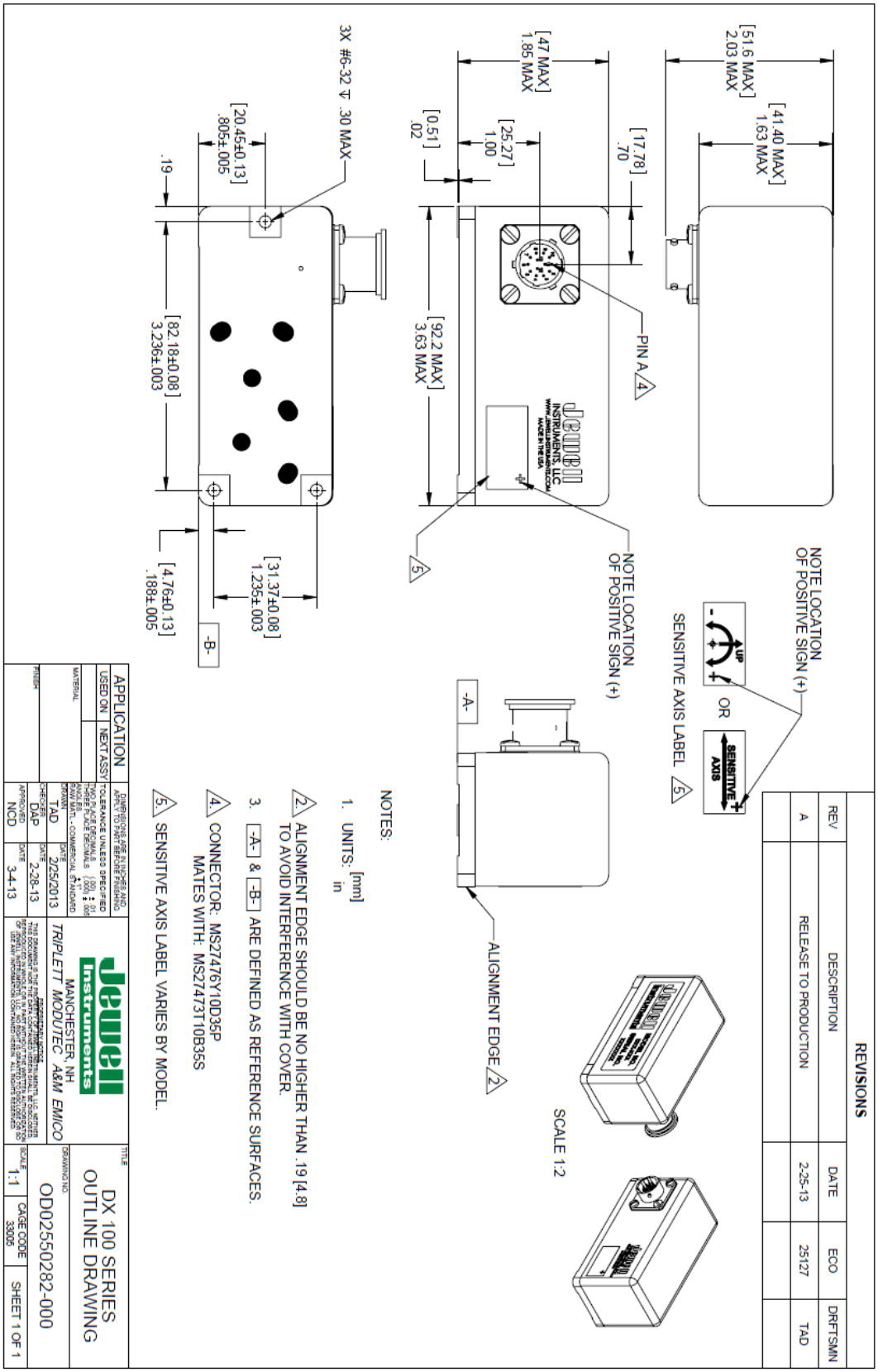
DX Series

User's Manual

For DXA-200/100 Accelerometers and DXI-200/100 Inclinometers

Jewell Instruments, LLC
850 Perimeter Road
Manchester, NH 03103
Telephone: 800.227.5955 **Email: Sales@JewellInstruments.com**

Contents


REVISIONS
REV
DESCRIPTION
DATE
ECO
DRFTSMN
A
RELEASE TO PRODUCTION
2-25-13
25127
TAD
[51.6 MAX]
2.03 MAX
[41.40 MAX]
1.63 MAX
[47 MAX]
1.85 MAX
[25.27]
1.00
[0.51]
.02
[17.78]
.70
[92.2 MAX]
3.63 MAX
PIN A
4
NOTE LOCATION
OF POSITIVE SIGN (+)
5
NOTE LOCATION
OF POSITIVE SIGN (+)
OR
SENSITIVE AXIS LABEL
5
SENSITIVE
AXIS
-A-
ALIGNMENT EDGE
2
SCALE 1:2
3X #6-32 ∇ .30 MAX
[20.45±0.13]
.805±.005
.19
[82.18±0.08]
3.236±.003
[31.37±0.08]
1.235±.003
[4.76±0.13]
.188±.005
-B-
NOTES:
1. UNITS: [mm]
in
2. ALIGNMENT EDGE SHOULD BE NO HIGHER THAN .19 [4.8]
TO AVOID INTERFERENCE WITH COVER.
3. -A- & -B- ARE DEFINED AS REFERENCE SURFACES.
4. CONNECTOR: MS27476Y10D35P
MATES WITH: MS27473T10B35S
5. SENSITIVE AXIS LABEL VARIES BY MODEL.
APPLICATION
USED ON
NEXT ASSY
MATERIAL
FINISH
DIMENSIONS ARE IN INCHES AND APPLY TO PART BEFORE FINISHING
TOLERANCE UNLESS SPECIFIED
THREE PLACE DECIMALS ± .005
TWO PLACE DECIMALS ± .01
ANGLES ± 1°
DRAWN
TAD
DATE
2/25/2013
CHECKED
DAP
DATE
2-28-13
APPROVED
NCD
DATE
3-4-13
Jewell Instruments
MANCHESTER, NH
TRIPLETT MODUTEC A&M EMICO
TITLE
DX 100 SERIES
OUTLINE DRAWING
DRAWING NO.
OD02550282-000
SCALE
1:1
CAGE CODE
33006
SHEET 1 OF 1

1. General Information

This section will provide general notes and conventions about the product, its manual, and any other resources which may be available for assistance.

A. About this Manual

This manual will cover the installation and operation of a Jewell Instruments DX Series force balance digital sensor. Topics will include Operations, Specifications, Product Information, Best Practices, and more, all related to the DX Series sensor. In order to best understand the DX Series Sensor it is recommended to place the Unit under evaluation conditions and preform commands simultaneously with the user's guide.

i. Related Documentation / Applications

In addition to this manual there are additional resources that are available for interfacing with your DX Series Unit. The two additional resources in addition to the user guide would be:

- DX Series Quick Start Guide
- DX Series Interface Platform Application

The DX Series Quick Start Guide is an abbreviated version of this manual for fast set up time providing only the minimum of information in order to begin operation with the DX Series Unit. It should not be used as a substitute for this document however could prove useful for those new to the unit or where a reminder set of information is helpful.

The DX Series Interface Platform is a pre-packaged Labview application/vi which automates most of the functionality of a DX series into a GUI for simple operation. The Interface Platform is available for distribution via the Sales Department of Jewell Instruments or your local distributor at no cost. Please contact the distributor or Jewell Instruments Sales at 603.669.6400.

ii. Documentation Conventions

Throughout this manual certain language or mathematical conventions will be used in order to accurately represent and evaluate outputs. These conventions may break traditional grammar and spelling behaviors, including abbreviations, bitwise and hexadecimal representation with prefixes, etc…

- Hexadecimal notation for inputs and outputs is shown throughout this manual will be shown by a leading '$' or '0x' prefix followed by the MSB. Binary inputs will be designated as '0b' followed by the binary output MSB first.

  Example: 0xA6 = $A6 = [0b1010 0110]

- C programming language conventions are used in this manual to give practical decoding examples.

- RS485 Transmission protocol and the corresponding RS485 operational mode are referred to as 485. The same convention is used for RS422 and the corresponding RS422 Operational Mode as 422.

- The use of words and phrases which have an underline refer to executable commands for which further information is provided. They will most be used in the Operations section of this manual as well as Appendix A: Condensed Command List.

  Example: "In order to switch from 485 to 422 preform a RESET after saving to flash with the DX series unit in the correct operational mode

- When referring to the DX Series Model for communication, installation, or other functions the following alternative words and phrases may be used: UUT (Unit Under Test) , DUT (Device Under Test), DX Series Unit/Sensor.

- When referring to the Labview VI, Computer Application, PC, or other receiving body which is the primary destination for data from the DX Series device the following words and phrases may be used: Host, Terminal, Function, Application, Program.

B. Warranty

Seller warrants its products to be free from defects workmanship and material. Seller's liability is limited to replacing the instrument, or any part thereof, that is returned by the original purchaser, transportation paid, to the factory within on (1) year after date of shipment, provided that the Seller's examination shall disclose that a defect existed under proper and normal use. Seller shall not be responsible for consequential damages of any kind. This warranty is in lieu of all other warranties, liabilities or obligations to the Buyer or any other person.

2. Product Overview

The DX series is a Force Balance based approach to acceleration and inclination detection. It builds upon the established force balance servo based sensing system Jewell Instruments is known for and adds a digital backend for enhanced functionality. It utilizes a dedicated microcontroller for individual axis control, as well as error detection and multiple unit installation on a single bus.

A. Introduction to the DX Series

The DX Series by Jewell Instruments is comprised of Digital Accelerometers and Inclinometers. The sensors can have either single or dual sensitive axes on a single digital output bus. Each unit's analog front end works with the digital interface to produce data in one of two transmission modes which communicate via a 2-wire differential line. When using RS485 protocol (default) operating in half duplex each unit is capable of 2 way communication. In the inverse configuration the unit is operating in an emulated RS422 mode, more about this in the Operating modes. Transmissions to and from the device are in a custom binary format which require encoding and decoding for each operation or output. The output type of the unit is in either thousandths of a degree (DXI) or a fractional acceleration (DXA) format as outlined in each sub-section of this document DXA/DXI Decoding.

In addition to those digital characteristics each unit can take a wide single supply range of 10V to 30V. The unit connects to external sources by a 13 pin locking barrel plug; more information can be found in the Mechanical section. By default DX Series units are shipped in the following configuration: RS485, 38.4k Baud Rate, with No Averaging, Normal Polarity.

i. Features and Options

The DX series sensor enhances the capabilities of the standard servo output of our other product offerings. Each DX series unit has the capability of adjusting several out its output, transmission and data manipulation characteristics to best suit each individual application. In addition to the standard series of features each DX series model has a CENELAC / AREMA Model. The type "–R" models CENELAC and AREMA model is qualified for the following abbreviated standards:

CENELEC EN 55022:2010

CENELEC EN 50155:2007

CENELEC EN 61000-4-8:2010

AREMA Part 11.5.1

For the complete test plan and results please contact Jewell Instruments Sales Department.

ii. Internal Diagram

The core of the DX Series is based on the same force balance sensors which drive Jewell's analog sensor offerings. This front back end consists of a torquer, servo, position detector, and various signal conditioning circuits. The analog back end then is fed into the microcontroller and control system as shown in the following Figure 1: DX Series Top Level Hardware Block Diagram.

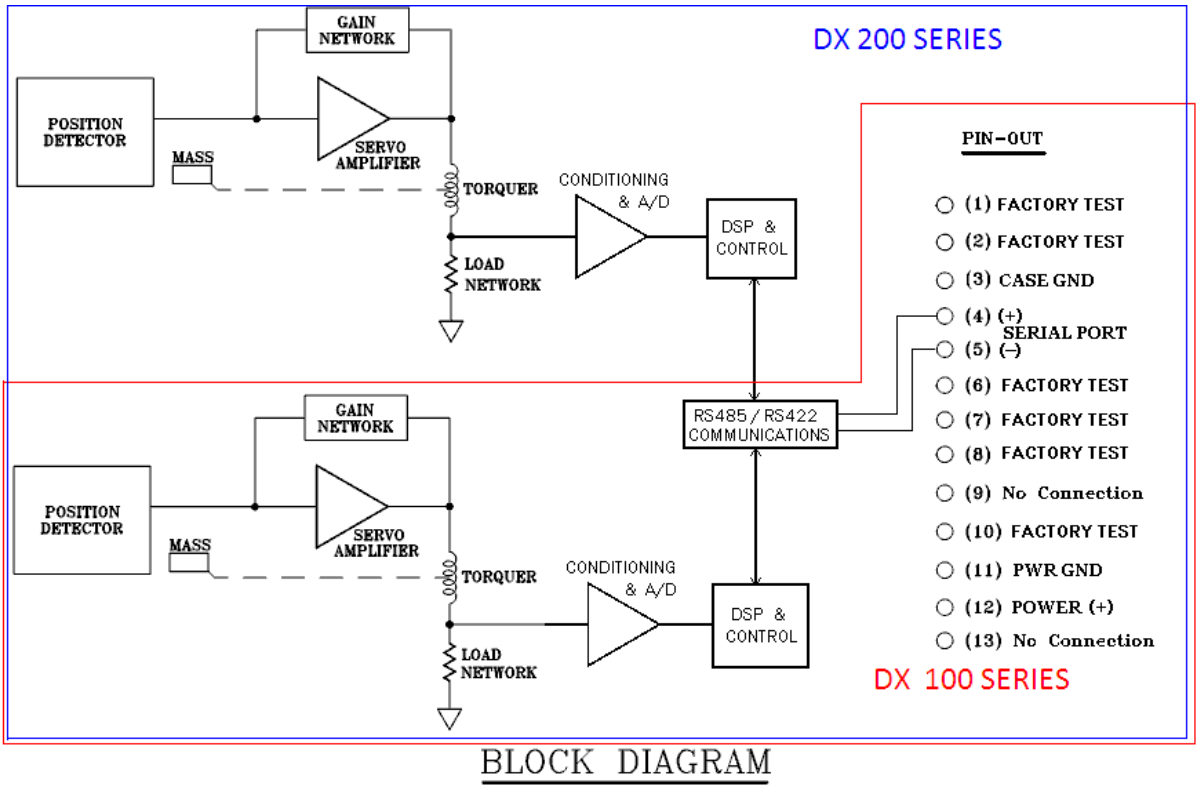


Figure 1: DX Series Top Level Hardware Block Diagram

Note: The analog / digital relationship is one way as the DSP does not actively modify the analog back end. It does however play a role in the conditioning and A/D sub-circuit.

iii. Typical Applications

The Jewell instruments DX series is suited for a wide range of applications and functions. Its differential line transmission allows for greater distance from unit to host and vice versa. For DXA/DXI applications we recommend it for the control and monitoring of the following types of systems:

Radar/Antenna	Structural	Linear Acceleration and Deceleration
Automatic Train Position	Seismic	Platform Leveling

Of course the applications for the DX series are not limited to these fields, but have potential anywhere a linear acceleration or incline needs to be accounted for.

