Quick Start Guide

for

eDXI-100/200 Inclinometer

REVISION HISTORY

Revision	Date	Change Description
1	5/18/15	Initial release

TABLE OF FIGURES

TABLE OF TABLES

MANUAL ACRONYMS & NOTATIONS

DXI   Digital series Precision Inclinometer

IP   Internet Protocol

PoE   Power over Ethernet

TCP   Transmission Control Protocol

UDP   User Datagram Protocol

0x__   0x indicates hexadecimal notation

1   OVERVIEW

This quick start guide covers the installation and operation of the Jewell Digital Servo Inclinometer, model number DXI-E-200-3/3. The inclinometer provides continuous dual axis measurements of inclination via TCP/IP over a linear range of +/-3 degrees with resolution down to 0.001 degree. Power is supplied via a standard 4 twisted pair category 5 cable or higher and is compliant to the IEEE 802.3af Power-over-Ethernet (PoE) standard.

2   INSTALLATION

The following sections cover mechanical connections, mechanical installation, and PoE installation.

2.1   MECHANICAL CONNECTIONS

The inclinometer is assembled with an Amphenol RJF Series receptacle which when mated to an RJF series plug is sealed to a rating of IP67. For convenience, Jewell has included the mating plug to ease installation. The receptacle and plug allow for fast connections using the ¼ turn bayonet coupling mechanism per MIL-C-26482. The figure below is borrowed from the Amphenol RJ Field Series datasheet; use it to help install the plug on the ethernet cable.
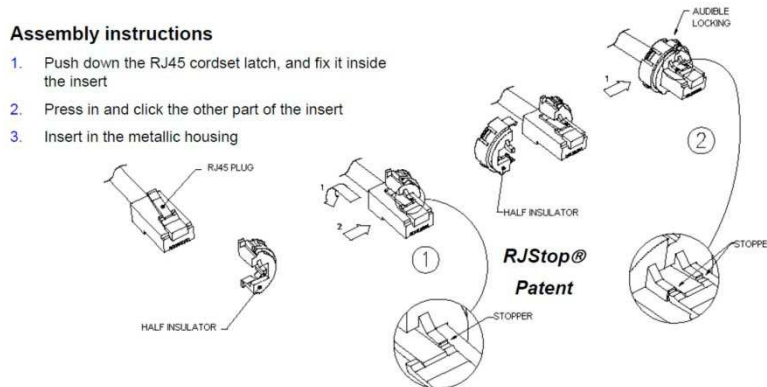
Figure 1: Amphenol RJ Field series plug assembly

IMPORTANT: The plug and receptacle are keyed. Be sure to match the keying between the plug and receptacle before the two half insulators (step 2 in the diagram above) are inserted into the metal housing. Once the RJ45 and half insulator are inserted into the metal housing, the connection cannot be undone!

2.2   MECHANICAL INSTALLATION

The figures below show the hole pattern required to mount the inclinometer and outline dimensions of the inclinometer. Dimensions are provided in Imperial units of inches, shown without brackets, and SI units of millimeters, shown within brackets. The base plate is clear anodized 7075 aluminum.

Figure 2: DXI-E-200 Outline & Dimensions
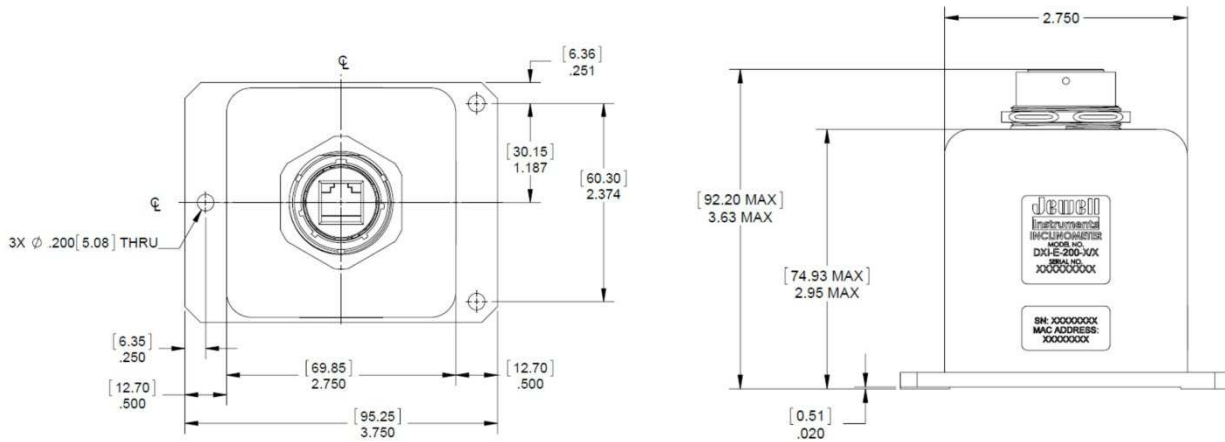
2.3   POWER OVER ETHERNET (POE)

