BioHarness Bluetooth Comms Link Specification

Document History

Version	Date	Description
1.0	26th May 2010	Extracted from General Comms Link v1.66
1.1	31st May 2010	Updated after review.

Document Notes

All numbers in this document are written in decimal, except hexadecimal numbers which are prefixed by ‘0x’. For example 5436 is decimal, while 0x5436 is hexadecimal.

1. References

Ref #	ID	Description

2. Abbreviations

Abbreviation	Description
ACK	**ACK**nowlege
ASCII	**A**merican **S**tandard **C**ode for **I**nformation **I**nterchange
BPM	**B**eats **P**er **M**inute
CRC	**C**yclic **R**edundancy **C**heck
DLC	**D**ata **L**ength **C**ode
ECG	**E**lectro**C**ardio **G**raph
ETX	**E**nd of **T**ext
ID	**Id**entifier
LS	**L**east **S**ignificant
MAC	**M**edia **A**ccess **C**ontrol
MS	**M**ost **S**ignificant
NAK	**N**egative **AC**K**n**owledge
PC	**P**ersonal **C**omputer
RF	**R**adio **F**requency
STX	**S**tart of **T**ext
SPP	**Serial Port Profile**
UART	**U**niversal **A**synchronous **R**eceiver **T**ransmitter
VCP	**Virtual Com Port**

3. Introduction

This document specifies the protocol used to enable communications between a Zephyr Bluetooth BioHarness and other (remote) units. The protocol enables communications to a PC via USB VCP or via Bluetooth SPP. The protocol is half-duplex and UART-based in nature, relying on a simple request/response format. The protocol is defined, together with the specification of each message and describes the data used within each packet.

4. Communications Overview

The serial communication is request/response based, with the local or remote unit acting as master in the data transfer. The delays between the request and response messages are dependant on the message type; some can reply instantly e.g. “get version”, whereas others take longer e.g. “delete log file”; the delay essentially depends on the work the unit is requested to perform. The protocol only accommodates a point-to-point communications transfer, with no addressing used. The unit replies to the request message with one of two basic message types. If the request message is found to be valid, the response is in the form of an ACK with required data included; if found to be invalid, the response is in the form of a NAK message and no data is inserted into the message.
There is an exception to the rule where ‘periodic’ messages are transferred between units. In this case, no responses are required.

4.1. Basic Message Format

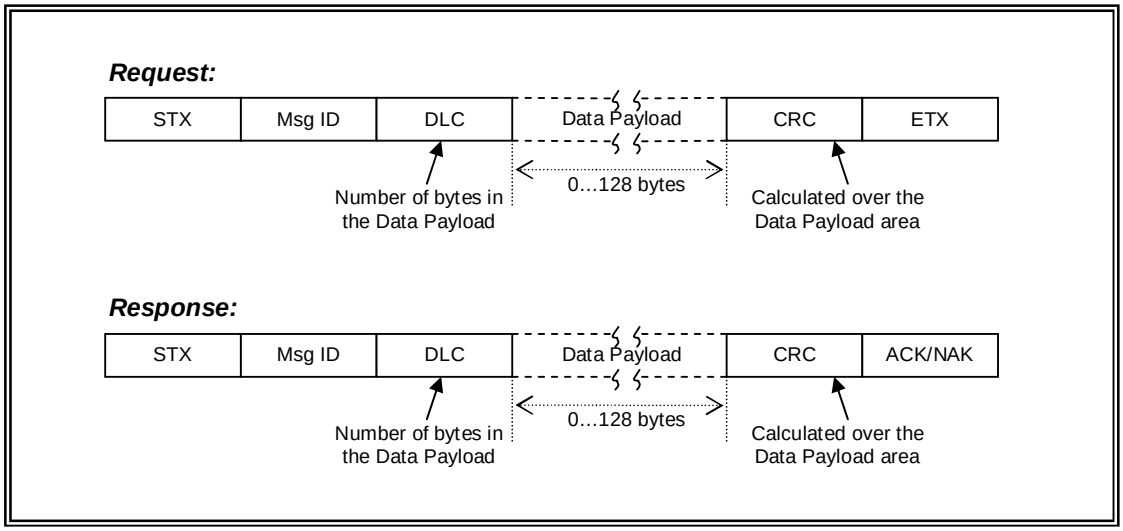

Figure 4-1 ***Basic Message Format***

4.1.1. STX

The STX field (Start of Text) is a standard ASCII control character (0x02) and denotes the start of the message. Although the protocol does not guarantee this value will not appear again within a message, it gives some delimiting to the message and therefore a start character to search for when receiving data.

4.1.2. Msg ID

The Message ID uniquely identifies each message and is in binary format. A response message uses the same Message ID as the corresponding request message.

4.1.3. DLC

The Data Length Code is used to specify the number of bytes within the data payload field of the message. Valid values range between zero and 128 (inclusive).

4.1.4. Data Payload

This field contains the actual data sent between the local and remote units and can contain anywhere between zero and 128 bytes of data. The number of bytes in this field is dictated by the DLC field.

4.1.5. CRC

An 8-bit CRC is used and has a polynomial of 0x8C (CRC-8), with the accumulator being initialised to zero before the CRC calculation begins. The following source code indicates the required implementation:

crc: The current CRC.
Ch: Value to add to the CRC calculation.

```
Void crc8PushByte(uint8_t *crc, uint8_t ch)
{
   uint8_t i;

   *crc = *crc ^ ch;
   for (i=0; i<8; i++)
   {
      if (*crc & 1)
      {
         *crc = (*crc >> 1) ^0x8C;
      }
      else
      {
         *crc = (*crc >> 1);
      }
   }
}
```

pcrc: Pointer to running CRC to update (set this to something before first call).
Block: Pointer to the.block of data to push through.
Count: The number of bytes to push.
Return: The computed CRC (result also updated in pcrc if that is non-NULL).

```
Uint8_t crc8PushBlock(uint8_t *pcrc, uint8_t *block, uint16_t count)
{
   uint8_t crc = pcrc ? *pcrc : 0;

   for (; count>0; --count,block++)
   {
      crc8PushByte(&crc, *block);
   }
   if (pcrc) *pcrc = crc;
   return crc;
}
```

For example, if the CRC has to be calculated over a block of 10 bytes of data (starting from the start of the buffer) the following function call could be used:

```
crc = crc8PushBlock( NULL, &dataBuffer[0], 10 );
```

4.1.6. ETX

The ETX field (End of Text) is a standard ASCII control character (0x03) and denotes the end of the message. Note that although this is the last field within the request message, the response message has no ETX value, instead using an ACK/NAK in its place.

4.1.7. ACK/NAK

This field is only present in the response message and indicates whether the unit found errors within the request message or couldn't perform the requested operation. If all data within the request was found to be valid and the unit could perform the requested operation, the response will have an ACK appended (standard ASCII control character, value = 0x06); otherwise a NAK is appended (standard ASCII control character, value = 0x15) – see section "Fixed Message Types" for more details on the NAK packet.

4.2. Reception Timeout

Although a general reception timeout is not utilized within the link processing, an inter-byte timeout is required (for direct links, e.g. to a PC), with the timeout period of 100ms being adequate. If there has been no data received within this period, reception within the unit should be reset, forcing a search for the beginning of a message again (the STX character). Although the response sent by the unit depends on the requested operation, it is recommended that a fixed timeout is used by the remote unit of 500ms (for direct PC links).

4.3. Transmission Mediums

The protocol can be used to transfer data over either a direct link to a PC via USB or via Bluetooth.

4.3.1. Direct Links

Given the example of a direct link to a PC via a virtual serial port via USB, the protocol can be used without being encapsulated in any other protocols.

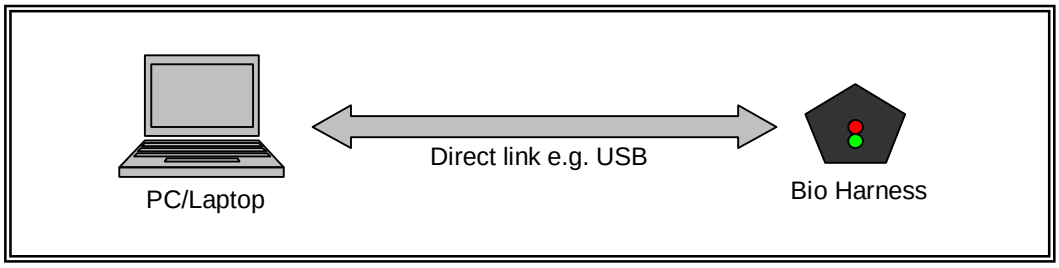

Figure 4-2 Direct Communications Example

In the figure shown above, a direct link between a PC and a Bio Harness can be implemented and requires no modification to the protocol described in this document. The link is point-to-point and therefore no unit addressing is required. Because the link is direct, the only delays between the request and response messages will be due to the amount of processing the receiver has to perform.

4.3.2. Wireless Links

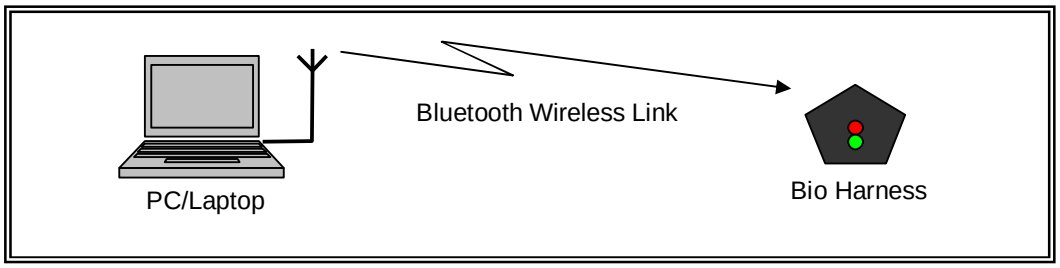

Figure 4-3 Wireless Communications Example

The figure above shows a wireless link using Bluetooth. As Bluetooth provides direct links between units on a PAN (Personal Area Network), the packets sent are generally point-to-point and therefore again there is no requirement to use addressing. The inherent addressing scheme of Bluetooth takes care of it for us. There may be several messages wrapped within one Bluetooth packet and therefore this must be taken into account.

