Version

1.03

MICROSTRAIN, INC.

Microminiature Sensors

Agile-Link™ 2.4 GHz Data Communications Protocol

©2007 by MicroStrain, Inc.
310 Hurricane Lane, Suite 4
Williston, VT 05495
Phone 802-862-6629
Fax 802-863-4093
www.microstrain.com
support@microstrain.com

REVISED: 8 November 2007

Table of Contents

Overview

This document describes the data communications protocol for the MicroStrain 2.4 GHz Agile-Link™ product line which includes the following devices:

Device	Firmware
V-Link® Node	4.99 and higher
G-Link® Node	4.99 and higher
SG-Link®-1CH Node	4.99 and higher
Serial Base Station	2.07 and higher
USB Base Station	2.07 and higher
Analog Base Station	2.07 and higher

If your equipment has firmware earlier than stated in the above table, there may be a requirement to upgrade in order to take full advantage of this protocol.

Communication between the Serial Base Station or the RS-232 interface of the Analog Base Station and the host computer is via a standard RS-232 connection as shown in the RS-232 Signals Definition and RS-232 Asynchronous Character Format 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

Communication between the USB Base Station or the USB interface of the Analog Base Station and the host computer is via a USB 2.0 compliant connection using a Silicon Laboratories CP2101 USB-to-UART Bridge chip on the Base Station supported by Silicon Laboratories Virtual Communications Port (VCP) drivers installed on the host PC.

Reference is made to the following Silicon Laboratories documents:

CP2101 Single-Chip USB to UART Bridge Product Brief

CP2101 Data Sheet

RS-232 + VCP 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 Agile-Link™ Base Station and wireless sensor Nodes. Please see the sections following for more detailed information on each command.

Base Station Commands

Command	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8	Byte 9	Byte 10
Ping Base Station	0x01									
Read Base Station EEPROM	0x72	EEprom Address								
Write Base Station EEPROM	0x77	EEprom Address	Write Value							
			MSB	LSB						

Node Commands

Command	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8	Byte 9	Byte 10
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		
Download Page	0x05	Node Address		Page Index						
		MSB	LSB	MSB	LSB					
Erase Session Data	0x06	Node Address		0x08	0x10	0x0C	0xFF			
		MSB	LSB							
Trigger Datalogging Session	0x0C	Node Address		Trigger ID (1-255)						
		MSB	LSB							
Initiate Real-Time Streaming	0x38	Node Address								
		MSB	LSB							
Initiate Sleep Mode	0x32	Node Address								
		MSB	LSB							
Read Single Sensor	0x03	Node Address		0x01	Channel Number (1-8)					
		MSB	LSB							
Auto-Balance Channel	0x62	Node Address		Channel Number (1-4)	Target Balance					
		MSB	LSB		MSB	LSB				
Long Ping	0xAA	0x05	0x00	Node Address		0x02	0x00	0x02	Checksum	
				MSB	LSB				MSB	LSB
Initiate Low Duty Cycle	0xAA	0x05	0x00	Node Address		0x02	0x00	0x38	Checksum	
				MSB	LSB				MSB	LSB
Initiate High-Speed Streaming	0xAA	0x05	0x00	Node Address		0x02	0x00	0x39	Checksum	
				MSB	LSB				MSB	LSB
Stop Node	0xAA	0xFE	0x00	Node Address		0x02	0x00	0x90	Checksum	
				MSB	LSB				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

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 PC and the Base Station are properly communicating.

Command:

Command Byte:	0x01

Command Data:

None

Response Success:

Byte 1	0x01

Response Fail:

No Response

Read Base Station EEPROM

Use the **Read Base Station EEPROM** command to read the value of a specific memory address from the Base Station's EEPROM.
See *Base Station EEPROM Map* for specific memory address details.

Command:

Command Byte:	0x72

Command Data:

Byte 1	EEPROM address

Response Success:

Byte 1	0x72
Byte 2	Value (MSB)
Byte 3	Value (LSB)

Response Fail:

No Response

See also: Data Format (MSB, LSB)

Write Base Station EEPROM

Use the **Write Base Station EEPROM** command to write a value to a specific memory address on the Base Station's EEPROM. Use the **Read Base Station EEPROM** command to confirm that the value was written correctly.
See *Base Station EEPROM Map* for specific memory address details.

