# Gilbarco Dispenser Two-Wire Protocol for Third Party Pump Controllers

**TWOTP-IS-IS2.26-P**
**Document ID: 4280**

REVISION
**2.26**

Issued to ZAO "NPF "SIBNEFTEKART"

**by**

**The Global Two-Wire Review Committee**

**Issued: September 2011**

Gilbarco Veeder-Root provides the material in this document “as is”, without warranty of any kind, either expressed or implied. Gilbarco may make changes and/or improvements in the program(s) described in this document at any time, without notice.

This product could contain technical inaccuracies or typographical errors. Changes are routinely made to the information herein which will be incorporated into new editions of the publication.

The techniques described in this document are the property of Gilbarco Veeder-Root. The last character in the document number printed on the title page of this document reflects the sensitivity of the material contained here, and indicates, as explained below, the handling required:

- If the document number ends in ‘S’ (as in T12G-PT-V20.0-S):

  This PROPRIETARY, SPECIAL HANDLING document contains extremely sensitive material. *It may not be duplicated or distributed either inside or outside Gilbarco without written approval by an authorized executive.*

- If the document number ends in ‘P’ (as in TCRG-DF-V43.1-P):

  The PROPRIETARY document may be duplicated as needed for general use *within Gilbarco, by Gilbarco employees*. All distributions should be reported to Software and Documentation Control so future revisions can be distributed to all those holding copies of the document.

  For distribution *outside Gilbarco*, this should be handled like a PROPRIETARY, SPECIAL HANDLING document (see above).

- If the document number ends in ‘M’ (as in TCRS-FS-V10.0-M):

  This non-confidential MANUAL may be duplicated *as needed* for use *either inside or outside Gilbarco*. All distributions should be reported to Software and Documentation Control so any future revisions can be distributed to all those holding copies of the document.

# TABLE OF CONTENTS

# 1. CHANGE HISTORY

| Revision | Date | Author | Changes |
|---|---|---|---|
| 1.0 | 01 DEC 1992 | Committee | • Initial Release |
| 2.0 | 01 JUL 2008 | Oldham | • Change to more readable format<br>• Add in Appendix to cover CRC Calculation<br>• Added Application Note 2, 3, 5, 6, 7 and 8 covering preset methodology, price change methodology, grade selecting, push to start and push to stop methodology<br>• Clarify preset messages, field lengths and use with 5 and 6 digit money mode |
| 2.10 | 02 SEP 2008 | Oldham | • Updated the document format<br>• Added QWB ID Number to document<br>• Added footer to document<br>• Update the general information, baud rates and options<br>• Added modern Two-Wire state diagram<br>• Added more pump state information |
| 2.11 | 28 JUL 2010 | Oldham | • Clarify the grade bit field of the “Range-of-Grades” Preset |
| 2.17 | 18 OCT 2010 | Oldham | • Clarified 2+1+1 and 3+1+1 style dispensers for Special Function 00E<br>• All Special Functions detailed with Field Descriptions |
| 2.18 | 20 JAN 2011 | Oldham | • Clarified 2+2 and 3+2 dispensers for Special Function 00E<br>• Pump Identification section to describe fields for totalizers, prices, grades by unit type code |
| 2.19 | 07 FEB 2011 | Oldham | • Correct field information for totals messages |
| 2.20 | 15 APR 2011 | Oldham | • Added SHF Unit Type Code<br>• Detailed SHF to Special Function 00E<br>• Special Functions in sequential order<br>• Special Function Data now details as command or response in each Special Function description |
| 2.21 | 21 APR 2011 | Oldham | • Added original Two-Wire State Transition Table and state transition descriptions back |
| 2.22 | 25 APR 2011 | Oldham | • Added Change History<br>• Clarified 2+2 and 3+2 style dispensers for Special Function 00E |
| 2.23 | 10 MAY 2011 | Oldham<br>Toogood | • Application Notes for China<br>• Clarified and Corrected Application Note 8<br>• Clarified Transaction Response (command 4) while pump is in Stop state<br>• Extra transitions on State Diagram & Tables<br>• Special Function 410 error code definitions<br>• Transaction Data Error Codes<br>• Application Notes re-numbered<br>• Application Note 9 added for China<br>• Added Sprint and Frontier to Application Node 8<br>• Removed approvals information from section 3.3 |
| 2.24 | 23 JUN 2011 | Oldham | • Unit type Codes for x+1+1, x+SHMPD2 and x+2 redefined.<br>• Grade and Totals mapping for 2+1+1, 3+1+1, 2+2 and 3+2 redefined.<br>• Unit Type 41 in v2.23 is split into Types 41 and 42.<br>• Unit Type 42 in v2.23 is split into Types 43, 44 and 45.<br>• Unit Type 43 in v2.23 is split into Types 46, 47, 48 and 49.<br>• Unit Type Code Table for SF00E set to reference Section 6. |

| | | | |
|---|---|---|---|
| | | | • SF405 Clarified that pumps may respond only with blend manifold one for Blender Unit Type Code 30 through 40.<br>• SF405 Clarified to only be valid for pure product totals of blend manifold 1, regardless of the unit type of the blender<br>• Document SF605 requirement for pure product totals of blend manifold 2 for Unit Types 46, 47, 48 and 49. |
| 2.25 | 15 SEP 2011 | Oldham | • Transaction Error Code tables corrected for improper listing of representative transaction error code numbers.<br>• Range of Grades Preset Field for Grade Data was denoted as 1 byte instead of 2 bytes. |
| 2.26 | 30 SEP 2011 | Oldham | • 3+2 Grade layout changed at request of Passport North America. |

# 2. INTRODUCTION

## 2.1 GENERAL DESCRIPTION

The purpose of this report is to describe the *Gilbarco* two-wire communication protocol which was designed to transmit commands and data between electronic pumps and consoles. This report describes the protocol level, message level, timing, and the effects of two-wire commands on the pump. This document will serve as a design and test tool for future pump and console design.

The two-wire protocol is a *Gilbarco* proprietary protocol which uses an 11 bit data format. The protocol was designed to support up to 16 active fueling positions. A single word command format and multiple word data block are used in this protocol to obtain maximum communication throughput. Throughout the specification, the console or controller means any device that can be connected to control the pumps.

This report provides a framework for a better understanding of the Two-Wire Protocol and the relationship between all Gilbarco consoles and pumps regardless of country of origin. It attempts to describe the basic, generic functional elements of pump control by a console or controller. It covers all aspects of basic pump control including:

- Bringing a pump on-line
- Configuring a pump
- Use of Special Functions
- Polling pumps
- Fuel Sale management
- Pump diagnostics
- Weights and Measures
- Safety

### 2.1.1 Compatibility

An important aspect of all of our products is compatibility. Backward compatibility between pump and consoles for existing functions is vitally important and a continuing Gilbarco customer service objective. The Two-Wire Protocol exists for that reason as much as any other. However, true compatibility requires that a pump and a console be compatible at not just the message level but also at the functional level.

### 2.1.2 Improved Communication

An important element in communication is standard, well understood terminology. This document provides accurate definitions of the terminology and functions of pump control for use by Gilbarco personnel worldwide.

### 2.1.3 Reduce Risk

This document reduces the risk of modifying the Two-Wire Protocol by improving our ability to evaluate proposed changes in terms of their impact on existing products.

### 2.1.4 Education

This document provides a source of accurate and complete information concerning pump control for employees new to the company or new to the pump control aspect of our products.

## 2.2 SCOPE

This document is a source for:

- An explanation of the relationship between a console/controller and pumps in the retail fuel marketplace;
- The interdependencies between a console/controller and a pump;
- A list of the assumptions of each product concerning the other;
- The chronological order of the Two-Wire message set in basic pump control functions;
- The status of the functional compliance with the Two-Wire of all Gilbarco pumps.

This section is:

- Not a functional specification for a specific pump or console;
- Not a definition of all current or future features of pumps and consoles;
- Not a source for hardware information.

## 2.3 ORGANIZATION

The 3 LINE LEVEL COMMUNICATION PROTOCOL, 4 MESSAGE LEVEL COMMUNICATION PROTOCOL, and 5 PUMP STATUS TRANSITION(S) sections provide three views of the pump/console relationship: line level communications, message level communications, and basic pump control functions. The Basic Pump Control section is written from the perspective of the console/controller with the topics appearing chronologically as they would be encountered by a console/controller from installation through operation and, if appropriate, pump programming and diagnostic control.

## 2.4 APPLICABLE REFERENCE DOCUMENTS

1. Two-Wire Driver Hardware Specification - TWO-HW-S1.0-S

## 2.5 GLOSSARY

The Glossary, normally the last section of a document, appears near the beginning in this one because it is vitally important that the reader know the meaning of key words and phrases used in this document. Some familiar terms may be used in a new way or new terms used in place of old standbys. A careful reading of this glossary along with reference to it during the reading of the other sections should help the reader gain a more complete understanding of the concepts contained in this document.

**Authorization** Is the site controller/console approving the dispensing of fuel by a pump. A pump may not dispense fuel prior to receiving authorization, however other elements may be necessary after authorization is received before dispensing of fuel can

| | |
|---|---|
| | begin. |
| **Blender** | Is any pump capable of blending two pure products to create a blended grade of fuel. This includes single hose blender pumps, as well as “fixed” blender pumps, which look to the user like a standard MPD. Electronic blenders may be blend ratio programmable or set. If they are programmable, their blend ratios cannot be changed over the Two-Wire. Set blenders do not allow their blend ratios to be changed. |
| **Mechanical Blender** | Can dispense 1 grade of blended fuel but the blend ratio is mechanically set and cannot be changed. Over the Two-Wire this pump cannot be distinguished from an MPD! |
| **Fixed Blender** | Looks to the user just like an MPD (e.g. a nozzle per grade) but uses the blending formats over the Two-Wire. Optionally it may have an additional hose which dispenses a non-blending grade. |
| **6 Hose Blender** | Same as Fixed Blender above. |
| **Command** | Is sent by the controller to the pump. A command contains data or instruction(s) for the receiving device. |
| **Communications Failure** | Is a lack of link level communications as defined by the Two-Wire link-level specification. |
| **Console** | May be a master (controller) or a secondary (to a controller) based on the individual product’s design. |
| **CRIND** | Is a Card Reader In a Dispenser. In the context of this document it is considered to be a separate physical device equivalent to a controller for pump control purposes. |
| **Customer** | The person manipulating the PUMP to cause it to deliver fuel. |
| **Device ID** | Or device address. The physical address of a device is part of the link level protocol. |
| **Dispenser** | Or device address. The physical address of a device is part of the link level protocol. |
| **Dispenser** | Is a unit which is used with a remote submerged turbine pump or STP. |
| **Download** | Is the term used to describe the communication from a controller to a pump of the parameters which control the operation of that pump. The latter may also be referred to as Parameter Download. |
| **Fuel Delivery Condition** | The logical condition of a pump with respect to its ability to dispense fuel. |
| **Fuel Sale** | A set of information/data related to a particular instance of dispensing of fuel. A fuel sale becomes one line item in a transaction, but is used interchangeably with transaction here and in the Two-Wire Specification. |
| **Fueling Position** | Is the term used to describe a single logically addressable device which dispenses fuel and with which the Controller communicates concerning Fuel Sales. A fueling |

| | |
|---|---|
| | position may control multiple hoses and dispense multiple grades of fuel. |
| **Grade** | Is the blended or pure product which is dispensed by a pump. |
| **Hose** | Is the physical hose which dispenses a grade (or grades of fuel). There is not necessarily a one to one ratio equivalence for grade to hose due to blending where one hose may dispense multiple grades and MPDs where the same grade may be dispensed from multiple hoses. |
| **LSD** | Least Significant Digit: The rightmost digit in a number. |
| **Link Level Protocol** | The physical communications protocol. |
| **Message Level** | Is the definition of the format, structure and meaning of the messages carried by the link level protocol. |
| **Messages** | Sent between the pumps and the controller via the link level protocol. |
| **MSD** | Most Significant Digit: The leftmost, non-zero digit in a number. It is the digit with the greatest value in the number. |
| **Operator** | The person directing the CONTROLLER/CONSOLE in its management of a FUEL SALE. |
| **PAM** | Is a Pump Access Module which allows a third party device to control Gilbarco pumps, providing limited pump control and protocol conversion. |
| **Polling** | Is the process of soliciting status from the pumps. |
| **Price level** | Selects one of the prices in the pump at which a grade of fuel may be dispensed (e.g. Cash price/Credit price). Price levels may be set by the controller or at the pump. |
| **Product flow** | Is defined to have begun when the pump no longer returns a zero volume amount for Transaction Data. For example, if a pump does not display any volume until X pulses have occurred, no product has flowed until X is reached. |
| **Program** | Used as a noun refers to the actual binary object file of a device. When it is used as a verb it means to set the operating parameters for a pump or a controller. |
| **Pump** | A pump is the generic term used when it is not necessary to differentiate between a dispenser and a self-contained unit. While a |
| **Pump Handle** | (Or lever) when present, is a mechanism the consumer must lift or turn to cause the pump to recognize that someone wants to use the pump. |
| **Pump Totals** | Are the totals of the volume and money amount dispensed by grade. These totals can be requested over the Two-Wire. These totals are never reset by the pump other than through Cold Start and at rollover (although some pumps may allow programming at the keypad or from the controller). |
| **Pure Product** | Is the fuel which is kept in the storage tanks at a site. Pure product may be dispensed as a non-blended grade or form a new grade in a blend with another pure product. |

| | |
|---|---|
| **Pure Product Totals** | Are the non-resettable volume totals kept by a blender for the pure products used to make the blended grades. These totals are never reset to zero except at Cold Start and rollover (although some pumps may allow programming at the keypad). |
| **Real Time Money** | Is the process of a controller monitoring a fuel sale so the operator may watch the money total for the sale increase as the fuel is dispensed. |
| **Response** | Is a message sent by a pump to a controller. Responses are the result of a command received previously and can not require a response in turn. |
| **Self Contained** | Is a unit which has the actual pump contained inside the unit housing, and does not depend on an STP. |
| **STP** | Submerged Turbine Pump. The pumps are |
| **Settlement** | Is the process of actually paying for the sale and removing it as an active sale at the console. |
| **Standalone** | Is the mode of operation in which a pump authorizes the dispensing of fuel without the controller. The pump will not communicate to the controller while it is operating in Standalone, but will continue to accumulate Pump Totals and Pure Product Totals. |
| **System** | Is the entity resulting from the interconnection of a controller/console with pumps. |
| **Terminated Fuel Sale** | Is a fuel sale which has been discontinued such that no more fuel may be delivered, however all conditions are not met for the fuel sale to be complete. |

# 3. LINE LEVEL COMMUNICATION PROTOCOL

This section describes the line level of the two-wire communications protocol between the pumps and the console.

The line level protocol is basically a multi-drop protocol which supports up to 16 terminals (pumps). When the master (i.e., Console) sends the data or the terminal (pumps) responds, every unit in the loop will receive the same data. Only the addressed terminal (pump) should respond and the rest of the terminals should ignore the data. The master (i.e., Console) will be the only unit in the loop which is allowed to initiate communication.

This document does not define the physical implementation of the link. The standard form of physical layer is a 45 mA current loop, as defined in the referenced documents “Two-Wire Hardware Specification - TWO-HW-S1.0-S” and “Electronic Dispenser Controller Interface Specification for Current Loop or RS422 - TWOTP-HW - V50.0-P”. Other physical layers may be used, such as RS485 or Ethernet. These will not be inter-operable with standard devices except with some form of hardware adapter. The descriptions and timings which follow assume standard asynchronous communications using a UART or equivalent, and are applicable to the standard current loop.

## 3.1 TWO WIRE CURRENT LOOP INTERFACE

The standard current loop operates at a nominal 45 mA +/- 5%, for a mark (loop idle or logic 1) and less than 2 mA for a space (logic 0). The current regulator must be designed to within +/- 5% of the nominal marking current over the operating temperature range of 0 to 50 ºC. The current loop open-loop supply voltage must be between 12 Volts D.C. and 42 Volts D.C. Some dispensers may clamp the voltage to prevent it exceeding the upper limit; most do not, and may be damaged if the upper limit is exceeded.

