GUIDANT

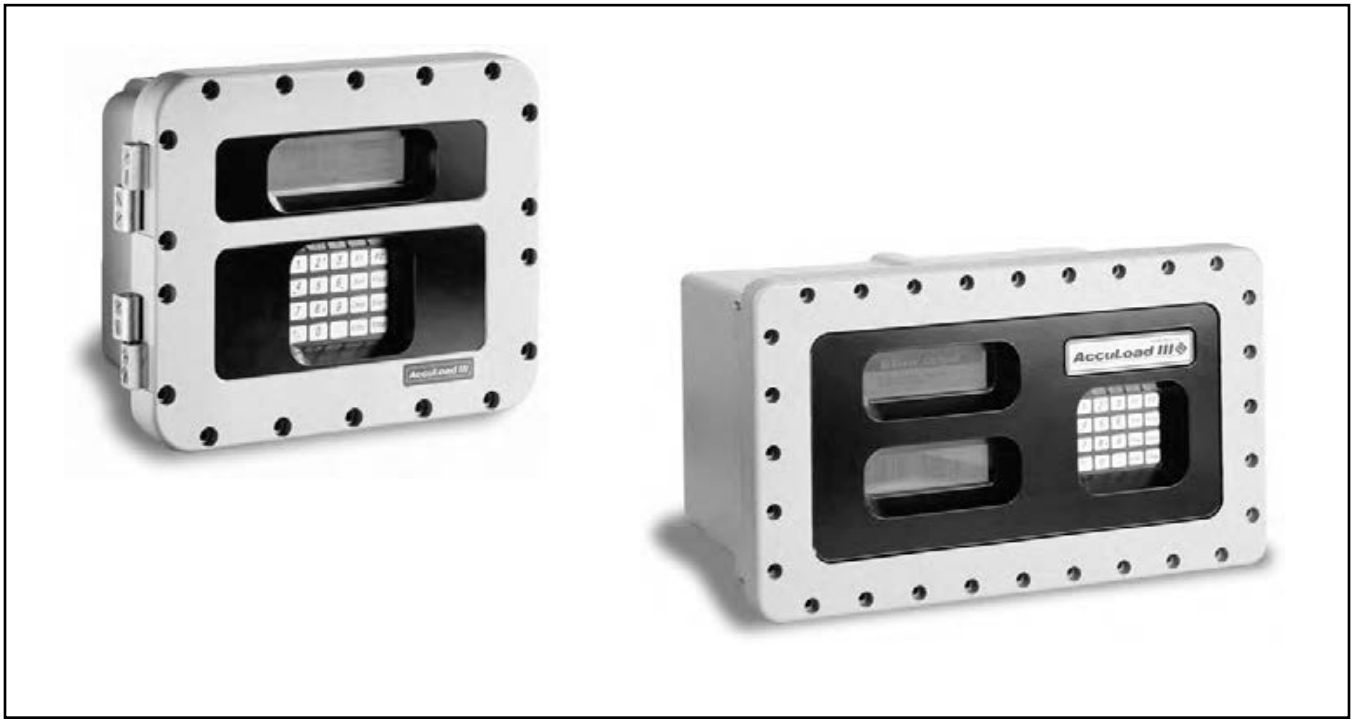
Electronic Preset Controller

# *Smith Meter® AccuLoad® III*

## Communications

Issue/Rev. 1.4 (2/13)

Bulletin MN06130L



(This Page Intentionally Left Blank)

***Caution***

The default or operating values used in this manual and in the program of the AccuLoad III 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 AccuLoad III.

## *Proprietary Notice*

This document contains information that is proprietary to Guidant and is available solely for customer information. The information herein shall not be duplicated, used, or disclosed with-out prior permission of Guidant.

Guidant will not be held responsible for loss of liquid or for damage of any kind or from any cause to the person or property of others, or for loss or profit, or loss of use, or any other special, incidental, or consequential damages caused by the use or misapplication of the contents stated herein.

# Table of Contents


Caution ..... i
Disclaimer ..... i
Proprietary Notice ..... ii
**Section I - Introduction ..... 1**
**Section II - Communications Primer ..... 2**
ASCII Code Table ..... 4
**Section III - Communication Protocol ..... 7**
Communication Types ..... 7
Communications for Terminal Mode of Operation ..... 7
Communications for Minicomputer Mode of Operation ..... 7
Text Format ..... 8
Communication Control Selections ..... 8
**Section IV - Communications with Smart Additive Injectors ..... 10**
Pass-by Communications Mode ..... 10
Command Code IC – Injector Command ..... 11
Command Code IR – Injector Response ..... 12
**Section V - Command Reference Guide ..... 13**
Command Code AB – Allocate Blend Recipes ..... 13
Command to Allocate Recipes, Character 1 ..... 14
Command to Allocate Recipes, Character 2 ..... 15
Command To Allocate Recipes, Character 3 ..... 16
Command To Allocate Recipes, Character 4 ..... 17
Command To Allocate Recipes, Character 5 ..... 18
Command To Allocate Recipes, Character 6 ..... 19
Command To Allocate Recipes, Character 7 ..... 20
Command To Allocate Recipes, Character 8 ..... 21
Command To Allocate Recipes, Character 9 ..... 22
Command To Allocate Recipes, Character 10 ..... 23
Command To Allocate Recipes, Character 11 ..... 24
Command To Allocate Recipes, Character 12 ..... 25
Command To Allocate Recipes, Character 13 ..... 26
Command Code AN – Read Analog Engineering Value ..... 27
Command Code AO – Set General Purpose Analog Output Engineering Value ..... 28
Command Code AP – Authorize Transaction to Preset ..... 29
Additive Selection Code A1 ..... 30
Additive Selection Code A2 ..... 31
Additive Selection Code A3 ..... 32
Additive Selection Code A4 ..... 33
Additive Selection Code A5 ..... 34
Additive Selection Code A6 ..... 35
Command Code AR – Alarm Reset ..... 36
Alarm Status Codes That Can Be Reset Through Communications – System ..... 37
Alarm Status Codes That Can Be Reset Through Communications – Injector ..... 38
Alarm Status Codes That Can Be Reset Through Communications – Arm ..... 39
Alarm Status Codes That Can Be Reset Through Communications – Meter ..... 40
Alarm Status Codes That Can Be Reset Through Communications – Product ..... 41
Command Code AT – Request Additive Volumes by Transaction ..... 42
Command Code AU – Authorize Transaction ..... 44
Command Code AV – Request Additive Volumes by Batch ..... 45
Command Code BA – Bay Assignment ..... 47
Command Code BR – Boolean/Algebraic Variable Read ..... 48
Command Code BW – Boolean/Algebraic Variable Write ..... 49
Command Code CD – Card Reader Data ..... 50
CD Command, Optional Character C1 ..... 51
CD Command, Optional Character C2 ..... 52
Response to CD Command, Character S1 ..... 53

# Table of Contents


Command Code CF – AccuLoad Configuration ..... 54
Command Code CF – AccuLoad Configuration ..... 54
Command Code CP – Request Contaminant Percentage ..... 55
Command Code CT – Clear Transactions Locked in Standby Mode ..... 56
Command Code CV – CIVACON Status ..... 57
Response to CV Command, Character C1 ..... 58
Response to CV Command, Character C2 ..... 59
Response to CV Command, Character C3 ..... 60
Response to CV Command, Character C4 ..... 61
Command Code DA – Release Keypad and Display ..... 62
Command Code DD – Delete Driver Card Data ..... 63
Command Code DI – Insert Driver Card Data ..... 64
Command Code DQ – Query Driver Card Data ..... 65
Command Code DS – Display Density Sample ..... 66
Command Code DU – Update Driver Database in Non-Volatile Memory ..... 67
Command Code DY – Request Dynamic Display Values ..... 68
System "DY_SYXX" Dynamic Display Values ..... 70
Injector Dynamic Displays ..... 71
Flow Control Type Response ..... 72
Product "DY_PNXX" Dynamic Display Values ..... 73
** E tables only (relative density @60F & current pressure) used in CPL calculation. ..... 74
Recipe "DY_DDXX" Dynamic Display Values ..... 74
Batch "DY_BNXX" Dynamic Display Values ..... 74
Transaction "DY_TRXX" Dynamic Display Values ..... 75
Flow Controlled Additives “DY_FAXX” Dynamic Display Values ..... 77
Command Code EA – Enquire Alarms ..... 79
Response to System Enquire Alarms Command, Character A1 ..... 81
Response to System Enquire Alarms Command, Character A2 ..... 82
Response to Enquire Alarms Command, Character A3 ..... 83
Response to Enquire Alarms Command, Character A4 ..... 84
Response to Enquire Alarms Command, Character A5 ..... 85
Response to Enquire Alarms Command, Character A6 ..... 86
Response to Enquire Alarms Command, Character A7 ..... 87
Response to Enquire Alarms Command, Character A8 ..... 88
Response to Enquire Alarms Command, Character A9 ..... 89
Response to Enquire Alarms Command, Character A10 ..... 90
Response to Enquire Alarms Command, Character A11 ..... 91
Response to Product Enquire Alarms Command, Character A1 ..... 92
Response to Product Enquire Alarms Command, Character A2 ..... 93
Response to Product Enquire Alarms Command, Character A3 ..... 94
Response to Product Enquire Alarms Command, Character A4 ..... 95
Response to Arm Enquire Alarms Command, Character A1 ..... 96
Response to Arm Enquire Alarms Command, Character A2 ..... 97
Response to Arm Enquire Alarms Command, Character A3 ..... 98
Response to Injector Enquire Alarms Command, Character A1 ..... 99
Response to Injector Enquire Alarms Command, Character A2 ..... 100
Response to Injector Enquire Alarms Command, Character A3 ..... 101
Response to Injector Enquire Alarms Command, Character A4 ..... 102
Response to Injector Enquire Alarms Command, Character A5 ..... 103
Response to Injector Enquire Alarms Command, Character A6 ..... 104
Response to Injector Enquire Alarms Command, Character A7 ..... 105
Response to Injector Enquire Alarms Command, Character A8 ..... 106
Response to Injector Enquire Alarms Command, Character A9 ..... 107
Response to Injector Enquire Alarms Command, Character A10 ..... 108
Response to Injector Enquire Alarms Command, Character A11 ..... 109
Response to Injector Enquire Alarms Command, Character A12 ..... 110
Response to Injector Enquire Alarms Command, Character A13 ..... 111

# Table of Contents


Response to Injector Enquire Alarms Command, Character A14 ..... 112
Response to Injector Enquire Alarms Command, Character A15 ..... 113
Response to Injector Enquire Alarms Command, Character A16 ..... 114
Response to Injector Enquire Alarms Command, Character A17 ..... 115
Response to Injector Enquire Alarms Command, Character A18 ..... 116
Response to Injector Enquire Alarms Command, Character A19 ..... 117
Response to Injector Enquire Alarms Command, Character A20 ..... 118
Response to Injector Enquire Alarms Command, Character A21 ..... 119
Response to Injector Enquire Alarms Command, Character A22 ..... 120
Response to Injector Enquire Alarms Command, Character A23 ..... 121
Response to Injector Enquire Alarms Command, Character A24 ..... 122
Response to Injector Enquire Alarms Command, Character A25 ..... 123
Response to Injector Enquire Alarms Command, Character A26 ..... 124
Response to Injector Enquire Alarms Command, Character A27 ..... 125
Response to Injector Enquire Alarms Command, Character A28 ..... 126
Response to Injector Enquire Alarms Command, Character A29 ..... 127
Response to Injector Enquire Alarms Command, Character A30 ..... 128
Response to Injector Enquire Alarms Command, Character A31 ..... 129
Response to Injector Enquire Alarms Command, Character A32 ..... 130
Response to Injector Enquire Alarms Command, Character A33 ..... 131
Response to Injector Enquire Alarms Command, Character A34 ..... 132
Response to Injector Enquire Alarms Command, Character A35 ..... 133
Response to Injector Enquire Alarms Command, Character A36 ..... 134
Response to Injector Enquire Alarms Command, Character A37 ..... 135
Response to Injector Enquire Alarms Command, Character A38 ..... 136
Response to Injector Enquire Alarms Command, Character A39 ..... 137
Response to Injector Enquire Alarms Command, Character A40 ..... 138
Response to Injector Enquire Alarms Command, Character A41 ..... 139
Response to Injector Enquire Alarms Command, Character A42 ..... 140
Response to Injector Enquire Alarms Command, Character A43 ..... 141
Response to Injector Enquire Alarms Command, Character A44 ..... 142
Response to Injector Enquire Alarms Command, Character A45 ..... 143
Response to Injector Enquire Alarms Command, Character A46 ..... 144
Response to Injector Enquire Alarms Command, Character A47 ..... 145
Response to Injector Enquire Alarms Command, Character A48 ..... 146
Response to Injector Enquire Alarms Command, Character A49 ..... 147
Response to Injector Enquire Alarms Command, Character A50 ..... 148
Response to Injector Enquire Alarms Command, Character A51 ..... 149
Response to Injector Enquire Alarms Command, Character A52 ..... 150
Response to Injector Enquire Alarms Command, Character A53 ..... 151
Response to Injector Enquire Alarms Command, Character A54 ..... 152
Response to Injector Enquire Alarms Command, Character A55 ..... 153
Response to Injector Enquire Alarms Command, Character A56 ..... 154
Response to Injector Enquire Alarms Command, Character A57 ..... 155
Response to Injector Enquire Alarms Command, Character A58 ..... 156
Response to Injector Enquire Alarms Command, Character A59 ..... 157
Response to Injector Enquire Alarms Command, Character A60 ..... 158
Response to Injector Enquire Alarms Command, Character A61 ..... 159
Response to Injector Enquire Alarms Command, Character A62 ..... 160
Response to Injector Enquire Alarms Command, Character A63 ..... 161
Response to Injector Enquire Alarms Command, Character A64 ..... 162
Response to Injector Enquire Alarms Command, Character A65 ..... 163
Response to Injector Enquire Alarms Command, Character A66 ..... 164
Response to Injector Enquire Alarms Command, Character A67 ..... 165
Response to Injector Enquire Alarms Command, Character A68 ..... 166
Response to Injector Enquire Alarms Command, Character A69 ..... 167
Response to Injector Enquire Alarms Command, Character A70 ..... 168
Response to Injector Enquire Alarms Command, Character A71 ..... 169
Response to Injector Enquire Alarms Command, Character A72 ..... 170

# Table of Contents


Response to Injector Enquire Alarms Command, Character A73 ..... 171
Response to Injector Enquire Alarms Command, Character A74 ..... 172
Response to Injector Enquire Alarms Command, Character A75 ..... 173
Response to Injector Enquire Alarms Command, Character A76 ..... 174
Response to Injector Enquire Alarms Command, Character A77 ..... 175
Response to Injector Enquire Alarms Command, Character A77- A83 ..... 175
Response to Meter Enquire Alarms Command, Character A1 ..... 176
Response to Meter Enquire Alarms Command, Character A2 ..... 177
Response to Meter Enquire Alarms Command, Character A3 ..... 178
Response to Meter Enquire Alarms Command, Character A4 ..... 179
Response to Recipe Enquire Alarms Command, Character A1 ..... 180
Command Code EB – End Batch ..... 181
Command Code EE – Enquire Status Extended ..... 182
Response to "EE" Enquire, Character A17 ..... 183
Response to "EE" Enquire, Character A18 ..... 184
Command Code EQ – Enquire Status ..... 185
Response to "EQ" Enquire, Character A1 ..... 186
Response to "EQ" Enquire, Character A2 ..... 187
Response to "EQ" Enquire, Character A3 ..... 188
Response to "EQ" Enquire, Character A4 ..... 189
Response to "EQ" Enquire, Character A5 ..... 190
Response to "EQ" Enquire, Character A6 ..... 191
Response to "EQ" Enquire, Character A7 ..... 192
Response to "EQ" Enquire, Character A8 ..... 193
Response to "EQ" Enquire, Character A9 ..... 194
Response to "EQ" Enquire, Character A10 ..... 195
Response to "EQ" Enquire, Character A11 ..... 196
Response to "EQ" Enquire, Character A12 ..... 197
Response to "EQ" Enquire, Character A13 ..... 198
Response to "EQ" Enquire, Character A14 ..... 199
Response to "EQ" Enquire, Character A15 ..... 200
Response to "EQ" Enquire, Character A16 ..... 201
Command Code ER – Event Recall ..... 202
Command Code ES – Last Event Recall ..... 203
Command Code ET – End Transaction ..... 204
Command Code FL – Read Flow Count ..... 205
Command Code FS – Full Screen ..... 207
Command Code GD – Get Date and Time ..... 208
Command Code GK – Get Key ..... 209
Command Code GP – Get PROM Signatures ..... 210
Command Code LD – Request Load Average Density ..... 211
Command Code LO – Request Logout of Program Mode ..... 213
Command Code LP – Request Load Average Pressure ..... 214
Command Code LS – Request Last Density Sample ..... 216
Command Code LT – Request Load Average Temperature ..... 217
Command Code MC – Mass Meter Send ..... 219
Command Code MP – Minimum Preset ..... 220
Command Code MR – Mass Meter Receive ..... 221
Command Code MS – Override MMI Swing Arm Position ..... 222
Command Code NR – New Recipe ..... 223
Command Code OA – Other Address ..... 224
Command Code OR – Output Relay ..... 225
Command Code PC – Change Program Code Values ..... 226
Command Code PF – Request Time of Power-Fail ..... 228
Command Code PN – Batch Net Mass for Vapor Recovery System ..... 229
Command Code PP – Print Report to Printer ..... 230

# Table of Contents


Command Code PR – Program Change Recall ..... 231
Command Code PS – Last Program Code Change Recall ..... 232
Command Code PT – Print Transaction to Host ..... 233
Command Code PV – Request Program Code Values ..... 234
Command Code PX – Transaction Net Mass for Vapor Recovery System ..... 236
Command Code RA – Request Alarm Status ..... 237
Command Code RB – Request Batch Totals ..... 238
Command Code RC – Request Recipe Composition ..... 240
Command Code RD – Request Current Transducer Value ..... 242
Command Code RE – Reset Status Conditions ..... 244
Valid Status Codes ..... 244
Command Code RF – Recipe Flow Rates ..... 245
Command Code RK – Read Keypad ..... 246
Command Code RL – Show Recipes Loaded ..... 248
Command Code RN – Show Recipe Numbers ..... 249
Command Code RP – Request Preset ..... 250
Command Code RQ – Request Current Flow Rate ..... 251
Command Code RR – Request Recipe ..... 252
Command Code RS – Request Status ..... 253
Request Status Codes ..... 254
Command Code RT – Request Transaction Totals ..... 256
Command Code SA – Remote Start ..... 258
Command Code SB – Set Batch ..... 259
Command Code SD – Set Date and Time ..... 260
Command Code SF – Authorize and Set Batch without Override ..... 261
Command Code SP – Remote Stop ..... 262
Command Code SR – Show Recipes Currently Allocated ..... 263
Command Code SS – Split Screen ..... 264
Command Code ST – Remote Stop on Arm ..... 265
Command Code SV – Extended Services Routines ..... 266
Command Code SW – Swing Arm Position ..... 267
Command Code TA – Set Transaction ..... 268
Command Code TD – Turbine Meter Diagnostics ..... 269
Command Code TI – Show Prompt Data Entry ..... 270
Command Code TN – Show Transaction Stop Date and Time ..... 271
Command Code TR – Read Turbine Meter Signature ..... 272
Command Code TS – Turbine Meter Status ..... 273
Command Code TW – Write Turbine Meter Signature ..... 274
Command Code VB – Batch Vapor Recovered Mass ..... 275
Command Code VC – Control Valve Solenoid Actuation Diagnostic Counters ..... 276
Comm. Modes: Polling only, poll and authorize, poll and program, remote control. ..... 276
Command Code VD – Valve (Closure) Diagnostic ..... 277
Command Code VT – Request Meter Totalizer Data from the AccuLoad III ..... 279
Command Code VT – Request Meter Totalizer Data from the AccuLoad III ..... 279
Command Code VX – Transaction Vapor Recovered Mass ..... 281
Command Code WA – Write Second Line of Message to Appear on Display ..... 282
Command Code WB – Write Third Line of Message to Appear on Display ..... 284
Command Code WC – Write Fourth Line of Message to Appear on Display ..... 286
Command Code WD – Write to Display ..... 288
Command Code WP – Write Delayed Prompt with Echo ..... 290
Command Code WQ – Write Delayed Prompt with Security Echo ..... 292
Command Code WX – Write to the Display in Security Mode ..... 294
Command Code XC – Change Parameter Security Level ..... 296
Command Code XV – Read Parameter Security Level ..... 297
**Section VI - Appendix ..... 298**
Appendix I – Reference for "NOXX" Responses ..... 298
Appendix II – Alphanumeric Character Set Used By the AccuLoad III ..... 300
Appendix III – Unauthorized Flow ..... 304

# Table of Contents


Appendix IV – Using the Bit-Map Tables ..... 305
Encoding a Bit-Mapped Character ..... 305
Decoding a Bit-Mapped Character ..... 305
Appendix V – Interfacing with the AccuLoad III.net via Ethernet (TCP/IP) ..... 306
Appendix VI – Windows Setup of SLIP Ports ..... 307
Appendix VII – Emulation of AccuLoad II Communications Using AccuLoad III-X ..... 308
Configuring AccuLoad II Style Communications ..... 308
Configuring Load Arms ..... 308
Configuring I/O ..... 309
Configuring Additives, Recipes and Products ..... 309
Unsupported Commands ..... 310
Limited PV/PC Support ..... 311
Other Considerations / Exceptions ..... 311
**Section VII - Communications Glossary ..... 312**
**Section VIII – Related Publications ..... 317**

# Section I – Introduction

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

Incorporated within the AccuLoad III is the ability to communicate directly (i.e., without a modem or multiplexer) with an EIA-232C and/or an EIA-485 Compatible Remote terminal or minicomputer. Depending on the communication type, certain key information from multiple AccuLoad IIIs can be requested (polled). If the AccuLoad IIIs 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 AccuLoad III program codes.

To communicate with a particular AccuLoad III 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.

| **System Program Codes 705 – 724** | | | | |
|---|---|---|---|---|
| **Port 1** | **Port 2** | **Port 3** | **Port 4** | |
| 707 | 712 | 717 | 722 | Function |
| 708 | 713 | 718 | 723 | Baud |
| 709 | 714 | 719 | 724 | Data/Parity |
| 710 | 715 | 720 | 725 | Control |
| 711 | 716 | 721 | 726 | Timeout |

| **System Codes** | |
|---|---|
| 701 | Load Arm #1 Address (1 – 99) |
| 702 | Load Arm #2 Address (1 – 99) |
| 703 | Load Arm #3 Address (1 – 99) |
| 704 | Load Arm #4 Address (1 – 99) |
| 705 | Load Arm #5 Address (1 – 99) |
| 706 | Load Arm #6 Address (1 – 99) |
| 731 | Comm Link Programming (Level of Access) |
| 732 | Modbus Endian Selection |

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 4-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.

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 AccuLoad.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 AccuLoad.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.***

## *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 |
| ? | 63 | 3F | 011 1111 |

| ASCII CHARACTER | DECIMAL | HEX | BINARY |
|---|---|---|---|
| @ | 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 |
| f | 102 | 66 | 110 0110 |
| g | 103 | 67 | 110 0111 |

| ASCII CHARACTER | DECIMAL | HEX | BINARY |
|---|---|---|---|
| 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 |
| *I* | 124 | 7C | 111 1100 |
| } | 125 | 7D | 111 1101 |
| ◦ | 126 | 7E | 111 1110 |
| DEL | 127 | 7F | 111 1111 |

**Table 1**

## *Communication Types*

The type of communicating device that is being used in the system with an AccuLoad III is programmable and can be defined in the communications directory of the AccuLoad III. Communicating devices can be used with either the EIA-232 Port or EIA-485 Port on the AccuLoad III.

**Terminal (Term Host)** – The AccuLoad III communications ports communicate with a terminal type device using a simplified communications protocol.

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

**Modbus Host** – The AccuLoad III communication ports support Modbus communications.

**SLIP (Serial Line Internet Protocol)** – The microLoad.net communications 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.

**Card Reader** – Permits the AccuLoad.net to communicate with the Smith Card Reader

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

**Network Printing** – Allows the ability of the AccuLoad to print to a network printer.

**Printer** – The AccuLoad III will automatically print a report at the end of each transaction. One AccuLoad III will communicate with a printer.

**Shared Printer** – The AccuLoad III will automatically print a report at the end of each transaction. Multiple AccuLoads may communicate with a single printer.

**Smart Injector** – Permits the AccuLoad III to communicate with smart additive injector systems. (Examples are Gate City Blend-Pak and Mini-Pak, Titan Pak3 and Smith AccuTroller.)

**CIVACON Monitor** – Permits the AccuLoad III to communicate with the CIVACON grounding and overfill system.

### *Communications for Terminal Mode of Operation*

The AccuLoad III 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 AccuLoad III, or

| | | | | |
|---|---|---|---|---|
| * | A1 A2 | text | CR | LF |

for a response from AccuLoad III

Where:

| | | |
|---|---|---|
| * | = | Asterisk Hex "2A" |
| Text | = | Character string containing instructional or response information |
| CR | = | Carriage return Hex "0D" |
| LF | = | Line feed Hex "0A" |
| A1 A2 | = | AccuLoad III Address (01 to 99) |

The universal or global address "00" is an invalid address and must not be assigned to any AccuLoad III. 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 AccuLoad III in the Terminal Mode of operation.

### *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 AccuLoad III, or

| | | | | | | |
|---|---|---|---|---|---|---|
| NL | STX | A1 A2 | text | ETX | LRC | PAD |

for a response from AccuLoad III

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 | = | AccuLoad III 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 AccuLoad. 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 AccuLoad III 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 AccuLoad III, 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 AccuLoad, 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 -** An AccuLoad III will ignore a command whose address does not match its own. The communication address is programmed into the AccuLoad III System program code 701 and following.

**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 AccuLoad III is programmable for various degrees of control.

**Polling Only –** Permits the EIA-232 or EIA-485 communication device to request information only.

**Poll and Authorize –** Permits the EIA-232 or EIA-485 communication device to request information and to authorize operation.

**Remote Control –** Permits the EIA-232 or EIA-485 communication device 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 AccuLoad stops sending data until the printer sends an Xon (11 hex) signifying that it is ready for more data.

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

**PTB-FX Protocol –** This security level is designed to support the PTB Weights and Measures Agency approved printer interface. This interface is currently supported by and applies to the Epson FX-850 printer. This interface uses a data block structure with handshaking. The AccuLoad initially sends an ENQ (enquire) to the printer. The printer then responds with an ACK (acknowledge) and is now ready to receive a block of data from the AccuLoad. A block of data is defined to be one line to be printed. The data is

wrapped between a STX (start of text) and an ETX (end of text) and is followed by the longitudinal redundancy checksum for the data block. After the AccuLoad has sent the data block, the printer will then respond with an ACK if the data was received correctly or a NAK (negative acknowledge) if not received correctly. If the printer responds with a NAK, the data block will be re-transmitted.

**PTB-LQ PROTOCOL –** This security level is designed to support a PTB Weights and Measures Agency- approved printer interface. This interface is currently supported by and applies to the Epson LQ-570 printer.

This interface uses a data block structure with handshaking. The AccuLoad initially sends an ENQ (enquire) to the printer. The printer then responds with an ACK (acknowledge) and is now ready to receive a block of data from the AccuLoad. A block of data is defined to be one line to be printed. The data is wrapped between a STX (start of text) and an ETX (end of text) and is followed by the longitudinal redundancy checksum for the data block. After the AccuLoad has sent the data block, the printer will then respond with an ACK if the data was received correctly or a NAK (negative acknowledge) if not received correctly. If the printer responds with a NAK, the data block will be re-transmitted.

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

The AccuLoad III 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 AccuLoad III will respond to the alarm as it has been instructed to do in the programming.

The AccuLoad III 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.

### *Pass-by Communications Mode*

The pass-by communications mode is designed to allow supervisory computers to communicate with smart additive injectors under the control of the AccuLoad III. Commands from the supervisory computer will be received from the AccuLoad III 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 AccuLoad III is in the Ready Mode. No queries or commands will be accepted by the AccuLoad while it is in the Run Mode or the Programming Mode.

Communications from a supervisory computer to a smart additive injector will be a three-step process. First, the supervisory computer will issue a pass-by command to the AccuLoad III. The AccuLoad III will acknowledge receipt of the command to the supervisory computer. Next, the AccuLoad 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 AccuLoad. The final step occurs when the supervisory computer requests the additive's response from the AccuLoad III.

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

## *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 AccuLoad 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" AccuLoad III Released
= "06" Operation Not Allowed
= "19" Option Not Installed

When the AccuLoad III 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 AccuLoad'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 AccuLoad, 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 pass-by command to an Additive Injection Subsystem.

**Command:**

**"IR"**

**Response:**

**"IR_r..r"**

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 AccuLoad receives the response from the injector, then strips off the STX, ETX, and LRC, and what remains is "r..r" above.*

or

**"NOXX"**

## *Command Code AB – Allocate Blend Recipes*

This command allocates the blend recipes that will be allowable for selection. This allocation is a subset of those enabled in the program mode.

**Command:**

**"AB_C1C2C3…C13"**

Where: C1C2C3…C13 is the three- to thirteen-character bit map described on the following pages.

**Responses:**

**"OK"** Good Response. Recipes have been allocated

or...

**"NOXX"** Recipes have not been allocated

**Remarks:**

1. A one allocates the corresponding recipe number for selection. Allocation is cancelled on transaction done status being set.
2. Allocation is also cancelled if, when in Remote Control Mode, a power-fail restart occurs between batches.
3. This command does not require all characters. A minimum of three characters is accepted. For those characters omitted, the corresponding recipes are not allocated.
4. This command will automatically exit from Program Mode when initiated.

