



**MODEL DXA/DXI-100/200
DRAFT USER MANUAL**

JEWELL INSTRUMENTS, LLC

**850 Perimeter Road
Manchester, NH 03103**

**PHONE: (800) 227-5955
E-MAIL: sales@jewellinstruments.com**

DXI-200/100 User Manual
Serial interface and Command set

SUBJECT INDEX

OVERVIEW -----	1.0
COMMUNICATIONS FORMAT -----	2.0
DATA MESSAGES FROM THE DEVICE -----	2.1
MESSAGES TO THE DEVICE -----	2.2
Short command (Poll Unit)	
Long commands	
Extended commands	
MAINTENANCE COMMAND SET -----	3.0
Long Commands -----	3.1
Reset	
Break	
Allow Update	
Update Configuration	
Assign Unit ID	
Set Baudrate	
ENQ (Ping)	
Query Setting xxx	
Send Configuration Vector	
RS422 Emulation Off	
RS422 Emulation On	
Averaging Off	
Averaging On	
Continuous Averaging Off	
Continuous Averaging On	
Select Reverse Polarity	
Select Normal Polarity	
Recall Saved Averaging and Polarity Settings	
Extended Commands -----	3.2
Set Minimum Response Delay	
Set RS422 Output Period	
Set Max Averaging Time Constant	
Set Max Averaging Time Constant and Enable Averaging	
Set Max Averaging Time Constant and Enable Continuous Averaging	
General Guidelines on Averaging Time Constant and Output Sample Rate -----	4.0
Flash Memory Verification -----	5.0
Electrical Installation -----	6.0
Multiple Unit Installation -----	6.1
Long Lines -----	6.2

1.0 OVERVIEW:

The DXI-200/100 is supplied as a 1- or 2-axis unit. It communicates via a 2-wire differential line: half-duplex when using RS485 protocol (default), and transmitter only when operating at RS422. The format is straight binary for both commands and data, and the measurement data is 18-bit signed scaled integer. The value is in thousandths of a degree and where the LSB is equivalent to 0.001 degrees and the most MSB represents sign.

Operating Modes:

While the default configuration is RS485 the device can also be configured as RS422, meaning it sends data periodically from both axes without being polled, and the serial line is never floating. The X and Y data are sent in separate packets (referred to as a "Twin Pack"). The output data are digitally filtered and contain measurements taken over a time interval significantly longer than the output sample period. In RS422 mode, the X and Y data in each Twin Pack correspond to the same time interval. The default output sample rate is 90 Hz (regardless of baud rate) and the filter pass band is 0 - 30 Hz. The output sample rate can optionally be set to any submultiples of 90 Hz (Continuous Averaging must be enabled).

In RS485 mode, the unit transmits only when polled. Addressing allows up to 30 units on a bus, and a poll can address either or both axes. If both, a "Twin Packet" is sent as in RS422 mode except the bus is released after each reply. The X and Y responses generally do not correspond to precisely the same time interval - they're just the most recent digital filter output values available when each packet is assembled.

Data Averaging is a mode setting which reduces noise and aliasing, especially when the unit is polled at long intervals in RS485 mode. A uniformly weighted average is computed from one poll to the next, up to a specified maximum time interval. Beyond this time interval, the weighting becomes exponential and the process becomes equivalent to a 1-pole low-pass filter following the fixed 30 Hz filter. A variation on this is Continuous Averaging. When Continuous Averaging is selected, the average is not reset after each output - averaging is always exponential with the selected time constant. In both Averaging modes, the number of output samples currently in the average is reported in the data packet. When the averaging is exponential, it's the effective number.

The data packet includes a status byte which indicates the on/off status of Averaging and Sign Reversal. The Status byte also indicates whether the analog front-end may have been in saturation during some part of the time interval which contributed to an output sample. This status bit is valid only for the most recent filter output - it is not maintained through all the potentially affected output samples when Averaging is enabled.

Besides Data Mode commands, there are Communications Mode commands to change baud rate, device address, response delay (in case the Master does not release the bus quickly) and turn RS422 mode on and off. Some commands can be broadcast to all units on the bus, all X axes, or all Y axes. Commands to a single unit can address one or both axes. To avoid temporary conflicts, some commands do not take effect until the change is saved in Flash and the unit is reset, both of which commands can be broadcast so the changes take effect simultaneously in all units on the bus. Saving configuration to Flash is a protected command which must be immediately preceded by an "Allow Update" command.