B. DX Series Installation

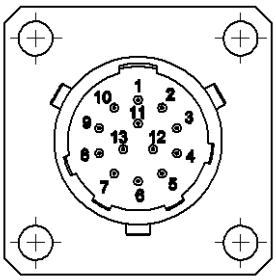
This section will cover the electrical and mechanical installation of the DX Series digital senor. It will provide diagrams and other illustrations to assist in the initial physical setup of the sensor. All functional elements of the unit including communication, operating modes, and decoding will be covered in the Operations section of this User Guide.

i. Mechanical

The DX series was designed for maximum compatibility with the Jewell Instruments LCF-2330 Sensor therefore it uses the same form factor and dimensions. It is a sealed unit with a military style locking barrel plug and an IP rating of 67. In addition to its static response, the sensor is designed to resist corrosion and is unit is rated to IP67 and 1500g, 1msec, ½ sine shock tolerance.

1. Matting Connectors

The standard connector on any DX series model is a 13 pin locking barrel plug with male pins, model type: MS27476Y10D35P.

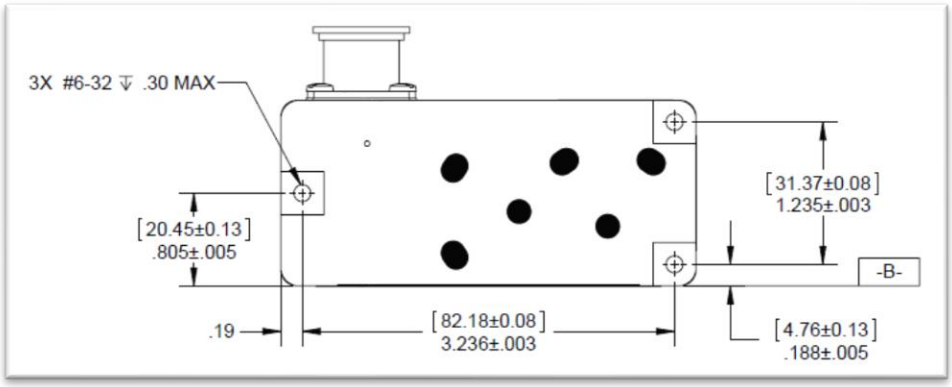


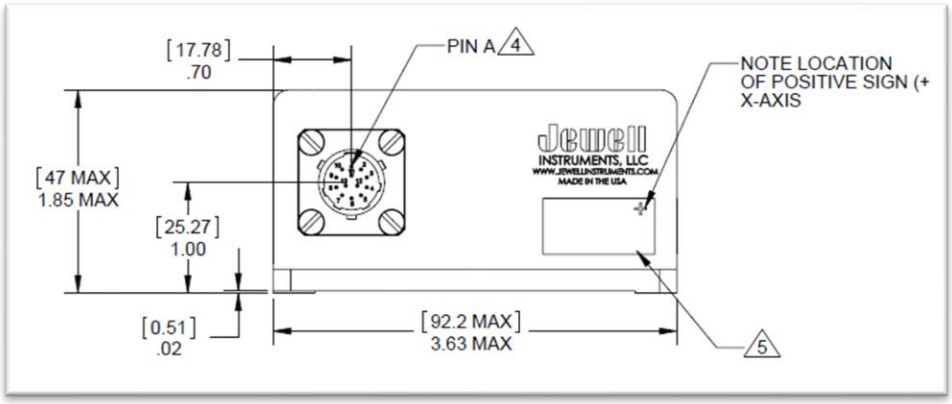
CONNECTOR: MS27476Y10D35P
MATES WITH: MS27473T10B35S

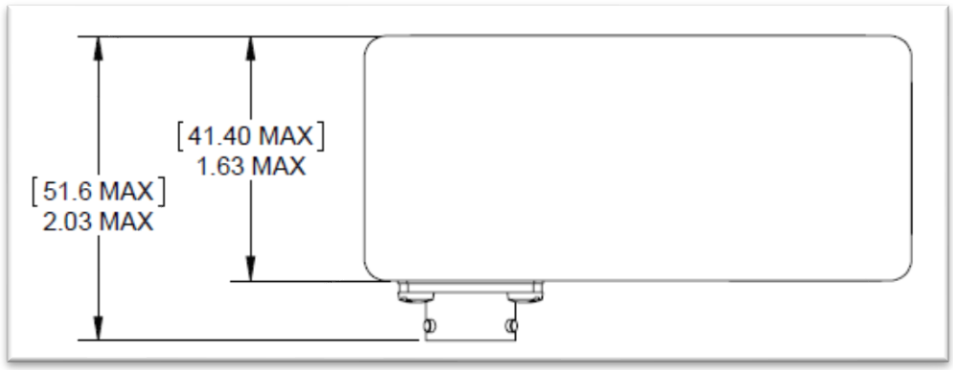
The interlocking female inverse for the unit's connector is: MS27473T10B35S or equivalent based on the specific application. The exterior of the unit is non-conductive however each housing on the connector.

2. *Unit Dimensions*

The sensor is tightly packed within the enclosure to allow for the smallest footprint. Below you can see a partial engineering drawing and its dimensions for the DX series unit.







3. Alignment Surfaces

In order to achieve the most accurate results from your DX series unit, mounting should be done with reference to the reference alignment edge. Those surfaces are used to define the calibration and are recommended as a best practice for your DX series unit.

ii. Electrical

The DX Series operates from a single positive supply voltage from 10-30 V with average current consumption of approximately 50mA for DX-200 series models.

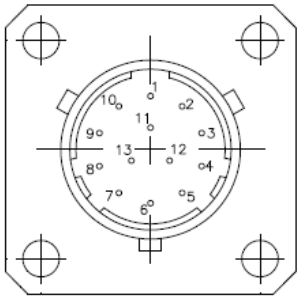
Connections should be made via shielded cable having a twisted pair for the Serial I/O leads for maximum noise isolation. Non-twisted pair cabling can be used in applications with lower point to point transmission distances. The shield should be connected to Case Ground at the device end. To avoid creating a ground loop, if there is continuity or a voltage difference between the device case and the chassis of the RS422/RS485 adaptor at the Host end, the shield should be left unconnected at that end. In noisy electrical environments, a clamp-on Ferrite sleeve (common-mode choke) suitable for the cable used is recommended at the device end of the cable. Good EMI-preventive wiring practice is to route the cable along conductive structures so as to minimize the area enclosed by the cable and other paths between the Device and the Host. A line termination resistor matching the Z_0 of the twisted-pair is needed at the Host end. If RS485 mode is used, the Device end must likewise be terminated. If RS422 mode is used, the device end does not require a termination resistor. However, installing one there will ensure communications integrity should the need arise to send the device commands to change the configuration settings.

A bias network is commonly used in RS485 installations to maintain the line at Mark level when no node is transmitting. The DX series does not require bias, but your RS485 adaptor might – follow the recommendations given for the adaptor.

1. Wiring

The standard wiring for a DX a series consists of 5 of the 13 pins. The remaining are used for factory test / unpopulated and should remain connection free to ensure proper operation of the DX series. For CENELAC/AREMA Type R DX series units, Pin 3 is unpopulated and a no connect in order to comply with standards.

Pin	Function
3	Case Ground (N/C on R type models)
4	[D+] / [Serial I/O+]
5	[D-] / [Serial I/O-]
11	Supply Return
12	Supply V+



2. Power Supply Requirements

The best power supply for the DX series is a linear supply set to any voltage within its operating range of 10 to 30V (Single Sided). Switching power supplies will work as well however it may produce additional noise on the output lines, this noise in most cases will be insignificant. Theoretically each DX unit can be run from a set of batteries which meet the minimum voltage requirements when near depletion. This scenario has not been tested and no resulting effects have been documented.

3. Basic Operating Tenants

The Operations section is going to cover the general operating principals of the DX series, operational modes under which your DX Series can function, output configuration options and non-format functionality. In addition to the different operational modes and transmission protocols, the DX 200/100 Series offers more functionality that can be used in order to customize the unit to its specific application. The modifiable parameters can be made to adjust the measurement output, transmission parameters, and maintenance based effects.

A. Getting Started

The factory default for any DX series is 485. This mode's primary characteristic is a need to send input to the unit to enact change or receive data as a response to query. Therefore upon initial installation the DX series will not be transmitting until otherwise engaged by the user. The quickest test of the DX unit is a Poll command which will return a 7 byte data packet or an ENQ which will return a "hello world" equivalent to a terminal emulator. The DX Interface platform can be used to quickly automate and configure a unit without binary encoding/decoding.

i. Default Settings

By default, each unit is shipped in the following configuration:

Operating Mode: RS485
Baud Rate: 38400 Baud
Output Rate: 90 Hz/60 Hz (DXA/DXI)
Parity Bit: None
Character Length: 8 bits
Stop Bit: 1 Stop Bit
Averaging: None
Polarity: Normal

ii. Terminal Programs

In order to avoid writing an application from scratch when not using the DXI-DXA Interface Platform programs, several terminal programs are available to quickly and efficiently communicate with a unit in a binary/hexadecimal format. Jewell Instruments LLC did not develop these programs and are not in charge of maintenance. Terminal programs we recommend are below:

Program Name	Source	Cost/License
Realterm	realterm.sourceforge.net	Free
Docklight V2.0 (Evaluation Copy)	docklight.de	Free Evaluation Edition

With any 3rd party software there are limitations to the capabilities. But these have been used in the development and troubleshooting of the DX series in the past so they have some of the key features required for communication and troubleshooting. The '$' hexadecimal notation was based on the realterm standard procedure for inputs.

iii. Connecting to the Serial Interface

The serial interface can be integrated into an embedded system or connected directly to a terminal interface. With all serial interfaces make sure the logic levels are compatible with all equipment on the bus to prevent damage to the master as well as all slaves. Pin outs and electrical diagrams are within the electrical subsection of the DX series installation section. If you are having trouble communicating with the unit, one of the first things to check is the differential lines to ensure their polarity is not reversed in error.

B. Operational Modes

Under normal operating conditions the DX series only operates in one of 3 operational modes: RS-422, RS-485, or Temporary RS-485. Both RS-422 and RS-485 are permanent modes; they will remain the active mode until otherwise intentionally switched. The Temporary RS-485 operational mode is an intermediary that can be activated in a unit which is already set to RS 422 operational mode. These transmission modes dictate method of retrieving and inputting data into the device. The operational modes for the unit are primarily controlled through the RS-422 Emulation parameter and

can only be changed in one of the two RS-485 modes. In short, the DX series unit always starts in RS485 mode while it loads parameters and preforms memory checking functions. After a period of about 30mS, the unit completes its startup routine and determines if it should be operating in RS422 mode. If so it shuts down its Rx ports and begins to send data immediately. Once the unit enters this emulated 422 state it does not have an escape routine and will not respond or otherwise react to any other action on the serial line. In order to escape the 422 operational modes, the unit must be issued a break command, more on this later in the Using DX Series Command subsection.

i. RS422

In the RS-422 operational mode, each unit is sends data periodically from its axis/axes without being polled, and the serial line is never floating. For a DX 200 unit it will send at an output iteration will be a Twin packet that corresponds to the same approximate time interval. In RS-422 mode the default output rate is 90Hz per axis for DXA models and 60Hz for DXI Models at all baud rates. The low pass filter is tuned to 30Hz in most DX Models. Exceptions include the 1° and 3° DXI models, in which filter is targeted at 3Hz and 6Hz respectively. The output sample rate can be decreased via input commands while in one of the RS-485 modes. Installing multiple units operating in RS422 mode on the line has a high probability to create data collisions and such a setup is not recommended.

ii. RS485

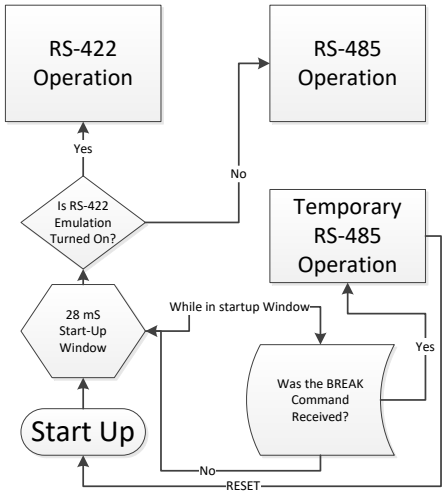
In the RS485 operational mode, a unit only sends data when polled at its unique address. These unique addresses allow for up to 30 units on a single bus; and each poll command sent can request data from either X, Y, or both axes. In the event that 2 units on the bus have the same address both will respond and the risk of collision is high. In addition, should the two packets arrive successfully the packets will be non-differentiable to the originating sensors.

In the case that data from both axes are requested, the unit sends a Twin Packet. A Twin Packet consists of data from both axes send sequentially, without releasing the line. After the Twin Packet completes the unit releases the differential line. Twin Packets do not necessarily correspond to the same time interval but rather the most recent calibrated digital filter output. While in the RS-485 operational mode, the unit is capable to change any of the operating parameters via input commands.

iii. Temporary RS485

In the temporary RS485 operational mode, the unit behaves identical to "normal" RS485 mode with one exception. A unit which was placed in temporary RS485 mode from RS422 and then is either RESET or power cycled it will default to RS422 immediately unless Emulation was turned off and saved to flash. In addition temporary RS485 mode is not reachable when a unit is initialized in "normal" RS485 mode. The

break command can be applied but it will have no effect on the unit's behavior. Below is a flow chart containing the operational modes relationships along with outside influences.



C. Serial Communications

The DX series uses a fixed serial protocol type compatible with RS-232/COM port communication on desktop PC systems. Mark Level is asserted slightly more than 1 character time before the start of a transmission to allow any false start bit due to switching the driver on to clock out of the receiving UART. See default setting in the previous section for configuration details.

i. Packet Structure

The generalized data packet is as follows, however this structure varies in length based on the data being sent or received. Below is a brief explanation of each portion of the data packet for purpose, range of length, and calculation.

| Prefix | UAID | Content (arguments, data, etc...) | Checksum |

All packets to and from the device, follow this format. Prefix, UAID, and Checksum are always one byte in width. The Content Byte(s) are of varying length and contain all functional elements of commands or outputs. The Content bytes can be 0 bytes wide. In DX 200 series units, the name given to dual axis information requested via a single unified command or output iteration is called a Twin Packet.

1. Prefix/Sync

The 1st byte is known as the Prefix/Sync; this byte serves two purposes it indicates the packet type and serves as a sync byte. It is always a variation of $A_ and the second nibble is determined by data length.

The set of Prefix values {**$A0**, **$A3**, **$A6**} is reserved for messages **from** the unit.

$A0: Variable Length
$A3: 4 Byte Response
$A6 7 Byte Response / Measurement Output Packet

The set {**$A9**, **$AC**, **$AF**} is reserved for messages from the master **to** the unit.

$A9: Short Command / Poll / 3 Byte Command
$AC: Long Command / 4 Byte Command
$AF: Extended Command / 5 Byte Command

In output only driven operation (485), only the **$A9** (poll) and **$A6** (data) packets are involved. The remainders are for the command and response sets. All types except **$A0** have fixed packet lengths. **$A0** packets have variable length and the 3rd byte's value or first byte of the content bytes is the total length of the packet.

2. *UAID*

UAID is the unit/axis address and is always the second byte in the sequence. In packets from the device, b0 always indicates the X axis and b1 the Y axis. A command to the device may be addressed to either or both axes (Adress+b1+b0) according to b0 and b1. A command addressed to no axis, will result in no action being performed by the unit. The UAID is never more than 1 byte in length.

