Wave Glider™

Interface Control Description

Version 2.10

(Through WGMS Release 3.7.2)

14 March 2011

LIQUID ROBOTICS, INC.
1329 MOFFETT PARK DR.
SUNNYVALE, CA 94089
(650) 493-6300

LRI Part Number: 030-00026-02

This Page Intentionally Left Blank

REVISION HISTORY

Rev	Changes	WGMS	Author(s)
1.00	Initial Production Release		FJ
1.10	Split Chapter 5 into 5, 6 & 8. Add Chapter 7		FJ
1.20	Add DB3. Update Commands/Msgs and XBee.		FJ
1.30	Update Commands. Fix XBee. Update connectors.	3.5.1	FJ
1.40	Mod Part #. Add WGMS note. Add MC Chapt 7 changes	3.6.1	FJ, MC
2.00	Incorporate API info. Split Comm chapters. Entities into Appendix A. New Roles description.	3.7.2	FJ

TRADEMARKS

This Page Intentionally Left Blank

Table of Contents

1. Overview 9
1.1 Wave Glider Documents 9
1.2 Wave Glider Primary Components 10
1.3 Float Assembly 11
1.4 Command & Control Drybox 12
1.5 Fore/Aft Payload Dryboxes 14
1.6 Float-Glider Umbilical 15
1.7 Submerged Glider 16
1.8 Electrical/Electronics 17
1.9 Communications and Control 19

2. Float Mechanical 23
2.1 Float Overview 23
2.2 Float Payload Bays 25
2.3 C&C Drybox 28
2.4 Payload Dryboxes 29
2.5 Float Considerations 33
2.6 External Float Payloads 34

3. Glider Mechanical 37
3.1 Glider Overview 37
3.2 Glider Payload Considerations 39

4. Electrical Specifications 45
4.1 Electrical Overview 45
4.2 Standard Electrical Connections 47
4.3 External Charging 52
4.4 Power Considerations 53

5. Vehicle Communications and Control 57
5.1 Communications 57
5.2 C&C Functions 60
5.3 Data Types 61
5.4 Wave Glider Management System 63
5.5 WGMS Components 64
5.6 Common WGMS Scenarios 66
5.7 Alternate WGMS Configurations 71

6. Communications Methods 73
6.1 Iridium/Relay Communications 73
6.2 Payload Communications 75
6.3 XBee Radio Communications 76
6.4 Communication Data Formats 77

6.5 Navigation Modes 79
6.6 Roles and Permissions 79

7. Commands 85
7.1 Command Data Packet Format 85
7.2 Command Summary 86
7.3 Command Descriptions 87
7.4 Restricted Commands 106

8. Vehicle Responses 111
8.1 ACK/NAK Protocol 111
8.2 ACK Response Data 113
8.3 NAK Response Data 122
8.4 Restricted ACK Responses 123
8.5 Unsolicited Messages 124
8.6 Relay-Only Message 125

9. Vehicle Telemetry 127
9.1 Telemetry Packet Format 127
9.2 Telemetry – NAL Tracker Format 128
9.3 Standard Telemetry Data 129
9.4 Telemetry – Power Controller Status 131
9.5 Payload Telemetry 132

10. Payload Communications 137
10.1 Communication Hardware 137
10.2 Communication Method 139
10.3 Protocol Message Format 142
10.4 Addresses 144
10.5 Other Considerations 146
10.6 Payload Message Types 146
10.7 Payload Message Packet Descriptions 147
10.8 Message Protocol Use Cases 159

11. XBee Communications 169
11.1 XBee Command Menu 169
11.2 XBee Menu Options 170
11.3 Status Response 179
11.4 Menu Error Responses 179

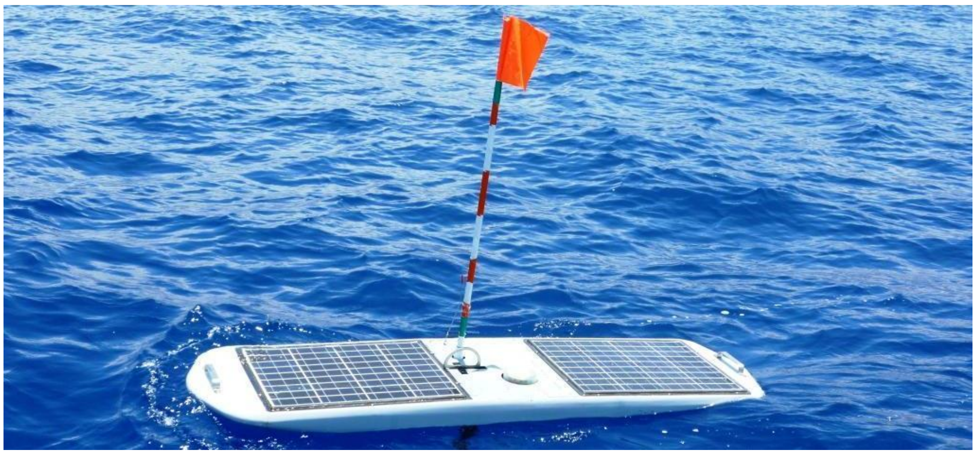
12. TroubleShooting 181
12.1 Overview 181
12.2 Problems and Responses 182
12.3 Interpreting C&C Alarms 184
12.4 Contacting Liquid Robotics Inc. 187

13. Notes 189
13.1 Glossary 189
13.2 Abbreviations and Acronyms 195
Appendix A – XML, SOAP and Entities 197
A.1 XML 197
A.2 Soap Envelopes 198
A.3 Management System Function Summary 200
A.4 Function Descriptions 201
Appendix B – Iridium/Relay Header Data 207
B.1 Iridium Ship-to-Shore Data Packets 207
B.2 Relay Ship-to-Shore Packets 209
B.3 Iridium Shore-to-Ship Data Packets 211
B.4 Relay Shore-to-Ship Packets 212
Appendix C – CRC-16 Calculation 213

This Page Intentionally Left Blank

1. Overview

This chapter provides a general overview of the Wave Glider system. It is assumed that the reader has reviewed the appropriate chapters in the Wave Glider *User Manual*, and is generally familiar with the Wave Glider terminology. However this chapter is intended as a more extensive description of the system, with an emphasis on information related to those doing payload configuration and systems integration on the Wave Glider.



1.1 Wave Glider Documents

There are several documents for the Wave Glider in addition to this one, each with a different intended audience and purpose. For the payload integrator/developer, the primary documents of interest are:

1.1.1 User Manual

The LRI *User Manual* provides overall information about the Wave Glider and the Shore-based User Interface. It is intended primarily for persons provisioning a Wave Glider, planning missions, and executing/monitoring mission operations.

The *User Manual* should be reviewed by all persons working with the Wave Glider, in order to get an overall understanding of the Wave Glider and its operation.

1.1.2 Payload Electronics Guide

The LRI *Payload Electronics Guide* contains detailed specifications for the standard electrical components used within payloads. It is intended for those people interfacing sensors and other equipment to the payload electronics.

The *Payload Electronics Guide* provides more detailed electronics information than this document.

1.1.3 System Administration

The LRI *System Administration* manual is intended for those people who are installing and configuring the Wave Glider Management System for different deployment scenarios.

Generally, the information in the *System Administration* manual will not be needed by a payload designer/integrator, but it may prove useful in special cases.

1.2 Wave Glider Primary Components

The primary components of the Wave-Glider Vehicle are shown in Figure 1-1 below. They are:

1.2.1 Float

This is the component that rides on the surface of the water. It contains the batteries, control and communications electronics, and optional user-supplied payloads. Two solar panels on the top maintain the battery charge. The upper surface of the Float may also have additional devices for sensing and communications.

1.2.2 Umbilical

This is a high-strength streamlined connector between the Float and the Glider. The Umbilical provides electrical power to the Glider and also contains electrical wires for data communications between the Float and Glider. The Umbilical is longer than shown in the illustration.

1.2.3 Glider

The Glider (also called the "Sub") is always submerged well below the Float during operation. It provides the propulsion and steering for the Wave Glider system. The Glider can also be used as a platform to mount underwater sensing equipment.

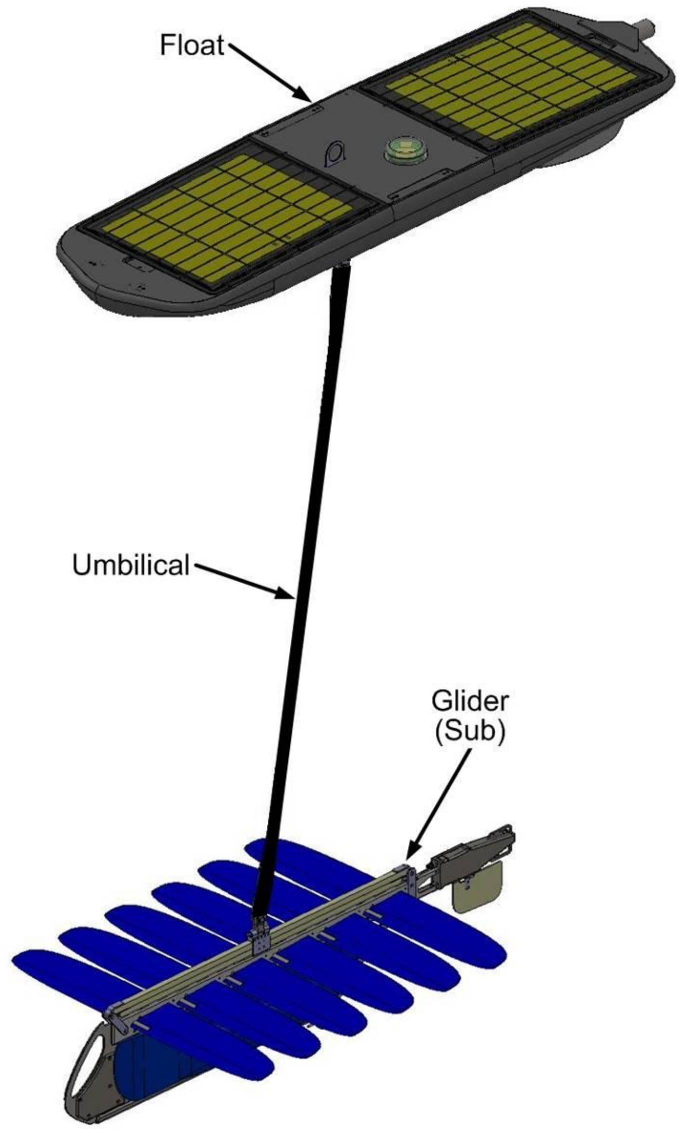


Figure 1-1: Wave Glider Main Components

The Wave Glider main components are described briefly below and in more detail in subsequent chapters.

1.3 Float Assembly

The main Float assembly components are shown in the figure, and generally described below.

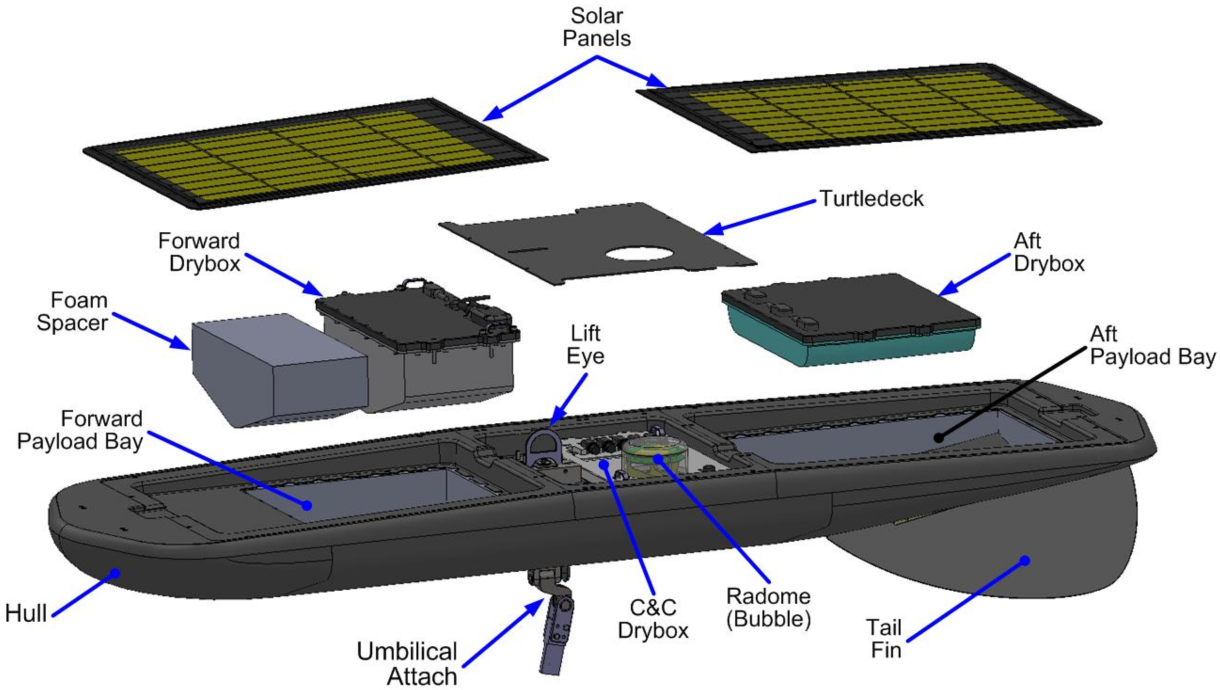


Figure 1-2: Float Components Exploded View

Aft Drybox: Optional watertight container for the aft payload.

Aft Payload Bay: Aft compartment. Shallower than the Forward Payload Bay.

C&C Drybox: The Command and Control Drybox contains the Vehicle electronics and the batteries. See section 1.4 "Command & Control Drybox" for additional information.

Foam Spacer: Optional spacer. Size/shape will vary depending upon the size of the drybox.

Forward Drybox: Optional watertight container for the forward payload.

Forward Payload Bay: Forward compartment. Deeper than the Aft Payload Bay.

Hull: The single-piece composite main shell of the Float assembly.

Lift Eye: Attach point for lifting an assembled Wave Glider. The Lift Eye connects directly to the Umbilical Attach.

Radome: (Also called the "Bubble"). Part of the C&C Drybox. Exposed above the Turtledeck. Encloses the Vehicle lights (IR and/or visible), as well as the Iridium, GPS and XBee antennas.

Solar Panels: Photovoltaic ("PV") cells used to charge the electronics batteries.

Tail Fin: Rear stabilizing fin. This is non-moving and non-adjustable.

Turtledeck: Non-watertight cover for the Center Drybox. It is made of composite and is radio wave transparent.

Umbilical Attach: Gimballing fitting to which the Umbilical is connected.

The basic Hull has some additional features that should be noted:

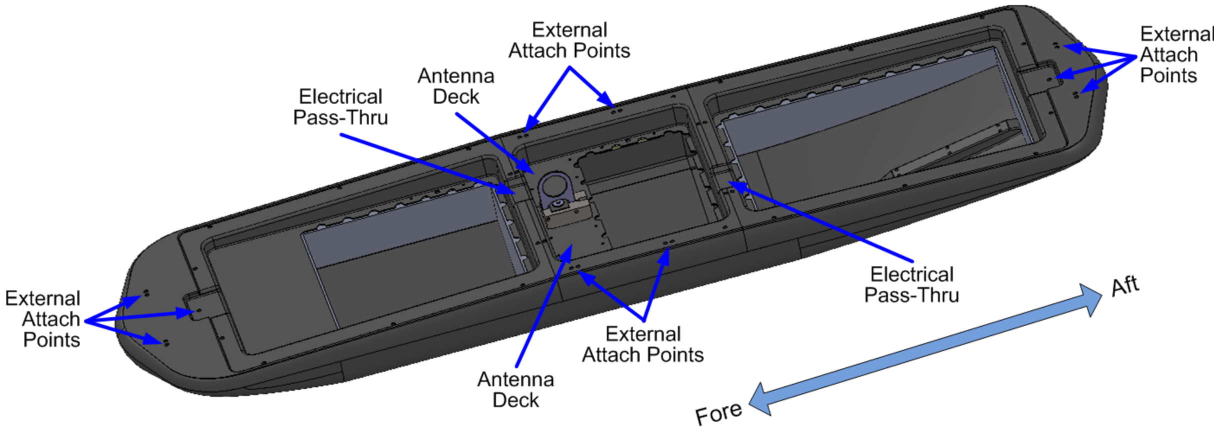


Figure 1-3: Basic Hull External Features

Antenna Decks: Two adjacent platforms that can be used to mount auxiliary communication antennas or external sensors.

Electrical Pass Thrus: Cutouts for routing electrical cables between the payload bays and the central C&C Drybox, or to fore and aft external payloads.

External Attach Points: Threaded thru-holes that can be used to attach external payloads.

1.4 Command & Control Drybox

The C&C Drybox is a factory sealed module that mounts in the center bay of the Wave Glider. This Drybox contains the computing and communications equipment. It also contains the batteries to store energy from the solar panels. There are two versions of the C&C Drybox.

1.4.1 Original Drybox

The Original Drybox is shown in Figure 1-4. It has seven connectors on top, all on one side.

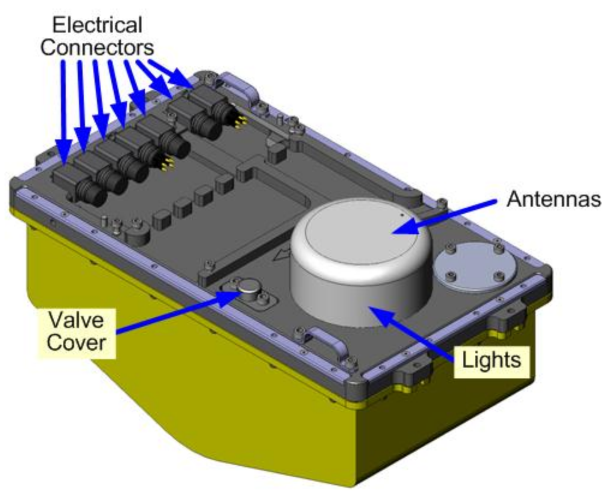


Figure 1-4: Original C&C Drybox

The main external components of the Original C&C Drybox are:

Antennas: GPS, Iridium and XBee antennas.

Electrical Connections: Seven watertight electrical connectors for the following:

1) Umbilical (Tether)
2) Aft solar panel
3) Forward solar panel
4) Payload 1
5) Payload 2
6) I/O 1
7) I/O 2

Lights: External lights, both visible and Infra-Red.

Valve Cover: Protective cover for the box pressurization valve.

1.4.2 Revised Drybox

The Revised Drybox is shown in Figure 1-5. It has ten connectors on top on both sides.

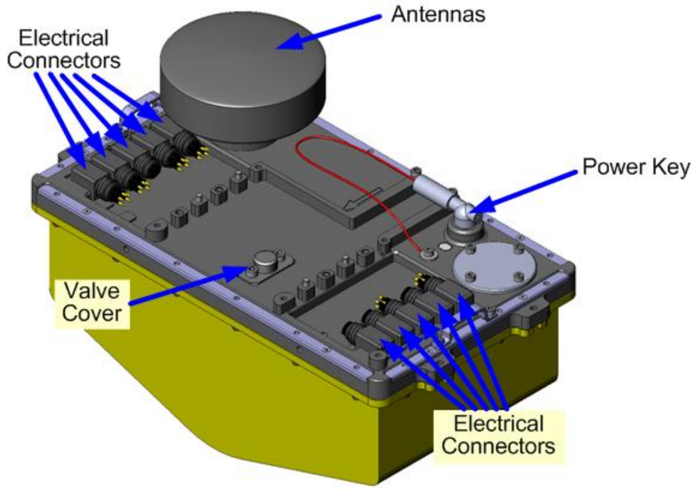


Figure 1-5: Revised C&C Drybox

The main external components of the revised C&C Drybox are:

Antennas: GPS, Iridium and XBee antennas.

Electrical Connections: Ten watertight electrical connectors for the following:

1) Tether (Umbilical)	6) Weather
2) Forward solar panel	7) Speed
3) Aft solar panel	8) Camera
4) Payload 1	9) Radar
5) Payload 2	10) Satcom

Power Key: Insertion key that turns the C&C On or Off.

Valve Cover: Protective cover for the box pressurization valve.

1.5 Fore/Aft Payload Dryboxes

The Fore and Aft Dryboxes are used to store user-supplied payloads. The Dryboxes shown below are representative of those available from Liquid Robotics Inc. Note that since the aft payload bay is both longer and shallower than the forward bay, the two Dryboxes have different shapes.

The Payload Dryboxes are described in more detail in Chapter 2.

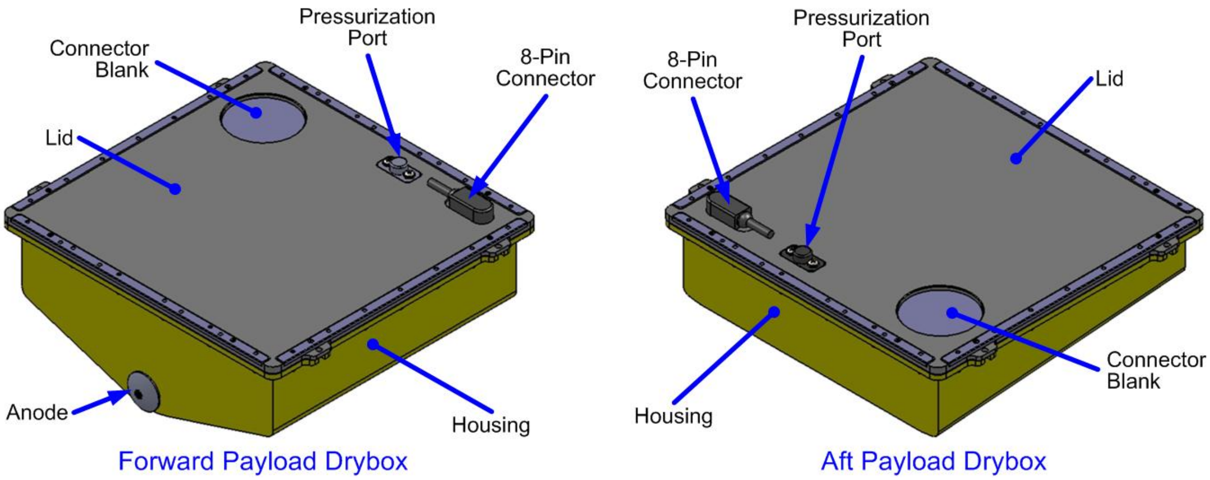


Figure 1-6: Payload Dryboxes

8-pin Connector: Watertight electrical connector. Used for power and communications.

Anode: Bolted-on sacrificial anode.

Connector Blank: Removable plastic panel for adding additional connectors. An optional lid with two connector panels is available.

Lid: The top of the Drybox. Typically, the payload electronics are mounted to the underside. The lid is retained with multiple bolts and sealed with an O-ring around the edges.

Pressurization Port: Valve (with cover) for pressurizing the drybox.

Housing: The lower part of the Drybox. Typically has no internal attachments.

Note that the Dryboxes are designed to be watertight only, and are not intended to withstand significant water pressure.

1.6 Float-Glider Umbilical

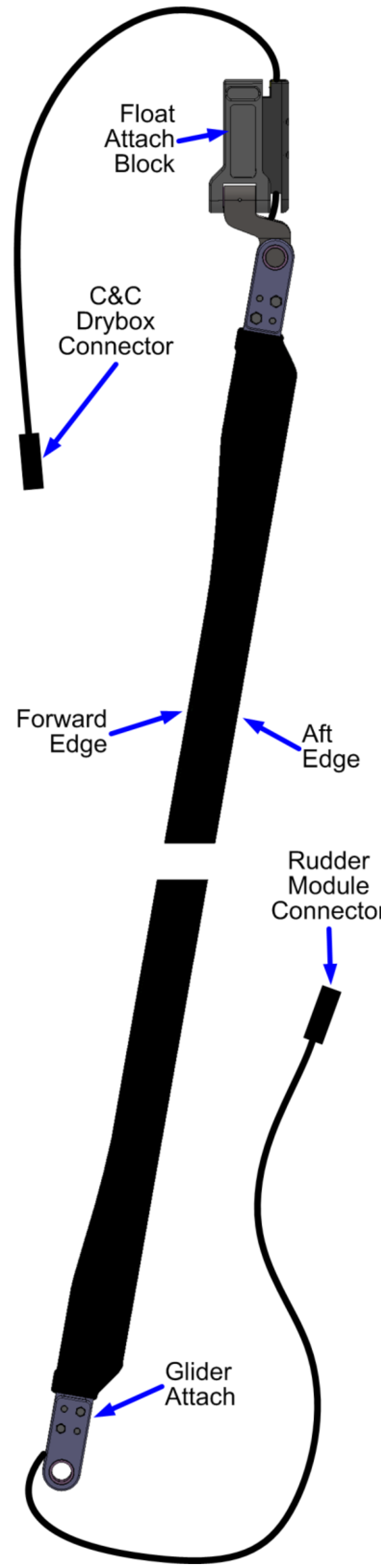
The Umbilical (also referred to as the "Tether") is a flexible cable that connects the Float to the submerged Glider. Since it is fairly long (approx. 7 meters), it is streamlined to reduce drag.

The Umbilical contains electrical wiring for power, rudder control and payload communications. It also contains high-strength reinforcement cables.

The attach fittings at each end have both mechanical and electrical connections.

The mechanical connections at each end are different. The Float attach is a quick connect and release, while the Glider attach is intended to be more permanently connected.

Likewise the electrical connections at each end are different. The Glider attach end has a long electrical cable to connect to the aft Rudder Module. The Float attach end has a shorter cable for connecting to the C&C electronics.



Aft Edge: For streamlining, the Aft Edge tapers to a sharp edge.

C&C Drybox Connector: The watertight electrical connector that connects to the top of the C&C Drybox.

Float Attach Block: The quick-release mechanical connector to the Float. The electrical connection passes through this fitting.

Forward Edge: For streamlining, the Forward edge is rounded. Note that the forward edge is also slightly concave at each end.

Glider Attach: The mechanical connection to the Glider. This is the upper part of the Glider gimballing system and requires assembly.

Rudder Module Connector: The watertight electrical connector from the Umbilical to the Rudder Module.

Note: To fit on the page, the figure does not show the full length of the Umbilical.

Figure 1-7: Umbilical Descriptors

1.7 Submerged Glider

The Glider (also called the "Sub") is the primary component of the Wave Glider. It is used to provide both thrust and steering. However, it can also be used to carry sub-marine payloads.

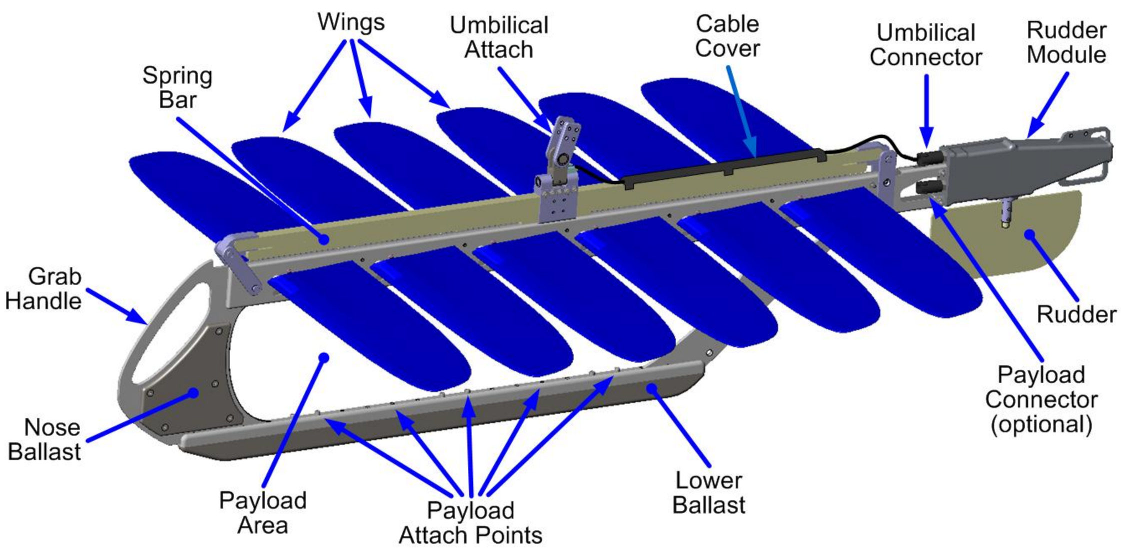


Figure 1-8: Glider (Sub) Basic Descriptors

Cable Cover: Protects and guides Umbilical-to-Rudder cable.

Grab Handle: Used for launch and retrieval.

Lower Ballast: Weight that may be removed or adjusted to compensate for added payload weight and/or floatation.

Nose Ballast: Forward-located weight that can be changed to balance the Glider, in order to compensate for off-CG mounting of payloads.

Payload Area: Area for locating submarine payloads. The holes on the bottom go completely through, for mounting payloads on the bottom of the Glider.

Payload Attach Points: M5 threaded holes. Bolts here are used hold on the Lower Ballast.

Payload Connector: Optional watertight electrical connector for payload electrical and communications. Connects within the Rudder Module to the Umbilical Connector.

Rudder: The component used to steer the Wave Glider.

Rudder Module: Factory sealed. Contains the servo motor and mechanism for the Rudder. Has electrical connectors for the Umbilical and optional payload.

Spring Bar: A flexible connection from the Umbilical Attach to the Glider. Used to reduce the stress on the Umbilical.

Umbilical Attach: Mechanical attachment point for the Umbilical.

Umbilical Connector: Watertight electrical connector from the Umbilical to the Rudder Module.

Wings: Six pairs of wings connected through the center. Each left-right pair of Wings rotates independently to provide thrust.

When the Glider is carrying payloads, it may use streamlined fairings to cover the primary Payload Area. An example is shown in the figure below. The actual shape of the cover can vary depending upon the type, shape and volume of the payload.

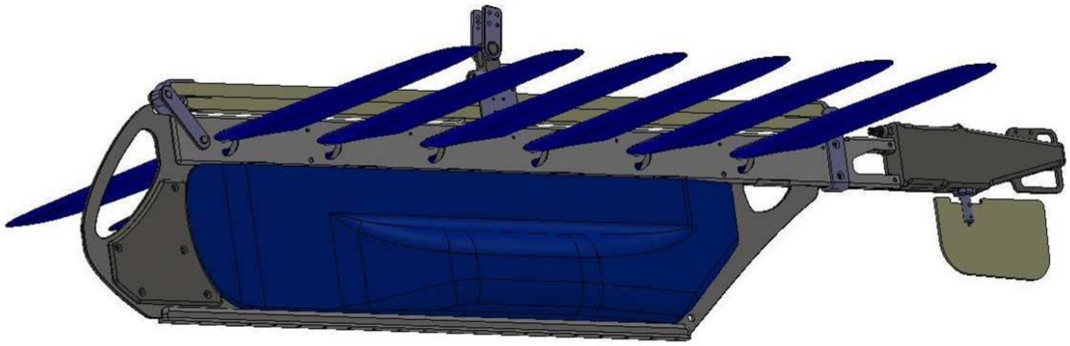


Figure 1-9: Glider With Custom Payload Cover

See section 3.2 for additional design considerations for Glider payloads.

1.8 Electrical/Electronics

The basic control and communications electronics are in the C&C Drybox. Additional electronics may be added as payload, generally interfacing with the C&C Drybox.

Wave Glider electrical and electronic systems are described in more detail in Chapter 4.

1.8.1 Original Drybox Connections

The top of the original C&C Drybox contains two male 5-pin connectors for the two Solar Panels. The other five connectors are 8-pin female.

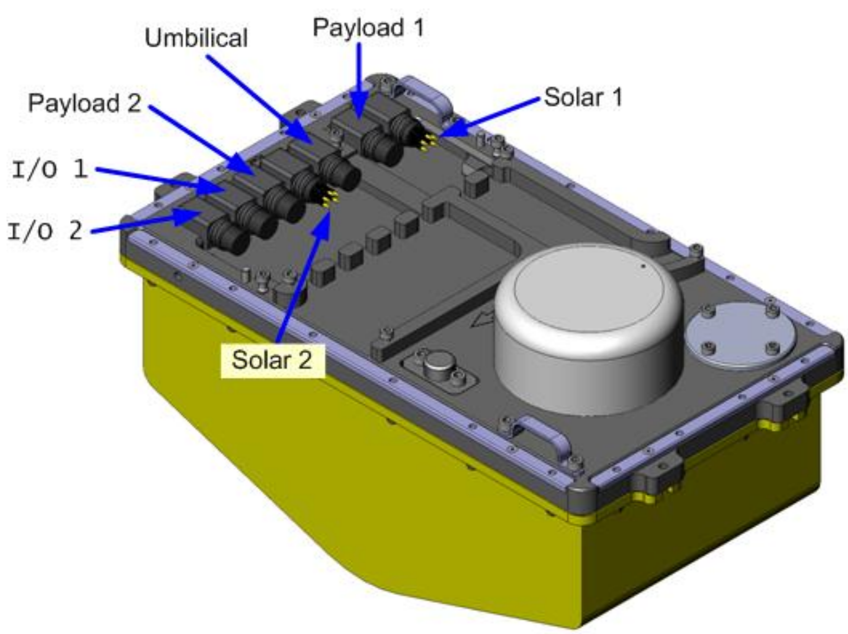


Figure 1-10: Connections on Original C&C Drybox

1.8.2 Revised Drybox Connections

The top of the revised C&C Drybox contains the two male 5-pin connectors for the two Solar Panels. The other eight connectors are 8-pin female.

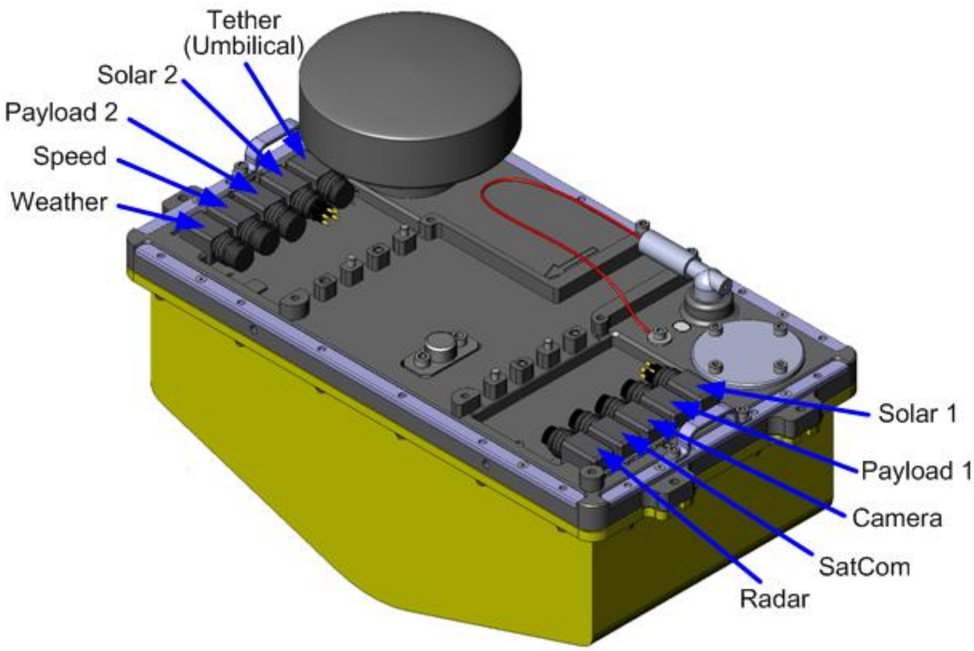


Figure 1-11: Connections on Revised C&C Drybox

1.8.3 Rudder Module Connections

The Rudder Module has an 8-pin male connector on the, which is used for the Umbilical electrical connection. In addition, there is an option for an 8-pin female connector for a Glider payload.

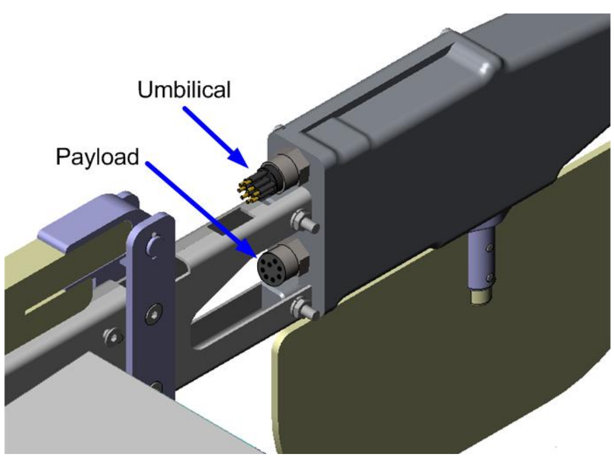


Figure 1-12: Rudder Module Connectors

Note: In the above figure, the rudder box cover is removed to show the connectors.

1.8.4 Electrical Connectivity

The general connectivity of the Wave Glider electrical system is shown below. A complete description is included in Chapter 4 "Electrical Specifications".

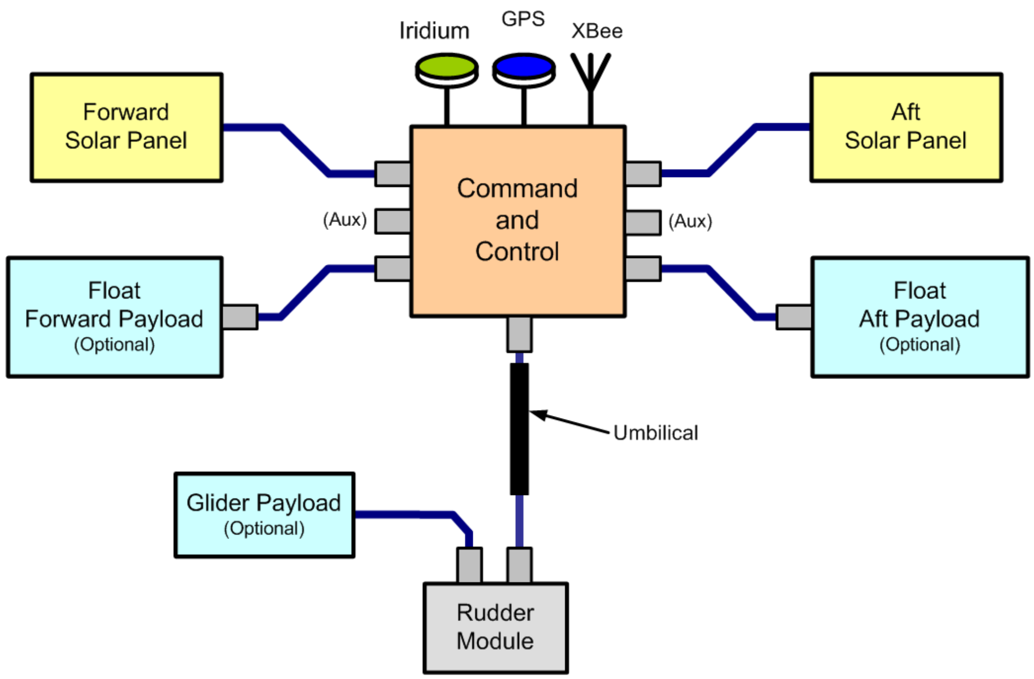


Figure 1-13: Basic Electrical System

As is shown in the figure, the C&C Drybox is the central point to which all the main electrical components connect.

1.9 Communications and Control

On-board communications is managed by the computer in the C&C Drybox. Standard Wave Glider data communications can take four forms:

1) Iridium satellite link between the C&C and a control station (“Console”).
2) An alternate satellite “Relay” link between the C&C and a Console.
3) RS-232 wire link from the C&C to the Fore and/or Aft payloads.
4) XBee radio link from the C&C to a nearby receiver.

Each of these is discussed briefly below. Refer to Chapter 5 "Vehicle Communications" for more detailed information.

Also, a Payload can have its own separate communications link to an off-Vehicle location. This can utilize additional antennas or other communications devices that are attached to the Vehicle.

1.9.1 Iridium Satellite Link

The primary communications link for the Wave Glider is an Iridium satellite link to a control workstation, which is typically on-shore but can also be shipboard. The Iridium link is typically used through an Internet-hosted Graphical User Interface that hides the details of the commands and responses.

Command packets are sent from the workstation to either control the Wave Glider's behavior, or to request information from the Vehicle.

The Iridium link periodically sends out telemetry data packets. The time between telemetry messages is user-variable. After the telemetry data is sent, any command data from Shore is downloaded.

When the command data packet has been processed by the Wave Glider, the Vehicle responds with an ACK (acknowledge) data packet containing the requested data. If the command also requests some action by the ship (such as moving the rudder), that action is executed also.

If the command data packet sent to the Wave Glider is in some way incorrect, the Vehicle responds with a NAK (not-acknowledged) data packet, that indicates the nature of the error.

Note that Iridium command data is processed only after Vehicle-initiated communications (telemetry intervals or ACK/NAK responses), and not necessarily when the command is entered by the User.

1.9.2 Relay Communications

Communications can also be accomplished via a “Relay” connection that bypasses the standard Iridium system. Relay communications is similar to Iridium communications except that data is sent directly to and from the Vehicle, through either an alternate Iridium system, or else some other satellite or radio means.

1.9.3 Payload Communications

Each of the Float payloads can communicate by sending messages to the C&C using their respective RS-232 links. Payload boards (and other electronic devices) can communicate with each other using the same payload communications protocol.

Communications can be C&C-to-payload, payload-to-C&C or payload-to-payload. Payloads can also send messages out through the Iridium link.

Payload messages are sent using data packets with formats different than the Iridium packets. As with the Iridium packets, the recipient responds with an ACK or NAK.

1.9.4 XBee Radio Link

The XBee radio link is used for short-range (approx. 100 meters) communications to and from the Vehicle. XBee is typically used for communications during Vehicle launch and retrieval.

The XBee link is convenient for shore-based payload development and system integration, since commands are received and processed immediately.

XBee communications are done using interactive text-based commands and responses.

1.9.5 Navigation

Navigation of the Vehicle is typically specified using "Waypoints", which are each defined by three values:

1) The Waypoint number: A user-assigned integer value from 1 to 254. No two Waypoints can have the same number.
2) A Latitude value, defined with a precision of arc-minutes / 10,000.
3) A Longitude value, also defined with a precision of arc-minutes / 10,000.

The basic "Navigation Modes" of a Vehicle are:

- Move to a Waypoint and maintain station there.
- Move from a start Waypoint through a sequence of Waypoints to an ending Waypoint.
- Continually move through a series of Waypoints, in a cyclic fashion.
- Maintain a constant heading.
- Hold a fixed rudder position

Waypoints and heading values are described in more detail in sections 6.4.2 and 6.4.3.

This Page Intentionally Left Blank

2. Float Mechanical

2.1 Float Overview

In the drawings shown on the following pages, dimensions are shown as millimeters [inches].

2.1.1 Float Overall Dimensions

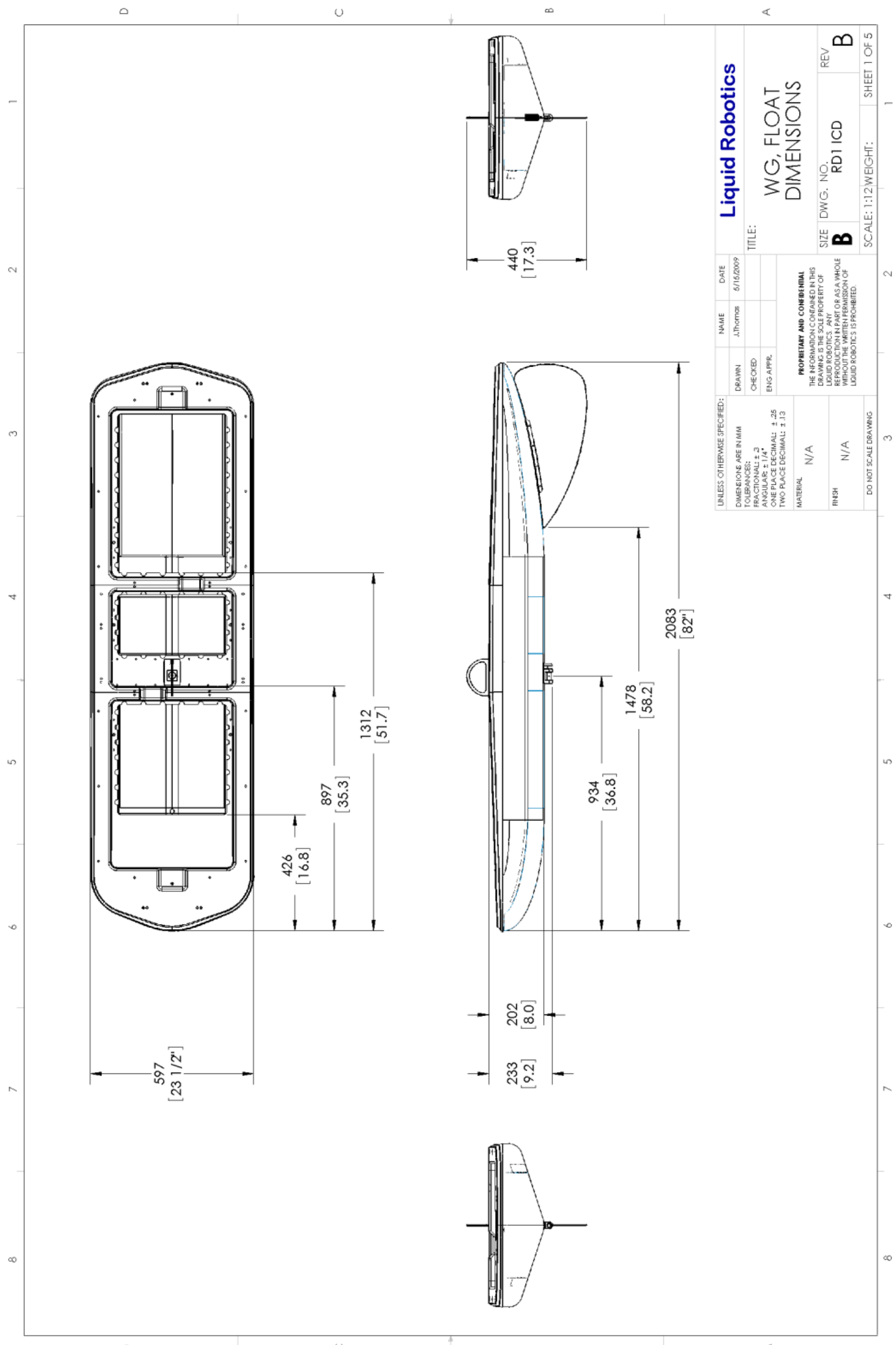


Figure 2-1: Float Overall Dimensions

2.1.2 Float Attachment Hardpoints

The payload attachment points on the upper deck of the Float are shown in the drawing below. All attachment points are thru-hole M5 x ¾ thread.

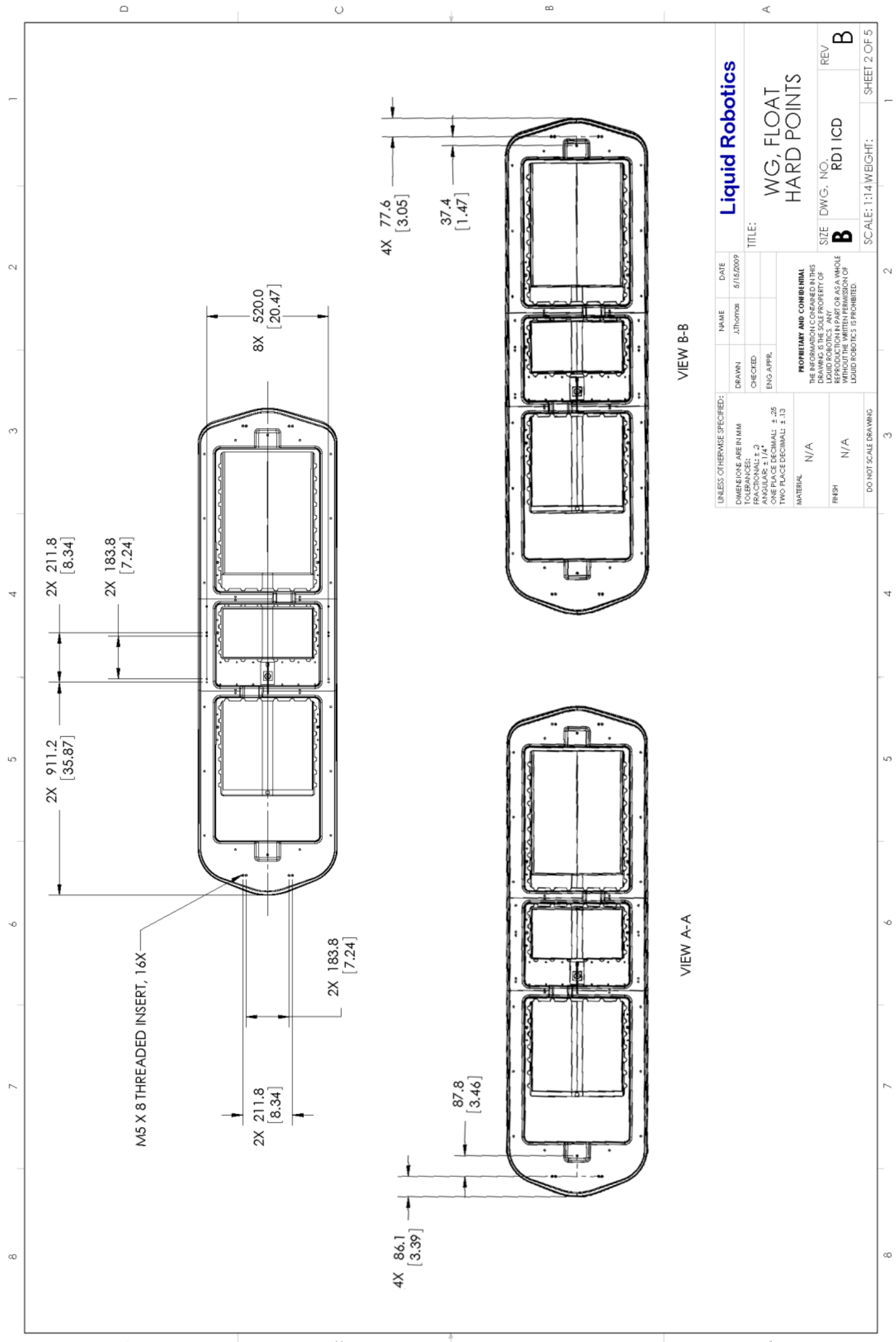


Figure 2-2: Float Attachment Hardpoints

2.2 Float Payload Bays

The Float Payload bays all have different dimensions, as shown on the following pages.

