ProSoft TECHNOLOGY

Where Automation Connects.

MVI56E-AFC / MVI69E-AFC

Enhanced Liquid and Gas Flow Computer

Backplane Communication

July 14, 2021

TECHNICAL NOTE

Your Feedback Please

We always want you to feel that you made the right decision to use our products. If you have suggestions, comments, compliments or complaints about our products, documentation, or support, please write or call us.

How to Contact Us

ProSoft Technology, Inc.
+1 (661) 716-5100
+1 (661) 716-5101 (Fax)
www.prosoft-technology.com
support@prosoft-technology.com

MVIxxE-AFC Technical Note Backplane Communications

July 14, 2021

Content Disclaimer

This documentation is not intended as a substitute for and is not to be used for determining suitability or reliability of these products for specific user applications. It is the duty of any such user or integrator to perform the appropriate and complete risk analysis, evaluation and testing of the products with respect to the relevant specific application or use thereof. Neither ProSoft Technology nor any of its affiliates or subsidiaries shall be responsible or liable for misuse of the information contained herein. Information in this document including illustrations, specifications and dimensions may contain technical inaccuracies or typographical errors. ProSoft Technology makes no warranty or representation as to its accuracy and assumes no liability for and reserves the right to correct such inaccuracies or errors at any time without notice. If you have any suggestions for improvements or amendments or have found errors in this publication, please notify us.

No part of this document may be reproduced in any form or by any means, electronic or mechanical, including photocopying, without express written permission of ProSoft Technology. All pertinent state, regional, and local safety regulations must be observed when installing and using this product. For reasons of safety and to help ensure compliance with documented system data, only the manufacturer should perform repairs to components. When devices are used for applications with technical safety requirements, the relevant instructions must be followed. Failure to use ProSoft Technology software or approved software with our hardware products may result in injury, harm, or improper operating results. Failure to observe this information can result in injury or equipment damage.

For professional users in the European Union
If you wish to discard electrical and electronic equipment (EEE), please contact your dealer or supplier for further information.

Prop 65 Warning – Cancer and Reproductive Harm – www.P65Warnings.ca.gov

Contents

1 Preface

1.1 Purpose of this Backplane Communications Technical Note

The purpose of this document is to provide technical details of the backplane communication blocks used with the MVI56E-AFC and MVI69E-AFC Enhanced Liquid and Gas Flow Computers.

1.2 Additional Information

The following resources contain additional information that can assist the user with the module installation and operation.

Table 1.1. - Additional Information

Resource	Link
MVI56E-AFC / MVI69E-AFC Reference Guide	www.prosoft-technology.com
MVI56E-AFC / MVI69E-AFC Setup and Configuration Guide	www.prosoft-technology.com

1.3 Support

Technical support will be provided via the Web (in the form of user manuals, FAQ, datasheets etc.) to assist with installation, operation, and diagnostics.

For additional support the user can use either of the following:

Table 1.2. – Support Details

Resource	Link
Contact Us web link	www.prosoft-technology.com
Support email	www.prosoft-technology.com

2 Overview

2.1 Multiplexing

Set aside at least 2 words of the 248-word output block array:

- word 0 = the "sentinel"
- word 247 = the "anchor"

A PLC-output (i.e. EAFC input) block array is recognized by the module as valid only if all three of these conditions are true:

- The values of the sentinel and the anchor are the same; we call this value the "transaction number".
- The transaction number is non-zero.
- The transaction number is different from the preceding one.

This scheme allows the 248-word PLC-output block array to be multiplexed, and also provides the block integrity required for portions of it.

The PLC-input block array returned by the EAFC has the same sentinel and anchor in similar locations; the sentinel is returned in word 0 and the anchor in word 249. The PLC may then verify its validity by checking the same three conditions, and the transaction number can be used to match the input (the EAFC's output) with the originating output (the EAFC's input).

The EAFC uses the transaction number only for the block-validation logic described above and does not use it in any other way. It cares nothing about specific transaction number values (except 0), which therefore may be chosen arbitrarily by the PLC program subject only to the constraints of the validation logic.

2.2 Output and Input Block Array Contents

The two block arrays, output and input, have different lengths; the output (EAFC input) has length 248, but the input (EAFC output) has length 250. The sentinel and the anchor occupy the first and the last words of each block respectively, words 0 and 247 on output and words 0 and 249 on input. The two extra input words are returned with these data:

Input Word	Bits	Contents
Input word 247	Bits 0 to 7	Copy of low-order 8 bits of Site Status at Ph00006
	Bit 8	Modbus master poll pending
	Biit 9	Modbus master poll complete (results waiting)
	Bit 10	Pass-thru input pending
	Bit 11	Transmitter calibration mediation pending; see “Transmitter Calibration Mediation” on page 39.
	Bit 12	Prove enabled (version 2.07+)
	Bits 13 to 15	[reserved]
Input word 248	Bits 0 to 15	Bitmap of meters in alarm: bit 0 => M01, bit 1 => M02, etc.

Word 1, immediately following the sentinel, contains the length of the data region, not including the sentinel, the anchor, or the length word themselves; hence its maximum value is 245. Words between the end of the data region and the anchor are ignored on output (to the EAFC) and returned as zero on input (from the EAFC), except for the two extra input words.

The "length" word returned by the module in the input block array has the high-order (sign) bit set if any formatting error is detected in the data region; we call this a "format alarm". The remainder of the input length word contains the number of data words successfully processed before any format alarm, which will equal the output length if no alarm is raised. If the output length word itself raises an alarm (e.g. value > 245), then the returned input length is -1.

The data region is divided into function blocks, allocated in the output block array in the sequence and quantity required by the end-user application at the discretion of the implementer. The EAFC returns function blocks of the input in locations and sizes matching those of the output. The EAFC completes the actions implied by all function blocks before responding with the input block array to the PLC. If the EAFC is unable to determine the size of any function block, or the implied size would overlap the block array's anchor, then that and all following function blocks are not processed and a format alarm is raised. Invalid parameters delivered in some function blocks may also raise format alarms; details below.

In the following, use of the word "block" without any qualifiers means such a function block.

2.3 Block ID

The first word of each block, called the "block ID", specifies the block type, control flags, and addressing and additional information depending on type.

Types include (see complete list below):

- null
- wallclock
- Modbus gateway
- meter process inputs
- meter analysis
- meter signals

Control flags include:

- output (not) present (i.e. block contains valid output to the EAFC)
- input (not) required (i.e. the EAFC populates matching returned block)

Type-dependent addressing and information include, where appropriate:

- meter number
- Modbus database table (primary vs virtual slave, holding vs input bank)
- null block length

The above lists are examples only and are not intended to be complete.

Values in the block ID of all types, flags, addressing, and other information are chosen such that a word value of zero indicates a null block of length 1 (the block ID only).

For each block type, the matching input block (returned by the EAFC) also contains a block ID. Portions of this word echo the output verbatim, while the remainder may contain status information about the result of the action.