If the unit is configured as RS422, the UAID byte serves mainly to identify the axis; given the com port will be dedicated to the one constantly streaming unit. The initial device address field is $70 (8 bit left aligned) or $1C (right aligned). Thus, if unit has never been assigned a new address, the default UAID for X-axis is $71 and the default UAID for Y axis is $72.

a. Addressing

In order to allow for multiple units on a bus when the DX series is operating in 485 mode; each unit on the bus is given an address. The address is a combination of two parts; a 6 bit unit address and a 2 bit axis identifier. In total the combined address is 1 byte wide. The 6-bit unit address portion ranges from $01 - $27. When combined with the 2-bit differentiation the addresses range from $05 - $9D as the 2 bit axis identifier occupies the 2 least significant bits. See below for example.

Example: $01 (6 bit) => $04 (6 bit left shifted by 2 in a 8 bit)
+ $01/$02 (Axis Identification)

$05/$06

Or

0b000001 (6 bit) => 0b00000100 (6 bit left shifted by 2 in a 8 bit)

```
                                +       0b00000001   (Axis Identification)
                                ------------------------
                                        0b00000101

                0b000001 (6 bit) =>  0b00000100   (6 bit left shifted by 2 in a 8 bit)
                                +       0b00000010   (Axis Identification)
                                ------------------------
                                        0b00000110
```

```
                        $1C (6 bit) =>          $70             (6 bit left shifted by 2 in a 8
bit)
                                +       $01/$02         (Axis Identification)
                                ----------------------
                                        $71/$72
Or
                0b011100 (6 bit) =>  0b01110000   (6 bit left shifted by 2 in a 8 bit)
                                +       0b00000001   (Axis Identification)
                                ------------------------
                                        0b01110001

                0b011100 (6 bit) =>  0b01110000   (6 bit left shifted by 2 in a 8 bit)
                                +       0b00000010   (Axis Identification)
                                ------------------------
                                        0b01110010
```

---

3. *Content*

Any data, command or argument bytes follow, LSB first in the Content section. This portion of the Packet Structure can be anywhere from 0 to 20+ bytes depending on the function being executed and the resulting response. In addition the content bytes have no fixed data format, the contents can be binary, ascii, non-decoded Hex/integer, etc… Most commands are 1 or 2 bytes in length, containing a functional arguments and parameter selection or value. As briefly discussed before, The poll function does not rely on any functional argument therefore contains no content bytes, its special case can be considered that the functional argument is contained within the Prefix. The longest response will be that to ENQ or Ping command as an ASCII output detailing unit specifics. Specific response lengths can be found in Appendix A along with typical executions and corresponding responses.

4. *Checksum*

The last byte in the generalized packet structure is Checksum and serves as a non-parity data corruption detector. It is always one byte long and is included on transmissions to and from the DX series device. Checksum calculations can are DX

series specific so care should be taken to follow the procedure for checksum's on messages to the device. Each DX unit will verify a valid command via the checksum byte. When receiving data comparing them against the included checksum is not required and plays no part of the decode cycle however we recommend it for result accuracy. The method for calculating checksum is detailed in the Building Command Structures section later in this user's guide.

ii. Using DX Series Commands

This section is going to cover the specifics of structuring DX series commands for delivery and execution by the sensor. This includes how the unit verifies each received packet as a valid command via the checksum calculation and other information related to messages to the device.

1. Type of Commands

a. Short

Short profile commands consist of 3 bytes: a pre-fix, address, and checksum. There are no content bytes in the short profile therefore only has one command; POLL. The poll command requests data out of a unit in RS485 operation. POLL is the only command which uses the prefix $A9 then the UAID and finally the checksum.

b. Long

Long profile commands consist of 4 bytes: a $AC prefix, UAID, 1 content byte, and the checksum. Commands under this profile generally have I/O purposes rather than ranged effects though there are some exceptions. Some of the commands under this category are RS422 emulation on/off, save to flash, and activating/deactivating averaging. Full listings of commands which use this profile are in this manual under Operations, List of Serial Commands.

c. Extended

The Extended profile is 5 bytes long, consisting of the following: Prefix ($AF), UAID, 2 Content bytes, and the checksum. This command set is used for setting options which have multiple selections both discrete and ranged. Ranged options include Output Rate in 422 or Setting Max Averaging Time Constant.

2. Building Command Structures

The first 3 sections of the generalized packet structure are independent, only checksum is dependent on the contents of all previous bytes. In order to build a complete command, Take each section of the Packet structure and concatenate it with the previous in order to generate a transmission string that can send to the unit at once. Example:

	Hexadecimal Byte
Type of Command: Long	$AC
UAID: Default Dual Axis	$73
Contents: Ping	$B7
Checksum	$28(Calculated)

By concatenating the strings results in: AC73B728 sent sequentially as an uninterrupted serial input to the device. The most difficult part of generating a successful command is creating the checksum for data verification by the unit. The unit always compares a received transmission to its included checksum and will reject and clear the buffer of any data associated with a non-verified packet.

a. Calculating Checksum

In order to calculate the checksum; first sum all the preceding bytes (Prefix through the last content byte) in hexadecimal notation. Add to this the number of carries from the high byte into the low byte of the accumulator or use the 3rd LSB Nibble if calculating checksum manually. Ignore any carry from this addition. Using only the least significant byte of the result, preform a binary inverse or one's compliment.

Example #1:	$AC	PREFIX (Extended Profile)
	+ $03	ADDRESS (Broadcast)
	+ $03	CONTENT (RESET)
	$B2	SUM OF ALL BYTES
	+ $00	HIGH BYTE (No carry from addition past 8 bits)
	~$B2	Negation / One's Compliment
	$4D	FINAL CHECKSUM VALUE

Example #2:	$A9	PREFIX (Short Profile)
	+ $71	ADDRESS (Default X axis only)
	+ NULL	CONTENT (Poll requires no content)
	$[01]1A	SUM OF ALL BYTES
	+ $[01]	HIGH BYTE (No carry from addition past 8 bits)
	~$1B	Negation / One's Compliment
	$E4	FINAL CHECKSUM VALUE

For some pre-calculated checksum's please refer to Appendix A.

4. **Building Command Structures**

Device does not respond to the message if it gets a checksum error. Commands must be sent as continuous strings at the selected baud rate. Any gap within the string

will cause a timeout. The unit will also behave erratically if an extra stop bit is used. Commands which do not trigger replies can be concatenated to simplify the host's string table, but its recommended to separate them with 2 or more nulls or rubout's ($FF), especially at max baud rate. Replies to most commands begin within 2 character periods after the end of the command string. There is one exception: Reply to Update Configuration is delayed approx. 32 milliseconds to allow time for Flash Memory Write in both axes. Communications are not receivable / ineffective during the write period for flash altering commands.

A. Full List of Serial Commands

This section will cover each command that is accepted by the DX series, its effects, subsequent considerations, and a brief implementation summary. Each command that can be issued to the DX series falls into one of three super classes organized by functional group. The three groups are Configuring Measurement Outputs, Configuring Transmission Parameters, Maintenance and Infrastructure Inputs. The serial commands available to each DX series user are aimed to provide the most customizable experience to suit a specific application.

i. Configuring Measurement Outputs

The Configuring Measurement Outputs section of the DX series will cover commands which directly affect the quantitative value of data received from the unit. Topics to be covered include Averaging Functionality with functional states & Polarity reversal and indication, as well as the implementation of the jointed averaging commands.

Changing Polarity

Dependent on the Host, documentation, environment definitions, unit mounting and/or if the application calls for a negation, the data output polarity may be reverse of intended polarity can be switched via the serial interface without the need for physical manipulation of the unit. This function eliminates the need to tap new connection points or other physical corrections. In order to do this a Select Normal Polarity (Default) or Select Reverse Polarity needs to be executed via the serial interface. The argument for changing polarity can be found below in the content byte of the table:

	Prefix	UAID	Content	Checksum
Select Normal Polarity	$AC	$<UAID>	$C9	$<Checksum>
Select Reverse Polarity	$AC	$<UAID>	$C8	$<Checksum>

This change only affects the sign on the unit for reference to an external plane system. It will not change the amplitude or transient response of the measurement data.

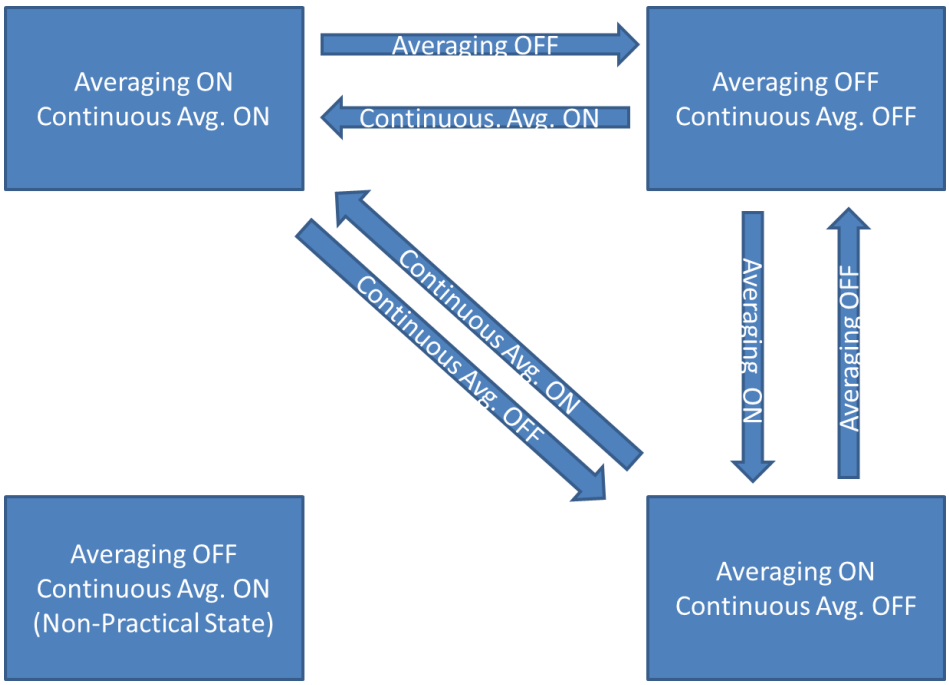
Averaging Functionality

An important set of functions within the DX series unit are the data averaging commands which are used to reduce noise. There are two conditions where the use of the averaging functionality is suggested; when the unit is polled at long intervals in RS485 or in high noise environments. The averaging state of a sensor is determined by 3 parameters within the DX series firmware: Averaging, Continuous Averaging, Max Averaging Time Constant. Continuous Averaging is a secondary modification applied to a unit already under standard averaging. For standard averaging a uniformly weighted average is computed from one output to the next, up to a specified maximum time interval (set by max averaging time constant). Beyond this time interval (continuous averaging), the weighting becomes exponential and the process becomes equivalent to a 1-pole low-pass filter in addition to the fixed frequency digital and analog filters. The set max averaging time constant allow the user to choose a sample inclusion set determined on the output period or desired responsiveness. The averaging functionality of the DX series unit operates under the operation of 5 primary commands, in addition there are a several complex commands which provide the combinations of at least 2 primary commands.

The five primary commands are listed below:

1:	Averaging ON
2:	Averaging OFF
3:	Continuous Averaging ON
4:	Continuous Averaging OFF
5:	Set Max Averaging Time Constant

Commands 1 and 2 operate as imperfect inverse functions. Averaging OFF will set both continuous averaging and averaging parameters to FALSE while Averaging ON will only set averaging to TRUE, the command will not re-enable or set the continuous averaging parameter TRUE if it was previously enabled. Commands 3 & 4 are also not inverses as Continuous Averaging ON affects both the standard and continuous parameters. Commands 4 will not affect the standard Averaging parameter; if the intent is to turn off all averaging please use command 2. Below is a graphical representation of the averaging states as well as the effects of functional commands on those states.



Standard Averaging

Standard averaging uses the summation and division of samples gathered between outputs to reduce noise and aliasing that may be introduced through a variety of sources. Each sample is given equal weight in the average. This technique is more prone to short term influences and can be further controlled through a max averaging time constant or the continuous averaging sub-functionality, for more information on either process/parameter see its specific section. The argument for Averaging ON is $C5, and the argument for Averaging OFF is $C4. Turning on or off Standard Averaging is done through two I/O commands structures when fully concatenated:

Averaging ON	$AC	$<UAID>	$C5	$<Checksum>
Averaging OFF	$AC	$<UAID>	$C4	$<Checksum>

As explained in the flowchart above these functions are not perfectly inversed as they depend on sub-functionality to determine final state.

Continuous Averaging

In addition to the Standard Averaging previously described, the continuous averaging sub-set has a different response and calculation profile. When Continuous Averaging is selected, the average is not reset after each output, averaging is always given an exponential weight with the selected time constant for each additional

output sample. In both Averaging modes, the number of output samples currently in the average is reported in the data packet.

In order to activate continuous averaging, the unit does not need to be in a specific state because the Continuous Averaging ON command will set both the standard and continuous averaging parameter to true. When using the Continuous Averaging OFF command, it will only disable the averaging subset of features leaving the unit in the standard averaging mode. The value of the argument for Continuous Averaging ON is $C6, while its counterpart Continuous Averaging OFF has an argument value of $C7 as shown below in these generalized command strings.

Continuous Averaging OFF	$AC	$<UAID>	$C6	$<Checksum>
Continuous Averaging ON	$AC	$<UAID>	$C7	$<Checksum>

Averaging Time Constant

The Set Max Averaging Time Constant command is used in order to limit the amount of samples in an averaged output packet. It can be used to further define the behavior of averaging functionality for both continuous and basic averaging modes by limiting or expanding the influence of impulse noise. The value of the parameter can also be used in conjunction with Set RS422 Output Period in order to apply effective filter behavior to the unit's output please refer to section Guidelines on Effective Filter Selection. In general, the more samples with a single output packet the more stable and immune to noise the packet will be.

Set Max Averaging Constant	$AF	$<UAID>	$E4	$<Value>	$<Checksum>

Jointed Averaging Commands

There are several jointed commands available to the user for single line modification of multiple averaging operation parameters. These commands always have the same effect as the two commands they are comprised of if executed individually. The following commands are available in a jointed format:

Set Max Averaging Time Constant and Enable Averaging

Command is equivalent to Set Max Averaging Time Constant in addition to Averaging ON except it does not reset the average. Use for one command configuration of a unit which will need an averaging modification.

Set Max Averaging Time Constant and Enable Continuous Averaging.

Command is equivalent to Set Max Averaging Time Constant in addition to Continuous Averaging ON. Use for one command configuration of a unit which will need an averaging modification. Does not reset the Average.

For an example of the implementation of these jointed commands please refer to Appendix A.

ii. Configuring Transmission Parameters

Transmission Parameters are items which affect the Meta characteristics of a data packet are covered in this section however it does not cover commands which affect the value of data packets. These modifiable parameters which change the various attributes of the signal, such as baud rate, RS 422 emulation, output packet rate, response delay, etc… are modifiable via serial commands. Any commands which would affect both a transmission setting and another non-transmission setting within a single string are reserved for the jointed command section of Appendix A.

- *Set Unit Address*

In order to set the UAID to a value each DX series unit includes a set unit address command. This command has a long profile and is similar to the select baud rate command as both the issuing command and its ranged value exist with a single byte. The argument byte reserves the 6 most significant bits for the value of the new address you with to assign while leaving the last two to indicate this is an Assign Unit ID command.