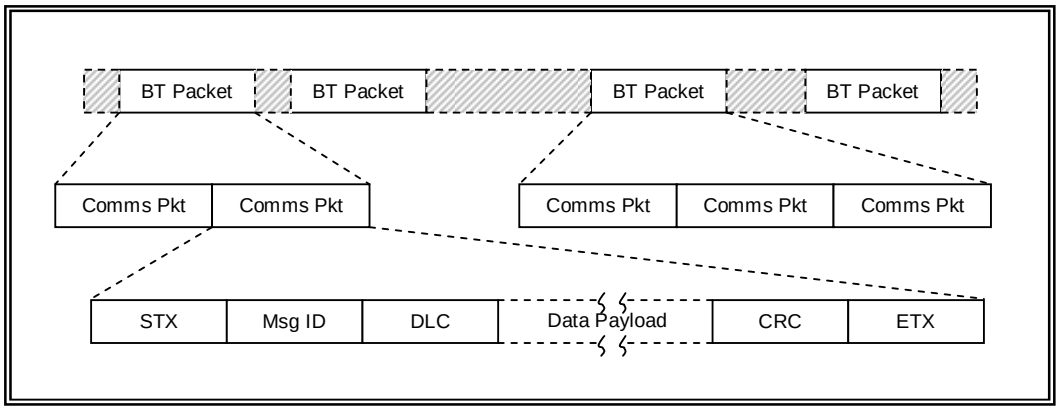

Figure 4-4 RF Packet Encapsulation

The figure above shows that when sending standard communications packets over a Bluetooth link, the possibility must be taken into account that multiple communication packets (specified in this document) may be concatenated into one Bluetooth packet and sent to the remote unit. Therefore the message parsing method employed by the receiver must be able to extract one or several communications packets from within a Bluetooth packet.

5. Standard Messages

This section specifies request/response messages that are implemented in the BioHarness, valid range of values and approximate response timeouts. Both the request and response formats are specified.

5.1. MSG:0x01 - Read Logging Data

5.1.1. Request Message

This message requests logged data from the Bio Harness. The offset within the log and the number of bytes to be read is specified.

Byte/Bit	7	6	5	4	3	2	1	0	Field
0	STX								STX
1	0x01								Msg ID
2	6								DLC
3	File Number								Payload
4	Offset (LS Byte)								
5									
6									
7	Offset (MS Byte)								
8	Number of bytes to read								
9	CRC								CRC
10	ETX								ETX

Table 5-1 ***Request: Read Data from the Log***

- The File Number is used to reference a particular log file in the log. At present, only one log file is stored, but there may be a need to use more than one file in the future. At present however, the value zero can be used in this field.
- The Offset is a 32-bit binary value in Little Endian format. The LS byte is located first within the message. The minimum value is zero and the maximum is one less than the file size. If a request is made to read data beyond the end of the file, a NAK response will be issued.
- The number of bytes to read must be either 1 or a multiple of 2. The minimum value is 1 and the maximum is 128. Odd values other than 1 are not permitted.

5.1.2. Response Message

Byte/Bit	7	6	5	4	3	2	1	0	Field
0	STX								STX
1	0x01								Msg ID
2	Number of bytes read								DLC
3	Data read from the Log								Payload
3+DLC	CRC								CRC
4+DLC	ACK								ACK/NAK

Table 5-2 ***Response: Read Data from the Log***

- The number of bytes read (DLC) is the same as the requested number of bytes in the request message.
- The data payload could be anywhere between 1 and 128 bytes in length.
- An ACK is appended to the message.
- If the data couldn't be read from the Log for any reason, a NAK is sent in response.

5.2. MSG:0x02 - Delete Log File

5.2.1. Request Message

This message requests that the Bio Harness deletes a log file.

Byte/Bit	7	6	5	4	3	2	1	0	Field
0	STX								STX
1	0x02								Msg ID
2	1								DLC
3	File Number								Payload
4	CRC								CRC
5	ETX								ETX

Table 5-3 ***Request: Delete Log File***

- The File Number is used to reference a particular log file in the log. At present, only one log file is stored, but we may use more in the future. At the moment, the value zero can be used in this field.

5.2.2. Response Message

Byte/Bit	7	6	5	4	3	2	1	0	Field
0	STX								STX
1	0x02								Msg ID
2	0								DLC
3	CRC								CRC
4	ACK								ACK/NAK

Table 5-4 ***Response: Delete Log File***

- The response simply indicates to the PC that the log file has been deleted.
- If the file couldn't be deleted, a NAK is sent.

5.3. MSG:0x07 - Set RTC Date/Time

5.3.1. Request Message

This message sets the date and time within the remote unit.

Byte/Bit	7	6	5	4	3	2	1	0	Field
0	STX								STX
1	0x07								Msg ID
2	7								DLC
3	Day								Payload
4	Month								
5	Year (LS Byte)								
6	Year (MS Byte)								
7	Hours								
8	Minutes								
9	Seconds								
10	CRC								CRC
11	ETX								ETX

Table 5-5 ***Request: Set RTC Date/Time***

- The Day field is the day of the month, valid values are 1…31 inclusive.
- The Month field is the month of the year, valid values are 1…12 inclusive.
- Year is in Little Endian format (LS byte first), valid values are 2000…2099 inclusive.
- Hours is the hour of the day, valid values are 0…23 inclusive.
- Minutes is the minute of the hour, valid values are 0…59 inclusive.
- Seconds is the second of the minute, valid values are 0…59 inclusive.
- If any of the payload values are found to be invalid, a NAK is sent back to the PC.

5.3.2. Response Message

Byte/Bit	7	6	5	4	3	2	1	0	Field
0	STX								STX
1	0x07								Msg ID
2	0								DLC
3	CRC								CRC
4	ACK								ACK/NAK

Table 5-6 ***Response: Set RTC Date/Time***

- There is no data payload within the message as the packet merely acknowledges that the data was accepted and the RTC in the remote unit was updated.

5.4. MSG:0x08 - Get RTC Date/Time

5.4.1. Request Message

This message requests the current date and time within the remote unit.

Byte/Bit	7	6	5	4	3	2	1	0	Field
0	STX								STX
1	0x08								Msg ID
2	0								DLC
3	CRC								CRC
4	ETX								ETX

Table 5-7 ***Request: Get RTC Date/Time***

- There is no data payload within the message as it is simply a command to get the current date and time from the remote unit.

5.4.2. Response Message

Byte/Bit	7	6	5	4	3	2	1	0	Field
0	STX								STX
1	0x08								Msg ID
2	7								DLC
3	Day								Payload
4	Month								
5	Year (LS Byte)								
6	Year (MS Byte)								
7	Hours								
8	Minutes								
9	Seconds								
10	CRC								CRC
11	ACK								ACK/NAK

Table 5-8 ***Response: Get RTC Date/Time***

- The Day field is the day of the month, valid values are 1…31 inclusive.
- The Month field is the month of the year, valid values are 1…12 inclusive.
- Year is in Little Endian format (LS byte first), valid values are 2000…2099 inclusive.
- Hours is the hour of the day, valid values are 0…23 inclusive.
- Minutes is the minute of the hour, valid values are 0…59 inclusive.
- Seconds is the second of the minute, valid values are 0…59 inclusive.

5.5. MSG:0x09 - Get Boot Software Version

5.5.1. Request Message

This message requests the Bootloader software version from the remote unit.

Byte/Bit	7	6	5	4	3	2	1	0	Field
0	STX								STX
1	0x09								Msg ID
2	0								DLC
3	CRC								CRC
4	ETX								ETX

Table 5-9 ***Request: Get Boot Software Version***

- There is no data payload within the message as it is simply a command to get the Boot Software Version from the remote unit.

5.5.2. Response Message

Byte/Bit	7	6	5	4	3	2	1	0	Field
0	STX								STX
1	0x09								Msg ID
2	8								DLC
3	Major Version (LS Byte)								Payload
4	Major Version (MS Byte)								
5	Minor Version (LS Byte)								
6	Minor Version (MS Byte)								
7	Reserved (LS Byte)								
8	Reserved (MS Byte)								
9	Build Number (LS Byte)								
10	Build Number (MS Byte)								
11	CRC								CRC
12	ACK								ACK/NAK

Table 5-10 ***Response: Get Boot Software Version***

- The Major Version field is in binary and in Little Endian format (LS Byte first).
- The Minor Version field is in binary and in Little Endian format (LS Byte first).
- The Reserved field is not currently used and should be set to zero.
- The Build Number is in binary and in Little Format format (LS Byte first).

5.6. MSG:0x0A - Get Application Software Version

5.6.1. Request Message

This message requests the Application software version from the remote unit.

Byte/Bit	7	6	5	4	3	2	1	0	Field
0	STX								STX
1	0x0A								Msg ID
2	0								DLC
3	CRC								CRC
4	ETX								ETX

Table 5-11 ***Request: Get Application Software Version***

- There is no data payload within the message as it is simply a command to get the Application Software Version from the remote unit.

