Wave Glider™ Interface Control Description

Wave Glider™ Interface Control Description

Publication date February 2013
Copyright © 2013 Liquid Robotics, Inc.

Legal Notice

Confidential computer software. Valid license from Liquid Robotics, Inc. required for possession, use or copying. Consistent with FAR 12.211 and 12.212, Commercial Computer Software, Computer Software Documentation, and Technical Data for Commercial Items are licensed to the U.S. Government under vendor's standard commercial license.

Trademarks

Liquid Robotics™, LRI, Wave Glider, Wave Glider Management System, WGMS, and SHARC are worldwide trademarks of Liquid Robotics, Incorporated. Other products and company names mentioned herein may be the trademarks of their respective owners.

Patents & Inventions

The Wave Glider mechanism is patented under US Patent Number 7,371,136 and other patents pending. All aspects of the Wave Glider product have been developed at private expense by Liquid Robotics, Inc.

Warranty

The information contained herein is subject to change without notice. The only warranties for Liquid Robotics, Inc. products and services are set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. Liquid Robotics, Inc. shall not be liable for technical or editorial errors or omissions contained herein.

Revision History

Document Part Number	Revision	Publication Date
030-04438	A	February 2013

Table of Contents

1. Sensor Integration Overview 1
2. Wave Glider System Overview 2
    Wave Glider Major Components 2
    Float Assembly 2
    Solar Panels 4
        Solar Panel Specification 4
        Shading and Shade Compensation 4
    Command & Control Dry Box 5
    Forward and Aft Payload Dry Boxes 6
    Float to Glider Umbilical 7
    Glider 9
    Towfish Options 10
        Gateway Towfish 11
        4" Modular Towfish 11
        Towfish Tether 12
    Electrical 14
        C&C Dry Box Connections 14
        Rudder Module Connections 14
        Electrical Connectivity 15
    Communications and Control 15
        Iridium Satellite Network 15
        Relay Communications 16
        Payload Communications 16
        XBee Radio Link 16
        Navigation 16
3. Float Mechanical 18
    Float Overall Dimensions 18
    Float Attachment Hard-points 19
    Float Payload Bays 19
        Float Forward Payload Bay 20
        Float Aft Payload Bay 20
    C&C Dry Box 21
    Payload Dry Boxes 22
        Forward Payload Dry Box 22
        Aft Payload Dry Box 23
        Short Aft Payload Dry Box 24
        Forward and Aft Payload Dry Box Lid Mounting 25
        Forward and Aft Payload Dry Box Lid With Payload Interface Board(s) 26
        Forward and Aft Payload Dry Box Lid With Dual Connector Blanks 26
    Float Payload Considerations 27
        Maximum and Minimum Weight 27
        Center-of-Gravity Range 27
        Float Sail Area and Masts 28
        Deck mounting 28
        Towing 29
4. Glider Mechanical 30
    Glider Dimensions and Payload Attachment Points 30
    Glider Payload Considerations 31
        Drag 31
        Glider Payload Bay 32
        Glider Ballast Weights 32

Wing-Payload Interference 32
Balance and Flotation 33
Poor Sail Area Locations 33
Acceptable Sail Area Locations 34
Trailing Wires 34
5. Electrical Specifications 36
Physical Connectivity 36
Solar Panels 37
Float C&C 37
Umbilical 37
Rudder Module 37
Water Speed Sensor 37
Payload Dry Boxes and Tow Bodies 37
PEP Port 38
Weather Station and Light Bar 39
Payload Connectivity 39
Payload Communication Connectivity 39
Payload Power Connectivity 40
GPS and PPS Connectivity 40
Payload Maximum Currents 40
Maximum Total Current Output 41
WGMS Side Connectivity 41
Iridium Connectivity 41
XBee Connectivity 41
Command and Control Dry Box 42
Power Key 42
Indicator Light 42
Unused Connections 42
Standard Electrical Connections 42
PAYLD1, PAYLD2, and PAYLD3 Connectors 42
UMBILICAL Connector 44
AUX PWR Connector 46
SOLAR 1 AFT and SOLAR 2 FORWARD Connectors 47
WEATHER Connector 49
SPEED Connector 50
LIGHT Connector 51
Power System 52
Batteries 52
Core System Power Draw 52
Payload Regulator 52
Power Input 53
External Charging 53
Overview 53
Charger Connections 53
Liquid Robotics-Supplied Charger 54
Custom-Made Charger 54
Power Considerations 55
Sustainable Payload Power 55
C&C Dry Box Power Draw 55
Battery System Power 56
Battery Charge Level 56
Battery Discharge Characteristics 56
Rudder Module 57
Overview 57

Unused Connections 57
Rudder Umbilical Connector 58
Rudder Payload Connector 59
6. Vehicle Communications and Control 61
Communications 61
Iridium-Internet Link 61
Mobile-to-Mobile Link 61
XBee Radio Link 62
Payload Communications 62
Communications Management 63
GPS 63
C&C Functions 64
Telemetry and Polling 64
Command Processing 64
Payload Message Handling 65
Command and Message Responses 65
Data Types 65
Telemetry 65
Commands and Messages 65
Responses 66
Communication Packets 66
Addresses 68
XBee 68
Wave Glider Management System 68
WGMS Overall Structure 68
Command Communications 69
Vehicle Responses 69
WGMS Components 69
User Interface 70
User Application 70
WGMS Kernel 71
Data 71
Com Server 71
Packet Reflector 71
Alternate Kernels 71
Iridium Relay 71
Communication Relays 71
Common WGMS Scenarios 72
Liquid Robotics-Hosted WGMS 72
Mobile-to-Mobile Deployment 73
Customer-Hosted (Self-Hosted) Shared WGMS 75
Payload Development and Testing Using XBee 75
Alternate WGMS Configurations 76
Customized Relay 76
Multiple Copies of WGMS 76
7. Communication Methods 77
Overview 77
Mobile-Originated Communications 77
Mobile-Terminated Communications 77
Commands and Responses 77
Vehicle Unique ID 78
Transaction ID 78
Iridium (Direct IP) and Alternate Iridium Relay Communications 78
Short-Burst Data 78

Packet Levels 78
Iridium Message Buffering 79
XBee Radio Communications 79
Xbee Relay 79
Debug Console 80
Operational Considerations (Channel and PAN ID) 81
Communication Data Formats 81
Little-Endian Byte Format 82
Waypoint Values 82
Heading Values 82
Distance Values 82
Time Values 82
Heading Values 83
Distance Values 83
Temperature Values 83
File Names 83
Navigation Modes 83
Roles and Permissions 84
8. Commands 89
Command Data Packet Format 89
Command Summary 90
9. Vehicle Responses 93
ACK/NAK Protocol 93
ACK/NAK Packet Format 93
ACK Data Type Numbers 94
NAK Data Type Numbers 94
Unsolicited Response Type Numbers 95
Relay Messages 95
ACK Response Data 95
ACK - Status with Waypoint Read-Back (0X10) 95
ACK - Status Return (0x11) 96
ACK - Vehicle Identification Data (0x12) 97
ACK - No Payload Response (0x13) 99
ACK - Power Controller Status (0x14) 100
ACK - Rudder Control Data (0x15) 101
ACK - Read File Block (0x1B) 102
ACK - Detailed File List Data (0x1C) 103
ACK – Unique ID (0x1E) 104
ACK – File at Destination (0x1F) 105
ACK – File List Data (0x31) 105
NAK Response Data 106
NAK - Invalid Command (0x22) 106
NAK – Invalid Data (0x22) 106
NAK – File Error (0x24) 106
Restricted ACK Responses 107
Special ACK Data Types 107
ACK – Debug Buffer (0x16) 108
ACK – Sub Heading Buffer (0x17) 108
ACK – PID Integral Cache (0x18) 108
ACK – PID Operating Parameters (0x19) 108
Unsolicited Messages 108
New File Available (0x60) 108
Send Console Output to Client (0x1D) 109
Relay-Only Message 109

Relay Status (0x70) 110
10. Vehicle Telemetry 111
Telemetry Packet Format 111
Telemetry Types 111
Telemetry Buffering 112
NAL Tracker Telemetry 0x03 112
Shout Tracker Telemetry 0x05 113
Standard Telemetry Data 0x06 115
Alarm Bytes 117
Float Status Byte 117
Telemetry - Power Controller Status 0x7 118
DB3.5 Power Control Status 0x08 118
Twist Status 0x09 122
Relay Status 0x70 123
System Information 0x71 124
Single Payload Packet 0x80 124
Weather Station Packet (Type 1) 125
Payload Status Packet (Type 2) 125
Payload AISyG Status Packet (Type 16) 126
Payload AISyG Static Record (Type 17) 127
Payload AISyG Position Report (Type 18) 128
1.1.1 Sensor Sample Record (Unparsed) 129
Fluorometer (C3) Data Type 96 129
Compressed CTD Data Type 112 129
Compressed CTD Data 129
Compressed CTD With Oxygen Sensor Data 130
Payload ECO Puck Sensor Data Type 123 130
Payload CStar Sensor Data Type 124 132
Payload Benthos Modem Data Type 125 133
Payload Datawell MOSE Sensor Data Type 126 133
Payload VEMCO VR2C Data Type 127 134
Payload ADCP Data Type 128 135
Payload Aquatracka Data Type 130 136
Payload WHOI Micro Modem Type 131 136
Harmony Payload Type 132 137
Cefas OrbComm Payload Type 134 138
LISST Payload Type 135 138
Payload CO2 Sensor Type 144 139
Payload Thermistor Array Data Type 192 139
Multi-Payload Packet 0x81 140
Payload Packet 0x91 140
11. Payload Communications Protocol 141
Communication Electronics 141
Payload Interface Board (PIB) 141
Personality Modules 141
Payload Board Devices 142
Configuration Scenarios 142
Communication Method 143
Payload Communications Requirements 144
Serial Communication Parameters 144
Payload Data Packet Format 144
Message Routing 145
Polling 145
Enumeration Request and Response 146

ACKs and NAKs 146
Protocol Message Format 146
Start-of-Frame Character (Section 1) 147
Message Size (Section 2) 147
Destination Address (Section 3) 147
Source Address (Section 4) 147
Transaction ID (Section 5) 147
Message Type (No ACK) (Section 6) 148
Message Data (Section 7) 148
CRC-16 (Section 8) 148
SOF and ESC Characters and Escaping the Message Body 148
Addresses 149
Address Format 149
Board ID 149
Task ID 149
Other Considerations 150
Message Routing 150
Timeouts and Message Usage 150
Payload Message Types 151
Payload Message Packet Descriptions 152
Payload Header Abbreviation 152
Generic Payload NAK Data Packet (0x0000) 153
Generic Payload ACK Data Packet (0x0001) 154
Request ID/Enumerate (0x0010) 155
ACK Reply to Enumerate Message 156
Device Information Block Format 156
C&C Telemetry (0x0015) 157
Request Status (0x0022) 158
ACK Reply to Request Status 158
Status Block, Version = 1 159
Send/Forward Message (0x0024) 159
ACK for Forward 160
ACK for Send 160
Power Status and Control (0x0030) 161
ACK Response to Power Status and Control 161
Request Queued Message (0x0040) 162
NAK Response to Request Queued Message 162
ACK Response to Request Queued Message 163
ACK/NAK Queued Message (0x0041) 163
Message Protocol Use Cases 164
Normal Polling Update Time and Location 164
Start-up Identification Broadcast 164
Device Receives File from WGMS 165
Send File from Device to WGMS 166
Device Receives Power Off Notification 168
Send Message to Console 168
12. Payload Communications Hardware 169
Sensor Management Computer (SMC) 169
Overview 169
Console Port 169
Full RS-232 Ports (UART0 and UART1) 170
Configurable UART Ports (UART2/3, UART4/5, and UART6/7) 171
GPS/PPS Port 172
Power Ports 173

CAN Port 173
USB Ports (not USB OTG) 174
USB OTG Port 174
Ethernet Port 174
SD Card Expansion Port 174
SPI Expansion Port 176
McBSP Expansion Port 177
I2C Ports 178
Payload Interface Board (PIB) 179
Payload Interface Board Architecture 179
Payload Interface Board Interfaces 181
Pressure and Temperature Sensor 182
Reset Button 182
Power Indicator 182
Humidity and Temperature Sensor 182
Payload Port Interface – J7 183
AUX Lines - J6 184
Atmel ISP – J5 185
Atmel JTAG – J4 186
Personality Interface - J1, J2, J3 187
Liquid Leak Detector - J8 191
Glider (Sub) Payload Interface Board (SubPIB) 192
Sub Payload Interface Board Architecture 192
Sub Payload Interface Board Interfaces 193
Pressure and Temperature Sensor 193
Reset Button 193
Power Indicator 193
Humidity and Temperature Sensor 193
Payload Port Interface - J7 194
Optional Power Input Port Interface - J10 (510-01581 only) 195
AUX Lines - J6 196
GPS and Secondary AUX Lines - J9 197
Atmel ISP - J5 198
Atmel JTAG - J4 201
Personality Interface - J1, J2, J3 203
Liquid Leak Detector - J8 208
P-COM Personality Boards 210
Personality Interface - J1 210
Output Connector - J4 213
Configuration Switch - SW1 215
Switching Regulator - U1 219
Linear Regulator - U3 220
13. XBee Communications 221
XBee Command Menu 221
Menu Error Responses 225
14. Troubleshooting (Sensor Integration Issues) 226
Overview 226
XBee and Iridium Interfaces 226
General Checks 226
Re-Booting 226
Contacting Liquid Robotics 227
Problems and Responses 227
No XBee Connection 227
No Iridium Connection 227

No GPS Reception. 227
LED Light Will Not Turn ON/OFF 228
IR Light Won't Turn ON/OFF 228
No Float Payload Power From C&C 228
No Sub Payload Power From C&C 228
No C&C-to-Payload RS-232 Communications 229
Float Continuously Re-Boots 229
Sub Continuously Re-Boots 229
Interpreting C&C Alarms 229
Sub Leak Alarm (Byte 0 | bit 0) 230
Float Leak Alarm (Byte 0 | bit 1) 230
Sub-to-Float Comms Alarm (Byte 0 | bit 2) 230
Payload Alarm (Byte 0 | bit 3) 230
Float Battery Low Alarm (Byte 0 | bit 4) 231
Umbilical Fault Alarm (Byte 0 | bit 5) 231
GPS Not Functioning (Byte 0 | bit 7) 231
Sub Reboot Alarm (Byte 1 | bit 1) 231
Float-to-Sub Comms Error (Byte 1 | bit 2) 231
Float Temperature Threshold Exceeded (Byte 1 | bit 4) 232
Float Pressure Threshold Exceeded (Byte 1 | bit 5) 232
EPROM Error (Byte 1 | bit 6) 232
Float Reboot Alarm (Byte 1 | bit 7) 232
Contacting Liquid Robotics, Inc. 232
A. Command Descriptions 234
Unrestricted Commands 234
No Command (0x0000) 234
Hold Current Position (0x0001) 234
Hold Station at Waypoint (0x0002) 234
Follow Sequential Waypoint Course (0x0003) 235
Rudder Right/Left/Center (0x0004, 0x0005, 0x0006) 235
Follow Fixed Heading (0x0008) 236
Set Default Communication Channel (0x000A) 236
Set Unique ID (0x000F) 237
Twist Umbilical (0x1007) 237
Retrieve Debug Buffer (0x7000) 237
Retrieve Sub Heading Buffer (0x7001) 238
Retrieve PID Integral Cache (0x7002) 238
Retrieve PID Operating Params (0x7003) 239
New Waypoint (0x8000) 239
Read Unique ID (0x000E) 239
Set Rudder Position (0x1006) 240
Visible Light Off/On (0x2001, 0x2002) 240
IR Light Off/On (0x2003, 0x2004) 240
XBee Comm Off/On (0x2005, 0x2006) 241
Retrieve Status (0x4001) 241
Set Status (0x4002) 241
Retrieve Identification (0x4003) 243
Retrieve Power Status (0x5001) 243
Retrieve Rudder Control (0x5002) 243
Set Rudder Control (0x5003) 244
Set Parameters (0x6000) 245
Set Single Waypoint (0x8000) 247
Read Waypoint (0x8003) 247
Follow Course and Loop (0x8101) 248

Follow Course With Options (0x8102) 248
Add MMSI to Watch List (0x8200) 249
Initialize File Write (0x9000) 249
Write File Block (0x9001) 251
Delete File (0x9002) 251
Read File Request (0x9003) 252
Get Detailed File List (0x9004) 253
Select Program (0x9005) 254
Write Console Input to Vehicle (0x9006) 255
Read File Transaction Closed (0x9007) 256
Get File List (0x9008) 256
Write Console Script (0x9100) 257
Write Console Binary (0x9101) 258
Float Reboot (0xA000) 258
Reserved (0xA51E) 259
Secure On (0xA500) 259
Secure Authenticate (0xA501) 259
Secure Register Client (0xA502) 260
Secure Off (0xA504) 260
Secure Enable Partition (0xA505) 260
Restricted Commands 261
Set Unique ID (0x000F) 261
Set Restricted Parameters (0x6000) 262
Retrieve Debug Buffer (0x7000) 263
Retrieve Sub Heading Buffer (0x7001) 263
Retrieve PID Integral Cache (0x7002) 263
Retrieve PID Operating Parameters (0x7003) 264
Float Reboot (0xA000) 264
B. XML, Soap and Entities 265
XML 265
Entities 265
XML Data Types 265
Response Strings 266
Soap Envelopes 266
SOAP 1.1 Format 266
SOAP 1.2 Format 267
Management System Function Summary 267
Function Descriptions 268
Create Entity 268
Create Login Session 269
Delete Entity 269
Get Entity Column Metadata 269
Get Entity Column Metadata Set 270
Get Entity Column Select Metadata Set 270
Get Entity Metadata 270
Get Entity Set 271
Get Related Entity Set 271
Get Vehicle Commands 272
Get Vehicle Crumbs 272
Get Vehicle Power Control Statuses 273
Read Entity 273
Update Entity 273
C. Iridium and Relay Header Data 275
Iridium Vehicle-to-WGMS Data Packets 275

Mobile Originated Lat/Long Data 276
Relay Vehicle-to-WGMS Packets 276
Iridium WGMS-to-Vehicle Data Packets 278
Relay WGMS-to-Vehicle Packets 279
D. CRC-16 Calculation 281

List of Figures

2.1. Wave Glider Major Components 2
2.2. Float Components Exploded View 3
2.3. Basic Float External Features 4
2.4. Examples of Shading and How It Affects Power Output 5
2.5. C&C Dry Box 6
2.6. Forward and Aft Payload Dry Boxes 6
2.7. Short Aft Payload Dry Box 7
2.8. Umbilical 8
2.9. Glider (Sub) 9
2.10. Glider With Custom Payload Cover 10
2.11. Gateway Towfish 11
2.12. 4" Modular Towfish 12
2.13. 4" Modular Towfish Dimensions 12
2.14. Towfish with Tether 3D 13
2.15. Towfish Tether Dimensions 13
2.16. Connections on C&C Dry Box 14
2.17. Rudder Module Connectors 14
2.18. Basic Electrical System 15
3.1. Float Overall Dimensions 18
3.2. Float Attachment Hard points 19
3.3. Forward Payload Bay Dimensions 20
3.4. Aft Payload Bay Dimensions 21
3.5. C&C Dry Box Dimensions 22
3.6. Forward Payload Dry Box Dimensions 23
3.7. Aft Payload Dry Box Dimensions 24
3.8. Short Aft Payload Dry Box 25
3.9. Forward and Aft Payload Dry Box Lid Underside Dimensions 26
3.10. Forward and Aft Payload Dry Box Lid with PIB 26
3.11. Forward and Aft Payload Dry Box Lid With Two Connector Blanks 27
3.12. Float Max Forward Center-of-Gravity 28
3.13. Float Max Aft Center-of-Gravity 28
3.14. Possible Sail Area Problems 28
3.15. Deck Mounting of Payloads 29
3.16. Towing Behind Float 29
4.1. Glider Overall Dimensions and Attachment Points 30
4.2. Glider Moving Up 31
4.3. Glider Moving Down 31
4.4. Poor Payload Design/Location 31
4.5. Glider with Payload Box 32
4.6. Wing-Payload Interference Area 32
4.7. Checking Glider Underwater Balance 33
4.8. Excessive Forward Payload Sail Area 34
4.9. Excessive Aft Payload Sail Area 34
4.10. Acceptable Payload Sail Area Locations 34
4.11. Poor Trailing Wire Locations 35
4.12. Acceptable Trailing Wire Location 35
5.1. Wave Glider Connection Diagram 36
5.2. Payload Dry Box Electrical Components 38
5.3. Towfish Electrical Components 38
5.4. Payload Interconnectivity and Communication Diagram 39
5.5. Payload Power Routing 40

5.6. GPS and PPS Routing 40
5.7. Vehicle Wireless Connectivity Options 41
5.8. Liquid Robotics External Charger 54
5.9. Solar Power Relative To Time and Location 55
5.10. Battery Charge Level Relative to Voltage 56
5.11. Battery Discharge Time vs. Current Draw 57
5.12. Rudder Module 57
6.1. WGMS to Vehicle Iridium Communications 61
6.2. Mobile-to-Mobile Communications 61
6.3. On-Shore XBee Communications 62
6.4. Payload Communications Connections 62
6.5. Wave Glider Command and Control 63
6.6. GPS Satellite Constellation 64
6.7. Level 3 Iridium or Relay Data Packet Format 66
6.8. Level 3 Encrypted Iridium or Relay Data Packet Format 67
6.9. Level 2 Payload Data Packet Format 67
6.10. Basic Wave Glider Management System 68
6.11. Basic WGMS Components 70
6.12. Liquid Robotics-Hosted WGMS 72
6.13. Multiple Organizations Using Liquid Robotics-Hosted WGMS 73
6.14. Iridium Communications Corp ISU-ISU 74
6.15. Mobile-to-Mobile Configuration 74
6.16. Self-Hosted WGMS Configuration 75
6.17. XBee for Payload Testing 75
6.18. XBee Communications Flow 76
7.1. Level 3, Level 2, and Level 1 Data Packets 79
8.1. Command Packet Format 89
9.1. ACK/NAK Response Packet Format 93
10.1. Level 2 Telemetry Data Packet 111
11.1. Basic Payload Communications 141
11.2. Personality Module Connections 142
11.3. Daisy-Chain of PIBs 142
11.4. Example Payload Hardware Configuration 143
11.5. Payload Data Packet Format 145
11.6. Message Packet Format 147
12.1. SMC Connector Locations 169
12.2. Console Port Connection 170
12.3. GPS/PPS Connections to a PIB 173
12.4. Simple Single PIB with 3 Personality Modules Connected to 3 Devices 180
12.5. Simple Double PIB Connected to 6 Devices 180
12.6. Daisy Chained PIBs Used to Expand Device Count in a Single Dry Box 181
12.7. PIB Connector and Sensor Locations 182
12.8. Payload Port Pin Number Orientation 183
12.9. PIB to Penetrator Example Connector 184
12.10. AUX Lines J6 Pin Number Orientation 184
12.11. Atmel ISP - J5 Pin Number Orientation 185
12.12. Atmel ATJTAGICE2 without the adapter cable 186
12.13. Atmel ATAVRISP2 186
12.14. Atmel JTAG - J4 Pin Number Orientation 186
12.15. Atmel ATJTAGICE2 187
12.16. Personality Connector Locations 188
12.17. Pin Number Orientation 188
12.18. Liquid Leak Detector - J8 Pin Number Orientation 191
12.19. Single Sub PIB in a Towfish Connected to 3 Personality Modules and 3 Devices 192

12.20. Sub Payload Interface Board 510-01086 Connector and Sensor Locations 193
12.21. Payload Port Interface - J7 Pin Number Orientation 194
12.22. Optional Power Port Interface - J10 Pin Number Orientation 196
12.23. Pin Number Orientation 196
12.24. GPS/Secondary AUX Lines – J9 Pin Number Orientation 197
12.25. ISP and JTAG Connector Location 198
12.26. ISP Pin Number Orientation 198
12.27. Liquid Robotics Programming Adapter, Part Number 510-00537 200
12.28. Atmel ATJTAGICE2 without the Programming Adapter or the 10 Pin to 6 Pin Adapter 200
12.29. Atmel ATAVRISP2 without the Programming Adapter 200
12.30. ISP and JTAG Connector Location 201
12.31. JTAG Pin Number Orientation 201
12.32. Liquid Robotics Programming Adapter, Part Number 510-00537 202
12.33. Atmel ATJTAGICE2 without the Programming Adapter 203
12.34. Top Side Personality Connector Locations 204
12.35. Bottom Side Personality Connector Locations 205
12.36. Pin Number Orientation 205
12.37. Leak Detector Location 209
12.38. P-COM Personality Board 510-00097 Rev C 210
12.39. J1 As Viewed From The Bottom 210
12.40. Output Connector - J4 Pin Number Orientation 213
12.41. RS-232 based device connection diagram 214
12.42. RS-485 half duplex based device connection diagram 214
12.43. Output Example Connector 215
12.44. Configuration Switch SW1 With All Switches In The OFF Position 215
12.45. Switching Regulator and Potentiometer 219
12.46. Linear Regulator and Potentiometer 220
13.1. Launching the XBee Console from the XBee Relay 221
13.2. XBee Console Window for Vehicle 41414141 222
13.3. Example: Command 0 System Status Output for Vehicle 41414141 223
13.4. Example: Prompt that Appears when the **Add Waypoint** Command is Entered 223

List of Tables

2.1. Solar Panel Specifications 4
3.1. Float Maximum and Minimum Weight Limits 27
4.1. Minimum and Maximum Total Glider Weight - Underwater 33
5.1. Color and Label Conventions for System Diagram 36
5.2. C&C Dry Box Indicator Light States 42
14.1. C&C Alarms 229

Chapter 1. Sensor Integration Overview

The Wave Glider™ is an autonomous ocean-going platform for deploying custom ocean-to-surface sensor systems. This document provides detailed information on all of the parameters that impact Wave Glider sensor integration. There are a few key parameters that are most likely to determine the feasibility of a sensor integration project. Liquid Robotics recommends evaluating a sensor integration project with respect to these key parameters first:

- Total weight of the sensor system – see the section called “Float Payload Considerations”
- Size of sensors and associated components - see Chapter 3, *Float Mechanical*, Chapter 4, *Glider Mechanical*, and the section called “Towfish Options”
- Mounting requirements for the sensor system - see Chapter 3, *Float Mechanical*, Chapter 4, *Glider Mechanical*, and the section called “Towfish Options”
- Total power consumption for the sensor system - see the section called “Power System”

If a sensor integration project can be adapted within these key parameters, its feasibility is very likely. The other Wave Glider sensor integration parameters are typically easier to comply with.

Chapter 2. Wave Glider System Overview

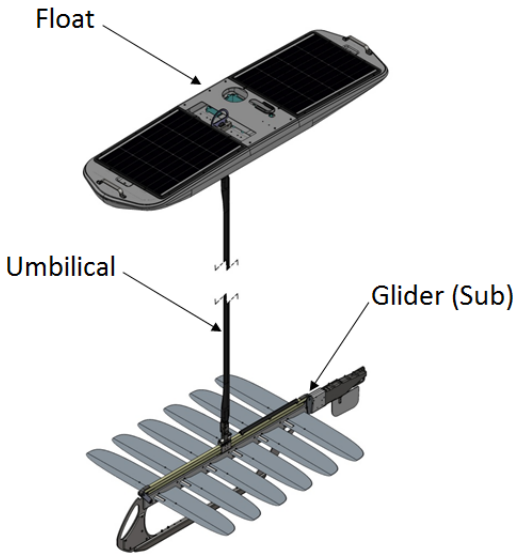
This chapter provides a general overview of the Wave Glider system. It is assumed that you have reviewed the appropriate chapters in the Wave Glider User Manual, and are generally familiar with Wave Glider terminology.

Wave Glider Major Components

The Wave Glider system has three major components:

- **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.
- **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.
- **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 2.1. Wave Glider Major Components

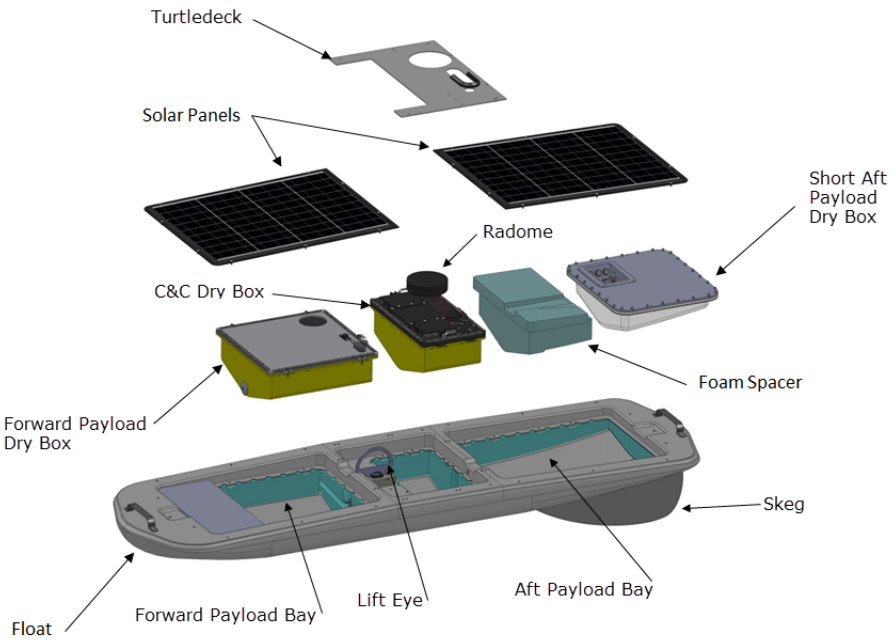


The following sections provide more information on Wave Glider components.

Float Assembly

The Float assembly is illustrated in Figure 2.2, “Float Components Exploded View”..

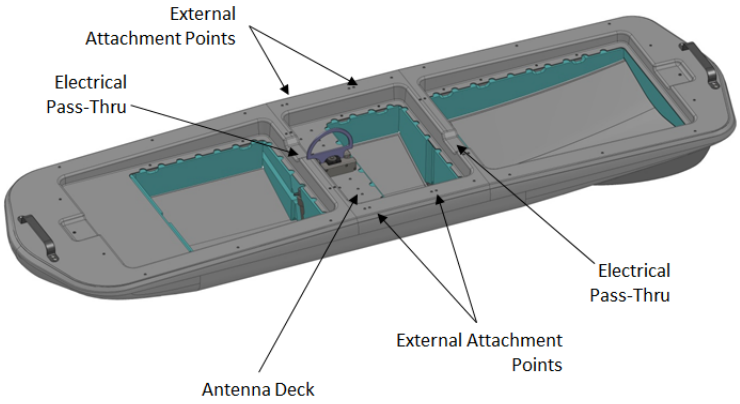
Figure 2.2. Float Components Exploded View



- **Short Aft Payload Dry Box:** Optional watertight container for the aft payload. There are also standard forward and aft payload dry boxes, for more information see the section called “Forward and Aft Payload Dry Boxes”.
- **Aft Payload Bay:** Aft compartment. Shallower than the Forward Payload Bay.
- **C&C Dry Box:** The Command and Control Dry Box contains the vehicle electronics and the batteries. See section 2.4 “Command & Control Dry Box" for additional information.
- **Foam Spacer:** Optional spacer. Size/shape will vary depending on the size of the dry box.
- **Forward Payload Dry Box:** Optional watertight container for the forward payload.
- **Forward Payload Bay:** Forward compartment. Deeper than the Aft Payload Bay.
- **Float:** The single-piece composite hull of the Wave Glider.
- **Lift Eye:** Attach point for lifting an assembled Wave Glider. The Lift Eye connects directly to the Umbilical Attach.
- **Radome:** Part of the C&C Dry Box. Exposed above the Turtledeck. Encloses the Wave Glider Iridium, GPS, and XBee antennas.
- **Solar Panels:** Photovoltaic (“PV”) cells used to charge the electronics batteries.
- **Skeg:** Rear stabilizing fin. This is non-moving and non-adjustable.
- **Turtledeck:** Non-watertight cover for the C&C Dry Box payload bay on the Float.

The Float has some additional features that should be noted:

Figure 2.3. Basic Float External Features



- **Antenna Deck:** A platform that can be used to mount auxiliary communication antennas or external sensors.
- **Electrical Pass Thru:** A cutout for routing electrical cables between the payload bays and the C&C Dry Box, or to forward and aft external payloads.
- **External Attach Points:** Threaded thru-holes that can be used to attach external payloads.

Solar Panels

This section describes the features of the standard Wave Glider solar panels.

Solar Panel Specification

The solar panel laminates are Kyocera KC40T or equivalent.

Table 2.1. Solar Panel Specifications

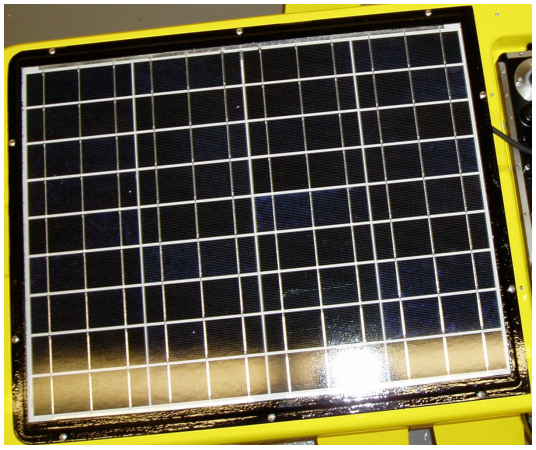
Description	Value	Units	Notes
Laminate Width	520	mm	
Power	40	Watts	
Vmpp	17.4	Volts	Maximum power point voltage
Impp	2.30	Amps	Maximum power point current
Voc maximum	22.2	Volts	Limiting voltage on the input
Isc maximum	5.0	Amps	Limited by the connector rating

Shading and Shade Compensation

Shading is a significant problem for solar panels. A solar panel laminate is typically made up of 36 cells and it only takes one shaded cell to create problems. If a single cell becomes shaded, current flow through the panel is reduced by 50%.

The design of the panels provides connections for bypass diodes that allow the use of half of a panel to provide power when a single cell is shaded. If one or more cells in each half of a panel is shaded, the panel provides zero output. The dividing line between the two halves of each panel is located half way between the forward and aft ends of the panel. The images below show examples of what constitutes shading of a single cell.

Figure 2.4. Examples of Shading and How It Affects Power Output



Shading in general comes from environmental conditions such as cloud cover, and from system components mounted on the vehicle. Shading from environmental conditions is typically temporary and is out of our control. To optimize total power input, all sensors should be mounted below the top of the solar panel. Since this is not always possible, the recommendations below should be followed.

If a tall element such as a mast is required, attempt to design it to be narrower than a single cell. This does not eliminate the shading effects, but only reduces the output proportional to the shading width.

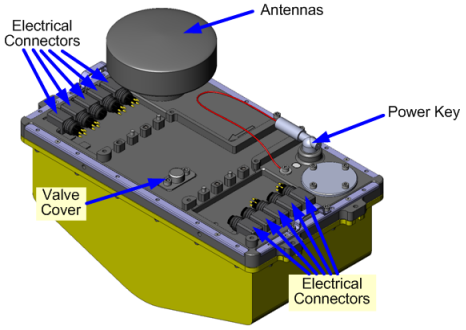
If it is a wide element, attempt to design it to be as short as possible. With this design, shading is not a problem during the middle of the day.

To maximize power input when there are shading elements on a vehicle, move to a course that is predominately North-South so the shadows fall off the sides of the vehicle for most of the day. In the East-West navigational direction, one of the panels may always suffer from shading.

Command & Control Dry Box

The C&C Dry Box is a factory sealed module that mounts in the center bay of the Float. This Dry Box contains the computing and communications equipment. It also contains the batteries that store energy from the solar panels.

Figure 2.5. C&C Dry Box



The main external components of the C&C Dry Box are:

Antennas: GPS, Iridium, and XBee antennas.

Electrical Connections: Ten watertight electrical connectors.

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

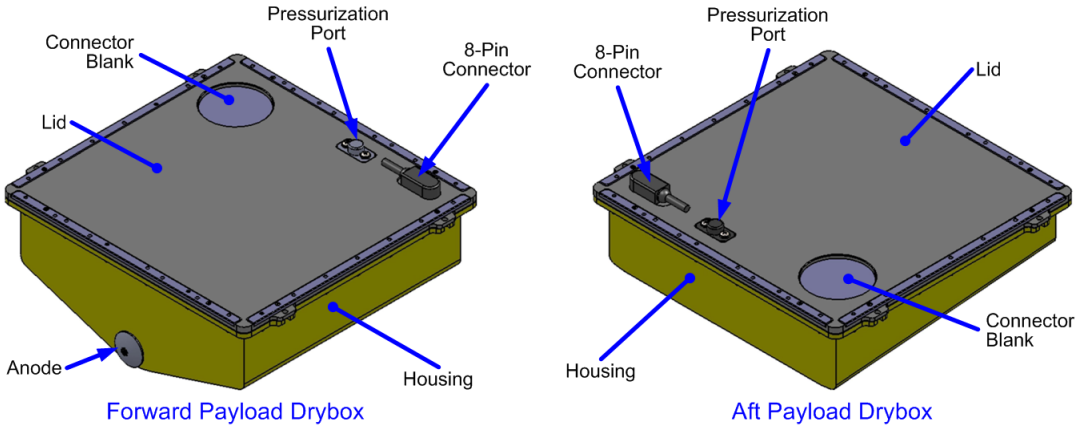
Valve Cover: Protective cover for the box pressurization valve.