**Constraints:** Recipe allocation remains in effect for the current transaction only.

**Special Case:** None.

**Comm. Modes:** Polling Only, Poll and Authorize, or Remote Control

## *Command to Allocate Recipes, Character 1*

| Character | | Recipe Request | | | |
|---|---|---|---|---|---|
| **Char** | **Hex** | **Recipe #4** | **Recipe #3** | **Recipe #2** | **Recipe #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 |
| A | 41 | X | | X | |
| B | 42 | X | | X | X |
| C | 43 | X | X | | |
| D | 44 | X | X | | X |
| E | 45 | X | X | X | |
| F | 46 | X | X | X | X |

**Command to Allocate Recipes - Character 1**

### *Command to Allocate Recipes, Character 2*

| Character | | Recipe Request | | | |
|---|---|---|---|---|---|
| **Char** | **Hex** | **Recipe #8** | **Recipe #7** | **Recipe #6** | **Recipe #5** |
| 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 |
| A | 41 | X | | X | |
| B | 42 | X | | X | X |
| C | 43 | X | X | | |
| D | 44 | X | X | | X |
| E | 45 | X | X | X | |
| F | 46 | X | X | X | X |

**Command to Allocate Recipes - Character 2**

### *Command To Allocate Recipes, Character 3*

| Character | | Recipe Request | | | |
|---|---|---|---|---|---|
| **Char** | **Hex** | **Recipe #12** | **Recipe #11** | **Recipe #10** | **Recipe #9** |
| 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 |
| A | 41 | X | | X | |
| B | 42 | X | | X | X |
| C | 43 | X | X | | |
| D | 44 | X | X | | X |
| E | 45 | X | X | X | |
| F | 46 | X | X | X | X |

**Command to Allocate Recipes - Character 3**

### *Command To Allocate Recipes, Character 4*

| Character | | Recipe Request | | | |
|---|---|---|---|---|---|
| **Char** | **Hex** | **Recipe #16** | **Recipe #15** | **Recipe #14** | **Recipe #13** |
| 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 |
| A | 41 | X | | X | |
| B | 42 | X | | X | X |
| C | 43 | X | X | | |
| D | 44 | X | X | | X |
| E | 45 | X | X | X | |
| F | 46 | X | X | X | X |

**Command to Allocate Recipes - Character 4**

### *Command To Allocate Recipes, Character 5*

| Character | | Recipe Request | | | |
|---|---|---|---|---|---|
| **Char** | **Hex** | **Recipe #20** | **Recipe #19** | **Recipe #18** | **Recipe #17** |
| 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 |
| A | 41 | X | | X | |
| B | 42 | X | | X | X |
| C | 43 | X | X | | |
| D | 44 | X | X | | X |
| E | 45 | X | X | X | |
| F | 46 | X | X | X | X |

**Command to Allocate Recipes - Character 5**

### *Command To Allocate Recipes, Character 6*

| Character | | Recipe Request | | | |
|---|---|---|---|---|---|
| **Char** | **Hex** | **Recipe #24** | **Recipe #23** | **Recipe #22** | **Recipe #21** |
| 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 |
| A | 41 | X | | X | |
| B | 42 | X | | X | X |
| C | 43 | X | X | | |
| D | 44 | X | X | | X |
| E | 45 | X | X | X | |
| F | 46 | X | X | X | X |

**Command to Allocate Recipes - Character 6**

### *Command To Allocate Recipes, Character 7*

| Character | | Recipe Request | | | |
|---|---|---|---|---|---|
| **Char** | **Hex** | **Recipe #28** | **Recipe #27** | **Recipe #26** | **Recipe #25** |
| 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 |
| A | 41 | X | | X | |
| B | 42 | X | | X | X |
| C | 43 | X | X | | |
| D | 44 | X | X | | X |
| E | 45 | X | X | X | |
| F | 46 | X | X | X | X |

**Command to Allocate Recipes - Character 7**

### *Command To Allocate Recipes, Character 8*

| Character | | Recipe Request | | | |
|---|---|---|---|---|---|
| **Char** | **Hex** | **Recipe #32** | **Recipe #31** | **Recipe #30** | **Recipe #29** |
| 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 |
| A | 41 | X | | X | |
| B | 42 | X | | X | X |
| C | 43 | X | X | | |
| D | 44 | X | X | | X |
| E | 45 | X | X | X | |
| F | 46 | X | X | X | X |

**Command to Allocate Recipes - Character 8**

## *Command To Allocate Recipes, Character 9*

| Character | | Recipe Request | | | |
|---|---|---|---|---|---|
| **Char** | **Hex** | **Recipe #36** | **Recipe #35** | **Recipe #34** | **Recipe #33** |
| 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 |
| A | 41 | X | | X | |
| B | 42 | X | | X | X |
| C | 43 | X | X | | |
| D | 44 | X | X | | X |
| E | 45 | X | X | X | |
| F | 46 | X | X | X | X |

**Command to Allocate Recipes - Character 9**

### *Command To Allocate Recipes, Character 10*

| Character | | Recipe Request | | | |
|---|---|---|---|---|---|
| **Char** | **Hex** | **Recipe #40** | **Recipe #39** | **Recipe #38** | **Recipe #37** |
| 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 |
| A | 41 | X | | X | |
| B | 42 | X | | X | X |
| C | 43 | X | X | | |
| D | 44 | X | X | | X |
| E | 45 | X | X | X | |
| F | 46 | X | X | X | X |

**Command to Allocate Recipes - Character 10**

### *Command To Allocate Recipes, Character 11*

| Character | | Recipe Request | | | |
|---|---|---|---|---|---|
| **Char** | **Hex** | **Recipe #44** | **Recipe #43** | **Recipe #42** | **Recipe #41** |
| 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 |
| A | 41 | X | | X | |
| B | 42 | X | | X | X |
| C | 43 | X | X | | |
| D | 44 | X | X | | X |
| E | 45 | X | X | X | |
| F | 46 | X | X | X | X |

**Command to Allocate Recipes - Character 11**

### *Command To Allocate Recipes, Character 12*

| Character | | Recipe Request | | | |
|---|---|---|---|---|---|
| **Char** | **Hex** | **Recipe #48** | **Recipe #47** | **Recipe #46** | **Recipe #45** |
| 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 |
| A | 41 | X | | X | |
| B | 42 | X | | X | X |
| C | 43 | X | X | | |
| D | 44 | X | X | | X |
| E | 45 | X | X | X | |
| F | 46 | X | X | X | X |

**Command to Allocate Recipes - Character 12**

### *Command To Allocate Recipes, Character 13*

| Character | | Recipe Request | | | |
|---|---|---|---|---|---|
| **Char** | **Hex** | **Reserved** | **Reserved** | **Recipe #50** | **Recipe #49** |
| 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 |
| A | 41 | X | | X | |
| B | 42 | X | | X | X |
| C | 43 | X | X | | |
| D | 44 | X | X | | X |
| E | 45 | X | X | X | |
| F | 46 | X | X | X | X |

**Command to Allocate Recipes - Character 13**

## *Command Code AN – Read Analog Engineering Value*

This command returns the current analog engineering value for the specified analog I/O point.

**Command:**

**"AN X"** Read Analog Engineering Value

**Where:** X… is the analog I/O point desired (1-6).

**Response:**

**"AN X VVVV.V"**

**Where:** X is the analog I/O point desired (1-6 VVVV.V is the current value in the programmed engineering scale units.

**Remarks:** (New in AccuLoad III-X Revision 10.07)

**Constraints:** None

**Special Care:** None

**Comm. Modes:** Polling only, Poll & Authorized, Remote Control, Poll & Program

## *Command Code AO – Set General Purpose Analog Output Engineering Value*

This command sets a new engineering value associated with the general purpose analog output, effectively changing the current/voltage out to a new value. The engineering value is scaled to the appropriate output voltage/current based on the programmed 4 ma or IV (min) and 20 mA or 5 V (max.) engineering values for the specified I/O point..

**Command:**

**"AO"** VVVV.V

**Where:** A1A2A3… is the optional six-character additive selection code.
VVVV.V is the new engineering value to set.

**Responses:**

**"OK"** AccuLoad III has set the new value.

or...

**"NOXX"** The new value was not set.

or...

**"NOO1"** The requested operation cannot be performed while the AccuLoad III is in program mode.

**Remarks:** None

**Constraints:** Output must be configured as general-purpose analog output.

**Special Case:** None

**Comm. Modes:** Polling Only, Poll & Authorize, Remote Control, Poll & Program.

## *Command Code AP – Authorize Transaction to Preset*

This command authorizes a transaction and displays the "PRESET" prompt.

**Command:**

**"AP"** Authorize Transaction
**"AP_A1A2A3A4A5A6"** Authorize AccuLoad to preset with additives

Where: A1A2A3… is the optional six-character additive selection code.

**Responses:**

**"OK"** AccuLoad III has been authorized for transaction.

or...

**"NOXX"** The transaction failed to release.

NO30 will be returned if the additive specified is not currently allocated to the arm to which the command was directed.
NO36 will be returned if the AccuLoad is being used with a card reader and the card data is invalid or has expired.

**Remarks:** 1. If the additive qualifier (A1A2A3…) is not used, the Additive selection will operate as currently programmed in the AccuLoad III. For authorization with additives, only one recipe may be allocated. Additives selected in the qualifier must be programmed for use.
2. If more than one recipe is enabled, the "Select Recipe" prompt will be displayed.

**Constraints:** The "AP" will reset the transaction done, batch done and keypad pending flags.

**Special Case:** None

**Comm. Modes:** Poll & Authorize

### *Additive Selection Code A1*

| Character Sent (A2) | | Add 4 | Add 3 | Add 2 | Add 1 |
|---|---|---|---|---|---|
| Char | Hex | | | | |
| 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 |
| A | 41 | X | | X | |
| B | 42 | X | | X | X |
| C | 43 | X | X | | |
| D | 44 | X | X | | X |
| E | 45 | X | X | X | |
| F | 46 | X | X | X | X |

**Codes for AP and AU**

### *Additive Selection Code A2*

| Character Sent (A2) | | Add 8 | Add 7 | Add 6 | Add 5 |
|---|---|---|---|---|---|
| Char | Hex | | | | |
| 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 |
| A | 41 | X | | X | |
| B | 42 | X | | X | X |
| C | 43 | X | X | | |
| D | 44 | X | X | | X |
| E | 45 | X | X | X | |
| F | 46 | X | X | X | X |

**Codes for AP and AU**

### *Additive Selection Code A3*

| Character Sent (A2) | | Add 12 | Add 11 | Add 10 | Add 9 |
|---|---|---|---|---|---|
| Char | Hex | | | | |
| 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 |
| A | 41 | X | | X | |
| B | 42 | X | | X | X |
| C | 43 | X | X | | |
| D | 44 | X | X | | X |
| E | 45 | X | X | X | |
| F | 46 | X | X | X | X |

**Codes for AP and AU**

### *Additive Selection Code A4*

| Character Sent (A2) | | Add 16 | Add 15 | Add 14 | Add 13 |
|---|---|---|---|---|---|
| Char | Hex | | | | |
| 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 |
| A | 41 | X | | X | |
| B | 42 | X | | X | X |
| C | 43 | X | X | | |
| D | 44 | X | X | | X |
| E | 45 | X | X | X | |
| F | 46 | X | X | X | X |

**Codes for AP and AU**

### *Additive Selection Code A5*

| Character Sent (A2) | | Add 20 | Add 19 | Add 18 | Add 17 |
|---|---|---|---|---|---|
| Char | Hex | | | | |
| 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 |
| A | 41 | X | | X | |
| B | 42 | X | | X | X |
| C | 43 | X | X | | |
| D | 44 | X | X | | X |
| E | 45 | X | X | X | |
| F | 46 | X | X | X | X |

**Codes for AP and AU**

### *Additive Selection Code A6*

| Character Sent (A2) | | Add 24 | Add 23 | Add 22 | Add 21 |
|---|---|---|---|---|---|
| Char | Hex | | | | |
| 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 |
| A | 41 | X | | X | |
| B | 42 | X | | X | X |
| C | 43 | X | X | | |
| D | 44 | X | X | | X |
| E | 45 | X | X | X | |
| F | 46 | X | X | X | X |

**Codes for AP and AU**

## *Command Code AR – Alarm Reset*

This command resets one of the alarms currently registered at the AccuLoad III.

**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 the specified injector |

Where: XX = Two-character alarm code
DD = Directory, where
SY = System
P1 = Product 1
P2 = Product 2
P3 = Product 3
P4 = Product 4
P5 = Product 5
P6 = Product 6
AR = Arm
M1 = Meter 1
M2 = Meter 2 (Ratio, hybrid blending configuration only)
M3 = Meter 3 (Ratio, hybrid blending configuration only)
M4 = Meter 4 (Ratio, hybrid blending configuration only)
M5 = Meter 5 (Ratio, hybrid blending configuration only)
M6 = Meter 6 (Ratio, hybrid blending configuration only)
IN = Additive Injector
RR = Recipe (01 – 50)
YY = Injector (01 – 24)

**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:** Poll & Authorize or Remote Control.

## *Alarm Status Codes That Can Be Reset Through Communications – System*

| Code | Condition |
|---|---|
| CF | ComFlash Alarm – Indicates a failure with the optional ComFlash. This alarm will be set if there is no SD card in the ComFlash board or if communications with the ComFlash fails. |
| CM | Communications Alarm – Communications failure on one of the communications channels. |
| CV | CIVACON Comm Failure – Indicates a failure on the communications line between the AccuLoad III and the CIVACON ground/overfill system. |
| MC | MMI Communications Failure – Indicates a communication failure between the MMI and FCM when using the AccuLoad III-SA. |
| ME | MMI Excess Arms Active – Indicates that more than six arms are active when using the AccuLoad III-SA. No more than six arms can be active at once. |
| PA | Power-fail Alarm – The unit either had a power failure or a hardware reset occurred. |
| PP | PTB Printer Failure – The PTB printer failure alarm is set when the AccuLoad III fails to get a good response from a PTB printer after sending data to be printed. The AccuLoad III will continue to resend the data to the printer until the communication port timeout setting expires, at which time the PTB printer alarm is set. |
| 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 timeout. |
| A1 | Add-Pak Power-fail – Indicates that a power failure has occurred on Add-Pak #1. |
| A2 | Add-Pak Power-fail – Indicates that a power failure has occurred on Add-Pak #2. |
| C1 | Add-Pak Communications Alarm – Communications failure on Add-Pak #1. |
| C2 | Add-Pak Communications Alarm – Communications failure on Add-Pak #2. |
| D1 | Add-Pak Diagnostic Alarm – Indicates a failure on Add-Pak #1. |
| D2 | Add-Pak Diagnostic Alarm – Indicates a failure on Add-Pak #2. |
| P1 | Add-Pak #1 Auto-Detect Failed – Indicates that the AccuLoad has failed to detect Add-Pak Board #1 |
| P2 | Add-Pak #2 Auto-Detect Failed – Indicates that the AccuLoad has failed to detect Add-Pak Board #2 |
| U1 | User Alarm #1 |
| U2 | User Alarm #2 |
| U3 | User Alarm #3 |
| U4 | User Alarm #4 |
| U5 | User Alarm #5 |
| U6 | User Alarm #6 |
| U7 | User Alarm #7 |
| U8 | User Alarm #8 |
| U9 | User Alarm #9 |
| U0 | User Alarm #10 |

**Resettable Alarms – System**

## *Alarm Status Codes That Can Be Reset Through Communications – Injector*

| Code | Condition |
|---|---|
| AD | Auto-Detect Failed – Indicates that a specific Smart Injector was unable to be located on any comm port. |
| FA | Additive Feedback Error – Indicates that the additive feedback has exceeded the programmed number of errors. |
| AC | Additive Communications Error – Indicates a failure on the master/slave communications line between the AccuLoad III and the Additive Injector Subsystem. |
| 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. |
| 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. |
| UA | "Unauthorize" Error – Indicates that the unauthorize command failed at the end of the batch for an additive. Authorization may have to be removed manually (by power cycling the additive system) to prevent unwanted additive in subsequent batches/transactions. |
| GA | General Additive Error – Indicates that there is an additive injector error. |
| OR | Overspeed Injector – Indicates that the meter on the metered injector has exceeded its specified maximum frequency. |
| CR | Command Refused Error – Indicates that the command from the AccuLoad to the injector was rejected. |
| AH | Additive High Temperature Alarm – Used only for flow controlled additive types. Temperature probe or transducer is out of range with high alarm setting. |
| AL | Additive Low Temperature Alarm – Used only for flow controlled additive types. Temperature probe or transducer is out of range with low alarm setting. |
| AT | Additive Temperature Probe – Used only for flow controlled additive types. Short or open condition in temperature probe. |
| PS | Pulse Security – Used only with the security option. Indicated that an excessive number of out-of-sequence errors in the A-B pulse stream have been detected on one of the flow controlled additive meters. |
| XA | Indicates a collision in the incoming pulse stream. (The signals on A and A-bar or B and B-bar are the same). |
| CT | Indicates that the additive total received from the smart additive system may be erroneous |

**Resettable Alarms - Injector**

## *Alarm Status Codes That Can Be Reset Through Communications – Arm*

| Code | Condition |
|---|---|
| CA | Additive Clean Line – The additive clean line volume delivered has under-run the additive clean line volume programmed. |
| CL | Clean Line – The clean line volume delivered has under-run the clean line volume programmed by at least the amount of clean line alarm limit. |
| DE | DE Head Alarm (unloading only) – Indicates an illegal combination of inputs. If the low flow switch is covered, the stop float must be covered. If the high flow switch is covered, both the stop float and the low flow switches must also be covered. |
| HF | High Flow – Flow rate has exceeded limit set by Excess High Flow program parameter for more than 4 seconds. |
| OA | Overrun – Volume delivered has exceeded the preset amount by at least the number of units set in the Overrun Limit program code. |
| SF | Storage Full – Indicates that the archived transaction area is full. The operator must clear stored standby transactions in order to continue. |
| TK | Ticket – Ticket was cranked from ticket printer while the batch loading was in progress. |
| ZF | Zero Flow – The AccuLoad did not see flow through the system before the zero flow timer expired. |
| CD | Card Removed – The card was removed from the card reader prior to the end of the transaction. |

**Resettable Alarms – Arm**

## *Alarm Status Codes That Can Be Reset Through Communications – Meter*

| Code | Condition |
|---|---|
| DR | Density Transducer – Density transducer failure or out-of-range condition |
| FR | Indicates that reverse flow occurring during a batch has exceeded the programmed “reverse flow limit” (system 203). |
| LA | Leakage Alarm – Indicates that leakage between transactions has exceeded the programmed “leakage alarm limit” (system 202). |
| MM | 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.) |
| PM | The Promass Meter has a system problem status alarm. This alarm indicates that the meter should be checked for programming errors. |
| PO | Predict Overrun – Indicates that the AccuLoad has stopped the batch because it suspects a valve fault condition will result in an overrun occurring. |
| 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 |
| SC | Solenoid Count – Indicates that the solenoid count has exceeded the “solenoid alarm count” (system 201). This alarm will only be set when the load arm is idle. |
| TP | Temperature Probe – Short or open condition in the temperature probe circuit |
| TM | Turbine Meter |
| VF | Valve Fault – Indicates that the meter was still registering flow when the valve fault timer expired after the AccuLoad commanded the valve to close |
| XA | Indicates a collision in the incoming pulse stream. (The signals on A and A-bar or B and B-bar are the same) |

**Resettable Alarms - Meter**

## *Alarm Status Codes That Can Be Reset Through Communications – Product*

| Code | Condition |
|---|---|
| BH | Blend High Alarm – An overflow on this product has caused a blend tolerance to be exceeded. |
| BL | Blend Low Alarm – An underflow on this product has caused the blend tolerance to be exceeded. |
| BP | Back Pressure – Back pressure per the entries set cannot be maintained during flow enough to maintain entry set in product program code. |
| BV | Block Valve Alarm – The valve did not close within 10 seconds after receiving the signal to close. |
| HD | High Density – Density transducer is out of range of the high alarm setting. |
| HF | High Flow Alarm – Flow rate has exceeded limit set by Excess High Flow program code 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 than 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. |
| OA | Overrun Alarm – Volume delivered has exceeded the preset amount by at least the number of units set in the overrun limit program code. |
| PA | Product Stop Alarm – Hybrid Blending Arms only – This alarm indicates that there was an under-run of product after the ratio product was shut down. Therefore, the line may not be clear of the ratio product. |
| UF | Unauthorized Flow – The AccuLoad has detected unauthorized flow. |
| ZF | Zero Flow – The AccuLoad did not see flow through the system before the zero flow timer expired. |

**Resettable Alarms - Product**

## *Command Code AT – Request Additive Volumes by Transaction*

This command retrieves additive transaction volumes from AccuLoad III.

**Command:**

**Current Transaction**

| | |
|---|---|
| **"AT_XX"** | Cumulative transaction additive volume |
| **"AT_XX_Z"** | Total additive, by volume type, of transaction in progress if that type is available |

**Local Storage Transaction**

| | |
|---|---|
| **"AT_XX_NNN"** | Total additive in an historic transaction |
| **"AT_XX_Z_NNN"** | Total additive, by volume type, in an historic transaction for flow controlled additives |

Where: XX = 1 through 24 (Additive #)
Z = G (GV Total); N (GST Total); M (Mass Total)
NNN = number of transactions back into local storage to retrieve the data

**Responses:**

**Good Response:**

**Current Transaction**

| | |
|---|---|
| **"AT_XX_YY_VVVVV.VVV"** | For "AT_XX" |
| **"AT_XX_YY_Z_VVVVV.VVV"** | For "AT_XX_Z" |

Local Storage Transaction

| | |
|---|---|
| **"AT_XX_YY_VVVVV.VVV_NNN"** | For "AT_XX_YY_NN' |
| **"AT_XX_YY_Z_VVVVV.VVV_NNN"** | For "AT_XX_YY_Z_NNN" |

Where:

| | |
|---|---|
| XX | = 1 through 24 (Additive #) |
| YY | = Total Number of Batches Completed |
| VVVVV.VVV | = Total Additive Transaction Volume |
| Z | = R for Raw or Indicated (IV) Total<br>G for Gross (GV) Total<br>N for Gross at Standard Temperature (GST) Total<br>M for Mass Total |
| NNN | = # of transactions 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 AccuLoad III. Additive must be assigned to the arm that is addressed. Volume types are available for flow control additives only.

**Special Case:** If bays are configured, the returned value is the bay transaction total.

**Comm. Modes:** Polling only, Poll & Authorize, Remote Control, Poll and Program.

## *Command Code AU – Authorize Transaction*

This command authorizes a transaction at the AccuLoad III. The AccuLoad will continue to display READY until the SET key is pressed. Then the "PRESET" or recipe selection prompt will be displayed.

**Command:**

**"AU"** Authorize AccuLoad

**"AU_A1A2A3A4A5A6"** Authorize AccuLoad with Additives

Where:

A1A2A3A3A4A5A6 = the optional, six-character additive selection code.
(See tables after AP command).

**Responses:**

**"OK"** Good Response. The AccuLoad III has been authorized for the transaction.

or...

**"NOXX"** The transaction was not authorized.

(NO30 will be returned if the additive specified is not currently allocated to the arm to which the command was directed.)

**Remarks:**

1. If the additive qualifier (A1A2…) is not used, the Additive selection will operate as currently programmed in the AccuLoad III. For authorization with additives, only one recipe may be allotted. Additives selected in the qualifier must be programmed for use.
2. If more than one recipe is enabled, the "Select Recipe" prompt will be displayed.

**Constraints:** The "AU" will reset the transaction done, batch done and keypad pending flags. Additive must be assigned to the arm that is being authorized.

**Special Case:** None.

**Comm. Modes:** Poll & Authorize.

## *Command Code AV – Request Additive Volumes by Batch*

This command retrieves additive batch volumes from AccuLoad III.

**Command:**

**Current Transaction**

| | |
|---|---|
| **"AV_XX"** | Total additive of batch in progress (delivery type for flow controlled additive) |
| **"AV_XX_Z"** | Total additive, by volume type, of batch in progress if that type is available |
| **"AV_XX_YY"** | Total additive of a specified batch |
| **"AV_XX_YY_Z"** | Total additive of specified batch, specified volume type if volume was stored |

**Local Storage Transaction**

| | |
|---|---|
| **"AV_XX_YY_NNN"** | Total additive of a specified batch in an historic transaction |
| **"AV_XX_YY_Z_NNN"** | Total additive, volume type, of specified batch in an historic transaction for flow controlled additives |

Where: XX = 1 through 24 (Additive #)
YY = Batch Number Requested
Z = G (GV Total); N (GST Total); M (Mass Total)
NNN = number of transactions back into Local Storage to retrieve the data

**Responses:**

**Good Response**

| | |
|---|---|
| **"AV_XX_YY_VVVVV.VVV"** | For "AV_XX", "AV_XX_YY" |
| **"AV_XX_Z_VVVVV.VVV"** | For "AV_XX_Z" |
| **"AV_XX_YY_Z_VVVVV.VVV"** | For "AV_XX_YY_Z" |

**Local Storage Transaction**

| | |
|---|---|
| **"AV_XX_YY_VVVVV.VVV_NNN"** | For "AV_XX_YY_NN" |
| **"AV_XX_YY_Z_VVVVV.VVV_NNN"** | For "AV_XX_YY_Z_NNN" |

Where:
XX = 1 through 24 (Additive #)
YY = Batch number requested
Z = R for Raw or Indicated (IV) Total
G for Gross (GV) Total
N for Gross at Standard Temperature (GST) Total
M for Mass Total
VVVVV.VVV = Total Additive Batch Volume
NNN = Number of transactions back into Local Storage to retrieve the data

or...

**"NOXX"**

**Remarks:** If bays are configured, the AV for the current batch returns the current batch on the arm addressed (or NO39 if no batch is in progress for the arm). For an AV command for a completed batch, the batch argument specified in the batch is the bay-based transaction, and hence may return a batch completed on an arm that is not the arm being addressed.

**Constraints:** Volume units are assumed as the type programmed into AccuLoad III. Volume types are available for flow controlled additives only.

**Special Case:** None

**Comm. Modes:** Polling only, Poll & Authorize, Remote Control, Poll and Program

## *Command Code BA – Bay Assignment*

This command retrieves the position of each arm in the bay.

**Command:**

"BA"

**Responses:**

**Good Response:**

**"BA_A1A2A3A4A5A6"**

Where:

An ***n*** is a character representing the position of arm ***n***:
'A' = Bay A
'B' = Bay B
'?' = In transit

or...

**"NOXX"**

**Remarks:** None

**Constraints:** None

**Special Case:** None

**Comm. Modes:** Polling only, Poll & Authorize, Remote Control, Poll and Program

## *Command Code BR – Boolean/Algebraic Variable Read*

This command retrieves the Boolean/Algebraic data from AccuLoad III.

**Command:**

**"BR_X_YYY"** Reads the Boolean / Algebraic Variable.

Where: X = F – Algebraic Variable (single precision, floating point)
B – Boolean Variable
T – Timer Variable
S – String* (for revision II & above)

YYY = Variable number

**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

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

or...

**"NOXX"**

**Remarks:** String can be up to 32 characters in length.

**Constraints:** None

**Special Case:** None

**Comm. Modes:** Polling only, Poll & Authorize, Remote Control, Poll and Program.

## *Command Code BW – Boolean/Algebraic Variable Write*

This command writes to the Boolean/Algebraic registers in AccuLoad III.

**Command:**

**"BW_X_YYY_D…D"** Writes to the Boolean/Algebraic Registers.

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* (revision II & above)

YYY = Variable number
D…D = The data; up 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:**

**"BW_X_YYY_D…D"**

Where: X = F – Algebraic Variable (floating point)
B – Boolean Variable
T – Timer Variable
A – Set User Alarm (no _D…D is returned)
S – String
YYY = Variable number
D…D = The data; up to 255 for Boolean variables and numeric string for algebraic and Timer variables

or...

**"NOXX"**

**Remarks:** Strings can be up to 32 characters in length.

**Constraints:** None.

**Special Case:** None.

**Comm. Modes:** Polling only, Poll & Authorize, Remote Control, Poll & Program.

## *Command Code CD – Card Reader Data*

This command returns card reader data to the host.

**Command:**

**"CD"**
**"CD_C1C2"**
**"CD_NNN"**
**"CD_V"**

Where:
C1C2 (optional) is a two-character bitmap
NNN (optional is the number of transactions back in storage)

**Responses:**

**Good Response:**

| | |
|---|---|
| **"CD_S1_HHHHHHHHH"** | For CD (and CD_C1C2 when no card read error) |
| **"CD_VVVVVV_K"** | |
| **"CD_VVVVVV_K"** | For CD_V |
| **"CD_S1"** | For CD_C1C2 (when card read error) |
| **"CD_HHH_N…_NNN"** | For CD_NNN |

Where:
S1 is a one-character status indication – 0 = Data good
S1 = 1 – Timed out (after 3 seconds from the time the card was swiped)

HHHHHHHHH is the data encoded on the card
NNN is the number of transactions back
Both 26-bit and 37-bit card data formats are supported
'VVVVVV' is the tag ID returned for the vehicle, or alternatively the entered response to the prompt for Vehicle ID.
'K' is a terminating character; 'R' – if data comes from a Nedap Reader; 'E'- if entered as a prompt response

For the 26-bit data format, the nine characters of card data are returned as follows:

002NNXXXX

Where:
002 is fixed and is added to the data read from the card
NN is the facility code in hexadecimal (decimal range is 0 – 255)
XXXX is the card ID in hexadecimal (decimal range is 0 – 65535)

For the 37-bit data format, the nine characters of card data can consist of any digits 0-9, A-F

**Remarks:** None.

**Constraints:** Override bit must be set to control I/O on the card reader. Override bit cleared returns control of the I/O to the card reader. Red and green LED both on, equal amber LED on reader.

**Special Case:** None.

## *CD Command, Optional Character C1*

| Character | | Hex Value | | | |
|---|---|---|---|---|---|
| **Char** | **Hex** | **0x08 Override I/O Control** | **0x04 Reserved for Future Use** | **0x02 Reserved for Future Use** | **0x01 Reserved for Future Use** |
| 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 CD - Character C1**

***Note:*** *Override bit must be set to control I/O on card reader. Override bit cleared returns control of these I/O to the card reader.*

## *CD Command, Optional Character C2*

| Character | | Hex Value | | | |
|---|---|---|---|---|---|
| **Char** | **Hex** | **0x08 Contact Closed** | **0x04 Beep** | **0x02 Red LED On** | **0x01 Green LED On** |
| 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 CD - Character C2**

***Note:*** *Red and Green LED both On = Amber LED on Reader.*

### *Response to CD Command, Character S1*

| Character | | Hex Value | | | |
|---|---|---|---|---|---|
| **Char** | **Hex** | **0x08 Reserved for Future Use** | **0x04 Reserved for Future Use** | **0x02 Card Validated** | **0x01 Card Read 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 CD - Character S1**

## *Command Code CF – AccuLoad Configuration*

This command provides the hardware configuration of the requested AccuLoad. This corresponds to the hardware configuration jumpers on the KDC boards (J23 and J24). If J23 is present, this implies that there is a BSE available on the unit. If J24 is present, this indicates that there is a second display available to the AccuLoad III.

**Command:**

**"CF"**

**Responses:**

**Good Response:**

**"CF_xy"**

Where: X = J23 status
Y = J24 status

or

**"NOXX"**

**Remarks:** These parameters may take one of two values. An ASCII zero will indicate that the jumper is not present, while an ASCII 1 will indicate that the jumper is present.

*Example:* An AccuLoad is configured as a 6-product ratio blender without a second display. The response should be as follows: CF 10.

**Constraints:** None

**Special Case:** None

**Comm. Modes:** Polling only, Poll & Authorize, Remote Control, Poll & Program

## *Command Code CP – Request Contaminant Percentage*

This command requests the percentage of contaminant from the AccuLoad III.