2.2.1 Float Center Bay

The Center Bay holds the C&C Drybox and also contains the Antenna deck.

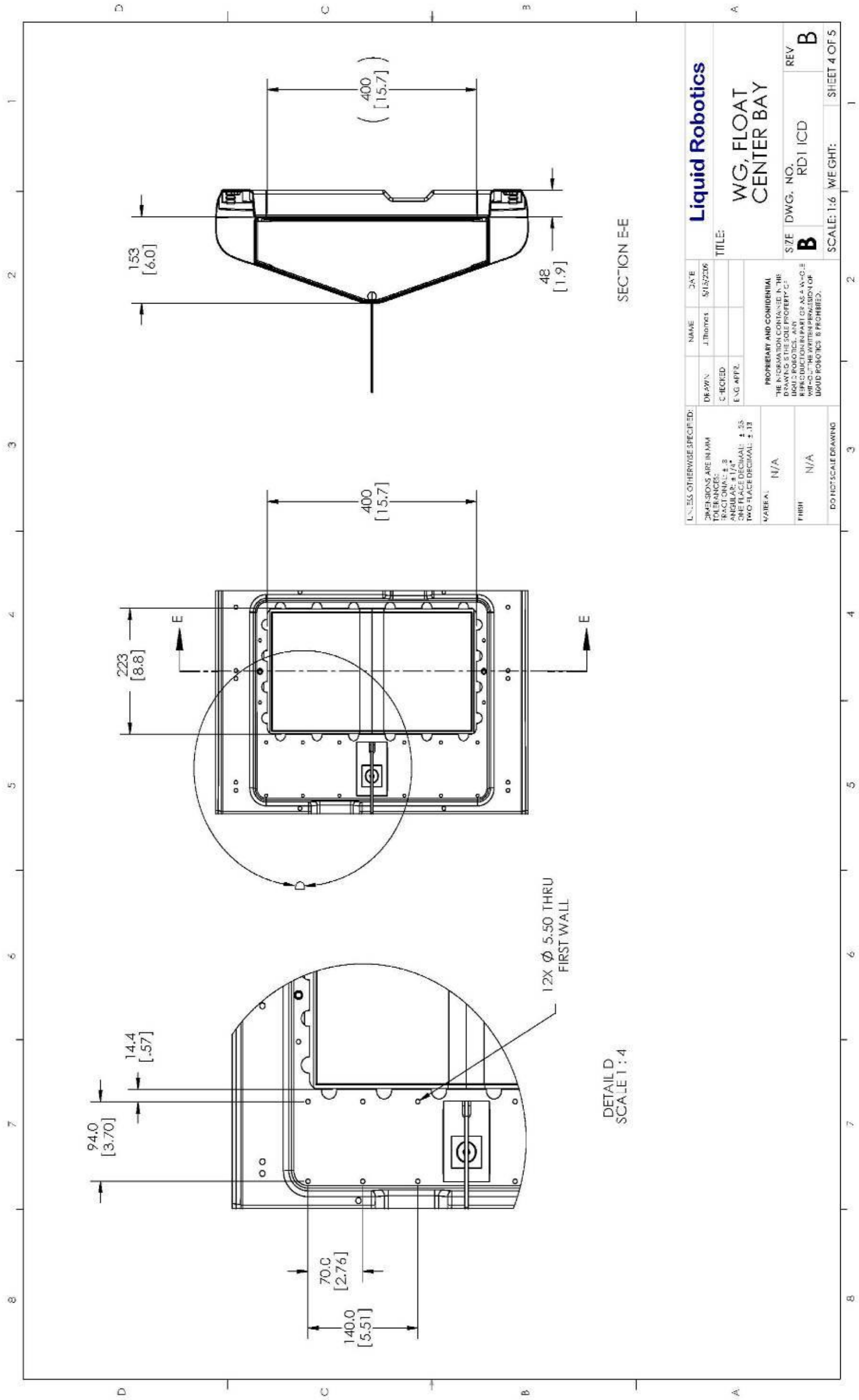


Figure 2-3: Float Center Bay Dimensions

2.2.2 Float Forward Payload Bay

The Forward Payload Bay is shorter but deeper than the Aft Payload Bay. It also has a constant fore-aft cross-section.

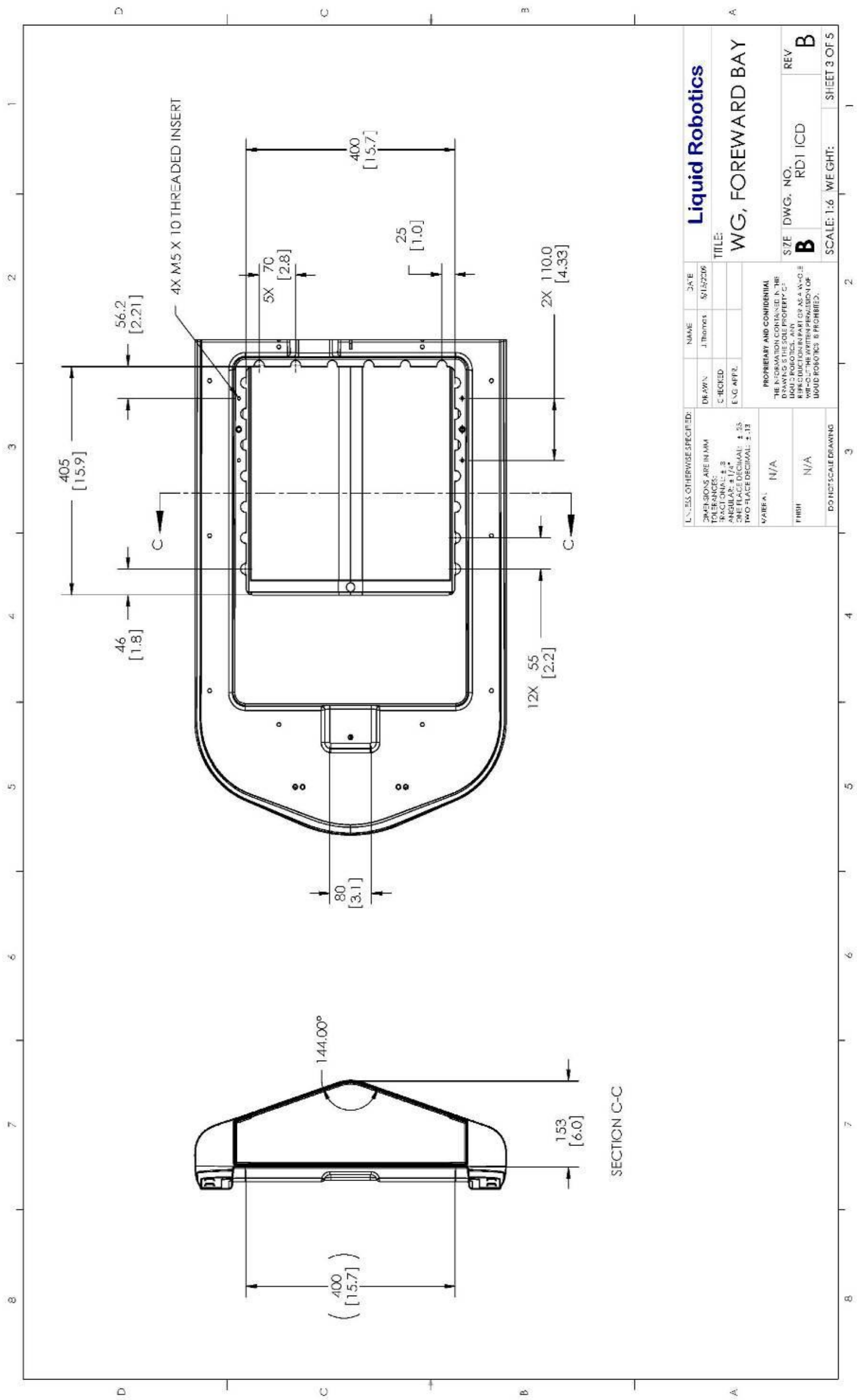


Figure 2-4: Forward Payload Bay Dimensions

2.2.3 Float Aft Payload Bay

The Aft Payload Bay is longer but shallower than the Forward Payload Bay. It also does not have a constant cross-section, as shown below.

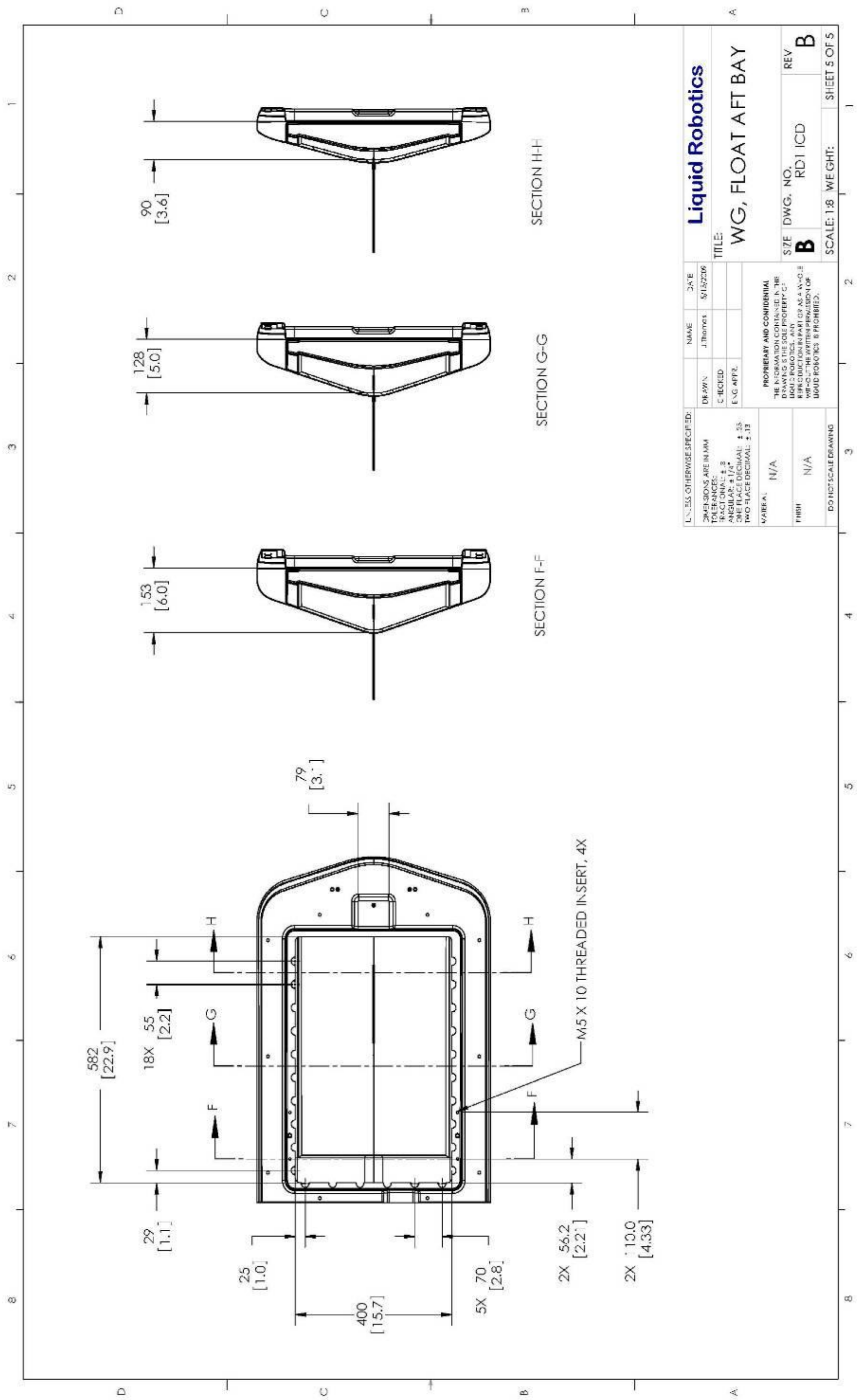


Figure 2-5: Aft Payload Bay Dimensions

2.3 C&C Drybox

For reference, the dimensions of the C&C Drybox are shown below. Refer to section 4.2 “Standard Electrical Connections" for detail information on the connector pinouts.

The basic dimensions of the Original Drybox and the Revised Drybox are the same; they are basically interchangeable in the Float. The primary external differences are on the top of the box.

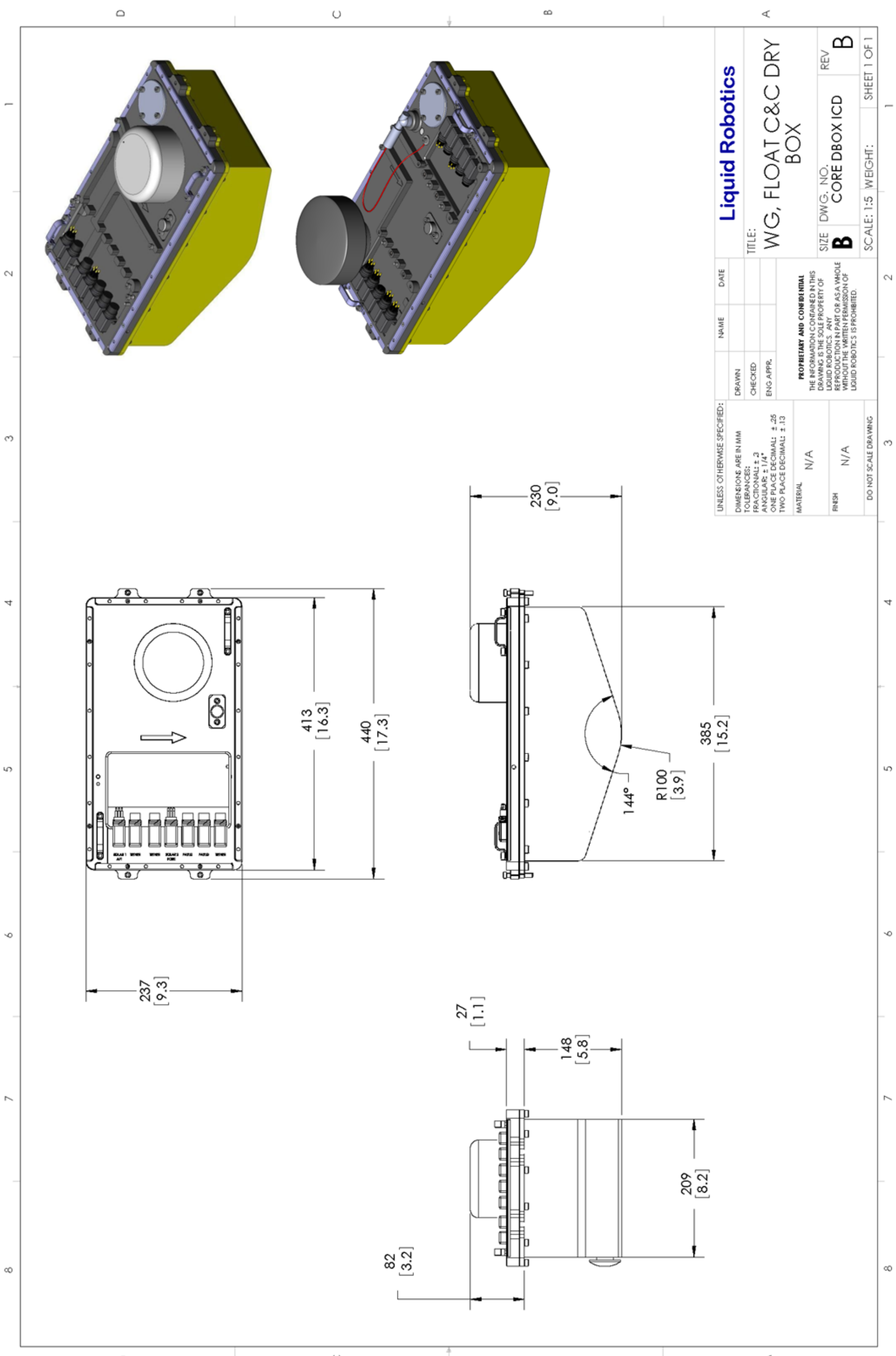


Figure 2-6: C&C Drybox Dimensions

2.4 Payload Dryboxes

2.4.1 Forward Payload Drybox

The overall dimensions of the Forward Drybox are shown below. Note that the lids for both the Forward and Aft Payload dryboxes are identical. Only the bottom casing is different.

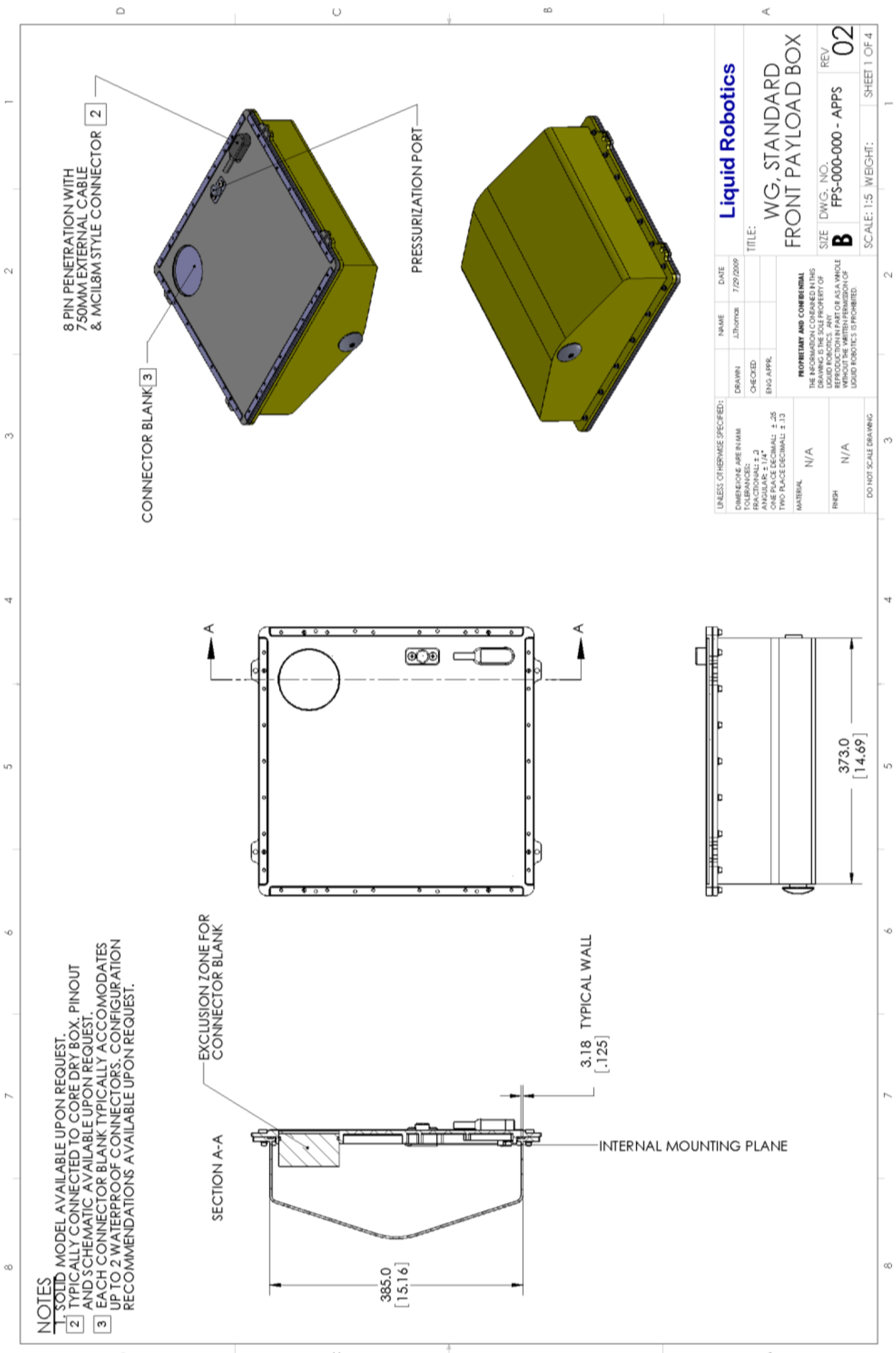


Figure 2-7: Fwd Drybox Dimensions

2.4.2 Aft Payload Drybox

The overall dimensions of the Aft Drybox are shown below. Note that the lids for both the Aft and Forward Payload dryboxes are identical. Only the bottom casing is different.

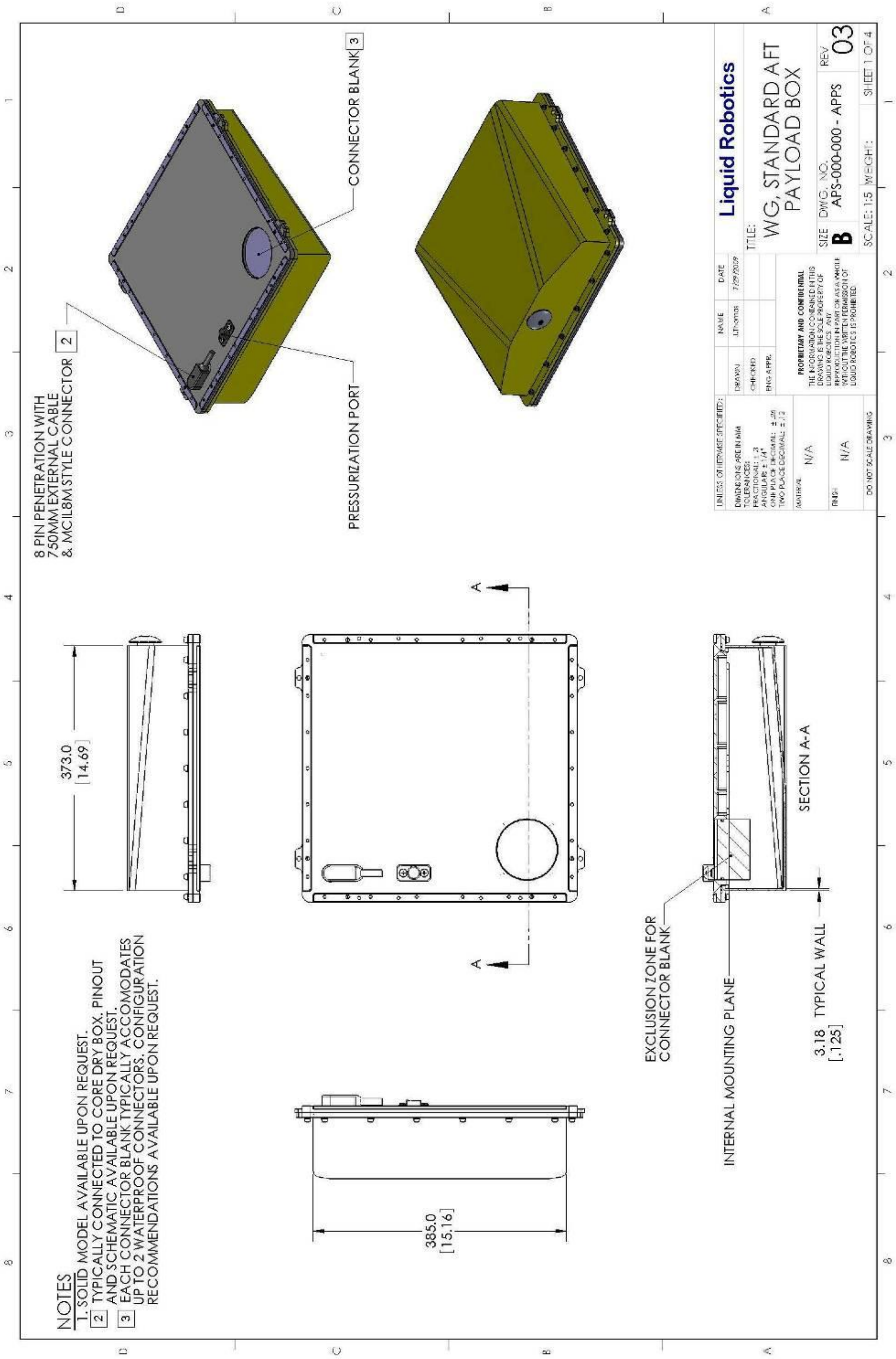


Figure 2-8: Aft Drybox Dimensions

2.4.3 Drybox Lid Mounting

The fore and aft drybox lids are identical. The lid underside has several bosses for mounting electronics and other equipment. There is also a removable panel for adding additional pass-thru connectors.

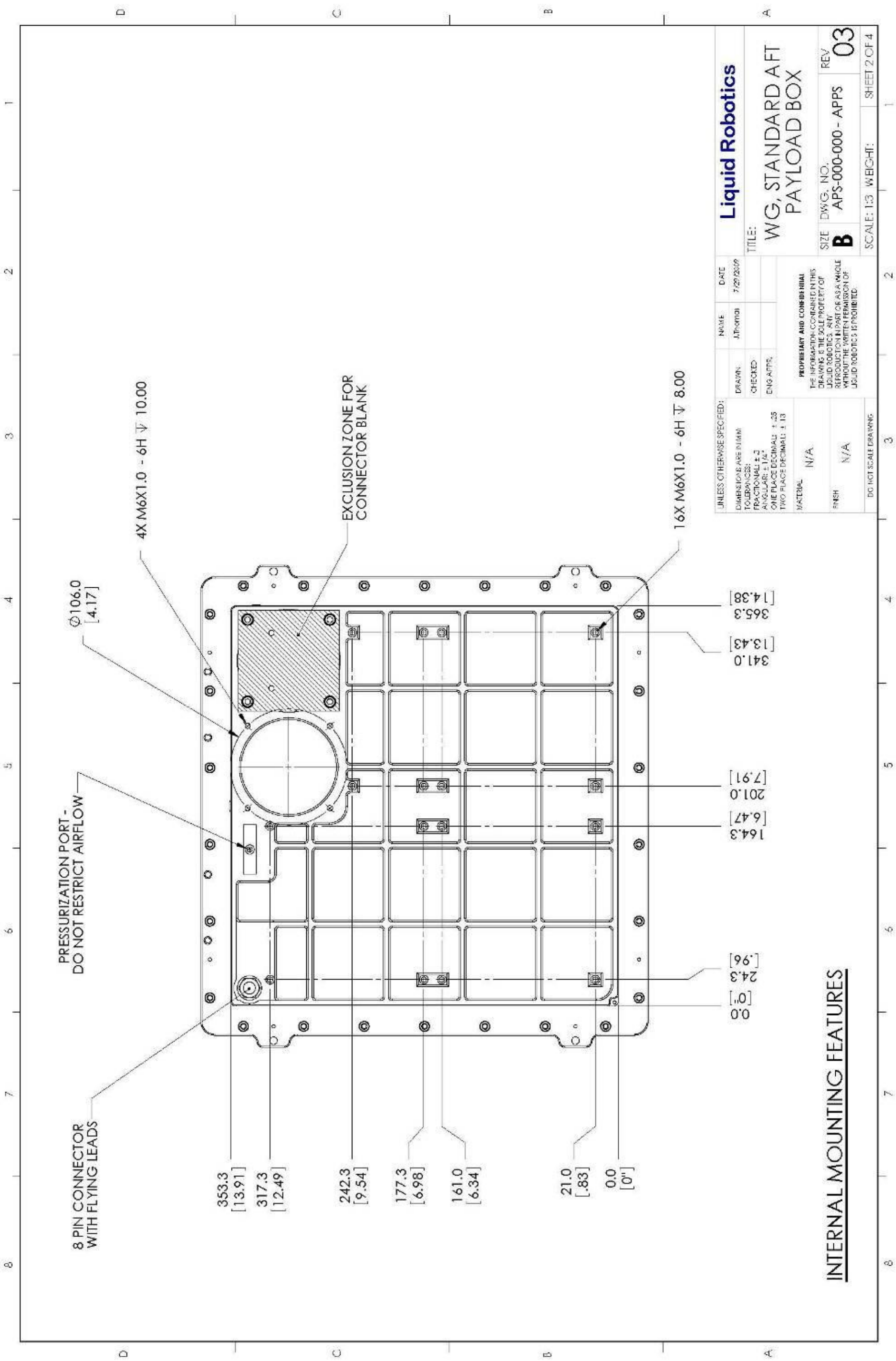


Figure 2-9: Drybox Lid Underside Dimensions

2.4.4 Aft Drybox Lid With Payload Interface

The drybox lid can be optionally equipped with a Payload Interface Board ("PIB"). Up to three PIBs can be mounted on the underside of the lid. Refer to the *Payload Electronics Guide* for more information on PIBs.

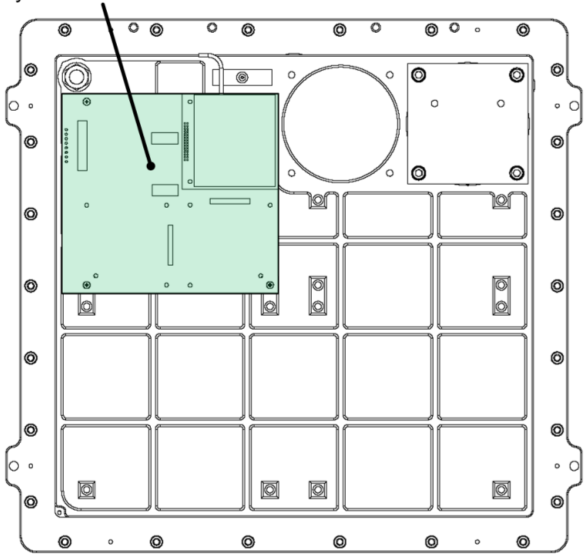


Figure 2-10: Drybox Lid With Interface Module

2.4.5 Aft Drybox Lid With Dual Connector Blanks

The lid can be optionally equipped with two Connector Blanks for pass-thrus. The exclusion areas are shown below.

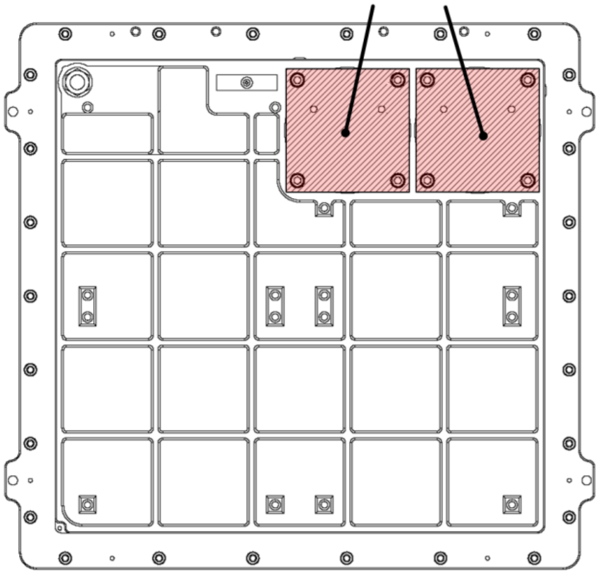


Figure 2-11: Drybox Lid With Two Connector Blanks

2.5 Float Considerations

2.5.1 Min/Max Weight

The maximum allowed weight of the Float depends upon the underwater weight of the Glider. The table below shows the maximum allowed weight of the Float.

Glider Configuration	Float Maximum Weight
Glider at minimum weight	80 Kg (175 lb)
Glider at maximum allowed weight	60 Kg (130 lb)

Table 2-1: Float Min/Max Weights

2.5.2 Center-of-Gravity Range

For maximum efficiency, the horizontal CG of the Float has to be within a narrow range.

The CG value can be verified by placing the Float (without the Umbilical+Sub) in calm water and checking that the Turtledeck angle does not exceed the fore and aft ranges shown in the figures below. Ideally, the float should be slightly nose-high.

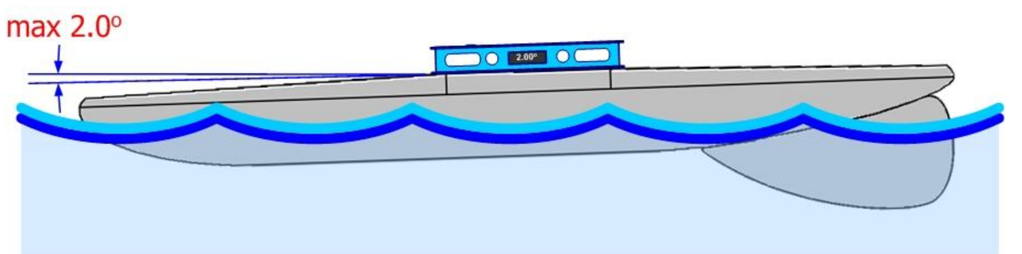


Figure 2-12: Float Max Forward CG

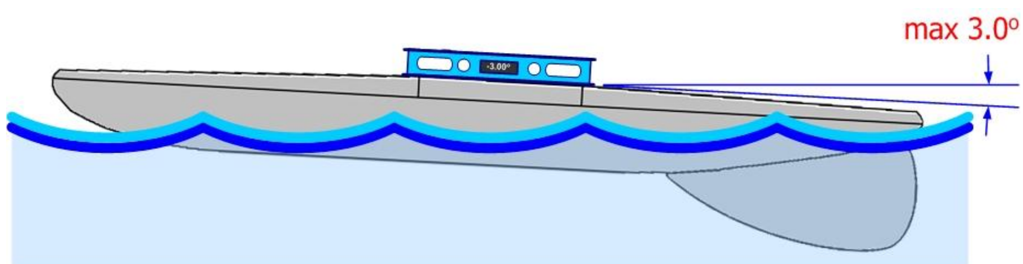


Figure 2-13: Float Max Aft CG

Also, any payloads must be ballasted, if necessary, so that the Float is <u>exactly</u> level from side-to-side, and does not list.

2.5.3 Float Sail Area

There are limits to the amount of sail area mounted to the top of the Float. The presence of the underwater Glider acts like a large keel to stabilize the Float, but tall masts with large or heavy tops should be avoided. Also, payloads with large sail areas in either the front or back should be avoided.

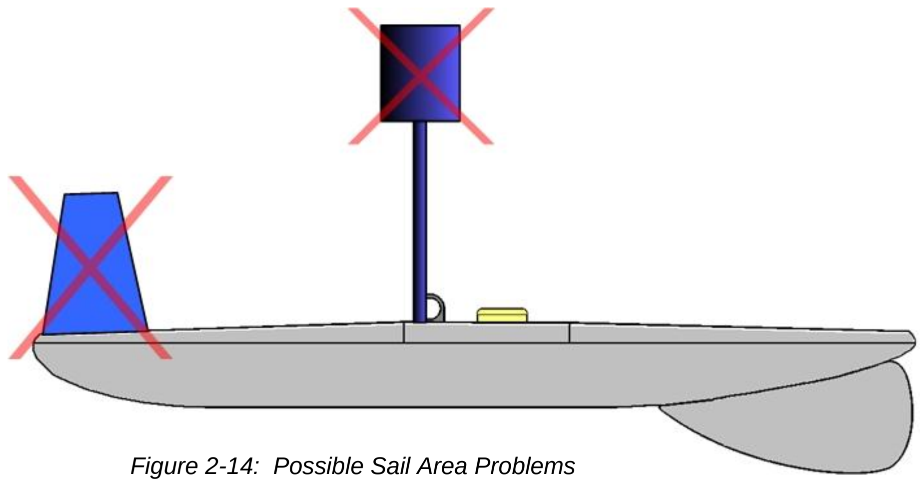


Figure 2-14: Possible Sail Area Problems

2.6 External Float Payloads

Payloads may be attached externally to the Float. However, consideration needs to be made for CG location, sail area, and covering of the solar panels.

2.6.1 Deck mounting

Mounting payloads to the top of the Float deck is acceptable if the payload does not significantly "shadow" the solar panels. The figure below shows a possible gunwale mounting and a possible stern mounting of payloads. Note that the payload on the left will partially shield the rear solar panel, reducing its output slightly at low sun angles.

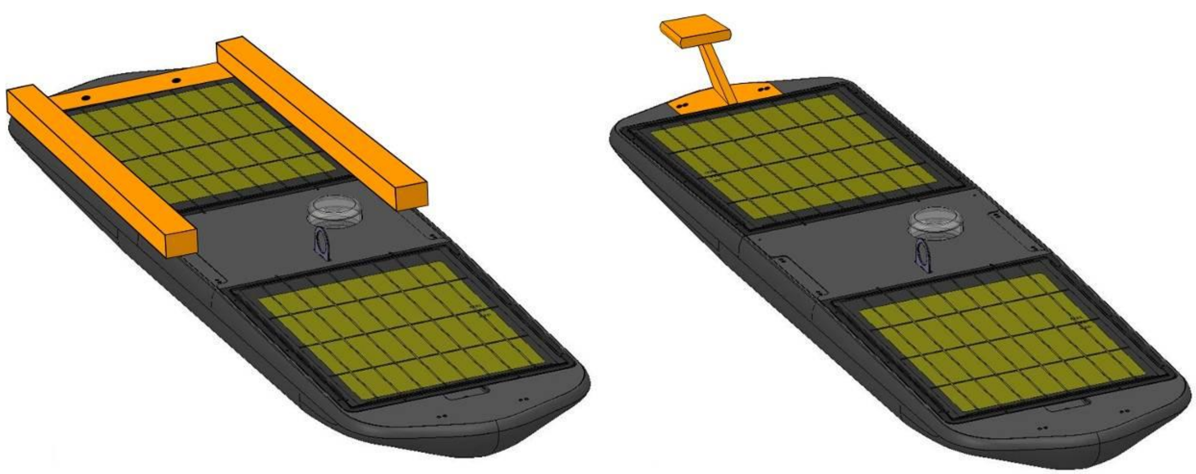


Figure 2-15: Deck Mounting of Payloads

2.6.2 Towing

Small payloads may be towed behind the Float, although the additional drag will slow the Vehicle. An example is shown in the figure below.

Figure 2-16: Towing Behind Float

2.6.3 Float Attach Points and Limits

There are 18 threaded attachment points on the top of the Float. Some are used to attach lift handles, but they can be used also for payload mounting. All attachments are M5 x ¾ threaded thru holes. The attachment points are shown in the figure below.

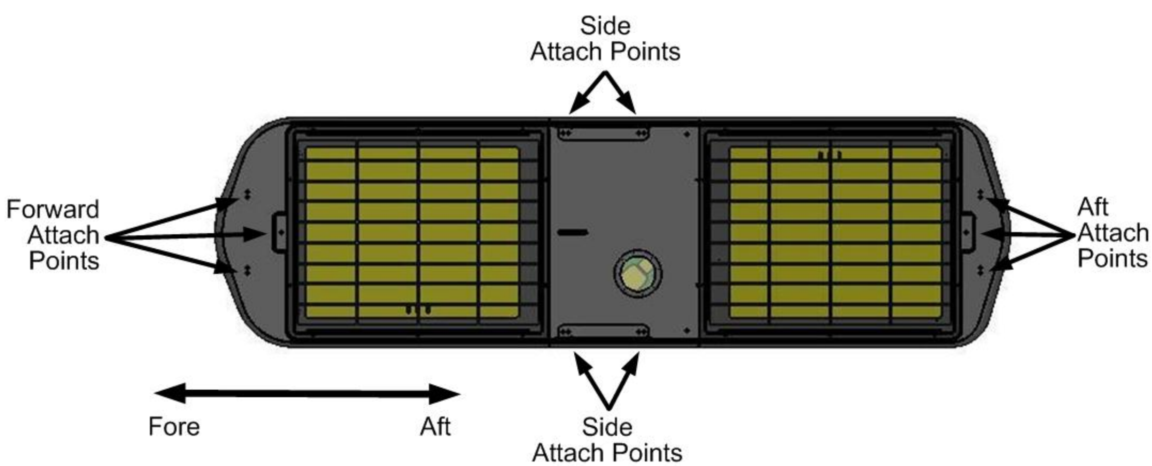


Figure 2-17: Float Attachment Points

The attachment points use metal threaded inserts. The maximum allowed stress on each attach points depends on its location, and is given in the Table below.

Position	Max Tension	Max Shear
Fore/Aft Attach	20 Kg (44 lbs)	60 kg (132 lbs)
Side Attach	15 Kg (33 lbs)	50 Kg (120 lbs)

Table 2-2: Attach Point Tension/Shear Limits

This Page Intentionally Left Blank

3. Glider Mechanical

3.1 Glider Overview

In the drawings shown on the following pages, dimensions are shown as millimeters [inches].

3.1.1 Glider Dimensions

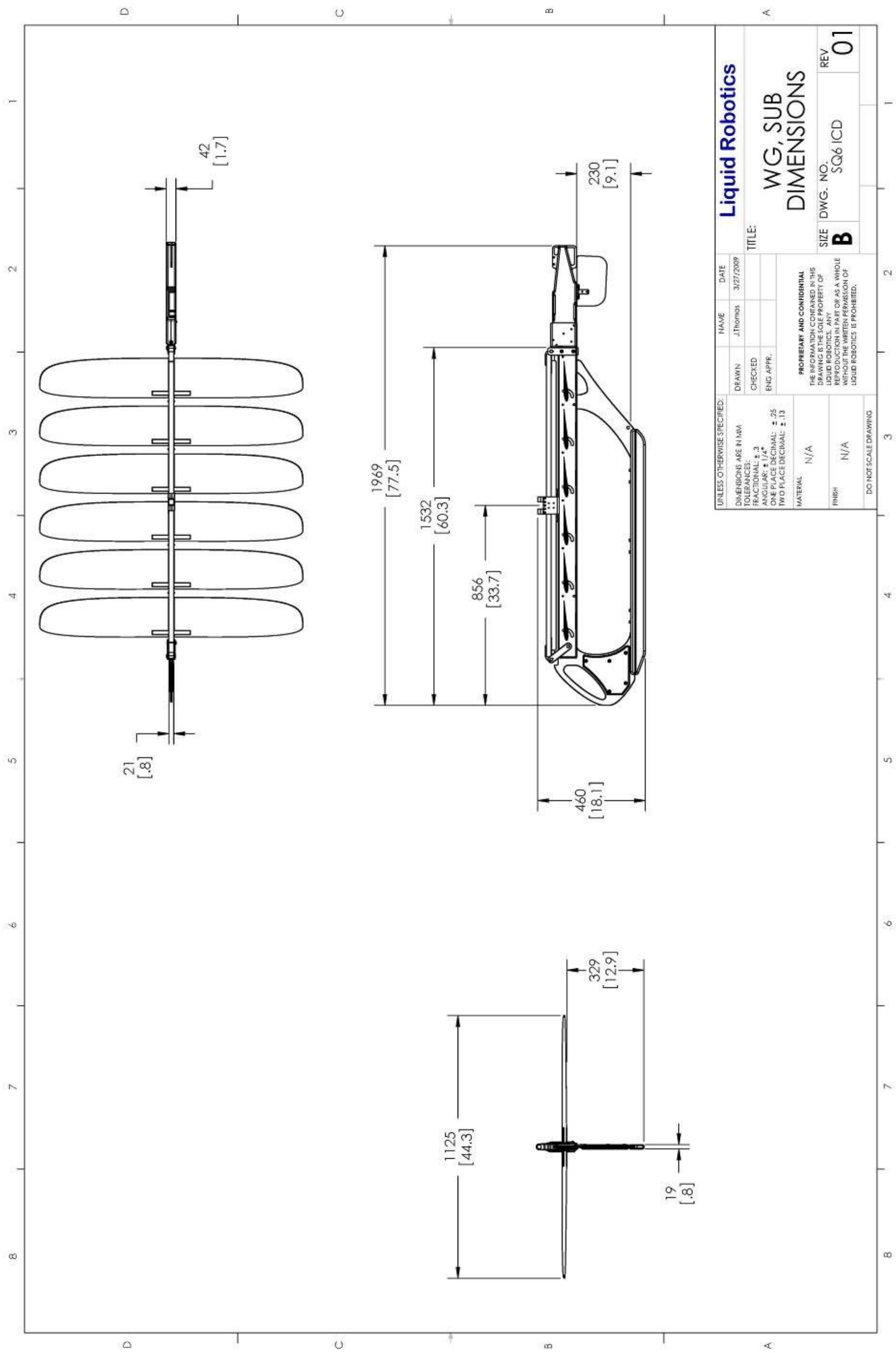


Figure 3-1: Glider Overall Dimensions

3.1.2 Attachment Point Locations

The Glider payload attachment points and dimensions are shown in the drawing below.
All attachment points use M5 x ¾ threads.

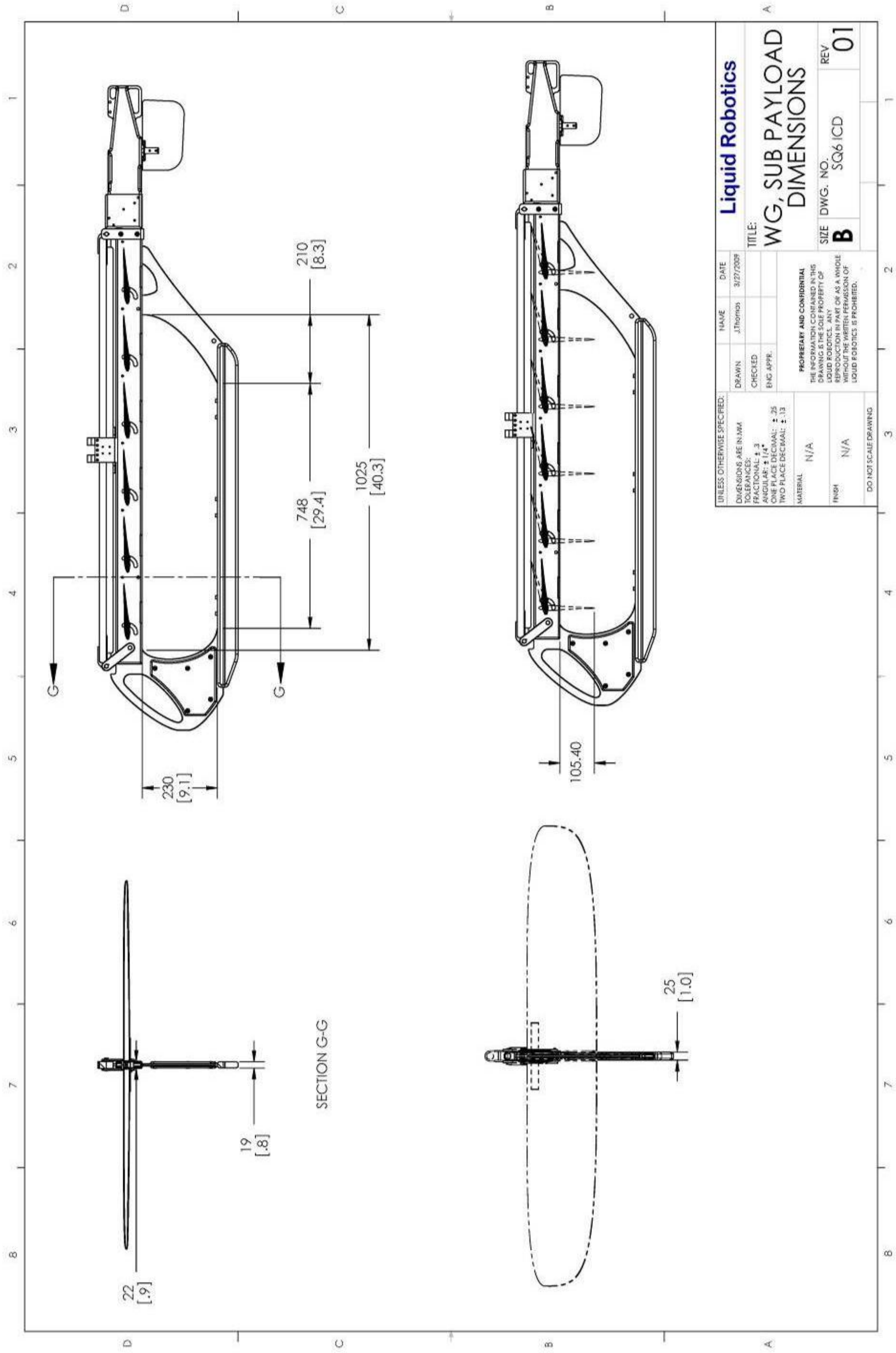


Figure 3-2: Glider Payload Attachment Points

3.2 Glider Payload Considerations

Since the Glider is always moving underwater, it is important to consider all drag, balance and floatation issues when designing a payload.

3.2.1 Drag

To maximize the efficiency of the Wave Glider system, the drag of any payload added to the Glider should be minimized both by reducing the frontal cross-sectional area, and by adding streamlined fairings if necessary.

Contrary to what might be assumed, the water flow direction over the Glider is not exactly from front to back. As shown in the two figures below, there is a substantial vertical component to the flow, especially when the Glider is being lifted up in heavy swells.

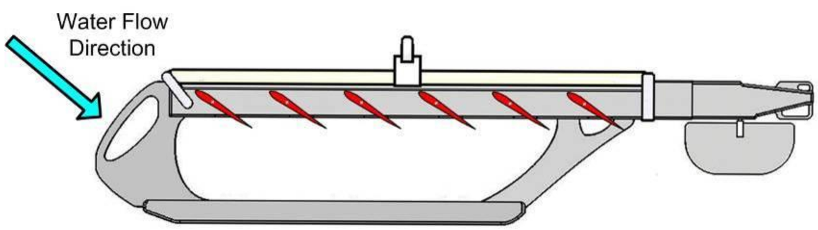


Figure 3-3: Glider Moving Up

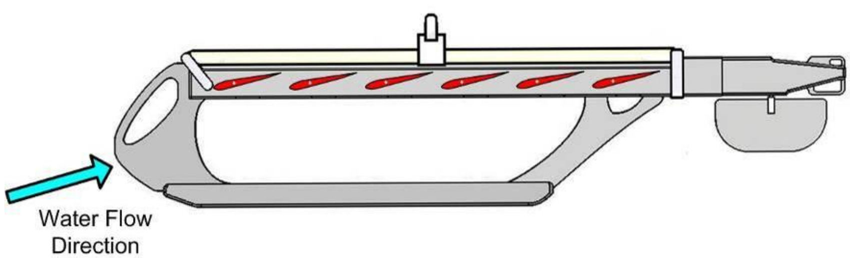


Figure 3-4: Glider Moving Down

Thus, a seemingly "streamlined" payload, like that shown in Figure 3-5, would create a substantial nose-down force (as well as drag) when the Glider is moving up.