The Block ID layout is as follows:

Table 2.1. – Block ID Layout

Bits	Output	Input
0 to 7	data length, or meter number &c	result of action, or echoed
8	EAFC ignores received output	result of action
9	EAFC skips returning input	result of action
10 to 15	function code, with options	echoed

The function descriptions below give the formats of both the output and the input block IDs, laid out as bit fields according to this partitioning. In those descriptions, these symbols have these meanings:

Table 2.2. – Format Symbols

Symbol	Meaning
0	Bit has value 0
1	Bit has value 1
i	1 => EAFC skips returning input
o	1 => EAFC ignores received output
a	1 => meter is in alarm
l	7- or 8-bit data length, *excluding* the block ID
m	Meter number 1 thru 16 (5 bits), or 0 for site/prover when permitted and other symbols are described where they occur. If the "skip input" bit of the output block ID is set, the returned input block ID contains the function code only unless otherwise indicated.

Each output block ID is verified to be unambiguous: any value that is out of range, or any undefined bit that is set to 1, causes a format alarm.

2.4 Block Type Summary

Table 2.3. – Block Type Summary

Block Number	Contents	Code
1	Null	000000 binary 0 decimal
2	Firmware version information	100000 binary 32 decimal
3	Wallclock	000001 binary 1 decimal
4	Meter type and product group fetch	100100 binary 36 decimal
5	Meter process inputs	101000 binary 40 decimal
6	Meter component analysis	101010 binary 42 decimal
7	Active stream select	001111 binary 15 decimal
8	Site or meter signals	001100 binary 12 decimal
9	Meter archive fetch	101110 binary 46 decimal
10	Modbus gateway	010drs binary 16 thru 21 decimal
11	Meter enable/disable	01110e binary 28 thru 29 decimal
12	Stream enable/disable	11001e binary 50 thru 51 decimal
13	Meter proving	01111p binary 30 thru 31 decimal
14	Modbus master	0110dr binary 24 thru 27 decimal
15	Modbus pass-thru	0001ks binary 4 thru 7 decimal
16	Transmitter calibration mediation (v4.01+)	10110k binary 44 thru 45 decimal

All other codes are undefined. Use of an undefined code causes a format alarm, as described above.

3 Block Type Details

This section describes the block types, giving their lengths (which *include* the block ID), the meanings of their outputs (to the EAFC), those of their inputs (from the EAFC), and a brief description of the EAFC action implied.

3.1 Null

Code 000000 binary, 0 decimal

Output block ID: 000000 i o lllllll

Input block ID: 000000 0 0 00000000

Length: 1 (block ID) + data length

Output: ignored

Input: zero

This block acts as a space-filler only, implying no action by the EAFC. As the null block has no associated action, the "i" and "o" bits of the output block ID have no function and their values are irrelevant.

Note that an output block array whose data region is set to all zero is interpreted by the EAFC as a sequence of 245 null blocks of length 1 each.

Note also, however, that a null block whose length "lllllll" is too large (extending beyond the block array length) will cause a format alarm.

3.2 Firmware Version Information

Code 100000 binary, 32 decimal

Output block ID: 100000 i o 00000000

Input block ID: 100000 0 0 00000000

Length: 3

Output: [none]

Input: Firmware version

This block fetches the version code of the EAFC application firmware, which may be used by the PLC to tailor its ladder logic appropriately. Since there is no significant output, the "ignore output" bit has no function.

3.2.1 Input Block Format

Table 3.1. – Input Block Format

Word	Contents
Word 1	Firmware major and minor version numbers: ▪ Lo-order byte: minor version number ▪ Hi-order byte: major version number
Word 2	Firmware revision number

3.3 Wallclock

Code 000001 binary, 1 decimal

Output block ID: 000001 i o 00000000

Input block ID: 000001 0 0 000000nf

Bit symbols are described in the following table:

Table 3.2. – Bit Symbols for Wallclock

Bit Symbol	Description
f(ail to set)	clock-set action failed
n(ot set)	clock has not been set

Length: 7

Output: YMDhms; 24-hour clock, 4-digit year

Input : current wallclock YMDhms

This block (re)sets and/or returns the wallclock.

If the output fails to set the clock, bit 0 of the input block ID is set.

If the input returns a clock that has never been set, bit 1 of the input block ID is set and the clock value returned is all zero.

3.4 Meter Type and Product Group Fetch

Code 100100 binary, 36 decimal

Output block ID: 100100 i o 000mmmmm

Input block ID: 100100 a 0 000mmmmm

Length: 4

Output: [none]

Input: Meter type and product group, streams enabled/configured

This block fetches a summary of the meter type and product group, which may be used by the PLC to tailor its ladder logic for providing process input values to the EAFC. Since there is no significant output, the "ignore output" bit has no function.

3.4.1 Input Block Format

Table 3.3. – Input Block Format for Meter Type and Product Group Fetch

Word	Contents	
Word 1	Meter type and product group bitmap:	
	Bits	**Contents**
	Bit 0	Meter is in alarm
	Bit 1	Meter is enabled
	Bit 2	[spare]
	Bit 3	[spare]
	Bit 4	Device linear
	Bit 5	Phase liquid
	Bit 6	Flow rate / frequency integration
	Bit 7	[spare]
	Bits 8 to 11	Number of active stream (0-based: 0 thru 15)
	Bit 12	[spare]
	Bit 13	[spare]
	Bit 14	[spare]
	Bit 15	[spare]
Word 2	Bitmap of streams enabled (bit 0 => stream 1, etc.)	
Word 3	Bitmap of streams configured	

3.5 Meter Process Inputs

Code 101000 binary, 40 decimal

Output block ID: 101000 i o 000mmmmm

Input block ID: 101000 a 0 000mmmmm

Length: 15

Output: process input alarms and values, as described below

Input : alarms and status, selected values from the meter database

This block provides fresh values of the process inputs to the meter calculation logic and returns results from the latest such calculation.

Process inputs in the output block are meaningful depending on the meter's configuration; for example, the "primary" input is interpreted according to the meter type, water content is meaningful only for liquid NGL/crude, and pressure is not meaningful for water product.

The input block always returns an alarm indicator as part of the block ID; the remaining words are freely selectable from the entire meter database, with the default selection including alarm detail, net volume accumulator, net volume flow rate, and other values depending on configuration. For default layouts, see below.

3.5.1 Output Block Format

Table 3.4. – Output Block Format for Meter Process Inouts

Word	Contents
Words 1 to 2	Bitmap of process input alarms, as determined by the PLC or the associated transmitter:
	Bits / **Contents**
	Word 1, bits 0 to 3 — alarms for primary input
	Word 1, bits 4 to 7 — alarms for temperature
	Word 1, bits 8 to 11 — alarms for pressure
	Word 1, bits 12 to 15 — alarms for density
	Word 2, bits 0 to 3 — alarms for water content
	Word 2, bits 4 to 7 — alarms for pulse count
Words 3 to 4	Primary input; depending on configuration, one of the following: • Differential pressure (orifice, v-cone, etc.) • Flow rate (flow rate integration) • Pulse frequency (linear)
Words 5 to 6	Temperature
Words 7 to 8	Pressure
Words 9 to 10	Density or densitometer frequency (liquids)
Words 11 to 12	Water content (unrefined or semi-refined liquids) Engineering units are percent of volume, range 0% to 100%.
Words 13 to 14	Pulse count (linear pulse train)

3.5.2 Output Alarm Bits, Per Non-Pulse Process Input

Table 3.5. – Output Alarm Bits, Per Non-Pulse Process Input

Bit	Contents
0	Transmitter failure (no input available)
1	Stale data (value not refreshed from source since last delivery)
2	Range limit exceeded, lower
3	Range limit exceeded, upper

3.5.3 Output Alarm Bits, Pulse Process Input

Table 3.6. – Output Alarm Bits, Pulse Process Input

Bit	Contents
0	Transmitter failure (no input available)
1	Stale data (value not refreshed from source since last delivery)
2	[reserved]
3	Pulse fidelity error (with reduced-reliability count available)

The remaining (unspecified) bits in word 2 (bits 8 to 15) are ignored.

