# Bluefin Sandshark
# Standard Payload Interface (SPI) Specification

**Version 0.1 (draft 09/03/2015)**

www.bluefinrobotics.com

# Table of Contents

## Software Interface

This specification describes the communications interface between the Bluefin Adapt Module (BAM) and an attached payload.

The communications interface presented in this document is a strict subset of the Bluefin Standard Payload Interface (SPI). It is not intended to be a replacement for the SPI, nor is it intendended to be a new interface.

### 1 Limitations

The following limitations apply:

- This specification is a subset of the Bluefin Standard Payload Interface, and applies only to Sandshark vehicles.

- The software interfaces defined by this specification are likely to change over time. Any new messages will follow the format outlined in the Bluefin Standard Payload Interface. Bluefin will attempt to maintain backwards compatibility as the specification evolves, but to avoid uncertainty, clients of this interface are advised to ensure that the version of the specification they are referencing corresponds to the version of the software they are using.

## 2  Timekeeping

The BAM keeps the reference time for all mission navigation log data.  The vehicle will be disciplined from GPS hits when locked.

Each message sent from the BAM will have as the first data field a timestamp indicating when the message was created.  The timestamps are guaranteed to include precision at least to the millisecond.  However, network delays and operating system latencies will reduce the quality of synchronization based on timestamps.  Some messages will contain an additional timestamp at the end of the message indicating when the data in the message were generated or received by the vehicle.

## 3  Data Logging

The payload is expected to store all necessary output data internally.  Data analysis can be performed after mission completion using the Ethernet connection.  Small amounts (a few bytes) of data or data validity indicators can be sent during the mission through the Bluefin payload interface.

The validity/status indicators provided by the payload via the payload interface are utilized in fixed safety rules to help maintain the safety of the vehicle and payload.  If the specifics of these rules are of interest, please discuss with a Bluefin representative.

## 4  Session Details

### 4.1      Session Initialization and Configuration

All navigation, sensor, and control information between the payload and BAM will be sent using a TCP connection over the onboard Ethernet.  The BAM (Bluefin Adapt Module) will act as a TCP server listening on port 29500.  The payload should connect to this port after booting.

The payload driver can be configured with the expected IP address of an expected client, along with a symbolic name for that client.  Associating names with particular payload clients allows Bluefin vehicle applications and topside tools to look for and act on messages from specific named entities.

BLUEFIN
ROBOTICS

## 4.2    Message Format

All messages sent between the payload and BAM take the form of NMEA messages.  Many messages have common field types which are described in Table 1.  In addition, each message sent by the Bluefin vehicle includes a standard NMEA checksum.  Messages sent by the payload may or may not include the checksum.  Line termination is the NMEA standard carriage return followed by line feed.  As per the NMEA spec, comma (',') and asterisk ('*') characters are reserved characters for use only as field and checksum delimiters respectively. All messages defined in this spec are considered to be case-sensitive.

Future versions of this interface are likely to define additional messages, and perhaps to add new fields to existing messages.  Payload client software should be written to ignore unexpected messages received from the BAM.  Also, it is recommended that client software ignore unexpected extra fields at the end of messages.  (That is, if new fields are ever added to the end of existing messages, they will always be optional, and safe to ignore.)

*Table 1: Field Types*

| *Field Type* | *Format* | *Definition* |
|---|---|---|
| Latitude | llll.ll | Fixed/variable length standard NMEA latitude.  The first two digits are degrees, the remaining digits are variable precision minutes.  When providing data, the Bluefin vehicle will send the maximum amount of precision available. |
| Longitude | yyyyy.yy | Fixed/variable length standard NMEA longitude.  The first 3 digits are decimal degrees.  The remaining digits are variable precision minutes. |
| Timestamp | hhmmss.ss | Hours, minutes, and variable-length seconds of the current day in UTC time.  The seconds will be transmitted with a minimum precision to the nearest millisecond, but may be more precise. |
| Integer Numeric | x<br>xx | A variable-length integer numeric field. Optional leading and trailing zeros.  If negative, the first non-whitespace character will be the minus sign. |
| Floating-point Numeric | x.x | A variable-length, variable-precision numeric field.  The decimal point may be omitted.  Optional leading and trailing zeros.  If negative, the first non-whitespace character will be the minus sign. |
| Variable Text | c--c | Variable length character field.  Any non-NMEA reserved character may be included. |

BLUEFIN
ROBOTICS

## 4.3      Message Initiation and Synchronization

Messages are sent both from the payload to the Bluefin BAM, and from the BAM to the payload. In general, messages can be sent independently and asynchronously in both directions. There is not a rigid command/response model.