**Command:**

**Current Transaction**

| | |
|---|---|
| **"CP"** | Current percentage of contaminant for the current batch |
| **"CP_YY"** | Percentage of contaminant for a specified batch |

**Local Storage**

| | |
|---|---|
| **"CP_YY_NNN"** | Returns the requested percentage of contaminant for a previously completed batch |

Where:

YY = Batch Number (01 – 10)
NNN = Number of transactions back into Local Storage to retrieve data

**Responses:**

**Current Transaction**

| | |
|---|---|
| **"CP_YY_XXX.X"** | Good response |

**Local Storage**

| | |
|---|---|
| **"CP_YY_XXX.X_NNN"** | Good response |

Where:

YY = Batch Number (01 – 10)
XXX.X = Percentage of contaminant
NNN = Number of transactions back into Local Storage to retrieve data

or...

| | |
|---|---|
| **"NOXX"** | The percentage of contaminant was not read |

Where:

XX = "03" Value rejected
= "05" No transaction ever done
= "19" Option not installed

**Remarks:** None

**Constraints:** Applies to arms configured as unloading arms only. Other arm configurations will result in NO19 being returned.

**Special Case:** None

**Comm. Modes:** Polling only, Poll & Authorize, Remote Control, Poll & Program

## *Command Code CT – Clear Transactions Locked in Standby Mode*

This command resets the transaction archive markers that track transactions completed in standby mode. This restores normal transaction archive operation and re-enables read access to all of the transactions in the archive. This command also resets the Standby Transactions (ST) status.

**Command:**

**"CT"**

**Responses:**

**Good Response:**

**"OK"**

**Remarks:**

1. When the ST status is asserted indicating standby transactions exist, only those standby transactions may be accessed via host data retrieval commands such as RT, RB, TN, etc. This is to prevent loss of transaction data generated when the host was offline.

2. If bays are configured, the command clears all transactions for the bay.

**Constraints:** None

**Special Case:** None

**Comm. Modes:** Poll & Authorize, Remote Control

## *Command Code CV – CIVACON Status*

This command returns the status of the CIVACON monitor.

**Command:**

**"CV"**

**Responses:**

**Good Response:**

**"CV_X_Y_Z_C1C2C3C4"**

| | | |
|---|---|---|
| Where: | X | = 0 (sensors are wet) or 8 (no sensors are wet) (prefill detection status) |
| | Y | = 0 (not grounded) or 4 (grounded) (ground input status) |
| | Z | = 0 (not bypassed) or 1 (CIVACON bypassed) (bypass status) |
| | C1C2C3C4 | = Two-character hexadecimal bitmaps |

or

**"NOXX"**

**Remarks:** None

**Constraints:** None

**Special Case:** None

**Comm. Modes:** Polling only, Poll & Authorize, Remote Control, Poll & Program

### *Response to CV Command, Character C1*

| Character | | Hex Value | | | |
|---|---|---|---|---|---|
| **Char** | **Hex** | **Thermistor Sensors** | **Optic Sensors** | **Pulse to Truck** | **Pulse from Truck** |
| 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 CV - Character C1**

### *Response to CV Command, Character C2*

| Character | | Hex Value | | | |
|---|---|---|---|---|---|
| **Char** | **Hex** | **Sensor 4** | **Sensor 3** | **Sensor 2** | **Sensor 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 CV - Character C2**

### *Response to CV Command, Character C3*

| Character | | Hex Value | | | |
|---|---|---|---|---|---|
| **Char** | **Hex** | **Sensor 8** | **Sensor 7** | **Sensor 6** | **Sensor 5** |
| 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 CV - Character C3**

### *Response to CV Command, Character C4*

| Character | | Hex Value | | | |
|---|---|---|---|---|---|
| **Char** | **Hex** | **Sensor 12** | **Sensor 11** | **Sensor 10** | **Sensor 9** |
| 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 CV - Character C4**

## *Command Code DA – Release Keypad and Display*

This command returns the control of the keypad and display to the AccuLoad III.

**Command:**

| |
|---|
| **"DA"** |

**Responses:**

| | |
|---|---|
| **"OK"** | **Good Response.** Keypad and display have been released to AccuLoad III and the keypad data pending flag is reset. |
| or... | |
| **"NOXX"** | The keypad and display were not released. |

**Remarks:** None.

**Constraints:** A "NO01" response will be returned if another arm is in the program mode locally or via communications.

**Special Case:** None.

**Comm. Modes:** Poll & Authorize or Remote Control.

## *Command Code DD – Delete Driver Card Data*

This command removes a driver card from the list of valid cards in the AccuLoad III-X. It modifies the current run database but does not update the table in non-volatile memory. After any combination of DI and DD commands are sent, the DU command should be used to make the changes permanent by writing the data to non-volatile (flash) memory.

**Command:**

**"DD_HHHH…H"**

Where:
HHHH…H is the card data value for the record to delete

**Responses:**

**Good Response:**

**"OK"**

or

**"NOXX"**

**Remarks:** None.

**Constraints:** None.

**Special Case:** None.

**Comm. Modes:** Polling only, Poll & Authorize, Remote Control, Poll & Program.

## *Command Code DI – Insert Driver Card Data*

The DI command inserts (adds) a driver card to the list of valid cards in the AccuLoad III-X. It modifies the current run database but does not update the table in non-volatile memory. After any combination of DI and DD commands are sent, the DU command should be used to make the changes permanent by writing the data for non-volatile (flash) memory.

**Command:**

**"DI_HHHH…H"**

Where:
HHHH…H is the card data value for the record to be inserted

**Responses:**

**Good Response:**

**"OK"**

or

**"NOXX"**

**Remarks:** None.

**Constraints:** None.

**Special Case:** None.

**Comm. Modes:** Polling only, Poll & Authorize, Remote Control, Poll & Program.

## *Command Code DQ – Query Driver Card Data*

The DQ command returns the card data for the specified record index. This command is included to allow a host to retrieve the list of valid cards from the AccuLoad III-X. Note that the index for any one card entry may change when records have been inserted or deleted, as they are maintained in sorted order internally.

**Command:**

**"DQ_NNN"**

Where:
NNN is the index of the card database record that is being requested

**Responses:**

**Good Response:**

**"DQ_HHHH…H"**

Where:
HHHH…H is the card data for the requested record

or

**"NOXX"**

**Remarks:** None.

**Constraints:** None.

**Special Case:** None.

**Comm. Modes:** Polling only, Poll & Authorize, Remote Control, Poll & Program.

## *Command Code DS – Display Density Sample*

This command retrieves one of ten density samples from the AccuLoad III for the current batch.

**Command:**

| | |
|---|---|
| **"DS_X"** | For the current batch – returns the density sample requested<br>Where X = Specific Density Sample (0 – 9) |

**Responses:**

**Good Response**
**"DS_X_VVVV.V"**

Where: X = 0 – 9, # of density sample
VVVV.V = Density of sample in programmed density units

or...

**"NOXX"**
Where: "XX" = "03" Value rejected
= "05" No transaction ever done
= "19" Option not installed

**Remarks:** None

**Constraints:** Only available with Unloading arms

**Special Case:** None

**Comm. Modes:** Polling only, Poll & Authorize, Remote Control, Poll & Program

## *Command Code DU – Update Driver Database in Non-Volatile Memory*

The DU command rebuilds the driver card data table in non-volatile memory, after freeing any memory used by deleted records and re-sorting the table by card number. This command is to be used by the host after the host has issued any combination of DI or DD commands to modify the driver card database at the AccuLoad III-X. Failure to issue this command after modifying the database via DI or DD will cause those changes to be lost if the AccuLoad is powered down.

**Command:**

**"DU"**

**Responses:**

**Good Response:**

**"OK"**

or

**"NOXX"**

**Remarks:** None.

**Constraints:** None.

**Special Case:** None.

**Comm. Modes:** Polling only, Poll & Authorize, Remote Control, Poll & Program.

## *Command Code DY – Request Dynamic Display Values*

This command retrieves a current dynamic display value from the AccuLoad III.

**Command:**

**"DY_ddxx"**

Where: dd = dynamic display type
SY = system
IN = injector
P1 = product #1
P2 = product #2
P3 = product #3
P4 = product #4
P5 = product #5
P6 = product #6
CP = current product (straight and sequential blending only)
RR= recipe number (01 – 50 for a specific recipe)
Bz = batch, where "z" is 1…9 for batches 1-9, and "A" for batch 10
CB = current batch
CR = current recipe
TR = transaction
FA = flow controlled additives
xx = variable number to access

**Responses:**

**"DY_V..V_D..D"** Good Response.

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

**"NOXX"** Requested display not returned.

**Remarks:** No response exceeds 31 characters.

For bay configurations, the following information applies:

1. All additives on the bay can be accessed via any valid arm address located on the bay.
2. Batch values are for the bay-based batch.
3. Not all transaction "DY" values are available when bays are configured, i.e., average temperature, pressure, meter factor, etc.

**Constraints:** NO06 will be returned if the requested batch has not been delivered (Bz qualifier).
NO05 will be returned if there is no transaction in progress and no transaction has ever been done (TR qualifier).
NO30 will be returned if a product or recipe is requested which is not currently allocated to the arm to which the command was directed.
NO31 will be returned if the command format does not match the current configuration for the arm.

**Special Case:** None.

**Comm. Modes:** Polling only, Poll & Authorize, Remote Control, or Poll & Program.

### *System "DY_SYXX" Dynamic Display Values*

| Index Number ("XX") | Description | Response |
|---|---|---|
| 00 | Current Flow Rate Units/Min, Arm 1 | DY Flow (Arm 1)XXXXX.X Gal/Min |
| 01 | Current Flow Rate Units/Min, Arm 2 | DY Flow (Arm 2)XXXXX.X Gal/Min |
| 02 | Current Flow Rate Units/Min, Arm 3 | DY Flow (Arm 3)XXXXX.X Gal/Min |
| 03 | Current Flow Rate Units/Min, Arm 4 | DY Flow (Arm 4)XXXXX.X Gal/Min |
| 04 | Current Flow Rate Units/Min, Arm 5 | DY Flow (Arm 5)XXXXX.X Gal/Min |
| 05 | Current Flow Rate Units/Min, Arm 6 | DY Flow (Arm 6)XXXXX.X Gal/Min |
| 06 | Current Flow Rate Units/Hr, Arm 1 | DY Flow (Arm 1)XXXXXXX.X Gal/Hr |
| 07 | Current Flow Rate Units/Hr, Arm 2 | DY Flow (Arm 2)XXXXXXX.X Gal/Hr |
| 08 | Current Flow Rate Units/Hr, Arm 3 | DY Flow (Arm 3)XXXXXXX.X Gal/Hr |
| 09 | Current Flow Rate Units/Hr, Arm 4 | DY Flow (Arm 4)XXXXXXX.X Gal/Hr |
| 10 | Current Flow Rate Units/Hr, Arm 5 | DY Flow (Arm 5)XXXXXXX.X Gal/Hr |
| 11 | Current Flow Rate Units/Hr, Arm 6 | DY Flow (Arm 6)XXXXXXX.X Gal/Hr |
| 12 | Current Recipe Name, Arm 1 | DY Recipe (Arm 1)  XXXXXXXXX |
| 13 | Current Recipe Name, Arm 2 | DY Recipe (Arm 2)  XXXXXXXXX |
| 14 | Current Recipe Name, Arm 3 | DY Recipe (Arm 3)  XXXXXXXXX |
| 15 | Current Recipe Name, Arm 4 | DY Recipe (Arm 4)  XXXXXXXXX |
| 16 | Current Recipe Name, Arm 5 | DY Recipe (Arm 5)  XXXXXXXXX |
| 17 | Current Recipe Name, Arm 6 | DY Recipe (Arm 6)  XXXXXXXXX |
| 18 | Current Preset, Arm 1 | DY Preset (Arm 1)  XXXXXX Gal |
| 19 | Current Preset, Arm 2 | DY Preset (Arm 2)  XXXXXX Gal |
| 20 | Current Preset, Arm 3 | DY Preset (Arm 3)  XXXXXX Gal |
| 21 | Current Preset, Arm 4 | DY Preset (Arm 4)  XXXXXX Gal |
| 22 | Current Preset, Arm 5 | DY Preset (Arm 5)  XXXXXX Gal |
| 23 | Current Preset, Arm 6 | DY Preset (Arm 6)  XXXXXX Gal |
| 24 | Current Delivered Volume, Arm 1 | DY Deliv. (Arm 1) XXXXXXX.XX Gal |
| 25 | Current Delivered Volume, Arm 2 | DY Deliv. (Arm 2) XXXXXXX.XX Gal |
| 26 | Current Delivered Volume, Arm 3 | DY Deliv. (Arm 3) XXXXXXX.XX Gal |
| 27 | Current Delivered Volume, Arm 4 | DY Deliv. (Arm 4) XXXXXXX.XX Gal |
| 28 | Current Delivered Volume, Arm 5 | DY Deliv. (Arm 5) XXXXXXX.XX Gal |
| 29 | Current Delivered Volume, Arm 6 | DY Deliv. (Arm 6) XXXXXXX.XX Gal |
| 30 | Current Remaining Volume, Arm 1 | DY Remain. (Arm 1) XXXXXXX.XX Gal |
| 31 | Current Remaining Volume, Arm 2 | DY Remain. (Arm 2) XXXXXXX.XX Gal |
| 32 | Current Remaining Volume, Arm 3 | DY Remain. (Arm 3) XXXXXXX.XX Gal |

| Index Number ("XX") | Description | Response |
|---|---|---|
| 33 | Current Remaining Volume, Arm 4 | DY Remain. (Arm 4) XXXXXXX.XX Gal |
| 34 | Current Remaining Volume, Arm 5 | DY Remain. (Arm 5) XXXXXXX.XX Gal |
| 35 | Current Remaining Volume, Arm 6 | DY Remain. (Arm 6) XXXXXXX.XX Gal |
| 36 | Time of Last Power Fail | DY HH:MM:SS AM MM-DD-YY |

## *Injector Dynamic Displays*

***Note:*** *If Injectors 1 through 4 are flow control type, responses to DY_INXX appear in the table immediately following the table below.*

| 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 5 Current Pulse Rate | DY Inj 5 Cal XXXX.XX |
| 05 | Injector 6 Current Pulse Rate | DY Inj 6 Cal XXXX.XX |
| 06 | Injector 7 Current Pulse Rate | DY Inj 7 Cal XXXX.XX |
| 07 | Injector 8 Current Pulse Rate | DY Inj 8 Cal XXXX.XX |
| 08 | Injector 9 Current Pulse Rate | DY Inj 9 Cal XXXX.XX |
| 09 | Injector 10 Current Pulse Rate | DY Inj 10 Cal XXXX.XX |
| 10 | Injector 11 Current Pulse Rate | DY Inj 11 Cal XXXX.XX |
| 11 | Injector 12 Current Pulse Rate | DY Inj 12 Cal XXXX.XX |
| 12 | Injector 13 Current Pulse Rate | DY Inj 13 Cal XXXX.XX |
| 13 | Injector 14 Current Pulse Rate | DY Inj 14 Cal XXXX.XX |
| 14 | Injector 15 Current Pulse Rate | DY Inj 15 Cal XXXX.XX |
| 15 | Injector 16 Current Pulse Rate | DY Inj 16 Cal XXXX.XX |
| 16 | Injector 17 Current Pulse Rate | DY Inj 17 Cal XXXX.XX |
| 17 | Injector 18 Current Pulse Rate | DY Inj 18 Cal XXXX.XX |
| 18 | Injector 19 Current Pulse Rate | DY Inj 19 Cal XXXX.XX |
| 19 | Injector 20 Current Pulse Rate | DY Inj 20 Cal XXXX.XX |
| 20 | Injector 21 Current Pulse Rate | DY Inj 21 Cal XXXX.XX |
| 21 | Injector 22 Current Pulse Rate | DY Inj 22 Cal XXXX.XX |
| 22 | Injector 23 Current Pulse Rate | DY Inj 23 Cal XXXX.XX |
| 23 | Injector 24 Current Pulse Rate | DY Inj 24 Cal XXXX.XX |
| 24 | Injector 1 Programmed Pulse Rate | DY Inj 1 Prg XXXX.XX |
| 25 | Injector 2 Programmed Pulse Rate | DY Inj 2 Prg XXXX.XX |

| Index Number ("XX") | Description | Response |
|---|---|---|
| 26 | Injector 3 Programmed Pulse Rate | DY Inj 3 Prg XXXX.XX |
| 27 | Injector 4 Programmed Pulse Rate | DY Inj 4 Prg XXXX.XX |
| 28 | Injector 5 Programmed Pulse Rate | DY Inj 5 Prg XXXX.XX |
| 29 | Injector 6 Programmed Pulse Rate | DY Inj 6 Prg XXXX.XX |
| 30 | Injector 7 Programmed Pulse Rate | DY Inj 7 Prg XXXX.XX |
| 31 | Injector 8 Programmed Pulse Rate | DY Inj 8 Prg XXXX.XX |
| 32 | Injector 9 Programmed Pulse Rate | DY Inj 9 Prg XXXX.XX |
| 33 | Injector 10 Programmed Pulse Rate | DY Inj 10 Prg XXXX.XX |
| 34 | Injector 11 Programmed Pulse Rate | DY Inj 11 Prg XXXX.XX |
| 35 | Injector 12 Programmed Pulse Rate | DY Inj 12 Prg XXXX.XX |
| 36 | Injector 13 Programmed Pulse Rate | DY Inj 13 Prg XXXX.XX |
| 37 | Injector 14 Programmed Pulse Rate | DY Inj 14 Prg XXXX.XX |
| 38 | Injector 15 Programmed Pulse Rate | DY Inj 15 Prg XXXX.XX |
| 39 | Injector 16 Programmed Pulse Rate | DY Inj 16 Prg XXXX.XX |
| 40 | Injector 17 Programmed Pulse Rate | DY Inj 17 Prg XXXX.XX |
| 41 | Injector 18 Programmed Pulse Rate | DY Inj 18 Prg XXXX.XX |
| 42 | Injector 19 Programmed Pulse Rate | DY Inj 19 Prg XXXX.XX |
| 43 | Injector 20 Programmed Pulse Rate | DY Inj 20 Prg XXXX.XX |
| 44 | Injector 21 Programmed Pulse Rate | DY Inj 21 Prg XXXX.XX |
| 45 | Injector 22 Programmed Pulse Rate | DY Inj 22 Prg XXXX.XX |
| 46 | Injector 23 Programmed Pulse Rate | DY Inj 23 Prg XXXX.XX |
| 47 | Injector 24 Programmed Pulse Rate | DY Inj 24 Prg XXXX.XX |

## *Flow Control Type Response*

| Index Number ("XX") | Description | Response |
|---|---|---|
| 00 | Injector 1 Current Pulse Rate | DY Meter Inj 1 XXX.X Gal/per min |
| 01 | Injector 2 Current Pulse Rate | DY Meter Inj 2 XXX.X Gal/per min |
| 02 | Injector 3 Current Pulse Rate | DY Meter Inj 3 XXX.X Gal/per min |
| 03 | Injector 4 Current Pulse Rate | DY Meter Inj 4 XXX.X Gal/per min |

### *Product "DY_PNXX" Dynamic Display Values*

| Index Number ("XX") | Description | Response |
|---|---|---|
| 00 | Current Flow Rate, Units/Minute | DY Flow XXXXX.X Gal/Min |
| 01 | Current Flow Rate, Units/Hour | DY Flow XXXXXXX.X Gal/Hr |
| 02 | Batch Average Temperature | DY Temperature SXXXX.X F |
| 03 | Batch Average Density | DY Density XXXX.X Kg/M3 |
| 04 | Batch Average API | DY API SXXX.X |
| 05 | Batch Average Reference Density | DY Ref Dens XXXX.X Lb/F3 |
| 06 | Batch Average Relative Density | DY Rel Dens X.XXXX |
| 07 | Batch Average Pressure | DY Pressure XXXX.X PSI |
| 08 | Batch Average Vapor Pressure | DY Avg Vapor Press XXXX.X PSI |
| 09 | Batch Average Meter Factor | DY Cur Meter Factor X.XXXXX |
| 10 | Batch Average CTL | DY Batch Avg CTL X.XXXXX |
| 11 | Batch Average CPL | DY Batch Avg CPL X.XXXXX |
| 12 | Current Temperature | DY Current Temp SXXXX.X F |
| 13 | Current Density | DY Current Density SXXXX.X Kg/M3 |
| 14 | Current Meter Factor | DY Current Meter Fact. X.XXXXX |
| 15 | Current Valve Requested Position | DY Valve Requested Closed |
| 16 | Current Percentage of Batch | DY Current % XXX % |
| 17 | Desired Percentage of Batch | DY Desired % XXX % |
| 18 | Current Batch Raw Volume | DY Raw Batch XXXXXXX.XX Gal |
| 19 | Current Batch Gross Volume | DY Gross Batch XXXXXXX.XX Gal |
| 20 | Current Batch GST Volume | DY GST Batch XXXXXXX.XX Gal |
| 21 | Current Batch GSV Volume | DY GSV Batch XXXXXXX.XX Gal |
| 22 | Current Batch Mass Total | DY Mass Batch XXXXXXX.XX Lbs |
| 23 | Transaction Raw Volume | DY Raw Trans XXXXXXX.XX Gal |
| 24 | Transaction Gross Volume | DY Gross Trans XXXXXXX.XX Gal |
| 25 | Transaction GST Volume | DY GST Trans XXXXXXX.XX Gal |
| 26 | Transaction GSV Volume | DY GSV Trans XXXXXXX.XX Gal |
| 27 | Transaction Mass Total | DY Mass Trans XXXXXXX.XX Lbs |
| 28 | Current Reference Density (Calculated) | DY Cur Ref Dens XXXX.X Lb/F3 |
| 29 | Batch Average CTPL | DY Batch Avg CTPL X.XXXXX |
| 30 | Instantaneous Blend Percentage | DY Inst Blend XXX.XX% |
| 31 | Deviation Count | DY Deviation Cnt XXXXXXX.X |
| 32 | Ref Dens @ Ref Dens Temp | DY Ref Dens@DensTemp XXXX.X Lb/Ft3 |

| Index Number ("XX") | Description | Response |
|---|---|---|
| 33** | Avg Rel Dens@60F & Prs | DY Avg Rel Dens@60F & Prs X.XXXX |

** E tables only (relative density @60F & current pressure) used in CPL calculation.

### *Recipe "DY_DDXX" 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 Total | DY R#XX Mass Trans XXXXXXX Gal |

### *Batch "DY_BNXX" Dynamic Display Values*

| Index Number ("XX") | Description | Response |
|---|---|---|
| 00 | Recipe Name and Number | DY Recipe XX = XXXXXXXXX |
| 01 | Indicated (Raw) Batch Volume | DY Raw Batch XXXXXXX.XX Gal |
| 02 | Gross Batch Volume | DY Gross 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 Temperature | DY Batch Avg Temp SXXXX.X F |
| 07 | Batch Average Density | DY Batch Avg Dens XXXX.X Lb/F3 |
| 08 | Batch Average Pressure | DY Batch Avg Pres XXXX.X PSI |
| 09 | Batch Average Meter Factor | DY Batch Avg Mtr Factor X.XXXXX |
| 10 | Batch Average CTL | DY Batch Avg CTL X.XXXXX |
| 11 | Batch Average CPL | DY Batch Avg CPL X.XXXXX |
| 12 | Additive #1 Batch Volume | DY Add 1 Batch XXXXXXXX.XXX |
| 13 | Additive #2 Batch Volume | DY Add 2 Batch XXXXXXXX.XXX |
| 14 | Additive #3 Batch Volume | DY Add 3 Batch XXXXXXXX.XXX |
| 15 | Additive #4 Batch Volume | DY Add 4 Batch XXXXXXXX.XXX |
| 16 | Additive #5 Batch Volume | DY Add 5 Batch XXXXXXXX.XXX |

| Index Number ("XX") | Description | Response |
|---|---|---|
| 17 | Additive #6 Batch Volume | DY Add 6 Batch XXXXXXXX.XXX |
| 18 | Additive #7 Batch Volume | DY Add 7 Batch XXXXXXXX.XXX |
| 19 | Additive #8 Batch Volume | DY Add 8 Batch XXXXXXXX.XXX |
| 20 | Additive #9 Batch Volume | DY Add 9 Batch XXXXXXXX.XXX |
| 21 | Additive #10 Batch Volume | DY Add 10 Batch XXXXXXXX.XXX |
| 22 | Additive #11 Batch Volume | DY Add 11 Batch XXXXXXXX.XXX |
| 23 | Additive #12 Batch Volume | DY Add 12 Batch XXXXXXXX.XXX |
| 24 | Additive #13 Batch Volume | DY Add 13 Batch XXXXXXXX.XXX |
| 25 | Additive #14 Batch Volume | DY Add 14 Batch XXXXXXXX.XXX |
| 26 | Additive #15 Batch Volume | DY Add 15 Batch XXXXXXXX.XXX |
| 27 | Additive #16 Batch Volume | DY Add 16 Batch XXXXXXXX.XXX |
| 28 | Additive #17 Batch Volume | DY Add 17 Batch XXXXXXXX.XXX |
| 29 | Additive #18 Batch Volume | DY Add 18 Batch XXXXXXXX.XXX |
| 30 | Additive #19 Batch Volume | DY Add 19 Batch XXXXXXXX.XXX |
| 31 | Additive #20 Batch Volume | DY Add 20 Batch XXXXXXXX.XXX |
| 32 | Additive #21 Batch Volume | DY Add 21 Batch XXXXXXXX.XXX |
| 33 | Additive #22 Batch Volume | DY Add 22 Batch XXXXXXXX.XXX |
| 34 | Additive #23 Batch Volume | DY Add 23 Batch XXXXXXXX.XXX |
| 35 | Additive #24 Batch Volume | DY Add 24 Batch XXXXXXXX.XXX |

### *Transaction "DY_TRXX" Dynamic Display Values*

| Index Number ("XX") | Description | Response |
|---|---|---|
| 00 | Recipe Name and Number | DY Multiple Recipes |
| 01 | Indicated (Raw) Transaction Volume | DY Raw Trans XXXXXX.XX Gal |
| 02 | Gross Transaction Volume | DY Gross Trans XXXXXX.XX Gal |
| 03 | GST Transaction Volume | DY GST Trans XXXXXX.XX Gal |
| 04 | GSV Transaction Volume | DY GSV Trans XXXXXX.XX Gal |
| 05 | Mass Transaction Totals | DY Mass Trans XXXXXX.XX Gal |
| 06 | Transaction Average Temperature | DY Trans Avg Temp SXXXX.X F |
| 07 | Transaction Average Density | DY Trans Avg Dens XXXX.X Lb/F3 |
| 08 | Transaction Average Pressure | DY Trans Avg Pres XXXX.X PSI |
| 09 | Transaction Average Meter Factor | DY Trans Avg Mtr Factor X.XXXXX |
| 10 | Transaction Average CTL | DY Trans Avg CTL X.XXXX |

| Index Number ("XX") | Description | Response |
|---|---|---|
| 11 | Transaction Average CPL | DY Trans Avg CPL X.XXXX |
| 12 | Additive #1 Transaction Volume | DY Add 1 Trans XXXXXXXX.XXX |
| 13 | Additive #2 Transaction Volume | DY Add 2 Trans XXXXXXXX.XXX |
| 14 | Additive #3 Transaction Volume | DY Add 3 Trans XXXXXXXX.XXX |
| 15 | Additive #4 Transaction Volume | DY Add 4 Trans XXXXXXXX.XXX |
| 16 | Additive #5 Transaction Volume | DY Add 5 Trans XXXXXXXX.XXX |
| 17 | Additive #6 Transaction Volume | DY Add 6 Trans XXXXXXXX.XXX |
| 18 | Additive #7 Transaction Volume | DY Add 7 Trans XXXXXXXX.XXX |
| 19 | Additive #8 Transaction Volume | DY Add 8 Trans XXXXXXXX.XXX |
| 20 | Additive #9 Transaction Volume | DY Add 9 Trans XXXXXXXX.XXX |
| 21 | Additive #10 Transaction Volume | DY Add 10 Tran s XXXXXXXX.XXX |
| 22 | Additive #11 Transaction Volume | DY Add 11 Trans XXXXXXXX.XXX |
| 23 | Additive #12 Transaction Volume | DY Add 12 Trans XXXXXXXX.XXX |
| 24 | Additive #13 Transaction Volume | DY Add 13 Trans XXXXXXXX.XXX |
| 25 | Additive #14 Transaction Volume | DY Add 14 Trans XXXXXXXX.XXX |
| 26 | Additive #15 Transaction Volume | DY Add 15 Trans XXXXXXXX.XXX |
| 27 | Additive #16 Transaction Volume | DY Add 16 Trans XXXXXXXX.XXX |
| 28 | Additive #17 Transaction Volume | DY Add 17 Trans XXXXXXXX.XXX |
| 29 | Additive #18 Transaction Volume | DY Add 18 Trans XXXXXXXX.XXX |
| 30 | Additive #19 Transaction Volume | DY Add 19 Trans XXXXXXXX.XXX |
| 31 | Additive #20 Transaction Volume | DY Add 20 Trans XXXXXXXX.XXX |
| 32 | Additive #21 Transaction Volume | DY Add 21 Trans XXXXXXXX.XXX |
| 33 | Additive #22 Transaction Volume | DY Add 22 Trans XXXXXXXX.XXX |
| 34 | Additive #23 Transaction Volume | DY Add 23 Trans XXXXXXXX.XXX |
| 35 | Additive #24 Transaction Volume | DY Add 24 Trans XXXXXXXX.XXX |

### *Flow Controlled Additives "DY_FAXX" Dynamic Display Values*

| Index Number ("XX") | Description | Response |
|---|---|---|
| 00 | Additive 1 Current Batch IV (Raw) | DY A1 IV Batch XXXXX.XXX Gal |
| 01 | Additive 1 Current Batch GV (Gross) | DY A1 GV Batch XXXXX.XXX Gal |
| 02 | Additive 1 Current Batch GST Volume | DY A1 GST Batch XXXXX.XXX Gal |
| 03 | Additive 1 Current Batch Mass Total | DY A1 Mass Batch XXXXX.XXX Lbs |
| 04 | Additive 1 Current Batch Temperature | DY A1 Current Temp XXX.X F |
| 05 | Additive 1 Batch Average Temperature | DY A1 Avg Temp XXX.X F |
| 06 | Additive 1 Batch Average Density | DY A1 Avg Density XXX.X Lb/f3 |
| 07 | Additive 1 Batch Average CTL | DY A1 CTL X.XXXX |
| 08 | Additive 1 IV (Raw) Transaction Total | DY A1 IV Trans XXXXXX.XXX Gal |
| 09 | Additive 1 GV (Gross) Trans Total | DY A1 GV Trans XXXXXX.XXX Gal |
| 10 | Additive 1 GST Transaction Total | DY A1 GST Trans XXXXXX.XXX Gal |
| 11 | Additive 1 Mass Transaction Total | DY A1 Mass Trans XXXXXX.XXX Lbs |
| 12 | Additive 2 Current Batch IV (Raw) | DY A2 IV Batch XXXXX.XXX Gal |
| 13 | Additive 2 Current Batch GV (Gross) | DY A2 GV Batch XXXXX.XXX Gal |
| 14 | Additive 2 Current Batch GST Volume | DY A2 GST Batch XXXXX.XXX Gal |
| 15 | Additive 2 Current Batch Mass Total | DY A2 Mass Batch XXXXX.XXX Lbs |
| 16 | Additive 2 Current Batch Temperature | DY A2 Current Temp XXX.X F |
| 17 | Additive 2 Batch Average Temperature | DY A2 Avg Temp XXX.X F |
| 18 | Additive 2 Batch Average Density | DY A2 Avg Density XXX.X Lb/f3 |
| 19 | Additive 2 Batch Average CTL | DY A2 CTL X.XXXX |
| 20 | Additive 2 IV (Raw) Transaction Total | DY A2 IV Trans XXXXXX.XXX Gal |
| 21 | Additive 2 GV (Gross) Trans Total | DY A2 GV Trans XXXXXX.XXX Gal |
| 22 | Additive 2 GST Transaction Total | DY A2 GST Trans XXXXXX.XXX Gal |
| 23 | Additive 2 Mass Transaction Total | DY A2 Mass Trans XXXXXX.XXX Lbs |
| 24 | Additive 3 Current Batch IV (Raw) | DY A3 IV Batch XXXXX.XXX Gal |
| 25 | Additive 3 Current Batch GV (Gross) | DY A3 GV Batch XXXXX.XXX Gal |
| 26 | Additive 3 Current Batch GST Volume | DY A3 GST Batch XXXXX.XXX Gal |
| 27 | Additive 3 Current Batch Mass Total | DY A3 Mass Batch XXXXX.XXX Lbs |
| 28 | Additive 3 Current Batch Temperature | DY A3 Current Temp XXX.X F |
| 29 | Additive 3 Batch Average Temperature | DY A3 Avg Temp XXX.X F |
| 30 | Additive 3 Batch Average Density | DY A3 Avg Density XXX.X Lb/f3 |
| 31 | Additive 3 Batch Average CTL | DY A3 CTL X.XXXX |
| 32 | Additive 3 IV (Raw) Transaction Total | DY A3 IV Trans XXXXXX.XXX Gal |

| Index Number ("XX") | Description | Response |
|---|---|---|
| 33 | Additive 3 GV (Gross) Trans Total | DY A3 GV Trans XXXXXX.XXX Gal |
| 34 | Additive 3 GST Transaction Total | DY A3 GST Trans XXXXXX.XXX Gal |
| 35 | Additive 3 Mass Transaction Total | DY A3 Mass Trans XXXXXX.XXX Lbs |
| 36 | Additive 4 Current Batch IV (Raw) | DY A4 IV Batch XXXXX.XXX Gal |
| 37 | Additive 4 Current Batch GV (Gross) | DY A4 GV Batch XXXXX.XXX Gal |
| 38 | Additive 4 Current Batch GST Volume | DY A4 GST Batch XXXXX.XXX Gal |
| 39 | Additive 4 Current Batch Mass Total | DY A4 Mass Batch XXXXX.XXX Lbs |
| 40 | Additive 4 Current Batch Temperature | DY A4 Current Temp XXX.X F |
| 41 | Additive 4 Batch Average Temperature | DY A4 Avg Temp XXX.X F |
| 42 | Additive 4 Batch Average Density | DY A4 Avg Density XXX.X Lb/f3 |
| 43 | Additive 4 Batch Average CTL | DY A4 CTL X.XXXX |
| 44 | Additive 4 IV (Raw) Transaction Total | DY A4 IV Trans XXXXXX.XXX Gal |
| 45 | Additive 4 GV (Gross) Trans Total | DY A4 GV Trans XXXXXX.XXX Gal |
| 46 | Additive 4 GST Transaction Total | DY A4 GST Trans XXXXXX.XXX Gal |
| 47 | Additive 4 Mass Transaction Total | DY A4 Mass Trans XXXXXX.XXX Lbs |

## *Command Code EA – Enquire Alarms*

This command requests the alarm status from the AccuLoad III (short form).

