DCP

Version 1.0.0

EmbedSense®

Data Communications Protocol

MicroStrain® Little Sensors, Big Ideas®

www.microstrain.com

459 Hurricane Lane, Suite 102
Williston, VT 05495
Phone 802-862-6629
Fax 802-863-4093
www.microstrain.com
support@microstrain.com

ISSUED: 5 August 2009

Table of Contents

Document Overview

This document describes the data communications protocol for the EmbedSense® wireless sensor and data acquisition system which includes: 1) passively-powered wireless sensor **Nodes** which acquire and send strain, voltage, temperature, pressure, load and/or other sensor data, 2) an **Interrogation Antenna** which powers the nodes as well as receives the sensor data, and 3) a **Reader Assembly** which powers the Interrogation Antenna, digitizes the sensor data, and interfaces the host computer.

Communication Interface

Communication between the Reader Assembly and the host computer is via a standard RS-232 connection as shown in the RS-232 Signals Definition and RS-232 Serial Port Configuration tables.

RS-232 Signals Definition

Signal	Name	Direction	Function
TxD	Transmit Data	Host to Base Station	Asynchronous Serial Data from Host
RxD	Receive Data	Base Station to Host	Asynchronous Serial Data to Host
GND	Signal Ground	N/A	Signal Ground Reference

RS-232 Serial Port Configuration

Baud Rate	115.2K
Parity	None
Data Bits	8
Stop Bits	1

Commands Overview

The following table presents a quick reference list of communication commands for controlling the Reader Assembly and the Nodes. Please see the sections following for more detailed information on each command.

Base Station Commands

Command	Cmd Byte	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8	Byte 9
Ping Base Station	0x01									

Node Commands

Command	Cmd Byte	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8	Byte 9
Short Ping	0x02	Node Address								
		MSB	LSB							
Read Node EEPROM	0x03	Node Address		EEprom Address						
		MSB	LSB	MSB	LSB					
Write Node EEPROM	0x04	Node Address		EEprom Address	Write Value		Checksum			
		MSB	LSB		MSB	LSB	MSB	LSB		
		MSB	LSB							
Initiate Real-Time Streaming	0x38	Node Address								
		MSB	LSB							

Data Format (MSB, LSB)

All communication is performed at the byte level. To represent values greater than a single byte the value is broken down into several bytes, transmitted, received as several bytes and reassembled into a single value. Throughout this document the notation of MSB (Most Significant Byte) and LSB (Least Significant Byte) will be used to describe 2 byte values.

Example:
A value of 476 would yield an MSB of 1 and an LSB of 220.

	Decimal	Hex	Binary
2 Byte Value	476	01 DC	00000001 1101100
MSB	1	01	00000001
LSB	220	DC	1101100

Code Snippets

Sample code to convert between a 2 byte value and an MSB:LSB pair

Sample C++ Code

```
Void ValueToMSBandLSB(WORD value, BYTE& msb, BYTE& lsb)
{
     msb = value >> 8;      //shift 8 bits right, drop the lower byte
     lsb = value & 0x00ff; //mask out the upper byte
}

WORD ValueFromMSBandLSB(BYTE msb, BYTE lsb)
{
     return (msb << 8)| lsb; //shift msb 8 bits left and ‘or’ the lsb in
}
```

Sample VB 6.0 code

```
Function ValueToMSBandLSB(value As Long, msb As Byte, lsb As Byte)
    msb = value \ 256
    lsb = value Mod 256
End Function

Function ValueFromMSBandLSB(msb As Byte, lsb As Byte) As Long
    ValueFromMSBandLSB = (CLng(msb) * 256) + lsb
End Function
```

Checksum

Some commands and responses require or supply a checksum. The checksum is used to ensure that the data was transmitted without error. The documentation for each command or response that uses a checksum will detail how the checksum is calculated. The checksum is calculated by summing all the bytes protected by the checksum and taking the modulo N of the sum, where N is 255 for a 1 byte checksum and 65535 for a 2 byte checksum.