All process inputs are delivered as 32-bit floating point values in the engineering units configured for the meter run, except for "pulse count" which is a non-resetting (but rolling over) 32-bit unsigned integer obtained from the input device's raw pulse count register (value is progressive, *not* incremental).

3.5.4 Input Block Format (Default)

Table 3.7. – Input Block Format (Default) for Meter Process Inputs

Meter Type	Word	Contents
Gas meter	Words 1 to 2	meter alarms (dword bitmap)
	Words 3 to 4	non-resettable accumulator, totalizer, net (dword)
	Words 5 to 6	non-resettable accumulator, residue, net (float)
	Words 7 to 8	flow rate, net (float)
	Words 9 to 10	flow rate, gross (float)
	Words 11 to 12	Fpv (float)
	Words 13 to 14	C-prime (float)
Liquid meter	Words 1 to 2	meter alarms (dword bitmap)
	Words 3 to 4	non-resettable accumulator, totalizer, net (dword)
	Words 5 to 6	non-resettable accumulator, residue, net (float)
	Words 7 to 8	flow rate, net (float)
	Words 9 to 10	non-resettable accumulator, totalizer, gross (dword)
	Words 11 to 12	non-resettable accumulator, totalizer, gross std (dword)
	Words 13 to 14	non-resettable accumulator, totalizer, mass (dword)

Depending on the alignment of this block within the containing block array, 32-bit quantities in both output and input blocks may be *unaligned*. Ladder code must compensate accordingly. However, if word 0 (the block ID) is aligned on an odd word boundary then all elements are correctly aligned (that is, alignment is consistent).

3.6 Meter Component Analysis

Code 101010 binary, 42 decimal

Output block ID: 101010 i o 000mmmmm

Input block ID: 101010 a r 000mmmmm

Bit symbols are described in the following table:

Table 3.8. – Bit Symbols for Meter Component Analysis

Bit Symbol	Description
(e)r(ror)	0: OK 1: error occurred (see below)

Length: 66

Output: stream number and analysis

Input: active stream, error indicators, selected values from the meter database

This block supplies for the targeted stream the fluid analysis for several metering calculations, including those of compressibility and energy. The analysis is supplied as a sequence of component concentrations (molar fractions) in 32-bit floating point form.

The input block always returns an alarm indicator as part of the block ID and error indicators; the remaining 64 words are freely selectable from the entire meter database, with the default selection being nothing (all zero).

The EAFC clips component concentrations received in this block to the range 0.0000 through 6.5536, without raising a range alarm for any value that is actually clipped. However, if clipping is performed then the analysis itself will raise a normalization alarm.

3.6.1 Output Block Format

Table 3.9. – Ouput Block Format for Meter Component Analysis

Word	Contents
Word 1	1-based number of the meter's (configured) stream targeted by this analysis, or 0 for the active stream.
Words 2 to 3	Mole fraction concentration of component #1, by default AGA-8 component #1, C1.
Words 4 to 5	Mole fraction concentration of component #2, by default AGA-8 component #2, N2.
Words 6 to 7	Mole fraction concentration of component #3, by default AGA-8 component #3, CO2.
Words 8 to 9	Mole fraction concentration of component #4, by default AGA-8 component #4, C2.
Words 10 to 11	Mole fraction concentration of component #5, by default AGA-8 component #5, C3.
Words 12 to 13	Mole fraction concentration of component #6, by default AGA-8 component #6, H20.
Words 14 to 15	Mole fraction concentration of component #7, by default AGA-8 component #7, H2S.
Words 16 to 17	Mole fraction concentration of component #8, by default AGA-8 component #8, H2.`

Word	Contents
Words 18 to 19	Mole fraction concentration of component #9, by default AGA-8 component #9, CO.
Words 20 to 21	Mole fraction concentration of component #10, by default AGA-8 component #10, O2.
Words 22 to 23	Mole fraction concentration of component #11, by default AGA-8 component #11, iC4.
Words 24 to 25	Mole fraction concentration of component #12, by default AGA-8 component #12, nC4.
Words 26 to 27	Mole fraction concentration of component #13, by default AGA-8 component #13, iC5.
Words 28 to 29	Mole fraction concentration of component #14, by default AGA-8 component #14, nC5.
Words 30 to 31	Mole fraction concentration of component #15, by default AGA-8 component #15, C6.
Words 32 to 33	Mole fraction concentration of component #16, by default AGA-8 component #16, C7.
Words 34 to 35	Mole fraction concentration of component #17, by default AGA-8 component #17, C8.
Words 36 to 37	Mole fraction concentration of component #18, by default AGA-8 component #18, C9.
Words 38 to 39	Mole fraction concentration of component #19, by default AGA-8 component #19, C10.
Words 40 to 41	Mole fraction concentration of component #20, by default AGA-8 component #20, He.
Words 42 to 43	Mole fraction concentration of component #21, by default AGA-8 component #21, Ar.
Words 44 to 45	Mole fraction concentration of component #22, by default unassigned.
Words 46 to 47	Mole fraction concentration of component #23, by default unassigned.
Words 48 to 49	Mole fraction concentration of component #24, by default unassigned.
Words 50 to 51	Mole fraction concentration of component #25, by default unassigned.
Words 52 to 53	Mole fraction concentration of component #26, by default unassigned.
Words 54 to 55	Mole fraction concentration of component #27, by default unassigned.
Words 56 to 57	Mole fraction concentration of component #28, by default unassigned.
Words 58 to 59	Mole fraction concentration of component #29, by default unassigned.
Words 60 to 61	Mole fraction concentration of component #30, by default unassigned.
Words 62 to 63	Mole fraction concentration of component #31, by default unassigned.
Words 64 to 65	Mole fraction concentration of component #32, by default unassigned.

Later versions of the EAFC will allow configuration of the component selection list and sequence, including selection from an expanded list. FOR NOW, however, this sequence is fixed (though component *selection* from those of AGA-8 is permitted).

3.6.2 Input Block Format (Default)

Table 3.10. – Input Block Format (Default) for Meter Component Analysis

Word	Contents	
Word 1, low byte	Currently active stream	
Word 1, high byte	Error bitmap	
	Bit	**Description**
	8	Requested stream number out of range
	9	Requested stream not configured
Words 2 to 65	default all zero	

Depending on the alignment of this block within the containing block array, 32-bit quantities in both output and input blocks may be *unaligned*. Ladder code must compensate accordingly. However, if word 0 (the block ID) is aligned on an even word boundary then all elements are correctly aligned (that is, alignment is consistent).