**Command:**

**"EA_DD"** (Request alarm status of directory)

Where: DD = directory specification
SY = System
P1 = Product 1
P2 = Product 2
P3 = Product 3
P4 = Product 4
P5 = Product 5
P6 = Product 6
AR = Arm
IN = Injector
M1 = Meter 1
M2 = Meter 2 (ratio, hybrid blending configuration only)
M3 = Meter 3 (ratio, hybrid blending configuration only)
M4 = Meter 4 (ratio, hybrid blending configuration only)
M5 = Meter 5 (ratio, hybrid blending configuration only)
M6 = Meter 6 (ratio, hybrid blending configuration only)
RR = Recipe

**Responses:**

**"EA_SY_A1A2A3A4A5A6A8A9A10A11" Good Response.** Ten characters for System

**"EA_Px_A1A2A3A4" Good Response.** Four characters for Products

**"EA_AR_A1A2A3" Good Response.** Three characters for Arm

**"EA_IN_A1A2A3 … A83" Good Response.** Seventy-seven characters for Additive Injectors

**"EA_Mx_A1A2A3" Good Response.** Three characters for Meter

**"EA_RR_A1" Good Response.** One character for Recipe

**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:** NO30 will be returned if the meter, product or additive specified is not currently configured to the arm which the request was sent.
NO31 will be returned if the command format does not match the current configuration for the arm.

**Special Case:** None.

**Comm. Modes:** Polling only, Poll & Authorize, Remote Control, Poll & 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** | **EAAI Failure (DA)** | **Sys Prog Error (DA)** | **Watchdog (DA)** | **Flash 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** | **Communication (CM)** | **Power Fail (PA)** | **Passcode Reset (DA)** | **BSE Failure (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 A3 (System)**

### *Response to System Enquire Alarms Command, Character A4*

| Character | | System Alarm Requests | | | |
|---|---|---|---|---|---|
| **Char** | **Hex** | **User Alarm 1 (U1)** | **PTB Printer (PP)** | **Shared Printer (SP)** | **CIVACON Comm Failure (CV)** |
| 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 Request | | | |
|---|---|---|---|---|---|
| **Char** | **Hex** | **User Alarm 5 (U5)** | **User Alarm 4 (U4)** | **User Alarm 3 (U3)** | **User Alarm 2 (U2)** |
| 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 Request | | | |
|---|---|---|---|---|---|
| **Char** | **Hex** | **User Alarm 9 (U9)** | **User Alarm 8 (U8)** | **User Alarm 7 (U7)** | **User Alarm 6 (U6)** |
| 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 Request | | | |
|---|---|---|---|---|---|
| **Char** | **Hex** | **Add-Pak #1 Diagnostic (D1)** | **Add-Pak #2 Power-fail (A2)** | **Add-Pak #1 Power-fail (A1)** | **User Alarm #10 (U0)** |
| 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 Request | | | |
|---|---|---|---|---|---|
| **Char** | **Hex** | **Add-Pak #1 Comm Failed (C1)** | **Add-Pak #2 Auto-Detect Failed (P2)** | **Add-Pak #1 Auto-Detect Failed (P1)** | **Add-Pak #2 Diagnostic (D2)** |
| 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 Request | | | |
|---|---|---|---|---|---|
| **Char** | **Hex** | **MMI Excess Arms Active (ME)** | **MMI Comm Fail (MC)** | **Display Failure (DA)** | **Add-Pak #2 Comm Fail (C2)** |
| 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 Request | | | |
|---|---|---|---|---|---|
| Char | Hex | Sening Comm | Network Printer | ComFlash (CF) | Data Retention (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 A10 (System)**

### *Response to System Enquire Alarms Command, Character A11*

| Character | | System Alarm Request | | | |
|---|---|---|---|---|---|
| **Char** | **Hex** | **Reserved** | **Reserved** | **Reserved** | **Factory Reserved** |
| 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 A11 (System)**

### *Response to Product Enquire Alarms Command, Character A1*

| Character | | Product Alarm Request | | | |
|---|---|---|---|---|---|
| **Char** | **Hex** | **High Flow (HF)** | **High Density (HD)** | **Back Pressure (BP)** | **Product Prog Error (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 (Product)**

### *Response to Product Enquire Alarms Command, Character A2*

| Character | | Product Alarm Request | | | |
|---|---|---|---|---|---|
| **Char** | **Hex** | **Low Flow (LF)** | **Low Density (LD)** | **High Temp Alarm (HT)** | **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 A2 (Product)**

### *Response to Product Enquire Alarms Command, Character A3*

| Character | | Product Alarm Request | | | |
|---|---|---|---|---|---|
| **Char** | **Hex** | **Overrun (OA)** | **Zero Flow (ZF)** | **Low Temper-ature (LT)** | **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 A3 (Product)**

### *Response to Product Enquire Alarms Command, Character A4*

| Character | | Product Alarm Request | | | |
|---|---|---|---|---|---|
| **Char** | **Hex** | **Product Stop (PA)** | **Blend Low (BL)** | **Blend High (BH)** | **Block Valve (BV)** |
| 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 (Product)**

### *Response to Arm Enquire Alarms Command, Character A1*

| Character | | Arm Alarm Requests | | | |
|---|---|---|---|---|---|
| **Char** | **Hex** | **Ticket (TK)** | **Overrun (OA)** | **Zero Flow (ZF)** | **Arm Program Error (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 (Arm)**

### *Response to Arm Enquire Alarms Command, Character A2*

| Character | | Arm Alarm Requests | | | |
|---|---|---|---|---|---|
| **Char** | **Hex** | **Storage Full (SF)** | **Arm Recipe Program Error (DA)** | **Additive Clean Line (CA)** | **Clean Line (CL)** |
| 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 (Arm)**

### *Response to Arm Enquire Alarms Command, Character A3*

| Character | | Arm Alarm Requests | | | |
|---|---|---|---|---|---|
| **Char** | **Hex** | **Reserved** | **Reserved** | **Card Removed (CD)** | **DE Head Alarm (DE)** |
| 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 (Arm)**

### *Response to Injector Enquire Alarms Command, Character A1*

| Character | | Additive Alarm Requests | | | |
|---|---|---|---|---|---|
| **Char** | **Hex** | **Injector 4 Feedback (FA)** | **Injector 3 Feedback (FA)** | **Injector 2 Feedback (FA)** | **Injector 1 Feedback (FA)** |
| 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 8 Feedback (FA)** | **Injector 7 Feedback (FA)** | **Injector 6 Feedback (FA)** | **Injector 5 Feedback (FA)** |
| 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 12 Feedback (FA)** | **Injector 11 Feedback (FA)** | **Injector 10 Feedback (FA)** | **Injector 9 Feedback (FA)** |
| 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 16 Feedback (FA)** | **Injector 15 Feedback (FA)** | **Injector 14 Feedback (FA)** | **Injector 13 Feedback (FA)** |
| 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 20 Feedback (FA)** | **Injector 19 Feedback (FA)** | **Injector 18 Feedback (FA)** | **Injector 17 Feedback (FA)** |
| 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 24 Feedback (FA)** | **Injector 23 Feedback (FA)** | **Injector 22 Feedback (FA)** | **Injector 21 Feedback (FA)** |
| 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 Comm Error (AC)** | **Injector 3 Comm Error (AC)** | **Injector 2 Comm Error (AC)** | **Injector 1 Comm Error (AC)** |
| 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 8 Comm Error (AC)** | **Injector 7 Comm Error (AC)** | **Injector 6 Comm Error (AC)** | **Injector 5 Comm Error (AC)** |
| 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 12 Comm Error (AC)** | **Injector 11 Comm Error (AC)** | **Injector 10 Comm Error (AC)** | **Injector 9 Comm Error (AC)** |
| 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 16 Comm Error (AC)** | **Injector 15 Comm Error (AC)** | **Injector 14 Comm Error (AC)** | **Injector 13 Comm Error (AC)** |
| 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)**

### *Response to Injector Enquire Alarms Command, Character A11*

| Character | | Additive Alarm Requests | | | |
|---|---|---|---|---|---|
| **Char** | **Hex** | **Injector 20 Comm Error (AC)** | **Injector 19 Comm Error (AC)** | **Injector 18 Comm Error (AC)** | **Injector 17 Comm Error (AC)** |
| 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 A11 (Injector)**

### *Response to Injector Enquire Alarms Command, Character A12*

| Character | | Additive Alarm Requests | | | |
|---|---|---|---|---|---|
| **Char** | **Hex** | **Injector 24 Comm Error (AC)** | **Injector 23 Comm Error (AC)** | **Injector 22 Comm Error (AC)** | **Injector 21 Comm Error (AC)** |
| 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 A12 (Injector)**

### *Response to Injector Enquire Alarms Command, Character A13*

| Character | | Additive Alarm Requests | | | |
|---|---|---|---|---|---|
| **Char** | **Hex** | **Injector 4 Low Add Error (KA)** | **Injector 3 Low Add Error (KA)** | **Injector 2 Low Add Error (KA)** | **Injector 1 Low Add Error (KA)** |
| 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 A13 (Injector)**

### *Response to Injector Enquire Alarms Command, Character A14*

| Character | | Additive Alarm Requests | | | |
|---|---|---|---|---|---|
| **Char** | **Hex** | **Injector 8 Low Add Error (KA)** | **Injector 7 Low Add Error (KA)** | **Injector 6 Low Add Error (KA)** | **Injector 5 Low Add Error (KA)** |
| 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 A14 (Injector)**

### *Response to Injector Enquire Alarms Command, Character A15*

| Character | | Additive Alarm Requests | | | |
|---|---|---|---|---|---|
| **Char** | **Hex** | **Injector 12 Low Add Error (KA)** | **Injector 11 Low Add Error (KA)** | **Injector 10 Low Add Error (KA)** | **Injector 9 Low Add Error (KA)** |
| 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 A15 (Injector)**

### *Response to Injector Enquire Alarms Command, Character A16*

| Character | | Additive Alarm Requests | | | |
|---|---|---|---|---|---|
| **Char** | **Hex** | **Injector 16 Low Add Error (KA)** | **Injector 15 Low Add Error (KA)** | **Injector 14 Low Add Error (KA)** | **Injector 13 Low Add Error (KA)** |
| 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 A16 (Injector)**

### *Response to Injector Enquire Alarms Command, Character A17*

| Character | | Additive Alarm Requests | | | |
|---|---|---|---|---|---|
| **Char** | **Hex** | **Injector 20 Low Add Error (KA)** | **Injector 19 Low Add Error (KA)** | **Injector 18 Low Add Error (KA)** | **Injector 17 Low Add Error (KA)** |
| 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 A17 (Injector)**

### *Response to Injector Enquire Alarms Command, Character A18*

| Character | | Additive Alarm Requests | | | |
|---|---|---|---|---|---|
| **Char** | **Hex** | **Injector 24 Low Add Error (KA)** | **Injector 23 Low Add Error (KA)** | **Injector 22 Low Add Error (KA)** | **Injector 21 Low Add Error (KA)** |
| 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 A18 (Injector)**

### *Response to Injector Enquire Alarms Command, Character A19*

| Character | | Additive Alarm Requests | | | |
|---|---|---|---|---|---|
| **Char** | **Hex** | **Injector 4 Excess Pulses (MA)** | **Injector 3 Excess Pulses (MA)** | **Injector 2 Excess Pulses (MA)** | **Injector 1 Excess Pulses (MA)** |
| 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 A19 (Injector)**

### *Response to Injector Enquire Alarms Command, Character A20*

| Character | | Additive Alarm Requests | | | |
|---|---|---|---|---|---|
| **Char** | **Hex** | **Injector 8 Excess Pulses (MA)** | **Injector 7 Excess Pulses (MA)** | **Injector 6 Excess Pulses (MA)** | **Injector 5 Excess Pulses (MA)** |
| 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 A20 (Injector)**

### *Response to Injector Enquire Alarms Command, Character A21*

| Character | | Additive Alarm Requests | | | |
|---|---|---|---|---|---|
| **Char** | **Hex** | **Injector 12 Excess Pulses (MA)** | **Injector 11 Excess Pulses (MA)** | **Injector 10 Excess Pulses (MA)** | **Injector 9 Excess Pulses (MA)** |
| 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 A21 (Injector)**

### *Response to Injector Enquire Alarms Command, Character A22*

| Character | | Additive Alarm Requests | | | |
|---|---|---|---|---|---|
| **Char** | **Hex** | **Injector 16 Excess Pulses (MA)** | **Injector 15 Excess Pulses (MA)** | **Injector 14 Excess Pulses (MA)** | **Injector 13 Excess Pulses (MA)** |
| 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 A22 (Injector)**

### *Response to Injector Enquire Alarms Command, Character A23*

| Character | | Additive Alarm Requests | | | |
|---|---|---|---|---|---|
| **Char** | **Hex** | **Injector 20 Excess Pulses (MA)** | **Injector 19 Excess Pulses (MA)** | **Injector 18 Excess Pulses (MA)** | **Injector 17 Excess Pulses (MA)** |
| 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 A23 (Injector)**

### *Response to Injector Enquire Alarms Command, Character A24*

| Character | | Additive Alarm Requests | | | |
|---|---|---|---|---|---|
| **Char** | **Hex** | **Injector 24 Excess Pulses (MA)** | **Injector 23 Excess Pulses (MA)** | **Injector 22 Excess Pulses (MA)** | **Injector 21 Excess Pulses (MA)** |
| 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 A24 (Injector)**

### *Response to Injector Enquire Alarms Command, Character A25*

| Character | | Additive Alarm Requests | | | |
|---|---|---|---|---|---|
| **Char** | **Hex** | **Injector 4 No Pulses (NA)** | **Injector 3 No Pulses (NA)** | **Injector 2 No Pulses (NA)** | **Injector 1 No Pulses (NA)** |
| 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 A25 (Injector)**

### *Response to Injector Enquire Alarms Command, Character A26*

| Character | | Additive Alarm Requests | | | |
|---|---|---|---|---|---|
| **Char** | **Hex** | **Injector 8 No Pulses (NA)** | **Injector 7 No Pulses (NA)** | **Injector 6 No Pulses (NA)** | **Injector 5 No Pulses (NA)** |
| 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 A26 (Injector)**

### *Response to Injector Enquire Alarms Command, Character A27*

| Character | | Additive Alarm Requests | | | |
|---|---|---|---|---|---|
| **Char** | **Hex** | **Injector 12 No Pulses (NA)** | **Injector 11 No Pulses (NA)** | **Injector 10 No Pulses (NA)** | **Injector 9 No Pulses (NA)** |
| 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 A27 (Injector)**

### *Response to Injector Enquire Alarms Command, Character A28*

| Character | | Additive Alarm Requests | | | |
|---|---|---|---|---|---|
| **Char** | **Hex** | **Injector 16 No Pulses (NA)** | **Injector 15 No Pulses (NA)** | **Injector 14 No Pulses (NA)** | **Injector 13 No Pulses (NA)** |
| 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 A28 (Injector)**

### *Response to Injector Enquire Alarms Command, Character A29*

| Character | | Additive Alarm Requests | | | |
|---|---|---|---|---|---|
| **Char** | **Hex** | **Injector 20 No Pulses (NA)** | **Injector 19 No Pulses (NA)** | **Injector 18 No Pulses (NA)** | **Injector 17 No Pulses (NA)** |
| 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 A29 (Injector)**

### *Response to Injector Enquire Alarms Command, Character A30*

| Character | | Additive Alarm Requests | | | |
|---|---|---|---|---|---|
| **Char** | **Hex** | **Injector 24 No Pulses (NA)** | **Injector 23 No Pulses (NA)** | **Injector 22 No Pulses (NA)** | **Injector 21 No Pulses (NA)** |
| 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 A30 (Injector)**

### *Response to Injector Enquire Alarms Command, Character A31*

| Character | | Additive Alarm Requests | | | |
|---|---|---|---|---|---|
| **Char** | **Hex** | **Injector 4 Frequency (RA)** | **Injector 3 Frequency (RA)** | **Injector 2 Frequency (RA)** | **Injector 1 Frequency (RA)** |
| 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 A31 (Injector)**

### *Response to Injector Enquire Alarms Command, Character A32*

| Character | | Additive Alarm Requests | | | |
|---|---|---|---|---|---|
| **Char** | **Hex** | **Injector 8 Frequency (RA)** | **Injector 7 Frequency (RA)** | **Injector 6 Frequency (RA)** | **Injector 5 Frequency (RA)** |
| 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 A32 (Injector)**

### *Response to Injector Enquire Alarms Command, Character A33*

| Character | | Additive Alarm Requests | | | |
|---|---|---|---|---|---|
| **Char** | **Hex** | **Injector 12 Frequency (RA)** | **Injector 11 Frequency (RA)** | **Injector 10 Frequency (RA)** | **Injector 9 Frequency (RA)** |
| 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 A33 (Injector)**

### *Response to Injector Enquire Alarms Command, Character A34*

| Character | | Additive Alarm Requests | | | |
|---|---|---|---|---|---|
| **Char** | **Hex** | **Injector 16 Frequency (RA)** | **Injector 15 Frequency (RA)** | **Injector 14 Frequency (RA)** | **Injector 13 Frequency (RA)** |
| 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 A34 (Injector)**

### *Response to Injector Enquire Alarms Command, Character A35*

| Character | | Additive Alarm Requests | | | |
|---|---|---|---|---|---|
| **Char** | **Hex** | **Injector 20 Frequency (RA)** | **Injector 19 Frequency (RA)** | **Injector 18 Frequency (RA)** | **Injector 17 Frequency (RA)** |
| 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 A35 (Injector)**

### *Response to Injector Enquire Alarms Command, Character A36*

| Character | | Additive Alarm Requests | | | |
|---|---|---|---|---|---|
| **Char** | **Hex** | **Injector 24 Frequency (RA)** | **Injector 23 Frequency (RA)** | **Injector 22 Frequency (RA)** | **Injector 21 Frequency (RA)** |
| 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 A36 (Injector)**

### *Response to Injector Enquire Alarms Command, Character A37*

| Character | | Additive Alarm Requests | | | |
|---|---|---|---|---|---|
| **Char** | **Hex** | **Injector 4 Unauth. Failed (UA)** | **Injector 3 Unauth. Failed (UA)** | **Injector 2 Unauth. Failed (UA)** | **Injector 1 Unauth. Failed (UA)** |
| 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 A37 (Injector)**

### *Response to Injector Enquire Alarms Command, Character A38*

| Character | | Additive Alarm Requests | | | |
|---|---|---|---|---|---|
| **Char** | **Hex** | **Injector 8 Unauth. Failed (UA)** | **Injector 7 Unauth. Failed (UA)** | **Injector 6 Unauth. Failed (UA)** | **Injector 5 Unauth. Failed (UA)** |
| 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 A38 (Injector)**

### *Response to Injector Enquire Alarms Command, Character A39*

| Character | | Additive Alarm Requests | | | |
|---|---|---|---|---|---|
| **Char** | **Hex** | **Injector 12 Unauth. Failed (UA)** | **Injector 11 Unauth. Failed (UA)** | **Injector 10 Unauth. Failed (UA)** | **Injector 9 Unauth. Failed (UA)** |
| 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 A39 (Injector)**

### *Response to Injector Enquire Alarms Command, Character A40*

| Character | | Additive Alarm Requests | | | |
|---|---|---|---|---|---|
| **Char** | **Hex** | **Injector 16 Unauth. Failed (UA)** | **Injector 15 Unauth. Failed (UA)** | **Injector 14 Unauth. Failed (UA)** | **Injector 13 Unauth. Failed (UA)** |
| 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 A40 (Injector)**

### *Response to Injector Enquire Alarms Command, Character A41*

| Character | | Additive Alarm Requests | | | |
|---|---|---|---|---|---|
| **Char** | **Hex** | **Injector 20 Unauth. Failed (UA)** | **Injector 19 Unauth. Failed (UA)** | **Injector 18 Unauth. Failed (UA)** | **Injector 17 Unauth. Failed (UA)** |
| 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 A41 (Injector)**

### *Response to Injector Enquire Alarms Command, Character A42*

| Character | | Additive Alarm Requests | | | |
|---|---|---|---|---|---|
| **Char** | **Hex** | **Injector 24 Unauth. Failed (UA)** | **Injector 23 Unauth. Failed (UA)** | **Injector 22 Unauth. Failed (UA)** | **Injector 21 Unauth. Failed (UA)** |
| 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 A42 (Injector)**

### *Response to Injector Enquire Alarms Command, Character A43*

| Character | | Additive Alarm Requests | | | |
|---|---|---|---|---|---|
| **Char** | **Hex** | **Injector 4 General Additive Error (GA)** | **Injector 3 General Additive Error (GA)** | **Injector 2 General Additive Error (GA)** | **Injector 1 General Additive Error (GA)** |
| 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 A43 (Injector)**

### *Response to Injector Enquire Alarms Command, Character A44*

| Character | | Additive Alarm Requests | | | |
|---|---|---|---|---|---|
| **Char** | **Hex** | **Injector 8 General Additive Error (GA)** | **Injector 7 General Additive Error (GA)** | **Injector 6 General Additive Error (GA)** | **Injector 5 General Additive Error (GA)** |
| 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 A44 (Injector)**

### *Response to Injector Enquire Alarms Command, Character A45*

| Character | | Additive Alarm Requests | | | |
|---|---|---|---|---|---|
| **Char** | **Hex** | **Injector 12 General Additive Error (GA)** | **Injector 11 General Additive Error (GA)** | **Injector 10 General Additive Error (GA)** | **Injector 9 General Additive Error (GA)** |
| 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 A45 (Injector)**

### *Response to Injector Enquire Alarms Command, Character A46*

| Character | | Additive Alarm Requests | | | |
|---|---|---|---|---|---|
| **Char** | **Hex** | **Injector 16 General Additive Error (GA)** | **Injector 15 General Additive Error (GA)** | **Injector 14 General Additive Error (GA)** | **Injector 13 General Additive Error (GA)** |
| 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 A46 (Injector)**

### *Response to Injector Enquire Alarms Command, Character A47*

| Character | | Additive Alarm Requests | | | |
|---|---|---|---|---|---|
| **Char** | **Hex** | **Injector 20 General Additive Error (GA)** | **Injector 19 General Additive Error (GA)** | **Injector 18 General Additive Error (GA)** | **Injector 17 General Additive Error (GA)** |
| 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 A47 (Injector)**

### *Response to Injector Enquire Alarms Command, Character A48*

| Character | | Additive Alarm Requests | | | |
|---|---|---|---|---|---|
| **Char** | **Hex** | **Injector 24 General Additive Error (GA)** | **Injector 23 General Additive Error (GA)** | **Injector 22 General Additive Error (GA)** | **Injector 21 General Additive Error (GA)** |
| 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 A48 (Injector)**

### *Response to Injector Enquire Alarms Command, Character A49*

| Character | | Additive Alarm Requests | | | |
|---|---|---|---|---|---|
| **Char** | **Hex** | **Injector 4 Over-Speed (OR)** | **Injector 3 Over-Speed (OR)** | **Injector 2 Over-Speed (OR)** | **Injector 1 Over-Speed (OR)** |
| 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 A49 (Injector)**

### *Response to Injector Enquire Alarms Command, Character A50*

| Character | | Additive Alarm Requests | | | |
|---|---|---|---|---|---|
| **Char** | **Hex** | **Injector 8 Over-Speed (OR)** | **Injector 7 Over-Speed (OR)** | **Injector 6 Over-Speed (OR)** | **Injector 5 Over-Speed (OR)** |
| 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 A50 (Injector)**

### *Response to Injector Enquire Alarms Command, Character A51*

| Character | | Additive Alarm Requests | | | |
|---|---|---|---|---|---|
| **Char** | **Hex** | **Injector 12 Over-Speed (OR)** | **Injector 11 Over-Speed (OR)** | **Injector 10 Over-Speed (OR)** | **Injector 9 Over-Speed (OR)** |
| 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 A51 (Injector)**

### *Response to Injector Enquire Alarms Command, Character A52*

| Character | | Additive Alarm Requests | | | |
|---|---|---|---|---|---|
| **Char** | **Hex** | **Injector 16 Over-Speed (OR)** | **Injector 15 Over-Speed (OR)** | **Injector 14 Over-Speed (OR)** | **Injector 13 Over-Speed (OR)** |
| 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 A52 (Injector)**

### *Response to Injector Enquire Alarms Command, Character A53*

| Character | | Additive Alarm Requests | | | |
|---|---|---|---|---|---|
| **Char** | **Hex** | **Injector 20 Over-Speed (OR)** | **Injector 19 Over-Speed (OR)** | **Injector 18 Over-Speed (OR)** | **Injector 17 Over-Speed (OR)** |
| 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 A53 (Injector)**

### *Response to Injector Enquire Alarms Command, Character A54*

| Character | | Additive Alarm Requests | | | |
|---|---|---|---|---|---|
| **Char** | **Hex** | **Injector 24 Over-Speed (OR)** | **Injector 23 Over-Speed (OR)** | **Injector 22 Over-Speed (OR)** | **Injector 21 Over-Speed (OR)** |
| 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 A54 (Injector)**

### *Response to Injector Enquire Alarms Command, Character A55*

| Character | | Additive Alarm Requests | | | |
|---|---|---|---|---|---|
| **Char** | **Hex** | **Injector 4 Command Refused (CR)** | **Injector 3 Command Refused (CR)** | **Injector 2 Command Refused (CR)** | **Injector 1 Command Refused (CR)** |
| 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 A55 (Injector)**

### *Response to Injector Enquire Alarms Command, Character A56*

| Character | | Additive Alarm Requests | | | |
|---|---|---|---|---|---|
| **Char** | **Hex** | **Injector 8 Command Refused (CR)** | **Injector 7 Command Refused (CR)** | **Injector 6 Command Refused (CR)** | **Injector 5 Command Refused (CR)** |
| 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 A56 (Injector)**

### *Response to Injector Enquire Alarms Command, Character A57*

| Character | | Additive Alarm Requests | | | |
|---|---|---|---|---|---|
| **Char** | **Hex** | **Injector 12 Command Refused (CR)** | **Injector 11 Command Refused (CR)** | **Injector 10 Command Refused (CR)** | **Injector 9 Command Refused (CR)** |
| 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 A57 (Injector)**

### *Response to Injector Enquire Alarms Command, Character A58*

| Character | | Additive Alarm Requests | | | |
|---|---|---|---|---|---|
| **Char** | **Hex** | **Injector 16 Command Refused (CR)** | **Injector 15 Command Refused (CR)** | **Injector 14 Command Refused (CR)** | **Injector 13 Command Refused (CR)** |
| 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 A58 (Injector)**

### *Response to Injector Enquire Alarms Command, Character A59*

| Character | | Additive Alarm Requests | | | |
|---|---|---|---|---|---|
| **Char** | **Hex** | **Injector 20 Command Refused (CR)** | **Injector 19 Command Refused (CR)** | **Injector 18 Command Refused (CR)** | **Injector 17 Command Refused (CR)** |
| 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 A59 (Injector)**

### *Response to Injector Enquire Alarms Command, Character A60*

| Character | | Additive Alarm Requests | | | |
|---|---|---|---|---|---|
| **Char** | **Hex** | **Injector 24 Command Refused (CR)** | **Injector 23 Command Refused (CR)** | **Injector 22 Command Refused (CR)** | **Injector 21 Command Refused (CR)** |
| 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 A60 (Injector)**