There is also a command to query each command-settable configuration value except baud rate, which is included in a general configuration query.

Even a unit configured as an RS422 "talker" can be reconfigured by commands via the serial interface. There is a 28 ms window at startup before the unit begins sending data wherein a "Break" command addressed to both axes will put the unit in RS485 mode to accept commands. Depending on the baud rate and line length, this may require a termination resistor at the device end to work properly.

2.0 COMMUNICATIONS FORMAT:

Message Protocol is 8 Data Bits, 1 Stop Bit, No Parity, RS485 or RS422 differential line. Available baud rates are: 19200, 38400, 57600, 115200 and 230400 baud. 38400 is the default. 19200 is the lowest baud rate possible for sending both axes at 90 updates/second (the filter output rate), whether polled or not. 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.

There are 3 command packet types and 3 data/reply packet types, for different information length requirements. All packet types have the same general format:

| Prefix | UAID | arguments, data, etc. | Checksum |

The 1st byte, Prefix, indicates the packet type and serves as a sync byte.

The next byte, UAID, is the unit/axis address. 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 according to b0 and b1. Any command which does not require a reply can be broadcast to all units by setting UAID=3 (Unit#0, both axes). Any data, command or argument bytes follow, LSB first.

The last byte is Checksum, computed thus: Sum all the preceding bytes. Add to this the number of carries from the low byte into the high byte of the accumulator (this is the value in the high byte if you're using a 16 bit accumulator). Ignore any carry from this addition. Ones compliment the result.

Henceforth, '\$' signifies hexadecimal byte value, not a character in the packet.

The set of Prefix values {\$A0, \$A3, \$A6} is reserved for messages from the unit; The set {\$A9, \$AC, \$AF} is reserved for messages from the master to the unit.

In normal operation, only the \$A9 (poll) and \$A6 (data) packets are involved. The remainder 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 is the total length of the packet.

2.1 Data Messages from the Device:

(Other messages are described elsewhere with the commands which trigger them)

The filter output rate is 90 samples per second. This is the default output packet rate in RS422 mode.

\$A6, UAID, D0, D1, D2, Aux, Checksum Data Packet

If the unit is configured as RS422, the UAID byte serves mainly to identify the axis. The initial device address field is \$70. Thus, if unit has never been assigned a new address,

UAID (X) = \$71, and

UAID (Y) = \$72

Valid address range for assignment is \$04 through \$9C, although a future version may restrict the upper limit to \$7C. The two lsb's are reserved for Axis ID.

D0, D1 & D2 are the Data bytes, sent LSB first.

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. b0 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.

In normal operation, Status bits 4 & 5 are 0 and bits 1 & 2 indicate non-default setup modes. 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 last section).

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.

Data Format (D0,D1,D2):

C programming language conventions are used in this description.

Byte order is "little endian", so D2 is MSB. The data is 18-bit signed integer, left-justified in the 3 data bytes. The 2 MSb's of D0 are the LSb's of the data and the rest of D0 are Flag bits. Value is in milli-degrees, with 17 incremental bits. Available integer sizes in C are generally $(8 < n)$ bits. If 16 bits are enough, you can store just D2 and D1 as a Q15 (i.e., 15 bits with a minimum resolution of 0.004) value. To include the 2 LSb's, you would have to use a 32-bit integer. If you store it left-justified in the 32-bit integer and mask off the Flag bits, scale will be Q31. Reading will "peg" at these values, but the instrument is specified only over its calibrated range - these end values may not even be reachable.

The calibrated Full Scale + and - inclination values for a unit with +/- 60 degree "full-scale reading" correspond to +/- 60 deg. In the Q31 scaled integer above with 18 actual data bits, +60 degrees corresponds to 0x3A980000, and -60 degrees corresponds to -0x3A980000 / [0x3A980000+0x80000000] / 0xBA980000. Converting these back to angle yields +/- 60.000 degrees.

```
SaturationFlag= D0 & 1;           // ==1 if Analog front end may have been in
                                   // saturation for part of the sample interval.
                                   // The sample intervals of multiple output samples
                                   // overlap, so a single input sample can set this
                                   // bit for several output samples.
                                   // If Data Averaging is enabled, this flag is not
                                   // kept ON for the additional output samples
                                   // potentially affected.

// Reading is 32 bit fixed signed integer, type int or long depending on your
// architecture.
// Store the bytes D0,D1,D2 into Reading with D2 occupying the MSB of the data element.
```

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

if(Reading & 0x80000000)         // Determine if signed
    { Set_Negative = TRUE;
      Reading -= 0x80000000; }

Reading = Reading >> 12;         // Shift for scaling in 32b

if(Set_Negative)
    {Reading *= -1000;}          // Reading should represent as -xx.xxx degrees.
else
    {Reading *= +1000;}          // Reading should represent as +xx.xxx degrees.

