IB-11A003

DELTAMAX MOTION CONTROL SYSTEM March, 1997

Executive Port Serial Communications

Application Note

INDUSTRIAL INDEXING SYSTEMS, Inc.

Rev. 0
Approved By:

DeltaMax Serial Commands

The DeltaMax Controller can be interfaced to a host computer system (such as a PC using COM1 or COM2) using RS232 serial communications. The host can control the operation of the DeltaMax in the following ways:

1. Read and Write Integer and Floating Point Variables.

2. Read the DeltaMax Controller "Status Word".

3. Read flags and I/O.

4. Set Flags and I/O.

5. Provide data for running application programs.

Serial commands are implemented using a **Packet** concept. A **Packet** consists of:

1. The **Packet Header** which is a 4 byte area defined by a **start of packet** character, the **packet type** (1 byte) and **packet body length** (2 bytes).

2. The **Packet Body** which is a variable length area containing ASCII characters and binary data representing addresses, values and other parameters.

3. The **Packet Checksum** which is a 2 byte area representing the checksum of the **Packet** beginning with the **packet type** byte to and including the end of the **Packet Body.**

All communications are transferred at 19200 baud with 8 data bits and 1 stop bit. Parity is not used.

The host must initiate all serial communications as the DeltaMax is a passive device with respect to Packet Protocol.

The **Packet Header**, **Packet Body** and **Packet Checksum** are generated using the protocol described in the remainder of this document.

Data Transfer Protocol

This protocol allows data to be sent between the host computer and the DeltaMax.

In data communications, the protocol defines the rules for the electrical, physical and functional characteristics of the communication link. The protocol contains procedures required to ensure an orderly exchange of information through the link, to and from the executing programs.

The DeltaMax is initially a “passive” device with respect to **Packet** transmission. It will wait for **Packet** requests from other devices and then respond to those requests where appropriate. It is the responsibility of the host to initiate communications with the DeltaMax.

When the DeltaMax receives a **Packet**, it responds to the host with either an ACK (acknowledgement) character of a NAK (negative acknowledgement) character. An ACK character is an indication that the **Packet** received was valid. A NAK character is an indication that the **Packet** received was invalid or a timeout occurred.

It is the responsibility of the host device to retransmit a **Packet** in the event that it receives a NAK character from the DeltaMax.

Data Transfer Sequence

The following describes a typical sequence of events for the transfer of **Packets** between devices:

1. The host device creates and sends a **Packet** to the DeltaMax.

2. The DeltaMax receives the **Packet**. The **Packet** is examined for a valid **Packet Header**, **Packet Body** and **Packet Checksum**.

3. The DeltaMax will respond with an ACK (acknowledgement) character if the **Packet Header**, **Packet Body** and **Packet Checksum** are correct.

   The DeltaMax will respond with a NAK (negative acknowledgement) character if a problem was found with the **Packet** (for example, an incorrect **Packet Checksum** was received or transmission fails to complete within a given period of time resulting in a timeout).

4. If the **Packet** from the host is requesting information, the DeltaMax will then retrieve the information, build a return **Packet** and transmit this **Packet** back to the host device.

   The DeltaMax will also expect an ACK character (06H).or NAK (15H) character from the host device

5. The DeltaMax will retransmit the **Packet** up to 5 times if necessary.

DeltaMax Memory Partitions

DeltaMax memory is partitioned in the following manner:

1. Macroprogram Memory

2. Data Memory
   a. Integer Variable Area
   b. Floating Point Variable Area
   c. Integer Constants Area
   d. Floating Point Constants Area

Macroprogram Memory

This is the area containing the users application program. There are 64,000 avaliable bytes for the Macroprogram Area. The physical memory type of this area is Flash ROM.

Data Memory - Integer / Floating Point Variable Memory

This is the area that contains the variables used by the application program. There are 16,000 avaliable bytes for the Variable Area. Both Integer and Floating Point variables are stored here, in separate areas. The combined size of both the Integer Variable Area and the Floating Point Variable Area cannot exceed 16,000 bytes. The physical memory type of this area is NOVRAM.

Addressing for each area is zero based. The DeltaMax will maintain an offset into the appropriate area and will know which area is to be “read from” or “written to” based on the format byte [F] in the DREAD, DWRITE, BREAD or BWRITE instruction.