### *Response to Injector Enquire Alarms Command, Character A61*

| Character | | Additive Alarm Requests | | | |
|---|---|---|---|---|---|
| **Char** | **Hex** | **Injector 4 Autodetect Fail (AD)** | **Injector 3 Autodetect Fail (AD)** | **Injector 2 Autodetect Fail (AD)** | **Injector 1 Autodetect Fail (AD)** |
| 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 A61 (Injector)**

### *Response to Injector Enquire Alarms Command, Character A62*

| Character | | Additive Alarm Requests | | | |
|---|---|---|---|---|---|
| **Char** | **Hex** | **Injector 8 Autodetect Fail (AD)** | **Injector 7 Autodetect Fail (AD)** | **Injector 6 Autodetect Fail (AD)** | **Injector 5 Autodetect Fail (AD)** |
| 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 A62 (Injector)**

### *Response to Injector Enquire Alarms Command, Character A63*

| Character | | Additive Alarm Requests | | | |
|---|---|---|---|---|---|
| **Char** | **Hex** | **Injector 12 Autodetect Fail (AD)** | **Injector 11 Autodetect Fail (AD)** | **Injector 10 Autodetect Fail (AD)** | **Injector 9 Autodetect Fail (AD)** |
| 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 A63 (Injector)**

### *Response to Injector Enquire Alarms Command, Character A64*

| Character | | Additive Alarm Requests | | | |
|---|---|---|---|---|---|
| **Char** | **Hex** | **Injector 16 Autodetect Fail (AD)** | **Injector 15 Autodetect Fail (AD)** | **Injector 14 Autodetect Fail (AD)** | **Injector 13 Autodetect Fail (AD)** |
| 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 A64 (Injector)**

### *Response to Injector Enquire Alarms Command, Character A65*

| Character | | Additive Alarm Requests | | | |
|---|---|---|---|---|---|
| **Char** | **Hex** | **Injector 20 Autodetect Fail (AD)** | **Injector 19 Autodetect Fail (AD)** | **Injector 18 Autodetect Fail (AD)** | **Injector 17 Autodetect Fail (AD)** |
| 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 A65 (Injector)**

### *Response to Injector Enquire Alarms Command, Character A66*

| Character | | Additive Alarm Requests | | | |
|---|---|---|---|---|---|
| **Char** | **Hex** | **Injector 24 Autodetect Fail (AD)** | **Injector 23 Autodetect Fail (AD)** | **Injector 22 Autodetect Fail (AD)** | **Injector 21 Autodetect Fail (AD)** |
| 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 A66 (Injector)**

### *Response to Injector Enquire Alarms Command, Character A67*

| Character | | Additive Alarm Requests | | | |
|---|---|---|---|---|---|
| **Char** | **Hex** | **Injector 4 High Temp (AH)** | **Injector 3 High Temp (AH)** | **Injector 2 High Temp (AH)** | **Injector 1 High Temp (AH)** |
| 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 A67 (Injector)**

### *Response to Injector Enquire Alarms Command, Character A68*

| Character | | Additive Alarm Requests | | | |
|---|---|---|---|---|---|
| **Char** | **Hex** | **Injector 4 Low Temp (AL)** | **Injector 3 Low Temp (AL)** | **Injector 2 Low Temp (AL)** | **Injector 1 Low Temp (AL)** |
| 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 A68 (Injector)**

### *Response to Injector Enquire Alarms Command, Character A69*

| Character | | Additive Alarm Requests | | | |
|---|---|---|---|---|---|
| **Char** | **Hex** | **Injector 4 Temp Probe (AT)** | **Injector 3 Temp Probe (AT)** | **Injector 2 Temp Probe (AT)** | **Injector 1 Temp Probe (AT)** |
| 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 A69 (Injector)**

### *Response to Injector Enquire Alarms Command, Character A70*

| Character | | Additive Alarm Requests | | | |
|---|---|---|---|---|---|
| **Char** | **Hex** | **Injector 4 Pulse Security (PS)** | **Injector 3 Pulse Security (PS)** | **Injector 2 Pulse Security (PS)** | **Injector 1 Pulse Security (PS)** |
| 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 A70 (Injector)**

### *Response to Injector Enquire Alarms Command, Character A71*

| Character | | Additive Alarm Requests | | | |
|---|---|---|---|---|---|
| **Char** | **Hex** | **Injector 4 Transmitter Integrity (XA)** | **Injector 3 Transmitter Integrity (XA)** | **Injector 2 Transmitter Integrity (XA)** | **Injector 1 Transmitter Integrity (XA)** |
| 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 A71 (Injector)**

## *Response to Injector Enquire Alarms Command, Character A72*

| Character | | Additive Alarm Requests | | | |
|---|---|---|---|---|---|
| **Char** | **Hex** | **Injector 4 Add Comm Totals (CT)** | **Injector 3 Add Comm Totals (CT)** | **Injector 2 Add Comm Totals (CT)** | **Injector 1 Add Comm Totals (CT)** |
| 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 A72 (Injector)**

### *Response to Injector Enquire Alarms Command, Character A73*

| Character | | Additive Alarm Requests | | | |
|---|---|---|---|---|---|
| **Char** | **Hex** | **Injector 8 Add Comm Totals (CT)** | **Injector 7 Add Comm Totals (CT)** | **Injector 6 Add Comm Totals (CT)** | **Injector 5 Add Comm Totals (CT)** |
| 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 A73 (Injector)**

## *Response to Injector Enquire Alarms Command, Character A74*

| Character | | Additive Alarm Requests | | | |
|---|---|---|---|---|---|
| **Char** | **Hex** | **Injector 12 Add Comm Totals (CT)** | **Injector 11 Add Comm Totals (CT)** | **Injector 10 Add Comm Totals (CT)** | **Injector 9 Add Comm Totals (CT)** |
| 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 A74 (Injector)**

### *Response to Injector Enquire Alarms Command, Character A75*

| Character | | Additive Alarm Requests | | | |
|---|---|---|---|---|---|
| **Char** | **Hex** | **Injector 16 Add Comm Totals (CT)** | **Injector 15 Add Comm Totals (CT)** | **Injector 14 Add Comm Totals (CT)** | **Injector 13 Add Comm Totals (CT)** |
| 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 A75 (Injector)**

### *Response to Injector Enquire Alarms Command, Character A76*

| Character | | Additive Alarm Requests | | | |
|---|---|---|---|---|---|
| **Char** | **Hex** | **Injector 20 Add Comm Totals (CT)** | **Injector 19 Add Comm Totals (CT)** | **Injector 18 Add Comm Totals (CT)** | **Injector 17 Add Comm Totals (CT)** |
| 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 A76 (Injector)**

## *Response to Injector Enquire Alarms Command, Character A77*

| Character | | Additive Alarm Requests | | | |
|---|---|---|---|---|---|
| **Char** | **Hex** | **Injector 24 Add Comm Totals (CT)** | **Injector 23 Add Comm Totals (CT)** | **Injector 22 Add Comm Totals (CT)** | **Injector 21 Add Comm Totals (CT)** |
| 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 A77 (Injector)**

***Note:***
***Responses to Command Code EA - Characters A78 – A83 (Injector) Are All Reserved for Future Use.***

### *Response to Meter Enquire Alarms Command, Character A1*

| Character | | Meter Alarm Requests | | | |
|---|---|---|---|---|---|
| **Char** | **Hex** | **Valve Fault (VF)** | **Pulse Security (PS)** | **Transmitter Integrity (XA)** | **Meter Program Error (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 (Meter)**

### *Response to Meter Enquire Alarms Command, Character A2*

| Character | | Meter Alarm Requests | | | |
|---|---|---|---|---|---|
| **Char** | **Hex** | **Turbine Meter (TM)** | **Density Trans. (DR)** | **Pressure Trans. (PR)** | **Temperature Probe (TP)** |
| 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 (Meter)**

### *Response to Meter Enquire Alarms Command, Character A3*

| Character | | Meter Alarm Requests | | | |
|---|---|---|---|---|---|
| **Char** | **Hex** | **Solenoid Counts (SC)** | **Mass Meter Tube (MT)** | **Mass Meter Overdrive (MO)** | **Mass Meter Comm Fail (MF)** |
| 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 (Meter)**

### *Response to Meter Enquire Alarms Command, Character A4*

| Character | | Meter Alarm Requests | | | |
|---|---|---|---|---|---|
| **Char** | **Hex** | **Reverse Flow (FR)** | **Leakage (LA)** | **Predict Overrun (PO)** | **Promass (PM)** |
| 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 (Meter)**

### *Response to Recipe Enquire Alarms Command, Character A1*

| Character | | Recipe Alarm Requests | | | |
|---|---|---|---|---|---|
| **Char** | **Hex** | **Reserved** | **Reserved** | **Reserved** | **Program Error (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 (Recipe)**

## *Command Code EB – End Batch*

This command cancels the remaining batch volume.

**Command:**

**"EB"**

**Responses:**

**"OK"** **Good Response.** Batch flagged as complete if batch is in progress. Batch authorization is removed.

or...

**"NOXX"** The batch volume has not been canceled.

**Remarks:** None.

**Constraints:** NO01 will be returned if the other side is in the program mode locally or via communications.

**Special Case:** Valve will be commanded to close if it has not already been commanded to do so.

**Comm. Modes:** Remote Control.

## *Command Code EE – Enquire Status Extended*

This command retrieves the operational status of AccuLoad III and is simply an extended version of the EQ command. Data is returned in a bit-mapped format.

**Command:**

**"EE "**

**Responses:**

**OK: Good Response** Command accepted and AccuLoad will attempt to print reports.

$A_1$ through $A_{20}$

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

or...

**"NOXX"** Data not retrieved.

***Note:*** *A1- A16 are in the same format as the EQ command; A19 – A20 not used.*

**Remarks:** For bay configurations, SA, SF, TD and TP refer to the bay; other flags continue to be specific to the arm.

If the AccuLoad is unable to start a transaction because it would require over-writing a protected or pending transaction report, the AccuLoad will respond to SB, AU and AP commands with a NO10 (storage full).

**Constraints:** None

**Special Case:** See notes under tables.

**Comm. Modes:** Polling Only, Poll & Authorize, Remote Control, Poll & Program.

## *Response to "EE" Enquire, Character A17*

| Character | | Condition | | | |
|---|---|---|---|---|---|
| **Char** | **Hex** | **Pending Report*** | **Pending Report Storage Full*** | **Printer Standby*** | **New Valve Diagnostic Data**** |
| 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 EE – Character A17**

* See PP command for information.

** See VD command for information.

### *Response to "EE" Enquire, Character A18*

| Character | | Condition | | | |
|---|---|---|---|---|---|
| **Char** | **Hex** | **Reserved** | **Reserved** | **Nedap Comm Fail** | **New Vehicle Tag Data** |
| 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 EE – Character A18**

## *Command Code EQ – Enquire Status*

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

**Command:**

| |
|---|
| **"EQ"** Request Status |

**Responses:**

| | |
|---|---|
| **"AAAAAAAAAAAAAAAA"**<br><br>Where each "A" is a quasi hex value;<br>"0 1 2 3 4 5 6 7 8 9 : ; < > ?". | **Good Response:** 16 Characters. For descriptions of each of the characters, see the following pages. |

**Remarks:** Allow for additional characters to be added on the end for future status indicators. For bay configurations – SA, SF, TD, and TP – refer to the bay. Other flags continue to be specific to the load arm.

**Constraints:** None.

**Special Case:** See notes under tables.

**Comm. Modes:** Polling only, Poll & Authorize, Remote Control, Poll & 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 AccuLoad III 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. "Alarm Conditions" are not conditions that will result in an AccuLoad alarm being generated; they are combinations of status conditions that should be unable to occur at the same time.*

## *Response to "EQ" Enquire, Character A3*

| Character | | Condition | | | |
|---|---|---|---|---|---|
| **Char** | **Hex** | **Alarm On** | **Standby Transactions Exist** | **Storage Full** | **In Standby Mode** |
| 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 reset through the communication channel. (See Alarm Reset command "AR.")*

***Note:*** *The "(not used)" positions in this table reflect the attempt to remain mostly compatible with AccuLoad II. Many of these values were used for standby mode operations; they shall be reported as "off" in the AccuLoad III-X.*

## *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 #2** | **Input #3** |
| 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** | **Input #4** | **Input #5** | **Input #6** | **Input #7** |
| 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 A6**

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

## *Response to "EQ" Enquire, Character A7*

| Character | | Condition | | | |
|---|---|---|---|---|---|
| **Char** | **Hex** | **Input #8** | **Input #9** | **Input #10** | **Input #11** |
| 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 A7**

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

### *Response to "EQ" Enquire, Character A8*

| Character | | Condition | | | |
|---|---|---|---|---|---|
| **Char** | **Hex** | **Input #12** | **Input #13** | **Input #14** | **Input #15** |
| 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 A8**

## *Response to "EQ" Enquire, Character A9*

| Character | | Condition | | | |
|---|---|---|---|---|---|
| **Char** | **Hex** | **Input #16 (BIO #1) *** | **Input #17 (BIO #2) *** | **Input #18 (BIO #3) *** | **Input #19 (BIO #4) *** |
| 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 A9**

* If configured as an input

## *Response to "EQ" Enquire, Character A10*

| Character | | Condition | | | |
|---|---|---|---|---|---|
| **Char** | **Hex** | **Input #20 (BIO #5) *** | **Input #21 (BIO #6) *** | **Input #22 (BIO #7) *** | **Input #23 (BIO #8) *** |
| 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 A10**

* If configured as an input

### *Response to "EQ" Enquire, Character A11*

| Character | | Condition | | | |
|---|---|---|---|---|---|
| **Char** | **Hex** | **Input #24 (AICB #1)** | **Input #25 (AICB #1)** | **Input #26 (AICB #1)** | **Input #27 (AICB #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 A11**

### *Response to "EQ" Enquire, Character A12*

| Character | | Condition | | | |
|---|---|---|---|---|---|
| **Char** | **Hex** | **Input #28 (AICB #1)** | **Input #29 (AICB #1)** | **Input #30 (AICB #1)** | **Input #31 (AICB #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 A12**

### *Response to "EQ" Enquire, Character A13*

| Character | | Condition | | | |
|---|---|---|---|---|---|
| **Char** | **Hex** | **Input #32 (AICB #1)** | **Input #33 (AICB #1)** | **Input #34 (AICB #2)** | **Input #35 (AICB #2)** |
| 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 A13**

### *Response to "EQ" Enquire, Character A14*

| Character | | Condition | | | |
|---|---|---|---|---|---|
| **Char** | **Hex** | **Input #36 (AICB #2)** | **Input #37 (AICB #2)** | **Input #38 (AICB #2)** | **Input #39 (AICB #2)** |
| 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 A14**

### *Response to "EQ" Enquire, Character A15*

| Character | | Condition | | | |
|---|---|---|---|---|---|
| **Char** | **Hex** | **Input #40 (AICB #2)** | **Input #41 (AICB #2)** | **Input #42 (AICB #2)** | **Input #43 (AICB #2)** |
| 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 A15**

## *Response to "EQ" Enquire, Character A16*

| Character | | Condition | | | |
|---|---|---|---|---|---|
| **Char** | **Hex** | **Printing In Progress** | **Permissive Delay** | **Card Data Present** | **Preset in Progress** |
| 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 A16**

## *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:** Poll & Authorize, Remote Control, Poll only, Poll & Program.

## *Command Code ES – Last Event Recall*

This command requests the sequence number for a specified event stored by the AccuLoad III.

**Command:**

**"ES"** (deprecated form – use "ES_N")
**"ES_X"**

Where: X = "N" for newest (most recent) event
"O" for oldest event
"D" for most recent event on or before date/time specified
For "ES_D", the date argument above must be in the format specified.

Where:
DDDDDDDD = Date (MMDDYYYY standard convention or DDMMYYYY military convention,
HHNN = Time
X = Time type (A for AM, P for PM, M for military)

**Responses:**

**Good Response -**
**"ES_SSSSSSSSSS"** (for deprecated form)
**"ES_X_SSSSSSSSSS"** (for all others)
Where:
SSSSSSSSSS = Sequence number

or...

**"NOXX"** Data not available

**Remarks:** None.

**Constraints:** None

**Special Case:** None

**Comm. Modes:** Poll & Authorize, Remote Control, Poll only, Poll & Program.

## *Command Code ET – End Transaction*

This command removes authorization and flags the transaction as complete.

**Command:**

**"ET"**

**Responses:**

**"OK"** **Good Response.** Transaction is flagged as complete if a transaction is in progress. Authorization is removed.

or...

**"NOXX"** Transaction was not ended

**Remarks:** If bays are configured, this command will end the transaction for the entire bay.

**Constraints:** None.

**Special Case:** None.

**Comm. Modes:** Poll & Authorize, Poll & Program, or Remote Control

## *Command Code FL – Read Flow Count*

This command retrieves raw pulse and volume counts from the AccuLoad III.

**Command:**

| | |
|---|---|
| **"FL"** | Read flow count for the arm (straight, sequential, ratio, hybrid sequential meter) |
| **"FL_Px"** | Read raw pulse counts for specified product (ratio blending only). Read raw pulse counts for specified ratio products (only) on a hybrid arm. |
| **"FL_R"** | Read raw volume or mass for the arm (straight, sequential, ratio, hybrid sequential meter). |
| **"FL_Px_R"** | Read raw volume or mass for specified product (ratio blending only and products only on a hybrid arm.) |

Where: x = Product Number 1, 2, 3, 4, 5, or 6 for ratio blending
X = Product Number 1, 2, 3, 4, 5 for hybrid arm (only if ratio product)

**Responses:**

| | |
|---|---|
| **"FL_VVVVVVVVV"** | **Good Response.** Flow count for arm (straight product or ratio blending) |
| **"FL_VVVVVVVVV_Px"** | **Good Response.** Flow count for specified product (ratio, sequential, hybrid.) |
| **"FL_VVVVVVV_VV"** | **Good Response.** Raw volume or mass amount for the arm. |
| **"FL_VVVVVVV.VV_Px"** | **Good Response.** Raw volume or mass amount for specified product. |

Where: VVVVVVVVV = Raw Pulse Count

**"NOXX"** Did not receive the raw counts from the AccuLoad

**Remarks:**

For straight product applications the response will not include the Px parameter. The count is reset to zero at the start and end of each transaction.

**Constraints:**

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

Sequential blending applications and Hybrid Arm Sequential Products: The Px parameter will indicate the component currently flowing. In this case, the flow count may not reflect the pulses for the component indicated. For example, if the first component delivered 1000 pulses and the second component delivered 750 pulses, the response will show FL 1000 P1 after component one has completed its delivery and FL 1750 P2 after component two completes its delivery.

Ratio blending applications and Hybrid Arm Sequential Products: FL without the Px parameter will return the current pulse count through the arm, i.e., the sum of the two component products' flow count. The Px parameter will not be returned.

NO31 will be returned if the command FL_Px is directed to an arm currently configured for straight product or sequential blending. NO 31 will be returned if the product regulated is a sequential product on a hybrid arm.

**Special Case:** Px reflects the product currently being loaded or the last product that was loaded, if loading is not currently active. Stray pulses from the meter may accumulate and register in the FL command between loads. In the event of unauthorized flow such as this, the Px product designation does not necessarily indicate which product is leaking.

**Comm. Modes:** Polling only, Poll & Authorize, Remote Control, Poll & Program

## *Command Code FS – Full Screen*

This command will cause the AccuLoad to change the viewport currently associated with it to a full screen mode. If the AccuLoad is currently full screen, there will be no action. If the AccuLoad is in the 5- or 6-arm mode of operation, with a summary screen on the second display, then the AccuLoad shall move the focus to the viewport specified via the address.

**Command:**

**"FS"**

**Responses:**

**"OK"** Good Response

OR

**"NOXX"**
Where: X = "00" If there are any trailing characters after the second character in the two-byte code
= "01" If the unit is in Program Mode
= "06" if the other arm is in use (prompts or authorized)
= "07" If the port is not in "Remote Control" or "Poll & Authorize" Mode

**Remarks:** If transaction is in progress and at dynamic displays, the dynamic displays will be exited and the delivery screen will be displayed. If the AccuLoad is displaying the "Main Menu" or any of its sub-menus, that menu will be exited and the ready screen will be displayed.

**Constraints:** If the other arm using this display is in use (i.e., displaying WD prompts or authorized), the FS command is not allowed and will return an "NO06." This prevents an arm in use from being hidden behind another arm.

**Comm. Modes:** Polling only, Poll & Authorize, Remote Control, Poll & Program

## *Command Code GD – Get Date and Time*

This command retrieves the current date and time from the AccuLoad III.

**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 AccuLoad III

**Remarks:** None.

**Constraints:** None.

**Special Case:** None.

**Comm. Modes:** Polling only, Poll & Authorize, Remote Control, Poll & Program

## *Command Code GK – Get Key*

This command retrieves the last key that was pressed at the AccuLoad III keypad.

**Command:**

| |
|---|
| **"GK"** |

**Responses:**

| | |
|---|---|
| **"GK_XX"** | **Good Response.** Last key pressed. |
| **"GK*XX"** | **Good Response.** Last key pressed, this arm has current focus. |
| Where: XX | = "_0" through "_9" for numeric keys<br>= "E1" for the "ENTER" key<br>= "P1" for the "PRINT" key<br>= "A1" for the "START" key<br>= "B1" for the "SET" key<br>= "C1" for the "CLEAR" key<br>= "S1" for the "STOP" key<br>= "F1" for the "F1" key<br>= "F2" for the "F2" key<br>= "- +" for the "+/-" key<br>= "_." for the "." key |

**Remarks:** None.

**Constraints:** NO32 will be returned if no key has ever been pressed.

**Special Case:** None.

**Comm. Modes:** Polling only, Poll & Authorize, Remote Control, Poll & Program

## *Command Code GP – Get PROM Signatures*

This command requests that the computed CRC-16 check on each of the PROMs in the AccuLoad III.

**Command:**

| | |
|---|---|
| **"GP"** | Retrieve Firmware CRC |
| **"GP_R"** | Retrieve Firmware Revision |

**Responses:**

| | | |
|---|---|---|
| **"GP_SSSSSSSS"** | **Good Response.** | Firmware CRC |
| **"GP_XX.XX Revision"** | **Good Response.** | Firmware Revision |

Where: SSSSSSSS = eight character hexadecimal digit signature of the CRC
XX.XX = firmware revision

or...

**"NOXX"** Did not read the EPROM CRC signatures.

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

**Constraints:** None.

**Special Case:** None.

**Comm. Modes:** Polling only, Poll & Authorize, Remote Control, Poll & Program

## *Command Code LD – Request Load Average Density*

This command requests load average density from the AccuLoad III.

**Command:**

**Current Transaction**

**"LD_R"** Current batch average density

**"LD_P"** Current product batch average density

**"LD_Px"** Current batch average density of the specified product

**"LD_YY"** Batch average density for the specified batch

**"LD_YY_Px"** Batch average density for the specified batch and the specified product

**Local Storage**

**"LD_YY_NNN"** Batch average density of a stored transaction.

**"LD_YY_Px_NNN"** Batch average density for a specified product of a stored transaction.

Where: R = Indicated Current Batch
P = Constant (Current Product)
Px = Product 1, 2, 3, 4, 5, or 6
NNN = Number of transactions back into Local Storage to retrieve data.
YY = Batch Number requested (01-10)

**Responses:**

**Current Transaction**

**"LD_YY_RR_SVVVV.V"** **Good Response.** For commands LD_R, LD_YY

**"LD_YY_Px_VVVV.V"** **Good Response.** For commands LD_P, LD_Px, LD_YY_PX

**Local Storage**

**"LD_YY_RR_SVVVV.V_NNN"** **Good Response.** For commands LD_YY_NNN

**"LD_YY_Px_VVVV.V_NNN"** **Good Response.** For commands LD_Px_NNN

Where:

VVVV.V = average value (where most significant digit can be "-" minus sign)
NNN = Number of transactions back into Local Storage to retrieve data.
YY = Batch number (01-10)
RR = Recipe Number (01-50)
Px = Product 1, 2, 3, 4, 5, or 6
S = Sign (+/-)

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 AccuLoad III.

NO06 will be returned for a product request if the product requested was not delivered in the transaction.

NO30 will be returned if the product specified is not currently configured to the arm to which the command was directed.

NO31 will be returned if product load averages are requested for an arm currently configured for straight product delivery.

NO39 (No current batch on this arm) will be returned if transaction is in progress on bay but there is no active batch on the addressed arm. "YY" specifies the bay-based batch number.

**Special Case:** None.

**Comm. Modes:** Polling only, Poll & Authorize, Remote Control, Poll & Program

## *Command Code LO – Request Logout of Program Mode*

This command forces an immediate logout of Program Mode and all changes made by the PC command to be saved.

**Command:**

**"LO"**

**Response:**

**OK**

or

**NOXX**

**Remarks:** Ten seconds after issuing a PC command, the AccuLoad 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:** Poll & Authorize, Remote Control, Poll & Program

## *Command Code LP – Request Load Average Pressure*

This command requests the value of the load average pressure from the AccuLoad III.

**Command:**

**Current Transaction**

| | |
|---|---|
| **"LP_R"** | Current load average pressure for the current recipe. |
| **"LP_YY"** | Load average pressure for previous batch of the current transaction. |
| **"LP_P"** | Current load average pressure for the current product. |
| **"LP _Px"** | Current load average pressure of specified product (ratio, sequential). |
| **"LP_YY_Px"** | Batch average pressure for a specified product (sequential, ratio). |

**Local Storage Transaction**

| | |
|---|---|
| **"LP_YY_NNN"** | Load average pressure for the batch indicated in local storage. |
| **"LP_YY_Px_NNN"** | Batch average pressure for a specified product stored in local storage (sequential ratio). |

Where: R = Current Batch
P = Current Product
Px = Product 1, 2, 3, 4, 5, or 6
YY = Batch Number requested (01-10)
NNN = Number of transactions back into local storage to retrieve data

**Responses:**

**Current Transaction**

**"LP_YY_RR_VVVV.V"** For "LP_R" and "LP_YY"

**"LP_YY_Px_VVVV.V"** For command "LP_P," "LP_Px, LP_YY_Px"

**Local Storage Transaction**

**"LP_YY_Px_VVVV.V_NNN"** For "LP_YY_Px_NNN"

**"LP_YY_RR_VVVV.V_NNN"** For "LP_YY_NNN"

Where: YY = Batch Number requested (01-10)
RR = Recipe (01-50)
Px = Product 1, 2, 3, 4, 5, or 6
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:** Pressure units are as programmed for the AccuLoad III.

NO06 will be returned for a product request if the product requested was not delivered in the transaction.

NO30 will be returned if the product specified is not currently configured to the arm to which the command was directed.

NO31 will be returned if product load averages are requested for an arm currently configured for straight product delivery.

NO39 (No current batch on this arm) will be returned if transaction is in progress on bay but there is no active batch on the addressed arm. "YY" specifies the bay-based batch number.

**Special Case:** None.

**Comm. Modes:** Polling only, Poll & Authorize, Remote Control, Poll & Program

## *Command Code LS – Request Last Density Sample*

This command requests the last density sample taken during an unloading batch.

**Command:**

**Current Transaction**
**"LS"** Last density sample from the current batch
**"LS_YY"** Last density sample for a specified batch

**Local Storage**
**"LS_YY_NNN"** Returns the density sample for a previously completed batch

Where:
YY = Batch Number (01 – 10)
NNN = Number of transactions back into Local Storage to retrieve data

**Responses:**

**Current Transaction**
**"LS_YY_XXXX.X"** Good response

**Local Storage**
**"LS_YY_XXXX.X_NNN"** Good Response

Where:
YY = Batch Number (01 – 10)
XXXX.X = Last density sample in programmed density units
NNN = Number of transactions back into Local Storage to retrieve data

**Or…**

**"NOXX"** The last density sample was not read.

Where:
XX = "03" Value rejected
= "05" No transaction ever done
= "19" Option not installed

**Remarks:** If bays are configured, LS will return the sample for the current batch on the arm addressed, and CP YY will return the percentage for the bay batch number specified by YY (which may be a batch from another arm on the bay).

**Constraints:** (1) Density units are as programmed in the AccuLoad III.
(2) Applies to arms configured as unloading arms only. Other arm configurations will result in NO19 being returned.

**Special Case:** None.

**Comm. Modes:** Polling only, Poll & Authorize, Remote Control, Poll & Program

## *Command Code LT – Request Load Average Temperature*

This command requests the value of the load average temperature from the AccuLoad III.

**Command:**

**Current Transaction**

| | |
|---|---|
| **"LT_R"** | Current batch average |
| **"LT_YY"** | Batch average |
| **"LT_YY_NNN"** | Batch average for local storage transaction |
| **"LT_P"** | Current average of the current product (sequential, hybrid sequential product) |
| **"LT_Px"** | Current average of specified product (ratio, sequential, hybrid) |
| **"LT_YY_Px"** | Batch average for specified product (ratio, sequential, hybrid) |
| **"LT_YY_Px_NNN"** | Batch average for specified product for local storage transaction (sequential, ratio, hybrid) |

Where: R = Current Batch
P = Current Product
Px = Product 1, 2, 3, 4, 5, or 6
YY = Batch Number requested
NNN = Number of transactions back into local storage to retrieve data

**Responses:**

**Current Transaction**

**"LT_YY_RR_SVVVV.V"** For command(s) "LT_R", "LT_XX"
**"LT_YY_RR_SVVVV.V_NNN"** For command(s) "LT_XX_NNN"
**"LT_YY_Px_SVVVV.V"** For command(s) "LT_P", "LT_Px", "LT_XX_Px"
**"LT_YY_Px_SVVVV.V_NNN"** For command(s) "LT_XX_Px_NNN"

Where: YY = Batch Number requested
RR = Recipe 01-50
S = Sign (+ or -)
Px = Product 1, 2, 3, 4, 5, or 6
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 AccuLoad III. Negative temperature is possible.

NO06 will be returned for a product request if the product requested was not delivered in the transaction.

NO30 will be returned if the product specified is not currently configured to the arm to which the command was directed.

NO31 will be returned if product load averages are requested for an arm currently configured for straight product delivery.

NO39 (No current batch on this arm) will be returned if transaction is in progress on bay but there is no active batch on the addressed arm. "YY" specifies the bay-based batch number.

**Special Case:** None.

**Comm. Modes:** Polling only, Poll & Authorize, Remote Control, Poll & Program

## *Command Code MC – Mass Meter Send*

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

**Command:**

**"MC_m…m"**

Where:

m…m = Mass Meter command text as defined in MN0M008 for the S-Mass, or MN0M012 for the Apollo

**Responses:**

**OK** Good response

Or …

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

Where:

XX = "01" if the AccuLoad is in Program Mode
= "02" if the AccuLoad 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:** Polling only, Poll & Authorize, Remote Control, Poll & Program

## *Command Code MP – Minimum Preset*

This command requests the minimum preset allowed for the recipe specified.