Assign Unit ID	$AC $<UAID> 0bxxxx xx11 $<Checksum>

The range of the input for the Assign Unit ID command is from 0b0000 01 to 0b1001 11, in other forms it can be read 0x01 to 0x27 right aligned. These representations in combination with the argument identifier will serve to create the completed argument in this Long profile command. It should be noted that each dual axis unit requires the same base address in order to operate correctly for the two axes. Below is a table of corresponding UAIDs, arguments, Binary, integer equivalents and right grouped inputs.

Binary	Right Grouped Hexadecimal	Completed Argument	UAID X/Y (0x)	Integer Address
0000 01	01	07	05/06	1
0000 10	02	0B	09/0A	2
0000 11	03	0F	0D/0E	3
0001 00	04	13	11/12	4
0001 01	05	17	15/16	5

0001 10	06	1B	19/1A	6
0001 11	07	1F	1D/1E	7
0010 00	08	23	21/22	8
0010 01	09	27	25/26	9
0010 10	0A	2B	29/2A	10
0010 11	0B	2F	2D/2E	11
0011 00	0C	33	31/32	12
0011 01	0D	37	35/36	13
0011 10	0E	3B	39/3A	14
0011 11	0F	3F	3D/3E	15
0100 00	10	43	41/42	16
0100 01	11	47	45/46	17
0100 10	12	4B	49/4A	18
0100 11	13	4F	4D/4E	19
0101 00	14	53	51/52	20
0101 01	15	57	55/56	21
0101 10	16	5B	59/5A	22
0101 11	17	5F	5D/5E	23
0110 00	18	63	61/62	24
0110 01	19	67	65/66	25
0110 10	1A	6B	69/6A	26
0110 11	1B	6F	6D/6E	27
0111 00	1C	73	71/72	28
0111 01	1D	77	75/76	29
0111 10	1E	7B	79/7A	30
0111 11	1F	7F	7D/7E	31
1000 00	20	83	81/82	32
1000 01	21	87	85/86	33
1000 10	22	8B	89/8A	34
1000 11	23	8F	8D/8E	35
1001 00	24	93	91/92	36
1001 01	25	97	95/96	37
1001 10	26	9B	99/9A	38
1001 11	27	9F	9D/9E	39

Each unit takes its assigned address (right group hexadecimal) and combines it with the two LSB reserved portion for axis identification/argument. In order to determine the no axis UAID (for reference) each unit, the 6 bit input to the Assign Unit ID function should be left shifted 2 places and filled with 0's.

- *RS-422 Emulation*

There are two commands used to change the operational mode of a DX Series: 422 Emulation ON and 422 Emulation OFF. These two commands are inverse functions, exactly negating each other's functional purpose. The commands are used in order to set the value of the 422 emulation parameter TRUE or FALSE within the DX series firmware. 422 Emulation is a software generated state by where the output is constantly driven once the unit identifies the 422 emulation parameter is set to true. This suggests that the unit is always in 485 however the window in which 485 functionality is maintained is small, past that point the unit assumes dedicated 422 operations. The narrow window should not be used for anything accept the BREAK command.

Once the unit starts transmitting data measurement packets without interruption in the emulated 422 state, it will no longer respond to commands as the line will constantly be driven by the output from the unit. While the 422 emulation parameter is set false (OFF) the unit will act a 485 unit only outputting data when polled and capable of receiving and executing commands. A unit currently in 485 mode has the ability to set the 422 emulation parameter True (ON), however the unit will not change to that operational mode until this parameter is saved to flash then RESET or power cycled. The two applicable commands use the Long profile with the prefix $AC. In order to issue the RS422 Emulation ON command the sequence is as follows with the argument of $C3:

RS422 Emulation ON Command	$AC $<UAID> $C3 $<Checksum>

Similar to the immediately above command, the RS422 Emulation OFF is a long profile command however its argument byte consists of the hexadecimal value $C2. A complete sequence can be seen be below:

RS422 Emulation OFF Command	$AC $<UAID> $C2 $<Checksum>

- *Baud Rate*

The DX series unit uses standard baud rates of 19.2K, 38.4K, 57.6K, 115.2K, and 230.4K. They are selectable via the Select Baud rate command listed in Appendix A and in this section. The lower limit is dictated by the ability to send 90 samples per second on a DXA-200 series unit while the upper limit is due to hardware limitation. While lower baud rates are possible for lower output rates or in RS485 mode, there is currently no plan to make them available. The select Baud Rate command operates differently as compared to many of the other Long profile commands; Select Baud Rate is not a binary operation and contains both the argument and parameter in the content byte. The parameter is encapsulated within the last 3 bits of the content bite in an unsigned integer format.

Select Baud rate	$AC $<UAID> 0b10110XXX $<Checksum>

The XXX represents the user selected field. To populate this field follow the following table for the unsigned integer encoded baud rate values:

Baud Rate	Encoded Unit Value	Binary Insertion
19.2K	0	000
38.4K	1	001
57.6K	2	010
115.2K	3	011
230.4K	4	100

Baud rate changes only take effect at RESET after being saved to flash.

- *Output Rate (RS-422 Only)*

Output rate when operating in RS 422 mode is controlled by a user definable parameter within the DX series firmware. This extended profile command uses a parameter to define a wide range of values selectable between 0 (by default) and 255 for the 8 bit parameter/character width. For DXI Models, the unit operates at 60Hz by default (Parameter = 0) while the DXA operates at 90Hz by default (Parameter = 0). The value of the parameter can be converted into an effective operating output frequency using the following algebraic equation:

$$\frac{90}{X+1} = f(Hertz)$$

$$Where\ X\ is\ an\ integer\ between\ 0\ and\ 255$$

Below is a limited selection table for output rates:

DXI Output Rate Table			
Output Freq. (Hz)	Input Parameter	Output Freq. (Hz)	Input Parameter
60	0	5	11
30	1	4	14
20	2	3	19
15	3	2	29
12	4	1	59
10	5	0.5	119
6	9	0.234375	255
DXA Output Rate Table			
Output Freq. (Hz)	Input Parameter	Output Freq. (Hz)	Input Parameter
90	0	6	14
45	1	5	17

30	2	3	29
18	4	2	44
15	5	1	89
10	8	0.5	179
9	9	0.3515625	255

Outputs rate which cannot be generated within the limits and constraints of the equation are not available. In order to issue this command the following structure should be adhered to. This command does not take effect until saved to flash and the unit is RESET.

Set RS422 Output Rate	$AF $<UAID> $E2 $<Parameter> $<Checksum>

- *Minimum Response Delay (RS-485 Only)*

Response Delay is an option built into DX series units in order to add additional time between when a unit receives a command and its response. This operation can be applied to both operational modes; however given the perpetual nature of 422 operations each sample would be delayed the same amount resulting in no apparent effect. The response delay feature is currently designed for use in systems where the master/host is slow to release the differential line after a transmission has been sent. The line additional time added to the response time is a function of this function:

$$Delay > (mS) = \frac{X}{32.768} mS$$

Response delay uses an extended profile command with a full byte of range selection in addition to the argument. The argument is always issued as $CD while the range selection (parameter) is an integer value between 0 and 255. The default value of the parameter is 0 for DX series units. The complete command string can be found below:

Set Minimum Response Delay	$AF $<UAID> $CD $<Parameter> $<Checksum>

iii. Maintenance and Infrastructure Inputs

All non-formatting, non-measurement related functionality fall under the maintenance sub-set of commands and functionality. These are diagnostic and infrastructure related, dealing with mode changes, unit identification, and memory settings.

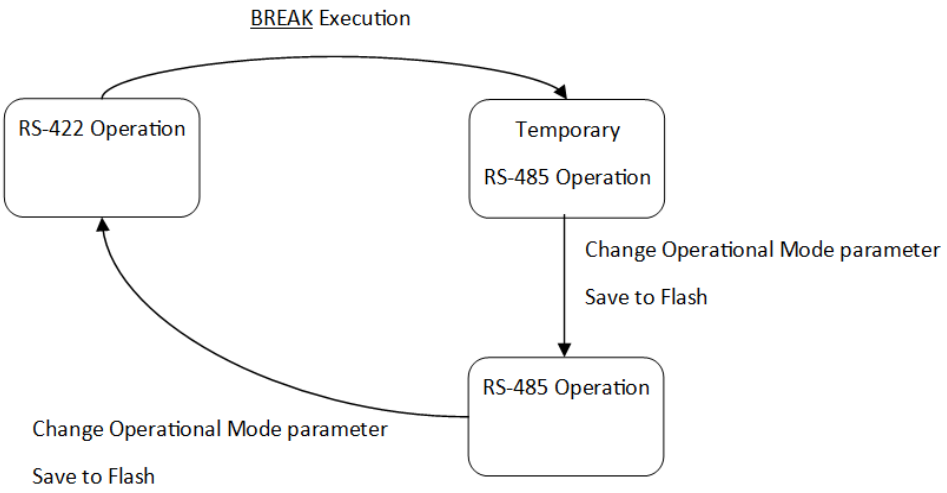
Configuration Segments

The DX series sensor controls its operation based on a set of internal memory banks, these primary and editing memory segments contain all unit specific information assigned to each DX series unit. Upon initialization/power up of the DX series unit, the primary configuration segment is loaded from memory into the active configuration

bank of the DX series sensor. This defines its operational mode and other parameters which affect its output formatting and contents. In addition at unit initialization a DX series unit will load the contents of its primary configuration segment into the editing segment for manipulation. This editing configuration serves as an inactive copy under which changes can be made without compromising unit performance. IE: assigning a new baud rate to the editing configuration without reinitializing the COM system; only enacting change once configuration has been completed. To save most changes from editing to primary memory segments the following commands are used: Allow Update and Update Configuration Commands. The effects of issued commands can vary whether they take effect immediately or not. For more information please refer to each individual command in Appendix A. Re-initialization of the unit can be performed from a RESET command or power cycle.

Break

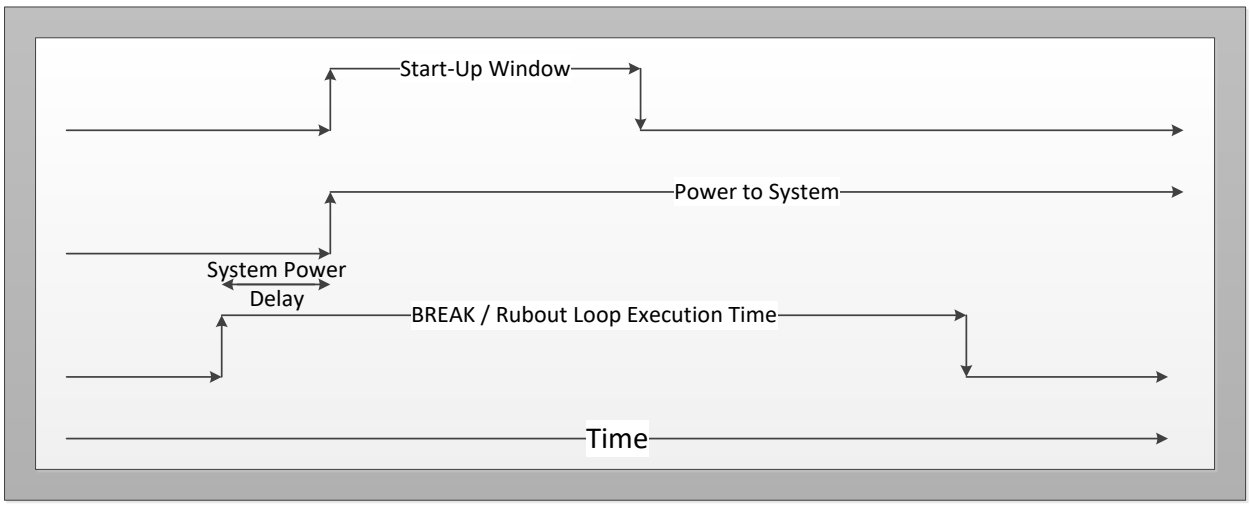
The DX Series unit is always in one of three operational modes: 422, 485 or Temporary 485 also known as T485 as previously discussed; in order to change a unit's operational mode from 422 mode to 485 a few steps need to be taken. The first of which is to place the unit in Temporary 485 mode using the BREAK command. Built into the startup sequence of every DX series sensor is a 28 mS window at startup before the unit begins sending data. This window is where a BREAK command addressed to both axes will put the unit in Temporary RS485 mode. The BREAK command can also be sent in a unit



which is already in RS485 mode; however this will have no visible effect.

As with the standard practices of a RS485/RS422 differential line and dependent on baud rate and line length, sending commands may require a termination resistor at

the device end to work properly. Once a unit is in the temporary 485 mode, it will only remain in that mode until a power cycle or RESET command. The most effective way to apply the BREAK command to a unit under test is done in 3 parts. To start, a cycle of BREAK commands buffered by two or more "Rubouts" or 0xFF should be applied to the unit before power is turned on. Secondly, power should be applied to the unit(s) while the stream of BREAK commands are being sent on the serial line. This will result in break command being acknowledged within the unit's start-up window. The 3rd part would be to stop sending the strings of commands and verify that the unit is not sending data. If the unit continues to stream data the break process has failed and should be repeated.



Above is an action/execution timing sequence for applying the break command to the unit. Below is the generalized command structure for the break command with included Rubout's.

BREAK with "Rub-outs"	$AC $<UAID> $02 $<Checksum> $FF $FF

Ping

Ping is a unique command with respect to the typical response from the device. It does not respond in an encoded fixed length format, however instead replies in an ASCII with the model number, range, and unit ID, and its configuration. This response can vary slightly from model, and set up configurations therefore this command uses the variable length prefix of $A0. The most common use of the ping command is to determine an unknown unit address via the Ping command in broadcast; however other purposes include the identification of range, firmware revision changes, etc… For sake of simplicity the best time to use the Ping command is when operating in a terminal interface for inclusive display of ASCII characters. Below is the BROADCAST definition of a ping command:

Ping/ENQ	$AC $03 $B7 $98

A typical response for the ping command can be found below for a DXI-100-14.5:

DX SERIES +/- 14.5 Rev 2, Axis ID = 71, Single

RESET

Standard on most digital systems, reset provides a method for total system restart without disruption of the power system or physical manipulation. The RESET command for the DX series is identical allowing for the unit to re-initialize the system operating parameters. Each unit is resets individually and should be treated as such with regards to operating mode or expected behavior. For obvious reasons the reset command does not have a response to the input command however post-reset operation will confirm.

RESET	$AC $<UAID> $03 $<Checksum>

Allow Update

Allow update is the first portion of a two part execution for flash memory access. In order to save to the editing configuration to flash memory, the Update Configuration is used. However to protect the Update Configuration command from premature execution, the DX Series firmware required it be preceded by Allow Update. This command serves as precursor mechanism for those protected functions only allowing the execution of the second portion of the command sequence if the first sequence was completed. There are other protected commands which require this pre-requisite however they are exclusively for manufacturing and advance diagnostic works.

Allow Update	$AC $<UAID> $01 $<Checksum>

Update Configuration

Update Configuration is used to commit operational changes to non-volatile memory. While some commands take effect immediately, those changes will not be retained unless an Update Configuration command is executed successfully. Update configuration has the longest standard response time of 32mS due to the operation within the flash memory segment, in which time the editing configuration segment is copied back into the primary memory segment for use on all future initializations.