Command:

Command Byte:	0x77

Command Data:

Byte 1	EEPROM address
Byte 2	Value (MSB)
Byte 3	Value (LSB)

Response Success:

Byte 1	0x77

Response Fail:

No Response

See also: Data Format (MSB, LSB)

Node Commands

Short Ping

Use the **Short Ping** command to check the communication between the Base Station 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.
See *Node EEPROM Map* for specific memory address details.

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.
See *Node EEPROM Map* for specific memory address details.

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

Download Page

When the Node is in datalogging mode, it stores the data to memory. Use the **Download Page** command to retrieve a logged data session from the Node.

Command:

Command Byte:	0x05

Command Data:

Byte 1	Node Address (MSB)
Byte 2	Node Address (LSB)
Byte 3	Page Index (MSB)
Byte 4	Page Index (LSB)

Response Success:

Byte 1	0x05
Byte 2	Data Point 1 (MSB)
Byte 3	Data Point 1 (LSB)
Bytes 4-263	Data Points 2-131 (MSB:LSB)
Byte 264	Data Point 132 (MSB)
Byte 265	Data Point 132 (LSB)
Byte 266	Checksum (MSB)
Byte 267	Checksum (LSB)

Response Fail:

Byte 1	0x21

Page Index

Each Node contains 2MB of memory. These 2MB are mapped to 8192 pages of data, each page containing 132 data points. A data point is a 2 byte value.

Pages are numbered sequentially from 0 to 8191.

- Page 0 always contains data points that represent the current value in each of the Node's EEPROM locations (0-254).
- Page 1 is a 'blank' page and contains no usable information or data.
- Page 2 is the first page that will contain datalogging data.

Data Sessions

The Node can contain multiple consecutive datalogging sessions. Each of these sessions has a leading 12 byte header which resolves into 6 data points. This header is used to identify the start of a new datalogging session and the end of the previous session. The header contains the datalogging environment during the particular session.

Session Header

The datalogging session header is a 12 byte leading string indicating the start of the next session. The header can be found anywhere on a downloaded page and it can wrap between pages. The bytes resolve as follows:

Data Point	Byte	Byte Value	Data Point Value	Description
1	1	255	65535	Fixed Header
	2	255		
2	3	254	65025-65279	Trigger ID
	4	1-255		
3	5	xxx	100-65500	Samples per Data Set
	6	xxx		
4	7	xxx	1-65535	Session Index
	8	xxx		
5	9	0	1-255	Active Channel Mask
	10	1-255		
6	11	0	1-7	Datalogging Sampling Rate
	12	1-7		

Fixed Header

Each new session is always marked with an 0xFFFF (decimal 65535).

Trigger ID

The Trigger ID supplied to Trigger Datalogging Session command for the datalogging session. If the trigger was started as a result of an event driven trigger, the Trigger ID will be 0.

Samples per Data Set

The number of expected samples per data set. If the session was logged as a continuous datalogging session, this value is not applicable. If the session ended prematurely due to power failure or because the memory was completely filled, the value will not correspond to the actual number of samples. When parsing session data, the Samples per Data Set value should not be used to determine the end of the session. The end of the session should be determined by the start of the next session header.

Session Index

Each data logging session is stored with an index. The first session's index is 1 and subsequent sessions would have consecutive index numbers. When the logged sessions are erased the index will be reset, and the first data logging session after the erase will have a Session Index of 1.

Channel Mask

The channel mask indicates the active channels that were logged in this session. Each of the 8 bits of the mask correspond to one of the Nodes channels. Bit 1 corresponds to channel 1, bit 2 to channel 2, etc. When the bit is set to 1, the session will contain data for that channel. If the bit's value is 0, no data for that channel was logged.

Sampling Rate
The Sample Rate that was used during the data logging session. The table below describes the value and the associated rate that it represents.

Value	Rate
1	2048 Hz
2	1024 Hz
3	512 Hz
4	256 Hz
5	128 Hz
6	64 Hz
7	32 Hz

Session Data
During the actual datalogging session, the data from each active channel on the Node is written consecutively to the memory pages. For example, a Node with 3 active channels would write data as CH1, CH2, CH3, CH1, CH2, CH3 and so forth. This data comes off the Node in the same pattern during download.