Data Memory - Integer / Floating Point Constants Memory

This is the area that contains the constants used by the application program. There are 16,000 avaliable bytes for the Constants Area. Both Integer and Floating Point constants are stored here, in separate areas. The combined size of both the Integer Constants Area and the Floating Point Constants Area cannot exceed 16,000 bytes. The physical memory type of this area is Flash ROM.

Addressing for each area is zero based. The DeltaMax will maintain an offset into the appropriate area and will know which area is to be “read from” based on the format byte [F] in the DREAD or BREAD instruction.

Data Memory Addressing

Since each data area is zero based, there may be as many as four types of data with the same address. The serial commands, however, will define the area to be accessed by the use of the ***type*** or ***format*** *byte.*

For example, doing a DREAD on an Integer Variable at address 8 (decimal) would be accomplished with the following packet:

[Header][09]DREAD,[03][00][08][Checksum]

where the ***[03]*** indicates an Integer Variable type and ***[00][08]*** indicates an address 8 bytes in from the start of the Integer Variable Area.

The ***format*** codes are defined below.

Format Byte Codes

A ***format*** byte is used to indicate to the DeltaMax the type of data being addressed. The format codes are as follows:

01 = Integer Constant
02 = Floating Point Constant
03 = Integer Variable
04 = Floating Point Variable

Any other code is considered illegal.

Packet Data Structure

Please refer to Figure 1 for a diagram of the **Packet** protocol.

Header

The first byte is the **start of packet** character. It will always be a STX (02H) character and is used to signal the beginning of a **Packet**.

The second byte is the **packet type** character. It will always be a (01H) character.

The next 2 bytes define the **packet body length** (the **start of packet** character, the **packet type** and the **Packet Checksum** are not included). The length is given in bytes and may be an even or odd value. The most significant byte of the length is transmitted first.

Packet Body

The variable length **Packet Body** is next. The maximum length is 240 bytes.

Checksum

Following the **Packet Body**, there is a two byte field which contains the **Packet Checksum** for the **Packet**. The checksum is calculated by summing the contents of each **byte** of the **Packet**, beginning with the **packet type** field and continuing through the last byte of the **Packet Body**. The most significant byte of the **Packet Checksum** is transmitted first.

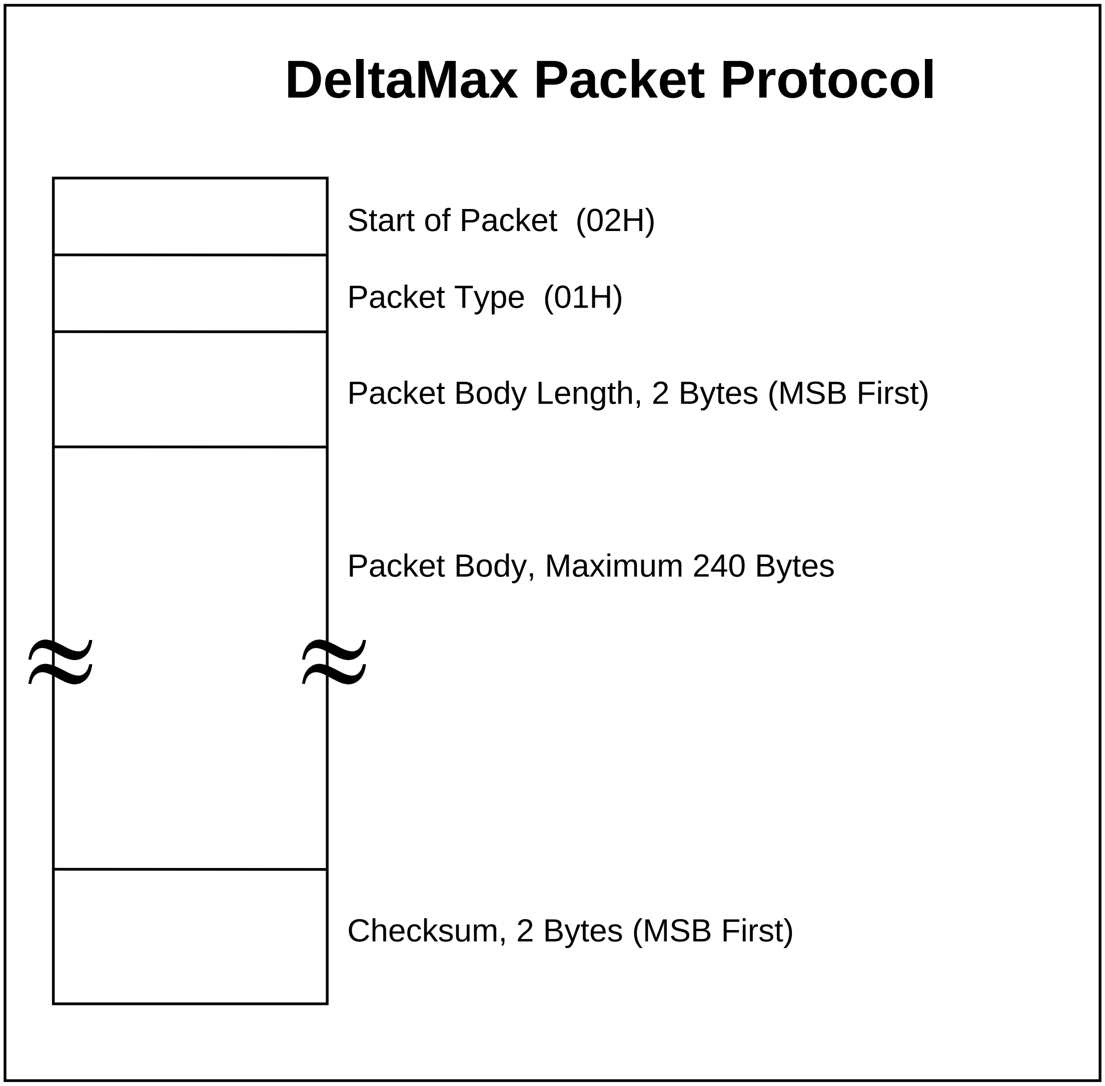
DeltaMax Packet Protocol
Start of Packet (02H)
Packet Type (01H)
Packet Body Length, 2 Bytes (MSB First)
Packet Body, Maximum 240 Bytes
Checksum, 2 Bytes (MSB First)