Update Configuration	$AC $<UAID> $00 $<Checksum>

Query Setting

Query setting is a diagnostic tool for evaluating the value of selectable parameters in the DX series firmware. It cannot be broadcast, and is intended for use to confirm the operation settings of a unit. Most settings are able to be queried using selectable inputs which will return in either an operational parameter or the

configuration byte. The parameter for the Query Setting command is embedded in the argument 0x1011 1xxx where the parameter being queried is determined by the value of xxx, the 3 LSB's of the argument. Below in the table are the corresponding 3 bit codes to each parameter.

Function	Completed Argument	3 Bit Code
Configuration Byte	B8	000
Minimum Response Delay Parameter	B9	001
RS 422 Output Period Parameter	BA	010
Maximum Averaging Time Constant	BB	011
RESERVED	BC	100

The configuration bit consists of the binary status of several parameters at once. Each bit within the byte then represents a particular status or function on or off. For decoding of the status byte use the following table, please be sure to note the active low modification on functions with "~":

Bit	Representative Value or Function
0	Normal Polarity
1	~Averaging Enabled
2	~Continuous Averaging Enabled
7	RS 422 Emulation

When combining the argument byte with the completed command string, the complete issued command appears below.

Query Setting	$AC $<UAID> 0b1011 1xxx $<Checksum>

In addition to querying the configuration byte this command can be used to determine the ranged value of operational parameters used in complex commands.

001: < Minimum Response Delay Parameter >
(Minimum Response Delay setting)

010: < RS 422 Output Period Parameter >
(RS422 Output Period setting – 1)

011: < Maximum Averaging Time Constant >
(Max Averaging Time setting - 1)

All values returned are the values used in the corresponding command arguments.

Send Configuration Vector

The Send Configuration Vector command is similar to that of query command (configuration byte); it provides several parameters via a single execution command. The configuration vector response is comprised of the following values and functional information: if Primary and Editing Configuration segments match, total packet length, baud rate, minimum response delay, the configuration byte, max averaging time constant, output period, and a reserved byte for in house diagnostics. The argument to send the Configuration vector is $BF. Completed the command packet appears below:

Send Configuration Vector	$AC $<UAID> $BF $<Checksum>

The response to this command is a fixed length response as seen below:

1	2	3	4	5, 6, 7, 8, 9, 10	11
A0	UAID	N	X	Vector Bytes	Checksum
Prefix	Address	Number of bytes in the packet	Primary/Editing configuration mismatch and location	See Below	Calculated Checksum

Each of those bytes has a specific operation to them which will be expanded upon below. The "N" Byte gives the number of bytes within the packet; this is standard operation for prefix variable length type A0 responses from the unit. All firmware revision 2 or greater units use an 11 byte packet. This is subject to change in further firmware revisions but will be noted in any updated documentation.

Byte 4 or the "X" byte is used in order to determine equivalency between Primary (Startup) configuration segments and the editing configuration segment. In the event of a mismatch, the numerical value of "X" is the location of the first mismatch within the Vector bytes. This value starts with 1 representing position 5 (baud rate) in the send configuration vector response. The next byte in the response will be byte 2 indicating a (minimum response delay) mismatch, etc. So if the baud rate in the editing configuration was not equivalent to that of the Primary configuration, the value of the "X" byte within the configuration vector would be 0x01. This applies for all values in the

Vector bytes. The vector bytes themselves are representations of various values within the DX series firmware. See below:

Byte	Function
5	Baud Rate Selection
6	Minimum Response Delay
7	Configuration Byte
8	Max Averaging Time Constant
9	Output Period
10	RESERVED

Vector Byte - 5: This value is equivalent to that as selected by the Select Baud rate command. In addition a value greater than 4 is undefined and represents the unit is at the default baud rate of 38400 or vector byte value 1.

Vector Byte -6: The value of vector byte 6 is the negation of the value for the current minimum response delay. In order to determine the time equivalent value of the delay the value should be input into the following equation:

$$Delay\ (in\ milliseconds) = \frac{255 - [\sim Byte\ 6]}{32.768}$$

Vector Byte -7: Byte 7 is the configuration byte. The byte is comprised of several binary indicators stating which modes and functions the unit is currently operating under on a per bit basis. Below is a table of functionality encapsulated within the configuration byte:

Bit	Representative Value or Function
0	Normal Polarity
1	~Averaging Enabled
2	~Continuous Averaging Enabled
7	RS 422 Emulation

Note that parameters with "~" preceding the parameter names are active low signals.

Vector Byte -8: This byte is the value of the parameter set by Set Max Averaging Time Constant.

Vector Byte- 9: This value of this byte is the negation of the assigned output parameter. This parameter is only relevant to DX Series Sensor operation when operating in the RS-422 mdoe. The byte returned by Send Configuration Vector can be converted into a numerical form equivalency output period using the following equation:

$$Period\ (in\ seconds) = \frac{256 - [\sim Byte\ 9]}{90}$$

Vector Byte-10: This byte is reserved for diagnostics and no specific user relevant information should be derived from it.

Polling

In order to extract data from a unit operating in 485, the POLL command is used in order to extract measurement data from the unit. The POLL command is specific to 485 operations and it is the only command that uses the short profile. The Poll Command is assembled as followed and returns a 7 byte measurement packet as a response:

$A9 $<UAID> $<Checksum>

There is no argument in this command as the prefix serves as both the packet length designator and the command being issued. Using the UAID, and calculating checksum have been covered earlier in this document; any questions on their use should be referenced back to the originating sections.

5. Responses from the DX Series Sensor

This section will cover specifics of received packets from your DX series unit. It will also cover the full contents of the 7 byte measurement packet including status conditions and the use of the auxiliary byte. In response to commands sent to the device most exchanges on the serial interface will result in an encoded response to confirm unit preformed requested action, this response can be in many forms: ACK (Acknowledge), NAK (Non-Acknowledge), Measurement packet, ASCII test string, or a function specific encoding.

A. ACK and NAK

The ACK and NAK functionality of the DX series firmware is based on the generalized command and packet structure which follow a constant format for most input commands consisting of Prefix, UAID, function that was executed and checksum. For Extended profile commands the response does not include an echo of the parameter but rather just replies with the primary function call. When executing a dual

axis command each unit responds individually with its own UAID even if the originating command was addressed to both axes. For a list of sample ACKs please refer to the following table:

Command	Hexadecimal Input	Acknowledge (ACK)
Set 19.2K Baud Rate Both Axis	$AC 73 B0 2F	A3 71 B0 3A A3 72 B0 39
Turn ON RS-422 Emulation	$AC 71 C3 1E	A3 71 C3 27
Update Configuration	$AC 73 00 DF	A3 71 00 EA A3 72 00 E9

For commands which adhere to the checksum protocols however do not correspond to executable operations, the NAK is used. The NAK is identical to the ACK except it will return an ones complement of the command issued to indicate operation failure. Not all commands will return an ACK or NAK. These can be due to the nature of the operation such as poll or query setting where in the desired action/information is contained within the response or if the inverted argument would mimic the successful completion of an alternative command. Each ACK/NAK is outlined in Appendix A along with which commands do not provide responses.

B. 7 Byte Measurement Output

Each DX series output measurement packet has the same 7 byte format. This format is the same from both axes in a dual element sensor bringing each Twin packet response up to 14 bytes. The measurement packet follows the generalized packet structure allowing for a prefix, UAID, 4 content bytes, and Checksum. The 4 content bytes consist of 3 distinct sections: 18 bits dedicated to the measurement data, 6 bits dedicated to a status indication and fault condition reporter, as well as a byte/ 8 bit section reserved for secondary information (currently only number of samples for averaged data when active). Below is a complete measurement packet using the individual bytes names and functionally divided by section.

| $A6 | UAID | D0, D1, D2, Aux | Checksum | == Measurement Packet

The prefix/sync for the measurement packet is always $A6, the only response from the unit of this length. The UAID byte is consistent with the rest of the document where an X axis produces an address+01, and y address+02.

D0 is the "status" byte while D2 &D1 are the data bytes, sent least significant byte first. Byte order is "little endian", so D2 is the most significant byte of the 18 bit measurement data within the 7 byte data packet.

The 18-bit measurement value is left-justified in the 3 data bytes, D2, D1, D0. The 2 MSB's of D0 are the LSB's of the data and the rest of D0 are status indicators bits within the Status byte. The DX series unit uses a different data encoding within the18-bit output depending on the base model of Inclinometer or Accelerometer. DXA models use a fractional representation while DXI models use a signed incremental notation to store the measurement information. In subsequent sections this manual will outline the data formats for each respective unit.

D0 is referred to as the Status byte, but it also contains in its MSB's (b6, b7) the LSB's of the 18-bit data. In normal operation, Status bits 4 & 5 are 0 and bits 1 & 2 indicate non-default setup modes. Bit b1 indicates Reverse Polarity (rotation direction) and b2 indicates Averaging Mode. If a Memory Checksum error is detected at startup, b4 is set and bits 1 & 2 are commandeered to indicate what part of memory (see memory verification section). Bit 0 of the status byte is a flag to indicate that the analog front-end may have been in saturation during some portion of the input for this output sample. If Data Averaging is turned on, the saturation flag is not maintained ON for the additional output samples potentially affected.

When Averaging is enabled, the Aux byte indicates the number (or effective number if exponential averaging) of output samples currently in the average. If unit is polled a 2nd time before the next filter output and Averaging is not in Continuous mode, the Aux byte is 0 and the data value is the most recent filter output which was part of the previous average. The Aux byte's meaning when Averaging is not enabled is undefined. It is kept in the packet to avoid having a different packet length when Averaging is on. Checksum is computed as previously described.

i. DXI Decoding

The measurement value is stored in the 18 bits as 1 designated sign bit (MSB) and 17 bits of unsigned integer, each of which represents one thousandth of degree with one designated sign bit. If this value is being imported to a C based system, integer sizes in C are 2^n bits. Based on the application, if a resolution of 0.004 degrees is sufficient in terms of accuracy, a 16 bit variable is suitable for storing the measurement data. Store only D2 and D1 as 1 sign + 15 unsigned integer bits where the scaling changes to 0.004 degrees per count. To include the 2 least significant bits and maintain a resolution of 0.001 degrees, 32-bit integer must be used. Using this data size we can show actual outputs of units under the previously described scaling and formatting. The D0, D1, D2 for a unit with 60 degree "full-scale reading" correspond to +/- 60 deg. +60 degrees corresponds to 0x3A980000, and -60 degrees corresponds to -0x3A980000 / [0x3A980000+Ox80000000] / 0XBA980000 with the 18 bit data structure left justified within the total variable length. Below are two examples of the same decoding method used above

In the following manual procedure and example, we will be using the 18 bit data within a 24 bit variable length scenario to maximize the resolution. Below are the data bytes in the order they would have been received, we assume the addressing and checksum information was verified previous to this decoding process:

D0: 0b0000 0010 / $02 D1: 0b1001 1000 / $98 D2: 0b0011 1010 / $3A

The bytes need to re-ordered so that D0 is the least significant byte and D2 the most significant byte of the n-wly formed data structure. After the new structure is populated in the correct order it has the value below:

D2 [xxxxxxxx] => D1[xxxxxxxx] => D0[xxxxxxxx]

To

0b [0011 1010][1001 1000][0000 0010] or $3A9802

The whole D0 byte in included in our concatenation, the byte contains non data bits in the status portion. To remove the status bits truncate the 24 bit format down the most significant 18 bits which contain the measurement data only. Below in red are the bits to be removed from the format, then every value regrouped to form bytes with no leading zeroes.

0x3A9800 = [0011 1010] [10011000] [00000000] => [00] [1110 1010] [0110 0000] = 0x0EA60

The remaining is the 18 bits of data from the unit with the most significant bit of the data being a sign bit to indicate whether the data is negative or positive. This sign bit is shown in red below.

[00][1110 1010][0110 0000] = 0x0EA60

If the value of the sign bit is zero, the value is positive. If the bit is 1, the value is negative. Once we know if the value is positive or negative, we can remove the sign bit from the remaining data.

[0] [1110 1010] [0110 0000] => [0] [1110 1010] [0110 0000]

Now that the data has been formatted, we can translate the binary data into an integer count using standard binary positional values, where each count is 1/1000 of a degree. So the 17 bits data above becomes an integer value of:

[0][1110 1010][0110 0000] => 60,000

The result is 60,000 counts. Now recall each count is $\frac{1}{1000}$ of a degree and that we determined the value was positive. The resulting angle measurement is:

$$60{,}000(counts) * \frac{1}{1000}\left(\frac{deg}{count}\right) = 60.000\ degrees * 1(positive\ polarity) = Degree\ Reading$$

Degree reading = +60.000 Degrees

In the C based example below, Reading is defined as a 32 bit signed integer. The value of reading is the left justified concatenation of D2, D1, and D0 respectively.

```
Boolean Set_Negative=FALSE; //Define variable to hold polarity state

Reading &= 0xffffc000;                  // Mask off the status bits

If (Reading & 0x80000000)   // Determine if sign is negative

        { Set_Negative = TRUE; Reading -= 0x80000000; }

// set polarity state variable and remove extra bit from the remaining data

Reading = Reading >> 13;  // Shift for scaling in so LSB data fits into LSB variable lentgh

If (Set_Negative)                  //determine which constant to scale by

        {Reading *= -1000;}  // Reading should represent as -xx.xxx degrees.

else    {Reading *= +1000;} // Reading should represent as +xx.xxx degrees.
```

ii. DXA Decoding

The value of the measurement is in signed fractional g's, with 17 fraction bits and one incremental integer bit. These results in the MSB representing a unit place and each sequential bit representing half of the previous bit's value. For instance b16 (second MSB) = 0.5, b15=0.25, etc… Below is the equation which shows how the value of any bit in the 18 bit measurement data can be determined.

$$\frac{1}{2^{17-n}}$$

As with the DXI units the available integer sizes in most systems are (2^n) bits. Most of the same data size and resolution tradeoffs exist. If 16 bits are enough, you can store just D2 and D1 as a **Q15** (i.e., 15 fraction bits) value. To include the 2 LSBs a 32-bit variable must be used. Using a left-justified sample within 32-bit variable, mask off the Flag bits from D0 and the scale will be Q31. Under the assumption that the data has been stored in a 32 bit variable, the Q31 format results in 0x00000000 being equivalent to +/- 0g. 0x7fffc000 and 0x80000000 represent +/- 1g, minus 1 resolution unit at the positive end. As shown above the Reading can saturate at these values (but may vary with range), but the instrument is specified only over its calibrated range - these end values may not even be reachable for lower ranged units. Below is an example calculation of G's from the DXAs proprietary output format.

$$0x[1][6000] = \frac{1}{2^{17-16}} + \frac{1}{2^{17-14}} + \frac{1}{2^{17-13}} = 0.6875g$$

For manual decoding of the DXA 18 bit data please use the following procedure. In the following manual procedure and example for DXA decoding, we will be using the 18 bit data within a 24 bit variable length. Below are the data bytes in the order they would have been received, we assume the addressing and checksum information was verified previous to this decoding process:

D0: 0b1100 0000 / $C0 D1: 0b1101 1010 / $DA D2: 0b0110 1110 / $6E

The bytes need to re-ordered so that D0 is the least significant byte and D2 the most significant byte of the newly formed data structure. After the new structure is populated in the correct order it has the value below:
u

D2 [xxxxxxxx] =>D1[xxxxxxxx] => D0[xxxxxxxx]

To

0b [0110 1110][1101 1010][1100 0000] or $6EDAC0

The whole D0 byte in included in our concatenation, the byte contains non data bits in the status portion. To nullify the status bits set last 6 bits of the original 24 to 0. We are maintaining the Left justified 24 bit data variable length for summation and 2's

complement. Below in red are the bits to be removed from the format, then every value regrouped to form bytes with no leading zeroes.