3.7 Active Stream Select

Code 001111 binary, 15 decimal

Output block ID: 001111 i o 000mmmmm

Input block ID: 001111 a r 000mmmmm

Bit symbols are described in the following table:

Table 3.11. – Bit Symbols for Active Stream Select

Bit Symbol	Description
(e)r(ror)	0: OK 1: error occurred (see below)

Length: 2

Output: desired active stream, 1-based

Input: actual active stream, 1-based; error bitmap

This block is used to make active a specific (enabled) stream for the selected meter.

3.7.1 Output Block Format

Table 3.12. – Output Block Format for Active Stream Select

Word	Contents
Word 1	Desired active stream, 1-based; 0 means no action

3.7.2 Input Block Format

Table 3.13. – Input Block Format for Active Stream Select

Word	Contents	
Word 1, low byte	Actual active stream, 1-based	
Word 1, high byte	Error bitmap	
	Bit	**Description**
	8	Requested stream number out of range
	9	Requested stream not configured
	10	Requested stream not enabled
	11	Event log full, action not performed

To use this block only for obtaining the current active stream, either set the "ignore output" bit or set the desired active stream to zero.

3.8 Site or Meter Signals

Code 001100 binary, 12 decimal

Output block ID: 001100 i o 000mmmmm

Input block ID: 001100 0 0 000mmmmm

Length: 2

Output: signal bits

Input: pending undischarged signals

This block issues signals to the logic in the EAFC. A signal is an instruction for the EAFC to perform a function once. Signals are latched in the EAFC and remain pending until they are discharged. Typical signals are: reset resettable accumulators; write ad-hoc archive record. The EAFC returns this block immediately, without delaying it until the signal is discharged; however, it is still possible for the signal to be discharged before the block is returned, so it is not guaranteed that output signal bits will be echoed in the input. To schedule a signal, the PLC should latch the appropriate bit in the output signal word. To prevent the inadvertent reissuing of signals, the PLC should clear to zero the output signal word immediately after writing it to the module.

3.9 Meter Archive Fetch

Code 101110 binary, 46 decimal

Output block ID: 101110 i o 000mmmmm

Input block ID: 101110 a c 000mmmmm

Bit symbols are described in the following table:

Table 3.14. – Bit Symbols for Meter Archive Fetch

Bit Symbol	Description
c(rc error)	0: OK 1: fetched record reported CRC error

Length: 102

Output: archive age and file select

Input: selected archive, or zero

This block returns an archive record to the PLC, selected according to the criteria in the output block. Archive fetch may be controlled through the use of the "ignore output" bit of the block ID; the "skip input" bit has no function and is ignored.

3.9.1 Output Block Format

Table 3.15. – Output Block Format for Meter Archive Fetch

Word	Contents
Word 1	Archive file select; 0 daily, 1 hourly
Word 2	Age of requested archive
Words 3 to 101	[ignored]

3.9.2 Input Block Format

Table 3.16. – Input Block Format for Meter Archive Fetch

Word	Contents	
Word 1	Error bitmap:	
	Bit	**Description**
	Bit 0	archive file selection invalid
	Bit1	age outside configured file size
	Bit 2	age OK, but archive record never written (new or reconfigured meter)
Words 2 to 101	Archive record data (up to 100 words), or 0 on error	

3.10 Modbus Gateway

Code 010drs binary, 16 thru 21 decimal

Output block ID: 010drs i o lllllll

Input block ID: 010drs 0 0 0000mvaf

Bit symbols are described in the following table:

Table 3.17. – Bit Symbols for Modbus Gateway

Bit Symbol	Description
d(irection)	0: read (from EAFC) 1: write (to EAFC)
r(egister bank)	0: holding register bank 1: input register bank
s(lave)	0: primary slave 1: virtual slave
f(unction)	Modbus exception: illegal function
a(ddress)	Modbus exception: illegal address
v(alue)	Modbus exception: illegal data value
(slave) m(emory)	Modbus exception: slave fail (memory error)

Bits "d" and "r" both set ("write to input") raises a format alarm.

Length: 1 (block ID) + data length- Output: Modbus register address, number of registers, data to be written (if any)

Input: data read (if any)

This block performs an arbitrary data transfer between the PLC and the EAFC data table. Any data transfer that can be performed with a Modbus command to either EAFC slave through any of the module's ports may be implemented using the gateway. Any data words not relevant to the command are ignored upon output (to the EAFC) and zero upon input (from the EAFC). Gateway data transfer may be controlled through the use of the "ignore output" bit of the block ID; the "skip input" bit has no function and is ignored.

3.10.1 Output Block Format

Table 3.18. – Output Block Format for Modbus Gateway

Word	Contents
Word 1	Register address
Word 2	Number of registers
Word 3 to end	Data to be written, if any

3.10.2 Input Block Format

Table 3.19. – Input Block Format for Modbus Gateway

Word	Contents
Word 1 to end	Data read, if any

Since the output block always contains the two words described above, the block's data length must be at least 2 else a format alarm is raised.

The number of registers given in the output block must be such that the implied Modbus data fits entirely within the block as specified by its data length, else a format alarm is raised. This means that for a Modbus read function the number of registers must not exceed the data length, and for a Modbus write function the number of registers must not exceed the data length minus 2.

As may be inferred from the 8-bit data length "llllllll", the size of the Modbus data transfer is not limited to the standard Modbus maximum of 125 words; a Modbus gateway read that occupies the entire 245-word data region of the output block array may transfer up to 244 words.

3.11 Meter Enable/Disable:

Code 01110e binary, 28 thru 29 decimal

Output block ID: 01110e i o 00000000

Input block ID: 01110e w 0 00000000

Bit symbols are described in the following table:

Table 3.20. – Bit Symbols for Meter Enable/Disable

Bit Symbol	Description
e(nable)	0: disable 1: enable
(overflo)w	0: action performed and event-logged 1: event log full, action not performed

Length: 2

Output: meter selection map

Input: map of enabled meters

This block may be used to enable or disable multiple meters with a single transaction. Place into output word 1 the bitmap of meters to be enabled or disabled (bit 0 => meter 1, etc.).

To use this block only for obtaining the bitmap of enabled meters, either set the "ignore output" bit or set the selection map to all zeros.

3.12 Stream Enable/Disable

Code 11001e binary, 50 thru 51 decimal

Output block ID: 11001e i o 000mmmmm

Input block ID: 11001e a r 000mmmmm

Bit symbols are described in the following table:

Table 3.21. – Bit Symbols for Stream Enable/Disable

Bit Symbol	Description
e(nable)	0: disable 1: enable
(e)r(ror)	0: OK 1: error occurred (see below)

Length: 3

Output: stream selection map

Input: map of enabled streams, map of configured streams

This block may be used to enable or disable multiple streams for the selected meter with a single transaction. Place into output word 1 the bitmap of streams to be enabled or disabled (bit 0 => stream 1, etc.).