**Command:**

**"MP_RR"**

Where: RR = recipe number (01-50)

**Responses:**

| | |
|---|---|
| **"MP_RR_VVVVVV"** | **Good Response.** |
| Where: RR | = Recipe Number (01-50) |
| VVVVVV | = Minimum Preset |
| or... | |
| **"NOXX"** | Minimum preset not returned |

**Remarks:** None.

**Constraints:** NO30 will be returned if the recipe specified is not currently configured to the arm (or bay, if bays are configured) to which the command was directed.

**Special Case:** None.

**Comm. Modes:** Polling only, Poll & Authorize, Remote Control, Poll & Program

## *Command Code MR – Mass Meter Receive*

This command allows a host system to receive commands from a mass meter connected to an AccuLoad.

**Command:**

**"MR"**

**Responses:**

**"MR_r…r"** Good response

Where:
r…r = Mass Meter response text as defined in MN0M008 for the S-Mass, or for MN0M012 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:** Polling only, Poll & Authorize, Remote Control, Poll & Program

## *Command Code MS – Override MMI Swing Arm Position*

This command allows the host to override the current state of the swing arm detect switch, and force an arm to be displayed on a particular MMI regardless of the swing arm's current position. This is useful when a driver needs to be notified to swing the arm to the other loading position.

**Command:**

**"MS_X"**
Where:
X = A to move arm's display to side A (MMI_A)
X = B to move arm's display to side B (MMI_B)

**Responses:**

**Good Response: OK**

or

**"NOXX"**

Where:
XX = "01" In the Program Mode
"03" Value Rejected
"07" Wrong Control Mode
"13" AccuLoad Authorized
"19" Option Not Installed

**Remarks:** Once the host has sent a command to override the detect switch, the AccuLoad will continue to ignore the detect switch setting until either the swing arm is moved into the requested position or the host sends another "MS" command to move the arm to the correct MMI.

If an automation command is sent to a KDC to authorize a transaction (i.e., AU, AP or SB) and the swing arm is not in the correct position, the AccuLoad will respond with the following new error code: **"NO35" – Swing Arm not in correct position**.

**Constraints:** None.

**Special Case:** A "NO19" response occurs when there are no detect switches configured.

**Comm Modes:** Poll & Authorize, Remote Control

## *Command Code NR – New Recipe*

This command requests is valid only for Unlimited Preset arms. It allows for on-the-fly recipe changes during delivery. This command will also authorize an unlimited preset arm as if an AU/AP or SB command was sent.

**Command:**

**"NR_RR"**

Where: RR = Recipe Number (1-50)

**Responses:**

**OK: Good Response.**

or...

**"NOXX"**

**Remarks:** To specify an additive mask via communications you should continue to use the AB command to select the recipe, followed by the AU/AP or SB to specify the additive mask.

If the NR is received during a batch, the recipe will be changed immediately. The portion of the batch already delivered will not be affected (except any existing deviation from the desired ratio will still be taken into account if possible). The remainder of the batch will be delivered using the blend percentages specified by the new recipe.

If Recipe Select inputs are defined, the NR command will override the recipe selected via the digital inputs.

**Constraints:** Unlimited preset mode only.

**Special Case:** None.

**Comm. Modes:** Polling only, Poll & Authorize, Remote Control, Poll & Program.

## *Command Code OA – Other Address*

This command requests the address of the other arm(s) configured in this AccuLoad III.

**Command:**

**"OA"**

**Responses:**

**"OA_WW_XX_YY_ZZ_AA_BB" Good Response.**

Where: WW = address of Arm 1 (00 – 99)
XX = address of Arm 2 (00 – 99)
YY = address of Arm 3 (00 – 99)
ZZ = address of Arm 4 (00 – 99)
AA = address of Arm 5 (00 – 99)
BB = address of Arm 6 (00 – 99)

or...

**"NOXX"** The address was not returned.

**Remarks:** A value of zero indicates the arm cannot be addressed via communications.

**Constraints:** None.

**Comm. Modes:** Polling only, Poll & Authorize, Remote Control, Poll & 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 – 78)
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 AccuLoad 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:** Polling only, Poll & Authorize, Remote Control, Poll & Program.

## *Command Code PC – Change Program Code Values*

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

**Command:**

**"PC_DD_XXX_V...V"**

or…

**"PC_DD_XXX+VVV.V"** For alternate flowing point response (see note)

Where: DD = Major Directory
= CF – Configuration
= SY – System
= AR – Arm
= Mx – Meter 1, 2, 3, 4, 5, or 6
= Px – Product 1, 2, 3, 4, 5, or 6
= BA – Bay A
= BB – Bay B
= 01-50 – 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 amount 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.

NO01 will be returned if another load arm is in the program mode locally or via communications.

**Special Case:** None.

**Comm. Modes:** Poll & Authorize, Remote Control, Poll & 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 01 005 | PV 01 005 000.0 1st Percentage |
| 01PC 01 005 23.36 | PC 01 005 023.4 1st Percentage |
| 01PV 01 005 | PV 01 005 023.4 1st Percentage |
| 01PV 01 005+ | PV 01 005 023.36 1st Percentage |
| 01PC 01 005+23.64 | PC 01 005 023.64 1st Percentage |
| 01PV 01 005 | PV 01 005 023.6 1st Percentage |
| 01PV 01 005+ | PV 01 005 023.64 1st Percentage |

## *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_AAAA_X" Good Response.**

| | | |
|---|---|---|
| Where: | DDDDDDDD | = Power-fail Date |
| | | = (MMDDYYYY for Standard Time) |
| | | = (DDMMYYYY for Military Time) |
| | AAAA | = Power-fail Time (HHMM) |
| | X | = A (Standard Time – A.M.) |
| | | = P (Standard Time – P.M.) |
| | | = M (Military Time) |

**Remarks:** None.

**Constraints:** None.

**Special Case:** None.

**Comm. Modes:** Polling only, Poll & Authorize, Remote Control, Poll & Program.

## *Command Code PN – Batch Net Mass for Vapor Recovery System*

This command retrieves the net mass for a batch. The net mass is the mass measured from the delivery meter (ex. Meter #1) less the mass measured by the vapor recovery meter (ex. Meter #2).

**Command:**

| | |
|---|---|
| **"PN"** | for the current batch net mass |
| **"PP YY"** | for the completed batch net mass |

**Responses:**

| | | |
|---|---|---|
| **"PN_MMMMMMM.MM"** | **Good Response.** | Current batch net mass |
| **"PN_YY_MMMMMMM.MM"** | **Good Response.** | Completed batch net mass |

Where: YY = batch number (1-10)
MMMMMMM.MM = batch net mass
or

**"NOXX"**

**Remarks:** None.

**Constraints:** None.

**Special Case:** None.

**Comm. Modes:** Polling only, Poll & Authorize, Remote Control, Poll & Program

## *Command Code PP – Print Report to Printer*

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

***Note:*** *This command code is available only when using Revision 10 and above firmware.*

**Command:**

| | |
|---|---|
| **"PP"** | for the most recently completed transaction |
| **"PP NNN"** | for NNN transactions back in local storage |
| **"PP ST"** | for standby transactions that have failed to print |

**Responses:**

**Good Response:**

**"OK"**

or

**"NOXX"**
Where: XX = "41" No pending reports to print

**Remarks:** None.

**Constraints:** A printer port must be configured.

**Special Case:** None.

**Comm. Modes:** Polling only, Poll & Authorize, Remote Control, Poll & Program

## *Command Code PR – Program Change Recall*

This command retrieves program mode changes from the AccuLoad'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_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 | = 0_WW_ZZZ <Tab> Old Data <Tab> New Data |
| WW | = Directory designation |
| 00 | = System |
| P | = Product Number (01 – 06) |
| 00 | = Recipe Number (01 – 50) |
| ZZZ | = Program Code Number |

Or…

**"NOXX"** Data not retrieved.

**Remarks:** None.

**Constraints:** None.

**Special Case:** None.

**Comm. Modes:** Poll & Authorize, Remote Control, Poll Only, Poll & Program.

## *Command Code PS – Last Program Code Change Recall*

This command retrieves the last changed program codes sequence number for a program mode change stored by AccuLoad III in the program change (audit) log.

**Command:**

**"PS"** (Deprecated form use "PS_N")
**"PS_X"**
**"PS_X_DDDDDDDD_HHNN_Y"** (for X = "D")

Where: X = N = Newest (most recent) entry
O = Oldest entry still available
D = Most recent entry on or before a specified date/time

For **"PS_D"** the date argument above must be in the format specified

Where: DDDDDDDD is the date (MM DD YYYY for Standard Convention and DD MM YYYY for Military Convention)

HHH is the time

X is the time type (A for AM, P for PM, M for Military)

**Responses:**

**"PS_SSSSSSSSSS"** **Good Response.** (Deprecated form)
**"PS_X_SSSSSSSSSS"**(all others)

Where:
SSSSSSSSSS = Sequence number

or...

**"NOXX"** Sequence number not retrieved.

**Remarks:** None.

**Constraints:** None.

**Special Case:** None.

**Comm. Modes:** Poll & Authorize, Remote Control, Poll only, Poll & 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 AccuLoad 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.

***Note:*** *This command code is available only when using Revision 10 and above firmware.*

**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:** A printer port must be configured.

**Special Case:** This case is not supported via TCP/IP (Ethernet or SLIP)

**Comm. Modes:** Polling only, Poll & Authorize, Remote Control, Poll & Program

## *Command Code PV – Request Program Code Values*

This command requests program values from the AccuLoad III.

**Command:**

**"PV_DD_XXX"**

or…

**"PV_DD_XXX+ "** For alternate floating point response (see note)

Where: DD = Major directory
= CF – Configuration
= SY – System
= AR – Arm
= Mx – Meter 1, 2, 3, 4, 5, or 6
= Px – Product 1, 2, 3, 4, 5, or 6
= 01-50 – Recipe Number
= BA – Bay A
= BB – Bay B
XXX = Program Code Number
\+ = See note on next page

**Responses:**

**"PV_DD_XXX_V...V"** Good Response for three-digit codes

Where: DD = Directory
= CF – Configuration
= SY – System
= AR – Arm
= Mx – Meter 1-6
= Px – Product 1-6
= 01-50 – Recipe Number
= BA – Bay A
= BB – Bay B
XXX = Three-digit Program Code Number
V...V = Value of the parameter

or...

**"NOXX"** Program value not read.

**Remarks:** None.

**Constraints:** None.

**Special Case:** None.

**Comm. Modes:** Polling only, Poll & Authorize, Remote Control, Poll & 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 01 005 | PV 01 005 000.0 1st Percentage |
| 01PC 01 005 23.36 | PC 01 005 023.4 1st Percentage |
| 01PV 01 005 | PV 01 005 023.4 1st Percentage |
| 01PV 01 005+ | PV 01 005 023.36 1st Percentage |
| 01PC 01 005+23.64 | PC 01 005 023.64 1st Percentage |
| 01PV 01 005 | PV 01 005 023.6 1st Percentage |
| 01PV 01 005+ | PV 01 005 023.64 1st Percentage |

## *Command Code PX – Transaction Net Mass for Vapor Recovery System*

This command retrieves the net mass for a transaction. The net mass is the mass measured from the delivery meter (ex. Meter #1) less the mass measured by the vapor recovery meter (ex. Meter #2).

**Command:**

| | |
|---|---|
| **"PX"** | for the current transaction net mass |
| **"PP NNN"** | for the historic transaction request |

**Responses:**

| | | |
|---|---|---|
| **"PN_MMMMMMM.MM"** | **Good Response.** | Current transaction net mass |
| **"PN_YY_MMMMMMM.MM"** | **Good Response.** | Historical transaction net mass |

Where: NNN = number of transactions back into local storage to retrieve data
MMMMMMM.MM = transaction net mass

or

**"NOXX"**

**Remarks:** None.

**Constraints:** None.

**Special Case:** None.

**Comm. Modes:** Polling only, Poll & Authorize, Remote Control, Poll & Program

## *Command Code RA – Request Alarm Status*

This command requests current alarm conditions from the AccuLoad III.

**Command:**

**"RA_DD"**

Where: DD = Directory
= SY – System
= P1 – Product 1
= P2 – Product 2
= P3 – Product 3
= P4 – Product 4
= P5 – Product 5
= P6 – Product 6
= AR – Arm
= IN – Injector
= M1 – Meter 1
= M2 – Meter 2 (ratio, hybrid blending configuration only)
= M3 – Meter 3 (ratio, hybrid blending configuration only)
= M4 – Meter 4 (ratio, hybrid blending configuration only)
= M5 – Meter 5 (ratio, hybrid blending configuration only)
= M6 – Meter 6 (ratio, hybrid blending configuration only)

**Responses:**

| | |
|---|---|
| **"HF OA TP"** | **Good Response.** A character string consisting of from 1 to 5 status codes separated by a single space. Each status code is two characters. |
| **"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:** NO31 will be returned if the command format does not match the current configuration of the arm.

**Special Case:** If no alarm condition is set, an “OK” response is issued.

**Comm. Modes:** Polling only, Poll & Authorize, Remote Control, Poll & Program.

## *Command Code RB – Request Batch Totals*

This command retrieves Batch Data from the AccuLoad III.

**Command:**

**Current Transaction**

| | |
|---|---|
| **"RB"** | Current batch volume, delivery volume type |
| **"RB_Px"** | Current batch, specified product (ratio, sequential, hybrid) |
| **"RB_YY"** | Completed batch volume, delivery volume type |
| **"RB_YY_X"** | Completed batch volume, specified volume type |
| **"RB_YY_Px"** | Product volume of batch, delivery volume type |
| **"RB_YY_X_Px"** | Product volume of batch, specified volume type |

**Local Storage Transaction**

| | |
|---|---|
| **"RB_YY_NNN"** | Historic transaction requests |
| **"RB_YY_Px_NNN"** | Returns the product volume type |
| **"RB_YY_X_NNN"** | Returns the batch in the volume type requested if that volume type was stored. |
| **"RB_YY_Z_Px_NNN"** | Returns the product in the volume type requested if that volume type was stored. |

Where: X = R for raw total (Indicated Volume – IV)
= G for gross volume (GV)
= N for gross @ standard temperature volume (GST)
= P for net pressure & temperature volume (GSV)
= M for mass total
Px = Product 1, 2, 3, 4, 5, or 6
YY = Batch Number (01 – 10)
NNN = number of transactions back into Local Storage to retrieve data

**Response:**

**Current Transaction**

| | |
|---|---|
| **"RB_YY_X_A1A2A3A4A5A6_RR_VVVVVVV"** | For "RB," "RB_YY," and "RB_YY_X" |
| **"RB_YY_X_A1A2A3A4A5A6_Px_VVVVVVV"** | For "RB_YY_Px," "RB_YY_Px_X," and "RB_Px" |

**Local Storage Transaction**

| | |
|---|---|
| **"RB_YY_X_A1A2A3A4A5A6_RR_VVVVVVV_NNN"** | For "RB_XX_NNN" and "RB_XX_X_NNN" |
| **"RB_YY_X_A1A2A3A4A5A6_Px_VVVVVVV_NNN"** | For "RB_XX_Px_NNN" and "RB_XX_Px_X_NNN' |

YY = Batch number requested or current batch
X = R for raw total
= G for gross total
= N for gross at standard temperature total
= P for net temperature & pressure
= M for mass totals
A1 = Additive selection code character 1
A2 = Additive selection code character 2
A3 = Additive selection code character 3
A4 = Additive selection code character 4
A5 = Additive selection code character 5
A6 = Additive selection code character 6
Px = Product 1, 2, 3, 4, 5, or 6
RR = Recipe Number 01 – 50
VVVVVVV = Batch Volume
NNN = # of transactions back into local storage to retrieve data

or...

**"NOXX"**

**Remarks:** The batch is not considered complete until the start key is pressed for the next batch, or the transaction is ended. RB_YY will always fail on the current batch even after BD status is reported in Command EQ or RS.

See AU and AP for additive code (A1A2A3…).

If bays are configured, the returned value is the bay-based batch total.

**Constraints:** Batch volume units are assumed as the type programmed into AccuLoad III.

NO03 will be returned for a product request if the product requested was not delivered in the batch.

NO30 will be returned if the product specified is not currently configured to the arm to which the command was directed.

NO31 will be returned if the product batch totals are requested for an arm currently configured for straight product delivery.

**Special Case:** None.

**Comm. Modes:** Polling only, Poll & Authorize, Remote Control, Poll & Program.

## *Command Code RC – Request Recipe Composition*

This command requests the blend composition of the completed batch.

**Command:**

**Current Transaction**
**"RC_YY"** Request composition of the completed batch.

**Local Storage Transaction**
**"RC_YY_NNN"** Request composition of a transaction stored in local storage.

Where: YY = Batch Number
NNN = Number of transactions back into local storage.

**Responses:**

**Current Transaction**

**"RC_YY_RR_AAA.A_BBB.B_CCC.C_DDD.D_EEE.E_FFF.F" Good Response.**
Sequential Blending

**Local Transaction Storage**

**"RC_YY_RR_AAA.A_BBB.B_CCC.C_DDD.D_EEE.E_FFF.F_NNN" Good Response.**

Where: RR = Recipe Number 01-50
YY = Batch Number
AAA.A = Percentage of Product 1
BBB.B = Percentage of Product 2
CCC.C = Percentage of Product 3
DDD.D = Percentage of Product 4
EEE.E = Percentage of Product 5
FFF.F = Percentage of Product 6
NNN = Number of Transactions Back (for historical recall)

or...

**"NOXX"** The value was not read.

**Note:** Percentages are returned only for products configured for the arm. For example, if three products are configured for the load arm, the response will be shortened as the AccuLoad will suppress the product 4 percentage. Products appear in order (i.e., Product 1, Product 2…)

**Remarks:** Zero will be returned for products not delivered in the batch.

If bays are configured, the returned value is the bay-based batch number's recipe composition.

**Constraints** NO03 will be returned if the requested batch is incomplete or has not been delivered.

NO31 will be returned if the recipe composition is requested for an arm currently configured for straight product delivery.

**Special Case:** None.

**Comm. Modes:** Polling only, Poll & Authorize, Remote Control, Poll & Program.

***Note for AccuLoad II Users:*** *The RE SA command duplicates the AccuLoad II RC command functionality. The RC command in the AccuLoad III is used for recipe component data.*

## *Command Code RD – Request Current Transducer Value*

This command requests the value of one of the probes or transducers installed and wired to the AccuLoad III.

**Command:**

**"RD_Z"** Request transducer value.

**"RD_Z_Px"** Request transducer value for the specified product.

Where: Z = T (current temperature)
= P (current pressure)
= D (current density)
Px = Product 1, 2, 3, 4, 5, or 6.

**Responses:**

**"RD_Z_VVVV.V"** **Good Response.** Pressure or density, straight product

**"RD_Z_SVVVV.V"** **Good Response**. Temperature, straight product.

**"RD_Z_VVVV.V_Px"** **Good Response.** Pressure or density, ratio or sequential
or hybrid blending

**"RD_Z_SVVVV.V_Px"** **Good Response.** Temperature, ratio or sequential or hybrid blending.

*** "RD_Z_VVVV.V_VVVV.V_VVVV.V_VVVV.V"** **Good Response.** Pressure or density, ratio blender

**"RD_Z_SVVVV.V_SVVVV.V"** **Good Response.** Temperature, ratio blender

*** "RD_Z_SVVVV.V_SVVVV.V_SVVVV.V_SVVVV.V"** Good Response. Temperature, ratio blender

Where: Z = T (current temperature)
= P (current pressure)
= D (current density)
VVVV.V = current pressure or density value
SVVVV.V = current temperature value
Px = Product 1, 2, 3, 4, 5, or 6.

or...

**"NOXX"** The value was not read.

**Remarks:** Length of Response * depends on the number of meters programmed for the arm.

**Constraints** NO30 will be returned if the requested product is not allocated to the arm to which the command was directed.

NO31 will be returned if the command format does not match the current configuration for the arm.

On a hybrid arm “RD_Z” returns value for the sequential products meter. “RD_Z_Px” is for ratio assigned products only.

**Special Case:** None.

**Comm. Modes:** Polling only, Poll & Authorize, Remote Control, Poll & Program.

## *Command Code RE – Reset Status Conditions*

This command resets or acknowledges pending status conditions of the AccuLoad III.

**Command:**

**"RE_XX"**

Where: XX = two character code as detailed below.

**Responses:**

**"OK"** **Good Response.** Status condition has been reset. Status condition will no longer appear in response to status requests (EQ and RS).

or...

**"NOXX"** The status condition has not been reset.

**Remarks:** For bay configurations, the RE TD command clears the transaction done flag for the bay. RE BD still affects only the arm addressed.

**Constraints:** When using a card reader, RE CD will cancel card validation and prevent additional transactions without another card-in (if AccuLoad III is configured for card-in required options).

**Special Case:** "TD" and "BD" statuses are also reset on authorize commands.

**Comm. Modes:** Polling only, Poll & Authorize, Remote Control, Poll & Program

## *Valid Status Codes*

| Code | Condition |
|---|---|
| BD | Batch Done Status |
| CD | Card Data Is Present |
| NC | Nedap Reader Communications Fail |
| PC | Program Mode Value Has Changed |
| PF | Unit Has Power-failed |
| SA | Standby Mode |
| TD | Transaction Done Status (also resets BD) |
| VT | Vehicle Data Request |

**Valid Status Code for RE**

***Note:*** *If the status code is already reset, a "NO06" will be returned.*
***Note for AccuLoad II Users:*** *The RE SA command duplicates the AccuLoad II RC command functionality. The RC command in the AccuLoad III is used for recipe component data.*

## *Command Code RF – Recipe Flow Rates*

This command returns calculated first and second high flow rates and low flow rates for each of the products for a ratio blender. These flow rates are not the programmed flow rates. They represent the flow rates calculated by the AccuLoad according to the percentages programmed for this recipe.

**Command:**

**"RF_RR"**

Where: RR = Recipe Number (01 – 50)

**Responses:**

**"RF_RR_HHHHH_SSSSS_LLLLL_HHHHH_SSSSS_LLLLL [_HHHHH_SSSSS_LLLLL] [_HHHHH_SSSSS_LLLLL] [_HHHHH_SSSSS_LLLLL] [_HHHHH_SSSSS_LLLLL]"**

Where: HHHHH = Product high flow rate
SSSSS = Product second high flow rate
LLLLL = Product low flow rate

***Note:*** *"[ ]" denote optional values; data is returned starting with Product 1 and counting up until the number of products configured for that arm is reached.*

or...

**"NOXX"** Flow rates not returned.

**Remarks:** These flow rates are not the programmed flow rates, they represent the flow rates calculated by the AccuLoad in order to create a ratio blend which remains in spec throughout the duration of the batch, given the programmed blend ratio.

**Constraints:** NO31 will be returned if the command format does not match the current configuration of the arm/bay.

**Special Case:** None.

**Comm. Modes:** Polling only, Poll & Authorize, Remote Control, Poll & Program

## *Command Code RK – Read Keypad*

This command instructs the AccuLoad III to transmit any pending data entered at the keypad to the requesting device.

**Command:**

**"RK"**

**Responses:**

**"RK_NNNN....N"** **Good Response.**

Where:

NNNN....N is a string of from 1 to 25 characters, where the final character is a non-numeric terminating key represented as indicated below:

Numeric keypad data zero through nine will be represented by their ASCII character equivalents.

The following are accepted as terminating keys only and will be represented as follows:

Enter key ("ENTER") = "E"

Print key ("PRINT") = "P"

Start key ("START") = "A"

Set key ("SET") = "B"

The Clear key ("CLEAR") has a function and therefore will not be represented on the display. If pressed, the last character entered will be cleared and another character may be entered in its place. If pressed a second time, the entire field will be cleared and the entry will have to be repeated.

or...

**"NOXX"** Pending data was not available.

**Remarks:** The response string NNNN...N will contain leading zeros to the length of response specified in the prompt that was issued. If the expected input length for a prompt is 4 digits and only 2 digits are entered, 2 leading zeros and the 2 digits input at the keypad will be displayed in the response to the "RK." If the prompt was issued with the option for variable input field length (adding 40 to expected data length), leading zeros are still returned. If the prompt was issued with the delimiter "[" or "]," the initiating keystroke is NOT returned in the response to the "RK."

**Constraints:** Once a terminating key is pressed at the keypad the message "PLEASE WAIT" will be displayed. Keypad data is not cleared from the AccuLoad III until a "WX," "WD," or "DA" command has been issued. After data pending, the keypad is locked until a "WX" or "WD" command has been issued for prompt or a "DA" command removes minicomputer control. If none of these occur within 3 minutes, AccuLoad III will time-out and release the keypad and display from minicomputer control.

**Special Case:** None.

**Comm. Modes:** Poll & Authorize (1) or Remote Control (2)

## *Command Code RL – Show Recipes Loaded*

This command requests a bitmap 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…C13"** **Good Response.** Bit map of the loaded recipes

**Local Storage Transaction**
**"RL_C1C2C3…C13_NNN"** **Good Response.** Bit map of the loaded recipes

Where: C1…C13 are bit mapped characters indicating recipes 01-50 (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 Poll/Authorize or Remote Control) or when the AccuLoad is released for delivery for the first batch of a transaction (Polling Only). For bay-based transactions, this bitmap will show all recipes loaded for the bay.

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

**Special Case:** None.

**Comm. Modes:** Polling only, Poll & Authorize, Remote Control, Poll & Program

## *Command Code RN – Show Recipe Numbers*

This command returns the recipe numbers for each of the batches loaded in the current or previous transaction.

**Command:**

**Current Transaction**
**"RN"** Requests recipe numbers loaded in the transaction.

**Local Storage Transaction**
**"RN_NNN"** Requests recipe numbers loaded in a historical Transaction.
Where: NNN = number of transactions back into local storage

**Responses:**

**Current Transaction**
**"RN_RR_RR_RR_RR_RR_RR_RR_RR_RR_RR" Good Response.** Recipe numbers.

**Local StorageTransaction**
**"RN_RR_RR_RR_RR_RR_RR_RR_RR_RR_RR_NNN" Good Response.**
Recipes loaded in the historical transaction

Where: RR = Recipe number 01-50
NNN = number of transaction back into local storage.

or...

**"NOXX"** The recipe number has not been returned

**Remarks:** Up to ten batches may be reported. The number of batches reported will be determined by the number of batches per transaction parameter and not the number of batches actually loaded in any given transaction. For bay-based transactions, the response will include all recipes loaded for the bay.

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

If a batch was not delivered, 00 will be returned as the recipe number for the undelivered batch.

**Special Case:** None.

**Comm. Modes:** Polling only, Poll & Authorize, Remote Control, Poll & Program

## *Command Code RP – Request Preset*

This commands requests the current preset value being loaded at the AccuLoad III.

**Command:**

**"RP"**

**Response:**

**"RP_XXXXXX" Good Response.**

Where: XXXXXX = Preset Volume

**"NOXX"** The current preset value was not read.

**Remarks:** The preset value is returned with leading spaces e.g., RP_ _ _1000.

**Constraints:** The preset value is only available from the AccuLoad III while the AccuLoad III is Authorized, Released, Flowing, and Transaction in Progress. RP will return the actual preset value from the time the load is started ("SA" command or pressing the "START" key) until batch done status is set.

**Special Case:** None. If bays are configured, an RP to an arm on the bay with no batch in progress will return " 'NO39' – No current batch on arm."

**Comm. Modes:** Polling only, Poll & Authorize, Remote Control, Poll & Program

## *Command Code RQ – Request Current Flow Rate*

This command retrieves the current flow rate(s) from the AccuLoad III.

**Command:**

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

**"RQ_Px"** Flow rate for the specified product (ratio, hybrid ratio products)

**"RQ_P"** Flow rate for all component products (ratio blending, hybrid blending)

Where P = Constant for ratio blending applications
Px = Products 1, 2, 3, 4, 5, or 6 (P is a constant for ratio and hybrid blending applications)

**Responses:**

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

**"RQ_XXXXX_Px"** **Good Response.** Flow rate of a specific product

**"RQ_XXXXX_XXXXX [_XXXXX] [_XXXXX] [_XXXXX] [_XXXXX]"**
**Good Response.** Flow rates for component products (ratio only)

***Note:*** *Length of response will depend on the number of meters programmed for the arm.*

Where: XXXX = current flow rate
Px = Products 1, 2, 3, 4, 5, or 6

**"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:** Hybrid Blending: RQ_P will return flow rates for all ratio products. RQ returns flow rate for the sequential product currently flowing.

**Comm. Modes:** Polling only, Poll & Authorize, Remote Control, Poll & Program

## *Command Code RR – Request Recipe*

This command requests the current recipe number from AccuLoad III.

**Command:**

**"RR"**

**Response:**

**"RR_NN"** **Good Response.**

Where: NN = Recipe Number (01 – 50)

**"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 AccuLoad 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.

NO39 will be returned if bays are configured and an RR is sent to an arm with no batch in progress, when a bay transaction is in progress.

**Special Case:** None.

**Comm. Modes:** Polling only, Poll & Authorize, Remote Control, Poll & Program

## *Command Code RS – Request Status*

This command requests the operational status of the AccuLoad III. 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 …"** | **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 AccuLoad III 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.)
For bay configurations – SA, SF, TD, and TP – refer to the bay. Other flags continue to be specific to the load arm.

**Constraints:** None.

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

**Comm. Modes:** Polling only, Poll & Authorize, Remote Control, Poll & Program

## *Request Status Codes*

| Code | Condition |
|---|---|
| AL | Alarm active |
| AU | Arm authorized |
| BD | Arm batch done |
| CD | Card data present |
| CE | Checking entries |
| DP | Delayed prompt is in effect (“WP” or “WQ” command) |
| FL | Arm flowing |
| I1 | Input 1 on |
| I2 | Input 2 on |
| I3 | Input 3 on |
| I4 | Input 4 on |
| I5 | Input 5 on |
| I6 | Input 6 on |
| I7 | Input 7 on |
| I8 | Input 8 on |
| I9 | Input 9 on |
| IA | Input 10 on |
| IB | Input 11 on |
| IC | Input 12 on |
| ID | Input 13 on |
| IE | Input 14 on |
| IF | Input 15 on |
| IG | Input 16 (BIO 1, if configured as an input) on |
| IH | Input 17 (BIO 2, if configured as an input) on |
| II | Input 18 (BIO 3, if configured as an input) on |
| IJ | Input 19 (BIO 4, if configured as an input) on |
| IK | Input 20 (BIO 5, if configured as an input) on |
| IL | Input 21 (BIO 6, if configured as an input) on |
| IM | Input 22 (BIO 7, if configured as an input) on |
| IN | Input 23 (BIO 8, if configured as an input) on |
| JA | Input 24 (AICB, optional) on |
| JB | Input 25 (AICB, optional) on |
| JC | Input 26 (AICB, optional) on |

| Code | Condition |
|---|---|
| JD | Input 27 (AICB, optional) on |
| JE | Input 28 (AICB, optional) on |
| JF | Input 29 (AICB, optional) on |
| JG | Input 30 (AICB, optional) on |
| JH | Input 31 (AICB, optional) on |
| JI | Input 32 (AICB, optional) on |
| JJ | Input 33 (AICB, optional) on |
| JK | Input 34 (AICB, optional) on |
| JL | Input 35 (AICB, optional) on |
| JM | Input 36 (AICB, optional) on |
| JN | Input 37 (AICB, optional) on |
| JO | Input 38 (AICB, optional) on |
| JP | Input 39 (AICB, optional) on |
| JQ | Input 40 (AICB, optional) on |
| JR | Input 41 (AICB, optional) on |
| JS | Input 42 (AICB, optional) on |
| JT | Input 43 (AICB, optional) on |
| KY | Keyboard data pending |
| LR | Pending (locked) reports |
| NC | Nedap Reader Communications Fail |
| PC | Program parameter changed |
| PD | Permissive delay active |
| PF | Power-fail has occurred since last reset |
| PP | Printing in progress |
| PR | Preset in Progress |
| PS | Printer standby (printer is down) |
| PW | In program mode |
| RL | Arm released |
| RS | Pending report storage full |
| TD | Arm transaction done |
| TO | Display message timed out |
| TP | Arm transaction in progress |
| SA | In standby mode |
| SF | Storage full |
| ST | Standby transactions exist |
| VT | New vehicle ID tag available |

**Status codes for RS**

## *Command Code RT – Request Transaction Totals*

This command instructs AccuLoad III to transmit the transaction data.