1 Byte checksum = SumOfBytes Mod 255
2 Byte Checksum = SumOfBytes Mod 65535

1 Byte Checksum Example

Protected Command Bytes

Byte	Value
Byte 1	10
Byte 2	121
Byte 3	37
Byte 4	235

```
Sum:        403 = 10 + 121 + 37 + 235
Checksum:   148 = 403 MOD 255
```

2 Byte Checksum Example

Protected Command Bytes

Byte	Value
Byte 1	10
Byte 2	121
Byte 3	37
Byte 4	235

```
Sum:        403 = 10 + 121 + 37 + 235
Checksum:   403 = 403 MOD 65535
```

Base Station Commands

Ping Base Station

Use the **Ping Base Station** command to ensure that the host computer and the Reader Assembly are properly communicating.

Command:

Command Byte:	0x01

Command Data:

None

Response Success:

Byte 1	0x01

Response Fail:

No Response

Node Commands

Short Ping

Use the **Short Ping** command to check the communication between the Reader Assembly and the Node.

Command:

Command Byte:	0x02

Command Data:

Byte 1	Node Address (MSB)
Byte 2	Node Address (LSB)

Response Success:

Byte 1	0x02

Response Fail:

Byte 1	0x21

See also: Data Format (MSB, LSB)

Read Node EEPROM

Use the **Read Node EEPROM** command to read the value of a specific memory address from the Node's EEPROM.
Important Note: The only EEPROM Address that should be read or written is Address 50. Address 50 contains the unique Node ID and can be a value of 1 to 65535. By default the Node ID is 16384.

Command:

Command Byte:	0x03

Command Data:

Byte 1	Node Address (MSB)
Byte 2	Node Address (LSB)
Byte 3	EEPROM Address (MSB)
Byte 4	EEPROM Address (LSB)

Response Success:

Byte 1	0x03
Byte 2	Value (MSB)
Byte 3	Value (LSB)
Byte 4	Checksum (MSB)
Byte 5	Checksum (LSB)

Response Fail:

Byte 1	0x21

Response Packet Checksum
This response packet contains a checksum of bytes 2 - 3.

See also: Data Format (MSB, LSB) , Checksum

Write Node EEPROM

Use the **Write Node EEPROM** command to write a value to a specific memory address on the Node's EEPROM. Use the **Read Node EEPROM** command to confirm that the value was written correctly.
Important Note: The only EEPROM Address that should be read or written is Address 50. Address 50 contains the unique Node ID and can be a value of 1 to 65535. By default the Node ID is 16384.

Command:

Command Byte:	0x04

Command Data:

Byte 1	Node Address (MSB)
Byte 2	Node Address (LSB)
Byte 3	EEPROM Address
Byte 4	Value (MSB)
Byte 5	Value (LSB)
Byte 6	Checksum (MSB)
Byte 7	Checksum (LSB)

Response Success:

Byte 1	0x04

Response Fail:

No Response

Command Packet Checksum
This command packet contains a checksum of bytes 1 – 5.

See also: Data Format (MSB, LSB) , Checksum

Initiate Real-Time Streaming

Use the **Initiate Real-Time Streaming** command to start a real-time streaming session on a Node. The Node will respond by immediately sending a stream of data packets as the sensors are read.

Command:

Command Byte:	0x38

Command Data:

Byte 1	Node Address (MSB)
Byte 2	Node Address (LSB)

Response:
The Node will initially return 10-20 'garbage' bytes followed by a 0xFF byte (decimal 255). The 0xFF byte is the header of the first valid data packet shown below as Byte 1. The initial garbage bytes should be discarded and parsing should begin with the first 0xFF byte. Each successive 0xFF indicates the start of a new data packet and the end of the previous packet.

Response Packet:

Byte 1	0xFF
Byte 2	Channel 1 Value (MSB)
Byte 3	Channel 1 Value (LSB)
…	
Byte (*N**2)-2	Channel *N* Value (MSB)
Byte (*N**2)-1	Channel *N* Value (LSB)
Byte (*N**2)	Checksum Byte

(N is the total number of channels)

Data Format
There are no 0xFF's in the actual data bytes as the data is sent in 12 bit format, shifted 1 bit to the left. To convert the data to the actual value shift the data 1 bit to the right. Dividing by 2 will also have the same effect as shifting 1 bit right.