## 3.2 NOTATION CONVENTIONS

This defines some notation and terms that are used in this section.

a. Command Code
   A command code is an 8 bit word which is sent by the console to command the pump to perform defined actions. (See 3.7DATA AND DATA CONTROL WORD (DCW)).

b. Data Control Word (DCW)
   DCW is an 8 bit word, as part of a data block, which is used to indicate what the next words are. The most significant nibble of a data control word is always F.

c. LRC Check Character
   LRC is a 4 bit word, as part of a data block, which is used to detect longitudinal bit errors (See 3.6LRC CHECK CHARACTER).

d. Data Length (DL)
   DL is an 8 bit word, as part of a data block, which is used to indicate how many words are being transmitted. (Section 3.4 LINE PROTOCOL, TIMING REQUIREMENTS AND ERROR RECOVERY).

e. No Response Timer
   No response timer is a time interval which is set up by the console to wait for pump responses before retransmitting the request or taking any actions.

f. Retry Counter
   Retry counter indicates the number of tries a requesting device will attempt to obtain the appropriate response before it takes an appropriate action.

g. Pump ID
Pump ID is a four bit address which is set at the pump at installation time to indicate the address of the pump.

h. Data Word
Data Word is an 8 bit word, as part of a data block, which conveys the actual message. The most significant nibble of a Data Word is always 'E'.

i. Pumps, dispenser, fueling positions are used in this specification interchangeably. It indicates a unique addressable communication position. For example, a dual MPD will have two fueling positions, but could have up to 8 hoses.

j. The following notation will be used throughout the report.

| | |
|---|---|
| < > | Encloses symbols reference |
| { } | Enclosed optional data |
| ' ' | Encloses a four bit nibble |
| DL | Data Length |
| STX | Start of Text |
| ETX | End of Text |
| LRC | LRC - 4 check character (least significant nibble) |
| DCW | Data Control Word |

## 3.3 WORD FORMAT

Each word on the two-wire system consists of 11 bits. They are:

- Start bit (space)
- 8 data bits (least significant bit transmitted first)
- Even parity
- Stop bit

The bit rate is 5787 ± 0.5% bits/second, including variations due to frequency and clock phase shift tolerance.

### 3.3.1 Exception

Mandates for 1200, 2400, 4800, 9600 and 19200 bits/second are used for special configurations worldwide.

## 3.4 LINE PROTOCOL, TIMING REQUIREMENTS AND ERROR RECOVERY

This section describes the five types of data and their timing requirements and error recovery. They are:

- polling (status)
- a single word command
- send data to pump
- send data to console
- special function command

Note that all the timing requirements in this section indicate the time interval between the last bit of the previous word and the first bit of the next word.

### 3.4.1 POLLING (Status)

The pump will respond with one of the status words (see 3.9PUMP STATUS WORDS) upon receipt of a STATUS REQUEST Command Code with a matched pump ID, from the console. It will ignore the STATUS REQUEST Command Code if it does not have a matched pump ID. The protocol exchanges are as follows:

| **Console** | | **Pump** |
|---|---|---|
| <STATUS REQUEST> | ----------> | |
| | t1 | |
| | <---------- | <PUMP STATUS> |
| | t2 | |
| <NEXT TRANSMISSION TO ANY PUMP> | -------> | |

t1 = 0ms - 68ms
t2 = at least 5ms

Exception: Some very old pumps might exceed (be slower than) the t1 68ms limit.

Error Recovery Rules:

1. No-response timer should be at least 68ms.
2. Retry counter should be set by the console, depending on the application program. It is recommended that it should be at least 5.
3. It is advised that the console check whether or not the response comes from the polled pump.
4. The console should check the pump status frequently. The appropriate actions should be taken if the pump state is not compatible with the console state.

### 3.4.2 A SINGLE WORD COMMAND

A single word command can be sent from the console to direct the pumps to perform certain tasks. There is no response to a single word command from the pump. Therefore, it is advised that the console request the status from the affected pump and ensure the proper command has been received and processed. All single word commands except All Stop (FC) are for one pump at a time. The protocol exchanges are as follows:

| **Console** | | **Pump** |
|---|---|---|
| <COMMAND CODE> | ----------> | |
| | t0 | |
| <STATUS REQUEST> | ----------> | |
| | t1 | |

```
                                        <----------          <PUMP STATUS>
                                            t2
<NEXT TRANSMISSION TO ANY PUMP>         ---------->
```

t0 = at least 68ma
t1 = 0ms - 68ms
t2 = at least 5ms

Exception: Some very old pumps might exceed (be slower than) the t1 68ms limit.

Error Recovery Rules:

1. No-response timer should be at least 68ms.
2. Retry counter for status should be set by the console depending on the application program. It is recommended that it should be at least 5.
3. It is advised that the console check whether or not the response comes from the polled pump.
4. The console should check if the command is received and processed correctly. A re-transmission of the command should be attempted by the console if it detects that the pump failed to process the command correctly. The exact retry counter should depend on the application program and at least 5 retries are recommended.
5. If the pump is not in one of the valid pump states for the command, it shall ignore the command.
6. If the pump receives a single word command with transmission error, it shall ignore the command.

### 3.4.3 DATA TO PUMP

The two-wire protocol utilizes a command word to place the pump in a DATA state so that the console can send the data to the pump. The protocol exchanges are as follows:

```
          Console                                        Pump
<DATA NEXT COMMAND>             ---------->
                                    t1
                                <----------          <SEND DATA>
                                    t2
      <DATA BLOCK>              ---------->
                                    t3
   <STATUS REQUEST>             ---------->
                                    t1
                                <----------          <PUMP STATUS>
                                    t2
<NEXT TRANSMISSION TO ANY PUMP> ---------->

          <DATA BLOCK> :=   <STX>
                                t6
                            <DCL>
                            <message>
                            <LRC next>
                            <LRC>
                            <ETX>
```

t1 = 0ms - 68ms
t2 = at least 5ms
t3 = at least 68ms
t6 = The time interval between words in the data block should be at least 68ms.

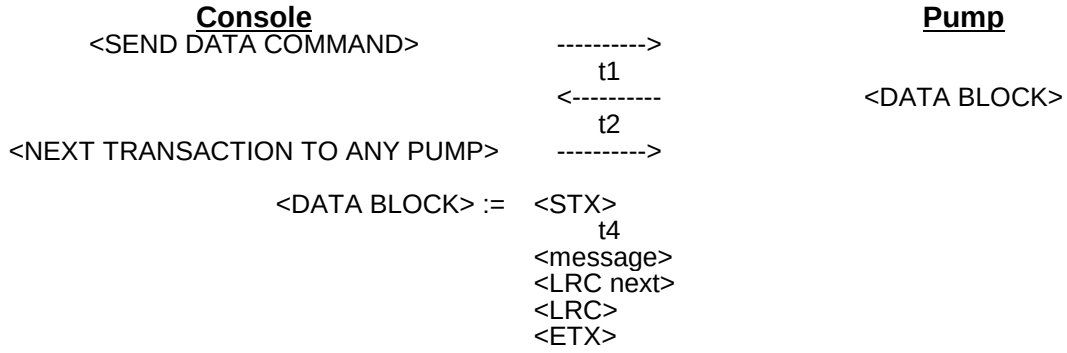
Exception: Some very old pumps might exceed (be slower than) the t1 68ms limit.

Error Recovery Rules:

1. No-response timer should be at least 68ms.
2. If any transmission error occurs in DATA BLOCK, the pump will send ERROR status in response to the next STATUS REQUEST or DATA NEXT command. The term “transmission error” includes parity error, block length error, framing error, overrun error and checksum error. The console must re-transmit the whole sequence starting from <DATA NEXT> Command, unless a correct status is received. The exact retry counter should depend on the application program and at least 5 retries are recommended.
3. The pump should enter the DATA STATE when it receives DATA NEXT Command and exit from the DATA STATE as soon as it receives any word, which is not a data or a data control word. The most significant nibble of a data word is always an E and that of a data control word is F (See 3.7DATA AND DATA CONTROL WORD (DCW)).
4. The console must not transmit a DATA BLOCK to a pump, which has a preset transaction pending. “Preset transaction pending” means that:
   - a pump has received either a money or a volume preset message.
   - the pump did not enter the ERROR state in response to the preset message,
   - the pump has not yet “completed” the preset transaction by:
     - receiving a pump stop or universal stop command (without an authorization command), OR
     - receiving an authorization command followed by a handle on to handle off transition, with or without fuel being dispensed).

### 3.4.4 DATA TO CONSOLE

These commands are used by the console to obtain data; such as transaction data or pump totals, from the pumps. The line level protocol exchange is as follows.




t1 = 0 - 68ms
t2 = at least 5ms
t4 = all data blocks transmitted to the console must meet this condition:
The interval between words must be between 2ms and 68ms.

Data blocks with 8 or more words must also meet this condition:

For any group of 8 consecutive words, the average word transmission rate shall not be faster than 6.5ms, as computed below:

$$\frac{\textit{Sum of 8 word transmission times + sum of 7 inter-word delays}}{8}$$

must be greater than or equal to 6.5ms.

Exceptions:

Sandpiper-based and E101-based pumps may transmit message words contiguously.
Some very old pumps might exceed (be slower than) the t1 68ms limit.

Error Recovery Rules:

1. No-response timer should be at least 68ms.
2. Retry counter should be set by the console depending on the application program. It is recommended that it should be at least 5.

## 3.5 DATA LENGTH (DL)

The data length is included only in the data block from the console to the pump. The lower nibble of DL is the modulo 16 of the Two's complement of number of words in the data block excluding Start of Text and DL.

## 3.6 LRC CHECK CHARACTER

The LRC check character is included in every data block transmitted between the console and the pumps except the Special Function Data Request response sequences from the pumps (refer to 4.8.3 PRICE CHANGE DATA). It is a 2's complement of the summation of the least significant nibbles of all words in a data block.

## 3.7 DATA AND DATA CONTROL WORD (DCW)

The message in a data block consists of Data and Data Control Words. Each group of data must be preceded by a Data Control Word (DCW) to indicate the type of data. The most significant nibble of Data is always E and that of Data Control Words F. Table 1 describes all DCW.

### Table 1: DATA and DATA CONTROL WORD

| | | Standard Word Format | |
|---|---|---|---|
| **A.** | **Data Words** | E | 0-F |
| **B.** | **Data Control Words** | | |
| | End of Text | F | 0 |
| | Volume Preset | F | 1 |
| | Money Preset | F | 2 |
| | Fill-Up | F | 3 |
| | Level 1 | F | 4 |
| | Level 2 | F | 5 |
| | Grade Data Next | F | 6 |
| | PPU Data Next | F | 7 |
| | Pump Identifier Next (from Pump)<br>Preset Quantity Next (from Console) | F | 8 |
| | Volume Totals Next | F | 9 |
| | Money Totals Next | F | A |
| | LRC Next | F | B |
| | Unused | F | D |
| | Special Function Mode Next | F | E |
| | Start of Text | F | F |

## 3.8 COMMAND CODE

Each Command code except ALL STOP has the following format (Note: command code can only be sent by the console):

### Table 2: PUMP ID and COMMAND CODES

| Four least significant bits | ---- | Pump ID |
|---|---|---|
| 1 | ---- | Pump 1 |
| 2 | ---- | Pump 2 |
| : | ---- | : |
| F | ---- | Pump 15 |
| 0 | ---- | Pump 16 |

| Four most significant bits | ---- | Command Code |
|---|---|---|
| 0 | ---- | Status Request |
| 1 | ---- | Authorization |
| 2 | ---- | Data Next |
| 3 | ---- | Pump Stop |
| 4 | ---- | Transaction Data Request |
| 5 | ---- | Pump Totals Data Request |
| 6 | ---- | Real-Time Money Request |
| FC | ---- | ALL STOP* |

* The ALL STOP command has a different data format.
Every pump should process it
(see 4.4BROADCAST COMMANDS TO ALL PUMPS)

## 3.9 PUMP STATUS WORDS

Upon the receipt of a STATUS REQUEST Command, the pump with a matched pump ID will respond with one of the following pump status to the console. The pumps without a matched pump ID shall ignore the command. The pump status words have the following data format. Note that only the pumps can send the pump status.

### Table 3: PUMP ID and STATUS WORDS

| Four least significant bits | ---- | Pump ID |
|---|---|---|
| 1 | ---- | Pump 1 |
| 2 | ---- | Pump 2 |
| : | ---- | : |
| F | ---- | Pump 15 |
| 0 | ---- | Pump 16 |

| Four most significant bits | ---- | Status Code |
|---|---|---|
| 0 | ---- | Data Error |
| 6 | ---- | Off |
| 7 | ---- | Call |
| 8 | ---- | Authorized/Not Delivering |
| 9 | ---- | Busy |
| A | ---- | Transaction Complete (PEOT) |
| B | ---- | Transaction Complete (FEOT) |
| C | ---- | Pump Stop |
| D | ---- | Send Data* |

* This status is returned only to a <DATA NEXT COMMAND>.
For details, please refer to 4.1STATUS POLLS: COMMAND '0'

# 4. MESSAGE LEVEL COMMUNICATION PROTOCOL

This section describes the message level of the two-wire protocol. It will also describe the appropriate pump response and console responses in order to facilitate the design and testing of future pumps and consoles. Please refer to Section 2 for line level, error recovery and timing requirement. We will also include the permissible pump status after the pump accepts a command.

## 4.1 STATUS POLLS: COMMAND '0'

The command may be sent by the console to request the current pump status from the pump. Only the pump with a matched ID should respond.

| Command/Response Syntax<br>Valid Pump State: Any pump state | | | | | |
|---|---|---|---|---|---|
| Command | '0' '<p>' | | | | |
| | | <p> | Pump Number | | |
| | | | 1 | ---- | Pump 1 |
| | | | 2 | ---- | Pump 2 |
| | | | : | | : |
| | | | F | ---- | Pump 15 |
| | | | 0 | ---- | Pump 16 |
| Response | '<r>' '<p>' | | | | |
| | | <r> | Pump Status | | |
| | | | 0 | ---- | DATA ERROR |
| | | | 6 | ---- | OFF |
| | | | 7 | ---- | CALL |
| | | | 8 | ---- | AUTH<br>(authorized but not yet delivered) |
| | | | 9 | ---- | BUSY<br>(delivering product) |
| | | | A | ---- | PEOT<br>(transaction complete) |
| | | | B | ---- | FEOT<br>(transaction complete) |
| | | | C | ---- | Stop<br>(pump stop) |
| State After Command Acceptance | | A Pump may change its two-wire status as a result of a status poll only as follows:<br>1. Clear ERROR condition<br>2. Transition from FEOT or PEOT to OFF or CALL | | | |

Note:

1. Pumps will respond FEOT/PEOT repeatedly to status polls, until another pump ID is polled or the pump receives some command other than status poll.