2.2 Messages to the Device:
```

Short command (there's only one), 3 bytes:

Poll Unit (either or both axes)

Maintenance command set, see next section.

Maintenance command set, see later section.

Negative Acknowledge (~arg = 1's compliment)

If UAID byte specifies both axes, both axes will reply in a Twin Packet.

3.0 Maintenance Command Set

Unless flagged (*immediate*), configuration changes do not affect operation until they are saved in Flash. Baudrate and RS422 Emulation ON/OFF don't take effect until Reset (by command or by cycling power).

Broadcast Mode versions are shown below for fixed commands only, just to provide examples and precalculated Checksum values.

3.1 Long Commands

arg (binary)
\$AC, UAID, 0000 0011, Checksum
Broadcast X: \$AC,\$01,\$03,\$4F
Broadcast Y: \$AC,\$02,\$03,\$4E
Broadcast X+Y:\$AC,\$03,\$03,\$4D

Reset (no reply).

\$AC, UAID, 0000 0010, Checksum
Broadcast X+Y:\$AC,\$03,\$02,\$4E

Break (no reply, just doesn't start sending data). Terminates RS422 mode to allow commands. Send this repeatedly for at least 2 seconds at power on. Command will be received in 28 ms window before first transmission. Can be sent as a long string of repeats padded with 2 or more nulls or rubouts between them.

\$AC, UAID, 0000 0001, Checksum
Broadcast X: \$AC,\$01,\$01,\$51
Broadcast Y: \$AC,\$02,\$01,\$50
Broadcast X+Y: \$AC,\$03,\$01,\$4F

Allow Update. Must immediately precede Update Configuration. Any intervening bus traffic except the Acknowledge messages will void the command even if previously Acknowledged. In fact, a Nak (Negative Acknowledge) message from either axis will void it for both axes.

\$AC, UAID, 0000 0000, Checksum
Broadcast X: \$AC,\$01,\$00,\$52
Broadcast Y: \$AC,\$02,\$00,\$51
Broadcast X+Y: \$AC,\$03,\$00,\$50

Update Configuration. Writes the editing copy of Configuration Vector to Flash. Must immediately follow Allow Update. *Note:* Ack/Nak reply is delayed about 32 ms to allow updating Flash. Wait for reply or a longer timeout than this before sending another command.

\$AC, UAID, xxxx xx11, Checksum

0 < xxxx xx <= 1001 11
Assign Unit ID (Invalid if Broadcast; *Takes effect upon save to Flash*). No conflict with any other commands due to range limit. *Note:* The upper limit may be changed to 0111 11 in a future version.

To avoid confusion with Query replies, there is no NAK (Negative Acknowledge) message for the 1011 xxxx group:

\$AC, UAID, 1011 0xxx, Checksum

0 <= xxx <= 100

Select Baudrate

(Must save to Flash, takes effect at Reset).
{19.2=0, 38.4=1, 57.6=2, 115.2=3, 230.4=4}KB

\$AC, UAID, \$B7, Checksum

Broadcast X: \$AC,\$01,\$B7,\$93

Broadcast Y: \$AC,\$02,\$B7,\$92

Broadcast X+Y:\$AC,\$03,\$B7,\$91

ENQ (Ping) (Broadcast is legal if only one unit is connected.)

Returns \$A0-packaged ASCII string with program rev, angle range, any compile-time options, axis ID (hex) and 'Dual', 'Single' or 'Error!':

\$A0 | UAID | N | ASCII string | Checksum

N = total packet length.

\$AC, UAID, 1011 1xxx, Checksum

Query Setting xxx (Invalid if Broadcast).