Data is rebuilt into 12 bit format by the following relationship:
(Channel N MSB*256 + Channel N LSB)>>1
or
(Channel N MSB*256 + Channel N LSB)/2

End of Stream

The normal end of a stream occurs when the Node is withdrawn from the Interrogation Antenna. The Node loses power and the stream ends. This end of stream can be interpreted by the host computer using a timeout. For example, if the host detects no further stream (packets) after 5 seconds, the application can signal the user that the stream has ended.

Response Packet Checksum

This response packet contains a checksum of data bytes 2 – (N-1) where **N** is the total number of bytes. Unlike most checksums which are 2 bytes, this checksum is a single byte.

See also: Data Format (MSB, LSB) , Checksum

Suggested Debugging Tools

We have found that software port *sniffers* are incredibly valuable when building applications with serial communication. Here are some of our favorites; we highly recommend their use.

LookRS232

http://www.lookrs232.com/

- Look RS232 is a tool for debugging computer connection with peripheral devices using COM port, such as modem, mini-ATS, projector etc. Its easy to use interface guarantees an easy work with Com-port.
- Look RS232 can send data through COM port, receive data from an outer device through COM port, it has port indication as well.
- Look RS232 supports connection at the standard speed of 110…115200 kbit/s, it supports commands of COM port program control.
- Look RS232 supports data input in ASCII, BIN, HEX, OCT formats, keeps log of the sent and received data and commands in ASCII, BIN, HEX, OCT formats, it has an option of time link (relating to COM port activation).
- Look RS232 works with system initiated COM ports regardless of whether they are installed on motherboard or on supplementary in/output boards, e.g. VS-Com (Roadrunner) PCI-800H 8-port PCI.

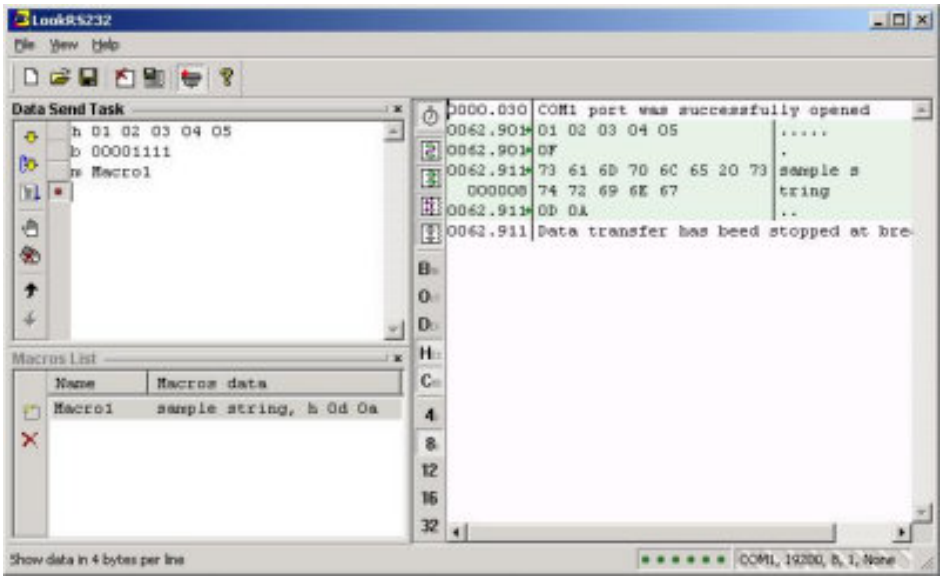

Figure 6

Serial Port Monitor

http://www.serial-port-monitor.com/index.html

Free Serial Port Monitor allows you to intercept, display and analyze all data exchanged between the Windows application and the serial device. It can be successfully used in application development, device driver or serial hardware development and offers the powerful platform for effective coding, testing and optimization.