## Table 4: PUMP STATE/STATUS DEFINITIONS

| | | |
|---|---|---|
| 0 | DATA ERROR | The pump will enter the DATA ERROR state under the following conditions:<br>1. The pump is in DATA STATE when it receives an invalid data block including any word that the most significant nibble is not an E or F and any transmission error such as parity error, checksum, overrun or framing error<br>2. The pump receives a correctly formatted message containing incorrect fields. |
| 6 | OFF | Pump handle is off and it is not authorized. |
| 7 | CALL | The pump handle is ON, but not authorized. The CASH/CREDIT option, preset option or others at the pump may already be activated. |
| 8 | AUTH | The pump is authorized but not ready to deliver fuel. This may mean that one of delivery conditions is not met such as CASH/CREDIT option, key lock option, Push-to-Start, pump handle, price-change delay, LPG pressurization period, or other options. Refer to individual pump software report for those options. |
| 9 | BUSY | The pump is authorized and all delivery conditions (CASH/CREDIT option, key lock option, Push-to-Start option, pump handle or other options) are met. The pump is in a state which is ready to deliver fuel, delivering fuel, has reached a preset goal, or is performing lamp test. Refer to individual pump software report for those options. |
| A | PEOT | The pump has completed a delivery with the pump handle at 'OFF' position. |
| B | FEOT | The pump has completed a delivery with the pump handle at 'OFF' position. |
| C | STOP | The pump is in a PUMP STOP state.<br><br>The pump will enter DATA ERROR STATE if it is in DATA STATE and detect a break condition as defined in the RS232 standard. Most pumps will process an ALL-STOP command if they detect a loop-break condition for (typically) at least 500ms. |

## 4.2 AUTHORIZE OR RE-AUTHORIZE: COMMAND ’1’

The command is used to enable a pump to deliver or to resume the delivery of the fuel product. In some European pumps, the Authorization command is used when the pump is in state IDLE/OFF to clear the transaction displays to zero. Only the pump with a matched pump ID should accept this command.

| Command/Response Syntax<br>Valid Pump State: OFF or CALL for authorization, STOP for re-authorization | | | | | |
|---|---|---|---|---|---|
| Command | ’1’ ’<p>’ | | | | |
| | | <p> | Pump Number | | |
| | | | 1 | **----** | Pump 1 |
| | | | 2 | **----** | Pump 2 |
| | | | : | | : |
| | | | F | **----** | Pump 15 |
| | | | 0 | **----** | Pump 16 |
| Response | *None* | | | | |
| State After Pump Accepts the Command | | | AUTH or BUSY | | |

## 4.3 PUMP STOP: COMMAND ’3’

This command is used to de-authorize an authorized pump. It may also be used to cancel a Preset so that a different limit can be sent (see Appendix A: Application Note #005). Some European dispensers can be configured by the Manager Keypad so that a Pump Stop command received in the OFF state will turn off the Red Drive-off Light. This is used to show that the sale has been paid. Only the pump with a matched pump ID should accept this command.

| Command/Response Syntax<br>Valid Pump State: AUTH or BUSY for de-authorization, OFF or CALL for Preset cancel | | | | | |
|---|---|---|---|---|---|
| Command | ’3’ ’<p>’ | | | | |
| | | <p> | Pump Number | | |
| | | | 1 | **----** | Pump 1 |
| | | | 2 | **----** | Pump 2 |
| | | | : | | : |
| | | | F | **----** | Pump 15 |
| | | | 0 | **----** | Pump 16 |
| Response | *None* | | | | |
| State After Command Acceptance | | | OFF, CALL or STOP | | |

## 4.4 BROADCAST COMMANDS TO ALL PUMPS

### 4.4.1 ALL STOP: COMMAND ’F’ ’C’

This command is used by the console to stop every pump in the loop.

| Command/Response Syntax<br>Valid Pump State: OFF, BUSY, CALL or STOP | |
|---|---|
| Command | ’F’ ’C’ |
| Response | *None* |
| State After Command Acceptance | OFF, CALL or STOP |

## 4.5 REQUEST FOR TRANSACTION DATA: COMMAND '4'

The command is used by the console to request the transaction data. Dispensers should return data for the *current transaction* in the STOP state (and Call *after* Authorization), the just-finished transaction in the FEOT and PEOT states, and previous transaction in OFF and CALL states before Authorization. The transaction data is finalized before the pump transmits EOT status (or returns to OFF in the case of a zero transaction).

Controllers should be aware that not all dispensers will do this. In particular, some will continue to send the *previous transaction* data even after Authorization in the CALL and STOP states.

Command/Response Syntax
Valid Pump State: OFF, CALL, FEOT, PEOT and STOP

Command '4' '<p>'

<p> Pump Number

| | | |
|---|---|---|
| 1 | ---- | Pump 1 |
| 2 | ---- | Pump 2 |
| : | | : |
| F | ---- | Pump 15 |
| 0 | ---- | Pump 16 |

Response

| Symbol | Words | Bytes | Definition |
|---|---|---|---|
| <STX> | FF | 1 | Start of text |
| <pt> | FX | 1 | Reserved: (preset type: obsolete)<br>Pump will send X=1, 2, or 3.<br>Console should ignore this word. |
| <pin> | F8 | 1 | Pump identifier next |
| <pi>*[1] | EY<br>EX<br>EX<br>EX<br>EX | 5 | Pump identifier data, see below for details<br>Y = A or B<br>X = 0 ~ F |
| <gn> | F6 | 1 | Grade data next |
| <g> | EX | 1 | Grade<br>X = 0 – Grade 1<br>\|<br>\|<br>X = F – Grade 16 |
| <tt> | FX | 1 | Transaction Type i.e.,<br>what PPU is used for this transaction:<br>X = 4 – Level 1<br>X = 5 – Level 2 |
| <ppun> | F7 | 1 | PPU data next |
| <ppu> | EX<br>EX<br>EX | 4 | Price per unit in XXXX format (BCD)<br>X = 0 – 9<br>Least significant digit is transmitted first. |

| | | | |
|---|---|---|---|
| | EX | | LSD is the LSD of the displayed PPU at the pump.<br><br>Normally the tenth penny digit (U.S. currency) for non-Advantage pumps. For Advantage series, decimal position is defined in 4.9.4.2 Miscellaneous Pump Data Request (X3X2X1 = 00E). |
| <vn> | F9 | 1 | Transaction Volume data next |
| <v> | EX<br>EX<br>EX<br>EX<br>EX<br>EX | 6 | Transaction volume in XXX.XXX format (BCD)<br>X = 0 9<br>Least significant digit is transmitted first |
| <cn> | FA | 1 | Transaction Money data next |
| <c> | EX<br>EX<br>EX<br>EX<br>EX<br>EX | 6 | Transaction Money in XXXXXX format (BCD)<br>X = 0 – 9<br>Least significant digit (LSD) is transmitted first. In 5 digit mode, LSD is the hidden digit which is not displayed at the pump. It is a tenth penny digit in US currency. In 6 digit mode, all 6 digits are significant and the value depends on the money decimal point configuration programmed. |
| <LRCn> | FB | 1 | LRC check character next |
| <LRC> | EX | 1 | LRC check character<br>X = 0 – F |
| <ETX> | F0 | 1 | End of text |
| Total Bytes = | | 33 | |

State After Command Acceptance OFF, CALL or STOP

Refer to section 3.4.4 DATA TO CONSOLE for the line level communication requirement.

*[1] There are two types of data which can be included in the pump identifier. The EA response form must not include ERROR information. The console should ignore EX data words. EC-EF shall be interpreted as EA.

A. EA Ep Ex Ex Ex

p: Pump number
0 ---- pump 1
|
|
F ---- pump 16

x: 0 to F don't care

B. EB Ep Eh Ea Eb

p: Pump number

0 --- pump 1
|
|
F --- pump 16

h: 0 ---- no error
1 - F --- hose number
1 ---- left most hose at side A or right most hose at side B
2 ---- next to hose 1
: :
a,b: a and b are used as error code and identification bits.

Error code: b1, a4, a3, a2, a1 ---- a combined 5 bit word.

b1 is the least significant bit of b.

| E | $b_4$ | $b_3$ | $b_2$ | $b_1$ | | E | $a_4$ | $a_3$ | $a_2$ | $a_1$ |
|---|---|---|---|---|---|---|---|---|---|---|

$b_4b_3b_2$ indicate the number of grades per fueling position.*

$b_4b_3b_2$ = 000 1 - 3 grade pump (non-blending)
$b_4b_3b_2$ = 001 4 - 6 grade pump (non-blending)
$b_4b_3b_2$ = 010 6 grade blending pump

*$b_4b_3b_2$ will always equal 000 for non-blending Multiline pumps, regardless of the number of grades in the pump's configuration. Blending Multiline pumps will return 010, the same as all other blenders.

## Table 5: TRANSACTION DATA ERRORS

Many modern pumps use the error field to report error states. Many of these states are common across the pumps that support this. Sandpiper based SK700 and E500, E101 based SK700 and Horizon, Greensboro Advantage, Legacy, Performer, Endeavor and China Advantage pumps may have support for this. Common errors that can be reported are listed below.

| Error Code | 5-Bit Decimal Error | Binary b1, a4, a3, a2, a1 | E500 Horizon SK700 | Optimized Advantage China Advantage Endeavor Legacy Performer E300 |
|---|---|---|---|---|
| 50 | 31 | 11111 | POS Communications Failure | |
| 49 | 30 | 11110 | | |
| 48 | 29 | 11101 | | |
| 47 | 28 | 11100 | | |
| 46 | 27 | 11011 | Minimum Preset | |
| 45 | 26 | 11010 | | |
| 44 | 25 | 11001 | Pump Handle Error<br>Pump Handle on at Power Up | Pump Handle Error<br>Pump Handle on at Power Up |
| 43 | 24 | 11000 | General Use<br>Must use Special Function 410 | General Use<br>Must use Special Function 410 |
| 42 | 23 | 10111 | | Price Posting Option Changed |
| 41 | 22 | 10110 | | Pump Exists Option Changed |
| 40 | 21 | 10101 | | |
| 39 | 20 | 10100 | | |
| 38 | 19 | 10011 | | |
| 37 | 18 | 10010 | | |

| | | | | |
|---|---|---|---|---|
| 36 | 17 | 10001 | | |
| 35 | 16 | 10000 | | Configuration Data Error |
| 34 | 15 | 01111 | Battery Charge Low | Battery Charge Low |
| 33 | 14 | 01110 | Stop Button Activated | Stop Button Activated |
| 32 | 13 | 01101 | Maximum Auth Time<br>Maximum Fill Time | Pulser Count Error |
| 31 | 12 | 01100 | DEF/Adblue Freeze Error | Totals Data Error |
| 30 | 11 | 01011 | Vapor Recovery Error | VaporVac Shutdown |
| 29 | 10 | 01010 | Valve Stuck<br>Pump Timeout<br>Dispenser Timeout<br>Initial Dispenser Timeout | Dispenser Timeout Error<br>Initial Dispenser Timeout |
| 28 | 9 | 01001 | | |
| 27 | 8 | 01000 | | |
| 26 | 7 | 00111 | Not Calibrated | Not Calibrated (Where Applicable) |
| 25 | 6 | 00110 | | Two-Wire/Standalone Changed |
| 24 | 5 | 00101 | Conversion Factor Error | Conversion Factor Error |
| 23 | 4 | 00100 | | |
| 22 | 3 | 00011 | | |
| 21 | 2 | 00010 | | Non-existent memory access |
| 20 | 1 | 00001 | Pulser Error<br>Pulser Failure | Pulser Failure |
| -- | 0 | 00000 | No Error | No Error |

## 4.6 REQUEST FOR PUMP TOTALS: COMMAND '5'

The command is used by the console to request the pump electronic volume and money totals that are stored in the pump by grade. Only the pump with a matched pump ID should respond to the command. Also the PPU is included in the data block which can be used to check if the pump has the correct price per unit.

Command/Response Syntax
Valid Pump State: OFF, CALL, FEOT, PEOT, and STOP

Command '5' '<p>'

<p> Pump Number

| | | |
|---|---|---|
| 1 | ---- | Pump 1 |
| 2 | ---- | Pump 2 |
| : | | : |
| F | ---- | Pump 15 |
| 0 | ---- | Pump 16 |

Response

| Symbol | Words | Bytes | Definition |
|---|---|---|---|
| <STX> | FF | 1 | Start of text |
| <gn> | F6 | 1 | Grade data next |
| <g> | EX | 1 | Grade<br>X = 0 – Grade 1<br>\|<br>\|<br>X = F – Grade 16 |
| <vn> | F9 | 1 | Pump Volume data next |
| <v>*[1] | EX<br>EX<br>EX<br>EX<br>EX<br>EX<br>EX<br>EX | 8 | Pump volume totals in the format<br>XXXXXX.XX (BCD)<br>X = 0 – 9<br>Least significant digit is transmitted first |
| <cn> | FA | 1 | Pump Money data next |
| <c> | EX<br>EX<br>EX<br>EX<br>EX<br>EX<br>EX<br>EX | 8 | Pump money totals in the format<br>XXXXXXXX (BCD).<br>X = 0 – 9<br>Least significant digit (LSD) is transmitted first. LSD is the LSD of the displayed pump totals at the pump. The number of decimal places will be equal to the number programmed via the money decimal point configuration programming. |
| <ppun> | F4 | 1 | Level 1 PPU data next |
| <ppu1> | EX<br>EX | 4 | Level 1 Price per unit in XXXX format (BCD) |

| | | | |
|---|---|---|---|
| | EX<br>EX | | X = 0 – 9<br>Least significant digit (LSD) is transmitted first. LSD is the LSD of the displayed PPU at the pump. In non-Advantage pumps, it is a tenth penny digit in the US currency. In Advantage series pumps the decimal position of this field matches the PPU decimal point configuration. |
| <ppun> | F5 | 1 | Level 2 PPU data next |
| <ppu2> | EX<br>EX<br>EX<br>EX | 4 | Level 2 Price per unit in XXXX format (BCD)<br>X = 0 – 9<br>Least significant digit (LSD) is transmitted first. LSD is the LSD of the displayed PPU at the pump. In non-Advantage pumps, it is a tenth penny digit in the US currency. In Advantage series pumps the decimal position of this field matches the PPU decimal point configuration. |

*(Repeat from <gn> Grade Data Next through <ppu2> digits for Grade 2, 3, 4, 5, and 6)*

| | | | |
|---|---|---|---|
| <LRCn> | FB | 1 | LRC check character next |
| <LRC> | EX | 1 | LRC check character<br>X = 0 – F |
| <ETX> | F0 | 1 | End of text |

| | | |
|---|---|---|
| Total Bytes = | 34 | Pump sending 1 grade |
| | 64 | Pump sending 2 grades |
| | 94 | Pump sending 3 grades |
| | 124 | Pump sending 4 grades |
| | 154 | Pump sending 5 grades |
| | 184 | Pump sending 6 grades |

State After Command Acceptance OFF, CALL or STOP

Refer to section 3.4.4 DATA TO CONSOLE for the line level communication requirement.

*[1] <v> and <c> will be updated when a transaction that has at least 10 pulses is complete in Advantage based pumps or when .01 units of fuel have been dispensed on Tulip, Sandpiper and E101 based pumps.

The pump totals are updated before the pump transmits EOT status.

## 4.7 REQUEST FOR REAL-TIME MONEY: COMMAND ’6’

This command is used by the console to request the currently running transaction amount while the pump is delivering fuel. Only the pump with a matched pump ID should respond to this command.

| Command/Response Syntax<br>Valid Pump State: BUSY | | | | |
|---|---|---|---|---|
| Command | ’6’ ’<p>’ | | | |
| | | <p> | Pump Number | |
| | | | 1 ---- | Pump 1 |
| | | | 2 ---- | Pump 2 |
| | | | : | : |
| | | | F ---- | Pump 15 |
| | | | 0 ---- | Pump 16 |
| Response | | | | |
| | EX | | 6 | The Current Money Display in the format XXXXXX (BCD).<br>X = 0 to 9 in BCD format, the current money display.<br>The least significant digit is transmitted first. The decimal position of this field matches the transaction data money decimal point position. |
| | EX | | | |
| | EX | | | |
| | EX | | | |
| | EX | | | |
| | EX | | | |
| Total Bytes = | | | 6 | |
| State After Command Acceptance | | | The pump should not change the STATE unless other events occur, such as pump handle position, during transmission. | |

Note that some dispensers can be configured to return Real-Time Volume *instead of* Real-Time Money. In that case, the format of the Volume data matches that in the Transaction Data (Command 4) response message.

## 4.8 SEND DATA TO PUMP: COMMAND ’2’

This section describes how the console sends the data to the pump. Refer to 3.4.3 DATA TO PUMP for line level protocol information. Each data block message is processed by one pump only. In order for a pump to process a message, it must first receive a Data Next single-byte command from the controller.

### 4.8.1 “RANGE-OF-GRADES” PRESET DATA

This is an alternative Standard format for “range of grades” money or volume preset.

