

# JDX Modbus Series User Manual



Contact us: Jewell Instruments, LLC  
850 Perimeter Road  
Manchester, NH 03103

Phone: 603-669-6400  
Fax: 603-622-2690  
Website: [www.jewellinstruments.com](http://www.jewellinstruments.com)  
Email: [sales@jewellinstruments.com](mailto:sales@jewellinstruments.com)



Copyright © 2021 by Jewell Instruments LLC. All rights reserved.



**REVISION HISTORY**

<b>Revision</b>	<b>Date</b>	<b>Change Description</b>
1	2/19/2020	Initial Release-prototype
2	3/6/2021	Updates to Configuration Register Bank (300-327), Moving Average Filter, Relative Zero Command additions, pinout and installation detail added, minor table corrections.

<b>TABLE OF TABLES AND FIGURES</b>	<b>iii</b>
<b>MANUAL ACRONYMS &amp; NOTATIONS</b>	<b>iv</b>
<b>1.0 OVERVIEW</b>	<b>1</b>
<b>2.0 INSTALLATION REQUIREMENTS</b>	<b>2</b>
2.1 Power Requirements	2
2.2 Mechanical Connections	2
2.3 Mechanical Installation	8
2.4 RS-485 Installation	10
2.5 Galvanic Isolation & JDx Grounding & Transient Protection	10
<b>3.0 COMMUNICATION</b>	<b>11</b>
3.1 Modbus Interface – General Overview	12
3.2 Identity Register Bank	14
3.3 Data1 Register Bank	15
3.4 Data2 Register Bank	17
3.5 Configuration Register Bank	19
3.5.1 Command Registers – Nonvolatile Save Command	23
3.5.2 Command Registers – Nonvolatile Restore Command	23
3.5.3 Command Registers – Test Mode Off Command	24
3.5.4 Command Registers – Test Mode On Mode 1 Command	24
3.5.5 Command Registers – Test Mode On Mode 2 Command	25
3.5.6 Command Registers – User Relative Zero Off Command	26
3.5.7 Command Registers – User Relative Zero On Command	27
3.5.8 Configuration Register Bank – Additional Examples	29
3.6 Communication Register Bank	33
3.7 Coil Bank	35
3.8 Fault/Status Register Bank	37
<b>4.0 MODBUS FUNCTIONS SUPPORTED</b>	<b>39</b>
4.1 Exception Response Codes	39
4.2 FC01 (01) Read Coils	39
4.3 FC03 (03) Read Holding Register	40
4.4 FC04 (04) Read Input Registers	40

4.5	FC05 (05) Write Single Coil	41
4.6	FC06 (06) Write Single Register	41
4.7	FC16 (0X10) Write Multiple Registers	42
4.8	FC08 (08) Diagnostics	43
4.8.1	FC08 (08), Subfunction 0, Return Query Data	43
4.8.2	FC08 (08), Subfunction 1, Restart Communications Option	43
4.8.3	FC08 (08), Subfunction 2, Return Diagnostic Register	44
4.8.4	FC08 (08), Subfunction 10, Clear Counters and Diagnostic Register	44
4.8.5	FC08 (08), Subfunction 12, Return Bus Communication Error Count	44
4.8.6	FC08 (08), Subfunction 13, Return Bus Exception Count	45
4.8.7	FC08 (08), Subfunction 14, Return Server Message Count	45
4.8.8	FC08 (08), Subfunction 18, Return Bus Character Overrun Count	46
<b>5.0</b>	<b>TROUBLESHOOTING</b>	<b>47</b>

## TABLE OF TABLES AND FIGURES

TABLE 1	POWER REQUIREMENTS	2
TABLE 2:	JDX SERIES DB9 PINOUT	3
TABLE 3:	JDX SERIES M12 PINOUT	4
TABLE 4:	JDX SERIES C06 PINOUT	5
TABLE 5:	JDX SERIES DEUTSCH 4-PIN PINOUT	6
TABLE 6:	JDX SERIES UNTERMINATED CABLE (FLYING LEADS)	7
TABLE 7:	MODBUS TIMING	11
TABLE 8:	MODBUS REGISTER MAP	12
FIGURE 1:	JDX DB9 OUTLINE & DIMENSIONS	8
FIGURE 2:	JDX M12 OUTLINE & DIMENSIONS	8
FIGURE 3:	JDX C06 OUTLINE & DIMENSIONS	9
FIGURE 4:	JDX FLYING LEADS, DEUTSCH 4P OUTLINE & DIMENSIONS	9

## MANUAL ACRONYMS & NOTATIONS

JDI	Jewell Digital Inclinator
JDA	Jewell Digital Accelerometer
JDx	JDI or JDA Sensor
0x__	0x indicates hexadecimal notation
CR	Carriage Return, hexadecimal 0x0D
LF	Line Feed, hexadecimal 0x0A (also known as a new line)
CRLF	Abbreviation for the line termination sequence “Carriage Return & Line Feed”
Modbus	A compact communications protocol designed for automation and control systems by Modicon, now part of Schneider Electric SE
Modbus RTU	A compact communications protocol designed for automation and control systems by Modicon, now part of Schneider Electric SE. Modbus RTU uses binary encoded commands and data, including a CRC-16 field as part of all messages. This is a different protocol scheme from Modbus ASCII.
CRC16	16-bit cyclic redundancy check value
RS-485	Also known as EIA-485 or TIA-485(A), a common differential signaling communications network which allows multiple devices to share a common set of wires for transmission and reception

## 1.0 OVERVIEW

This user manual covers the installation and operation of the Modbus RTU protocol models of the Jewell Instruments Digital Inclinometer. The inclinometer provides automatic dual axis measurements of inclination via Modbus register banks over 2-wire RS-485 across the sensor's linear range with resolution down to 0.001 degree. Power is supplied through two pins on the connector.

The sensors are shipped with generic settings and the intention is that the user will reconfigure the sensor to meet the needs of their RS-485 Modbus-RTU application. This manual provides documentation of the supported Modbus-RTU commands which may be used to configure and collect data from the JDx sensor.

For additional product information not covered by this manual, please consult the product data sheet available on our website, or contact our customer support office: 1-800-227-5955 or [sensors@jewellinstruments.com](mailto:sensors@jewellinstruments.com).

## 2.0 INSTALLATION REQUIREMENTS

Some general information for the installation of the JDx sensor is included here. Please refer to the product-specific datasheet or product drawing for connector pinout, mechanical dimensions and mounting patterns. These are also shipped with the product and are available by contacting Jewell Instrument's Customer Support at 1-800-227-5955 or [sensors@jewellinstruments.com](mailto:sensors@jewellinstruments.com).

## 2.1 POWER REQUIREMENTS

The inclinometer's minimum, maximum, and typical power requirements are shown in the following Table 1.

Power Supply Voltage, Minimum	5V DC
Power Supply Voltage, Typical	9V – 36V DC
Power Supply Voltage, Maximum	38V DC
Power, Maximum	800 mW

**Table 1 Power Requirements**

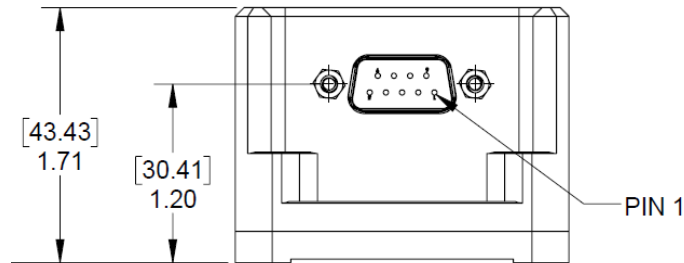
## 2.2 MECHANICAL CONNECTIONS

The sensor is available in several models with various connectors, including an IP67 rated male DB9 connector. In order to achieve the full IP67 rating, an appropriate mating connector with an IP67 rated backshell must be employed and fully engaged. IP67 backshells are readily available from suppliers such as NorComp, Conec and Digi-Key.

The power & communication information for the various JDx sensor models are provided below.



**Table 2: JDX Series DB9 Pinout**



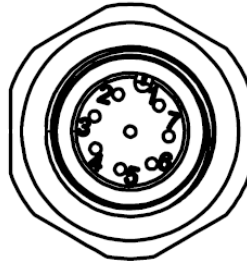
DB9 CONNECTOR RS-485 VERSION	
COLOR	ASSIGNMENT
1	DATA-
2	DATA+
3	N/C
4	N/C
5	GND
6	N/C
7	N/C
8	POWER-
9	POWER+

DB9 CONNECTOR RS-232 VERSION	
COLOR	ASSIGNMENT
1	N/C
2	RXD
3	TXD
4	N/C
5	GND
6	N/C
7	N/C
8	POWER-
9	POWER+

Pins 1 and 2 are polarity specific and are used for the half-duplex RS-485 communication lines. Take note of the polarity. Pin 5 contains the ground reference for the RS-485 lines and depending on installation may not be required. This GND reference IS needed for proper operation of the RS-232 models, as the RS-232 standard requires a Ground reference.

Pins 8 and 9 are the power pins for the JDX sensor. Reverse polarity protection (via a diode bridge) has been implemented internal to the sensors. This means either power or ground can be applied to either pin 8 or 9 without causing electrical damage. **APPLY DC VOLTAGE ONLY.**

Table 3: JDx Series M12 Pinout



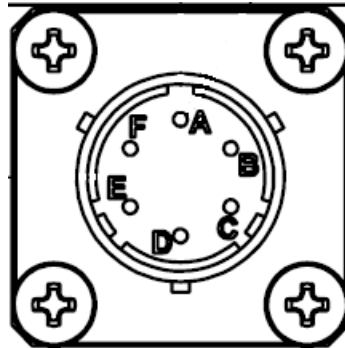
M12 CONNECTOR RS-485 VERSION	
COLOR	ASSIGNMENT
1	N/C
2	POWER+
3	POWER-
4	DATA-
5	GND
6	DATA+
7	N/C
8	GND

M12 CONNECTOR RS-232 VERSION	
COLOR	ASSIGNMENT
1	TXD
2	POWER+
3	POWER-
4	N/C
5	GND
6	N/C
7	RXD
8	GND

Pins 4 and 6 are polarity specific for the half-duplex RS-485 communication lines. Take note of the polarity. Pin 5 contains the ground reference for the RS-485 lines and depending on installation may not be required. This Pin 5 GND reference IS needed for proper operation of the RS-232 models, as the RS-232 standard requires a Ground reference.

Pins 2 and 3 are the power pins for the JDx sensor. Reverse polarity protection (via a diode bridge) has been implemented internal to the sensors. This means either power or ground can be applied to either pin 2 or 3 without causing electrical damage. **APPLY DC VOLTAGE ONLY.**

Table 4: JDX Series C06 Pinout



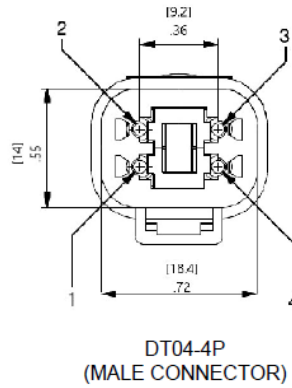
6-PIN CIRCULAR (C06) CONNECTOR RS-485 VERSION	
COLOR	ASSIGNMENT
A	POWER+
B	POWER-
C	N/C
D	DATA+
E	DATA-
F	GND

6-PIN CIRCULAR (C06) CONNECTOR RS-232 VERSION	
COLOR	ASSIGNMENT
A	POWER+
B	POWER-
C	N/C
D	RXD
E	TXD
F	GND

Pins D and E are polarity specific for the half-duplex RS-485 communication lines. Take note of the polarity. Pin F contains the ground reference for the RS-485 lines and depending on installation may not be required. This Pin F GND reference IS needed for proper operation of the RS-232 models, as the RS-232 standard requires a Ground reference.

Pins A and B are the power pins for the JDX sensor. Reverse polarity protection (via a diode bridge) has been implemented internal to the sensors. This means either power or ground can be applied to either pin A or B without causing electrical damage. **APPLY DC VOLTAGE ONLY.**

**Table 5: JDX Series Deutsch 4-Pin Pinout**



4-PIN DEUTSCH (MALE) CONNECTOR RS-485 VERSION ONLY	
PIN	ASSIGNMENT
1	POWER+
2	POWER-
3	DATA+
4	DATA+

Pins 3 and 4 are polarity specific for the half-duplex RS-485 communication lines. Take note of the polarity.

Pins 1 and 2 are the power pins for the JDX sensor. Reverse polarity protection (via a diode bridge) has been implemented internal to the sensors. This means either power or ground can be applied to either pin 1 or 2 without causing electrical damage. **APPLY DC VOLTAGE ONLY.**

**Table 6: JDx Series Unterminated cable (Flying Leads)**

RS-485 VERSION	
COLOR	ASSIGNMENT
RED	POWER+
BLACK	POWER-
GREEN	DATA+
BLUE	DATA-
WHITE	GND

RS-232 VERSION	
COLOR	ASSIGNMENT
RED	POWER+
BLACK	POWER-
GREEN	RXD
BLUE	TXD
WHITE	GND

Green and Blue wires are polarity specific for the half-duplex RS-485 communication lines. Take note of the polarity. White wire contains the ground reference for the RS-485 lines and depending on installation may not be required. This White GND wire reference IS needed for proper operation of the RS-232 models, as the RS-232 standard requires a Ground reference.