0x6EDAC0 = [0110 1110][1101 1010][1100 0000] => 0x6EDAC0 = [0110 1110][1101 1010][1100 0000]

The unaltered 18 bits of data in the 24 bit format use a 2's complement fractional representation. Full scale output is equivalent to ~ 1G (0.99999…). The use of 2's complement allows the MSB to act as a sign indicator with reference to absolute zero (0x0000 0000 left justified).

[0110 1110][1101 1010][1100 0000] = 0x6EDAC0

In the above example the value is positive.

If the value of the sign bit is zero, the value is positive. If the bit is 1, the value is negative. Once known if the value is positive or negative, the value can be manipulated as necessary. In the event of a negative value, a 2's complement routine is used. The value is subjected to a binary inversion and single count addition with carry to create the positive representation. Below is the 2's complement of the value used in this example.

	[1001 0001][0010 0101][0100 0000] =0x912540
Binary Inversion:	[0110 1110][1101 1010][1011 1111] =0x6ED93F
+1	[0000 0000][0000 0000][0000 0001] = 0x000001

	[0110 1110][1101 1010][1100 0000] =0x6EDAC0

(same as the example data value)

Now that the data has been formatted, we can remove the extra formatting and status bits. Afterward translating the binary data into the fractional equivalencies based on binary location. Using standard position values, each location is valued according to the following equation, where only high bit values are used in the summation for final value:

$$\frac{1}{2^{17-n}} \; where\ n\ is\ the\ bit\ position\ within\ the\ 18\ bit\ data$$

To show this using our example 0x6EDAC0 we break it into each bit, positions are defined LSB as 0:

Binary Position	Hex Value	Binary Value	Decimal Equivlent		Summation
17	6	0	Sign		
16		1	0.50000000000000000	=	0.5
15		1	0.25000000000000000	+=	0.75
14		0	0.12500000000000000	+=	0.75
13	E	1	0.06250000000000000	+=	0.8125
12		1	0.03125000000000000	+=	0.84375
11		1	0.01562500000000000	+=	0.859375
10		0	0.00781250000000000	+=	0.859375
9	D	1	0.00390625000000000	+=	0.86328125
8		1	0.00195312500000000	+=	0.865234375000
7		0	0.00097656250000000	+=	0.865234375000
6		1	0.00048828125000000	+=	0.865722656250
5	A	1	0.00024414062500000	+=	0.865966796875
4		0	0.00012207031250000	+=	0.865966796875
3		1	0.00006103515625000	+=	0.866027832031
2		0	0.00003051757812500	+=	0.866027832031
1	C	1	0.00001525878906250	+=	0.866043090820
0		1	0.00000762939453125	+=	0.866050720215
N/A		X			
N/A		X	N/A		
N/A	0	X	N/A		

N/A		0	N/A		
N/A		0	N/A		
N/A		0	N/A		

So the 18 bits data above becomes a decimal value after summation of: 0.872718811 G

Below is a C decoding routine for the DXA model provided to more quickly integrate the DXA into its target application. For use in custom digital systems data can be read as a 32 bit signed integer, type int or long depending on your architecture. Store the bytes D0, D1, D2 into the 32 bit int *Reading* with D2 occupying the MSB of the data element and D0 occupying the least significant byte.

Reading &= 0xffffc000; *// Mask off the Flag bits*

// Acceleration is type float or double

Acceleration = (float) Reading/0x80000000ul; *//ul is the variable type unsigned long. Result in g's.*

6. Additional Information

This section will contain general operating information which can be useful for troubleshooting, rapid implementation, or other characteristics which do not fall under the previous sections.

A. Errors and Flash Memory Verification

In mission-critical applications or in the event in remote installation, the device is designed to report a problem but continue to operate if at all possible. In a redundant system, this tells the controller which device to start ignoring when they don't match. The probability of a single-bit Flash memory error affecting normal operation is low because normal operation involves a small portion of the complete program. However, the checksum test doesn't report the number of bits affected. Any memory error is abnormal and serves as a warning that memory content has been altered.

The program verifies the contents of the Program, Calibration Data and Filter Coefficient areas of Flash on startup. It also checks the unused area for bits that no longer read back as erased, since this is potentially an early warning. If Program Checksum passes, the Filter Coefficient area is checked against reference data which was included in the Program Checksum. If a mismatch is found, the coefficients are rewritten and no error would be seen at the next startup. The Calibration Checksum is

verified next; its error code would overwrite the error code for Filter Coefficients. The error is reported in the Status byte (D0) of the Data Packet.

All of these errors set bit D0.4 of the status byte. D0.1/2 are used to identify Memory Verification errors, in this order:

(D0 is the Status byte in the Data Packet. Bits 6 & 7 are the lsb's of the data and b0 is the Saturation flag.)

D0 bits:
7 6 5 4 3 2 1 0
x x 0 1 x 0 1 x Program Checksum Error (highest priority)
x x 0 1 x 1 1 x Calibration Checksum Error
x x 0 1 x 1 0 x Filter Coefficient Mismatch
x x 0 1 x 0 0 x Unused Flash not blank (lowest priority)
x x x 0 x x x x No Checksum Error nor Filter Coefficient Mismatch;
D0.(2,1) then indicate non-default data modes.

This is the priority for reporting. Filter Coefficient Mismatch will not be reported if either of the higher ones exists, but the Filter Coefficients will still be refreshed if mismatch unless Program Checksum Error is set.

B. Troubleshooting

Effectively troubleshooting the DX series is done in a few sequential steps. Ensure the device is connected as described in the outline drawing or this manual.

Problem: The unit is transmitting, but data is corrupted.

Source:

1. Unit data line polarity is reversed
2. Unit has exceeded its operational temperature and unit may be in risk for memory corruption.
3. Line length is reducing transmission voltage too drastically. See section 4.E for details.

Problem: Unit fails to send Data Automatically

Source:

1. Unit is not powered.
2. Host is incorrectly addressed to COM / serial port.
3. Unit is in RS485 Operational Mode. Use PING to determine mode.

Problem: Unit fails to respond to POLL or other commands.

Source:

1. Unit is being addressed incorrectly. Use PING to determine address.
2. Unit command checksum is invalid, causing rejection of whole command string.

Problem: Unit output does not reach range it currently positioned at.

1. Unit's Analog limitations have caused it to reach saturation; each model's over range capability varies between 10% and 100% by design. If you have over reached saturation, please consider a higher ranged model for replacement.

C. Guidelines and Effective Output Rate Selection

The averaging function/filter built into each DX series unit only a 1-pole response. Therefore the parameters in Set RS422 Output Period command and Set Max Averaging Time Constant or related command must result in the output sample rate being much higher than twice the averaging filter's cutoff frequency to avoid aliasing. Aliasing will occur momentarily anyway while the averaging pool builds up to the specified count. Given the above consideration the following statements are true: M=<Output Period Parameter>+1 and N=<Max Averaging Time Constant parameter>+1, the Nyquist frequency is 90/ (2*M) Hz and attenuation at the Nyquist frequency is

$-dB = 10*\log(1 + (\pi/(M*\mathrm{Ln}(N/(N-1))))^2)$.

The above does not apply when the Output Period is 1/90 (for DXA models) sec (M = 1), because then the Nyquist frequency is in another (sharper cutoff) filter's stop band.

Some considerations for selection of these parameters: Per the above, with M = N-1 and 3 <= N <= 256, attenuation at the Nyquist frequency ranges from 12.0 to 10.4 dB. With N = 256 and M = 2, it's 52.1 dB. Any remaining signal component above the Nyquist frequency will in effect heterodyne with the Output Sample Rate. A coherent vibration signal at exactly the Output Sample Rate (twice the Nyquist frequency) would yield a zero-beat, or DC error value. The Averaging Filter's attenuation at that frequency is 6 dB greater than at the Nyquist frequency.

D. Mounting Considerations

In multiple-unit installations, the cable should be routed along a path from node to node, including the Host, which can be anywhere along the run. Line termination resistors should be installed only at each end of the line. Long stubs between the line and individual nodes should be avoided. The same ground-loop considerations apply as above. If after connecting the shield at the end farthest from the Host there would be continuity or a voltage difference between any device case and

the cable shield, then the shield connection should not be made at this device. The choice of which nodes should be connected to the shield should also consider the noise environment. For example, if one device is close to a transmitting antenna it is a prime candidate for the choice of where to ground the shield because noise voltage between the cable shield and the device case may show up as scatter in the measurements before it shows up as communication errors. A Ferrite sleeve on the cable will attenuate this noise, and having the cable shield grounded on the device side of the sleeve makes the sleeve more effective because the sleeve introduces a high series impedance for the common-mode signals and the case connection is a low-impedance load. The noise current will flow through the shield rather than the inner conductors.

E. Line Resistance Effects

The major issue with long lines in differential transmission is DC resistance. 4000' of AWG#24 pair has a loop resistance of about 204Ω at room temperature. If Z_0 is 100Ω, then 100Ω line termination resistors would be connected at each end and signal voltage at the receiving end reduced to 1/3 (actually lower than that due to R_s of the line driver). If a bias network is present, sufficient signal voltage is required at each node to overcome the bias. Device operating current, typically 35-40 mA when not transmitting, must increase to supply the signal current into the effective load resistance seen by its line driver. Since only one device can be transmitting at any one time, worst-case voltage drop should be calculated on the basis of all devices drawing the specified maximum of 50 mA, and the device farthest from the power supply drawing additionally 3.3v/(30+R_L), where R_L is the resistance you would measure across the serial line pair where that device is connected (30Ω is the approximate output resistance of the line driver). With the supply voltage set to the maximum of 30v, the supply voltage at that farthest device must not be less than 9v.

An additional effect of voltage drop along the line is that the Power Gnd voltage is different at each node. This changes the common-mode voltage of each talker as seen by a listener on the bus, which must be considered if a bias network is used. Ensure that the effective resistances to ground or fixed voltage on both sides of the line are equal so that V_{cm} changes do not generate differential voltage changes.

With RS485 protocol, since the driver is off when not sending, increasing baud rate reduces average current consumption. The maximum baud rate, 230400 Baud, is not recommended for cable lengths exceeding 2000' if the Host and Power Supply are at the end of the cable if a bias network is used. The DX 200's line driver is slew-rate limited for RFI reduction, and the voltage drop due to bias current can shift the zero crossings enough to cause transmission errors.

Appendix A: Condensed Command Set

1. Measurement Outputs

Changing Polarity

Both polarity commands take effect immediately.

Normal Polarity:

	Hexadecimal String	Realterm Copy and Paste
Generic	AC, "UAID", C9, "Checksum"	N/A
Default UAID X Axis	AC, 71, C9, 18	$AC $71 $C9 $18
Default UAID Y Axis	AC, 72, C9, 17	$AC $72 $C9 $17
Default UAID X+Y Axis	AC, 73, C9, 16	$AC $73 $C9 $16

Reverse Polarity:

	Hexadecimal String	Realterm Copy and Paste
Generic	AC, "UAID", C8, "Checksum"	N/A
Default UAID X Axis	AC, 71, C8, 19	$AC $71 $C8 $19
Default UAID Y Axis	AC, 72, C8, 18	$AC $72 $C8 $18
Default UAID X+Y Axis	AC, 73, C8, 17	$AC $73 $C8 $17

Averaging Functionality

Averaging Commands take effect immediately. When Averaging, the Aux byte contains the number of updates since last Data Packet, = # of samples held in the average if > 0. If 0 or 1, value is the last sample.

Standard Averaging ON:

This command can also be used to reset average when continuous averaging is enabled. Sets Status bit 2.

	Hexadecimal String	Realterm Copy and Paste
Generic	AC, "UAID", C5, "Checksum"	N/A
Default UAID X Axis	AC, 71, C5, 1C	$AC $71 $C5 $1C
Default UAID Y Axis	AC, 72, C5, 1B	$AC $72 $C5 $1B
Default UAID X+Y Axis	AC, 73, C5, 1A	$AC $73 $C5 $1A

Standard Averaging OFF:

Turns off Averaging and resets average and cancels selection of Continuous mode.

	Hexadecimal String	Realterm Copy and Paste
Generic	AC, "UAID", C4, "Checksum"	N/A
Default UAID X Axis	AC, 71, C4, 1D	$AC $71 $C4 $1D
Default UAID Y Axis	AC, 72, C4, 1C	$AC $72 $C4 $1E
Default UAID X+Y Axis	AC, 73, C4, 1B	$AC $73 $C4 $1F

Continuous Averaging ON:

This command does not reset the count and start a new average. Can be used in RS422 mode, with the selectable Output Period setting. Sets Status bit 2.

	Hexadecimal String	Realterm Copy and Paste
Generic	AC, "UAID", C7, "Checksum"	N/A
Default UAID X Axis	AC, 71, C7, 1A	$AC $71 $C7 $1A
Default UAID Y Axis	AC, 72, C7, 19	$AC $72 $C7 $19
Default UAID X+Y Axis	AC, 73, C7, 18	$AC $73 $C7 $18

Continuous Averaging OFF:

Only turns off Continuous mode.

	Hexadecimal String	Realterm Copy and Paste
Generic	AC, "UAID", C6, "Checksum"	N/A
Default UAID X Axis	AC, 71, C6, 1B	$AC $71 $C6 $1B
Default UAID Y Axis	AC, 72, C6, 1A	$AC $72 $C6 $1A
Default UAID X+Y Axis	AC, 73, C6, 19	$AC $73 $C6 $19

Set Max Averaging Time Constant:

This command does not reset the average and current selection of Averaging Mode remains in effect. <Acount> (1 byte) is max number of $AF, $02, $E4, <Acount>, Checksum samples held in the average.

	Hexadecimal String	Realterm Copy and Paste
Generic	AF, "UAID", E4, "Acount", "Checksum"	N/A
Default X+Y Axis -255	AF, 73, E4, FF, F7	$AF $71 $E4 $1B
Default X+Y Axis -128	AF, 73, E4, 80, 77	$AF $72 $E4 $1A
Default X+Y Axis -000	AF, 73, E4, 00, F7	$AF $73 $E4 $19

Set Max Averaging Time Constant and Enable Averaging:

Functionally identical to executing Set Max Averaging Time Constant and Averaging

ON individually. The Acount parameter only has an effect for the Set Max Averaging Time Constant portion of the jointed command. Command does not reset average.

	Hexadecimal String	Realterm Copy and Paste
Generic	AF, "UAID", E5, "Acount", "Checksum"	N/A
Default X+Y Axis -255	AF, 73, E5, FF, F6	$AF $73 $E5 $FF $F7
Default X+Y Axis -128	AF, 73, E5, 80, 76	$AF $73 $E5 $80 $77
Default X+Y Axis -000	AF, 73, E5, 00, F6	$AF $73 $E5 $00 $F7

Set Max Averaging Time Constant and Enable Continuous Averaging:

	Hexadecimal String	Realterm Copy and Paste
Generic	AF, "UAID", E5, "Acount", "Checksum"	N/A
Default X+Y Axis -255	AF, 73, E7, FF, F4	$AF $73 $E7 $FF $F4
Default X+Y Axis -128	AF, 73, E7, 80, 74	$AF $73 $E7 $80 $74
Default X+Y Axis -000	AF, 73, E7, 00, F4	$AF $73 $E7 $00 $F4

2. Transmission Parameters

Set Unit ID (also known as Assign Unit ID):

This commands assignable portion of the command string ranges from 0x01 to 0x27. Command only takes effect when saved to flash.