Forward and Aft Payload Dry Boxes

The Forward and Aft Payload Dry Boxes are typically used to house sensor related electronics payloads. The dry boxes shown below are representative of those available from Liquid Robotics Inc. The forward and aft dry boxes have different shapes, because the aft payload bay is both longer and shallower than the forward bay.

The payload dry boxes are described in more detail in Chapter 3, *Float Mechanical*.

Figure 2.6. Forward and Aft Payload Dry Boxes



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

Anode: Bolted-on sacrificial anode.

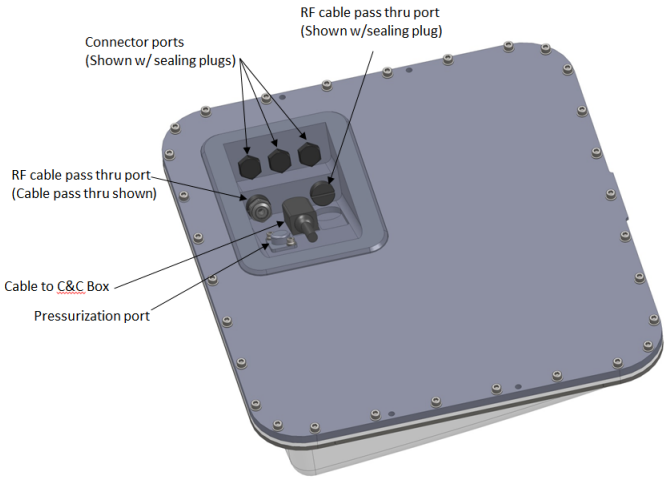
Connector Blank: Removable plastic panel for adding additional connectors. An optional lid with two connector panels is available.

Lid: The top of the Dry Box. 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 dry box.

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

Figure 2.7. Short Aft Payload Dry Box



The Short Aft Payload Dry Box is typically used when a Teledyne RDI ADCP sensor is integrated with the Wave Glider.

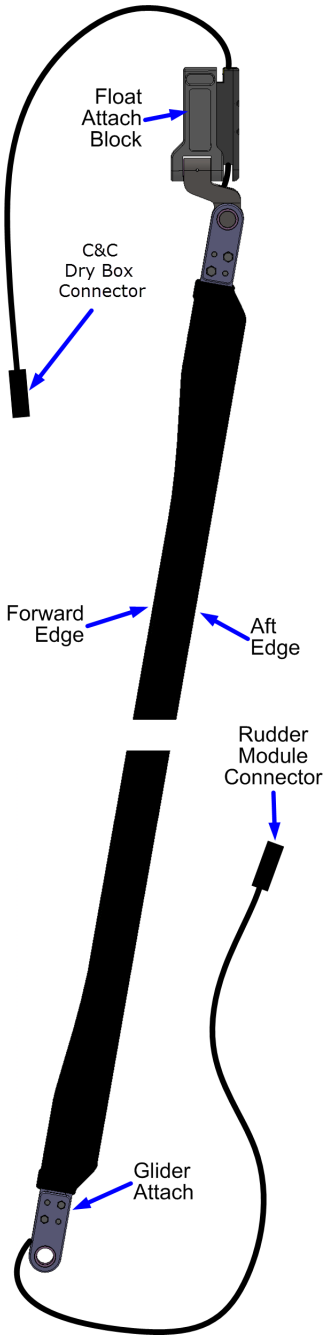
Note that the payload dry boxes are designed to be watertight only, and are not intended to withstand significant water pressure.

Float to Glider Umbilical

The Umbilical (also referred to as the "Tether") is a flexible cable that connects the Float to the submerged Glider. It is fairly long (5.8 meters or 4 meters) and streamlined to reduce drag.

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

Figure 2.8. Umbilical



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

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 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 is tapered.

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

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.

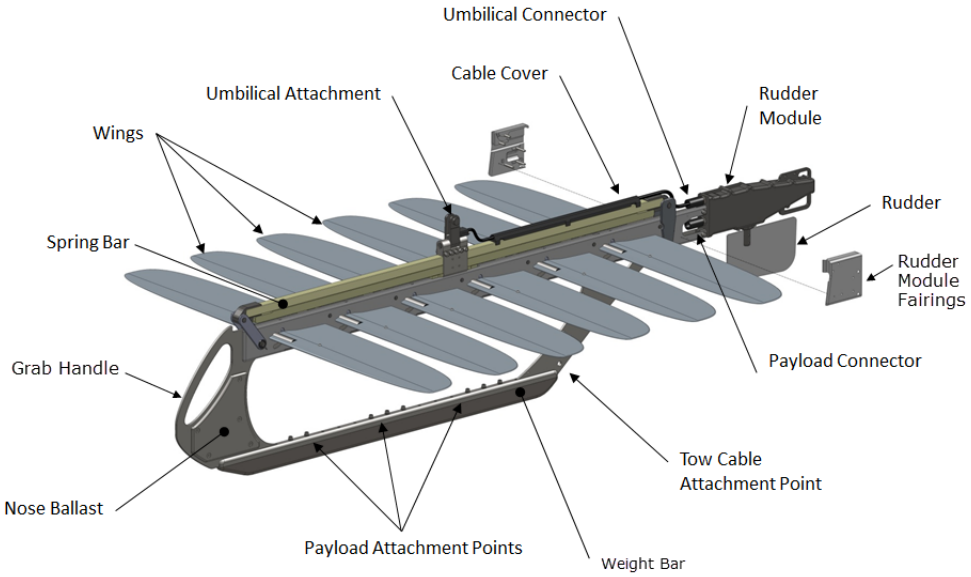
Glider Attach: The mechanical connection to the Glider. This is the upper part of the Glider gimbal attachment system and requires assembly.

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

Glider

The Glider (also called the "Sub"), provides both thrust and steering. It can also be used to carry underwater payloads, although attaching payloads to the glider is discouraged.

Figure 2.9. Glider (Sub)



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 flotation.

Nose Ballast: Forward-located weight that can be changed to balance the Glider to compensate for payload mounting that impacts the Glider center-of-gravity.

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

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

Rudder: For steering the Wave Glider.

Rudder Module: Factory sealed. Contains the servo motor and mechanism for the Rudder. Has electrical connectors for the Umbilical and for Glider or Towfish payloads.

Rudder Module Fairings: Covers that protect Rudder Module electrical connections.

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

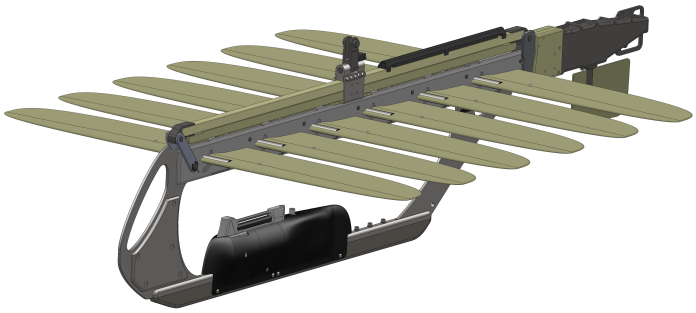
Umbilical Attach: Mechanical attachment point for the Umbilical.

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, streamlined fairings can be fabricated to cover the payload. An example is shown in the figure below. The actual shape of the cover can vary depending on the type, shape and volume of the payload.

Figure 2.10. Glider With Custom Payload Cover



See the section called “Glider Payload Considerations” for additional design considerations for Glider payloads.

Towfish Options

A Towfish is tethered to the Glider providing a satellite communication gateway for on-demand data transmissions between a sub-sea sensor and data gathering on-shore. This section provides an overview of the Wave Glider Towfish options. There are a wide range of variables that impact Towfish sensor integration. Contact Liquid Robotics for additional information necessary to evaluate the feasibility of a Towfish sensor integration project.

There are two Towfish options for the Wave Glider:

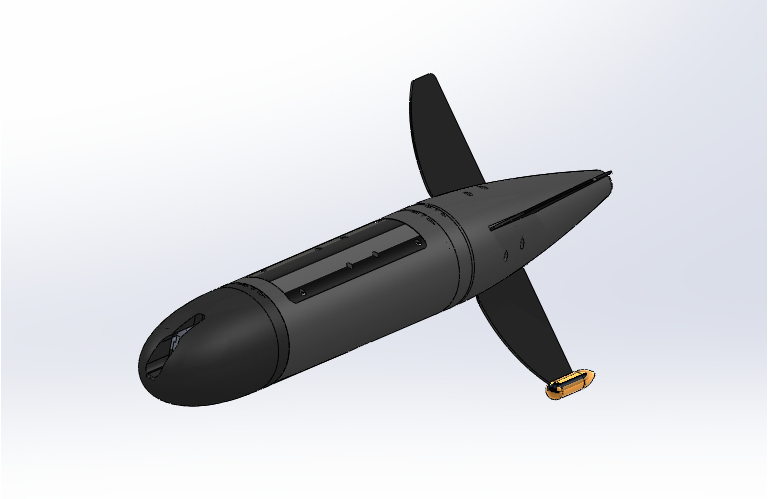
- **Gateway** For information on the Gateway Towfish see the section called “Gateway Towfish”.

- **4" Modular Towfish** For information on the 4" Modular Towfish see the section called “4" Modular Towfish”.

Gateway Towfish

The Gateway Towfish is a sub-sea extension of the Wave Glider system that provides a dynamically isolated location to mount an acoustic transducer and accompanying electronics for communication with sub-sea sensors. It is a streamlined, buoyancy neutral vessel that is tethered to the Rudder Module on the Glider. The Gateway Towfish communicates over a shock-absorbing tether, transmitting acoustic data through the Rudder Module to a payload on the Wave Glider Float. From the payload on the Float, data can be transmitted through the C&C and over the Iridium Satellite network to shore.

Figure 2.11. Gateway Towfish



The Gateway Towfish is designed to carry an acoustic transducer and a modem, with related electronics located in a payload dry box on the Float. Its payload bay can accept a 4.5" diameter transducer weighing up to 7 lbs. and it is designed to accept the Benthos modem or the WHOI micro-modem, with up to 3.5 lbs of total modem weight. Once the contents of the payload are installed, the Towfish must be adjusted to neutral buoyancy and level trim.

Power requirements depend on the sensor and modem that will be used. A high voltage board (located in a Float payload dry box) may be required to power the electronics in the Towfish. Power available to the payload is dependent on Rudder Module power limitations, as well as over-all Wave Glider system power limitations.

The Gateway Towfish has been proven in high sea states and has been successful in missions off shore operating to depths of 4000 meters.

4" Modular Towfish

The 4-inch Modular Towfish is sub-sea extension of the Wave Glider system that provides mounting points, power and communications channels for a wide variety of underwater payloads. It is a streamlined, buoyancy neutral vessel that is tethered to the Rudder Module on the Glider.

The 4" Modular Towfish can carry multiple payloads, each with dedicated electrical power and communications provided by the on-board Sensor Management Computer that is connected to the Float C&C through the Rudder Module and a Float payload dry box.

The number of payloads the 4" Modular Towfish can support depends on the weight, physical dimensions, and power requirements of the payloads. Power available to the payloads is dependent on Rudder Module power limitations, as well as over-all Wave Glider system power limitations.

Once the contents of the payloads are installed, the Towfish must be adjusted to neutral buoyancy and level trim.

Figure 2.12. 4" Modular Towfish

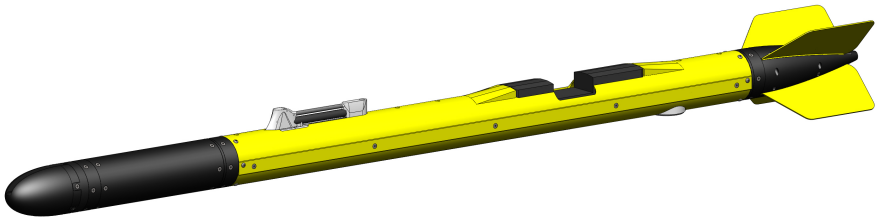
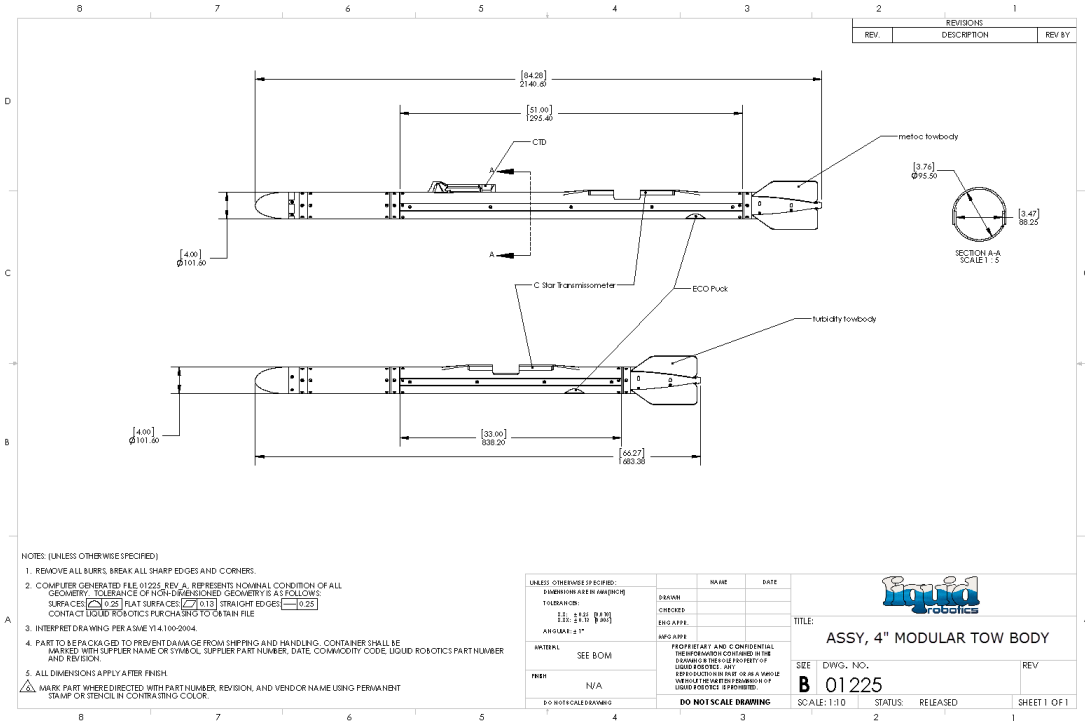


Figure 2.13. 4" Modular Towfish Dimensions



Towfish Tether

The same 5.75 meter long tether is used to connect the Gateway or Metoc Towfish to the Glider. It is designed with attached floats and weights to absorb shock, protecting the towfish from the surging action of the Glider. The speed of the Wave Glider is reduced by 3 to 5% when a properly configured Towfish is attached.

Figure 2.14. Towfish with Tether 3D

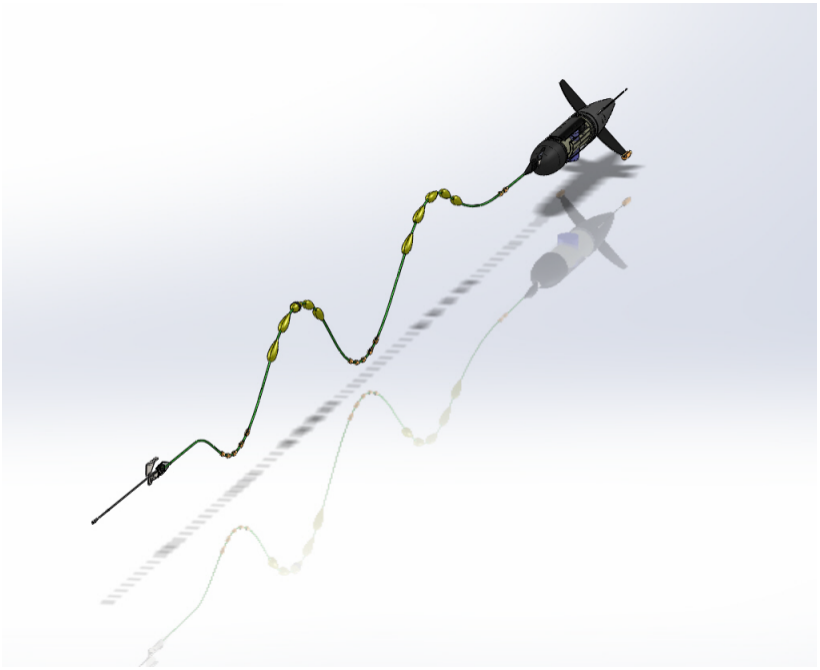
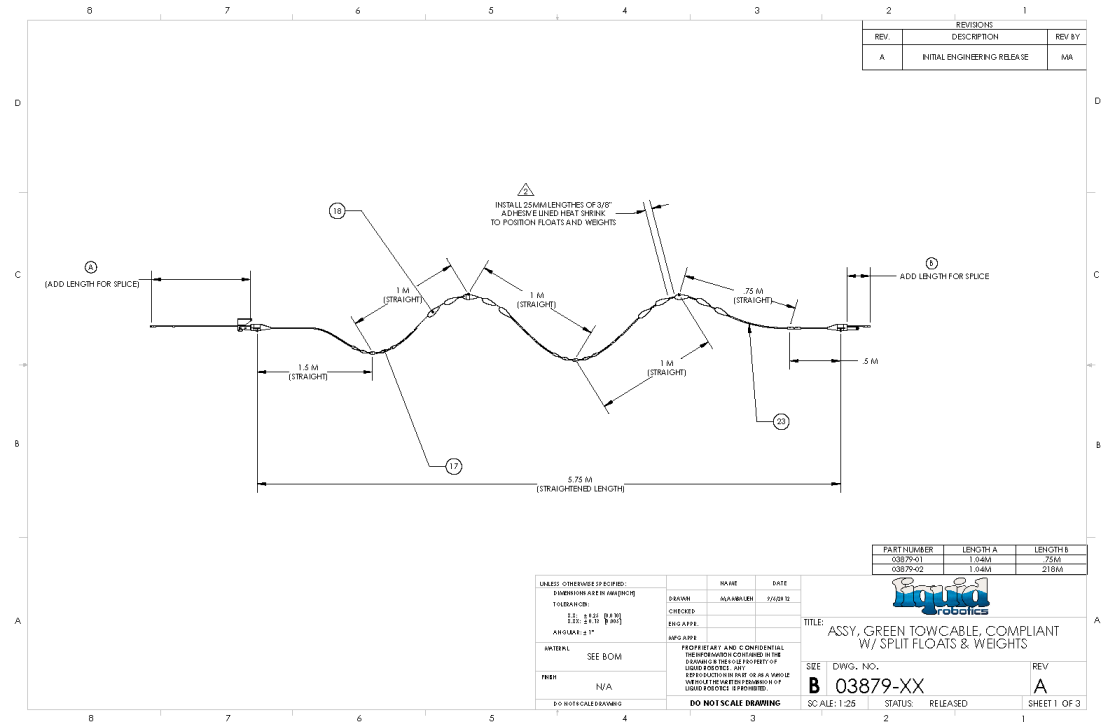


Figure 2.15. Towfish Tether Dimensions



Note

If you are planning shallow water operations, be aware that the Towfish flies approximately 1 to 2 meters below the sub.

Electrical

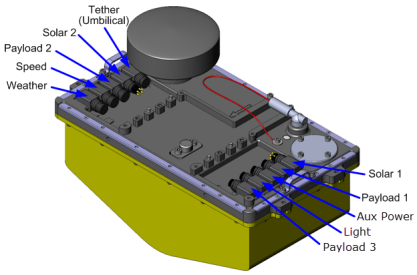
The basic control and communications electronics are in the C&C Dry Box. Additional electronics can be added as a payload, generally interfacing with the C&C Dry Box.

For details on Wave Glider electrical systems, see Chapter 5, *Electrical Specifications*.

C&C Dry Box Connections

The top of the C&C Dry Box contains two male 5-pin connectors for the two Solar Panels. The other eight connectors are 8-pin female. Figure 2.16, “Connections on C&C Dry Box” identifies the various C&C Dry Box connections.

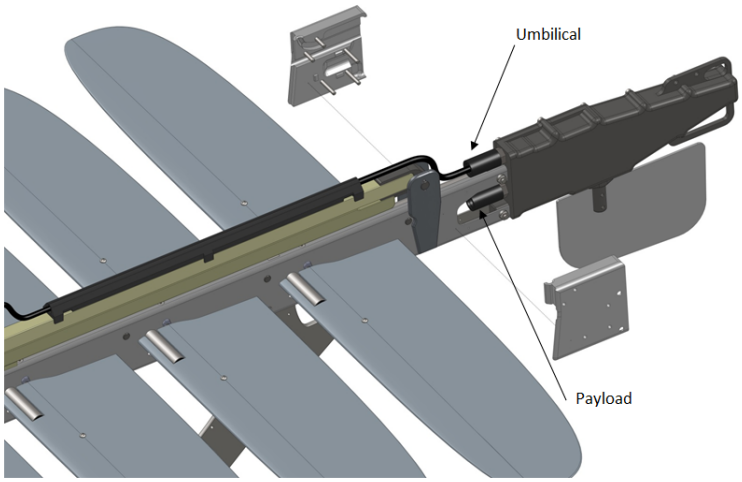
Figure 2.16. Connections on C&C Dry Box



Rudder Module Connections

The Rudder Module has an 8-pin male connector that is used for the Umbilical electrical connection. There is also an 8-pin female connector for a Glider payload.

Figure 2.17. Rudder Module Connectors



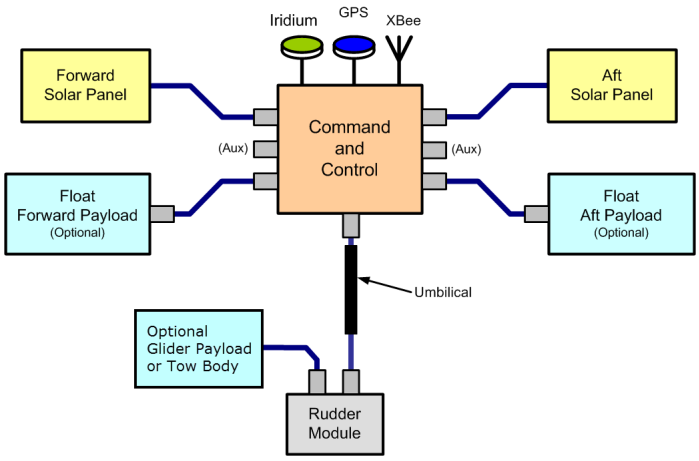
Note

In the above figure, the Rudder Module Fairings are removed to show the connectors.

Electrical Connectivity

The general connectivity of the Wave Glider electrical system is shown below. A complete description is included in Chapter 5, *Electrical Specifications*.

Figure 2.18. Basic Electrical System



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

Communications and Control

On-board communications are managed by the computer in the C&C Dry Box. Standard Wave Glider data communications can be in the following formats:

- An Iridium satellite link between the C&C and a remote Console.
- An RS-232 wire link from the C&C to the Fore and/or Aft payloads.
- An XBee radio link from the C&C to a nearby receiver.

Iridium Satellite Network

The Iridium satellite network is the primary communications link for the Wave Glider.

A Wave Glider's Iridium modem sends and receives data using Iridium Short Burst Data (SBD) packet sessions. Iridium computers on-shore exchange data between Wave Gliders and the Wave Glider Management System (WGMS). WGMS is a Liquid Robotics-developed application that provides a GUI based Wave Glider navigation and communications management console.

Communication sessions are initiated by the Wave Glider. The Wave Glider periodically sends telemetry data packets from the vehicle C&C to WGMS over the Iridium link. The time span between telemetry messages is user-configurable. Command data packets from WGMS are sent to the Iridium computers, where they are queued until the Wave Glider initiates a communication session. After telemetry data is sent from a vehicle to WGMS, any commands sent from WGMS are downloaded to the vehicle C&C.

When a 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 requests some action by the vehicle (for example, moving the rudder), that action is also executed.

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 you enter the command.

See Chapter 6, *Vehicle Communications and Control* for more information.

Relay Communications

It is sometimes beneficial to communicate with the vehicle through means other than the standard Iridium Server. Relay communications are similar to Iridium Satellite Network communications except that data is sent directly to and from the vehicle through some other satellite or radio means. Considerations can include bandwidth, cost, latency, privacy, security and distance. Communication relays can be one of four types: Iridium, Alternate Iridium, XBee, or Custom).

See Chapter 6, *Vehicle Communications and Control* for more information.

Payload Communications

Each of the Float mounted 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.

Glider mounted payloads and towfish mounted payloads use auxiliary electrical lines in the Umbilical to connect with electronics in the Float mounted payload dry boxes, and from there they communicate with the C&C over RS-232 links.

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

For more information see Chapter 11, *Payload Communications Protocol*.

XBee Radio Link

The XBee radio link is used for short-range (approx. 100 meters) communications to and from the vehicle. XBee communications consist of interactive text-based commands and responses. XBee is typically used during vehicle launch and retrieval. The XBee link is also convenient for shore-based payload development and system integration, since commands are received and processed immediately.

See Chapter 6, *Vehicle Communications and Control* for more information.

Navigation

Navigation of the vehicle is typically achieved by referencing Waypoints. Each Waypoint is 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 - 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.

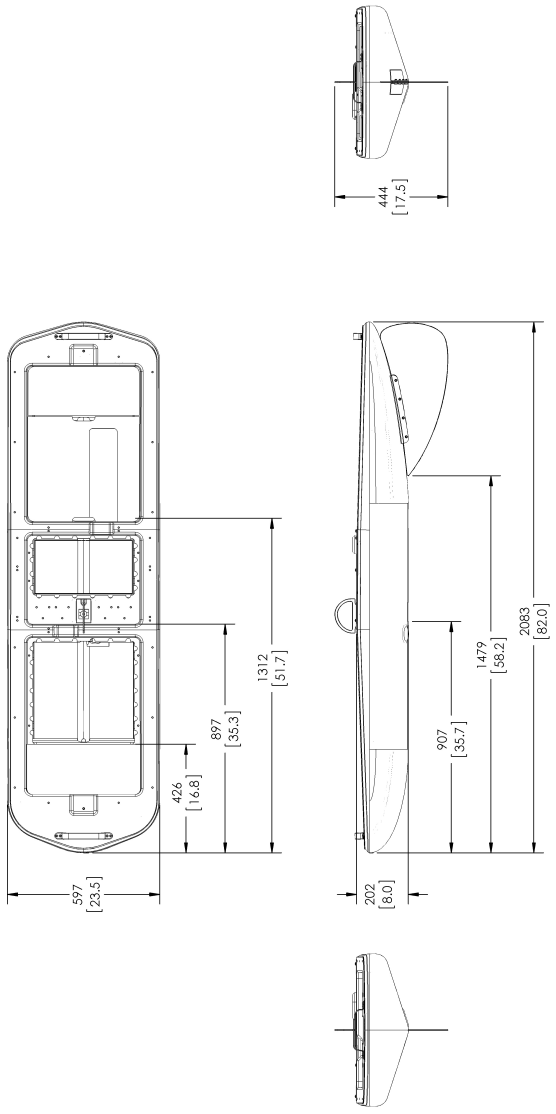
For more information on waypoints and heading values, see the section called “Communication Data Formats”.

Chapter 3. Float Mechanical

The drawings in this chapter show dimensions in millimeters and [inches].

Float Overall Dimensions

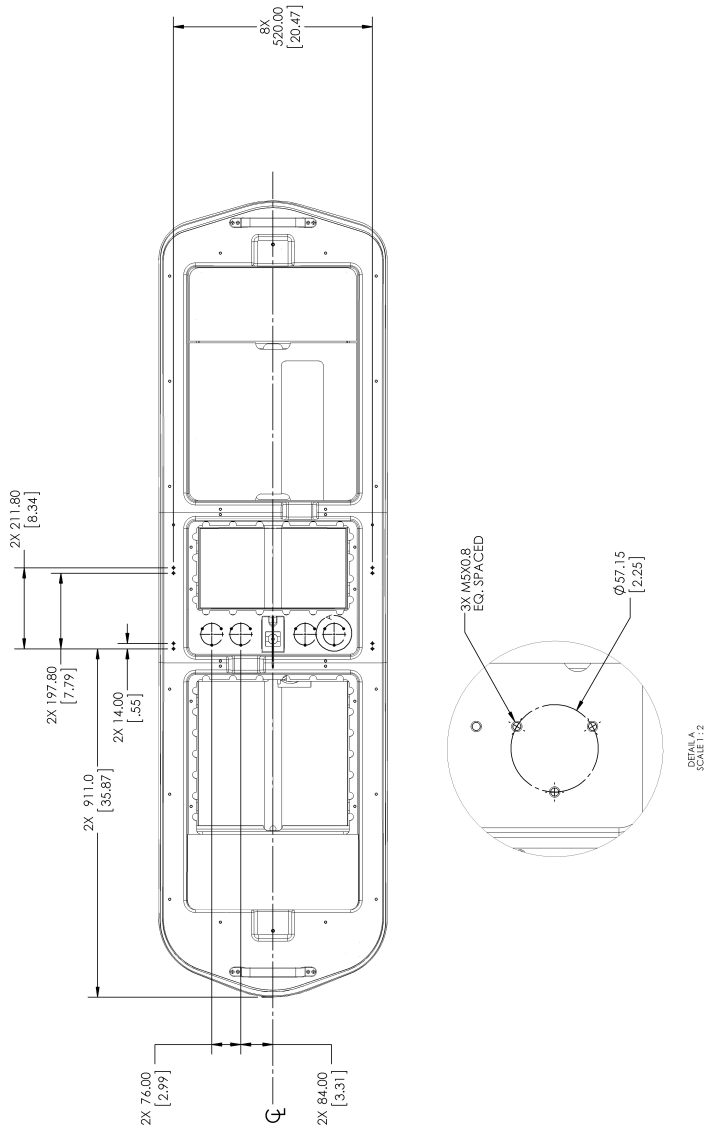
Figure 3.1. Float Overall Dimensions



Float Attachment Hard-points

The payload attachment points on the upper deck of the Float are shown in the drawing below. There are 20 attachment points. All attachment points are grade 5 titanium thru-hole M5 x 0.8 mm pitch thread.

Figure 3.2. Float Attachment Hard points



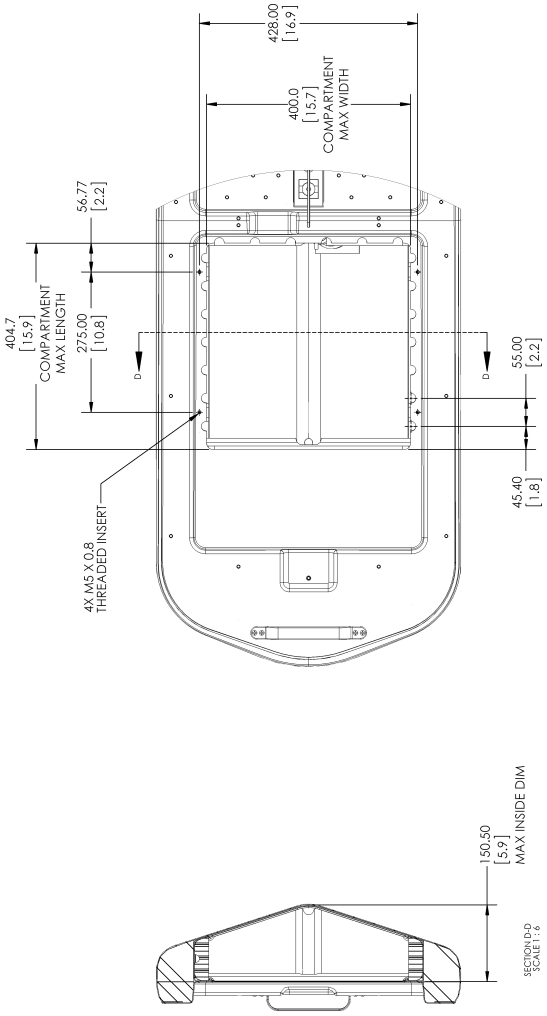
Float Payload Bays

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

Float Forward Payload Bay

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

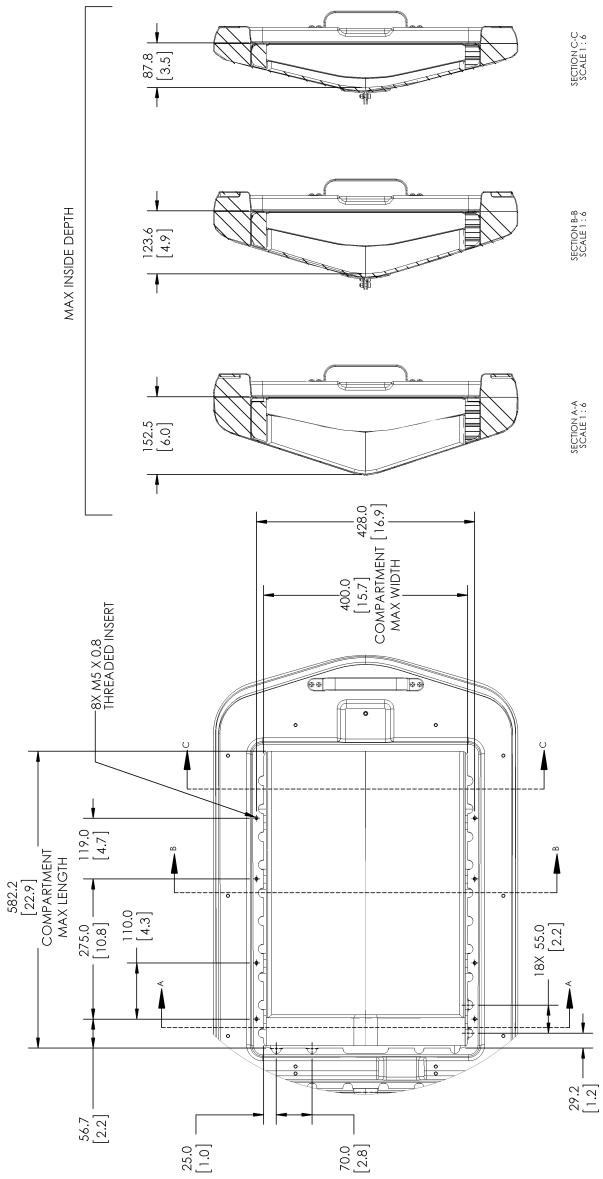
Figure 3.3. Forward Payload Bay Dimensions



Float Aft Payload Bay

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

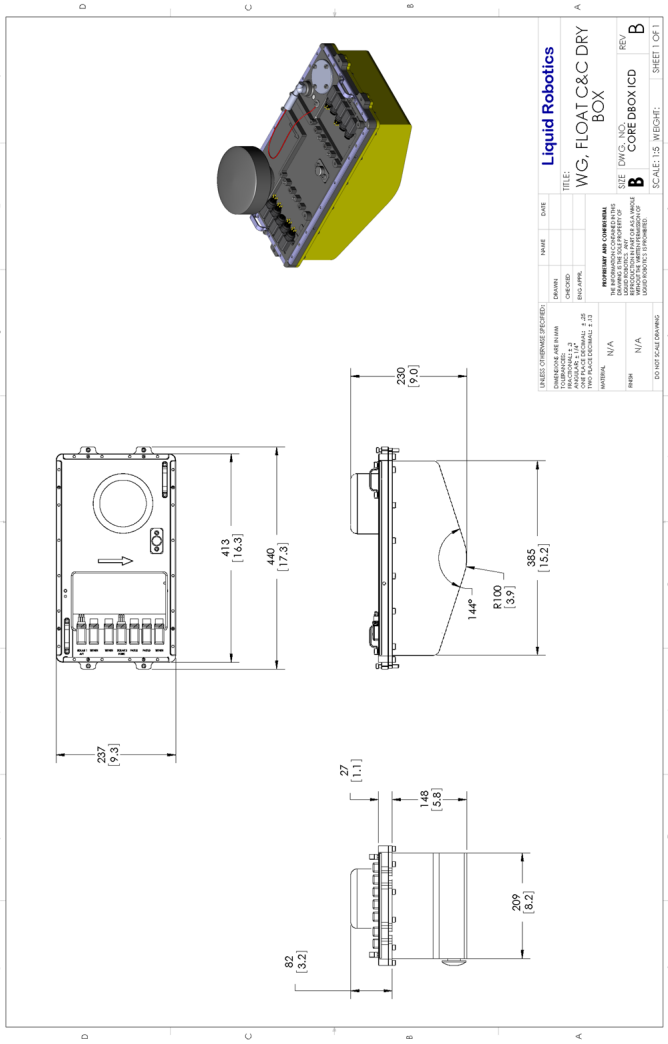
Figure 3.4. Aft Payload Bay Dimensions



C&C Dry Box

For reference, the dimensions of the C&C Dry Box are shown below. The C&C Dry Box contains no customer-accessible components. For information on connector pin outs, see the section called “Standard Electrical Connections”.