The client payload may send any payload message to the BAM at any time. Payload messages are described in section 7 .

By default, only command messages are sent from the vehicle to the payload. On connect, the payload can control what status messages are sent to it via the BPLOG message (see section 7.1 ). Command messages are described in section 5 and status messages are described in section 6 .

A future revision to this specification may define additional mechanisms to allow predefinition of Bluefin status messages to be sent.

## 4.4      Payload Command Acknowledgements

Optionally, the payload can request that the BAM send acknowledgements for messages sent from the payload to the BAM. If requested, acknowledgements are sent immediately after the acknowledged payload message is received. However, since other Bluefin messages are being sent asynchronously and might be received by the payload at any time, the payload should not assume that the first message it receives after sending a payload message will necessarily be the Bluefin acknowledgement of that payload message.

The acknowledgement lets the payload know (a) that the message was received; (b) whether or not it was interpreted correctly; and, if the message requested an action on the part of Bluefin, (c) whether that action completed successfully or not.

Some payload messages request that Bluefin perform complex actions (such as adding additional mission elements) for which a success/failure determination cannot be made immediately. In these cases, the time necessary before success or failure can be determined is at odds with the goal of providing an immediate acknowledgement. Therefore, it is possible for Bluefin to send two acknowledgements. The first, immediate acknowledgement contains a success/failure indication of "pending". The second acknowledgement, which will be returned in at most 2 seconds, will contain the actual success/failure indication.

The descriptions of the various payload commands below contain additional information about what "success" or "failure" of these commands mean. Most commands always return a success/failure status immediately. Those commands which might return an initial "pending" acknowledgement are explicitly noted.

See section 6.4 "$BFACK - Message Acknowledgement" below for more information about message acknowledgements.

## 5   Command Messages from Bluefin Vehicle to Payload

These messages can be sent from the BAM to a connected payload at any time.

## 5.1      $BFSHT - Payload Shutdown

This message is sent by the Bluefin vehicle at least 1 minute before the payload will be electrically shut down.  The payload can use this time to perform normal cleanup operations.

| $BFSHT,hhmmss.ss*hh | |
|---|---|
| hhmmss.ss | Timestamp |

## 5.2      $BFVER - Vehicle Interface Version

This message contains the Bluefin Standard Payload Interface version that the vehicle supports. The vehicle will send this message at the beginning of communications with the payload.

| $BFVER,hhmmss.ss,v-v*hh | |
|---|---|
| hhmmss.ss | Timestamp |
| v-v | Version number |

## 6  Status Messages from Bluefin Vehicle to Payload

These messages are sent from the BAM to a connected payload only if explicitly requested by the payload (see section  7.1 ).

## 6.1      $BFNVG - Navigation Update

| $BFNVG,hhmmss.ss,llll.ll,a,yyyyy.yy,b,g,c.c,d.d,h.h,r.r,p.p,hhmmss.ss*hh | |
|---|---|
| hhmmss.ss | Timestamp |
| llll.ll | Latitude |
| a | Hemisphere (N/S) |
| yyyyy.yy | Longitude |
| b | Hemisphere (E/W) |
| g | Quality of position estimate, if no GPS is available this will be 0. When the position solution includes GPS, the value will be 1. |
| c.c | Altitude (m) |

| $BFNVG,hhmmss.ss,llll.ll,a,yyyyy.yy,b,g,c.c,d.d,h.h,r.r,p.p,hhmmss.ss*hh | |
|---|---|
| d.d | Depth (m) |
| h.h | Heading (deg True North in current UTM coordinate space, i.e. has the UTM convergence factor applied) |
| r.r | Roll (deg) |
| p.p | Pitch (deg) |
| hhmmss.ss | Timestamp (of computed navigation solution) |