	Hexadecimal String	Realterm Copy and Paste
Generic	AC, "UAID", "XXXX XX11", "Checksum"	N/A
Default X+Y Axis -025	AC, 73, 97, 48	$AC $73 $97 $48
Default X+Y Axis -010	AC, 73, 2B, B4	$AC $73 $2B $B4
Default X+Y Axis -001	AC, 73, 07, D8	$AC $73 $07 $D8

RS-422 Emulation Control

All RS 422 Emulation control commands must be saved to Flash, Will take effect at reset.

RS-422 Emulation ON:

	Hexadecimal String	Realterm Copy and Paste
Generic	AC, "UAID", C3, "Checksum"	N/A
Default X+Y Axis -025	AC, 73, C3, 1C	$AC $73 $C3 $1C
Default X Axis -010	AC, 71, C3, 1E	$AC $71 $C3 $1E
Default Y Axis -001	AC, 72, C3, 1D	$AC $72 $C3 $1D

RS-422 Emulation OFF:

	Hexadecimal String	Realterm Copy and Paste
Generic	AC, "UAID", C2, "Checksum"	N/A
Default X+Y Axis -025	AC, 73, C2, 1D	$AC $73 $C2 $1D
Default X Axis -010	AC, 71, C2, 1F	$AC $71 $C2 $1F
Default Y Axis -001	AC, 72, C2, 1E	$AC $72 $C2 $1E

Select Baud Rate:

Baud Rate selection must be performed on both axes on a dual axis unit otherwise Y axis will fail to operate upon reset. In order to select Baud rate, the binary selection is completed is done within the function identification byte. This command will only take effect at reset after it has been saved to flash.

	Hexadecimal String	Realterm Copy and Paste
Generic	AC, "UAID", 1011 0xxx, "Checksum"	N/A
Default X+Y Axis -19.2 KHz	AC, 73, B0, 2F	$AC $73 $B0 $2F
Default X+Y Axis -38.4 KHz	AC, 73, B1, 2E	$AC $73 $B1 $2E
Default X+Y Axis -57.6 KHz	AC, 73, B2, 2D	$AC $73 $B2 $2D
Default X+Y Axis -115.2 KHz	AC, 73, B3, 2C	$AC $73 $B3 $2C
Default X+Y Axis -230.4 KHz	AC, 73, B4, 2B	$AC $73 $B4 $2B

Select Minimum Response Delay:

Takes effect immediately, including the ACK for this command.

	Hexadecimal String	Realterm Copy and Paste
Generic	AF, "UAID", CD, "Delay", "Checksum"	N/A
Default X+Y Axis – 0.0 mS	AF, 73, CD, FF, 0F	$AF $73 $CD $FF $0F
Default X+Y Axis – 3.9 mS	AF, 73, CD, 80, 8E	$AF $71 $CD $80 $8E
Default X+Y Axis – 7.8 mS	AF, 73, CD, 00, 0F	$AF $72 $CD $00 $0F

Set RS 422 Output Period:

This section will be covered in 2 portions to reflect the differences in DXA/DXI maximum output rates. Command takes effect upon reset, save to flash and operation in RS 422 mode only.

DXA Output rate Table:

	Hexadecimal String	Realterm Copy and Paste
Generic	AF, "UAID", E2, "Period", "Checksum"	N/A
Default X+Y Axis – 90 Hz	AF, 73, E2, 00, F9	$AF $73 $E2 $00 $F9

Default X+Y Axis – 45 Hz	AF, 73, E2, 01, F8	$AF $73 $E2 $01 $F8
Default X+Y Axis – 15 Hz	AF, 73, E2, 05, F4	$AF $73 $E2 $05 $F4
Default X+Y Axis – 05 Hz	AF, 73, E2, 12, E7	$AF $73 $E2 $12 $E7

Note: Commands issued between DXA and DXI Models are identical. The difference in the default output rate creates 2 command tables based on the respective equation in each case.

	Hexadecimal String	**Realterm Copy and Paste**
Generic	AF, "UAID", E2, "Period", "Checksum"	N/A
Default X+Y Axis – 60 Hz	AF, 73, E2, 00, F9	$AF $73 $E2 $00 $F9
Default X+Y Axis – 30 Hz	AF, 73, E2, 01, F8	$AF $73 $E2 $01 $F8
Default X+Y Axis – 10 Hz	AF, 73, E2, 05, F4	$AF $73 $E2 $05 $F4
Default X+Y Axis – 02 Hz	AF, 73, E2, 1D, DC	$AF $73 $E2 $12 $E7

3. Maintenance & Infrastructure

Polling:

Polling is a unique case, where the prefix A9 with a 3 byte length only has one corresponding command.

	Hexadecimal String	Realterm Copy and Paste
Generic	A9, "UAID", "Checksum"	N/A
Default X+Y Axis	A9, 73, E2	$A9 $73 $E2
Default X Axis	A9, 71, E4	$A9 $71 $E4
Default Y Axis	A9, 72, E3	$A9 $72 $E3

Break:

	Hexadecimal String	Realterm Copy and Paste
Generic	A9, "UAID", "Checksum"	N/A
Broadcast X+Y Axis	AC, 03, 02, 4E	$AC $03 $02 $4E
Broadcast X+Y Axis with Rubout's	AC, 03, 02, 4E, FF, FF	$AC $03 $02 $4E $FF $FF

Reset:

	Hexadecimal String	Realterm Copy and Paste
Generic	AC, "UAID", 03, "Checksum"	N/A
Default X+Y Axis	AC, 73, 03, DC	$AC $73 $03 $DC
Broadcast X+Y Axis	AC, 03, 03, 4D	$AC $03 $03 $4D

Allow Update:

This command must be used preceding a protected command execution. Any interference resulting in the failure of a single axis will cause both axes on a dual axis unit to reject the command.

	Hexadecimal String	Realterm Copy and Paste
Generic	AC, "UAID", "Checksum"	N/A
Default X+Y Axis	AC, 73, 01, DE	$AC $73 $01 $DE

Update Configuration:

	Hexadecimal String	Realterm Copy and Paste
Generic	AC, "UAID", 00, "Checksum"	N/A
Default X+Y Axis	AC, 73, 00, DF	$AC $73 $00 $DF

ENQ/Ping:

This example is set to broadcast and only 1 unit should be connected to the bus.

	Hexadecimal String	Realterm Copy and Paste
Generic	AC, "UAID", B7, "Checksum"	N/A
Broadcast X+Y Axis	AC, 03, B7, 91	$AC $03 $B7 $91

Note: The return string is ASCII, and parameter reflect startup conditions

Query Setting:

	Hexadecimal String	Realterm Copy and Paste
Generic	AC, "UAID", B 1xxx, "Checksum"	N/A
Default X+Y Axis : Configuration Byte	AC, 73, B0, 2F	$AC, $73, $B0, $2F
Default X+Y Axis : Response Delay	AC, 73, B1, 2E	$AC, $73, $B1, $2E
Default X+Y Axis : Output Period	AC, 73, B2, 2D	$AC, $73, $B2, $2D
Default X+Y Axis : Max Avg. Time Const.	AC, 73, B3, 2C	$AC, $73, $B3, $2C

Note: Returned values are from the editing segment of the UUT.

Send Configuration Vector:

	Hexadecimal String	Realterm Copy and Paste

Generic	AC, "UAID", BF, "Checksum"	N/A
Default X+Y Axis :	AC, 73, BF, 20	$AC, $73, $B0, $2F

Note: Returned values are from the editing segment of the UUT.

Additional Notes:

Long commands return an Ack/Nak message of the form:

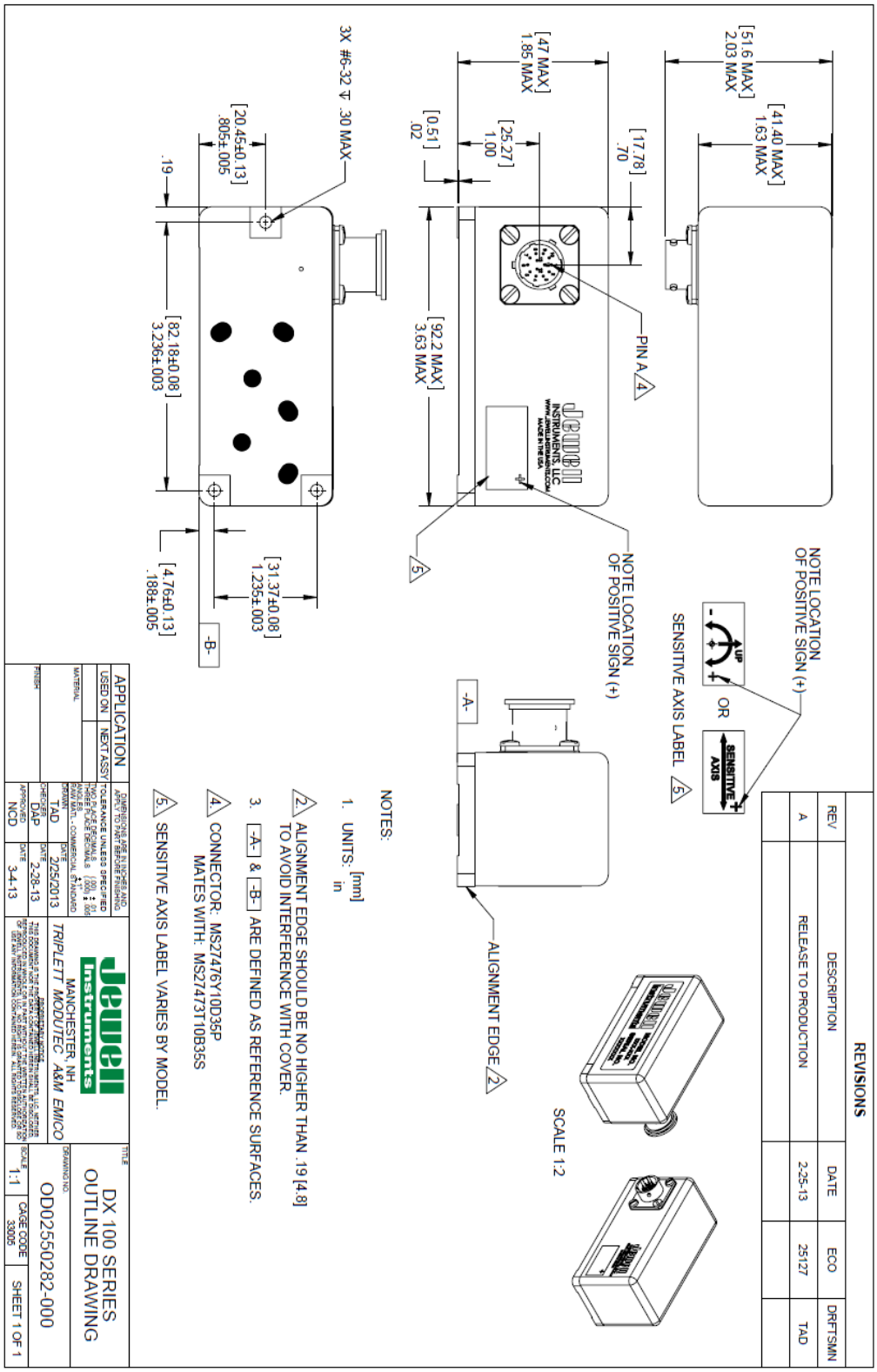
$A3, UAID, Argument, Checksum Acknowledge

Negative Acknowledge uses the binary inverse of the argument to indicate a failure.

$A3, UAID, ~Argument, Checksum Negative Acknowledge
(~Argument = 1's compliment)

The Acknowledge message is generated after command executes. Thus if Unit ID is changed, the Acknowledge from Update Configuration will contain the new UAID. The Acknowledge from Set Minimum Response Delay reflects the new delay setting.

Appendix B: Technical Documentation



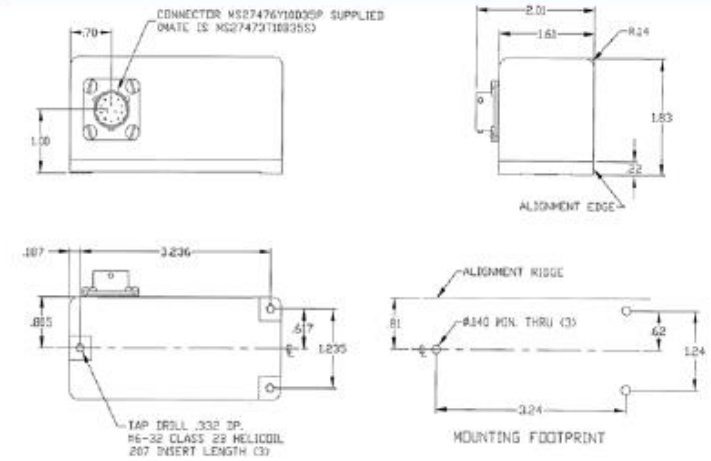
DXI-100/200 Series Inclinometer

Making Sense out of Motion...

Digital Output - Single or Dual Axis for a wide variety of applications.

The Jewell **DXI-100/200 Series** single or dual digital inclinometer takes Jewell's highly accurate analog closed loop sensor technology to the next level.