Red and Black wires are the power connections for the JDx sensor. Reverse polarity protection (via a diode bridge) has been implemented internal to the sensors. This means either power(+) or power(-)/ground can be applied to either Red or Black without causing electrical damage. **APPLY DC VOLTAGE ONLY.**

### 2.3 MECHANICAL INSTALLATION

The figures below show the hole pattern required to mount the JDx sensor. The figures also provide the dimensions of the inclinometer. Dimensions are provided in Imperial units of inches, shown without brackets, and SI units of millimeters, shown within brackets.

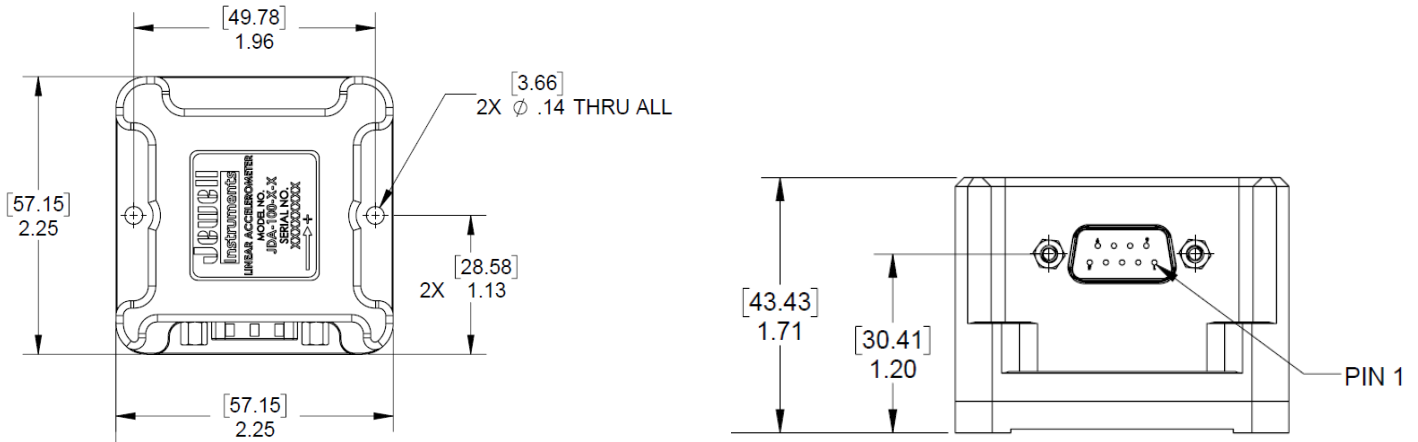


Figure 1: JDx DB9 Outline & Dimensions

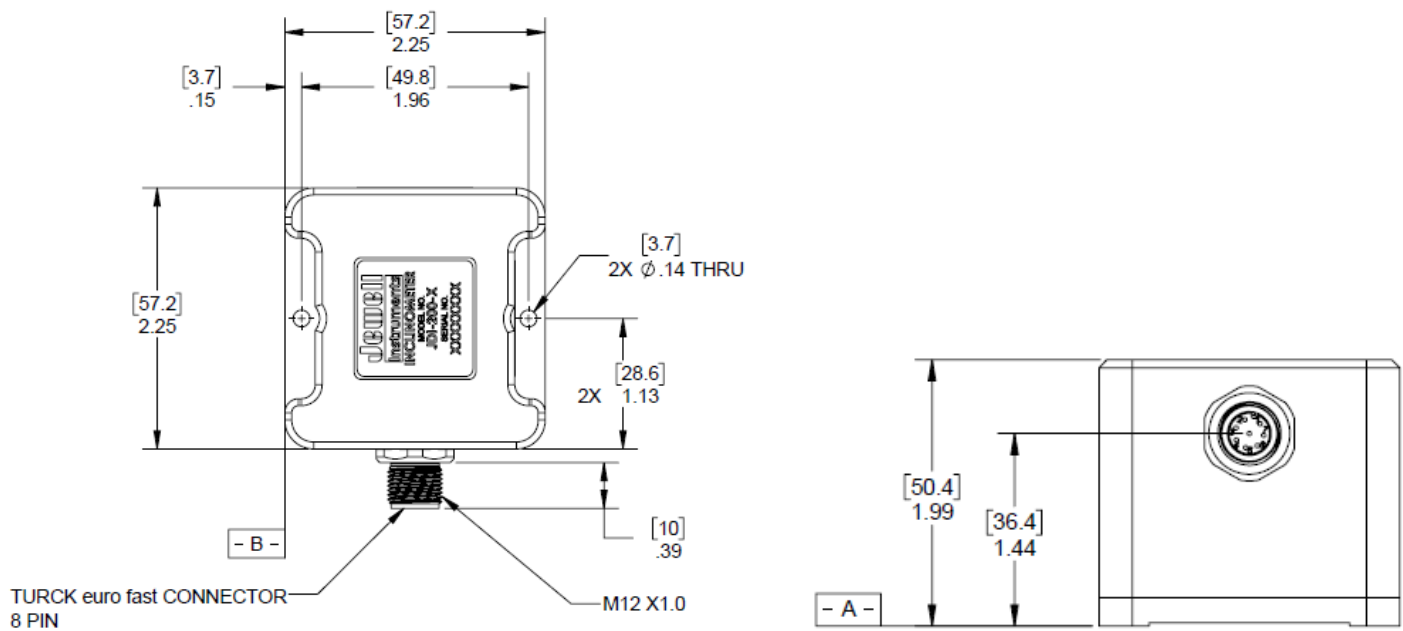


Figure 2: JDx M12 Outline & Dimensions

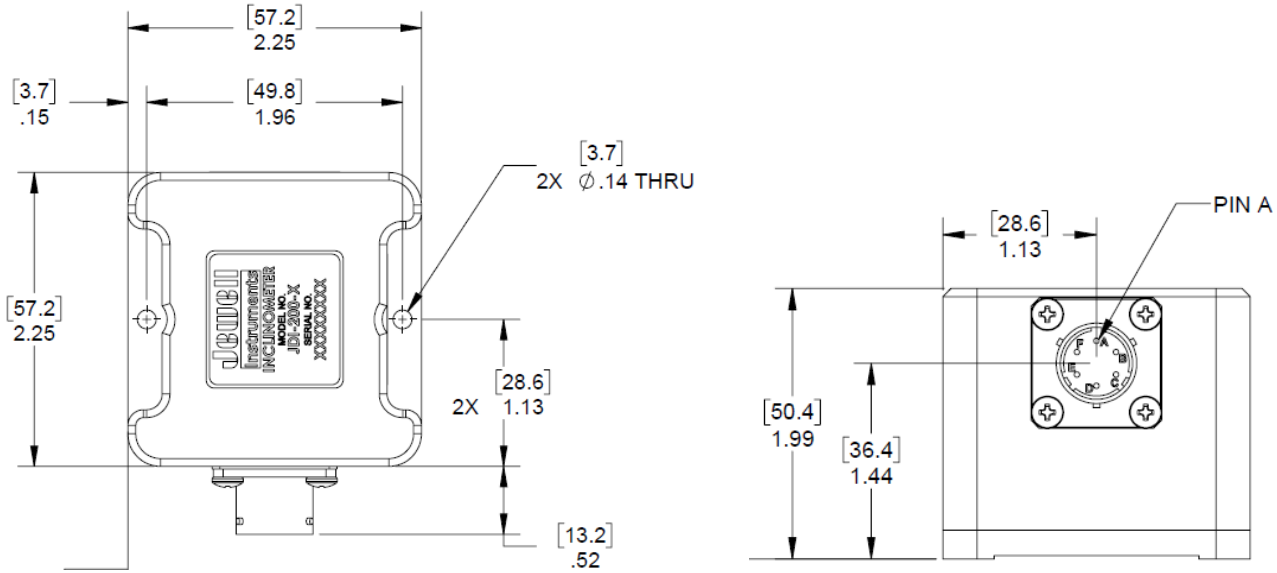


Figure 3: JDX C06 Outline & Dimensions

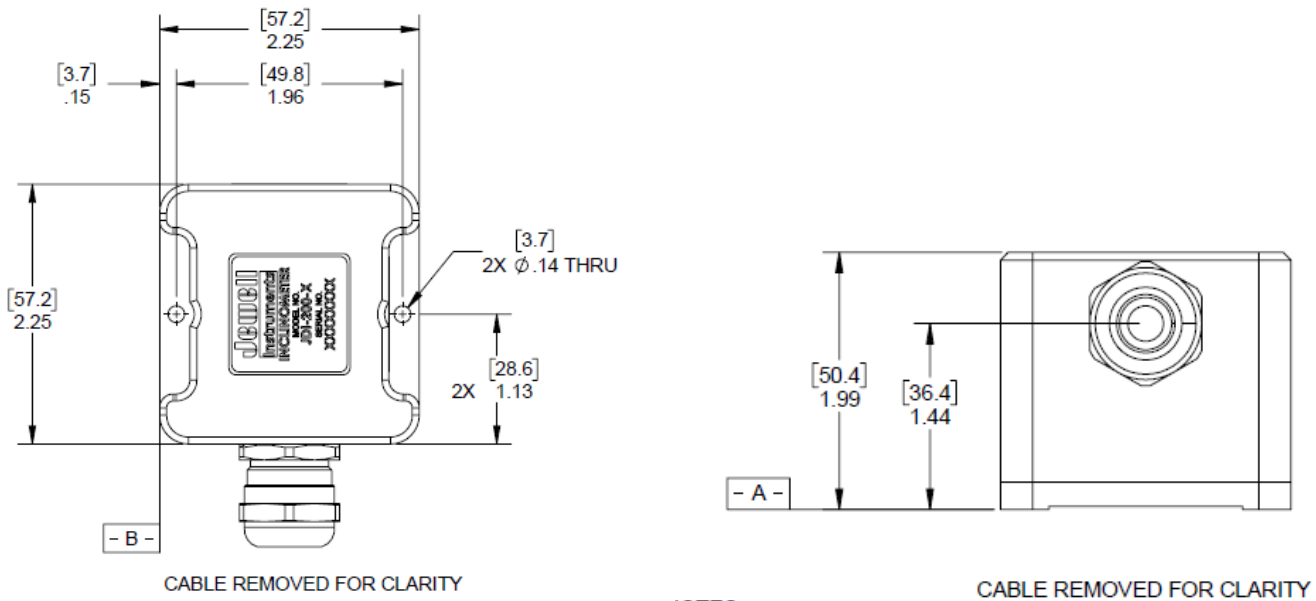


Figure 4: JDX Flying Leads, Deutsch 4P Outline & Dimensions

## 2.4 RS-485 INSTALLATION

The JDx sensor is intended to be installed on an RS-485 bus. Jewell Instruments recommends that the installer review appropriate guidelines for RS-485 bus installations, particularly when multiple devices are installed on a common bus. The JDx sensor contains a 120  $\Omega$  termination resistor at the PCB level near the internal RS-485 transceiver. Enabling and disabling of the termination resistor is accomplished through Modbus-RTU commands.

## 2.5 GALVANIC ISOLATION & JDx GROUNDING & TRANSIENT PROTECTION

Both the power connections and RS-485 connections are galvanically isolated from the internal sensor. The power inputs are rated to 2kVAC whereas the RS-485 connections are rated to 5kVAC isolation. This ensures that common mode noise which may develop over long cable lengths will not adversely affect the sensitive measurements of the device. The power and RS-485 signals are separated from each other, but their separation is not rated.

The shell of the DB9, M12, and C06 connectors have a low impedance path to the baseplate of the sensor. This ensures that cable noise is diverted through the aluminum enclosure and into the mounting structure of the sensor.

To further protect the JDx sensor in harsh electrical environments, all inputs and outputs have bi-directional transorbs and gas discharge tubes to dissipate any transient impulses. Further, all inputs and outputs have differential transorbs to dissipate transients between power and communication signals.



### 3.0 COMMUNICATION

The JDx communicates using the Modbus RTU protocol as a Modbus slave device. By default, the slave address of the unit will be the last (2) digits of the factory assigned serial number. This can be reassigned by the user (valid slave address values are from 1 – 247). The communication port hardware is configured as 2-wire, half duplex RS-485.

The JDI-xxxx communication port supports the following settings: (which can be changed by the user).

**Baud rates:** 2400, 4800, 9600, 19200, 38400, 57600, 115200, 230400, 460800, 500000 bps

**Parity:** Even, Odd, None (“None” forces the port to use 2 stop bits per Modbus RTU protocol definition)

**Stop bits:** 1, 2

**THE DEFAULT FACTORY CONFIGURATION OF THE UNIT WHEN SHIPPED IS 19200 bps, Even Parity, 1 Stop bit, and the Slave / Node Address will match the last two digits of the unit serial number. If the last two digits of the serial number are “00”, then the Slave / Node Address will be set = “100”.**

**Modbus RTU Turn-around or Inter-Frame delay** time intervals have been defined as (3.5) character durations (which is 38.5 bit widths at 11 bits/character). Strictly by Modbus-RTU standard, the timing interval was originally defined in this manner for lower baud rate applications (below 19200 bps). At signal rates above 19200 bps, the Modbus-RTU standard defines a minimum constant turn-around delay interval of 1.75 milliseconds. With modern equipment, the full (3.5 character) time intervals are often much longer than necessary, and can greatly reduce the effective throughput and scan rate of a system. Consequently, many devices do not implement the full, strict time interval at very low baud rates. The turn-around delay is provided so that Master devices (and Gateway Devices) have some time to switch their half-duplex RS-485 ports from transmit (after their last stop bit of its message frame has been transmitted) to receive (to catch the first start bit of the Slave’s response message frame). A similar amount of time is also used as an interframe-gap timing interval, which can be used and enforced by Master devices. The JDI firmware has been designed to simplify compatibility with other Modbus-RTU devices. It does not strictly enforce interframe gap timing on accepting/receiving messages, but it will delay sending out any response to a received message according to the interval timing table shown in this section.