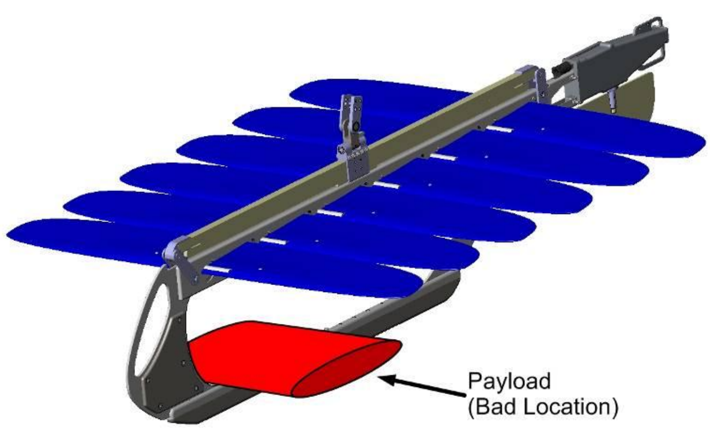


Figure 3-5: Poor Payload Design/Location

3.2.2 Glider Payload Bay

The open frame of the Glider is designed to hold an 800mm x 200mm x 20mm electronics housing, as shown in the figure below. As is shown, the payload electrical connection is through the Rudder Module. (The Rudder Cover is removed to show the connector detail.)

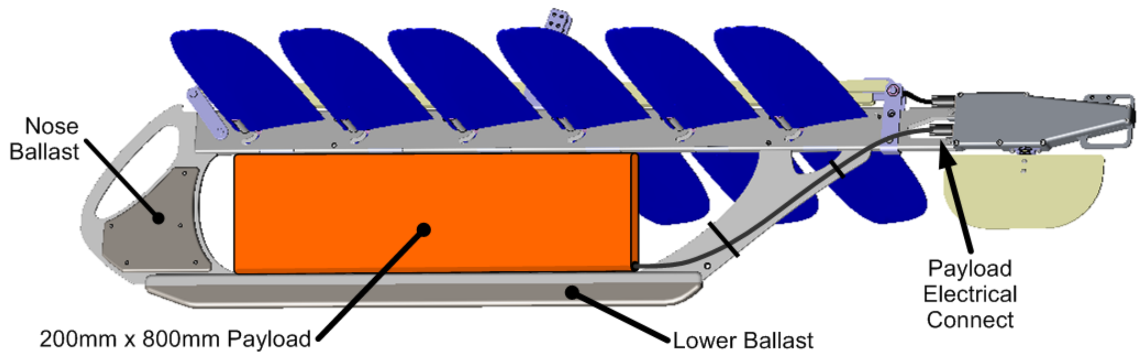


Figure 3-6: Glider with Payload Box

Any payload within the frame of the Glider has to consider the wing clearance issue as described in section 3.2.4 "Wing-Payload Interference".

3.2.3 Glider Ballast Weights

Ballast weights are attached to the nose and bottom of the Glider frame as shown in Figure 3-6 above. As payload is added, these weights may be removed or replaced with other weights to adjust the balance and buoyancy of the Glider.

The lower ballast weight can be shifted fore and aft. However, this should be done only for interim testing purposes, not for extended water operations.

3.2.4 Wing-Payload Interference

Note that since the Glider wings can rotate almost 90° to the vertical, any payload (or payload cover) mounted within the main frame of the glider has to consider the clearance of the wings in their fully rotated position. This is shown in the figure below. The clearance values are specified in the drawing in section 3.1.2 "Attachment Point Locations".

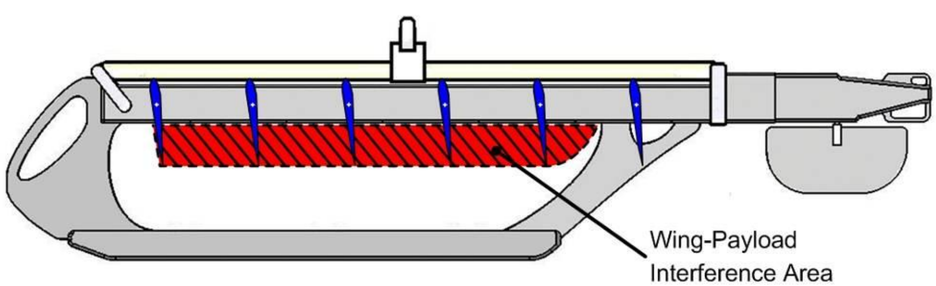


Figure 3-7: Wing-Payload Interference Area

3.2.5 Balance and Floatation

Like the Float, a Glider with added payload needs to be balanced about the Umbilical pivot point. Since the glider is submerged, the balance needs to be established underwater, to account for both the floatation and weight of the payload.

The best way to compute the floatation/balance of the Glider is to support it by the Umbilical attach point and check the deck angle while the Glider is submerged.

The Glider should be either level, or else slightly nose-down. The maximum nose-down angle is shown in the figure below.

<u>The Glider should never be ballasted nose-up</u>.

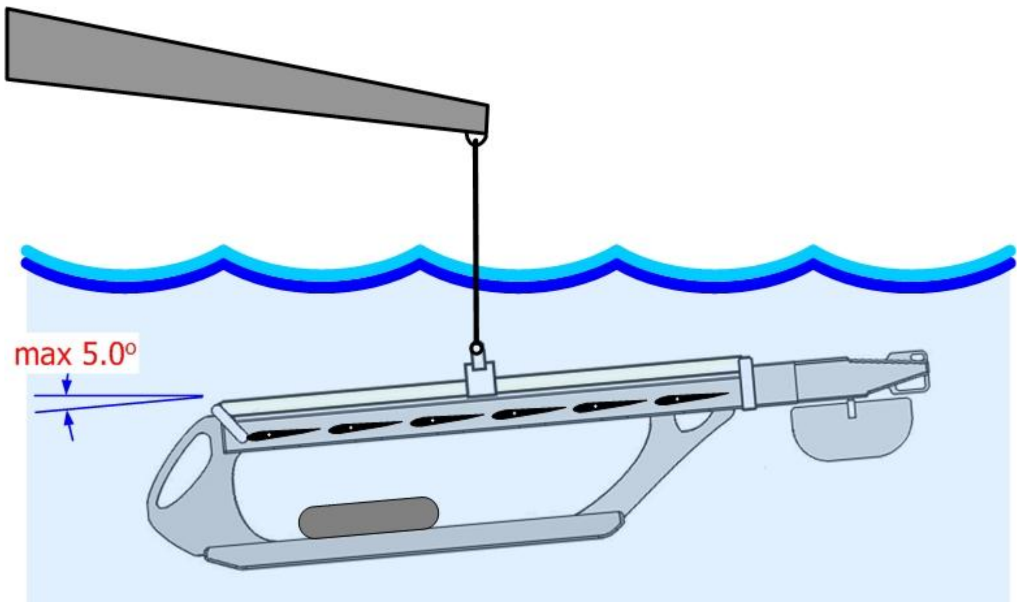


Figure 3-8: Checking Glider Underwater Balance

The Glider+Payload floatation likewise needs to be checked while the Glider is underwater, to account for both the weight and flotation of the payload. The minimum and maximum <u>submerged</u> weights are given in the Table below:

Minimum Weight	40 Kg (88 lbs)
Maximum Weight	60 Kg (130 lbs)

Table 3-1: Min/Max Underwater Glider Weights

3.2.6 Poor Sail Area Locations

Glider payloads with a large sail area can be a problem if the payload is mounted forward or aft of the Glider, as shown in the figures below.

A large sail area in front of the Glider can destabilize and make it over-sensitive to rudder inputs.

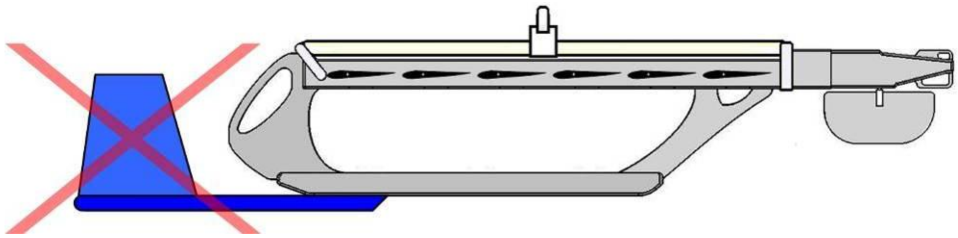


Figure 3-9: Excessive Forward Payload Sail Area

Conversely, a large sail area aft of the Glider can over-stabilize and reduce the rudder effectiveness.

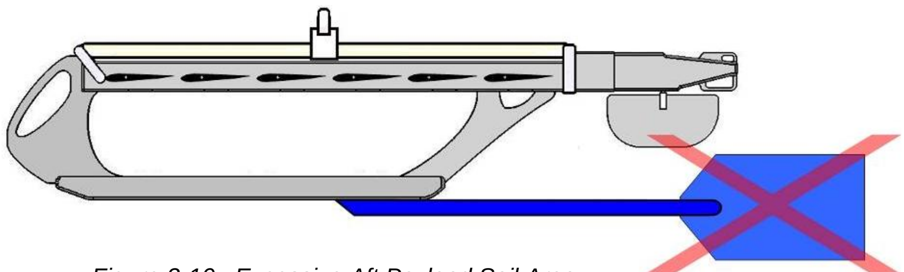


Figure 3-10: Excessive Aft Payload Sail Area

3.2.7 Acceptable Sail Area Locations

Generally, a reasonably-sized payload sail area located generally below the Umbilical attach point should not materially affect stability, though there can be a significant increase in drag for payloads mounted outside the frame.

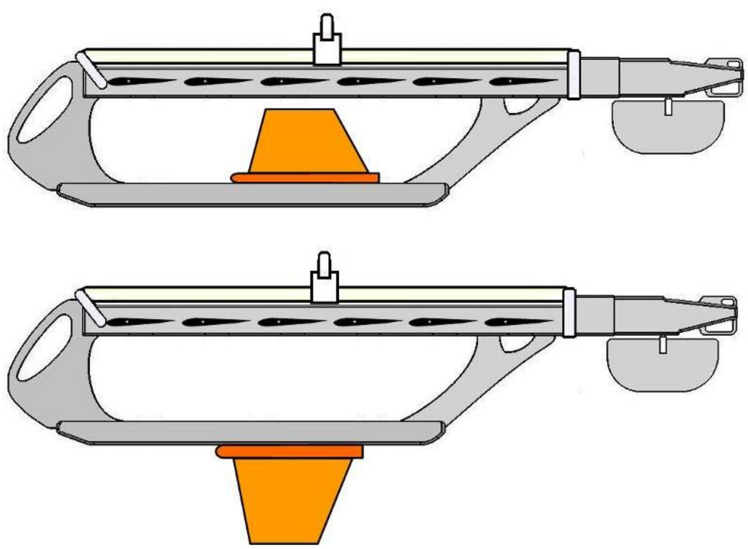


Figure 3-11: Acceptable Payload Sail Area Locations

3.2.8 Trailing Wires

Trailing wires behind the Glider is generally not recommended, as they can potentially foul the Rudder. However, if such a wire is necessary, then the wire should be mounted near the center of the Glider, and not trail off the back. In addition to fouling, rear trailing wires can have the same over-stabilizing effect as a large rear sail area.

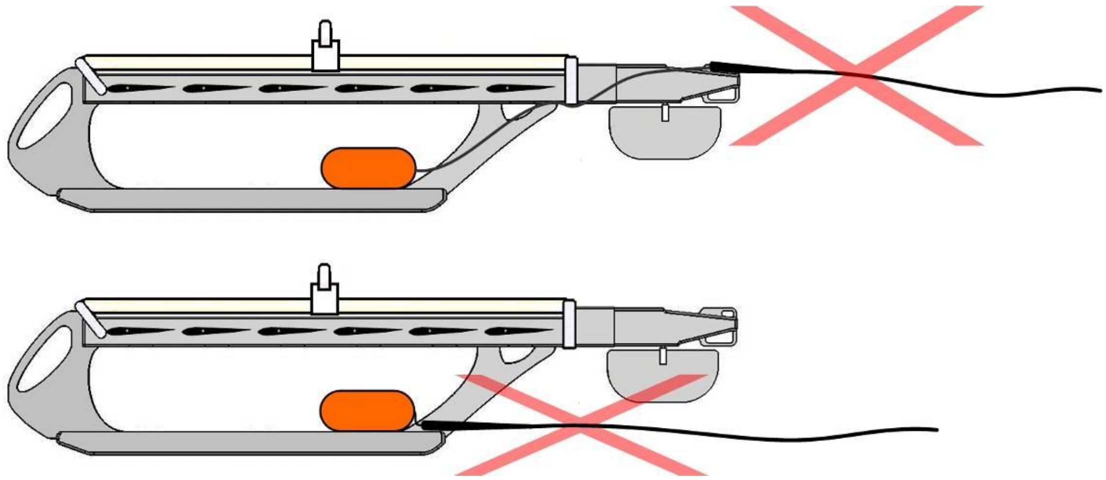


Figure 3-12: Poor Trailing Wire Locations

The best location for a trailing wire mounting is on a pylon directly below the Umbilical attach point, as shown in the figure below.

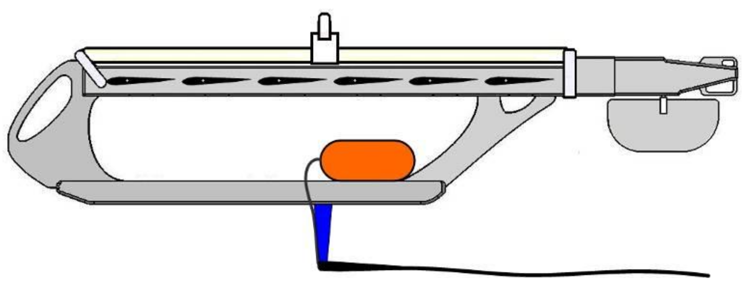


Figure 3-13: Acceptable Trailing Wire Location

Keep in mind that the Glider is continually moving up and down while moving forward. Any trailing wire must be short enough and mounted such that it cannot foul the rudder, regardless of the angle of the wire.

This Page Intentionally Left Blank

4. Electrical Specifications

4.1 Electrical Overview

4.1.1 Electrical System Schematic

A detailed drawing of the Wave Glider electrical system is shown in the figure below. This shows a fully-configured system with both Float and Glider Payloads. Not shown are the auxiliary connections on both the original and revised C&C

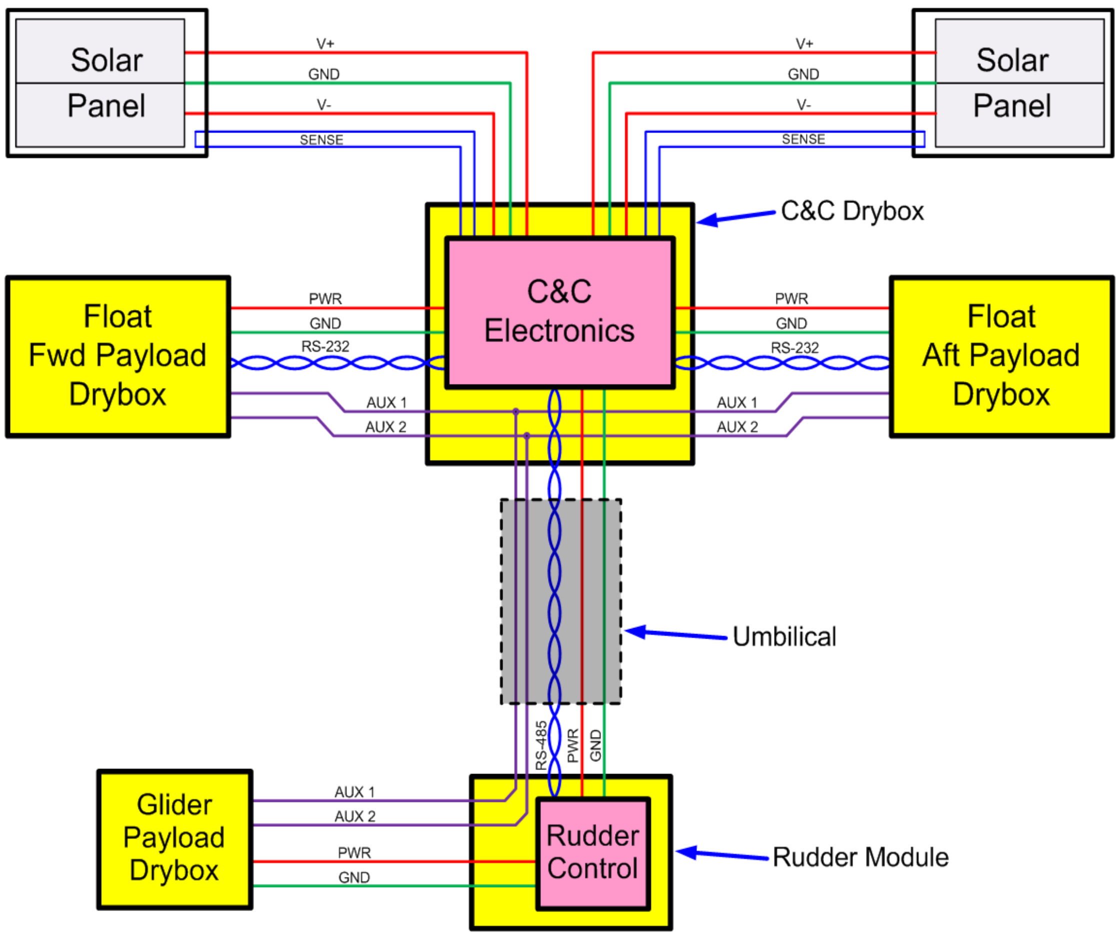


Figure 4-1: Wave Glider Electrical Schematic

4.1.2 Solar Panels

Battery charging is via Solar Panels mounted on the Float. The Float can operate with a single solar panel if electrical loads are kept to a minimum.

A two-wire sensing line is also connected to the solar panels. On the Original Drybox, the C&C computer is turned on by shorting at least one set of lines (i.e., connecting one panel). On the Revised Drybox, the sensing lines are used by the C&C to check whether a panel is connected.

Out-of-water battery charging is possible through the solar panel connections. Refer to section 4.3 "External Charging" for more information.

4.1.3 Auxiliary Common Payload Connections

As shown in Figure 4-1**, AUX 1** and **AUX 2** are connected to both the Float and the (optional) Glider payload, but not to the C&C electronics. These lines can be used for digital or analog signaling between the payloads. However, the voltages on these lines are not connected to, or otherwise sensed by, the C&C computer.

Communication speed on these lines will depend on the protocol used. It is recommended that for RS-232, the speed be limited to 115,200 bps when communicating with a Glider payload.

4.1.4 Fore/Aft Float Payloads

The Fore and Aft payloads each have a 12V power (and ground) connection. For payload communications, there is a two-wire RS-232 connection to the C&C. The RS-232 communications should not be operated at speeds higher than 115,200 bps.

The payloads are also connected to the auxiliary 2-wire "bus" (ref. 4.1.3).

4.1.5 Glider Payload

The Glider payload has a 12V power and ground connection. There is no specific communication channel for the Glider payload.

However, any Glider payload can connect to the 2-wire payload "bus" (ref. 4.1.3). Thus, any communications with the Glider payload can be done via **Aux 1** and **Aux 2** to either of the Float payloads. Note that the Glider payload cannot communicate directly with the C&C.

4.1.6 Payload Maximum Currents

Each of the Float payload (PAYLD 1 and PAYLD 2) connectors, as well as the Umbilical connector provide a power source. However, the voltage value and maximum currents for each of the connectors depends upon the version of the C&C used.

- For the Original Drybox, the output voltage is essentially the battery voltage, which can vary from about 18.5 VDC (fully charged) down to about 11.5 VDC (discharged). The maximum current for each payload is **1.25 Amps**.
- For the Revised Drybox, the output voltage is regulated to 13.2 VDC. The allowed current for each payload is **4.0 Amps**.
- For all Glider payloads combined, the maximum current allowed is **1.0 Amp**.

Attempting to exceed the current limits on any output will cause a shut-down of the voltage to that output.

4.1.7 Maximum Total Current Output

The above limits apply to each payload separately. However, there is a maximum allowed total current for all payloads combined. It is **5.0 Amps**.

4.2 Standard Electrical Connections

There are basically two types of watertight connectors used on the Wave Glider: 5-Pin and 8-Pin. The tables below describe the connector type and pinouts for each location.

4.2.1 C&C Solar Panel Connections

The two **SOLAR1** and **SOLAR2** connectors on the top of the C&C Drybox are SubConn 5-pin Male type. These connections can be used with an external charger to recharge the batteries. Refer to section 4.3 "External Charging".

Pin #	SOLAR Function	Description	Notes
1	V+	Positive Solar Panel voltage	
2	V–	Negative Solar Panel voltage	
3	GND PWR	Return for V+ and V- inputs	
4	Sense	Continuity Sensor for Solar Panel	[1] [2]
5	Sense Ref	Reference for Continuity Sensor	[1] [2]

SubConn MCBHRA5M

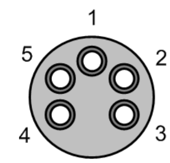


Table 4-1: Solar Panel Connector Specifications

[1] On the Original C&C drybox, the **Sense** and **Sense Ref** are used to turn the C&C on or off.

[2] On the Revised C&C, the **Sense** and **Sense Ref** are used to verify that the solar panel is connected. Thus, no solar panels are required to be connected for power to be turned on.

4.2.2 C&C Payload Connections

The two Float **PAYLD1** and **PAYLD2** connectors on the top of the C&C Drybox are SubConn 8-pin female type, with the following connections:

Pin #	PAYLD Function	Description	Notes
1	GPS NMEA Data	4800 Baud GGA, ZDA, and VTG, using RS-232 levels	[4] [5]
2	PWR	Original C&C: 11.6 – 16.6 Volts Revised C&C: 13.2 Volts regulated	[1] [2]
3	Tx	RS-232 Transmit Line	
4	Rx	RS-232 Receive Line	
5	AUX 1	Payload common #1	[3]
6	AUX 2	Payload common #2	[3]
7	GND	Ground reference for all pins	
8	GPS PPS	GPS pulse-per-second, using RS-232 levels	[5]

SubConn MCBHRA8F

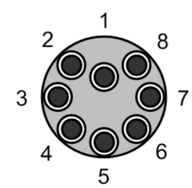


Table 4-2: C&C Payload Connector Specifications

[1] For Original C&C Drybox, the maximum allowed current is **1.0 Amps**. See also section 4.4 "Power Considerations".

[2] For the Revised C&C Drybox, the maximum current is **4.0 Amps**. See also section 4.4.

[3] **AUX 1** and **AUX 2** connect also to pins 5 and 6 on the Glider rudder box payload connector (ref. 4.2.4).

[4] GPS: 4800 baud GGA (position), ZDA (date/time) and VTG (velocity) messages.
These GPS signals are on the Revised C&C only.

[5] For the Original C&C, Pins 1 and 8 are No-Connects.

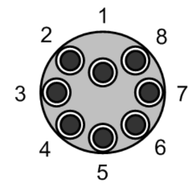
4.2.3 C&C Umbilical Connections

There is generally no need to interface directly with the Umbilical connection cable. However, for possible electrical troubleshooting, it is described here.

The Umbilical (Tether) connector on the C&C Drybox is a SubConn 8-pin **female** type. The connector on the Rudder Box is an 8-pin **male**. Both use the connections shown in Table 4-3 below.

Pin #	UMBILICAL Function	Description	Notes
1	SHLD	Shielding (grounded)	
2	PWR	Original C&C: 11.6 – 16.6 Volts Revised C&C: 13.2 Volts regulated	[1] [2]
3	A	Rudder Control A	
4	B	Rudder Control B	
5	AUX 1	Payload common #1	[3]
6	AUX 2	Payload common #2	[3]
7	GND	Ground reference for all pins	
8	Open	Not Used	

SubConn MCBHRA8F



Female Face View

Table 4-3: C&C Umbilical Connector Specifications

[1] For Original C&C Drybox, the maximum allowed current is **1.0 Amps**. See also section 4.4 “Power Considerations”.

[2] For the Revised C&C Drybox, the maximum current is **4.0 Amps**. See also section 4.4.

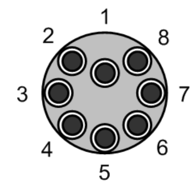
[3] **AUX 1** and **AUX 2** connect also to pins 5 and 6 on the Glider rudder box payload connector (ref. 4.2.4).

4.2.4 Glider Payload Connection

The optional lower connector on the Glider's Rudder Module is a SubConn 8-pin female type. It has the connections:

Pin #	Function	Description	Notes
1	SHLD	Shielding (grounded)	
2	PWR	Original C&C: 11.6 – 16.6 Volts Revised C&C: 13.2 Volts regulated	[1]
3	(reserved)	DO NOT USE	[2]
4	(reserved)	DO NOT USE	[2]
5	AUX 1	Payload common #1	[3]
6	AUX 2	Payload common #2	[3]
7	GND	Ground reference for all pins	
8	Open	Not Used	

SubConn MCBHRA8F



Female Face View

Table 4-4: Glider Payload Connector Specifications

[1] The maximum allowed current is **1.0 Amps**, (the same for both the Original and Revised C&C dryboxes). See also section 4.4 "Power Considerations".

[2] Pins 3 and 4 are reserved for use by the Wave Glider. These pins <u>must not</u> be connected to any Glider payload.

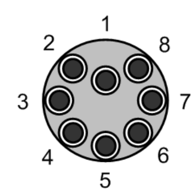
[3] **AUX 1** and **AUX 2** connect also to pins 5 and 6 on both of the Float Payload connectors (ref. 4.2.2).

4.2.5 Revised C&C "WEATHER" Connections (Original "I/O 1")

The **I/O 1** connector on the Original C&C and the **WEATHER** connector on the Revised C&C have the same pinouts. Both use a SubConn 8-pin female type connector.

Pin #	WEATHER Function	Description	Notes
1	N/C	Not Used	
2	PWR	Original C&C: 11.6 – 16.6 Volts Revised C&C: 13.2 Volts regulated	
3	RXD-	RS-422 receive from station	[1]
4	RXD+	RS-422 receive from station	[1]
5	TXD-	Not used	[2]
6	TXD+	Not used	[2]
7	GND	Ground	
8	N/C	Not Used	

SubConn MCBHRA8F



Female Face View

Table 4-5: Revised C&C Weather Connector Specifications

[1] Only the receive portion of the RS-422 is used.

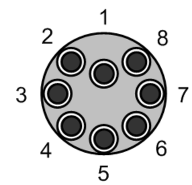
[2] The transmit RS-422 connections are available on the connector, but are not used.

4.2.6 Revised C&C "SPEED" Connection (Original "I/O 2")

The **I/O 2** connector on the Original C&C and the **SPEED** connector on the Revised C&C have the same pinouts. Both use a SubConn 8-pin female type, with the connections:

Pin #	SPEED Function	Description	Notes
1	GND	Ground	
2	N/C	Not Used	
3	PWR	Original C&C: 11.6 – 16.6 Volts Revised C&C: 13.2 Volts regulated	
4	Data	Speed Data	[1]
5	N/C	Not Used	
6	N/C	Not Used	
7	N/C	Not Used	
8	N/C	Not Used	

SubConn MCBHRA8F



Female Face View

Table 4-6: Revised C&C Speed Connector Specifications

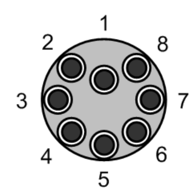
[1] Speed data is a variable-frequency square wave.

4.2.7 Revised C&C "SATCOM" Connections

The **SATCOM** connector on the Revised C&C Drybox is a SubConn 8-pin female type, with the connections:

Pin #	SATCOM Function	Description	Notes
1	N/C	Not Used	
2	PWR	13.2 Volts regulated	[1]
3	(reserved)	DO NOT USE	[2]
4	(reserved)	DO NOT USE	[2]
5	Aux 1	Payload common #1	[3]
6	Aux 2	Payload common #2	[3]
7	GND	Ground reference for all pins	
8	Open	Not Used	

SubConn MCBHRA8F



Female Face View

Table 4-7: Revised C&C Satcom Connector Specifications

[1] The maximum allowed current is **1.0 Amps**.

[2] Pins 3 and 4 are reserved for use by the Wave Glider. These pins <u>must not</u> be connected.

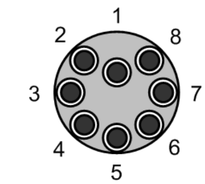
[3] **AUX 1** and **AUX 2** connect also to pins 5 and 6 on the Float Payload connectors (ref. 4.2.2).

4.2.8 Revised C&C "CAMERA" Connections

The **CAMERA** connector on the Revised C&C Drybox is a SubConn 8-pin female type, with the connections:

Pin #	CAMERA Function	Description	Notes
1	TBD		
2	TBD		
3	TBD		
4	TBD		
5	TBD		
6	TBD		
7	TBD		
8	TBD		

SubConn MCBHRA8F



Female Face View

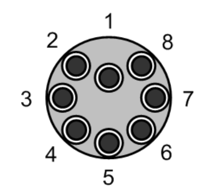
Table 4-8: Revised C&C Camera Connector Specifications

4.2.9 Revised C&C "RADAR" Connections

The **RADAR** connector on the Revised C&C Drybox is a SubConn 8-pin female type, with the connections:

Pin #	RADAR Function	Description	Notes
1	TBD		
2	TBD		
3	TBD		
4	TBD		
5	TBD		
6	TBD		
7	TBD		
8	TBD		

SubConn MCBHRA8F



Female Face View

Table 4-9: Revised C&C Radar Connector Specifications

4.3 External Charging

4.3.1 Charger Connections

An external charger can be connected to either or both of the C&C solar panel connectors. This allows charging or “topping-off” of the batteries pre-mission, or when bench testing.

The maximum charging current for each connection will be internally limited to **2.5 Amps**, regardless of what the charger is capable of supplying. For faster charging, connect a charger to each of the two solar panel connectors.

4.3.2 LRI-Supplied Charger

Liquid Robotics Inc. offers a pre-made external charger specifically designed for the Wave Glider.

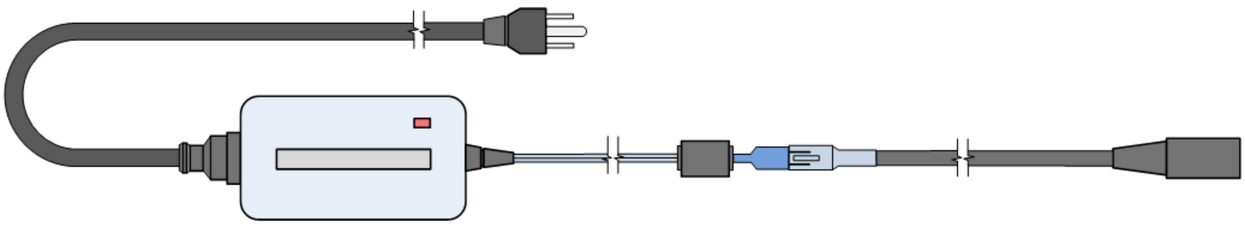


Figure 4-2: LRI-Supplied Charger

Contact Liquid Robotics for more information.

4.3.3 Custom-Made Charger

A custom Wave Glider battery charger can be made from a commercial power supply and a SubConn connector.

The power supply voltage must be **+9.0 VDC**. Charging is done using a fixed voltage. <u>Do not use a constant-current source</u>.

In order to minimize charging time, the charger should be capable of delivering at least **2 Amps**. A higher rating is recommended.

The connector required to mate with the solar panel connector is a MCBHRA5F SubConn 5-pin Female type, with the following connections:

Pin #	Function	Description
1	V+	To P/S Positive (+9VDC)
2	V–	N/C
3	GND	To P/S Negative
4	Sense	Connected Together
5	Sense Ref	

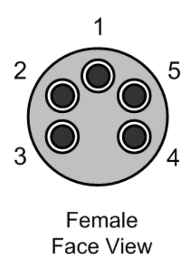


Table 4-10: External Charger Connections

With the Original C&C, the Wave Glider will power-up when the charger is connected.

With the Revised C&C, the power must be turned on (via the key) during charging.

4.4 Power Considerations

The Wave Glider has some specific limitations on the power used by the payloads.

Since the Wave Glider derives all of its long-term operating power from the solar panels, the average energy usage by the C&C Drybox plus payload(s) should not exceed the average energy input from the solar panels. Allowance also has to be made for the possibility that, due to storms and cloud cover, the solar energy may be considerably less than the "average" for several days at a time.

4.4.1 Maximum Current Limits

Table 4-11 below summarizes the maximum allowed currents for connections to both the Original and the Revised C&C Drybox. If any of these limits is exceeded, the C&C will shut down power to the offending component.

Max Current	Original Drybox	Revised Drybox	Notes
Total to all External	5.0 A	5.0 A	[1]
PAYLD 1	1.25 A	4.0 A	[2]
PAYLD 2	1.25 A	4.0 A	[2]
Glider Payloads (all)	1.0 A	1.0 A	[3
WEATHER (I/O 1)	TBD	TBD	
SPEED (I/O 2)	TBD	TBD	
SATCOM	TBD	n/a	
CAMERA	TBD	n/a	
RADAR	TBD	n/a	

Table 4-11: Maximum Currents for C&C Connections

[1] For the Revised Drybox, the sum of the allowed payload currents exceeds the maximum current draw. The maximum current draw must not be exceeded.

[2] If the payload current exceeds this limit, power will be shut down to the payload.

[3] If the Glider payload current exceeds this limit, power will be shut down to both the payload and the rudder control.

4.4.2 Transient Power Surges

It is extremely important that start-up and other transient currents be taken into consideration, as the C&C has very limited ability to absorb excess power when currents exceed the maximum values. If a payload's transient current value will exceed the limit, some kind of local energy storage (batteries or super-caps) is required.

Refer to the LRI *Payload Electronics Guide* for more detailed information on the transient power handling capabilities of the C&C electronics.

4.4.3 Sustainable Payload Power

Figure 4-3 shows the computed power available to a Wave Glider at different Northern latitudes and dates. The graph assumes that cloud cover reduces the solar flux by 25%. However, it does not take into account the larger atmospheric absorption at far-North latitudes.

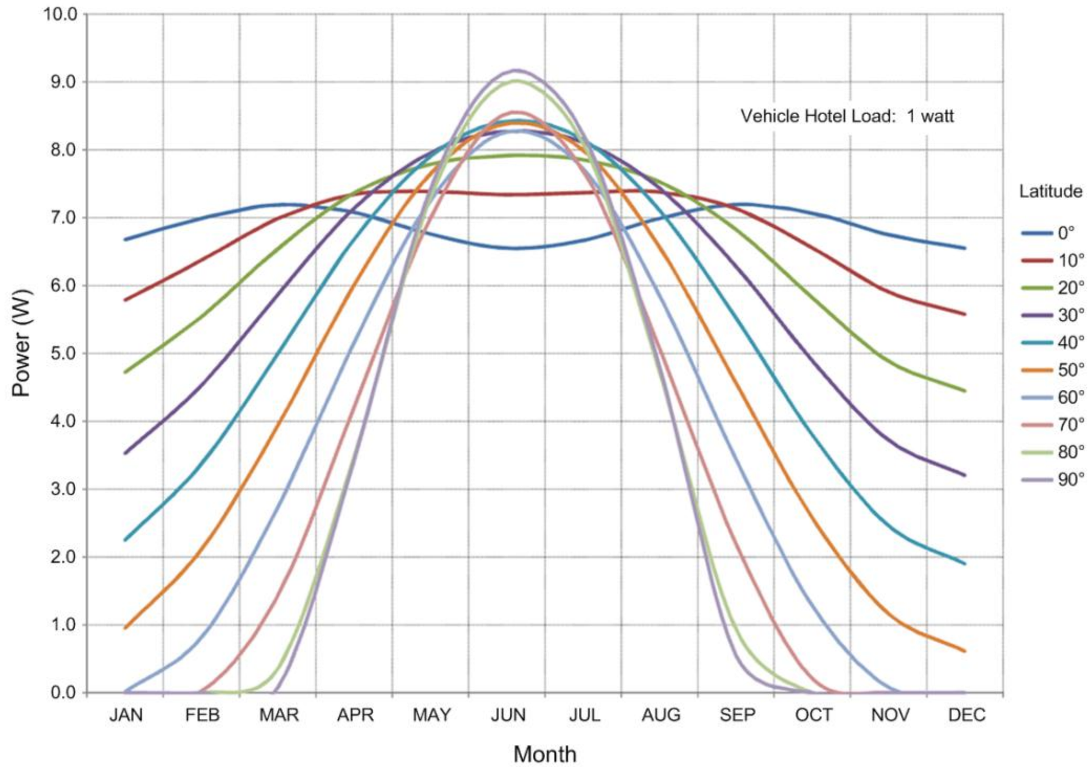


Figure 4-3: Solar Power vs. Time and Location

4.4.4 C&C Drybox Power Draw

The C&C Drybox runs off of the internal battery pack with a theoretical maximum capacity of 650 Wh. The amount of power draw from by the C&C box depends upon the electrical activity, as shown in Table 4-12 below.

Operation	Average Power
Idle	700 mW
Iridium On	700 mW
XBee On	250 mW
RS-232 to Float Payload	50 mW
RS-232 to Float Payload	50 mW

Table 4-12: C&C Drybox Current Draw

4.4.5 Battery System Power

The Wave Glider fully-charged Lithium-Ion battery pack has the following characteristics:

Battery Max Voltage	16.6 Volts
Maximum Continuous Current	20 Amps
Energy Capacity (fully charged) [theoretical maximum]	665 Watt-hours

Table 4-13: Battery Pack Characteristics

The above numbers assume that all seven batteries are installed in the C&C Drybox. However, note that the Maximum Continuous Current is not available on any external connections.

4.4.6 Battery Charge Level

The figure below shows how the battery voltage indicates the charge level. Note the characteristic rapid drop-off near full discharge.

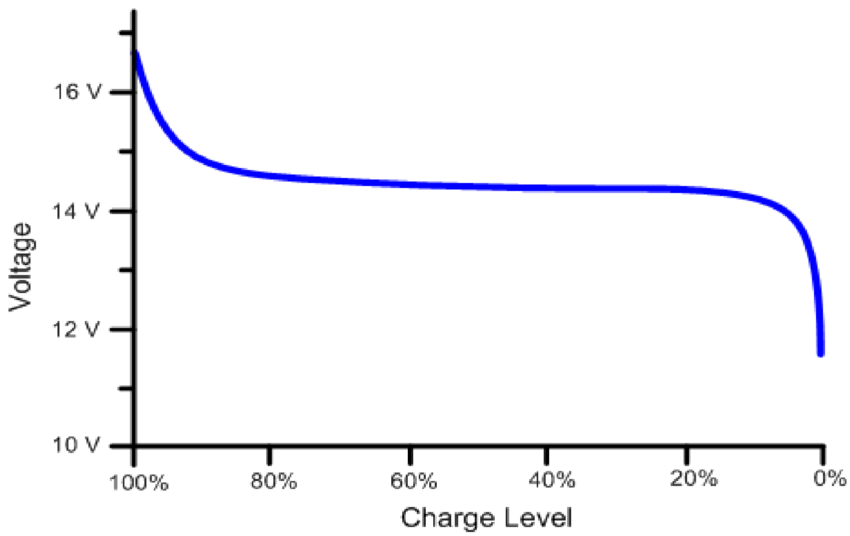


Figure 4-4: Battery Charge Level vs. Voltage

This Page Intentionally Left Blank

5. Vehicle Communications and Control

5.1 Communications

There are multiple methods of communicating with the Vehicle. The types and characteristics of this communication are generally described below. Details are provided in subsequent chapters.

5.1.1 Communications Management

All primary communications to and from the Vehicle are managed by the Command and Control ("C&C") electronics contained in the center Float drybox. Likewise, central control of the Vehicle systems is handled exclusively by the C&C electronics.

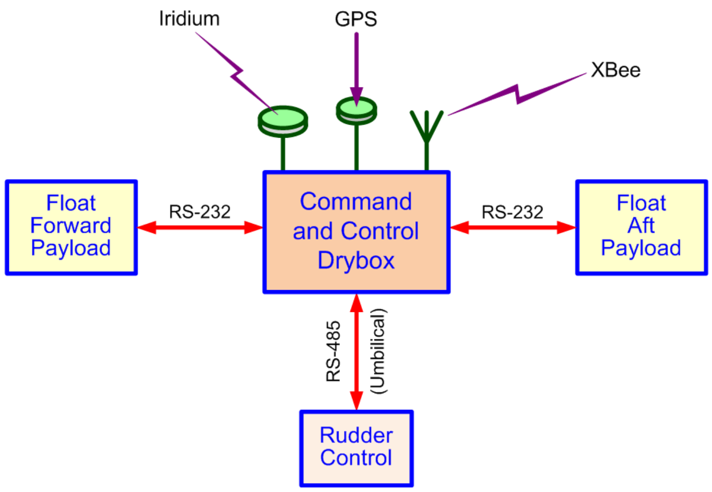


Figure 5-1: Wave Glider Command and Control

5.1.2 GPS

The GPS constellation location signals are essential to Vehicle operation. Most commands for Wave Glider positioning are based on Latitude and Longitude values. Telemetry and other status responses from the Vehicle contain the Lat/Long location based upon GPS data.

Figure 5-2: GPS Satellite Constellation

GPS fixes are typically obtained every 5 seconds. However, there can be short periods (usually less than a minute) when no signal is received.

5.1.3 Iridium-Internet Link

As shown in Figure 5-3, the Wave Glider is typically remotely linked to the shore-based control interface through an Internet connection to the Iridium satellite system. This is the standard means of operational control and monitoring of the Vehicle.

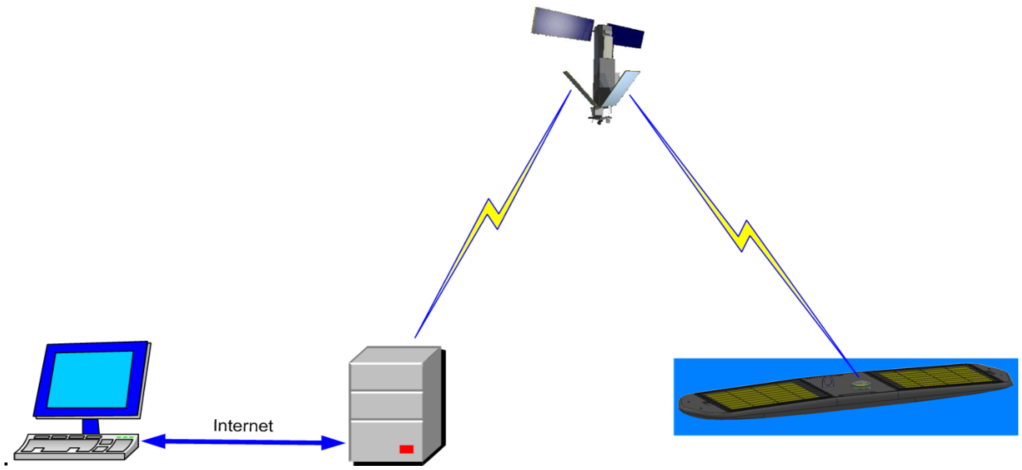


Figure 5-3: Shore/Ship Iridium Communications

The Iridium communications are via variable-length data packets sent in "short burst mode".

Communications can also be accomplished via a "Relay" connection that bypasses the Iridium system. Relay communications is similar to Iridium communications except that data is sent directly to the Vehicle, rather than through the Iridium constellation.

Refer to section 5.5 "WGMS Components" for more detailed information.

5.1.4 Mobile-to-Mobile Link

Mobile-to-Mobile communications provides an alternative to the Web-based Iridium link. In this case, the controlling computer is directly linked to an Iridium modem, as shown in Figure 5-4.

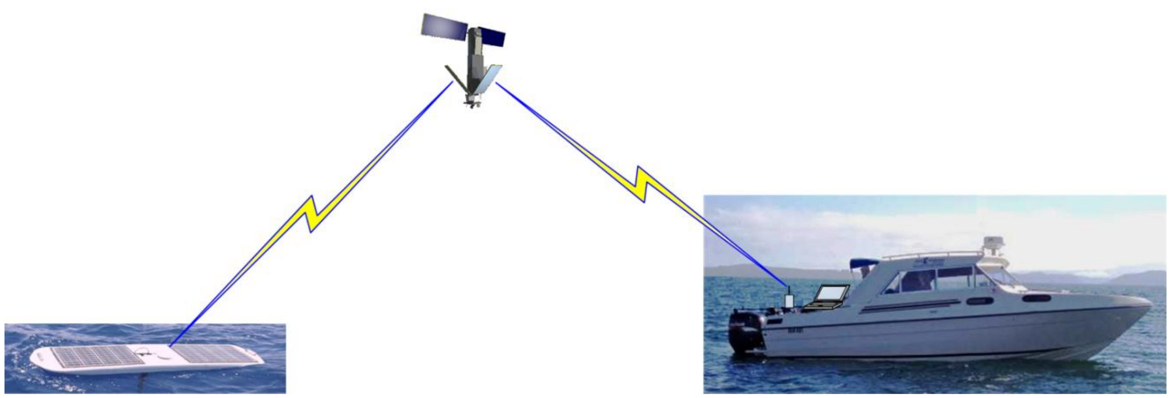


Figure 5-4: Mobile-to-Mobile Communications

The Mobile-to-Mobile link allows sophisticated control of one or more Vehicles from a ship or other remote site. The direct link also provides a more secure method of communications.

Refer to section 5.6.2 "Mobile-to-Mobile Deployment" for more detailed information.

5.1.5 XBee Radio Link

The Wave Glider has an XBee short-range radio. Though intended primarily for communications between a ship and the Vehicle, it can be a convenient link for on-shore development and testing of Wave Glider systems.

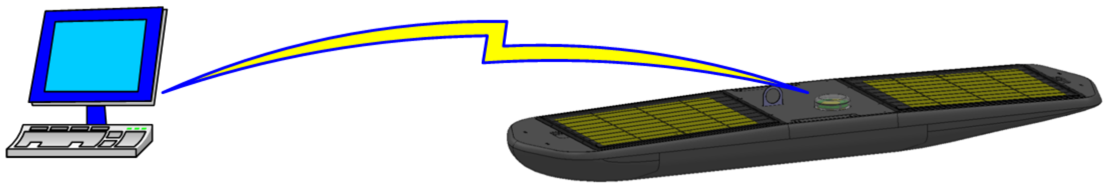


Figure 5-5: On-Shore XBee Communications

The XBee communications is through a computer screen text-based system. The user can select from a menu of options, and the computer prompts for additional data if necessary. Responses from the Vehicle are displayed on-screen in text format.

Refer to section 5.6.4 "Payload Development and Testing Using XBee" for more detailed information.

5.1.6 Payload Communications

There is an optional RS-232 communications link between each payload and the C&C Drybox computer. This link can be used for indirectly sending or receiving data using the Iridium system.

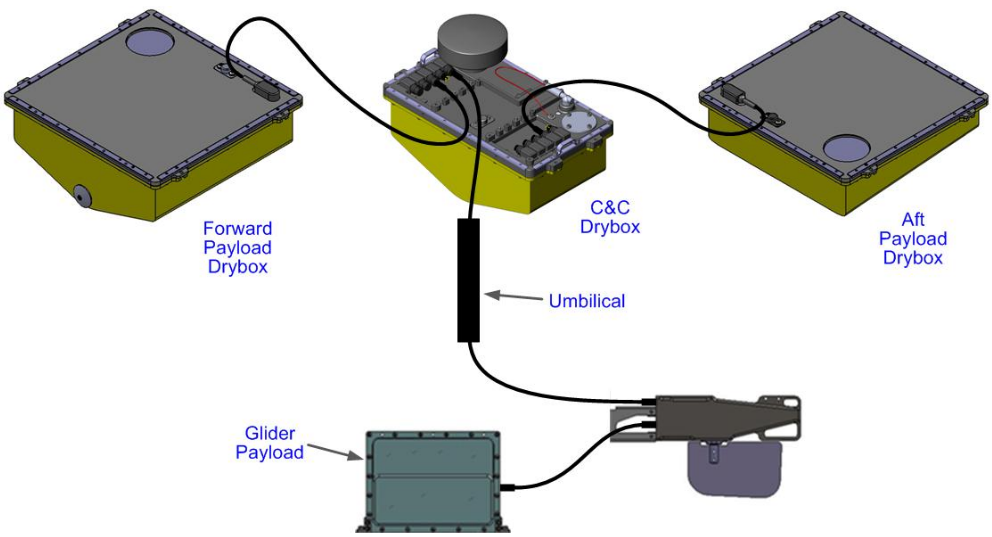


Figure 5-6: Payload Communications Connections

In addition, there is a two-wire uncommitted "Payload Bus" that connects to each payload, but does <u>not</u> connect to the C&C computer. This can be used to transfer data between payloads using whatever 2-wire method is preferred by the payload integrator.

A Glider-mounted payload can be connected through the Umbilical to the 2-wire Payload Bus, but cannot communicate directly with the C&C computer. Any Glider payload datacomm to the C&C computer has to be routed via one of the Float payload's RS-232 links.

Refer to section 6.2 "Payload Communications" for more detailed information.

5.2 C&C Functions

As the central point of Vehicle communications, the C&C computer has several functions, described below.

5.2.1 Telemetry and Polling

Via the Iridium (or a Relay), the C&C automatically sends telemetry data at regular intervals. Telemetry data indicates primarily the position, but also the general status of the Vehicle and if there are any alarms. There are different telemetry data types, and each type can be sent out on a different schedule.

In addition, the C&C sends periodic polling messages to the payloads. These messages may convey information to the payloads, but also give the opportunity for the payloads to send messages to the C&C.

An important point to understand is that all outside communications with the C&C are initiated on the C&C, and depend upon its telemetry and polling.

5.2.2 Command Processing

The C&C receives commands for controlling the Vehicle operation, or for reporting the status of Vehicle systems.

Wave Glider commands typically used for development and testing are:

- Make some change to the Vehicle (e.g., move rudder, turn a light on).
- Set system parameters (e.g., telemetry rates).
- Request Vehicle systems status (e.g., battery level).
- Communicate with a payload.

Additional commands generally used during on-water operations are:

- Define a course (e.g., set/delete course Waypoints).
- Follow a course (e.g., hold position, follow a set of Waypoints).

The Vehicle response to commands depends upon the communications method, as described below.

5.2.3 Payload Message Handling

The C&C may receive a message from a payload. The C&C action will depend upon where the message is addressed to.

- If addressed to the C&C payload message handler, the C&C will take an appropriate action.
- If addressed to the C&C command processor, the C&C will process the command.
- If addressed to another payload, the C&C will route the message to that payload.
- If addressed to the Iridium system, the C&C will send it out in Iridium format.
- If addressed to the XBee system, it will be routed to XBee.

5.2.4 Command/Message Responses

When an Iridium command or payload message is received by the C&C, it will respond in basically two different ways:

- Acknowledge (ACK) responses let the requesting system know that the command/message was received. The ACK response may also contain requested status information.
- Not-Acknowledge (NAK) responses indicate that the data was received but is in error, and could not be processed. The NAK response also indicates the general nature of the error.