Command/Response Syntax
Valid Pump State: OFF, CALL
*Do not issue this command to a pump which has a preset transaction pending (see 3.4.3 DATA TO PUMP).*

Command ’2’ ’<p>’

<p> Pump Number

| | | |
|---|---|---|
| 1 | ---- | Pump 1 |
| 2 | ---- | Pump 2 |
| : | | : |
| F | ---- | Pump 15 |
| 0 | ---- | Pump 16 |

Response ’D’ ’<p>’

<p> Pump Number

Command <DATA BLOCK>

| Symbol | Words | Bytes | Definition |
|---|---|---|---|
| <STX> | FF | 1 | Start of text |
| <DL> | EX | 1 | Data length*[5]<br>X = 2 – grade group and price level specified<br>X = 3 – grade group specified only<br>X = 5 – price level specified only<br>X = 6 – neither grade group or price level specified<br><br>Data Length for 6 Digit Money Modes: *[5]<br>X = 1 – grade group and price level specified*[5]<br>X = 2 – grade group specified only*[5]<br>X = 4 – price level specified only*[5]<br>X = 5 – neither grade group or price level specified*[5] |
| <pt> | FX | 1 | Preset type for this preset<br>X = 1 – Volume Preset<br>X = 2 – Money Preset |
| <pput> *[1] | FX | 1 | Price level for this preset<br>X = 4 – Level 1<br>X = 5 – Level 2 |
| <gn> *[2]*[3] | F6 | 1 | Grade group next |

| | | | |
|---|---|---|---|
| <g> *2*3 | EX | 2 | Grade group data |

| Grade Group: 1st Byte | | | |
|---|---|---|---|
| Bit 3 | Bit 2 | Bit 1 | Bit 0 |
| G4 | G3 | G2 | G1 |

| Grade Group 2nd Byte | | | |
|---|---|---|---|
| Bit 3 | Bit 2 | Bit 1 | Bit 0 |
| G8 | G7 | G6 | G5 |

| | | | |
|---|---|---|---|
| <pn> | F8 | 1 | Preset amount next |
| <p> *4 | EX<br>EX<br>EX<br>EX<br>EX | 5 | Preset amount XXXXX<br>X = 0 – 9<br>The least significant digit is transmitted first.<br>The LSD is the LSD of the displayed transaction cash amount at the pump in case of a money preset or a hundredth of a volume unit in case of volume preset. It is the penny digit in the US currency.<br>XXXXX must be greater than or equal to 00010. |
| | | -OR- | |
| <p> *4 | EX<br>EX<br>EX<br>EX<br>EX<br>EX | 6 | Preset amount XXXXXX<br>X = 0 – 9<br>The LSD is the LSD of the displayed amount of the transaction with the decimal position matching the money decimal configuration for money presets.<br>Volume presets are not possible since this field is for a 6 digit money display only.<br>The pump will return an Error status in response to the next poll. XXXXXX must be greater than or equal to 000010 |
| <LRCn> | FB | 1 | LRC check character next |
| <LRC> | EX | 1 | LRC check character<br>X = 0 – F |
| <ETX> | F0 | 1 | End of text |

| | | |
|---|---|---|
| Total Bytes = | 12 | 5 digit mode with no grade or level specified |
| | 13 | 5 digit mode with level specified only |
| | 15 | 5 digit mode with grade specified only |
| | 16 | 5 digit mode with grade and level specified |
| | 13 | 6 digit mode with no grade or level specified*6 |
| | 14 | 6 digit mode with level specified only*6 |
| | 16 | 6 digit mode with grade specified*6 |
| | 17 | 6 digit mode with grade and level specified*6 |

State After Command Acceptance  Same as before command

Refer to section 3.4.4 DATA TO CONSOLE for the line level communication requirement.

*1 Price level is optional for a money preset. The pump will assume the previous transaction price level for this transaction if it is not specified in the block.

*2 The grade 4 or higher grade can be sent only to the pumps that support them. A volume preset requires grade(s) and level in the preset block. The grades may

optionally be sent in a money preset.

*3 The pump will enter ERROR state and ignore the preset data block if the grade or level fields (volume or money preset) do not match a grade or level selection at the pump. The console shall not transmit 'AUTH' command to the pump until the console receives a CALL or OFF response from this pump. Once the price level/grade(s) is accepted by the pump, it shall not allow a delivery from any other level/grade(s).

*4 The comments associated with the Preset Amount are the same for this alternative format as the ones shown for the previous format.

*5 Data length for 6 digit money fields is not available in Modular Advantage pumps or before.

*6 Byte lengths for 6 digit mode field only apply to money preset messages. Volume preset messages will only use 5 digit lengths.

NOTE: The only valid commands that may follow a preset message are status poll, ALL STOP, pump stop, and Authorize after the pump accepts the preset message.

## 4.8.2 STANDARD PRESET DATA

This command sequence allows the console to send the preset amount to the pump. An authorization command code is required to authorize the pump even if the pump accepts the preset data. Note that the pump needs to take all necessary caution to ensure that the preset amount will not be lost or reset before it receives an authorization command. All authorizations for a preset amount/volume should be immediately preceded by a preset data message. Also, once the preset amount is accepted by the pump, it must not allow the customer to set or change price level or grade selection (volume preset only) at the pump. The pumps must support both the money preset and volume preset.

This section only describes the format and contents of the data block. Refer to 3.4.3 DATA TO PUMP for line level protocol information.

Command/Response Syntax
Valid Pump State: OFF, CALL
*Do not issue this command to a pump which has a preset transaction pending (see 3.4.3 DATA TO PUMP).*

Command ’2’ ’<p>’

<p> Pump Number

| | | |
|---|---|---|
| 1 | ---- | Pump 1 |
| 2 | ---- | Pump 2 |
| : | | : |
| F | ---- | Pump 15 |
| 0 | ---- | Pump 16 |

Response ’D’ ’<p>’

<p> Pump Number

Command <DATA BLOCK>

| Symbol | Words | Bytes | Definition |
|---|---|---|---|
| <STX> | FF | 1 | Start of text |
| <DL> | EX | 1 | Data length<br>X = 3 – grade and price level specified<br>X = 5 – price level specified<br>X = 6 – neither grade nor price level specified<br><br>Data Length for 6 Digit Money Modes: *[5]<br>X = 2 – grade and price level specified*[5]<br>X = 4 – price level specified*[5]<br>X = 5 – neither grade nor price level specified*[5] |
| <pt> | FX | 1 | Preset type for this preset<br>X = 1 – Volume Preset<br>X = 2 – Money Preset |
| <pput> *[1] | FX | 1 | Price level for this preset<br>X = 4 – Level 1<br>X = 5 – Level 2 |
| <gn> *[2]*[3] | F6 | 1 | Grade digit next |
| <g> *[2]*[3] | EX | 1 | Grade<br>X = 0 – Grade 1 |

| | | | |
|---|---|---|---|
| | | | \| <br> \| <br> X = F – Grade 16 |
| <pn> | F8 | 1 | Preset amount next |
| <p> *[4] | EX<br>EX<br>EX<br>EX<br>EX | 5 | Preset amount XXXXX<br>X = 0 – 9<br>The least significant digit is transmitted first.<br>The LSD is the LSD of the displayed transaction cash amount at the pump in case of a money preset or a hundredth of a volume unit in case of volume preset. It is the penny digit in the US currency.<br>XXXXX must be greater than or equal to 00010. |
| | | -OR- | |
| <p> *[4] | EX<br>EX<br>EX<br>EX<br>EX<br>EX | 6 | Preset amount XXXXXX<br>X = 0 – 9<br>The LSD is the LSD of the displayed amount of the transaction with the decimal position matching the money decimal configuration for money presets.<br>Volume presets are not possible since this field is for a 6 digit money display only. The pump will return an Error status in response to the next poll. XXXXXX must be greater than or equal to 000010 |
| <LRCn> | FB | 1 | LRC check character next |
| <LRC> | EX | 1 | LRC check character<br>X = 0 – F |
| <ETX> | F0 | 1 | End of text |

| | | |
|---|---|---|
| Total Bytes = | 12 | 5 digit mode with no grade or level specified |
| | 13 | 5 digit mode with level specified only |
| | 15 | 5 digit mode with grade and level specified |
| | 13 | 6 digit mode with no grade or level specified*[6] |
| | 14 | 6 digit mode with level specified only*[6] |
| | 16 | 6 digit mode with grade and level specified*[6] |

State After Command Acceptance Same as before command

Refer to section 3.4.4 DATA TO CONSOLE for the line level communication requirement.

*[1] Price level is optional for a money preset. The pump will assume the previous transaction price level for this transaction if it is not specified in the block.

*[2] The grade 4 or higher grade can be sent only to the pumps that support them. A volume preset requires grade and level in the preset block. The grade must not be sent in a money preset.

*[3] The pump will enter ERROR state and ignore the preset data block if the grade or level fields (volume or money preset) do not match a grade or level selection at the pump. The console shall not transmit 'AUTH' command to the pump until the console receives a CALL or OFF response from this pump. Once the price level/grade is accepted by the pump, it shall not allow a delivery from any other level/grade.

*[4] The dispenser will continue to use a field length of five for volume presets when configured for six digit money.

*[5] Data length 2, 4, 5 is not available in Modular Advantage pumps or before.

*[6] Byte lengths for 6 digit mode field only apply to money preset messages. Volume preset messages will only use 5 digit lengths.

NOTE: The only valid commands that may follow a preset message are status poll, ALL STOP, pump stop, and Authorize after the pump accepts the preset message.

### 4.8.3 PRICE CHANGE DATA

This command sequence allows the console to set the price at the pump. This section describes the format and contents of the data block. Refer to 3.4.3 DATA TO PUMP for protocol level information.

Command/Response Syntax
Valid Pump State: OFF, CALL
*Do not issue this command to a pump which has a preset transaction pending (see note 4, 3.4.3 DATA TO PUMP).*

Command ’2’ ’<p>’

<p> Pump Number

| | | |
|---|---|---|
| 1 | ---- | Pump 1 |
| 2 | ---- | Pump 2 |
| : | | : |
| F | ---- | Pump 15 |
| 0 | ---- | Pump 16 |

Response ’D’ ’<p>’

<p> Pump Number

Command <DATA BLOCK> For price change

| Symbol | Words | Bytes | Definition |
|---|---|---|---|
| <STX> | FF | 1 | Start of text |
| <DL> | E5 | 1 | Data length |
| <pput> | FX | 1 | Price level<br>X = 4 – Level 1<br>X = 5 – Level 2 |
| <gn> | F6 | 1 | Grade digit next |
| <g> *[1] | EX | 1 | Grade<br>X = 0 – Grade 1<br>\|<br>\|<br>X = F – Grade 16 *[1] |
| <ppun> | F7 | 1 | PPU digit next |
| <ppu> | EX<br>EX<br>EX<br>EX | 4 | PPU XXXX<br>X = 0 – 9<br>The least significant digit is transmitted first.<br>The LSD is the LSD of the displayed PPU at the pump. |
| <LRCn> | FB | 1 | LRC check character next |
| <LRC> | EX | 1 | LRC check character<br>X = 0 – F |
| <ETX> | F0 | 1 | End of text |
| Total Bytes = | | 13 | |

| | |
|---|---|
| State After Command Acceptance | Same as before command |

Refer to section 3.4.4 DATA TO CONSOLE for the line level communication requirement.

*[1] The grade 4 or higher grade can be sent only to the pumps that support them.

## 4.8.4 LEVEL CHANGE DATA

This command sequence allows the console to switch the price level at the pump without sending the whole price per unit. This section describes the format and contents of the data block. Refer to 3.4.4 DATA TO CONSOLE for protocol level information.

Command/Response Syntax
Valid Pump State: OFF, CALL
*Do not issue this command to a pump which has a preset transaction pending (see note 4, 3.4.3 DATA TO PUMP).*

Command ’2’ ’<p>’

<p> Pump Number

| | | |
|---|---|---|
| 1 | ---- | Pump 1 |
| 2 | ---- | Pump 2 |
| : | | : |
| F | ---- | Pump 15 |
| 0 | ---- | Pump 16 |

Response ’D’ ’<p>’

<p> Pump Number

Command <DATA BLOCK> For level change

| Symbol | Words | Bytes | Definition |
|---|---|---|---|
| <STX> | FF | 1 | Start of text |
| <DL> | E5 | 1 | Data length |
| <pput> | FX | 1 | Price level<br>X = 4 – Level 1<br>X = 5 – Level 2 |
| <LRCn> | FB | 1 | LRC check character next |
| <LRC> | EX | 1 | LRC check character<br>X = 0 – F |
| <ETX> | F0 | 1 | End of text |
| Total Bytes = | | 6 | |

State After Command Acceptance Same as before command

Refer to section 3.4.4 DATA TO CONSOLE for the line level communication requirement.

## 4.9 SPECIAL FUNCTION COMMAND

Special Functions are an extension to the basic Two-Wire Protocol which allow controllers to communicate with pumps which provide features not anticipated by the initial Two-Wire design. Discussion in this section will be limited to general information concerning the use of Special Functions. Use of specific Special Functions will be discussed under the appropriate section of this document. For example, Special Functions for blending control will be discussed under Blend Ratio Management.

Pumps that do not support Special Functions, or that support only a subset, are not easily identified by an ASC (Authorized Service Contractor) as part of the system installation process. It is more desirable for the controller to automatically determine which pumps on a loop support the Special Functions.

The following procedures are recommended for each Console, CRIND or PAM which will utilize the New Special Functions.

A. To determine if a pump supports the Special Functions. First send a S.F. 001 (version #) command. If the pump does not respond with a version #, it does not support the S.F.s. If it does respond with a version #, send a S.F. 00E (Misc. Pump Data) command to it. If it responds with a configuration data block, the pump supports at least the minimum subset of the Special Functions. If it does not respond or responds with a Null block, it does not support the Special Functions.

B. Use of specific pump software version numbers returned by pumps to determine S.F. support should not occur. This will leave us free to assign any version numbers we choose to new pump models or software releases of existing models.

C. If the pump did not respond to S.F. 001 and you want to determine whether it is a blender, send it S.F 401 (Request for Blend Ratios). All blenders will respond to this S.F., and non-blenders will not.

### Table 6: SPECIAL FUNCTION COMMANDS

| *Category Code* $X_3$ | *Sub Command* $X_2\ X_1$ | *Functions* |
|---|---|---|
| 0 | 0 0 | Reserved |
| 0 | 0 1 | Version Number Request, Software Release Date, Elapsed Operation Time And Power Up Counter Data |
| 0 | 0 E | Miscellaneous Pump Data Request |
| 0 | 0 F | Configuration Data Request |
| 0 | 1 0 | Extended Pump Status |
| | | |
| 4 | 0 1 | Blend Ratio Data Request |
| 4 | 0 5 | Pure Product Totals Request |
| 4 | 1 0 | Specific Error Code |
| 4 | 1 2 | Keyboard Blend Ratio Change Counters Request |

### 4.9.1 COMMAND STRUCTURE

Only the pump with a matched pump ID should send the data requested.

| Command/Response Syntax<br>Valid Pump State<br>OFF, CALL, AUTH - for commands sending data to the pump<br>OFF, CALL, AUTH, STOP, FEOT/PEOT for commands requesting data | | | | |
|---|---|---|---|---|
| Command | ’2’ ’<p>’ | | | |
| | <p> | Pump Number | | |
| | | 1 | ---- | Pump 1 |
| | | 2 | ---- | Pump 2 |
| | | : | | : |
| | | F | ---- | Pump 15 |
| | | 0 | ---- | Pump 16 |
| Response | ’D’ ’<p>’ | | | |
| | <p> | Pump Number | | |
| Command | <DATA BLOCK> | | | |
| Symbol | Words | Bytes | | Definition |
| <STX> | FF | 1 | | Start of text |
| <DL> | EX | 1 | | Data length<br>X = 0 - F |
| <sf> | FE | 1 | | Special Function Command next |
| <sfc> | EX1<br>EX2<br>EX3 | 3 | | Special Function Command<br>See Table 6 for details<br>$X_1$, $X_2$, $X_3$ = 0 – F<br>$X_3$ = Category code<br>$X_2$, $X_1$ = Sub command |
| <m> | EX | Up to 32 | | The message of the command up to 32 words |
| <LRCn> | FB | 1 | | LRC check character next |
| <LRC> | EX | 1 | | LRC check character<br>X = 0 – F |
| <ETX> | F0 | 1 | | End of text |
| Total Bytes = | | Up to 41 | | |
| Response | <DATABLOCK> | | | |
| State After Command Acceptance | | The pump should not change the STATE unless other events occur, such as pump handle position, during transmission. | | |

Refer to section 3.4.4 DATA TO CONSOLE for the line level communication requirement.

NOTE 1: Special Function Commands which have no data to transmit to the pump use the above format with <m> omitted and a total message size of 9 bytes.

NOTE 2: If the pump does not return Send Data, the controller must not send the Data Block regardless of the status of the pump prior to the initiation of the exchange.

## 4.9.2 RESPONSE STRUCTURE

This section defines some general rules, which are used throughout the Special Function Responses.