The Modbus-Compliant JDI firmware implements a baud-rate dependent turn-around delay interval to delay the transmission of a response message frame to a received query or command from a master device. The delay intervals are implemented according to the following table. These baud rates and intervals have been compliance tested by Modbus.org and found to be compatible with a range of Modbus-RTU communication devices and systems.

Modbus-RTU Baud Rate	Turn-Around Delay Interval
2400 bps	12 milliseconds
4800 bps	8 milliseconds
9600 bps	4 milliseconds
19200 bps	2 milliseconds
38400 bps and all higher rates	1.75 milliseconds

Table 7: Modbus Timing

### 3.1 MODBUS INTERFACE – GENERAL OVERVIEW

The JDX Modbus RTU protocol provides a set of registers to:

1. Identify the device and its revision information
2. Read sensor diagnostic status and sensor data in Most Significant Byte/Most Significant Word (“Big Endian”) format
3. Read sensor diagnostic status and sensor data in Least Significant Byte/Least Significant Word (“Little Endian”) format
4. Configure the sensor’s bandwidth, update rate, and averaging functions.
5. Configure the sensor’s communication parameters.
6. A coil bank register to select communication line termination, initiate reconfiguration of the sensor, reset the sequence counter and internal FIFO, or reset/reboot the sensor.

The map of these registers, their Modbus RTU Slave addresses and functional groupings are summarized in the following table.

**Table 8: Modbus Register Map**

Register Bank Name	Address	Register #	Name	Type	Size (bytes)	Order	R/W	Supported Function Codes	
Config Coils	0		RS485 Termination	bit	1	N/A	R/W	FC01,FC05	
	1		Comm Reset	bit	1	N/A	R/W	FC01,FC05	
	2		Act On Config Data	bit	1	N/A	R/W	FC01,FC05	
	5		Sequence Counter Reset	bit	1	N/A	R/W	FC01,FC05	
	7		Sensor Reset Control	bit	1	N/A	R/W	FC01,FC05	
	8-15		Reserved						
Register Bank Name	Address	Register #	Name	Type	Size (bytes)	Order	R/W	Supported Function Codes	
Identity Bank	0	1	Device Model (ASCII)	ASCII/2-chars per register	16	char string"JDI-200-1.00", msb	R	FC03,FC04	
	8	9	Manufacture Date	ASCII/2-chars per register	12-padded(16)	"2018MAR31",msbf	R	FC03,FC04	
	16	17	Calibration Date	ASCII/2-chars per register	12-padded(16)	"2018MAR31", msbf	R	FC03,FC04	
	24	25	Serial Number	ASCII/2-chars per register	16	"01234567890ABCD", msbf	R	FC03,FC04	
	32	33	Output "Style"	UINT16	2	UINT16, msbf	R	FC03,FC04	
	33	34	Firmware Revision, Major	UINT16	2	UINT16, msbf	R	FC03,FC04	
	34	35	Firmware Revision, Minor	UINT16	2	UINT16, msbf	R	FC03,FC04	
	35	36	Firmware Revision, Dev	UINT16	2	UINT16, msbf	R	FC03,FC04	
	36-45	37	Customer Programmable Field	ASCII/2-chars per register	20	string"NW_Column_SN",n	R	FC03,FC04	
Register Bank Name	Address	Register #	Name	Type	Size (bytes)	Order	R/W	Supported Function Codes	
Data Bank 1	100	101	FAULT/STATUS REG0	UINT16	2	lsrf	R	FC03,FC04	
	little endian	101	102	FAULT/STATUS REG1	UINT16	2	lsrf	R	FC03,FC04
		102	103	Sequence Counter	Int32	4	lsrf	R	FC03,FC04
		104	105	Sensor Temperature	Float (REAL32)	4	lsrf	R	FC03,FC04
		106	107	X-Axis Inclination	Float (REAL32)	4	lsrf	R	FC03,FC04
		108	109	Y-Axis Inclination	Float (REAL32)	4	lsrf	R	FC03,FC04
		110-117		Reserved					
Register Bank Name	Address	Register #	Name	Type	Size (bytes)	Order	R/W	Supported Function Codes	
Data Bank 2	200	201	FAULT/STATUS REG0	UINT16	2	msrf	R	FC03,FC04	
	big endian	201	202	FAULT/STATUS REG1	UINT16	2	msrf	R	FC03,FC04
		202	203	Sequence Counter	Int32	4	msrf	R	FC03,FC04
		204	205	Sensor Temperature	Float (REAL32)	4	msrf	R	FC03,FC04
		206	207	X-Axis Inclination	Float (REAL32)	4	msrf	R	FC03,FC04
		208	209	Y-Axis Inclination	Float (REAL32)	4	msrf	R	FC03,FC04
		210-217		Reserved					

Register Bank Name	Address	Register #	Name	Type	Size (bytes)	Order	R/W	Supported Function Codes
Config Bank	300	301	Samples per Measurement	UINT16	2	msbf	R/W	FC03,FC04/FC06,FC16
	301	302	Sampling Rate/LPF	UINT16	2	msbf	R/W	FC03,FC04/FC06,FC16
	302	311	Custom Programmable Field	ASCII/2chars per register, 10 Regs	20	msbf	R/W	FC03,FC04/FC06,FC16
	312	313	CMD_REG0	ASCII/2 chars/UINT16	2	msbf	R/W	FC03,FC04/FC06,FC16
	313	314	CMD_REG1	ASCII/2 chars/UINT16	2	msbf	R/W	FC03,FC04/FC06,FC16
	314	315	CMD_REG2	ASCII/2 chars/UINT16	2	msbf	R/W	FC03,FC04/FC06,FC16
	315	316	CMD_REG3	ASCII/2 chars/UINT16	2	msbf	R/W	FC03,FC04/FC06,FC16
	316	317	CMD_REG4	ASCII/2 chars/UINT16	2	msbf	R/W	FC03,FC04/FC06,FC16
	317	318	CMD_REG5	ASCII/2 chars/UINT16	2	msbf	R/W	FC03,FC04/FC06,FC16
	318	319	CMD_REG6	ASCII/2 chars/UINT16	2	msbf	R/W	FC03,FC04/FC06,FC16
	319	320	CMD_REG7	ASCII/2 chars/UINT16	2	msbf	R/W	FC03,FC04/FC06,FC16
	320	321	Moving Avg Filter Enable	UINT16	2	msbf	R/W	FC03,FC04/FC06,FC16
	321	322	Moving Avg Filter Length	UINT16	2	msbf	R/W	FC03,FC04/FC06,FC16
	322	323	Relative Zero X	Float (REAL32)	4	msrf	R	FC03,FC04
324	325	Relative Zero Y	Float (REAL32)	4	msrf	R	FC03,FC04	
326	327	Relative Zero Z	Float (REAL32)	4	msrf	R	FC03,FC04	
Register Bank Name	Address	Register #	Name	Type	Size (bytes)	Order	R/W	Supported Function Codes
Comms Bank	500	501	Network Address	UINT16	2	UINT16, msbf	R/W	FC03,FC04/FC06,FC16
	501	502	Parity	UINT16	2	UINT16, msbf	R/W	FC03,FC04/FC06,FC16
	502	503	StopBits	UINT16	2	UINT16, msbf	R/W	FC03,FC04/FC06,FC16
	503-504	504	BaudRate	UINT32	4	UINT16, msbf	R/W	FC03,FC04/FC06,FC16
Register Bank Name	Address	Register #	Name	Type	Size (bytes)	Order	R/W	Supported Function Codes
Fault/Status Bank	700	701	Diagnostic / Fault Status Reg0	UINT16	2	UINT16, msbf	R	FC03,FC04 , FC08,SFC02
				Bit 0: COM Port Parity Error				0x0001
				Bit 1: COM Port Framing and/or Overrun Error				0x0002
				Bit 2: Measured temperature is above the maximum calibrated range of the device				0x0004
				Bit 3: Measured temperature is below the minimum calibrated range of the device				0x0008
				Bit 4: Measured value (angle, acceleration) is above the maximum calibrated range of the device - depends on LUT calibration settings				0x0010
				Bit 5: Measured value (angle, acceleration) is below the minimum calibrated range of the device -- depends on LUT calibration settings				0x0020
				Bit 6: Failure to load Calibration (LUT) data				0x0040
				Bit 7: Failure to load Nonvolatile Settings data from EEPROM chip - using default configuration setting				0x0080
				Bit 8: Failure to ID, configure, or self test Accelerometer IC				0x0100
				Bit 11: Test Mode Active in Accelerometer				0x0800
				Bit 12: The system has been reset due to a software fault/Watchdog timeout error - operational stability may or may not be maintained.				0x1000
				Bit 13: BrownOut/Power Cycle Detect. System has been reset from power brownout condition or has just been powered up				0x2000
		701	702	Reserved	UINT16	2	UINT16, msbf	R

Details about each register section are covered in the following sections.

### 3.2 IDENTITY REGISTER BANK

The Identity Register Bank is comprised of (46) 16-bit registers which are READ ONLY. The information is presented in the following order. The registers may be read individually, or in multiples, using Modbus Function Code 03 or 04. The order of character string information is presented in ASCII string format, so that the characters are transmitted in sequential order. The data is transmitted according to Modbus protocol with the Most Significant Byte/Most Significant Word first.

Register Bank Name	Address	Register #	Name	Type	Size (bytes)	Order	R/W	Supported Function Codes
Identity Bank	0	1	Device Model (ASCII)	ASCII/2-chars per register	16	char string"JDI-200-1.00", msbf	R	FC03,FC04
	8	9	Manufacture Date	ASCII/2-chars per register	12-padded(16)	"2018MAR31",msbf	R	FC03,FC04
	16	17	Calibration Date	ASCII/2-chars per register	12-padded(16)	"2018MAR31", msbf	R	FC03,FC04
	24	25	Serial Number	ASCII/2-chars per register	16	"01234567890ABCD", msbf	R	FC03,FC04
	32	33	Output "Style"	UINT16	2	UInt16, msbf	R	FC03,FC04
	33	34	Firmware Revision, Major	UINT16	2	UINT16, msbf	R	FC03,FC04
	34	35	Firmware Revision, Minor	UINT16	2	UINT16, msbf	R	FC03,FC04
	35	36	Firmware Revision, Dev	UINT16	2	UINT16, msbf	R	FC03,FC04
	36-45	37	Customer Programmable Field	ASCII/2-chars per register	20	char string"NW_Column_SN",msbf	R	FC03,FC04

The "Customer Programmable Field" is a read only character field in this register bank and provides a user-customizable section to facilitate identifying the device. The contents of this field are changed through the configuration register bank, which is described later.

**Example:** Read the (46) 16-bit register contents of IDENTITY Register Bank

**Master Sends:** 53 03 00 00 00 2E C8 64

**Sensor Responds:** 53 03 5C 4A 44 49 2D 32 30 30 2D 33 30 2E 30 20 00 00 00 32 30 31 38 6A 75 6E 31 34  
00 00 00 00 00 00 00 32 30 32 31 6A 75 6E 31 34 00 00 00 00 00 00 41 42 43 44 45 46  
47 31 00 00 00 00 00 00 00 14 5E 00 02 00 03 00 0F 43 55 53 54 4F 4D 45 52 20 49 4E  
46 4F 20 48 45 52 45 21 00 1F 14

The sensor's data maps into register information as shown:

String16	0	4A44 492D	JDI-200-30.0	Model
String16	8	3230 3138	2018jun14	Mfr Date
String16	16	3230 3231	2021jun14	Cal Date
String16	24	4142 4344	ABCDEFG1	Serial Number
16bit UINT	32	145E	5214	Style
16bit INT	33	0002	2	FW Major
16bit INT	34	0003	3	FW Minor
16bit INT	35	000F	15	FW Dev
String16	36	4355 5354	CUSTOMER INFO HE	CustomField
32b String4	44	5245 2100	RE!	CF_Ext

**Errors:** Attempting to read outside the defined register address range will result in the sensor returning an Address Error Exception response. Attempting to write into this address range will result in the sensor returning an Invalid Function Code Exception response.

### 3.3 DATA1 REGISTER BANK

The DATA1 Register Bank is comprised of (10) 16-bit registers which are READ ONLY. The information is presented in the following order. The registers may be read individually, or multiples, using Modbus Function Codes 03 or 04. The order of bytes and words when reading from this register bank are transmitted to the Master in “Little Endian” format (Least Significant Byte/Least Significant Word) for user convenience, if desired. The exact same data is also available in DATA2 Register Bank, but information read from that register bank is transmitted to the Master in “Big Endian” format (Most Significant Byte/Most Significant Word first) for user convenience. In the case of 32-bit Integer or 32-bit floating point values, if we consider the (4) bytes which comprise those values to be in the order of “ABCD” for Big Endian format, they will be in order of “DCBA” (byte position reversed) in Little Endian format.