**Command:**

**Current Transaction**

| | |
|---|---|
| **"RT_Z"** | Transaction Volume – sum of the recipes. |
| **"RT_Z_RR"** | Transaction single recipe volume. |
| **"RT_Z_Px"** | Transaction volume for the product specified. |

**Local Storage Transaction**

| | |
|---|---|
| **"RT_Z_NNN"** | Transaction volume - sum of the recipes. |
| **"RT_Z_RR_NNN"** | Transaction - single recipe volume. |
| **"RT_Z_Px_NNN"** | Transaction volume for the product specified. |

Where:

| | |
|---|---|
| Z | = 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 |
| Px | = Product 1, 2, 3, 4, 5, or 6. |
| RR | = Recipe 01 – 50, or "MR" to indicate multiple recipes |
| NNN | = Number of transactions back into local storage to retrieve data |

**Responses:**

**Current Transaction**

**"RT_Z_YY_RR_VVVVVVVV" Good Response.** Transaction volume – sum of the recipes
**"RT_Z_YY_Px_VVVVVVVV" Good Response**. Transaction volume – specified product

**Local Storage Transaction**

**"RT_Z_YY_RR_VVVVVVVV_NNN" Good Response.** Transaction volume – sum of the products

**"RT_Z_YY_Px_VVVVVVVV_NNN" Good Response.** Transaction volume – specified product

Where: Z = R for raw total (Indicated Volume – IV)
= G for gross volume (GV)
= N for gross at standard temperature volume (GST)
= P for gross @ standard temperature and pressure volume (GSV)
= M for mass totals
Px = Products 1, 2, 3, 4, 5, or 6
YY = Total number of batches
RR = Recipe 01-50
VVVVVVVV = Total transaction volume
NNN = Number of transactions back into local storage to retrieve data

or...

**"NOXX"** No transaction 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.

For bay configurations, the returned values represent bay transaction totals.

**Constraints:** Transaction Volume Units are assumed to be as programmed into AccuLoad III.

NO03 will be returned for a product or recipe request if the product or recipe requested was not delivered in the transaction.

NO30 will be returned if the product or recipe specified is not currently configured to the arm to which the command was directed.

NO31 will be returned if product transaction totals are requested for an arm currently configured for straight product delivery.

**Special Case:** None.

**Comm. Modes:** Polling only, Poll & Authorize, Remote Control, Poll & Program

## *Command Code SA – Remote Start*

This command remotely starts the AccuLoad III as though the "START" key was pressed.

**Command:**

**"SA"**

**Responses:**

**"OK"** **Good Response.** AccuLoad III is released for flow to begin.

or...

**"NOXX"** AccuLoad III is not released for remote start.

**Remarks:** None.

**Constraints:** NO01 will be returned if another load arm is in the program mode locally or via communications.

**Special Case:** None.

**Comm. Modes:** Poll & Authorize or Remote Control

## *Command Code SB – Set Batch*

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

**Command:**

**"SB_VVVVVV"**

**"SB_A1A2A3A4A5A6_VVVVVV"**

Where:

A1A2A3A4A5A6 = Additive selection code
VVVVVV = Volume to preset

**Response:**

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

or...

**"NOXX"** The batch volume has not been set.

**Remarks:** If the additive qualifier (A1A2A3…) is not used, the additive selection will operate as currently programmed in the AccuLoad III. For authorization with additives, only one recipe may be allotted. Additives selected in the qualifier must be programmed for use. Additives will be injected for all components of the blend recipe. 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 A1A2… 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 AccuLoad III 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 maximum batch amount being displayed; a non-zero set batch amount will override the maximum batch amount as the auto preset amount and cannot be overridden by the driver while in the Auto Preset Mode.

**Comm. Modes:** Remote Control.

## *Command Code SD – Set Date and Time*

This command sets the date and time in the AccuLoad III.

**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 AccuLoad III - 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:** Polling only, Poll & Authorize, Remote Control, Poll & Program.

## *Command Code SF – Authorize and Set Batch without Override*

This command allows an automation system to authorize a batch in remote control with a predefined batch volume that cannot be overridden by the driver. The display appears in full screen mode.

**Command:**

**"SF_VVVVVV"**

**"SF_A1A2A3A4A5A6_VVVVVV"**

Where:

A1A2A3A4A5A6 = Additive selection code
VVVVVV = Volume to preset

**Response:**

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

or...

**"NOXX"** The batch volume has not been set.

**Remarks:** If the additive qualifier (A1A2A3…) is not used, the additive selection will operate as currently programmed in the AccuLoad III. For authorization with additives, only one recipe may be allotted. Additives selected in the qualifier must be programmed for use. Additives will be injected for all components of the blend recipe. 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 A1A2A3… 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 AccuLoad III for units of measurement.

**Comm. Modes:** Remote Control.

## *Command Code SP – Remote Stop*

This command instructs the AccuLoad III to stop, halting product delivery on all arms.

**Command:**

**"SP"**

**Responses:**

**"OK" Good Response.**

or...

**"NOXX"** The AccuLoad III was not stopped.
NO36 will be returned if the AccuLoad is being used with a card reader and the card data is invalid or has expired.

**Remarks:** None.

**Constraints:** Valve and pump are shut down whether flow is present or not. If a batch is in progress, the "START" key or the Remote Start command "SA" must be used to continue the batch.

**Special Case:** None.

**Comm. Modes:** Polling only, Poll & Authorize, Remote Control, Poll & Program

## *Command Code SR – Show Recipes Currently Allocated*

This command requests from the AccuLoad III recipes that are currently allocated.

**Command:**

| |
|---|
| **"SR"** |

**Responses:**

| |
|---|
| **"SR_C1C2C3…C13" Good Response.**<br><br>Where: C1…C13 are bit mapped characters indicating recipes 01-50. See AB command for recipe bit maps for C1 – C13. |

**Remarks:**

1. Before recipes are allocated with the "AB" command, the response from the "SR" will show all recipes that are currently programmed for this load arm at the AccuLoad III. After recipes are allocated, the response from the "SR" will show only those recipes selected in the "AB." Recipe allocation is canceled when transaction done status is set.
2. See also Remarks for "AB" - Allocate Blend recipes command. (*)

**Constraints:** None.

**Special Case:** None.

**Comm. Modes:** Polling only, Poll & Authorize, Remote Control, Poll & Program.

## *Command Code SS – Split Screen*

This command causes the AccuLoad to change the viewport currently associated with it to a split screen mode. If the AccuLoad is currently split screen, there is no action. If the AccuLoad is in the 5- or 6-arm mode of operation, with a summary screen on the second display, or in any other state where a split screen does not make sense, then the AccuLoad responds with an error response.

**Command:**

**"SS"**

**Responses:**

**"OK"** Good Response

OR

**"NOXX"**
Where: X = "01" In the Programming Mode
= "06" Operation Not Allowed
= "07" Not in "Remote Control" or "Poll & Authorize" Mode

**Remarks:** If transaction is in progress and at dynamic displays, the dynamic displays will be exited and the delivery screen will be displayed. If the AccuLoad is displaying the "Main Menu" or any of its sub-menus, that menu will be exited and the ready screen will be displayed.

**Constraints:** If this arm or the other arm sharing the same display has proving enabled, the SS command is not allowed and will return an "NO06." Also, in situations where there is only one arm using this display, the SS command is not allowed. This would be the case when there are only two arms configured (one arm per display) and also when there are three arms configured and Arm #2 has the bottom display to itself.

**Comm. Modes:** Polling only, Poll & Authorize, Remote Control, Poll & Program.

## *Command Code ST – Remote Stop on Arm*

This command instructs the AccuLoad III to stop flow on an individual arm, halting product delivery on that arm only. Any other arms with flow in progress are not affected.

**Command:**

| | |
|---|---|
| **"ST"** | Stop Arm Flow |

**Responses:**

**"OK"** Good Response.

or...

**"NOXX"** The flow to the arm was not stopped.

**Remarks:** None.

**Constraints:** Valve and pump are shut down whether flow is present or not. If a batch is in progress, the "START" key or the Remote Start command "SA" must be used to continue the batch.

**Special Case:** None.

**Comm. Modes:** Polling only, Poll & Authorize, Remote Control, Poll & Program.

## *Command Code SV – Extended Services Routines*

This command is for future use.

**Command:**

**"SV_<packet>"**

Where: <packet> is a binary buffer, no longer than 1K bytes.

**Responses:**

**Good Response -**
**"SV_<packet>"**

Where: <packet> is a binary buffer, no longer than 1K bytes.

or...

**"NOXX"** Data not retrieved.

**Remarks:** None.

**Constraints:** None.

**Special Case:** None.

**Comm Modes:** Polling Only, Poll & Authorize, Remote Control, Poll & Program

## *Command Code SW – Swing Arm Position*

This command allows the host communications protocol to detect the current position of the swing arm.

**Command:**

**"SW"**

**Responses:**

**Response: SW_S**
Where:
S = “A” for Side A
“B” for Side B
“?” for arm in transit

or

**“NOXX”**
Where:
XX = “03” Value Rejected
“07” Wrong Control Mode
“19” Option Not Installed

**Remarks:** None.

**Constraints:** None.

**Special Case:** A “NO19” response occurs when there are no detect switches configured for that arm.

**Comm Modes:** Polling Only, Poll & Program, Poll & Authorize, and Remote Control

## *Command Code TA – Set Transaction*

This command sets the maximum transaction volume.

**Command:**

**"TA_V…V"**

Where: V…V = Maximum Preset Total for Transaction
Range = 0 – 999999

**Responses:**

**"OK"** **Good Response.** The transaction volume has been accepted.

or...

**"NOXX"** The transaction volume was not accepted.

**Remarks:** None.

**Constraints:** Unit value must correspond to what is programmed into the AccuLoad III for units of measurement and for delivery and preset display.

NO01 will be returned if another load arm is in the program mode locally or via communications.

**Special Case:** None.

**Comm. Modes:** Poll & Authorize or Remote Control.

## *Command Code TD – Turbine Meter Diagnostics*

This command sets the maximum transaction volume.

**Command:**

| | |
|---|---|
| **"TD_Z"** | Set Turbine Meter Diagnostics State. |
| **"TD_Z_Mx"** | Set Turbine Meter Diagnostics for the specified meter |
| Where: Z | = E – Enabled<br>= D – Disabled<br>= L – Learn Mode |
| Mx | = Meter 1, 2, 3, 4, 5, or 6 (ratio blending configuration only) |

**Responses:**

| | |
|---|---|
| **"OK"** | Good Response. State of the Turbine Meter diagnostics has been changed. |
| or... | |
| **"NOXX"** | The state of the Turbine Meter diagnostics has not been changed. |

**Remarks:** None.

**Constraints:** NO07 will be returned if the arm is not configured for Poll & Authorize or Remote Control Mode.

NO31 will be returned if the command format does not match the current configuration of the arm.

**Special Case:** None.

**Comm. Modes:** Poll & Authorize or Remote Control.

## *Command Code TI – Show Prompt Data Entry*

This command retrieves the current or stored prompts data entered in the AccuLoad III by the operator in response to one of five local prompts configured at and issued by the AccuLoad.

**Command:**

**Current Transaction**

**"TI_X"**

**Local Storage Transaction**

**"TI_X_NNN"**

Where: X = Prompt number (1 through 5)
NNN = Number of transactions back into local storage to retrieve data

**Responses:**

**Current Transaction**

**"TI_X_AAAAAAAAA"** **Good Response.**

**Local Storage Transaction**

**"TI_X_AAAAAAAAA_NNN"** **Good Response.**

Where:

X = Prompt Number (1 – 5)
AAAAAAAAA = Prompt Data Entry (Numeric)
NNN = Number of transactions back into local storage to retrieve data.

or...

**"NOXX"** No prompt data retrieved.

**Remarks:** Length of this data field will always be nine characters (right justified, no lead zero blanking).

**Constraints:** This information does not include responses to prompts issued by the communication commands WA, WD, WP, WQ or WX..

**Special Case:** None.

**Comm. Modes:** Polling only, Poll & Authorize, Remote Control, Poll & Program.

## *Command Code TN – Show Transaction Stop Date and Time*

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

**Command:**

**Current Transaction**

**"TN"**

**Local Storage Transaction**

**"TN_NNN"**

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

**Responses:**

**Current Transaction**

**"TN_IIII_DDDDDDDD_AAAA_X"** **Good Response.**

**Local Storage Transaction**

**"TN_IIII_DDDDDDDD_AAAA_X"** **Good Response.**

Where:

| | |
|---|---|
| IIII | = Internal Transaction Number |
| DDDDDDDD | = Transaction Stop Date |
| | = (MMDDYYYY for Standard Time) |
| | = (DDMMYYYY for Military Time) |
| AAAA | = Transaction Stop Time (HHMM) |
| X | = A (Standard Time - A.M.) |
| | = P (Standard Time - P.M.) |
| | = M (Military Time) |

or...

**"NOXX"** The transaction stop date and time were not retrieved.

**Remarks:** None.

**Constraints:** None.

**Special Case:** None.

**Comm. Modes:** Polling only, Poll & Authorize, Remote Control, Poll & Program.

## *Command Code TR – Read Turbine Meter Signature*

This command returns the signature set up for the Turbine Meter Diagnostics.

**Command:**

**"TR_Z_Y"** Request Signature

**"TR_Z_Y_Mx"** Request Signature for a specific meter

Where: Z = M – Meter Value
= R – Rotational Value
= B – Blade Value
Y = S – Signature Value
= D – Deviation Value
Mx = Meter 1, 2, 3, 4, 5, or 6 (ratio, hybrid blending only)

**Responses:**

**"TR_Z_Y_N…N"** **Good Response.** Current setting

**"TR_Z_Y_Mx_N…N"** **Good Response.** Current setting of the specified meter

Where: Z = M – Meter Value
= R – Rotational Value
= B – Blade Value
Y = S – Signature Value
= D – Deviation Value
Mx = Meter 1, 2, 3, 4, 5, or 6

or...

**"NOXX"** The transaction volume was not accepted.

**Remarks:** None.

**Constraints:** NO31 will be returned if the command format does not match the current configuration for the arm.

NO01 will be returned if another load arm is in the program mode locally or via communications.

**Special Case:** None.

**Comm. Modes:** Polling Only, Poll & Authorize or Remote Control.

## *Command Code TS – Turbine Meter Status*

This command returns the status of the turbine meter diagnostics.

**Command:**

| | |
|---|---|
| **"TS"** | Request turbine meter diagnostics state |
| **"TS_Mx"** | Request turbine meter diagnostics state for specified meter |

Where: Mx = Meter 1, 2, 3, 4, 5, or 6 (ratio, hybrid blending only)

**Responses:**

| | |
|---|---|
| **"TS_Z"** | **Good Response.** Current Turbine Meter Diagnostics State |
| **"TR_Z_Mx"** | **Good Response.** Current Turbine Meter Diagnostics State for the specified meter. |

Where: Z = E – Enabled
= D – Disabled
= L – Learn Mode
= A – Alarm
Mx = Meter 1, 2, 3, 4, 5, or 6

or...

| | |
|---|---|
| **"NOXX"** | The Diagnostic State was not reported. |

**Remarks:** None.

**Constraints:** NO31 will be returned if the command format does not match the current configuration for the arm.

**Special Case:** None.

**Comm. Modes:** Polling Only, Poll & Authorize, Remote Control, Poll & Program

## *Command Code TW – Write Turbine Meter Signature*

This command writes the signature set up for the turbine meter diagnostics.

**Command:**

| | |
|---|---|
| **"TW_Z_Y_N...N"** | Set signature |
| **"TW_Z_Y_Mx_N...N"** | Set signature for a specific meter |

Where: Z = M – Meter Value
= R – Rotational Value
= B – Blade Value
Y = S – Signature Value
= D – Deviation Value
Mx = Meter 1, 2, 3, 4, 5, or 6 (ratio, hybrid blending only)

**Responses:**

| | |
|---|---|
| **"OK"** | **Good Response.** Setting Changed |
| or... | |
| **"NOXX"** | The setting was not changed |

**Remarks:** None.

**Constraints:** NO31 will be returned if the command format does not match the current configuration for the arm.

**Special Case:** None.

**Comm. Modes:** Polling Only, Poll & Authorize, Remote Control, or Poll & Program.

## *Command Code VB – Batch Vapor Recovered Mass*

This command retrieves the vapor recovered mass for a batch.

**Command:**

| | |
|---|---|
| **"VB"** | Current batch recovered mass |
| **"VB_YY"** | Completed batch mass |
| **"VB_YY_NNN"** | Historical transaction request |

**Responses:**

| | |
|---|---|
| **"VB_MMMMMMM.MM"** | Current batch mass |
| **"VB_YY_MMMMMMM.MM"** | Completed batch mass |
| **"VB_YY_NNN_MMMMMMM.MM"** | Historic |

Where:

YY = Batch number (1-10)
NNN = Number of transactions back into local storage to retrieve data
MMMMMMM.MM = Batch mass

or

**"NOXX"**

**Remarks:** None.

**Constraints:** None.

**Special Case:** None.

**Comm Modes:** Polling Only, Poll & Program, Poll & Authorize, and Remote Control

## *Command Code VC – Control Valve Solenoid Actuation Diagnostic Counters*

This command retrieves the solenoid actuation counts from the AccuLoad III.

**Command:**

| | |
|---|---|
| **"VC"** | Read solenoid actuation counts (straight, sequential, unloading arms) |
| **"VC_Mx"** | Read solenoid actuation counts (specific meter's control valve on ratio/hybrid arm) |
| **"VC_Ix"** | Flow controlled injector |
| **"VC_S_VVVVVVVVV"** | Set/reset actuation count (straight, sequential, unloading arms) |
| **"VC_Mx_S_VVVVVVVVV"** | Set/reset actuation count (ratio, hybrid arms) |
| **"VC_Ix_S_VVVVVVV"** | Set/reset actuation count for flow controlled injector valve |

**Responses:**

| | |
|---|---|
| **"VC_ XXXXXXXXX_YYYYYYYYY"** | Good response. Solenoid counts (straight, sequential arms) |
| **"VC_Mx_XXXXXXXXX_YYYYYYYYY"** | Good response. Solenoid counts (straight, Hybrid arms) |
| **"VC_Ix_XXXXXXXX_YYYYYYYYY"** | Good response. Solenoid counts (flow controlled injector) |

**"OK"** Good response. Sets/resets actuation count (VC_S_VVV.V; VC_Mx_S_VVV.V)

Note: the solenoid actuation count values returned will be padded with leading zeroes.

Where:

X = "1" through "6" for meter 1 through 6 respectively
S = "U" for Upstream solenoid, "D" for Downstream solenoid
VVVVVVVVV = Count value (reset to 0 assumed if not specified)
XXXXXXXXX = Current value of upstream solenoid counter
YYYYYYYYY = Current value of downstream solenoid counter

**Constraints:** Not available with transaction in progress (NOXX).

**Special Case:** For injectors (Ix), must be a flow controlled type.

**Comm. Modes:** Polling only, poll and authorize, poll and program, remote control.

## *Command Code VD – Valve (Closure) Diagnostic*

This command retrieves the time it takes for the product flow control valve to completely close, the volume of product that has been delivered and the flow rate at the time after the STOP button (or remote stop) has been pressed. This command will also retrieve the same informartion if there is a loss of permissive resulting in valve closure or if a communication command is issued to stop the batch.

**Command:**

**"VD"** Capture latest valve diagnostic data for valve on meter 1 (straight, sequential, unloading arms).

**"VD_Mx"** Capture latest valve diagnostic data for specified valve (ratio/hybrid arms).

**"VD_R"** Reset valve diagnostic data on load arm (straight, sequential, unloading arms).

**"VD_Mx_R"** Reset valve diagnostic data for a specified valve on a load arm (ratio/hybrid arms).

Where:

X = Meter number on the load arm

**Response:**

**"VD_TT.T_AAA_FFFFFFF" Good Response.**

**"VD_Mx_TT.T_AAA_FFFFFFF" Good Response.**

**"OK"** Good Response. Valve diagnostic reset.

**"OK"** Specific valve diagnostic reset.

Where:

TT.T = Time in seconds that was required before zero flow occurred.
AAA = Amount of volume delivered after valve commanded to close.
FFFFFFF = Flow rate of product when batch was commanded to stop.

Or…

**"NOXX"** XX = "31" Invalid for current configuration

**Remarks:** To indicate to the host computer system that a new set of valve diagnostic data is available, the response from the EE command will be updated with a “New Valve Diagnostic Data Available” flag. The bit for the status condition will be set when the batch is stopped due to the conditions indicated above. When the batch is resumed or ended the flag will be cleared.

**Constraints:** NO31 will be returned if the command format does not match the current configuration of the arm.

NO31 will be returned if the arm is not configured for “Straight with VRS.”

**Special Cases:** None.

**Comm. Modes:** Polling only, Poll & Authorize, Remote Control, Poll & Program.

## *Command Code VT – Request Meter Totalizer Data from the AccuLoad III*

This command requests a totalizer value from the AccuLoad III.

**Command:**

**"VT_Z_Px"** Product Non-resettable totals. Valid for sequential and ration blending configurations.

**"VT_Aaa"** Additive non-resettable totals.

**"VT_Z_RR"** Recipe non-resettable totals. Valid for any configuration.

**"VT_Z_Y_Px"** Product starting or ending non-resettable total for current transaction.

**"VT_Z_Y_Px_NNN"** Product starting or ending non-resettable totals for previous transactions.

**"VT_V"** Non-resettable totals for the vapor recovered mass for the load arm (Straight with VRS)

Where:

| | |
|---|---|
| Aaa | = A01 (Additive 1) |
| | = A02 (Additive 2) |
| | = • : |
| | = • : |
| | = • : |
| | = A24 (Additive 24) |
| Z | = 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 |
| Px | = Product 1, 2, 3, 4, 5, or 6 |
| RR | = Recipe Number 01 – 50 |
| Y | = S for starting non-resettable total, E for ending non-resettable |
| NNN | = Number of transactions back |
| V | = Vapor recovered mass for the load arm |

If the AccuLoad is programmed for “dynamic leakage update” (System 336), the non-resettable totals will be updated with leakage volume while the AccuLoad is idle. The “VT Z Px” command will return the current value of the non-resettable totals which will include any leakage that has occurred since the end of the last transaction.

If the AccuLoad is programmed for “update non-resettables with leakage at transaction start” (System 336), the non-resettable totals will remain static while the AccuLoad is idle (i.e. leakage will not be added until start of next transaction). The “VT Z Px” command will return the value of the non-resettable totals captured at the end of the last transaction.

As always, if a transaction is in progress, the “VT Z Px” command will return the current value of the dynamically updated non-resettable totals.

The “VT Z S” command may be sent at any time during or after a transaction to retrieve the non-resettable totals captured at the start of the transaction.

If “VT X E” is sent while the AccuLoad is idle, the non-resettable totals captured at the end of the last transaction will be returned. This value will not include any leakage that may have occurred after the last transaction is ended.

If “VT Z E” is sent to retrieve the ending transaction non-resettable totals while a transaction is still in progress, and will be appended to the volume indicating the value returned may not be the final value.

**Responses:**

**"VT_Z_Px_VVVVVVVVV"** **Good Response.** Product Totals, blending configuration.

**"VT_Aaa_VVVVVV.VVV"** **Good Response.** Additive Totals

**"VT_Z_RR_VVVVVVVVV"** **Good Response.** Recipe Totals

**"VT_Z_Y_Px_VVVVVVVVV"** **Good Response.** Product starting or ending non-resettable total for current transaction.

**"VT_Z_RR_VVVVVVVVV_NNN"** **Good Response.** Product starting or ending non-resettable totals for previous transactions.

**"VT_V_VVVVVVVVV "** **Good Response.** Vapor recovered mass.

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

VVVVVVVVV = 9-digit totalizer volume
Aaa = A01 (Additive 1)
= A02 (Additive 2)
= A03 (Additive 3)
= • :
= • :
= A24 (Additive 24)
Px = Products 1, 2, 3, 4, 5, or 6
RR = Recipe Number 01 – 50
Y = S for starting non-resettable total
= E for ending non-resettable total
NNN = Number of transactions back

or...

**"NOXX"** Totals were not retrieved.

**Remarks:** Products, Recipes, and Additives must be allocated to the arm addressed.

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

NO31 will be returned if the arm is not configured for “Straight with VRS.”

**Special Cases:** None.

**Comm. Modes:** Polling only, Poll & Authorize, Remote Control, Poll & Program.

## *Command Code VX – Transaction Vapor Recovered Mass*

This command retrieves the vapor recovered mass for a transaction.

**Command:**

| | |
|---|---|
| **"VX"** | Current transaction recovered mass |
| **"VX _NNN"** | Historic transaction request |

**Responses:**

| | |
|---|---|
| **“VX_MMMMMMM.MM”** | Current transaction recovered mass |
| **“VX_ NNN_MMMMMMM.MM”** | Historic transaction request |

Where:
NNN = Number of transactions back into local storage to retrieve data
MMMMMMM.MM = batch mass

or

**“NOXX”**

**Remarks:** None.

**Constraints:** None.

**Special Case:** None.

**Comm Modes:** Polling Only, Poll & Program, Poll & Authorize, and Remote Control

## *Command Code WA – Write Second Line of Message to Appear on Display*

This command instructs the AccuLoad III to display the second line of a message, up to 30 characters in length. The message is displayed on the line below the message specified with a WD, WP, WQ or WX command. The format is identical to these commands; however the time out value is ignored. The delimiter character is not ignored; the last command set determines the expected input form.

**Command:**

**"WA_NNN_X...Xd99"**

| | | | |
|---|---|---|---|
| Where: | X...X | = | An alphanumeric character string of up to 30 characters (see Remarks). |
| | NNN | = | timeout value in seconds. An entry of 000 will cause the timeout value to default to what was entered for the first display message or if a non-zero entry is used, it will override the previous value entered. |
| | d | = | A required delimiter used to signify the input field. There are three different delimiters each indicating a unique handling of the input process. They are as follows: |
| | & | = | Input data is entered directly with no initiator required; any function key except the "CLEAR" or the "STOP" may be used to terminate the input sequence. |
| | [ | = | An "ENTER" key must initiate the input sequence; any function key except the "CLEAR" or "STOP" may be used to terminate the input sequence. |
| | ] | = | An "ENTER" key must initiate the input sequence; only an "ENTER" key will be accepted for terminating the input sequence. |

***Note:*** *The above delimiters are acceptable as part of the alphanumeric message.*

| | | |
|---|---|---|
| 99 | = | Represents the input field length. This may be encoded to allow a variable length input field up to the length specified here. The range of the input length is 00 to 20. To encode the option of the variable length input, add 40 to the desired input field length. |

**EXAMPLE:** To request an input field length of 5, enter "05." To also include the option, enter "45." If the option is not included the operator will have to put in 5 digits and then press the terminating key, any more than 5 digits and the message "Error Press Clear" will appear on the AccuLoad III display. If the option is included, the terminating key can be pressed at any time sending data to the Host.

**Responses:**

**"OK"** **Good Response.** Message has been accepted for the alphanumeric display (left justified).

or...

**"NOXX"** The message was not accepted.

**Remarks:** The following characters can be displayed on the AccuLoad III display. (X...X part of command).

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
a b c d e f g h i j k l m n o p q r s t u v w x y z
0 1 2 3 4 5 6 7 8 9
+ ! " # $ % & ' ( ) - . / : ; < = > @ [ \ ^ _ ' {|} ° ? * space

The message will be displayed on the third line of the display and will wrap to the fourth line where necessary.

**Constraints:** "WD" or "WX" must be issued first. Input field length must be zero if one was previously defined in the first message issued. If it is not, a "NO06" will be returned. "WA" cannot be used with delayed prompts "WP" or "WQ."

NO01 will be returned if another load arm is in the program mode locally or via communications.

**Special Case:** None.

**Comm. Modes:** Poll & Authorize or Remote Control.

## *Command Code WB – Write Third Line of Message to Appear on Display*

The WB command allows writing a 30-character message which is displayed two lines below the message specified with a WD, WP, WQ, or WX command. The format is identical to these commands; however, the timeout value is ignored. The delimiter character is not ignored. The last command set determines the expected input form.

The commands that write to lines other than the first (WA, WB, WC) must be preceded by one of the prompt commands that write to the first line (WD, WP, WQ, WX).

**Command:**

**"WB_NNN_X...Xd99"**

Where:

| | | |
|---|---|---|
| X...X | = | An alphanumeric character string of up to 30 characters |
| NNN | = | Prompt timeout value (001-999) |
| d | = | A required delimiter used to signify the input field. There are three different delimiters each indicating a unique handling of the input process. They are as follows: |
| & | = | Input data is entered directly with no initiator required; any function key except the "CLEAR" or the "STOP" may be used to terminate the input sequence. |
| [ | = | An "ENTER" key must initiate the input sequence; any function key except the "CLEAR" or "STOP" may be used to terminate the input sequence. |
| ] | = | An "ENTER" key must initiate the input sequence; only an "ENTER" key will be accepted for terminating the input sequence. |

***Note:*** *The above delimiters are acceptable as part of the alphanumeric message.*

| | | |
|---|---|---|
| 99 | = | Represents the input field length. This may be encoded to allow a variable length input field up to the length specified here. The range of the input length is 00 to 20. To encode the option of the variable length input, add 40 to the desired input field length. |

**EXAMPLE:** To request an input field length of 5, enter "05." To also include the option, enter "45." If the option is not included the operator will have to put in 5 digits and then press the terminating key, any more than 5 digits and the message "Error Press Clear" will appear on the AccuLoad III display. If the option is included, the terminating key can be pressed at any time sending data to the Host.

**Responses:**

**"OK"** **Good Response.** Message has been accepted for the alphanumeric display (left justified).

or...