A. All data blocks must conform to the Data Block format, which will be described in the 4.9.2.1 Data Block Format.

B. In general, Most Significant Digit (MSD) will be transmitted first unless otherwise specified.

C. All words are transmitted in a modified ASCII format. The most significant bit of the word is set to 1. The following table describes the words.

| *Transmitted Code* | *ASCII Code* | *Interpretation* |
|---|---|---|
| BA | 3A | Colon, Data Length Next |
| B0 – B9 | 30 – 39 | Digit 0 – 9 |
| C1 – C6 | 41 – 46 | Letter A – F |
| 8D | 0D | Carriage Return |
| 8A | 0A | Line Feed |

D. The numbers are transmitted in BCD format unless otherwise specified.

E. Block length equals the number of words in the message (not including overhead, refer to 4.9.2.1 Data Block Format) divided by 2.

F. Check sum is 2's complement of summation of all translated words in the Data Block excluding Block - Length Next control word and Carriage Return and Line Feed. (Refer to 4.9.2.1 Data Block Format for Data Block Format). The translated words are the hexadecimal representation of two consecutive words in the data block starting with MSD of Data Length (see examples).

| *Data Field* | *Data on the Block* | *Translated Words* |
|---|---|---|
| Block Length Next | BA | (not included in checksum) |
| Block Length MSD | B0 | 01 |
| Block Length LSD | B1 | |
| Pump Number | B2 | 23 |
| Category Command (SF3) | B3 | |
| MS Command Nibble (SF2) | B4 | 45 |
| LS Command Nibble (SF1) | B5 | |
| Remaining Block Count MSD | B6 | 67 |
| Remaining Block Count LSD | B7 | |
| 1st Byte Data | B8 | 89 |
| 2nd Byte Data | B9 | |
| Checksum MSD | C1 | A7 |
| Checksum LSD | B7 | |
| Carriage Return | 8D | (not included in checksum) |
| Line Feed | 8A | (not included in checksum) |

The modulo 100h sum of the translated bytes is 59h
The twos complement of 59h is A7h

G. Data Block is a block of words (data) including the Block Length, pump number, echoed commands, block number, message, Carriage Return, and Line Feed.

H. In general, the counters are cumulative and can be reset only at the Master RAM Reset, unless otherwise specified.

I. Pump shall respond with null data block, instead of the requested data, under a number of conditions as follows:

- Pump receives a valid Special Function but it is not supported by the pump's version of software. (i.e.; blender Special Function sent to non-blender pump)
- Pump receives a valid special function but some data in the data block is invalid (out of range; i.e.; memory address or incorrect configuration data

J. Supported Special functions with valid data, which cannot be executed for any reason unrelated to the command content will cause the pump to respond with an Error status to the next poll.

K. Under certain conditions the pump is allowed to ignore Special Function Commands (e.g.; blend ratios sent to a pump whose ratios are not programmable). The conditions are indicated in the sections dealing with the relevant Commands/Responses. Ignore means a Null Block will not be returned in response to the Command and an Error Status will not be returned in response to the poll following the Special Function Command which has been ignored.

### 4.9.2.1 Data Block Format

This section describes the format of the Response Data Block. The response to a Special Function Command must conform to this format.

| Symbol | Words | Bytes | Definition |
|---|---|---|---|
| <pLn> | BA | 1 | Block Length Next |
| <pL> | XY<br>XY | 2 | Block Length<br>XY = B0 – B9 or C1 – C6<br>MSD transmitted first |
| <pn> | XY | 1 | Pump Number<br>XY = B0 – B9 or C1 – C6<br>B0 ---- Pump 16<br>B1 ---- Pump 1<br>: :<br>C1 ---- Pump 10<br>: :<br>C6 ---- Pump 15 |
| <sfc> | SF1<br>SF2<br>SF3 | 3 | Special Function Command Code<br>Code is echoed from request<br>SF1, SF2, SF3 = B0 – B9 or C1 – C6<br>SF3 = Modified ASCII of $X_3$<br>SF2 = Modified ASCII of $X_2$,<br>SF1 = Modified ASCII of $X_1$ |
| <bn> | XY<br>XY | 2 | Remaining Blocks<br>Represents the remaining blocks to be transmitted for this request<br>XY = B0 – B9 or C1 – C6<br>MSD is transmitted first<br>The last block will have B0 B0 |
| <m> | Message up to 32 bytes | | The message of the response.<br>An even number of bytes (2, 4, 6, etc) up to 32. |
| <CS> | XY1<br>XY2 | 1 | Checksum<br>XY1 = B0 – B9<br>XY2 = C1 – C6<br>MSD transmitted first |
| <cr> | 8D | 1 | Carriage Return |
| <lf> | 8A | 1 | Line Feed |

NOTE: Modular Advantage non-blenders and some early Advantage non-blending pumps will respond to a S.F. 001 with a data block which contains an additional character (F0). This character is after the line feed which should be the last character (e.g. 8A F0)

## 4.9.3 This section describes the response messages in the Data Block.

### 4.9.3.1 The Definition of a Null Block

| *Word* | *Interpretation* | *Comments* |
|---|---|---|
| | No Data <m> in the data block | |

When a pump responds with a Null Block to a S.F. Command, it means one of two things:

- The Command is not supported by this pump
- Some or all of the data in the data block is invalid

When a controller receives a Null Block from a pump, there is no point in resending the S.F. Command.

## 4.9.4 Command and Response Structure

### 4.9.4.1 Version Number Request ($X_3X_2X_1$ = 001)

| Response Format | | |
|---|---|---|
| *Word* | *Interpretation* | *Comments* |
| BX | Software Version Number | Version number MS digit (X = 0 – 9) |
| BX | (C6 is a blank) | Version Number Digit (X = 0 – 9) |
| BX | | Version Number Digit (X = 0 – 9) |
| BX | | Version Number LS Digit (X = 0 – 9) |
| BX | Software Release Date | Month 10's Digit (X = 0 – 1) |
| BX | | Month 1's Digit (X = 0 – 9) |
| BX | | Day 10's Digit (X = 0 – 3) |
| BX | | Day 1's Digit (X = 0 – 9) |
| BX | | Year 10's Digit (X = 0 – 9) |
| BX | | Year 1's Digit (X = 0 – 9) |
| BX | Accumulated operation time since last | 100000 Hours Digit (X = 0 – 9) |
| BX | cold start | 10000 Hours Digit (X = 0 – 9) |
| BX | | 1000 Hours Digit (X = 0 – 9) |
| BX | | 100 Hours Digit (X = 0 – 9) |
| BX | | 10 Hours Digit (X = 0 – 9) |
| BX | | 1 Hours Digit (X = 0 – 9) |
| BX | | 10 Minutes Digit (X = 0 – 9) |
| BX | | 1 Minutes Digit (X = 0 – 9) |
| BX | Accumulated number of warm starts | 1000's Digit (X = 0 – 9) |
| BX | | 100's Digit (X = 0 – 9) |
| BX | | 10's Digit (X = 0 – 9) |
| BX | | 1's Digit (X = 0 – 9) |

Note: Epsilon units will send the last 4 digits of their 8 digit software version number.

### 4.9.4.2 Miscellaneous Pump Data Request ($X_3X_2X_1$ = 00E)

| Response Format | | |
|---|---|---|
| *Word* | *Interpretation* | *Comments* |
| BX | Unit Type Code | MSD (X = 0 – 9) |
| BX | Unit Type Code | LSD (X = 0 – 9) |
| BX | Conversion Factor Code | MSD (X = 0 – 9) |
| BX | Conversion Factor Code | LSD (X = 0 – 9) |
| BX | Reserved | Reserved |
| BX | Reserved | Reserved |
| BX | Reserved | Reserved |
| BX | Reserved | Reserved |
| BX | Reserved | Reserved |
| BX | Reserved | Reserved |
| BX | 5/6 Digit Money Mode | MSD (X = 0 – 9) |
| BX | 5/6 Digit Money Mode | LSD (X = 0 – 9) |
| BX | Auto On/Push to Start Mode | MSD (X = 0 – 9) |
| BX | Auto On/Push to Start Mode | LSD (X = 0 – 9) |

Miscellaneous Pump Data available from the pump:

| Unit Type Code |
|---|
| Refer to the Unit Type Code definition in Section 6:  Pump Identification. |

| Conversion Factor Code/Volume Unit Type for ECAL | | |
|---|---|---|
| 00 | | Not Programmed |
| 01 | | U. S. Gallons |
| 02 | | UK Gallons (Imperial Gallons) |
| 03 | | Liters |
| 04 | | 1012 pulse/gallon (Hawaii) |

| 5/6 Digit Money Mode | | |
|---|---|---|
| 00 | | 5 digits |
| 01 | | 6 digits |

| Auto-On/Push-To-Start | | |
|---|---|---|
| 00 | | Not Push-To-Start |
| 01 | | Auto-On/Push-To-Start |

### 4.9.4.3 Configuration Request ($X_3X_2X_1$ = 00F)

| Command Format | | |
|---|---|---|
| *Word* | *Interpretation* | *Comments* |
| BX | Command Code MSD | Received in Command |
| BX | Command Code LSD | |
| | | |
| XY | Data MSD first | XY = B0 – B9 or C1 – C6 |
| XY | Up to 14 Bytes | |
| : | | |
| XY | | |

Note: This Special Function may not be implemented in all pumps and dispensers for all features.

### 4.9.4.4 Extended Pump Status ($X_3X_2X_1$ = 010)

| Response Format | | |
|---|---|---|
| *Word* | *Interpretation* | *Comments* |
| XY | Command Code MSD | B0 Standard<br>XY = B0- B9<br>XY = C1 – C6 Extended More-Buttons |
| BX | Price Level Selection Needed | 0 = Needed<br>1 = Not Needed |
| BX | Grade Selection Needed | 0 = Needed<br>1 = Not Needed |
| BX | Pump Handle / Nozzle Operation | 0 = Off/In<br>1 = On/Out |
| BX | Push-To-Start Needed | 0 = Needed<br>1 = Not Needed |
| BX | Selected Grade | 0 = Unknown<br>1 – F = Grade Digit |

#### 4.9.4.4.1 Extended Status Information

Post-Modular pumps provide Extended Status information for use by the controller (S.F. 010). This Information allows the controller to determine more information about the customer's operation of the dispenser for display of additional, perhaps more accurate, information to the console operator or pump user.

True means operation not needed. False means operation needed. All operations are pump related. For example, if the controller has selected the price level then it will have no effect on the Extended Pump Status. Only selecting price level at the pump can do that. Any option not present at the pump (e.g., Push-To-Start) will default to Not Needed. An Auto-On SHMPD with Cash/Credit could set Price Level Selection Needed, Grade Selection Needed and Push-To-Start needed all to True when in the CALL state. Subsequently pressing one of the price level buttons would set all 3 flags to False.

The information available in the Extended Status includes:

Price Level Selection Needed

Selection of Price Level may be independent of the safety Second Step. This merely indicates whether a price level needs to be selected, not which one.

Grade Selection Needed

Grade selection may occur independent of the safety Second Step. Once it has occurred, the grade information will be available also.

Pump Handle/Nozzle Operation

Under certain conditions the controller may not be able to determine the handle/nozzle position from the response to the Status Poll (e.g. ERROR status). This information may be important for user/operator prompting.

Push-To-Start Needed

This indicates whether the Push-To-Start button has been pressed. This too is independent of the safety Second Step. For example, if the Push-To-Start status is true and the Pump Handle is IN, it may indicate the user is confused about the proper sequence for pump activation.

Selected Grade

In addition the Extended Pump Status tells the console which grade has been selected. This is in the form of a grade number which is the same grade number used in other Two-Wire messages. The Grade Selected will always be non-zero once a grade is selected by the customer, even if that grade is not available. For example, if a pump is preset for grades 1 or 3 (only) and the customer selects grade 2 the Selected Grade will be 2 but the Grade Selection Needed flag will still be True since a grade that can be authorized has not yet been selected, the value of this field is zero or “unknown.”

### 4.9.4.5 Blend Ratio Data Request ($X_3X_2X_1$ = 401)

| Response Format | | |
|---|---|---|
| *Word* | *Interpretation* | *Comments* |
| BX | Grade 1 Blend Ratio MSD | X = 0 – 1 |
| BX | Grade 1 Blend Ratio Digit | X = 0 – 9 |
| BX | Grade 1 Blend Ratio LSD | X = 0 – 9 |
| BX | Grade 2 Blend Ratio MSD | X = 0 – 1 |
| BX | Grade 2 Blend Ratio Digit | X = 0 – 9 |
| BX | Grade 2 Blend Ratio LSD | X = 0 – 9 |
| BX | Grade 3 Blend Ratio MSD | X = 0 – 1 |
| BX | Grade 3 Blend Ratio Digit | X = 0 – 9 |
| BX | Grade 3 Blend Ratio LSD | X = 0 – 9 |
| BX | Grade 4 Blend Ratio MSD | X = 0 – 1 |
| BX | Grade 4 Blend Ratio Digit | X = 0 – 9 |
| BX | Grade 4 Blend Ratio LSD | X = 0 – 9 |
| BX | Grade 5 Blend Ratio MSD | X = 0 – 1 |
| BX | Grade 5 Blend Ratio Digit | X = 0 – 9 |
| BX | Grade 5 Blend Ratio LSD | X = 0 – 9 |
| BX | Grade 6 Blend Ratio MSD | X = 0 – 1 |
| BX | Grade 6 Blend Ratio Digit | X = 0 – 9 |
| BX | Grade 6 Blend Ratio LSD | X = 0 – 9 |

This data block contains the Grade 1 - Grade 6 blend ratio that will be used for the next transaction.

### 4.9.4.6 Pure Product Totals Request 405 ($X_3X_2X_1$ = 405)

| Response Format | | |
|---|---|---|
| *Word* | *Interpretation* | *Comments* |
| BX | Blend Manifold 1 Pure product total for high product (volume) | 1000000 Digit X = 0 – 9 |
| BX | | 100000 Digit X = 0 – 9 |
| BX | | 10000 Digit X = 0 – 9 |
| BX | | 1000 Digit X = 0 – 9 |
| BX | | 100 Digit X = 0 – 9 |
| BX | | 10 Digit X = 0 – 9 |
| BX | | 1 Digit X = 0 – 9 |
| BX | | .1 Digit X = 0 – 9 |
| BX | | .01 Digit X = 0 – 9 |
| BX | | .001 Digit X = 0 – 9 |
| BX | Blend Manifold 1 Pure product total for low product (volume) | 1000000 Digit X = 0 – 9 |
| BX | | 100000 Digit X = 0 – 9 |
| BX | | 10000 Digit X = 0 – 9 |
| BX | | 1000 Digit X = 0 – 9 |
| BX | | 100 Digit X = 0 – 9 |
| BX | | 10 Digit X = 0 – 9 |
| BX | | 1 Digit X = 0 – 9 |
| BX | | .1 Digit X = 0 – 9 |
| BX | | .01 Digit X = 0 – 9 |
| BX | | .001 Digit X = 0 – 9 |
| BX | Blend Manifold 2 Pure product total for high product (volume) | 1000000 Digit X = 0 – 9 |
| BX | | 100000 Digit X = 0 – 9 |
| BX | | 10000 Digit X = 0 – 9 |
| BX | | 1000 Digit X = 0 – 9 |
| BX | | 100 Digit X = 0 – 9 |
| BX | | 10 Digit X = 0 – 9 |

| | | |
|---|---|---|
| BX | | 1 Digit<br>X = 0 – 9 |
| BX | | .1 Digit<br>X = 0 – 9 |
| BX | | .01 Digit<br>X = 0 – 9 |
| BX | | .001 Digit<br>X = 0 – 9 |
| BX | Blend Manifold 2 | 1000000 Digit |
| | Pure product total for low product | X = 0 – 9 |
| BX | (volume) | 100000 Digit<br>X = 0 – 9 |
| BX | | 10000 Digit<br>X = 0 – 9 |
| BX | | 1000 Digit<br>X = 0 – 9 |
| BX | | 100 Digit<br>X = 0 – 9 |
| BX | | 10 Digit<br>X = 0 – 9 |
| BX | | 1 Digit<br>X = 0 – 9 |
| BX | | .1 Digit<br>X = 0 – 9 |
| BX | | .01 Digit<br>X = 0 – 9 |
| BX | | .001 Digit<br>X = 0 – 9 |

**NOTES:**

1. This response violates the maximum 32 bytes of data in <m> limit specified in Data Block Format of the Gilbarco Dispenser Two-Wire Protocol for Third Party Pump Controllers.
2. This function is only valid for pure products of standard single blend manifold blenders unit types. The field for blend manifold 2 is optional and may be omitted in responses. Some dispensers may respond with blend manifold 2 values populated with zeroes.
3. For Unit Type Code 46, 47, 48 and 49 (2+2 and 3+2 blenders with two sets of blend manifolds), pure product totals for blend manifold 1 are for the corresponding pure products for the standard blender portion of the dispenser. The pure products for blend manifold 1 are the only pure products in a response for a SF405 request. Special Function 605 (detailed in Gilbarco Dispenser Two-Wire Protocol for Third Party Pump Controllers – Extended Two-Wire Appendix) is required for all products totals for these dispenser types.