Figure 1 - Packet Protocol Diagram

Example #1 - Data Write

The following is an example of a **Packet** that would be created to write the value 1234 (decimal) to address 2008 (decimal) in a DeltaMax Controller:

Byte	ASCII Value	Hex Value	Decimal Value	Description
0		02	2	Start of Packet
1		01	1	Packet Type
2		00	0	Packet Body Length
3		0E	14	"
4	D	44	68	Serial Command
5	W	57	87	"
6	R	52	82	"
7	I	49	73	"
8	T	54	84	"
9	E	45	69	"
10	,	2C	44	"
11		03	03	Integer Variable Type
12		07	7	Address of Data (2008 dec.)
13		D8	216	"
14		00	0	Data Value (1234 dec.)
15		00	0	"
16		04	4	"
17		D2	210	"
18		03	3	Checksum of Bytes 1-18 (962 dec.)
19		C2	194	"

Example #2 - Data Read

The following is an example of a DREAD **Packet** that would be created to read an Integer Variable at address 2408 decimal (0x0968) of a DeltaMax.

This **Packet** would be ***sent*** by the host device to a DeltaMax.

Byte	ASCII Value	Hex Value	Decimal Value	Description
0		02	2	Start of Packet
1		01	1	Packet Type
2		00	9	Packet Body Length
3		09	9	"
4	D	44	68	DeltaMax Serial Command
5	R	52	82	"
6	E	45	69	"
7	A	41	65	"
8	D	44	68	"
9	,	2C	44	"
10		03	03	Integer Variable Type
11		09	9	Integer Variable Addr (2408 dec.)
12		68	104	"
13		02	2	Checksum of Bytes 1-12 (531 dec.)
14		13	19	"

This **Packet** would be ***returned*** by a DeltaMax as a response to the host from the **Packet** above. In this example, the returned value is 100 decimal (0x00000064).

Byte	ASCII Value	Hex Value	Decimal Value	Description
0		02	2	Start of Packet
1		01	1	Packet Type
2		00	0	Packet Body Length
3		0A	10	"
4	D	44	68	DeltaMax Serial Command
5	R	52	82	"
6	E	45	69	"
7	A	41	65	"
8	D	44	68	"
9	,	2C	44	"
10		00	0	Data Value (100 dec.)
11		00	0	"
12		00	0	"
13		64	100	"
14		01	1	Checksum of Bytes 1-13 (507 dec.)
15		FB	251	"

General Serial Command Format

The DeltaMax will accept two basic types of serial commands:

1 - The first type of command is a simple transfer of data from a host device to the DeltaMax. Once the DeltaMax receives this type of **Packet**, it simply responds to the host with an ACK or NAK character. Whatever data or command was received is processed by the DeltaMax.