Figure 3.5. C&C Dry Box Dimensions



Payload Dry Boxes

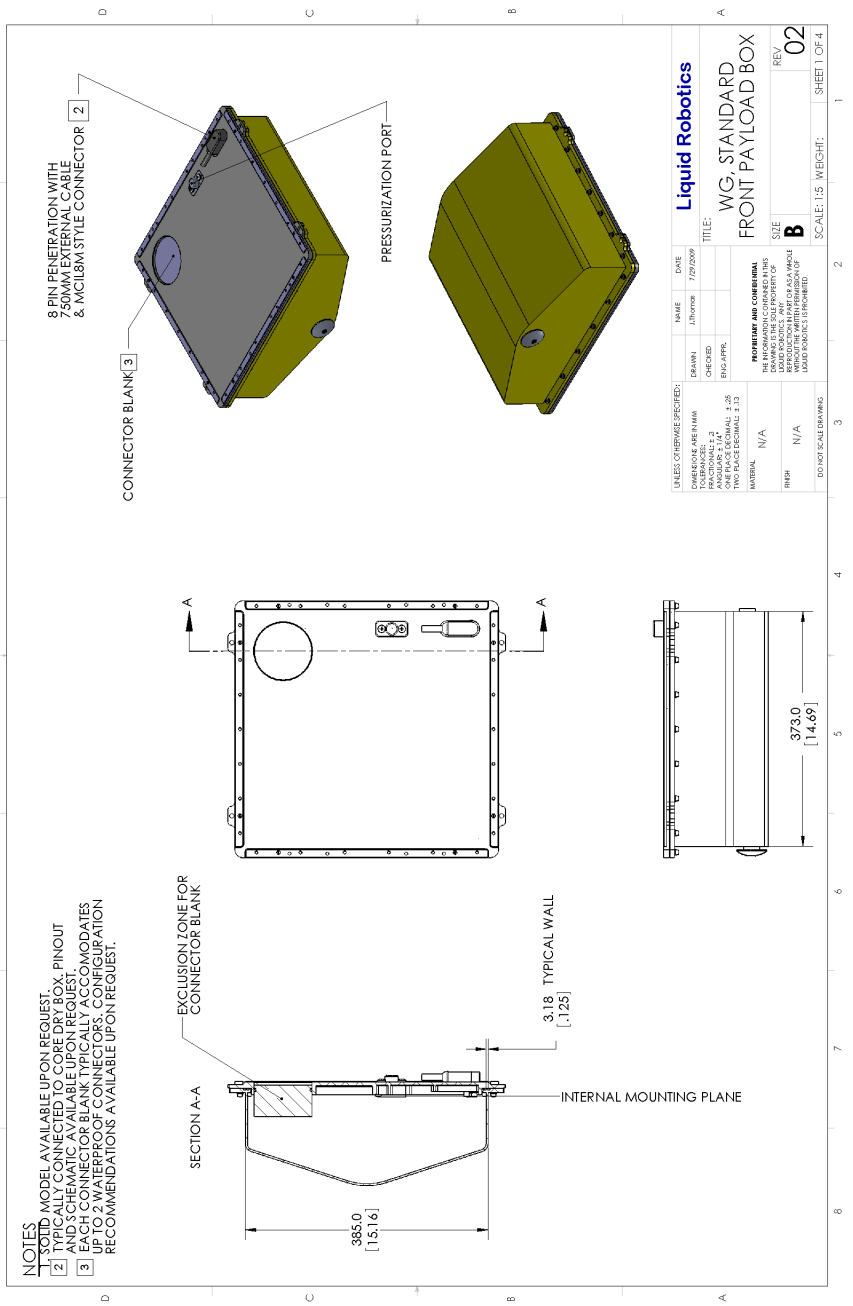
Liquid Robotics offers three standard payload dry boxes:

- Forward Payload Dry Box
- Aft Payload Dry Box
- Short Aft Payload Dry Box

Forward Payload Dry Box

The overall dimensions of the Forward Payload Dry Box are shown below. Note that the lids for both the forward and aft payload dry boxes are identical. Only the bottom casing is different.

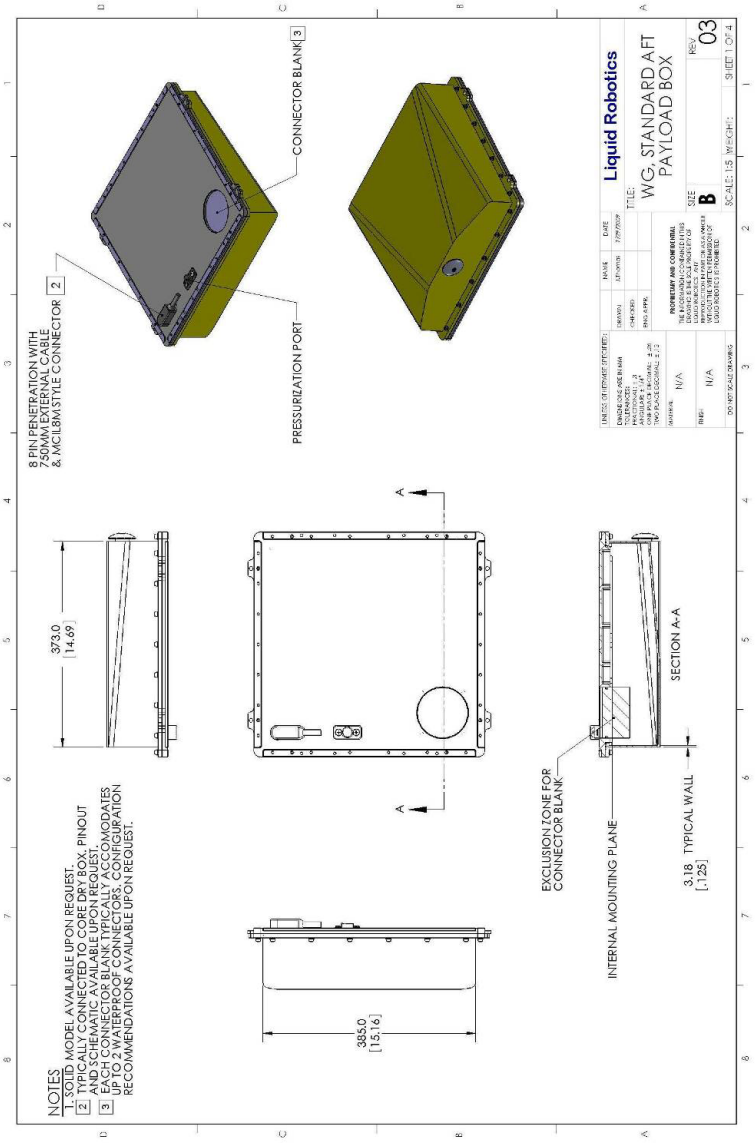
Figure 3.6. Forward Payload Dry Box Dimensions



Aft Payload Dry Box

The overall dimensions of the Aft Payload Dry Box are shown below. Note that the lids for both the aft and forward payload dry boxes are identical. Only the bottom casing is different.

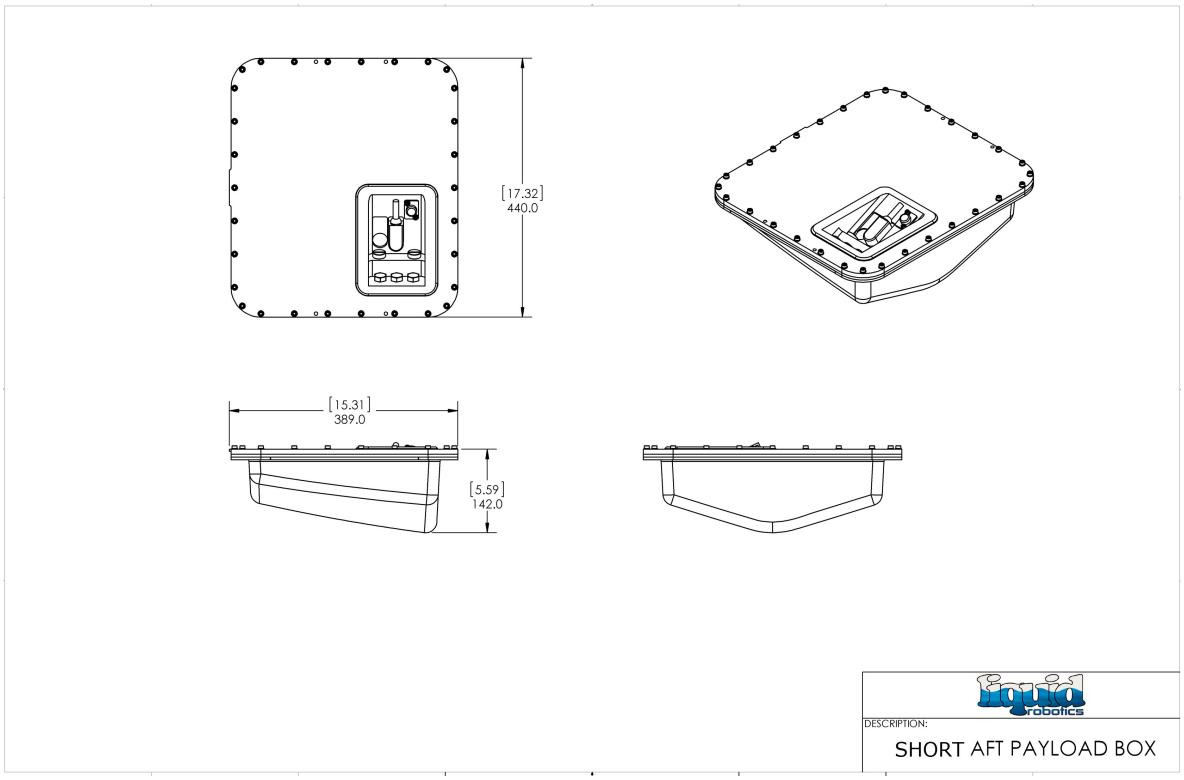
Figure 3.7. Aft Payload Dry Box Dimensions



Short Aft Payload Dry Box

The overall dimensions of the Short Aft Payload Dry Box are shown below.

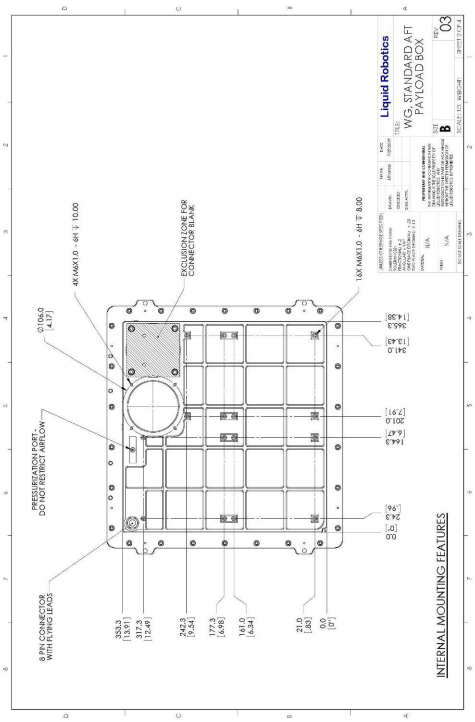
Figure 3.8. Short Aft Payload Dry Box



Forward and Aft Payload Dry Box Lid Mounting

The forward and aft payload dry box 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.

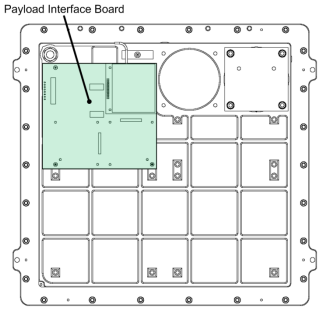
Figure 3.9. Forward and Aft Payload Dry Box Lid Underside Dimensions



Forward and Aft Payload Dry Box Lid With Payload Interface Board(s)

A forward or aft payload dry box lid can be optionally equipped with a Payload Interface Board (PIB). Up to three PIBs can be mounted on the underside of the lid. For more information on PIBs, see Chapter 12, *Payload Communications Hardware*.

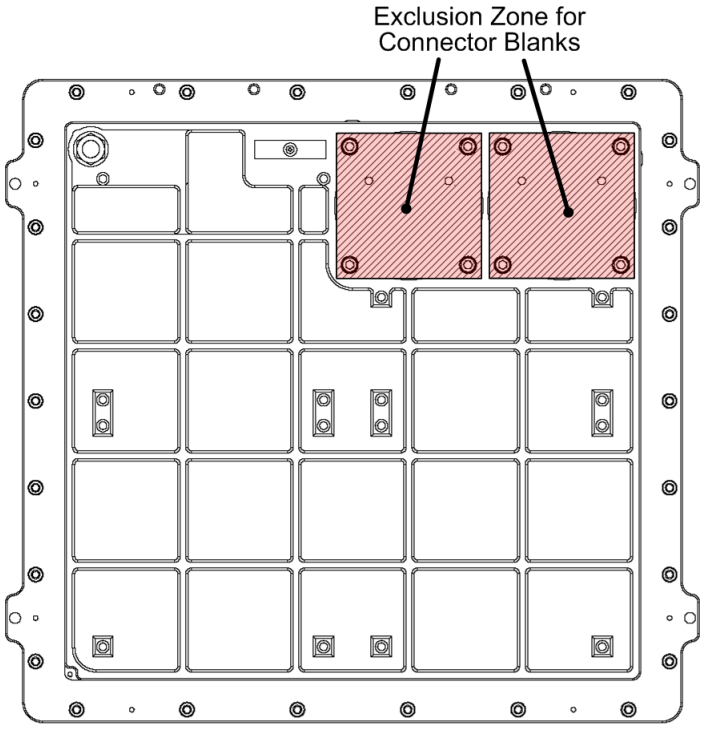
Figure 3.10. Forward and Aft Payload Dry Box Lid with PIB



Forward and Aft Payload Dry Box Lid With Dual Connector Blanks

The forward and aft payload dry box lid can be optionally equipped with two Connector Blanks that can be replaced with a variety of external connector options for the dry box. The exclusion areas are shown below.

Figure 3.11. Forward and Aft Payload Dry Box Lid With Two Connector Blanks



Float Payload Considerations

Maximum and Minimum Weight

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

Table 3.1. Float Maximum and Minimum Weight Limits

Glider Configuration	Float Maximum Weight
Glider at minimum weight	70.31 Kg (155 lb)
Glider at maximum allowed weight	63.5 Kg (140 lb)

Center-of-Gravity Range

For maximum efficiency, the horizontal center-of-gravity of the Float must be within a narrow range.

The center-of-gravity value can be verified by placing the Float (without the Umbilical and Glider attached) 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 3.12. Float Max Forward Center-of-Gravity

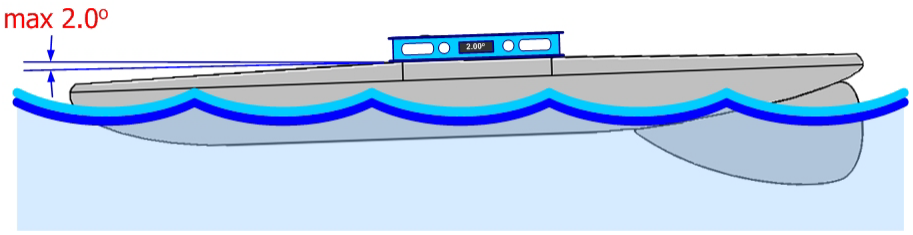
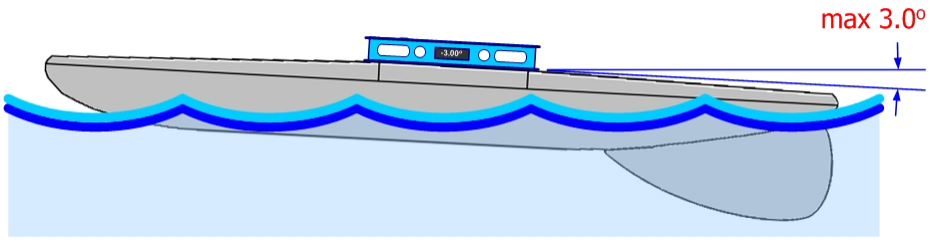


Figure 3.13. Float Max Aft Center-of-Gravity

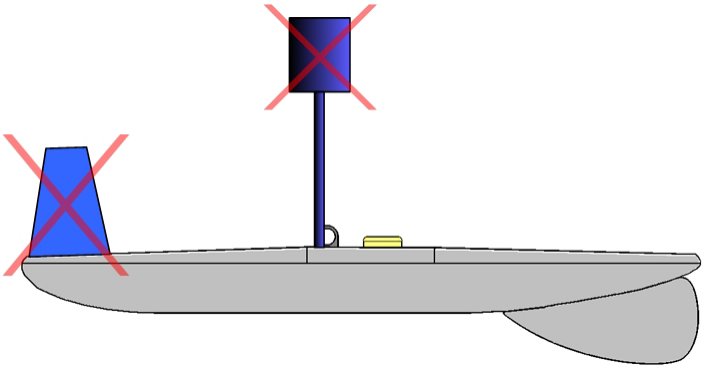


IMPORTANT - Payloads must be ballasted, if necessary, so the Float is exactly level from side-to-side when resting in calm water (it cannot list).

Float Sail Area and Masts

There are limits to the amount of sail area that can be mounted on top of the Float. Payloads with large sail areas should be avoided. Although the underwater Glider acts like a large keel to stabilize the Float, tall masts with large or heavy tops should also be avoided. Contact Liquid Robotics Customer Support before mounting objects on the Float that could create significant sail effects or impact the stability of the Float.

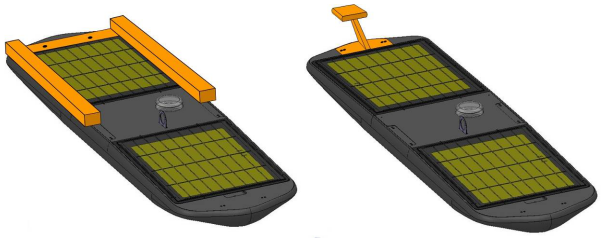
Figure 3.14. Possible Sail Area Problems



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 3.15. Deck Mounting of Payloads



Towing

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

Figure 3.16. Towing Behind Float



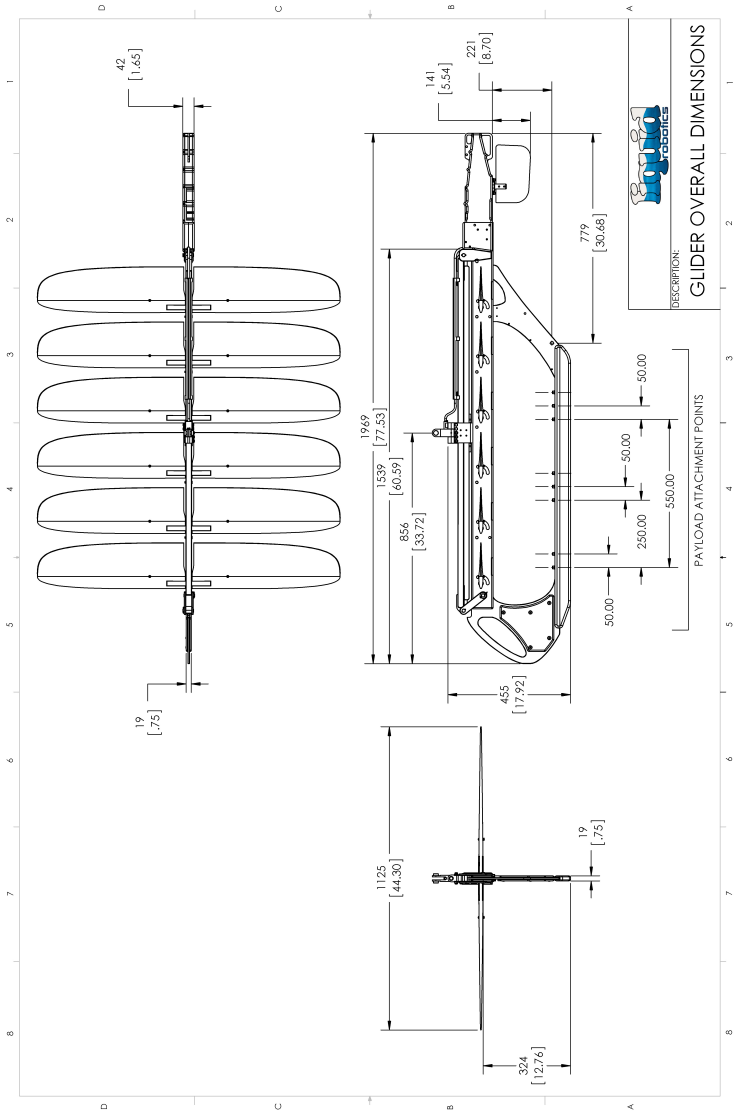
Chapter 4. Glider Mechanical

The drawings in this chapter show dimensions in millimeters and [inches].

Glider Dimensions and Payload Attachment Points

The Glider attachment points are detailed in the following drawing along with the Glider overall dimensions. All attachment points have M5 x .08 mm pitch threads.

Figure 4.1. Glider Overall Dimensions and Attachment Points



Glider Payload Considerations

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

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 upward in heavy swells.

Figure 4.2. Glider Moving Up

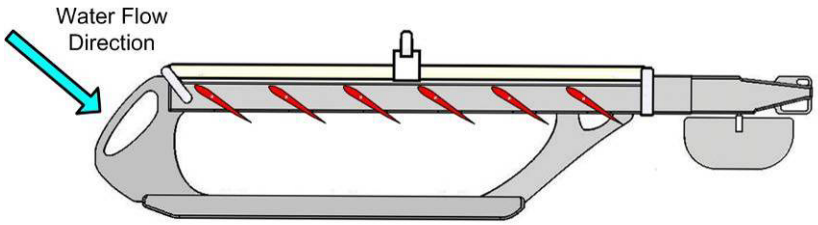
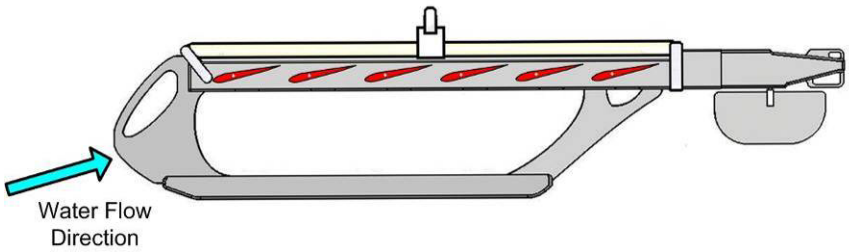
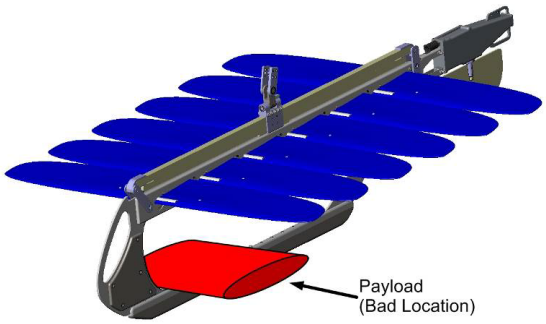


Figure 4.3. Glider Moving Down



Thus, a seemingly "streamlined" payload, like that shown in Figure 4.4, “Poor Payload Design/Location”, would create a substantial nose-down force (as well as drag) when the Glider is moving up.

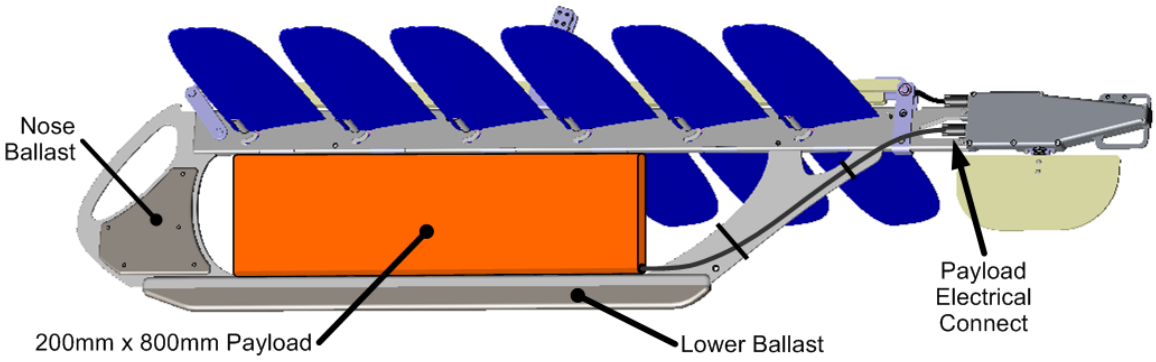
Figure 4.4. Poor Payload Design/Location



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. The payload electrical connection is through the Rudder Module. (The Rudder Fairings are removed to show the connector detail.)

Figure 4.5. Glider with Payload Box



Any payload mounted within the frame of the Glider must conform to the wing clearance requirements described in the section called “Wing-Payload Interference”. Payloads mounted in the Glider frame can create a sail area and cause drag. Consult with Liquid Robotics before attempting to mount a payload in the Glider frame.

Glider Ballast Weights

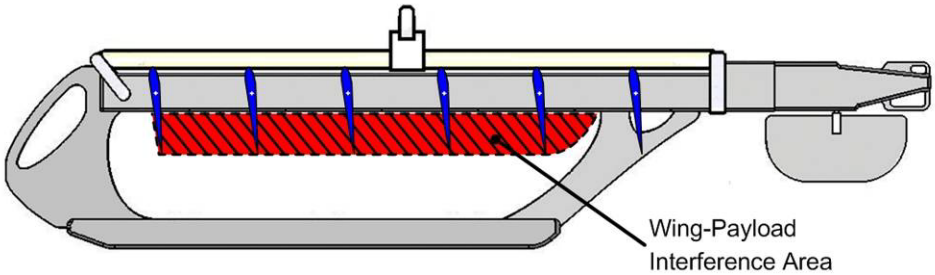
Ballast weights are attached to the nose and bottom of the Glider frame as shown in Figure 4.5, “Glider with Payload Box”. As payload is added, these weights can be removed or replaced with other weights to adjust the balance and buoyancy of the Glider.

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

Wing-Payload Interference

Since the Glider wings can rotate upward or downward almost 90° to the vertical position, any payload (or payload cover) mounted within the main frame of the Glider has to accommodate clearance of the wings in their fully rotated position as illustrated in the figure below.

Figure 4.6. Wing-Payload Interference Area



Balance and Flotation

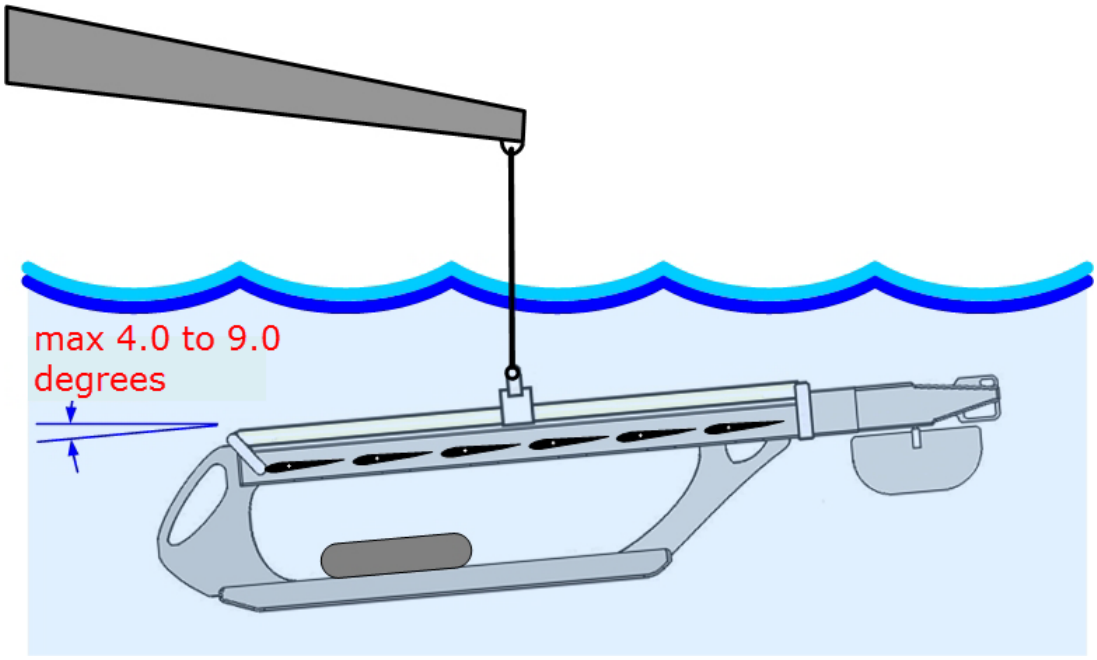
A Glider with an added payload needs to be balanced about the Umbilical pivot point. Since the Glider is submerged, the balance needs to be established underwater, to account for both the flotation and the weight of the payload.

The best way to compute the flotation and 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 slightly nose-down. The maximum nose-down angle is shown in the figure below.

The Glider should never be ballasted nose-up.

Figure 4.7. Checking Glider Underwater Balance



The flotation of the Glider with the payload installed also needs to be checked while the Glider is underwater, to account for both the weight and the flotation of the payload. The minimum and maximum submerged weights are listed in the Table below:

Table 4.1. Minimum and Maximum Total Glider Weight - Underwater

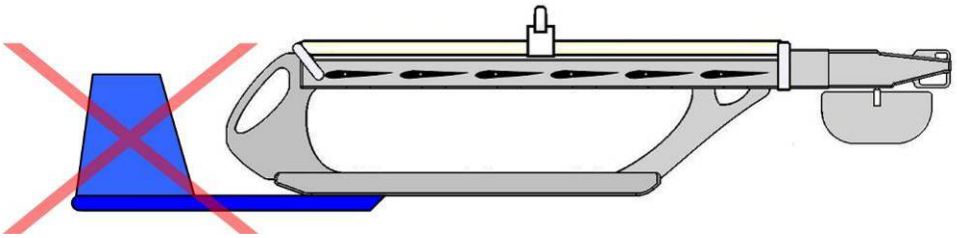
Minimum Weight	40 Kg (88 lbs)
Maximum Weight	60 Kg (130 lbs)

Poor Sail Area Locations

A Glider payload 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.

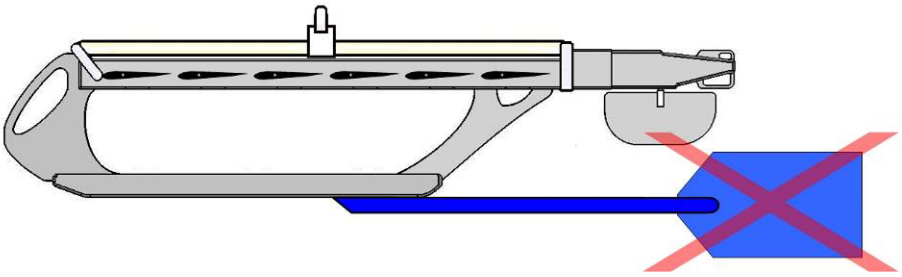
A large sail area in front of the Glider can destabilize and make it over-sensitive to the rudder.

Figure 4.8. Excessive Forward Payload Sail Area



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

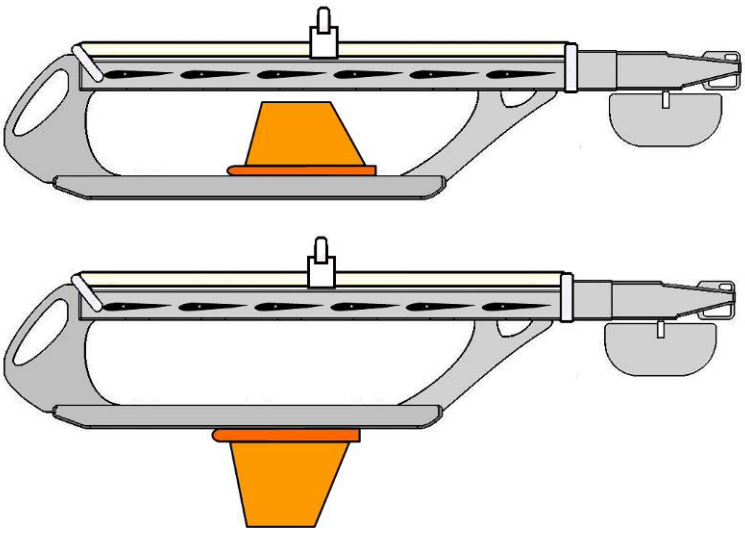
Figure 4.9. Excessive Aft Payload Sail Area



Acceptable Sail Area Locations

Generally, a reasonably-sized payload sail area located 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 4.10. Acceptable Payload Sail Area Locations

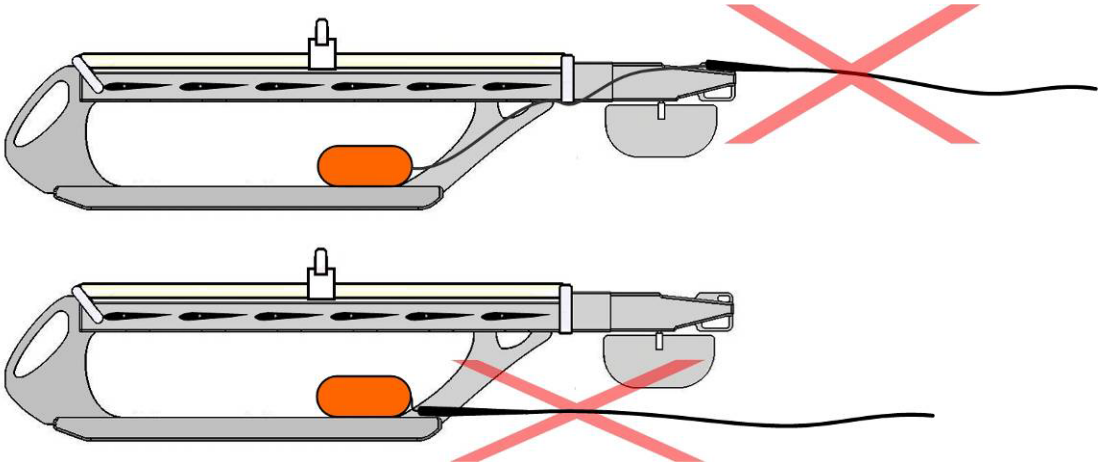


Trailing Wires

Trailing wires behind the Glider are generally not recommended, as they can potentially foul the Rudder. However, if a trailing wire is necessary, then the wire should be mounted near the center of the Glider,

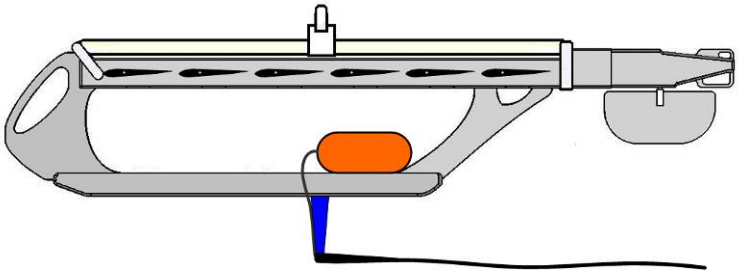
and not trail off of the back. In addition to fouling, rear trailing wires can have the same over-stabilizing effect as a large sail area at the aft end of the Glider.

Figure 4.11. 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 4.12. 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 so it cannot foul the rudder, regardless of the angle of the wire.

Chapter 5. Electrical Specifications

Physical Connectivity

Figure 5.1, “Wave Glider Connection Diagram” illustrates a Wave Glider system with all Float and Glider payloads connected. This is only an example of what is possible; it should not be considered a limiting factor in the overall design and implementation of a Wave Glider system. The diagram Wave Glider components in the diagram are described in this section. The technical details are provided in later sections of this chapter.

Figure 5.1. Wave Glider Connection Diagram

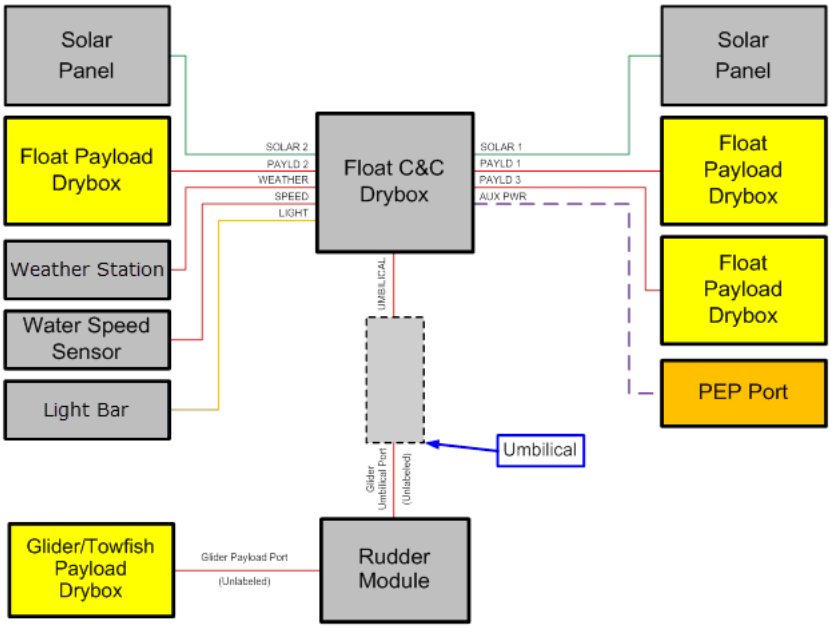


Table 5.1. Color and Label Conventions for System Diagram

Indication	Group	Description
Grey Box	Core System	These components are a part of the core system
Yellow Box	Payloads	End user equipment installation points
Green Line	5-pin whip	These are the 5-pin connections for power into the system
Red Line	8-pin whip	These are the 8 pin connections for power and data to components
Orange Line	2-pin whip	This is the 2 pin connection for power only
Purple dash	6-pin whip	This is the whip that provides extra power out and an additional input power point
Orange Box	PEP Port	This connection provides supplemental power for a variety of uses (see the section called “PEP Port”)

Indication	Group	Description
Names	Port Labels	Names in all caps are found on the Float C&C lid

Solar Panels

The solar panels are used to power the system and to charge the back-up batteries. A system does not require solar panels for short deployments.