The DXI-E-200 inclinometer extracts power from a conventional twisted pair category 5 cable (or better) and the interface is compliant to the IEEE 802.3af Power-over-Ethernet (PoE) standard. The power over the twisted pairs may either be supplied by endspan or midspan Power Sourcing Equipment (PSE). Per the IEEE standard, the inclinometer is defined a powered device and DC powered may be supplied on any of the 4 twisted pairs within the ethernet cable. More specifically, the DXI-E-200 inclinometer is a Type 1 class 1 powered device which allows the PSE to supply no more than 3.84W[1].

3   NETWORK CONFIGURATION

The DXI-E-200 inclinometer uses TCP/IP for network communications. Other noteworthy supported protocols include: Telnet, DHCP, and HTTP.

Upon connection to the PSE and a network, the DXI-E-200 inclinometer will use by default DHCP to automatically assign an IP address. Utilize the hardware address of the inclinometer to identify the IP address on the network. The hardware address is also known as the MAC address. The MAC address is provided on a label on the side of the inclinometer. The first three bytes of the MAC address will start with 00-80-A3. The full MAC address will be 00-80-A3-97-E7-46.

The IP address can be located using the following steps on a Windows based system: Open the Windows command line interpreter (CMD) to get to a DOS prompt. Type "arp -a" in the command prompt. This lists a number of MAC addresses with the associated IP addresses on the local network. Scroll down the list to find the IP address associated with the MAC address 00-80-A3-97-E7-46 and take note of the IP address.

---

[1] The class 1 physical layer classification allows PSE equipment capable of active power management to reduce power to the inclinometer to 3.84W.

3.1 MANUAL IP ADDRESS CHANGE

The IP address can be changed manually and caution should be exercised when doing so otherwise duplicate network IP address will occur. Duplicate IP addresses will most likely result in temporary loss of network communication with the DXI-E inclinometer[2]. Before manually changing the IP address, the new IP address should be checked on the network to ensure it is not used. Perform this check by using the DOS ping command with the new IP address from the DOS prompt. If no response is received, then the IP address is okay to use.

To change the IP address, perform the following steps. Change only the IP address fields within the "Server" configuration, as highlighted in Figure 3 below.

1. Telnet into port 9999 using the existing IP address
2. Press 0 to enter the Server setup
3. Type the new IP address into the fields and press return after each octet0
4. Press enter through the remaining Server entries
5. When back at the Change Setup menu, press 9 to save.

```
Change Setup:
  0 Server
  1 Channel 1
  2 Channel 2
  5 Expert
  6 Security
  7 Defaults
  8 Exit without save
  9 Save and exit           Your choice ? 0

IP Address : (192) .(168) .(002) .(240)
Set Gateway IP Address (N) ?
Netmask: Number of Bits for Host Part (0=default) (0)
Set DNS Server IP addr  (N) ?
Change Telnet/Web Manager password (N) ?
```

Figure 3: Telnet, Port 9999 Menu

Upon completion, it is best to disconnect the ethernet cable, then reconnect; this cycles power to the inclinometer. You should now be able to Telnet into the DXI-E with new IP address.

IMPORTANT: Do not change any other settings of the DXI-E inclinometer. Doing so may result in loss of X & Y packet data from the inclinometer. Please contact Jewell Instruments for issues with network communication.

4   DXI-E COMMUNICATION PROTOCOLS

The DXI-E inclinometer transmits measurements of inclination at an output rate of 60 samples per second; this is the default setting for the output rate. This data may be accessed by Telnet connection into the sensor at port 10001. The Data is packetized into 7 hexadecimal bytes. X and y axes are transmitted in different packets as follows:

---

[2] If duplicate IP addresses are accidently entered, the user may have to release/renew the conflicting IP address. The DNS server should then reassign a new address to the "other" network device. The inclinometer's IP address should now be acceptable.

<Prefix><UAID ><D0><D1><D2><AUX><Checksum>

Where:

Prefix = this byte wide packet always indicates the start of either X or Y data. The value is constant.

UAID = this byte wide packet is the Unit Address/Identifier. For X-data the byte value is 0x71; Y data the byte value is 0x72

D0 = inclination data

D1 = inclination data

D2 = inclination data + flags

AUX = number of samples average (not covered in this Quick Start Guide)

Checksum = the byte wide ones complement sum of all prior packet values.