Example:

$BFNVG,081025.987,4221.81092,N,07106.84603,W,1,45.3,20.1,203.1,-3.4,4.5,081025.980*5b

## 6.2      $BFNVR - Velocity and Rate Update

| $BFNVR,hhmmss.ss,e.e,n.n,v.v,s.s,t.t,g.g*hh | |
|---|---|
| hhmmss.ss | Timestamp |
| e.e | East velocity (m/s) |
| n.n | North velocity (m/s) |
| v.v | Down velocity (m/s) |
| s.s | Pitch rate (deg/s) |
| t.t | Roll rate (deg/s) |
| q.q | Yaw rate (deg/s) |

## 6.3      $BFMIS - Mission Status

This message is sent when the mission status changes, such as when a mission starts or stops.

| $BFMIS,hhmmss.ss,d--d,t—t,a--a*hh | |
|---|---|
| hhmmss.ss | Timestamp |
| d--d | Current dive file.  If current behavior has not been requested by the mission plan, this field will be empty. |

| $BFMIS,hhmmss.ss,d--d,t—t,a--a*hh | |
|---|---|
| t--t | Textual string for mission status. Currently defined mission status types include, but are not limited to "Offline", "Waiting", "Invalid Dive File", "Invalid Wrapup File", "Ready", "Running", "WrappingUp", "Aborting", "Completed", "Aborted", and "Stopped". The status string starting with "Running" may be followed by the logging directory for that mission. |
| a--a | Additional mission status details such as error information for an abort. The text will be human readable for debug and logging purposes. |

## 6.4      $BFACK - Message Acknowledgement

Optionally, the Bluefin vehicle will respond to each payload command with an acknowledgement.

Broadly, there are four different acknowledgements. A command can be rejected as invalid (an unrecognized command name, or one with an incorrect number or types of parameters). A command can be recognized, processing of the command can be initiated, but the processing can fail. A command can be recognized and successfully processed. Finally, it may not always be immediately possible for the Bluefin vehicle to determine whether a command has succeeded or failed, in which case a pending acknowledgement may be issued.

When a pending acknowledgement is returned, the payload knows that its request was at least syntactically valid, and that the vehicle is working on processing the request. At some later time (generally within 2 seconds), the vehicle will transmit an actual success or failure acknowledgement corresponding to the same payload request. Within this specification, those payload commands which have the possibility to return these "pending" acknowledgements are explicitly noted. Otherwise, commands can be assumed to always return requested acknowledgements immediately.

The acknowledgement also contains a message string which may contain additional details about the failure, or certain extra status information in case of success. The format of these message strings is not specified. Except by prearrangement with Bluefin Robotics, payload software should not attempt to interpret the contents of the message string. Message strings in acknowledgements are intended for debugging and human consumption, not machine parsing.

By default, these acknowledgements are *not* sent. The payload can request that acknowledgements be sent by using the BPLOG command.

| $BFACK,hhmmss.ss,c-c,hhmmss.ss,iiii,s,c,m-m,*hh | |
|---|---|
| hhmmss.ss | Timestamp |

| $BFACK,hhmmss.ss,c-c,hhmmss.ss,iiii,s,c,m-m,*hh | | |
|---|---|---|
| c-c | Name of payload command being acknowledged | |
| hhmmss.ss<br><br>iiii | Timestamp of payload command being acknowledged<br><br>If the command being acknowledged is a new behavior insertion (e.g. BPRTK), the payload-assigned identifier for that behavior | |
| s | *Acknowledgement Status Code* | |
| | 0 | Invalid Request (unknown message, or improper parameters) |
| | 1 | Request unsuccessfully processed |
| | 2 | Request successfully processed |
| | 3 | Pending: request received and processing initiated |
| c | Future use; always 0 in this revision of this specification | |
| m-m | String describing reason for failure (or acknowledgement of success) in possibly more detail | |