Reply: \$A3, UAID, <arg>, Checksum.

Returns <arg> matching the corresponding command's parameter value computed from the Editing copy of the Configuration Vector. Config Byte bits are altered by several commands, and so is an exception to this.

xxx = 000: Config Byte

Config_byte: b0 = Normal Polarity

b1 = NOT(Averaging Enabled)

b2 = NOT(Continuous Averaging Mode)

b7 = RS422 Emulation

001: <delay> (Minimum Response Delay setting)

010: <Pcount> (RS422 Output Period setting - 1)
Note, RS422 Output Period setting applies only if Continuous Averaging and Averaging are both ON (Config Byte bits 1 & 2 both 0).

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

<delay>, <Pcount> and <Acount> values returned are the values used in the corresponding command arguments.

\$AC, UAID, \$BF, Checksum

Send Configuration Vector (Query command 7)

(Invalid if Broadcast)

Returns the Editing Copy of the configuration, not the copy saved in Flash memory. Also returns status indicating mismatch between the two.

Reply to Send Configuration Vector (subject to continuing development):

\$A0, UAID, N, x, {vector bytes}, Checksum

N = total packet length = 11 presently.

x = 0 if Editing Copy matches Flash,
else x = byte # (Baud_Sel byte is #1) of 1st mismatch.

{vector bytes} = {Baud_Sel, ~RepDly, Config_byte, N_avg, ~Output_Pd, dcf}

Baud_Sel: As shown for Select Baudrate command;
>4 implies default baudrate.

~RepDly: 1's compliment of <delay> in Set Minimum Response Delay
command. Min Delay (milliseconds) = delay/32.768.

Config_byte: b0 = Normal Polarity
b1 = NOT(Averaging Enabled)
b2 = NOT(Continuous Averaging Mode)
b7 = RS422 Emulation

N_avg = <Acount> value in Set Max Averaging Time Constant command.

~Output_Pd: 1's compliment of <Pcount> value in Set RS422 Output
Period command. Period = (Pcount + 1)/90 seconds.

dcf: (reserved for Jewell Instruments use)

\$AC, UAID, 1100 0010, Checksum

RS422 Emulation Off (Invalid if Broadcast;
Must save to Flash, takes effect at Reset).

\$AC, UAID, 1100 0011, Checksum

RS422 Emulation On (Invalid if Broadcast;
Must save to Flash, takes effect at Reset).

\$AC, UAID, 1100 0100, Checksum

Broadcast X: \$AC,\$01,\$C4,\$8D

Broadcast Y: \$AC,\$02,\$C4,\$8C

Broadcast X+Y:\$AC,\$03,\$C4,\$8B

Averaging OFF (*immediate*).

Turns off Averaging and resets average.
Cancels selection of Continuous mode, so a
subsequent Averaging ON command will
restart with Continuous mode OFF.

\$AC, UAID, 1100 0101, Checksum

Broadcast X: \$AC,\$01,\$C5,\$8C

Broadcast Y: \$AC,\$02,\$C5,\$8B

Broadcast X+Y:\$AC,\$03,\$C5,\$8A

Averaging ON (*immediate, resets average*).

Can be used to reset average in Continuous
mode. 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. At 255 (this is alterable, see Set
Max Averaging Time Constant for equations),
averaging becomes exponential. In RS422
mode, it stays thus. If near +/- "peg"
value, the count may decrement to avoid an
overflow error. Sets Status bit 2.

\$AC, UAID, 1100 0110, Checksum