The solar panel inputs can be used for other sources of energy as well, such as a wall charger. The alternate forms of energy must meet the Float C&C connection requirements.

Float C&C

The Float C&C acts as the central hub for communication and power. Its main purpose is to maintain power for the end user equipment and to provide a communication path from the user to the vehicle. It contains no customer-serviceable components.

Umbilical

The Umbilical is the physical and electrical connection between the Float and the Glider. The Umbilical provides for power to be transferred from the Float to the Rudder Module, and to Glider-attached payloads and Tow Bodies. The Umbilical also provides for data to be transferred between the Float Payload Dry Boxes (as well as the C&C), and the Glider Payload Dry Boxes, Tow Bodies, and the Rudder Module.

Rudder Module

The Rudder Module steers the Wave Glider vehicle and provides electrical and communications connectivity to Glider-attached payloads or Tow Bodies.

Water Speed Sensor

The water speed sensor measures speed through water. When combined with GPS-derived speed over ground it is possible for pilots to estimate local currents.

Payload Dry Boxes and Tow Bodies

Payload Dry Boxes and Tow Bodies house sensors and other application payloads. They are designed to be customer accessible. The Liquid Robotics-provided Payload Interface Board (PIB) provides electrical and communications connectivity between a Payload Dry Boxes or Tow Bodies and the Float C&C. A large number of configurations are possible.

Figure 5.2. Payload Dry Box Electrical Components

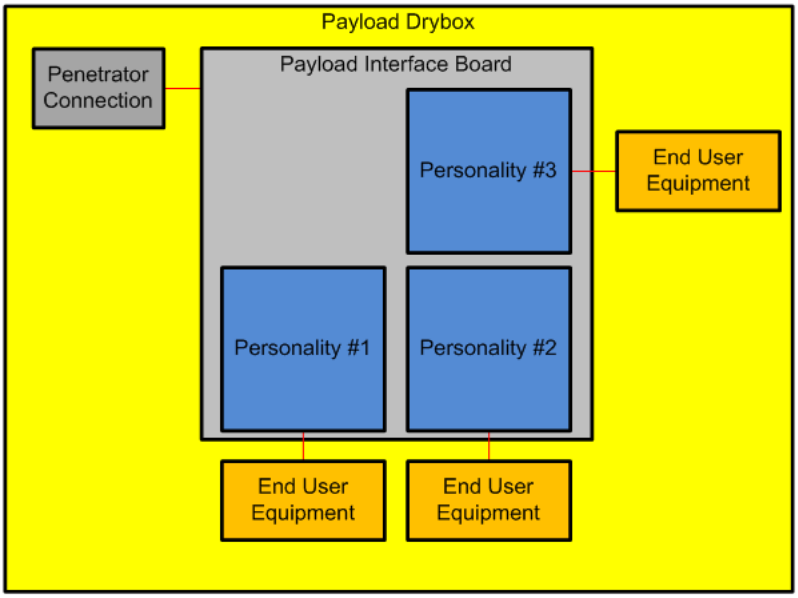
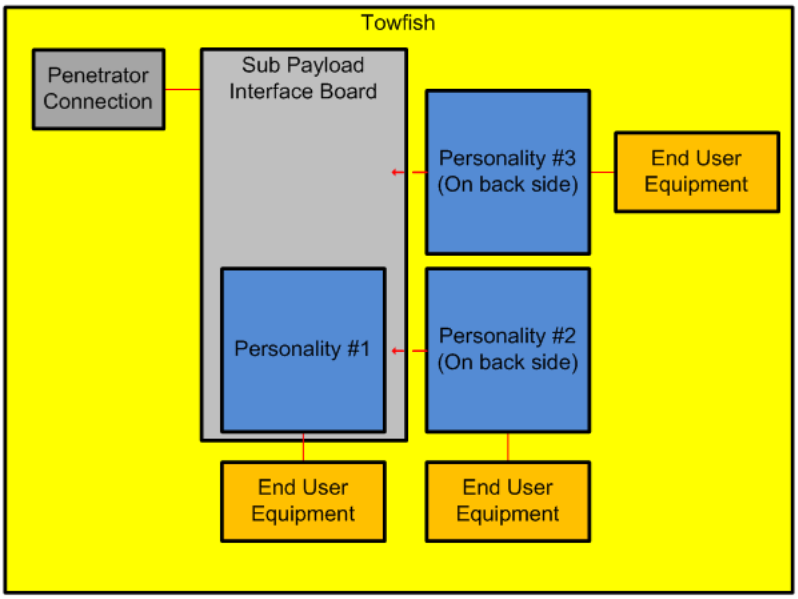


Figure 5.3. Towfish Electrical Components



PEP Port

The PEP (Payload Extra Power) port is used to provide up to 6 Amps of supplemental power to a payload dry box that requires more power than a standard payload connector can provide (3 Amp limit). The PEP port can also be used as the only power supply to a payload dry box that does not require any communications connection to the C&C (the PEP port does not provide any communication lines). You also have the option to use the PEP port to supplement the power supply for the Wave Glider electrical

system, by placing extra batteries in a payload dry box and connecting them to the PEP port (contact Liquid Robotics for details).

Weather Station and Light Bar

Liquid Robotics provides a weather station and light bar as standard equipment on the Wave Glider vehicle. Each has a dedicated port on the C&C Dry Box.

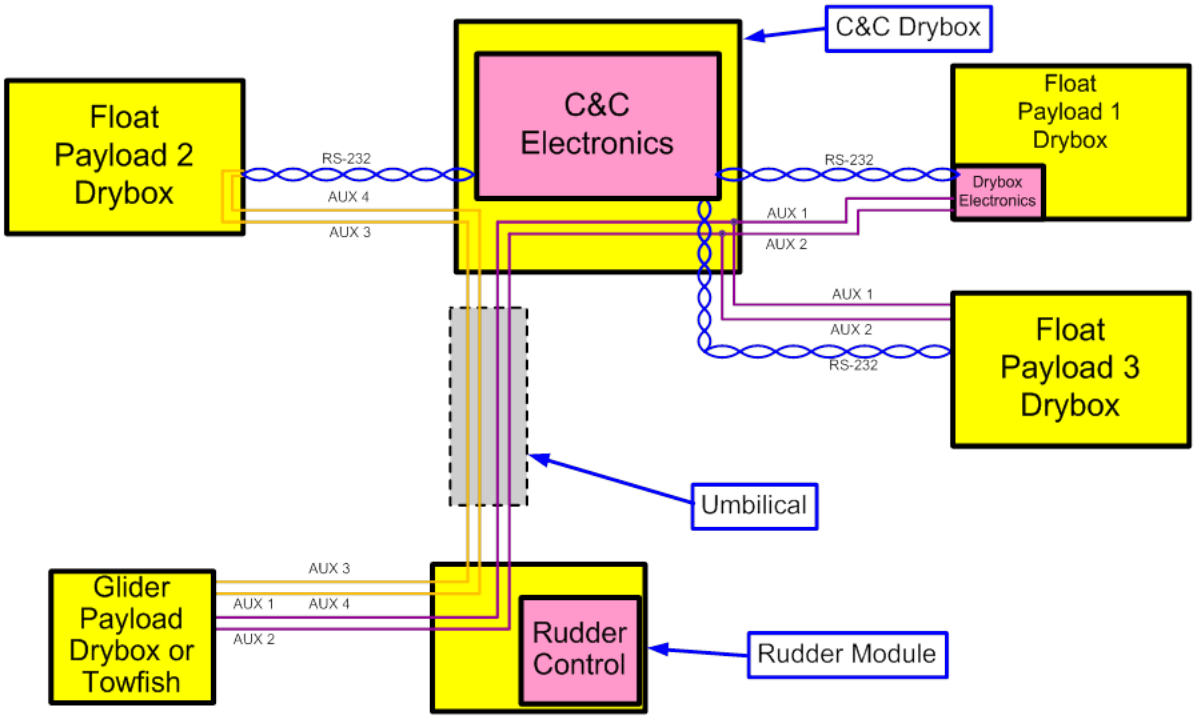
Payload Connectivity

This section describes the physical connections between the Payload Dry Boxes and the Float C&C.

Payload Communication Connectivity

Figure 5.4, “Payload Interconnectivity and Communication Diagram” shows the possible Wave Glider communication paths. There are dedicated RS-232 communications paths between each of the Float payloads and the Float C&C for transfering data to shore, or to other payloads. The Glider and Towfish payloads do not have a direct connection to the Float C&C and consequently must communicate through a payload in one of the Float payload dry boxes. The diagram below shows two different approaches for getting data from the Glider, or a Towfish, to the C&C. The path through Payload Dry Box 1 involves connecting the AUX lines to the electronics within the Dry Box to aggregate the data with other equipment installed in the Payload Dry Box 1. The path through Payload Dry Box 2 is uses jumper wires within the Payload Dry Box 2.

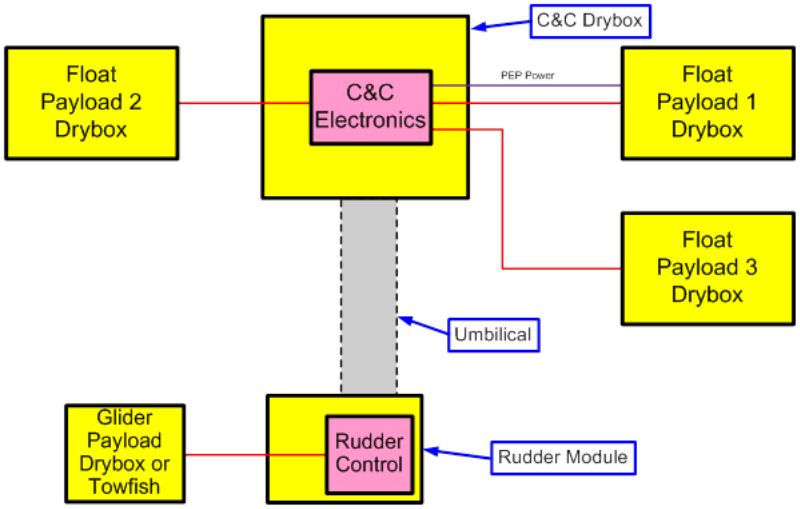
Figure 5.4. Payload Interconnectivity and Communication Diagram



Payload Power Connectivity

Energizing payloads is done using one of the five WGMS controlled power outputs for payloads on the Float and Rudder Module. On the Float, there are three ports labeled PAYLDx and a fourth port that is labeled AUX PWR (PEP Power). There is a male and a female connector on the Rudder Module. The female connector is the Rudder Module payload power connector. Figure 5.5, “Payload Power Routing” shows the AUX PWR port as the purple line and the other payload power paths as red lines. The AUX PWR port is typically connected in parallel with another payload port.

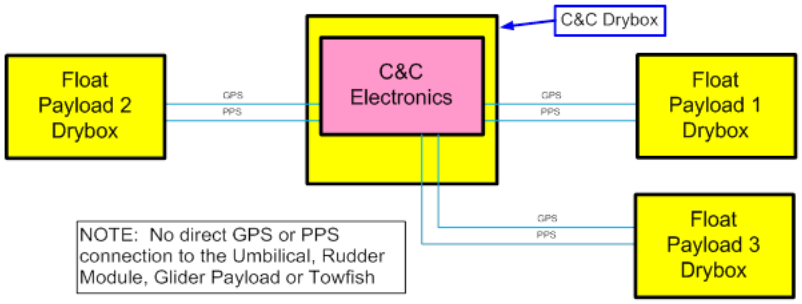
Figure 5.5. Payload Power Routing



GPS and PPS Connectivity

GPS and PPS are routed directly from the GPS unit to Float-mounted payloads through the standard payload connectors. As a customization, those signals can be routed to Glider-attached payloads by using the AUX lines.

Figure 5.6. GPS and PPS Routing



Payload Maximum Currents

There are 3 C&C payload connectors for Float payloads. Each of the C&C Float payload connections is limited to 3 amps.

There is a C&C Payload Extra Power (PEP) port that is limited to 6 amps (see the section called “PEP Port”).

The C&C Umbilical connector provide a power source to the Rudder Module that is limited to 3 amps.

The Rudder Module has a payload connection for Glider or Towfish mounted payloads that is limited to 1 amp.

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

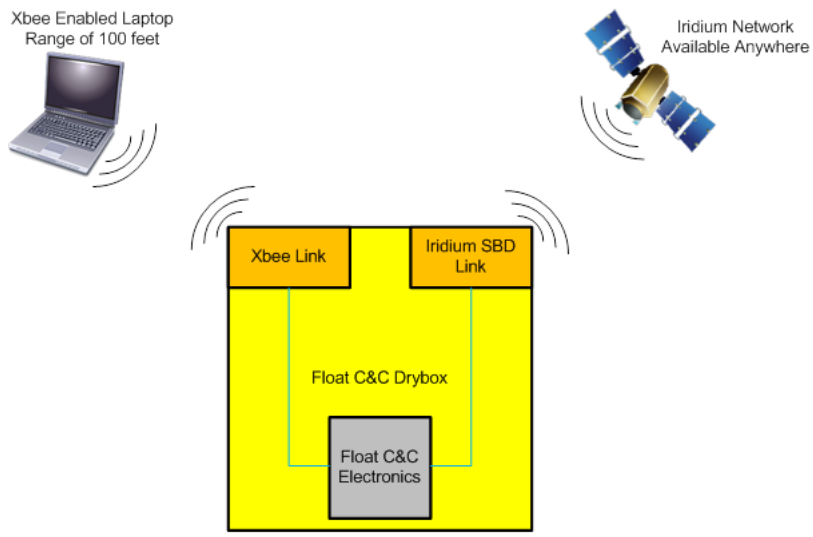
Maximum Total Current Output

The C&C Dry Box output voltage is regulated to 13.2 VDC. The maximum total current allowed for all payloads combined is **10 Amps**.

WGMS Side Connectivity

The Float C&C has two paths for communicating with shore-side applications. vehicles on a mission typically use an Iridium SBD session. The XBee path is used for pre-launch checks and short-range communications.

Figure 5.7. Vehicle Wireless Connectivity Options



Iridium Connectivity

The iridium network is an “always on” satellite-based system with 100% global coverage. This is the main link when on mission and deployed.

XBee Connectivity

XBee is a Zigbee based short-range RF connection that requires a base station to be configured as an XBee relay. This connection has no per use costs associated with it and it consumes less power than the Iridium communication channel. The typical XBee range is 100ft with line of sight. This range can be greatly reduced if objects block the path, including waves.

Command and Control Dry Box

This section describes the interfaces to the C&C Dry Box. This dry box is the core component in navigation and overall vehicle control. This dry box provides connections for payloads, sensors and other navigational components.

Power Key

The power key turns the system on and off. When the power key is pulled out, the system will be shutdown. When the system is shutdown, the batteries are individually disconnected and the solar panels and auxiliary power inputs are disconnected from the system. The system is powered on when the key is fully inserted.

Indicator Light

There is an indicator light next to the power key. This light can be used to verify that the vehicle is ready. The table below shows the potential states of the indicator. Please note that if the XBee communication channel is turn off the Indicator is turned off, because the vehicle is assumed to be out of reach of any users.

Table 5.2. C&C Dry Box Indicator Light States

Indicator Color	State	Description
Off	Off or XBee disabled	System Off Turn on XBee to verify system state if desired
Red	Error	See WGMS or the XBee console to determine the source of the error
Orange	Indeterminate	System has just powered up and has not had enough time to finish system checks and/or GPS not yet fixed
Green	OK	All systems are functional for launch

Unused Connections

All unused connections must have a dummy plug installed on them. If a connection is left open, it is susceptible to corrosion and leakage. Dummy plugs should be installed when testing on a bench to protect the connectors from mechanical damage.

Standard Electrical Connections

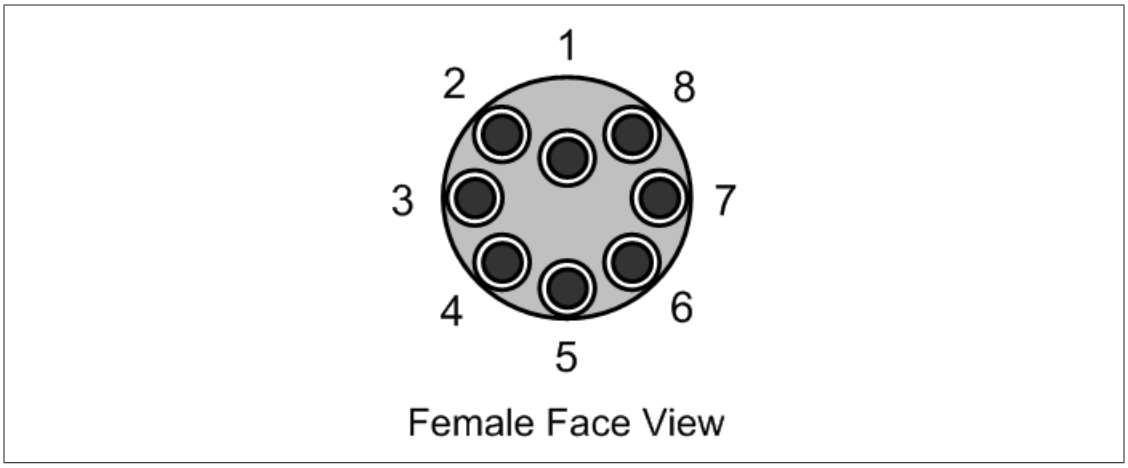
There are two types of watertight connectors used on the Wave Glider: 5-Pin and 8-Pin. The tables below describe the connector type and pin outs for each location.

PAYLD1, PAYLD2, and PAYLD3 Connectors

The three Payload connectors interface to equipment mounted in the Float. These connections provide power, communications with the C&C, connectivity to other payloads including Glider mounted components, and GPS data.

PAYLDx Connector Face

SubConn MCBHRA8F



PAYLDx Pin Out

Pin #	Signal Name	Description
1	PPS	GPS pulse-per-second, using RS-232 levels
2	PWR	13.2 Volts regulated at 3.0A continuous
3	TX	RS-232 Transmit Line
4	RX	RS-232 Receive Line
5	AUX1 or AUX3	Glider Connections; exact connections are determined by payload port
6	AUX2 or AUX4	Glider Connections; exact connections are determined by payload port
7	GND	Ground reference for all pins
8	GPS Data	4800 Baud GGA, ZDA, and VTG, using RS-232 levels, 1 second report rate

PWR and GND

The power pin provides a regulated 13.2 volts at up to 3.0A continuous draw. The port will handle surges of up to 3.5A. Any transient current draw greater than 3.5A for more than 160ms will constitute a fault and cause the output to shut off to protect itself.

When sizing the load, care must be taken in sizing the total input capacitance seen by the power switch. If the capacitance is excessively large, the switch sees the output as being shorted, which causes a fault and the power will turn off to protect itself. The maximum allowable input capacitance for each payload port is ~50,000uF.

If the equipment's transients are larger than 3.5A or the input capacitance is greater than 50,000uF, additional circuitry is required to buffer the power output. This can take the form of an NTC inrush limiter, local batteries, super capacitors, etc.

TX and RX

These communication lines connect directly into the C&C processor. This link is used to send data to other payloads and between the vehicle and WGMS. This connection is RS-232 at 115,200 baud rate. These signals are provided via a MAX232 compatible circuit.

PPS and GPS Signals

The PPS signal is a 4200ns pulse once per second, with positive polarity RS-232 pulses. The GPS data output is in NMEA format with the GGA, ZDA and VTG messages reported either once every 1 second, or every 5 seconds, depending on the release level of the installed C&C firmware. The order of the messages is not fixed. These signals are present on all payload ports if any single payload port has been turned on.

PAYLD1 Power	**PAYLD2 Power**	**PAYLD3 Power**	**GPS/PPS Data Stream to all PAYLDx ports**
Off	Off	Off	Off
On	Don’t Care	Don’t Care	On
Don’t Care	On	Don’t Care	On
Don’t Care	Don’t Care	On	On

Auxiliary Lines

The auxiliary lines provide direct access to other payloads on the Float and are connected to the Glider assembly. The table below shows how each of the lines is connected. Typical applications have these pins connected as two wire communication paths or low speed analog signals.

Signal Name	**UMBILICAL Pin #**	**PAYLD 1 Pin #**	**PAYLD 2 Pin #**	**PAYLD 3 Pin #**
AUX1	5	5	No Connection	5
AUX2	6	6	No Connection	6
AUX3	1	No Connection	5	No Connection
AUX4	8	No Connection	6	No Connection

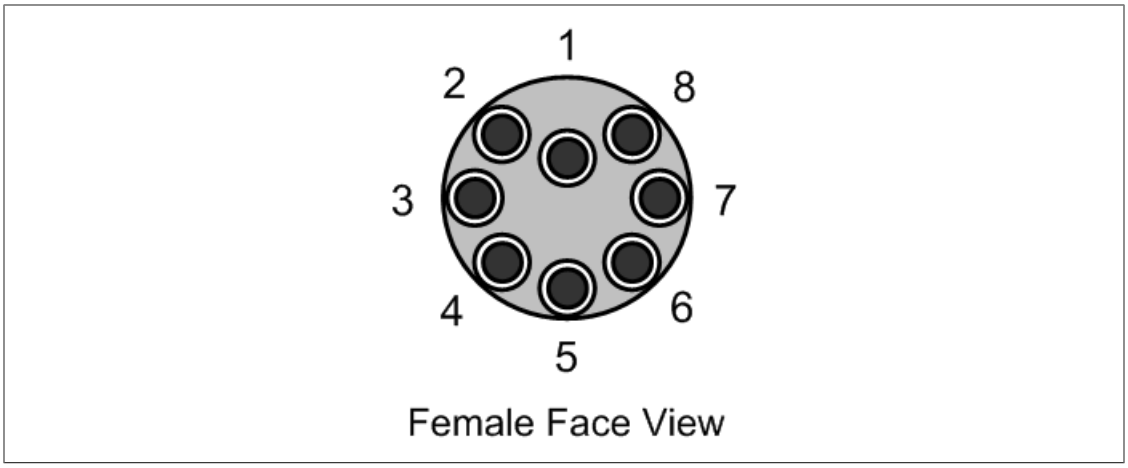
PAYLDx Mating Connector

The PAYLDx ports are designed to mate to a wet mate-able connector from Teledyne or SubConn. The mating connector should be an 8 pin, male, whip type assembly. An example part number from SubConn is PNARA MCIL8M, which is the penetrator assembly used on the Float Forward and Aft Payload Dry Boxes.

UMBILICAL Connector

The Umbilical connection provides power and communications to the Glider and connectivity between the Rudder Module and the payloads.

SubConn MCBHRA8F



UMBILICAL Pin Out

Pin #	Signal Name	Description
1	AUX3	Connects to PAYLD 2 pin 5
2	PWR	13.2 Volts regulated at 3.0A continuous
3	A	RS-485 Positive Line
4	B	RS-485 Negative Line
5	AUX1	Connects to PAYLD 1 and PAYLD3 pin 5
6	AUX2	Connects to PAYLD 1 and PAYLD3 pin 6
7	GND	Ground reference for all pins
8	AUX4	Connects to PAYLD 2 pin 6

PWR and GND

The power pin provides a regulated 13.2 volts at up to 3.0A continuous draw. The port will handle surges of up to 3.5A. Any transient current draw greater than 3.5A for more than 160ms will constitute a fault and cause the output to shut off to protect itself.

When sizing the load, care must be taken in sizing the total input capacitance seen by the power switch. If the capacitance exceeds ~50,000uF, the switch sees the output as being shorted, which causes a fault and the power will turn off to protect itself

If the equipment’s transients are larger than 3.5A or the input capacitance is greater than 50,000uF, additional circuitry is required to buffer the power output. This can take the form of an NTC inrush limiter, local batteries, super capacitors, etc.

The Glider payload port further limits the output current to 1.0A. The voltage drop across the Umbilical also limits the Glider current to no more than 1.0A. All of the current is available at the Float connection.

A and B

These are the communication lines that connect directly into the C&C processor from the Rudder Module. This link cannot be used by any end user equipment.

Auxiliary Lines

The auxiliary lines are provided to give direct access to other payloads on the Float or connected to the Glider assembly. The table below shows how each of the lines is connected. Typical applications have these pins connected as two wire communication paths or low speed analog signals.

Signal Name	UMBILICAL Pin #	PAYLD 1 Pin #	PAYLD 2 Pin #	PAYLD 3 Pin #
AUX1	5	5	No Connection	5
AUX2	6	6	No Connection	6
AUX3	1	No Connection	5	No Connection
AUX4	8	No Connection	6	No Connection

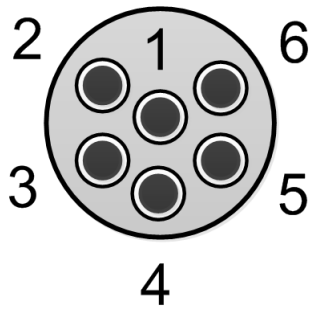
UMBILICAL Mating Connector

The UMBILICAL port is designed to mate to a wet mate-able connector from Teledyne or SubConn. The mating connector should be an 8 pin, male, whip type assembly. An example part number from SubConn is MCIL8M, which is the whip assembly used on the Umbilical assembly.

AUX PWR Connector

The auxiliary power connector interfaces to equipment mounted in the Float that needs more power than provided by the standard payload ports. It also provides an auxiliary input to the charging circuit. An AIS data stream is also provided, if the AIS option is installed and turned on.

SubConn MCBHRA6F



Female Face View

AUX PWR Pin Out

Pin #	Signal Name	Description
1	V-AUX	Auxiliary charging input
2	AIS	AIS data stream
3	PWR	13.2 Volts regulated

4	PWR	13.2 Volts regulated
5	GND	Ground reference for all pins
6	GND	Ground reference for all pins

PWR and GND

The power pin provides a regulated 13.2 volts at up to 6.0A continuous draw. The port will handle surges of up to 8.0A. Any transient current draw greater than 8.0A for more than 160ms will constitute a fault and cause the output to shut off to protect itself.

When sizing the load, care must be taken in sizing the total input capacitance seen by the power switch. If the capacitance exceeds ~50,000uF the switch sees the output as being shorted, which causes a fault and the power will turn off to protect itself.

If the equipment's transients are larger than 8.0A or the input capacitance is greater than 50,000uF, additional circuitry must be added to buffer the power output. This can take the form of an NTC inrush limiter, local batteries, super capacitors, etc.

AIS

This communication line connects directly into the Liquid Robotics AIS receiver option. This connection is RS-232 at 38,400 baud rate. The signal is provided by a MAX232 compatible circuit.

V-AUX

This is the auxiliary power input pin for external power generation and/or storage into the battery charging circuit and payload regulator.

Voltage Range	**Functionality**	**Current Limit**
0-9 Volts	Off/indeterminate	None
9-17	Powers system and small payloads	1.0A
17-22	Full system functionality including charging batteries	3.0A

The 3.0A current limit is based on the cabling and connectors to get into the C&C. The 1.0A current limit is based on power dissipation within the Float C&C itself.

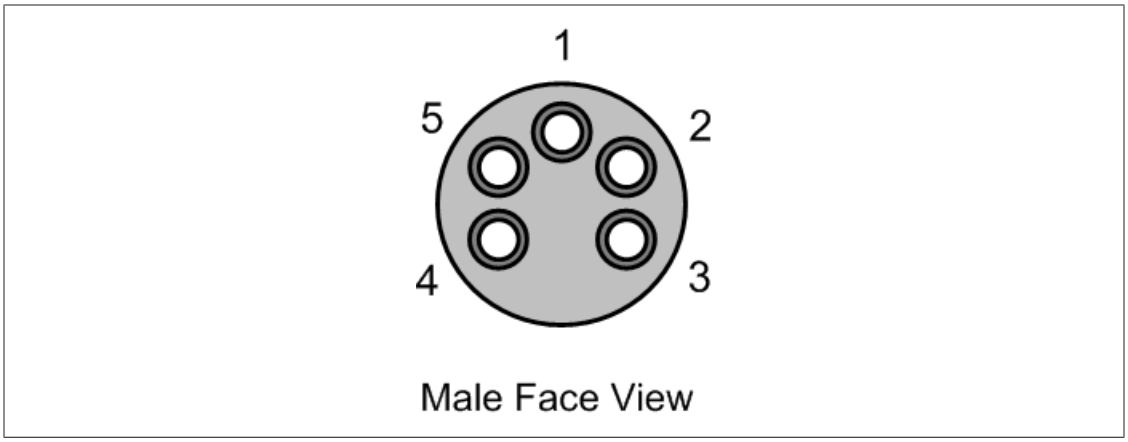
AUX PWR Mating Connector

The AUX PWR port mates to a wet mate-able connector from Teledyne or SubConn. The mating connector must be a 6 pin, male, whip type assembly. An example part number from SubConn is MCIL6M, which is a whip assembly.

SOLAR 1 AFT and SOLAR 2 FORWARD Connectors

The two solar panel connectors interface to solar panels or other power generation equipment. When on shore they often connect to a battery charger.

SubConn MCBHRA5M



SOLAR Pin Out

Pin #	Signal Name	Description
1	V+	Highest voltage on the solar panel
2	V-	Lowest voltage on the solar panel
3	MID	Mid panel voltage used for shade compensation and bypass diodes
4	DETECT	Signal shorted to pin 5 to detect that the panel is present
5	DETECT	Typically connected to ground

Table 11: SOLAR connection pin out

V+, V-, MID

The system is expecting a solar panel input equivalent to a Kyocera KC40T solar panel. The maximum voltage from V- to V+ is 22.0 volts. The minimum voltage required to charge the system at maximum efficiency is 16.8 volts between V+ and V-. The system can also charge from voltages as low as 9.0 volts between V+ and V-, but the charging system takes a 20-30% efficiency hit.

The maximum current per pin is 3.0A.

The MID pin is used when one-half of the panel is shaded. This can be caused by sensor masts on the vehicle or by environmental conditions. This MID point is used to draw power from the half panel that is still in full sunlight. This voltage is then boosted to a voltage usable by the charging circuit. If the power generation source does not provide this point or does not have shading issues, the charging system is still operational.

For information on solar panel specifications and how to avoid shading the solar panels, see the section called “Solar Panels”.

Detect

These two pins are shorted together so the Float C&C can detect the presence or absence of the solar panel. These pins cannot connect to other voltages or inputs, because they can cause damage to the system, or a misreading of the panel presence.

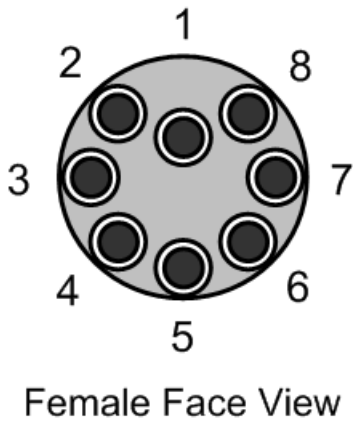
SOLAR Mating Connector

The SOLAR ports are designed to mate to a wet mate-able connector from Teledyne or SubConn. The mating connector should be a 5 pin, male, whip type assembly. An example part number from SubConn is MCIL5F, which is the whip assembly used on the aft and forward solar panels.

WEATHER Connector

The weather station connector interfaces to the Liquid Robotics Airmar PB200 weather station. The pin out is similar to a payload to avoid causing damage to either component, but the power and communication type are not compatible with standard payloads.

SubConn MCBHRA8F



Female Face View

WEATHER Pin Out

Pin #	Signal Name	Description
1	N/C	Not Used
2	PWR	13.2 Volts regulated, 300mA maximum
3	RXD-	RS-422 receive from station
4	RXD+	RS-422 receive from station
5	TXD-	RS-422 transmit to station
6	TXD+	RS-422 transmit to station
7	GND	Ground reference for all pins
8	N/C	Not Used

PWR and GND

The power pin provides a regulated 13.2 volts at up to 300mA continuous draw. The port will handle surges of up to 500mA. Any transient current draw greater than 500mA for more than 160ms will constitute a fault and cause the output to shut off to protect itself.

TX+, TX-, RX+ and RX-

These are the communication lines that connect directly into the C&C processor. This link connects to the weather station using an RS-422 interface. Current systems have all four pins connected, but do not transmit data to the weather station. The Float C&C only listens to the data being reported.

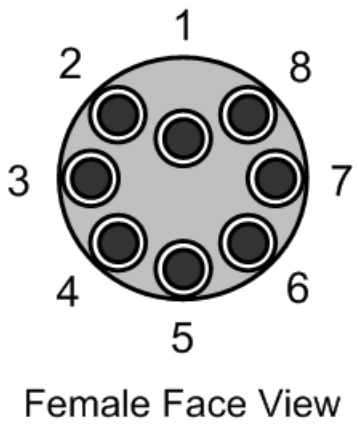
WEATHER Mating Connector

The WEATHER port is designed to mate to a wet mate-able connector from Teledyne or SubConn. The mating connector should be an 8 pin, male, whip type assembly. An example part number from SubConn is MCIL8M, which is the assembly used on the Liquid Robotics Weather Stations Assembly.

SPEED Connector

The water speed sensor connector interfaces to the Liquid Robotics provided Airmar CS4500 ultrasonic water speed sensor. The pin out is such that if a payload is accidentally connected here, no damage will occur to either component, but the power and communication type are not compatible with standard payloads.

SubConn MCBHRA8F



SPEED Pin Out

Pin #	Signal Name	Description
1	GND	Ground reference for all pins
2	N/C	Not Used
3	PWR	8.0 Volts Regulated, 250mA Maximum
4	Data	Paddle wheel water speed sensor input
5	N/C	Not Used
6	N/C	Not Used
7	N/C	Not Used
8	N/C	Not Used

PWR and GND

The power pin provides a regulated 8.0 volts at up to 250mA continuous draw. The port will handle surges of up to 300mA. Any transient current draws greater than 300mA will fold back the current to protect the regulator.

Data

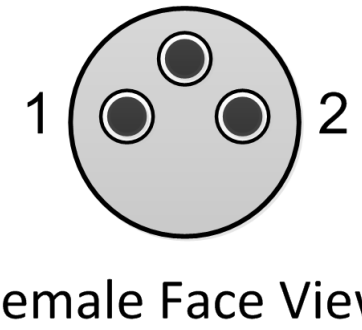
This pin is designed to accept an open collector output compatible with the CS4500 water speed sensor. This pin has a weak pull-up (100k) to 3.3 volts.

SPEED Mating Connector

The SPEED port is designed to mate to a wet mate-able connector from Teledyne or SubConn. The mating connector should be an 8 pin, male, whip type assembly. An example part number from SubConn is MCIL8M, which is the assembly used on the Liquid Robotics Water Speed Sensor Assembly.

LIGHT Connector

The light connector interfaces to a Liquid Robotics provided light mounted on a mast with a weather station.

SubConn MCBHRA2F


LIGHT Pin Out

Pin #	Signal Name	Description
1	PWR	13.2 Volts regulated, 500mA maximum
2	GND	Ground reference for all pins

PWR and GND

The power pin provides a regulated 13.2 volts at up to 500mA continuous draw. The port will handle surges of up to 1.0A. Any transient current draw greater than 1.0A for more than 160ms will constitute a fault and cause the output to shut off to protect itself.

LIGHT Mating Connector

The LIGHT port is designed to mate to a wet mate-able connector from Teledyne or SubConn. The mating connector should be a 2 pin, male, whip type assembly. An example part number from SubConn is MCIL2M, which is used on the Liquid Robotics Light Bar Assembly.

Power System

Batteries

The Float C&C contains a Li-Ion battery pack that supplies power to the payloads and the core system as needed to supplement the solar panel input. The battery pack can store up to 665 Watt Hours of energy. This energy is shared between the core electronics and the various payloads connected to the system.

Core System Power Draw

The core system typically draws 1.5 to 3.0 watts depending on operating conditions (telemetry rates, wave wash-over, etc.). Use this draw when calculating system sustainability.

Payload Regulator

The PAYLDx, UMBILICAL, and AUX PWR ports all share a common regulator. This regulator is limited to 10.0A output current when the batteries are more than 20% full. When the batteries are less than 20% full the regulator's maximum output current declines linearly to 5.0A at empty. The payloads should never exceed these values because this can cause the Rudder Module to go into a low power mode. The tables below are examples of potential current draws. As long as the individual port draws are not exceeded, any combination of currents can flow through each port.

