmission capability from keyboard only.

LAN: Local Area Network; A data communications system handling a few nodes up to several hundred, confined to a few buildings within a few thousand meters of one another.

Line Driver: A signal converter which conditions a digital signal to ensure reliable transmission over an extended distance.

Line Turnaround: The reversing of transmission direction from sender to receiver or vice versa when using a half-duplex circuit.

Local Line, Local Loop: A channel connecting the subscriber's equipment to the line terminating equipment in the central office. Usually a metallic circuit (either 2-wire or 4-wire).

LRC (Longitudinal Redundancy Check): An error detection scheme in which the check character is a 7 bit ASCII character calculated as the exclusive (OR) sum of all characters excluding itself in the packet of transmitted information.

Magnetic Disk: A storage device of magnetically coated disks, on the surface of which information is stored in the form of magnetic spots arranged in a manner to represent binary data. These data are arranged in circular tracks around the disks, are accessible to reading and writing heads on an arm that can be moved mechanically to the desired disk, and then to the desired track on that disk. Data from a given track is read or written sequentially as the disk rotates.

Magnetic Tape: An external storage medium in the form of a ferrous oxide coating on a reel of metallic or plastic tape on which bits may be recorded magnetically as a means of retaining data.

Mark: Presence of signal. In telegraph communication, a mark represents the closed condition or current flowing. A mark impulse is equivalent to a binary 1.

Message Format: Rules for the placement of such portions of a message as message heading, address text, and end of message.

Minicomputer: A computer usually weighing less than 50 pounds, that contains a relatively small internal memory and that can accept peripherals such as disk storage, magnetic tape units and line printers.

Mnemonic Code: Instructions for the computer written in a form that is easy for the programmer to remember. A program written in mnemonics must be converted to machine code prior to execution.

Modem (Modulator-Demodulator): A device used to convert serial digital data from a transmitting terminal to a signal suitable for transmission over a telephone channel or to reconvert the transmitted signal to serial digital data for acceptance by a receiving terminal.

Multiplexer: A device used for division of a transmission facility into two or more sub-channels either by splitting the frequency band into narrower bands (frequency division), or by allotting a common channel to several different transmitting devices, one at a time (time division).

Noise: In communication theory, an undesired disturbance in a communication system. Noise can generate errors or spurious messages. Contrast with signal.

Null Modem: A device that connects two DTE devices directly by emulating the physical connections of a DCE device.

Off-line: Pertaining to equipment or devices not under direct control of the central processing unit.

On-line: Pertaining to equipment or devices in direct communication with the central processing unit.

Operating System: The operating system supplies all services and utilities to the applications task necessary to run the system efficiently. The operating system provides priorities and schedules of the different applications tasks.

Packet: A group of binary digits, including data and call control signals, which is switched as a whole. The packet information is arranged in a specific format.

Parallel Transmission: Byte-wide data transmission that allocates a data line for each bit in a word. Transmission is usually unidirectional.

Parity Check: Addition of non-information bits to data, making the number of ones in a byte (bit group) either always odd or always even. This permits detection of errors in blocks that have a single error.

Perforator: A keyboard device for punching paper tape.

Polling: A centrally controlled method of calling a number of devices, by sequential inquiry, to permit them to transmit information.

Port: An interface on a computer configured as data terminal equipment and capable of attaching a modem for communication with a remote data terminal.

Priority or Precedence: Controlled transmission of messages in order of their designated importance; e.g., urgent or routine.

Program: An explicit set of steps or instructions that directs the computer and coordinates the operation of the various hardware components.

PROM (Programmable Read Only Memory): Non-volatile memory chip that allows a program to reside permanently in a piece of hardware.

Protocol: A formal set of conventions governing the formatting and relative timing of message exchange between two communicating systems.

Punched Paper Tape: A strip of paper on which characters are represented by combinations of punched holes.

Real Time: Pertaining to the actual time during which a physical process takes place. Pertaining to the performance of a computation during a period, short in comparison, with the actual time that the related physical process takes place in order that results of the computations can be used in guiding the physical process.

Queue: A waiting line or area.

RAM: Random Access Memory. Semiconductor read-write volatile memory. Data stored is lost if power is turned off.