XBee menu commands do not respond with an ACK/NAK. However, status commands will result in a response, and error messages will be returned if there is a problem.

5.3 Data Types

There are essentially three types of communication data types: Telemetry, Commands and Responses.

5.3.1 Telemetry

The Vehicle automatically sends telemetry data packets at regular intervals. Telemetry Data indicates primarily the position, but also the general status of the Vehicle and if there are any alarms. There are different telemetry data types, and each type may be sent out on a different schedule.

5.3.2 Commands and Messages

When the shore-based system receives an Iridium packet (usually telemetry), it can then send a command to the Vehicle. Likewise, if a payload on the Vehicle receives a message, it can send a command to the C&C. The C&C then processes the command if possible.

Commands generally fall into the following categories:

- Make some change to the Vehicle (e.g., move rudder, turn a light on).
- Define a course (e.g., set course Waypoints).
- Follow a course (e.g., hold position, follow a set of Waypoints).
- Set system parameters (e.g., telemetry rates).
- Request Vehicle systems status (e.g., battery level).
- Route a message to a payload or the Iridium.

5.3.3 Responses

When a command or payload message is sent to the Vehicle, the C&C will respond in basically two different ways:

- Acknowledge (ACK) responses let the requesting system know that the command was received. Depending upon the type of command/message, the ACK response may also contain requested status information.
- Not-Acknowledge (NAK) responses indicate that the data was received but is in error, and could not be processed. The NAK response also indicates the general nature of the error.

When an XBee command is sent to the Vehicle, there will either be a text response, or there will be no response at all, depending upon the command type.

5.3.4 Communication Packets

All communications to and from the Vehicle are done with "packets" of information containing a fairly small (approx. 200 to 500) number of bytes. Note that Iridium and Relay packets themselves contain packets that contain the message data and an error check.

Communications to and within the payloads are also done with data packets, though the message types and formats are different than for Iridium/Relay.

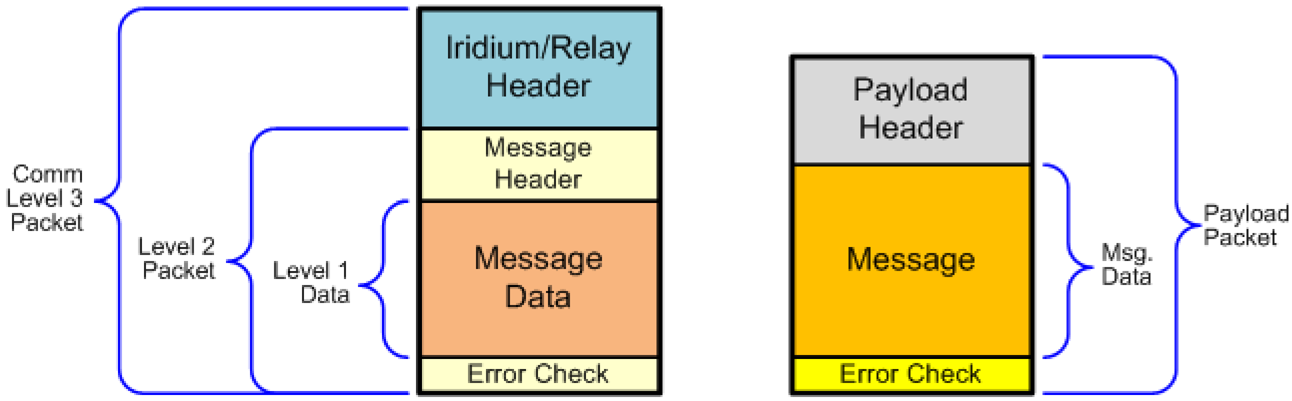


Figure 5-7: Data Packet Formats

As shown in the figure, the Iridium and Relay packets have a similar format. However, the header data – though similar – is not identical. The Iridium and Relay data packets are described in more detail in Chapter 6.

The Message in a payload packet supports many different types of data, including data that can be put into an Iridium/Relay packet. Payload message data is described in detail in Chapter 10.

As indicated above, packet communications all carry two CRC-16 bytes that can be checked by the receiving party. All Iridium communications to the Shore will have the error code checked. Payload message packets may or may not be checked, at the option of the message recipient.

5.3.5 Addresses

Addresses are values used to route messages between the C&C, payloads and other devices (e.g., sensors). Addresses are composed of a two-byte value. The MSB is the “Board ID” value which is usually put in Non-Volatile Memory on the board. The LSB is the “Task ID” which is usually assigned by the firmware on each board.

Addresses are described in more detail in section 10.4.

5.3.6 XBee

XBee communications use character-string commands and responses, rather than data packets. There is no error checking.

Commands sent via XBee to the Vehicle are interpreted by the C&C computer, which performs any requested action, then sends the appropriate response back to the XBee message originator. Software on the User's computer displays the responses as on-screen text.

XBee communications are described in more detail in Chapter 11.

5.4 Wave Glider Management System

The Wave Glider Management System (WGMS) is the Shore-based system used to control all aspects of Wave Glider operation. It is used to define the Graphical User Interface, handle inputs from the UI, and deal with Iridium and Relay communications both to and from the Vehicle. It also manages the storage and retrieval of data in the Database.

The WGMS interfaces through the Internet (Worldwide Web) to both the User and the Iridium. It uses the Database both to configure the User Interface, to store Vehicle data, and to recover stored data for presentation on the UI. Web and Database communications are done using XML entities. The WGMS XML Entities are described in detail in Appendix A.

5.4.1 WGMS Overall Structure

Figure 5-8 shows the overall operational structure of the WGMS.

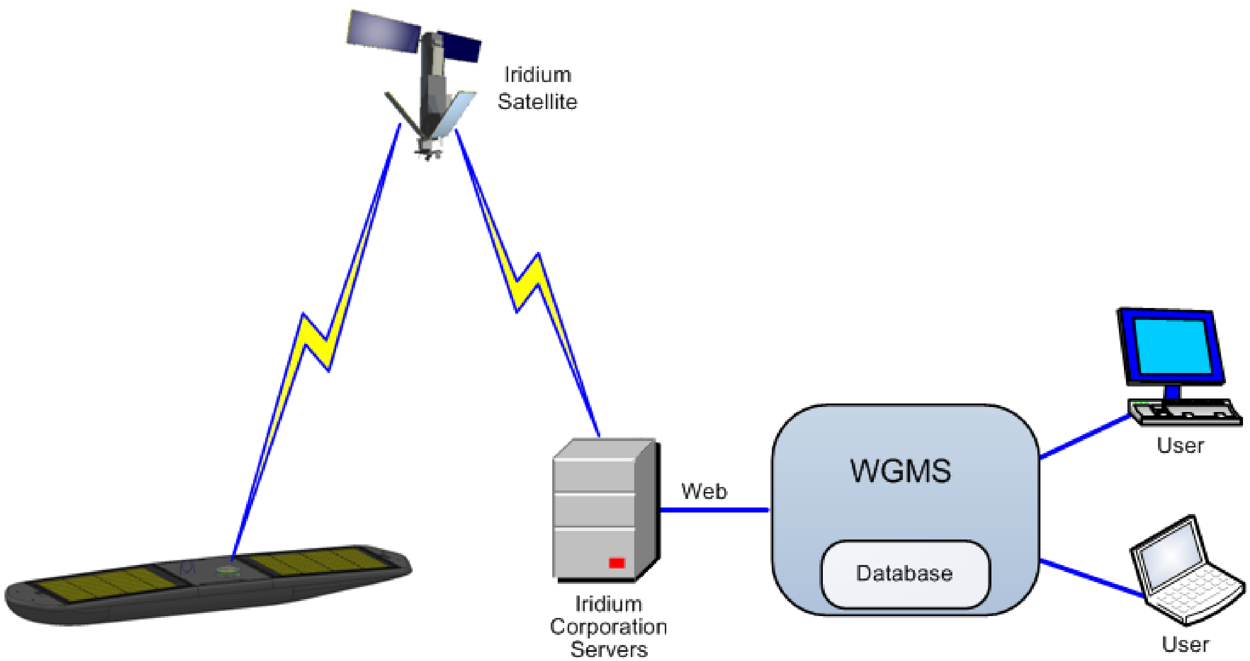


Figure 5-8: Basic Wave Glider Management System

The figure above shows one of several possible WGMS configurations. Other configurations are described in more detail below in section 5.6 "Common WGMS Scenarios".

5.4.2 Command Communications

For most applications, commands are sent from a browser UI or a customer-provided application to the WGMS Kernel. The Kernel performs authentication, authorization and validation checks on the request. Once approved, it passes the command in an internal format to a Packet Reflector. The Packet Reflector forwards the command to the Vehicle C&C via a specified "Relay" communications channel. The Relay manages the details of the underlying communications technology (e.g., Iridium, XBee, Custom). Depending on the addressing in the command, the C&C either executes the command locally or forwards it to a payload for processing.

5.4.3 Vehicle Responses

Vehicle responses such as telemetry are returned from the Wave Glider through essentially an inverse of the Command communications mechanism. Responses sent to the appropriate Relay (when on mission, typically the Iridium Server) which forwards it to the appropriate Packet Reflector. The Packet Reflector performs validations and some protocol processing and then forwards the data to one or more WGMS Kernels, based on configuration. The WGMS Kernel stores the data in its database, where it can be accessed by the user for evaluation.

5.5 WGMS Components

Figure 5-9 illustrates a full WGMS operational configuration, indicating all communications, routing and processing options. Specific deployments, such as payload development or mobile-to-mobile solutions, typically will use a subset of the modules represented by the illustration.

In the figure, circles represent software modules. Blue circles indicate LRI-supplied components which may be customer or user configurable. Crosshatched circles represent user-modifiable or user-generated components. As shown in the figure, the WGMS components communicate with each other via TCP/IP and need not all be resident on the same computer.

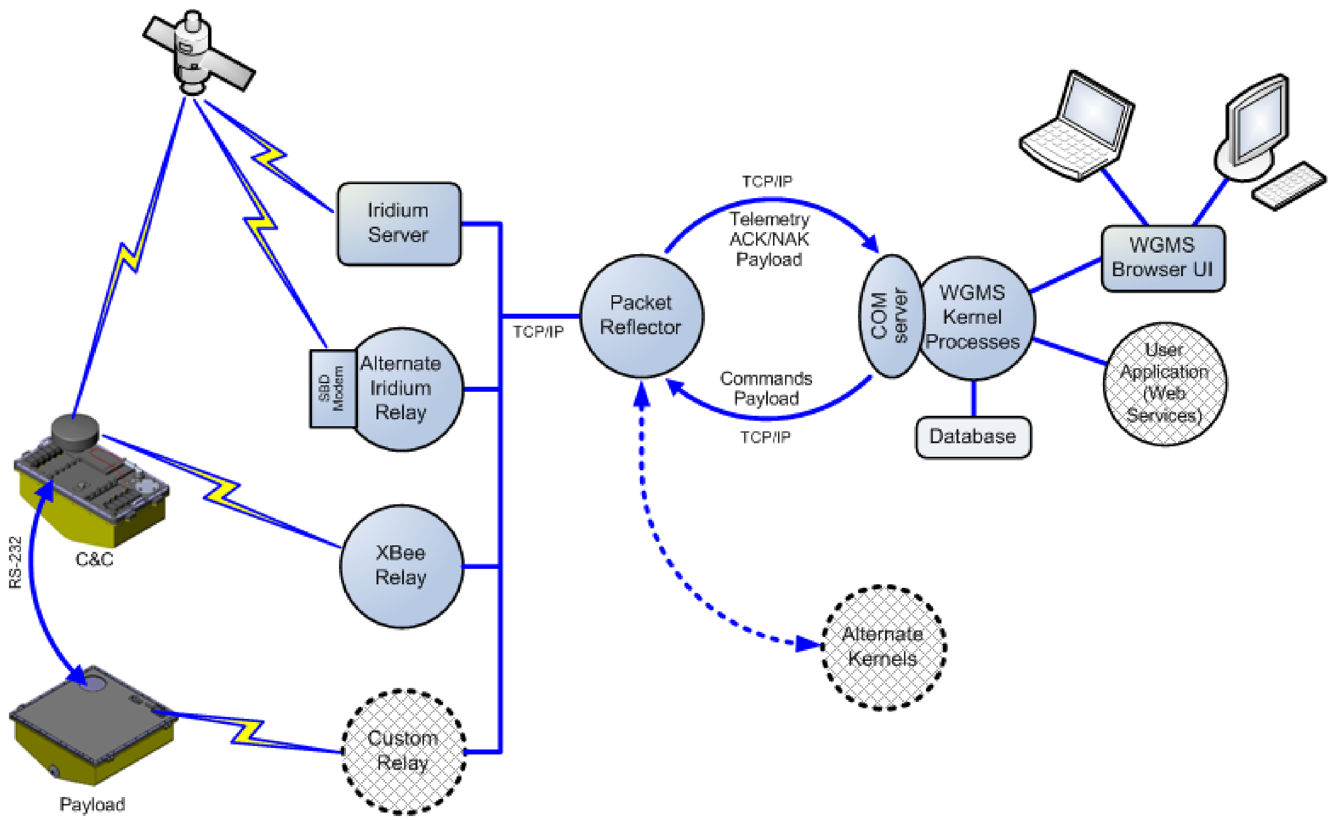


Figure 5-9: Basic WGMS Components

The Components of the WGMS (going generally from right to left in Figure 5-9 above) are as follows:

5.5.1 User Interface

The User Interface is the Web browser-based graphical interface to the Wave Glider. The GUI has access to selected database values for each Vehicle and displays the following elements:

- An annotated Google Map showing the position of Waypoints and Vehicles.
- Summaries of the telemetry and command data from any selected Vehicle.
- Buttons, icons and lists for sending commands to the Vehicle.
- Selected data associated with users and Vehicles

The appearance and use of the User Interface are described in the Wave Glider *User Manual*.

The User Interface is used to both control and monitor Vehicle operations. It communicates with the Internet using HTTP(S) and translates to and from XML for interface with Web Services.

5.5.2 User Application

As an alternative to the conventional user interface, the WGMS Kernel can be accessed by a non-LRI application using Web Services (SOAP 1.1 or SOAP 1.2) calls over HTTPS. Authentication and authorization are the same as for browser-based access. No additional configuration is required for Web Services access.

Web Services is essentially an XML processing system. It serves as the central communication and data management center of the WGMS. It manages:

- The forms, views and components of the Graphical User Interface.
- Communications to and from the Iridium.
- Object-Oriented Entities for transfer to and from the Database.

5.5.3 WGMS Kernel

The WGMS Kernel is essentially the "brains" of the WGMS system. It controls the GUI, Web Services, communications and database access. The Kernel can also use email for sending alarms to users.

Primary services include filters for access control, data consistency and concurrency management. Communications with application programs is through Web Services technologies. Browser communications rely on standard HTML and JavaScript.

5.5.4 Database

WGMS relies on Microsoft SQL Server, a relational database that contains both data and metadata.

Data stored in the database contains all the telemetry, alarm and other response received from the Vehicle, as well as the Shore-based commands sent to the Vehicle. This allows retrieval and presentation of all data from both active and previous missions.

Metadata contains XML entities used to configure and operate the Web Services, the User Interface and Vehicle communications.

5.5.5 Com Server

The COM Server interfaces between the Packet Reflector and the kernel WGMS server application. It communicates with the kernel via Web Services.

The Com Server restructures Iridium data into XML for processing by Web Services. It also queries the system through Web Services to determine if any commands need to be sent to the Vehicle. Commands in XML are restructured into the Iridium format for sending.

5.5.6 Packet Reflector

The Packet Reflector is basically a bidirectional router for the Iridium data packets. Applications "subscribe" to communicate with the Packet Reflector. Normally, it routes packets between the the Com Server and Relays, but it can also route messages to other applications.

The Packet Reflector normally communicates via TCP/IP, but defaults to email if the Internet service connection is down.

5.5.7 Alternate Kernels

A Packet Reflector can direct messages to, or receive messages from other kernels. These can be other copies of WGMS, or they can be custom applications.

5.5.8 Iridium Server

For most deployed Wave Gliders, the primary mode of communications is through the use of Short Burst Data (SBD) packets sent via the Iridium satellite constellation. Actual satellite communications to the Vehicle is done via a group of servers maintained by Iridium Communications, Inc. The Packet Reflector communicates to these servers through the through the Web, using TCP/IP. If necessary, data transfer can be done via Email, but at a lower data rate.

5.5.9 Communications Relays

Communication relays are generally three types: Alternate Iridium, XBee and Custom). Their usage is described in section 5.6 below.

It is sometimes beneficial to communicate with the Vehicle through means other than the standard Iridium Server. Considerations can include bandwidth, cost, latency, privacy, security and distance. For example, in a payload development scenario, the Wave Glider’s Iridium satellite antenna might be unable to see the sky, necessitating the use of XBee.

Relays mask the variations between the underlying communications technologies by converting to/from TCP/IP for handling by the Packet Reflector. Specific duties may include fragmenting and reassembling packets, managing queues and driving the physical interface.

Custom Communications Relays may be developed by customers and third parties. They need not even reside on the same computer as the Packet Reflector or WGMS kernel.

5.6 Common WGMS Scenarios

This section describes the four most common deployment scenarios. The descriptions below show the types of communications that may be encountered by payload developers. Detailed descriptions of how to configure different scenarios are provided in the LRI *System Administration* document.

Your specific usage may require combining more than one scenario. For example, you might run a private shared copy of WGMS that must support both payload developers as well as Wave Gliders deployed on a mission. LRI Support would be pleased to advise you on configuration options specific to your needs.

5.6.1 LRI-Hosted Shared WGMS

A popular deployment option is letting LRI manage the hosting of WGMS for your Organization. The WGMS software is deployed on redundant, high-performance servers in a secure, high availability data center. In general, communications are via the commercial Iridium system. LRI's Operations staff configures the system for you and administers your system and Wave Gliders on a 24x7 basis. Figure 5-10 and Figure 5-11 illustrate this option.

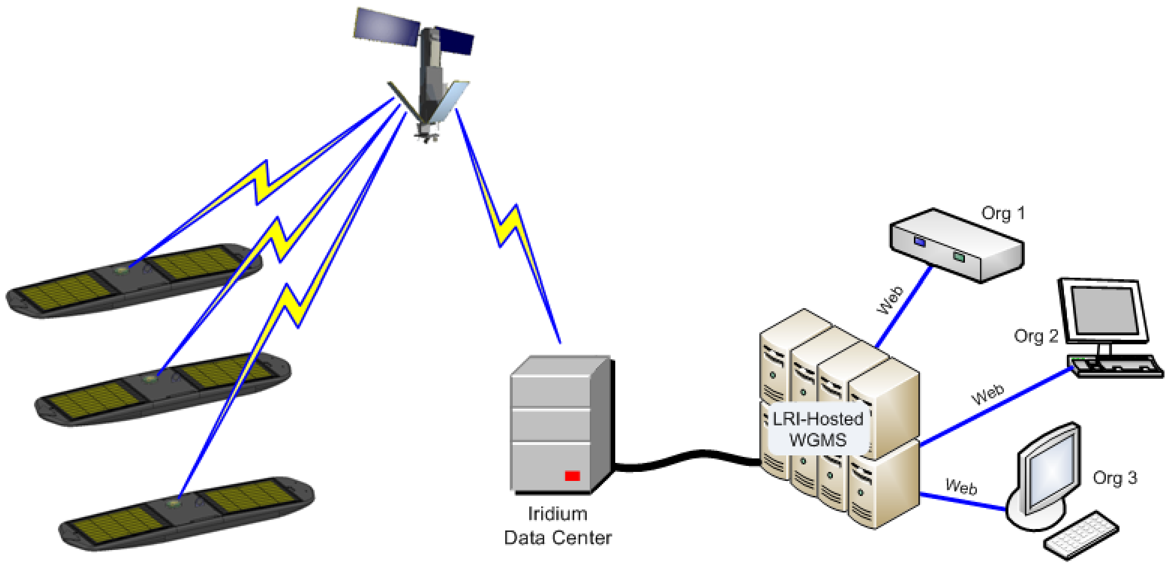


Figure 5-10: LRI-Hosted Shared WGMS

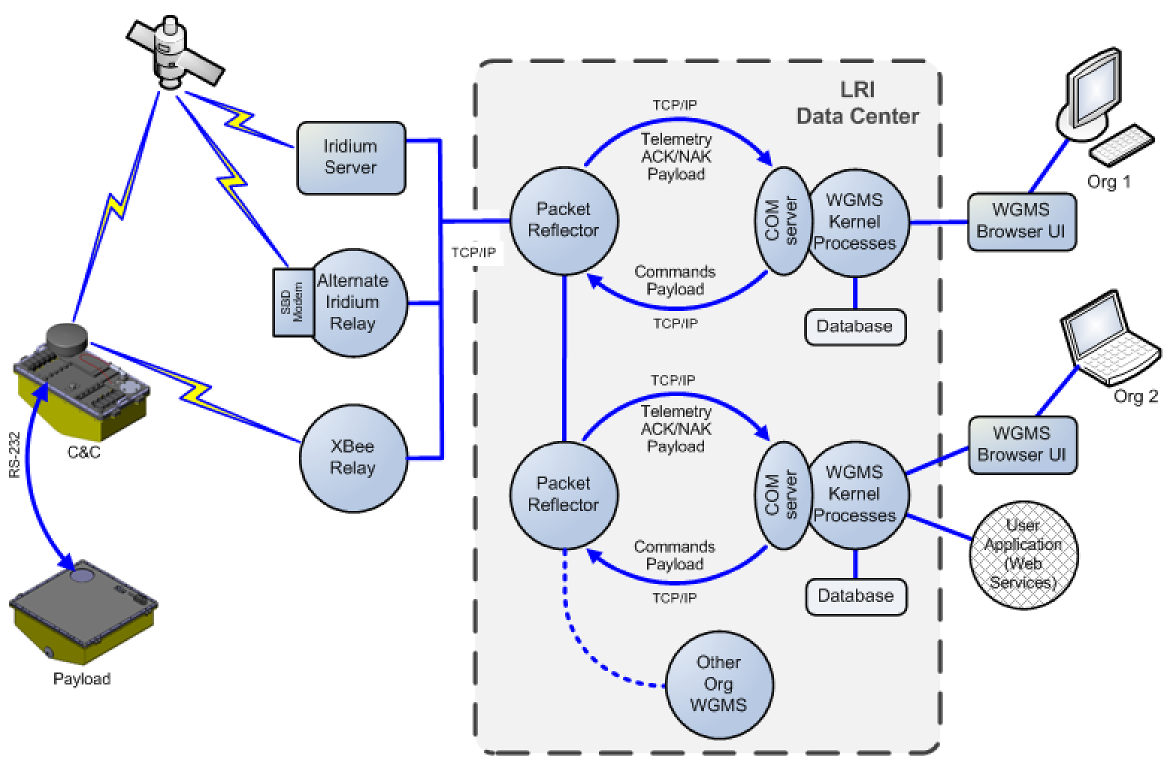


Figure 5-11: Multiple Orgs Using LRI-Hosted WGMS

5.6.2 Mobile-to-Mobile Deployment

The Mobile-to-Mobile ("M2M") configuration supports Wave Glider deployments when Internet access is inconvenient or expensive, such as on-board a ship. The Iridium satellite network is still used, but the communications are directly between the Iridium modem on the Wave Glider and a paired Iridium modem on a local computer (the "M2M Host"), such as a laptop.

For M2M, the entire WGMS stack is present on the M2M host. Communications between WGMS elements are still via TCP/IP, using an internal networking capability. Although elements of the mobile-to-mobile stack can connect over the Internet to other Packet Reflectors and other Internet-based services, it also can run completely independent of the Internet. One consequence is that Internet-based services, such as email and Google Maps, may be unavailable.

Iridium communications in the M2M scenario use a protocol called "ISU-ISU", (ISU = "Iridium Subscriber Unit"). The ISU-ISU protocol is very similar to the Direct IP protocol except that the sent messages are queued in servers in the Iridium Corporation data center until the second modem requests them. This is illustrated in Figure 5-12.

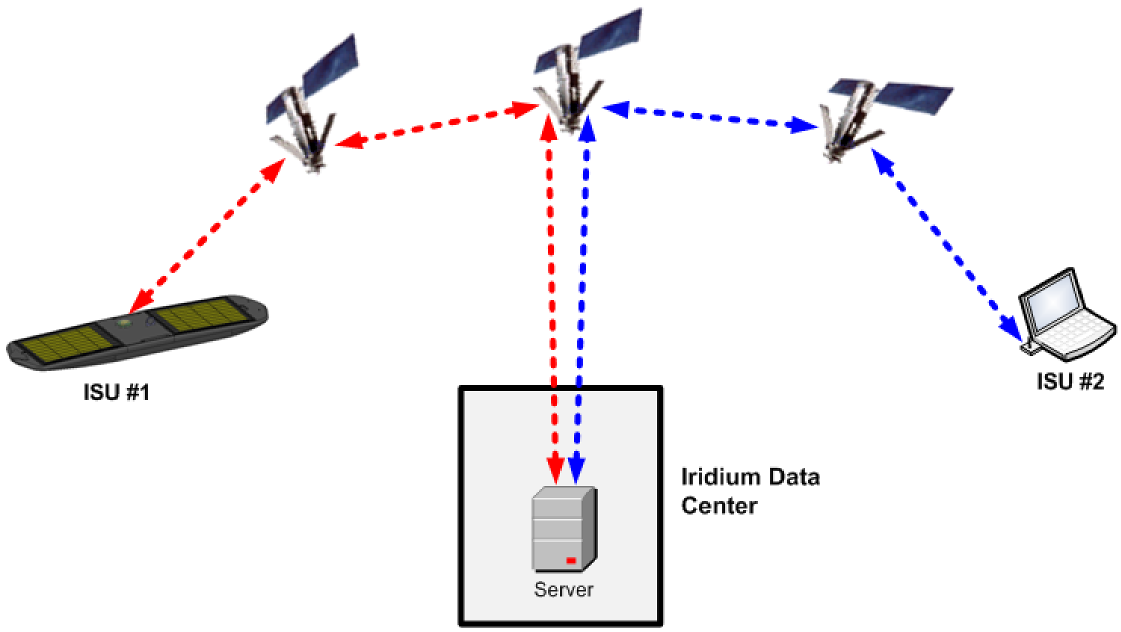


Figure 5-12: Iridium Communications Corp ISU-ISU

In Mobile-to-Mobile, the paired Iridium modem is connected to the M2M host computer via a serial interface. Messages from the Wave Glider go up to the Iridium satellite, back down to the local Iridium modem, through the serial interface to the Alternate Iridium Relay. This relay reformats the messages before passing them on to the M2M host's Packet Reflector, which then forwards the message to the WGMS Kernel and local database. A web server runs on the M2M Host and is accessed by a browser on the M2M host. As with other versions of WGMS, application programs can interact with the WGMS Kernel through Web Services interfaces even though they run on the same computer.

The M2M communications architecture is shown in Figure 5-13.

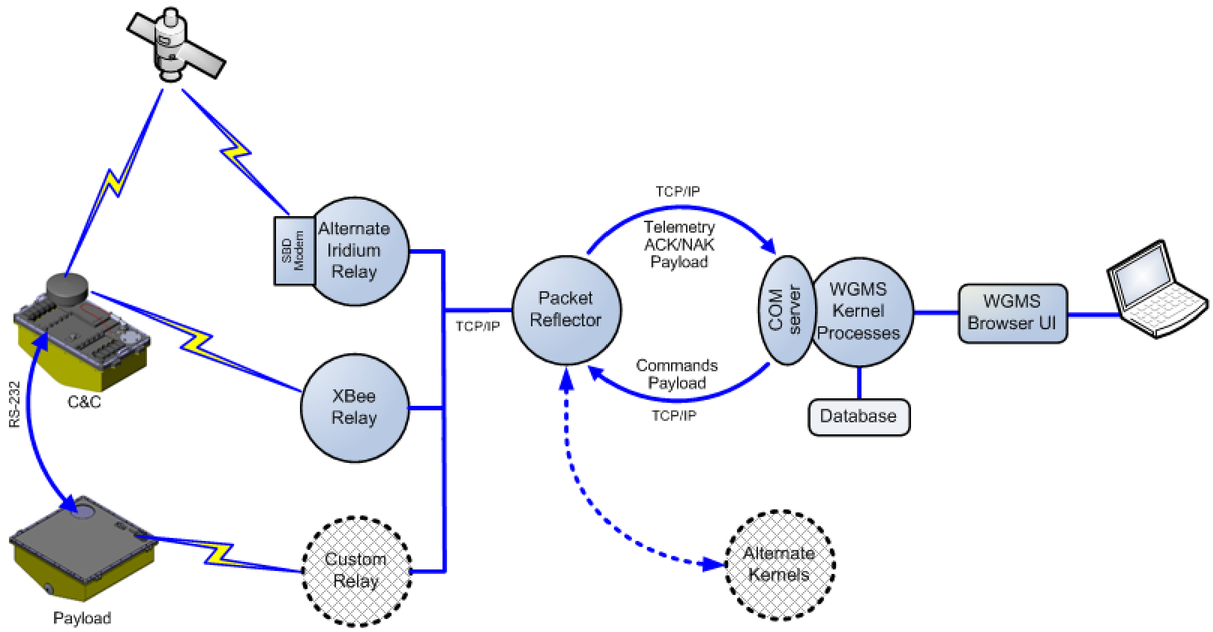


Figure 5-13: Mobile-to-Mobile Configuration

5.6.3 Customer-Hosted (Self-Hosted) Shared WGMS

An Organization can license WGMS for hosting on its own servers inside its own network. Advantages of this option include direct oversight of physical security, better control over performance parameters, and the ability to tightly integrate WGMS services with other aspects of the Organization.

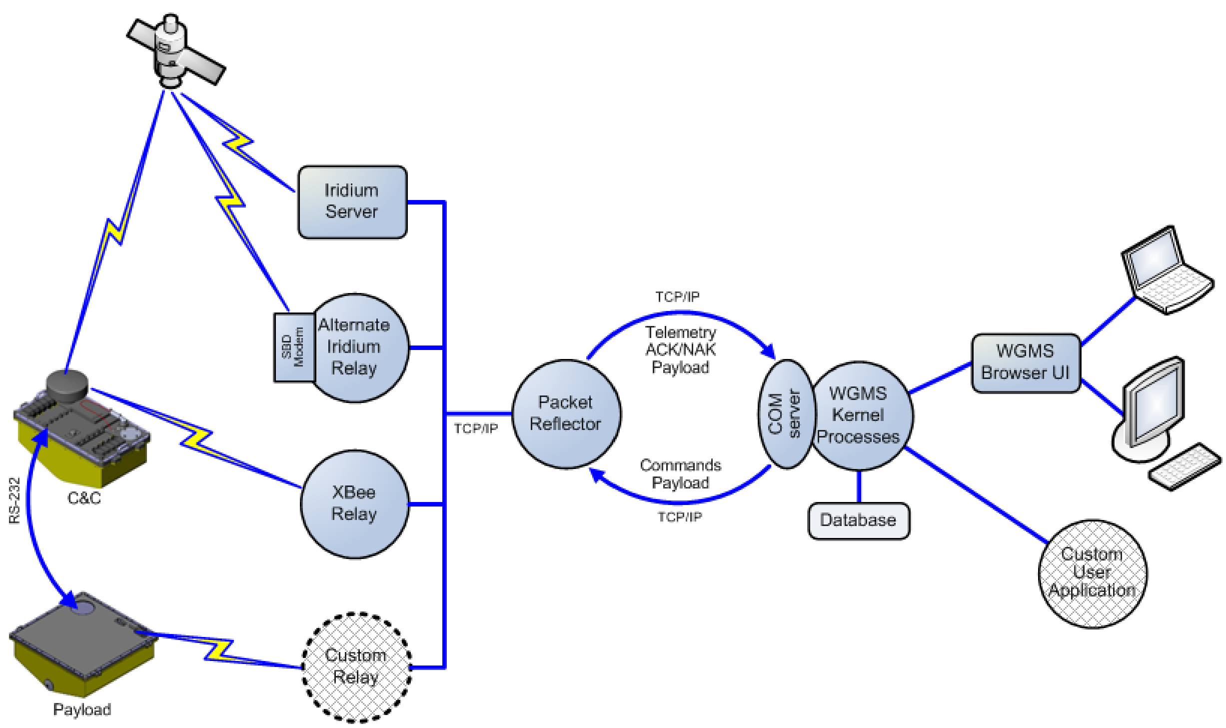


Figure 5-14: Self-Hosted WGMS Configuration

5.6.4 Payload Development and Testing Using XBee

A common workflow is to develop and test a payload in a development lab, then launch the integrated vehicle on its mission. The recommended way to do this is to combine usage of an XBee modem and Iridium services. This use case usually is combined with one of the previous three use cases – LRI hosted, self-hosted or mobile-to-mobile.

For a Wave Glider on an offshore mission, the only practical means of communications is via satellite, typically the Iridium Communications Inc. constellation. This works well with the unobstructed view of the sky from the surface of the sea. However, during payload development, using satellite communications is impractical since the Vehicle is typically indoors. In addition, satellite bandwidth can be expensive and slow compared to other technologies.

To meet this need, the Wave Glider C&C is equipped with an XBee modem. A nearby computer can be equipped with an XBee modem which communicates through a serial port with the LRI-provided XBee Relay.

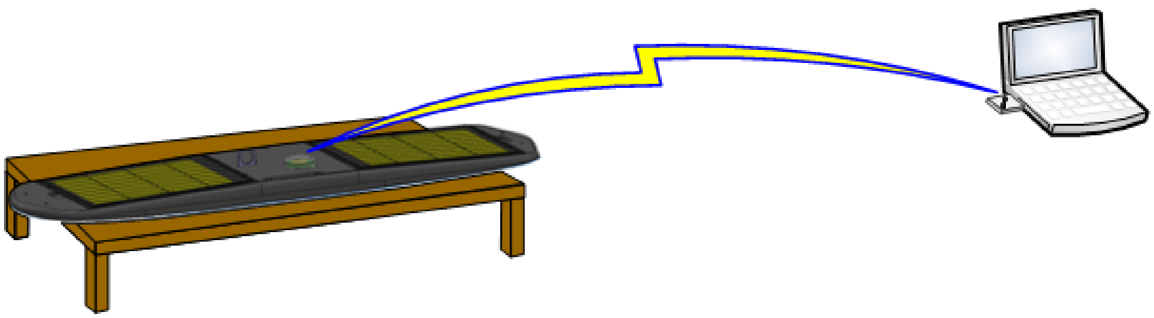


Figure 5-15: XBee for Payload Testing

The XBee communications flow is illustrated below.

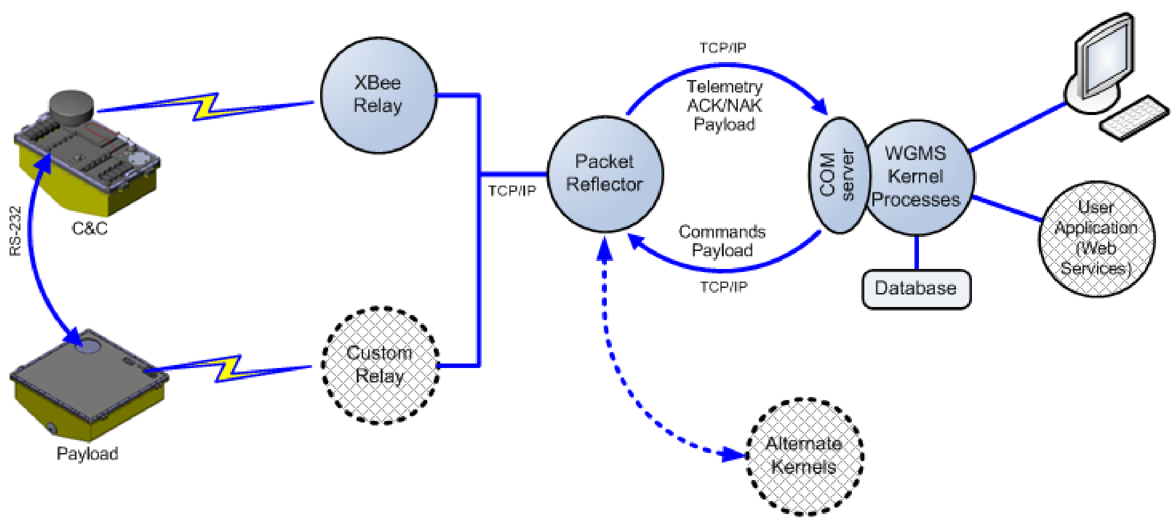


Figure 5-16: XBee Communications Flow

A very common scenario is the pre-launch checkout of a Wave Glider. Many of these checks can be performed using the XBee relay, though LRI recommends that the Iridium communications also be exercised prior to launch.

5.7 Alternate WGMS Configurations

The WGMS architecture supports a number of other less common usage scenarios. Two are mentioned briefly below. Please contact LRI Support for advice on configuring your system if you require one of these, or any other configuration not mentioned.

5.7.1 Customized Relay

It is possible for an Organization to implement their own Relay to supplement or replace an LRI supplied Relay. Two circumstances where this might be desirable are:

1) You need to support a communications protocol other than XBee or Iridium, such as 802.11 or BGAN. In this case, you will need to implement the hardware and software stack on the Wave Glider side via the payload electronics in one of the Float dryboxes. You also will need to supply the shore-side hardware and software stack and implement a custom Relay to interface between that stack and the Packet Reflector.
2) You would like to monitor traffic between the Wave Glider and the Packet Reflector. This would allow you, for example, to inspect payload-specific data packets in near real-time rather than waiting for them to be delivered through WGMS and Web Services.

5.7.2 Multiple Copies of WGMS

A Packet Reflector can forward a copy of a packet inbound from a Wave Glider to other Packet Reflectors. This allows multiple instances of WGMS to see the same Wave Glider telemetry. This can be useful in large-scale testing environments.

This Page Intentionally Left Blank

6. Communications Methods

Iridium, Relay and Payload communications are via packets, with headers and data sub-packets. XBee communications uses ASCII character streams.

6.1 Iridium/Relay Communications

6.1.1 Short-Burst Data

The particular Iridium protocol used is the "Short-Burst-Data" (SBD), which allows economical transmission of relatively small data packets.

Vehicle communication can also be done directly with a “Relay” that bypasses the standard Iridium system. Relay communications use packets whose headers are similar to the Iridium packets.

All of these communications are handled by the C&C and the WGMS. However, since payloads may need to communicate via the Iridium or Relay, the formats of these data packets should be understood.

6.1.2 Packet Levels

Iridium/Relay packets are described as “Level 3”, “Level 2” and “Level 1”.

The Level 3 packet is used only for communications to and from the Vehicle, using either the Iridium system or an alternate “Relay”, as described in sections 1.9.1 and 1.9.2. A Level 3 packet contains a header, followed by a Level 2 data packet.

The Level 2 packet itself contains its own header, message data, and an error check. The Level 1 packet can be raw data without any header. This hierarchy is shown in Figure 6-1.

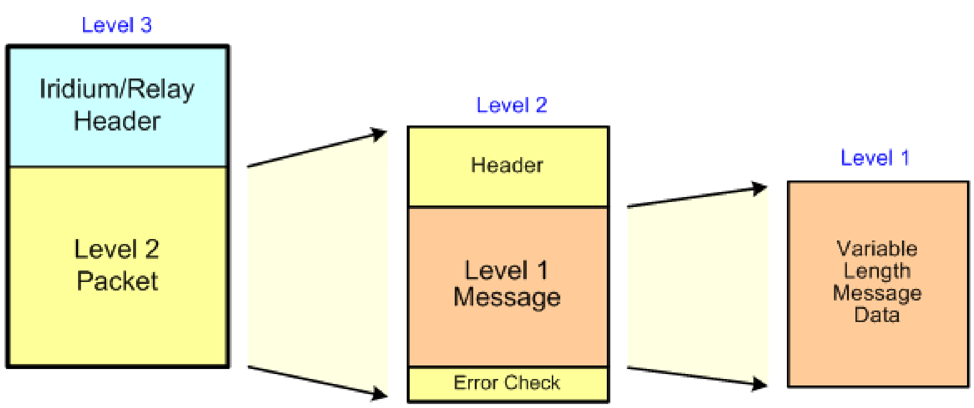


Figure 6-1: Level 3, Level 2 and Level 1 Data Packets

Level 2 packets within the Iridium/Relay packets comprise the types: Telemetry, Commands, Responses and Alarms. The length and structure of the different packets will vary depending upon the data type, and the intended recipient. These packets are described briefly in sections 6.1.3 and 6.1.4 below, and in more detail in Chapters 7, 8, 9 and 10.

The details of the Iridium and Relay headers are managed by the C&C and the WGMS, so they need not be understood for payload development. For reference, however, the Iridium and Relay header formats are described in Appendix B.

6.1.3 Shore-to-Ship Commands

A Shore-to-Ship packet carries Commands to the Vehicle. It has 30 bytes of header content and can carry up to 135 bytes of Command data.

Iridium/Relay Shore-to-Ship Header	30 bytes
Command Data Packet	≤ 135 bytes

Figure 6-2: Shore-to-Ship Packet Format

6.1.4 Ship-to-Shore Telemetry/Responses

Ship-to-Shore Iridium packets are ACK/NAK responses, status data, telemetry or alarms. These packets have a 51 byte header, followed by up to 205 bytes of response data.

Iridium/Relay Ship-to-Shore Header	51 bytes
Telemetry, Response, Alarm Data Packet	≤ 205 bytes

Figure 6-3: Ship-to-Shore Packet Format

The details of the headers are handled by the C&C and the WGMS and are not of concern for payload development.

6.1.5 Mobile-Originated Communications

Communications from the Vehicle are categorized as "Mobile Originated" ("MO"). That is, all Iridium communications originate from the Vehicle. The general types of Mobile Originated Communications are:

1) Position Data
2) Vehicle Status
3) Data forwarded from payload(s)
4) Alarms

6.1.6 Mobile-Terminated Communications

Communications from the Shore-side are categorized as "Mobile-Terminated". These are basically commands to the Vehicle, of the following type:

1) Request that a Vehicle take some kind of action
2) Request status or other data from the Vehicle

However, when a command is issued by an On-Shore user, it is not immediately sent to the Vehicle. Commands are only sent to the Vehicle when requested by the Vehicle, after the Vehicle has sent an MO communication.

6.1.7 Commands and Responses

Communications from the Vehicle are always initiated on the Vehicle. Thus, if a command is issued on shore, it will not be immediately sent to the Vehicle, but will instead be queued for sending.

The C&C periodically uploads data (such as telemetry) to Shore, and then immediately downloads the next waiting command from the Shore system. The C&C has to first process the command before it can provide an ACK/NAK response. When the command has been processed, the C&C then initiates an upload of the ACK/NAK to shore. If there is an additional command queued, that will be downloaded to the Vehicle. This process continues until there are no more commands waiting

In most cases, the ACK/NAK response to any command sent from the Shore would be in the next communication from the Ship. However, if communication was interrupted, there may be multiple Vehicle data packets queued-up. When communications resumes, the Shore will see multiple incoming data packets after a command has been sent, and before the ACK/NAK will be received.

Any software designed for real-time processing of Vehicle responses needs to consider this communications characteristic.

6.1.8 WGMS Identification Number

The WGMS ID is a unique 4-byte identification number stored in Non-Volatile Memory in the Vehicle's C&C. This number is assigned and uploaded when the Vehicle is first brought online. It is used to identify the Vehicle when data packets are sent by means other than Iridium.

6.1.9 Iridium Message Buffering

Iridium service may occasionally be interrupted for up to several minutes. In this case, the C&C electronics will buffer any responses and telemetry, and then send out multiple responses and telemetry when service is resumed.

The message buffer is 16 responses deep. Once the 16-message limit is reached, any additional messages will be added to the buffer, and the oldest message discarded. Thus, for very long Iridium disruptions, some data may be lost.

6.2 Payload Communications

Communication between the C&C and payloads is similar in concept to the Iridium/Relay communications in that they use fixed format, variable-length data packets. However, the data packet formats are different.

Refer to Chapter 10 for detailed information on payload communications.

6.2.1 Payload Interface Boards

Within each payload is one or more electronic Payload Interface Boards ("PIBs"). Each board has a unique hard-coded serial number that is used as the board's "address". The C&C itself has its own reserved addresses.

Payload communications are between the PIBs, the C&C and customer-designed payload boards.

6.2.2 Payload Message Packets

All messages are transmitted in the form of fixed-structure, variable-length data packets.

Packets used for payload communications are hierarchical in that there is a main packet with a header, a message body and an error check. However, the header and message data formats are specific to payload communications.

The general structure of a payload data packet is shown in Figure 6-4.

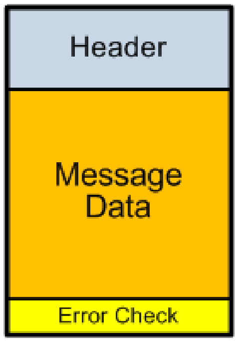


Figure 6-4: Message Packet Format

Payload data packets by themselves are not considered to be "Level 2" or "Level 3". However, depending upon the message, the Message Data can contain a Level 1 or Level 2 Command (ref. Chapter 7), Response (ref. Chapter 8) or Telemetry (ref. Chapter 9) packet.

6.3 XBee Radio Communications

XBee communications are substantially different than Iridium and Payload communications. However, note that the Iridium interface can be used to turn Off/On the XBee communications.

6.3.1 XBee Menu Format

XBee commands are issued using a menu system. A menu table is presented that contains a list of numbered menu commands. Entering a number executes the command. The computer interface will prompt for additional data as necessary.

The XBee Menu is described in detail in Chapter 11.

6.3.2 ASCII XBee Strings

XBee communication does not use packets. Rather XBee sends and receives variable-length ASCII (1-byte) character strings, ending with either a 1-byte carriage-return (CR) or a 2-byte carriage-return/line-feed combination. There is no header.

XBee commands sent by the User have a Carriage-Return termination, as represented below.

byte	*byte*	*byte*	. . .	. . .	. . .	*byte*	*byte*	**CR**

Vehicle XBee responses have a Carriage-Return/Line-Feed termination, as shown.

byte	*byte*	*byte*	. . .	. . .	. . .	*byte*	*byte*	**CR**	**LF**

The maximum length of any XBee data string is 255 bytes, including the terminating byte(s). Communication speed is fixed at 115,200 bps.

6.3.3 XBee Data Processing

Commands sent via XBee to the Vehicle are interpreted by the C&C computer, which performs any requested action. If the command requires a response, the C&C sends the appropriate response back to the XBee message originator. If the command requests a Vehicle action (e.g., move rudder), there is usually no XBee response. Any responses are displayed as on-screen text.

Use of XBee requires a serial port terminal emulation program, such as HyperTerminal or Indigo.

6.4 Communication Data Formats

Several of the message packets and XBee responses contain multi-byte representations of certain values. The formats of these are defined below.

6.4.1 Little-Endian Byte Format

All multi-byte data values are represented in "little-endian" format. That is, values are stored with the least-significant byte first, and the most-significant byte last. This includes multi-byte values stored in XBee data strings.

As an example of 2-byte little-endian, the value **0xAABB** will be stored as:

LSB	MSB
0xBB	**0xAA**

Figure 6-5: 2-Byte Little-Endian Byte Order

The 4-byte value **0xAABBCCDD** will be stored as:

LSB			MSB
0xDD	**0xCC**	**0xBB**	**0xAA**

Figure 6-6: 4-Byte Little-Endian Byte Order

Some data values are encoded in bit fields within bytes. The numbering of the bits is right-to-left, as shown in the figure below.

7	**6**	**5**	**4**	**3**	**2**	**1**	**0**

Figure 6-7: Bit Order Within Bytes

6.4.2 Waypoint Values

Each waypoint is defined by a waypoint number, a Latitude value and a Longitude value.

The waypoint number is a single-byte value from 1 to 254 (the value zero cannot be used), and two waypoints <u>cannot</u> use the same number. However, the numbers are not required to be sequential, although there are advantages to defining a course using sequential numbers.

Latitude and Longitude values are each stored as a 4-byte value (little-endian format). They are defined in units of 1/10,000 of an arc-minute, providing a resolution of about 0.2 meters.
The sign (+ or –) indicates the direction as follows.

Latitude: (+) → North (–) → South [–54,000,000 to +54,000,000]
Longitude: (+) → East (–) → West [–108,000,000 to +108,000,000]

6.4.3 Heading Values

Heading values are in whole Degrees from 0 to 359. The value cannot be negative. Headings are stored as a 2-byte value in little-endian format.

6.4.4 Distance Values

Distances are indicated in meters, and are stored as 2-byte little-endian values.

6.4.5 Time Values

For any given command or response value, time values are specified in whole-number minutes, seconds or milliseconds.

Minutes: Stored as a 2-byte value with a typical range of 1 to 1440 minutes.
(1440 minutes = 24 hours).

Seconds: Stored as a 1-byte value with a range of 1 to 255.

Milliseconds: Stored as 1-byte value with a range of 1 to 255.

6.4.6 Heading Values

Heading values are in whole Degrees from 0 to 359. The value cannot be negative. Headings are stored as a 2-byte value in little-endian format.

6.4.7 Distance Values

Distances are indicated in meters, and are stored as a 2-byte value.

6.4.8 Temperature Values

Temperature values are stored as a 2-byte signed value in units of °C / 10, thus giving 1/10th degree resolution.

6.4.9 File Names

A file name specified in a command field must be an 8+3, 11-byte ASCII string padded with spaces.
Only the characters 0x20 thru 0x7E are allowed (i.e., standard keyboard characters).
The separating dot (between the name and extension) is omitted.

For example, `DATA.TXT` is stored as:

0	1	2	3	4	5	6	7	8	9	10
D	A	T	A					T	X	T

`foo.c` is stored as:

0	1	2	3	4	5	6	7	8	9	10
f	o	o						c		

6.5 Navigation Modes

Once a set of one or more Waypoints have been defined, the Vehicle is instructed to proceed in one of several Navigation Modes.

The Navigation Modes and their associated reference number are shown in the table below. The reference number is used in commands to specify or indicate the mode.

Ref. Number	Navigation Mode	Explanation
0x00	(not defined)	
0x01	Course Loop	Continuous loop thru sequential Waypoints
0x02	Course Hold	Move thru sequential Waypoints, then hold position at last Waypoint
0x03	Custom Loop	Continuous loop thru non-sequential Waypoints
0x04	Custom Hold	Move thru non-sequential Waypoints, then hold position at last Waypoint
0x10	Custom Heading	Maintain a fixed heading
0x20	Rudder Position	Maintain a fixed rudder position
0x40	Target Waypoint	Move to a Target Waypoint and hold there
0x41	Hold Position	Maintain current position

Table 6-1: Navigation Modes

6.6 Roles and Permissions

Users on the Shore-based system are assigned "Roles", depending upon who they are and what they need to do. Each Role has differing Permissions that restrict what actions they can take and what commands they can use.