Register Bank Name	Address	Register #	Name	Type	Size (bytes)	Order	R/W	Supported Function Codes
Data Bank 1	100	101	FAULT/STATUS REG0	UINT16	2	lsrf	R	FC03,FC04
little endian	101	102	FAULT/STATUS REG1	UINT16	2	lsrf	R	FC03,FC04
	102	103	Sequence Counter	Int32	4	lsrf	R	FC03,FC04
	104	105	Sensor Temperature	Float (REAL32)	4	lsrf	R	FC03,FC04
	106	107	X-Axis Inclination	Float (REAL32)	4	lsrf	R	FC03,FC04
	108	109	Y-Axis Inclination	Float (REAL32)	4	lsrf	R	FC03,FC04
	110-117		Reserved					

#### Register Descriptions:

- 100 Fault/Status Register0 contains fault bits for a variety of fault or diagnostic conditions in the sensor. Refer to the section “FAULT/STATUS Register Bank” for detailed explanation.
- 101 Fault/Status Register1 contains fault bits for a variety of fault or diagnostic conditions in the sensor. Refer to the section “FAULT/STATUS Register Bank” for detailed explanation.
- 102 Sequence Counter is a 32 bit value ((2) 16-bit registers) which is incremented for each new measurement performed by the sensor. At present, it will simply rollover back to “0” when it reaches a value of 65535 and continue incrementing. The sensor’s measurement rate is controlled by several parameters, and this rate is independent of the rate of Modbus scanning. Therefore, this counter may appear to increment by more than “1” with successive Modbus scans. By simply looking for changes in value of the counter, the Master can verify that the sensor is operating and providing new measurement data.
- 104 Sensor Temperature is an IEEE-754 32-bit floating point value ((2) 16-bit registers) representing the measured internal temperature of the sensor (in degrees Celsius).
- 106 X-Axis Inclination is an IEEE-754 32-bit floating point value ((2) 16-bit registers) representing the temperature compensated inclination measurement of the sensor’s X axis.
- 108 Y-Axis Inclination is an IEEE-754 32-bit floating point value ((2) 16-bit registers) representing the temperature compensated inclination measurement of the sensor’s Y axis, if the unit is a dual axis unit. Otherwise it will read “0”.

**Example:** Read the entire (10) 16-bit register contents of DATA1 Register Bank (Little-Endian)

**Master Sends:** 53 04 00 64 00 0A 3C 60

**Sensor Responds:** 53 04 14 00 00 00 00 10 49 00 00 F5 27 00 42 9D 82 BA 40 C3 26 54 41 F5 F0

The sensor's data maps into register information as shown:

16bit UINT	100	0000	0	STATUS0
16bit UINT	101	0000	0	STATUS1
32bit UINT	102	0000 4910	18704	Sequence
32bit Float	104	4200 27F5	32.0390205	Temp
32bit Float	106	40BA 829D	5.82844400	X Angle
32bit Float	108	4154 26C3	13.2594633	Y Angle

**Errors:** Attempting to read outside the defined register address range will result in the sensor returning an Address Error Exception response. Attempting to write into this address range will result in the sensor returning an Invalid Function Code Exception response.

### 3.4 DATA2 REGISTER BANK

The DATA2 Register Bank is comprised of (10) 16-bit registers which are READ ONLY. The information is presented in the following order. The registers may be read individually, or multiples, using Modbus Function Codes 03 or 04. The order of bytes and words when reading from this register bank are transmitted to the Master in “Big Endian” format (Most Significant Byte/Most Significant Word) for user convenience, if desired. The exact same data is also available in DATA1 Register Bank, but information read from that register bank is transmitted to the Master in “Little Endian” format (Least Significant Byte/Least Significant Word first) for user convenience. In the case of 32-bit Integer or 32-bit floating point values, if we consider the (4) bytes which comprise those values to be in the order of “ABCD” for Big Endian format, they will be in order of “DCBA” (byte position reversed) in Little Endian format.

Register Bank Name	Address	Register #	Name	Type	Size (bytes)	Order	R/W	Supported Function Codes
Data Bank 2	200	201	FAULT/STATUS REG0	UINT16	2	msrf	R	FC03,FC04
big endian	201	202	FAULT/STATUS REG1	UINT16	2	msrf	R	FC03,FC04
	202	203	Sequence Counter	Int32	4	msrf	R	FC03,FC04
	204	205	Sensor Temperature	Float (REAL32)	4	msrf	R	FC03,FC04
	206	207	X-Axis Inclination	Float (REAL32)	4	msrf	R	FC03,FC04
	208	209	Y-Axis Inclination	Float (REAL32)	4	msrf	R	FC03,FC04
	210-217		Reserved					

#### Register Descriptions:

- 200 Fault/Status Register0 contains fault bits for a variety of fault or diagnostic conditions in the sensor. Refer to the section “FAULT/STATUS Register Bank” for detailed explanation.
- 201 Fault/Status Register1 contains fault bits for a variety of fault or diagnostic conditions in the sensor. Refer to the section “FAULT/STATUS Register Bank” for detailed explanation.
- 202 Sequence Counter is a 32 bit value ((2) 16-bit registers) which is incremented for each new measurement performed by the sensor. At present, it will simply rollover back to “0” when it reaches a value of 65535 and continue incrementing. The sensor’s measurement rate is controlled by several parameters, and this rate is independent of the rate of Modbus scanning. Therefore, this counter may appear to increment by more than “1” with successive Modbus scans. By simply looking for changes in value of the counter, the Master can verify that the sensor is operating and providing new measurement data.
- 204 Sensor Temperature is an IEEE-754 32-bit floating point value ((2) 16-bit registers) representing the measured internal temperature of the sensor (in degrees Celcius).
- 206 X-Axis Inclination is an IEEE-754 32-bit floating point value ((2) 16-bit registers) representing the temperature compensated inclination measurement of the sensor’s X axis.
- 208 Y-Axis Inclination is an IEEE-754 32-bit floating point value ((2) 16-bit registers) representing the temperature compensated inclination measurement of the sensor’s Y axis, if the unit is a dual axis unit. Otherwise it will read “0”.

**Example:** Read the entire (10) 16-bit register contents of DATA2 Register Bank (Big-Endian)

**Master Sends:** 53 03 00 C8 00 0A 49 81

**Sensor Responds:** 53 03 14 00 00 00 00 00 51 D3 41 FF 98 0B 40 BB 32 6B 41 54 2C 21 35 7A

The sensor's data maps into register information as shown:

16bit UINT	200	0000	0	STATUS0
16bit UINT	201	0000	0	STATUS1
32bit UINT	202	0000 51D3	20947	Sequence
32bit Float	204	41FF 980B	31.9492397	Temp
32bit Float	206	40BB 326B	5.84990454	X Angle
32bit Float	208	4154 2C21	13.2607737	Y Angle

**Errors:** Attempting to read outside the defined register address range will result in the sensor returning an Address Error Exception response. Attempting to write into this address range will result in the sensor returning an Invalid Function Code Exception response.



### 3.5 CONFIGURATION REGISTER BANK

The Configuration Register Bank is comprised of (22) 16-bit registers which are READ/WRITE. The information is presented in the following order. The registers may be read or written individually, or multiples, using Modbus Function Codes 03, 04, 06, or 16. The order of bytes and words when reading or writing to this register bank is transmitted in “Big Endian” format (Most Significant Byte/Most Significant Word). In the case of 32-bit Integer or 32-bit floating point values, if we consider the (4) bytes which comprise those values to be in the order of “ABCD” for Big Endian format, they will be in order of “DCBA” (byte position reversed) in Little Endian format.

Register Bank Name	Address	Register #	Name	Type	Size (bytes)	Order	R/W	Supported Function Codes
Config Bank	300	301	Samples per Measurement	UINT16	2	msbf	R/W	FC03,FC04/FC06,FC16
	301	302	Sampling Rate/LPF	UINT16	2	msbf	R/W	FC03,FC04/FC06,FC16
	302	311	Custom Programmable Field	ASCII/2chars per register, 10 Regs	20	msbf	R/W	FC03,FC04/FC06,FC16
	312	313	CMD_REG0	ASCII/2 chars/UINT16	2	msbf	R/W	FC03,FC04/FC06,FC16
	313	314	CMD_REG1	ASCII/2 chars/UINT16	2	msbf	R/W	FC03,FC04/FC06,FC16
	314	315	CMD_REG2	ASCII/2 chars/UINT16	2	msbf	R/W	FC03,FC04/FC06,FC16
	315	316	CMD_REG3	ASCII/2 chars/UINT16	2	msbf	R/W	FC03,FC04/FC06,FC16
	316	317	CMD_REG4	ASCII/2 chars/UINT16	2	msbf	R/W	FC03,FC04/FC06,FC16
	317	318	CMD_REG5	ASCII/2 chars/UINT16	2	msbf	R/W	FC03,FC04/FC06,FC16
	318	319	CMD_REG6	ASCII/2 chars/UINT16	2	msbf	R/W	FC03,FC04/FC06,FC16
	319	320	CMD_REG7	ASCII/2 chars/UINT16	2	msbf	R/W	FC03,FC04/FC06,FC16
	320	321	Moving Avg Filter Enable	UINT16	2	msbf	R/W	FC03,FC04/FC06,FC16
	321	322	Moving Avg Filter Length	UINT16	2	msbf	R/W	FC03,FC04/FC06,FC16
	322	323	Relative Zero X	Float (REAL32)	4	msrf	R	FC03,FC04
	324	325	Relative Zero Y	Float (REAL32)	4	msrf	R	FC03,FC04
	326	327	Relative Zero Z	Float (REAL32)	4	msrf	R	FC03,FC04

The functionality of this register bank is multi-purpose. For registers #300 – 311, the present value of the sensor’s parameters are displayed and may be read by the master. Also, the master can write to these registers to change them according to the application demands. Registers #312-319 are used in a special way to command certain operations in the sensor. To avoid potential race-conditions or conflicting operations, none of the parameter changes in this register bank will be acted upon until the master issues a write to turn on sensor coil #2, “Act On Config Data”.

When the master issues a write to turn on the “Act On Config Data” Coil, the sensor will issue a response to the write command, and a short while later (approximately 1 second) it will update its internal parameters from the registers #300 – 311, 320, 321, or perform the command sequence defined in the command registers #312-319.

NOTE: While the sensor is reconfiguring its parameters in response to an “Act On Config Data” coil write, output data and reported measurements and command responses may be unstable. It is strongly recommended to pause sending commands to the sensor for approximately 1.5 to 2 seconds to allow it to complete its reinitialization of all configuration register bank parameters. After this interval, it should be fine to resume sending it commands or poll for data.

#### Register Descriptions:

300 This register specifies the number of samples from the internal sensing device to average per each update or measurement that is output by the product. The allowable range of values for this register is 1 through 4096. Attempting to write values outside of this range will result in the value being set to the maximum or minimum number in the range. The microcontroller in the device will collect the specified number of data samples from the internal accelerometer and average them for each published update or measurement. The Sampling Rate parameter (in Register 301) in combination with this parameter affects how frequently new measurements are made available and updated in the Modbus interface. A value of “1” provides no averaging of internal sample

data. In effect, it decimates or reduces the configured Output Date Rate (ODR) while averaging if this register value is not = “1”.

301 The Sampling Rate register has only a specific range of values which it will accept. These values specify the sampling rate, output data rate and low-pass filter bandwidth of the sensing device. The supported values and their corresponding behavior are shown in the following table:

Register 301 Value	Sampling & Output Data Rate (Internal)	LowPass Filter Bandwidth
0	4000 Hz	1000 Hz
1	2000 Hz	500 Hz
2	1000Hz	250 Hz
3	500 Hz	125 Hz
4	250 Hz	62.5 Hz
5	125 Hz	31.25 Hz
6	62.5 Hz	15.625 Hz
7	31.25 Hz	7.813 Hz
8	15.625	3.906 Hz
9	7.813	1.953 Hz
10	3.906	0.977 Hz

The selection of values for registers 300 and 301 should be considered carefully with respect to the application measurement requirements. Together, they also impact the rate at which new measurements are made available at the Modbus interface. For example, selecting Register 300 = 4096 (4K samples) along with selecting Register 301= 10 will yield a new measurement update rate of  $(4096/3.906) = 1048$  seconds, or approximately 17.5 minutes. Selecting Register 301=0, and Register 300=100 yields a measurement update rate of  $(100/4000)=0.025$  seconds, which may be too rapid for some communication system configurations to maintain. However, in various applications it may still be desired to have the more rapid measurement response even if the communications configuration cannot transfer every new update before more new updates occur. Noting the values of the Sequence Counter Register on successive Modbus scans will provide insight as to how many measurement updates are occurring between bus scans.

302-311 Registers 302-311 provide a means where the customer may, if desired, write a custom character string to provide application-specific identification information. Although the (10) registers provide a theoretical storage space for (20) ASCII characters, the implementation is presently limited to (19) printable characters, with a NULL (0x00) termination character.

312 - 319 Registers #312 – 319 correspond to Command Registers CMD\_REG0 – CMD\_REG7 which provide a simple command interface by which certain operational functions can be invoked in the sensor. The currently supported command syntaxes are defined in the following “User Command” Sections.

320 Register #320 is used to ENABLE or DISABLE the Moving Average, or “Boxcar” Filter. The Moving Average Filter maintains an internal sample buffer length of the most recent “N” samples, where N= <the value located in register #321>. The moving average filter outputs data at the configured Output Date Rate (ODR) in register #301, it doesn’t reduce the effective ODR, like programming a value other than “1” in register #300 will. To ENABLE the Moving Average filter, WRITE the value “1” (0x0001) to this register. Writing “0” or any other value will result in turning the register #320 contents = “0” and will DISABLE the filter when the Act On Config Data Coil is set.