2 - The second type of command is a request for data from a host device to the DeltaMax. Once the DeltaMax receives this type of **Packet**, it responds to the host with an ACK or NAK character.

After responding with an ACK character, the DeltaMax will process the request, construct a **Packet** of information to be sent back to the host device, then transmit that **Packet**.

The DeltaMax will wait for either an ACK or NAK character from the host device, making sure that the **Packet** was received by the host. If the DeltaMax receives a NAK character, it will retransmit the **Packet**. The DeltaMax will retransmit the **Packet** up to 5 times if necessary.

Each serial command consists of ASCII characters, optionally followed by a comma and one or more parameters. The complete command must be formatted into a **Packet** by defining the **Packet Header**, followed by the **Packet Body** and the **Packet Checksum.**

If the command is one that requires a response from the DeltaMax, the DeltaMax will return the response in the same **Packet** format. Further, the DeltaMax will place the command name and the comma in the **Packet**. Details for each serial command follow.

In the format description for each **Packet**, [Header] represents the **Packet Header**, and [Checksum] represents the **Packet Checksum**. The brackets themselves are for purposes of clarity and are NOT part of the actual **Packet**. <u>All numbers in binary format are most significant byte first, followed by least significant byte(s)</u>.

The BREAD Command

The BREAD (block read) serial command serves to read a group of data values from the DeltaMax. Each data value is stored sequentially in the DeltaMax Integer or Floating Point Data Area beginning at the address specified in the BREAD command.

Packet sent to the DeltaMax:

[Header]BREAD,[F][AA][Checksum]

F: A 1-byte value indicating the format of the data to be read from the DeltaMax.

01 = Integer Constant
02 = Floating Point Constant
03 = Integer Variables.
04 = Floating Point Variables

AA: A 2-byte value indicating the address within the Integer Data Area or Floating Point Data Area where the values will be read.

The ***packet body length*** of this **Packet** will always be 9 bytes.

***Packet returned by the DeltaMax*:**

[Header]BREAD,[LL][V1][V2]...[Vn][Checksum]

LL: A 2-byte value indicating the number of values actually read. This value can be as high as 32 for Integer Variables or 16 for Floating Point Variables. This value will be less than 32 Integer or 16 Float if an attempt has been made to read ***outside*** the area defining Integer or Floating Point Variables.

V1...Vn: These represent the returned values for the type being read. There will be 32 4-byte values returned for Integer Variables or 16 8-byte values for Floating Point Variables.

The ***packet body length*** of this **Packet** will always be 136 bytes.

The BWRITE Command

The BWRITE (block write) serial command serves to transmit a group of data values to the DeltaMax. Each data value is stored sequentially in the DeltaMax Integer Data Area or Floating Point Data Area beginning at the address specified in the BWRITE command. Caution should be used to ensure that sufficient space has been reserved in the data area to receive the 128 bytes. If fewer than 128 bytes are to be transmitted, the user must "pad" the command with a sufficient number of dummy values in order to make a total of 128 bytes.

Data arrays longer than 128 bytes can be transmitted by using multiple BWRITE commands.

Packet sent to the DeltaMax:

[Header]BWRITE,[F][AA][V1][V2]...[Vn][Checksum]

F: A 1-byte value indicating the format of the data to be written to the DeltaMax.

03 = Integer Variables
04 = Floating Point Variables

NOTE**:** The DWRITE command cannot write to the ***Constants Area.***

AA: A 2-byte value indicating the address within the Integer Data Area or Floating Point Data Area where the values will be written. The value for [AA] can be determined from the Integer / Floating Point Data Area of a Macroprogram SYMBOL FILE.

V1...Vn: These represent the values being written for the type specified. There will be 32 4-byte values written for Integer Variables or 16 8-byte values for Floating Point Variables.

When placing numeric values in the **Packet**, the most significant byte goes first.

The DeltaMax will not allow writing past the boundary of a specified data area.

The ***packet body length*** of this **Packet** will always be 138 bytes.

***Packet returned by the DeltaMax*:**

*** **None *****

If, for example, a Macroprogram contains the statement:

```
array        dim              32
```

then the Integer Variable Data table will contain an entry for the symbol called "array" and will indicate its address within the data area. This is the value that should be used for [AA] above.