The Roles apply only to Iridium/Relay communications. There are no restrictions on what features can be used via XBee communications.

6.6.1 Entities

Entities are the primary components of the WGMS. Entities are typically groups of data, but can also be definitions of data structures and presentation formats. Entities are described in more detail in Appendix A.

6.6.2 Roles

The defined Roles are

Guest User: A Guest User can view the status of Vehicles that have been specifically marked for display to guests.

Data Analyst: A Data Analyst looks at existing Vehicle and WGMS data and can download database data.

Master: A Master manages Vehicle provisioning, mission planning and operations, as well as payload and web services development.

Admin: A customer System Administrator has all the permissions required to go through the day-to-day administration of the WGMS, add users, manage customizations, and deal with data.

Root: A Root user has full access to all aspects of the Vehicle and WGMS. This user will be either an LRI developer or a customer under close direction from LRI.

Note: Admin users can assign themselves a Root Role. However, this should not be done unless the Admin is in close contact with an LRI person. Incorrect use of Root permissions can result in corruption of the WGMS, loss of data, and failure of Vehicle communication.

Roles and permissions are shown in the table below. In the table, the "C-R-U-D" codes indicate the following permissions:

C – Create. User can add a new entity.

R – Read. User can access information about an entity.

U – Update. User can update a previously-created entity.

D – Delete. User can delete an entity.

Permission Type	Guest User	Data Analyst	Master	Admin	Root	Description
Area Permissions						
Home Pages	• R • •	• R • •	• R • •	• R • •	C R U D	Grids and top levels of the main window.
System Alarms	• R • •	• R • •	• R U •	• R U •	C R U D	Information for alarm escalation
Records	• • • •	• • • •	• R • •	• R U •	C R U D	Top-level metadata records. (outputs, vehicles, permissions)
Lists	• • • •	• R • •	C R U D	C R U D	C R U D	Escalation lists, user groups (future)
Dashboard Pages	• R • •	• R • •	• R • •	• R • •	C R U D	Allows user to edit dashboard pages.
List Items	• • • •	• R • •	C R U D	C R U D	C R U D	Items in a list
Report	• • • •	C R U D	C R U D	C R U D	C R U D	Data reports

Permission Type	Guest User	Data Analyst	Master	Admin	Root	Description
Users, Roles Permissions						
Users	• • • •	• R U •	• R U •	C R U D	C R U D	User data
Organizations	• • • •	• R • •	• R • •	• R U •	C R U D	Organization parameters
Notes	• • • •	C R U D	C R U D	C R U D	C R U D	Any entity can have a note attached (user, vehicle) (html or text)
Accounts	• • • •	• • • •	• • • •	• • • •	C R U D	(For future use)
Catalog Images	• • • •	• R • •	C R U D	C R U D	C R U D	Logos, images, photographs
Roles	• • • •	• • • •	• • • •	• R U •	C R U D	Editing the roles available in the system
User Roles	• • • •	• • • •	• • • •	C R U D	C R U D	Mapping between users and roles
Audit Records	• • • •	• R • •	• R • •	• R • •	C R U D	Entity change history (parts, vehicle changes)
Vendors	• • • •	• • • •	• • • •	• • • •	C R U D	For future use
Attachments	• • • •	• R • •	C R U D	C R U D	C R U D	Files attached to an entity
Permissions	• • • •	• • • •	• • • •	• R • •	C R U D	What permissions are allowed in the system
Role Permissions	• • • •	• • • •	• • • •	• R • •	C R U D	Mapping between roles and permissions
Workflow Events	• • • •	• • • •	• • • •	• • • •	C R U D	For future use
Inventory Items	• • • •	• • • •	• • • •	• • • •	C R U D	For future use
Record Customization						
Fields	• • • •	• R • •	• R • •	• R U •	C R U D	Metadata for each entity field (user name) (change the name of a column)
Data Views	• • • •	• • • •	• R • •	C R U D	C R U D	Metadata defining how entities are displayed (column grids) (change column in a grid)
Relationships	• • • •	• • • •	• R • •	• R • •	C R U D	Structure that tells how entities are related. (inverse of foreign keys)
Criterias	• • • •	• • • •	• R • •	C R U D	C R U D	Additional conditions on grids (filters)
Selects	• • • •	• • • •	• R U •	C R U D	C R U D	Values for fields with multiple values
Criteria Groups	• • • •	• • • •	• R • •	C R U D	C R U D	Grouping class for criteria
Payload	• • • •	• • • •	• • • •	• • • •	C R U D	Edit Payload display structures

Permission Type	Guest User	Data Analyst	Master	Admin	Root	Description
Vehicle Permissions						
Vehicles	• • • •	• R • •	C R U D	C R U D	C R U D	Access to records on vehicles in database (Guests can see vehicles marked "visible for guest")
Commands	• • • •	• R • •	C R U •	C R U •	C R U D	Entities representing commands
Telemetry	• R • •	• R • •	• R • •	• R • •	C R U D	Data coming back from vehicles
Statuses	• • • •	• R • •	C R U •	C R U •	C R U D	Statuses coming and going from vehicle
Event Definitions	• • • •	• R • •	C R U D	C R U D	C R U D	Alarm definitions
Rudder Controls	• • • •	• R • •	• R • •	C R • •	C R U D	Parameters for rudder module control loop.
Parameters	• • • •	• R • •	C R U •	C R U •	C R U D	Data user can send to vehicle
Communication Channels	• • • •	• • • •	• • • •	• R U •	C R U D	Automatically created by relays
Parts	• R • •	• R • •	C R U D	C R U D	C R U D	Manufacturing track listing of what's on vehicle
Courses	• • • •	• R • •	C R U D	C R U D	C R U D	Vehicle routes
Part Identifications	• • • •	• R • •	• R • •	• R • •	C R U D	Vehicle-generated list of parts
Missions	• • • •	• R • •	C R U •	C R U D	C R U D	Time-delineated vehicle status that states a vehicle is in water.
Operation Polygons	• • • •	• R • •	C R U D	C R U D	C R U D	Geographical area for vehicle presence
File Notifications	• • • •	• • • •	• • • •	• • • •	C R U D	Entities used during file transfers
Errors	• • • •	• R • •	• R U •	• R U •	C R U D	Vehicle Events
Raw Outputs	• R • •	• R • •	• R • •	• R • •	C R U D	Telemetry received from the vehicle
Waypoints	• R • •	• R • •	C R U D	C R U D	C R U D	Waypoint data associated with a vehicle.
Course Waypoints	• • • •	• R • •	C R U D	C R U D	C R U D	How a course is mapped to a waypoint
Power Ctl Statuses	• • • •	• R • •	• R • •	• R • •	C R U D	Telemetry
Mission Vehicles	• • • •	• R • •	C R U •	C R U •	C R U D	Mapping between missions and vehicles
Operation Coordinates	• • • •	• R • •	C R U D	C R U D	C R U D	Coordinates of an operational polygon

Permission Type	Guest User	Data Analyst	Master	Admin	Root	Description
Payload Data Permissions						
Payload Packets	• R • •	• R • •	• R • •	• R • •	C R U D	Generic data from the payloads
Payload Statuses	• R • •	• R • •	• R • •	• R • •	C R U D	Parsed payload packets
AISyG Locations	• R • •	• R • •	• R • •	• R • •	C R U D	AIS updates
Weather Stations	• R • •	• R • •	• R • •	• R • •	C R U D	Weather station telemetry
AISyG Statuses	• R • •	• R • •	• R • •	• R • •	C R U D	AIS updates
Weather Reports	• R • •	• R • •	• R • •	• R • •	C R U D	Weather station telemetry
AISyG Records	• R • •	• R • •	• R • •	• R • •	C R U D	AIS vehicles seen
Site Permissions						
Site Areas	• • • •	• • • •	• • • •	• • • •	C R U D	Allows different orgs to change page structure (home, vehicles…)
Site Groups	• • • •	• • • •	• • • •	• • • •	C R U D	Groups of items (Vehicles, Payload data)
Site Items	• • • •	• • • •	• • • •	• • • •	C R U D	Items that appear within a group
Custom Entity						
Added Records	• • • •	• • • •	• • • •	• • • •	C R U D	Ability to add new entities without changing the database
Added Fields	• • • •	• • • •	• • • •	• • • •	C R U D	
Added Mappings	• • • •	• • • •	• • • •	• • • •	C R U D	Relationships between added records and existing entities
Collision Avoidance Permissions						
Obstacles	• • • •	• • • •	• • • •	• • • •	C R U D	Obstacle in the glider's path
Obstacle Locations	• • • •	• • • •	• • • •	• • • •	C R U D	Obstacle telemetry
Vehicle Obstacles	• • • •	• • • •	• • • •	• • • •	C R U D	Mapping between obstacles and vehicles

This Page Intentionally Left Blank

7. Commands

Commands are primarily used either to specify some action by the Vehicle, or else to request information from the Vehicle. Commands are usually sent via the Iridium system, although alternate methods are possible.

Command packets are received by the C&C and processed to remove any Level 3 transmission headers (Iridium or Relay). The C&C then interprets the data in the command packet and responds appropriately.

7.1 Command Data Packet Format

The Level 2 Command Packets have a leading header and a trailing CRC error check. In addition, the Command message itself always has a 3-byte header that defines the command type and structure. This is shown in Figure 7-1.

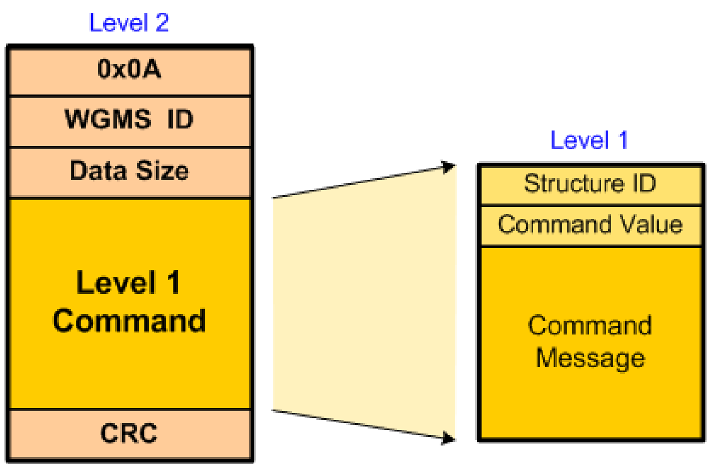


Figure 7-1: Command Packet Format

In the Level 2 Command Packet:

0x0A: [1 byte] Indicates a Command Data Packet.

WGMS ID: [4 bytes] An LRI-defined ID stored in the WGMS database.

Size: [2 bytes] The size of the Command message that follows.

Command: [max 217 bytes] The Level 1 Command Packet. This packet always starts with a Structure ID and a Command Value.

CRC: [2 bytes] The CRC16 of all bytes in the data packet (except the two CRC bytes).

In the Level 1 Command Message:

Structure ID: [1 byte] Defines the expected size and format of the Command Data.

Command Value: [2 bytes] A number specific to the Command Data type. Tells the computer the type of command (ref. 7.2).

Command Message: Variable-length field containing the actual command.

A summary of the Command Structure IDs and Values are shown in the summary table below. The Command Data descriptions are shown in section 7.3.

7.2 Command Summary

Listed below are the current user-accessible commands, with their corresponding Structure IDs and Command Values. Detailed descriptions are given in the sections referenced.

There are additional restricted-use commands, described in section 7.4.

Command Name	Command Value	Struct. ID	Ref.
No Command	**0x0000**	**0x10**	7.3.1
Hold Current Position	**0x0001**	**0x10**	7.3.2
Hold Station At Waypoint	**0x0002**	**0x10**	7.3.3
Follow Sequential Waypoint Course	**0x0003**	**0x10**	7.3.4
Right Rudder Left Rudder Center Rudder	**0x0004** **0x0005** **0x0006**	**0x10**	7.3.5
Follow Fixed Heading	**0x0008**	**0x10**	7.3.6
Set Default Comm Channel	**0x000A**	**0x10**	7.3.7
Read Vehicle WGMS ID	**0x000E**	**0x10**	7.3.8
Set Rudder Position	**0x1006**	**0x10**	7.3.9
Visible Light Off Visible Light On	**0x2001** **0x2002**	**0x10**	7.3.10
IR Light Off IR Light On	**0x2003** **0x2004**	**0x10**	7.3.11
XBee Comm Off XBee Comm On	**0x2005** **0x2006**	**0x10**	7.3.12
Retrieve Status	**0x4001**	**0x10**	7.3.13
Set Status	**0x4002**	**0x11**	7.3.14
Retrieve Identification	**0x4003**	**0x10**	7.3.15
Retrieve Power Status	**0x5001**	**0x10**	7.3.16
Retrieve Rudder Control	**0x5002**	**0x10**	7.3.17
Set Rudder Control	**0x5003**	**0x16**	7.3.18
Set Parameters	**0x6000**	**0x15**	7.3.19
Set Single Waypoint	**0x8000**	**0x10**	7.3.20
Read Waypoint	**0x8003**	**0x10**	7.3.21
Follow Course and Loop	**0x8101**	**0x12**	7.3.22
Follow Course With Options	**0x8102**	**0x12**	7.3.23
Initialize File Write	**0x9000**	**0x18**	7.3.24
Write File Block	**0x9001**	**0x19**	7.3.25
Delete File	**0x9002**	**0x1A**	7.3.26
Read File Request	**0x9003**	**0x1B**	7.3.27
Get Detailed File List	**0x9004**	**0x1C**	7.3.28

Command Name	Command Value	Struct. ID	Ref.
Select Program	**0x9005**	**0x10**	7.3.29
Write Console Input to Vehicle	**0x9006**	**0x1E**	7.3.30
Read File Transaction Closed	**0x9007**	**0x10**	7.3.31
Get File List	**0x9008**	**0x31**	7.3.32

Note in the table that several commands can share the same Structure ID.

7.3 Command Descriptions

Listed below are descriptions and explanations of the commands, and the expected response for each. The ACK/NAK format types are described in Chapter 8.

Note in the description below, that all fields marked "(reserved)" or "(not used)" must have a value of zero, or the CRC will trigger an error.

7.3.1 No Command (0x0000)

A "No Command" is <u>not sent to the Vehicle</u> but is stored in the WGMS vehicle command log along with a time-stamp and a comment text string. This can be useful for annotating tests.

The “No Command” should not be sent by a payload.

ACK/NAK – None

Field Name	Bytes	Value
Structure ID	1	**0x10**
Command Value	2	**0x0000**

7.3.2 Hold Current Position (0x0001)

Commands the Vehicle to maintain position at a specified Latitude/Longitude.

ACK – Status Return (ref. 8.2.2)

NAK – Data Error (ref. 8.3.2) if Latitude or Longitude error

Field Name	Bytes	Value
Structure ID	1	**0x10**
Command Value	2	**0x0001**
(reserved)	5	0
Latitude	4	[ref. 6.4.2]
Longitude	4	[ref. 6.4.2]

7.3.3 Hold Station at Waypoint (0x0002)

Commands the Vehicle to move to the specified Target Waypoint and maintain position there.

ACK – Status with Waypoint Read-Back (ref. 8.2.1)

NAK – Data Error (ref. 8.3.2) if Target Waypoint has not been defined

Field Name	Bytes	Value [range]
Structure ID	1	**0x10**
Command Value	2	**0x0002**
Target Waypoint Number	1	[1 to 254]
(reserved)	4	0
Latitude	4	0 (not used)
Longitude	4	0 (not used)

7.3.4 Follow Sequential Waypoint Course (0x0003)

Commands Vehicle to move through a contiguous sequence of Waypoints (e.g., 11, 12, 13, 14, 15, 16), specified by using a Begin-Waypoint and an End-Waypoint number (e.g., 11 and 16).
The Vehicle will repeatedly cycle through the Waypoint sequence until commanded otherwise (e.g., . . . 14, 15, 16, 11, 12, 13, 14, 15, 16, 11, 12, 13, 14, . . .).

The Target Waypoint is the first Waypoint that the Vehicle goes to at the start, and need not be the same as the Begin-Waypoint number. However, the Target Waypoint must be in the sequence.

ACK – No Payload (ref. 8.2.4)

NAK – Data Error (ref. 8.3.2) if missing a Waypoint number in the sequence. For example, if the course is from Waypoint 9 to Waypoint 20, but Waypoint 17 was never defined.

NAK – Data Error (ref. 8.3.2) if the Target Waypoint is not within the sequence. For example, if the sequence is from 5 to 18, but the Target Waypoint is 4.

Field Name	Bytes	Value [range]
Structure ID	1	**0x10**
Command Value	2	**0x0003**
Target Waypoint #	1	[within sequence]
Begin Waypoint #	1	[1 to 254]
End Waypoint #	1	[1 to 254]
(reserved)	2	0
Latitude	4	0 (not used)
Longitude	4	0 (not used)

7.3.5 Rudder Right/Left/Center (0x0004, 0x0005, 0x0006)

Commands the Glider's rudder to move either to full left, full right, or to center.

ACK – No Payload (ref. 8.2.4)

Field Name	Bytes	Value
Structure ID	1	**0x10**
Command Value	2	**0x0004** (Right Rudder) **0x0005** (Left Rudder) **0x0006** (Center Rudder)
(reserved)	5	0
Latitude	4	0 (not used)
Longitude	4	0 (not used)

7.3.6 Follow Fixed Heading (0x0008)

Commands the Vehicle to immediately proceed on a specified heading from its current position. The heading value is specified in unit degrees.

ACK – No Payload (ref. 8.2.4)

NAK – Data Error (ref. 8.3.2) if the heading value exceeds 359 degrees.

Field Name	Bytes	Value
Structure ID	1	**0x10**
Command Value	2	**0x0008**
(reserved)	3	0
Heading in Degrees	2	[0 to 359]
Latitude	4	0 (not used)
Longitude	4	0 (not used)

7.3.7 Set Default Communication Channel (0x000A)

The command allows specification of the Vehicle's communication channel. The Vehicle normally uses the Iridium channel.

Note: If the specified channel is not available, the C&C will switch to Iridium communications.

ACK – No Payload (ref. 8.2.4)

Field Name	Bytes	Value
Structure ID	1	**0x10**
Command Value	2	**0x000A**
Channel Number	1	Ref. chart →
(reserved)	4	0
Latitude	4	0 (not used)
Longitude	4	0 (not used)

Channnel Numbers

1	Iridium (default)
2	XBee
3	Serial
4	WiFi
5	Alternate Iridium

7.3.8 Read Vehicle WGMS ID (0x000E)

A unique WGMS ID should have been written into the C&C's Non-Volatile Memory when the Vehicle was first brought online.

If an ID was never set, the returned value will be **0xFFFFFFFF** (–1 decimal).

Note: This command can only be issued via Iridium communications.

ACK – Structure type 0x1E (ref. 8.2.9) containing the WGMS ID

Field Name	Bytes	Value
Structure ID	1	**0x10**
Command Value	2	**0x000E**
(reserved)	5	0
Latitude	4	0 (not used)
Longitude	4	0 (not used)

7.3.9 Set Rudder Position (0x1006)

Commands the Glider's rudder to move to a position, specified in unit degrees.

ACK – No Payload (ref. 8.2.4)

Field Name	Bytes	Value [range]	Note
Structure ID	1	**0x10**	
Command Value	2	**0x1006**	
Position	1	Degrees [–30° to +30°]	[1]
(reserved)	4	0	
Latitude	4	0 (not used)	
Longitude	4	0 (not used)	

Note:

[1] The sign of the "Position" value is interpreted as follows:

Negative → Left Rudder
Positive → Right Rudder

7.3.10 Visible Light Off/On (0x2001, 0x2002)

Turns On or Off the visible light.

ACK – No Payload (ref. 8.2.4)

Field Name	Bytes	Value
Structure ID	1	**0x10**
Command Value	2	**0x2001** (Light Off) **0x2002** (Light On)
(reserved)	5	0
Latitude	4	0 (not used)
Longitude	4	0 (not used)

7.3.11 IR Light Off/On (0x2003, 0x2004)

Turns On or Off the Infrared (IR) light.

ACK – No Payload (ref. 8.2.4)

Field Name	Bytes	Value
Structure ID	1	**0x10**
Command Value	2	**0x2003** (IR Off) **0x2004** (IR On)
(reserved)	5	0
Latitude	4	0 (not used)
Longitude	4	0 (not used)

7.3.12 XBee Comm Off/On (0x2005, 0x2006)

Turns On or Off the XBee communications channel.

ACK – No Payload (ref. 8.2.4)

Field Name	Bytes	Value
Structure ID	1	**0x10**
Command Value	2	**0x2005** (XBee Off) **0x2006** (XBee On)
(reserved)	5	0
Latitude	4	0 (not used)
Longitude	4	0 (not used)

7.3.13 Retrieve Status (0x4001)

Retrieves the Status Return data packet. Refer to section 8.2.2 for a detailed description of the Status Return packet.

ACK – Status Return (ref. 8.2.2)

Field Name	Bytes	Value
Structure ID	1	**0x10**
Command Value	2	**0x4001**
(reserved)	5	0
Latitude	4	0 (not used)
Longitude	4	0 (not used)

7.3.14 Set Status (0x4002)

Sends new operational parameters to the Vehicle. This command has to send many parameters at one time. For sending only certain parameters, the Set Parameters command (ref. 7.3.19) can be used instead.

ACK – Status Return (ref. 8.2.2)

Field Name	Bytes	Value [range] (units)	Notes
Structure ID	1	**0x11**	
Command Value	2	**0x4002**	
(not used)	1	0	
Telemetry Rate	2	[1 – 1440] (minutes)	[1]
Low_Battery Report Rate	2	[1 – 1440] (minutes)	[2]
Alarm Report Rate	2	[1 – 30] (minutes)	[3]
(reserved)	1		
Sub_Float Timeout	1	[20 – 200] (millisecs)	[4]
(reserved)	2	0	
Float Low_Battery Power	1	[1 – 65] (10Wh steps)	[5]
Float Critical_Battery Power	1	[1 – 65] (10Wh steps)	[6]
(reserved)	3	0	
Float Pressure_Threshold High	1	[0 – 255] (kPa)	[7]
Float Pressure_Threshold Low	1	[0 – 255] (kPa)	[7]
Float Temperature_Threshold	1	[0 – 255] (°C)	[8]
Waypoint Arrival Distance	2	[5 – 5000] (meters)	[9]
(reserved)	1	0	
Power Controller Telemetry Rate	2	[5 – 2880] (minutes)	[10]
(reserved)	2	0	
Float-to-Sub Comm Period	1	[1 – 240] (seconds)	[11]
(not used)	8	0	

Notes:

[1] "Telemetry Rate" is the number of minutes between telemetry transmissions.

[2] "Low_Battery Report Rate" telemetry rate is used when battery is below the "Float Low_Battery Power" value (ref. note [5]).

[3] "Alarm Report Rate" is the telemetry rate used when an alarm has been triggered.

[4] "Sub_Float Timeout". Failure of the Sub to communicate with the C&C within this time triggers an alarm.

[5] "Float Low_Battery Power". Level at which the battery power is considered "Low".

[6] "Float Critical_Battery Power". Level at which the battery power is considered "Critical".

[7] "Float Pressure_Threshold High/Low". Max and Min allowed C&C Drybox pressures before an alarm is triggered. Thresholds are adjusted for temperature using the Ideal Gas Law, pV=nRT, referenced to 23°C.

[8] "Float Temperature_Threshold" is the maximum C&C Drybox temperature allowed before an alarm is triggered.

[9] " Waypoint Arrival Distance" is the distance from the exact Waypoint coordinates at which the Vehicle is considered to have "arrived" at the Waypoint.

[10] "Power Controller Telemetry Rate" is the telemetry rate for the Power Controller Status.

[11] "Float-to-Sub Comm Period" is the rate at which the Float sends a command to the Sub.

7.3.15 Retrieve Identification (0x4003)

Requests the Vehicle to send back the ID Packet. Refer to 8.2.3 for a description of the ID Packet format.

ACK – Identification Packet (ref. 8.2.3)

Field Name	Bytes	Value
Structure ID	1	**0x10**
Command Value	2	**0x4003**
(reserved)	5	0
Latitude	4	0
Longitude	4	0

7.3.16 Retrieve Power Status (0x5001)

Requests the Vehicle to send back the Power Controller Status. Refer to 8.2.5 for a description of the Power Controller Status Packet format.

ACK – Power Controller Packet (ref. 8.2.5)

Field Name	Bytes	Value
Structure ID	1	**0x10**
Command Value	2	**0x5001**
(reserved)	5	0
Latitude	4	0
Longitude	4	0

7.3.17 Retrieve Rudder Control (0x5002)

Requests the Vehicle to send back the Rudder Control Packet. Refer to 8.2.6 for a description of the Rudder Control Packet format.

ACK – Rudder Control Packet (ref. 8.2.6)

Field Name	Bytes	Value
Structure ID	1	**0x10**
Command Value	2	**0x5002**
(reserved)	5	0
Latitude	4	0
Longitude	4	0

7.3.18 Set Rudder Control (0x5003)

Sends the Rudder Control data parameters to the Vehicle. Note that these parameters define the control laws for the rudder, not the rudder position.

There are three parameter sets:

- "Normal" values are the default.
- "Low" values are used when the battery energy is low.
- "Critical" values are used when the batteries are close to depletion.

Note that the initial default values are the same for all parameter sets. If different values are desired for the Low and Critical states, they have to be explicitly set.

ACK – Rudder Control Packet (ref. 8.2.6)

Field Name	Bytes	Value (units)	Default	Notes
Structure ID	1	**0x16**		
Command Value	2	**0x5003**		
Normal Gain	2	[1 to 100]	**0x7FFF**	[1]
Normal Timebase	1	[1 to 15] (15Hz/N)	**7** (2 Hz)	[2]
Normal Deadband	1	[0 to 60] (0.5°)	**30** (15°)	[3]
Normal Goal	1	[0 to 10] (0.5°)	**1** (0.5°)	[4]
Normal RudderMax	1	[0 to 60] (0.5°)	**40** (20°)	[5]
Normal MoveMin	1	[0 to 10] (0.5°)	**2** (1°)	[6]
Normal TrimMax	1	[0 to 10] (0.5°)	**0**	[7]
Low Gain	2	[1 to 100]	**0x7FFF**	
Low Timebase	1	[1 to 15] (15Hz/N)	**7** (2 Hz)	
Low Deadband	1	[0 to 60] (0.5°)	**30** (15°)	
Low Goal	1	[0 to 10] (0.5°)	**1** (0.5°)	
Low RudderMax	1	[0 to 60] (0.5°)	**40** (20°)	
Low MoveMin	1	[0 to 10] (0.5°)	**2** (1°)	
Low TrimMax	1	[0 to 10] (0.5°)	**0**	
Critical Gain	2	[1 to 100]	**0x7FFF**	
Critical Timebase	1	[1 to 15] (15Hz/N)	**7** (2 Hz)	
Critical Deadband	1	[0 to 60] (0.5°)	**30** (15°)	
Critical Goal	1	[0 to 10] (0.5°)	**1** (0.5°)	
Critical RudderMax	1	[0 to 60] (0.5°)	**40** (20°)	
Critical MoveMin	1	[0 to 10] (0.5°)	**2** (1°)	
Critical TrimMax	1	[0 to 10] (0.5°)	**0**	
(unused)	4	0	0	
CRC32	4			

Notes:

[1] "Gain" is the rudder "Proportional" control multiplier, used as:

Correction = (**Gain** * 200 * Rudder_Deflection) / Heading_Error

Where Heading_Error is the error that exceeds the specified Deadband value. Deflection and Error are in degrees.

[2] "Timebase" is the integer number of control loop samples to be taken within the 15 Hz timing intervals. Thus a value of 5 will result in a 3 Hz control loop. The default value of 7 results in a 2 Hz control loop.

[3] "Deadband" is the minimum heading error at which the control system will start to take action. Defined in units of 0.5 degrees.

[4] "Goal" The rudder will be moved to the trim value when the heading error falls to this value. Defined in units of 0.5 degrees.

(Since the algorithm straightens the rudder when the heading error changes sign, this value can be made as small as needed without risk of the control loop going unstable.)

[5] "RudderMax" is the maximum allowed deflection of the rudder. Defined in units of 0.5 degrees.

[6] "MoveMin" is the minimum rudder movement. The desired position of the rudder has to be at least this much different from the actual position for the rudder to actually move. Defined in units of 0.5 degrees.

[7] "TrimMax" is the maximum trim value. Trim is an offset angle put into the rudder to compensate for any mechanical misalignment. Defined in units of 0.5 degrees.

7.3.19 Set Parameters (0x6000)

Sets the value of one or more specific parameters. This command can be used instead of the Set Status Command when only a few Vehicle parameters need to be changed.

There are also restricted-use parameters, as described in section 7.4.2.

ACK – No Payload (ref. 8.2.4)

Field Name	Bytes	Value [range]
Structure ID	1	**0x15**
Command Value	2	**0x6000**
Number of parameters	1	**N**
Parameter 1 ID	2	
Parameter 1 Length	1	
Parameter 1 Data	Bytes in Param 1	
.....	...	…
.....	...	…
.....	...	…
Parameter N ID	2	
Parameter N Length	1	
Parameter N Data	Bytes in Param N	

Parameter IDs and data that are currently supported:

Parameter	ID	Bytes	Type	Value [range]	Notes
Iridium Modem Always On	**0x0014**	1	Boolean	0 → No, 1 → Yes	[1]
Iridium Modem Default On	**0x0015**	1	Boolean	0 → No, 1 → Yes	[2]
Alternate Iridium Polling Rate	**0x001A**	1	Numeric	0 → Off, [1–120 min]	
Temporary Channel Retries	**0x001B**	1	Numeric	[1–32]	
Bubble Power	**0x0020**	1	Boolean	0 → Off, 1 → On	
Sub Power	**0x0021**	1	Boolean	0 → Off, 1 → On	
Payload 1 Power	**0x0100**	1	Boolean	0 → Off, 1 → On	
Payload 2 Power	**0x0200**	1	Boolean	0 → Off, 1 → On	
Sub Payload Power	**0x0300**	1	Boolean	0 → Off, 1 → On	
C&C Internal Payload Power	**0x0400**	1	Boolean	0 → Off, 1 → On	[3]
AIS Receiver Power	**0x0401**	1	Boolean	0 → Off, 1 → On	
Weather Station Power	**0x0402**	1	Boolean	0 → Off, 1 → On	
Water Sensor Power	**0x0403**	1	Boolean	0 → Off, 1 → On	
GPS Alarm Threshold	**0x000F**	1	Count	[0 to 255]	[4]

Notes:

[1] "Iridium Modem Always On" specifies that the Iridium modem should <u>not</u> turn OFF between transmissions.

[2] "Iridium Modem Default On" indicates that the Iridium modem will only turn ON when it is transmitting or expecting a message.

[3] The C&C interfaces with – and supplies power to – a payload within the C&C drybox. This parameter controls the power to that payload.

[4] "GPS Alarm Threshold" is the count of continuous invalid GPS fixes detected before the "GPS Not Functioning" alarm, is triggered.

7.3.20 Set Single Waypoint (0x8000)

Used to set a single new Waypoint Lat/Long value. Refer to 6.4.2 for a description of the Lat/Long values.

Waypoint numbers must be between 1 and 254.

ACK – No Payload (ref. 8.2.4)

NAK – Data Error (ref. 8.3.2) if no Waypoints in the system have been added or modified.

Field Name	Bytes	Value [range]
Structure ID	1	**0x10**
Command Value	2	**0x8000**
Waypoint Number	1	[1 to 254]
(reserved)	4	0
Latitude	4	[ref. 6.4.2]
Longitude	4	[ref. 6.4.2]

Note: When this command is sent from the WGMS, the system will respond to the ACK by automatically issuing a "Read Waypoint" command (ref. 7.3.21). Thus, using this command will result in two ACKs being returned.

7.3.21 Read Waypoint (0x8003)

Requests the Lat/Long coordinates of a specific Waypoint number.

ACK – Status with Waypoints Read-Back (ref. 8.2.1)

NAK – Data Error (ref. 8.3.2) if the specified Waypoint number hasn't been defined.

Field Name	Bytes	Value [range]
Structure ID	1	**0x10**
Command Value	2	**0x8003**
(reserved)	4	0
Waypoint Number	1	[1 to 254]
Latitude	4	0
Longitude	4	0

Note: When the ACK is returned, the WGMS will then compare the returned Lat/Long values against the values stored in the database. If they do not agree, an alarm will be issued.

7.3.22 Follow Course and Loop (0x8101)

Commands the Vehicle to follow a course from Waypoint to Waypoint in the order specified. The Waypoint numbers need not be sequential. When the Vehicle reaches the last Waypoint, it goes to the first Waypoint and continues looping until terminated by another command.

ACK – No Payload (ref. 8.2.4)

NAK – Data Error if: a) any undefined Waypoint numbers.
b) count does not match number of Waypoints specified.
c) the Target Waypoint is not one of the course Waypoints specified.

Field Name	Bytes	Value [range]	Notes
Structure ID	1	**0x12**	
Command Value	2	**0x8101**	
Target Waypoint	1	[1 to 254]	[1] [2]
Count of Waypoints	1	[1 to 213]	
Waypoint Number	1	[1 to 254]	
Waypoint Number	1	[1 to 254]	
…..	...	…	
…..	...	…	
Waypoint Number	1	[1 to 254]	

Notes:

[1] The Target Waypoint is the first Waypoint that the Vehicle goes to, and must be the same as one of the course Waypoints specified in the command.

[2] If the Target Waypoint is specified more than once in the list of waypoint numbers, the Vehicle will initially proceed toward the first course Waypoint in the list

7.3.23 Follow Course With Options (0x8101)

Commands Vehicle to follow a course from Waypoint to Waypoint in the order specified. An additional option field specifies whether to loop continuously, or to hold at the last Waypoint.

ACK – No Payload (ref. 8.2.4)

NAK – Data Error if: a) any undefined Waypoint numbers.
b) count does not match number of Waypoints specified.
c) the Target Waypoint is not one of the course Waypoints specified.
d) the Navigation Option is not 0 or 1.

Field Name	Bytes	Value [range]	Notes
Structure ID	1	**0x12**	
Command Value	2	**0x8102**	
Target Waypoint	1	[1 to 254]	[1]
Navigation Option	1	**0x00** or **0x01**	[2]
Count of Waypoints	1	[1 to 213]	
Waypoint Number	1	[1 to 254]	
Waypoint Number	1	[1 to 254]	
. . . . .	. . .	. . .	
. . . . .	. . .	. . .	
Waypoint Number	1	[1 to 254]	

Notes:

[1] The "Target Waypoint" is the first Waypoint that the Vehicle goes to, and must be the same as one of the course Waypoints specified in the command.

[2] The "Navigation Option" indicates what the Vehicle should do when it reaches the last Waypoint. The options are:

0x00 → Loop Continuously
0x01 → Hold at Last Waypoint

7.3.24 Initialize File Write (0x9000)

Start a data file download to the C&C or to a Payload.

ACK – No Payload (ref. 8.2.4)

NAK – Command Error (ref. 8.3.1)
Data Error (ref. 8.3.2)
File Error (ref. 8.3.3)

Field Name	Bytes	Value [range]	Notes
Structure ID	1	**0x18**	
Command Value	2	**0x9000**	
Storage Controller Address	2	[1 to 256] + [1 to 256]	[1]
File Name	11	8+3 ASCII string (ref. 6.4.9)	[2] [4]
File ID	4	WGMS-generated (or 0xFFFFFFFF)	[3] [4]
File Length	4	Number of bytes [≥1]	[5]
Creation Date	6	Time+Date	[6] [7]
Modified Date	6	Time+Date	[6] [7] [8]
File Attributes	1	Bit flags	[9]
Overwrite File Flag	1	**0x00** or **0xFF**	[10]
CRC-16 of content	2		[11]
MD5 of content	16		[11]
Number of bytes to write	2	**N** [≥0]	
File Data	0 to N		[12]

Notes:

[1] The "Storage Controller Address" is the 1-byte Board Address, and the 1-byte Task ID (ref. 5.3.5). The Float C&C has the reserved address value: **0x0000**

[2] The "File Name" is an ASCII string padded by spaces. Only the characters 0x20 thru 0x7E are allowed. (ref. 6.4.9)

[3] The "File ID" is typically generated by the WGMS. If this command is not generated by the WGMS, the value **0xFFFFFFFF** must be used.

[4] The following filenames and WGMS ID numbers are reserved; they cannot be written or deleted:
"**EEPROM**" (ID = **0x1**): Non-Volatile memory at the Storage Controller Address
"**PRGMEM**" (ID = **0x2**): Program memory at the Storage Controller Address
"**RAM**" (ID = **0x3**): Volatile memory at the Storage Controller Address

[5] The "File Length" is the total number of bytes to allocate for storing the file. It cannot be zero.

[6] Dates use 1 byte per value in the order shown below. Value ranges are enclosed in [ ].
Hour [0 to 23]
Minute [0 to 59]
Second [0 to 59]
Month [1 to 12]
Day [1 to 31]
Year-2000 [0 to 99]

[7] If all the time/date value fields are set to zero, the time/date value stored in the Vehicle's C&C will be used.

[8] The Modified Date field will be updated by any subsequent writes to the file content.

[9] The "File Attributes" flag bit flags indicate the following:
0x01 set → Read Only

0x08 set → (reserved)
All other bits are user-defined

[10] The "Overwrite File Flag" indicates what to do if the file already exists:
0x00: Don't overwrite the file (returns NAK)
0xFF: Overwrite the file

[11] The CRC-16 and MD5 values should be provided for later checking by the WGMS.

[12] There will be no "Data" field if there are no bytes to be written with this command.

7.3.25 Write File Block (0x9001)

Send a single block of data from the file.

ACK – No Payload (ref. 8.2.4)

NAK – Command Error (ref. 8.3.1)
Data Error (ref. 8.3.2)
File Error (ref. 8.3.3)

Field Name	Bytes	Value [range]	Notes
Structure ID	1	**0x19**	
Command Value	2	**0x9001**	
Storage Controller Address	2	[1 to 256] + [1 to 256]	[1]
File ID	4	**0xFFFFFFFF** (unless WGMS-generated)	[2]
Offset (Seek)	4	[0 to (File_Length - 1)]	[3]
Number of Bytes to Write	2	**N** [>0]	[4]
File Data	1 to N	[1 to (Packet_Size – 19)]	

Notes:

[1] The "Storage Controller Address" is the 1-byte Board Address, and the 1-byte Task ID (ref. 5.3.5). The Float C&C has the reserved address value: **0x0000**

[2] The "File ID" value **0xFFFFFFFF** must be used if this command is generated by a payload. (The ID is otherwise generated by the WGMS.)

[3] The "Offset: is the number of bytes from the start of the file at which the write begins.
Offset = 0 → Start of File

[4] The "Number of Bytes" cannot be zero. The limit is determined by the overall packet size.

7.3.26 Delete File (0x9002)

Delete a file from the selected storage address.

ACK – No Payload (ref. 8.2.4)

NAK – Command Error (ref. 8.3.1)
Data Error (ref. 8.3.2)
File Error (ref. 8.3.3)

Field Name	Bytes	Value [range]	Notes
Structure ID	1	**0x1A**	
Command Value	2	**0x9002**	
Storage Controller Address	2	[1 to 256] + [1 to 256]	[1]
File Name	11	8+3 ASCII string. (ref. 6.4.9)	[2]
File ID	4	**0xFFFFFFFF** (unless WGMS-generated)	[3]
Override Read Flag	1	**0x00** or **0xFF**	[4]

Notes:

[1] The "Storage Controller Address" is the 1-byte Board Address, and the 1-byte Task ID (ref. 5.3.5). The Float C&C has the reserved address value: **0x0000**

[2] The "File Name" is an ASCII string padded by spaces (ref. 6.4.9). This name will be ignored if the File ID is not **0xFFFFFFFF**.

[3] The "File ID" value **0xFFFFFFFF** must be used if this command is generated by a payload. (The ID is otherwise generated by the WGMS.)

[4] The "Override Read Flag" value indicates the following:
0x00 = Do not delete file if marked Read-Only
0xFF = Delete file even if the file is marked as Read-Only

7.3.27 Read File Request (0x9003)

Send a request to read a file from the Vehicle.

ACK – Read File ACK

NAK – Command Error (ref. 8.3.1)
Data Error (ref. 8.3.2)
File Error (ref. 8.3.3)

Field Name	Bytes	Value [range]	Note
Structure ID	1	**0x1B**	
Command Value	2	**0x9003**	
Storage Controller Address	2	[1 to 256] + [1 to 256]	[1]
File Name	11	8+3 ASCII string. (ref. 6.4.9)	[2]
Transaction ID	4	See note →	[3]
Offset (Seek)	4	[0 to (File_Length – 1)]	[4]
Read Length	4	**0** or [1 to File_Length]	[5]
Max Response Size	2	**175**	[6]

Notes:

[1] The "Storage Controller Address" is copied from the New File Available request.

[2] The "File Name" is an ASCII string padded by spaces. (ref. 6.4.9)

[3] The “Transaction ID” is -1 for the first read. For retries, it is the original Read File ID (ref. 7.3.31).

[4] The "Offset" is the number of bytes from the start of the file at which the write begins.
Offset = 0 → Start of File.

[5] The "Read Length" indicates how many bytes to read from the Offset.
Read Length = 0 → Read entire file starting from the Offset.

[6] The "Max Response Size" is the maximum number of bytes to allocate for the response.
The Vehicle may choose to report fewer bytes than requested.

7.3.28 Get Detailed File List (0x9004)

Used to request a directory listing or the listing for a specific file. Optionally, this command can be used to request a check of the validity of a file using MD5 and CRC, and delete files that fail the validity check.

If the Float C&C is implementing a “store and forward” protocol, a successful file check will transfer the file to its target storage controller, resulting in a second ACK.

ACK – Get Detailed File List Data (ref. 8.2.8)

ACK – File at Destination [2nd response if file transferred] (ref. 8.2.10)

NAK – Command Error (ref. 8.3.1)
Data Error (ref. 8.3.2)
File Error (ref. 8.3.3)

Field Name	Bytes	Value [range]	Notes
Structure ID	1	**0x1C**	
Command Value	2	**0x9004**	
Storage Controller Address	2	[1 to 256] + [1 to 256]	[1]
File Name	11	8+3 ASCII string. (ref. 6.4.9)	[2]
File ID	4	**0xFFFFFFFF** (unless WGMS-generated)	[3]
Action Flags	1	Bit Flags	[4]
Max Response Size	2	[1 to 300]	[5]

[1] The "Storage Controller Address" is the 1-byte Board Address, and the 1-byte Task ID (ref. 5.3.5). The Float C&C has the reserved address value: **0x0000**

[2] The "File Name" is an ASCII string padded by spaces (ref. 6.4.9). This name will be ignored if the File ID is <u>not</u> **0xFFFFFFFF**.

[3] The "File ID" value **0xFFFFFFFF** <u>must</u> be used if this command is generated by a payload. (The ID is otherwise generated by the WGMS.)

[4] The "Action Flags" are bit fields used to request the following:
0x01 bit = 1 → Files should be MD5 and CRC checked before reporting
0x02 bit = 1 → Files failing check should be deleted

[5] The "Max Response Size" is the maximum number of bytes to allocate for the response.
Note that the Vehicle may choose to respond with a smaller packet than requested.

7.3.29 Select Program (0x9005)

This command is used to select one or more program images. Each program can be specified as either the current selection, as a default selection, or as both.

For each program, the file MD5 will be checked, unless it has previously been checked. If the MD5 check fails for a program, that program will not be selected. Thus, this command can be used in place of a "Get File List" command to verify the validity of program files.

If the Float C&C is implementing a "store and forward" protocol, a successful file check will transfer the file to its target storage controller, resulting in a second ACK.

ACK – No Payload (ref. 8.2.4)

ACK – File at Destination [2nd response if file transferred] (ref. 8.2.10)

NAK – Command Error (ref. 8.3.1)
Data Error (ref. 8.3.2)
File Error (ref. 8.3.3)

Field Name	Bytes	Value [range]	Notes
Structure ID	1	**0x10**	
Command Value	2	**0x9005**	
Storage Controller Address	2	[1 to 256] + [1 to 256]	
Number of Program Selections	2	**N** [≥1]	[1]
Program Record 1	9	See below	
Program Record 2	9		
. . . . .	. . .		
. . . . .	. . .		
Program Record N	9		

Program Record Types are as follows:

Field Name	Bytes	Value [range]	Notes
Program Type Select	1	**0**, **1** or **2**	[2]
File ID	4	WGMS-generated only	[3]
Board Address	2	[1 to 256] + [1 to 256]	[4]
(reserved)	2	0	

Notes:

[1] At least one program must be specified.

[2] "Program Type Select" specifies the type of one program. Any other program on the board is de-selected. The select flags are:

0 = Select program as the default
1 = Select program as current (the one to be used)
2 = Select program as both default and current

[3] The "File ID" is the WGMS-generated number that will be used to tag all the ACKs with file blocks. These will then have to be assembled back to form the actual file.
(The ID **0xFFFFFFFF** cannot be used)

[4] The "Board Address" is the location (C&C or payload) where the program will be executed.

7.3.30 Write Console Input to Vehicle (0x9006)

Writes a stream of ASCII data to the C&C or a payload board on the Vehicle.

ACK – No Payload (ref. 8.2.4)

NAK – Command Error (ref. 8.3.1)
Data Error (ref. 8.3.2)

Field Name	Bytes	Value [range]	Notes
Structure ID	1	**0x1E**	
Command Value	2	**0x9006**	
Destination Address	2	[1 to 256] + [1 to 256]	[1]
Function	1	**0x00** for C&C, or else user-assigned	[2]
Number of Bytes in Data	2	**N** (≥1)	
Data	1 to N	ASCII data for Vehicle console input	

Notes:

[1] The "Destination Address" is the 1-byte Board Address, and the 1-byte Task ID (ref. 5.3.5). The Float C&C has the reserved address value: **0x0000**.

[2] The "Function" is a user-assigned code value that is passed to the destination board. If the destination board is the C&C, the Function value should be **0x00**.

7.3.31 Read File Transaction Closed (0x9007)

Sends a transaction closed message to the Vehicle to signify successful reception of the file. The C&C then removes the file from the Vehicle's memory.

ACK – No Payload (ref. 8.2.4)

NAK – Command Error (ref. 8.3.1)
Data Error (ref. 8.3.2)

Field Name	Bytes	Default Value	Note
Structure ID	1	**0x10**	
Command Value	2	**0x9007**	
Storage Controller Address	2	[1 to 256] + [1 to 256]	[1]
(reserved)	3	0	
File ID	4	WGMS-generated	
Longitude	4	0 (not used)	

Note:

[1] The "Storage Controller Address" is the 1-byte Board Address, and the 1-byte Task ID (ref. 5.3.5). The Float C&C has the reserved address value: **0x0000**.

7.3.32 Get File List (0x9008)

Used to request a directory listing or the listing for a specific file. Optionally, this command can request a check of the validity of a file using MD5 and CRC, and delete files that fail the validity check.

If the Float C&C is implementing a "store and forward" protocol, a successful file check will transfer the file to its target storage controller, resulting in a second ACK.

ACK – Detailed File List Data (ref. 8.2.8)

ACK – File at Destination [2nd response if file transferred] (ref. 8.2.10)

NAK – Command Error (ref. 8.3.1)
Data Error (ref. 8.3.2)
File Error (ref. 8.3.3)

Field Name	Bytes	Value [range]	Notes
Structure ID	1	**0x31**	
Command Value	2	**0x9008**	
Storage Controller Address	2	[1 to 256] + [1 to 256]	[1]
File Name	11	8+3 ASCII string. (ref. 6.4.9)	[2]
File WGMS ID	4	See note →	[3]
Action Flags	1	Bit Flags	[4]
Max Response Size	2	[1 to 300]	[5]

Notes:

[1] The "Storage Controller Address" is the 1-byte Board Address, and the 1-byte Task ID (ref. 5.3.5). The Float C&C has the reserved address value: **0x0000**

[2] The "File Name" is an ASCII string padded by spaces (ref. 6.4.9). This name will be ignored if the File ID is not **0xFFFFFFFF**.

[3] The "File ID" value **0xFFFFFFFF** must be used if this command is generated by a payload. (The ID is otherwise generated by the WGMS.)

[4] The "Action Flags" are bit fields used to request the following:
0x01 bit = 1 → files should be MD5 and CRC checked before reporting
0x02 bit = 1 → files failing check should be deleted

[5] The "Max Response Size" is the maximum number of bytes to allocate for the response. Note that the Vehicle may choose to respond with a smaller packet than requested.

7.4 Restricted Commands

Admins and (in some cases) Masters have access to these commands for development, debugging and error response.

Note: These commands are subject to change without notice.

Command Name	Command Value	Structure ID	Ref.
Set WGMS ID	**0x000F**	0x10	7.4.1
Set Parameters	**0x6000**	0x10	7.4.2
Retrieve Debug Buffer	**0x7000**	0x10	7.4.3
Retrieve Sub Heading Buffer	**0x7001**	0x10	7.4.4
Retrieve PID Integral Cache	**0x7002**	0x10	7.4.5
Retrieve PID Operating Parameters	**0x7003**	0x10	7.4.6
Float Reboot	**0xA000**	0x10	7.4.7

Table 7-1: Special Commands

7.4.1 Set WGMS ID (0x000F)

Available to Admins only.

Writes the unique WGMS ID into the C&C Non-Volatile Memory. This command is usually used only once in the Vehicle lifetime: When the vehicle is first brought online.

Note: This command can only be issued by via Iridium. It will only be enabled on the Main WGMS system and not be allowed on any secondary WGMS.

ACK – Structure type 0x1E containing the WGMS ID (8.2.9)

Field Name	Bytes	Value
Structure ID	1	**0x10**
Command Value	2	**0x000F**
WGMS ID	4	Ref. 6.1.8
(reserved)	1	0
Latitude	4	0 (not used)
Longitude	4	0 (not used)

7.4.2 Set Restricted Parameters (0x6000)

Some individual parameters can only be set by Admins and Masters.