3.12.1 Output Block Format

Table 3.22. – Output Block Format for Stream Enable/Disable

Word	Contents
Word 1	Stream selection map
Word 2	[ignored]

3.12.2 Input Block Format

Table 3.23. – Input Block Format for Stream Enable/Disable

Word	Contents
Word 1	Map of enabled streams
Word 2	Map of configured streams

If the error bit "r" is set, it means either:

- Inconsistent request (some requested stream not configured)
  This will be the case if *some* of the bits set in output word 1 (the stream selection map) *are* *not* set in input word 2 (map of configured streams).

  or:

- Event log full, action not performed
  This will be the case if *all* of the bits set in output word 1 (the stream selection map) *are* set in input word 2 (map of configured streams).

The simplest logic to distinguish these cases is to calculate:

(out_word_1) AND NOT (in_word_2)

using boolean AND and NOT operations on bitmaps, for which the result:

!= 0 => case (1), inconsistent request

== 0 => case (2), event log full

In either case, no action is taken.

A stream-disable request can fail partially and silently (error bit "r" clear) because it is not permitted to disable an active stream. In such a case unaffected streams are disabled successfully and the stream that was not disabled is that for which the corresponding bit in output word 1 (the stream selection map) remains set in input word 1 (map of enabled streams). There will be at most one such bit, for the currently active stream. Boolean logic for this is:

(out_word_1) AND (in_word_1)

which yields the map of streams for which the disable request was ignored.

Note that this consideration applies ONLY to DISABLE-stream requests; active stream conditions do not affect enable-stream requests.

If the error bit "r" is clear, then all stream actions were performed and event-logged, subject to the disable-active-stream exception described above.

To use this block only for obtaining the bitmaps of enabled and configured streams, either set the "ignore output" bit or set the selection map to all zeros.

3.13 Meter Proving

Code 01111p binary, 30 thru 31 decimal

Output block ID: 01111p i o 00000000

Input block ID: 01111p 0 0 00000000

Bit symbols are described in the following table:

Table 3.24. – Bit Symbols for Meter Proving

Bit Symbol	Description
p(rocess input)	0: process input not present 1: process input present

Length: 26

Output: see below

Input: see below

This block is the conduit between the EAFC and the PLC for the proving support functionality of the EAFC.

3.13.1 Output Block Format

Table 3.25. – Output Block Format for Meter Proving

Word	Contents
Words 1 to 15	**Prover process input**
Word 1	Prover sequencing **Bit** / **Contents** 0 to 3 — (Inlet) temperature 4 to 7 — (Inlet) temperature 8 to 11 — Switch bar temperature 12 to 15 — (Inlet) pressure
Word 2	Bitmap of process input alarms, as determined by the PLC or the associated transmitter (see below): **Bit** / **Contents** 0 to 3 — Outlet pressure 4 to 7 — Prover density 8 to 11 — Meter pulse count 12 to 15 — Master meter pulse count
Words 4 to 5	(Inlet) temperature
Words 6 to 7	Outlet temperature (if dual)
Words 8 to 9	Switch bar temperature
Words 10 to 11	(Inlet) pressure
Words 12 to 13	Outlet pressure (if dual)
Words 14 to 15	Prover density
Word 16	Prove signals
Word 17	Prove controls
Words 18 to 19	**Parameters for prove enable**
Word 18	Meter number requested
Word 19	Zero (reserved)
Word 20	Runs to select (future)word 21 - Parameters for prove acceptance
Word 21	Requested interpolation point, 0 nearest
Words 22 to 25	Prover run/pass input
Words 22 to 23	Meter pulse count for run/pass
Words 24 to 25	Master meter pulse count for run/pass

3.13.2 Output Alarm Bits, Per Non-Pulse Process Input

Table 3.26. – Output Alarm Bits, Per Non-Pulse Process Input

Bit	Contents
Bit 0	[reserved]
Bit 1	Transmitter failure (no input available)
Bit 2	Range limit exceeded, lower
Bit 3	Range limit exceeded, upper

3.13.3 Output Alarm Bits, Per Pulse Process Input

Table 3.27. – Output Alarm Bits, Per Pulse Process Input

Bit	Contents
Bit 0	Transmitter failure (no input available)
Bit 1	[reserved]
Bit 2	[reserved]
Bit 3	Pulse fidelity error (with reduced-reliability count available)

This layout is the same as that for meter process input (section 3.5 above).

All process inputs are delivered as 32-bit floating point values in the engineering units configured for the prover, except for the pulse counts which are obtained from the input devices' incremental run/pass counter registers, whose values may be either 32-bit unsigned integer or 32-bit floating point and may be scaled to represent fractional pulses (prover configuration must match).

3.13.4 Input Block Format

Table 3.28. – Input Block Format for Meter Proving

Word	Contents
Words 1 to 6	**Prover configuration**
Word 1	Prover type
Word 2	**Prover options**
Word 3	Runs per prove, total
Word 4	Runs per prove, selected
Word 5	Max total runs before abort
Word 6	Short passes/run, bidir 2, else 0
Word 7	Zero (spare)
Word 8	Zero (spare)
Word 9	Prove signals pending (remaining oustanding)
Word 10	Prove status
Word 11	Prove error code (latest enable attempt)
Word 12	Prove controls echo
Word 13	Requested meter echo
Word 14	[reserved]
Word 15	Requested curve point echo
Words 16 to 23	**Run and prove results**
Word 16	Run count, attempted
Word 17	Run count, completed
Word 18	Run count, selected
Word 19	Runs selected (bitmap)
Words 20 to 21	New meter factor
Words 22 to 23	Change in meter factor
Words 24 to 25	[reserved]

Details of Bitmapped Fields

Following are details on the bitmapped fields listed above.

Prove Signals

Table 3.29. – Prove Signals

Bit	Contents
Bit 0	[reserved]
Bit 1	Enable prove
Bit 2	Accept prove
Bit 3	Reject prove
Bit 4	Run start
Bit 5	Run complete
Bit 6	Run cancel
Bit 7	Run delete (!! will be obsolete)
Bit 8	[reserved]

Bit	Contents
Bit 9	Pass complete
Bit 10	[reserved]
Bit 11	[reserved]
Bit 12	Pause prove
Bit 13	Select runs (!! future)
Bit 14	Abort prove
Bit 15	[reserved]

Prover Sequencing

Table 3.30. – Prover Sequencing

Bit	Contents
Bits 0 to 2	Prover phase (except short prover) Value: 0 – Not selected; 1 – Active, not counting; 2 – Counting, past 1st switch (fwd leg Bdir); 3 – Past 2nd switch fwd leg (Bdir only); 4 – Past 1st switch return leg (Bdir only); 5 – Run complete; 6 – ? (Prover inactive, new store count A); 7 – ? (Prover inactive, new store count B)
Bit 3	[reserved]
Bit 4	[reserved]
Bit 5	[reserved]
Bit 6	[reserved]
Bit 7	[reserved]
Bit 8	Ball position: Ready (Udir & Bdir only; fwd leg Bdir)
Bit 9	Ball position: Sealed (Udir & Bdir only; fwd leg Bdir)
Bit 10	Ball position: Ready bwd leg (Bdir only)
Bit 11	Ball position: Sealed bwd leg (Bdir only)
Bit 12	[reserved]
Bit 13	[reserved]
Bit 14	[reserved]
Bit 15	[reserved]