The CFLAG Command

The CFLAG serial command can be used to turn off a specified output or to clear a specified user flag.

Packet sent to the DeltaMax:

[Header]CFLAG,[NN][Checksum]

NN: A 2-byte value indicating the flag number to be cleared.

The ***packet body length*** of this **Packet** will always be 8 bytes.

***Packet returned by the DeltaMax*:**

***** None *****

The DREAD Command

The DREAD function is used to read the contents of a specified address in the DeltaMax Integer Data Area or Floating Point Data Area.

Packet sent to the DeltaMax:

[Header]DREAD,[F][AA][Checksum]

F: A 1-byte value indicating the format of the data to be read from the DeltaMax.

01 = Integer Constant
02 = Floating Point Constant
03 = Integer Variables
04 = Floating Point Variables

AA: A 2-byte value indicating the address within the Integer Data Area or Floating Point Data Area where the values will be read. The value for [AA] can be determined from the Integer / Floating Point Variable area of a Macroprogram SYMBOL FILE.

The ***packet body length*** of this **Packet** will be 9 bytes.

***Packet returned by the DeltaMax*:**

[Header]DREAD,[V1][Checksum]

V1: If [F] is an Integer Variable Type, a 4-byte value will be returned. If [F] is a Floating Point Variable Type, an 8-byte values will be returned.

The ***packet body length*** of this **Packet** will always be 10 bytes when an Integer Variable is returned and 14 bytes when a Floating Point Variable is returned.

The DWRITE Command

The DWRITE serial command transfers a single data value to the specified address in the DeltaMax Integer or Floating Point Data Area.

Packet sent to the DeltaMax:

[Header]DWRITE,[F][AA][V1][Checksum]

F: A 1-byte value indicating the format of the data to be written to the DeltaMax.

03 = Integer Variables
04 = Floating Point Variables

NOTE: The DWRITE command cannot write to the ***Constants Area*****.**

AA: A 2-byte value indicating the address within the Integer Data Area or Floating Point Data Area where the value willl be written. The value for [AA] can be determined from the Integer / Floating Point Variable area of a Macroprogram SYMBOL FILE.

V1: This represents the value being written for the type specified. This will be a 4-byte value for an Integer Variable type or an 8-byte value for a Floating Point Variable type.

When placing numeric values in the **Packet**, the most significant byte goes first.

The DeltaMax will not allow writing past the boundary of a specified data area.

The ***packet body length*** of this **Packet** will always be 14 bytes when writing an Integer Variable and 18 bytes when writing a Floating Point Variable.

Packet returned by the DeltaMax**:**

***** None *****

The GINFO Command

The GINFO serial command is used to retrieve the size of the Program Area and the four distinct Data Areas from the DeltaMax Controller.

Packet sent to the DeltaMax:

[header]GINFO[checksum]

The ***packet body length*** of this **Packet** will always be 5 bytes.

***Packet returned by the DeltaMax*:**

[Header]GINFO,[PP][IC][FC][IV][FV][Checksum]

PP: A 2 byte unsigned value indicating the length of the PROGRAM AREA.

IC: A 2 byte unsigned value indicating the length (in bytes) of the INTEGER CONSTANTS AREA.

FC: A 2 byte unsigned value indicating the length (in bytes) of the FLOATING POINT CONSTANTS AREA.

IV: A 2 byte unsigned value indicating the length (in bytes) of the INTEGER VARIABLE AREA.

FV: A 2 byte unsigned value indicating the length (in bytes) of the FLOATING POINT VARIABLE AREA.

The ***packet body length*** of this **Packet** will always be 16 bytes.

The RFLAG Command

The RFLAG serial command can be used to read the status of DeltaMax inputs, outputs, PLS, timers, motor status flags and user flags.

Packet sent to the DeltaMax:

[Header]RFLAG,[NN][Checksum]

NN: A 2-byte value indicating the flag number to be read.

The ***packet body length*** of this **Packet** will always be 8 bytes.

***Packet returned by the DeltaMax*:**

[Header]RFLAG,[SS][Checksum]

SS: A 2-byte value indicating the state of the flag number read. If the flag is SET / ON, [SS] will be 1. If the flag is CLEAR / OFF , [SS] will be 0.

The ***packet body length*** of this **Packet** will always be 8 bytes.