### 4.9.4.7 Specific Error Code Request ($X_3X_2X_1$ = 410)

| Response Format | | |
|---|---|---|
| *Word* | *Interpretation* | *Comments* |
| BX | Grade 1 Error Code MSD | X = 0 – 9 |
| BX | Grade 1 Error Code LSD | X = 0 – 9 |
| BX | Grade 2 Error Code MSD | X = 0 – 9 |
| BX | Grade 2 Error Code LSD | X = 0 – 9 |
| BX | Grade 3 Error Code MSD | X = 0 – 9 |
| BX | Grade 3 Error Code LSD | X = 0 – 9 |
| BX | Grade 4 Error Code MSD | X = 0 – 9 |
| BX | Grade 4 Error Code LSD | X = 0 – 9 |
| BX | Grade 5 Error Code MSD | X = 0 – 9 |
| BX | Grade 5 Error Code LSD | X = 0 – 9 |
| BX | Grade 6 Error Code MSD | X = 0 – 9 |
| BX | Grade 6 Error Code LSD | X = 0 – 9 |
| BX | Specific Blend Error Code MSD | Error in current transaction<br>X = 0 – 9 |
| BX | Specific Blend Error Code LSD | Error in current transaction<br>X = 0 – 9 |

### 4.9.4.8 Error Codes

Errors are reported in the data response as a sequence of 9 bits concatenated from 3 nibbles of the response data Eh, Ea, Eb. The bits used are h4-h1, b1 and a4-a1 presenting a nine bit pattern as follows:

| Hose | Error |
|---|---|
| h4, h3, h2, h1 | b1, a4, a3, a2, a1 |

A non-zero value in the Hose number portion indicates the presence of an error code in the error portion. A zero value for the Hose number means bits b1 and a4-a1 are to be ignored by the Console. The binary value of b1, a4, a3, a2, and a1 plus 19 results in the decimal error code displayed by the pump. For example, if the binary value of the error bits is 24, then add 19 to get the displayed error of 43 or General Error. This is a list of the current decimal error codes and their meanings returned by Post Modular, Modular and some pre-modular pumps.

**Table 7: PUMP ERROR CODES (SF410)**

| Specific Code | E300<br>Optimized Advantage<br>Modular Advantage<br>Legacy<br>Performer | Pre-Modular |
|---|---|---|
| 01 | Communication Timeout | N/A |
| 02 | Blend Controller Busy | N/A |
| 03 | Communication Limit | N/A |
| 04 | Communication Limit | N/A |
| 05 | Received Data Error | N/A |
| 06 | Received Block Error | N/A |
| 07 | Blends Not Programmed | N/A |
| 11 | Lo Product Low Flow | N/A |
| 12 | Hi Product Low Flow | N/A |

| | | |
|---|---|---|
| 13 | Lo Product No Flow | N/A |
| 14 | Hi Product No Flow | N/A |
| 15 | Hi Contaminated by Lo | N/A |
| 16 | Lo Contaminated by Hi | N/A |
| 17 | Lo Product Surge | N/A |
| 18 | Hi Product Surge | N/A |
| 19 | Undefined Surge | N/A |
| 20 | No Power to Hi Valve | N/A |
| 21 | No Power to Lo Valve | N/A |
| 22 | No Power to Valves | N/A |
| 23 | Pulser Fail | N/A |
| 24 | Invalid Command | N/A |
| 25 | Maximum Pulse Lag Exceeded | N/A |
| 26 | Lost Communication | N/A |
| 27 | Software Stack Error | N/A |
| 28 | BPC Too Slow in Removing Pulses | N/A |
| 29 | Leaky Valve | N/A |
| 51 | Grade 1 Leaks (On State) *[1] | N/A |
| 52 | Grade 2 Leaks (On State) *[1] | N/A |
| 53 | Grade 3 Leaks (On State) *[1] | N/A |
| 54 | Grade 4 Leaks (On State) *[1] | N/A |
| 55 | G1/G2 Leaks (Off State) *[1] | N/A |
| 56 | G1/G3 Leaks (Off State) *[1] | N/A |
| 57 | G1/G4 Leaks (Off State) *[1] | N/A |
| 58 | G2/G3 Leaks (Off State) *[1] | N/A |
| 59 | G2/G4 Leaks (Off State) *[1] | N/A |
| 60 | G3/G4 Leaks (Off State) *[1] | N/A |

*[1] Not Available in Modular Advantage Dispensers

### 4.9.4.9 Blend Ratio Change Counters Request ($X_3X_2X_1$ = 412)

| Response Format | | |
|---|---|---|
| *Word* | *Interpretation* | *Comments* |
| BX | Grade 1 blend ratio change | 1000's Digit (X=0-9) |
| BX | Counter from pump keyboard | 100's Digit (X=0-9) |
| BX | | 10's Digit (X=0-9) |
| BX | | 1's Digit (X=0-9) |
| BX | Grade 2 blend ratio change | 1000's Digit (X=0-9) |
| BX | Counter from pump keyboard | 100's Digit (X=0-9) |
| BX | | 10's Digit (X=0-9) |
| BX | | 1's Digit (X=0-9) |
| BX | Grade 3 blend ratio change | 1000's Digit (X=0-9) |
| BX | Counter from pump keyboard | 100's Digit (X=0-9) |
| BX | | 10's Digit (X=0-9) |
| BX | | 1's Digit (X=0-9) |
| BX | Grade 4 blend ratio change | 1000's Digit (X=0-9) |
| BX | Counter from pump keyboard | 100's Digit (X=0-9) |
| BX | | 10's Digit (X=0-9) |
| BX | | 1's Digit (X=0-9) |
| BX | Grade 5 blend ratio change | 1000's Digit (X=0-9) |
| BX | Counter from pump keyboard | 100's Digit (X=0-9) |
| BX | | 10's Digit (X=0-9) |
| BX | | 1's Digit (X=0-9) |
| BX | Grade 6 blend ratio change | 1000's Digit (X=0-9) |
| BX | Counter from pump keyboard | 100's Digit (X=0-9) |
| BX | | 10's Digit (X=0-9) |
| BX | | 1's Digit (X=0-9) |

# 5. PUMP STATUS TRANSITION(S)

## 5.1 Pump State Diagram

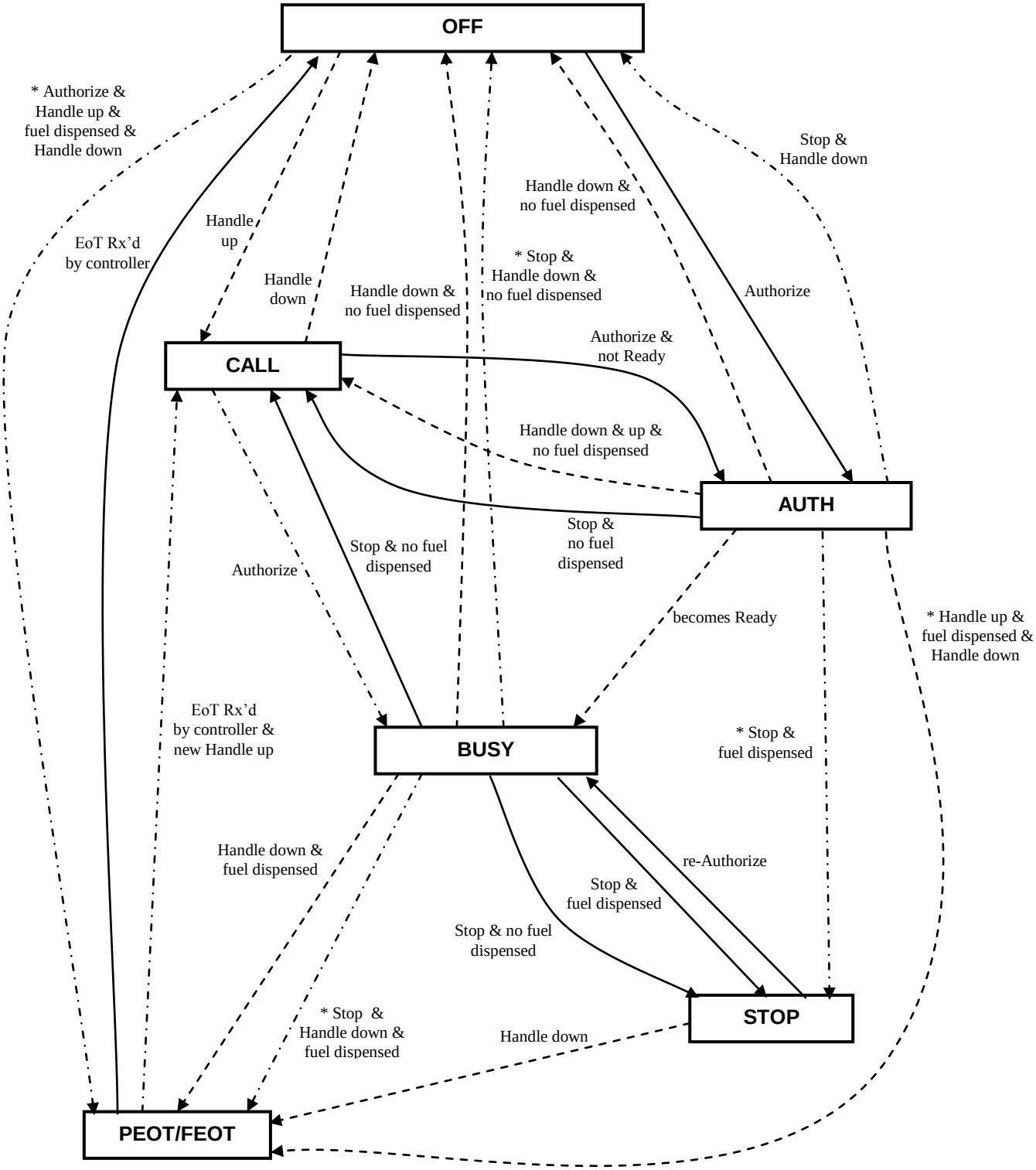
In the following diagram, pump or controller initiated status transitions are shown. The notation ⟶ indicates a controller initiated transition and - - - ▶ indicates a pump initiated transition. A line of - · - ▶ indicates that both an action at the pump and a controller command are needed for this transition to occur. These may occur simultaneously from the controller's view and the key factor governing some of these transitions will be whether fuel was dispensed. Transitions to and from the ERROR status are omitted since they are all pump initiated and can occur from and to any other status.

These transitions represent the sequence of external status responses the controller may see based on conditions and actions occurring at a pump.

"EoT" represents the End-of-Transaction status. "Authorize" is the Authorize command from the controller. "Stop" is the Pump Stop command from the controller. "Ready" indicates that all conditions have been met for beginning a sale (price/grade selection, handle up, valid grade, etc.). "Not Ready" implies that some of these conditions have not been met.

| | |
|---|---|
| * Note 1: | Transitions marked with asterisks are unlikely to be seen by the controller except when status polling is very infrequent. |
| Note 2: | Transitions 5 (Call to Off) and 6 (Auth to Off) could occur when fuel has been dispensed for some pre-modular pumps (refer to Sale Completion section). |
| Note 3: | For most pumps, Transition 7 (Busy to Off) occurs when no fuel has been dispensed. |
| Note 4: | Transition 10 (Call to Auth) is deprecated. Pumps should transition immediately to Busy once Authorized from Call (Transition 11). Controllers should be aware that some pumps will nevertheless go from Call via Auth to Busy (Transition 10 followed by 15). |
| Note 5: | Transitions 13 (Busy to Call) & 19 (Busy to Stop) are alternatives. When no fuel has been dispensed, some pumps will return to the Call state, others will go to the Stop state the same as if fuel had been dispensed (Transition 18). |

### 5.1.1 PUMP STATUS DIAGRAM

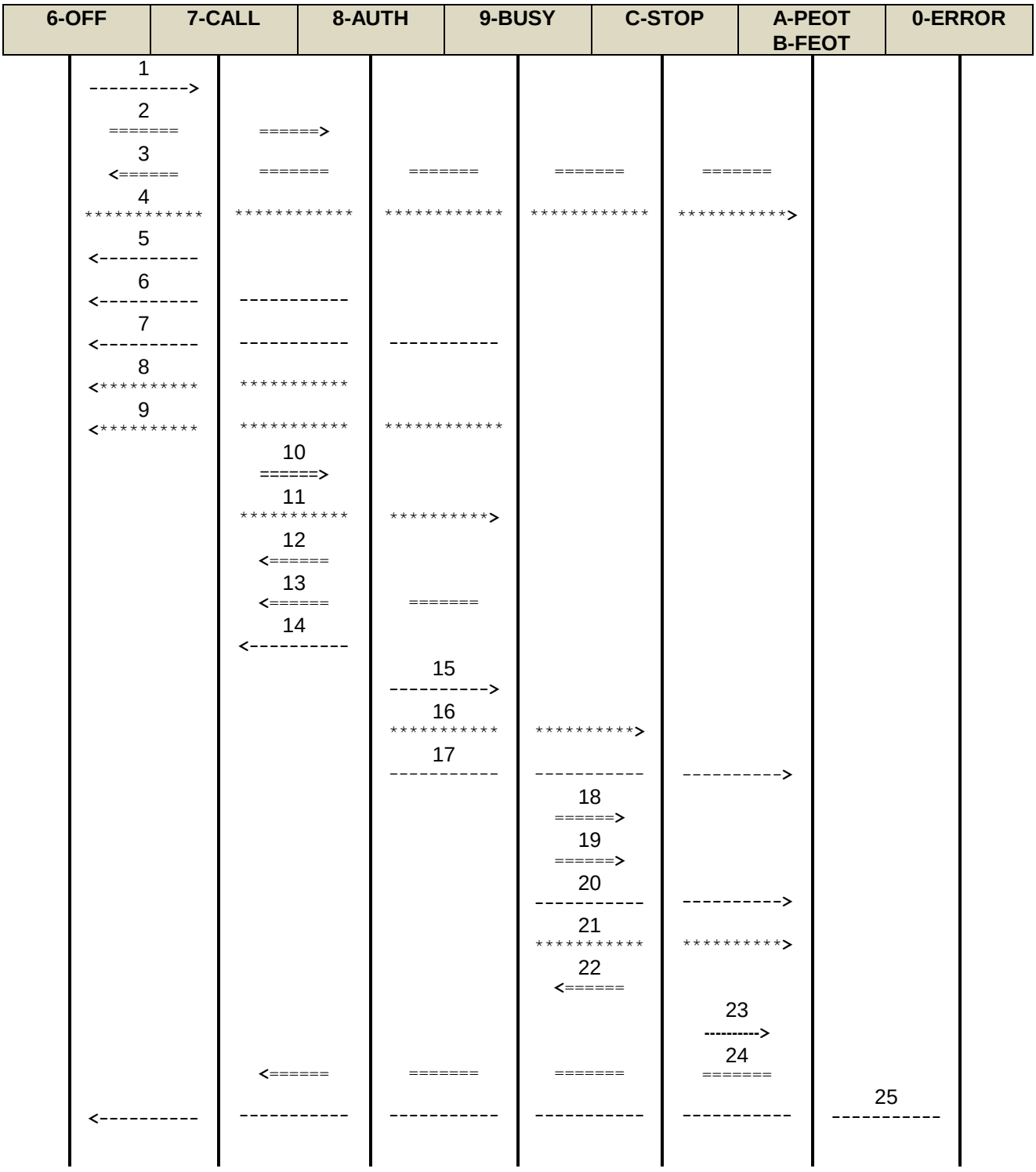


### 5.1.2 Pump Status Transition Table

In the following diagram, pump or controller initiated status transitions are shown. The notation "**==>**" indicates a controller initiated transition and "**-->**" indicates a pump initiated transition. A line of "******>**" indicates that both an action at the pump and a controller command are needed for this transition to occur. These may occur simultaneously from the controller's view and the key factor governing some of these transitions will be whether fuel was dispensed. Transitions to the ERROR status are omitted since they are all pump initiated and can occur from any other status.