The Moving Average Filter controlled by registers #320, #321 emits the moving averaged data for each axis at the configured ODR, removing the oldest sample point in the process of adding the latest sample point. As such, it exhibits linear phase response. When enabling the Moving Average Filter, it should be noted that until it has accumulated a buffer’s worth of “N” samples, it will emit the unfiltered/non-averaged data, rather than incorrect data scaled by 1/N. One can reference the sequence number count in Register 102 or 202 to identify at what point the output sample data is being filtered/averaged after initially enabling the Moving Average Filter. For example, if the MA Filter is enabled by default, and the MA Filter Length (N) = 20 in register #321, then the first datapoints emitted with sequence number values < 20 will be “unfiltered” or “raw”. Datapoints with sequence numbers > 20 will be filtered or averaged, as expected. As with all registers in this register bank, writing the register value does not immediately take effect on the sensor’s function. When the master issues a write to turn on the Act On Config Data Coil, the sensor will issue a response to the write command, and a short while later (approximately 1 second) it will update its internal parameters from the current register values.

321 Register #321 is used to set the buffer sample length, or window size (N) for the Moving Average, or “Boxcar” Filter. When Enabled, the Moving Average Filter maintains an internal sample buffer length of the most recent “N” samples, where N= <the value in register #321>. The allowable range of values to be written in Register #321 are 3 – 32. Using lower numbers provides less smoothing/filtering of noise but reduces the phase delay from input stimulus. Using higher values provide a lot of smoothing, but with considerably more phase delay and step attenuation. As with all registers in this register bank, writing the register value does not immediately take effect on the sensor’s function. When the master issues a write to turn on the Act On Config Data Coil, the sensor will issue a response to the write command, and a short while later (approximately 1 second) it will update its internal parameters from the Configuration Register Bank’s register values. If an out of range value had been written to this register before the Act On Config Data coil is set, the register will revert back to it previous valid value, or default to a buffer length value of “4”.

322-323 Register Pair 322, 323 are used to indicate the “User Relative Zero” value for the X-Axis. This register is Read Only. The value is an IEEE-754 format 32-bit floating point value stored in “big-endian” format, like the 32-bit floating point values stored in Data Register Bank 200. The value in the register reflects the offset value which is subtracted from X axis output measurements. It functions as an Offset or “Tare” value in cases where the user wishes to compensate for a small mechanical offset or bias in the sensor’s installation. For more information on this feature, please refer to “Command Registers - User Relative Zero” section.

324-325 Register Pair 324, 325 are used to indicate the “User Relative Zero” value for the Y-Axis (if configured). This register is Read Only. The value is an IEEE-754 format 32-bit floating point value stored in “big-endian” format, like the 32-bit floating point values stored in Data Register Bank 200. The value in the register reflects the offset value which is subtracted from Y axis output measurements. It functions as an Offset or “Tare” value in cases where the user wishes to compensate for a small mechanical offset or bias in the sensor’s installation. For more information on this feature, please refer to “Command Registers - User Relative Zero” sections.

326-327 Register Pair 326, 327 are used to indicate the “User Relative Zero” value for the Z-Axis (if configured). This register is Read Only. The value is an IEEE-754 format 32-bit floating point value stored in “big-endian” format, like the 32-bit floating point values stored in Data Register Bank 200. The value in the register reflects the offset value which is subtracted from Y axis output measurements. It functions as an Offset or “Tare” value in cases where the user wishes to compensate for a small mechanical offset or bias in the sensor’s installation. For more information on this feature, please refer to “Command Registers - User Relative Zero” sections.

**3.5.1 COMMAND REGISTERS – NONVOLATILE SAVE COMMAND**

Address	Register Name	Contents for Command Function
312	CMD_REG0	“NS” ASCII characters (0x4E53)
313	CMD_REG1	XXXX (Don’t care)
314	CMD_REG2	XXXX (Don’t care)
315	CMD_REG3	XXXX (Don’t care)
316	CMD_REG4	XXXX (Don’t care)
317	CMD_REG5	XXXX (Don’t care)
318	CMD_REG6	XXXX (Don’t care)
319	CMD_REG7	XXXX (Don’t care)

Writing the values in the CMD\_REGx registers as shown above, and then writing a “1” to the Act On Config Data coil (Coil 2) will cause the sensor to perform a “Save” of all its current parameter settings to internal nonvolatile memory. The device will automatically restore the updated parameter values upon a power cycle or device reset. These include all communications port settings, custom fields, sampling, filtering, averaging, and user-relative zero parameters.

**3.5.2 COMMAND REGISTERS – NONVOLATILE RESTORE COMMAND**

Address	Register Name	Contents for Command Function
312	CMD_REG0	“NR” ASCII characters (0x4E52)
313	CMD_REG1	XXXX (Don’t care)
314	CMD_REG2	XXXX (Don’t care)
315	CMD_REG3	XXXX (Don’t care)
316	CMD_REG4	XXXX (Don’t care)
317	CMD_REG5	XXXX (Don’t care)
318	CMD_REG6	XXXX (Don’t care)
319	CMD_REG7	XXXX (Don’t care)

Writing the values in the CMD\_REGx registers as shown above, and then writing a “1” to the Act On Config Data coil (Coil 2) will cause the sensor to perform a “Restore” or reload of its last saved parameter configuration from nonvolatile EEPROM. This may be useful when experimenting with parameter changes and needing to revert to the last saved sensor settings. These include all communications port settings, custom fields, sampling, filtering, averaging, and user-relative zero parameters.

### 3.5.3 COMMAND REGISTERS – TEST MODE OFF COMMAND

Address	Register Name	Contents for Command Function
312	CMD_REG0	“T0” ASCII characters (0x5430)
313	CMD_REG1	XXXX (Don’t care)
314	CMD_REG2	XXXX (Don’t care)
315	CMD_REG3	XXXX (Don’t care)
316	CMD_REG4	XXXX (Don’t care)
317	CMD_REG5	XXXX (Don’t care)
318	CMD_REG6	XXXX (Don’t care)
319	CMD_REG7	XXXX (Don’t care)

Writing the values in the CMD\_REGx registers as shown above, and then writing a “1” to the Act On Config Data coil (Coil 2) will cause the sensor to exit “test mode” (if it had been previously initiated).

### 3.5.4 COMMAND REGISTERS – TEST MODE ON MODE 1 COMMAND

Address	Register Name	Contents for Command Function
312	CMD_REG0	“T1” ASCII characters (0x5431)
313	CMD_REG1	XXXX (Don’t care)
314	CMD_REG2	XXXX (Don’t care)
315	CMD_REG3	XXXX (Don’t care)
316	CMD_REG4	XXXX (Don’t care)
317	CMD_REG5	XXXX (Don’t care)
318	CMD_REG6	XXXX (Don’t care)
319	CMD_REG7	XXXX (Don’t care)

Writing the values in the CMD\_REGx registers as shown above, and then writing a “1” to the Act On Config Data coil (Coil 2) will cause the sensor’s microcontroller to place the internal accelerometer in Test Mode 1, which activates an internal test mode within the instrument’s accelerometer and rejects external forces applied to the instrument. When this mode is active the SelfTest Status bit will be set in the Status/Fault Register0. All measurements read during Self Test Mode 1 are baseline, isolated accelerometer readings which can be used to compare against Self Test Mode 2 readings.

### 3.5.5 COMMAND REGISTERS – TEST MODE ON MODE 2 COMMAND

Address	Register Name	Contents for Command Function
312	CMD_REG0	"T2" ASCII characters (0x5432)
313	CMD_REG1	XXXX (Don't care)
314	CMD_REG2	XXXX (Don't care)
315	CMD_REG3	XXXX (Don't care)
316	CMD_REG4	XXXX (Don't care)
317	CMD_REG5	XXXX (Don't care)
318	CMD_REG6	XXXX (Don't care)
319	CMD_REG7	XXXX (Don't care)

Writing the values in the CMD\_REGx registers as shown above, and then writing a "1" to the Act On Config Data coil (Coil 2) will cause the sensor's microcontroller to place the internal accelerometer in Test Mode 2, which applies an internal reference test force within the accelerometer and rejects external forces applied to the instrument. When this mode is active the SelfTest Status bit will be set in the Status/Fault Register0. All measurements read during Self Test Mode 2 are correspond to an internally generated test force equivalent to approximately 0.5g acceleration (approximately 30 degree angular displacement). The Test mode is not intended to be used for calibration purposes but may be useful for some basic operational checks. BE SURE TO DISABLE TEST MODE (T0 command) when accurate measurements are needed.

**3.5.6 COMMAND REGISTERS – USER RELATIVE ZERO OFF COMMAND**

Address	Register Name	Contents for Command Function
312	CMD_REG0	“Z0” ASCII characters (0x5A30)
313	CMD_REG1	XXXX (Don’t care)
314	CMD_REG2	XXXX (Don’t care)
315	CMD_REG3	XXXX (Don’t care)
316	CMD_REG4	XXXX (Don’t care)
317	CMD_REG5	XXXX (Don’t care)
318	CMD_REG6	XXXX (Don’t care)
319	CMD_REG7	XXXX (Don’t care)

Writing the values in the CMD\_REGx registers as shown above, and then writing a “1” to the Act On Config Data coil (Coil 2) will cause the sensor to disable its internal relative zero offset adjustment and will clear out (zero) the working zero offset values for X, Y, Z axes. To store these changes so they are restored by default upon a sensor reset or power cycle, you must perform the “Nonvolatile Save” Command sequence described elsewhere in this chapter. The register contents for the Relative Zero Offsets for the X, Y, Z axes will be cleared as shown below in the Configuration Register Bank when the Relative Zero Offset is disabled:

String16	302	0000 0000		CustomField
32b String4	310	0000 0000		Customfield Ext
16bit UINT	312	0000	0	CMD_REG0
16bit UINT	313	0000	0	CMD_REG1
16bit UINT	314	0000	0	CMD_REG2
16bit UINT	315	0000	0	CMD_REG3
16bit UINT	316	0000	0	CMD_REG4
16bit UINT	317	0000	0	CMD_REG5
16bit UINT	318	0000	0	CMD_REG6
16bit UINT	319	0000	0	CMD_REG7
16bit UINT	320	0000	0	MAF_Enable
16bit UINT	321	0004	4	MAF_Length
32bit Float	322	0000 0000	0.00000000	RelZero_X
32bit Float	324	0000 0000	0.00000000	RelZero_Y
32bit Float	326	0000 0000	0.00000000	RelZero_Z



**3.5.7 COMMAND REGISTERS – USER RELATIVE ZERO ON COMMAND**

Address	Register Name	Contents for Command Function
312	CMD_REG0	“Z1” ASCII characters (0x5A31)
313	CMD_REG1	XXXX (Don’t care)
314	CMD_REG2	XXXX (Don’t care)
315	CMD_REG3	XXXX (Don’t care)
316	CMD_REG4	XXXX (Don’t care)
317	CMD_REG5	XXXX (Don’t care)
318	CMD_REG6	XXXX (Don’t care)
319	CMD_REG7	XXXX (Don’t care)

Writing the values in the CMD\_REGx registers as shown above, and then writing a “1” to the Act On Config Data coil (Coil 2) will cause the sensor to enable its internal relative zero offset adjustment and set its working zero offset values for X, Y, Z axes equal to the current measured data values for X,Y,Z axes (according to the sensor configuration). These axis-specific values will then automatically be subtracted from subsequent axis measurement readings before they are updated in the Register Bank 100 or Register Bank 200 measurement registers. To keep these changes so they are restored by default upon a sensor reset or power cycle, you must perform the “Nonvolatile Save” Command sequence described elsewhere in this chapter.

For example, assume that the User Relative Zero function is Off/Disabled. The Configuration Register Bank contents of Registers #322 - 327 will be as shown in the following image:

String16	302	0000 0000		CustomField
32b String4	310	0000 0000		Customfield Ext
16bit UINT	312	0000	0	CMD_REG0
16bit UINT	313	0000	0	CMD_REG1
16bit UINT	314	0000	0	CMD_REG2
16bit UINT	315	0000	0	CMD_REG3
16bit UINT	316	0000	0	CMD_REG4
16bit UINT	317	0000	0	CMD_REG5
16bit UINT	318	0000	0	CMD_REG6
16bit UINT	319	0000	0	CMD_REG7
16bit UINT	320	0000	0	MAF_Enable
16bit UINT	321	0004	4	MAF_Length
32bit Float	322	0000 0000	0.00000000	RelZero_X
32bit Float	324	0000 0000	0.00000000	RelZero_Y
32bit Float	326	0000 0000	0.00000000	RelZero_Z

Also, assume the Data Register 200 Bank Readings are currently as shown in the following image:

16bit UINT	200	2000	8192	STATUS0
16bit UINT	201	0000	0	STATUS1
32bit UINT	202	0000 19F8	6648	Sequence
32bit Float	204	41C6 9345	24.8219090	Temp
32bit Float	206	B850 6D60	-0.20354223	X Angle
32bit Float	208	3E85 21A9	0.26002243	Y Angle
32bit Float	210	0000 0000	0.00000000	Z Angle

We will write the “Z1” command into the Command Registers as indicated above and then issue a write to turn on the Act On Config Data coil (Coil 2).

Now, a subsequent reading of the Data Register 200 Bank indicates values as shown in the following image, reflecting the automatic subtraction of the “Relative Zero” offsets values, as displayed in the Configuration Register 300 Bank (following images):

16bit UINT	200	2003	8195	STATUS0
16bit UINT	201	0000	0	STATUS1
32bit UINT	202	0000 2861	10337	Sequence
32bit Float	204	41CA EB6D	25.3649540	Temp
32bit Float	206	3A8B FE00	0.00106806	X Angle
32bit Float	208	3B11 7400	0.00221944	Y Angle
32bit Float	210	0000 0000	0.00000000	Z Angle

String16	302	0000 0000		CustomField
32b String4	310	0000 0000		Customfield Ext
16bit UINT	312	0000	0	CMD_REG0
16bit UINT	313	0000	0	CMD_REG1
16bit UINT	314	0000	0	CMD_REG2
16bit UINT	315	0000	0	CMD_REG3
16bit UINT	316	0000	0	CMD_REG4
16bit UINT	317	0000	0	CMD_REG5
16bit UINT	318	0000	0	CMD_REG6
16bit UINT	319	0000	0	CMD_REG7
16bit UINT	320	0000	0	MAF_Enable
16bit UINT	321	0004	4	MAF_Length
32bit Float	322	B852 05C0	-0.20510006	RelZero_X
32bit Float	324	3E83 B0BF	0.25720784	RelZero_Y
32bit Float	326	0000 0000	0.00000000	RelZero_Z

### 3.5.8 CONFIGURATION REGISTER BANK – ADDITIONAL EXAMPLES

Some additional READ and WRITE examples of Configuration Register Bank operations are provided as an aid to the user.