The RFLAGS Command

The RFLAGS serial command can be used to read the status of all DeltaMax inputs, outputs, PLS, timers, motor status flags, and user flags with a single instruction.

Packet sent to the DeltaMax:

[Header]RFLAGS,Checksum]

NN: A 2-byte value indicating the flag number to be read.

The ***packet body length*** of this **Packet** will always be 7 bytes.

Packet returned by the DeltaMax:

[Header]RFLAGS,[B1][B2]...[B32][Checksum]

[B1]...[B32]: These are 8 bit unsigned values representing the state of the 256 DeltaMax internal status flags. The flag states are returned as follows:

BYTE	BIT #	FLAG #
B1	7 - 0	7 - 0
B2	7 - 0	15 - 8
B3	7 - 0	23 - 16
B4	7 - 0	31 - 24
.	.	.
.	.	.
.	.	.
.	.	.
B32	7 - 0	255 - 248

The ***packet body length*** of this **Packet** will always be 39 bytes.

The RQSTAT Command

The RQSTAT serial command requests the DeltaMax Controller to transmit its status bits as represented in the table below.

Packet sent to the DeltaMax:

[Header]RQSTAT,[Checksum]

The ***packet body length*** of this **Packet** will always be 7 bytes.

***Packet returned by the DeltaMax*:**

[Header]RQSTAT,[SS][FF][AAAA][Checksum]

SS: A 2-byte value representing the 16 Macroprogram Status flags. The Macroprogram status word definition is shown in Table 1.

FF: Not currently used.

AAAA: Not currently used.

The ***packet body length*** of this **Packet** will always be 15 bytes.

As an example, if the returned Macroprogram status word is 65, the program is running (1) and the auto start bit is enabled.

Macroprogram Status Bits

BIT	VALUE	DEFINITION
0	1	PROGRAM RUNNING.
1	2	BAD ARGUMENT.
2	4	FLASH MEMORY NOT RELIABLE.
3	8	SYSTEM HAS BEEN RESET, PROGRAM WAS CLEARED.
4	16	SYSTEM RETURN INTERRUPT ENCOUNTERED.
5	32	SYSTEM FAULT INTERRUPT ENCOUNTERED.
6	64	AUTO START ENABLED. PROGRAM STARTS ON POWER-UP.
7	128	SYSTEM TEST IN PROGRESS.
8	256	ERROR - LOADING OVER RUNNING PROGRAM.
9	512	ERROR - CHECKSUM IS INCORRECT.
10	1024	PROGRAM LOAD IN PROGRESS.
11	2048	ERROR - PROGRAM LOAD OUT OF SEQUENCE.
12	4096	ERROR - BAD OPCODE ENCOUNTERED.
13	8192	ERROR - STACK OVERFLOW ENCOUNTERED.
14	16384	ERROR - STACK UNDERFLOW ENCOUNTERED.
15	32768	ERROR - SOFTWARE INTERRUPT TABLE FULL.

The SFLAG Command

The SFLAG serial command can be used to turn on a specified output or to set a specified user flag.

Packet sent to the DeltaMax:

[Header]SFLAG,[NN][Checksum]

NN: A 2-byte value indicating the flag number to be set.

The ***packet body length*** of this **Packet** will always be 8 bytes.

***Packet returned by the DeltaMax*:**

***** None *****

The SINFO Command

The SINFO serial command is used to define the size of the Program Area and the four distinct Data Areas.

Packet sent to the DeltaMax:

[Header]SINFO,[PP][IC][FC][IV][FV][Checksum]

PP: A 2 byte unsigned value indicating the length of the PROGRAM AREA.

IC: A 2 byte unsigned value indicating the length (in bytes) of the INTEGER CONSTANTS AREA.

FC: A 2 byte unsigned value indicating the length (in bytes) of the FLOATING POINT CONSTANTS AREA.

IV: A 2 byte unsigned value indicating the length (in bytes) of the INTEGER VARIABLE AREA.

FV: A 2 byte unsigned value indicating the length (in bytes) of the FLOATING POINT VARIABLE AREA.

The ***packet body length*** of this **Packet** will always be 16 bytes.

***Packet returned by the DeltaMax*:**

***** None *****

IB-11A003

INDUSTRIAL INDEXING SYSTEMS INC.

626 FISHERS RUN
VICTOR, NEW YORK 14564

(585) 924-9181
FAX: (585) 924-2169