These transitions represent the sequence of external status responses the controller may see based on conditions and actions occurring at a pump. Each transition is numbered and the corresponding number in the Transition Explanation list, explains the reason for the transition.

**Table 8: PUMP STATE TRANSITION TABLE**




Legend:

| | |
|---|---|
| ======= | Controller Initiated |
| ---------- | Pump Initiated |
| ********** | Pump or Controller Initiated |

### 5.1.3 State Transition Explanation

1. Customer initiated fuel sale request.
2. Authorization received by pump from controller.
3. Completed sale status received by controller.
4. * Pump authorized by controller, handle up, customer dispenses some fuel, and then terminates transaction.
5. Customer or pump terminated a fuel sale with fuel being dispensed.
6. * Customer or pump terminated a fuel sale with fuel being dispensed.
7. * Customer or pump terminated a fuel sale with fuel being dispensed.
8. STOP issued by controller, no fuel dispensed and pump no longer requesting authorization.
9. STOP issued by controller, no fuel dispensed and pump no longer requesting authorization.
10. Pump authorization by controller, not all conditions met for beginning of transaction.
11. Pump authorized by controller, all conditions met for beginning of transaction.
12. Pump stopped by controller, but still requesting authorization.
13. Pump stopped by controller, but still requesting authorization (no fuel dispensed).
14. Customer or pump discontinued a fuel sale without fuel being dispensed (e.q. handle down then immediately back up).
15. All prerequisites for beginning dispensing have been completed (i.e. price selection, handle up).
16. * Pump is authorized then stopped by controller with no intervening status polls and customer dispensed some fuel before stop occurred.
17. * Handle up, customer dispenses some fuel, and then terminates the transaction.
18. Controller issues STOP and customer has not terminated transaction (fuel has been dispensed).
19. Pump stopped by controller, and customer has not terminated transaction (no fuel dispensed).
20. Customer terminated fuel transaction.
21. * Controller issues STOP, fuel was dispensed and customer has terminated fuel transaction.
22. Controller re-authorized pump before customer terminated fuel transaction.
23. Customer terminated fuel transaction while pump was stopped.
24. Controller registers transaction and next customer is initiating fuel transaction.
25. A repaired pump (could also go to CALL)

| | |
|---|---|
| * NOTE 1: | Transitions marked with asterisks are unlikely to be seen by the controller except when status polling is very infrequent. |
| NOTE 2: | Transitions 5 (Call to Off) and 6 (Auth to Off) could occur when fuel has been dispensed for some pre-modular pumps (refer to Sale Completion section). |
| NOTE 3: | Transition 10 is deprecated. Pumps should transition immediately to Busy once Authorized from Call (Transition 11). Controllers should be aware that some pumps will nevertheless go from Call via Auth to Busy (Transition 10 followed by 15). |
| NOTE 4: | Transitions 13 (Busy to Call) & 19 (Busy to Stop) are alternatives. When no fuel has been dispensed, some pumps will return to the Call state, others will go to the Stop state the same as if fuel had been dispensed (Transition 18). |

# 6. Pump Identification

The console needs to identify the pump's functional type and make certain it agrees with the console's current site configuration "map". There are two methods for soliciting functional type data via the Two-Wire. The first way is to use the Request for Transaction Data command. The Pump Identifier field contained in the pump's response has some basic functional information about the pump including blender versus non-blender and an indication of the maximum number of grades (3 or 6). Not all pump models and software groups support this method, for example, Multiline models do not.

If more precise information is required, some pump models/software groups support a Special Function called Miscellaneous Pump Data Request which contains more specific pump configuration information. This method is only supported by Post-Modular pump models including Advantage, Encore E300, Dimension, Euroline and Euro-Advantage. Refer to the section on Special Functions to find out how a console can determine if a pump supports this function.

Miscellaneous Pump Data available from the pump

| Unit Type | Configuration (see details in Special Function 00E in Section 4.9.4.2) | T1[*1] | T2 | T3 | T4 | T5 | T6 |
|---|---|---|---|---|---|---|---|
| 10 | MPD with 1 grade/hose per side | G1[*2] | | | | | |
| 11 | MPD with 2 grade/hose per side | G1 | G2 | | | | |
| 12 | MPD with 3 grade/hose per side | G1 | G2 | G3 | | | |
| 13 | MPD with 4 grade/hose per side | G1 | G2 | G3 | G4 | | |
| 14 | MPD with 5 grade/hose per side | G1 | G2 | G3 | G4 | G5 | |
| 15 | MPD with 6 grade/hose per side | G1 | G2 | G3 | G4 | G5 | G6 |
| 20 | Single Hose MPD 2+0 (2 grades, 1 hose/side) | G1 | G2 | | | | |
| 21 | Single Hose MPD 2+1 (3 grades, 2 hoses/side) | G1 | G2 | G3 | | | |
| 22 | Single Hose MPD 3+0 (3 grades, 1 hose/side) | G1 | G2 | G3 | | | |
| 23 | Single Hose MPD 3+1 (4 grades, 2 hoses/side) | G1 | G2 | G3 | G4 | | |
| 28 | Super-High Flow (SHF) (2 meters, 1 hose/side) | G1 | | | | | |
| 30 | Blender 3+0 (3 blends, 3 hoses/side) | G1 | G2 | G3 | | | |
| 30 | Blender 3+1 (3 blends + 1 non-blend, 4 hoses/side) | G1 | G2 | G3 | | | G6 |
| 40 | Blender 2+0 (2 blends, 1 hose/side) | G1 | G2 | | | | |
| 40 | Blender 2+1 (2 blends + 1 non-blend, 2 hoses/side) | G1 | G2 | | | | G6 |
| 40 | Blender 3+0 (3 blends, 1 hose/side) | G1 | G2 | G3 | | | |
| 40 | Blender 3+1 (3 blends + 1 non-blend, 2 hoses/side) | G1 | G2 | G3 | | | G6 |
| 40 | Blender 4+0 | G1 | G2 | G3 | G4 | | |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| | (4 blends, 1 hose/side) | | | | | | |
| 40 | Blender 4+1<br>(4 blends + 1 non-blend, 2 hoses/side) | G1 | G2 | G3 | G4 | | G6 |
| 40 | Blender 5+0<br>(5 blends, 1 hose/side) | G1 | G2 | G3 | G4 | G5 | |
| 40 | Blender 5+1<br>(5 blends + 1 non-blend, 2 hoses/side) | G1 | G2 | G3 | G4 | G5 | G6 |
| 41 | Blender 2+1+1<br>(2 blends + 2 non-blends, 3 hoses/side) | G1 | G2 | G3[*5] | | | G6 |
| 42 | Blender 3+1+1<br>(3 blends + 2 non-blends, 3 hoses/side) | G1 | G2 | G3 | G4[*5] | | G6 |
| 43[*3] | Blender 2 + Single-Hose MPD 2<br>(2 blends + 2 non-blends, 2 hoses/side) | G1 | G2 | G3[*6] | | | G6 |
| 44[*3] | Blender 3 + Single-Hose MPD 2<br>(3 blends + 2 non-blends, 2 hoses/side) | G1 | G2 | G3 | G4[*6] | | G6 |
| 45[*3] | Blender 4 + Single-Hose MPD 2<br>(4 blends + 2 non-blends, 2 hoses/side) | G1 | G2 | G3 | G4 | G5[*6] | G6 |
| 46[*4] | Blender 2+2<br>(4 Pure, 2 manifolds, 2 blend/1hose + 2 blend/2 hose) | G1 | G2 | G3[*7] | G4[*7] | | |
| 47[*4] | Blender 3+2<br>(4 Pure, 2 manifolds, 3 blend/1hose + 2 blend/2 hose) | G1 | G2 | G3 | G4[*7] | G5[*7] | |
| 48[*4] | Blender 2+2<br>(4 Pure, 2 manifolds, 2 blend/1hose + 2 blend/1 hose) | G1 | G2 | G3[*7] | G4[*7] | | |
| 49[*4] | Blender 3+2<br>(4 Pure, 2 manifolds, 3 blend/1hose + 2 blend/1 hose) | G1 | G2 | G3 | G4[*7] | G5[*7] | |

[*1] T*x* = Position in Totals Message.
[*2] G*x* = Grade Number in Two-Wire Messages.
[*3] Denotes a Single Hose blender with Single Hose MPD Combination
[*4] Denotes a blender with two blend manifolds.
[*5] For Unit Type 41 and 42 – Denotes Extra +1 Grade
[*6] For Unit Types 43, 44 and 45 – Denotes the first of the extra MPD Grades
[*7] For Unit Types 46, 47, 48 and 49 – Denotes One of the +2 Blended Grades

# 7. APPENDIX A

## 7.1 Application Note #001

| | |
|---|---|
| Affected Document: | TWO WIRE PROTOCOL REPORT VOL. 1<br>TWO-IS-S1.0-S |
| Issue Date: | 10 September 1990 |
| Author: | K. Eric Lee |
| Reason: | To support Push-To-Stop function and to meet the following criteria.<br>• Low risk and minimum change to pump software.<br>• No change to Console/PAM/CRIND software.<br>• Console/PAM shall be able to close out transactions when Pump Handle is turned off or close out shift/PPU change if the pump is isolated.<br><br>In a “Push-To-Stop” equipped pump, it is very likely that the “Pump Handle” will be fixed and the Removal of nozzle will automatically activate Pump Handle signal to the Console. |
| Functional Impact: | 1. Pump will turn off valves as soon as PUSH to STOP button is pushed.<br>2. Valve can not be turned on again until Pump Handle is turned off and pump authorized by the console.<br>3. Pump will display an error code as long as Pump Handle is ON. |
| Two Wire Impact: | 1. The pump will not change the current state (i.e. Two Wire Status.)<br>2. The pump will respond to Two Wire commands and pump handle as specified in Two Wire Protocol Report.<br>3. The ERROR Code will be available to the Console in the transaction data at PUMP STOP state.<br>4. Note the error code condition will be removed if Pump Handle is at “OFF” condition or is turned off. |

## 7.2 Application Note #002

| | |
|---|---|
| Affected Document: | TWO WIRE PROTOCOL REPORT VOL. 1<br>TWO-IS-S1.0-S |
| Issue Date:<br>Updated: | 10 September 1990<br>26 October 2007 |
| Author: | K. Eric Lee |
| Reason: | Reduce PRESET and PPU transmission time |
| Two Wire Impact: | 1. All pumps that support version number polling will also support a fast PRESET and PPU Data transmission. This includes all modular electronic pumps V50.2 and up, Blender and Post-Modular Advantage. This currently does not include Epsilon, Sandpiper or E101 based dispensers.<br>2. The delay time for word-to-word interval can be reduced from 68ms (as specified in 3.4.3 DATA TO PUMP) to 6ms for this class of pumps.<br>3. A proposed algorithm from Two Wire Review Committee to take advantage of the fast PPU/Preset Data transmission time.<br>a. Poll the pump version number when the pump comes on-line.<br>b. Store "response" or "no response" status for each pump in the Console/Controller.<br>c. The Console/Controller will use the fast PPU/PRESET transmission algorithm if all pumps at site respond to the version number poll.<br><br>Note that this means the system may switch back to slow PRESET/PPU transmission scheme if a "non-responding" pump (such as H-111B V10.7) is added to the loop. |
| New Note 2007: | Some pumps, such as Optimized Advantage and E300, support an even faster Data transmission, called "Fast PPUs". These pumps all respond to the version number polling as above.<br><br>The delay time for word-to-word interval t6 can be reduced from 6ms (as above) to 0ms. In addition, the delays for turn-around (t2), and between data block and polling (t3) can both also be reduced to 0ms.<br><br>Note not all pumps that respond to the version number polling will support the "Fast PPUs" enhanced timing. "Fast PPUs" should only be used if all pumps on the site support it. |

## 7.3 Application Note #003

| | |
|---|---|
| Affected Document: | TWO WIRE PROTOCOL REPORT VOL. 1<br>TWO-IS-S1.0-S |
| Issue Date: | 20 June 1991 |
| Author: | T. E. Dickson |
| Reason: | Definition of modifications to the pump's interface to consoles/CRINDs to support changing of a preset (dollar or volume) with the pump handle on or nozzle out. This allows more flexible control of a pump specifically by a CRIND/Cash Acceptor. |
| Functional Impact: | 1. No changes should be required for any existing console/CRIND.<br>2. All Advantage series pumps and non-blending pumps will receive this capability. Other Modular Advantage versions and all pump versions prior to Modular Advantage will not receive this capability.<br>3. A pump equipped with this feature will allow a preset (volume or money) to be cancelled by the console/CRIND and accept a new preset (volume or money) while the pump handle remains in the ON position or the nozzle is out (auto-on pump). A preset of one type (volume or money) may be replaced by a preset of the same or different type.<br>4. Installing systems which depend on this feature may require upgrading the software of existing pumps. |
| Two Wire Impact: | 1. Preset/authorized pumps receiving a STOP (or ALL STOP) while the handle is ON or the nozzle is OUT and with no fuel dispensed will transition from BUSY to CALL. They will then accept a new preset (and authorize) from the console/CRIND. Note that the pump only goes BUSY when it enters lamp test which is after any prerequisite operation such as Push-To-Start, keylock authorization or cash/credit selection/confirmation by the customer.<br>2. If the pump handle is OFF (or the nozzle is IN) when the pump receives a STOP, the pump will remain in IDLE (no change from current operation).<br>3. There is no change to any message format or content as a result of this modification of the pump's operation.<br>4. A grade or price level change is possible if the customer has not yet confirmed the price level or selected the grade at the pump from the previous preset. A grade and/or price level mismatch between the preset command and the customer's selection will be handled per the Two Wire Specification.<br>5. If a pump receives an Authorize after the STOP, instead of a new preset, it will dispense up to the value of the previous preset received (no change from current operation.) Exception: Sandpiper and later based pumps clear the limit and any grade restriction, and so will not stop at the previous preset limit. |

## 7.4 Application Note #004

| | |
|---|---|
| Affected Document: | TWO WIRE PROTOCOL REPORT VOL. 1<br>TWO-IS-S1.0-S |
| Issue Date: | 5 September 1995 |
| Author: | J. Ronchetti |
| Reason: | Define proper operation of pre-Optimized Advantage Dispensers with the Programmable Pump Preset option while used in conjunction with two-wire presets. |
| Functional Impact: | 1. No changes should be required for any existing console/CRIND. |
| Two Wire Impact: | 1. If the console authorizes the pump for a fill-up, the customer selected money or volume amount will terminate the sale.<br>2. If the console authorizes the pump for a preset and it matches the customer preset type, i.e., both are money or both are volume, then the lesser of the two presets will terminate the sale.<br>3. If the console authorizes the pump for a preset and it does not match the customer preset type, then if the customer preset was received by the pump before the console preset, then the pump will enter the ERROR state and ignore the console preset data block and the customer selected money or volume amount will terminate the sale. The console must not send the AUTH command (or a matching preset) to the pump until the pump has returned to either the OFF or CALL state, i.e., the console must poll the pump again before sending the next command. Or if the customer preset was initiated after the pump received the console preset, then the customer preset will be ignored by the pump and an error tone sounded by the pump. |