ACK – No Payload (ref. 8.2.4)

Field Name	Bytes	Value [range]
Structure ID	1	**0x15**
Command Value	2	**0x6000**
Number of parameters	1	**N**
Parameter 1 ID	2	Ref Table below
Parameter 1 Length	1	[1 or 2]
Parameter 1 Data	Length of Param 1	
.....	...	...
.....	...	...
Parameter N ID	2	Ref Table below
Parameter N Length	1	[1 or 2]
Parameter N Data	Length of Param N	

Restricted parameter IDs and data that are currently supported:

Parameter	ID	Bytes	Type	Value (default) [range]	Notes
XBee PAN ID	**0x0016**	2	Numeric	(0x5000)	[1]
Default XBee Channel	**0x0017**	1	Numeric	(12) [12 to 23]	[2]
Enable PI Navigation	**0x0030**	1	Boolean	0 → Off, 1 → On	[3]
P Gain Value	**0x0031**	2	Numeric	(0) [0 to 32767]	[4]
I Gain Value	**0x0032**	2	Numeric	(0) [0 to 32767]	[5]
Integral Sat Limit	**0x0033**	2	Numeric	[0 to 90]	[6]

Notes:

[1] The "XBee PAN ID" is the user-defined Personal Area Network ID for the XBee system.

[2] The "Default XBee Channel" is user-defined.

[3] "Enable PI Navigation" enables/disables navigation using proportional-integral course correction.

[4] The "P Gain Value" is the proportional gain filter coefficient for PI navigation. The specified value is scaled internally to the range [0.0 to 0.5].

[5] The "I Gain Value" is the integral gain filter coefficient for PI navigation. The specified value is scaled internally to the range [0.0 to 0.5].

[6] The "Integral Sat Limit" the maximum allowed integral value for PI navigation.

7.4.3 Retrieve Debug Buffer (0x7000)

Requests retrieval of the Debug Buffer Packet.

ACK – Debug Buffer Packet (ref. 8.4.2)

Field Name	Bytes	Value
Structure ID	1	**0x10**
Command Value	2	**0x7000**
(reserved)	5	0
Latitude	4	0 (not used)
Longitude	4	0 (not used)

7.4.4 Retrieve Sub Heading Buffer (0x7001)

Requests retrieval of the Sub Heading Buffer.

ACK – Sub Heading Buffer (ref. 8.4.3)

Field Name	Bytes	Value
Structure ID	1	**0x10**
Command Value	2	**0x7001**
(reserved)	5	0
Latitude	4	0 (not used)
Longitude	4	0 (not used)

7.4.5 Retrieve PID Integral Cache (0x7002)

Requests retrieval of the PID Integral Cache.

ACK – PID Integral Cache (ref. 8.4.4)

Field Name	Bytes	Default Value
Structure ID	1	**0x10**
Command Value	2	**0x7002**
(reserved)	5	0
Latitude	4	0 (not used)
Longitude	4	0 (not used)

7.4.6 Retrieve PID Operating Parameters (0x7003)

Requests retrieval of the PID Operating Parameters.

ACK – PID Operating Parameters (8.4.5)

Field Name	Bytes	Default Value
Structure ID	1	**0x10**
Command Value	2	**0x7003**
(reserved)	5	0
Latitude	4	0 (not used)
Longitude	4	0 (not used)

7.4.7 Float Reboot (0xA000)

Available to Admins only.

Used to reset the Float C&C in case of a no-response, or during system debugging.

This command is not expected to be ACK'ed by the Vehicle and will be followed up by a Float Reboot alarm on the next telemetry transmission.

Field Name	Bytes	Value
Structure ID	1	**0x10**
Command Value	2	**0xA000**
(reserved)	5	0
Latitude	4	0 (not used)
Longitude	4	0 (not used)

This Page Intentionally Left Blank

8. Vehicle Responses

Most response messages are ACK/NAK data packets sent in response to Commands. There can also be unsolicited messages sent by the Vehicle (Ref. 8.4).

8.1 ACK/NAK Protocol

Commands sent to the C&C will be acknowledged with an ACK or NAK. The ACK/NAK response contains the command ID to allow the receiving system to track the command being acknowledged, even if a data transmission from the Vehicle is lost.

Responses are as follows:

- ACK signals successful packet reception and acceptance of its content.
- NAK signals successful packet reception but rejection of its content.
- No response (timeout) signals non-reception of a packet.

Note that the "Float Reboot" command is not expected to ACK or NAK.

8.1.1 ACK/NAK Packet Format

The ACK/NAK response packet format is shown in Figure 8-1.

Note that, unlike command packets, the Level 1 ACK/NAK message packet has no standard header structure. The different ACK/NAK "Type" values indicate the format of the Level 1 packet.

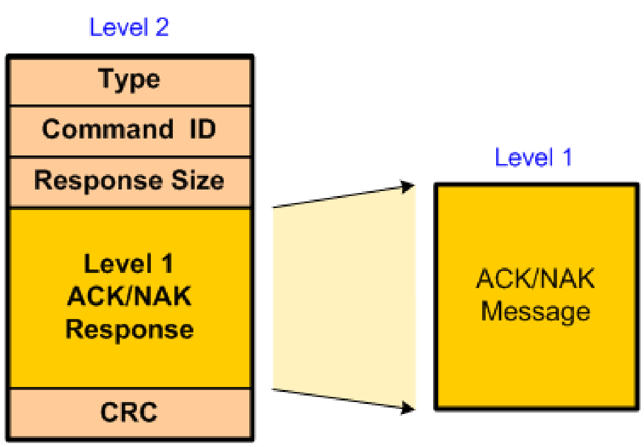


Figure 8-1: ACK/NAK Response Packet Format

where:

Type: [1 byte] ACK or NAK Data Type. See sections 8.1.2 and 8.1.3 below.

Command ID: [4 bytes] The ID value of the command being ACK'ed or NAK'ed (ref. 7.2).

Size: [2 bytes] Size of the ACK or NAK Message that follows. Size = 0 → No Data.

ACK/NAK Response: [max 196 bytes] Level 1 ACK or NAK data/message return, if any.

CRC: [2 bytes] CRC16 of all previous bytes in the Level 2 ACK/NAK packet.

8.1.2 ACK Data Type Numbers

Allowed Data Types for the **ACK** return are:

Type	ACK Return Payload	Ref.
0x10	Status With Waypoint Read-Back	8.2.1
0x11	Status Return	8.2.2
0x12	Identification Data	8.2.3
0x13	No Payload	8.2.4
0x14	Power Controller Status	8.2.5
0x15	Rudder Control Data	8.2.6
0x1B	Read File	8.2.7
0x1C	Detailed File List Data	8.2.8
0x1E	WGMS ID	8.2.9
0x1F	File at Destination	8.2.10
0x31	File List Data	8.2.11

8.1.3 NAK Data Type Numbers

Allowed data Types for the **NAK** return are:

Type	NAK Type	Ref.
0x20	(reserved)	
0x21	(reserved)	
0x22	Invalid Command	8.3.1
0x23	Invalid Data	8.3.2
0x24	File Error	8.3.3

8.1.4 Unsolicited Response Type Numbers

Allowed data Types for an unsolicited response are:

Type	Message Type	Ref.
0x60	New File Available	8.5.1
0x1D	Write Console Output to Client	8.5.2

8.1.5 Relay Messages

These are messages from a Relay to the WGMS.

Type	Message Type	Ref.
0x70	Relay Status	8.6.1

8.2 ACK Response Data

The formats of data returned in the ACK packets are described below.

8.2.1 ACK – Status with Waypoint Read-Back (0x10)

This type of Status packet is sent in response to the commands:

Set Waypoint (ref. 7.3.20)
Waypoint Read (ref. 7.3.21)

The format is identical to the regular "Status Return" (8.2.2), except for the last two fields (Waypoint Lat/Long).

Field Name	Bytes Used	Byte / Bit	Value [range] (units)	Default	Note
Follow Course	1	0 / 0	0 → not, 1→ Enabled	**1**	[1]
Manual Navigation		0 / 1	0 → not, 1→ Enabled	**0**	[1]
Follow Heading		0 / 2	0 → not, 1→ Enabled	**0**	[1]
Hold Station		0 / 3	0 → not, 1→ Enabled	**0**	[1]
(unused)		0 / 4	0		
XBee State		0 / 5	0 → Off, 1→ On		
Marker Light State		0 / 6	0 → Off, 1→ On		
IR Light State		0 / 7	0 → Off, 1→ On		
Telemetry Rate	2	1-2	[1 – 1440] (minutes)	**2**	
Low Battery Telemetry Rate	2	3-4	[1 – 1440] (minutes)	**20**	
Alarm Telemetry Rate	2	5-6	[1 – 30] (minutes)	**2**	
(reserved)	1	7	0		
Sub-to-Float Timeout	1	8	[20 – 200] (millisecs)	**20**	
(reserved)	2	9-10	0		
Float Low Battery	1	11	[1 – 65] (10 Wh)	**20**	
Float Critical Battery	1	12	[1 – 65] (10 Wh)	**3**	
(reserved)	3	13-15	0		
Float Pressure Threshold High	1	16	[0 – 255] (kPa)	**255**	
Float Pressure Threshold Low	1	17	[0 – 255] (kPa)	**115**	
Float Temperature Threshold	1	18	[0 – 255] (°C)	**70**	
Waypoint Arrival Distance	2	19-20	[5 – 5000] (meters)	**20**	
(reserved)	1	21	0		
Power Controller Telemetry Rate	2	22-23	[5 – 2880] (minutes)	**60**	
(reserved)	2	24-25	0		
Float-to-Sub Comm Period	1	26	[1 – 240] (seconds)	**15**	
Waypoint Latitude	4	27-30	[ref. 6.4.2]		
Waypoint Longitude	4	31-34	[ref. 6.4.2]		

Note:

[1] Only one of these bits will be enabled at a time.

8.2.2 ACK – Status Return (0x11)

This type of Status packet is sent in response to the commands:

Set Status (ref. 7.3.14)
Retrieve Status (ref. 7.3.13)
Hold Current Position (ref. 7.3.2)

Note that this packet contains read-only fields that are not used in the “Set Status” Command.

Field Name	Bytes Used	Byte / Bit	Value Range (units)	Default	Notes
Follow Course [1]	1	0 / 0	0 → not, 1→ Enabled	**1**	[1]
Manual Navigation [1]		0 / 1	0 → not, 1→ Enabled	**0**	[1]
Follow Heading [1]		0 / 2	0 → not, 1→ Enabled	**0**	[1]
Hold Station [1]		0 / 3	0 → not, 1→ Enabled	**0**	[1]
(unused)		0 / 4	0		
XBee State		0 / 5	0 → Off, 1→ On		
Marker Light State		0 / 6	0 → Off, 1→ On		
IR Light State		0 / 7	0 → Off, 1→ On		
Telemetry Rate	2	1	[1 – 1440] (minutes)	**2**	
Low Battery Telemetry Rate	2	3	[1 – 1440] (minutes)	**20**	
Alarm Telemetry Rate	2	5	[1 – 30] (minutes)	**2**	
(reserved)	1	7	0		
Sub Float Timeout	1	8	[20 – 200] (millisecs)	**20**	
(reserved)	2	9	0		
Float Low_Battery Power Level	1	11	[1 – 65] (10 Wh)	**20**	
Float Critical_Battery Power Level	1	12	[1 – 65] (10 Wh)	**3**	
(reserved)	3	13	0		
Float Pressure_Threshold High	1	16	[0 – 255] (kPa)	**255**	
Float Pressure_Threshold Low	1	17	[0 – 255] (kPa)	**115**	
Float Temperature Threshold	1	18	[0 – 255] (°C)	**70**	
Waypoint Arrival Distance	2	19	[5 – 5000] (meters)	**20**	
(reserved)	1	21	0		
Power Controller Telemetry Rate	2	22	[5 – 2880] (minutes)	**60**	
(reserved)	2	24	0		
Float-to-Sub Comm Period	1	26	[1 – 240] (seconds)	**15**	
(Hold Position Latitude)	4	27	0 [ref. 6.4.2]		[2]
(Hold Position Longitude)	4	31	0 [ref. 6.4.2]		[2]

Notes:

[1] Only one of these bits will be enabled at a time.

[2] Latitude and Longitude values are valid only when ACKing the "Hold Current Position" command. For the “Set Status” and “Retrieve Status” commands, the response returns a value of zero.

8.2.3 ACK – Vehicle Identification Data (0x12)

This response contains data on the Vehicle software and hardware. Sent in response to the command:

Retrieve Identification (ref. 7.3.15)

All Version numbers are stored as [Major] [Minor], using 2 bytes (1 byte each).

Identification Packet Field Name	Bytes Used	Byte #	Value [range] (units)
Hour	1	0	[0 – 23] (hours)
Minute	1	1	[0 – 59] (minutes)
Second	1	2	[0 – 59] (seconds)
Month	1	3	[1 – 12]
Day	1	4	[1 – 31]
Year	1	5	[0 – 99] (20xx)
Float_C&C Version	2	6-7	[major] [minor]
Float_C&C Build Number	2	8-9	[1 – 255]
Reported IMEI	15	10-24	[15 digits]
Sub Firmware Version	2	25-26	[major] [minor]
Sub Hardware Version	2	27-28	[major] [minor]
Sub Firmware CRC-32	4	29-32	
Sub EEPROM CRC-32	4	33-36	
Sensor Firmware Version	2	37-38	[major] [minor]
Sensor Hardware Version	2	39-40	[major] [minor]
Sensor Firmware CRC-32	4	41-44	
Sensor EEPROM CRC-32	4	45-48	
Bubble Firmware Version	2	49-50	[major] [minor]
Bubble Hardware Version	2	51-52	[major] [minor]
Bubble Firmware CRC-32	4	53-56	
Bubble EEPROM CRC-32	4	57-60	
Batteries Firmware Version	2	61-62	[major] [minor]
Batteries Hardware Version	2	63-64	[major] [minor]
Batteries Firmware CRC-32	4	65-68	
Batteries EEPROM CRC-32	4	69-72	
Untwist Firmware Version	2	73-74	[major] [minor]
Untwist Hardware Version	2	75-76	[major] [minor]
Untwist Firmware CRC-32	4	77-80	
Untwist EEPROM CRC-32	4	81-84	
COM Spec Version	2	85-86	[major] [minor]
COM Spec Revision	1	87	
Iridium Modem Revision	2	88-89	
XBee Channel	1	90	[12 to 23]
XBee PAN ID	2	91-92	
XBee Address	2	93-94	

Identification Packet Field Name	Bytes Used	Byte #	Value [range] (units)
(reserved)	4	95-101	
Count of Program Board Records	2	102-103	**N**
Record #1	17	104-120	(ref. below)
Record #2	17	121-137	(ref. below)
. . . . .	. . .	. . .	
. . . . .	. . .	. . .	
. . . . .	. . .	. . .	
Record #N	17		(ref. below)

Program Board Records have the following structure:

Field Name	Bytes	Value [range]	Note
Storage Controller Address	2		[1]
Board Address	2		[2]
Selected Program WGMS ID	4		[3]
Default Program WGMS ID	4		[4]
Currently Reported Program WGMS ID	4		[5]
Has Files	1	0x00 or 0xFF	[6]

Notes:

[1] The "Storage Controller Address" is the location where the program originates.

[2] The "Board Address" is the board where the program executes.

[3] The "Selected Program" is the designated standard program to run.

[4] The "Default Program" is the fall-back program.

[5] The "Currently Reported Program" is the one that last ran.

[6] Boolean value indicates whether any files are stored on the board.
0x00 → No
0xFF → Yes (board is a Storage Controller)

8.2.4 ACK – No Payload Response (0x13)

This ACK contains no payload data (Size = 0) and is sent in response to Vehicle action commands:

Follow Sequential Waypoint Course (ref. 7.3.4)
Follow Fixed Heading (ref. 7.3.6)
Follow Course and Loop (ref. 7.3.22)
Follow Course With Options (ref. 7.3.23)
Rudder Left/Right/Center (ref. 7.3.5)
Light, IR, XBee Off/On (ref. 7.3.10, 7.3.11, 7.3.12)
Initialize File Write (ref. 7.3.24)
Write File Block (ref. 7.3.25)
Delete File (ref. 7.3.26)
Select Program (ref. 7.3.29)
Write Console Input to Vehicle (ref. 7.3.30)
Read File Transaction Closed (ref. 7.3.31)

8.2.5 ACK – Power Controller Status (0x14)

Sent in response to the command: **Retrieve Power Status** (ref. 7.3.16)

Field Name	Bytes Used	Byte #	Value [Range] (units)
Hour	1	0	[0 – 23] (hours)
Minute	1	1	[0 – 59] (minutes)
Second	1	2	[0 – 59] (seconds)
Month	1	3	[1 – 12]
Day	1	4	[1 – 31]
Year	1	5	[20xx]
Payload_1 Current	1	6	(10 mA)
Payload_2 Current	1	7	(10 mA)
Umbilical Current	1	8	(10 mA)
Untwist Current	1	9	(10 mA)
Total Power Remaining	2	10-11	(10 mWh)
Battery Current	2	12-13	[signed] (10 mA)
System Current	2	14-15	(10 mA)
System Voltage	2	16-17	(1 mV)
Solar_Power_1 Indicator	1	18	[0 – 255] 0 → Off
Solar_Power_2 Indicator	1	19	[0 – 255] 0 → Off
Solar_Power_3 Indicator	1	20	[0 – 255] 0 → Off
Solar_Power_4 Indicator	1	21	[0 – 255] 0 → Off
Battery_0 Status	8	22-29	[ref. below]
Battery_1 Status	8	30-37	[ref. below]
Battery_2 Status	8	38-45	[ref. below]
Battery_3 Status	8	46-53	[ref. below]
Battery_4 Status	8	54-61	[ref. below]
Battery_5 Status	8	62-69	[ref. below]
Battery_6 Status	8	70-77	[ref. below]

Battery Status	Bytes	Bit	Value Range (units)
Battery Present	1	0	0→No 1→Yes
Battery Discharged		1	0→No 1→Yes
Battery Charged		2	0→No 1→Yes
Battery Discharging		3	0→No 1→Yes
Battery Terminate_Discharge		4	0→No 1→Yes
Battery Terminate_Charge		5	0→No 1→Yes
Battery Over_Temp		6	0→No 1→Yes
Battery Online		7	0→No 1→Yes
Battery Temp	1		[-10 to 100] (°C)
Battery Power_Remaining	2		(10 mWh)
Battery Current	2		(10 mA)
Battery Voltage	2		(1 mV)

8.2.6 ACK – Rudder Control Data (0x15)

Sent in response to the commands:

Set Rudder Control (ref. 7.3.18)
Retrieve Rudder Control (ref. 7.3.17)

There are three sets of parameters:

- "Normal" values are the default.
- "Low" values are used when the battery energy is low.
- "Critical" values are used when the batteries are very close to depletion.

See section 7.3.18 for a more detailed description of the Rudder Control parameters.

Field Name	Bytes Used	Byte #	Value (Units)
Normal Gain	2	0-1	(integer)
Normal Timebase	1	2	15Hz/Value
Normal Deadband	1	3	(0.5°)
Normal Goal	1	4	(0.5°)
Normal RudderMax	1	5	(0.5°)
Normal MoveMin	1	6	(0.5°)
Normal TrimMax	1	7	(0.5°)
Low Gain	2	8-9	(integer)
Low Timebase	1	10	15Hz/Value
Low Deadband	1	11	(0.5°)
Low Goal	1	12	(0.5°)
Low RudderMax	1	13	(0.5°)
Low MoveMin	1	14	(0.5°)
Low TrimMax	1	15	(0.5°)
Critical Gain	2	16-17	(integer)
Critical Timebase	1	18	15Hz/Value
Critical Deadband	1	19	(0.5°)
Critical Goal	1	20	(0.5°)
Critical RudderMax	1	21	(0.5°)
Critical MoveMin	1	22	(0.5°)
Critical TrimMax	1	23	(0.5°)
(unused)	4	24-27	
CRC32	4	28-31	

8.2.7 ACK – Read File Block (0x1B)

Sent in response to the command:

Read File Request (ref. 7.3.27)

If the file length exceeds the length of the data field, there may be multiple responses to a single "Read File Request" command.

Field Name	Bytes	Value [range]
Storage Controller Address	2	Matches command request
Offset (Seek)	4	Value specified in request
EOF Flag	1	0x00 = More responses will be sent 0xFF = End of file, no more responses
Number of Bytes in Data	2	**N** [≥ 1]
Data	1 to N	File data block

8.2.8 ACK – Detailed File List Data (0x1C)

Sent in response to the command:

Get Detailed File List (ref. 7.3.28).

Returns a list of file records. If the number of files to be reported exceeds the limit of this message, there will be multiple ACK responses.

Field Name	Bytes	Value	Note
Storage Controller Address	2	Matches command request	
Offset	2	Number of files previously reported	
Multiple-Report Flag	1	**0x00** = Additional packets coming **0xFF** = Last packet	
Number of Files in this Message	2	**N** [≥ 1]	[1]
Detailed File Record 1	68	See below	
Detailed File Record 2	68	See below	
. . . . .	. . .	. . . . .	
. . . . .	. . .	. . . . .	
Detailed File Record N	68		

Each of the Detailed File Records has the following format:

Field Name	Bytes	Field	Notes

Field Name	Bytes	Field	Notes
File Name	11	8+3 ASCII string. (ref. 6.4.9)	
File ID	4	WGMS-generated	
File Size	4	Bytes	
Creation Date	6	Date + Time	[2]
Modified Date	6	Date + Time	[2]
File Attributes	1	0x01 bit = 1 → read-only 0x80 bit = 1 → MD5 check was successful All other bits are user defined	
Provided CRC-16	2		[3]
Provided MD5	16		[3]
Calculated CRC-16	2		[4]
Calculated MD5	16		[4]

Notes:

[1] The number of File Records that can be returned in this message depends on the maximum message size allowed in the Level 3 packet.

[2] Dates use 1 byte per value in the order shown below. Value ranges are shown in [ ].
Hour [0 to 23]
Minute [0 to 59]
Second [0 to 59]
Month [1 to 12]
Day [1 to 31]
Year-2000 [0 to 99]

[3] The "Provided" CRC and MD5 are the values that were specified in the "Initialize File Write" command (ref. 7.3.24).

[4] The "Calculated" CRC and MD5 are the values calculated on a previous file access command. If they have not been calculated, the default value is 0.

8.2.9 ACK – WGMS ID (0x1E)

Sent in response to the command:

Read Vehicle WGMS ID (ref. 7.3.8)

Returns the unique pre-set value from the Vehicle's Non-Volatile Memory.

Field Name	Bytes	Value	Note
WGMS ID	4	Ref. 6.1.8	[1]

Note:

[1] If the ID value has not been set, returns: **0xFFFFFFFF**

8.2.10 ACK – File at Destination (0x1F)

This is an additional response to the commands:

Get File List (ref. 7.3.32)
Get Detailed File List (ref. 7.3.28)
Select Program (ref. 7.3.29)

where the file was checked and then forwarded to its final destination.

Field Name	Bytes	Value
Storage Controller Address	2	Matches command request
File WGMS ID	4	

8.2.11 ACK – File List Data (0x31)

This is the primary response to the command:

Get File List (ref. 7.3.32)

Returns a list of file records. If the number of files exceeds the limit of this message, there will be multiple ACK responses.

Field Name	Bytes	Value	Note
Storage Controller Address	2	Matches command request	
Offset	2	Number of files previously reported	
Multiple-Report Flag	1	**0x00** = Additional packets coming **0xFF** = Last packet	
Number of Files in this Message	2	**N** [≥ 1]	[1]
File Record 1	20	See below	
File Record 2	20	See below	
. . . . .	. . .	. . . . .	
. . . . .	. . .	. . . . .	
File Record N	20		

Each of the File Records has the following format:

Field Name	Bytes	Value
File Name	11	8+3 ASCII string. (ref. 6.4.9)
File WGMS ID	4	(ref. 6.1.8)
File Size	4	Bytes
File Attributes	1	0x01 bit = 1 → read-only 0x80 bit = 1 → MD5 check was successful All other bits are user defined

Note:

[1] The number of File Records that can be returned in this message depends on the maximum message size allowed in the Level 3 packet.

8.3 NAK Response Data

8.3.1 NAK – Invalid Command (0x22)

The command code number was not recognized.

This NAK contains no payload data (Size = 0).

8.3.2 NAK – Invalid Data (0x22)

The data in the command is not correct.

This NAK contains no payload data (Size = 0).

8.3.3 NAK – File Error (0x24)

Response given for a file error in the commands:

Initialize File Write (ref. 7.3.24)
Write File Block (ref. 7.3.25)
Delete File (ref. 7.3.26)
Read File Request (ref. 7.3.27)
Get Detailed File List (ref. 7.3.28)
Select Program (ref. 7.3.29)
Get File List (ref. 7.3.32)

The NAK message returns one or more disposition codes indicating the type of error.

Field Name	Bytes	Value [range]	Notes
Count of Dispositions	2	**N** [≥ 1]	
Disposition 1	1	[3 to 17] (ref. below)	[1]
. . . . .	. . .	. . . . .	
. . . . .	. . .	. . . . .	
Disposition N	1	[3 to 17]	

Note:

[1] Disposition values are:

3 = Some items failed, others succeeded
5 = Bad data
7 = File exists but is marked for no-change
8 = File would be overwritten
9 = File is read-only
10 = Missing storage device
11 = Failed CRC-16 check. File was deleted if requested.
12 = Failed MD5 check. File was deleted if requested.
13 = Storage full
14 = Storage I/O error
15 = File control structures corrupted
16 = Program selection configuration on Vehicle is corrupt
17 = The number of active file commands exceeds handling capability

8.4 Restricted ACK Responses

Certain ACK responses are only available to Admin and Master Roles.

8.4.1 Special ACK Data Types

Allowed Data Types for the **ACK** return are:

Type	ACK Return Payload	Ref.
0x16	Debug Buffer	8.4.2
0x17	Sub Heading Buffer	8.4.3
0x18	PID Integral Cache	8.4.4
0x19	PID Operating Parameters	8.4.5

8.4.2 ACK – Debug Buffer (0x16)

The Debug Buffer response is given in response to the **Retrieve Debug Buffer** command. This is only for debugging use by System Admins and developers.

8.4.3 ACK – Sub Heading Buffer (0x17)

This response is given in response to the **Sub Heading Buffer** command. This is only for development use by System Admins and developers.

8.4.4 ACK – PID Integral Cache (0x18)

This response is given in response to the **PID Integral Cache** command. This is only for development use by System Admins and developers.

8.4.5 ACK – PID Operating Parameters (0x19)

This response is given in response to the **PID Operating Parameters** command. This is only for development use by System Admins and developers.

8.5 Unsolicited Messages

These Messages can be sent by the Vehicle without being prompted by a Command.

8.5.1 New File Available (0x60)

The content has following format:

Field Name	Bytes	Value	Note
Storage Controller Address	2		[1]
File Name	11	8+3 ASCII string. (ref. 6.4.9)	
File Size	4	Bytes	
Creation Date	6	Date + Time	[2]
Modified Date	6	Date + Time	[2]
File Attributes	1	0x01 bit = 1 → read-only 0x80 bit = 1 → MD5 check was successful All other bits are user-defined	
Provided CRC-16	2		
Provided MD5	16		
Calculated CRC-16	2		[3]
Calculated MD5	16		[3]

Notes:

[1] The "Storage Controller Address" is the 1-byte Board Address, and the 1-byte Task ID (ref. 5.3.5). The Float C&C has the reserved address value: **0x0000**

[2] Dates use 1 byte per value in the order shown below. Value ranges are shown in [ ].

Hour	[0 to 23]
Minute	[0 to 59]
Second	[0 to 59]
Month	[1 to 12]
Day	[1 to 31]
Year-2000	[0 to 99]

[3] The "Calculated" CRC and MD5 are the values calculated on a previous file access command. If they have not been calculated, the default value is 0.

8.5.2 Send Console Output to Client (0x1D)

Sends a stream of ASCII data from the Vehicle to a Console (WGMS user).

Field Name	Bytes	Value [range]	Notes
Source Address	2	[1 to 256] + [1 to 256]	[1]
Function	1	Program-defined	[2]
Number of Bytes in Data	2	**N** (≥1)	
Data	1 to N	ASCII data being sent to the Console	

Notes:

[1] The "Source Address" is the address of the board sending the data. Float C&C = **0x0000**.

[2] The "Function" is a value defined by the Vehicle program. It can be used by the Client.

8.6 Relay-Only Message

Certain commands can only be issued by a Relay to WGMS. These typically are instructions to the WGMS, sent when the Vehicle connects to the Relay.

8.6.1 Relay Status (0x70)

This is sent by a Relay to the WGMS. It provides the characteristics and the operating status of the Relay.

Field Name	Bytes	Byte #	Bit #	Value (default)	Notes
Channel State	1	0	0	0 or 1	[1]
Message Not Sent Alarm			1	0 or 1	[2]
Communications Error Alarm			2	0 or 1	[3]
(reserved)			3	0	
(reserved)			4	0	
(reserved)			5	0	
(reserved)			6	0	
(reserved)			7	0	
Channel Revision major	1	1			
Channel Revision Minor	1	2			
Channel Revision Sub-Minor	1	3			
Date	6	4 - 9			[4]

Notes:

[1] The "Channel State" values indicate the following:
  0 → Channel not active
  1 → Channel active

[2] The "Message Not Sent Alarm" indicates:
  Relay tried to forward message to the Vehicle, but the file was not accepted.

[3] The "Communications Error Alarm" indicates:
  The Relay was getting errors while trying to communicate with the Vehicle.

[4] Dates use 1 byte per value in the order shown below. Value ranges are shown in [ ].

Hour	[0 to 23]
Minute	[0 to 59]
Second	[0 to 59]
Month	[1 to 12]
Day	[1 to 31]
Year-2000	[0 to 99]

This Page Intentionally Left Blank

9. Vehicle Telemetry

Telemetry data packets are automatically sent by the C&C at the rates defined by the **Set Status** command (ref. 7.3.14). Note that the "Standard" Telemetry rate can differ from the rate for the **Power Controller Status** command (ref. 7.3.16).

Telemetry can be sent to an off-Vehicle location (via Iridium/Relay or XBee), or it can be sent to a payload (ref. Chapter 10).

9.1 Telemetry Packet Format

The Level 2 Telemetry packet format is similar to the ACK/NAK packet, but with different data Type values. Also, the Level 1 data has no heading structure.

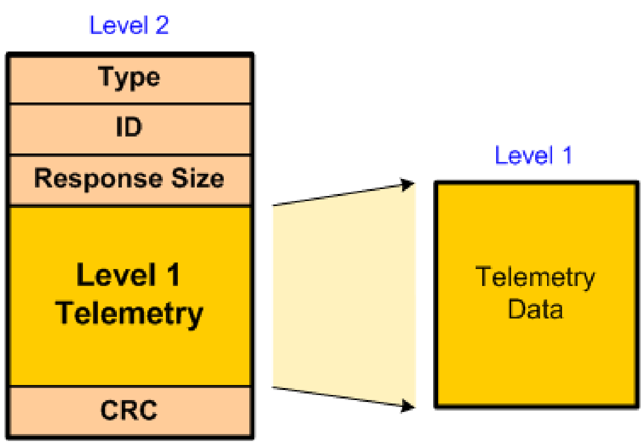


Figure 9-1: Level 2 Telemetry Data Packet

where:

Type: [1 byte] Telemetry Type. See 9.1.1 below.

ID: [4 bytes] This field is not used and is set to **0x00000000**

Size: [2 bytes] The size of the data that follows.

Telemetry Data: [≤ 196 bytes] Telemetry Level 1 data payload.

CRC: [2 bytes] The CRC-16 of all bytes in the data packet (except the two CRC bytes).

9.1.1 Telemetry Types

Currently-defined Telemetry Types are:

Type	Description	Ref.
0x03	NAL Tracker Format	9.2
0x06	Standard Telemetry	9.3
0x07	Power Controller Status	9.4
0x80	Payload Telemetry	9.5
0x81	Multi-Packet Telemetry	TBD

9.1.2 Telemetry Buffering

As with ACK/NAK responses, Iridium service may occasionally be interrupted for up to several minutes. In this case, the C&C electronics will buffer any telemetry, and then send out buffered messages when service is resumed.

Since the message buffer is 16 responses deep, once the 16-message limit is reached, any additional messages will be added to the buffer, and the oldest message discarded. Thus, for very long Iridium disruptions, some data may be lost.

9.2 Telemetry – NAL Tracker Format

The NAL tracker is a commercially-available device used to log time/position data, which can then be sent out as Level 1 data. The Type 0x03 NAL tracker data format is:

NAL Tracker Telemetry Field Name	Bytes Used	Byte #	Value [range]	Note
Decade + Hour	1	0	Decade after 2000 | 24 Hour Time	[1]
Minute	1	1	Time Minutes [0 to 59]	
Second	1	2	Time Integral Seconds	[2]
Decimal Seconds	1	3	0.## [0 to 100]	
Year + Month	1	4	Month | Year after decade	[3]
Day	1	5	[0 to 31]	
Latitude	1	6	Signed Integer Degrees [-90 to 90]	
Latitude	1	7	Integer Minutes [0 to 59]	
Longitude	1	8	Signed Integer Degrees [-180 to 180]	
Longitude	1	9	Integer Minutes [0 to 59]	
Latitude	1	10	Fraction Minutes 0.## [0 to 100]	
Latitude	1	11	Fraction Minutes 0.00## [0 to 100]	
Longitude	1	12	Fraction Minutes 0.#	[4]
Longitude	1	13	Fraction Minutes 0.0## [0 to 100]	
Longitude	1	14	Fraction Minutes 0.00## [0 to 100]	
(not used)	14	15-28		
Configuration Byte	1	29		

Notes:

[1] The Decade (after 2000) and the Hour are encoded into a single Byte as follows:
Decade = Byte modulo 10
Hour = Byte / 10

[2] Integral Time Seconds area encoded into a single Byte as follows:
Seconds = Byte modulo 0x40

[3] The Year and Month are encoded into a single Byte as follows:
Year = Byte modulo 10
Month = Byte / 10

[4] The decimal fraction of longitude is encoded as follows:
0.# = Byte modulo 10

9.3 Standard Telemetry Data

This is the standard telemetry package sent at the defined intervals.

The Type 0x06 Telemetry format is:

Standard Telemetry Field Name	Bytes Used	Byte #	Value [Range] (units)	Note
Alarm Bytes	2	0-1	[ref. below]	
Surface Temperature	2	2-3	(°C)	
Latitude	4	4-7	[ref. 6.4.2]	
Longitude	4	8-11	[ref. 6.4.2]	
Last_Fix_Time Hours	1	12	[0 – 23] (hours)	[1]
Last_Fix_Time Minutes	1	13	[0 – 59] (minutes)	[1]
Last_Fix_Time Seconds	1	14	[0 – 59] (seconds)	[1]
Last_Fix_Date Month	1	15	[1 – 12]	[1]
Last_Fix_Date Day	1	16	[1 – 31]	[1]
Last_Fix_Date Year	1	17	[0 – 99] (20xx)	[1]
Unused	1	18		
Temperature Sub	2	19-20	(°C)	
Distance Over Ground	2	21-22	(meters)	[2]
Rudder Count	4	23-26	(counts)	[3]
Heading of Sub	2	27-28	[0 to 359] (degrees)	
Iridium Signal Strength	1	29	[0 to 5]	
Target Waypoint	1	30	[1 to 254]	
Desired Bearing	2	31-32	[0 to 359] (degrees)	
Heading Float	2	33-34	[0 to 359] (degrees)	
Pressure_Sensor Float	1	35	(kPa)	
Pressure_Sensor Sub	1	36	(kPa)	
Navigation Mode	1	37	[ref. 6.5]	
Float Status Byte	1	38	[ref. below]	
Reset Flags	1	39		[4]
Total Power Remaining	2	40-41	(10 mWh)	
Iridium Download Errors	1	42	(counts)	[5]
Iridium Upload Errors	1	43	(counts)	[5]
Iridium Contact Failures	1	44	(counts)	[5]
(reserved)	10	45-54		

Notes:

[1] Last_Fix is the time when a valid GPS signal was last received.

[2] Distance covered since last report.

[3] The number of times the rudder has moved since the last re-boot.

[4] Count of C&C CPU resets.

[5] Number of errors since re-boot.

9.3.1 Alarm Bytes

For the Alarm bytes 0 & 1, the bit values indicate the following:

0 → OK
1 → Alarm Active

Alarm Type	Byte	Bit
Sub Leak Alarm	0	0
Float Leak Alarm		1
Sub-to-Float Comm Alarm		2
Payload Error Condition Alarm		3
Float Battery Low Alarm		4
Umbilical Fault Alarm		5
Sub Pressure Threshold Alarm		6
GPS Not Functioning		7
Unused	1	0
Sub Reboot Alarm		1
Float-to-Sub Comm Error		2
Unused		3
Float Temperature Threshold Exceeded		4
Float Pressure Threshold Exceeded		5
Sub Temperature Threshold Exceeded		6
Float Reboot Alarm		7

9.3.2 Float Status Byte

For the Float Status byte 38, the bit values indicate the following:

0 → Off
1 → On

Status Type	Byte	Bit
Light State	38	0
IR State		1
XBee State		2
Payload 1 Power		3
Payload 2 Power		4
Unused		5
Unused		6
Unused		7

9.4 Telemetry – Power Controller Status

A condensed format for the Power Controller Status is shown in the table below. The complete description is given in section 8.2.5 "ACK – Power Controller Status (0x14)".

The Type 0x07 Telemetry format is:

Power Controller Data Type	Bytes Used	Byte #
Hours/Minutes/Seconds	3	0-2
Day/Month/Year	3	3-6
Payload 1 & 2 Currents	2	6-7
Umbilical Current	1	8
Untwist Current	1	9
Total Power Remaining	2	10-11
Battery Current	2	12-13
System Current	2	14-15
System Voltage	2	16-17
Solar Power 1/2/3/4 Indicators	4	18-21
Battery_0 Status	8	22-29
Battery_1 Status	8	30-37
Battery_2 Status	8	38-45
Battery_3 Status	8	46-53
Battery_4 Status	8	54-61
Battery_5 Status	8	62-69
Battery_6 Status	8	70-77

9.5 Payload Telemetry

This telemetry type sends data packets that originate from the Vehicle payload section. The Payload Packet data is given a type-indicating header by the C&C, then sent to Shore using the Iridium modem. There is no checking or processing of the payload data by the C&C.

Field Name	Bytes
Packet Type Value	4
Packet Data	[variable]

The first four bytes of the packet indicate the data type. The Payload Telemetry types are:

Packet Type	Value	Ref.
Weather Station	1	9.5.1
Payload Status	2	9.5.2
Payload AISyG Status	16	9.5.3
Payload AISyG Static Record	17	9.5.4
Payload AISyG Position Report	18	9.5.5

9.5.1 Weather Station Packet (Type 1)

A Type 1 Weather Station data packet has 35 bytes and contains the following data.

Field Name	Bytes Used	Byte #	Value (units)	Notes
Weather Data Version	2	0-1	(internal)	
Flags	2	2-3	(internal)	
Temperature	2	4-5	°C / 10	
Pressure	2	6-7	(millibars)	
Latitude	4	8-11	(± minutes / 10,000)	[1]
Longitude	4	12-15	(± minutes / 10,000)	[2]
Avg. Wind Speed	2	16-17	(knots / 10)	
Max Wind Speed	2	18-19	(knots / 10)	
Std. Dev. Wind Speed	2	20-21	(knots / 10)	
Avg. Wind Direction	2	22-23	(degrees / 10)	
Std. Dev. Wind Dir	2	24-25	(degrees / 10)	
Number of Wind Samples	2	26-27	(count)	
Hour	1	28	[0 – 23] (UTC)	
Minute	1	29	[0 – 59]	
Second	1	30	[0 – 59]	
Day	1	31	[1 – 31]	
Month	1	32	[1 – 12]	
Year	2	33-34	[2000 – 2099]	

Notes:

[1] Latitude: (+) → North; (–) → South

[2] Longitude: (+) → East; (–) → West

9.5.2 Payload Status Packet (Type 2)

A Type 2 Payload Status packet has 16 bytes and contains the following data.

Field Name	Bytes Used	Byte #	Bit #	Value (units)
Hour	1	0		
Minute	1	1		
Second	1	2		
Month	1	3		
Day	1	4		
Year	1	5		
Version	2	6		
Alarm0 – Leak1	1	8	0	**0** → OK, **1** → Alarm
Alarm1 – Leak2			1	**0** → OK, **1** → Alarm
Alarm2 – Pressure Low			2	**0** → OK, **1** → Alarm
Alarm3 – Pressure High			3	**0** → OK, **1** → Alarm
Alarm4 – Humidity			4	**0** → OK, **1** → Alarm
Alarm5 – Temp Humid			5	**0** → OK, **1** → Alarm
Alarm6 – Temp Pressure			6	**0** → OK, **1** → Alarm
Alarm7			7	
Alarm8	1	9	0	
Alarm9			1	
Alarm10			2	
Alarm11			3	
Alarm12			4	
Alarm13			5	
Alarm14			6	
Alarm15			7	
Leak1	1	10		
Leak2	1	11		
HumidTemp	1	12		(°C)
Humidity	1	13		(%)
PressureTemp	1	14		(°C)
Pressure	1	15		(kPa)

9.5.3 Payload AISyG Status Packet (Type 16)

A Payload AISyG Status packet has 16 bytes and contains the following data:

Field Name	Bytes Used	Byte #	Value (units)
Hour	1	0	
Minute	1	1	
Second	1	2	
Month	1	3	
Day	1	4	
Year	1	5	
Latitude	4	6	
Longitude	4	10	
Temperature	2	14	
Pressure	2	16	
Humidity	2	18	
Alarms	2	20	(TBD)

9.5.4 Payload AISyG Static Record (Type 17)

A Payload AISyG Static Record packet has 86 bytes and contains the following data:

Field Name	Bytes Used	Byte #	Default
Hour	1	0	
Minute	1	1	
Second	1	2	
Month	1	3	
Day	1	4	
Year	1	5	
MMSI	4	6	
AISClass	1	10	A/B
Ship Name (20 characters)	20	11	
Call Sign (7 characters)	7	31	
Ship and Cargo Type	1	38	
Device Used for the Position Fixes	1	39	
Destination (20 characters)	20	40	
ETA (Month, Day, Hour, Minutes)	4	60	
Dimension and Point Of Reference	4	64	
IMO	4	68	
Latitude	4	72	
Longitude	4	76	
Course Over Ground	2	78	
Speed Over Ground	2	80	
Rate Of Turn	1	82	
Heading	2	83	
Navigation Status	1	85	

9.5.5 Payload AISyG Position Report (Type 18)

A Payload AISyG Static Record packet has a variable number of bytes depending upon the number of records contained. It has the following format:

Field Name	Bytes Used	Byte #	Value
Number Of Records in Message	1	0	**N**
Record 1	26	1	(ref below)
Record 2	26	27	(ref below)
. . .	. . .	. . .	
. . .	. . .	. . .	
Record N	26		(ref below)

Each 26-byte Record has the following format:

Field Name	Bytes Used	Byte #	Default
Hour	1	0	
Minute	1	1	
Second	1	2	
Month	1	3	
Day	1	4	
Year	1	5	
MMSI	4	6	
Latitude	4	10	
Longitude	4	14	
Course Over Ground	2	18	
Speed Over Ground	2	20	
Rate Of Turn	1	22	
Heading	2	23	
Navigation Status	1	25	

10. Payload Communications

Customers using application-specific electronics on board a Wave Glider need to be able to interact with other resources on the Vehicle as well as with the shore-side WGMS. LRI has developed a common generic communications protocol, described below. Using this protocol allows LRI and its customers to work independently on a project, yet retain communication interoperability.

10.1 Communication Hardware

Payload communications hardware can be both of LRI-supplied and customer-supplied or created electronic payload boards and devices.

10.1.1 Peripheral Interface Board

The Payload Interface Board ("PIB") is a standard LRI-designed circuit board designed to be mounted within a payload box. The PIB is a basic component of payload communications.

The Float's C&C computer is connected to the payloads via serial ports. Within each payload is a PIB that manages communications between the payload and the C&C. The PIBs in turn connect to other LRI or customer-designed payload boards and/or devices.

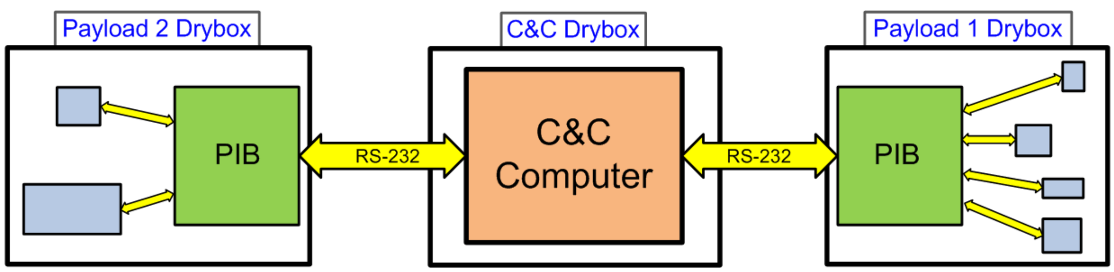


Figure 10-1: Basic Payload Communications

The PIB board itself does not connect directly with the additional payload devices. It communicates via "Personality Modules", as described below.

10.1.2 Personality Module

Each PIB board is designed to have to up to three daughter boards mounted to it, called "Personality Modules" ("PMs"). The PMs connect to different devices, including other PIBs.

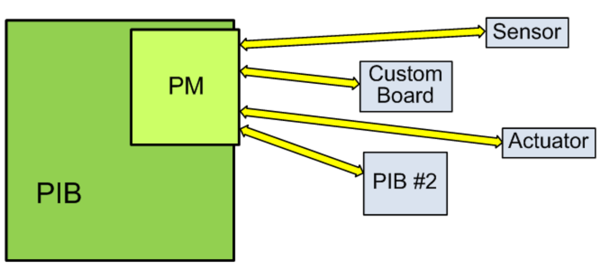


Figure 10-2: Personality Module Connections

Communications between a PM and the other payload devices is typically via RS-232, but is not necessarily restricted to that communication method.

LRI can provide a standard PM called the Power/Comm ("P/COM") board. The P/COM can provide both power and communications (RS-232, RS-422 or RS-485) to connected devices.

Customers can create their own custom PMs. Hardware descriptions of the PIB and P/COM are provided in the LRI *Payload Electronics Guide*.

10.1.3 Payload Board Devices

Application-specific payload board devices can take many forms. Some possible types of these devices are:

- Standard LRI-supplied PIBs
- Sensors
- Actuators
- Storage devices (e.g., Flash memory)
- External communications devices
- Processing nodes

Since a PM can be connected to additional PIBs, it is possible to daisy-chain PIBs, as shown in Figure 10-3 below.

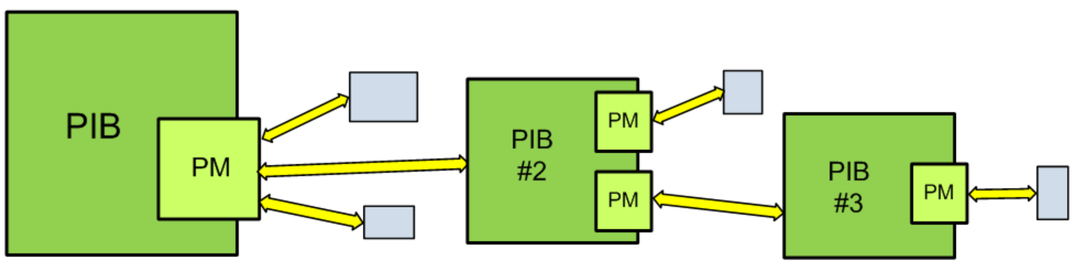


Figure 10-3: Daisy-Chain of PIBs

10.1.4 Configuration Scenarios

Some possible payload configuration scenarios are:

- Payload boards and sensors connected to PIBs via P/COMs.
- PIB acting as a Relay
- PIB connected to another PIB as a daisy chain back to the Float C&C PIB port.
- PIB running custom software and with custom PMs and acting as an end device (the custom PMs are the "payload boards".)
- Payload board connected directly to the Float C&C (which essentially acts as the "PIB")

Figure 10-4 below shows a combination of the above-described scenarios. Note that the Float C&C can be connected via radio links to WGMS. Thus, the Iridium and XBee links are considered part of the "configuration".

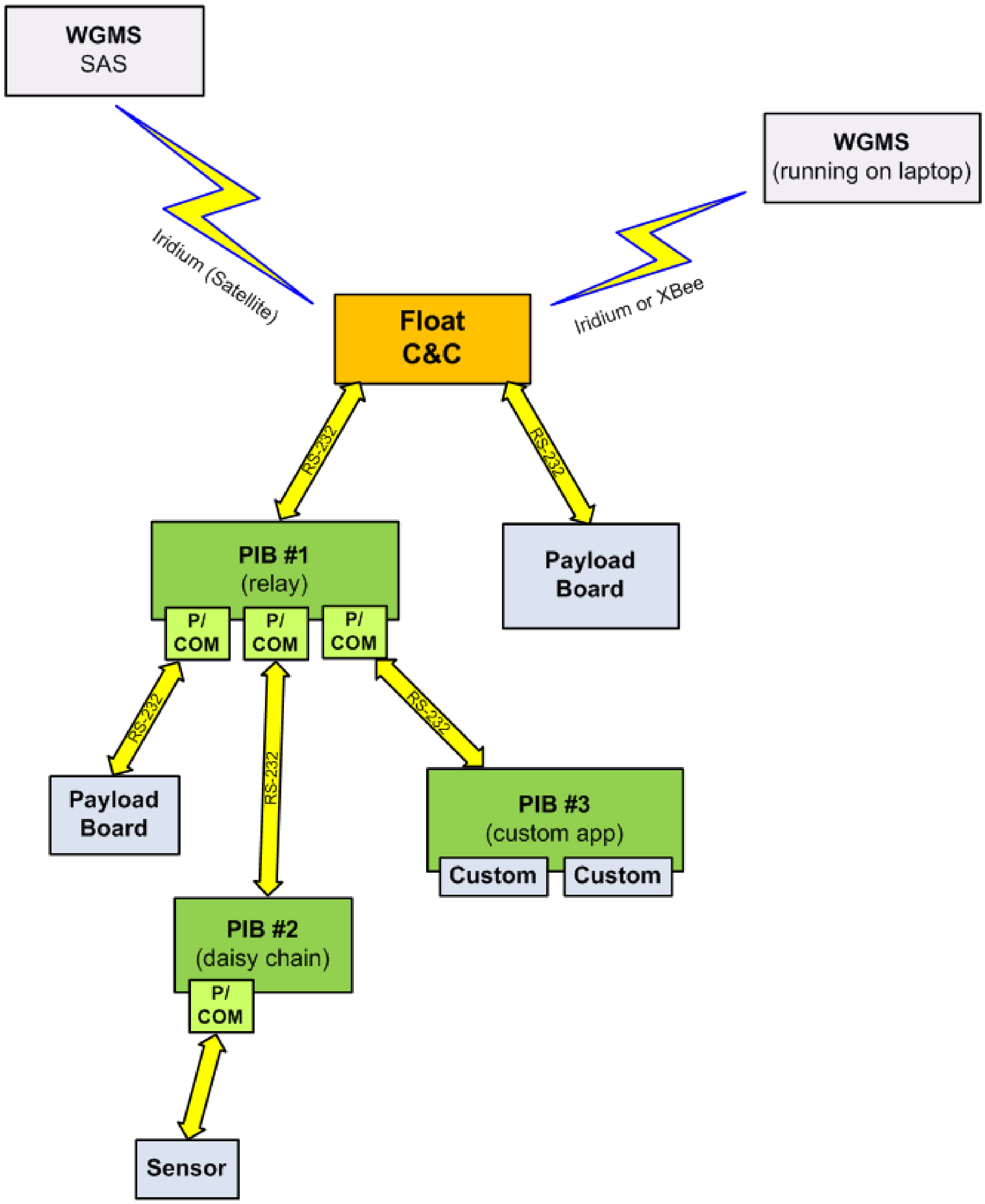


Figure 10-4: Example Payload Hardware Configuration

The above configuration shows most communications below the Float C&C as using RS-232. Although this is recommended, it is not necessarily required if using a custom PM or a direct electrical connection to a P/COM.