Response Packet Checksum
This response packet contains a checksum of bytes 2 -265.

Relevant Memory Locations
Node EEPROM 0: Current Log Page
Node EEPROM 2: Current Page Offset
Node EEPROM 4: Data Sets Stored

See also: Data Format (MSB, LSB) , Checksum, Trigger Datalogging Session, Erase Session Data

Erase Session Data

Use the **Erase Session Data** command to erase all datalogging sessions and data stored on the Node's memory.

Command:

Command Byte:	0x06

Command Data:

Byte 1	Node Address (MSB)
Byte 2	Node Address (LSB)
Byte 3	0x08
Byte 4	0x10
Byte 5	0x0C
Byte 6	0xFF

Response Success:

Byte 1	0x06

Response Fail:

Byte 1	0x21

Response
The response is returned immediately when the erasing processes begins, and no acknowledgment is sent when the process completes. The process will take approximately 5 seconds, and the Node will not respond to any commands until the erasing is complete. Completion of the erase can be detected by repeated short pinging of the Node; when the erase is complete, the Node will come back on-line and respond successfully to the ping.

See also: Data Format (MSB, LSB) , Trigger Datalogging Session, Short Ping

Trigger Datalogging Session

Use the **Trigger Datalogging Session** command to initiate a data capture session on-board the Node. The data will be stored in the Node's 2MB memory and may be downloaded at a later time.

Command:

Command Byte:	0x0C

Command Data:

Byte 1	Node Address (MSB)
Byte 2	Node Address (LSB)
Byte 3	Trigger ID

Response Success:

Byte 1	0x0C

Response Fail:

No Response

Trigger ID
The trigger ID is a value between 1 and 255 that is stored with the datalogging session.

Notes

- If a Sensor Event Driven Trigger (SEDT) is being used, the equivalent of a trigger command is automatically issued within the Node's firmware as a result of an output ceiling, or output floor being reached. Note that if SEDT is enabled, the Node will not automatically enter Sleep mode after a user inactivity timeout.

- During a datalogging session the Node will not respond to any commands.

- If continuous datalogging is enabled, the Node will continue datalogging until it uses all available memory or until it is powered off.

Relevant Memory Locations
Node EEPROM 12: Active Channel Mask
Node EEPROM 14: Datalogging Sample Rate
Node EEPROM 16: Samples per Data Set
Node EEPROM 102: Continuous Datalogging Flag

See also: Data Format (MSB, LSB) , Download Page, Erase Session Data

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

Terminating Streaming
Continuous or Finite Streaming may be stopped at any time by issuing any byte to the Base Station. This will cause the Base Station to stop streaming. It will not however cause the Node to stop streaming. The Node will continue to stream until the set (finite) duration has elapsed or the power on the Node is cycled. The Node may be stopped from streaming by issuing the **Stop Node** command.

End of Stream
The normal end of a Finite stream is marked by 4-6 consecutive 0xAA (decimal 170) bytes. When parsing the stream, these bytes should be used as a signal that finite streaming has ended.

If, during streaming, the Base Station receives no data from the Node as a result of battery draw-down in the Node, radio interference between the Base Station and the Node, the Node's power is switched off, an out-of-radio-range condition between the Base Station and the Node, etc., random noise may "dribble" from the Base Station; this takes the form of a few odd bytes every second or so.

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.

Relevant Memory Locations
Node EEPROM 12: Active Channel Mask
Node EEPROM 16: Samples per Data Set
Node EEPROM 100: Continuous Streaming/LDC Flag

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

Initiate Sleep Mode

Use the **Initiate Sleep Mode** command to put the Node in a low power state.

Command:

Command Byte:	0x32

Command Data:

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

Response Success:

No Response

Response Fail:

No Response

Waking Node

A Node in sleep mode periodically awakes, listens for a command, and if none is received, returns to sleep. To wake a sleeping Node, send the **Short Ping** command repetitively until a valid response is received indicating the Node is communicating. The purpose of the repetitive pings is to insure pings are in the air when the Node temporarily awakens. The **Stop Node** command can also be used to wake a Node.