5.6.2. Response Message

Byte/Bit	7	6	5	4	3	2	1	0	Field
0	STX								STX
1	0x0A								Msg ID
2	8								DLC
3	Major Version (LS Byte)								Payload
4	Major Version (MS Byte)								
5	Minor Version (LS Byte)								
6	Minor Version (MS Byte)								
7	Reserved (LS Byte)								
8	Reserved (MS Byte)								
9	Build Number (LS Byte)								
10	Build Number (MS Byte)								
11	CRC								CRC
12	ACK								ACK/NAK

Table 5-12 ***Response: Get Application Software Version***

- The Major Version field is in binary and in Little Endian format (LS Byte first).
- The Minor Version field is in binary and in Little Endian format (LS Byte first).
- The Reserved field is not currently used and should be set to zero.
- The Build Number is in binary and in Little Format format (LS Byte first).

5.7. MSG:0x0B - Get Serial Number

5.7.1. Request Message

This message requests the Serial Number from the remote unit.

Byte/Bit	7	6	5	4	3	2	1	0	Field
0	STX								STX
1	0x0B								Msg ID
2	0								DLC
3	CRC								CRC
4	ETX								ETX

Table 5-13 ***Request: Get Serial Number***

- There is no data payload within the message as it is simply a command to get the Serial Number from the remote unit.

5.7.2. Response Message

Byte/Bit	7	6	5	4	3	2	1	0	Field
0	STX								STX
1	0x0B								Msg ID
2	12								DLC
3	Serial Number String (ASCII) – 1st Byte								Payload
4									
5									
6									
7									
8									
9									
10									
11									
12									
13									
14	Serial Number String (ASCII) – 12th Byte								
15	CRC								CRC
16	ACK								ACK/NAK

Table 5-14 ***Response: Get Serial Number***

- The Serial Number string is in ASCII and is 12 characters in length e.g. “ZBH123456”.
- There is no NULL terminator at the end of the string; thus the receiving application must append a NULL to the string data.
- If the serial number is less than 12 bytes in length, the end of the string is padded with spaces.

5.8. MSG:0x0C - Get Hardware Part Number

5.8.1. Request Message

This message requests the Hardware Part Number from the remote unit.

Byte/Bit	7	6	5	4	3	2	1	0	Field
0	STX								STX
1	0x0C								Msg ID
2	0								DLC
3	CRC								CRC
4	ETX								ETX

Table 5-15 ***Request: Get Hardware Part Number***

- There is no data payload within the message as it is simply a command to get the Hardware Part Number from the remote unit.

5.8.2. Response Message

Byte/Bit	7	6	5	4	3	2	1	0	Field
0	STX								STX
1	0x0C								Msg ID
2	12								DLC
3	Hardware Part Number String (ASCII) – 1st Byte								Payload
4									
5									
6									
7									
8									
9									
10									
11									
12									
13									
14	Hardware Part Number String (ASCII) – 12th Byte								
15	CRC								CRC
16	ACK								ACK/NAK

Table 5-16 ***Response: Get Hardware Part Number***

- The Hardware Part Number string is in ASCII and is 12 characters in length e.g. “9900.0085v1a”.
- There is no NULL terminator at the end of the string; thus the receiving application must append a NULL at the end of the string data.
- If the part number is less than 12 bytes in length, the end of the string is padded with spaces.

5.9. MSG:0x0D - Get Boot Part Number

5.9.1. Request Message

This message requests the Boot Part Number from the remote unit.

Byte/Bit	7	6	5	4	3	2	1	0	Field
0	STX								STX
1	0x0D								Msg ID
2	0								DLC
3	CRC								CRC
4	ETX								ETX

Table 5-17 ***Request: Get Boot Part Number***

- There is no data payload within the message as it is simply a command to get the Boot Part Number from the remote unit.

5.9.2. Response Message

Byte/Bit	7	6	5	4	3	2	1	0	Field
0	STX								STX
1	0x0D								Msg ID
2	12								DLC
3	Boot Part Number String (ASCII) – 1st Byte								Payload
4									
5									
6									
7									
8									
9									
10									
11									
12									
13									
14	Boot Part Number String (ASCII) – 12th Byte								
15	CRC								CRC
16	ACK								ACK/NAK

Table 5-18 ***Response: Get Boot Part Number***

- The Boot Part Number string is in ASCII and is 12 characters in length e.g. “9500.0001”.
- There is no NULL terminator at the end of the string; thus the receiving application must append a NULL to the end of the string data.
- If the part number is less than 12 bytes in length, the end of the string is padded with spaces.

5.10. MSG:0x0E - Get Application Part Number

5.10.1. Request Message

This message requests the Application Part Number from the remote unit.

Byte/Bit	7	6	5	4	3	2	1	0	Field
0	STX								STX
1	0x0E								Msg ID
2	0								DLC
3	CRC								CRC
4	ETX								ETX

Table 5-19 ***Request: Get Application Part Number***

- There is no data payload within the message as it is simply a command to get the Application Part Number from the remote unit.

5.10.2. Response Message

Byte/Bit	7	6	5	4	3	2	1	0	Field
0	STX								STX
1	0x0E								Msg ID
2	12								DLC
3	Application Part Number String (ASCII) – 1st Byte								Payload
4									
5									
6									
7									
8									
9									
10									
11									
12									
13									
14	Application Part Number String (ASCII) – 12th Byte								
15	CRC								CRC
16	ACK								ACK/NAK

Table 5-20 ***Response: Get Application Part Number***

- The Application Part Number string is in ASCII and is 12 characters in length e.g. “9500.0002”.
- There is no NULL terminator at the end of the string; thus the receiving application must append a NULL to the end of the string data.
- If the part number is less than 12 bytes in length, the end of the string is padded with spaces.

5.11. MSG:0x10 - Set Network ID

5.11.1. Request Message

This message sets the Bluetooth Network ID within the device. Note that the Network ID makes up part of the Bluetooth Friendly name:

"<Unit Type> <Network ID>"

e.g.

"BH John Smith 5"

Contains a unit type of **BH** (Bio Harness) and a Network ID of **John Smith 5**.

Byte/Bit	7	6	5	4	3	2	1	0	Field
0	STX								STX
1	0x10								Msg ID
2	Number of characters in string.								DLC
3	Bluetooth Network ID (ASCII) – 1st Byte								Payload
:									
:									
:									
:									
:									
DLC+3	Bluetooth Network ID (ASCII) – Last Byte								
DLC+4	CRC								CRC
DLC+5	ETX								ETX

Table 5-21 ***Request: Set Network ID***

- The Network ID string is in ASCII.
- The string has a minimum of 2 a maximum of 29 characters in length.
- There is no NULL terminator at the end of the string.

5.11.2. Response Message

Byte/Bit	7	6	5	4	3	2	1	0	Field
0	STX								STX
1	0x10								Msg ID
2	0								DLC
3	CRC								CRC
4	ACK								ACK/NAK

Table 5-22 ***Response: Set Network ID***

- There is no data payload within the message as the packet merely acknowledges that the remote unit accepted the new Bluetooth Network ID.

5.12. MSG:0x11 - Get Network ID

5.12.1. Request Message

This message gets the Bluetooth Network ID from the device. Note that the Network ID makes up part of the Bluetooth Friendly name:

"**<Unit Type> <Network ID>**"

e.g.

"**BH John Smith 5**"

Contains a unit type of **BH** (Bio Harness) and a Network ID of **John Smith 5**.

Byte/Bit	7	6	5	4	3	2	1	0	Field
0	STX								STX
1	0x11								Msg ID
2	0								DLC
3	CRC								CRC
4	ETX								ETX

Table 5-23 ***Request: Get Network ID***

- There is no data payload within the message as it is simply a command to get the Bluetooth Network ID from the remote unit.

5.12.2. Response Message

Byte/Bit	7	6	5	4	3	2	1	0	Field
0	STX								STX
1	0x11								Msg ID
2	Number of characters in string.								DLC
3	Bluetooth Network ID (ASCII) – 1st Byte								Payload
:									
:									
:									
:									
:									
DLC+3	Bluetooth Network ID (ASCII) – Last Byte								
DLC+4	CRC								CRC
DLC+5	ACK								ACK/NAK

Table 5-24 ***Response: Get Network ID***

- The Bluetooth string is in ASCII.
- The string has a minimum of 2 a maximum of 29 characters in length.
- There is no NULL terminator at the end of the string.

5.13. MSG:0x12 - Get Unit MAC Address

5.13.1. Request Message

This message gets the Bluetooth MAC address from the device.

Byte/Bit	7	6	5	4	3	2	1	0	Field
0	STX								STX
1	0x12								Msg ID
2	0								DLC
3	CRC								CRC
4	ETX								ETX

Table 5-25 ***Request: Get Unit MAC Address***

- There is no data payload within the message as it is simply a command to get the Bluetooth MAC Address from the device.

5.13.2. Response Message

Byte/Bit	7	6	5	4	3	2	1	0	Field
0	STX								STX
1	0x12								Msg ID
2	17								DLC
3	Bluetooth MAC Address String (ASCII) – 1st Byte								Payload
:									
:									
:									
:									
:									
19	Bluetooth MAC Address String (ASCII) – 17th Byte								
20	CRC								CRC
21	ACK								ACK/NAK

Table 5-26 ***Response: Get Unit MAC Address***

- The MAC Address string is in ASCII.
- The string has a fixed number of characters (17) e.g. "**00:07:80:82:7A:61**".
- There is no NULL terminator at the end of the string and therefore it is the responsibility of the receiving application to append a NULL to the string data.

5.14. MSG:0x13 - Get Bio Harness MAC Address

5.14.1. Request Message

This message gets the Bio Harness Bluetooth MAC address from the device. Each device connected to a Bio Harness over a Bluetooth link stores its MAC address; certain implementations involving a Bio Harness require that the Bio Harness is not visible in a Bluetooth network inquiry, therefore its MAC address (required to connect) must be retrieved from another unit on the network. When an RID responds with a MAC address of all 00's, this indicates that the RID is configured as a gateway.