## 6.5 $BFCTL - Backseat Control

This message explicitly hands over control from Bluefin (frontseat) to the payload (backseat). A value of 1 (true) will be sent to the backseat indicating that it now has control over the vehicle. This will be sent at a time in the mission when the internal value "allowDynamicRequests" transitions to true. It must be programmed as such in the mission plan. The value of 0 (false) will be sent to the backseat when the internal value "allowDynamicRequests" transitions to false, or when the mission state transitions out of "Running". Alternatively, the vehicle can be configured to hand control to the backseat when the mission starts and returned to the frontseat at mission stop.

| $BFVER,hhmmss.ss,c*hh | |
|---|---|
| hhmmss.ss | Timestamp |
| c | Control (boolean true = 1, false = 0) |

# 7 Messages From Payload to Bluefin Vehicle

## 7.1 $BPLOG - Logging Control

When a payload initially connects to the vehicle, all status messages are turned off. The BPLOG command is used to turn ON or OFF a specific message type. The special keyword ALL can also be used to turn on or off all messages (with the exception of BFACK, which must always be requested explicitly).

If this message is acknowledged (see section 6.4 ), a 'success' acknowledgement indicates that the requested logging action (ON or OFF) was performed.

| $BPLOG,rrr,a-a*hh | |
|---|---|
| rrr | Requested 3-letter message identifier (e.g. NVG to turn on/off logging of navigation data to this payload), or the keyword ALL to turn on/off logging of all data to this payload |
| a-a | ON or OFF |

## 7.2 $BPSTS - Payload Status Message

| $BPSTS,hhmmss.ss,f,m--m*hh | |
|---|---|
| hhmmss.ss | Timestamp |
| f | Status flag. 0 = Failed, 1 = All OK. |
| m--m | Variable length textual status message. |

## 7.3 $BPRMB - Modify Current Behavior

This message may be used to provide the Bluefin vehicle with a temporary alternate heading, depth, and/or speed to traverse along.  The current behavior will not be terminated immediately, but may be at some future point if the total time duration or deviation from original behavior is too large.  Not all of the fields are required; any empty fields do not modify that aspect of the current behavior.

If the payload wishes to continue modifying the current behavior this message must be sent at least every 5 seconds or the Bluefin vehicle will resume control of the mission.

Note that this message can not be used to command the vehicle beyond its safe operating parameters.

The ability of the payload to modify the vehicle's behavior is a complex feature that may

negatively impact mission reliability.  The vehicle's willingness to accept dynamic mission plan requests is configurable.  Dynamic requests are enabled by setting the onboard "allowDynamicRequests" parameter to true.  If a $BPRMB request is received during a mission or a phase of a mission where "allowDynamicRequests" is not enabled, the request will be ignored, or acknowledged with a failure response if acknowledgements are enabled.

| $BPRMB,hhmmss.ss,h.h,d.d,dm,v.v,vm,hm*hh | |
| --- | --- |
| hhmmss.ss | Timestamp |
| h.h | Variable precision heading in degrees.  Empty means use the current behavior's value. |
| d.d | Variable precision depth or altitude in meters.  May be empty. |
| dm | Depth mode.  0 signifies the previous field is a depth; 1 signifies the previous field is an altitude; 2 signifies the previous field is a pitch; 3 signifies the previous field is an elevator adjustment.  May be empty. |
| v.v | RPM to command the thruster or speed in m/s to command the vehicle.  May be empty. |
| vm | Speed mode. 0 signifies the previous field is a thruster RPM; 1 signifies the previous field is a speed in meters per second. May be empty. |
| hm | Horizontal mode. 0 signifies the first field is a heading; 1 signifies a rudder adjustment. |

## 7.4     $BPABT - Abort Mission

This command causes the current mission to end, and its wrapup behavior executed (such as a powered ascent to the surface).

| $BPABT,hhmmss.ss,c--c,a*hh | |
| --- | --- |
| hhmmss.ss | Timestamp (when message is sent) |
| c--c | Message.  This may be displayed to the user to explain the abort. |
| a | Abort reason (0 = successful mission, 1 = abort with errors).  Abort with errors (1) is the default. |

## 7.5    $BPVER - Payload Interface Version

This message reports the version of the Bluefin Standard Payload Interface that the payload supports.  The payload must send this message at the beginning of communications.

| $BPVER,hhmmss.ss,v-v*hh | |
|---|---|
| hhmmss.ss | Timestamp |
| v-v | Version number |