Prove Controls

Table 3.31. – Prove Controls

Bit	Contents
Bits 0 to 7	**Controls at enable**
Bit 0	Auto-accept prove
Bit 1	Auto-continue run
Bit 2	[reserved]

Bit 3	[reserved]
Bit 4	[reserved]
Bit 5	[reserved]
Bit 6	[reserved]
Bit 7	[reserved]
Bits 8 to 15	**Controls at accept (if auto-accept, must be available at enable)**
Bit 8	[reserved]
Bit 9	[reserved]
Bit 10	[reserved]
Bit 11	[reserved]
Bit 12	[reserved]
Bit 13	Apply change in MF to all streams
Bit 14	Apply change in MF to all curve points
Bit 15	Apply new meter factor

Prove Status

Table 3.32. – Prove Status

Bit	Contents
Bit 0	Prove accepted
Bit 1	Meter factor applied
Bit 2	Change in MF applied to all curve points
Bit 3	Change in MF applied to all streams
Bit 4	Prove rejected by command
Bit 5	Prove rejected: empty prove (no runs)
Bit 6	Prove rejected: divide by zero
Bit 7	[reserved]
Bit 8	Prove enabled
Bit 9	Proving run in progress
Bit 10	Prove paused
Bit 11	Run cancelled
Bit 12	Run timed out
Bit 13	[reserved]
Bit 14	Prove rejected: abort
Bit 15	[reserved]

Depending on the alignment of this block within the containing block array, 32-bit quantities in both output and input blocks may be *unaligned*.

Ladder code must compensate accordingly. However, if word 0 (the block ID) is aligned on an even word boundary then all elements are correctly aligned (that is, alignment is consistent).

3.14 Modbus Master

Code 0110dr binary, 24 thru 27 decimal

Output block ID: 0110dr i o lllllll

Input block ID: 0110dr 0 0 0000000p

Bit symbols are described in the following table:

Table 3.33. – Bit Symbols for Modbus Master

Bit Symbol	Description
d(irection)	0: read (from slave) 1: write (to slave)
r(egister bank)	0: holding registers / output coils 1: input registers / input status
p(ending)	0: master poll completed or never issued 1: master poll scheduled but not yet completed

Bits "d" and "r" both set ("write to input register/status") is a format error.

Length: 1 (block ID) + data length

Output: Transaction number, data item size, slave address, Modbus register address, number of data items, data to be written (if any)

Input: transaction number echo, error code, data read (if any)

This block performs an arbitrary data transfer between the PLC and external Modbus slaves connected to the last EAFC serial port, provided that such port is configured as a Modbus master. Any data transfer to or from a slave's holding registers, input registers, output coils, or input status may be implemented using this block; equivalent Modbus function codes are 1, 2, 3, 4, 15, and 16. In addition, capability is provided for access to a slave's "long remote" (32-bit) registers where the slave implements them; in particular, Enron-style long integer (5000 series) and floating point (7000 series) registers are accessible. Any data words not relevant to the command are ignored upon output (to the EAFC) and zero upon input (from the EAFC). The "transaction number" is provided as a resource to the PLC programmer for implementing multiplexing, if required; the EAFC copies it verbatim from output to input and does not use it in any other manner.

This block *is* *not* capable of implementing Enron-style transactions that do not conform to the Modbus standard, such as those that retrieve event and alarm logs and historical (archive) records.

Use the "ignore output" bit of the block ID to control the scheduling of master mode transactions and the "skip input" bit to control the retrieval of the result. The output block is processed and a master transaction scheduled when "ignore output" is clear and there is no transaction currently pending; in all other cases the output is ignored. While "skip input" is clear, a pending transaction is indicated by the "pending" bit, if no transaction is pending then the results of the latest transaction are returned, and in all cases the "transaction number" of the latest or current transaction is returned.

3.14.1 Output Block Format

Table 3.34. – Output Block Format for Modbus Master

Word	Contents
Word 1	Transaction number
Word 2	Data item size and swap options: **Value** / **Description** 0 – bit (in the EAFC, packed 16 to a word) 1 – word (16-bit registers) 2 – long (32-bit items as register pairs) 3 – long remote (32-bit items as single registers) To this, add 10 for byte swap (except size 0), and/or 20 for word swap (sizes 2 and 3 only).
Word 3	Slave address
Word 4	Modbus register address
Word 5	Number of data items This is the number of data items as determined by the "data item size", and is the same as the number of Modbus registers *except* for data item size 2 (which allocates 2 Modbus registers for each data item).
Word 6 to end	Data to be written, if any When word-swap is applied to a data packet containing an odd number of words, the last word is swapped with a word of zero.

3.14.2 Input Block Format

Table 3.35. – Input Block Format for Modbus Master

Word	Contents
Word 1	Transaction number echo
Word 2	Error code: **Value** / **Meaning** =0 – No error >0 – Modbus exception code or communication error; see below. <0 – Configuration, parameter, or logic error; see below.
Word 3 to end	Data read, if any When word-swap is applied to a data packet containing an odd number of words, the last word is swapped with a word of zero.

Modbus Exception Codes

Modbus exception codes are issued by the responding slave and are listed in commonly available Modbus protocol manuals; they lie between 1 and 127, and include:

Table 3.36. – Modbus Exception Codes

Code	Description
1	Illegal function
2	Illegal address
3	Illegal data value

Communication Errors

Communication errors are issued by the EAFC.

Table 3.37. – Communication Error Codes

Code	Description
500	CTS timeout
501	Receive timeout
502	Bad framing
503	Buffer overrun
504	Bad checksum/CRC
505	Wrong slave
506	Wrong function code
507	Wrong length

Configuration, Parameter, or Logic Error

Table 3.38. – Configuration, Parameter, or Logic Error Codes

Code	Description
-1	Master port not configured
-2	Master port never used
-3	Bad slave address
-4	Bad direction / target
-5	Bad datum size / swap options
-6	Bad number of data items

Since the output block always contains the five words described above, the block's data length must be at least 5 else a format alarm is raised.

The number of data items given in the output block must be such that the implied Modbus data fits entirely within the block as specified by its data length, else a format alarm is raised. This means that for a Modbus read function the number of words occupied by the data must not exceed the block data length minus 2, and for a Modbus write function the number of words occupied by the data must not exceed the block data length minus 5.

3.15 Modbus Pass-Thru

Code 0001ks binary, 4 thru 7 decimal

Output block ID: 0001ks i o 0lllllll

Input block ID: 0001ks 0 0 00000wbr

Bit symbols are described in the following table:

Table 3.39. – Bit Symbols for Modbus Pass-Thru

Bit Symbol	Description
s(wap words)	swap words of word pass-thru
(ac)k(nowledge)	acknowledge receipt (clear pending)
r(eceived)	command present (pending)
b(it)	command is bit-write
(overflo)w	command too long (overflow)

Length: 1 (block ID) + data length

Output: ignored

Input: Modbus write command, or zero

This block fetches any pass-thru Modbus write command sent by an external host, which is returned to the PLC essentially verbatim (see below).