Byte/Bit	7	6	5	4	3	2	1	0	Field
0	STX								STX
1	0x13								Msg ID
2	0								DLC
3	CRC								CRC
4	ETX								ETX

Table 5-27 ***Request: Get Bio Harness MAC Address***

- There is no data payload within the message as it is simply a command to get the Bio Harness's Bluetooth MAC Address from the device.

5.14.2. Response Message

Byte/Bit	7	6	5	4	3	2	1	0	Field
0	STX								STX
1	0x13								Msg ID
2	17								DLC
3	Bio Harness Bluetooth MAC Address String (ASCII) – 1st Byte								Payload
:									
:									
:									
:									
:									
19	Bio Harness Bluetooth MAC Address String (ASCII) – 17th Byte								
20	CRC								CRC
21	ACK								ACK/NAK

Table 5-28 ***Response: Get Bio Harness Unit MAC Address***

- The MAC Address string is in ASCII.
- The string has a fixed number of characters (17) e.g. "**00:07:80:82:7A:61**".
- There is no NULL terminator at the end of the string and therefore it is the responsibility of the receiving application to append a NULL to the string data.

5.15. MSG:0x14 - Set General Data Packet Transmit State

5.15.1. Request Message

This message requests that the Bio Harness enables or disables the Bluetooth transmission of the periodic General Data Packet

Byte/Bit	7	6	5	4	3	2	1	0	Field
0	STX								STX
1	0x14								Msg ID
2	1								DLC
3	Transmission State								Payload
4	CRC								CRC
5	ETX								ETX

Table 5-29 ***Request: Set General Data Packet Transmit State***

- This command enables (payload = 1) or disables (payload = 0) the General Data Packet transmission. When enabled, the general data packet is transmitted periodically (every 1.008 seconds).

5.15.2. Response Message

Byte/Bit	7	6	5	4	3	2	1	0	Field
0	STX								STX
1	0x14								Msg ID
2	0								DLC
3	CRC								CRC
4	ACK								ACK/NAK

Table 5-30 ***Response: Set General Data Packet Transmit State***

- There is no data payload within the message as the packet merely acknowledges that the data was accepted and that the remote unit has set the transmission state to the requested mode.

5.16. MSG:0x15 - Set Breathing Waveform Packet Transmit State

5.16.1. Request Message

This message requests that the Bio Harness enables or disables the Bluetooth transmission of the periodic Breathing Waveform Packet.

Byte/Bit	7	6	5	4	3	2	1	0	Field
0	STX								STX
1	0x15								Msg ID
2	1								DLC
3	Transmission State								Payload
4	CRC								CRC
5	ETX								ETX

Table 5-31 ***Request: Set Breathing Waveform Packet Transmit State***

- This command enables (payload = 1) or disables (payload = 0) the Breathing Waveform Packet transmission. When enabled, the packet is transmitted periodically (every 1.008 seconds).

5.16.2. Response Message

Byte/Bit	7	6	5	4	3	2	1	0	Field
0	STX								STX
1	0x15								Msg ID
2	0								DLC
3	CRC								CRC
4	ACK								ACK/NAK

Table 5-32 ***Response: Set Breathing Waveform Packet Transmit State***

- There is no data payload within the message as the packet merely acknowledges that the data was accepted and that the remote unit has set the transmission state to the requested mode.

5.17. MSG:0x16 - Set ECG Waveform Packet Transmit State

5.17.1. Request Message

This message requests that the Bio Harness enables or disables the Bluetooth transmission of the periodic ECG Waveform Packet.

Byte/Bit	7	6	5	4	3	2	1	0	Field
0	STX								STX
1	0x16								Msg ID
2	1								DLC
3	Transmission State								Payload
4	CRC								CRC
5	ETX								ETX

Table 5-33 ***Request: Set ECG Waveform Packet Transmit State***

- This command enables (payload = 1) of disables (payload = 0) the ECG Waveform Packet transmission. When enabled, the packet is transmitted periodically (every 252ms).

5.17.2. Response Message

Byte/Bit	7	6	5	4	3	2	1	0	Field
0	STX								STX
1	0x16								Msg ID
2	0								DLC
3	CRC								CRC
4	ACK								ACK/NAK

Table 5-34 ***Response: Set ECG Waveform Packet Transmit State***

- There is no data payload within the message as the packet merely acknowledges that the data was accepted and that the remote unit has set the transmission state to the requested mode.

5.18. MSG:0x17 - Get Unit Bluetooth Friendly Name

5.18.1. Request Message

This message gets the Bluetooth 'Friendly Name' from the device. The Bluetooth name consists of the Unit Type and the Network ID:

"**<Unit Type> <Network ID>**"

e.g. Friendly Name:

"**BH John Smith 5**"

Contains a unit type of **BH** (Bio Harness) and a Network ID of **John Smith 5**.

Byte/Bit	7	6	5	4	3	2	1	0	Field
0	STX								STX
1	0x17								Msg ID
2	0								DLC
3	CRC								CRC
4	ETX								ETX

Table 5-35 ***Request: Get Bluetooth Friendly Name***

- There is no data payload within the message as it is simply a command to get the Bluetooth Friendly Name from the remote unit.

5.18.2. Response Message

Byte/Bit	7	6	5	4	3	2	1	0	Field
0	STX								STX
1	0x17								Msg ID
2	Number of characters in string.								DLC
3	Bluetooth Friendly Name (ASCII) – 1st Byte								Payload
:									
:									
:									
:									
:									
DLC+3	Bluetooth Friendly Name (ASCII) – Last Byte								
DLC+4	CRC								CRC
DLC+5	ACK								ACK/NAK

Table 5-36 ***Response: Get Bluetooth Friendly Name***

- The Bluetooth string is in ASCII.
- The string has a minimum of 4 a maximum of 32 characters in length.
- There is no NULL terminator at the end of the string.

5.19. MSG:0x19 - Set R to R Data Packet Transmit State

5.19.1. Request Message

This message requests that the Bio Harness enables or disables the Bluetooth transmission of the periodic R to R Data Packet.

Byte/Bit	7	6	5	4	3	2	1	0	Field
0	STX								STX
1	0x19								Msg ID
2	1								DLC
3	Transmission State								Payload
4	CRC								CRC
5	ETX								ETX

Table 5-37 ***Request: Set R to R Data Packet Transmit State***

- This command enables (payload = 1) of disables (payload = 0) the R to R Data Packet transmission. When enabled, the packet is transmitted periodically (every 1008ms).

5.19.2. Response Message

Byte/Bit	7	6	5	4	3	2	1	0	Field
0	STX								STX
1	0x19								Msg ID
2	0								DLC
3	CRC								CRC
4	ACK								ACK/NAK

Table 5-38 ***Response: Set R to R Waveform Packet Transmit State***

- There is no data payload within the message as the packet merely acknowledges that the data was accepted and that the remote unit has set the transmission state to the requested mode.

5.20. MSG:0x1E - Set Accelerometer Packet Transmit State

5.20.1. Request Message

This message requests that the Bio Harness enables or disables the Bluetooth transmission of the periodic Accelerometer Packet.

Byte/Bit	7	6	5	4	3	2	1	0	Field
0	STX								STX
1	0x1E								Msg ID
2	1								DLC
3	Transmission State								Payload
4	CRC								CRC
5	ETX								ETX

Table 5-39 ***Request: Set Accelerometer Packet Transmit State***

- This command enables (payload = 1) of disables (payload = 0) the Accelerometer Packet transmission. When enabled, the packet is transmitted periodically (every 252ms).

5.20.2. Response Message

Byte/Bit	7	6	5	4	3	2	1	0	Field
0	STX								STX
1	0x1E								Msg ID
2	0								DLC
3	CRC								CRC
4	ACK								ACK/NAK

Table 5-40 ***Response: Set Accelerometer Packet Transmit State***

- There is no data payload within the message as the packet merely acknowledges that the data was accepted and that the remote unit has set the transmission state to the requested mode.

5.21. MSG:0x9B - Set ROG Settings

5.21.1. Request Message

This message sets the ROG thresholds that the device uses for the ROG algorithm.

Byte/Bit	7	6	5	4	3	2	1	0	Field
0	STX								STX
1	0x9B								Msg ID
2	16								DLC
3	Heart Rate High (0…240) – LS Byte								Payload
4	Heart Rate High (0…240) – MS Byte								
5	Heart Rate Low (0…240) – LS Byte								
6	Heart Rate Low (0…240) – MS Byte								
7	Respiration Rate High (0…70) – LS Byte								
8	Respiration Rate High (0…70) – MS Byte								
9	Respiration Rate Low (0…70) – LS Byte								
10	Respiration Rate Low (0…70) – MS Byte								
11	Activity High (0…3.3) – LS Byte								
12	Activity High (0…3.3) – MS Byte								
13	Activity Low (0…3.3) – LS Byte								
14	Activity Low (0…3.3) – MS Byte								
15	Green to Orange Time – LS Byte								
16	Green to Orange Time – MS Byte								
17	Orange to Red Time – LS Byte								
18	Orange to Red Time – MS Byte								
19	CRC								CRC
20	ETX								ETX

Table 5-67 ***Request: Set ROG Settings***

- The data payload has the following specifications:
  - Heart Rate: 0…240 with 1 unit resolution e.g. 132 = 132 BPM
  - Respiration Rate: 0…70 with 0.1 unit resolution e.g. 173 = 17.3 BPM
  - Activity: 0…3.3 with 0.01 unit resolution e.g. 256 = 2.56 VMU
  - The times are in seconds.