**Example 1:** Read the first (20) 16-bit register contents of Configuration Register Bank (Big-Endian)

**Master Sends:** 53 03 01 2C 00 14 88 42

**Sensor Responds:** 53 03 28 00 01 00 0A 43 55 53 54 4F 4D 45 52 20 49 4E 46 4F 20 48 45 52 45 21 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 BC 0B

The sensor’s data maps into register information as shown:

▲ ▼	16bit UINT	300	0001	1	AveragesPerMeasure
▲ ▼	16bit UINT	301	000A	10	ODR
▲ ▼	String16	302	4355 5354	CUSTOMER INFO HE	CustomField
▲ ▼	32b String4	310	5245 2100	RE!	CustomFieldExt
▲ ▼	16bit UINT	312	0000	0	CMD0
▲ ▼	16bit INT	313	0000	0	CMD1
▲ ▼	16bit INT	314	0000	0	CMD2
▲ ▼	16bit INT	315	0000	0	CMD3
▲ ▼	16bit UINT	316	0000	0	CMD4
▲ ▼	16bit UINT	317	0000	0	CMD5
▲ ▼	16bit UINT	318	0000	0	CMD6
▲ ▼	16bit UINT	319	0000	0	CMD7

**Example 2:** Configure the sensor to perform 16 point sample averages per measurement (Reg300=16), with a sampling rate of 31.25 Hz (Reg301=7), and set the custom field to identify the sensor as “NW PYLON 1”.

**Master Sends:** 53 10 01 2C 00 0C 18 00 10 00 07 4E 57 20 50 59 4C 4F 4E 20 31 00 00 00 00 00 00 00 00 00 00 00 00 00 D4 97

**Sensor Responds:** 53 10 01 2C 00 0C 0D 8B

The sensor’s new data maps into register information as shown:

16bit UINT	300	0010	16	AveragesPerMeasure
16bit UINT	301	0007	7	ODR
String16	302	4E57 2050	NW PYLON 1	CustomField
32b String4	310	0000 0000		CustomFieldExt
16bit UINT	312	0000	0	CMD0
16bit INT	313	0000	0	CMD1
16bit INT	314	0000	0	CMD2
16bit INT	315	0000	0	CMD3
16bit UINT	316	0000	0	CMD4
16bit UINT	317	0000	0	CMD5
16bit UINT	318	0000	0	CMD6
16bit UINT	319	0000	0	CMD7

Note that the data is now written into the Config Reg Bank registers, but no working parameter changes will occur until the Act On Config Data coil is written to by the master. Also note that these parameter changes will not be permanent (restored on next power cycle or device reset) until a Nonvolatile Save (NS) Command is executed.

Examples of these steps follow in Example 3.

**Example 3:** Write Act On Config Data coil to initiate the sensor to change its working values.

After performing the steps in Example #2, the master must then perform a Coil Write (FC05) to the Act On Config Data coil to actually change the working values of the parameters modified in Example 2.

**Master Sends:** 53 05 00 02 FF 00 20 48

**Sensor Responds:** 53 05 00 02 FF 00 20 48

The sensor has accepted the coil write command and sent the proper response. Within approximately 1 second from this point, it will reconfigure the parameters as specified, or to the allowed limits if a value is out of range.

The sensor’s new data maps into register information as shown:

16bit UINT	300	0010	16	AveragesPerMeasure
16bit UINT	301	0007	7	ODR
String16	302	4E57 2050	NW PYLON 1	CustomField
32b String4	310	0000 0000		CustomFieldExt
16bit UINT	312	0000	0	CMD0
16bit INT	313	0000	0	CMD1
16bit INT	314	0000	0	CMD2
16bit INT	315	0000	0	CMD3
16bit UINT	316	0000	0	CMD4
16bit UINT	317	0000	0	CMD5
16bit UINT	318	0000	0	CMD6
16bit UINT	319	0000	0	CMD7

Further, the custom field in the sensor’s Identity bank is updated to reflect the change:

String16	0	4A44 492D	JDI-200-30.0	Model
String16	8	3230 3138	2018jun14	Mfr Date
String16	16	3230 3231	2021jun14	Cal Date
String16	24	4142 4344	ABCDEFG1	Serial Number
16bit UINT	32	145E	5214	Style
16bit INT	33	0002	2	FW Major
16bit INT	34	0003	3	FW Minor
16bit INT	35	000F	15	FW Dev
String16	36	4E57 2050	NW PYLON 1	CustomField
32b String4	44	0000 0000		CF_Ext

**Example 4:** Configure and activate a Nonvolatile Save (NS) Command to make the Average, Sample Rate, and Custom Name Field changes permanent so they will be restored upon future power cycles or device resets.

Write the “NS” Command to the appropriate CMD\_REG0 register:

**Master Sends:** 53 10 01 38 00 01 02 4E 53 22 D6

**Sensor Responds:** 53 10 01 38 00 01 8C 4A

The sensor’s new data maps into register information as shown:

16bit UINT	300	0010	16	AveragesPerMeasure
16bit UINT	301	0007	7	ODR
String16	302	4E57 2050	NW PYLON 1	CustomField
32b String4	310	0000 0000		CustomFieldExt
16b String2	312	4E53	NS	CMD0
16bit INT	313	0000	0	CMD1
16bit INT	314	0000	0	CMD2
16bit INT	315	0000	0	CMD3
16bit UINT	316	0000	0	CMD4
16bit UINT	317	0000	0	CMD5
16bit UINT	318	0000	0	CMD6
16bit UINT	319	0000	0	CMD7

Then, the master sends a coil write (FC05) to Coil #2 (Act On Config Data) to initiate the command processing.

**Master Sends:** 53 05 00 02 FF 00 20 48

**Sensor Responds:** 53 05 00 02 FF 00 20 48

The sensor has accepted the coil write command and sent the proper response. Within 1 second from this point, it will perform the requested save operation to nonvolatile storage of the current values of all its parameters. When the sensor processes the command, it will automatically clear the command register contents back to “0”, as shown:

16bit UINT	300	0010	16	AveragesPerMeasure
16bit UINT	301	0007	7	ODR
String16	302	4E57 2050	NW PYLON 1	CustomField
32b String4	310	0000 0000		CustomFieldExt
16b String2	312	0000		CMD0
16bit INT	313	0000	0	CMD1
16bit INT	314	0000	0	CMD2
16bit INT	315	0000	0	CMD3
16bit UINT	316	0000	0	CMD4
16bit UINT	317	0000	0	CMD5
16bit UINT	318	0000	0	CMD6
16bit UINT	319	0000	0	CMD7

**Errors:** Attempting to read outside the defined register address range will result in the sensor returning an Address Error Exception response. Attempting to write outside this address range will result in the sensor returning an Invalid Function Code Exception response.

### 3.6 COMMUNICATION REGISTER BANK

The Communication Register Bank is comprised of (5) 16-bit registers which are READ/WRITE. The registers may be read or written individually, or multiples, using Modbus Function Codes 03, 04, 06, or 16. The order of bytes and words when reading or writing to this register bank is transmitted in “Big Endian” format (Most Significant Byte/Most Significant Word). In the case of 32-bit Integer or 32-bit floating point values, if we consider the (4) bytes which comprise those values to be in the order of “ABCD” for Big Endian format, they will be in order of “DCBA” (byte position reversed) in Little Endian format.

Register Bank Name	Address	Register #	Name	Type	Size (bytes)	Order	R/W	Supported Function Codes
Comms Bank	500	501	Network Address	UINT16	2	UINT16, msbf	R/W	FC03,FC04/FC06,FC16
	501	502	Parity	UINT16	2	UINT16, msbf	R/W	FC03,FC04/FC06,FC16
	502	503	StopBits	UINT16	2	UINT16, msbf	R/W	FC03,FC04/FC06,FC16
	503-504	504	BaudRate	UINT32	4	UINT32, msbf	R/W	FC03,FC04/FC06,FC16

#### Register Descriptions:

- 500 This register specifies the network address for the sensor. The allowable range of addresses for Modbus slave devices are from 1 – 247. By default from the factory, this network address will be set to match the last (2) digits of the sensor unit’s serial number. The customer may change/reprogram the address to suit the application. If an invalid address value has been written and saved, the sensor will default to using an address value of “1”.
- 501 This register specifies the Parity setting for each byte transmitted & received on the serial communications port. The factory default setting for this register is Even parity, specified with an ASCII character ‘E’ (0x45). Other allowed settings are Odd (ASCII character ‘O’ (0x4F)) or None (ASCII character ‘N’ (0x4E)). Per Modbus specifications, if ‘N’ (None) parity is selected, the sensor will default to using (2) stop bits per character. The customer can change the parity setting to suit the application need, but should do so with care.
- 502 This register specifies the number of Stop bits setting for each byte transmitted & received on the serial communications port. The factory default setting for this register is 1 Stop bit, specified with a value of ‘1’. The only other allowed setting is a value of ‘2’, which is the value which Modbus specification expects to be employed if ‘No’ parity is being used. The customer can change the Stop bits setting to suit the application need, but should do so with care.
- 503-504 This 32-bit register specifies the baud rate setting on the serial communications port. The clock frequencies in the sensor do not allow some common baud rate values to be exact matches (for example, 115.2 kbps is actually closer to 116kb, but any difference is generally within allowable tolerances of communications systems (no more than 2% error). When the user writes a specific baud rate into register 502/503 and activates the COMM\_Reset Coil the sensor will configure the closest baud rate match that it can from the available selections. Allowed baud rate values to write include: 2400, 4800, 9600, 19200, 38400, 57600, 115200, 230400,

46800, and 500000 bps for under 2% error limits. The sensor will calculate its closest supportable matching value and update the value in the register with the configured value when the Comm Reset Coil is activated. If the register value written by the application doesn't come within 2% of one of the standard values specified, the sensor will default to using 19200 bps and update the register content accordingly.

The Modbus master device can read and write these register values, but the sensor will not take action to reconfigure its communications port setting until the master performs a Coil Write (FC05) command to Comm Reset Coil (Coil #1). An example of reconfiguring the communications port setting follows:

**Example 1:** Change the communications port baud rate to 19200 bps (or closest equivalent).

The initial state of the Communications Register Bank is as follows:

16bit UINT	500	0053	83	Address
16b String2	501	0045	E	Parity
16bit UINT	502	0001	1	Stopbits
32bit UINT	503	0001 C637	116279	Baudrate

**Master Sends:** 53 10 01 F7 00 02 04 00 00 4B 00 8C D4

**Sensor Responds:** 53 10 01 F7 00 02 FC 74

The contents of the registers are now:

16bit UINT	500	0053	83	Address
16b String2	501	0045	E	Parity
16bit UINT	502	0001	1	Stopbits
32bit UINT	503	0000 4B00	19200	Baudrate

Note that communications baud rate change will not occur until after the master writes the Comm Reset\_Coil (Coil #1) as follows:

**Master Sends:** 53 05 00 01 FF 00 D0 48

**Sensor Responds:** 53 05 00 01 FF 00 D0 48

And the sensor has changed its communications baud rate. After changing the baud rate on the master and reading the register bank again, we observe, that it has changed and the new baud rate is actually 19230 (less than 1% error):

16bit UINT	500	0053	83	Address
16b String2	501	0045	E	Parity
16bit UINT	502	0001	1	Stopbits
32bit UINT	503	0000 4B1E	19230	Baudrate



**Errors:** Attempting to read outside the defined register address range will result in the sensor returning an Address Error Exception response. Attempting to write outside this address range will result in the sensor returning an Invalid Function Code Exception response.

To make Communication Port Settings changes the new default values upon sensor reset/power cycle, perform a “User Command - Nonvolatile Save” sequence as described in the Configuration Register Bank chapter. It is advised to not save these changes until you have verified they are working properly for your application and you have recorded the changed values in some manner for future reference.

### 3.7 COIL BANK

The Coil Register Bank is comprised of (5) Coils which are READ/WRITE. The Coils can be read or written individually or in read multiples. Only single coil WRITE operations are allowed to avoid triggering multiple, coincident reconfiguration or reset sequences. The defined coil names and numbers are shown in the table below. Coil #8-15 are reserved for future use and do not have a defined state/function at this time.

Address	Name	Type	Size (bytes)	Order	R/W	Supported Function Codes
0	RS485 Termination	bit	1	N/A	R/W	FC01,FC05,FC15
1	Comm Reset	bit	1	N/A	R/W	FC01,FC05,FC15
2	Act On Config Data	bit	1	N/A	R/W	FC01,FC05,FC15
5	Sequence Counter Reset	bit	1	N/A	R/W	FC01,FC05,FC15
7	Sensor Reset Control	bit	1	N/A	R/W	FC01,FC05,FC15

**Coil 0** RS-485\_Termination Control Coil. This coil represents the status of the onboard 120Ω RS-485 termination resistor. To disable RS-485 termination on the communication port, write a “0” to this coil. To enable the termination resistor, write a “1” to this coil. The setting of this coil is also stored in nonvolatile memory through the use of the “N,S” command (Reference Configuration Register Bank Command Registers) so that the desired termination resistor state is restored upon power cycle or device reset.