Broadcast X: \$AC,\$01,\$C6,\$8B

Broadcast Y: \$AC,\$02,\$C6,\$8A

Broadcast X+Y:\$AC,\$03,\$C6,\$89

Continuous Averaging OFF (*immediate, doesn't
reset average*).

Only turns off Continuous mode, does not
cancel Averaging ON, so is not the exact
compliment of Continuous Averaging ON cmd.

\$AC, UAID, 1100 0111, Checksum

Broadcast X: \$AC,\$01,\$C7,\$8A

Broadcast Y: \$AC,\$02,\$C7,\$89

Broadcast X+Y:\$AC,\$03,\$C7,\$88

Continuous Averaging ON (*immediate, doesn't
reset average*).

Same as Averaging ON except in RS485 mode,
polling does not reset the count and start
a new average. In RS422 mode, enables using
Output Period setting. Sets Status bit 2.

\$AC, UAID, 1100 1000, Checksum

Broadcast X: \$AC,\$01,\$C8,\$89

Broadcast Y: \$AC,\$02,\$C8,\$88

Broadcast X+Y:\$AC,\$03,\$C8,\$87

Select Reverse Polarity (*immediate*)

\$AC, UAID, 1100 1001, Checksum

Broadcast X: \$AC,\$01,\$C9,\$88

Broadcast Y: \$AC,\$02,\$C9,\$87

Broadcast X+Y:\$AC,\$03,\$C9,\$86

Select Normal Polarity (*immediate*)

\$AC, UAID, 1100 1010, Checksum

Broadcast X+Y:\$AC,\$03,\$CA,\$85

Recall Saved Averaging & Polarity Settings
(*immediate, resets average*)

3.2 Extended Commands (5 bytes):

The <argument> is 1 byte.

All comments above re: Long commands apply to these also.

Broadcast variants are shown for convenience, and their absence here has no significance. They're all valid unless noted otherwise.

\$AF, UAID, \$CD, <delay>, Checksum
Broadcast X+Y:
\$AF, \$03, \$CD, <delay>, Checksum

Set Minimum Response Delay (*immediate*).
Adds <delay>/32.768 milliseconds to latency for all replies. Intended for fixing bus collision problems with masters that are slow to release the bus. Timing of command's Acknowledge (if not Broadcast Mode) reflects the change. This should be set long enough so the Pre-Mark Interval, wherein device holds the bus at Mark level before sending, does not overlap the master's Output Enable. Otherwise noise may be increased due to driver voltage offsets.

\$AF, UAID, \$E2, <Pcount>, Checksum

Set RS422 Output Period
(*Must save to Flash, takes effect at Reset. <Pcount> = 0 restores full output rate*).
RS422 output rate will be $90/(1+\text{Pcount})$ Hz if Continuous Averaging is ON, otherwise output rate is 90 Hz.

\$AF, UAID, \$E4, <Acount>, Checksum
Broadcast X:
\$AF, \$01, \$E4, <Acount>, Checksum
Broadcast Y:
\$AF, \$02, \$E4, <Acount>, Checksum
Broadcast X+Y:
\$AF, \$03, \$E4, <Acount>, Checksum

Set Max Averaging Time Constant (*immediate*).
Does not reset the average, and current selection of Averaging Mode remains in effect. <Acount> (1 byte) is max number of samples held in the average. When this many samples accumulate in the average, the Aux byte reaches <Acount> samples and stops (if near a +/- "peg" limit, it may occasionally decrement to prevent an overflow error). With the next sample, the averaging becomes exponential, with time constant $\text{Tau} = 1/(90 \cdot \ln(N/(N-1)))$ seconds ($N = \text{Acount} + 1$) and a single-pole cutoff frequency $F_c = 1/(2 \cdot \pi \cdot \text{Tau})$ Hz. For a desired pole frequency, use this equation: $N = 1/(1 - \exp(-F_c \cdot 2 \cdot \pi / 90))$; <Acount> = N-1.