Relevant Memory Locations

Node EEPROM 66: Sleep Interval

See also: Data Format (MSB, LSB) , Short Ping, Stop Node

Read Single Sensor

Use the **Read Single Sensor** command to read the current value of the specified channel.

Command:

Command Byte:	0x03

Command Data:

Byte 1	Node Address (MSB)
Byte 2	Node Address (LSB)
Byte 3	0x01
Byte 4	Channel Number (1-8)

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

Channel Number

The Channel Number of the sensor to read. If a channel number is used that doesn't exist on the Node, erroneous response data will be returned.

Value

The Value returned must be divided by 2 to get the correct sensor value.

Response Packet Checksum

This response packet contains a checksum of bytes 2 – 3.

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

Auto-Balance Channel

Use the **Auto-Balance** Channel command to auto-balance a particular channel on the Node. This command is only applicable to the differential channels on the V-Link® and SG-Link®.

Command:

Command Byte:	0x62

Command Data:

Byte 1	Node Address (MSB)
Byte 2	Node Address (LSB)
Byte 3	Channel to Balance (hardware specific, 1-4)
Byte 4	Target Balance Value (MSB)
Byte 5	Target Balance Value (LSB)

Response Success:

No Response

Response Fail:

No Response

Channel to Balance
Channels 1, 2, 3 and/or 4 on the V-Link®.
Channel 1 on the SG-Link®.

Target Balance Value
The target balance value represents the desired sensor output value in bits, with a range of 0-4096. All differential inputs have a programmable offset feature that allows the user to trim sensor offset (see hardware user manual for more information.) This programmable offset can be manually altered by writing to Node EEPROM locations 26-32, or auto-tuned such that the sensor output is balanced to a user-defined target. For example, a common use is to auto-balance to mid-scale (2048 bits) to obtain maximum bipolar dynamic range.

Relevant Memory Locations
Node EEPROM 26: PGA Offset 1
Node EEPROM 28: PGA Offset 2
Node EEPROM 30: PGA Offset 3
Node EEPROM 32: PGA Offset 4

See also: Data Format (MSB, LSB)

Long Ping

Use the **Long Ping** command to check the communication between the Base Station and the Node.

Command Packet:

Byte 1	SOP(Start Of Packet)	0xAA
Byte 2	Message Type	0x05
Byte 3	Address Mode	0x00
Byte 4	Node Address (MSB)	0xXX
Byte 5	Node Address (MSB)	0xXX
Byte 6	Payload Length	0x02
Byte 7	Command ID (MSB)	0x00
Byte 8	Command ID (LSB)	0x02
Byte 9	Checksum (MSB)	0xXX
Byte 10	Checksum (LSB)	0xXX

Response:
The 1 byte response comes directly from the Base Station to verify that the Long Ping command was correctly received and sent to the Node.

Byte 1	0xAA

Response Packet:

Byte 1	SOP	0xAA
Byte 2	Message Type	0x07
Byte 3	Address Mode	0x02
Byte 4	Node Address (MSB)	0xXX
Byte 5	Node Address (LSB)	0xXX
Byte 6	Payload Length	0x02
Byte 7	Payload (not used)	0x00
Byte 8	Payload (not used)	0x00
Byte 9	LQI	0xXX
Byte 10	RSSI	0xXX
Byte 11	Checksum (MSB)	0xXX
Byte 12	Checksum (LSB)	0xXX

Command Packet Checksum
This command packet contains a checksum of bytes 2 – 8. The SOP is not included in the checksum.

Response Packet Checksum
This response packet contains a checksum of bytes 2 – 8. SOP, LQI, and RSSI bytes are not included in the checksum.

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

Initiate Low Duty Cycle

Use the **Initiate Low Duty Cycle (LDC)** command to put the Node in LDC mode. The Node will send an LDC packet for every sample at the defined interval for the defined sample rate.

Command Packet:

Byte 1	SOP	0xAA
Byte 2	Message Type	0x05
Byte 3	Address Mode	0x00
Byte 4	Node Address (MSB)	0xXX
Byte 5	Node Address (LSB)	0xXX
Byte 6	Payload Length	0x02
Byte 7	Command ID (MSB)	0x00
Byte 8	Command ID (LSB)	0x38
Byte 9	Checksum (MSB)	0xXX
Byte 10	Checksum (LSB)	0xXX

Response:
The 1 byte response comes directly from the Base Station to verify that the Initiate Low Duty Cycle command was correctly received and sent to the Node.