Example packets are given below and will be used for discussion in .

Table 1: Example X and Y data packet values

Byte Num:	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
Name	Prefix	UAID	D0	D1	D2	AUX	Checksum
Ex. X Value	0xA6	0x71	0x80	0x0F	0x80	0x42	0x95
Ex. Y Value	0xA6	0x72	0xC0	0x2A	0x00	0x6D	0x8E

4.1 CHECKSUM & CHECKSUM CALCULATION

The checksum byte is provided to help identify packet bits errors. It should be used to ensure data integrity at the host equipment. The calculation of the checksum is a four step process outlined below.

1. Sum bytes 1-6
2. Right shift the most significant byte 8 bits & sum again
3. Find the ones complement of only the least significant byte

4.1.1 EXAMPLE

Using the X value packet from Table 1 the checksum is calculated as follows.

1. 0xA6 + 0x71 + 0x80 + 0x0F + 0x80 + 0x42 = 0x268
2. 0x02 + 0x68 = 0x6A
3. 0xFF – 0x6A = 0x95

The calculated value above from the example matches the checksum packet value from Table 1.

4.1.2 PYTHON EXAMPLE

The python code below can be used to calculate the checksum. The packet data is contained in the decimal array named DXIpacket[] in elements 0 through 6. Because Python does not handle calculation in hexadecimal format all that well, all calculations were performed in decimal.

```
checksumA = sum(DXIpacket[0:6])       # sums all elements of DXIpacket except the checksum in element #7
checksumLO = checksumA&255            # find the low byte                     ( 255 dec = 0x00FF)
checksumHI = (checksumA&65280) >> 8   # find the high byte, then right shift 8 bits (65280dec = 0xFF00)
checksumB =  checksumLO + checksumHI  # add the low and high
checksumC = checksumB&255             # keep only the low byte of the sum
checksum = 255 - checksumC            # now take the binary complement, but in decimal notation.
```

checksumA, checksumB, checksumB, checksumLO, checksumHI are all temporary variables. The final checksum result is stored in checksum.

4.2 DATA CALCULATION

Calculation of the data bytes are more involved because the D0 packet contains both data and flag bits. The remaining D0 data bits must be concatenated with D1 and D2 to form an 18 bit value, then the calculation can be performed. The resulting units are degrees. Calculation is an X step process outlined below.

1. Determine the sign of the final value contained D2 bit 7
2. Right shift D0 by 6 bits
3. Left shift D1 by 2 bits
4. Mask off the most significant bit of D2, then left shift 10 bits
5. Sum the left and right shifted values from D0, D1, & D2
6. Multiply the sum by the sign and divide by 1000

4.2.1 EXAMPLE

Using the X value packet from Table 1 the inclination value is calculated as follows.

1. D2 = 0x80. Thus D2 bit 8 = 1; this indicates a negative sign. (0 indicates a positive sign)
2. D0 = 0x80. D0 right shifted six bits = 0x00.
3. D1 = 0x0F. D1 light shifted 2 bits = 0x3C.
4. D2 = 0x80. D2.8 masked off, then left shifted 10 bits = 0x000.
5. D2 + D1 + D0 = 0x000 + 0x3C +00x00 = 0x03C.
6. -1 *(0x03C / 1000) = -1* 60/1000 = -0.060 degrees.

4.2.2 PYTHON EXAMPLE

The python code below can be used to calculate the data value of inclination. The packet data is contained in the decimal array named Packet[] in elements 2 through 4. Because Python does not handle calculation in hexadecimal format all that well, all calculations were performed in decimal.

```
D0   = Packet[2]                    # data & flags
D1   = Packet[3]                    # data
D2   = Packet[4]                    # data

if (D2>>7 == 1):
        signbit = -1
else:
        signbit = 1

Data0 = (D0>>6)                     # right shift the D0 data 6 bits
Data1 = (D1<<2)                     # left shift the D1 data 2 bits
Data2 = ((D2&127)<<10)              # this masks off the sign bit first, then left shifts the value
Data[1] = signbit*(Data0 + Data1 + Data2)/1000      # calculate the inclination
```

D0, D1, D2, Data0, Data1, and Data2 are all temporary variables. The final checksum result is stored in the array named Data within element 1.

4.3 FLAG CALCULATION

Flags bits are provided by the DXI-E to report error conditions within the sensor. The four flag bits listed below are located within the D0 byte of the packet, bits 5 to 0. Decoding the bits is relatively straightforward as shown below.