**Coil 1** Comm Reset Control Coil. This coil is written (“1”) by the master device to cause the sensor to reconfigure its communication port settings according to the values that are currently in the Communications Register Bank. The state of this coil will automatically clear back to OFF (“0”) when the sensor reconfigures its communications port. The reconfiguration will be complete in approximately 1 second from the time this coil is activated (“1”) by the master write command. An example of the use of this coil has been provided in the Comm Register Bank Section.

**Coil 2** Act On Config Data Coil. This coil is written (“1”) by the master device to cause the sensor to act upon any parameter changes made in the Configuration Register Bank, or to execute a command sequence that has been written to the CMD\_REGx registers in the Configuration Register Bank. The state of this coil will automatically clear back to OFF (“0”) when the sensor reconfigures its parameter(s). The reconfiguration will be complete in approximately 1 second from the time this coil is activated (“1”) by the master write command. An example of the use of this coil has been provided in the Configuration Register Bank Section. It is recommended to wait 1.5-2.0 seconds after receiving the sensor’s response to the write command before issuing new commands.

**Coil 5** Sequence Counter Reset Coil. This coil is written (“1”) by the master device to cause the sensor to flush any accumulated sample data from its internal FIFO and to reset the Sequence Number counter. This may be used

to ensure no previous sample data is used for new measurements when changing the sample or averaging rates, or to reset the Sequence counter under application program control. The state of this coil will automatically clear back to OFF (“0”) when the sensor completes flushing its internal FIFO data and resets the counter. The reconfiguration will be complete in approximately 1 second from the time this coil is activated (“1”) by the master write command.

**Example:** Reset the sequence counter and flush any current FIFO data from the sensor’s internal buffers.

Prior to issuing the command, the Data1 Bank contains:

16bit UINT	200	0003	3	STATUS0
16bit UINT	201	0000	0	STATUS1
32bit UINT	202	0000 978C	38796	Sequence
32bit Float	204	4200 41EC	32.0643768	Temp
32bit Float	206	4046 B4A8	3.10477638	X Angle
32bit Float	208	C0C2 854C	-6.07877159	Y Angle

**Master Sends:** 53 05 00 05 FF 00 91 89

**Sensor Responds:** 53 05 00 05 FF 00 91 89

And the contents of the DATA2 Register Bank now show the sequence number has been reset:

16bit UINT	200	0003	3	STATUS0
16bit UINT	201	0000	0	STATUS1
32bit UINT	202	0000 0002	2	Sequence
32bit Float	204	4200 6D04	32.1064606	Temp
32bit Float	206	4046 B06C	3.10451794	X Angle
32bit Float	208	C0C2 873B	-6.07900763	Y Angle

**Coil 7** Sensor Device Reset Coil. This coil is written (“1”) by the master to cause the sensor to perform a self reset, or “reboot”. The sensor will not cycle its own power, but will fully reboot in a manner similar to performing a power cycle. Any parameter changes which were made but not saved to nonvolatile memory via the “N,S” command and Act On Config Data coil writing sequence will be lost. Any changes which were saved prior to writing this bit will be restored upon the sensor’s return to operating status. The reset/reboot sequence will require approximately 4-5 seconds after the master writes to this coil.

**Errors:** Attempting to read outside the defined register address range will result in the sensor returning an Address Error Exception response. Attempting to write into this address range will result in the sensor returning an Invalid Function Code Exception response.

### 3.8 FAULT/STATUS REGISTER BANK

The Fault/Status Register Bank is comprised of (2) 16-bit registers which are READ ONLY. The information is presented in Most Significant Byte First (Big Endian) order. The contents of this register bank are also mapped into the first (2) 16-bit register words of the DATA1 and DATA2 Register Banks for user convenience. In this way, the user’s application can monitor the devices’ fault/status bits along with position and temperature information in a single scan/read operation.

The user’s application can retrieve this data with a direct read (FC03 or FC04) of Register Address 700, or this same 16-bit register address fault/status value can be obtained with Diagnostic Function (FC08), Sub-function 02 (Return Diagnostic Register). Additionally, this same status register data is also made available at Data2 Register Bank address 200 and at Data1 Register Bank address 100 (Data1 Bank byte order is least-significant byte first, Data Bank 2 is most-significant first) where it can be read along with sensor output data with a single read command.

The register bits can be cleared by issuing a MODBUS Diagnostic command “Clear Counters and Diagnostic Register” command (Function Code 8, Sub Code 10). The register bits correspond to sensor status conditions in the table.

Register Bank Name	Address	Register #	Name	Type	Size (bytes)	Order	R/W	Supported Function Codes
Fault/Status Bank	700	701	Diagnostic / Fault Status Reg0	UINT16	2	UINT16, msbf	R	FC03,FC04 , FC08:SFC02
				Bit 0: COM Port Parity Error		0x0001		
				Bit 1: COM Port Framing and/or Overrun Error		0x0002		
				Bit 2: Measured temperature is above the maximum calibrated range of the device		0x0004		
				Bit 3: Measured temperature is below the minimum calibrated range of the device		0x0008		
				Bit 4: Measured value (angle, acceleration) is above the maximum calibrated range of the device - depends on LUT calibration settings		0x0010		
				Bit 5: Measured value (angle, acceleration) is below the minimum calibrated range of the		0x0020		
				Bit 6: Failure to load Calibration (LUT) data		0x0040		
				Bit 7: Failure to load Nonvolatile Settings data from EEPROM chip - using default		0x0080		
				Bit 8: Failure to ID, configure, or self test Accelerometer IC		0x0100		
				Bit 11: Test Mode Active in Accelerometer		0x0800		
				Bit 12: The system has been reset due to a software fault/Watchdog timeout error - operational stability may or may not be maintained.		0x1000		
				Bit 13: BrownOut/Power Cycle Detect. System has been reset from power brownout condition or has just been powered up		0x2000		
	701	702	Reserved	UINT16	2	UINT16, msbf	R	FC03,FC04

Register Fault/Status Bit definitions:

700 Fault/Status\_REG0 presently contains all of the main fault or status bits in the sensor. They are presently defined as follows:

Bit 0: COM Port Parity Error – Parity Error(s) have been detected on the communication port interface. This Fault Bit can be cleared by the master with a write of “1” to the COMM\_RESET Coil.

Bit 1: COM Port Framing Error – Character Framing or Overrun Error(s) have been detected on the communication port interface. This Fault Bit can be cleared by the master with a write of “1” to the COMM\_RESET Coil.

- Bit 2: Sensor Temperature detected to be above the calibrated range of the device. The application can attempt to clear this Bit is to reset the device via Power Cycle, or by the master writing a “1” to the Sensor Reset Control Coil, or with the MODBUS Diagnostic Reset Command. If the measured temperature returns to within the calibrated range of the device, the bit can reset automatically.
- Bit 3: Sensor Temperature detected to be below the calibrated range of the device. The application can attempt to clear this Bit is to reset the device via Power Cycle, or by the master writing a “1” to the Sensor Reset Control Coil, for with the MODBUS Diagnostic Reset Command. If the measured temperature returns to within the calibrated range of the device, the bit can reset automatically.
- Bit 4: Measured value (angle, acceleration) is above the maximum calibrated range of the device, output is clamped/saturated at positive limit.
- Bit 5: Measured value (angle, acceleration) is below the minimum calibrated range of the device, output is clamped/saturated at negative limit.
- Bit 6: Calibration data table load failure. This is a serious configuration fault . The only way to attempt to clear this Bit is to reset the device via Power Cycle, or by the master writing a “1” to the Sensor Reset Control Coil.
- Bit 7: Nonvolatile Settings Load/Restore Failure. This is a serious configuration fault. The only way to attempt to clear this Bit is to reset the device via Power Cycle, or by the master writing a “1” to the Sensor Reset Control Coil.
- Bit 11: Test Mode Active Status bit. This bit is set when Test Mode 1 or Test Mode 2 have been activated. It is present to indicate to the user that the reported measurements are not actual values, but test reference-only values. This bit is cleared by exiting Test Mode with the “T0” command.
- Bit 12: WatchDog Fault Bit. This bit is set when the sensor’s microcontroller detects that it has been reset through a Watchdog timeout fault. This can be the result of a firmware problem or upon receipt of a “Sensor Reset” Command or Control bit activation.
- Bit 13: Brownout Fault Bit. This bit is set when the sensor’s microcontroller detects that it has powered up from a power cycle or brownout fault condition. The bit is often set upon initial powerup, but if it has been cleared, and later found to be set again, the user will know that a power interruption or brownout had occurred.

**Register address 701 is reserved for future use.**

**Errors:** Attempting to read outside the defined register address range will result in the sensor returning an Address Error Exception response. Attempting to write into this address range will result in the sensor returning an Invalid Function Code Exception response.

## 4.0 MODBUS FUNCTIONS SUPPORTED

The JDx sensor supports the following MODBUS command functions and response codes, depending on the register bank address(es) or diagnostic functions specified by the command.

### 4.1 EXCEPTION RESPONSE CODES

If the JDx sensor receives a properly formed MODBUS-RTU command that attempts to access an invalid Coil or Register address, attempts an invalid function, or specifies an invalid value in a data field of the request – an exception response is generated according to MODBUS-RTU protocols.

The Exception Response Codes generated by the JDx sensor are as indicated in the table below:

Code	Name	Description
0x01	ILLEGAL FUNCTION	The function code received in the request is not allowed by the server for the specified address range. For example, trying to issue Function Code 06 (Write Single Register) in a Read-Only range of Addresses will result in this exception response being returned.
0x02	ILLEGAL DATA ADDRESS	The address specified (or the number of values requested) exceeds the allowable range for that function. For example, trying to issue Function Code 01 (Read Coils) to read 16 Coils starting with Coil#2 – which would be an attempt to read more than the number of defined coils in the unit- will result in this exception response being returned.
0x03	ILLEGAL DATA VALUE	A Data field in the message contains an invalid value. For example, trying to issue a Function Code 04 (Read Input Registers) with “0” as the Number of Registers to read will result in this exception response being returned.

### 4.2 FC01 (01) READ COILS

This function (0x01) is used to read from 1 to 16 contiguous bit/ virtual coil states in the JDx sensor, though a number of these are reserved for future use. The command message is comprised of the Slave Address; Function Code 0x01(1 byte); the starting Coil Address (2 bytes), the number of Coils to be read (2 bytes), and a 2-byte checksum. Coils are addressed beginning at 0, thus the (16) coils are addressed as 0 – 15.

Example: The following command reads the state of 8 Coil/bits beginning with Address/Coil # 0, on JDx sensor with Unit Address #77:

Address	FC	Start Addr High	Start Addr Low	Number of Coils-High	Number of Coils-Low	CRC_High	CRC-Low
0x4D	0x01	0x00	0x00	0x00	0x08	0x33	0xC0

Successful response from JDx sensor:

Address	FC	Num data bytes	Coil Data#0-7	CRC_High	CRC-Low
0x4D	0x01	0x01	0x00	0x47	0x18

### 4.3 FC03 (03) READ HOLDING REGISTER

This function (0x03) is used to read from 1 or more contiguous data registers in the JDX sensor. The command message is comprised of the Slave Address (1 byte); Function Code 0x03(1 byte); the starting Register Address (2 bytes), the number of Registers to be read (2 bytes), and a 2-byte checksum. Registers are addressed beginning at 0.

Example: The following command reads the contents of (10) Data Registers, beginning with Address #200, on JDx sensor with Unit Address #77. This would retrieve the contents of Data Registers #200 through #209, inclusive:

Address	FC	Start Addr High	Start Addr Low	Number of Regs-High	Number of Regs-Low	CRC_High	CRC-Low
0x4D	0x03	0x00	0xC8	0x00	0x0A	0x4A	0x3F

Successful response from JDx sensor (total length, 25 bytes, intermediate data bytes (0x00) not shown):

Address	FC	Num data bytes	Reg200,MSB	Reg200,LSB	Reg201,MSB	...	Reg209,LSB	CRC_High	CRC-Low
0x4D	0x03	0x14	0x21	0x00	0x00	...	0x00	0x78	0xB7

### 4.4 FC04 (04) READ INPUT REGISTERS

This function (0x04) is used to read from 1 or more contiguous data registers in the JDX sensor. The command message is comprised of the Slave Address (1 byte); Function Code 0x04(1 byte); the starting Register Address (2 bytes), the number of Registers to be read (2 bytes), and a 2-byte checksum. Registers are addressed beginning at 0.

Example: The following command reads the contents of (10) Data Registers, beginning with Address #200, on JDx sensor with Unit Address #77. This would retrieve the contents of Data Registers #200 through #209, inclusive:

Address	FC	Start Addr High	Start Addr Low	Number of Regs-High	Number of Regs-Low	CRC_High	CRC-Low
0x4D	0x04	0x00	0xC8	0x00	0x0A	0xFF	0xFF

Successful response from JDx sensor (total length, 25 bytes, intermediate data bytes (0x00) not shown):

Address	FC	Num data bytes	Reg200,MSB	Reg200,LSB	Reg201,MSB	...	Reg209,LSB	CRC_High	CRC-Low
0x4D	0x04	0x14	0x21	0x00	0x00	...	0x00	0x4E	0x51

### 4.5 FC05 (05) WRITE SINGLE COIL

This function (0x05) is used to write data into (1) bit “output” in the JDX sensor. The JDx sensor product doesn’t have any physical output coils. These single bits/outputs are actually “virtual” coils or settings bits which are used to activate specific functions in the sensor. Refer to section **3.7 Coil Bank** for more specific details on the defined bits and their effects. The command message is comprised of the Slave Address (1 byte); Function Code 0x05 (1 byte); the Coil/Output Bit Address (2 bytes), the coil/bit value to be written (2 bytes), and a 2-byte checksum. Coils/Output bits are addressed beginning at 0. There are only (2) acceptable values for the coil/bit state in the command: ON (0xFF00) or OFF (0x0000). All other byte values are illegal and can result in a DATA VALUE EXCEPTION response.