Byte 1	0xAA

Response Packet:
These packets are received at various, defined intervals after an initiate low duty cycle command has been sent.

Byte 1	SOP	0xAA
Byte 2	Message Type	0x07
Byte 3	Address Mode	0x04
Byte 4	Node Address(MSB)	0xXX
Byte 5	Node Address(LSB)	0xXX
Byte 6	Payload Length	0x0XX
Byte 7	Application Identifier	0x02
Byte 8	Channel Mask	0xXX
Byte 9	Sample Rate	0xXX
Byte 10	Data Type	0x01
Byte 11	Timer Tick (MSB)	0xXX
Byte 12	Timer Tick (LSB)	0xXX
Byte 13	Channel 1 Data (MSB)	0xXX
Byte 14	Channel 1 Data (LSB)	0xXX
Bytes 15 through N-4	Repeat bytes 13 and 14 for the channels that are active	0xXX
Byte N-3	LQI	0xXX
Byte N-2	RSSI	0xXX
Byte N-1	Checksum (MSB)	0xXX
Byte N	Checksum (LSB)	0xXX

Command Packet Checksum
This command packet contains a checksum of bytes 2 – 8. The SOP is not included in the checksum.

Response Packet Checksum
The response packet checksum is calculated on bytes 2 – (N-4) where N is the total number of bytes in the packet. SOP, LQI, and RSSI bytes are not included in the checksum.

Relevant Memory Locations
Node EEPROM 12: Active Channel Mask
Node EEPROM 16: Samples Per Data Set
Node EEPROM 72: LDC Rate
Node EEPROM 100: Continuous Streaming/LDC Flag

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

Initiate High-Speed Streaming

Use the **Initiate High-Speed Streaming (HSS)** command to put the Node in high-speed streaming mode. The Node will send HSS packets for the defined interval at the defined sample rate. The high-speed streaming mode allows for real-time data to be transmitted at rates up to 4 kHz.

Command Packet:

Byte 1	SOP	0xAA
Byte 2	Message Type	0x05
Byte 3	Address Mode	0x00
Byte 4	Node Address (MSB)	0xXX
Byte 5	Node Address (LSB)	0xXX
Byte 6	Payload Length	0x02
Byte 7	Command ID (MSB)	0x00
Byte 8	Command ID (LSB)	0x39
Byte 9	Checksum (LSB)	0xXX
Byte 10	Checksum (LSB)	0xXX

Response:
The 1 byte response comes directly from the Base Station to verify that the Initiate High-Speed Streaming command was correctly received and sent to the Node.

Byte 1	0xAA

Response Packet:

Byte 1	SOP	0xAA
Byte 2	Message Type	0x01
Byte 3	Address Mode	0x05
Byte 4	Node Address (MSB)	0xXX
Byte 5	Node Address (LSB)	0xXX
Byte 6	Payload Length	0xXX
Byte 7	PPI (Packet Payload Identifier)	0x02
Byte 8	Channel Mask	0xXX
Byte 9	Stream Rate	0xFF
Byte 10	# of samples in packet	0xXX
Byte 11	Timer value (Byte 1)	0xXX
Byte 12	Timer value (Byte 2)	0xXX
Byte 13	Timer value (Byte 3)	0xXX
Byte 14	Timer value (Byte 4)	0xXX
Byte 15 to N-2	See below for structure of high speed stream payload	
Byte N-1	Checksum (MSB)	0xXX
Byte N	Checksum (LSB)	0xXX

Sample Rate

The sample rate is completely dependent on the number of active channels. The actual sample rate is 4 kHz / N, where N is the number of active channels programmed on the Node. The high bandwidth is achieved by buffering as many samples as possible on the Node before transmission, greatly reducing the average packet overhead per sample. The sampling loop for this mode is timer-interrupt driven with a crystal reference, providing excellent timing accuracy.