Acceptable System Current Draws

Port	Current
PAYLD 1	1.0A
PAYLD 2	1.0A
PAYLD 3	1.0A
UMBILICAL (via the Rudder Module)	0.5A
AUX PWR	1.0A
Total	4.5A

Acceptable System Current Draws Above 20% Battery Charge

Port	Current
PAYLD 1	2.0A
PAYLD 2	2.0A
PAYLD 3	2.0A
UMBILICAL (via the Rudder Module)	0.5A

AUX PWR	3.0A
Total	9.5A

Unacceptable System Current Draws

Port	Current
PAYLD 1	2.0A
PAYLD 2	2.0A
PAYLD 3	2.0A
UMBILICAL (via the Rudder Module)	0.5A
AUX PWR	5.0A
Total	11.5A

Power Input

The Float C&C is expecting a solar panel or similar type power source to provide energy to recharge the batteries. To minimize losses, the main path to charge the batteries is unregulated. For this reason, the power input to the batteries must fall within the ranges specified in the SOLAR connections. There is a boost regulator built in to accommodate solar panel shade compensation, but the charging efficiency typically drops by 20-30% by using this stage. The shade compensation circuit is always active and the Float C&C is continuously checking to see if it is needed. An algorithm periodically checks to see whether unregulated or boosted voltage modes provide more power to the system and then leaves the most efficient mode in place until the next time it checks the voltage mode.

There is also an auxiliary charging input available from the AUX PWR port. Refer to the AUX PWR port section for specifications and requirements.

External Charging

Overview

This section describes the Liquid Robotics Wall Charger and provides the electrical specifications necessary to build a replacement charger.

Charger Connections

An external charger can be connected to either (or both) of the Float C&C solar panel connectors. This allows charging or “topping-off” of the batteries pre-mission, or when bench testing. For charging to occur, turn on the Float C&C by inserting the power key.

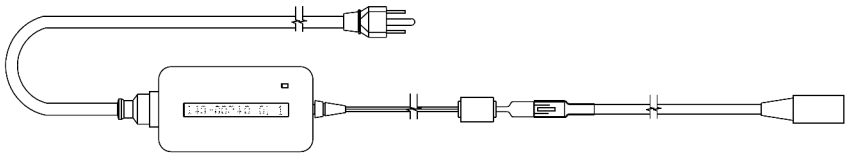
The maximum charging current for each connection is internally limited to 3.0 Amps, regardless of what the charger is capable of supplying. For faster charging, connect a charger to each of the two solar panel connectors.

When using an external charger it is recommended that all payloads and sensors be turned off. This is because the charging energy is shared between the batteries and any enabled payloads or sensors, but the 3.0 amp input limit still applies. The wall charger can add a significant amount of switching power supply noise that may affect the operation of the payloads or sensors.

Liquid Robotics-Supplied Charger

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

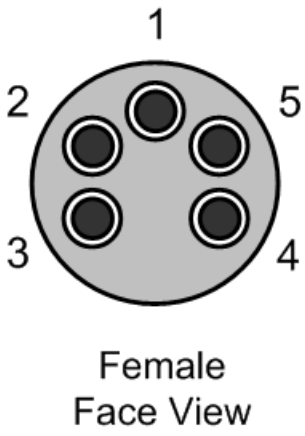
Figure 5.8. Liquid Robotics External Charger



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 9-12 VDC. To minimize charging time, the charger should be capable of delivering at least 3.0 Amps. The connector required to mate with the solar panel connector is an MCBHRA5F SubConn 5-pin Female type, with the pin out listed below.

Figure 17: Charger Connector Face

SubConn MCBHRA5F


Charger Connector Pin Out

Pin #	Signal Name	Description
1	V+	Connect to power supply positive pin
2	GND	Connect to power supply negative pin
3	GND	Connect to power supply negative pin
4	DETECT	Signal shorted to pin 5 to detect that the panel is present

5	DETECT	Typically connected to ground

Power Considerations

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

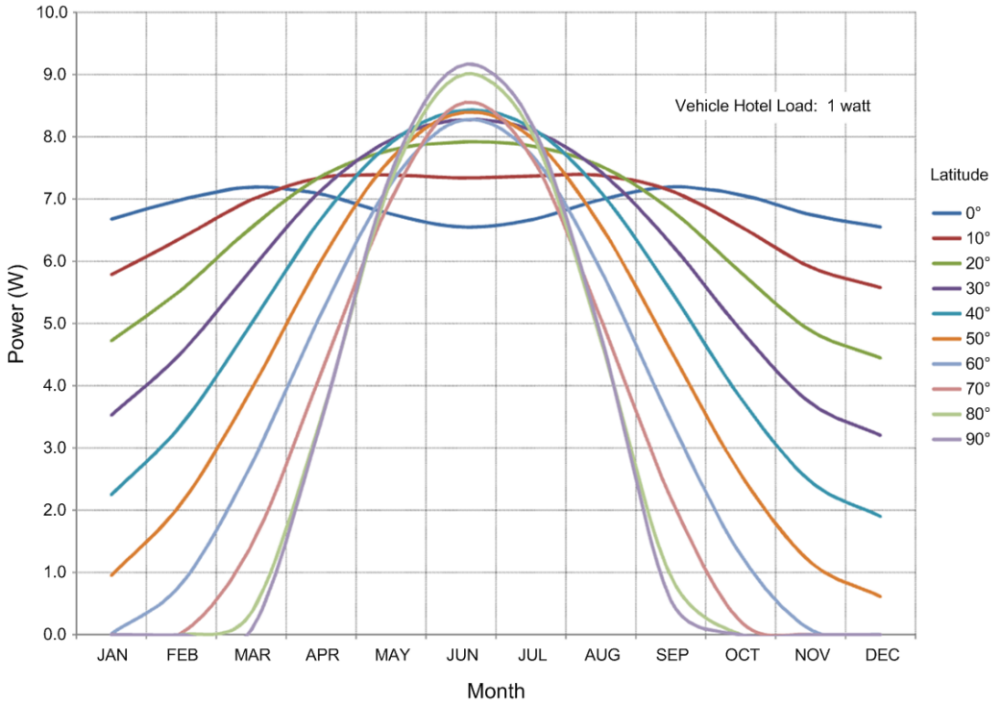
Since the Wave Glider derives all of its long-term operating power from the solar panels, the average energy usage by the C&C Dry Box plus payload(s) should not exceed the average energy input from the solar panels. Allowance also has to be made for the possibility that, due to storms and cloud cover,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

Sustainable Payload Power

Since the Wave Glider™ derives all of its long-term operating power from the solar panels, the average energy usage by the C&C Dry Box 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 average for several days at a time.

Figure 5.9, “Solar Power Relative To Time and Location” 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 5.9. Solar Power Relative To Time and Location



C&C Dry Box Power Draw

The C&C Dry Box runs off of the internal battery pack with a theoretical maximum capacity of 650 Wh. The amount of power the C&C box draws depends on electrical activity.

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

Battery System Power

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

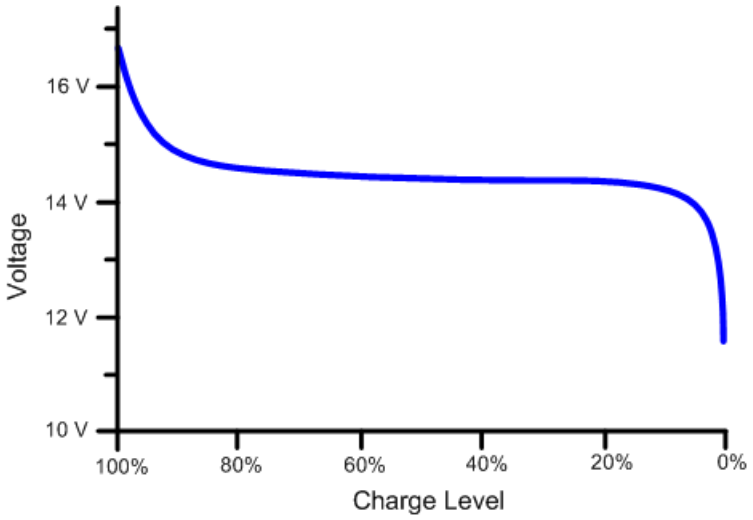
Battery Max Voltage	16.6 Volts
Maximum Continuous Current	20 Amps
Energy Capacity (fully charged) [theoretical maximum]	665 Watt-hours

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

Battery Charge Level

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

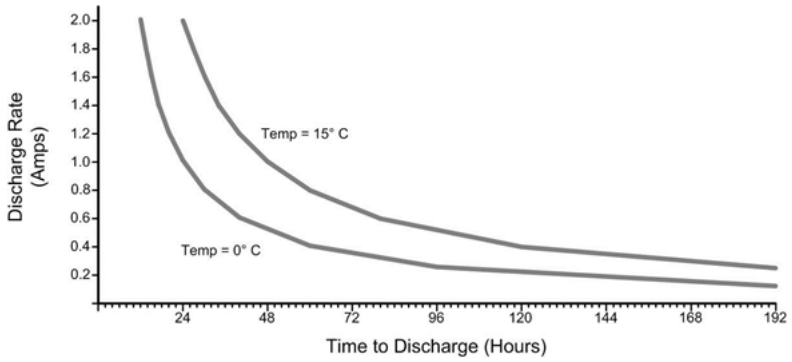
Figure 5.10. Battery Charge Level Relative to Voltage



Battery Discharge Characteristics

Assuming the batteries are fully charged, but there is no input from the solar panels, the discharge time will depend upon the Wave Glide current draw as follows:

Figure 5.11. Battery Discharge Time vs. Current Draw

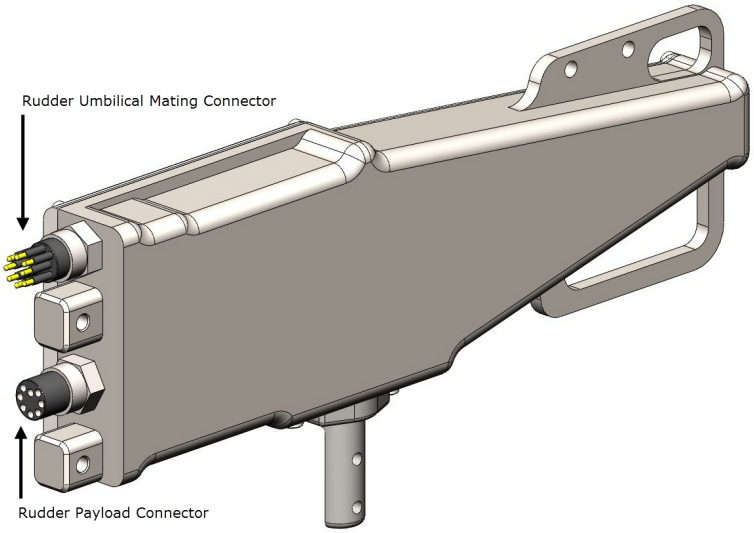


Rudder Module

Overview

This section describes the interfaces to the Wave Glider Rudder Module. The Rudder Module is the core component in compass based navigation and a control point for Glider based payloads. The male (upper) connection is intended to interface through the Umbilical to the C&C Dry Box. The female (lower) connection is intended to interface to Glider or Towfish payloads.

Figure 5.12. Rudder Module



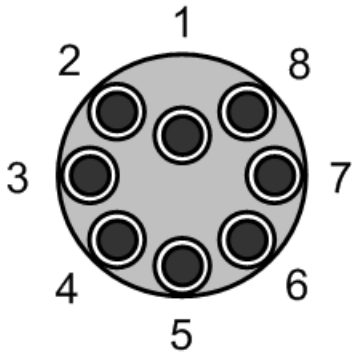
Unused Connections

Unused connections must have dummy plugs installed on them. If a connection is left open, it is susceptible to corrosion and leakage. Dummy plugs should be installed when testing on a bench to protect the connectors from mechanical damage.

Rudder Umbilical Connector

The Umbilical connection provides power and communications to the Rudder Module from the Float. It also provides a pass through of the auxiliary lines for Glider payload connectivity through the Rudder Module to the Float payloads.

SubConn MCBHRA8F



Female Face View

Rudder Umbilical Pin Out

Pin #	Signal Name	Description
1	AUX3	Connects to PAYLD 2 pin 5
2	PWR	10-17 volts, current is limited to 3.0A by connector
3	A	RS-485 Positive Line
4	B	RS-485 Negative Line
5	AUX1	Connects to PAYLD 1 and PAYLD3 pin 5
6	AUX2	Connects to PAYLD 1 and PAYLD3 pin 6
7	GND	Ground reference for all pins
8	AUX4	Connects to PAYLD 2 pin 6

PWR and GND

The power pin accepts 10-17 volts at up to 3.0A continuous draw. The Rudder Module itself will typically draw 10-50 milliamps continuously and up to an additional 1.0A can be passed to the payload. The voltage drop across the Umbilical also limits the Glider current to no more than 3.0A.

A and B

These are the communication lines that connect directly into the Command and Control processor from the Rudder Module. This link cannot be used by any end user equipment.

Auxiliary Lines

The auxiliary lines provide direct connections between Float mounted payloads and payloads attached to the Glider or a Towfish. The table below shows how each of the lines is connected. Typical applications have these pins connected as two wire communication paths or low speed analog signals.

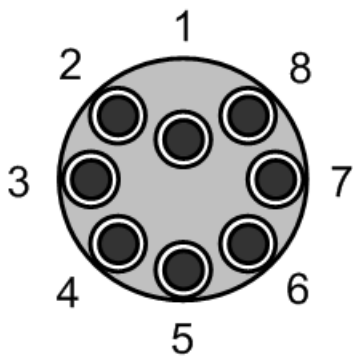
Signal Name	UMBILICAL Pin #	PAYLD 1 Pin #	PAYLD 2 Pin #	PAYLD 3 Pin #
AUX1	5	5	No Connection	5
AUX2	6	6	No Connection	6
AUX3	1	No Connection	5	No Connection
AUX4	8	No Connection	6	No Connection

Rudder Umbilical Mating Connector

The Rudder Module Umbilical port is designed to mate to a wet mate-able connector from Teledyne or SubConn. The mating connector should be an 8 pin, female, whip type assembly. An example part number from SubConn is MCIL8F, which is the whip assembly used on the Umbilical assembly.

Rudder Payload Connector

[To be consistent with the description of the Rudder Umbilical connector, I think we should be describing this Rudder Payload connector as a MALE rather than a female connector.] The Rudder Payload connector interfaces to payloads attached to the Glider or on a Towfish. These connections provide power to the payload and connectivity to the Float mounted components. This connection is designed to accept standard payload connections without damaging either component. Functionality of the rudder payload connection is limited in comparison to a Float payload connection. There are no dedicated GPS, PPS or data path connections. If GPS, PPS or data path connections are required, they must be provided using the auxiliary lines.

SubConn MCBHRA8F


Rudder Payload Pin Out

Pin #	Signal Name	Description
1	AUX3	Connects to PAYLD 2 pin 5
2	PWR	Umbilical Power with a 0.5A continuous limit
3		No Connect
4		No Connect

5	AUX1	Connects to PAYLD 1 and PAYLD3 pin 5
6	AUX2	Connects to PAYLD 1 and PAYLD3 pin 6
7	GND	Ground reference for all pins
8	AUX4	Connects to PAYLD 2 pin 6

PWR and GND

The power pin provides a pass through of the UMBILICAL power from the Float C&C with a 0.5A continuous draw limit. The port will handle surges of up to 1.0A. Any transient current draw greater than 1.0A for more than 160ms will constitute a fault and cause the output to shut off to protect itself.

When sizing the load, care must be taken in sizing the total input capacitance seen by the power switch. If the capacitance is excessively large, the switch sees the output as being shorted, which causes a fault and the power will turn off to protect itself. The maximum allowable input capacitance for the payload port is ~500uF.

If the equipment's transients are larger than 1.0A or the input capacitance is greater than 500uF, additional circuitry is required to buffer the power output. This can take the form of an NTC inrush limiter, local batteries, super capacitors, etc.

Auxiliary Lines

The auxiliary lines provide direct connections between Float mounted payloads and payloads attached to the Glider or a Towfish. The table below shows how each of the lines is connected. Typical applications have these pins connected as two wire communication paths or low speed analog signals.

Signal Name	**Payload Pin #**	**PAYLD 1 Pin #**	**PAYLD 2 Pin #**	**PAYLD 3 Pin #**
AUX1	5	5	No Connection	5
AUX2	6	6	No Connection	6
AUX3	1	No Connection	5	No Connection
AUX4	8	No Connection	6	No Connection

Rudder Payload Mating Connector

The Payload port is designed to mate to a wet mate-able connector from Teledyne or SubConn. The mating connector should be an 8 pin, male whip type assembly. An example part number from SubConn is MCIL8M, which is a whip assembly.

Chapter 6. Vehicle Communications and Control

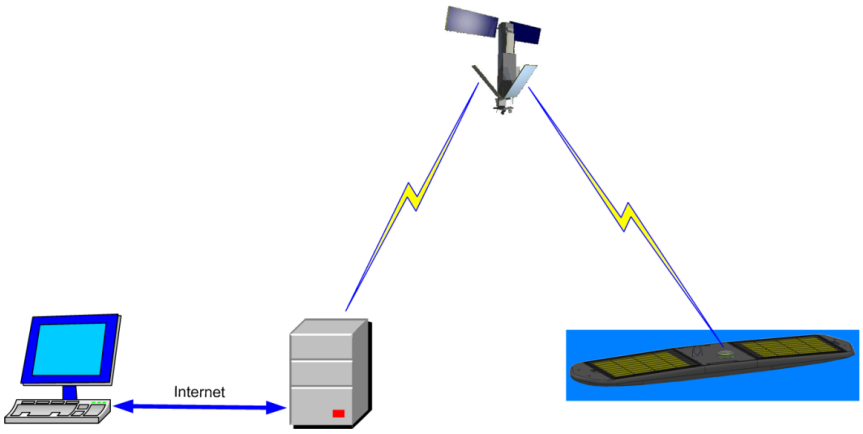
Communications

An overview of the most commonly used Wave Glider communications methods are described in this section.

Iridium-Internet Link

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 6.1. WGMS to Vehicle Iridium Communications

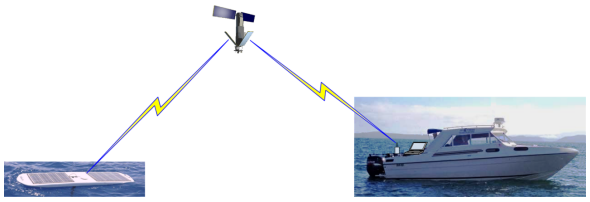


Iridium communications are transmitted in variable-length short burst data (SBD) packets. Communication can also be accomplished using a “Relay” connection that bypasses the Iridium system. Relay communication is similar to Iridium communication except that the data is sent directly to the vehicle, rather than through the Iridium constellation. For more information, see the section called “WGMS Components”.

Mobile-to-Mobile Link

Mobile-to-Mobile communication provides an alternative to the Web-based Iridium link. In this case, the controlling computer is directly linked to an Iridium modem.

Figure 6.2. Mobile-to-Mobile Communications

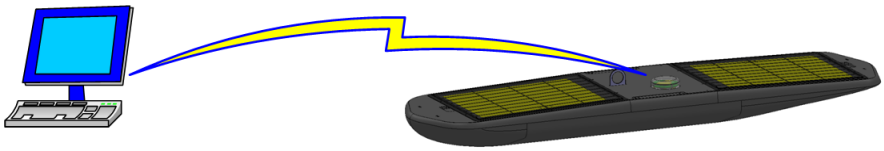


The Mobile-to-Mobile link allows sophisticated control of one or more vehicles from a ship or other remote site. The direct link also provides a more secure method of communications. For more information see the section called “Mobile-to-Mobile Deployment”.

XBee Radio Link

The Wave Glider has an XBee short-range radio. It can be used for communications between a ship and the vehicle, and as a convenient link for on-shore development and testing of Wave Glider systems.

Figure 6.3. On-Shore XBee Communications



XBee communication with the vehicle is established through the XBee Relay software. The XBee Relay has two interfaces:

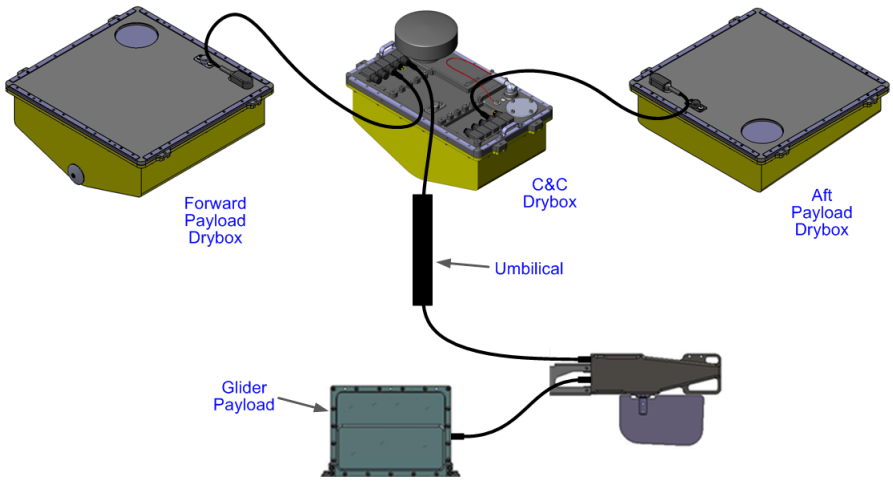
- A text-based terminal emulation to communicate with the vehicle.
- An extension of WGMS communications.

The terminal emulation allows the user to select from a menu of command options; responses and status from the vehicle are displayed in text format. If the XBee Relay software is also in contact with WGMS over the Internet and VPN, WGMS can also issue commands and receive responses. For more information see the section called “Payload Development and Testing Using XBee”.

Payload Communications

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

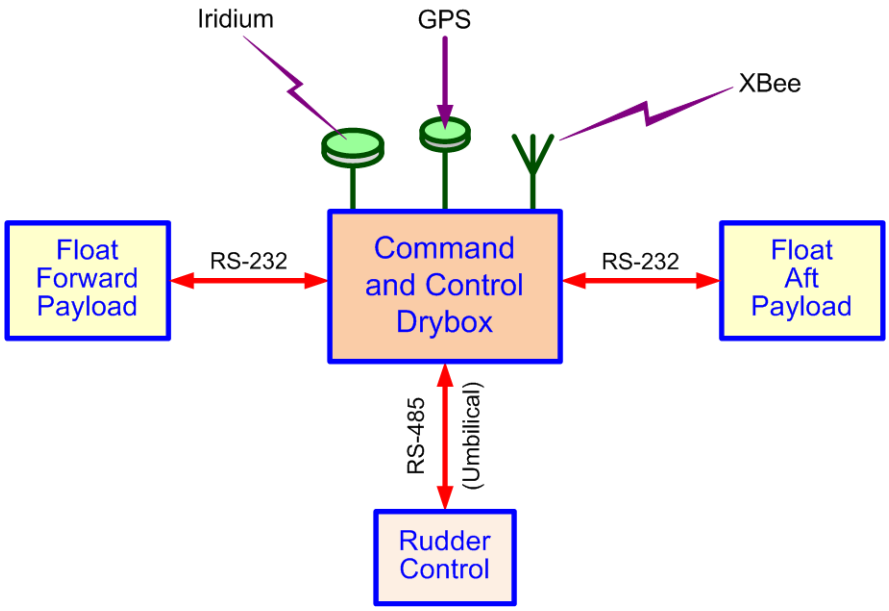
Figure 6.4. Payload Communications Connections



Communications Management

All primary communications to and from the vehicle are managed by the electronics contained in the C&C dry box. Likewise, central control of the vehicle systems is handled exclusively by the C&C electronics.

Figure 6.5. Wave Glider Command and Control



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 on GPS data.

Figure 6.6. GPS Satellite Constellation



GPS fixes are typically obtained every 1 seconds. However, there can be short periods (usually less than a minute) when no signal is received.

C&C Functions

This section provides an overview of the various C&C functions.

Telemetry and Polling

Via the Iridium communications channel (or a Relay), the C&C automatically sends telemetry data at regular intervals. Telemetry data includes the position and general status of the vehicle, as well as any alarms. There are different telemetry data types, and each type can be sent out on a different schedule.

The C&C also sends periodic polling messages to the payloads. These messages can convey information to the payloads and they provide payloads with the opportunity to send messages to the C&C.

It is important to understand that all communications with the C&C are initiated by the C&C based on C&C polling and telemetry.

Command Processing

After the C&C initiates communications by polling, it can receive commands for controlling vehicle operations and for reporting the status of vehicle systems.

Wave Glider commands typically used for development and testing include:

- Make some change to the vehicle (for example, move rudder, turn a light on).
- Set system parameters (for example, telemetry rates).
- Request vehicle systems status (for example, battery level).

- Communicate with a payload.

Additional commands generally used during on-water [off shore?] operations include:

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

The vehicle response to commands depends on the communications method, as described in the following sections.

Payload Message Handling

The C&C receives messages from payloads. The C&C action will depend on where the message is addressed:

- 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, the C&C will send it out in XBee format (then the XBee Relay will convert it to TCP IP packets that are compatible with Iridium packets.

Command and Message Responses

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

- Acknowledge (ACK) responses that let the requesting system know that the command/message was received. The ACK response may also contain requested status information.
- Not-Acknowledge (NAK) responses that 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.

Data Types

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

Telemetry

The vehicle automatically sends telemetry data packets at regular intervals. Telemetry Data indicates the position and the general status of the vehicle, as well as if there are any alarms. There are different telemetry data types, and each data type may be sent out on a different schedule. For more information on telemetry data types, see Chapter 10, *Vehicle Telemetry*.

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

Responses

When a command or payload message is sent to the vehicle, the C&C will respond in one of two ways:

- Acknowledge (ACK) responses that let the requesting system know that the command was received. Depending on the type of command/message, the ACK response may also contain requested status information.
- Not-Acknowledge (NAK) responses that 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.

Communication Packets

All communication to and from the vehicle is done with information packets containing a fairly small number of bytes (approx. 200 to 500 bytes). Note that Iridium and Relay packets contain other packets that contain the message data and an error check.

Communication within or between payloads is also done with data packets, but the message types and formats are different than what is used for Iridium and Relay packets.

Figure 6.7. Level 3 Iridium or Relay Data Packet Format

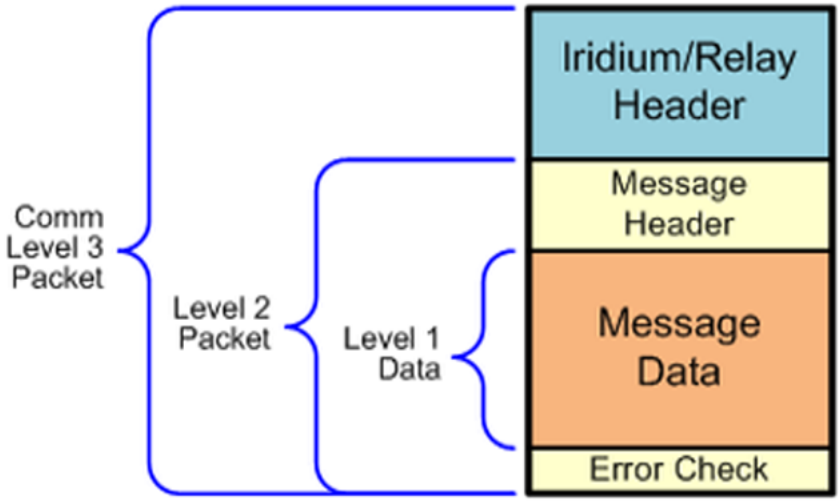


Figure 6.8. Level 3 Encrypted Iridium or Relay Data Packet Format

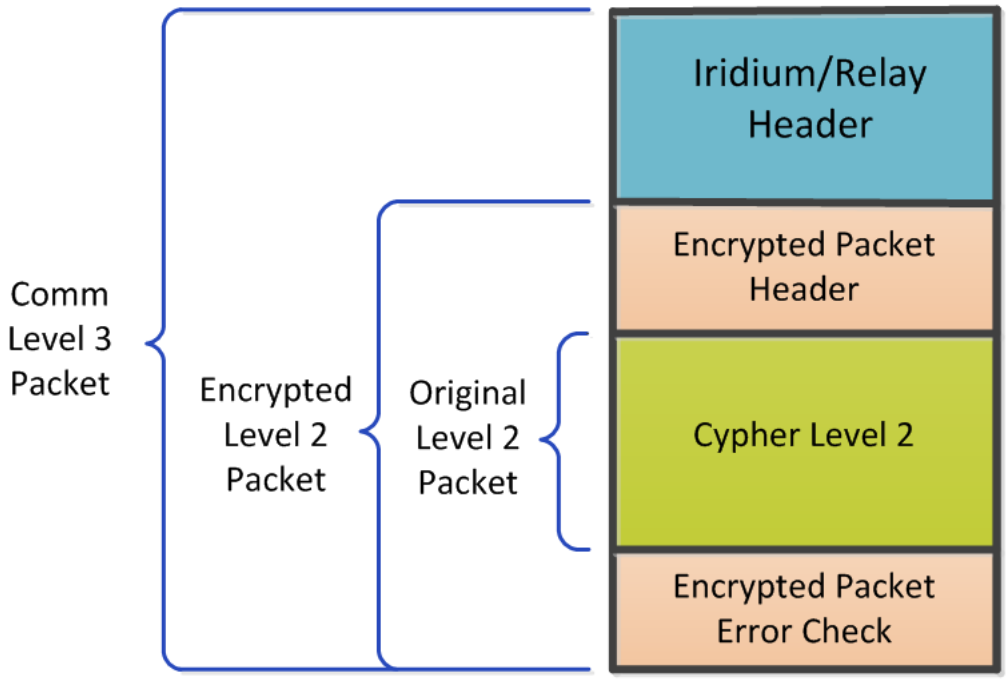
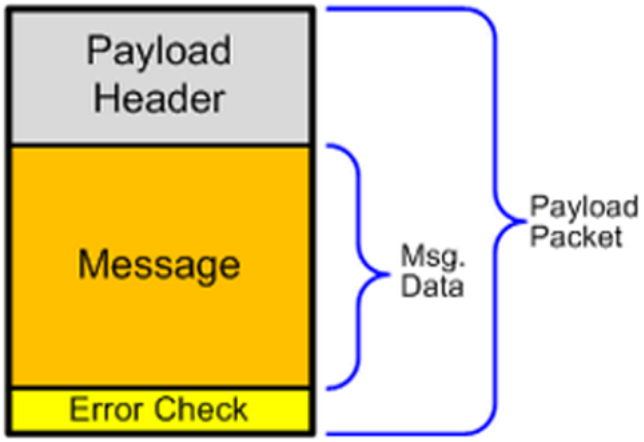


Figure 6.9. Level 2 Payload Data Packet Format



As shown in the figure, the Iridium and Relay packets have the same format, but the header data is different. For more information on Iridium and Relay data packets, see the section called “Iridium (Direct IP) and Alternate Iridium Relay Communications”.

The Message in a payload packet supports many different types of data, including data that can be put into an Iridium or Relay packet. For more information on payload message data types, see Chapter 11, *Payload Communications Protocol*.

All packet communications carry two CRC-16 bytes that can be checked by the receiving device. See Appendix D, *CRC-16 Calculation* for more information. All Iridium communications sent to WGMS have

the error code checked on receipt. Payload message packets may or may not be checked, at the option of the message recipient.

Addresses

Addresses are values used to route messages between the C&C, payloads and other devices (for example, 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. For more information, see the section called “Addresses”.

XBee

XBee communications with the vehicle are performed by the XBee Relay software running on a computer within range of the XBee module. The relay software provides two interfaces: a terminal emulation suitable for initiating interactive commands and monitoring status, and an extension of WGMS communications via XBee. For information on XBee commands see Chapter 13, *XBee Communications*.

Wave Glider Management System

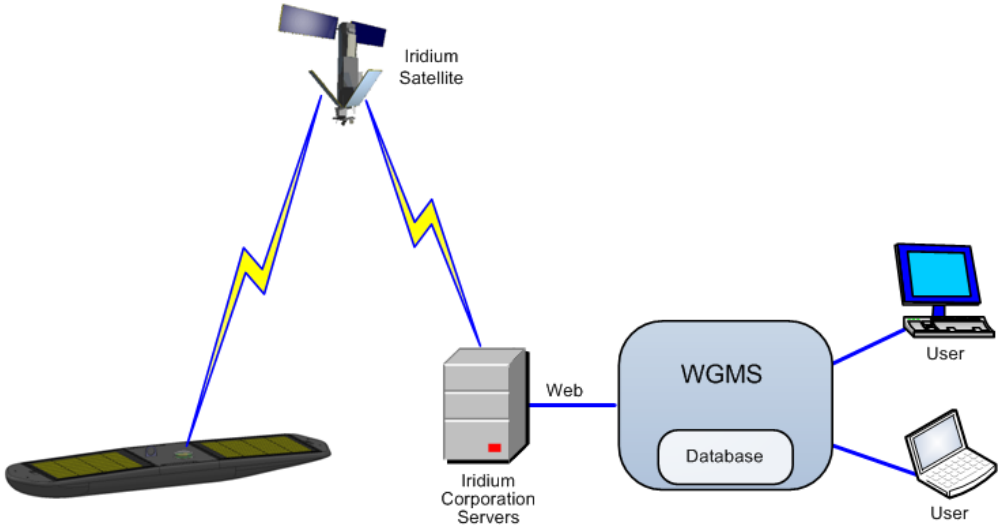
The Wave Glider Management System (WGMS) provides browser and programmatic interfaces to allow users and applications to send commands to the Wave Glider, including its payloads, and to receive data from the Wave Glider and its payloads. WGMS is easily configured to support a variety of deployment scenarios. WGMS interfaces with the Iridium Satellite Network over the Internet, or by way of a local Iridium modem. WGMS defines a browser UI, handles inputs from the browser UI, and handles Iridium and Relay communications to and from the vehicle. WGMS also manages the storage and retrieval of data in the Database. It uses the Database to configure the browser UI, to store vehicle data, and to recover stored data for presentation on the browser UI.

WGMS Web and Database communications are done using XML entities. For more information see Appendix B, *XML, Soap and Entities*.

WGMS Overall Structure

Figure 6.10, “Basic Wave Glider Management System” shows the overall operational structure of WGMS.

Figure 6.10. Basic Wave Glider Management System



The figure above shows one of several possible WGMS configurations. Other configurations are described in the section called “Common WGMS Scenarios”.

Command Communications

For most applications, commands are sent from the 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 (for example, 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.

Vehicle Responses

Vehicle responses such as telemetry are returned from the Wave Glider through an inverse of the command communications mechanism. Responses are sent to the appropriate Relay (when on mission, typically the Iridium Server), which forwards them 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 the configuration. The WGMS Kernel stores the data in its database, where it can be accessed for evaluation.

WGMS Components

Figure 6.11, “Basic WGMS Components” illustrates a full WGMS operational configuration, indicating all communications, routing and processing options. Specific deployments, such as payload development or mobile-to-mobile solutions, will typically use a subset of the modules represented by the illustration.

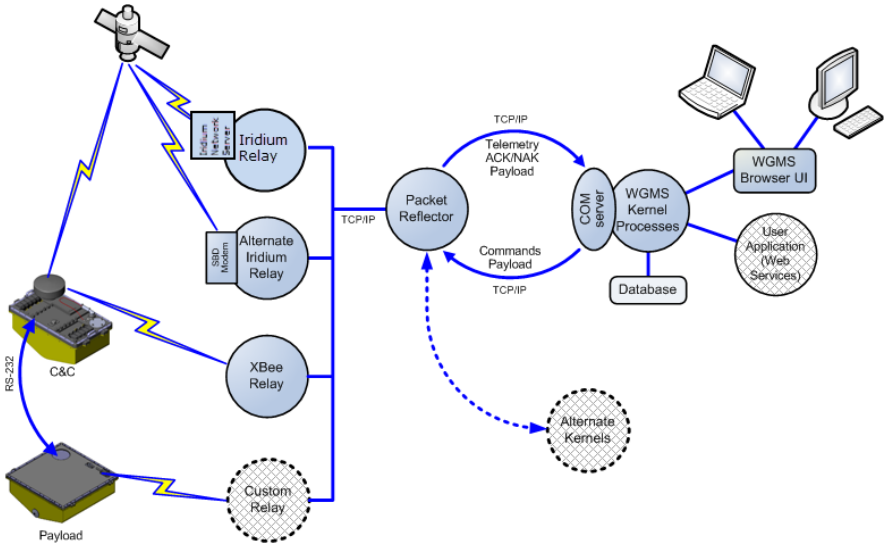
Circles represent software modules.

Blue circles indicate Liquid Robotics-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 using TCP/IP and need not all be resident on the same computer.

Figure 6.11. Basic WGMS Components



The following sections describe the components of WGMS.

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

For more information on the WGMS user interface, see the Liquid Robotics *Wave Glider User Manual*. The WGMS user interface both controls and monitors vehicle operations. It communicates with the Internet using HTTP(S) and translates to and from XML to interface with Web Services.