Table 2: D0 Error Flag Bits

7	6	5	4	3	2	1	0	Flag or Error
x	x	0	1	x	0	1	x	Program Checksum
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
x	x	x	0	x	x	x	x	No error

4.3.1 PYTHON EXAMPLE

The python code below can be used to determine the individual flag bits from the D0 packet. The packet data is contained in the decimal array named Packet[] in elements 2 through 4. Entering this code snippet, the D0 byte was converted into an integer value.   Because Python does not handle calculation in hexadecimal format all that well, all calculations were performed in decimal.

```python
FLAG_MASK = int('0x36',16)

# Look at the Flag bits for errors
Flags = D0 & FLAG_MASK                  # mask off data and don't care bits
if Flags == 18:                         # 18 dec = 0x12
        ProgChecksumErr = 1
        ErrorFlag = 1
elif Flags == 22:                       # 22 dec = 0x16
        CalibChecksumErr = 1
        ErrorFlag = 1
elif Flags == 20:                       # 20 dec = 0x14
        FilterCoeffErr = 1
        ErrorFlag = 1
elif Flags == 16:                       # 16 dec = 0x10
        FlashNotBlankErr = 1
        ErrorFlag = 1
elif Flags == 0:                        # 0 dec = 0x00
        ErrorFlag = 0
else:
        ErrorFlag = 0
```

The variables ProgChecksumErr, CalibChecksumErr, FilterCoeffErr , FlashNotBlankErr , and ErrorFlag are all reported to the higher level process.

APPENDIX B: WARRANTY & LIMITATION OF LIABILITY

Standard Force-balance products (those listed in Jewell Instruments' published sales literature, excluding software) manufactured by Jewell Instruments LLC are warranted against defects in materials and workmanship for twenty-four (24) months from the date of shipment from Jewell's premises with the following exceptions: Series 900 analog or digital clinometers are warranted against defects in materials and workmanship for 90 days from the delivery date. Jewell will repair or replace (at its option) goods that prove to be defective during the warranty period provided that they are returned prepaid to Jewell and:

(a) that the goods were used at all times for the purpose for which they were designed and in accordance with any instructions given by Jewell in respect of them,

(b) that notice is received by Jewell within 30 days of the defects becoming apparent, and

(c) that return authorization is received from Jewell prior to the goods being sent back.

Should goods be damaged in transit to the purchaser, Jewell will accept no liability unless the purchaser can show that such damage arose solely from Jewell's failure to pack the goods properly for shipment.

Software products are warranted to perform substantially in accordance with their documentation for 90 days following your receipt of the software. Jewell and its suppliers do not and cannot warrant the performance or results you may obtain by using the software or its documentation.

In respect of goods or parts thereof manufactured by others and resold by Jewell, Jewell will pass on to the customer the benefit of any guarantee or warranty received by Jewell from the original manufacturer insofar as such guarantee or warranty is assignable.

ANY OTHER CONDITIONS OR WARRANTIES WHETHER EXPRESS OR IMPLIED BY STATUTE OR OTHERWISE ARE EXCLUDED. THE REMEDIES PROVIDED HEREIN ARE THE BUYER'S SOLE AND EXCLUSIVE REMEDIES. JEWELL INSTRUMENTS LLC SHALL NOT BE LIABLE FOR ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES, INCLUDING LOST PROFITS OR LOST SAVINGS, WHETHER BASED ON CONTRACT, TORT, OR ANY OTHER LEGAL THEORY. THIS WARRANTY EXTENDS ONLY TO THE ORIGINAL PURCHASER AND IS EXPRESSLY IN LIEU OF ALL OTHER WARRANTIES, WHETHER OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR USE, AND OF ALL OTHER OBLIGATIONS AND LIABILITIES OF ANY KIND AND CHARACTER. THERE ARE NO WARRANTIES WHICH EXTEND BEYOND THE DESCRIPTION ON THE FACE HEREOF.

Jewell's liability arising out of the sale of its goods is expressly limited to the repair and/or replacement of defective parts or the cost of such repair and/or replacement.

If software does not perform substantially in accordance with the documentation, the entire and exclusive liability and remedy shall be limited to either, at Jewell's option, the replacement of the software or the refund of the license fee you paid for the software.

Liability for any other form of loss or damage is hereby expressly excluded.

Customer shall indemnify Jewell against any third party claim arising out of the use of goods and/or services supplied by Jewell, including any claim arising directly or indirectly out of alleged negligence on the part of Jewell, its employees, servants, representatives or agents.

DSPM *Industria*®
sensori & trasduttori
Via Paolo Uccello 4 - 20148 Milano
Tel +39 02 48 009 757   Fax +39 02 48 002 070   info@dspmindustria.it   www.dspmindustria.it

8 | P a g e