**"NOXX"** The message was not accepted.

**Remarks:** The following characters can be displayed on the AccuLoad III display. (X...X part of command).

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
a b c d e f g h i j k l m n o p q r s t u v w x y z
0 1 2 3 4 5 6 7 8 9
+ ! " # $ % & ' ( ) - . / : ; < = > @ [ \ ^ _ ' {|} ° ? * space

The message will be displayed on the third line of the display and will wrap to the fourth line where necessary.

**Constraints:** "WD" or "WX" must be issued first. Input field length must be zero if one was previously defined in the first message issued. If it is not, a "NO06" will be returned. "WB" cannot be used with delayed prompts "WP" or "WQ."

NO01 will be returned if another load arm is in the program mode locally or via communications.

**Special Case:** None.

**Comm. Modes:** Poll & Authorize or Remote Control.

## *Command Code WC – Write Fourth Line of Message to Appear on Display*

The WC command allows writing a 30-character message which is displayed three lines below the message specified with a WD, WP, WQ, or WX command. The format is identical to these commands; however, the timeout value is ignored. The delimiter character is not ignored. The last command set determines the expected input form.

The commands that write to lines other than the first (WA, WB, WC) must be preceded by one of the prompt commands that write to the first line (WD, WP, WQ, WX).

**Command:**

**"WC_NNN_X...Xd99"**

| | | | |
|---|---|---|---|
| Where: | X...X | = | An alphanumeric character string of up to 30 characters |
| | NNN | = | Prompt timeout value (001-999) |
| | d | = | A required delimiter used to signify the input field. There are three different delimiters each indicating a unique handling of the input process. They are as follows: |
| | & | = | Input data is entered directly with no initiator required; any function key except the "CLEAR" or the "STOP" may be used to terminate the input sequence. |
| | [ | = | An "ENTER" key must initiate the input sequence; any function key except the "CLEAR" or "STOP" may be used to terminate the input sequence. |
| | ] | = | An "ENTER" key must initiate the input sequence; only an "ENTER" key will be accepted for terminating the input sequence. |

***Note:*** *The above delimiters are acceptable as part of the alphanumeric message.*

| | | |
|---|---|---|
| 99 | = | Represents the input field length. This may be encoded to allow a variable length input field up to the length specified here. The range of the input length is 00 to 20. To encode the option of the variable length input, add 40 to the desired input field length. |

**EXAMPLE:** To request an input field length of 5, enter "05." To also include the option, enter "45." If the option is not included the operator will have to put in 5 digits and then press the terminating key, any more than 5 digits and the message "Error Press Clear" will appear on the AccuLoad III display. If the option is included, the terminating key can be pressed at any time sending data to the Host.

**Responses:**

**"OK"** **Good Response.** Message has been accepted for the alphanumeric display (left justified).

or...

**"NOXX"** The message was not accepted.

**Remarks:** The following characters can be displayed on the AccuLoad III display. (X...X part of command).

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
a b c d e f g h i j k l m n o p q r s t u v w x y z
0 1 2 3 4 5 6 7 8 9
+ ! " # $ % & ' ( ) - . / : ; < = > @ [ \ ^ _ ' {|} ° ? * space

The message will be displayed on the third line of the display and will wrap to the fourth line where necessary.

**Constraints:** "WD" or "WX" must be issued first. Input field length must be zero if one was previously defined in the first message issued. If it is not, a "NO06" will be returned. "WC" cannot be used with delayed prompts "WP" or "WQ."

NO01 will be returned if another load arm is in the program mode locally or via communications.

**Special Case:** None.

**Comm. Modes:** Poll & Authorize or Remote Control.

## *Command Code WD – Write to Display*

This command instructs the AccuLoad III to display a 30-character message. The time displayed and length of response are also specified in the command. Data will be echoed to the screen upon input by the operator. This command may be used alone or it may be used in conjunction with any combination of the WA, WB or WC commands.

**Command:**

**"WD_NNN_X...Xd99"**

Where: X...X = An alphanumeric character string of up to 30 characters (see Remarks).

NNN = time-out value in seconds. An entry of 000 will cause the time-out value to default to what was entered for the first display message or if a non-zero entry is used, it will override the previous value entered.

d = A required delimiter used to signify the input field. There are three different delimiters each indicating a unique handling of the input process. They are as follows:

& = Input data is entered directly with no initiator required; any function key except the "CLEAR" or the "STOP" may be used to terminate the input sequence.

[ = An "ENTER" key must initiate the input sequence; any function key except the "CLEAR" or "STOP" may be used to terminate the input sequence.

] = An "ENTER" key must initiate the input sequence; only an "ENTER" key will be accepted for terminating the input sequence.

***Note:*** *The above delimiters are acceptable as part of the alphanumeric message.*

99 = Represents the input field length. This may be encoded to allow a variable length input field up to the length specified here. The range of the input length is 00 to 20. To encode the option of the variable length input, add 40 to the desired input field length.

**EXAMPLE:** To request an input field length of 5, enter "05." To also include the option, enter "45." If the option is not included the operator will have to put in 5 digits and then press the terminating key, any more than 5 digits and the message "Error Press Clear" will appear on the AccuLoad III display. If the option is included, the terminating key can be pressed at any time sending data to the Host.

**Responses:**

**"OK"** **Good Response.** Message has been accepted for alphanumeric display, left justified.

or...

**"NOXX"** The message was not accepted.

**Remarks:** The following characters can be displayed on the AccuLoad III display. (X...X part of command).

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
a b c d e f g h i j k l m n o p q r s t u v w x y z
0 1 2 3 4 5 6 7 8 9
+ ! " # $ % & ' ( ) - . / : ; < = > @ [ \ ^ _ ' {|} ° ? * space

The message will wrap to the second line of the display where necessary.

**Constraints:** NO01 will be returned another load arm is in the program mode locally or via communications.

**Special Case:** Keypad data pending flag and data are cleared.

**Comm. Modes:** Poll & Authorize or Remote Control.

## *Command Code WP – Write Delayed Prompt with Echo*

This command sends a delayed prompt message to the AccuLoad III. This message will be displayed when the "SET" key is pressed enabling the keypad for input. The time displayed and length of response are also specified in the command. Data will be echoed to the screen upon input by the operator. This command may be used alone or it may be used in conjunction with any combination of the WA, WB or WC commands.

**Command:**

**"WP_NNN_X...Xd99"**

Where: X...X = An alphanumeric character string of up to 30 characters (see Remarks).

NNN = time-out value in seconds. An entry of 000 will cause the time-out value to default to what was entered for the first display message or if a non-zero entry is used, it will override the previous value entered.

d = A required delimiter used to signify the input field. There are three different delimiters each indicating a unique handling of the input process. They are as follows:

& = Input data is entered directly with no initiator required; any function key except the "CLEAR" or the "STOP" may be used to terminate the input sequence.

[ = An "ENTER" key must initiate the input sequence; any function key except the "CLEAR" or "STOP" may be used to terminate the input sequence.

] = An "ENTER" key must initiate the input sequence; only an "ENTER" key will be accepted for terminating the input sequence.

***Note:*** *The above delimiters are acceptable as part of the alphanumeric message.*

99 = Represents the input field length. This may be encoded to allow a variable length input field up to the length specified here. The range of the input length is 00 to 20. To encode the option of the variable length input, add 40 to the desired input field length.

**EXAMPLE:** To request an input field length of 5, enter "05." To also include the option, enter "45." If the option is not included the operator will have to put in 5 digits and then press the terminating key, any more than 5 digits and the message "Error Press Clear" will appear on the AccuLoad III display. If the option is included, the terminating key can be pressed at any time sending data to the Host.

**Responses:**

| | |
|---|---|
| **"OK"** | **Good Response.** Message has been accepted and will be displayed when the "SET" key is pressed at the start of a transaction only. Sets keypad pending status when keypad input is completed. Timer is not started until the "SET" key is pressed. |
| or... | |
| **"NOXX"** | The message was not accepted. |

**Remarks:** The following characters can be displayed on the AccuLoad III display. (X...X part of command).

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
a b c d e f g h i j k l m n o p q r s t u v w x y z
0 1 2 3 4 5 6 7 8 9
+ ! " # $ % & ' ( ) - . / : ; < = > @ [ \ ^ _ ' {|} ° ? * space

The prompt will wrap to the second line of the display where necessary.

**Constraints:** Cancellation occurs on receiving a "DA" command or on power-down.

NO01 will be returned if another load arm is in the program mode locally or via communications

**Special Case:** The "DP" status bit will be set on acceptance of the "WP" command and cleared on cancellation (issuance of DA). The "TO" status bit will be cleared on acceptance of a DA.

**Comm. Modes:** Poll & Authorize or Remote Control.

## *Command Code WQ – Write Delayed Prompt with Security Echo*

This command sends a delayed prompt message to the AccuLoad III. This message is displayed when the SET key is pressed enabling the keypad for input with security echo ("X"). The time displayed and length of response are also specified in the command. Data will be echoed to the screen upon input by the operator. This command may be used alone or it may be used in conjunction with any combination of the WA, WB or WC commands

**Command:**

**"WQ_NNN_X...Xd99"**

Where: X...X = An alphanumeric character string of up to 30 characters (see Remarks).

NNN = time-out value in seconds. An entry of 000 will cause the time-out value to default to what was entered for the first display message or if a non-zero entry is used, it will override the previous value entered.

d = A required delimiter used to signify the input field. There are three different delimiters each indicating a unique handling of the input process. They are as follows:

& = Input data is entered directly with no initiator required; any function key except the "CLEAR" or the "STOP" may be used to terminate the input sequence.

[ = An "ENTER" key must initiate the input sequence; any function key except the "CLEAR" or "STOP" may be used to terminate the input sequence.

] = An "ENTER" key must initiate the input sequence; only an "ENTER" key will be accepted for terminating the input sequence.

***Note:*** *The above delimiters are acceptable as part of the alphanumeric message.*

99 = Represents the input field length. This may be encoded to allow a variable length input field up to the length specified here. The range of the input length is 00 to 20. To encode the option of the variable length input, add 40 to the desired input field length.

**EXAMPLE:** To request an input field length of 5, enter "05." To also include the option, enter "45." If the option is not included the operator will have to put in 5 digits and then press the terminating key, any more than 5 digits and the message "Error Press Clear" will appear on the AccuLoad III display. If the option is included, the terminating key can be pressed at any time sending data to the Host.

**Responses:**

**"OK"** **Good Response.** Message has been accepted and will be displayed when the "SET" key is pressed at the start of a transaction only. Sets keypad pending status once keypad input is completed. Timer is not started until the "SET" key is pressed.

or...

**"NOXX"** The message was not received.

**Remarks:** The following characters can be displayed on the AccuLoad III display. (X...X part of command).

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
a b c d e f g h i j k l m n o p q r s t u v w x y z
0 1 2 3 4 5 6 7 8 9
+ ! " # $ % & ' ( ) - . / : ; < = > @ [ \ ^ _ ' {|} ° ? * space

The prompt will wrap to the second line of the display where necessary.

**Constraints:** Cancellation occurs on receiving a "DA" command or on power-down.

NO01 will be returned if another load arm is in the program mode locally or via communications.

**Special Case:** The "TO" status bit will be cleared on acceptance of a DA command. The "DP" status bit will be set on acceptance of the "WQ" command and cleared on cancellation (issuance of a DA).

**Comm. Modes:** Poll & Authorize or Remote Control.

## *Command Code WX – Write to the Display in Security Mode*

This command sends a message to the AccuLoad III and enables the keypad for input in an echo "X" or Security Mode. The time displayed and length of response are also specified in the command. Data will be echoed to the screen upon input by the operator. This command may be used alone or it may be used in conjunction with any combination of the WA, WB or WC commands.

**Command:**

**"WX_NNN_X...Xd99"**

Where: X...X = An alphanumeric character string of up to 30 characters (see Remarks).

NNN = time-out value in seconds. An entry of 000 will cause the time-out value to default to what was entered for the first display message or if a non-zero entry is used, it will override the previous value entered.

d = A required delimiter used to signify the input field. There are three different delimiters each indicating a unique handling of the input process. They are as follows:

& = Input data is entered directly with no initiator required; any function key except the "CLEAR" or the "STOP" may be used to terminate the input sequence.

[ = An "ENTER" key must initiate the input sequence; any function key except the "CLEAR" or "STOP" may be used to terminate the input sequence.

] = An "ENTER" key must initiate the input sequence; only an "ENTER" key will be accepted for terminating the input sequence.

***Note:*** *The above delimiters are acceptable as part of the alphanumeric message.*

99 = Represents the input field length. This may be encoded to allow a variable length input field up to the length specified here. The range of the input length is 00 to 20. To encode the option of the variable length input, add 40 to the desired input field length.

**EXAMPLE:** To request an input field length of 5, enter "05." To also include the option, enter "45." If the option is not included the operator will have to put in 5 digits and then press the terminating key, any more than 5 digits and the message "Error Press Clear" will appear on the AccuLoad III display. If the option is included, the terminating key can be pressed at any time sending data to the Host.

**Responses:**

**"OK"** **Good Response.** Message has been accepted for alphanumeric display, left justified. Keypad enabled for input in Echo 'X' Mode.

or...

**"NOXX"** Message was not accepted.

**Remarks:** The following characters can be displayed on the AccuLoad III display. (X...X part of command).

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
a b c d e f g h i j k l m n o p q r s t u v w x y z
0 1 2 3 4 5 6 7 8 9
+ ! " # $ % & ' ( ) - . / : ; < = > @ [ \ ^ _ ' {|} ° ? * space

The prompt will wrap to the second line of the display where necessary.

**Constraints:** NO01 will be returned if another load arm is in the program mode locally or via communications.

**Special Case:** The "TO" status bit will be cleared on acceptance of a "WD," "WX," "WP" or "WQ" command. Also, keypad data pending flag and data are cleared.

**Comm. Modes:** Poll & Authorize or Remote Control.

## *Command Code XC – Change Parameter Security Level*

This command instructs the AccuLoad III to modify the security level of a program mode parameter.

**Command:**

**"XC_XX_YYY...Z"**

Where: XX = Program mode major directory
- CF = Configuration
- SY = System
- AR = Arm
- Mx = Meter (1-6)
- Px = Product (1-6)
- 01…50 = Recipe

YYY = Parameter number

Z = Security Level (1-5)

**Responses:**

**"XC_XX_YYY_Z_A..A"** Message was accepted.

Where: XX = Program mode major directory
- CF = Configuration
- SY = System
- AR = Arm
- Mx = Meter (1-6)
- Px = Product (1-6)
- 01…50 = Recipe

YYY = Parameter number

Z = Security Level (1-5)

A..A = The programmed value

or...

**"NOXX"**

**Remarks:** None

**Constraints:** Security level for parameters in the security directory cannot be modified (pass codes etc.).

**Special Case:** None

**Comm. Modes:** Poll/Authorize, Remote Control, Poll/Program

## *Command Code XV – Read Parameter Security Level*

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

**Command:**

**"XV_XX_YYY "**

Where: XX = Program mode major directory
- CF = Configuration
- SY = System
- AR = Arm
- Mx = Meter (1-6)
- Px = Product (1-6)
- 01…50 = Recipe

YYY = Parameter number

**Responses:**

**"XV_XX_YYY_Z_A..A"** Message was accepted.

Where: XX = Program mode major directory
- CF = Configuration
- SY = System
- AR = Arm
- Mx = Meter (1-6)
- Px = Product (1-6)
- 01…50 = Recipe

YYY = Parameter number

Z = Currently programmed security level

A..A = The programmed value

or...

**"NOXX"** Message was not accepted.

**Remarks:** None

**Constraints:** Security level for parameters in the security directory cannot be modified (pass codes etc.).

**Special Case:** None

**Comm. Modes:** Poll/Authorize, Remote Control, Poll/Program

## *Appendix I – Reference for "NOXX" Responses*

| XX | Description |
|---|---|
| 00 | Command Nonexistent |
| 01 | In Program Mode |
| 02 | Released |
| 03 | Value Rejected |
| 04 | Flow Active |
| 05 | No Transaction Ever Done |
| 06 | Operation Not Allowed |
| 07 | Wrong Control Mode |
| 08 | Transaction In Progress |
| 09 | Alarm Condition |
| 10 | Storage Full |
| 11 | Operation Out Of Sequence |
| 12 | Power Fail During Transaction |
| 13 | Authorized |
| 14 | Program Code Not Used |
| 15 | Display/Keypad In Use |
| 16 | Ticket Not In Printer |
| 17 | No Keypad Data Pending |
| 18 | No Transaction In Progress |
| 19 | Option Not Installed |
| 20 | Start After Stop Delay |
| 21 | Permissive Delay Active |
| 22 | Print Request Pending |
| 23 | No Meter Enabled |
| 24 | Must Be In Program Mode |
| 25 | Ticket Alarm During Transaction |
| 26 | Volume Type Not Selected |
| 27 | Exactly One Recipe Must Be Enabled |
| 28 | Batch Limit Reached |
| 29 | Checking Entries |
| 30 | Product/Recipe/Additive Not Assigned |
| 31 | Invalid Argument for Configuration |
| 32 | No Key Ever Pressed |

| XX | Description |
|---|---|
| 33 | Maximum Active Arms in Use |
| 34 | Transaction Not Standby |
| 35 | Comm Swing Arm Out of Position |
| 36 | Card-In Required |
| 37 | Data Not Available |
| 38 | Too Many Shared Additives Selected |
| 39 | No Current Batch on This Arm |
| 40 | Invalid on Virtual Arm |
| 41 | No Pending Reports |
| 90 | Must Use Mini Protocol |
| 91 | Buffer Error |
| 92 | Keypad Locked |
| 93 | Data Recall Error |
| 94 | Not In Program Mode |
| 95 | Security Access Not Available |
| 96 | Data Request Queued Ask Later |
| 97 | Comflash Archiving |
| 99 | Internal Error |

## *Appendix II – Alphanumeric Character Set Used By the AccuLoad III*

The following characters are translated by the AccuLoad to display special characters not found on a typical keyboard: the tilde (~) will display as a degree sign at the AccuLoad; degree signs sent by the AccuLoad 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 AccuLoad, such as the product name, etc. The comma (,) may not be used within any prompt (WD, WA, 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 |

| ASCII | DECIMAL | HEX |
|---|---|---|
| 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 |
| @ | 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 |

| ASCII | DECIMAL | HEX |
|---|---|---|
| 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 |
| 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 |

| ASCII | DECIMAL | HEX |
|---|---|---|
| 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 AccuLoad 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 AccuLoad 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 AccuLoad. 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 "E," so the second character of the AB command will be E. 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 5E0000."

### *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 AccuLoad III Authorized and AccuLoad III 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 AccuLoad III.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'.

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 AccuLoad III.net from outside do not require any value to be programmed here; this entry is only used for initiating a connection from the AccuLoad III.net to a point outside the local network.

### *Using the Smith Meter protocol over TCP/IP*

The AccuLoad III.net has a reserved port that supports Smith Meter Terminal communications. This protocol is currently fixed at port 7734. The AccuLoad III.net expects a single complete command to be contained in each packet received. The AccuLoad III.net will currently ignore fragmented commands or any additional commands after the first in a packet. Due to the relatively small size of the Smith 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 AccuLoad III.net do not have complete commands and are ignored. The AccuMate has a built in tool - the Terminal Emulator - that sends an entire command in one packet. Custom software written to communicate with the AccuLoad.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 AccuLoad III.net information*

By pointing your Web browser at the IP address of the AccuLoad III.net, you can retrieve various Web pages from the device showing the current state of operation, etc. For example, assuming a AccuLoad III.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 Compute*" 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. "AccuLoad III.net 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 checkboxes 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 microLoad.net (i.e. if your AccuLoad III.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. "AccuLoad III.net 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 microLoad.net (i.e. if your AccuLoad III.net is 192.168.0.1, make the address for the SLIP client 192.168.0.9 or similar.

## *Appendix VII – Emulation of AccuLoad II Communications Using AccuLoad III-X*

An enhancement to AccuLoad III-X in Revision 8 enables an AccuLoad III-X to emulate the communications interface of an AccuLoad II. AccuLoad III-X now supports AccuLoad II firmware versions STD, SEQ, SQR, and RBM. This appendix outlines some requirements and guidelines for configuring and using the AccuLoad III-X with an automation system designed for the AccuLoad II.

### *Configuring AccuLoad II Style Communications*

New options have been added to the Communication Function Program Codes (System 707, 712, 717, and 722) to support emulation of the AccuLoad II communications interface. A Terminal Mode and a Minicomputer Mode version are available for each of the supported firmware packages. Select the appropriate communications protocol you wish to emulate and exit program mode to initiate emulation.

To emulate an AccuLoad II with the highest degree of compatibility with existing automation, care is required to configure the AccuLoad III so that I/O, products, recipes, etc. are correctly represented via the AccuLoad II protocol emulation. Some care has been taken to assure that implementation of the emulations provides the highest degree of compatibility with existing communications interfaces, while at the same time not artificially limiting the available functions. For example, the AccuLoad III contains 50 recipe slots; the AccuLoad II SEQ contains 24. If the emulation forced all 24 recipes to be available, it would only be possible to emulate two SEQ arms with the AccuLoad III-X firmware. Instead the software allows but does not force this full emulation. It also allows the scenario where all six arms could be configured as sequential blending arms, and each given a portion of the pool of 50 recipes. The AccuLoad III, when handling recipe data via the AccuLoad II protocol, calls the first recipe assigned to a particular arm recipe 1.

Generally, only features available in the emulated firmware package will be available when emulating that firmware in the AccuLoad III. For example, although the AccuLoad III supports density transducers on ratio blending arms, the AccuLoad II RBM protocol will not allow you to retrieve GSV totals since there was no live density input available on the RBM. There will be nothing preventing the assignment and use of the AccuLoad II feature, just no way of accessing that feature via the AccuLoad II protocol.

### *Configuring Load Arms*

When using the AccuLoad II communications emulation, there are new considerations when deciding how to allocate arm assignments in the AccuLoad III. One consideration is that all arms should be configured to match the protocol assigned for that box. If you have a load rack that contains STD and RBM units, for example, any AccuLoad IIIs installed on this rack should emulate either STD or RBM, and arms in that AccuLoad III should be configured either as all straight product arms or all ratio blending arms to match the emulation selected. While it may be possible to use an AccuLoad II protocol to communicate with arms configured differently than expected (i.e., use AccuLoad II STD protocol on an AccuLoad III with sequential blending arms configured), this may result in critical data not being available to the host and is not recommended. It is also possible to configure a straight-product arm in a box emulating an RBM, as it was in the original RBM firmware.

A second consideration is the number of recipes and additives available per arm. The AccuLoad III has a limit of 50 recipes and 24 additives per box. If you require all seven additives for your SEQ emulation then you would not want to configure more than three arms per box, for example. See the section *Configuring Additives, Recipes and Products* for more detailed information.

Another consideration is that full AccuLoad II emulation may be more I/O bound than the equivalent AccuLoad III arm, especially when emulating an AccuLoad II STD. This may result in fewer arms being configured for each AccuLoad III so that each can have a full complement of I/O. Since the STD had fixed I/O, to fully emulate the status responses requires at least that number of I/O to be configured for each arm. This may result in fewer arms being configured for each AccuLoad III so that each can have a full complement of I/O. If the input/output states are of no concern to the automation system, this factor can be ignored. See the following section, *Configuring I/O*, for more information.

### *Configuring I/O*

There was some difficulty accounting for all the differences in the way the I/O is handled in the AccuLoad II. Again, the approach taken allows for full compatibility where it is required and flexibility when it is not. For the most part, the AccuLoad II blending packages had configurable I/O, and so they are fairly easy to emulate on an AccuLoad III which also has configurable I/O. The AccuLoad II STD has a fixed I/O map in that input 1 was always function x, and input 2 was always function y, etc. This in itself is not a problem, but since the status of some of these inputs/outputs is reported by function rather than I/O point, care is required in configuring the I/O if you are using the input status bits in the EQ/RS responses to determine I/O state via the STD protocol. Again, the I/O is mapped to a particular arm. As in the recipe example above, the first input programmed for a particular arm is I1. Generally the $n^{th}$ input's state is reported in I*n.*

Order of mapped input functions for AccuLoad II STD:

| Order Programmed (by arm) | Input state reported |
|---|---|
| First | TS |
| Second | HC |
| Third | VS |
| Fourth | S1 |
| Fifth | VP |
| Sixth | S2 |

Some of these functions do not exist in the AccuLoad III. To emulate these functions, inputs and outputs configured as general-purpose now can be assigned to a particular arm. This only affects AccuLoad II emulation, and provides a way to map unused functions to an I/O point. For example, if your automation system is checking for the Valve Power input (VP) to be asserted, then you must configure at least five inputs for each arm, program the fifth as a general-purpose input, and jumper the input high so that it is always asserted. This will result in the VP status appearing in the EQ and RS responses. Program the first input for each STD arm as a ticket-tray switch if one is in use; otherwise, program a dummy general-purpose input as a placeholder. Program the second input for each STD arm for a first/second high flow contact if one is in use; and so on down the table until you have no need for any of the remaining states. (Note: There is no emulation of VS; if you are using a valve stem switch, bring this into the third general-purpose input programmed for the arm.)

Inputs do not have to be group programmed by arm at the AccuLoad III. They may be configured in any order desired, other than the considerations for STD above. Even though the inputs are not programmed contiguously by arm, the AccuLoad II protocols will treat the I/O for each arm as a set.

### *Configuring Additives, Recipes and Products*

Viewed from the AccuLoad II interface, each arm has a contiguous block of I/O and additives (and recipes, for the blending units) up to the limit for the particular AccuLoad II firmware being emulated. Again, it does not matter where in the AccuLoad III a particular additive is assigned. The $n^{th}$ additive assigned to arm 1 will be accessed when data for additive N is requested. For example, assume the following configuration:

| AccuLoad III Additive Position | Arm Assigned |
|---|---|
| 1 | 1 |
| 2 | 1 |
| 3 | 2 |
| 4 | 2 |
| 5 | 3 |
| 6 | 3 |
| 7 | 1 |
| 8 | 2 |
| 9 | 3 |

A request for the additive 3 non-resettable total on arm 2, for example, would return the non-resettable total for Additive 8 in the AccuLoad III. A request for an additive beyond the number assigned to that arm will result in a NO response.

Recipes are handled in the same manner. Requests for additives or recipes beyond the maximum for the firmware being emulated will always be rejected with a NO03 response. If the AccuLoad II STD is being emulated, all data returned is for Product 1. For sequential blending arms, a maximum of four products is supported by the AccuLoad II SEQ or SQR protocols. The product numbers map directly (P1 = Product 1, P2 = Product 2, etc.). For RBM emulation, an additional program code was added for each arm (System 761-766) to allow the AccuLoad II to emulate either the left side or the right side of an AccuLoad II RBM. This program code was required because product 1 and product 2 in the RBM must be on opposite sides if both arms are used. These program codes map the products to be used for the arm. It is possible to configure all RBM arms as left side arms, or all as right-side arms, there is no significance other than what product arguments are used to access the data via the AccuLoad II protocols. Example: In System 762, Arm 2 is mapped like the right side of an RBM, with products 2 and 4 selected. Internal to the AccuLoad III product 1 and product 2 will be used. AccuLoad II automation would be required to request P2 and P4, however.

### *Unsupported Commands*

The following AccuLoad II communications commands are not supported in the AccuLoad II emulations in AccuLoad III-X, Revision 8:

CD CF CI CL CR DB DS EI FR PT RI

These unsupported commands relate primarily to three areas of functionality: Special input/output relay functions in RBM (CI, EI, FR, RI); program code change mapping / NIST counter reporting (CD, CF, CL, DB, PB); and commands with multiple-line responses (CR, DS, and PT).

### *Limited PV/PC Support*

AccuLoad II PV and PC commands have limited support in the AccuLoad III. Briefly, the following limitations exist:

PV/PC commands to the Configuration directory (RBM) are not supported.
PV/PC commands to run/ready alarm clearing parameters (System 142/146 in RBM) are not supported.
PV/PC commands to configurable prompts (System 185 in RBM) are not supported.
PV/PC commands to configurable report parameters (System 734/759 in RBM) are not supported.
PV/PC commands to S-Mass specific parameters (STD 457-467) are not supported.

“NO14 - Program Code Not Used” will be returned for the above program code entries.

In addition, all parameters will be considered "PC Ready Only". No PC commands to arms with transactions in progress will be allowed. 'NO08 - Transaction in Progress' will be returned.

### *Other Considerations / Exceptions*

Every attempt has been made to return exception (NOXX) responses as closely to the original product as possible. Also, the priority of error responses was maintained as closely as practical. However, the emulation may return a different error code under some conditions, especially when multiple error conditions are present.

Features unique to AccuLoad II that are not supported include the following:

- No S-Mass communications support
- No additive stop option at first trip (stop by programmed volume only)
- No high-speed prover output
- There is no “Feedback only” additive feature in AccuLoad III. Attempting to select this option will result in a “NO03 - Value Rejected” response
- The number of standby mode transactions that can be stored may differ from that specified for the AccuLoad II being emulated. In most cases, more transactions will be available in the AccuLoad III.

Other considerations:

- The AccuLoad II configurable alarm action functionality, while implemented in the emulation, can only be programmed via communications
- Only nine inputs on the AccuLoad III are AC, compared to the twelve AC inputs that are available on the AccuLoad II. All other digital inputs on the AccuLoad III are DC
- Not all AccuLoad II alarms exist on the AccuLoad III, and vice versa. AccuLoad III alarms that do not have an equivalent AccuLoad II alarm will be mapped to the most nearly related alarm, or the DA alarm if no similar alarms exist. AccuLoad II alarms with no AccuLoad III equivalent will never be generated.

**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 subroutines 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.

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

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

**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.

**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.

**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.

**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.

**Reperforator:** 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.

**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.

**AccuMate for AccuLoad III**

Specification ..... Bulletin SS06038
Installation/Operation ..... Bulletin MN06136

**AccuLoad III**

Specification ..... Bulletin SS06036
Installation/Operation ..... Bulletin MN06135
Operator Reference ..... Bulletin MN06129
Communications ..... Bulletin MN06130L
Modbus Communications ..... Bulletin MN06131L

Revisions included in MN06130L Issue/Rev. 1.4 (2/13):

Page iv (TOC):
Page 41: Added Code PM (Promass) under “Alarm Status Codes…”
Page 50: CD_V added, revised response codes
Page 74: Added Index Number 29-33.
Page 80: Revised #1 & #4 Response Codes.
Page 90: Changed System Alert Request Reserved(column 3) to Sening Comm & Reserved (column 4) to Network Printer.
Page 91: Added Response to Command Code EA – Character A11 (System) table.
Page 175: Added Future Use for Response Codes Note at bottom of page.
Page 182: Revised Response Note: Changed A18 to A19
Page 184: Added Character A18 Table.
Page 204-205: Revised Command Code FL
Pages 246 and 257: NC & VT added to Valid Status Codes.

# Technical Support

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

The specifications contained herein are subject to change without notice and any user of said specifications should verify from the manufacturer that the specifications are currently in effect. Otherwise, the manufacturer assumes no responsibility for the use of specifications which may have been changed and are no longer in effect.

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

© Guidant Corporation 2025 All rights reserved.

www.GuidantMeasurement.com