## 7.5 Application Note #005

| | |
|---|---|
| Affected Document: | TWO WIRE PROTOCOL REPORT VOL. 1<br>TWO-IS-S1.0-S |
| Issue Date: | 30 July 1997 |
| Author: | J. Ronchetti |
| Reason: | Recommended sequence for sending a preset message to Advantage, Highline, Legacy and MPD dispensers. |
| Method: | 1. Console polls pump.<br>2. Pump responds with its status (must be OFF/IDLE or CALL).<br>3. Console immediately sends 'Data Next' (‘2’ ‘<p>’) command to pump.<br>4. Pump responds with 'Send Data' (‘D’ ‘<p>’).<br>5. Console immediately sends 'Preset Message' (<DATA BLOCK>).<br>6. Console immediately polls pump to check that the preset message was received without error.<br>7. If pump responds without error, the console immediately sends the Authorize command.<br>8. If pump responds 'Error', the console must start the sequence over (Poll, Data Next, Preset Message, Poll, Auth).<br><br>Please note that this sequence should be performed in the minimum amount of time, to minimize any lost presets due to the customer flipping the pump handle on and off. You should not poll any other pumps during this sequence. |

## 7.6 Application Note #006

| | |
|---|---|
| Affected Document: | TWO WIRE PROTOCOL REPORT VOL. 1<br>TWO-IS-S1.0-S |
| Issue Date: | 19 November 2002 |
| Author: | J. Ronchetti |
| Reason: | Recommended sequence for sending a price change message to Advantage, Highline, Legacy and MPD, Eclipse and Sandpiper based E500 dispensers. |
| Method: | 1. Console polls pump.<br>2. Pump responds with its status (must be OFF/IDLE or CALL).<br>3. Console immediately sends 'Data Next' (‘2’ ‘<p>’) command to pump.<br>4. Pump responds with 'Send Data' (‘D’ ‘<p>’).<br>5. Console immediately sends 'Price Change Data' (<DATA BLOCK>).<br>6. Console immediately polls pump to check that the price change message was received without error.<br>7. If pump responds without error, the price change was successful.<br>8. If pump responds 'Error', the console must start the sequence over (Poll, Data Next, Price Change Message, Poll).<br><br>Please note that the above sequence is performed for EACH price to be sent to a dispenser, e.g., for a 5+1 blender with two level pricing, the above sequence is performed a total of 12 times for each side, to update every price. |

## 7.7 Application Note #007

| | |
|---|---|
| Affected Document: | TWO WIRE PROTOCOL REPORT VOL. 1<br>TWO-IS-S1.0-S |
| Issue Date:<br>Updated: | 17 April 1996<br>19 February 2004 |
| Author: | K. Eric Lee |
| Reason: | Calculation of LRC checksums in the Gilbarco Two-Wire Protocol |
| Method: | The Longitudinal Redundancy Check (LRC) character is included in every data block transmitted between the console and the pump, except the Special Function Data Request Command sequence from the pump. Simple commands sent to the pump, e.g., the status poll or authorize command, do not include an LRC |

The pump commands that do not require the LRC check character are:

- Status Poll
- Authorize
- Pump Stop
- All Stop
- Request for Transaction Data
- Request for Pump Totals
- Request for Real Time Money

The following responses to the above commands DO include a check character to insure data integrity, and is calculated as follows:

- Request for Transaction Data Response
- Request for Pump Totals Response

Check character calculation:

- Beginning with the <STX> character, sum the least significant nibble of the bytes. Continue to sum the nibbles, up to and including the LRC next <LRCn> character. The character which follows this is the LRC character which was calculated by the pump. The least significant nibble is equal to the 2's compliment of the summation of all the least significant nibbles. This nibble should be identical to the one which was calculated by the receiving station to insure the data received is identical to the data sent.

Example:

Data stream:

FF F1 F8 E1 E1 E1 E1 E1 F6 E1 F4 FB ED F0

To calculate the LRC on this data stream sum:

0x0F
0x01
0x08
0x01
0x01
0x01
0x01
0x01
0x06

0x01
0x04
0x0B
---
0x33 = 00110011b

Take compliment of 00110011b = 11001100b

To get 2's compliment add 1 to the compliment: 11001100b + 1b = 11001101b

Take the least significant nibble of this: 1101b = 0x0D. This is the LRC, sent as 0xED.

Special functions however have modified rules as to the calculation of the checksum:

When sending a special function message the rules to apply to the LRC are the same as those outlined above. When creating the <DATA BLOCK>, begin summing with the <STX> character and end with the LRC next character <LRCn>. Using the rules described before, take the 2's compliment of this number and logical OR the least significant nibble with 0xF0 to create the LRC character.

When receiving a special function response, a different set of checksum rules apply. To enhance error detection the special function response uses a full byte in the checksum. To get a byte of checksum the character must be split, due to message level requirements. See section 3.11.2 of the Two Wire Specification. All special function responses will begin with Block Length Next characters. These characters are not to be included in the checksum.

In section RESPONSE STRUCTURE of the Two Wire specification is a simple example of the calculation of a checksum. First translate the received bytes. This involves stripping the most significant nibble off the byte and assembling it with the appropriate next character, i.e. Block Length MSD is B0. (B0 thru B9 represent hex 0 thru 9 and C1 thru C6 represent hex A thru F.) Convert B0 to hex 0 and you have 0x00, shift this left 4 bits and you get 0x00. Next you will receive the Block Length LSD, B1. Convert B1 to hex 1 and you are left with 0x01. Since this IS the least significant nibble it isn't necessary to shift the bits. Perform a logical OR operation on the two hex nibbles and 0x01 is the result. This is the translated hex byte. This result is the result that is included in the summation. In section 3.11.2 this is:

0x01
0x23
0x45
0x67
0x89
---
0x0159 since this is greater than a 1 byte value we ignore the most significant byte, thus we are left with 0x59

0x59 in binary is 01011001b and taking the compliment of this we get 10100110b

Adding 1 to this gives us the 2's compliment 10100111b = 0xA7 which is the checksum byte. This is sent as 0xC1 0xB7.

Note: The values in the data block that are C1 thru C6 are converted to A thru F hex, respectively.

## 7.8 Application Note #008

Affected Document: TWO WIRE PROTOCOL REPORT VOL 1
TWO-IS-S1.0-S

Issue Date: December 18, 2009

Author: A. Oldham

Reason: Decimal Point Synchronization across various recent dispenser platforms

| Pump Configuration | | Appearance Over Two-Wire Data | | | | | |
|---|---|---|---|---|---|---|---|
| | | Advantage<br>Atlas 8800<br>Dimension<br>E300<br>Infinity<br>Legacy<br>Performer<br>Titan 8800<br><br>*Pre-Modular, Modular & Optimized* | E500<br>E500S<br><br>*Sandpiper based* | 397G<br>Horizon<br>SK700<br>SK700-II<br><br>*Sandpiper, E101 based* | Frontier<br>Sprint<br><br>*Tulip based* | Advantage<br>Endeaver<br>Enterprise<br><br>*China Advantage* | Electroline<br>Enterpise<br>Euroline<br>EuroDimension<br>Fleetline<br>G-Line<br>G-MPD<br>Lowline<br>MPP<br><br>*Epsilon based *1* |
| **Standard Real-Time Money** | | | | | | | |
| 5D Money = X.XXX | | X.XXXX | X.XXXX | X.XXXX | X.XXX *2 | X.XXX *2 | X.XXXX |
| 5D Money = X.XX | | X.XXX | X.XXX | X.XXX | X.XXX | X.XX *2 | X.XXX |
| 5D Money = X.X | | X.XX | X.XX | X.XX | X.XX | X.X *2 | X.XX |
| 5D Money = X | | X.X | X.X | X.X | X.X | X *2 | X.X |
| | | | | | | | |
| 6D Money = X.XXX | | X.XXX | X.XXX | X.XXX | X.XXX | X.XXX | X.XXX |
| 6D Money = X.XX | | X.XX | X.XX | X.XX | X.XX | X.XX | X.XX |
| 6D Money = X.X | | X.X | X.X | X.X | X.X | X.X | X.X |
| 6D Money = X | | X | X | X | X | X | X |
| | | | | | | | |
| **Real-Time Volume** | | | | | | | |
| Volume = X.XXX | | N/A | X.XXX *3 | X.XXX *3 | N/A | N/A | N/A |
| Volume = X.XX | | N/A | X.XXX *3 | X.XXX *3 | N/A | N/A | X.XXX |
| | | | | | | | |
| **Transaction Data** | | | | | | | |
| 5D Money = X.XXX | | X.XXXX | X.XXXX | X.XXXX | X.XXX *2 | X.XXX *2 | X.XXXX |
| 5D Money = X.XX | | X.XXX | X.XXX | X.XXX | X.XXX | X.XX *2 | X.XXX |
| 5D Money = X.X | | X.XX | X.XX | X.XX | X.XX | X.X *2 | X.XX |
| 5D Money = X | | X.X | X.X | X.X | X.X | X *2 | X.X |
| | | | | | | | |
| 6D Money = X.XXX | | X.XXX | X.XXX | X.XXX | X.XXX | X.XXX | X.XXX |
| 6D Money = X.XX | | X.XX | X.XX | X.XX | X.XX | X.XX | X.XX |
| 6D Money = X.X | | X.X | X.X | X.X | X.X | X.X | X.X |
| 6D Money = X | | X | X | X | X | X | X |
| | | | | | | | |
| Volume = X.XXX | | X.XXX | X.XXX *3 | X.XXX *3 | X.XXX | X.XXX | N/A |
| Volume = X.XX | | X.XXX | X.XXX *3 | X.XXX *3 | X.XXX | X.XX *2 | X.XXX |
| | | | | | | | |
| **Standard Totals Data** | | | | | | | |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 5D Money = X.XXX | | X.XXX | X.XXX | X.XXX | X.XX *2 | X.XXX | X.XXX |
| 5D Money = X.XX | | X.XX | X.XX | X.XX | X.XX | X.XX | X.XX |
| 5D Money = X.X | | X.X | X.X | X.X | X.X | X.X | X.X |
| 5D Money = X | | X | X | X | X | X | X. |
| | | | | | | | |
| 6D Money = X.XXX | | X.XXX | X.XXX | X.XXX | X.XX *2 | X.XXX | X.XXX |
| 6D Money = X.XX | | X.XX | X.XX | X.XX | X.XX | X.XX | X.XX |
| 6D Money = X.X | | X.X | X.X | X.X | X.X | X.X | X.X |
| 6D Money = X | | X | X | X | X | X | X. |
| | | | | | | | |
| Volume = X.XXX | | X.XX | X.XX | X.XX | X.XX | X.XX | N/A |
| Volume = X.XX | | X.X *2 | X.XX *6 | X.XX | X.XX | X.X *2 | X.XX |
| | | | | | | | |
| **Standard Preset Data** | | | | | | | |
| 5D Money = X.XXX | | X.XXX | X.XXX *6 | X.XXX | X.XX *2 | X.XXX | X.XXX |
| 5D Money = X.XX | | X.XX | X.XX | X.XX | X.XX | X.XX | X.XX |
| 5D Money = X.X | | X.X | X.X *6 | X.X | X.X | X.X | X.X |
| 5D Money = X | | X | X *6 | X | X | X | X |
| | | | | | | | |
| 6D Money = X.XXX | | X.XXX | X.XXX | X.XXX | X.XX *2 | X.XXX | X.XXX |
| 6D Money = X.XX | | X.XX | X.XX | X.XX | X.XX | X.XX | X.XX |
| 6D Money = X.X | | X.X | X.X | X.X | X.X | X.X | X.X |
| 6D Money = X | | X | X | X | X | X | X |
| | | | | | | | |
| Volume = X.XXX | | X.XX | X.XX | X.XX *6 | X.XX | X.XXX *2 | N/A |
| Volume = X.XX | | X.XX | X.XX | X.XX | X.XX | X.XX | X.XX |
| | | | | | | | |
| 5 Digit Volume Allowed? | | Yes (XXX.XX) | Yes (XXX.XX) | Yes (XXX.XX) | Yes (XXX.XX) | Yes (XXX.XX) | Yes (XXX.XX) |
| 6 Digit Volume Allowed? | | No | No | Yes (XXXX.XX) *4 | Yes (XXXX.XX) *4 | Yes | Yes (XXXX.XX) *4 |
| 5 or 6 Digit Money, Regardles of Setting? | | No | No | Yes *5 | Yes *5 | No | Yes *5 |
| | | | | | | | |
| **Standard Range of Grades Preset Data** | | | | | | | |
| 5D Money = X.XXX | | X.XXX | X.XXX *6 | X.XXX | X.XX *2 | N/A | X.XXX |
| 5D Money = X.XX | | X.XX | X.XX | X.XX | X.XX | N/A | X.XX |
| 5D Money = X.X | | X.X | X.X *6 | X.X | X.X | N/A | X.X |
| 5D Money = X | | X | X *6 | X | X | N/A | X |
| | | | | | | | |
| 6D Money = X.XXX | | X.XXX | X.XXX | X.XXX | X.XX *2 | N/A | X.XXX |
| 6D Money = X.XX | | X.XX | X.XX | X.XX | X.XX | N/A | X.XX |
| 6D Money = X.X | | X.X | X.X | X.X | X.X | N/A | X.X |
| 6D Money = X | | X | X | X | X | N/A | X |
| | | | | | | | |
| Volume = X.XXX | | X.XX | X.XX | X.XX *6 | X.XX | N/A | N/A |
| Volume = X.XX | | X.XX | X.XX | X.XX | X.XX | N/A | X.XX |
| | | | | | | | |
| 5 Digit Volume Allowed? | | Yes (XXX.XX) | Yes (XXX.XX) | Yes (XXX.XX) | Yes (XXX.XX) | N/A | Yes (XXX.XX) |
| 6 Digit Volume Allowed? | | No | No | Yes (XXXX.XX) *4 | Yes (XXXX.XX) *4 | N/A | Yes (XXXX.XX) *4 |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 5 or 6 Digit Money, Regardles of Setting? | | No | No | Yes *5 | Yes *5 | N/A | Yes *5 |

Notes:
*1 – Early Epsilon-based systems do not accept Presets. Epsilon was always a 5-digit system.
*2 – Violates two-wire spec.
*3 – Matches transaction data format. Can be configured for XXX.XXX or follow the volume formats.
*4 – Later versions support both 5 & 6 digit fields for Volume Presets.
*5 – Later versions support either format, regardless of whether the pump is in 5-digit mode or 6-digit mode. Decimal follows the display configuration.
*6 – Early versions of software implement this field incorrectly. Later versions of software correct the fields as defined by this specification.

## 7.9 Application Note #009

| | |
|---|---|
| Affected Document: | TWO WIRE PROTOCOL REPORT VOL 1<br>TWO-IS-S1.0-S |
| Issue Date: | April 30, 2011 |
| Author: | F. Shan, S. Liu, Zou X., Zhao Y. |
| Reason: | China pumps Special pump States and data format |

Notes:

1. Newer Chinese pumps go from CALL status to BUSY (9x) via AUTH (8x). The transition from AUTH to BUSY happens spontaneously once the transaction money/volume has been reset and the pump and solenoid activated. Some earlier China pumps do not have AUTH state when authorizing the pump from CALL status; they will go to BUSY directly.
2. Some earlier China pumps do not send any EOT status (either PEOT or FEOT). The controller software will have to handle the situation where the pump goes to IDLE from BUSY directly as an exceptional case, and use the Totals increase to determine if it is a real transaction.
3. Earlier Endeavour pumps may not support Special Functions, and for Endeavour pump, each fuelling position only has one product, so there is no need to ask for Products' ID by Special Function or by command 4X at STOP status.
4. For China Endeavour pumps, even where there is only one product on each fuelling position, the Totals Response to command 5X are still composed of 3 products, for historical compatibility reasons.
5. Volume totals implemented in China pumps are configured as one decimal place less than the volume main display. So if the main display volume is (x)xxx.xxx, then the totals will be (xxxx)xxxxxx.xx, and if the main display volume is (x)xxxx.xx, then the totals will be (xxxx)xxxxxxx.x format. In some new pumps the volume totals decimal places can be selected as 1 or 2 from the keypad when the main display volume has 2 decimal places (x)xxxx.xx; refer to the pump manual.
6. Command 4X – transaction data is not updated at CALL status, but updated at STOP and EOT status.
7. Command 5X – totals data are not updated at STOP status, only updated at the EOT.
8. Extended data formats are used for the pumps with 7-7-6 main display.
9. Some old China pumps may also have a longer time to command response, >68ms.
10. For China pumps with 8051 CPU, the Two-Wire baud rate is 5760bps rather than 5787bps (Z80 CPU). This is compatible with FCC running at 5787bps baud rate.
11. Most China pumps have the 2-wire input clamp voltage, <18V.