Number of Channels	Sample Rate
1	4.000 kHz
2	2.000 kHz
3	1.333 kHz
4	1.000 kHz
5	0.800 kHz
6	0.666 kHz
7	0.570 kHz
8	0.500 kHz

Stream Payload format

The Node will stream with the number of channels prescribed in the channel mask and at the rate specified above. The stream contain a series of samples, each sample consists of one data point for each channel being sampled. Each data point (2 bytes) represents the actual value sampled. See the table below and the following examples for details.

Sample	Channel
Sample 1	Channel 1
	…
	Channel n
Sample 2	Channel 1
	…
	Channel n
…	Channel 1
	…
	Channel n
Sample N	Channel 1
	…
	Channel n

Where **N** is the number of samples in packet and **n** is the total number of channels.

Example 1

If the channel mask is 0x74 (0b01110100), there are 4 channels streaming (channels 3, 5, 6 and 7). The sample rate of this stream will be 1kHz. The sampled data will have the follow format:

Sample	1				2				3					N			
Channel	3	5	6	7	3	5	6	7	3	5	6	7	…	3	5	6	7
Data point	1	2	3	4	5	6	7	8	9	10	11	12		n-3	n-2	n-1	n

Where **N** is the number of samples in packet and **n** is the last data point.

Between each sample is a 1/sample rate delay, this is due to the sampling interval speed (sample rate) describe above.

Example 2

If the channel mask is 0x01 (0b00000001), there is 1 channel stream (channel 1). The sample rate of this stream will be 4kHz.

The sampled data will have the following format:

Sample	1	2	3	4	5	6	7		N
Channel	1	1	1	1	1	1	1	…	1
Data point	1	2	3	4	5	6	7		n

Between each sample there is a .25ms delay (due to the 4 kHz sample rate)

Timer Value

The Nodes contain an on-board clock. The 4 byte timer value resolves to an integer and represents the time the HSS packet was transmitted. The time will be an increment of 1/Sample Rate and will roll over as time elapses.

Command Packet Checksum

This command packet contains a checksum of bytes 2 – 8. The SOP byte is not included in the checksum.

Response Packet Checksum

The response packet checksum is calculated on bytes 2 – (N-2), where N is the total number of bytes in the packet. The SOP byte is not included in the checksum.

Relevant Memory Locations

Node EEPROM 12: Active Channel Mask
Node EEPROM 16: Samples per Data Set
Node EEPROM 100: Continuous Streaming/LDC Flag

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

Stop Node

Use the **Stop Node** command to stop a Node that is either in **Streaming, LDC** or **Sleep** mode. It will also stop a Node in **High Speed Streaming** mode if streaming a single channel.

Command Packet:

Byte 1	SOP	0xAA
Byte 2	Message Type	0xFE
Byte 3	Address Mode	0x00
Byte 4	Node Address (MSB)	0xXX
Byte 5	Node Address (LSB)	0xXX
Byte 6	Packet Length	0x02
Byte 7	Command Byte (MSB)	0x00
Byte 8	Command Byte (LSB)	0x90
Byte 9	Checksum (LSB)	0xXX
Byte 10	Checksum (LSB)	0xXX

Response:
The 1 byte response comes directly from the Base Station to verify that the **Stop Node** command was correctly received and the Base Station is attempting to stop the Node.

Byte 1	0xAA

Response Success:
The Node was stopped and will now communicate.

Byte 1	0x90

Response Fail:
The **Stop Node** process was aborted by user sending any random byte to the Base Station.

Byte 1	0x21

The Stop Node command sends small packets as fast as possible in an attempt to communicate with Node, and periodically checks, after 1000 packets, to see if the Node has responded to a ping request. The function will continue indefinitely until either the Node responds, or the user sends
any random byte to the Base Station.

Broadcast Special Case
When the broadcast Node address 65535 is used, the Base Station does not check for a ping response. It will continue sending the stop Node command until interrupted by the user (any single byte sent to the Base Station).

Command Packet Checksum
This command packet contains a checksum of bytes 2 – 8.

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

Other Details

Node Discovery

The Node has a specialized function whereby it sends out two identification packets every time it is turned on. The packets are actually sent out on all 16 radio channels, allowing any Base Station within range to receive the identification packets, regardless of the Base Station's current frequency assignment. The Base Station immediately passes these packets to the host serial port. The Node will only send out a Node discovery packet when its power is cycled.