5.21.2. Response Message

Byte/Bit	7	6	5	4	3	2	1	0	Field
0	STX								STX
1	0x9B								Msg ID
2	0								DLC
3	CRC								CRC
4	ACK								ACK/NAK

Table 5-68 ***Response: Set ROG Settings***

- There is no data payload within the message as it simply acknowledges that the ROG Settings were stored.

5.22. MSG:0x9C - Get ROG Settings

5.22.1. Request Message

This message gets the ROG thresholds that the device uses for the ROG algorithm.

Byte/Bit	7	6	5	4	3	2	1	0	Field
0	STX								STX
1	0x9C								Msg ID
2	0								DLC
3	CRC								CRC
4	ETX								ETX

Table 5-69 ***Request: Get ROG Settings***

- There is no data payload within the message as it is simply a command to get the ROG Settings from the device.

5.22.2. Response Message

Byte/Bit	7	6	5	4	3	2	1	0	Field
0	STX								STX
1	0x9C								Msg ID
2	16								DLC
3	Heart Rate High (0…240) – LS Byte								Payload
4	Heart Rate High (0…240) – MS Byte								
5	Heart Rate Low (0…240) – LS Byte								
6	Heart Rate Low (0…240) – MS Byte								
7	Respiration Rate High (0…70) – LS Byte								
8	Respiration Rate High (0…70) – MS Byte								
9	Respiration Rate Low (0…70) – LS Byte								
10	Respiration Rate Low (0…70) – MS Byte								
11	Activity High (0…3.3) – LS Byte								
12	Activity High (0…3.3) – MS Byte								
13	Activity Low (0…3.3) – LS Byte								
14	Activity Low (0…3.3) – MS Byte								
15	Green to Orange Time – LS Byte								
16	Green to Orange Time – MS Byte								
17	Orange to Red Time – LS Byte								
18	Orange to Red Time – MS Byte								
19	CRC								CRC
20	ACK								ACK/NAK

Table 5-70 ***Response: Get ROG Settings***

- The data payload has the following specifications:
  - Heart Rate: 0…240 with 1 unit resolution e.g. 132 = 132 BPM
  - Respiration Rate: 0…70 with 0.1 unit resolution e.g. 173 = 17.3 BPM
  - Activity: 0…3.3 with 0.01 unit resolution e.g. 256 = 2.56 VMU
  - The times are in seconds.

5.23. MSG:0xA2 - Set Bluetooth User Config

5.23.1. Request Message

This message sets the user configurable Bluetooth settings.

Byte/Bit	7	6	5	4	3	2	1	0	Field
0	STX								STX
1	0xA2								Msg ID
2	Payload Length								DLC
3	UC1- Device Discoverable – LS Byte								Payload
4	UC1 -Device Discoverable – MS Byte								
DLC+3	CRC								CRC
DLC+4	ETX								ETX

Table 5-67 ***Request: Set Bluetooth User Config***

- Payload Length
  - Length of data within message.
  - Length is 2 in this example but can grow up to 30 bytes when more Bluetooth settings are configurable.
- UCx
  - User Config Settings (UC1 – UC15)
  - UC1 Device Discoverable
    - 0 Device not visible in inquiry.
    - 1 Device visible in inquiry
  - UC2 – UC15 Reserved for future use.

5.23.2. Response Message

Byte/Bit	7	6	5	4	3	2	1	0	Field
0	STX								STX
1	0xA2								Msg ID
2	0								DLC
3	CRC								CRC
4	ACK/NAK								ACK/NAK

Table 5-68 ***Response: Set Bluetooth User Config***

- There is no data payload within the message as it simply acknowledges that the Bluetooth Settings were stored.

5.24. MSG:0xA3 - Get Bluetooth User Config

5.24.1. Request Message

This message requests the user configurable Bluetooth settings.

Byte/Bit	7	6	5	4	3	2	1	0	Field
0	STX								STX
1	0xA3								Msg ID
2	0								DLC
3	CRC								CRC
4	ETX								ETX

Table 5-69 ***Request: Get Bluetooth User Config***

- There is no data payload within the message as it is simply a command to get the user configurable Bluetooth settings from the device.

5.24.2. Response Message

Byte/Bit	7	6	5	4	3	2	1	0	Field
0	STX								STX
1	0xA3								Msg ID
2	Payload Length								DLC
3	UC1- Device Discoverable – LS Byte								Payload
4	UC1 -Device Discoverable – MS Byte								
DLC+3	CRC								CRC
DLC+4	ACK/NAK								ACK/NAK

Table 5-70 ***Response: Get Bluetooth User Config***

- Payload Length
  - Length of data within message.
  - Length is 2 in this example but can grow up to 30 bytes when more Bluetooth settings are configurable.
- UCx
  - User Config Settings (UC1 – UC15)
  - UC1 Device Discoverable
    - 0 Device not visible in inquiry.
    - 1 Device visible in inquiry
  - UC2 – UC15 Reserved for future use

5.25. MSG:0XA4 - Set BT Link Config

5.25.1. Request Message

This message sets the timeout & lifesign periods for a Bluetooth link.

Byte/Bit	7	6	5	4	3	2	1	0	Field
0	STX								STX
1	0xA4								Msg ID
2	4								DLC
3	Link Timeout (ms) – LS Byte								Payload
4	Link Timeout (ms) – MS Byte								
5	Lifesign Period (ms) – LS Byte								
6	Lifesign Period (ms) – MS Byte								
7	CRC								CRC
8	ETX								ETX

Table 5-67 ***Request: Set BT Link Config***

- Link Timeout
  - Link closed if no data received over link within this time in ms.
  - Set to 0 if no timeout required.
- Lifesign Period
  - Time in ms between sending Lifesign messages to remote device.
  - Set to 0 to never send Lifesign messages.

5.25.2. Response Message

Byte/Bit	7	6	5	4	3	2	1	0	Field
0	STX								STX
1	0xA4								Msg ID
2	0								DLC
3	CRC								CRC
4	ACK/NAK								ACK/NAK

Table 5-68 ***Response: Set BT Link Config***

- There is no data payload within the message as it simply acknowledges that the BT Link Settings were stored.

5.26. MSG:0xA5 - Get BT Link Config

5.26.1. Request Message

This message requests the timeout & lifesign periods for a Bluetooth link.

Byte/Bit	7	6	5	4	3	2	1	0	Field
0	STX								STX
1	0xA5								Msg ID
2	0								DLC
3	CRC								CRC
4	ETX								ETX

Table 5-69 ***Request: Get BT Link Config***

- There is no data payload within the message as it is simply a command to get the BT Link Settings from the device.

5.26.2. Response Message

Byte/Bit	7	6	5	4	3	2	1	0	Field
0	STX								STX
1	0xA5								Msg ID
2	Payload Length								DLC
3	Link Timeout (ms) – LS Byte								Payload
4	Link Timeout (ms) – MS Byte								
5	Lifesign Period (ms) – LS Byte								
6	Lifesign Period (ms) – MS Byte								
7	CRC								CRC
8	ACK/NAK								ACK/NAK

Table 5-70 ***Response: Get BT Link Config***

- Link Timeout
  - Link closed if no data received over link within this time in ms.
  - Set to 0 if no timeout required.
- Lifesign Period
  - Time in ms between sending Lifesign messages to remote device.
  - Set to 0 to never send Lifesign messages.

5.27. MSG:0xA6 - Set BioHarness User Config

5.27.1. Request Message

This message sets the BioHarness user configurable settings.

Byte/Bit	7	6	5	4	3	2	1	0	Field
0	STX								STX
1	0xA6								Msg ID
2	Payload Length								DLC
3	Log Enable								Payload
4	RF Enable								
5	Button Enable When Worn								
6	ECG Polarity								
DLC+3	CRC								CRC
DLC+4	ETX								ETX

Table 5-67 ***Request: Set BioHarness User Config***

- Payload Length
  - Length of data within message.
  - Length is 4 in this example but can grow up to 30 bytes when more BioHarness user settings are configurable.
- Log Enable
  - 0 Disable logging of physiological data.
  - 1 Enable logging of physiological data.
- RF Enable
  - 0 Disable RF communications.
  - 1 Enable RF communications.
- Button Enable when worn
  - 0 Button disabled when worn (pressing has no effect).
  - 1 Button enabled when worn.
- ECG Polarity
  - 0 Normal Polarity
  - 1 Reversed Polarity

5.27.2. Response Message

Byte/Bit	7	6	5	4	3	2	1	0	Field
0	STX								STX
1	0xA6								Msg ID
2	0								DLC
3	CRC								CRC
4	ACK/NAK								ACK/NAK

Table 5-68 ***Response: Set BioHarness User Config***

- There is no data payload within the message as it simply acknowledges that the BioHarness user Settings were stored.

5.28. MSG 0xA7 - Get BioHarness User Config

5.28.1. Request Message

This message requests the BioHarness user configurable settings.

Byte/Bit	7	6	5	4	3	2	1	0	Field
0	STX								STX
1	0xA7								Msg ID
2	0								DLC
3	CRC								CRC
4	ETX								ETX

Table 5-69 ***Request: Get BioHarness User Config***

- There is no data payload within the message as it is simply a command to get the BioHarness User Settings from the device.

5.28.2. Response Message

Byte/Bit	7	6	5	4	3	2	1	0	Field
0	STX								STX
1	0xA7								Msg ID
2	Payload Length								DLC
3	Log Enable								Payload
4	RF Enable								
5	Button Enable When Worn								
6	ECG Polarity								
DLC+3	CRC								CRC
DLC+4	ACK/NAK								ACK/NAK

Table 5-70 ***Response: Get BioHarness User Config***

- Payload Length
  - Length of data within message.
  - Length is 3 in this example but can grow up to 30 bytes when more BioHarness user settings are configurable.
- Log Enable
  - 0 Logging of physiological data disabled.
  - 1 Logging of physiological data enabled
- RF Enable
  - 0 Disable RF communications.
  - 1 Enable RF communications.
- Button Enable when worn
  - 0 Button disabled when worn (pressing has no effect).
  - 1 Button enabled when worn.
- ECG Polarity
  - 0 Normal Polarity
  - 1 Reversed Polarity

5.29. MSG:0xAC – Get Battery Status

5.29.1. Request Message

This message requests the current battery voltage in millivolts and the battery charge as a percentage of full charge.