3.15.1 Input Block Format

Table 3.40. – Input Block Format for Modbus Pass-Thru

Word	Contents
Word 1	Register address
Word 2	Number of registers, or 0, or -1
Word 3 to end	Data to be written When word-swap is applied to a data packet containing an odd number of words, the last word is swapped with a word of zero.

The EAFC buffers any such command until it is returned to the PLC via this function, at which time the buffer is made available for the next such command. While a command remains pending and unacknowledged by the PLC the buffer is occupied and cannot receive new commands; the response to a new command written while the buffer is occupied is the Modbus exception code 6, "slave busy".

Since a Modbus write command must write at least one word (bit) and cannot write more than 125 words (2000 bits), the data length of this function block must lie between 3 and 127.

Functions 4 and 5 may be used to "pre-inspect" the command -- data is returned, but still remains pending.

word 2 = length, 0 (no command pending), length w/ h/o bit set (overflow)

On overflow, command remains pending in the EAFC, regardless of the setting of the "acknowledge" bit.

Use "skip input" with function 6 or 7 to flush any pending command - get indicator bits, but no data - such never returns overflow indicator.

"skip input" with function 4 or 5 returns indicator bits only, no data, and command remains pending.

Use "ignore output" to disable function - indicator bits and data all zero, any command remains pending.

3.16 Transmitter Calibration Mediation (version 4.01 and later, only)

Code 10110k binary, 44 thru 45 decimal

Output block ID: 10110k i o 00000000

Input block ID: 10110k 0 p 00000fen

Bit symbols are described in the following table:

Table 3.41. – Bit Symbols for Transmitter Calibration Mediation

Bit Symbol	Description
(ac)k(nowledge)	acknowledge mediation completed; else poll
p(ending)	poll only: mediation pending for returned handle
n(ot found)	acknowledge only: transmitter handle not found
(output)e(rror)	acknowledge only: error code outside range 0-4
(file)f(ull)	acknowledge only: no room for calibration entry

Length: 36

Output: identification of calibrated transmitter on acknowledgement

Input: calibration actions requiring mediation, or 0

This block fetches details of any transmitter calibration actions that must be mediated by the PLC and acknowledges same to the EAFC.

3.16.1 Overview of PLC-Mediated Calibration

This procedure pertains only to those transmitters whose input scaling option "Calibration mediated by PLC" is set. When this option is clear, transmitter calibration adjustments do not make use of any mediation via the PLC and adjustment must be performed by other means, such as by the manual adjustments of traditional transmitters.

1 When the operator of a calibration session decides that calibration is needed, he records a sequence of calibration actions together with their applied and measured values; the EAFC collects the latest instances of each of these recorded actions for eventual delivery to the PLC.

2 When the operator has recorded a suitable set of calibration actions, he issues a "PLC calibration" action that requests PLC mediation. The EAFC then queues for the PLC the input block described below and raises input block array bit 247.11, "Calibration mediation pending", holding the "PLC calibration" action pending for discharge upon step 5 below.

3 Upon seeing input block array bit 247.11 high, the PLC issues function block code 44 ("poll"), to which the EAFC responds with the queued input block and setting input block ID bit "p" to indicate "mediation pending". See Note 2 below.

4 The PLC uses the information in the mediation request to calibrate the "smart" transmitter (such as by writing to it via Modbus) and informs the EAFC of its results by issuing function block 45 ("acknowledge"). See Note 3 below.

5 The EAFC records in the calibration file the "PLC calibration" action and, if there are no other mediation requests queued, drops input block array bit 247.11.

3.16.2 Output Block Format

Table 3.42. – Output Block Format for Transmitter Calibration Mediation

Function	Words	Description
Function 44, “poll”	Words 1 to 35	[unused]
Function 45, “acknowledge”	Word 1	"handle" of transmitter being calibrated (Note 1)
	Word 2	Error code == 0 ACK (calibration succeeded) != 0 NAK (range 1 thru 4; calibration failed) A value of this word outside the range 0 thru 4 is deemed to be NAK code 4. See Note 3 below.
	Word 3	[reserved]
	Words 4 to 5	Floating-point value to be recorded in the calibration file as the "PLC calibration" entry's "Applied" value. Will depend upon the needs of the specific "smart" transmitter; for many such this might be the "as-left low trim value".
	Words 6 to 7	Floating-point value to be recorded in the calibration file as the "PLC calibration" entry's "Measured" value. Will depend upon the needs of the specific "smart" transmitter; for many such this might be the "as-left high trim value".
	Words 8 to 35	[reserved]

3.16.3 Input Block Format

Table 3.43. – Input Block Format for Transmitter Calibration Mediation

Function	Words	Contents
N/A	Word 1	"handle" of transmitter being calibrated (Note 1)
Function 44, "poll"	Word 2	[reserved]
	Word 3	Bitmap of calibration actions present:
	Words 4 to 5	action 2, "Zero Shift", applied value

Bits	Contents
Bits 0 to 1	[reserved]
Bit 2	action 2, "Zero Shift"
Bit 3	action 3, "Set Zero"
Bit 4	action 4, "Set Span"
Bit 5	action 5, "Set Mid1"
Bit 6	action 6, "Set Mid2"
Bit 7	action 7, "Set Mid3"
Bits 8 to 15	[reserved]

Function	Words	Contents
	Words 6 to 7	action 2, "Zero Shift", measured value
	Words 8 to 9	action 3, "Set Zero", applied value
	Words 10 to 11	action 3, "Set Zero", measured value
	Words 12 to 13	action 4, "Set Span", applied value
	Words 14 to 15	action 4, "Set Span", measured value
	Words 16 to 17	action 5, "Set Mid 1", applied value
	Words 18 to 19	action 5, "Set Mid 1", measured value
	Words 20 to 21	action 6, "Set Mid 2", applied value
	Words 22 to 23	action 6, "Set Mid 2", measured value
	Words 24 to 25	action 7, "Set Mid 3", applied value
	Words 26 to 27	action 7, "Set Mid 3", measured value
	Words 28 to 35	[reserved]
Function 45, "acknowledge"	Words 2 to 35	[reserved]

All applied and measured values are floating-point.

Depending on the alignment of this block within the containing block array, 32-bit quantities in both output and input blocks may be *unaligned*. Ladder code must compensate accordingly. However, if word 0 (the block ID) is aligned on an even word boundary then all elements are correctly aligned (that is, alignment is consistent).