Node Discovery Packet:

Byte 1	SOP	0xAA
Byte 2	Message Type	(Byte 2 & 0x08) == 0
Byte 3	Address Mode	0x07
Byte 4	Node Address (MSB)	0xXX
Byte 5	Node Address (LSB)	0xXX
Byte 6	Packet Length	0x03
Byte 7	Radio Channel	0xXX
Byte 8	Model Number (MSB)	0xXX
Byte 9	Model Number (LSB)	0xXX
Byte 10	LQI	0xXX
Byte 11	RSSI	0xXX
Byte 12	Checksum (MSB)	0xXX
Byte 13	Checksum (LSB)	0xXX

Message Type
The message type byte must be vetted to insure a valid packet. The byte must be ANDed with 0x08 and the result should be 0; if <>0, the packet is not valid.

Radio Channel
The radio channel value corresponds to the channel on which the Node is currently communicating. This is the same value stored in Node EEPROM 90 and sends only values that are valid within EEPROM 90. Please, reference the Node EEPROM map for position 90 to identify valid radio channels.

Model Number
The model number corresponds to the model number of the Node stored in Node EEPROM 112. This value identifies the type of Node that had its power cycled. Please reference the Node EEPROM map for position 112 to identify valid model numbers.

Packet Checksum
This packet contains a checksum of bytes 2 – 9. SOP, LQI, and RSSI bytes are not included in the calculated checksum.

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

Channel Actions

Channel actions are linear scaling filters applied to convert a channel's raw bit value to physical or engineering units. Channel actions can be generated through Agile-Link calibration wizards or via manual calibration by the end user. Please see the Node hardware manual and Agile-Link manual for more information on general calibration procedures. This documentation explains how to use existing channel actions in custom software.

Channel action information is stored in non-volatile EEPROM memory on the Node, at locations 150 through 228 inclusive. Each channel requires 10 bytes to hold the id, gain, and offset values for that channel's particular action. Please note: 1) Channel action gain and offset are different from a Node's hardware gain and offset settings. 2) Channel actions represent a post-processing step because the conversion from bits to physical units takes place on a PC and not on the Node.

EEPROM Location	**Meaning**		**EEPROM Location**	**Meaning**
150	CH1 Channel action ID		190	CH5 Channel action ID
152	CH1 2 MSBs of the gain		192	CH5 2 MSBs of the gain
154	CH1 2 LSBs of the gain		194	CH5 2 LSBs of the gain
156	CH1 2 MSBs of the offset		196	CH5 2 MSBs of the offset
158	CH1 2 LSBs of the offset		198	CH5 2 LSBs of the offset
160	CH2 Channel action ID		200	CH6 Channel action ID
162	CH2 2 MSBs of the gain		202	CH6 2 MSBs of the gain
164	CH2 2 LSBs of the gain		204	CH6 2 LSBs of the gain
166	CH2 2 MSBs of the offset		206	CH6 2 MSBs of the offset
168	CH2 2 LSBs of the offset		208	CH6 2 LSBs of the offset
170	CH3 Channel action ID		210	CH7 Channel action ID
172	CH3 2 MSBs of the gain		212	CH7 2 MSBs of the gain
174	CH3 2 LSBs of the gain		214	CH7 2 LSBs of the gain
176	CH3 2 MSBs of the offset		216	CH7 2 MSBs of the offset
178	CH3 2 LSBs of the offset		218	CH7 2 LSBs of the offset
180	CH4 Channel action ID		220	CH8 Channel action ID
182	CH4 2 MSBs of the gain		222	CH8 2 MSBs of the gain
184	CH4 2 LSBs of the gain		224	CH8 2 LSBs of the gain
186	CH4 2 MSBs of the offset		226	CH8 2 MSBs of the offset
188	CH4 2 LSBs of the offset		228	CH8 2 LSBs of the offset

Below are more in depth explanations of each channel action field.

ID

The ID is an integer value used to determine which channel action is applied to the current channel. The ID value is the MSB of EEPROM 0. The table below shows the ID values and their corresponding channel action. The formula for each of these channel actions are described further on.