10.2 Communication Method

Payload messaging is based upon serial communications and data packets, using polling and message queues, as described below.

10.2.1 Payload Communications Requirements

Given the types of devices described above, the communications protocol has to allow the C&C, a PIB, or other device to:

- Identify itself and its software version number.
- Be updated with current location according to GPS.
- Subscribe to various messages that are sent to the shore.
- Issue commands directly to the Float C&C.
- Be updated with a new program, if the application supports it.
- Receive a file from WGMS or send a requested file to WGMS.
- Indicate when there is a new file available to be read.
- Read or store data in the Float C&C file system.
- Be notified when it is about to be powered off.
- Request a power cycle.
- Interact with a Debug Console.

10.2.2 Serial Communication Parameters

Communications between the C&C and the connected PIBs are:

- 115,200 bps
- 8 data bits
- 1 stop bit
- No parity

Communication is throttled using the protocol, so handshaking features like X-On/X-Off, RTS/CTS and DTR/DSR are disabled.

Communications between a PIB and its connected devices can use other parameters, depending upon the kind of PM used. However, using the above values is recommended.

10.2.3 Payload Data Packet Format

Payload data packets contain header data, followed by the message, ending with a CRC error check code. A special 1-byte Start-of-Frame (SOF) character is used to indicate the beginning of each packet ("frame").

The general format is:

Figure 10-5: Payload Data Packet Format

where:

SOF Character: Indicates the start of a payload message packet.

Packet Size: Number of bytes in the packet, excluding the SOF character.

Destination Address: Where the packet is to be sent.

Originator Address: Who sent the packet.

Tracking ID: Number created by the originator for message tracking.

Message Type: A code value indicating the type/format of data in the message.

Message Data: The actual message.

Error Check: A CRC code value.

10.2.4 Message Routing

A message packet contains the addresses of the originator and the recipient. When the C&C or a PIB creates a message packet, it can choose to broadcast it down to all connected child devices (PIBs, payload boards or other apps), or to send it only to a specific device. Depending upon the message, that PIB can process the message or route the message down to any of its child devices.

Note that a child device cannot send an unsolicited message to its parent device. Thus, a PIB cannot originate a message to the C&C. Likewise, a daisy-chained PIB cannot originate a message to its parent PIB.

When a child device has a packet to send to its parent, it puts in a message queue. The child then has to wait for the parent to poll it. When polled, the PIB responds with a packet that contains the message. This is described in more detail below.

10.2.5 Polling

Certain types of messages also have a polling function in addition to any data they may contain.

The C&C sends out periodic polling messages to the connected PIBs. The PIBs can respond with a "message waiting". If a PIB has a message waiting, the C&C requests the PIB to send it. The PIB then sends up the message. When the C&C receives the message, it then routes it to the designated recipient, which can be the C&C itself, another connected PIB, or an external communications device such as Iridium or XBee.

Likewise, a PIB can optionally send out periodic polling messages to its child devices. Each child device can respond with a "message waiting". The parent PIB then requests and receives the message, which it then process, route to another child device, or put into its message queue, waiting to be polled.

The polling rates are set by each PIB as part of the Enumeration Response, described below.

10.2.6 Enumeration Request/Response

At power-up, a specific "Enumeration" message is broadcast by the C&C. In response, each PIB reports its addresses, along with flags indicating the type of information it would like to receive (including the polling rate). Each PIB then broadcasts its own message to connected child devices, which respond with their own message requirements.

These addresses and flags (along with their physical connectivity) form the basis of the routing tables maintained by the C&C and PIBs.

10.2.7 ACKs and NAKs

When the C&C or a PIB receives a message packet, it typically responds to the sender with an ACK or NAK message packet.

An ACK ("Acknowledge") message indicates that the message was received and was able to be interpreted. When a PIB sends an ACK response to the C&C or parent PIB, it can also indicate that it has a message queued for sending.

A NAK ("Not Acknowledge") message indicates that there was a problem with the message. As with the ACK, when a PIB can use a NAK response to indicate that it has a message queued.

The original message sender has the option of notifying the recipient (via a flag bit) that it doesn't want an ACK/NAK response.

10.3 Protocol Message Format

Figure 10-6 below shows the format of the packet with a message containing “N” bytes.
Note that all fields are sent in “little endian” form (LSB first).

Field	Bytes	Format	Ref.

Start-of-Frame	1	**0x7E**	10.3.1
Message Size	2	[LSB, MSB]	10.3.2
Destination Address	2	[LSB, MSB]	10.3.3
Source Address	2	[LSB, MSB]	10.3.4
Transaction ID	2	[LSB, MSB]	10.3.5
Message Type	2	[LSB, MSB]	10.3.6
Message Data	N	**≤ 544** Bytes	10.3.7
CRC-16	2	[LSB, MSB]	10.3.8

Message Size Includes these bytes (Message Size through CRC-16)

Used to Calculate CRC16 (Message Size through Message Data)

Figure 10-6: Message Packet Format

Each of the message fields is described below.

10.3.1 Start-Of-Frame Character

The first character in a Message Packet must be the Start-of-Frame (SOF) character: **0x7E**

The SOF character can be escaped to allow it to be embedded within the remainder of the message packet (ref. 10.3.9).

10.3.2 Message Size

The number of bytes following the Start-of-Frame, including the CRC bytes. This must be **≤ 555**. However, the limit does not include any added Escape bytes (ref. 10.3.9).

10.3.3 Destination Address

The address of intended recipient of the message. There is also a special reserved address for broadcast messages. Ref. 10.4 below.

10.3.4 Source Address

The address of the message sender. Ref. 10.4 below. This allows the recipient to know who sent the message (especially in case of an ACK or NAK).

10.3.5 Transaction ID

A 2-byte number. This is an "arbitrary" number that is created by the sender. The value can be zero, but it is recommended that each message have a uniquely identifiable Transaction ID.

ACK and NAK messages do not assign Transaction IDs. They simply return the ID of the original message sender.

Transaction ID numbers are useful when a PIB sends out multiple messages. Each message should have a different Transaction ID. The response to each message will also contain the Transaction ID. This allows the sender to keep track of which messages were ACK'ed or NAK'ed.

A device may receive a message from two different sources, but with the same Transaction ID. There should not be any problem with this, as the recipient should consider the address of the sender, not the Transaction ID, when responding to the message.

Except for ACK and NAK packets, consecutive valid received messages with the same Transaction ID, Source, Size and CRC should be considered to be duplicate messages. In this case, the message recipient should simply send back duplicate responses. However, if a duplicated message requests an action, the recipient should not perform the requested action more than once.

10.3.6 Message Type

A 2-byte code that indicates the purpose and contents of this message (ref. section 10.6).

If the high bit of the MSB of the Message Type value is set to 1, it indicates that the sender doesn't want an ACK or NAK response.

10.3.7 Message Data

The message content to be forwarded. Some packets can contain no Message Data.

Normally, a maximum of **544** bytes is allowed, but characteristics of the recipient may impose additional length restrictions. For example, data intended to be sent out in the Iridium Short-Burst-Data format is limited to **192** bytes.

10.3.8 CRC-16

This error-check field contains a 16-bit CRC of the message starting at the byte following the **SOF** until the byte just before the CRC field (ref. Figure 10-6).

10.3.9 SOF and ESC Characters

The Start-of-Frame (SOF) character is: **0x7E**.

The Escape (ESC) character is: **0x7D**.

When either an SOF or ESC character is used within the packet (including the CRC), it must be preceded by the ESC character and also have its bit 5 toggled. Thus, for all bytes following the SOF:

0x7E → 0x7D,0x5E

0x7D → 0x7D,0x5D

The size values/limits for the messages exclude the count of any Escape characters.

10.4 Addresses

Addresses are 2-byte values used to route messages between the C&C, PIBs and other devices. These addresses are sometimes used within the body of a message, particularly when responding to broadcast messages.

10.4.1 Address Format

Source and Destination addresses are composed of a two-byte value. The MSB is the “Board ID” value. The LSB is the “Task ID”. These are both described in more detail below.

Note that, since the address is stored in little-endian format, the actual order of storage for the address will be:

LSB	MSB
Task ID	**Board ID**

10.4.2 Board ID

The MSB of the address contains the “Board ID”, which has defined values as follows:

Board ID	Description
0x00	Reserved ID of the Float C&C controller
0xFF	Broadcasts the message to all boards; Some scenarios do not support broadcasts
Other	An 8-bit ID that was configured to uniquely identify a device in this Vehicle

PIBs contain a Board ID that is embedded in Non-Volatile Memory on the board. Custom payload boards can assign their own Task IDs. However, the Board ID should not be duplicated by any device on any single Wave Glider Vehicle.

10.4.3 Task ID

The LSB of the address contains the Task ID, which is defined by the individual device. PIBs and other non-C&C devices can select their own addresses.

The following Task IDs are pre-defined for the C&C:

Task ID	Description	Notes
0x00	Main process running on the C&C	
0x01	The command interface on the Float C&C	[1]
0x02	Iridium modem (send Level 2 message)	
0x03	Debug Console interface	[2]
0x04	XBee interface (send Level 2 message)	
0x05	File transfer to shore.	[3]
0x06	Default Host (send Level 2 message)	

Notes:

[1] Level 2 messages (ref. 6.1.2) to be executed by the C&C should be directed to Task ID **0x02**.

[2] Using Task ID **0x03**, a device can request to send literal text to the Console.

[3] New File Available messages (ref. 8.5.1) should be sent to Task ID **0x05**.

10.5 Other Considerations

10.5.1 Message Routing

Devices that implement this protocol can act as message routers for connected child devices. Responsibilities of the message router are:

- Check messages received and handle any messages bound for the router device, or else forward to children. If the message has an unknown destination, queue for sending to parent when polled.
- Keep track of child devices that have indicated they have messages to send. When polled for pending messages, poll those children that have messages waiting. Message routers should not buffer messages from children.
- Messages with a broadcast Board ID (i.e., **0xFF**) should be forwarded to all children in addition to being processed by the message router.
- Route the **Power Status and Control** message promptly. Broadcast notifications to children when power removal is imminent.

10.5.2 Timeouts and Message Usage

Timeout and retries used for the exchange of messages should be controlled with configuration variables in the device software. The settings and suggested values are:

Setting	Description	Range	Suggest
Data Link Response Timeout	The maximum time (in milliseconds) to wait for Data Link ACK or NAK before attempting retry or discarding. This time is not end-to-end but between communicating peers	10 to 60000 ms	5000 ms
Data Link Retries	The maximum number of retries to make before failing the transaction. A parent has the option to retry. A child will not retry.	0 to100	3
Message Receive Timeout	The maximum time (in milliseconds) to wait between bytes of a message before abandoning it.	10 to 60000 ms	100 ms

Note: Messages indicated in any examples represent intended usage and were developed in response to Customer requests. While other use scenarios are possible, please consult with LRI on the best course of action if your requirements are not addressed.

10.6 Payload Message Types

Message IDs below **0x1000** are reserved for LRI use. Any customer-developed messages specific to their devices must use an ID value greater than **0x1000**.

The reserved Message Types are listed below. This list will expand as needed. The messages are described in detail in section 10.7.

Msg Type	Description	Direction	Detail	Broad cast?	Ref.

Msg Type	Description	Direction	Detail	Broad cast?	Ref.
0x0000	NAK	Child → Parent	Unable to handle message or request	No	10.7.2
0x0001	ACK	Child → Parent	Successfully handled message/ request	Yes [3]	10.7.3
0x0010	Request ID/ Enumerate	Parent → Child	Parent requests device ID and that of any attached devices	Yes	10.7.4
0x0015	Telemetry	Parent → Child	Telemetry uses a different interval than then shore-bound telemetry	Yes	10.7.6
0x0022	Request Status	Parent → Child	Parent requests status of Child	Yes	10.7.7
0x0024	Send/Forward Message	Parent → Child	Parent copies a requested message to child node	Yes	10.7.9
0x0030	Power Status and Control	Parent → Child	Parent notifies Child that power is about to turn off	Yes	10.7.12
0x0040	Request Queued Message	Parent → Child	Parent requests report of any queued message	No	10.7.14
0x0041	ACK/NAK Queued Message	Parent → Child	Parent forwards result of message execution	No	10.7.17

Notes:

[1] For response messages from child to parent, the high bit (bit 15) of the Original source Msg Type number can be set to indicate that there are messages queued for sending and that they should be read out using Msg Type **0x0040** (Request Queued Message).

[2] For messages from Parent to Child, the high bit of **Msg Type** (0x8000) can be set to indicate that no ACK should be sent in response to the message.

[3] If the ACK is in response to a Request Queued Message (0x0040), it can broadcast.

10.7 Payload Message Packet Descriptions

10.7.1 Payload Header Abbreviation

For the following explanations, the header for the message packets may be abbreviated thus:

Description	Bytes	Explanation
Header	9	SOF + Header data

Field	Bytes
Start-of-Frame	1
Message Size	2
Destination Address	2
Source Address	2

Transaction ID	2

10.7.2 Generic Payload NAK Data Packet (0x0000)

If a recipient cannot properly handle a message or request, it sends a NAK data packet back to the original sender.

The **Destination Address** is set to the original source address. The **Source Address** is set to the recipient's address. The **Transaction ID** value is set to the Transaction ID in the original request.

Note that the **Msg Type** of the original source message is included in the Message Data. Bit 15 (**0x8000**) can be set to indicate that there are additional NAK response messages.

Depending upon the type of NAK, to **541** bytes of additional information can be returned. Some responses can return an entire Level 2 NAK message (ref. 6.1.2), including the CRC.

Description	Bytes	Explanation
Header	1	Start-of-Frame
	2	Message Size
	2	Destination Address
	2	Source Address
	2	Transaction ID
Msg Type	2	**0x0000**
Message Data	2	Original source Msg Type number (bit 15 set if additional messages available)
	1	Message Code: **0**: Returning Level 2 NAK Message **1**: reserved **2**: reserved **3**: reserved **4**: Program Download Requested **5**: Request Power Cycle **6**: Request Power Off **7 → 31**: Reserved For LRI **32 → 255**: Customer definable
	≤ **541**	NAK details or reply data (if any).
Error Chk	2	CRC-16

NAK Message Types are as follows:

Code 0: Returning NAK Data Message

If the NAK Message Code is 0, the response contains a Level 2 enumerated NAK message (ref. 8.3), including the CRC.

Code 1, 2, 3: (Reserved)

These Message Codes are reserved for future LRI use.

Code 4: Program Download Requested (Not Implemented)

If the NAK Message Type = 4, the following data structure is returned in the message field:

Description	Bytes	Detail
Program Download Reason	1	**0x00**: Responding to request to load newly selected program **0x01**: Responding to request to load default program **0x02**: CRC did not match **0x03**: MD5 did not match **0x04**: Bad configuration in EEPROM **0x05**: Tentative image execution failed
CRC Expected	2	CRC expected for program image
CRC Actual	2	CRC actually computed
MD5 Expected	16	MD5 signature expected for program image
MD5 Actual	16	MD5 signature actually computed (Can be zero if boot loader is not able to compute MD5)
Number of Failed Starts This Image	2	Number of times the boot loader has attempted to start the program, but determined that it had failed (Never increments above 0xFFFF)
Program Type Loaded	1	**0x00**: Non-default program loaded **0x01**: Default program loaded

Code 5: Request Power Cycle (Not Implemented)

Code 6: Request Power Off (Not Implemented)

If the NAK Message Code = 5 or Code = 6, the following two 1-byte data values are returned in the message field:

Description	Bytes	Detail
Power Off Delay	1	Minimum number of seconds to wait after receiving the NAK before powering Off
Power On Delay	1	Minimum number of seconds to wait after powering Off before powering On again

10.7.3 Generic Payload ACK Data Packet (0x0001)

If a message is successfully handled, an ACK data packet will be sent to the original message sender.

The **Destination Address** is set to the original source address. The **Source Address** is set to the recipient's address. The **Transaction ID** value is set to the Transaction ID in the original request.

All ACK responses return the **Msg Type** of the original source message at the beginning of the Message Data. Bit 15 (**0x8000**) can be set to indicate that there are additional ACK response messages.

Up to **542** bytes of additional information can be added. Some ACK responses can return an entire Level 2 message (ref. 6.1.2), including the CRC.

Description	Bytes	Explanation
Header	1	Start-of-Frame
	2	Message Size
	2	Destination Address
	2	Source Address
	2	Transaction ID
Msg Type	2	**0x0001**
Message Data	2	Original source Msg Type number (bit 15 set if additional messages available)
	≤ 542	ACK details or reply data (if any).
Error Chk	2	CRC-16

10.7.4 Request ID/Enumerate (0x0010)

This payload message requests identifying information about a Device. The message contains bit-maps that indicate which protocols and commands are supported by the C&C. Each Device responds with an ACK or NAK.

An ACK response contains information specifying which C&C protocols that Device supports.

A NAK response indicates that the Device doesn't support any of the protocols. If there is a NAK (or no response), the C&C will not send any messages to that Device.

For this message, the **Message Data** contains two fields:

- A 2-byte bit-mask specifies what communication protocols are supported by the C&C.
- A 2-byte bit-mask specifies what command formats are supported by the C&C.

The Request ID soliciting command format is:

Description	Bytes	Explanation
Header	9	SOF + Header data
Msg Type	2	**0x0010**
Message Data	2	Bit mask of protocols supported by the Float C&C: **0x0001**: Existing Device Protocol All other bit mask values are reserved by LRI
	2	Mask of command formats accepted by the Float C&C **0x0001** = Existing Ship-to-Shore commands

Error Chk	2	CRC-16

In the specific case where the message is sent as a broadcast, the device should resend the request to its children and should concatenate responses. The receiving devices should only respond if they can communicate using at least one of the Float C&C protocols.

10.7.5 ACK Reply to Enumerate Message

Multiple response blocks may be returned using an ACK with the format shown below:

Note that given the Message Data size limit, this ACK response can contain information on no more than 10 devices.

If more than 10 Information Blocks need to be returned, the last Information Block will contain a bit flag indicating that additional ACK messages need to be issued. Subsequent messages will use the bit flag until all devices have been defined.

Description	Bytes	Data Type
Header	9	SOF + Header data
Msg Type	2	**0x0001**
Message Data	2	Original source Msg Type number (bit 15 set if additional messages available)
	2	Number of devices responding
	2	Number of devices in this message
	50	1st Device Information Block (see below)
	50	2nd Device Information Block
	. . .	. . . .
	50	10th Device Information Block (if necessary)
Error Chk	2	CRC-16

Device Information Block Format

There may eventually be different Device Information Block formats. Currently, there is only one.

Thus, all Device Information Blocks all use the following format:

Description	Bytes	Value	Detail
Version	2		Device Information Version Format number: **1:** This format
Device Type	2		**0x0000** – **0x1000** are reserved for LRI **0x1001** – **0x7FFF** are allowed for customer use Bit flag **0x8000** → More Device Blocks waiting
Address	2		[Board ID + Task ID] Address of responder [1]

Description	Bytes	Value	Detail
Serial Number	16	ASCII	Optional Manufacturer's serial number for the device, as ASCII characters. Padded on right with zeros (null terminators). A completely null string is allowed.
Location/Port	1	0	This field is reserved and should be set to zero (0)
Polling Frequency	4	Any	The requested number of seconds between polling operations, using the Telemetry message [2]
Want Extra Info	1	0x00 – 0x0F	Bit masks indicating whether a component wants to receive event notifications from the Float. **0x01**: Telemetry information desired **0x02**: Power information desired **0x04**: Event information desired **0x08**: Command ACK/NAK desired The upper 4 bits (0xF0) are reserved for LRI and should be set to 0
Firmware Major Revision	1		Integer
Firmware Minor Revision	1		Integer
Description	20	ASCII	Human-readable description of device. ASCII text field, padded on the right with spaces

Notes:

[1] The Board ID is typically programmed into non-volatile RAM during the manufacturing process. It also may be hard-coded by the customer.

[2] The polling frequency used will be the fastest (shortest interval) of the polling frequencies specified in all the Device Information blocks.

10.7.6 C&C Telemetry (0x0015)

If requested in a Device Information Block (ref. 10.7.5). The Float C&C can send Telemetry to each device at a user-configurable rate. The rate is set as the fastest of all the rates (shortest interval) requested in the Device Information Blocks.

The Telemetry message sent to the devices is a duplicate of the Level 1 telemetry sent to shore via the Iridium/Relay link (ref. 6.1). The structure ID from the telemetry message determines how the message content is decoded. The message packet format is:

Description	Bytes	Data Type

Header	9	SOF + Header data
Msg Type	2	**0x0015**
Message Data	1	Structure ID for Telemetry data
	≤ 544	Level 1 Telemetry data. Ref. Chapter 9
Error Chk	2	CRC-16

The response to this message is either an ACK or NAK with message content empty. As with any ACK/NAK, a device can signal if there are messages queued to be read.

10.7.7 Request Status (0x0022)

This is the basic polling message. It is also used to update various commonly-needed values.

Description	Bytes	Data Type	Detail
Header	9	SOF + Header data	
Msg Type	2	**0x0022**	
Message Data	6	Time	One byte per field. Bytes in order: 0: Second (0 – 59) 1: Minute (0 – 59) 2: Hour (0 – 23) 3: Day of Month (1 – 31) 4: Month (1 – 12) 5: Year since 2000 (0 – 255)
	4	Latitude	-90° < Latitude ≤ +90° Current Latitude encoded as a 32-bit floating-point number. Note [1]
	4	Longitude	-180° < Longitude ≤ +180° Current Longitude encoded as a 32-bit floating-point number. Note [2]
	1	Fix Valid?	Single character: '**Y**' = Latitude and Longitude are valid '**N**' = Latitude and Longitude are not valid
Error Chk	2	CRC-16	

Notes:

[1] Latitude = *degrees**600000 + *minutes**10000 + *minute_fraction*.
[-90 < *degrees* ≤ 90] [0 ≤ *minutes* < 60] [0 ≤ *minute_fraction* < 10000]
If *degrees* < 0, location is in the Southern hemisphere

[2] Longitude = *degrees**600000 + *minutes**10000 + *minute_fraction.*
-180 < *degrees* ≤ 180. 0 ≤ *minutes* < 60. 0 ≤ *minute_fraction* < 10000.
If *degrees* < 0, location is in the Western hemisphere

10.7.8 ACK Reply to Request Status

The response to the Request Status is returned in an ACK packet.

As with any ACK, the device can signal if there are messages queued to be read.

In the case of a broadcast, the C&C or a PIB will retransmit the message to its connected devices and will accumulate responses. Depending upon the message, multiple response messages may be needed.

The Status ACK returns one or more blocks of status information for the responding devices. As described below, there are two Status Block versions, with different formats.

Description	Bytes	Data Type
Header	9	SOF + Header data
Msg Type	2	**0x0001**
Message Data	2	Original source Msg Type number (bit 15 set if additional messages available)
	2	Total number of responses (must be ≥1) (may be more than can fit in this message)
	2	Number of responses in this message. Ref. Note [1]
	4 or 12	1st Status Block
	4 or 12	2nd Status Block
	4 or 12	3rd Status Block
	. . .	. . . .
CRC-16	2	

Note:

[1] Since the size of the Status Blocks is variable, the number of responses in the message will be variable also.

Status Block, Version = 0

If a Status Block has nothing to report, it uses Version 0, which contains 4 bytes.

Description	Bytes	Explanation
Version	2	**0x0000**
Address	2	Address of responding device

Status Block, Version = 1

If there is data to report in the Status Block, it uses Version 1, which contains 12 bytes.

Description	Bytes	Explanation
Version	2	**0x0001**
Address	2	Address of responding device
Alarms	2	Alarm flags (16 bit-field flags)
Leak Sensor 1	1	Status of leak sensor #1

Leak Sensor 2	1	Status of leak sensor #2
Humid Temp	1	Temperature from humidity sensor (°C)
Relative Humidity	1	% Humidity
Pressure Temp	1	Temperature from pressure sensor (°C)
Pressure	1	Air pressure (kPa)

10.7.9 Forward/Send Message (0x0024)

This message is used to either forward or send Level 2 data packets (messages that conform to the Ship-to-Shore protocol) to a device. These packets have either been requested as copies (e.g., of Telemetry) or are meant for interaction with systems external to the Vehicle.

The format is:

Description	Bytes	Data Type	Detail
Header	9	SOF + Header data	
Msg Type	2	**0x0024**	
Message Data	1	Flag	Indicates whether this is a Forward or a Send as: **0x00**: Forward [1] **0x01**: Send [2]
	2	Command Format	Mask of command formats accepted that this message conforms to: **0x0001**: Ship-to-Shore commands (ref. 7.2)
	≤ 541	Level 2 Packet	The Level 2 packet (ref. 6.1.2) containing the message, including the CRC.
Error Chk	2	CRC-16	

Notes:

[1] In the Forward case the command is only informational and typically no ACK is requested.

[2] In the Send case, the command is expected to be executed.

10.7.10 ACK for Forward

If an ACK is requested for the “Forward” case (not typical), the response is an empty ACK message (ref. 10.7.3).

10.7.11 ACK for Send

If an ACK is requested for the “Send” case, the ACK packet carries the response type expected by the Level 2 message (ref. 6.1.2), encapsulated as follows:

Description	Bytes	Data Type
Header	9	SOF + Header data
Msg Type	2	**0x0001**

Message Data	2	Original source Msg Type number (bit 15 set if additional messages available)
	≤ 542	Level 2 response message, including the CRC (ref. 6.1.2)
Error Chk	2	CRC-16

10.7.12 Power Status and Control (0x0030)

This message is sent to:

1) Notify a device of an intended power shutdown
2) Set the power state on a device
3) Get the power state of a device

Child devices only need to implement the notify scenario, but the C&C and PIBs should also implement the Get/Set scenario. PIBs receiving this message directly (i.e., not as a broadcast) should broadcast the command to its child devices as appropriate.

Description	Bytes	Data Type	Detail
Header	9	SOF + Header data	
Msg Type	2	**0x0030**	
Message Data	1	Command Type	Numerical flag value: **0x00**: Notification of power Off **0x01**: Set a new power state **0x02**: Get the existing power state
	1	Board Affected	Reserved value: **0x00**: Recipient Device **0xFF**: All Devices (broadcast) Otherwise, the Board ID of the Device (ref. 10.4.2)
	1	New Power State	For Set Power (command Type 1): **0x00**: Power Off **0x01**: Power On For Notification (Type 0) or Get State (Type 2): **0x00**
Error Chk	2	CRC-16	

10.7.13 ACK Response to Power Status and Control

Multiple response blocks may be returned in an ACK with the format shown below.

Description	Bytes	Data Type
Header	9	SOF + Header data
Msg Type	2	**0x0001**
Message	2	Original source Msg Type number

Data		(bit 15 set if additional messages available)
	2	Number of total devices responding
	2	Number of responses in this message
	6	1st Power State Record
	6	2nd Power State Record
	6	3rd Power State Record
	. . .	. . . .
Error Chk	2	CRC-16

Power State Record Format

The Power State Records all use the following format:

Description	Bytes	Detail
Address	2	Address of responder
Board ID Number	1	The ID of the board number affected
Power State	1	Flag value: **0x00**: Power Off **0x01**: Power On
Power Off Delay	1	The minimum number of seconds to wait after receiving the ACK before powering Off
Power On Delay	1	The minimum number of seconds to wait after powering Off before powering On again

10.7.14 Request Queued Message (0x0040)

This packet contains no message. It thus has the following format:

Description	Bytes	Explanation
Header	9	SOF + Header data
Msg Type	2	**0x0040**
Error Chk	2	CRC-16

10.7.15 NAK Response to Request Queued Message

If there is no data to be read, the following NAK is returned:

Description	Bytes	Explanation
Header	9	SOF + Header data
Msg Type	2	**0x0000**
Message Data	2	Original source Msg Type number (bit 15 set if additional messages available)
	1	**0x02** (Invalid Data)
	2	Length = **0x00**
Error Chk	2	CRC-16

10.7.16 ACK Response to Request Queued Message

If there is a message to be passed up, the format is:

Description	Bytes	Explanation
Header	9	SOF + Header data
Msg Type	2	**0x0001**
Message Data	2	Original source Msg Type number (bit 15 set if additional messages available)
	≤ 542	Data message to be passed to destination
Error Chk	2	CRC-16

10.7.17 ACK/NAK Queued Message (0x0041)

This message forwards the response that was made to a previously-read queued message.

The format is:

Description	Bytes	Explanation
Header	9	SOF + Header data
Msg Type	2	**0x0001**
Message Data	2	Command Format: **0x0001**: ICD Format

	1	Response Code: **0x00**: ACK **0x01**: NAK - Unknown Command **0x02**: NAK - Invalid Data **0x03**: NAK - Unable to Complete **0x04**: NAK - Unknown Format
	2	Length of the response message field
	≤ 540	The response message (if any)
Error Chk	2	CRC-16

The response to this message is an ACK (ref. 8.2.4) with no payload, or else a NAK (ref. 8.3.1) with no accompanying data.

10.8 Message Protocol Use Cases

Shown below are different use scenarios and the sequence of actions taken for each.

The text format indicates the following:

Bold indicates a command (or ACK/NAK response)

Italic indicates an action taken

Italic indicates a comment

10.8.1 Normal Polling Update Time and Location

This case shows a PIB trying to send a Request Status command to a Device.

PIB		Device
Request Status	→	*Data link verifies CRC and transport packet*
		Interprets message, sets timestamp and saves location
Data link verifies CRC and transport packet	←	**ACK** (*single response, no queued messages, version 0*)
Handles status request reply		

10.8.2 Startup Identification Broadcast

PIB		Device
System startup or device power up, PIB waits for a specified/assumed time after startup before attempting to communicate. Perhaps multiple devices are being queried, so PIB initializes a response buffer		
Request ID/Enumerate	→	*Data link verifies CRC and transport packet*
		Interprets message, verifies compatibility with Float C&C Protocol Mask *Reads its Serial Number and Asset ID from non-volatile RAM or other storage*
Data link verifies CRC and transport packet	←	**ACK** (*single response, no queued messages*)
Forwards response to requester, maps Address field of response to Port to build a routing table for future messages		

10.8.3 Device Program Download, Program Resident on Float C&C File System

PIB		Device
		Device starts up and finds that the loaded program is invalid or it has been requested to reload its program; it waits in its boot loader or the equivalent
Any message sent to device	→	*Boot loader verifies the message CRC and transport fields*
Message layer verifies the message CRC and transport fields	←	**NAK** (*Program Download Request*)

PIB		Device
Message goes back to requester, maybe the Float C&C; *PIB queues the request for next Request Queued Message* *Float C&C schedules file write to device*		
Send/Forward Message (*send Initialize File Write*)	→	*Boot loader verifies the message CRC and transport fields*
		Boot loader stores data to program space or equivalent
Message layer verifies the message CRC and transport fields	←	**ACK** (*Write File ACK*)
Reply goes back to the Float C&C, which schedules the next block		
Send/Forward Message (*send Write File Block*)	→	*Boot loader verifies the message CRC and transport fields*
Message layer verifies the message CRC and transport fields	←	**ACK** (*Write File ACK*)
Reply goes back to the Float C&C, which schedules the next block		*Boot loader stores data to program space or equivalent*
The Float C&C continues to send Write File Block until all file blocks are written.		
Send/Forward Message *(send Select Program)*	→	*Boot loader verifies the message CRC and transport fields*
		Boot loader checks loaded image against CRC (and MD5 if possible)
Message layer verifies the message CRC and transport fields	←	**ACK** *(Level 1 ACK)*
Reply goes back to Float C&C, which waits for several seconds before continuing with Request ID/Enumerate scenario		*Boot loader jumps into newly loaded program*

10.8.4 Device Receives File from Shore

PIB		Device
Shore side sends a file tagged for this Asset ID and the file is stored in the file system on the Float C&C; after the write completes, the file is checked, and this process begins. *(Both the PIB and the Device have to be powered on in order to forward the file.)*		

PIB		Device
Send/Forward Message (send Initialize File Write)	→	*Message layer verifies the message CRC and transport fields*
		Space is reserved for the file on the device, initial verification checks are made *Application-specific logic can be applied to determine what the file contains based on the file name or the content of the first block of data, which accompanies Initialize File Write*
Message layer verifies the message CRC and transport fields	←	**ACK** (Write File ACK)
ACK message is forwarded to Float C&C, which schedules the next packet		
Send/Forward Message (send Write File Block)	→	*Message layer verifies the message CRC and transport fields*
		Data is written to file or storage
Message layer verifies the message CRC and transport fields	←	**ACK** (Write File ACK)
ACK message is forwarded to Float C&C, which schedules the next packet		
Float C&C continues to send Write File Block until all blocks are transferred		
Send/Forward Message (send Get File List, indicating the specific file that was sent and asking for the file to be checked)	→	*Message layer verifies the message CRC and transport fields*
		The file CRC and MD5 are checked against the expected values from Initialize File Write)
Message layer verifies the message CRC and transport fields	←	**ACK** (Get File List Data)
Float C&C sends File At Destination event to Shore		*File is now resident on device; this event can be used to interpret the file if it is actually a command or parameter set*

10.8.5 Send File from Device to Shore

(Send File from Device to Shore)

PIB		Device
		Device finishes creating a new data set to be sent to the Shore-Side host

(Send File from Device to Shore)

PIB		Device
Request Status	→	*Message layer verifies CRC and transport packet*
		Interprets message, sets timestamp and saves location
Message layer verifies CRC and transport packet	←	**ACK** *(single response, have queued messages, version 0)*
Handles status request reply		
Request Queued Message	→	*Message layer verifies CRC and transport packet*
		Selects the next message in its queue to be sent
Message layer verifies CRC and transport packet	←	**ACK** *(response with New File Available message tagged for File Transfer to Shore Task ID)*
Float C&C sends New File Available message to shore side via the configured method		
ACK/NAK Queued Message *(simple ACK)*	→	*Message layer verifies CRC and transport packet*
Message layer verifies CRC and transport packet	←	**ACK** *(no accompanying data)*
Float C&C is informed of result		
Shore side system evaluates whether the file should be uploaded according to rules configuration and initiates a Read File request. Float C&C starts a Read File operation to buffer the file in the Float C&C file system before transmitting		
Send/Forward Message *(send Read File, indicating the specific file that was advertised)*	→	*Message layer verifies CRC and transport packet*
		Allocates a read operation context that can returned chunks of the file; saves the requester and the transaction ID as all resulting blocks will be returned through the same path
Message layer verifies CRC and transport packet	←	**ACK** *(Read File ACK containing the first block of data requested, indicates more data available if more blocks are to be read from the file)*
Float C&C stores the file chunk in its file system and schedules next check for this device		

(Send File from Device to Shore)

PIB		Device
Request Queued Message	⟶	*Message layer verifies CRC and transport packet*
		Selects the next message in its queue to be sent, which is Read File ACK
Message layer verifies CRC and transport packet	⟵	**ACK** *(responds with Read File ACK message tagged with requester's address)*
Float C&C stores the file chunk in its file system and schedules next check for this device		
The Float C&C continues its scheduled polling and receives all chunks of the file. It then begins to respond to the Read File request and sends the file in chunks to the remote requester. *When the requester has received the entire file, it issues Read File Transaction Closed to the Float C&C, which deletes the copy of the device's file from its file system.*		

10.8.6 Device Reads File from Float C&C File System

PIB		Device

PIB		Device
Request Status	→	*Message layer verifies CRC and transport packet*
		Interprets message, sets timestamp and saves location
Message layer verifies CRC and transport packet	←	**ACK** *(single response, have queued messages, version 0)*
Handles status request reply		
Request Queued Message	→	*Message layer verifies CRC and transport packet*
		Selects the next message in its queue to be sent
Message layer verifies CRC and transport packet	←	**ACK** *(responds with Read File message tagged for the command interface task ID on the Float C&C)*
Float C&C allocates and schedules Read File operation		
ACK/NAK Queued Message *(passing result of Read File request)*	→	*Message layer verifies CRC and transport packet*
		Stores resulting data
Message layer verifies CRC and transport packet	←	**ACK** *(simple ACK)*
Float C&C schedules the next block to be sent		
Float C&C continues this process until the read request data has been completely sent		
Request Queued Message	→	*Message layer verifies CRC and transport packet*
		Selects the next message in its queue to be sent
Message layer verifies CRC and transport packet	←	**ACK** *(responds with Read File Transaction Closed message tagged for the command interface task ID on the Float C&C)*
Float C&C de-allocates its Read File transaction		

10.8.7 Device Writes File to Float C&C File System

PIB		Device
Request Status	⟶	*Message layer verifies CRC and transport packet*
		Interprets message, sets timestamp and saves location
Message layer verifies CRC and transport packet	⟵	**ACK** *(single response, have queued messages, version 0)*
Handles status request reply		
Request Queued Message	⟶	*Message layer verifies CRC and transport packet*
		Selects next message in its queue to be sent
Message layer verifies CRC and transport packet	⟵	**ACK** *(responds with Initialize File Write message tagged for the command interface task ID on the Float C&C)*
Float C&C allocates file and file write operation, stores the first block to the file		
ACK/NAK Queued Message *(passes result of Initialize File Write)* *(marked as No ACK)*	⟶	*Message layer verifies CRC and transport packet*
Float C&C schedules the next device poll		
Float C&C continues to read out the Write File Block requests and completes the file.		
Request Queued Message	⟶	*Message layer verifies CRC and transport packet*
		Selects next message in its queue to be sent
Message layer verifies CRC and transport packet	⟵	**ACK** *(responds with Get File List message indicating the particular file that was just transferred and requesting a check, tagged for the command interface task ID on the Float C&C)*
File is checked and determined to be OK, CRC and MD5 signatures are matching		
ACK/NAK Queued Message *(passing Get File List Data response)*	⟶	*Message layer verifies CRC and transport packet*
Message layer verifies CRC and transport packet	⟵	**ACK** *(simple ACK)*
ACK/NAK Queued Message *(passing File At Destination response)* *(marked as No ACK)*	⟶	*Message layer verifies CRC and transport packet*
Float C&C de-allocates file write operation		

10.8.8 Device Receives Power Off Notification

PIB		Device
Power Status and Control *(Notification of Power Off)*	→	*Message layer verifies CRC and transport packet*
Message layer verifies CRC and transport packet	←	**ACK** (*single response, power is ON, power off delay 3 seconds, power on delay 60 seconds*)
Float C&C schedules power off		*Prepares to shut down*
Float C&C turns power off		

10.8.9 Device Requests Power Cycle (Not Implemented)

PIB		Device
Request Status	→	*Message layer verifies CRC and transport packet*
		Interprets message, sets timestamp and saves location
Message layer verifies CRC and transport packet	←	**NAK** *(Power Cycle Request, power off in 3 sec, power on 60 seconds later)*
Float C&C schedules power off		
Float C&C powers off 3 seconds later		
Float C&C powers on 60 seconds thereafter; it may automatically restart identification process		

10.8.10 Send Message to Console

PIB		Device
Request Status	→	*Message layer verifies CRC and transport packet*
		Interprets message, sets timestamp and saves location
Message layer verifies CRC and transport packet	←	**ACK** *(single response, have queued messages, version 0)*
Handles status request reply		
Request Queued Message	→	*Message layer verifies CRC and transport packet*
		Selects the next message in its queue to be sent
Message layer verifies CRC and transport packet	←	**ACK** *(responds with Write Console to Client message tagged for the command interface task ID on the Float C&C)*
Float C&C sends to console		
Send/Forward Message for Write Console Input to the Vehicle is likely to be sent without request for ACK, in which case the scenario stops here		
ACK/NAK Queued Message *(passing result of Initialize File Write request)* *(marked as No ACK)*	→	*Message layer verifies CRC and transport packet*

10.8.11 Receive Input from Console

PIB		Device
Send/Forward Message *(send Write Console Input to Vehicle)*	→	*Message layer verifies the message CRC and transport fields*
		Device queues input to its console. *Note: Device's console input may overflow if data is sent too quickly*
Send/Forward Message for Write Console Input to Vehicle is likely to be sent without a request for ACK, in which case the scenario stops here. Otherwise, the following occurs:		
Message layer verifies the message CRC and transport fields	←	**ACK** *(simple ACK)*
Reply goes back to Float C&C		

11. XBee Communications

11.1 XBee Command Menu

The following commands are shown on the XBee menu. XBee commands are submitted by entering the reference number of the command.

```
1) LED On
2) LED Off
3) IR On
4) IR Off
5) Turn XBee On
6) Rudder position
7) Rudder Right
8) Rudder Center
9) Rudder Left
10) Set Rudder Trim
11) Show Rudder Trim
12) Set Heading
13) Add Waypoint
14) Clear Waypoint
15) Go To Waypoint (0=hold)
16) Set Course
17) Set Declination
18) Set Sub Compass Cal
19) Show Compass Offsets
20) Rudder Delta
21) Set steering params
22) Set debug level and mask
23) Set float compass offset
24) Show steering params
25) Show WayPoint
26) Show System Charge
27) Show Ind. Battery
28) Pyld1 On
29) Pyld1 Off
30) Pyld2 On
31) Pyld2 Off
32) Show Rudder Compliance
33) Set Rudder Compliance
34) Set GPS Alarm Threshold
35) Bubble On
36) Bubble Off
37) Sub On
38) Sub Off
39) Set Float/Sub Nav Rate
40) Show Float/Sub Nav Rate
41) Set P, I, Sat. values
42) Show P, I, Sat. values
43) PID Navigation On
44) PID Navigation Off
45) Sub Pyld On
46) Sub Pyld Off
47) Ir. Modem Always On
48) Ir. Modem Idle=>Off
49) Show Ir. Mode
50) Cur. Ir. Mode=Default
51) I-PIB On
52) I-PIB Off
54) Set XBee Default
55) Set Ir. Default
91) Reset Pyld File Xfer
92) Set Default Radio     (rev 3 only)
93) Set Default Modem     (rev 3 only)
94) Show Error Counts
95) Show WayPoint Table
96) Show IMEI
97) Show Versions
98) XBee Off
99) Reboot Float
100) Display menu
```

For information on setting up XBee communications, refer to the LRI document: *System Administration*.

11.2 XBee Menu Options

Some XBee menu commands have no confirming responses. Others request additional information.

Hitting the Enter key without entering a number will cause the computer to print out status data. Refer to section 11.3 “Status Response".

1) LED On
2) LED Off

Turns the visible LED lights ON or OFF. The LEDs go on at full power, non-blinking. There is no text response.

3) IR On
4) IR Off

Turns the Infrared LED lights ON or OFF. The LEDs go on at full power. There is no text response.

5) Turn XBee On

This command has no effect.

6) Rudder position

Sets a specific rudder position (not a heading). Positive values indicate Right rudder, negative indicate Left rudder. The allowed range is from –30 to 30.

Request:

```
Enter Rudder Position in Degrees (+30(Right) ==> -30(Left)):
```

7) Rudder Right
8) Rudder Center
9) Rudder Left

Positions the rudder full Right, full Left, or in the center. Left or Right deflection is 30°. There is no text response.

10) Set Rudder Trim

Sets the rudder trim value, in unit degrees. The positive or negative trim value is added to the rudder position.

Request:

```
Enter Rudder Trim Value in Degrees (+30(Right) ==> -30(Left)):
```

11) Show Rudder Trim

Shows the rudder trim value, in degrees.

Output:

```
Rudder Trim: [degrees]
```

12) Set Heading

Sets a new heading value, in degrees:

Request:

```
Enter New Heading in Degrees  (0 ==> 359):
```

13) Add Waypoint

Adds a single new Waypoint number and its Latitude/Longitude. Seven items have to be entered in the following order:

Waypoint number (1 to 254)
Degrees Latitude (0 to 89)
Minutes Latitude (0 to 59.9999)
The letter N or S indicating Latitude direction
Degrees Longitude (0 to 179)
Minutes Longitude (0 to 59.9999)
The letter E or W indicating Longitude direction

Request:

```
Enter New Waypoint [# dd mm.mmmm n/s dd mm.mmmm e/w ]
```

14) Clear Waypoint

Deletes a single Waypoint, defined by the Waypoint number.

Request:

```
Enter Waypoint Number To Clear:
```

15) Go To Waypoint (0=hold)

Directs Vehicle to proceed to a previously-defined Waypoint. If the Waypoint number is zero, the Vehicle holds at the current position.

Request:

```
Enter Target WayPoint Number:
```

16) Set Course

Defines a course consisting of sequential Waypoint numbers, and instructs the Vehicle to proceed toward the Target Waypoint, and then begin looping the course.

Request:

```
Enter Course Info [First_WayPt Last_WayPt Initial_WP_Target]:
```

17) Set Declination

Sets the magnetic declination value at the current location. The value is specified in unit degrees, and can be positive or negative.

Request:

```
Enter The New Declination:
```

18) Set Sub Compass Cal

Adds a fixed offset to the Sub (Glider) compass reading. This is positive or negative value in unit degrees.

Request:

```
Enter The Calibration Factor For The Sub Compass:
```

19) Show Compass Offsets

Shows the compass offset values for both the Float and the Sub (Glider).

Output:

`Declination: [`*degrees*`]  -- Sub Offset: [`*offset*`]`

20) Rudder Delta

Sets the delta value for the rudder.

Used by LRI for engineering purposes only.

Request:

```
Enter rudder delta:
```

21) Set steering params

Sets the control parameters used for the rudder servo. All 21 values (3 sets of 7) must be specified in a single string (which may wrap onto multiple lines). The string is terminated with the ENTER key.

Request:

```
Settings are:  <Gain> <Timebase> <Deadband> <Goal> <RudderMax>
<MoveMin> <TrimMax>
Format: <Normal Power Settings> <Reduced Power Settings>
<Critical Power Settings>
```

22) Set debug level and mask

Sets the debug level and the debug bit mask value. The level is a value between 0 and 5. The mask is a single byte value between 0 and 255. Both values have to be entered.

Request:

```
Enter "Level Mask" debug parameters:
```

Output:

`Debug Level = [`*level*`]`

`Debug Mask  = [`*hex mask*`]`

23) Set float compass offset

Adds a fixed offset to the Float compass reading. This is positive or negative value in unit degrees.

Request:

```
Enter float offset:
```

Output:

```
Offset = [degrees]
```

24) Show steering params

Shows the various Vehicle heading and steering values. Values are given for all three power levels (Normal, Reduced and Critical). These are the same values specified in the same order shown in section 8.2.6, "ACK – Rudder Control Data (0x15)".

Output:

```
Normal Power
Gain = [gain], Timebase = [divisor], Deadband = [degrees], Goal = [degrees]
RudderMax = [degrees], MoveMin = [degrees], TrimMax = [degrees]
Reduced Power
Gain = [gain], Timebase = [divisor], Deadband = [degrees], Goal = [degrees]
RudderMax = [degrees], MoveMin = [degrees], TrimMax = [degrees]
Critical Power
Gain = [gain], Timebase = [divisor], Deadband = [degrees], Goal = [degrees]
RudderMax = [degrees], MoveMin = [degrees], TrimMax = [degrees]
```

25) Show WayPoint

Shows the Latitude and Longitude values for a specific Waypoint number.

Request:

```
Enter The WayPoint Number:
```

Output:

```
Way Point #[1→ 254]: [lat °] [min.dddd] [N,S]  [long °] [min.dddd]
[E,W]
```

26) Show System Charge

Shows the current draws, power usage and solar panel charging values.

Note: Each solar panel has two halves. Thus there are four panel charging values.

Output:

```
Pyld1=[nn](10mA), Pyld2=[nn](10mA), Umb=[nn](10mA), Twst=[nn](10mA)
          (128=Off, 129=Over Current)
Total Pwr=[nn](10mWh), Bat Cur=[mA]mA, Sys Cur=[mA]mA, Sys Volt=[mV]mV
Panel0=[nn], Panel1=[nn], Panel2=[nn], Panel3=[nn]
```

27) Show Ind. Battery

Shows the state of all seven batteries. The bits for the "Status" indicate the following:

Hex	Binary	State
0x00	0000 0000	No Battery
0x01	0000 0001	Battery Present
0x02	0000 0010	Discharged
0x04	0000 0100	Charged
0x08	0000 1000	Discharging
0x10	0001 0000	Discharge Alarm
0x20	0010 0000	Charge Alarm
0x40	0100 0000	Temperature Alarm
0x80	1000 0000	On-Line

Note that the Status bits can be OR'ed together to indicate more than one state.

Output:

```
Battery 0: Status=[0xhh], Temp = [deg]C
           Power=[nn] (10mWh), Current=[mA]mA, Voltage=[mV]mV
Battery 1: Status=[0xhh], Temp = [deg]C
           Power=[nn] (10mWh), Current=[mA]mA, Voltage=[mV]mV
  . . . . . . . . . . . . . . . . . . . .
  . . . . . . . . . . . . . . . . . . . .
Battery 5: Status=[0xhh], Temp = [deg]C
           Power=[nn] (10mWh), Current=[mA]mA, Voltage=[mV]mV
Battery 6: Status=[0xhh], Temp = [deg]C
           Power=[nn] (10mWh), Current=[mA]mA, Voltage=[mV]mV
```

28) Pyld1 On
29) Pyld1 Off

These options turn ON or OFF the power to Payload #1 (Forward). There is no text response.

Note: At power-up or re-boot, the Payload 1 power is OFF.

30) Pyld2 On
31) Pyld2 Off

These options turn ON or OFF the power to Payload #2 (Aft). There is no text response.

Note: At power-up or re-boot, the Payload 2 power is OFF.

32) Show Rudder Compliance
33) Set Rudder Compliance

Shows or sets the rudder compliance threshold. Used by LRI for engineering purposes only.

34) Set GPS Alarm Threshold

Sets the "GPS Alarm Threshold". This is the count of continuous invalid GPS fixes detected before the "GPS Not Functioning" alarm is set.