3.16.4 Notes

Note 1: Transmitter "handle"

The transmitter "handle" used throughout the EAFC, including by this function block, is a 16-bit unsigned integer formatted into bitfields as follows:

Table 3.44. – Transmitter handle bits

Bits	Contents
Bit 15	0: directory entry Remainder holds a non-zero index into the configurable transmitter directory (TBD). 1: direct indexing, bypass directory
Bit 14	0: meter process input 1: prover process input
Bits 13 to 12	[reserved]
Bits 11 to 8	0-based meter number; 0 for prover
Bits 7 to 4	[reserved]
Bits 3 to 0	0-based input-scaling index (see below) and having these special values: 0 : "invalid handle (BoL)" (no transmitter, beginning of list) 0xFFFF : "invalid handle (EoL)" (no transmitter, end of list)

The 0-based input-scaling indexes are as follows:

Table 3.45. – 0-Based Input-Scaling Indexes

	Index	Description
For meters:	0	Primary input; depending on meter type, this is: ▪ Type "differential standard", differential pressure ▪ Type "differential integration", flow rate ▪ Type "linear", pulse frequency
	1	Temperature
	2	Pressure
	3	Density (liquids only)
	4	Water content (liquids only)
For the prover	0	(Inlet) temperature
	1	Outlet temperature (dual transmitters only)
	2	Switch bar temperature (when configured)
	3	(Inlet) pressure
	4	Outlet pressure (dual transmitters only)
	5	Density

Subnotes

- The EAFC does not implement a configurable transmitter directory, hence all valid handles are of the direct-indexed type with bit 15 set.
- The EAFC does not implement calibration of prover inputs, hence handle bit 14 is always clear and the prover input scaling indexes are not used.
- The special handle values are used by the EAFC's calibration support in managing and accessing its database of calibratable transmitters; they are not relevant to this function block, except that such values are not valid transmitter identifiers.

Note 2: Function 44, "poll"

The "poll" function requests delivery to the PLC of the first queued mediation request.

Behavior

1 If there are no queued requests, the returned input block data are all zero with the block IDs "p" bit clear and block array bit 247.11 clear. Otherwise:

2 The EAFC populates the returned input block as described above, with the "p" bit in the block ID set.

3 The EAFC then rotates the queue, transferring the just-returned block to the tail of the queue, so that if multiple mediation requests are queued the PLC can step through them with multiple polls. At almost all times at most one request will be queued, so that a second poll will fetch the same queued block.

Subnotes

- The PLC uses the transmitter handle returned in the input block to IDentify the transmitter needing calibration. The PLC must deliver this handle to the EAFC in the subsequent "acknowledge" function block to be issued following the mediated calibration.
- The PLC, upon detecting input block array bit 247.11 high, must not assume that an immediately following poll will always deliver a valid queued mediation request, but should use 247.11 merely as an indicator that polling is appropriate at this time and must always check the returned input block for an actual request by inspecting its "p" bit.
- As there is no significant output for a poll, the "o" bit has no function.
- Setting the "i" bit causes return of all zeros after the block ID, so that no transmitter identification or calibration data are available for PLC mediation. However, all other behavior remains unchanged including determination of the returned block IDs "p" bit and the EAFC's rotation of the queue.

Note 3: Function 45, "acknowledge"

The "acknowledge" function informs the EAFC of the results of a PLC-mediated transmitter calibration, which may succeed or fail. The PLC populates the output block with the data described above, which data are used by the EAFC as follows:

1 The input block is prepared with the transmitter handle from the output block, as confirmation only.

2 The queue of pending mediation requests is searched for the entry matching the transmitter handle. If not found, the "n" bit of the input block ID is set and we skip to step (5) below; this may occur if the request is cancelled by the operator before completion by the PLC. Otherwise:

3 An entry is appended to the transmitter's open calibration session whose fields are populated as follows:

a. Action:

This is the output block's error code added to the base "PLC calibration" action value of 10, so that the base value of 10 (error code 0, an ACK) indicates successful completion while values of 11 thru 14 indicate failure of some kind. If the error code is outside the range 0 thru 4, it is deemed to be 4 (yielding action code 14) and the "e" bit of the input block ID is set.

b. As-found/As-left indicators:

Both clear.

c. Operator number:

That of the operator of the calibration session.

d. Applied value:

From words 4 and 5 of the output block.

e. Measured value:

From words 6 and 7 of the output block.

f. Timestamp:

From the EAFC's wallclock.

If there is no room in the calibration file for the new entry, the input block IDs "f" bit is set and the entry is discarded; as the calibration file is sized for up to 59 entries this should rarely become an issue.

4 The pending, now discharged, mediation request is removed from the queue.

5 The input block is returned with its block ID bits "n", "e", and "f" set as described above and bit "p" clear. NOTE that "p" clear does not mean that no more mediation requests remain queued; it means only that this request has now been discharged. Input block array bit 247.11, however, remains set if there are more queued requests.

Subnotes:

- Values of NAK error codes and of the two floating-point elements have no meaning to the EAFC and are merely recorded verbatim in the calibration file entry. The meanings of those values may be assigned as needed for the application, and it is up to the user to interpret calibration reports accordingly. Any "e"rror indication, while explicit in the block ID returned to the PLC, is not copied separately to the calibration file; NAK code 4 (action code 14) is its only indication.
- Setting the "o" bit delivers no data, not even the transmitter handle. No action occurs and input block ID bits "n", "e", "f", and "p" are all clear. Any pending mediation remains pending with input block array bit 247.11 set accordingly.
- The only effect of setting the "i" bit is to suppress return of the transmitter handle. The function is performed as described and the input block ID is fully populated.

4 Support, Service & Warranty

4.1 Contacting Technical Support

ProSoft Technology, Inc. is committed to providing the most efficient and effective support possible. Before calling, please gather the following information to assist in expediting this process:

- Product Version Number
- System architecture
- Network details

If the issue is hardware related, we will also need information regarding:

- Module configuration and associated ladder files, if any
- Module operation and any unusual behavior
- Configuration/Debug status information
- LED patterns
- Details about the interfaced serial, Ethernet or Fieldbus devices

Note: For technical support calls within the United States, ProSoft's 24/7 after-hours phone support is available for urgent plant-down issues.

North America (Corporate Location)	Europe / Middle East / Africa Regional Office
Phone: +1.661.716.5100 info@prosoft-technology.com Languages spoken: English, Spanish REGIONAL TECH SUPPORT support@prosoft-technology.com	Phone: +33.(0)5.34.36.87.20 france@prosoft-technology.com Languages spoken: French, English REGIONAL TECH SUPPORT support.emea@prosoft-technology.com
Latin America Regional Office	**Asia Pacific Regional Office**
Phone: +52.222.264.1814 latinam@prosoft-technology.com Languages spoken: Spanish, English REGIONAL TECH SUPPORT support.la@prosoft-technology.com	Phone: +60.3.2247.1898 asiapc@prosoft-technology.com Languages spoken: Bahasa, Chinese, English, Japanese, Korean REGIONAL TECH SUPPORT support.ap@prosoft-technology.com

For additional ProSoft Technology contacts in your area, please visit:

https://www.prosoft-technology.com/About-Us/Contact-Us.

4.2 Warranty Information

For complete details regarding ProSoft Technology's TERMS & CONDITIONS OF SALE, WARRANTY, SUPPORT, SERVICE AND RETURN MATERIAL AUTHORIZATION INSTRUCTIONS, please see the documents at: www.prosoft-technology/legal