Example: The following command force ON Coil/Output Bit #0, on JDx sensor with Unit Address #77. This message from a master would update the state of Bit #0 only:

Address	FC	Coil Addr High	Coil Addr Low	Coil Value-High	Coil Value-Low	CRC_High	CRC-Low
0x4D	0x05	0x00	0x00	0xFF	0x00	0x82	0x36

Successful response from JDx sensor – for this FC05 command a successful response looks the same as the command :

Address	FC	Coil Addr High	Coil Addr Low	Coil Value-High	Coil Value-Low	CRC_High	CRC-Low
0x4D	0x05	0x00	0x00	0xFF	0x00	0x82	0x36

### 4.6 FC06 (06) WRITE SINGLE REGISTER

This function (0x06) is used to write data into (1) 16-bit data register in the JDX sensor. The command message is comprised of the Slave Address (1 byte); Function Code 0x06 (1 byte); the Register Address (2 bytes), the value to be written (2 bytes), and a 2-byte checksum. Registers are addressed beginning at 0.

Example: The following command writes the numeric value “10” into the contents of Register #300, on JDx sensor with Unit Address #77. This message from a master would update the contents of Data Registers #300 only:

Address	FC	Reg Addr High	Reg Addr Low	Reg Value-High	Reg Value-Low	CRC_High	CRC-Low
0x4D	0x06	0x01	0x2C	0x00	0x0A	0xC7	0xF4

Successful response from JDx sensor – for the FC06 command a successful response looks the same as the command :

Address	FC	Reg Addr High	Reg Addr Low	Reg Value-High	Reg Value-Low	CRC_High	CRC-Low
0x4D	0x06	0x01	0x2C	0x00	0x0A	0xC7	0xF4

### 4.7 FC16 (0x10) WRITE MULTIPLE REGISTERS

This function (0x10) is used to write data into 1 or more contiguous data registers in the JDX sensor. The command message is comprised of the Slave Address (1 byte); Function Code 0x10(1 byte); the starting Register Address (2 bytes), the number of Registers to be read (2 bytes), a data length (1 byte), variable number of data bytes, and a 2-byte checksum. Registers are addressed beginning at 0.

Example: The following command writes the ASCII text “CUSTOMER” into the contents of (4) Data Registers, beginning with Address #302, on JDx sensor with Unit Address #77. This message from a master would update the contents of Data Registers #302 through #305, inclusive:

Address	FC	Start Addr High	Start Addr Low	Number of Regs-High	Number of Regs-Low	Num data byt	Reg302,MSB
0x4D	0x10	0x01	0x2E	0x00	0x04	0x08	0x43

Reg302,LSB	Reg303,MSB	Reg303,LSB	Reg304,MSB	Reg304,LSB	Reg305,MSB	Reg305,LSB	CRC_High	CRC-Low
0x55	0x53	0x54	0x4F	0x4D	0x45	0x52	0xD9	0x87

Successful response from JDx sensor:

Address	FC	Start Addr High	Start Addr Low	Number of Regs-High	Number of Regs-Low	CRC_High	CRC-Low
0x4D	0x10	0x01	0x2E	0x00	0x04	0xAE	0x33



## 4.8 FC08 (08) DIAGNOSTICS

This function (0x08) has several supported sub functions in the JDX sensor. The Modbus FC08 code is used to implement various diagnostics functions on the Modbus network. The diagnostic command message is comprised of the Slave Address (1 byte); Function Code 0x08 (1 byte); the Sub-function code (2 bytes); a Data field(2 bytes); and a 2-byte checksum.

The JDX sensor presently supports the following diagnostic sub-functions.

### 4.8.1 FC08 (08), SUBFUNCTION 0, RETURN QUERY DATA

The data passed in the diagnostic request data field is returned (looped back) to the master in the response message. The entire response message received by the master should appear identical to the message that it sent.

Example: The following command requests that the device at address 77 returns back the identical message as received to the master:

Address	FC	SubFunc-High	SubFunc Low	DataValue-High	DataValue-Low	CRC-High	CRC-Low
0x4D	0x08	0x00	0x00	0x00	0xFD	0x2F	0x86

Successful response from JDX sensor – for the FC08, SFC00 command a successful response looks the same as the original command :

Address	FC	SubFunc-High	SubFunc Low	DataValue-High	DataValue-Low	CRC-High	CRC-Low
0x4D	0x08	0x00	0x00	0x00	0xFD	0x2F	0x86

### 4.8.2 FC08 (08), SUBFUNCTION 1, RESTART COMMUNICATIONS OPTION

The JDX sensor will transmit a response and then reset/re-initialize its communications port, resetting its communications event counters in the process. The reset process should complete within approximately 1 second.

Example: The following command request that the device at address 77 resets its communications port:

Address	FC	SubFunc-High	SubFunc Low	DataValue-High	DataValue-Low	CRC-High	CRC-Low
0x4D	0x08	0x00	0x01	0x00	0x00	0xBF	0xC7

Successful response from JDX sensor – for the FC08, SFC01 command a successful response looks the same as the command :

Address	FC	SubFunc-High	SubFunc Low	DataValue-High	DataValue-Low	CRC-High	CRC-Low
0x4D	0x08	0x00	0x01	0x00	0x00	0xBF	0xC7

#### 4.8.3 FC08 (08), SUBFUNCTION 2, RETURN DIAGNOSTIC REGISTER

The JDx sensor will transmit a response with the value of its 16-bit Diagnostic Register (Register Address 700).

Example: The following command request that the device at address 77 returns its diagnostic register value:

Address	FC	SubFunc-High	SubFunc Low	DataValue-High	DataValue-Low	CRC-High	CRC-Low
0x4D	0x08	0x00	0x02	0x00	0x00	0x4F	0xC7

Successful response from JDx sensor (data values will vary depending on sensor status/condition):

Address	FC	SubFunc-High	SubFunc Low	DataValue-High	DataValue-Low	CRC_High	CRC-Low
0x4D	0x08	0x00	0x02	0x21	0x00	0x57	0x97

#### 4.8.4 FC08 (08), SUBFUNCTION 10, CLEAR COUNTERS AND DIAGNOSTIC REGISTER

The JDx sensor will clear, or reset its communications related counters and its 16-bit diagnostic register to “0”. If the conditions persist which caused bits in the diagnostic register or counters to not be “0”, these values may be reasserted immediately after the clearing function completes.

Example: The following command request that the device at address 77 to reset its communications event counters and diagnostic registers STATUS0, STATUS1:

Address	FC	SubFunc-High	SubFunc Low	DataValue-High	DataValue-Low	CRC-High	CRC-Low
0x4D	0x08	0x00	0x0A	0x00	0x00	0xCE	0x05

Successful response from JDx sensor will appear the same as the command request:

Address	FC	SubFunc-High	SubFunc Low	DataValue-High	DataValue-Low	CRC-High	CRC-Low
0x4D	0x08	0x00	0x0A	0x00	0x00	0xCE	0x05

#### 4.8.5 FC08 (08), SUBFUNCTION 12, RETURN BUS COMMUNICATION ERROR COUNT

The JDx sensor will reply with the number of CRC errors it has encountered since its last restart or clear counters operation. The value is a 16-bit integer and will rollover if the number of errors exceeds the numeric range of the register.

Example: The following command request that the device at address 77 returns its count of Bus Communication errors (CRC errors):

Address	FC	SubFunc-High	SubFunc Low	DataValue-High	DataValue-Low	CRC-High	CRC-Low
0x4D	0x08	0x00	0x0C	0x00	0x00	0x2E	0x04

Successful response from JDx sensor will have the CRC error count in the data value field(s). If there happen to be NO ERRORS detected yet, the response will appear similar to the request, as shown:

Address	FC	SubFunc-High	SubFunc Low	DataValue-High	DataValue-Low	CRC-High	CRC-Low
0x4D	0x08	0x00	0x0C	0x00	0x00	0x2E	0x04

#### 4.8.6 FC08 (08), SUBFUNCTION 13, RETURN BUS EXCEPTION COUNT

The JDx sensor will reply with the number of Exception errors it has encountered since its last restart or clear event counters operation. The count represents the total number of INVALID FUNCTION, DATA ADDRESS, or DATA VALUE exception errors the sensor has issued in response to command requests. The value is a 16-bit integer and will rollover if the number of errors exceeds the numeric range of the register.

Example: The following command request that the device at address 77 returns its count of Bus Exception errors:

Address	FC	SubFunc-High	SubFunc Low	DataValue-High	DataValue-Low	CRC-High	CRC-Low
0x4D	0x08	0x00	0x0D	0x00	0x00	0x7F	0xC4

Successful response from JDx sensor will have the exception error count in the data value field(s). In this example, the sensor is returning a count of (5) Exception Errors:

Address	FC	SubFunc-High	SubFunc Low	DataValue-High	DataValue-Low	CRC-High	CRC-Low
0x4D	0x08	0x00	0x0D	0x00	0x05	0xBF	0xC7

#### 4.8.7 FC08 (08), SUBFUNCTION 14, RETURN SERVER MESSAGE COUNT

The JDx sensor will reply with the number of messages it has received and processed since its last restart or clear event counters operation. The value is a 16-bit integer and will rollover if the number of processed messages exceeds the numeric range of the register.

Example: The following command request that the device at address 77 returns its count of processed messages:

Address	FC	SubFunc-High	SubFunc Low	DataValue-High	DataValue-Low	CRC-High	CRC-Low
0x4D	0x08	0x00	0x0E	0x00	0x00	0x8F	0xC4

Successful response from JDx sensor will have the message count in the data value field(s). In this example, the sensor is returning a count of 63 (0x3F) messages processed:

Address	FC	SubFunc-High	SubFunc Low	DataValue-High	DataValue-Low	CRC-High	CRC-Low
0x4D	0x08	0x00	0x0E	0x00	0x3F	0xCF	0xD4

#### 4.8.8 FC08 (08), SUBFUNCTION 18, RETURN BUS CHARACTER OVERRUN COUNT

The JDx sensor will reply with the number of occurrences of character overrun, framing, or parity errors it has encountered since its last restart or clear event counters operation. This count can represent one or more messages that could not be identified or processed by the device. The value is a 16-bit integer and will rollover if the number of errors exceeds the numeric range of the register.

Example: The following command request that the device at address 77 returns its count of processed messages:

Address	FC	SubFunc-High	SubFunc Low	DataValue-High	DataValue-Low	CRC-High	CRC-Low
0x4D	0x08	0x00	0x12	0x00	0x00	0x4E	0x02

Successful response from JDx sensor will have the message count in the data value field(s). If there are “0” errors encountered, the response message will appear similar to the request message, as shown here:

Address	FC	SubFunc-High	SubFunc Low	DataValue-High	DataValue-Low	CRC-High	CRC-Low
0x4D	0x08	0x00	0x12	0x00	0x00	0x4E	0x02

## 5.0 TROUBLESHOOTING

If the JDx sensor doesn't appear to respond to Modbus scans, first verify that it is powered with suitable power supply voltage and that it is consuming approximately 360mW to 800mW of power.

If the sensor is powered correctly, it is possible that the communications baud rate, parity setting, or slave address being used by the master does not correctly match the configuration of the sensor. As shipped from the factory, the JDx sensor will have its node or slave address set to match the last (2) digits of the unit's serial number. For example, if a unit has serial number "398736", the factory-programmed address will be "36". ". If the last (2) digits of the serial number happen to be "00", the factory programmed address will be "100". The communication port settings as configured from the factory will be set to 19.2 kbps, Even parity, 1 Stop bit. The user can reconfigure any or all of these parameters to suit the application. If one has reconfigured these parameters, and then is unable to communicate with the device, here are some common Modbus-specific suggestions for debugging:

Try using the factory default settings described above, or the *previous* user-modified communications settings. It is possible that the parameters changes had not been saved in nonvolatile memory in the device before it was power-cycled or reset. Also, some parameters, such as Slave Address, will remain at their previous value if an invalid value has been requested by the user.

Attempt to do a simple FC01 (Read Single Coil) command on Coil #0 or an FC08, Subfunction 0 (Diagnostic Return Query Data) and listen for/capture any responses as you cycle through possible addresses in the range of 1-247. If you obtain a response, use that address to attempt a Read Register (FC03) command on the Identity Register Bank (starting Register address "0"). The first (8) registers of the Identity Bank will hold the ASCII character string of the sensor product's Model Number. Reading more registers in this range will provide the serial number and other identifying information.

If you cycle through the address ranges without finding the unit, try a different baud rate or parity setting, and repeat.

If the sensor encounters a nonvolatile memory corruption or hardware failure, it will attempt to configure its communication port to the Modbus default values of 19.2 kbps, Even Parity, 1 Stop bit, and use a Modbus Slave address of "1".

**APPENDIX A: WARRANTY & LIMITATION OF LIABILITY**

Standard goods (those listed in Jewell Instruments' published sales literature, excluding software) manufactured by Jewell Instruments LLC are warranted against defects in materials and workmanship for twelve (12) 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 to 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.