User Application

As an alternative to the conventional user interface, the WGMS Kernel can be accessed by a non-Liquid Robotics 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 WGMS. It manages:

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

WGMS Kernel

The WGMS Kernel controls the WGMS user interface, Web Services, communications, and database access. The Kernel can also use email for sending alarms to users.

Data

Primary services include filters for access control, data consistency, and concurrency management. Communication with applications is accomplished using Web Services technologies. Browser communication relies on standard HTML and JavaScript.

Com Server

The COM Server interfaces between the Packet Reflector and the WGMS kernel.. 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.

Packet Reflector

The Packet Reflector is a bidirectional router for all Wave Glider data packets sent through their respective relays by any number of WGMS systems. Applications "subscribe" to communicate with the Packet Reflector. Normally, it routes packets between the Com Server and Relays, but it can also route messages to other applications. The Packet Reflector normally communicates using TCP/IP when commands are sent to a Wave Glider, but defaults to email if the Internet service connection is down.

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.

Iridium Relay

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

Communication Relays

Communication relays can be one of four types: Iridium, Alternate Iridium, XBee, or Custom). Their usage is described in the section called “Common WGMS Scenarios”.

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 the communication interface they control to Packet Reflector compatible TCP/IP packets, or by converting TCP/IP packets from the Packet Reflector to a format that is compatible with a specific communication interface. Specific duties may include fragmenting and reassembling packets, managing queues and driving the physical interface.

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

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 Liquid Robotics *WGMS System Administration Guide*.

Your circumstances may require combining more than one scenario. For example, you might run a private shared copy of WGMS that must simultaneously support payload developers and Wave Gliders deployed on a mission. The Liquid Robotics Support Organization is available to advise you on configuration options specific to your needs.

Liquid Robotics-Hosted WGMS

A popular deployment option is letting Liquid Robotics 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 completed through the commercial Iridium system. The Liquid Robotics Operations staff configures the system for you and administers your system and your Wave Gliders on a 24x7 basis. Figure 6.12, “Liquid Robotics-Hosted WGMS” and Figure 6.13, “Multiple Organizations Using Liquid Robotics-Hosted WGMS” illustrate this option.

Figure 6.12. Liquid Robotics-Hosted WGMS

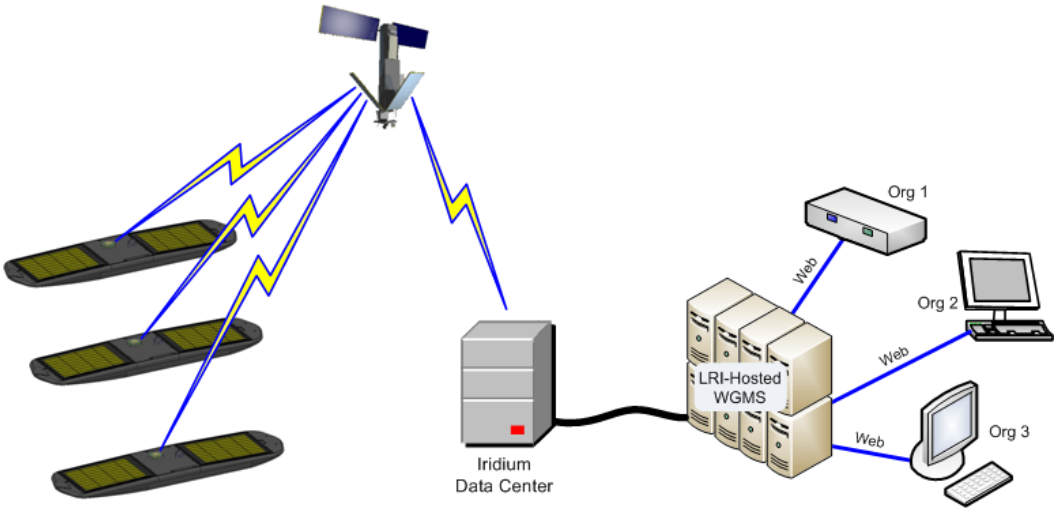
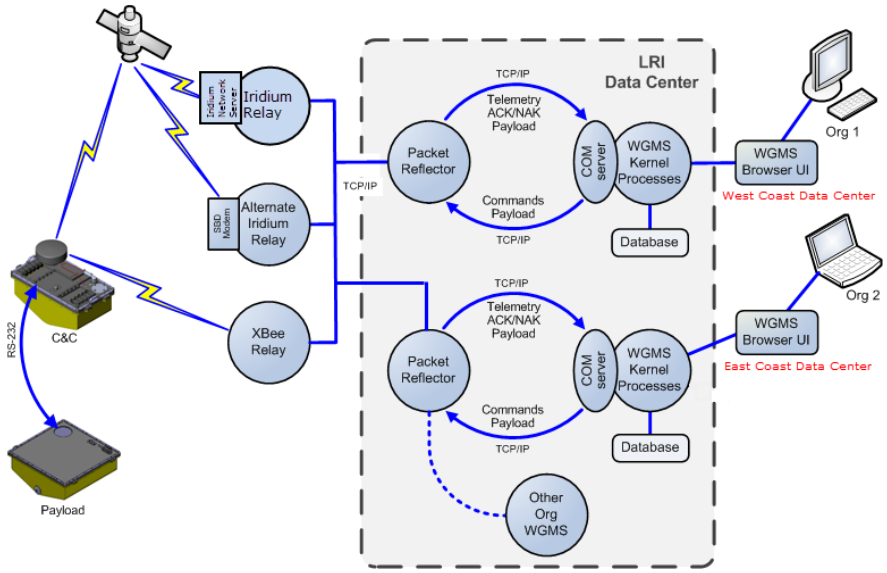


Figure 6.13. Multiple Organizations Using Liquid Robotics-Hosted WGMS



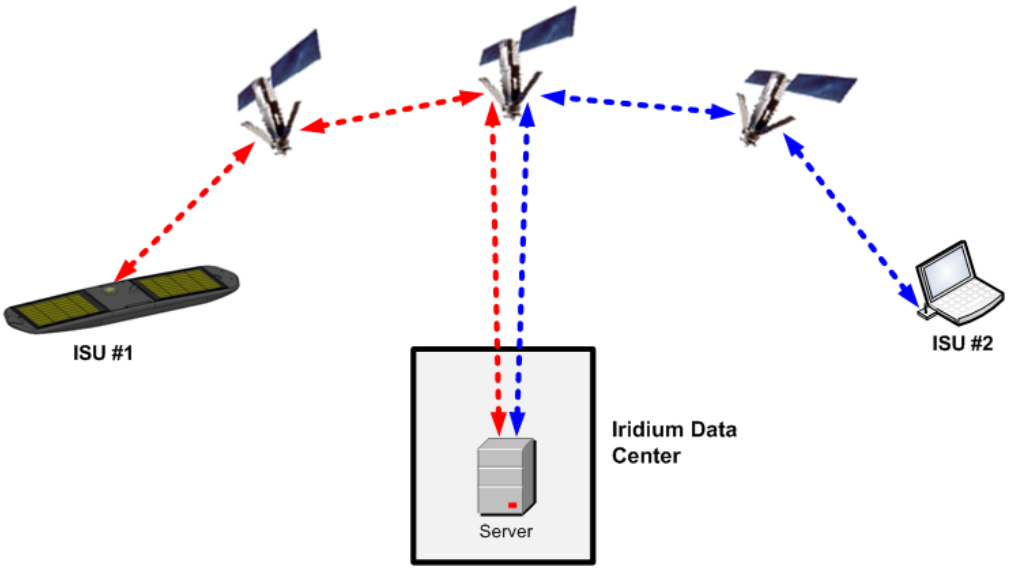
Mobile-to-Mobile Deployment

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. Communication between WGMS elements is still done with 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 can also run completely independent of the Internet. One consequence of the mobil-to-mobil stack running independent of the Internet is that Internet-based services, such as email and Google Maps will not be available. However, WGMS offers an option for using an offline map source to render maps.

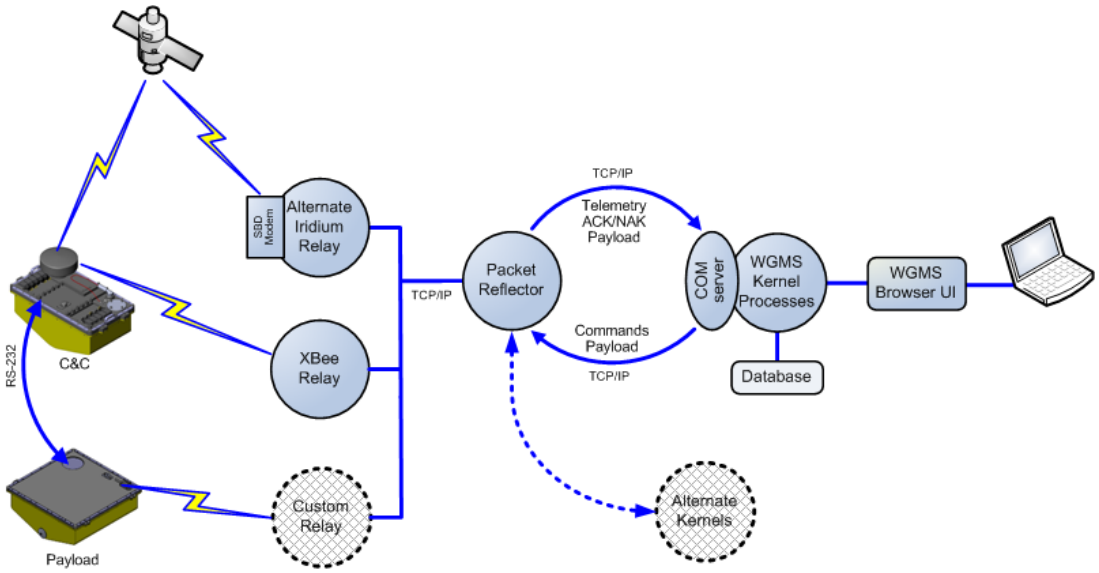
Iridium communications in the M2M scenario use a protocol called "ISU-ISU", (ISU = "Iridium Subscriber Unit"). The ISU-ISU protocol is very similar to the Direct IP protocol except that the sent messages are queued in servers in the Iridium Corporation data center until the second receiving modem requests them. This is illustrated in Figure 6.14, "Iridium Communications Corp ISU-ISU".

Figure 6.14. Iridium Communications Corp ISU-ISU



In a Mobile-to-Mobile configuration, the paired Iridium modem is connected to the M2M host computer using a serial interface. Messages from the Wave Glider go up to the Iridium satellite, back down to the local Iridium modem, and 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 messages to the WGMS Kernel and the 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, applications that interact with WGMS using the web services API 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 6.15, "Mobile-to-Mobile Configuration" .

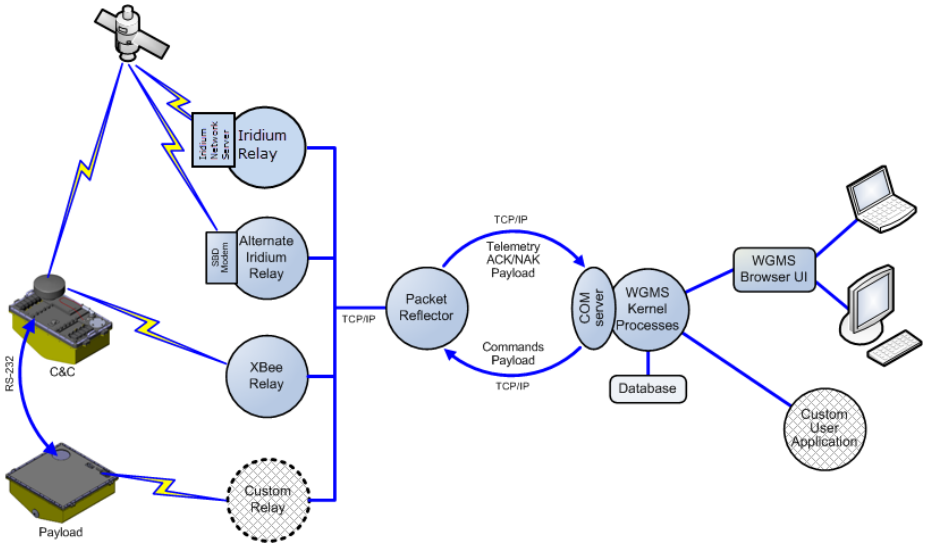
Figure 6.15. Mobile-to-Mobile Configuration



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 6.16. Self-Hosted WGMS Configuration



Payload Development and Testing Using XBee

A common work-flow 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 use of an XBee modem with use of Iridium services. This use case is usually combined with one of the previous three use cases – Liquid Robotics-hosted, Self-Hosted or Mobile-to-Mobile.

For a Wave Glider on an offshore mission, the only practical means of communications is by satellite, typically the Iridium Communications Inc. constellation. This works well with the unobstructed view of the sky available from the surface of the sea, but 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 that communicates through a serial port with the Liquid Robotics-provided XBee Relay.

Figure 6.17. XBee for Payload Testing

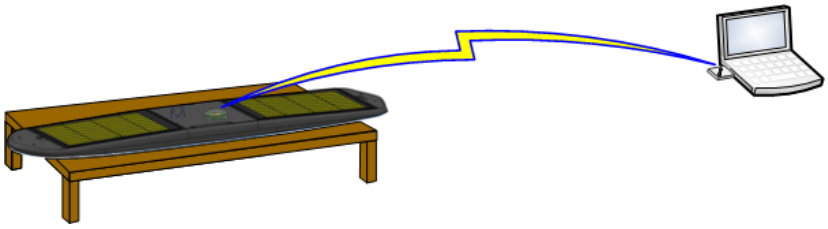
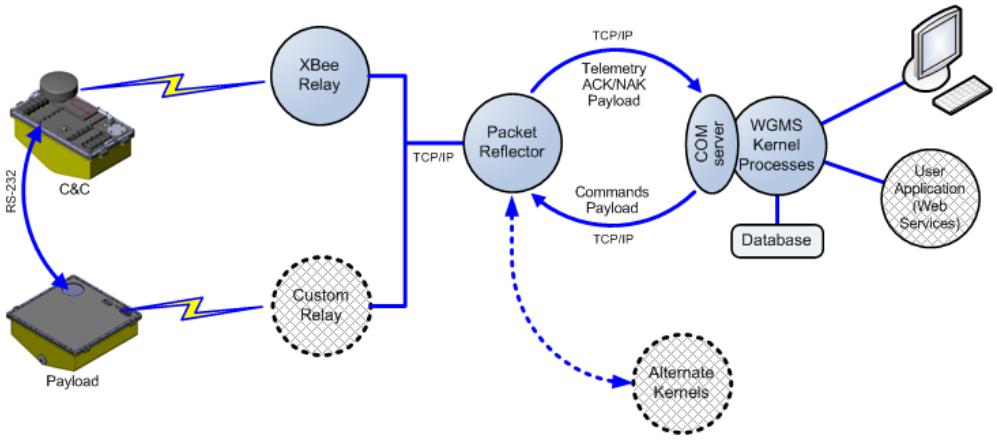


Figure 6.18. XBee Communications Flow



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 Liquid Robotics recommends that the Iridium communications also be exercised prior to launch.

Alternate WGMS Configurations

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

Customized Relay

It is possible for an organization to implement their own Relay to supplement or replace a Liquid Robotics-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 WiFi 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 dry boxes. You will also 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 to inspect payload-specific data packets in near real-time rather than waiting for them to be delivered through WGMS and Web Services.

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.

Chapter 7. Communication Methods

Iridium, Relay, and Payload communications are accomplished using packets with headers and data sub-packets.

Overview

Mobile-Originated Communications

Communications from the vehicle are categorized as "Mobile Originated" (MO). All Iridium communications originate from the vehicle. The general types of Mobile Originated Communications are:

- Position Data
- Vehicle Status
- Data forwarded from payload(s)
- Alarms
- Command ACK and NAK responses
- File read requests

Mobile-Terminated Communications

Communications from WGMS are categorized as "Mobile-Terminated" (MT). These are primarily commands to the vehicle as follows:

- Requests that a vehicle take some kind of action
- Requests for status or other data from the vehicle
- Encryption mode requests

However, when a command is issued by WGMS, 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.

Commands and Responses

Communications (to and from) the vehicle are always initiated by the vehicle when the Iridium SBD communication channel is used (this may not be the case with some other communication channels). If a command is issued from WGMS, it will not be immediately sent to the vehicle, but will instead be queued for sending when the vehicle requests that commands be sent.

When the Iridium SBD communication channel is used, the C&C periodically uploads data (for example, telemetry) to shore, and then immediately downloads the next command from the shore-side queue. The C&C has to first process a command before it can provide an ACK or NAK response. When the command has been processed, the C&C initiates an upload of the ACK/NAK to shore. If there is an additional command queued, it will be downloaded to the vehicle. This process continues until there are no more commands in the queue.

In most cases, the ACK/NAK response to any command sent from WGMS will be in the next communication from the vehicle. However, if communication is interrupted there may be multiple vehicle

data packets queued-up. When communication resumes, the shore-side will see multiple incoming data packets after a command is sent, but before the ACK/NAK for that command is received.

Any software designed for real-time processing of vehicle responses needs to consider this characteristic of communications with the C&C.

Vehicle Unique ID

The vehicle Unique 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 (for example, using relays).

Transaction ID

A Transaction ID is a 4 byte number assigned to each communication session (Command, ACK, NAK, and some types of Telemetry), between WGMS and a Wave Glider vehicle. The Transaction ID along with the Unique Vehicle ID and other metadata associated with the communication session provide a unique identifier for that transaction between WGMS and a specific vehicle. For more information see Chapter 9, *Vehicle Responses*, Chapter 10, *Vehicle Telemetry*, and Chapter 11, *Payload Communications Protocol*.

Iridium (Direct IP) and Alternate Iridium Relay Communications

Short-Burst Data

The Iridium "Short-Burst-Data" (SBD) protocol is used for Iridium (Direct IP) and Alternate Iridium Relay communications. SBD allows economical transmission of relatively small data packets. Iridium refers to messages from the modem as "mobile-originated" (MO) and messages to the modem as "mobile-terminated" (MT).

The Iridium Short-Burst-Data format supports packets of limited size, depending on the modem used and the direction of the packet:

- 9601/2/3 modems can send messages (MO) of up to 340 bytes and receive messages (MT) of up to 270 bytes.
- 9522B and 9523 modems can send messages (MO) of up to 1960 bytes and receive message (MT) of up to 1890 bytes.

The SBD modem on the vehicle is typically provisioned to deliver data to WGMS using Direct IP (over the Internet.) However, the SBD modem can be provisioned as a pair to another modem. In this case, the Alternate Iridium Relay is used to communicate with the vehicle modem's pair and acts as a communication extension of WGMS. Relay communication uses packets with headers that are similar to the Iridium packets.

Packet Levels

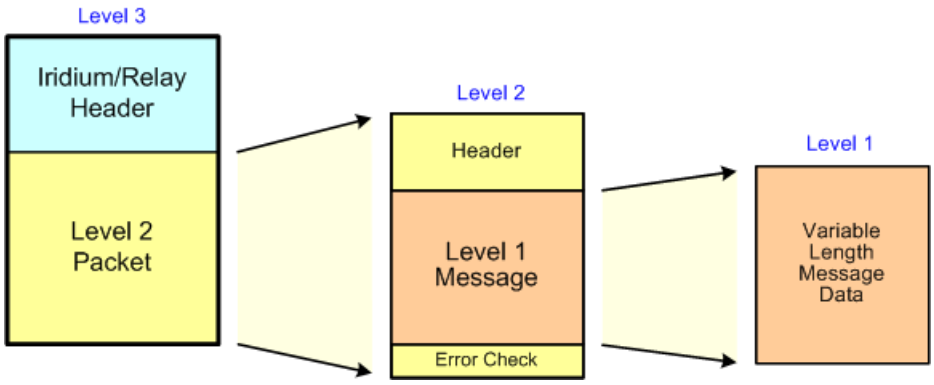
Iridium/Relay packets are described as "Level 3", "Level 2" and "Level 1".

The Level 3 packet is used for communications to and from the vehicle. The Iridium Direct IP service inserts a header to report the ID number of the vehicle's modem (IMEI) along with other metadata relevant

to the communication session. Relays also insert a header with identification numbers (Vehicle Unique ID) and metadata. These formats are described in Appendix C, *Iridium and Relay Header Data*. In each case, Level 2 packet data can follow the Level 3 header.

The Level 2 packet 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 7.1, “Level 3, Level 2, and Level 1 Data Packets”.

Figure 7.1. 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 on the data type, and the intended recipient. These packets are described in more detail in Chapter 9, *Vehicle Responses*, Chapter 10, *Vehicle Telemetry*, and Chapter 11, *Payload Communications Protocol*.

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

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.

XBee Radio Communications

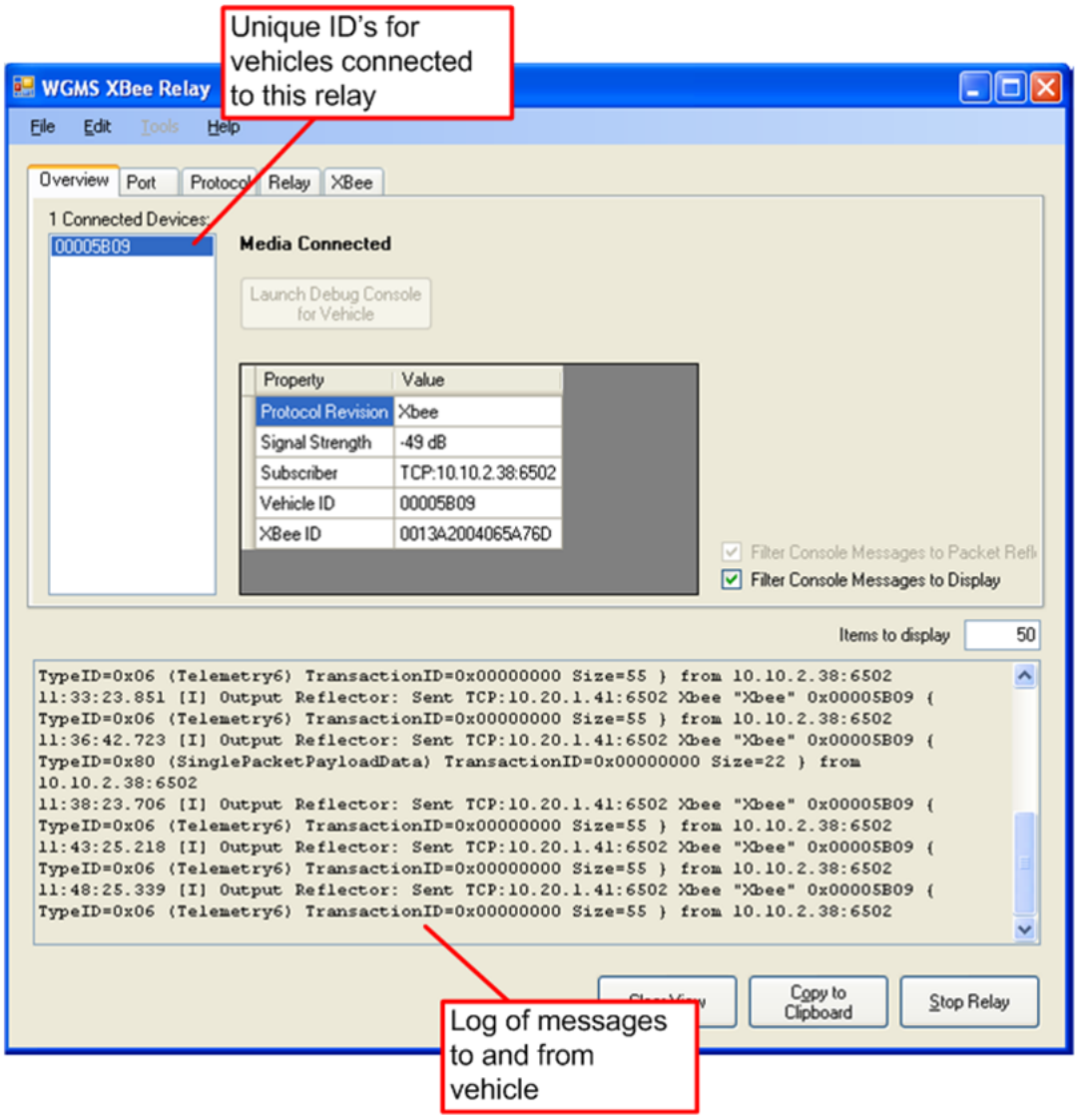
This section includes information on the XBee Relay, Debug Console and related operational considerations.

Xbee Relay

Vehicles are typically equipped with an XBee radio device (P/N 02908), which can be used as an alternate communication path between WGMS and the vehicle using the XBee Relay software (P/N 00775-01). The XBee Relay software must be installed and operated on a computer with network access to WGMS and within the communication range of the vehicle:

- Up to 300 feet indoors
- Up to 1 mile outdoors (line of sight)

As with the Alternate Iridium Relay, communications between WGMS and the XBee Relay include a Level 3 header that includes the 4-byte vehicle Unique ID to identify the vehicle (see Appendix C, *Iridium and Relay Header Data*) and it carries Level 2 messages as defined in Appendix A, *Command Descriptions*.





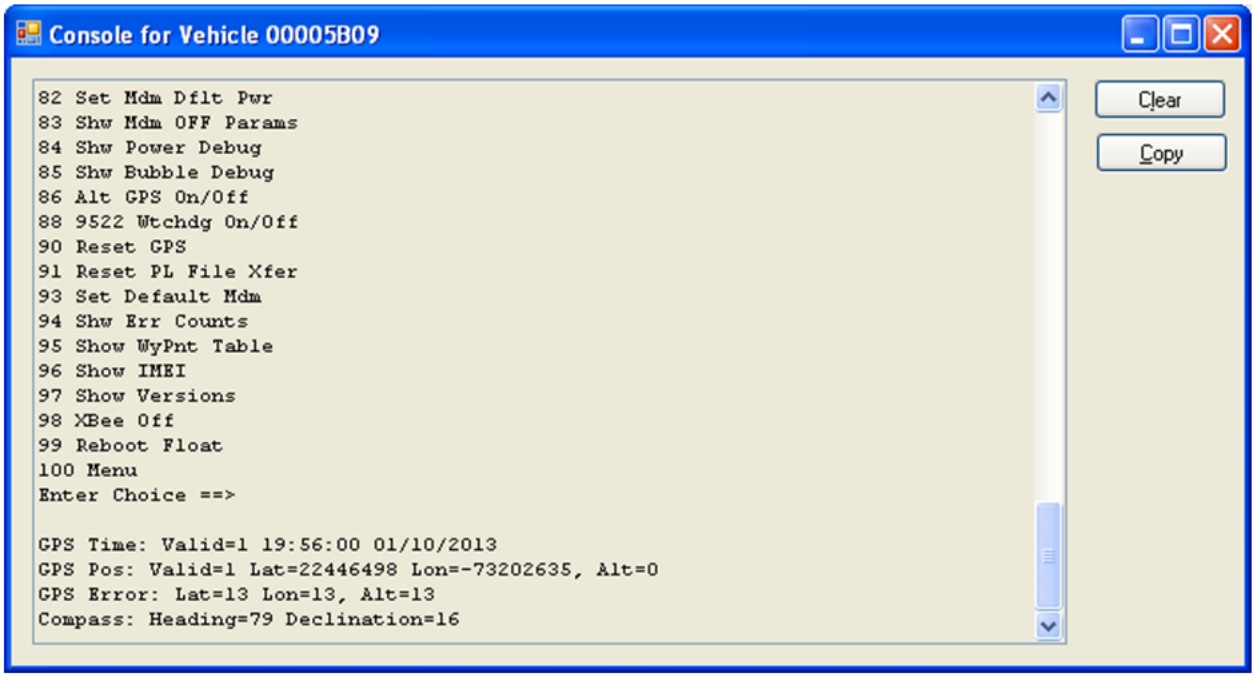
Note

The XBee modem can be powered on or off remotely using the **Set Parameter** command by configuring **Bubble Power (ID 0x20)**.

Debug Console

In addition to relaying messages between the vehicle and WGMS, the XBee Relay software implements a debug terminal interface, which interacts directly with the vehicle.>

The vehicle periodically outputs messages explaining system status at a user-specified verbosity. It also implements a list of basic commands that can be used for testing and troubleshooting. For more information, see Chapter 13, *XBee Communications*.



Operational Considerations (Channel and PAN ID)

XBee networks are subdivided using two operational parameters:

- **Default XBee Channel** (parameter 0x17, ranging from 12-23) configures the radio frequency within the XBee range that will be used to send and receive messages.

- **PAN ID** (parameter 0x16, ranging from 1 to 65534) configures a sub-network that allows multiple XBee relays to operate within the same area. Vehicles and relays must be configured with the same PAN ID in order to communicate with each other. Relays within communication range of each other must be configured with unique PAN ID's. This prevents the problem of relays not being able to establish communication with a vehicle as expected. Using unique PAN IDs also avoids unexpected loss of communications with a vehicle when another relay establishes communications.



Note

A single relay can communicate with many vehicles, but it's important not to overload a single relay with too many vehicles because that can result in unreliable communication.

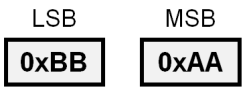
Communication Data Formats

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

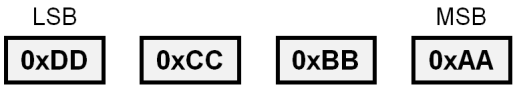
Little-Endian Byte Format

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.

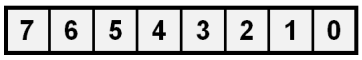
As an example of a value in 2-byte little-endian format, 0xAABB will be stored as:



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



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.



Waypoint Values

Each navigation 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 cannot 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]

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.

Distance Values

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

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.

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.

Distance Values

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

Temperature Values

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

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 (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**		

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 **Mode Names** and their associated **Values** are shown in the table below. The value is used in commands to specify the desired navigation mode. Vehicles return the navigation mode value in the telemetry as part of the command acknowledgement (ACK).

Mode Name	**Value**
NAV_MODE_UNDEFINED	0x00
NAV_COURSE_LOOP	0x01
NAV_COURSE_HOLD	0x02
NAV_CUSTOM_LOOP	0x03
NAV_CUSTOM_HOLD	0x04
NAV_CUSTOM_HEADING	0x10

NAV_RUDDER_POSITION	0x20
NAV_TARGET_WAYPOINT	0x40
NAV_HOLD_POSITION	0x41
NAV_MODE_UNDEFINED_ALT_GPS	0x80
NAV_COURSE_LOOP_ALT_GPS	0x81
NAV_COURSE_HOLD_ALT_GPS	0x82
NAV_CUSTOM_LOOP_ALT_GPS	0x83
NAV_CUSTOM_HOLD_ALT_GPS	0x84
NAV_COURSE_LOOP_PID	0x0A
NAV_COURSE_HOLD_PID	0x0B
NAV_CUSTOM_LOOP_PID	0x0C
NAV_CUSTOM_HOLD_PID	0x0D
NAV_COURSE_LOOP_PID_ALT_GPS	0x8A
NAV_COURSE_HOLD_PID_ALT_GPS	0x8B
NAV_CUSTOM_LOOP_PID_ALT_GPS	0x8C
NAV_CUSTOM_HOLD_PID_ALT_GPS	0x8D
NAV_CUSTOM_HEADING_ALT_GPS	0x90
NAV_RUDDER_POSITION_ALT_GPS	0xA0
NAV_TARGET_WAYPOINT_ALT_GPS	0xC0
NAV_HOLD_POSITION_ALT_GPS	0xC1

Roles and Permissions

Users on the shore-side system are assigned "roles", depending on 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.

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 B, *XML, Soap and Entities*.

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 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 a Liquid Robotics developer or a customer under close direction of Liquid Robotics.



Note

Admin users can assign themselves a Root role. However, this should not be done unless the Admin is in close contact with a Liquid Robotics person. Incorrect use of Root permissions can result in corruption of WGMS, loss of data, and failure of vehicle communications.

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
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

Permission Type	Guest User	Data Analyst	Master	Admin	Root	Description
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
Work flow 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)
Criteria	• • • •	• • • •	• 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
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")

Permission Type	Guest User	Data Analyst	Master	Admin	Root	Description
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 Control Status	• • • •	• 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
Payload Data Permissions						

Permission Type	Guest User	Data Analyst	Master	Admin	Root	Description
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

Chapter 8. Commands

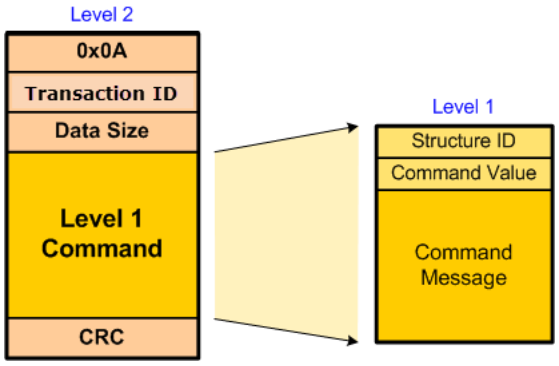
Commands are primarily used to specify some action by the vehicle, or to request information from the vehicle. Commands are typically sent using the Iridium system, although alternate methods are possible.

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.

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.

Figure 8.1. Command Packet Format



Level 2 Command Packet	
0x0A	[1 byte] Indicates a Command Data Packet.
Transaction ID	[4 bytes] A Liquid Robotics-defined ID stored in the WGMS database. For more information see the section called “Transaction ID”.
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).

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 C&C computer the type of command.
Command Message	Variable-length field containing the actual command.

Command values and structure IDs are summarized in the table below. For command data descriptions, see Appendix A, *Command Descriptions*.

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 in the Link to Command Description Appendix column.

There are additional restricted-use commands, described in the section called “Restricted Commands”.

Command Name	Command Value	Structure ID	Link to Command Description Appendix
No Command	**0x0000**	**0x10**	See the section called “No Command (0x0000)”
Set Default Comm Channel	**0x000A**	**0x10**	See the section called “Set Default Communication Channel (0x000A)”
Read Unique ID	**0x000E**	**0x10**	See the section called “Read Unique ID (0x000E)”
Set Unique ID	**0x000F**	**0x10**	See the section called “Set Unique ID (0x000F)”
Hold Current Position	**0x0001**	**0x10**	See the section called “Hold Current Position (0x0001)”
Hold Station At Waypoint	**0x0002**	**0x10**	See the section called “Hold Station at Waypoint (0x0002)”
Follow Sequential Waypoint Course	**0x0003**	**0x10**	See the section called “Follow Sequential Waypoint Course (0x0003)”
Right Rudder	**0x0004**	**0x10**	See the section called “Rudder Right/Left/ Center (0x0004, 0x0005, 0x0006)”
Left Rudder	**0x0005**	**0x10**	See the section called “Rudder Right/Left/ Center (0x0004, 0x0005, 0x0006)”
Center Rudder	**0x0006**	**0x10**	See the section called “Rudder Right/Left/ Center (0x0004, 0x0005, 0x0006)”
Follow Fixed Heading	**0x0008**	**0x10**	See the section called “Follow Fixed Heading (0x0008)”
Set Rudder Position	**0x1006**	**0x10**	See the section called “Set Rudder Position (0x1006)”
Twist Umbilical	**0x1007**	**0x16**	See the section called “Twist Umbilical (0x1007)”
Light Off Light On	**0x2001 0x2002**	**0x10**	See the section called “Visible Light Off/ On (0x2001, 0x2002)”
IR Off IR On	**0x2003 0x2004**	**0x10**	See the section called “IR Light Off/On (0x2003, 0x2004)”
XBee Off XBee On	**0x2005 0x2006**	**0x10**	See the section called “XBee Comm Off/ On (0x2005, 0x2006)”
Retrieve Status	**0x4001**	**0x10**	See the section called “Retrieve Status (0x4001)”
Set Status	**0x4002**	**0x11**	See the section called “Set Status (0x4002)”

Command Name	Command Value	Structure ID	Link to Command Description Appendix
Retrieve Identification	**0x4003**	**0x10**	See the section called “Retrieve Identification (0x4003)”
Retrieve Power Status	**0x5001**	**0x10**	See the section called “Retrieve Power Status (0x5001)”
Retrieve Rudder Control	**0x5002**	**0x10**	See the section called “Retrieve Rudder Control (0x5002)”
Set Rudder Control	**0x5003**	**0x16**	See the section called “Set Rudder Control (0x5003)”
Set Parameters	**0x6000**	**0x15**	See the section called “Set Parameters (0x6000)”
Retrieve Debug Buffer	**0x7000**	**0x16**	See the section called “Retrieve Debug Buffer (0x7000)”
Retrieve Sub Heading Buffer	**0x7001**	**0x16**	See the section called “Retrieve Sub Heading Buffer (0x7001)”
Retrieve PID Integral Cache	**0x7002**	**0x16**	See the section called “Retrieve PID Integral Cache (0x7002)”
Retrieve PID Operating Parameters	**0x7003**	**0x16**	See the section called “Retrieve PID Operating Params (0x7003)”
New Waypoint	**0x8000**	**0x10**	See the section called “New Waypoint (0x8000)”
Read Waypoint	**0x8003**	**0x10**	See the section called “Read Waypoint (0x8003)”
Follow Course and Loop	**0x8101**	**0x12**	See the section called “Follow Course and Loop (0x8101)”
Follow Course With Options	**0x8102**	**0x12**	See the section called “Follow Course With Options (0x8102)”
Add MMSI to Watch List	**0x8200**	**0x16**	See the section called “Add MMSI to Watch List (0x8200)”
Initialize File Write	**0x9000**	**0x18**	See the section called “Initialize File Write (0x9000)”
Write File Block	**0x9001**	**0x19**	See the section called “Write File Block (0x9001)”
File Delete	**0x9002**	**0x1A**	See the section called “Delete File (0x9002)”
Read File Request	**0x9003**	**0x1B**	See the section called “Read File Request (0x9003)”
Get Detailed File List (This command is implemented by the vehicle, it is not issued using WGMS)	**0x9004**	**0x1C**	See the section called “Get Detailed File List (0x9004)”
Select Program	**0x9005**	**0x10**	See the section called “Select Program (0x9005)”

Command Name	Command Value	Structure ID	Link to Command Description Appendix
Write Console Input to Vehicle (This command is implemented by the vehicle, it is not issued using WGMS)	**0x9006**	**0x1E**	See the section called “Write Console Input to Vehicle (0x9006)”
Read File Transaction Closed	**0x9007**	**0x10**	See the section called “Read File Transaction Closed (0x9007)”
Get File List	**0x9008**	**0x31**	See the section called “Get File List (0x9008)”
Write Console Script	**0x9100**	**0x25**	See the section called “Write Console Script (0x9100)”
Write Console Binary	**0x9101**	**0x25**	See the section called “Write Console Binary (0x9101)”
Float Reboot*	**0xA000**	**0x31**	See the section called “Float Reboot (0xA000)”
Reserved	**0xA51E**	**0x100**	See the section called “Reserved (0xA51E)”
Secure On	**0xA500**	**0x100**	See the section called “Secure On (0xA500)”
Secure Authenticate	**0xA501**	**0x101**	See the section called “Secure Authenticate (0xA501)”
Secure Register Client**	**0xA502**	**0x102**	See the section called “Secure Register Client (0xA502)”
Secure Off*	**0xA504**	**0x16**	See the section called “Secure Off (0xA504)”
Secure Enable Partition**	**0xA505**	**0x16**	See the section called “Secure Enable Partition (0xA505)”

Chapter 9. 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 see the section called “Restricted ACK Responses” for more information.