\$AF, UAID, \$E5, <Acount>, Checksum

Set Max Averaging Time Constant and Enable Averaging (*immediate*).
Equivalent to above + Averaging ON except does not reset the average.

\$AF, UAID, \$E7, <Acount>, Checksum

Set Max Averaging Time Constant and Enable Continuous Averaging (*immediate*).
Equiv. to above + Continuous Averaging ON. Does not reset the average.

4.0 General Guidelines on Averaging Time Constant and Output Sample Rate

Because the averaging filter has only a 1-pole response, the <Pcount> argument in Set RS422 Output Period command and the <Acount> argument in the 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. If $M = \text{Pcount} + 1$ and $N = \text{Acount} + 1$, the Nyquist frequency is $90/(2 \cdot M)$ Hz and attenuation at the Nyquist frequency is

$$-dB = 10 \cdot \log(1 + (\pi / (M \cdot \ln(N / (N - 1))))^2).$$

The above does not apply when the Output Period is 1/90 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 $3 \leq N \leq 256$, 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.

5.0 Flash Memory Verification:

In mission-critical applications, a device should report that it has a problem but nevertheless 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. 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:

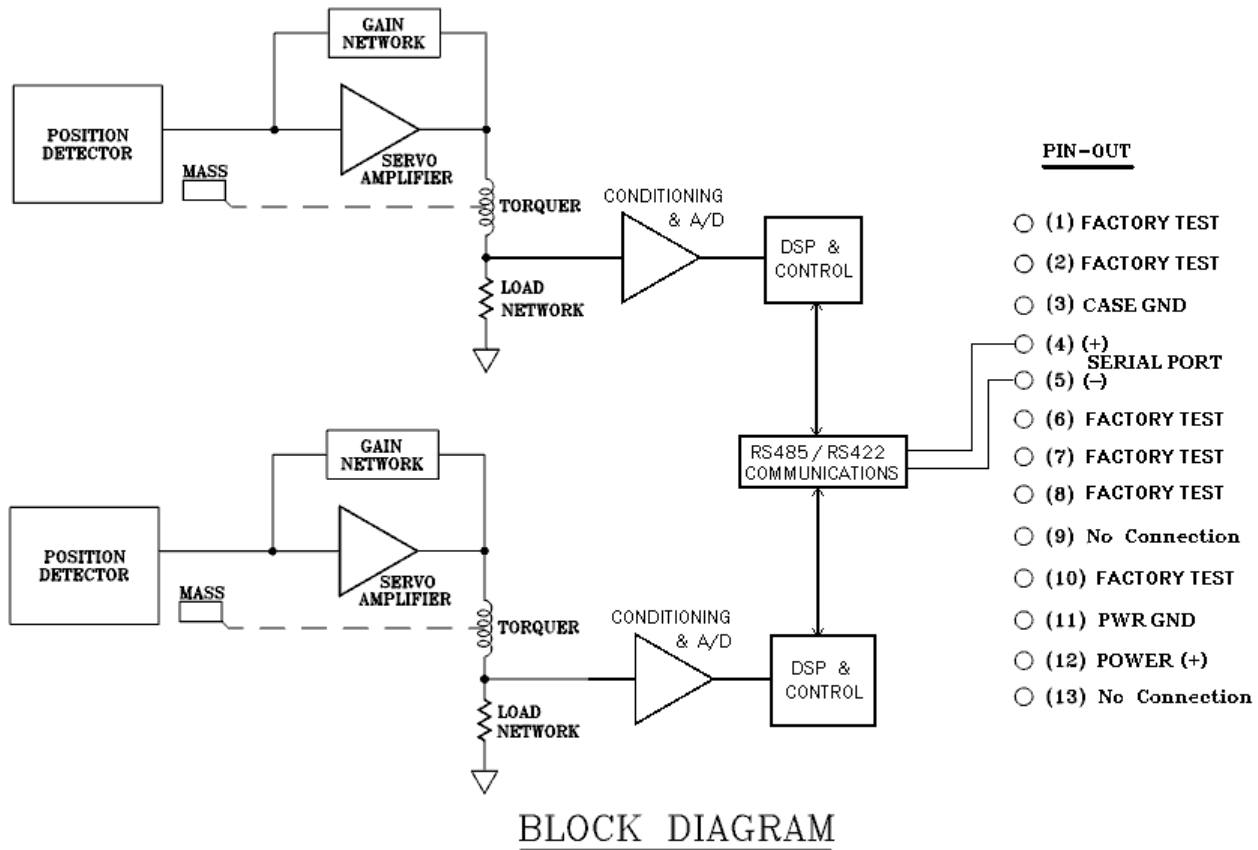
7	6	5	4	3	2	1	0	
x	x	0	1	x	0	1	x	Program Checksum Error (highest priority)
x	x	0	1	x	1	1	x	Calibration Checksum Error
x	x	0	1	x	1	0	x	Filter Coefficient Mismatch
x	x	0	1	x	0	0	x	Unused Flash not blank (lowest priority)
x	x	x	0	x	x	x	x	No Checksum Error nor Filter Coefficient Mismatch; D0.(2,1) then indicate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

6.0 Electrical Installation

The DXI-200 operates from a single positive supply voltage from 9 to 30 v. Its current drain is nearly independent of supply voltage as long as the voltage is within this range. Only 5 of the 13 pins in the connector are used in an installation:

3: Case Ground
4: Serial i/o +
5: Serial i/o -
11: Power Gnd
12: Power V+



Connections should be made via a 4 conductor shielded cable having a twisted pair for the Serial i/o leads. 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 (see Break command in section 3.1). A bias network is commonly used in RS485 installations to maintain the line at Mark level when no node is transmitting. The DXI-200 does not require bias, but your RS485 adaptor might - follow the recommendations given for the adaptor. See also section 6.2 below regarding Common Mode voltage.

6.1 Multiple Unit Installations

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.

6.2 Long Lines:

The major issue with long lines 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 DXI-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.

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.

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