Byte/Bit	7	6	5	4	3	2	1	0	Field
0	STX								STX
1	0xAC								Msg ID
2	0								DLC
4	CRC								CRC
5	ETX								ETX

Table 5-41 ***Request: Get Battery Status***

5.29.2. Response Message

Byte/Bit	7	6	5	4	3	2	1	0	Field
0	STX								STX
1	0xAC								Msg ID
2	3								DLC
3	Battery Voltage (mv) – LS Byte								Payload
4	Battery Voltage (mv) – MS Byte								
5	Battery Charge (%)								
6	CRC								CRC
7	ACK								ACK/NAK

Table 5-42 ***Response: Get Battery Status***

- Current Battery voltage in millivolts (0…max voltage).
- Current Battery Charge in % (0…100).

5.30. MSG:0x1F – Reboot Unit

5.30.1. Request Message

This message requests that the remote unit performs a reboot. As soon as the unit has sent the acknowledgement to the PC, it performs reboot via a watchdog timer reset.

Byte/Bit	7	6	5	4	3	2	1	0	Field
0	STX								STX
1	0x1F								Msg ID
2	7								DLC
3	Verification String (ASCII) – 1st Byte								Payload
4									
5									
6									
7									
8									
9	Verification String (ASCII) – 7th Byte								
10	CRC								CRC
11	ETX								ETX

Table 5-43 ***Request: Reboot Unit***

- The payload should contain a verification string "**ZReBoot**". If the data in the payload received by the unit does not match the string (case sensitive), a NAK is returned as the response.

5.30.2. Response Message

Byte/Bit	7	6	5	4	3	2	1	0	Field
0	STX								STX
1	0x1F								Msg ID
2	0								DLC
3	CRC								CRC
4	ACK								ACK/NAK

Table 5-44 ***Response: Reboot Unit***

- There is no data payload within the message as the packet merely acknowledges that the data was accepted and that the unit will now perform a reboot.
- After the response message has been sent to the PC, the unit reboots after approximately 125ms. The PC will have to wait a further 5 seconds before the unit can continue communications (unless Boot mode communications are required).

5.31. MSG:0xB0 – Bluetooth Peripheral Message

5.31.1. Request Message

This message requests that the remote unit performs the requested action on the specified Bluetooth device.

Byte/Bit	7	6	5	4	3	2	1	0	Field
0	STX								STX
1	0xB0								Msg ID
2	DLC								DLC
3	Sequence ID								Payload
4	Requested Action								
5	Bluetooth MAC Address - 1st Byte								
:	:								
10	Bluetooth MAC Address - 6th Byte								
:	Action Parameters								
DLC+3	CRC								CRC
DLC+4	ETX								ETX

Table 5-45 ***Request: Bluetooth Peripheral Message***

- The Sequence ID is incremented by the sender for every message sent.
- The Bluetooth MAC address is a 6-byte MAC address in binary representation used to identify the device.
- The action Parameters field depends on the Requested Action, see Table 5-46 below.

Action		Parameters		
Name	Code	Byte	Description	Notes
disconnect from device	0x00	none		
connect to device	0x01	0	Device Manufacturer Code (see Table 5-47)	0 .. 255
		1	Device Type Code (see Table 5-47)	0 .. 255
		2,3	Sampling period (see Table 5-48)	
		4,5	Logging period (see Table 5-48)	
		6	Bluetooth PIN (ASCII) - 1st Byte	optional
		:	:	
		n	Bluetooth PIN (ASCII) - nth Byte	
update configuration	0x02	0,1	Sampling period (see Table 5-48)	
		2,3	Logging period (see Table 5-48)	

Table 5-46 ***Actions and their Parameters***

- The Bluetooth PIN is an optional, variable length array of ASCII characters which is not null-terminated, its length can be determined from the DLC of the message.
- If the update configuration action is sent, the device will reset its sampling cycle and send the next sample immediately.

- If the device is unable to sample measurements at the rate specified by the sampling period, then it will send samples at the highest possible rate.

Manufacturer		Device	
Name	**Code**	**Name**	**Code**
Zephyr Technology	0x01	BioHarness	0x01
Nonin Medical	0x02	Nonin 9560 Onyx II SpO2 sensor	0x01
Mytech Technology	0x03	Mytech HPL-108 blood pressure sensor	0x01

Table 5-47 ***Device Manufacturers and Types***

Byte/Bit	7	6	5	4	3	2	1	0
0	Unit (see Table 5-49)				time period value (bits 11 .. 8)			
1	time period value (bits 7 .. 0)							

Table 5-48 ***Sampling and Logging Period***

code	unit
0	milliseconds
1	seconds
2 .. 15	reserved for future use

Table 5-49 ***Unit Codes***

5.31.2. Response Message

Byte/Bit	7	6	5	4	3	2	1	0	Field
0	STX								STX
1	0xB0								Msg ID
2	1								DLC
3	Sequence ID								Payload
4	CRC								CRC
5	ACK/NAK								ACK/NAK

Table 5-50 ***Response: Bluetooth Peripheral Message***

The response message will contain a NAK, if:

- The specified device/manufacturer combination is not known/supported
- Request to connect to a device, but the maximum number of simultaneous connections to the same type of device has already been reached
- Request to close a non-existing connection
- Requested action is not known

Otherwise an ACK response is returned.

The Sequence ID must be identical to the one received in the Request message.

5.32. MSG:0xB3 – Reset Configuration

5.32.1. Request Message

This message is used to reset the device configuration according to the specified mode. This command may also cause the device to be restarted.

Byte/Bit	7	6	5	4	3	2	1	0	Field
0	STX								STX
1	0xB3								Msg ID
2	1								DLC
3	Mode								Payload
4	CRC								CRC
5	ETX								ETX

Table 5-51 ***Request: Reset Configuration***

The Mode field determines how and what aspect of the configuration is reset, according to the following table:

Mode	Description
0	Reset all configuration to factory defaults
1-255	Reserved for future use

5.32.2. Response Message

An ACK response indicates that the command is understood and shall be carried out shortly. If the command or the specified mode is not supported by the command, or for any reason the command cannot be performed, a NAK response is returned.

Byte/Bit	7	6	5	4	3	2	1	0	Field
0	STX								STX
1	0xB3								Msg ID
2	0								DLC
3	CRC								CRC
4	ACK/NAK								ACK/NAK

Table 5-52 ***Response: Reset Configuration***

5.33. MSG:0xB4 - Set Accelerometer Axis Mapping

5.33.1. Request Message

This message sets the axis mapping information for the accelerometer. Any accelerometer axis can be mapped to any other axis as well as being inverted to allow the device to be used in a number of different orientations (e.g. worn on the front, worn on the side, upside down, etc.).

Byte/Bit	7	6	5	4	3	2	1	0	Field
0	STX								STX
1	0xB4								Msg ID
2	3								DLC
3	X-axis mapping (see Table 5-55)								Payload
4	Y-axis mapping (see Table 5-55)								
5	Z-axis mapping (see Table 5-55)								
6	CRC								CRC
7	ETX								ETX

Table 5-53 ***Request: Set Accelerometer Axis Mapping***

Mapping data is provided for each axis to determine which axis it should be mapped to and whether it should be inverted. For example if a BioHarness is used on the right side facing backwards, then the mapping is as follows (with negative sign showing inversion):
(-X) → (Y), (-Y) → (Z), (Z) → (X), encoded according to Table 5-55.

5.33.2. Response Message

Byte/Bit	7	6	5	4	3	2	1	0	Field
0	STX								STX
1	0xB4								Msg ID
2	0								DLC
3	CRC								CRC
4	ACK/NAK								ACK/NAK

Table 5-54 ***Response: Set Accelerometer Offset***

- There is no data payload within the message as it simply acknowledges that the Accelerometer Axis Mapping Settings were stored.
- If the Axis Index is illegal, a NAK is returned.

Byte/Bit	7	6	5	4	3	2	1	0
0	invert	Not used (should be set to 0)					Axis Index	

Table 5-55 **Axis Mapping Data**

Mapping Data	Description
Invert	0 = axis not inverted, 1 = axis inverted
Axis Index	Index of axis to map to: 0 = X Axis, 1 = Y Axis, 2 = Z Axis, 3 = illegal

5.34. MSG 0xB5 - Get Accelerometer Axis Mapping

5.34.1. Request Message

This message reads the accelerometer axis mapping.

Byte/Bit	7	6	5	4	3	2	1	0	Field
0	STX								STX
1	0xB5								Msg ID
2	0								DLC
3	CRC								CRC
4	ETX								ETX

Table 5-56 ***Request: Get Accelerometer Axis Mapping***

- There is no data payload within the message as it is simply a command to get the BioHarness User Settings from the device.

5.34.2. Response Message

Byte/Bit	7	6	5	4	3	2	1	0	Field
0	STX								STX
1	0xB5								Msg ID
2	Payload Length								DLC
3	X-axis mapping (see Table 5-55)								Payload
4	Y-axis mapping (see Table 5-55)								
5	Z-axis mapping (see Table 5-55)								
6	CRC								CRC
7	ACK/NAK								ACK/NAK

Table 5-57 ***Response: Get Accelerometer Axis Mapping***

6. Periodic Data Packets

This section specifies packets which do not require acknowledgement and/or are sent on a periodic basis.