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:

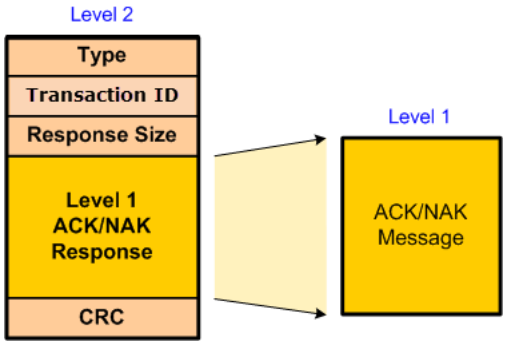
- ACK signals successful packet reception and acceptance of its content.
- NAK signals successful packet reception but rejection of its content.
- No response (timeout) signals non-reception of a packet.

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

ACK/NAK Packet Format

The ACK/NAK response packet format is shown in Figure 9.1, “ACK/NAK Response Packet Format”. 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 9.1. ACK/NAK Response Packet Format



Where:

Type: [1 byte] ACK or NAK Data Type. For more information see the section called “ACK Data Type Numbers” and the section called “NAK Data Type Numbers”.

Transaction ID: [4 bytes] The Transaction ID value for the communication session. For more information, see the section called “Transaction ID”.

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.

ACK Data Type Numbers

Allowed Data Types for the ACK return are:

Type	ACK Return Payload	Reference
0x10	Status With Waypoint Read-Back	See the section called “ACK - Status with Waypoint Read-Back (0X10)”
0x11	Status Return	See the section called “ACK - Status Return (0x11)”
0x12	Identification Data	See the section called “ACK - Vehicle Identification Data (0x12)”
0x13	No Payload	See the section called “ACK - No Payload Response (0x13)”
0x14	Power Controller Status	See the section called “ACK - Power Controller Status (0x14)”
0x15	Rudder Control Data	See the section called “ACK - Rudder Control Data (0x15)”
0x1B	Read File	See the section called “ACK - Read File Block (0x1B)”
0x1C	Detailed File List Data	See the section called “ACK - Detailed File List Data (0x1C)”
0x1E	Unique ID	See the section called “ACK – Unique ID (0x1E)”
0x1F	File at Destination	See the section called “ACK – File at Destination (0x1F)”
0x31	File List Data	See the section called “ACK – File List Data (0x31)”

NAK Data Type Numbers

Allowed data types for the NAK return are:

Type	NAK Type	Reference
0x20	(reserved)	
0x21	(reserved)	
0x22	Invalid Command	See the section called “NAK - Invalid Command (0x22)”
0x23	Invalid Data	See the section called “NAK – Invalid Data (0x22)”

Type	NAK Type	Reference
0x24	File Error	See the section called “NAK – File Error (0x24)”

Unsolicited Response Type Numbers

Allowed data Types for an unsolicited response are:

Type	Message Type	Reference
0x60	New File Available	See the section called “New File Available (0x60)”
0x1D	Write Console Output to Client	See the section called “Send Console Output to Client (0x1D)”

Relay Messages

These are messages from a Relay to the WGMS.

Type	Message Type	Reference
0x70	Relay Status	See the section called “Relay Status (0x70)”

ACK Response Data

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

ACK - Status with Waypoint Read-Back (0X10)

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

Set Single Waypoint - see the section called “Set Single Waypoint (0x8000)”

Read Waypoint - see the section called “Read Waypoint (0x8003)”

The format is identical to the regular “Status Return” - see the section called “ACK - Status Return (0x11)”, 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]

Field Name	Bytes Used	Byte / Bit	Value [range] (units)	Default	Note
(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	[See the section called “Waypoint Values”]		
Waypoint Longitude	4	31-34	[See the section called “Waypoint Values”]		

ACK - Status Return (0x11)

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

Set Status - see the section called “Set Status (0x4002)”

Retrieve Status - see the section called “Retrieve Status (0x4001)”

Hold Current Position- see the section called “Hold Current Position (0x0001)”

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 [See the section called “Waypoint Values”]		[2]
(Hold Position Longitude)	4	31	0 [See the section called “Waypoint Values”]		[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.

ACK - Vehicle Identification Data (0x12)

This response contains data on the vehicle software and hardware. It is sent in response to the command:

Retrieve Identification - see the section called “Retrieve Identification (0x4003)”

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	

Identification Packet **Field Name**	**Bytes** **Used**	**Byte** **#**	**Value** **[range] (units)**
XBee Address	2	93-94	
(reserved)	4	95-101	
Count of Program Board Records	2	102-103	**N**
Record #1	17	104-120	[See "Program Board Records" below]
Record #2	17	121-137	[See "Program Board Records" below]
. . . . .	. . .	. . .	
. . . . .	. . .	. . .	
. . . . .	. . .	. . .	
Record #N	17		[See "Program Board Records" 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)

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 - see the section called “Follow Sequential Waypoint Course (0x0003)”

Follow Fixed Heading - see the section called “Follow Fixed Heading (0x0008)”

Follow Course and Loop - see the section called “Follow Course and Loop (0x8101)”

Follow Course With Options - see the section called “Follow Course With Options (0x8102)”

Rudder Left/Right/Center - see the section called “Rudder Right/Left/Center (0x0004, 0x0005, 0x0006)”

Light, IR, XBee Off/On - see the section called “Visible Light Off/On (0x2001, 0x2002)”, the section called “IR Light Off/On (0x2003, 0x2004)”, the section called “XBee Comm Off/On (0x2005, 0x2006)”

Initialize File Write - see the section called “Initialize File Write (0x9000)”

Write File Block - see the section called “Write File Block (0x9001)”

Delete File - see the section called “Delete File (0x9002)”

Select Program - see the section called “Select Program (0x9005)”

Write Console Input to Vehicle - see the section called “Write Console Input to Vehicle (0x9006)”

Read File Transaction Closed - see the section called “Read File Transaction Closed (0x9007)”

ACK - Power Controller Status (0x14)

Sent in response to the command: **Retrieve Power Status** - see the section called “Retrieve Power Status (0x5001)”

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

Field Name	Bytes Used	Byte #	Value [Range] (units)
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	[see "Battery Status" below]
Battery_1 Status	8	30-37	[see "Battery Status" below]
Battery_2 Status	8	38-45	[see "Battery Status" below]
Battery_3 Status	8	46-53	[see "Battery Status" below]
Battery_4 Status	8	54-61	[see "Battery Status" below]
Battery_5 Status	8	62-69	[see "Battery Status" below]
Battery_6 Status	8	70-77	[see "Battery Status" 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)

ACK - Rudder Control Data (0x15)

Sent in response to the commands:

Set Rudder Control - see the section called “Set Rudder Control (0x5003)”

Retrieve Rudder Control - see the section called “Retrieve Rudder Control (0x5002)”

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 the section called “Set Rudder Control (0x5003)” 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	

ACK - Read File Block (0x1B)

Sent in response to the command:

Read File Request - see the section called “Read File Request (0x9003)”

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

ACK - Detailed File List Data (0x1C)

Sent in response to the command:

Get Detailed File List - see the section called “Get Detailed File List (0x9004)”

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 detailed file record format below]	
Detailed File Record 2	68	[see detailed file record format below]	
. . . . .	. . .	. . . . .	
. . . . .	. . .	. . . . .	
Detailed File Record N	68		

Each of the Detailed File Records has the following format:

Field Name	**Bytes**	**Field**	**Notes**
File Name	11	8+3 ASCII string. (see the section called “File Names”)	
File ID	4	WGMS-generated	

Field Name	Bytes	Field	Notes
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 (see the section called “Initialize File Write (0x9000)”).

[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.

ACK – Unique ID (0x1E)

Sent in response to the command:

Read Vehicle Unique ID - see the section called “Read Unique ID (0x000E)”

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

Field Name	Bytes	Value	Note
WGMS ID	4	see the section called “Vehicle Unique ID”	[1]

Note:

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

ACK – File at Destination (0x1F)

This is an additional response to the commands:

Get File List - see the section called “Get File List (0x9008)”

Get Detailed File List - see the section called “Get Detailed File List (0x9004)”

Select Program - see the section called “Select Program (0x9005)”

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	

ACK – File List Data (0x31)

This is the primary response to the command:

Get File List - see the section called “Get File List (0x9008)”

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 file records format below]	
File Record 2	20	[see file records format 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. (see the section called “File Names”)
File WGMS ID	4	(see the section called “Vehicle Unique ID”)
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.

NAK Response Data

NAK - Invalid Command (0x22)

The command code number was not recognized.

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

NAK – Invalid Data (0x22)

The data in the command is not correct.

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

NAK – File Error (0x24)

Response given for a file error in the commands:

Initialize File Write - see the section called “Initialize File Write (0x9000)”

Write File Block - see the section called “Write File Block (0x9001)”

Delete File - see the section called “Delete File (0x9002)”

Read File Request - see the section called “Read File Request (0x9003)”

Get Detailed File List - see the section called “Get Detailed File List (0x9004)”

Select Program - see the section called “Select Program (0x9005)”

Get File List - see the section called “Get File List (0x9008)”

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]	[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

Restricted ACK Responses

Certain ACK responses are only available to users with **Admin** and **Master** roles.

Special ACK Data Types

Allowed Data Types for the ACK return are:

Type	ACK Return Payload	Reference
0x16	Debug Buffer	See the section called “ACK – Debug Buffer (0x16)”
0x17	Sub Heading Buffer	See the section called “ACK – Sub Heading Buffer (0x17)”

Type	ACK Return Payload	Reference
0x18	PID Integral Cache	See the section called “ACK – PID Integral Cache (0x18)”
0x19	PID Operating Parameters	See the section called “ACK – PID Operating Parameters (0x19)”

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.

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.

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.

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.

Unsolicited Messages

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

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. (see the section called “File Names”)	
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	

Field Name	Bytes	Value	Note
		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 (see the section called “Addresses”). 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.

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.

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.

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]

Chapter 10. Vehicle Telemetry

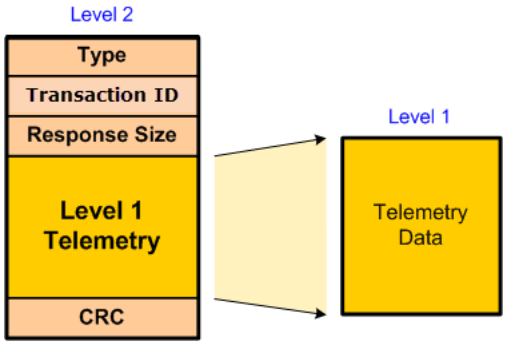
Telemetry data packets are automatically sent by the C&C at the rates defined by the Set Status command. For more information see the section called “Set Status (0x4002)”. Note that the “Standard” Telemetry rate can differ from the rate for the Retrieve Power Status command. For more information see the section called “Retrieve Power Status (0x5001)”.

Telemetry can be sent to an off-vehicle location (via Iridium/Relay or XBee), or it can be sent to a payload. For more information, see Chapter 11, *Payload Communications Protocol*.

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 10.1. Level 2 Telemetry Data Packet



Type: [1 byte] Telemetry Type (see the section called “Telemetry Types”, below)

Transaction 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).

Telemetry Types

Currently-defined Telemetry Types are:

Type	Description	References
0x03	NAL Tracker Telemetry	See the section called “NAL Tracker Telemetry 0x03”
0x05	Shout Tracker Telemetry	See the section called “Shout Tracker Telemetry 0x05”
0x06	Standard Telemetry	See the section called “Standard Telemetry Data 0x06”
0x07	Power Controller Status	See the section called “Telemetry - Power Controller Status 0x7”

Type	Description	References
0x08	DB3.5 Power Control Status	See the section called “DB3.5 Power Control Status 0x08”
0x09	Twist Status	See the section called “Twist Status 0x09”
0x70	Relay Status	See the section called “Relay Status 0x70”
0x71	System Information	See the section called “System Information 0x71”
0x80	Single Payload Packet	See the section called “Single Payload Packet 0x80”
0x81	Multi-Payload Packet	See the section called “Multi-Payload Packet 0x81”
0x91	Payload Packet	See the section called “Payload Packet 0x91”

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.

NAL Tracker Telemetry 0x03

The NAL tracker is a commercially-available device used to log time and 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]	

NAL Tracker Telemetry Field Name	Bytes Used	Byte #	Value [range]	Note
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

Shout Tracker Telemetry 0x05

Byte 0 = 0x05 (format type)

Byte 1 is a bit map of the input pins (read as 0)

Bytes 2--9 is a 64 bit unsigned integer -- convert it to the decimal value and then use DIV and MOD type functions to pull out the appropriate digits based on the dLLLMMFFFllmmfffHHHH format where:

d = latitude direction (0 North, 1 South)
l = latitude degrees
m = latitude minutes
f = latitude fraction minutes
L = longitude degrees with direction (<200 East, >200 West)
M = longitude minutes
F = longitude fraction minutes
H = horizontal dilution of precision which is set to always have 4 digits with the last 2 being decimal values

Bytes 10--17 is a 64 bit unsigned integer -- treat it as a decimal value and then use DIV and MOD type functions to pull out the appropriate digits based on the sMMDDYYYYHHmmSSTVVVV format where:

s = vertical velocity sign (1 positive, 0 negative) (Not Recorded)
MM = month of year
DD = day of month
YYYY = year
HH = hour
mm = minute of hour
SS = seconds
T = tenths of seconds (hundredths of second is always a 0 because of time alignment
VVVV = Last 4 digits in vertical velocity

Bytes 18--25 is a 64 bit unsigned integer -- treat it as a decimal value and then use DIV and MOD type functions to pull out the appropriate digits based on the sCCCCCaAAAAAgGGGGGOO format where:

s = altitude sign (1 positive, 0 negative) – Not Recorded
CCCCC = course in degrees with always 2 digits behind the decimal
a = altitude decimal place – Not Recorded
AAAAA = altitude value – Not Recorded
g = ground velocity decimal place
GGGGG = ground velocity value
OO = 6 output pins. This number can range from 00 to 63 and each bit represents one pin value. (Not Recorded)

Byte 26--27 is a 16 bit unsigned integer -- treat it as a decimal value and then use DIV and MOD type functions to pull out the appropriate digits based on the DVVVV format where:

D = The first digit is the decimal place in the vertical velocity.
VVVV = The next 4 digits are the VDOP value which is set to always have 4 digits with the last 2 being decimal values. (Not Recorded)

Byte 28 is an unsigned 8 bit value -- treat it as a decimal value and then use DIV and MOD type functions to pull out the appropriate digits based on the SSV format where:

SS = The first 2 digits are the number of GPS satellites used. (Not Recorded)
V = The last digit is the first digit in the vertical velocity. (Not Recorded)

Byte 29 is the GPS Quality Indicators and is decoded according to the following table:

bit	**Value**
0	2D-Fix (Not Recorded)
1	Dead Reckoning (Not Recorded)
2	Valid Position Fix (Not Recorded)
3	ID Included or extra data follows (Not Recorded)
4	Emergency Alert (Not Recorded)
5	Motion sensor triggered report, 1 if triggered by motion sensor. (Not Recorded)
6	Unused
7	Unused

Standard Telemetry Data 0x06

This is the standard telemetry package sent at the defined intervals. The Type 0x06 Telemetry format is:

Field Name	Byte Index	Size	Bit Index
Sub Leak Alarm	0	1	0
Float Leak Alarm	0	1	1
Sub to Float Comms Alarm	0	1	2
Payload Error Condition Alarm	0	1	3
Float Battery Low Alarm	0	1	4
Umbilical Fault Alarm	0	1	5
Sub Pressure Threshold Alarm	0	1	6
GPS Not Functioning	0	1	7
Internal vehicle Comm Error	1	1	0
Sub Reboot Alarm	1	1	1
Float to Sub Comms Error	1	1	2
Over Current Error	1	1	3
Float Temperature Threshold Exceeded	1	1	4
Float Pressure Threshold Exceeded	1	1	5
Sub Temperature Threshold Exceeded	1	1	6
Float Reboot Alarm	1	1	7
Temp_Surface	2	2	
Latitude	4	4	
Longitude	8	4	
Last_Fix_Time_Hours	12	1	
Last_Fix_Time_Minutes	13	1	
Last_Fix_Time_Seconds	14	1	
Last_Fix_Date_Month	15	1	
Last_Fix_Date_Day	16	1	
Last_Fix_Date_Year	17	1	
File Read Transaction Timeout Error	18	1	0
File Write Fallback Image Activated	18	1	1
File Write Program Image Burned	18	1	2
File Write Fallback Image Bad or Not Found	18	1	3
File Write Selected Image Bad or Not Found	18	1	4
File Write Tentative Image Retry	18	1	5
Boot Config Corrupt	18	1	6
IPIB Not Responding	18	1	7
Temperature_Sub	19	2	
Distance_Over_Ground (m)	21	2	

Field Name	Byte Index	Size	Bit Index
Rudder Count	23	4	
Heading_Sub (degrees)	27	2	
Iridium_Signal_Strength	29	1	
Target_WayPoint	30	1	
Desired_Bearing (Degrees)	31	2	
Heading_Float (Degrees)	33	2	
Pressure_Sensor_Float	35	1	
Pressure_Sensor_Sub	36	1	
Navigation_Mode	37	1	See Nav Modes
Light_State	38	1	0
IR_State	38	1	1
Xbee_State	38	1	2
Payload 1 Power	38	1	3
Payload 2 Power	38	1	4
Sub Payload Power	38	1	5
Bubble Com Error	38	1	6
Battery Com Error	38	1	7
Reset_Flags	39	1	
Total_Power (10mWh)	40	2	
Iridium_Read_Errors	42	1	
Iridium_Write_Errors	43	1	
Iridium_Session_Errors	44	1	
Water Speed	45	2	
Water Direction	47	2	
Water Speed Std Dev	49	1	
Water Direction Std Dev	50	1	
Humidity	51	1	
Payload 3 Status	52	1	0
Payload 4 Status	52	1	1
PEP Status	52	1	2
IPIB Status	52	1	3
Modem Power	52	1	4
Increased Mailbox Checks	52	1	5
Reserved	52	1	6
Reserved	52	1	7

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

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
Payload 3 Status	52	0
Payload 4 Status		1

Status Type	Byte	Bit
PEP Status		2
IPIB Status		3

Telemetry - Power Controller Status 0x7

A condensed format for the Power Controller Status is shown in the table below. The complete description is given in the section called “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

DB3.5 Power Control Status 0x08

Power Control Status packet format:

Field Name	Byte Index	Size	Bit Index	Default
Hour	0	1		
Minute	1	1		
Second	2	1		
Month	3	1		
Day	4	1		
Year	5	1		
Payload_1_Current	6	2		(mA) Signed
Payload_2_Current	8	2		(mA) Signed

Field Name	Byte Index	Size	Bit Index	Default
Umbilical_Current	10	2		(mA) Signed
Internal_Payload	12	2		(mA) Signed
Payload_3_Current	14	2		(mA) Signed
Payload_4_Current	16	2		(mA) Signed
Power_Extension_Port	18	2		(mA) Signed
Light_Bar	20	2		(mA) Signed
Total_Power_Remaining	22	4		(mWH)
Battery_Current	26	2		Signed
Payload_Voltage	28	2		(mV)
Float_Voltage	30	2		(mV)
Payload_Current	32	2		(mA)
Float_Current	34	2		(mA)
Modem_Current	36	2		(mA)
Solar_Power_1_Power	38	2		(mW)
Solar_Power_2_Power	40	2		(mW)
Auxiliary_Power	42	2		(mW)
Reserved_Power	44	2		(mW)
PM_Status	46	1		
Reserved_0	47	1		
Reserved_1	48	1		
Reserved_2	49	1		
Reserved_3	50	1		
Reserved_4	51	1		
Reserved_5	52	1		
Reserved_6	53	1		
Reserved_7	54	1		
Reserved_8	55	1		
Reserved_9	56	1		
Reserved_10	57	1		
Reserved_11	58	1		
Reserved_12	59	1		
Count of batteries in report	60	1		0 – 7 batteries >0 – count of batteries
Battery_0_Present	61	1	0	
Battery_0_Discharged	61	1	1	
Battery_0_Charged	61	1	2	
Battery_0_Discharging	61	1	3	

Field Name	Byte Index	Size	Bit Index	Default
Battery_0_Terminate_Discharge	61	1	4	
Battery_0_Terminate_Charge	61	1	5	
Battery_0_Over_Temp	61	1	6	
Battery_0_Online	61	1	7	
Battery_0_Temp	62	1		
Battery_0_Power_Remaining	63	2		
Battery_0_Current	65	2		Signed
Battery_0_Voltage	67	2		
Battery_1_Present	69	1	0	
Battery_1_Discharged	69	1	1	
Battery_1_Charged	69	1	2	
Battery_1_Discharging	69	1	3	
Battery_1_Terminate_Discharge	69	1	4	
Battery_1_Terminate_Charge	69	1	5	
Battery_1_Over_Temp	69	1	6	
Battery_1_Online	69	1	7	
Battery_1_Temp	70	1		
Battery_1_Power_Remaining	71	2		
Battery_1_Current	73	2		Signed
Battery_1_Voltage	75	2		
Battery_2_Present	77	1	0	
Battery_2_Discharged	77	1	1	
Battery_2_Charged	77	1	2	
Battery_2_Discharging	77	1	3	
Battery_2_Terminate_Discharge	77	1	4	
Battery_2_Terminate_Charge	77	1	5	
Battery_2_Over_Temp	77	1	6	
Battery_2_Online	77	1	7	
Battery_2_Temp	78	1		
Battery_2_Power_Remaining	79	2		
Battery_2_Current	81	2		Signed
Battery_2_Voltage	83	2		
Battery_3_Present	85	1	0	
Battery_3_Discharged	85	1	1	
Battery_3_Charged	85	1	2	
Battery_3_Discharging	85	1	3	
Battery_3_Terminate_Discharge	85	1	4	
Battery_3_Terminate_Charge	85	1	5	

Field Name	Byte Index	Size	Bit Index	Default
Battery_3_Over_Temp	85	1	6	
Battery_3_Online	85	1	7	
Battery_3_Temp	86	1		
Battery_3_Power_Remaining	87	2		
Battery_3_Current	89	2		Signed
Battery_3_Voltage	91	2		
Battery_4_Present	93	1	0	
Battery_4_Discharged	93	1	1	
Battery_4_Charged	93	1	2	
Battery_4_Discharging	93	1	3	
Battery_4_Terminate_Discharge	93	1	4	
Battery_4_Terminate_Charge	93	1	5	
Battery_4_Over_Temp	93	1	6	
Battery_4_Online	93	1	7	
Battery_4_Temp	94	1		
Battery_4_Power_Remaining	95	2		
Battery_4_Current	97	2		Signed
Battery_4_Voltage	99	2		
Battery_5_Present	101	1	0	
Battery_5_Discharged	101	1	1	
Battery_5_Charged	101	1	2	
Battery_5_Discharging	101	1	3	
Battery_5_Terminate_Discharge	101	1	4	
Battery_5_Terminate_Charge	101	1	5	
Battery_5_Over_Temp	101	1	6	
Battery_5_Online	101	1	7	
Battery_5_Temp	102	1		
Battery_5_Power_Remaining	103	2		
Battery_5_Current	105	2		Signed
Battery_5_Voltage	107	2		
Battery_6_Present	109	1	0	
Battery_6_Discharged	109	1	1	
Battery_6_Charged	109	1	2	
Battery_6_Discharging	109	1	3	
Battery_6_Terminate_Discharge	109	1	4	
Battery_6_Terminate_Charge	109	1	5	
Battery_6_Over_Temp	109	1	6	
Battery_6_Online	109	1	7	

Field Name	Byte Index	Size	Bit Index	Default
Battery_6_Temp	110	1		
Battery_6_Power_Remaining	111	2		
Battery_6_Current	113	2		Signed
Battery_6_Voltage	115	2		

Twist Status 0x09

Twist Status packet format:

Field Name	Byte Index	Size	Bit Index	Default
Hour	0	1		
Minute	1	1		
Second	2	1		
Month	3	1		
Day	4	1		
Year	5	1		
Motor_Activity	6	1	0, 1	
Twist_Direction	6	1	2	
Not_Used_3	6	1	3	
Twist_Activity	6	1	4	
Twist_Configuration	6	1	5	
Twist_Calibration_0	6	1	6	
Twist_Calibration_1	6	1	7	
Not_Used_8	7	1	0	
Not_Used_9	7	1	1	
Not_Used_10	7	1	2	
Not_Used_11	7	1	3	
Not_Used_12	7	1	4	
Not_Used_13	7	1	5	
Not_Used_14	7	1	6	
Twist_Debug	7	1	7	
Twist_Speed	8	2		(RPM)
Twist_Location	10	2		0-4795 = 0-360
Number_Of_Position	12	2		signed
Gearing_Ration	14	2		
Calibration_Offset	16	2		
Calibration_Threshold_A	18	2		Signed
Calibration_Threshold_B	20	2		Signed

Field Name	Byte Index	Size	Bit Index	Default
Home_Position_A	22	2		Signed
Max_Delta_A	24	2		Signed
Home_Position_B	26	2		Signed
Max_Delta_B	28	2		Signed
MaxEffect_A	30	2		Signed
Max_Delta_Home_A	32	2		Signed
MaxEffect_B	34	2		Signed
MaxDelta_B	36	2		Signed

Relay Status 0x70

Level 1 Format:

Field Name	Byte Index	Size	Bit Index	Default
Channel State	0	1	0	0=not active 1=active
Message Not Sent Alarm	0	1	1	
Communications Error Alarm	0	1	2	
Channel Not Authenticated Alarm	0	1	3	
Unused	0	1	4	0
Unused	0	1	5	0
Unused	0	1	6	0
Channel Encrypted Status	0	1	7	0=not active 1= active
Channel Revision major	1	1		2
Channel Revision Minor	2	1		0
Channel Revision Sub-Minor	3	1		0
Hour*	4	1		
Minute*	5	1		
Second*	6	1		
Month*	7	1		
Day*	8	1		
Year – 2000*	9	1		
Signal Strength**	10	1		0 – 5 bars

*Timestamp Date-Time in UTC.

** Only present in Virtual Relay (Structure Type >= 32)

Virtual Relay Channel Status Handshake

This 5 byte handshake between the Packet Reflector and the Virtual Relay in response to a RelayStatus, is sent prior to sending any pending commands in response to a connection.

Field Name	Bytes	Default Value
Relay Status Size	2	
Relay Status CRC16	2	
Disposition	1	0 – Channel Not Secure 1 – Channel is Secure

System Information 0x71

This message format is identical to that of Message type 0x91 except that the information contained in this message is a system information/configuration in nature.

Field Name	Bytes Used	Byte #	Format
System Information Type	4	0-3	
Board Id	1	4	Unsigned 8-bit integer
Task Id	1	5	Unsigned 8-bit integer
Data Format	1	6 (bits 0-3)	Unsigned 4-bit integer
Default Parser	1	6 (bit 4)	0=Binary/1=ASCII
Delivery Priority	1	6 (bits 5-7)	Unsigned 3-bit integer
Repeat Count	1	7	Unsigned 8-bit integer
System Information Record(s)	[variable]	8-	

Single Payload Packet 0x80

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 WGMS 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	Reference
Weather Station	1	the section called “Weather Station Packet (Type 1)”
Payload Status	2	the section called “Payload Status Packet (Type 2)”
Payload AISyG Status	16	the section called “Payload AISyG Status Packet (Type 16)”

Packet Type	Value	Reference
Payload AISyG Static Record	17	the section called “Payload AISyG Static Record (Type 17)”
Payload AISyG Position Report	18	the section called “Payload AISyG Position Report (Type 18)”

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

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)

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)

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	

Field Name	Bytes Used	Byte #	Default
Speed Over Ground	2	80	
Rate Of Turn	1	82	
Heading	2	83	
Navigation Status	1	85	

Payload AISyG Position Report (Type 18)

A Payload AISyG Static Record packet has a variable number of bytes depending on 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	(see record format below)
Record 2	26	27	(see record format below)
. . .	. . .	. . .	
. . .	. . .	. . .	
Record N	26		(see record format 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	

1.1.1 Sensor Sample Record (Unparsed)

All sensors must locate each sample in space and time using the following format. The software managing the sensor is responsible for providing the timestamp and the lat/lon. The SMC and PIBs should provide callable routines that report these values in a form that can be directly inserted into sensor sample records. Message types 0x90 and 0x91 provide for repeat counts for multiple samples within a message.

Field Name	Bytes Used	Byte #	Format
Latitude	4	0-3	Signed 32-bit integer
Longitude	4	4-7	Signed 32-bit integer
Unix Timestamp	4	8-11	Signed 32-bit integer
Sensor Data	[variable]	12-	

Fluorometer (C3) Data Type 96

Data Format 0:

Field Name	Bytes Used	Byte #	Format
Latitude	4	0-3	Signed 32-bit integer
Longitude	4	4-7	Signed 32-bit integer
Unix Timestamp	4	8-11	Signed 32-bit integer
DataC1 Int	2	12-13	Unsigned 32-bit integer
DataC1 Fraction	1	14	Unsigned 32-bit integer
DataC2 Int	2	15-16	Unsigned 32-bit integer
DataC2 Fraction	1	17	Unsigned 32-bit integer
DataC3 Int	2	18-19	Unsigned 32-bit integer
DataC3 Fraction	1	20	Unsigned 32-bit integer
Pressure Int	2	21-22	Unsigned 32-bit integer
Pressure Fraction	1	23	Unsigned 32-bit integer
Temperature	2	24-25	Unsigned 32-bit integer

Compressed CTD Data Type 112

The GPCTD Data comes back as payload type 0x00000070 or 112 decimal. The format does use the sub-type field for a couple of different possible data formats. Currently we are only supporting sub-type 8 which is the CTD Data with no O2 data included (details are in section 9 in the document). At some point in the future we will also have units that use sub-type 9, which will have an extra 4 bytes in every GPCTD sample for the oxygen.

Compressed CTD Data

This record reports the Pressure, Temperature, and Conductivity for each sample.

Field Name	Bytes Used	Byte #	Format
Latitude	4	0-3	Signed 32-bit integer
Longitude	4	4-7	Signed 32-bit integer

Field Name	Bytes Used	Byte #	Format
Unix Timestamp	4	8-11	Signed 32-bit integer
Pressure	4	12-15	Unsigned 32-bit integer
Temperature	4	16-19	Unsigned 32-bit integer
Conductivity	4	20-23	Unsigned 32-bit integer

Latitude and Longitude – the lat/lon reported by the C&C to the PIB – the location of the float at the time that the PIB obtained it from the C&C. Note that the reading of the GPS is not directly coordinated with the CTD, but should be close enough given other errors in the system.

Unix Timestamp – the standard Unix timestamp computed from the PIB RTC timestamp. The PIB clock is set from the C&C GPS and should be accurate within a second.

Pressure – The Pressure as measured by the CTD. P(in decibars) = (Value/100) - 10

Temperature – The Temperature as measured by the CTD. T(in deg C) = (Value/10,000) – 5

Conductivity – The Conductivity as measured by the CTD. C(in S/m) = (Value/100,000) – 0.05

Compressed CTD With Oxygen Sensor Data

This record reports the Pressure, Temperature, and Conductivity for each sample.

Field Name	Bytes Used	Byte #	Format
Latitude	4	0-3	Signed 32-bit integer
Longitude	4	4-7	Signed 32-bit integer
Unix Timestamp	4	8-11	Signed 32-bit integer
Pressure	4	12-15	Unsigned 32-bit integer
Temperature	4	16-19	Unsigned 32-bit integer
Conductivity	4	20-23	Unsigned 32-bit integer
Oxygen	4	24-27	Unsigned 32-bit integer

Latitude and Longitude – the lat/lon reported by the C&C to the PIB – the location of the float at the time that the PIB obtained it from the C&C. Note that the reading of the GPS is not directly coordinated with the CTD, but should be close enough given other errors in the system.

Unix Timestamp – the standard Unix timestamp computed from the PIB RTC timestamp. The PIB clock is set from the C&C GPS and should be accurate within a second.

Pressure – The Pressure as measured by the CTD. P(in decibars) = (Value/100) - 10

Temperature – The Temperature as measured by the CTD. T(in deg C) = (Value/10,000) – 5

Conductivity – The Conductivity as measured by the CTD. C(in S/m) = (Value/100,000) – 0.05

Oxygen – The Dissolved Oxygen as measured by the optional Oxygen sensor. O(in Hz) = (Value/10)

Payload ECO Puck Sensor Data Type 123

ECO Puck telemetry is returned as a type 0x91 structure as follows. See WET Labs documentation for definitions of the sensor specific values. Where appropriate, the text strings are converted to binary to save space.

FORMAT CODE 0 - ECO PUCK SAMPLE

Field Name	Bytes Used	Byte #	Format
PayloadType	4	0-3	123 (WET Labs ECO Puck)
Board Id	1	4	Unsigned 8-bit integer
Task Id	1	5	Unsigned 8-bit integer
Sensor Data Format	1	6 (bits 0-3)	Unsigned 4-bit integer – 0
Default Parser	1	6 (bit 4)	0=Binary/1=ASCII – 0
Delivery Priority	1	6 (bits 5-7)	Unsigned 3-bit integer – 0
Repeat Count	1	7	Unsigned 8-bit integer

Individual samples data format 0

Latitude	**4**	**0-3**	**Signed 32-bit integer**
Longitude	4	4-7	Signed 32-bit integer
Time	4	8-11	Unix time_t as read on the SMC
Raw date from sensor	8	12-19	String returned by sensor
Raw time from sensor	8	20-27	String returned by sensor
Reference frequency 1	2	28-29	Unsigned 16-bit integer
Reading 1	2	30-31	Unsigned 16-bit integer
Reference frequency 2	2	32-33	Unsigned 16-bit integer
Reading 2	2	34-35	Unsigned 16-bit integer
Reference frequency 3	2	36-37	Unsigned 16-bit integer
Reading 3	2	38-39	Unsigned 16-bit integer
Therm	2	40-41	Unsigned 16-bit integer

Example of a raw sample with 13 data columns of the EcoPuck:

99/99/99 99:99:99 532 4130 650 4130 460 4130 525 2011/06/28 7:05:45.430877 37.410927833333325 -122.00422849999999

How the first 9 columns of the payload should be interpreted:

	Column 1	Column 2	Column 3	Column 4	Column 5	Column 6	Column 7	Column 8	Column 9
Unit	Raw Date	Raw Time	Sample 1	532nm	Sample2	650nm	Sample3	CDOM	therm
Raw samples	99/99/99	99:99:99	532	4130	650	4130	460	4130	525
Type	8char String	8char String	INT16	INT16	INT16	INT16	INT16	INT16	INT16

Payload CStar Sensor Data Type 124

The telemetry is returned as a type 0x91 structure. See WET Labs documentation for the definition of the sensor specific values. Where appropriate, the text strings are converted to binary to save space.