Redundancy Check: A technique of error detection involving the transmission of additional data related to the basic data in such a way that the receiving terminal, by comparing the two sets of data, can determine to a certain degree of probability whether an error has occurred in transmission.

Re-perforator: A device that automatically punches a paper tape from received signals.

Response Time: The elapsed time between the generation of the last character of a message at a terminal and the receipt of the first character of the reply. It includes terminal delay and network delay.

ROM: Read-Only Memory. Non-volatile semiconductor memory manufactured with predefined data content, permanently stored.

RTS (Request to Send): Physical modem interface control signal from DTE, requesting clearance to transmit.

SDLC (Synchronous Data Link Control): IBM standard communication protocol superseding BSC.

SLIP (Serial Line Internet Protocol): The microFlow.net communications ports communicate with a minicomputer type device using TCP/IP over a serial communications line.

Secondary Storage: A storage that principally supplements primary storage. Secondary storage devices include magnetic disk units, magnetic drums, and magnetic tape. Secondary storage is characterized by slower speed of operation and correspondingly lower cost than those related to primary storage.

Sector: A portion of a track (from a magnetic disk) whose shape is similar to a slice of pie. Each track is equally divided into sectors, in which each sector may have its own distinct address.

Selective Calling: The ability of a transmitting station to specify which of several stations on the same line is to receive a message.

Serial Transmission: A method of data transmission in which each bit of information is sent sequentially on a single data channel. Serial transmission is the normal transmission mode for data communications.

Short Haul Modem: A signal converter which conditions a digital signal to ensure reliable transmission over DC continuous private line metallic circuits without interfering with adjacent pairs in the same telephone cable.

Signal: In communication theory, an intentional disturbance in a communication system. Contrast with noise.

Simplex Transmission: Data Transmission in one direction only.

Single-Address Message: A message to be delivered to only one destination.

Start Bit: In a synchronous transmission, the last bit or element in each character, normally a mark, to which is assigned a minimum duration during which the receiving equipment is returned to its rest condition in preparation for the reception of the next character.

Start Bit: In asynchronous transmission, the first bit or element in each character, normally a space, which prepares the receiving equipment for the reception and registration of the character.

Stop Bit: In start-stop transmission, the last bit or element in each character, normally a mark, to which is assigned a minimum duration, during which the receiving equipment is returned to its rest condition in preparation for the reception of the next character.

Storage: A general term for any device capable of retaining information.

STX (Start of Text): A transmission control character which precedes a text and which is used to terminate a heading.

Synchronous Transmission: Transmission where the data characters and bits are transmitted at a fixed rate with the transmitter and receiver synchronized. Synchronous transmission eliminates the need for start and stop bits.

Table: An organized collection of data, usually arranged in an array where each item in the array is uniquely identifiable by some label or by its relative position. Items in a table are easier to locate or identify, and thus provide a ready reference.

TC (Transmission Control): Category of control characters intended to control transmission of information over telecommunication networks. Samples of TC characters are as follows: ACK, DLE, ENQ, EOT, ETB, ETX, NAK, SOH, STX and SYN.

TCP/IP: Transfer Control Protocol/Internet Protocol; protocol used to reliably send messages across a network or the internet.

Voice Grade Channel: A channel suitable for transmission of speech, digital or analog data, or facsimile, generally with a frequency range of about 300 to 3000 Hertz.

Word: A set of characters that occupies one storage location and is treated by the computer circuits as a unit and is transported as such. Word lengths are fixed or variable, depending on the particular computer and program.

X-OFF (Transmitter Off, DC3): The communication control character used to instruct a terminal to suspend transmission.

X-ON (Transmitter On, DC1): The communication control character used to instruct a terminal to start or resume transmission.

Technical Support

Contact Information:
Field Service Response Center
24/7 Technical Support/Schedule a Technician: 1-844-203-4014
System Installation Supervision, Start-Up, Training, and Commissioning Services Available

USA Operation
1602 Wagner Avenue
Erie, Pennsylvania 16510 USA
P:+1 814.898.5000

Germany Operation
Smith Meter GmbH
Regentstrasse 1
25474 Ellerbek, Germany
P:+49 4101 304.0

www.GuidantMeasurement.com