6.1. MSG:0x20 - General Data Packet

This message contains the General Data transmitted by the Bio Harness. Once enabled, the packet is transmitted periodically.

Byte/Bit	7	6	5	4	3	2	1	0	Field
0	STX								STX
1	0x20								Msg ID
2	53								DLC
3	Sequence Number (0…255)								Payload
4	Timestamp – Year (LS Byte)								
5	Timestamp – Year (MS Byte)								
6	Timestamp – Month								
7	Timestamp – Day								
8	Timestamp – Milliseconds of day (LS Byte)								
9	:								
10	:								
11	Timestamp – Milliseconds of day (MS Byte)								
12	Heart Rate (0…240†) – LS Byte								
13	Heart Rate (0…240†) – MS Byte								
14	Respiration Rate (0…70†) – LS Byte								
15	Respiration Rate (0…70†) – MS Byte								
16	Skin Temperature (0…60) – LS Byte								
17	Skin Temperature (0…60) – MS Byte								
18	Posture (-90…90) – LS Byte								
19	Posture (-90…90) – MS Byte								
20	VMU (0…3.3) – LS Byte								
21	VMU(0…3.3) – MS Byte								
22	Peak Acceleration (0…3.3) – LS Byte								
23	Peak Acceleration (0…3.3) – MS Byte								
24	Battery Voltage								
26	Breathing Wave Amplitude								
28	ECG Amplitude†								
30	ECG Noise†								
32	Vertical Axis Acceleration Min								
34	Vertical Axis Acceleration Peak								
36	Lateral Axis Acceleration Min								
38	Lateral Axis Acceleration Peak								
40	Sagittal Axis Acceleration Min								
42	Sagittal Axis Acceleration Peak								
44	Zephyr System Channel								
46	GSR								
48	SPo2								
50	Blood Pressure								
52	ROG								
53	ALARM								
54	Battery Status(see 6.2 below)								
55	Button/Worn(see 6.2 below)								
56	CRC								CRC
57	ETX								ETX

Table 6-1 ***Message: General Data Packet***

†These fields have modified meanings or ranges for canine monitoring. Details can be found in Appendix B.

- Once enabled (by the “Set General Data Packet Transmission State” message), the packet is sent periodically to a remote unit (every 1.008 seconds). **See appendix B** for more details on each data item within this packet.

6.2. General Data packet content details

Byte 55 below

15	14	13	12	11	10	9	8
PMWS	UIBS	BHHSL	BHESC	Unused			

Byte 54 below

7	6	5	4	3	2	1	0
Unused	BPFC						

PMWS Bit 15 Physiological Monitor Worn Status
0 Not worn by user
1 Worn by user

UIBS Bit 14 User Interface Button Status
0 Button not pressed
1 Button Pressed

BHHSL Bit 13 BioHarness Heart-Rate Signal Low
0 BioHarness Heart-Rate Signal Quality is acceptable
1 BioHarness Heart-Rate Signal Quality Low (HR coasting)

BHESC Bit 12 BioHarness External Sensors Connected
0 BioHarness has no external sensors connected
1 BioHarness has external sensors connected

Unused Bits 11-7 Unused

BPFC Bits 6-0 Battery status as Percentage of Full Charge
0 0 % (Battery Cut-off Voltage)
….
100 100% (Battery Fully Charged)

6.3. MSG:0x21 - Breathing Waveform Packet

This message contains the Breathing Waveform data transmitted by the Bio Harness. Once enabled, the packet is transmitted periodically.

Byte/Bit	7	6	5	4	3	2	1	0	Field
0	STX								STX
1	0x21								Msg ID
2	32								DLC
3	Sequence Number (0…255)								Payload
4	Timestamp – Year (LS Byte)								
5	Timestamp – Year (MS Byte)								
6	Timestamp – Month								
7	Timestamp – Day								
8	Timestamp – Milliseconds of day (LS Byte)								
9	:								
10	:								
11	Timestamp – Milliseconds of day (MS Byte)								
12	Breathing Waveform Data (18 Samples) – see “Packing Format”								
35	CRC								CRC
36	ETX								ETX

Table 6-2 ***Message: Breathing Waveform Packet***

- Once enabled (by the “Set Breathing Waveform Packet Transmission State” message), the packet is sent periodically to a remote unit (every 1.008 seconds).
- Each Breathing Waveform sample is separated in time by 56ms.

6.3.1. Packing Format

Each Breathing Waveform sample is bit-packed into the message to minimise the amount of space used. The data is packed in the following format:

Byte/Bit	7	6	5	4	3	2	1	0
0								Bit 0
1						Bit 0	Bit 9	
2					Bit 0	Bit 9		
3			Bit 0	Bit 9				
4	Bit 9							
5	As Byte 0 (pattern repeats every 5 bytes).							

Table 6-3 ***Packing Format for Breathing Waveform Data.***

- Because each sample is 10-bits in length, the data is bit-packed into the message to conserve space.

6.4. MSG:0x22 - ECG Waveform Packet

This message contains the ECG Waveform data transmitted by the Bio Harness. Once enabled, the packet is transmitted periodically.

Byte/Bit	7	6	5	4	3	2	1	0	Field
0	STX								STX
1	0x22								Msg ID
2	88								DLC
3	Sequence Number (0…255)								Payload
4	Timestamp – Year (LS Byte)								
5	Timestamp – Year (MS Byte)								
6	Timestamp – Month								
7	Timestamp – Day								
8	Timestamp – Milliseconds of day (LS Byte)								
9	:								
10	:								
11	Timestamp – Milliseconds of day (MS Byte)								
12	ECG Waveform Data (63 Samples) – see “Packing Format”								
91	CRC								CRC
92	ETX								ETX

Table 6-4 ***Message: ECG Waveform Packet***

- Once enabled (by the “Set ECG Waveform Packet Transmission State” message), the packet is sent periodically to a remote unit (every 252ms). Each ECG Waveform sample is 4ms later than the previous one.

6.4.1. Packing Format

Each ECG Waveform sample is bit-packed into the message to minimise the amount of space used in the message. The data is packed in the following format:

Byte/Bit	7	6	5	4	3	2	1	0
0								Bit 0
1							Bit 0	Bit 9
2						Bit 0	Bit 9	
3					Bit 0	Bit 9		
4	Bit 9							
5	As Byte 0 (pattern repeats every 5 bytes).							

Table 6-5 ***Packing Format for ECG Waveform Data.***

- Because each sample is 10-bits in length, the data is bit-packed into the message to conserve space.

6.5. MSG:0x23 - Lifesign

This message can be sent from either side of a Bluetooth link to indicate to the remote unit that the link is still active (otherwise a timeout will occur).

Byte/Bit	7	6	5	4	3	2	1	0	Field
0	STX								STX
1	0x23								Msg ID
2	0								DLC
3	CRC								CRC
4	ETX								ETX

Table 6-6 ***Message: Lifesign***

- The message has no data content.

6.6. MSG:0x24 - R to R Packet

This message contains the R to R data transmitted by the Bio Harness. Once enabled, the packet is transmitted periodically.

Byte/Bit	7	6	5	4	3	2	1	0	Field
0	STX								STX
1	0x24								Msg ID
2	45								DLC
3	Sequence Number (0…255)								Payload
4	Timestamp – Year (LS Byte)								
5	Timestamp – Year (MS Byte)								
6	Timestamp – Month								
7	Timestamp – Day								
8	Timestamp – Milliseconds of day (LS Byte)								
9	:								
10	:								
11	Timestamp – Milliseconds of day (MS Byte)								
12	R to R Data (18 Samples) – see “Packing Format”								
48	CRC								CRC
49	ETX								ETX

Table 6-7 ***Message: R to R Packet***

- Once enabled (by the “Set R to R Packet Transmission State” message), the packet is sent periodically to a remote unit (every 1.008 seconds).
- Each R to R sample is separated in time by 56ms.

6.6.1. Packing Format

The R to R samples are packed in the following format:

Byte/Bit	7	6	5	4	3	2	1	0
0	R to R Sample 0 (LS Byte)							
1	R to R Sample 0 (MS Byte)							
2	R to R Sample 1 (LS Byte)							
3	R to R Sample 1 (MS Byte)							
4	R to R Sample 2 (LS Byte)							
5	R to R Sample 2 (MS Byte)							
34	R to R Sample 17 (LS Byte)							
35	R to R Sample 17 (MS Byte)							

Table 6-8 ***Packing Format for R to R Data.***

- Each sample is 16-bits in length so the data is packed into 2 bytes (LS Byte first).

6.7. MSG:0x25 - Accelerometer Packet

This message contains the Accelerometer data transmitted by the Bio Harness. Once enabled, the packet is transmitted periodically.

Byte/Bit	7	6	5	4	3	2	1	0	Field
0	STX								STX
1	0x25								Msg ID
2	84								DLC
3	Sequence Number (0…255)								Payload
4	Timestamp – Year (LS Byte)								
5	Timestamp – Year (MS Byte)								
6	Timestamp – Month								
7	Timestamp – Day								
8	Timestamp – Milliseconds of day (LS Byte)								
9	:								
10	:								
11	Timestamp – Milliseconds of day (MS Byte)								
12	Accelerometer Data (20 Sample Sets) – see “Packing Format”								
87	CRC								CRC
88	ETX								ETX

Table 6-9 ***Message: Accelerometer Packet***

- Once enabled (by the “Set Accelerometer Packet Transmission State” message), the packet is sent periodically to a remote unit (every 400ms). Each set of XYZ accelerometer samples is 20ms later than the previous one.