The default value is 12. The maximum value that can be set is 255.

Request:

```
Enter New GPS Error Count Threshold:
```

35) Bubble On
36) Bubble Off

Turns ON or OFF the power to the Bubble (Radome). There is no text response.

37) Sub On
38) Sub Off

Turns ON or OFF the power to the Glider (Sub). There is no text response.

39) Set Float/Sub Nav Rate

Sets the rate at which the Float sends navigation (rudder position) commands to the Glider. After the value has been entered, there is no text response.

Request:

```
Enter The New Float Navigation Rate in Seconds (2 --> 120):
```

40) Show Float/Sub Nav Rate

Shows the rate at which the Float sends navigation commands to the Glider.

Output:

Current Float Navigation Rate is once every [*2 → 120*] seconds

41) Set P, I, Sat. values

Sets the Proportional, Integral and Saturation values for PI control. For use by LRI only.

Request:

```
Enter the P and I Gains and the integral Saturation Limit:
```

42) Show P, I, Sat. values

Shows the Proportional, Integral and Saturation values. Also shows whether PI control is being used. For use by LRI only.

Output:

PID [*state*]: P Gain [*val*], I Gain [*val*], Integral Saturation Limit [*val*]

where:

[*state*] is "Enabled" or "Disabled"

43) PID Navigation On
44) PID Navigation Off

Enables or disables navigation using the PI filter parameters. There is no next response.

45) Sub Pyld On
46) Sub Pyld Off

Turns ON or OFF power to the Glider (Sub) Payload if present.

47) Ir. Modem Always On

Sets the power for the Iridium Modem to always be ON.

48) Ir. Modem Idle=>Off

Allows the Iridium Modem to power OFF when not in use.

49) Show Ir. Mode

Displays the current power mode for the Iridium Modem

50) Cur. Ir. Mode=Default

Sets the default power state of the Iridium Modem (i.e. when the system restarts) to whatever the current mode is (set by 47 or 48).

Note: Unless otherwise set, the default state on a vehicle is that the modem is always ON

51) I-PIB On
52) I-PIB Off

Turns ON or OFF power to the Internal Payload (weather, water speed and AIS).

54) Set XBee Default

Sets the XBee radio as the default communication channel (i.e., the channel over which telemetry is sent(.

55) Set Ir. Default

Sets the default communication channel (i.e. the path over which telemetry is sent) to the Iridium Modem. Iridium is always set to this channel at start-up.

91) Reset Pyld File Xfer

If a file transfer to a payload is in progress or has stalled, or if the user wants to reset it, this command will clear any active file transfers to payloads

92) Set Default Radio (Rev 3 only)
93) Set Default Modem (Rev 3 only)

These commands are used only for the Rev 3 hardware. This hardware will support either a 9522B modem or a 9601 for the satellite connection. It will eventually support either an XBee connection or a GPRS connection for the short range radio link and the user can specify which is in the system.

Requests:

```
Enter 0 for XBee and 1 for GPRS:
Enter 0 for 9601 and 1 for 9522:
```

94) Show Error Counts

Shows the counts for all errors since the last re-boot. The output shown below is abbreviated.

Note: The output response shown below is abbreviated.

```
Field Name                  | Count
--------------------------|------
Sub RX Timeout              | [cnts]
Sub RX CRC                  | [cnts]
Sub RX Overflow             | [cnts]
Sub RX Incomplete           | [cnts]
Sub Download Fail           | [cnts]
Sub Program Fail            | [cnts]
Sub Low Power               | [cnts]
Sub Local Power             | [cnts]
Sub Bad Heading             | [cnts]
Sub Magnetometer            | [cnts]
Sub Accelerometer           | [cnts]
Sub Pressure Fail           | [cnts]
Sub Brd Temp Fail           | [cnts]
Sub H2O Temp Fail           | [cnts]
Sub Cal Eeprom Bad          | [cnts]
Sub Nav Eeprom Bad          | [cnts]
. . . .
Charge Twist Over Current   | [cnts]
Bubble RX Timeout           | [cnts]
Bubble RX CRC               | [cnts]
Bubble RX Overflow          | [cnts]
Bubble RX Incomplete        | [cnts]
Bubble Download Fail        | [cnts]
Bubble Program Fail         | [cnts]
Bubble GPS Off              | [cnts]
. . . .
GPS Invalid Fix             | [cnts]
GPS Invalid Checksum        | [cnts]
GPS RX Timeout              | [cnts]
Leak Alarm Triggered        | [cnts]
Shield Fault Triggered      | [cnts]
Iridium SBDWB               | [cnts]
Iridium SBDRB               | [cnts]
Iridium SBDI                | [cnts]
Iridium SBDI Overflow       | [cnts]
Iridium SBDI Timeout        | [cnts]
Iridium IMEI                | [cnts]
Iridium Queue Full          | [cnts]
Iridium Signal Quality      | [cnts]
```

95) Show WayPoint Table

Displays a complete table of all possible Waypoint values. A line of all dashes indicates that no coordinates have been assigned to that Waypoint number.

Note: The output response shown below is abbreviated. The actual output is 256 lines long.

Output:

```
 Num |  Latitude     | N/S |  Longitude    |E/W
-----|-------------- |----|-------------- |---
#001 |  [deg] [mm.mmmm]  |[N,S]|  [deg] [mm.mmmm]  |[E,W]
#002 |-------------- |----|-------------- |---
#003 |-------------- |----|-------------- |---
#004 |  [deg] [mm.mmmm]  |[N,S]|  [deg] [mm.mmmm]  |[E,W]
#005 |  [deg] [mm.mmmm]  |[N,S]|  [deg] [mm.mmmm]  |[E,W]
#006 |  [deg] [mm.mmmm]  |[N,S]|  [deg] [mm.mmmm]  |[E,W]
#007 |  [deg] [mm.mmmm]  |[N,S]|  [deg] [mm.mmmm]  |[E,W]
  . . . . . . . . . . . . . . . . . . . .
  . . . . . . . . . . . . . . . . . . . .
#253 |-------------- |----|-------------- |---
#254 |-------------- |----|-------------- |---
```

96) Show IMEI

Shows the 15-digit International Mobile Equipment Identity number for the Iridium modem in the C&C Drybox.

Output:

```
Modem IMEI: [15 digits]
```

97) Show Versions

Shows the versions of all the Vehicle software and hardware. The Build number is a 4-digit value. Each version numbers are "major.minor", using two 2-digit numbers.

Output:

```
Float Software Version [nn.nn] Build [mmmm]
Rudder Software Version [nn.nn] -- Hardware Version [nn.nn]
Radome Software Version [nn.nn] -- Hardware Version [nn.nn]
Sensor Module Software Version [nn.nn] -- Hardware Version [nn.nn]
Battery Charger Software Version [nn.nn] -- Hardware Version [nn.nn]
```

98) XBee Off

Turns OFF the XBee. If this is done, the XBee can only be turned ON again by using an Iridium command. Thus, a confirmation is requested.

Request:

```
Are You Sure You Want To Turn Off The XBee (Y/N):
```

99) Reboot Float

Re-boots the main C&C processor. All Vehicle parameters will revert to the default values, and all Waypoint values will be lost. Thus, a confirmation is requested.

Request:

```
Are You Sure You Want To Reboot The Float (Y/N):
```

100) Menu

Re-displays the complete menu command list shown in 11.1.

11.3 Status Response

Simply hitting the Enter key without entering a number will get the response shown below.

```
Position: [Lat °] [min.dddd] [N/S]  [Long °] [min.dddd] [E/W]
Float Heading: [deg] -- Float Temp: [°C] -- Float Pressure: [kPa]
Sub Heading: [deg] -- Sub Temp: [°C] -- Sub Pressure: [kPa]
Sub Rudder Position: [deg] (*) -- Sub Rudder Driver Position: [deg] (*)
[course description (**)]
```

[*] The rudder position is indicated using symbols as follows:

| → Centered
^/ → Left
\^ → Right

[**] Possible course descriptions are:

```
Performing Sequential Course - Loop: Now heading to Waypoint [wpt #]
Performing Custom Course: Now heading to Waypoint [wpt #]
Holding Fixed Rudder Position [degrees]
Holding Fixed Heading [degrees]
```

11.4 Menu Error Responses

Shown below are the messages given in case of a Menu entry error.

```
This option is not supported yet!
Position **INVALID**: [Lat °] [min.dddd] [N,S]  [Long °] [min.dddd] [E,W]
One (or more) of the waypoints entered did not exist!
The waypoint requested does not exist!
```

This Page Intentionally Left Blank

12. TroubleShooting

12.1 Overview

On-Shore troubleshooting falls into two general categories:

1) Fixing non-functioning systems
2) Diagnosing and correcting error conditions

This chapter focuses on those kinds of problems that are likely to be encountered by someone developing and/or integrating payloads on the Wave Glider. Refer to the LRI *User Manual* for troubleshooting of on-water operational problems.

12.1.1 XBee and Iridium Interfaces

In general, most development and systems integration will be done using the XBee interface. However, it can be helpful to also have the Iridium interface available in order have an alternate method of communication for troubleshooting.

12.1.2 General Checks

Since the Vehicle is fully electric, many problems can be traced back to connectors and/or batteries. Thus the first checks for any problem should be:

1) Check that at least one of the Solar Panels is connected. Connecting either solar panel is what turns on the electrical system. The visible light is activated when power is turned on.
2) If the problem appears to be a connectivity issue, check that the connectors are tight. It is possible that the hold down sleeve is cross-threaded or otherwise jammed and the connector is not fully seated. If you're not sure, undo and re-install the connector.
3) Make sure that the electrical cables are plugged into the correct connectors. The payload and Umbilical connectors on the top of the C&C box are identical, and cross-connecting should not cause damage, but data communications will be incorrect.
4) Use the XBee or Iridium to get the Power Status and verify that the System voltage is at least 14 Volts. If not, the batteries need re-charging. Refer to section 4.3 “External Charging".

12.1.3 Re-Booting

Generally, re-booting the system is a solution of last resort. Re-booting may temporarily fix the problem, but not address the root cause.

A re-boot can be done via an XBee command (ref. 11.1) or. If you have Admin privileges, you can issue the Float Reboot command (ref. 7.4.7) via the Dashboard.

A “hard” re-boot can be initiated by powering the Vehicle OFF (remove the power key) (disconnect the solar panels]), waiting 20 seconds, and then turning the Vehicle back ON.

12.1.4 Contacting Liquid Robotics

If you've tried the solutions suggested here, and the problem persists, gather up the data and contact LRI, as described in 12.4 “ Contacting Liquid Robotics Inc.".

12.2 Problems and Responses

12.2.1 No XBee Connection

The XBee should power up when the Vehicle electronics are first turned on, or when the system is re-booted. Also, recall that XBee operates only at 115,200 bps.

Responses:

1) Verify that the serial port speed is set to 115,200 bps on the computer terminal software.
2) Check that the dongle for the XBee is fully plugged into the computer.
3) Turn on the XBee connection using the Dashboard Iridium link.
4) Power off. Wait 20 seconds. Then power-on again.

12.2.2 No Iridium Connection

The Iridium connection is initiated by the Vehicle at the Telemetry intervals. The interval can be set for an especially long time (as much as 24 hours). Also, if the batteries are low, the Vehicle may revert to the "Low Battery" telemetry rate, which is typically longer than the standard rate (to conserve battery power).

Also, the Iridium requires a clear view of the sky to operate. Being near a window is generally not adequate.

Responses:

1) Check that the Vehicle power is ON.
2) Check that the Iridium antenna (bubble) has a clear view of the sky.
3) Check that the correct Vehicle is selected on the Dashboard (ref. User Manual).
4) Refresh the Dashboard Web Page.
5) Use XBee to get the IMEI and verify that it is the correct one for the Vehicle.
6) Use the Dashboard to check the strength of the last-sent Telemetry data. The column "ISS" indicates the Iridium Signal Strength, from 0 to 5. A Level of zero may indicate a signal problem.

12.2.3 No GPS Reception

Like the Iridium, the GPS requires a view of the sky, though it can sometimes get an adequate fix if near a large window.

Responses:

1) Check that the Vehicle power is ON.
2) Check that the GPS antenna (bubble) has a reasonably clear view of the sky.
3) Check that Bubble Power has been turned on.

12.2.4 LED Light Won't Turn ON/OFF

The visible LED lights turn on whenever the system is powered on. If it fails to do this, there is probably an electrical failure in the C&C box, requiring repair. If the light was on at power-up, but fails to turn off, or if it turns off, but won't go back on, then it is probably a software or communications issue.

Responses:

1) Check that other commands (e.g., rudder position) work.
2) If using XBee, try using the Dashboard

12.2.5 IR Light Won't Turn ON/OFF

The IR light is not visible, even in a dark room. Typically, it can be seen on the view screen of a digital camera. It can also be checked using an IR "Viewing Card".

If it's verified that the light is not turning on, try the responses described in 12.2.4 "LED Light Won't Turn ON/OFF".

12.2.6 No Float Payload Power From C&C

When the system is powered up (or re-booted), power to the payloads is turned OFF. The power needs to be explicitly turned ON

Also, the Max current that can be supplied to any payload is limited to 1 Amp. Any attempt to use more current will result in power to that payload being shut off.

Responses:

1) Check that the payload is connected to the correct connector on the C&C Drybox.
2) Check the connectors on C&C and Payload Dryboxes.
3) Check for an over-current alarm.
4) Check the power status for low batteries.

12.2.7 No Sub Payload Power From C&C

As with the Float payloads, power is off when the Vehicle power is turned on. Also the max current that can be supplied to both the rudder and Sub payload combined is 1 Amp.

Responses:

1) Verify that rudder can move. If not, check both the Float and Rudder Module Umbilical connectors. Also check the Umbilical for cracks or tears.
2) Check for an over-current alarm.
3) Check the power status for low batteries.

12.2.8 No C&C-to-Payload RS-232 Communications

Lack of RS-232 communications can be caused by a problem with the C&C electronics, the Payload electronics, and/or the connection between them.

If communications were previously established, but no longer work, the best response is to try re-booting the system.

The responses below assume that communications has never been established using a specific payload.

Responses:

1) Check for correct cable connection to C&C Drybox.
2) Check that C&C has turned on power to the Payload.
3) Check for an over-current alarm.
4) Check payload electronics UART.

12.2.9 Float Continuously Re-Boots

This indicates a serious electronics problem. Contact LRI.

12.2.10 Sub Continuously Re-Boots

This indicates a serious electronics problem. Contact LRI.

12.3 Interpreting C&C Alarms

The Alarm indicator bits are returned in the first two bytes of the Telemetry Type 0x06 data packet. The interpretation for each Alarm bit is:

0 → No alarm
1 → Alarm set

The alarms are described below

12.3.1 Sub Leak Alarm (Byte 0 | bit 0)

Water leak was detected in the Rudder Module. This should never occur unless the Vehicle is in the water. The Rudder Module is factory-sealed and not field-repairable.

Responses:

1) If the Glider is under water, retrieve the Vehicle as soon as possible.
2) If the Glider is not under water, try re-booting. If the alarm persists, there is likely an electronics error.

12.3.2 Float Leak Alarm (Byte 0 | 1)

Water leak was detected in the C&C Drybox. This should never occur unless the Vehicle is in the water. The C&C Drybox is factory-sealed and not field-repairable.

Responses:

1) If the Float is in the water, retrieve the Vehicle as soon as possible.
2) If the Float is out of the water, try re-booting. If the alarm persists, there is likely an electronics problem in the C&C Drybox.

12.3.3 Sub-to-Float Comms Alarm (Byte 0 | 2)

The C&C computer detected a data comm error from the Glider.

Responses:

1) Check the electrical connectors on the C&C Drybox and the Rudder Module.
2) Check the Umbilical for cracks or tears.
3) Try re-booting. If the error occurs again, contact LRI.

12.3.4 Payload Alarm (Byte 0 | 3)

General problem with a payload. It may be a communications, electrical or environmental issue.

Responses:

1) Check the connectors between the C&C Drybox and each Payload Drybox.
2) Verify that the Payload is not drawing more than 1 Amp of current.
3) Try re-booting. If the alarm occurs again, contact LRI.

12.3.5 Float Battery Low Alarm (Byte 0 | 4)

C&C battery level is below the specified "Float Low_Battery Power".

Responses:

1) Power off the Vehicle and re-charge the batteries.
2) If the alarm persists after re-charging, contact LRI.

12.3.6 Umbilical Fault Alarm (Byte 0 | 5)

There is an integrity problem (a leak or broken conductor) on the Umbilical.

Responses:

1) If the Vehicle is in the water, retrieve as soon as possible.
2) If the Vehicle is out of the water, check the Umbilical for holes or tears.
3) Try re-booting. If the alarm persists, there is likely an electronics error.

12.3.7 GPS Not Functioning (Byte 0 | 7)

The Vehicle has not received a valid GPS fix for two minutes.

Responses:

1) Verify that the Float (the Bubble) has a clear view of the sky.
2) Try a re-boot, then wait for at least 2 minutes.

12.3.8 Sub Reboot Alarm (Byte 1 | 1)

A serious error occurred on the Rudder Module computer, causing a re-boot.

Responses:

1) Check for a "Sub Leak Alarm" or a "Sub-to-Float Comms Alarm".
2) Try powering OFF, then ON. If the error occurs again, contact LRI.

12.3.9 Float-to-Sub Comms Error (Byte 1 | 2)

The Sub computer detected a data comm error from the C&C computer.

Responses:

1) Check the electrical connectors on the C&C Drybox and the Rudder Module.
2) Check the Umbilical for cracks or tears.
3) Try re-booting. If the error occurs again, contact LRI.

12.3.10 Float Temperature Threshold Exceeded (Byte 1 | 4)

The temperature in the C&C Drybox is above the "Float Temperature Threshold". If the Float is not exposed to high outdoor temperatures, this indicates an electronics or battery problem.

Responses:

1) Immediately power the Vehicle OFF.
2) Check the outside air temperature. If the C&C interior is noticeably warmer than outside, there may be a serious internal problem. Call LRI and do not re-power the unit.

12.3.11 Float Pressure Threshold Exceeded (Byte 1 | 5)

The pressure in the C&C Drybox is above or below the "Float Pressure Threshold" values

Responses:

1) Remove the valve cover on the top of the C&C Drybox, and check the pressure. Adjust as necessary.
2) If the pressure values are within the acceptable range, re-boot and see if the alarm goes off.

12.3.12 EPROM Error (Byte 1 | 6)

During booting, the system transfers data from EEPROM to Flash memory, using error checking. This alarm is activated if there is an error in the data transfer.

Responses:

1) Try re-booting. If the error occurs again, contact LRI.

12.3.13 Float Reboot Alarm (Byte 1 | 7)

An error occurred on the C&C computer, causing a re-boot.

Responses:

1) If the error occurs continually, contact LRI.

12.4 Contacting Liquid Robotics Inc.

If you cannot determine or correct the cause of a problem, contact LRI.

To expedite resolving the problem, make a brief list beforehand with the following information:

1) Wave Glider Model Number
2) Any special or added payloads on the Vehicle
3) Time/date when the problem occurred
4) Location of Vehicle when problem occurred
5) Initial indication of problem
6) Any other indications or unusual readings
7) Attempts at corrective actions and their results

For time-critical problems, call LRI at:

1 (650) 493-6300

or

1 (888) 574-4574

and select the option for Customer Service

For non-critical problems, send the above information via email to:

support@liquidr.com

This Page Intentionally Left Blank

13. Notes

13.1 Glossary

ACK – Abbreviation for "Acknowledged" during data transmission. This is response to indicate to the sender that a sent message was received correctly.

Acknowledged (Alarm) – The alarm was reset by a User.

Active (Alarm) – The alarm was issued, but no other action has been taken.

Acoustic Wave and Current Profiler (AWAC) – A sensor that characterizes and records wave motion.

Admin (Role) – A User who has all the permissions required to go through the day-to-day administration of the WGMS, add users, manage customizations, and deal with data.

Alarm – A message either from the Vehicle C&C or from the Shore-based software indicating that a problem has occurred.

Alternate Iridium Relay – A connection to the Iridium system that bypasses the standard Iridium Server.

Arrival Distance – Distance from a Waypoint at which the C&C computer assumes that the Vehicle has "arrived" at the Waypoint.

Audit Records – Record of changes or updates made to a Vehicle and/or its parts.

Automatic Identification System (AIS) – A VHF radio system that automatically broadcasts the type and location of a seaborne vessel.

Broadband Global Area Network (BGAN) – A global satellite Internet network with telephony using portable terminals.

Channel (Packet Reflector) – The type of Subscriber to a Packet Reflector. Types are Command Source, Command Sink, Output Source and Output Sink.

Checkbox – A yes/no selection box in a Dialog Box. Checking one box does not preclude checking another. (ref. Radio Button)

Coastal Explorer – A mapping service that provides a DVD with nautical maps. Used as an alternative to Google Maps.

Command and Control Drybox – The center-mounted Drybox in the Wave Glider Float. Contains the main control computer and communications equipment.

Command Packets – Data packets containing Commands sent to the Vehicle.

Commands – Messages sent to the Vehicle for control or to request status.

Commercial, Off-The-Shelf (COTS) – Standard (as opposed to custom-made) components.

Compiled Views – Option to compile different views to speed up the response of the GUI.

Course – The path a Vehicle is attempting to follow. (ref. Track)

Create (Permission) – A Role permission that allows creation of a Vehicle-related system parameter.

Custom Relay – Software that provides special communications services between a Vehicle and a Packet Reflector.

Dashboard – The Home Page of the User Interface to the Wave Glider Management System.

Data Analyst (Role) – A User who looks at existing Vehicle and WGMS data and can download database data.

Database (WGMS) – The part of the WGMS that stores configuration and communications data for one or more Vehicles.

Delete (Permission) – A Role permission that allows deletion of a Vehicle-related system parameter.

Dialog Box – An on-screen box that typically requests data to be entered.

Direct IP – An individual computer that is allowed directly communicate to the Internet. Typically used for single-user machines

Drybox – A sealed watertight container on either the Glider or the Float. Typically used for housing electronics.

Errors – Messages sent by the Vehicle or the Shore-based software indicating a problem.

Escalation List – A prioritized list of people to contact in case of an error.

Events – Types of Alarms.

Float – The part of the Wave Glider Vehicle that rides on the surface of the water.

Glider – The submarine portion of the Vehicle, connected by the Umbilical to the Float.

Global Positioning System (GPS) – A satellite-based broadcasting system that allows determination of geographical coordinates.

Global System for Mobile Communications (GSM) – A worldwide standard for mobile telephone communication systems.

Google Maps – A mapping service provided by Google.

Guest User (Role) – Someone who can view the status of Vehicles that have been specifically marked for display to guests.

Home Page – The initial page that comes up on the browser after logging onto the WGMS system.

Hyperion – An alternate name for the WGMS Kernel

Icon – A small pictogram on a GUI. Typically, clicking on the Icon will start some other action.

International Mobile Equipment Identity (IMEI) – A 15-digit number used by the Iridium system to identify handsets and modems that can connect with the Iridium system.

Iridium – A constellation of satellites that provides world-wide voice and data transmission. Wave Glider Vehicles use the SBD form of transmission.

Iridium Relay – Software/hardware that provides direct communications to the Iridium system, bypassing an Iridium Server.

Iridium Network Server – A server maintained by the Iridium corporation that provides access to the Iridium system via the Internet.

Iridium Subscriber Unit (ISU) – Describes a stand-alone piece of hardware (phone or modem) that can communicate directly with the Iridium system.

IP Address – Equivalent to IP Number.

IP Number – Internet Protocol Number. A combination of numbers that defines the destination of TCP or UDP data. Of the form xxx.xxx.xxx.xxx, where “xxx” represents a number from 1 to 255 (e.g., 10.10.5.145).

Kernel (WGMS) – That part of the WGMS software that does the primary communications control and database management.

LRI-Hosted WGMS – A deployment option where an Organization uses LRI’s servers and WGMS software to operate their Vehicle.

Master (Role) – A User who manages Vehicle provisioning, mission planning and operations, as well as payload and Web services development.

Menu – Allows selection of one or more items from a list.

Metadata Version – Software version number. For reference use when contacting LRI about a software problem.

Mission – A set of Vehicle operational actions, either by itself, or in cooperation with other Vehicles.

Mobile-to-Mobile – A deployment option where a single computer with its own copy of WGMS communicates directly with a Vehicle, using either XBee or an Alternate Iridium Relay.

NAK – Abbreviation for “Not Acknowledged” for a data transmission. This is response to indicate to the sender that a sent message was received, but cannot be processed.

National Marine Electronics Association (NMEA) – An industry association that creates standards for, and certifies electronics used in marine applications.

NMEA Convert – Option to use the NMEA mapping available from Coastal Explorer. Typically used when there is no Internet connection to the WGMS.

Operation Polygon – A user-defined closed region in the water that defines limits on where a Vehicle should go.

Org – Abbreviation for Organization.

Org Internal Name – The name of the Organization used within the WGMS database. This is not necessarily the same as the "Organization Name" used on the Dashboard UI.

Organization – An entity that operates one or more Wave Glider Vehicles. Each Organization has its own data, and that data cannot be accessed by any other Organization.

Organization Name – The "public" name used on the Dashboard to identify the Organization. This name may be different than the "Org Internal Name".

Output Packets – Packets sent from the Vehicle. These include telemetry, ACK/NAK responses and (optional) payload-created data.

Packet Reflector – Software that is used to route Commands and Outputs between the Vehicle and the WGMS.

Page – A browser window that displays data, and typically has links to other pages and data.

Part Identifications – Lists of hardware and software version numbers.

Payload Drybox – A Drybox that is optionally installed on/in a Vehicle. The Float has provision for two internal dryboxes. Payload Dryboxes may also be mounted externally on the Glider.

Permissions – For each Role, the specifications of what system parameters are allowed to be Created, Read, Updated and/or Deleted.

Privileges – Refers to the combined set of Permissions for someone who has been granted multiple Roles.

Protocol (Internet) – Method by which data is sent from one computer to another on the Internet.

Radio Button – One of a set round-icon selection options. Only one option can be selected. (ref. Checkbox).

Raw Outputs – Data packets received from the Vehicle as-is, with no processing.

Read (Permission) – A Role permission that allows read-only viewing of WGMS system parameters.

Roles – A set of Permissions granted to a person that allows them specific types of access to the WGMS system.

Root (Role) – A super-user who has full access to all aspects of the Vehicle and WGMS. This user will be either an LRI developer or a customer under close direction from LRI.

Secure Iridium – A special DoD-only gateway to the Iridium system. (ref. EMSS)

Secure Sockets Layer (SSL) – A protocol for transmitting private documents over the Internet.

Self-Corrected (Alarm) – An Alarm situation corrected itself.

Self-Hosted WGMS – A deployment option where an Organization uses their own independent copy of WGMS to operate their Vehicle(s).

Shore-Side (Software) – Refers to any software that does not reside on the Vehicle. WGMS software in a laptop on a boat is still considered "Shore-Side".

Short Burst Data (SBD) – An Iridium data communications method that utilizes small data packets.

Status Data – Information sent via Telemetry indicating the operational status (position and system health) of the Vehicle.

Storefront Name – Legacy term for "Org Internal Name".

Structured Query Language (SQL): A language used for retrieval and management of database information. (Pronounced "Sequel").

Sub – An alternate (and discouraged) term used to refer to the sub-marine Glider part of the Vehicle.

Subscriber (Packet Reflector) – A Source or Sink that communicates with a Packet Reflector.

System Administrator – Someone who handles the day-to-day management of the WGMS and its users.

System Page – An alternate page used to display all System information for an Organization's Vehicle(s).

Tab – A label in the WGMS GUI. Tabs may be placed near the top of a window, or on the left side. Clicking on a tab selects different data on a Page.

Telemetry – Data sent from the Vehicle on a periodic basis. Typically sent every two minutes.

Tether – An alternate (and incorrect) term for the Umbilical.

Tool Bar – A bar, typically along the top of a window, that contains data entry boxes and icons.

Track – The path a Vehicle actually traverses in the water when it is trying to stay on a Course.

Transmission Control Protocol (TCP) – The standard protocol for transmitting data packets over the Internet (ref. User Datagram Protocol).

User Datagram Protocol (UDP) – An Internet data transfer method that provides higher transmission speed than TCP, but does no handshaking or error checking.

Umbilical – The flexible connecting cord between the Float and the Glider. Contains power and signal conductors, as well as strengthening elements.

Uniform Resource Locator (URL) – A string of names separated by a forward slash. Specifies an address on the Internet (e.g., http://www.nmea.org). A URL maps to an IP address.

Update (Permission) – A Role permission that allows updating of a Vehicle-related system parameter.

Vehicle – An alternate term used to describe the Wave Glider Float + Umbilical + Glider system.

Vehicles Page – An alternate page used to display all Vehicle information for an Organization.

Wave Glider Management System (WGMS) – The combined software package that is used to manage communications to and from a Vehicle.

Waypoint – A geographical point defined by Lat/Long coordinates. Used to define a Vehicle's Course or holding position.

Web Services – Automated information services that are conducted over the Internet, using standardized formats and protocols.

WiFi – A short-range high-bandwidth wireless communications method.

XBee – A short-range low-to-medium bandwidth radio communications method.

13.2 Abbreviations and Acronyms

AIS.......Automatic Identification System

AISyG..Automatic Identification System Gateway

API.......Application Programming Interface

AWAC..Acoustic Wave and Current Profiler

BGAN...Broadband Global Area Network

COTS....Commercial, Off-The-Shelf

CRUD....Create-Read-Update-Delete

EMSS.....Enhanced Mobile Satellite Service

GPS.......Global Positioning System

GSM......*Groupe Spécial Mobile* (Global System for Mobile Communications)

GUI.......Graphical User Interface

IMEI......International Mobile Equipment Identity

IP...........Internet Protocol

ISU........Iridium Subscriber Unit

LRI........Liquid Robotics, Inc.

M2M.....Mobile-to-Mobile

MTBF....Mean Time Between Failures

NMEA...National Marine Electronics Association

This Page Intentionally Left Blank

Appendix A – XML, SOAP and Entities

A.1 XML

XML both defines and is the structure of the WGMS.

A.1.1 Entities

The primary components of the WGMS are "Entities". Entities are XML strings used to either store or structure data. Entities can contain other entities. In the WGMS, entities are used to:

1) Define the structure of the GUI (using metadata stored in the Database).
2) Process Iridium data packets received from the Iridium Web Server.
3) Define the structure of data stored in the Database.
4) Create, delete and update other entities.

A.2.1 XML Data Types

All XML data types are strings of ASCII characters. If numerical values are contained in the string, the number values are stored as characters. For example, the number **`-12,345`** is stored as the string **`"-12345"`**.

The strings can be used to represent the following data types in entity description functions:

`int` – A signed 32-bit integer value.

`long` – A signed 64-bit integer value.

`string` – An ASCII string. This string may contain any of the data types described here, as well as XML data definitions.

`base64Binary` – An XML schema standard used to store binary data. Groups of 6 bits are encoded as one of 64 ASCII characters.

`boolean` – A single-character string, either "0" or "1". The represented value does not necessarily indicate "True" or "False".

`dateTime` – A date and time stamp value string, using the ISO 8601 format:

YYYY-**MM**-**DDThh**:**mm**:**ss**

where:

YYYY = four-digit year
MM = two-digit month (01=Jan, 02=Feb, etc.)
DD = two-digit day of month (01 through 31)
T = separator character
hh = two digit hour (00 through 23)
mm = two digit minute (00 through 59)
ss = two digit second (00 through 59)

Note: The time zone is the zone of the message originator.

A.3.1 Response Strings

All functions return data in the form of a character string. The type of string will be defined by the XML record type.

If a function fails to execute, the response string will contain the two-character "**`-1`**" string.

A.2 Soap Envelopes

The XML envelopes can be in SOAP 1.1 or 1.2 format. In the structure examples shown below:

length: Automatically-created indicating the content length
type: Data type appropriate to the parameter being defined
string: A URL Encoding String

A.1.1 SOAP 1.1 Format

```
POST /webservices/entityapi.asmx HTTP/1.1
Host: gliders.liquidr.com
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://schemas.LiquidRobotics.com/WebServices/EntityName"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
          xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
   <soap:Body>
      <EntityName xmlns="http://schemas.LiquidRobotics.com/WebServices">
         <ParamName>type</ParamName>
         <ParamName>type</ParamName>
         <...>
      </EntityName>
   </soap:Body>
</soap:Envelope>
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
          xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
   <soap:Body>
      <EntityNameResponse xmlns="http://schemas.LiquidRobotics.com/WebServices">
         <EntityNameResult>string</EntityNameResult>
      </EntityNameResponse>
   </soap:Body>
</soap:Envelope>
```

A.2.1 SOAP 1.2 Format

```
POST /webservices/entityapi.asmx HTTP/1.1
Host: gliders.liquidr.com
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
          xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
   <soap12:Body>
      <EntityName xmlns="http://schemas.LiquidRobotics.com/WebServices">
         <ParamName>type</ParamName>
         <ParamName>type</ParamName>
         <...>
      </EntityName>
  </soap12:Body>
</soap12:Envelope>
HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
          xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
   <soap12:Body>
      <EntityNameResponse xmlns="http://schemas.LiquidRobotics.com/WebServices">
         <EntityNameResult>string</EntityNameResult>
      </EntityNameResponse>
   </soap12:Body>
</soap12:Envelope>
```

A.3 Management System Function Summary

Below is a summary of the functions that are used to manage Entities.

Function Name	Description	Ref
`<CreateEntity>`	Create a new instance of an existing Entity	A.1.1
`<CreateLoginSession>`	Create and returns an encrypted session cookie	A.2.1
`<DeleteEntity>`	Delete a previously-created entity	A.3.1
`<GetEntityColumnMetadata>`	Retrieve from database a single column of metadata for a specified Entity type	A.4.1
`<GetEntityColumnMetadataSet>`	Retrieve from database a multi-column dataset for the specified entity type	A.5.1
`<GetEntityColumnSelectMetadataSet>`	Retrieve from database a named column dataset for the specified entity type	A.6.1
`<GetEntityMetadata>`	Retrieve from the database the complete metadata for the specified entity type	A.7.1
`<GetEntitySet>`	Retrieve from the database specific metadata for the entity type	A.8.1
`<GetRelatedEntitySet>`	Retrieve from the database related metadata for the entity type	A.9.1
`<GetVehicleCommands>`	Retrieve from the database date-specific Vehicle command data	A.10.1
`<GetVehicleCrumbs>`	Retrieve from the database date-specific Vehicle crumb data	A.11.1
`<GetVehiclePowerCtlStatuses>`	from the database date-specific Vehicle crumb data	A.12.1
`<ReadEntity>`	Retrieve a single row of entity data from the database	A.13.1
`<UpdateEntity>`	Update a specific entity field in the database	A.14.1

Table 13-1: Management System Function Summary

A.4 Function Descriptions

A.1.1 Create an Entity

Creates a new instance of an existing Entity.

Name:

`<CreateEntity>`

Parameters:

`<entityType>` **`int`** (A record type value assigned by the system)

`<entityFieldXml>` **`string`** (An XML string that defines the Entity)

Response:

`<CreateEntityResponse>`

`<CreateEntityResult>` **`string`** (The ID# of the Entity)

A.2.1 Create a Login Session

Creates and returns an encrypted session cookie for use in consecutive Web Services commands.

Name:

`<CreateLoginSession>`

Parameters:

`<login>` **`string`** (User's login name)

`<password>` **`string`** (User's password)

`<orgName>` **`string`** (Assigned organization name)

Response:

`<CreateLoginSessionResponse>`

`<CreateLoginSessionResult>` **`string`** (Cookie text string)

A.3.1 Delete an Entity

Deletes a previously-created entity.

Name:

`<DeleteEntity>`

Parameters:

`<entityType>` **`int:`** (System-assigned record-type value)

`<id>` **`long:`** (ID value of the Entity)

Response:

`<DeleteEntityResponse>`

`<DeleteEntityResult>` **`string`** ("0" → deleted; "-1" → failed)

A.4.1 Get Entity Column Metadata

Retrieves from the database the column metadata for a specified Entity type.

Name:

`<GetEntityColumnMetadata>`

Parameters:

`<entityType>` **int** (System-assigned record type value)

`<columnName>` **string** (Name of the column to be retrieved)

Response:

`<GetEntityColumnMetadataResponse>`

`<GetEntityColumnMetadataResult>` **string**
(The column metadata of the specified entity type)

A.5.1 Get Entity Column Metadata Set

Retrieves from the database a single-column dataset for the specified entity type.

Name:

`<GetEntityColumnMetadataSet>`

Parameters:

`<entityType>` **int** (System-assigned record type value)

Response:

`<GetEntityColumnMetadataSetResponse>`

`<GetEntityColumnMetadataSetResult>` **string**
(All metadata fields of the specified entity type)

A.6.1 Get Entity Column Select Metadata Set

Retrieve from the database a named column dataset for the specified entity type.

Name:

`<GetEntityColumnSelectMetadataSet>`

Parameters:

`<entityType>` **int** (System-assigned record type value)

`<columnName>` **string** (Name of the column to be retrieved)

Response:

`<GetEntityColumnSelectMetadataSetResponse>`

`<GetEntityColumnSelectMetadataSetResult>` **string**
(All selects associated with the vehicle type)

A.7.1 Get Entity Metadata

Retrieve from the database the table metadata for the specified entity type.

Name:

```
<GetEntityMetadata>
```

Parameters:

```
<entityType> int      (System-assigned record type value)
```

Response:

```
<GetEntityMetadataResponse>
  <GetEntityMetadataResult> string
                            (The table metadata for the specified entity type)
```

A.8.1 Get an Entity Set

Retrieve from the database specific sorted and filtered metadata for the entity type.

Name:

```
<GetEntitySet>
```

Parameters:

```
<entityType> int             (System-assigned record type value)
<columnXml> string           (Columns to get)
<pageNumber> int             (Page number to get)
<pageSize> int               (Number of entries per page)
<sortColumn> string          (Column to use for sorting results)
<sortDirection> string       (Direction for sorting)
<dataFilter> string          (Search string in the sorting column)
```

Response:

```
<GetEntitySetResponse>
  <GetEntitySetResult> string    (The sorted and filtered metadata)
```

A.9.1 Get Related Entity Set

Retrieve from the database related metadata for the entity type.

Name:

`<GetRelatedEntitySet>`

Parameters:

`<entityType>` **`int`**	(System-assigned record type value)
`<id>` **`long`**	(ID value of the entity)
`<relatedEntityType>` **`int`**	(Related entity type)
`<columnXml>` **`string`**	(Columns to get)
`<pageNumber>` **`int`**	(Page number to get)
`<pageSize>` **`int`**	(Number of entries per page)
`<sortColumn>` **`string`**	(Column to use for sorting results)
`<sortDirection>` **`string`**	(Direction for sorting)
`<dataFilter>` **`string`**	(Search string in the sorting column)

Response:

`<GetRelatedEntitySetResponse>`

`<GetRelatedEntitySetResult>` **`string`**

(The sorted and filtered metadata of the related entity)

A.10.1 Get Vehicle Commands

Retrieve from the database date-specific Vehicle command data.

Name:

`<GetVehicleCommands>`

Parameters:

`<fromTime>` **`dateTime`**	(Begin date for command data)
`<toTime>` **`dateTime`**	(End date for command data)
`<vehicleId>` **`int`**	(ID number for vehicle)

Response:

`<GetVehicleCommandsResponse>`

`<GetVehicleCommandsResult>` **`string`** (Vehicle command data)

A.11.1 Get Vehicle Crumbs

Retrieve from the database date-specific Vehicle crumb data. Data can be telemetry-only or all the combined crumb data.

Name:

<GetVehicleCrumbs>

Parameters:

<combined> **boolean** ("0" → Telemetry-only; "1" → All data combined)

<fromTime> **dateTime** (Begin date for crumb data)

<toTime> **dateTime** (End date for crumb data)

<vehicleId> **int** (ID number for vehicle)

Response:

<GetVehicleCrumbsResponse>

<GetVehicleCrumbsResult> **string**
(combined list of command, telemetry and alarms)

A.12.1 Get Vehicle Power Control Statuses

Retrieve from the database date-specific Vehicle crumb data.

Name:

<GetVehiclePowerCtlStatuses>

Parameters:

<fromTime> **dateTime** (Begin date for crumb data)

<toTime> **dateTime** (End date for crumb data)

<vehicleId> **int** (ID number for vehicle)

Response:

<GetVehiclePowerCtlStatusesResponse>

<GetVehiclePowerCtlStatusesResult> **string** (Status data)

A.13.1 Read Entity

Retrieve a single row of entity data from the database.

Name:

`<ReadEntity>`

Parameters:

`<entityType>` **`int:`** (System-assigned record-type value)

`<id>` **`long:`** (ID value of the Entity)

Response:

`<ReadEntityResponse>`

`<ReadEntityResult>` **`string`** (single row in the database table)

A.14.1 Update an Entity

Update a specific entity field in the database.

Name:

`<UpdateEntity>`

Parameters:

`<entityType>` **`int:`** (System-assigned record-type value)

`<id>` **`long:`** (ID value of the Entity)

`<fieldChangeXml>` **`string`** (New value for the Entity type/ID)

Response:

`<UpdateEntityResponse>`

`<UpdateEntityResult>` **`string`** ("0" → updated; "-1" → failed)

Appendix B – Iridium/Relay Header Data

B.1 Iridium Ship-to-Shore Data Packets

Ship-to-Shore Relay packets are command responses, telemetry or alarms. These packets have a 51 byte header, followed by up to 205 bytes of response data.

Iridium Ship-to-Shore Field Name	Byte #	Length (bytes)	Value	Format
Protocol Revision	0	1	1	Big Endian
Message Length	1-2	2	Total length - 3	
MO_Header IEI	3	1		
MO_Header Length	4-5	2		
CDR Reference	6-9	4	"Msg1"	
IMEI	10-24	15		
Session Status	25	1		
MOMSN	26-27	2		
MTMSN	28-29	2		
Time of Session	30-33	4		
MO_Payload IEI	34	1		
MO Payload Length	35-36	2	(# Level 2 bytes]	
MO Location Info IEI	37	1		
MO Location Info Length	38-39	2		
MO Lat/Long	40-46	7		
CEP Radius	47-50	4		
Level 2 Packet (Response or Telemetry)	51-255	≤ 205		Little Endian

where:

Protocol Revision – Iridium protocol revision number.

Message Length – Number of bytes in the remainder of the packet.

MO_Header IEI – Mobile Originated Header Information Element Identifier.

MO_Header Length – Mobile Originated Header length in bytes.

CDR Reference – Call Detail Record Reference.

IMEI – International Mobile Receiving Identity. 15 digit code of the receiving modem.

Session Status – Status of the connection link. Values larger than 2 indicate a problem.

MOMSN – Mobile Originated Message Sequence Number.

MTMSN – Mobile Terminated Message Sequence Number.

Time of Session – Time of day of session.

MO_Payload IEI – Mobile Originated Payload Information Element Identifier.

MO Payload Length – Mobile Originated Payload Length. (Number of Level 2 bytes)

MO Location Info IEI – Mobile Originated Location Information Element Identifier.

MO Location Info Length – Mobile Originated Location Information Length.

MO Lat/Long – Mobile Originated Latitude and Longitude data. See below.

CEP Radius – Circular Error Probability Radius. Distance (in Kilometers) around the center point within which the IMEI is located with an 80% probability of accuracy.

Level 2 Packet – Telemetry or Command response. Ref. Chapter 8, "Vehicle Responses" and Chapter 9, "Vehicle Telemetry".

B.1.1 MO Lat/Long Data

The Mobile Originated Latitude and Longitude data is encoded in a 7-byte value as follows:

Field Name	Bytes	Units	Value Description
Latitude N-S Bit Longitude E-W Bit	1	N/S/E/W	Bit 6: 0→North, 1→South Bit 7: 0→East, 1→West
Latitude (degrees)	1	Degrees	[0 to 90] (dec) [0x00 to 0x5A] (hex)
Latitude (minutes)	2	Minutes / 1,000	[0 to 59,999] (dec) [0x0000 to 0xEA5F] (hex)
Longitude (degrees)	1	Degrees	[0 to 180] (dec) [0x00 to 0xB4] (hex)
Longitude (minutes)	2	Minutes / 1,000	[0 to 59,999] (dec) [0x0000 to 0xEA5F] (hex)

B.2 Relay Ship-to-Shore Packets

Ship-to-Shore Relay packets are similar (but not identical) to the Iridium Ship-to-Shore packets. These packets also have a 51 byte header, followed by up to 205 bytes of response data.

Though these packets are structured like Iridium packets, note that an ID number and IP Address/Port replace the IMEI. Also, many of the fields are zero-valued placeholders.

Relay Ship-to-Shore Field Name	Byte #	Length (bytes)	Value
Protocol Version	0	1	See below
Message Length	1-2	2	Total length - 3
MO Header IEI	3	1	0
MO Header Length	4-5	2	0
CDR Reference	6-9	4	See below
WGMS ID	10-13	4	Ref. 6.1.8
IP Address	14-17	4	
IP Port	18-21	4	
(not used)	22-24	3	0
Session Status	25	1	0
MOMSN	26-27	2	0
MTMSN	28-29	2	0
Time of Session	30-33	4	0
MO Payload IEI	34	1	0
MO Payload Length	35-36	2	(# Level 2 bytes]
MO Location Info IEI	37	1	0
MO Location Info Length	38-39	2	0
MO Lat/Long	40-46	7	0
CEP Radius	47-50	4	0
Level 2 Packet	51-255	≤ 205	

The Relay Packet significant field types are:

Protocol Version – Number indicates the type of protocol to be expected in the Level 2 packet.

Protocol	Value	Notes
Iridium (plug-In)	1	[1]
XBee	2	
Serial	3	
802.11	4	
Alternate Iridium	5	[2]

Notes:

[1] Protocol 1 – Refers to the Iridium modem in the C&C. Uses SBD transmit/receive.

[2] Protocol 5 – Refers to another device on the Vehicle that sends and receives data in the Iridium format. This data transmission does <u>not</u> use SBD.

CDR Reference – A 4-character text sequence that indicates the type of message recipient, as follows:

Protocol	Value
Iridium (plug-In)	"Msg1"
XBee	"Xbee"
Serial	"Ser1"
802.11	"WiFi"
Alternate Iridium	"Irid"

Table 13-2: CDR Reference Numbers

WGMS ID – The unique 4-byte identification number that is stored in the Vehicle's Non-Volatile memory. Ref. 6.1.8.

IP Address – Internet Protocol destination address of the message recipient.

IP Port – Internet Protocol port number of the destination address.

MO Payload Length – Length of the Level 2 Content payload.

Level 2 Content – Data in the format specified by the "Protocol Version" number.

B.3 Iridium Shore-to-Ship Data Packets

A Shore-to-Ship Iridium packet carries commands to the Vehicle. It has 30 bytes of header content and can carry up to 135 bytes of command data.

Iridium Shore-to-Ship Field Name	Byte #	Length (bytes)	Default Value	Format
Protocol Revision	0	1	1	Big Endian
Message Length	1-2	2	Total length – 3	
MT_Header IEI	3	1	65	
MT_Header Length	4-5	2	21	
CDR Reference	6-9	4	"Msg1"	
IMEI	10-24	15		
MT_Disposition Flags	25-26	2	0	
MT_Payload IEI	27	1	66	
MT_Payload Length	28-29	2	[# Level 2 bytes]	
Level 2 Command Packet	30-164	≤ 135		Little Endian

Table 13-3: Iridium Shore-to-Ship Packet Format

where:

Protocol Revision – Iridium protocol revision number.

Message Length – Number of bytes in the remainder of the packet.

MT_Header IEI – Mobile Terminated Header Information Element Identifier.

MT_Header Length – Mobile Terminated Header length in bytes.

CDR Reference – Call Detail Record Reference.

IMEI – International Mobile Receiving Identity. 15 digit code of the receiving modem.

MT_Disposition Flags – Mobile Terminated Disposition Flags.

MT_Payload IEI – Mobile Terminated Payload Information Element Identifier.

MT_Payload Length – Number of bytes in the payload packet that follows.

Level 2 Packet – Command Data. Ref. Chapter 7, "Commands".

B.4 Relay Shore-to-Ship Packets

A Shore-to-Ship Relay packet carries commands to the Vehicle. It has 30 bytes of header content and can carry up to 135 bytes of command data.

Iridium Shore-to-Ship Field Name	Byte #	Length (bytes)	Default Value
Protocol Version	0	1	See below
Message Length	1-2	2	Total length – 3
MT Header IEI	3	1	65
MT Header Length	4-5	2	21
CDR Reference	6-9	4	See below
WGMS ID	10-13	4	Ref. 6.1.8
IP Address	14-17	4	
IP Port	18-21	4	
(not used)	22-24	3	0
MT Disposition Flags	25-26	2	0
MT Payload IEI	27	1	66
MT Payload Length	28-29	2	[# Level 2 bytes]
Level 2 Command Packet	30-164	≤ 135	

Table 13-4: Iridium Shore-to-Ship Packet Format

where:

The **Protocol Version** and **CDR Reference** numbers are the same as used for Ship-to-Shore Relay packets.

Protocol Version	Value
Iridium (plug-In)	1
XBee	2
Serial	3
802.11	4
Iridium (alternate)	5

CDR Reference	Value
Iridium (plug-In)	“Msg1”
XBee	“Xbee”
Serial	“Ser1”
802.11	“WiFi”
Iridium (alternate)	“Irid”

WGMS ID – The unique 4-byte identification number that is stored in the Vehicle's Non-Volatile memory. Ref. 6.1.8.

IP Address – Internet Protocol address of the command originator.

IP Port – Internet Protocol port number of the command originator.

Level 2 Packet – Data in the format specified by the "Protocol Version" number.

Appendix C – CRC-16 Calculation

A packet's error-check field contains a 16-bit CRC calculated from the bytes in a data packet. The specific bytes used for the calculation will depend upon the type of packet. However, the computational algorithm will be the same.

Here is an example of the algorithm written in C#:

```
public static CRC16 ComputeFromArray(byte[] buffer, uint offset, uint count)
{
    UInt16 crc = 0;

    for (uint idx = offset; count-- > 0; ++idx)
        crc = ComputeIncrementally(crc, buffer[idx]);

    return new CRC16(crc);
}

public static UInt16 ComputeIncrementally(UInt16 crc, byte a)
{
    int i;

    crc ^= a;

    for (i = 0; i < 8; i++)
    {
        if ((crc & 1) == 1)
        {
             crc = (ushort)((crc >> 1) ^ 0xA001);
        }
        else
        {
             crc = (ushort)(crc >> 1);
        }
    }

    return crc;
}
```

This Page Intentionally Left Blank