ID Value	Channel Action
0x01	Strain Wizard
0x02	G-Link Calibration Wizard
0x04	Custom Formula

Gain and Offset

The gain and offset are 4-byte floating point values in big endian format. For example, if EEPROM 2 and EEPROM 4 had the values of 44571 and 33220, respectively, their equivalent floating point value would be -1032.86499. The example below displays how this conversion works in C++.

```
/*  EEPROM 2 = 44571 and EEPROM 4 = 33220  */
unsigned short EEPROM2 = 44571, EEPROM4 = 33220;
/*  Allocate memory for a 4 byte float to hold the floating point result.  */
float fValue;
/*  point to the memory allocated to hold the floating point value.  */
unsigned char *pTemp = (BYTE*)&fValue;
pTemp[0] = (EEPROM2 >> 8);    //    MSB of EEPROM 2
pTemp[1] = (EEPROM2 & 0xff);  //    LSB of EEPROM 2
pTemp[2] = (EEPROM4 >> 8);    //    MSB of EEPROM 4
pTemp[3] = (EEPROM4 & 0xff);  //    LSB of EEPROM 4
/*  At this point, fValue contains the floating point value.  */
printf("Floating Point Value: %f\n", fValue);
```

Formulas

The formulas below are the actual formulas used by the channel action filters within Agile-Link™ software. Gain is the channel action gain, offset is the channel action offset, bit_value is the 2-byte, sample from a particular channel, and final_result is the units value of the channel action. These 4 values are in italics when an equation is shown below.

Strain Wizard Formula

final_result = *gain* * (*bit_value* + *offset*)

The final unit is in microstrain (µε).

G-Link Calibration Wizard Formula

final_result = (*bit_value* – *offset*) / *gain*

The final unit is in G's.

final_result = ((*bit_value* – *offset*) / *gain*) * *9.8*

The final unit is in m/s^2.

Custom Formula
*final_result = gain * bit_value + offset*
The unit value is defined by the user. Some of the more common gain and offset values are shown in the table below.

Unit	Gain	Offset	Applies To
Volts	.000732	0	V-Link – Channels 5, 6, 7 SG-Link – Channel 4
Temp C	.117188	-67.84	V-Link – Channel 8 G-Link – Channel 4 SG-Link – Channel 3
Battery Volts	.002197	0	SG-Link – Channel 2

Example

Below is an example of converting EEPROM addresses to their corresponding channel action. The EEPROM addresses listed below do not follow the previously mentioned addressing of EEPROM + base address. The base address is already included in the EEPROM address.

EEPROM Address	Value
190	1024
192	45283
194	16186
196	0
198	0

The above EEPROM values correspond to channel 5. This can be told because the base address, EEPROM 190, corresponds to channel 5 in the EEPROM map.

First determine which channel action is to be applied to channel 5. The channel action ID is the MSB of EEPROM 190. The MSB of 1024 is 4, 1024 >> 8 (i.e. 1024 / 256). The action ID therefore corresponds to the Custom Formula. Thus, the formula to use is gain * bit_value + offset as described earlier.

Second, the gain and offset must be calculated. The offset is very easy because both EEPROM 196 and 198 are zero. Thus, the offset is zero.

The gain has the values 45283 and 16186 for EEPROM addresses 192 and 194, respectively. Using the formula written earlier and knowing these 4 bytes are in big endian format, the floating point equivalent is 0.000732. As noted in the table of gains and offsets for the Custom Formula, this channel action converts incoming bits to volts for channel 5.

Important Disclaimer

The SDK is provided "as is" and any expressed or implied warranties, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose are disclaimed. In no event shall MicroStrain or its contributors be liable for any direct, indirect, incidental, special, exemplary, or consequential damages (including, but not limited to, procurement of substitute goods or services; loss of use, data, or profits; or business interruption) however caused and on any theory of liability, whether in contract, strict liability, or tort (including negligence or otherwise) arising in any way out of the use of this software, even if advised of the possibility of such damage. MicroStrain will make every effort to amplify the instructions contained in the Data Communications Protocol manual but will neither undertake to detail the functioning of the hardware or firmware in the Agile-Link™ family nor debug the purchaser's code.