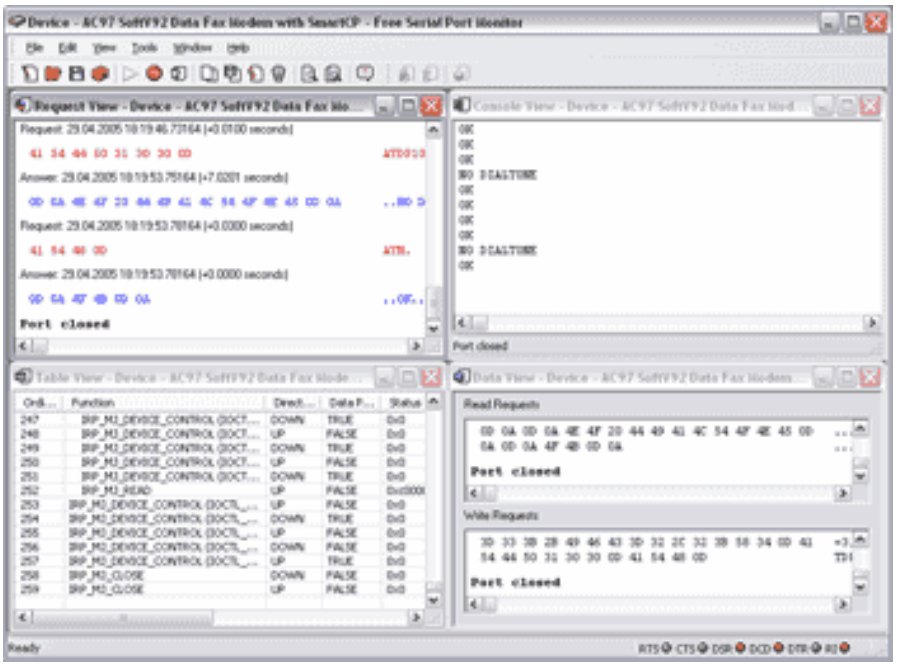

Figure 7

Comm Operator

http://www.serialporttool.com/CommOpV2Info.htm

- Comm Operator is a powerful tool for serial port communication. It can act as peripheral device emulator related RS232 COM port. It can also act as a debug tool for device's test and development.
- The built-in event driven auto send mechanics makes Comm Operator a full functioned device simulator with just a few mouse clicks. By this way, RS232 related software can be developed offsite, no need for device, only the rule script is enough.
- Comm Operator can send data through COM port, receive data from an outer device through COM port. The speed it supports range from 110 to 921600. It can detect all COM ports automatically regardless of whether they are real COM ports on motherboard or on supplementary in/output boards, or virtual COM ports created by special drivers.
- Comm Operator supports data input in ASCII, HEX as well as Decimal formats. It keeps log of the sent and received data in ASCII, HEX as well as Decimal formats. It also supports SYMBOL with which the invisible ASCII code became meaningful immediately. All setting can be saved to file and loaded later.

Support

Overview

- MicroStrain is committed to providing timely, knowledgeable, world-class support to its customers.
- We are open 24 X 7 through our web portal.
- We make every attempt to respond to your email the same business day.
- We are always available by telephone during business hours.
- We provide in-depth FAQs, manuals, quick start guides and technical notes.
- Firmware and software upgrades are made available on-line as they become available.
- Code samples in several languages are posted to aid your development.
- We support our customers as we would want to be supported.

Web

Our home page is at URL: www.microstrain.com
Our support page is at URL: http://www.microstrain.com/support_overview.aspx

Email

MicroStrain's Support Engineers make every attempt to respond to emails requesting product support within the same business day. The more detail you can provide, the quicker we will be able to understand your issues and find solutions. Data files, pictures, screen grabs, etc. are all very helpful in generating a well-thought-out solution.

Please email us at: support@microstrain.com

Telephone

MicroStrain's Support Engineers are available by phone Monday through Friday 9:00AM to 5:00PM local time. When calling MicroStrain, indicate to the receptionist that you are calling for product support and you will be promptly routed to a Support Engineer. Please have your equipment ready to test. Every attempt will be made to solve issues while you are on the line.

1.800.449.DVRT(3878) Toll Free in US
1.802.862.6629 telephone
1.802.863.4093 fax

Local time = GMT -05:00 (Eastern Time US & Canada)

SKYPE

MicroStrain's Support Engineers are available by SKYPE Monday through Friday 9:00AM to 5:00PM local. SKYPE name: **microstrain.wireless.support**