Outline Diagram: *DXI-100/200 Series Digital Inclinometer*

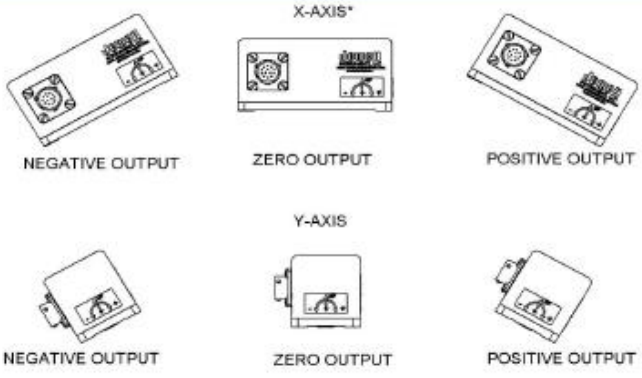


Features & Benefits

- Digital output
- Resolution 0.001°
- Mechanical Shock 1500g 1msec 1/2 sine
- Industry Standard EIA RS485 and EIA RS422 output
- For use in high shock and vibration environments
- High Precision and Performance
- Low Noise

Applications

- Radar/Antenna Control
- Structural Monitoring
- Linear Acceleration/Deceleration Measuring
- Automatic Train Position Control
- Seismic Monitoring
- Platform Leveling



*FOR DXI-100 SERIES SENSITIVE AXIS THIS DIRECTION ONLY

Jewell Instruments LLC, 850 Perimeter Road, Manchester, NH 03103
sales@jewellinstruments.com • www.jewellinstruments.com • Tel (800) 227-5955

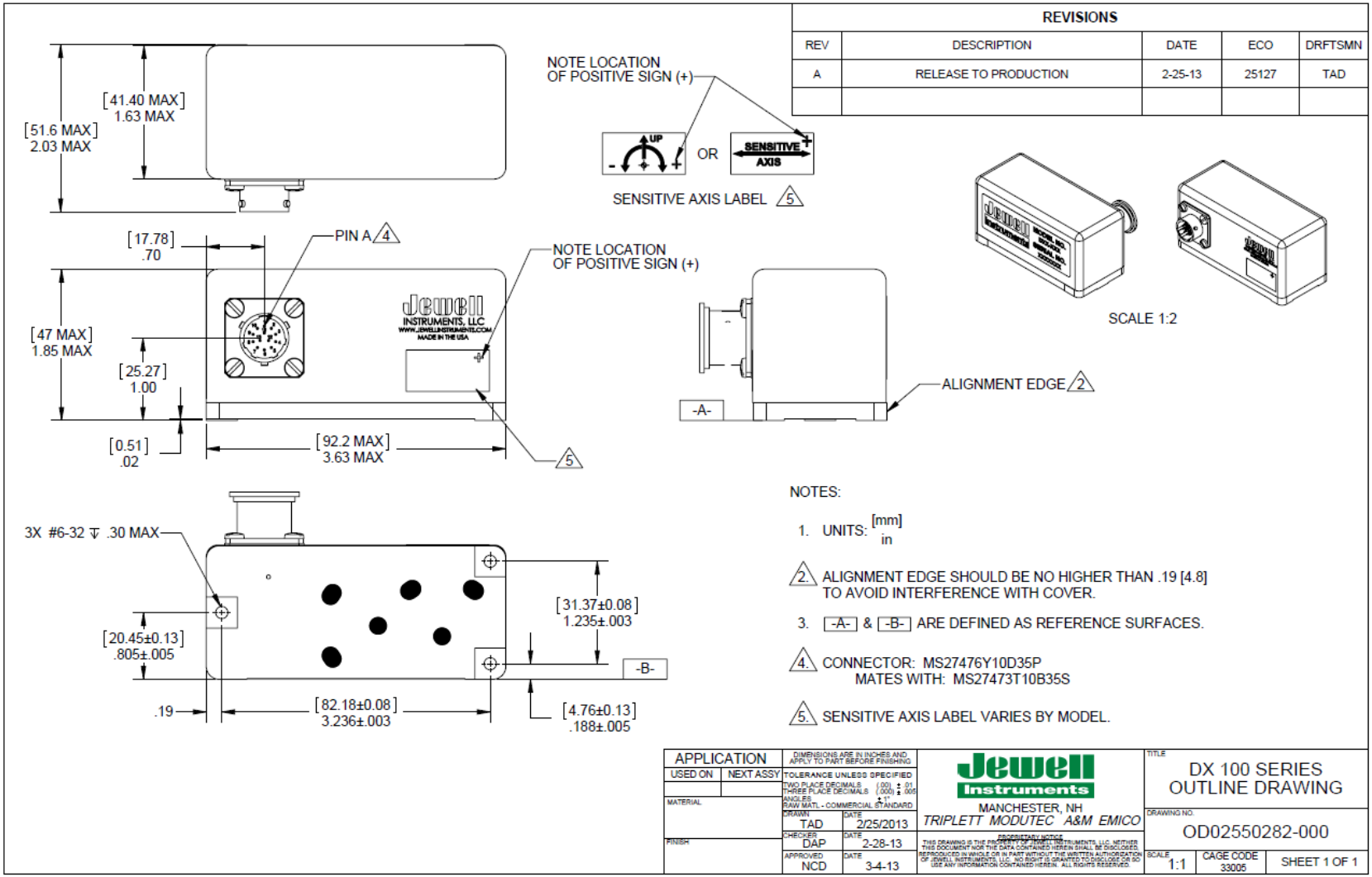
DXI-100/200 Series Inclinometer

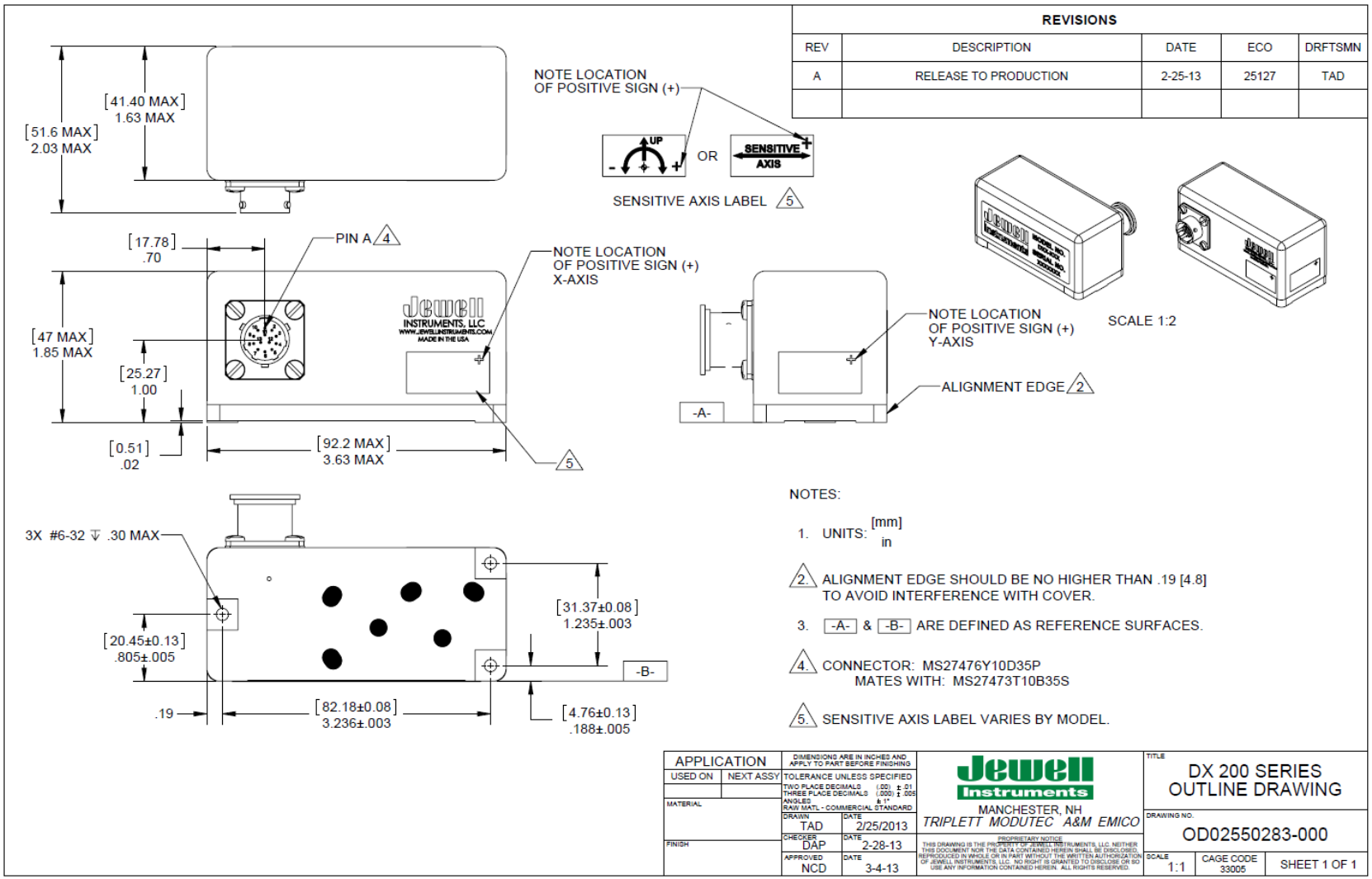
Making Sense out of Motion...

Performance					
Input Range[1], °	±1.0	±3.0	±14.5	±30.0	±60.0
Number of Axis	1,2	1,2	1,2	1,2	1,2
Non Linearity[2], %FRO, Max	0.02	0.015	0.02	0.02	0.03
Scale Factor Tolerance, % Max	0.05	0.05	0.05	0.05	0.05
Scale Factor Temperature Sensitivity, % reading/°C, Max	0.01	0.01	0.01	0.01	0.01
Output at 0° Tilt, °Max	0.01	0.01	0.05	0.05	0.05
0° Output Temperature Sensitivity, °/°C, Max	0.001	0.001	0.005	0.005	0.005
Bandwidth (-3dB), Hz, Nom[3]	3	6	30	30	30
Transverse Axis Misalignment, °, Max	0.15	0.15	0.5	0.5	0.5
Hysteresis, °, Max	0.001	0.001	0.001	0.001	0.001
Resolution and Threshold, °, Max	0.001	0.001	0.001	0.001	0.001
Power On Repeatability, °Max	0.001	0.001	0.001	0.001	0.001
Repeatability, °Max	0.001	0.001	0.002	0.002	0.003
Digital Output					
Interface	EIA-RS485 (default)/EIA-RS422				
Protocol	Custom				
Output Representation	Degrees				
Baud Rate[4]	19200, 38400, 57600, 115200, 230400				
Electrical					
Supply Voltage, Volts DC	10 to 30				
Input Current, mA, Max	DXI-100, 80 mA & DXI-200, 100 mA				
Environmental					
Operational Temp Range, °C	-40 to +70				
Storage and Temp Range, °C	-40 to +70				
Protection Class per IEC 529	IP67				
NEMA Enclosure Rating	6				
Seal	MILD-STD-202 Method 112				
Shock Survival	1500g, 1msec, ½ sine				
Vibration Survival, grms (20Hz to 2 KHz)	20				
Enclosure					
Housing Material	Anodized and Alodine Aluminum				
Weight	DXI-100 8oz [226.80 g]/ DXI-200 10oz [283.50 g]				
Connector Type	MS27476Y10D35P				
Recommended Mating Connector	MS27473T10B35S				

NOTES:
1- Full range is defined as "from negative full input angle to positive full input angle"
2 - Non-linearity is specified as deviation of output referenced to a best fit straight line, independent of misalignment.
3 - Factory default
4- Default Baud Rate is 38400

Jewell Instruments LLC, 850 Perimeter Road, Manchester, NH 03103
sales@jewellinstruments.com • www.jewellinstruments.com • Tel (800) 227-5955


REVISIONS
REV | DESCRIPTION | DATE | ECO | DRFTSMN
A | RELEASE TO PRODUCTION | 2-25-13 | 25127 | TAD
[51.6 MAX] 2.03 MAX
[41.40 MAX] 1.63 MAX
NOTE LOCATION OF POSITIVE SIGN (+)
UP
OR
SENSITIVE AXIS
SENSITIVE AXIS LABEL 5
SCALE 1:2
[17.78] .70
PIN A 4
NOTE LOCATION OF POSITIVE SIGN (+)
[47 MAX] 1.85 MAX
[25.27] 1.00
[0.51] .02
[92.2 MAX] 3.63 MAX
5
-A-
ALIGNMENT EDGE 2
3X #6-32 ↧ .30 MAX
[31.37±0.08] 1.235±.003
[20.45±0.13] .805±.005
-B-
.19
[82.18±0.08] 3.236±.003
[4.76±0.13] .188±.005
NOTES:
1. UNITS: [mm] in
2. ALIGNMENT EDGE SHOULD BE NO HIGHER THAN .19 [4.8] TO AVOID INTERFERENCE WITH COVER.
3. -A- & -B- ARE DEFINED AS REFERENCE SURFACES.
4. CONNECTOR: MS27476Y10D35P MATES WITH: MS27473T10B35S
5. SENSITIVE AXIS LABEL VARIES BY MODEL.
APPLICATION
USED ON | NEXT ASSY
MATERIAL
FINISH
DIMENSIONS ARE IN INCHES AND APPLY TO PART BEFORE FINISHING
TOLERANCE UNLESS SPECIFIED
TWO PLACE DECIMALS (.00) ± .01
THREE PLACE DECIMALS (.000) ± .005
ANGLES ± 1°
RAW MATL - COMMERCIAL STANDARD
DRAWN TAD 2/25/2013
CHECKER DAP 2-28-13
APPROVED NCD 3-4-13
Jewell Instruments
MANCHESTER, NH
TRIPLETT MODUTEC A&M EMICO
PROPRIETARY NOTICE
THIS DRAWING IS THE PROPERTY OF JEWELL INSTRUMENTS, LLC. NEITHER THIS DOCUMENT NOR THE DATA CONTAINED HEREIN SHALL BE DISCLOSED, REPRODUCED IN WHOLE OR IN PART WITHOUT THE WRITTEN AUTHORIZATION OF JEWELL INSTRUMENTS, LLC. NO RIGHT IS GRANTED TO DISCLOSE OR SO USE ANY INFORMATION CONTAINED HEREIN. ALL RIGHTS RESERVED.
TITLE
DX 100 SERIES OUTLINE DRAWING
DRAWING NO.
OD02550282-000
SCALE 1:1
CAGE CODE 33005
SHEET 1 OF 1


REVISIONS
REV | DESCRIPTION | DATE | ECO | DRFTSMN
A | RELEASE TO PRODUCTION | 2-25-13 | 25127 | TAD
[41.40 MAX] 1.63 MAX
[51.6 MAX] 2.03 MAX
NOTE LOCATION OF POSITIVE SIGN (+)
UP
OR
SENSITIVE AXIS
SENSITIVE AXIS LABEL 5
SCALE 1:2
[17.78] .70
PIN A 4
NOTE LOCATION OF POSITIVE SIGN (+) X-AXIS
[47 MAX] 1.85 MAX
[25.27] 1.00
[0.51] .02
[92.2 MAX] 3.63 MAX
5
NOTE LOCATION OF POSITIVE SIGN (+) Y-AXIS
ALIGNMENT EDGE 2
-A-
3X #6-32 ↧ .30 MAX
[20.45±0.13] .805±.005
[31.37±0.08] 1.235±.003
-B-
.19
[82.18±0.08] 3.236±.003
[4.76±0.13] .188±.005
NOTES:
1. UNITS: [mm] in
2. ALIGNMENT EDGE SHOULD BE NO HIGHER THAN .19 [4.8] TO AVOID INTERFERENCE WITH COVER.
3. -A- & -B- ARE DEFINED AS REFERENCE SURFACES.
4. CONNECTOR: MS27476Y10D35P MATES WITH: MS27473T10B35S
5. SENSITIVE AXIS LABEL VARIES BY MODEL.
APPLICATION
USED ON | NEXT ASSY
MATERIAL
FINISH
DIMENSIONS ARE IN INCHES AND APPLY TO PART BEFORE FINISHING
TOLERANCE UNLESS SPECIFIED
TWO PLACE DECIMALS (.00) ± .01
THREE PLACE DECIMALS (.000) ± .005
ANGLES ± 1°
RAW MATL - COMMERCIAL STANDARD
DRAWN TAD DATE 2/25/2013
CHECKER DAP DATE 2-28-13
APPROVED NCD DATE 3-4-13
Jewell Instruments
MANCHESTER, NH
TRIPLETT MODUTEC A&M EMICO
PROPRIETARY NOTICE
THIS DRAWING IS THE PROPERTY OF JEWELL INSTRUMENTS, LLC. NEITHER THIS DOCUMENT NOR THE DATA CONTAINED HEREIN SHALL BE DISCLOSED, REPRODUCED IN WHOLE OR IN PART WITHOUT THE WRITTEN AUTHORIZATION OF JEWELL INSTRUMENTS, LLC. NO RIGHT IS GRANTED TO DISCLOSE OR BO USE ANY INFORMATION CONTAINED HEREIN. ALL RIGHTS RESERVED.
TITLE
DX 200 SERIES OUTLINE DRAWING
DRAWING NO.
OD02550283-000
SCALE 1:1
CAGE CODE 33005
SHEET 1 OF 1