6.7.1. Packing Format

Each set of Accelerometer samples is bit-packed into the message to minimise the amount of space used in the message. The data is packed in the following format:

Byte/Bit	7	6	5	4	3	2	1	0
0								X-Bit 0
1						Y-Bit 0	X-Bit 9	
2				Z-Bit 0	Y-Bit 9			
3		X-Bit 0	Z-Bit 9					
4	X-Bit 9							
5								Y-Bit 0
6						Z-Bit 0	Y-Bit 9	
7				X-Bit 0	Z-Bit 9			
8		Y-Bit 0	X-Bit 9					
9	Y-Bit 9							
10								Z-Bit 0
11						X-Bit 0	Z-Bit 9	
12				Y-Bit 0	X-Bit 9			
13		Z-Bit 0	Y-Bit 9					
14	Z-Bit 9							
15	As Byte 0 (pattern repeats every 15 bytes).							

Table 6-10 ***Packing Format for Accelerometer Data.***

- Because each sample is 10-bits in length, the data is bit-packed into the message to conserve space.
- Note: the X/Y/Z axes correspond to the diagram below. The arrows indicate the direction for positive data values.

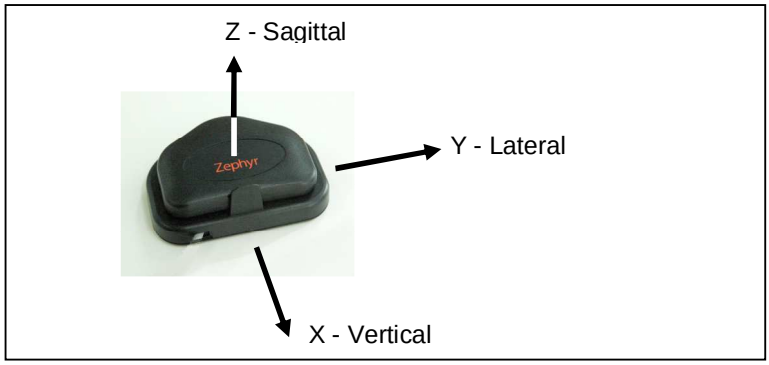

Fig. 6.11

6.8. MSG:0x27 – Bluetooth Device Data Packet

This message acts as a wrapper for data from third party Bluetoothdevices thus allowing the data to be sent between zephyr devices. A typical use is for the BioHarness to send this message containing data from a 3rd party Bluetooth device connected to the Bio Harness.

Byte/Bit	7	6	5	4	3	2	1	0	Field
0	STX								STX
1	0x27								Msg ID
2	DLC								DLC
3	Sequence Number (0…255)								Payload
4	Device Manufacturer Code (see Table 5-47)								
5	Device Type Code (see Table 5-47)								
6	Bluetooth MAC Address - 1st Byte								
:	:								
11	Bluetooth MAC Address - 6th Byte								
12	Status								
13	Bluetooth Device Data								
:	:								
DLC+3	CRC								CRC
DLC+4	ETX								ETX

Table 6-11 ***Message: Bluetooth peripheral Data Packet***

- The sequence number is incremented for every message sent
- The Bluetooth MAC address is a 6-byte MAC address in binary representation used to identify the device.
- **Status –** Indicates the status of the measurement
- **Bluetooth Device Data –** Data from the third party device. The length and contents of the data is specific to the type of device that the data originates from.

7	6	5	4	3	2	1	0
Unused				BDMS			

Table 6-12 Status Bits

Unused Bits 7-4 Unused

BDMS Bits 0-3 Bluetooth Device Measurement Status

0	Valid Measurement
1	Unreliable Measurement
2	Unable To Connect
3	Authentication Error
4	Communication Error
5	Failed Measurement
6-15	Reserved for future use

7. Fixed Message Types

7.1. NAK Response

This message is sent back to the request unit when either:

- The DLC value in the received request message didn't match the number of bytes in the data payload field.
- The received CRC and the calculated CRC didn't match.
- The ETX was not received/the value in the ETX field was incorrect.
- The data within the received payload was invalid e.g. RTC day was > 31.
- A failure to process the command occurred e.g. the Log could not be read.

Byte/Bit	7	6	5	4	3	2	1	0	Field
0	STX								STX
1	Message ID								Msg ID
2	0								DLC
3	CRC								CRC
4	NAK								ACK/NAK

Table 7-1 ***Response: NAK***

- The value in the Msg ID field is the same as the Msg ID in the request message.
- There is no data payload within the message as the packet merely indicates that the request failed.

8. Appendix A – Overview of Messages

Msg ID	Message	Description
0x01	Read Logging Data	Retrieve logging data
0x02	Delete Logging File	Delete logging file
0x07	Set RTC	Set the Date/Time
0x08	Get RTC	Get the Date/Time used
0x09	Get Bootloader Version Number	Get Firmware Boot Version
0x0A	Get Application Version Number	Get Firmware App Version
0x0B	Get Unit Serial Number	Get Firmware Serial Number
0x0C	Get Hardware Part Number	Get Firmware HW Part Number
0x0D	Get Bootloader Part Number	Get Firmware Boot Number
0x0E	Get Application Part Number	Get Firmware App Number
0x10	Set Network ID	Set the Bluetooth Network ID
0x11	Get Network ID	Get the Bluetooth Network ID
0x12	Get Unit MAC Address	Get the Bluetooth MAC address
0x13	Get Bio Harness MAC Address	Get connected Bio Harness MAC
0x14	Set General Data Packet Transmit State	Enable/Disable packet
0x15	Set Breathing Waveform Pkt Transmit State	Enable/Disable packet
0x16	Set ECG Waveform Packet Transmit State	Enable/Disable packet
0x17	Get Unit Bluetooth Name	Get Bluetooth Friendly Name
0x19	Set R to R Data Packet Transmit State	Enable/Disable packet
0x1E	Set Accelerometer Packet Transmit State	Enable/Disable packet
0x1F	Reboot Unit	Restart Code
0x20	General Data (streaming) Packet	No ACK required
0x21	Breathing Waveform (streaming) Packet	No ACK required
0x22	ECG Waveform (streaming) Packet	No ACK required
0x23	Lifesign Packet	No ACK required
0x24	R to R Data (streaming) Packet	No ACK required
0x25	Accelerometer Data (streaming) Packet	No ACK required
0x27	Bluetooth peripheral Data Packet	No ACK required
0x9B	Set ROG Settings	Set the ROG algorithm thresholds
0x9C	Get ROG Settings	Get the ROG algorithm thresholds
0xA1	BMS Set Sensor Slot	
0xA2	Set Bluetooth User Config	Set Bluetooth Module config
0xA3	Get Bluetooth User Config	Get Bluetooth Module config
0xA4	Set BT Link Config	Set Bluetooth Link config
0xA5	Get BT Link Config	Get Bluetooth Link config
0xA6	Set BioHarness User Config	Set BioHarness User Config
0xA7	Get BioHarness User Config	Get BioHarness User Config

0xAC	Get Battery Status	Current Battery Voltage & Charge
0xB0	BlueTooth Peripheral Message	For handling Bluetooth peripherals
0xB3	Reset Configuration	Resets device configuration
0xB4	Set Accelerometer Axis Mapping	Set mapping of accelerometer axes
0xB5	Get Accelerometer Axis Mapping	Get mapping of accelerometer axes

Table 8-1 Message Overview

9. Appendix B – General Data Packet

Byte Index	Element	Size (bytes)	Range	Units	Resolution
0	STX	1	0x02	-	-
1	Msg ID	1	0x20	-	-
2	DLC	1	53	-	-
3	Sequence Number	1	0…255	-	-
4	Year (LS Byte)	1	2000…2099	-	1
5	Year (MS Byte)	1			
6	Month	1	1…12	-	1
7	Day	1	1…31	-	1
8	Milliseconds of Day (LS)	1	0…86400000	ms	1
9		1			
10		1			
11	Milliseconds of Day (MS)	1			
12	Heart Rate†	2	0…240	BPM	1
14	Respiration Rate†	2	0…70	BPM	0.1
16	Skin Temperature	2	10…60	℃	0.1
18	Posture	2	-90…90	Degrees	1
20	VMU	2	0…3.3	VMU/s	0.01
22	Peak Acceleration	2	0…3.3	g	0.01
24	Battery Voltage	2	0…4.2	Volts	0.001
26	Breathing Wave Amp	2	0…3.3	Volts	0.001
28	ECG Amplitude†	2	0…0.2	Volts	0.000001
30	ECG Noise†	2	0…0.2	Volts	0.000001
32	X-axis Acceleration Min	2	± 3.3	g	0.01
34	X-axis Acceleration Peak	2	± 3.3	g	0.01
36	Y-axis Acceleration Min	2	± 3.3	g	0.01
38	Y-axis Acceleration Peak	2	± 3.3	g	0.01
40	Z-axis Acceleration Min	2	± 3.3	g	0.01
42	Z-axis Acceleration Peak	2	± 3.3	g	0.01
44	Zephyr System Channel	2	-	-	-
46	GSR	2	0…65535	nanoSiemens	1
48	SPO_2	2	0…100??	%	1
50	Blood Pressure	2	0…300??	Hg	0.001
52	Alarm Status	2	0…3	-	-
54	Button/Worn/Battery Status	2	See section 6.2		
56	CRC	1	0…255	-	-
57	ETX	1	0x06	-	-

Table 9-1 General Data Packet Format

- All data items from byte index 12 (Heart Rate) to 54 (LED Status) are 16-bit signed numbers and are LS byte first.
- The sequence number increments by 1 for each transmission.

†For canine monitoring, heart rate and breathing rate ranges extend to 350bpm.
ECG amplitude is 0 to 512 bits, and the ECG noise is replaced with Heart Rate Signal Quality, 0 to 100% with 1% resolution.