FORMAT CODE 0 – C-STAR SENSOR TELEMETRY

Field Name	Bytes Used	Byte #	Format
PayloadType	4	0-3	124 (WET Labs ECO Puck)
Board Id	1	4	Unsigned 8-bit integer
Task Id	1	5	Unsigned 8-bit integer
Sensor Data Format	1	6 (bits 0-3)	Unsigned 4-bit integer – 0
Default Parser	1	6 (bit 4)	0=Binary/1=ASCII – 0
Delivery Priority	1	6 (bits 5-7)	Unsigned 3-bit integer – 0
Repeat Count	1	7	Unsigned 8-bit integer

Individual samples

Latitude	**4**	**0-3**	**Signed 32 bit integer**
Longitude	4	4-7	Signed 32-bit integer
Unix Timestamp	4	8-11	Unix time_t as read on the SMC
Sensor serial number	10	12-21	String returned by sensor
Reference count	2	22-23	Unsigned 16-bit integer
Signal count	2	24-25	Unsigned 16-bit integer
Corr Signal Raw count	2	26-27	Unsigned 16-bit integer
Beam c Inverse	8	28-35	Double precision floating point
Temperature	2	36-37	Unsigned 16-bit integer

Example of a raw sample with 9 data columns of the CStar:

1234PG 11829 13838 13695 0.003 527 17:05:45.430877 37.410927833333325 -122.00422849999999

How the first 6 columns of the payload should be interpreted:

	Column 1	Column 2	Column 3	Column 4	Column 5	Column 6
Unit	Instrument serial number	Reference counts	Signal counts	Corrected Signal raw counts	Calculated Beam c inverse meters	Internal thermistor, counts
Raw samples	CST-1234PG	11829	13838	13695	0.003	527

Payload Benthos Modem Data Type 125

Format code 0 – Range telemetry results (ATY/ATR)

Field Name	Bytes Used	Byte #	Format
PayloadType	4	0-3	125 (Benthos Modem)
Board Id	1	4	Unsigned 8-bit integer
Task Id	1	5	Unsigned 8-bit integer
Sensor Data Format	1	6 (bits 0-3)	Unsigned 4-bit integer – 0
Default Parser	1	6 (bit 4)	0=Binary/1=ASCII – 0
Delivery Priority	1	6 (bits 5-7)	Unsigned 3-bit integer – 0
Repeat Count	1	7	Unsigned 8-bit integer – 1
Latitude	4	8-11	Latitude
Longitude	4	12-15	Longitude
Time	4	16-19	Unix time_t
Distance to Target	4	20-23	Unsigned 32-bit integer, tenths of meters

Records of the following format representing results for different baud rates, 6 sets (150 bytes):

Baud Rate	2	0-1	Signed 16-bit integer
Acquisition - SNR	2	2-3	Signed 16-bit integer
Acquisition - MPD	2	4-5	Signed 16-bit integer
Acquisition - SPD	2	6-7	Signed 16-bit integer
Header - CRC	1	8	Signed 8-bit integer (0=fail, 1=pass)
Header - SNR	2	9-10	Signed 16-bit integer
Header – CCERR	2	11-12	Signed 16-bit integer (-1 if not reported)
Data - MOD	2	13-14	Signed 16-bit integer
Data - ERR	2	15-16	Signed 16-bit integer
Data - SNR	2	17-18	Signed 16-bit integer
Data - AGC	2	19-20	Signed 16-bit integer
Data - SPD	2	21-22	Signed 16-bit integer
Data - CCERR	2	23-24	Signed 16-bit integer

Payload Datawell MOSE Sensor Data Type 126

Format code 0 – Reporting Hs/Tav/Tp/Dirp wave spectrum data.

Field Name	Bytes Used	Byte #	Format
PayloadType	4	0-3	126 (Datawell MOSE)

Field Name	Bytes Used	Byte #	Format
Board Id	1	4	Unsigned 8-bit integer
Task Id	1	5	Unsigned 8-bit integer
Sensor Data Format	1	6 (bits 0-3)	Unsigned 4-bit integer – 0
Default Parser	1	6 (bit 4)	0=Binary/1=ASCII – 0
Delivery Priority	1	6 (bits 5-7)	Unsigned 3-bit integer – 0
Repeat Count	1	7	Unsigned 8-bit integer

Records the following format:

Latitude	4	8-11	Latitude
Longitude	4	12-15	Longitude
Time	4	16-19	Unix time_t
Hs	8	20-27	double precision float
Tav	8	28-35	double precision float
Tp	8	36-43	double precision float
Dirp	8	44-51	double precision float
Number of Averaged Spectra	1	52	Signed int
Number of Samples per Spectrum	2	53-54	Signed int
Number of Sample Gaps	2	55-56	Signed int

Payload VEMCO VR2C Data Type 127

Format code 0 – Reporting a single tag reading.

Field Name	Bytes Used	Byte #	Format
PayloadType	4	0-3	127 (VEMCO VR2C)
Board Id	1	4	Unsigned 8-bit integer
Task Id	1	5	Unsigned 8-bit integer
Sensor Data Format	1	6 (bits 0-3)	Unsigned 4-bit integer – 0
Default Parser	1	6 (bit 4)	0=Binary/1=ASCII – 1
Delivery Priority	1	6 (bits 5-7)	Unsigned 3-bit integer – 0
Repeat Count	1	7	Unsigned 8-bit integer
Record Format:			
Latitude	4	8-11	Latitude
Longitude	4	12-15	Longitude
Time	4	16-19	Unix time_t

Field Name	Bytes Used	Byte #	Format
Tag Reading String	Varies	20-	string, variable length

Payload ADCP Data Type 128

Bathymetry Data:

FormatField Name	Bytes Used	Byte #	Format
Latitude	4	0-3	Signed 32-bit integer
Longitude	4	4-7	Signed 32-bit integer
Unix Timestamp	4	8-11	Signed 32-bit integer
Heading	2	12-13	Unsigned 16-bit integer
Pitch	2	14-15	Signed 16-bit integer
Roll	2	16-17	Signed 16-bit integer
B1	2	18-19	Unsigned 16-bit integer
B2	2	20-21	Unsigned 16-bit integer
B3	2	22-23	Unsigned 16-bit integer
B4	2	24-25	Unsigned 16-bit integer

C2 Data:

Field Name	Bytes Used	Byte #	Format
Latitude	4	0-3	Signed 32-bit integer
Longitude	4	4-7	Signed 32-bit integer
Unix Timestamp	4	8-11	Signed 32-bit integer
Bin size	2	12-13	Unsigned 16-bit integer
Blanking distance	2	14-15	Unsigned 16-bit integer
Bin 1, East	2	16-17	Signed 16-bit integer
Bin 1, North	2	18-19	Signed 16-bit integer
Bin 2, East	2	20-21	Signed 16-bit integer
Bin 2, North	2	22-23	Signed 16-bit integer
…	2	24-25	Signed 16-bit integer

C4 Data:

Field Name	Bytes Used	Byte #	Format
Latitude	4	0-3	Signed 32-bit integer
Longitude	4	4-7	Signed 32-bit integer
Unix Timestamp	4	8-11	Signed 32-bit integer
Bin size	2	12-13	Unsigned 16-bit integer
Blanking distance	2	14-15	Unsigned 16-bit integer
Bin 1, East	2	16-17	Signed 16-bit integer
Bin 1, North	2	18-19	Signed 16-bit integer

Field Name	Bytes Used	Byte #	Format
Bin 1, Up	2	20-21	Signed 16-bit integer
Bin 1, Error	2	22-23	Signed 16-bit integer
…	2	24-25	Signed 16-bit integer

Payload Aquatracka Data Type 130

Format code 0 – Reporting a single tag reading.

Field Name	Bytes Used	Byte #	Format
PayloadType	4	0-3	130 (Aquatracka)
Board Id	1	4	Unsigned 8-bit integer
Task Id	1	5	Unsigned 8-bit integer
Sensor Data Format	1	6 (bits 0-3)	Unsigned 4-bit integer – 0
Default Parser	1	6 (bit 4)	0=Binary/1=ASCII – 1
Delivery Priority	1	6 (bits 5-7)	Unsigned 3-bit integer – 0
Repeat Count	1	7	Unsigned 8-bit integer
Record Format:			
Latitude	4	8-11	Latitude
Longitude	4	12-15	Longitude
Time	4	16-19	Unix time_t
Raw Temp. Mean	4	20-23	float
Raw Ref. Mean	4	24-27	float
Raw Sig. Mean	4	28-31	float
Conc. Mean	4	32-35	float
Temp. Mean	4	36-39	float
Raw Temp. Stddev	4	40-63	float
Raw Ref. Stddev	4	44-47	float
Raw Sig. Stddev	4	48-53	float
Conc. Stddev	4	54-57	float
Temp. Stddev	4	58-61	float

Payload WHOI Micro Modem Type 131

Format code 0 – Informational Report.

Field Name	Bytes Used	Byte #	Format
PayloadType	4	0-3	131 (WHOI Micro Modem)
Board Id	1	4	Unsigned 8-bit integer

Field Name	Bytes Used	Byte #	Format
Task Id	1	5	Unsigned 8-bit integer
Sensor Data Format	1	6 (bits 0-3)	Unsigned 4-bit integer – 0
Default Parser	1	6 (bit 4)	0=Binary/1=ASCII – 1
Delivery Priority	1	6 (bits 5-7)	Unsigned 3-bit integer – 0
Repeat Count	1	7	Unsigned 8-bit integer
Latitude	4	8-11	Latitude
Longitude	4	12-15	Longitude
Time	4	16-19	Unix time_t
Report string	Varies	20-	string, variable length

Harmony Payload Type 132

Format Code 0 - String Report

Field Name	Bytes Used	Byte #	Format
PayloadType	4	0-3	132 (Harmony)
Board Id	1	4	Unsigned 8-bit integer
Task Id	1	5	Unsigned 8-bit integer
Sensor Data Format	1	6 (bits 0-3)	Unsigned 4-bit integer – 0
Default Parser	1	6 (bit 4)	0=Binary/1=ASCII – 1
Delivery Priority	1	6 (bits 5-7)	Unsigned 3-bit integer – 0
Repeat Count	1	7	Unsigned 8-bit integer
Latitude	4	8-11	Latitude
Longitude	4	12-15	Longitude
Time	4	16-19	Unix time_t
Report string	Varies	20-	string, variable length

Format Code 1 - Binary Report

Field Name	Bytes Used	Byte #	Format
PayloadType	4	0-3	132 (Harmony)
Board Id	1	4	Unsigned 8-bit integer
Task Id	1	5	Unsigned 8-bit integer
Sensor Data Format	1	6 (bits 0-3)	Unsigned 4-bit integer – 1
Default Parser	1	6 (bit 4)	0=Binary/1=ASCII – 0
Delivery Priority	1	6 (bits 5-7)	Unsigned 3-bit integer – 0

Field Name	Bytes Used	Byte #	Format
Repeat Count	1	7	Unsigned 8-bit integer
Latitude	4	8-11	Latitude
Longitude	4	12-15	Longitude
Time	4	16-19	Unix time_t
Report data	Varies	20-	Bytes, variable length

Cefas OrbComm Payload Type 134

Data Format 0:

Field Name	Bytes Used	Byte #	Format
PayloadType	4	0-3	133 (OrbComm)
Board Id	1	4	Unsigned 8-bit integer
Task Id	1	5	Unsigned 8-bit integer
Sensor Data Format	1	6 (bits 0-3)	Unsigned 4-bit integer – 0
Default Parser	1	6 (bit 4)	0=Binary/1=ASCII – 1
Delivery Priority	1	6 (bits 5-7)	Unsigned 3-bit integer – 0
Repeat Count	1	7	Unsigned 8-bit integer
Record Format:			
Latitude	4	8-11	Latitude
Longitude	4	12-15	Longitude
Time	4	16-19	Unix time_t
Data	270	20-varies	String

LISST Payload Type 135

Format Code 0:

Field Name	Bytes Used	Byte #	Format
PayloadType	4	0-3	135 (LISST)
Board Id	1	4	Unsigned 8-bit integer
Task Id	1	5	Unsigned 8-bit integer
Sensor Data Format	1	6 (bits 0-3)	Unsigned 4-bit integer – 0
Default Parser	1	6 (bit 4)	0=Binary/1=ASCII – 1
Delivery Priority	1	6 (bits 5-7)	Unsigned 3-bit integer – 0
Repeat Count	1	7	Unsigned 8-bit integer
Record Format:			
Latitude	4	8-11	Latitude
Longitude	4	12-15	Longitude
Time	4	16-19	Unix time_t
Total Concentration	4	20-23	float

Field Name	Bytes Used	Byte #	Format
Mean Particle Size	4	24-27	float
Sample Count	4	28-31	Unsigned 32-bit integer
Serial Number	2	32-33	Unsigned 16-bit integer
Name Length	1	34	byte
File Name	Varies	35	ASCII String

Payload CO2 Sensor Type 144

Field Name	Bytes Used	Byte #	Format
PayloadType	4	0-3	144 (pCO2)
Board Id	1	4	Unsigned 8-bit integer
Task Id	1	5	Unsigned 8-bit integer
Sensor Data Format	1	6 (bits 0-3)	Unsigned 4-bit integer – 0
Default Parser	1	6 (bit 4)	0=Binary/1=ASCII – 1
Delivery Priority	1	6 (bits 5-7)	Unsigned 3-bit integer – 0
Repeat Count	1	7	Unsigned 8-bit integer
Record Format:			
Latitude	4	8-11	Latitude
Longitude	4	12-15	Longitude
Time	4	16-19	Unix time_t
Custom Format	Varies	20-	Customer Specified

Payload Thermistor Array Data Type 192

Format code 0:

Field Name	Bytes Used	Byte #	Format
PayloadType	4	0-3	131 (WHOI Micro Modem)
Board Id	1	4	Unsigned 8-bit integer
Task Id	1	5	Unsigned 8-bit integer
Sensor Data Format	1	6 (bits 0-3)	Unsigned 4-bit integer – 0
Default Parser	1	6 (bit 4)	0=Binary/1=ASCII – 1
Delivery Priority	1	6 (bits 5-7)	Unsigned 3-bit integer – 0
Repeat Count	1	7	Unsigned 8-bit integer
Latitude	4	8-11	Latitude
Longitude	4	12-15	Longitude
Time	4	16-19	Unix time_t
T1	2	20-21	
T2	2	22-23	

Field Name	Bytes Used	Byte #	Format
T3	2	24-25	
T4	2	26-27	
T5	2	28-29	
T6	2	30-31	

Multi-Payload Packet 0x81

Sequence of Payload packets sent from ship to shore using glider core Iridium modem.

Field Name	Byte Index	Size	Bit Index	Default
PayloadType	0	4		
PacketUId	4	4		
PacketCount	8	2		
PacketTotal	10	2		
PacketData	12	Variable		

NOTE: Packet UId is a vehicle unique identifier that is assigned to each multi-packet sequence in order to uniquely identify all packets that belong to that sequence.

Payload Packet 0x91

(Telemetry Type 145) This message format is identical to that of Message type 0x90 except that the whole structure has a 4 byte PayloadType prepended. The PayloadType is identical in format and usage to the field of the same name and byte position in the deprecated Telemetry Type 0x80 message. The PacketData field of 0x80 would correspond to everything in 0x91 after the PayloadType field.

Field Name	Bytes Used	Byte #	Format
PayloadType	4	0-3	See Telemetry type 0x80
Board Id	1	4	Unsigned 8-bit integer
Task Id	1	5	Unsigned 8-bit integer
Sensor Data Format	1	6 (bits 0-3)	Unsigned 4-bit integer
Default Parser	1	6 (bit 4)	0=Binary/1=ASCII
Delivery Priority	1	6 (bits 5-7)	Unsigned 3-bit integer
Repeat Count	1	7	Unsigned 8-bit integer
Sensor Sample(s)	[variable]	8-	

Chapter 11. Payload Communications Protocol

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 WGMS. Liquid Robotics has developed a common generic communications protocol described below. Using this protocol allows Liquid Robotics and its customers to work independently on a project, yet retain communication interoperability.



Note

The Payload Communications Protocol carries vehicle commands and responses (Level 2 messages). For more information, see Appendix A, *Command Descriptions*.

Communication Electronics

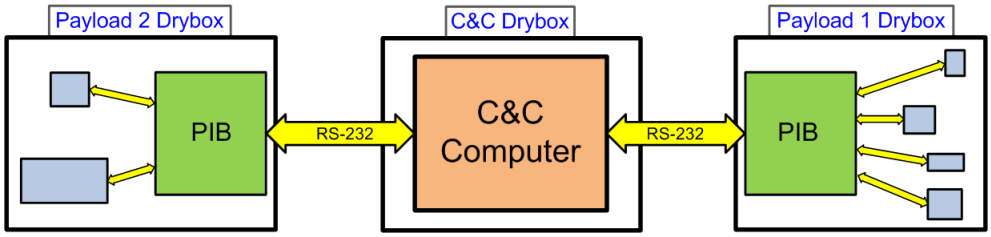
Payload communication electronics can be both Liquid Robotics-supplied and customer-supplied (or customer-developed) payload boards and devices. For more information on Liquid Robotics-supplied electronics, see Chapter 12, *Payload Communications Hardware*.

Payload Interface Board (PIB)

The Payload Interface Board (PIB) is a standard Liquid Robotics-developed circuit board designed to be mounted within a payload dry box. The PIB is a basic component of payload communications.

The Float C&C computer is connected to the payloads using serial ports. Within each payload dry box, each payload is associated with a PIB that manages communications between the payload and the C&C. PIBs can also connect to other Liquid Robotics, or customer-designed, payload boards and devices.

Figure 11.1. Basic Payload Communications

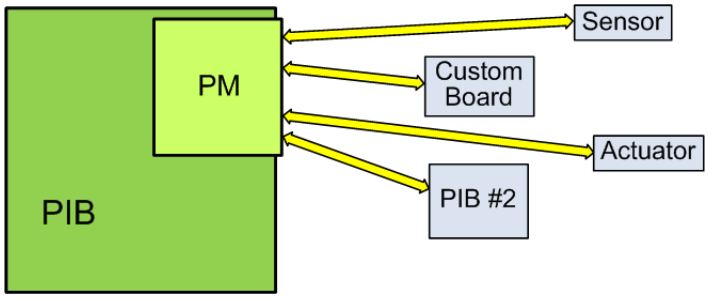


PIB boards do not connect directly with payload boards and devices. Intermediate boards known as "Personality Modules" provide the electrical interface between PIBs and other payload boards and devices.

Personality Modules

Each PIB board is designed to have to up to three Personality Module (PM) "daughter boards" mounted to it. The PMs can connect to different payload boards and devices, including other PIBs.

Figure 11.2. Personality Module Connections



Communications between a PM and the other payload boards and devices is typically accomplished using RS-232, but it is not necessarily restricted to that communication method.

Liquid Robotics provides a standard PM called the Power/Comm (P/COM) board. The P/COM provides both power and communications (RS-232, RS-422 or RS-485) to connected boards or devices.

Customers have the option to create custom PMs. Hardware descriptions of the PIB and P/COM are provided in Chapter 12, *Payload Communications Hardware*.

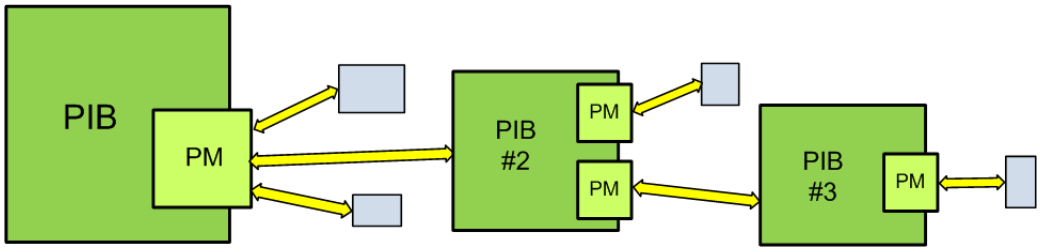
Payload Board Devices

Application-specific payload board devices can take many forms. Some possible types of payload board devices are:

- Standard Liquid Robotics-supplied PIBs
- Sensors
- Actuators
- Storage devices (for example, Flash memory)
- External communications devices
- Processing nodes

Since a PM can be connected to additional PIBs, it is possible to daisy-chain PIBs.

Figure 11.3. Daisy-Chain of PIBs



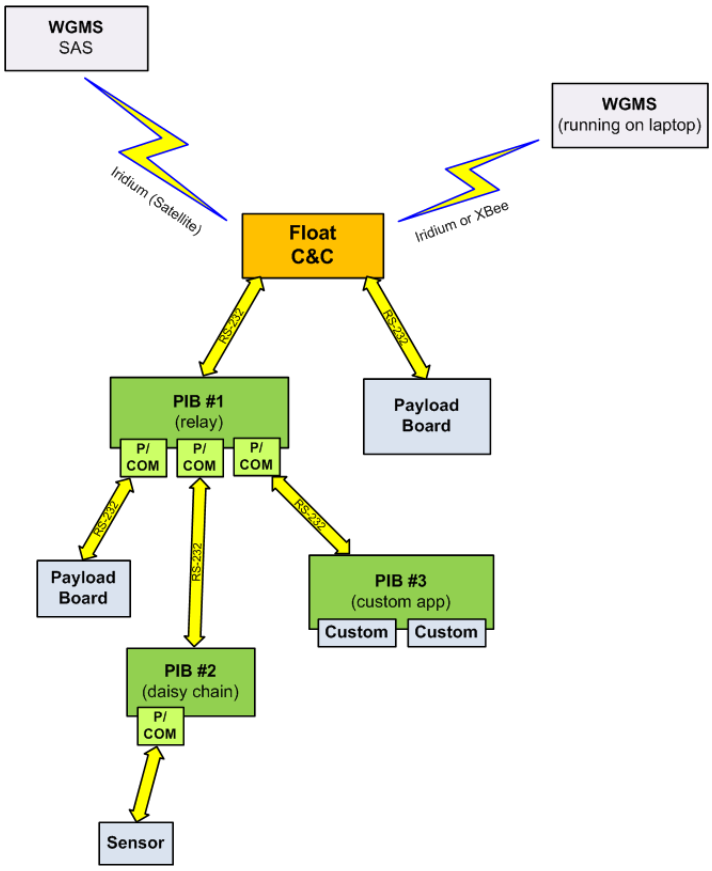
Configuration Scenarios

Some possible payload configuration scenarios are:

- Payload boards and sensors connected to PIBs using 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 acting as an end device, with custom PMs attached (the custom PMs are taking the place of payload boards.)
- Payload board connected directly to the Float C&C (taking the place of a PIB)

Figure 11.4, “Example Payload Hardware Configuration” shows a combination of the above-described scenarios. Note that the Float C&C is typically connected by radio links to WGMS (the Iridium and XBee links are considered part of the configuration).

Figure 11.4. Example Payload Hardware Configuration



This configuration shows most communications below the Float C&C using RS-232. Although this is recommended, it is not necessarily required if you are using a custom PM or a direct electrical connection to a P/COM.

Communication Method

Payload messaging relies on serial communications to transmit data packets, using polling to initiate communications and queues to accommodate messages between polling intervals.

Payload Communications Requirements

The communications protocol must allow the C&C, a PIB, or other device to:

- Identify itself and its software version number.
- Be updated with its current location according to GPS.
- Subscribe to various messages that are sent to WGMS.
- Issue commands directly to the Float C&C.
- Be updated with a new program, if the application running on the payload 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.

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 on the kind of PM used. However, Liquid Robotics recommends using the values listed above.

Payload Data Packet Format

Payload data packets contain header data, followed by the message and 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 11.5. Payload Data Packet Format



where:

- **SOF Character:** Indicates the start of a payload message packet.
- **Packet Size:** Is the number of bytes in the packet, excluding the SOF character.
- **Destination Address:** Is where the packet is to be sent.
- **Originator Address:** Is who sent the packet.
- **Tracking ID:** Is the number created by the originator for message tracking.
- **Message Type:** Is a code value indicating the type/format of data in the message.
- **Message Data:** The actual message.
- **Error Check:** A CRC code value.

Message Routing

A message packet contains the address of the originator and the address of the recipient. When the C&C or a PIB creates a message packet, it can choose to broadcast it to all connected child devices (PIBs, payload boards or applications), or to send it to a specific device. Depending on the message destination ID and the PIB address, the PIB can process the message or route the message 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 the packet 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.

Polling

Certain types of messages have a polling function in addition to any data they may contain.

The C&C sends out periodic polling messages to the PIBs connected to it. The PIBs can send a "message waiting" response to the C&C. If a PIB has a message waiting, the C&C requests the PIB to send it and

the PIB sends the message to the C&C. When the C&C receives the message, it 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.

Similarly, a PIB has the option to send out periodic polling messages to its child devices. Each child device can send a "message waiting" response to the parent PIB. The parent PIB then requests and receives the message, processes the message, and routes it to another child device (or puts it into its message queue and waits to be polled).

The polling rates are set by each PIB as part of the Enumeration Response.

Enumeration Request and 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.

ACKs and NAKs

When the C&C or a PIB receives a message packet, it typically responds to the sender with an ACK ("Acknowledge") or NAK ("Not Acknowledge") message packet.

An ACK 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.

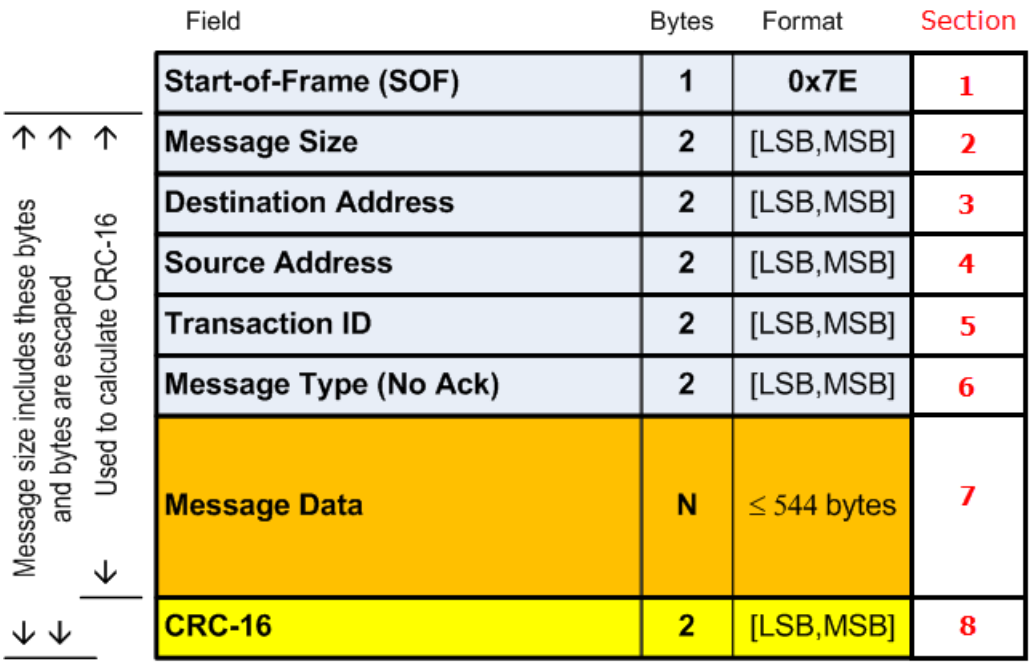
A NAK message indicates that there was a problem with the message. As with the ACK response, a PIB can also use a NAK response to indicate that it has a message queued.

The original message sender has the option of notifying the recipient (using a flag bit) that it doesn't want an ACK/NAK response to the message.

Protocol Message Format

Figure 11.6, “Message Packet Format” shows the format of the message packet for a message containing “N” bytes. Note that all fields are sent in “little-endian” form (LSB first).

Figure 11.6. Message Packet Format



Start-of-Frame Character (Section 1)

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 (Sections 2-8).

Message Size (Section 2)

The number of bytes following the Start-of-Frame, including the CRC bytes. This must be ≤ **556 bytes**, however, the limit does not include any added Escape (ESC) bytes.

Destination Address (Section 3)

This field indicates the intended recipient of the message by address. There is also a special reserved address for broadcast messages.

Source Address (Section 4)

This field indicates the sender of the message. This allows the recipient to know who sent the message (especially in case of an ACK or NAK).

Transaction ID (Section 5)

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 - 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.

Message Type (No ACK) (Section 6)

This is a 2-byte code that indicates the purpose and contents of this message in the least significant 15 bits (0-0x7FFF). The most significant bit (0x8000) contains a flag:

- Messages sent from Parent to Child set the flag to 1 if no acknowledgement should be sent (this is typical for broadcast or informational messages). If the flag is 0, the Child must reply with either an ACK or a NAK message.

Message Data (Section 7)

This field contains 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.

CRC-16 (Section 8)

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.

SOF and ESC Characters and Escaping the Message Body

Sections 2-8 of the message are escaped to prevent the SOF character (0x7E) from appearing more than once in the message. This simplifies interpretation of the incoming message since any SOF character restarts block reception.

The Escape (ESC) character is 0x7D. When either a SOF or an 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. For example, for all bytes following the SOF:

0x7E → 0x7D,0x5E

0x7D → 0x7D,0x5D

The size values and size limits for the messages exclude the count of any ESC characters.

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.

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**

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

PIB’s 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 must not be duplicated by any device on any single Wave Glider vehicle.

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 (board ID 0):

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)	[2]
0x03	Debug Console interface	[3]

Task ID	Description	Notes
0x04	XBee interface (send Level 2 message)	
0x05	File transfer to shore.	[4]
0x06	Default Host (send Level 2 message)	
0x20	Forward full message to target	[5]

Notes:

[1] Level 2 messages to be executed by the C&C should be directed to Task ID **0x01**.

[2] Telemetry messages should be sent without the Level 2 transaction ID field to Task ID **0x02**. The maximum Level 2 message size is 300 bytes.

[3] Using Task ID **0x03**, a device can request to send literal text to the Console.

[4] New File Available messages should be sent to Task ID **0x05**.

[5] Ack's to Write Console commands should be sent to Task ID **0x20**, prefixed by message routing instructions. See the note on Send/Forward ICD Message.

Other Considerations

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.

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	10 to 60000 ms	5000 ms

Setting	Description	Range	Suggest
	before attempting retry or discarding. This time is not end-to-end but between communicating peers		
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 Liquid Robotics on the best course of action, if your requirements are not addressed.

Payload Message Types

[There are issues with tables and page breaks in this section - consider cut and paste of this section into a small doc to facilitate transform time.] Message IDs below **0x1000** are reserved for Liquid Robotics use. Any customer-developed messages specific to their devices must use an ID value greater than **0x1000**.

The reserved Message Types are listed in the following table:

Message Type	Description	Direction	Detail	Broadcast	Reference
0x0000	NAK	Child → Parent	Unable to handle message or request	No	the section called “Generic Payload NAK Data Packet (0x0000)”
0x0001	ACK	Child → Parent	Successfully handled message/ request	No	the section called “Generic Payload ACK Data Packet (0x0001)”
0x0010	Request ID/ Enumerate	Parent → Child	Parent requests device ID and that of any attached devices	Yes	the section called “Request ID/ Enumerate (0x0010)”
0x0015	Telemetry	Parent → Child	Forwards telemetry	Yes	the section called “C&C Telemetry (0x0015)”

Message Type	Description	Direction	Detail	Broadcast	Reference
			message to payloads that have requested it		
0x0022	Request Status	Parent → Child	Parent requests status of Child	No	the section called “Request Status (0x0022)”
0x0023	*Reserved for future use*				
0x0024	Send/ Forward Message	Parent → Child	Parent copies a requested message to child node	Yes	the section called “Send/ Forward Message (0x0024)”
0x0030	Power Status and Control	Parent → Child	Parent notifies Child that power is about to turn off	Yes	the section called “Power Status and Control (0x0030)”
0x0040	Request Queued Message	Parent → Child	Parent requests report of any queued message	No	the section called “Request Queued Message (0x0040)”
0x0041	ACK/NAK Queued Message	Parent → Child	Parent forwards result of message execution	No	the section called “ACK/NAK Queued Message (0x0041)”

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.

Payload Message Packet Descriptions

Payload Header Abbreviation

Payload Header details:

Description	Bytes	Explanation

Header	9	Start-of-Frame + Header data

Start-of-Frame and Header Data details:

Field	Bytes
Start-of-Frame	1
Message Size	2
Destination Address	2
Source Address	2
Transaction ID	2

Generic Payload NAK Data Packet (0x0000)

If a recipient cannot properly handle a message or a 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 on the type of NAK, up to **541** bytes of additional information can be returned. Some responses can return an entire Level 2 NAK message 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** → **3**: Reserved For LRI **4**: Program Download Requested **5** → **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:

Returning NAK Data Message

If the NAK Message Code is 0, the response contains a Level 2 enumerated NAK message including the CRC.

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

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, 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

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 the 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 the 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	Start-of-Frame + 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 Vehicle-to-WGMS 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.

ACK Reply to Enumerate Message

Multiple device information blocks can be returned when multiple responses are aggregated, for example, in the case of a PIB that has daisy-chained devices or other PIB's. Up to 5 devices are currently supported on each payload port. This is done using an ACK with the format shown below:

Description	**Bytes**	**Data Type**
Header	9	Start-of-Frame + 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 (maximum 5)
	2	Number of devices in this message (must be equal to the number of devices responding)
	50	1st Device Information Block (see below)
	50	2nd Device Information Block
	. . .	. . . .
	50	Nth Device Information Block (if necessary)
Error Chk	2	CRC-16

Device Information Block Format

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
Address	2		[Board ID + Task ID] Address of responder [1]

Description	Bytes	Value	Detail
Serial Number	6	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 Request Status message
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)	1		Integer
Firmware (Minor)	1		Integer
Firmware (Revision)	2		Integer
Description	20	ASCII	Human-readable description of device. ASCII text field, padded on the right with spaces

C&C Telemetry (0x0015)

If requested in a Device Information Block, the Float C&C sends 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 using the Iridium/Relay link. 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	Start-of-Frame + Header data

Msg Type	2	**0x0015**
Message Data	1	Structure ID for Telemetry data
	≤ **544**	Level 1 Telemetry data
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. Sometimes telemetry is forwarded using a broadcast and the “no ack” flag is set, in which case no reply is sent.

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	Start-of-Frame + Header data	
Msg Type	2	**0x0022**	
Message Data	6	Time (UTC)	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

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	Start-of-Frame + 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)
	2	Number of responses in this message (must be 1)
	12	1st Status Block
CRC-16	2	

Status Block, Version = 1

The only defined return type is version 1. The Version and Address fields must be set with real values, but the remainder of the block can be set to zero (0) if there is nothing to report.

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)

Send/Forward Message (0x0024)

This message is used to send or to forward Level 2 data packets (messages that conform to the Vehicle-to-WGMS protocol) to a device. These packets have either been requested as copies (of telemetry, for example) or are meant for interaction with systems external to the vehicle. For example, the Level 2 Write Console Script message is sent using Send/Forward Message.

The format is:

Description	Bytes	Data Type	Detail
Header	9	Start-of-Frame + 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**: Vehicle-to-WGMS commands
	≤ **541**	Level 2 Packet	The Level 2 packet 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.

ACK for Forward

If an ACK is requested for the “Forward” case (not typical), the response is an empty ACK message.

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 encapsulated as follows:

Description	Bytes	Data Type
Header	9	Start-of-Frame + 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
Error Chk	2	CRC-16

Note that responding to Write Console is a special case: respond with a basic ACK (Level 2 type 0x13) and the transaction ID, but with a zero-length message payload. Responses to the Write Console message are queued up as follows:

Description	Bytes	Data Type
Header	9	Start-of-Frame + Header data
Msg Type	2	**0x0001**

Message Data	2	Request Queued Message type (0x40) (bit 15 set if additional messages available)
	1	Message format (**2**; 1=WGMS to Vehicle message, 2=Vehicle to WGMS message)
	2	Destination address (saved from the source of the Send/Forward Message)
	2	Source address
	≤ **537**	Level 2 response message, including the CRC
Error Chk	2	CRC-16

Power Status and Control (0x0030)

This message is sent to notify a device of an intended power shutdown.

Description	Bytes	Data Type	Detail
Header	9	Start-of-Frame + Header data	
Msg Type	2	**0x0030**	
Message Data	1	Command Type	Numerical flag value: **0x00**: Notification of power Off
	1	Board Affected	Reserved value: **0x00**: Recipient Device
	1	New Power State	**0x00**: Power Off
Error Chk	2	CRC-16	

ACK Response to Power Status and Control

Multiple response blocks can be returned in an ACK with the format shown below.

Description	Bytes	Data Type
Header	9	Start-of-Frame + Header data
Msg Type	2	**0x0001**
Message Data	2	Original source Msg Type number (bit 15 set if additional messages available)
	2	Number of total devices responding (1)
	2	Number of responses in this message (1)
	6	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**: Board is shut down **0x01**: Board is currently operating
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

Request Queued Message (0x0040)

This packet is sent when a previous message has indicated that there is at least one message waiting to be read.

Description	Bytes	Explanation
Header	9	Start-of-Frame + Header data
Msg Type	2	**0x0040**
Error Chk	2	CRC-16

NAK Response to Request Queued Message

Normally, there is no need to negative-acknowledge Request Queued Message since it is only sent when the board has previously indicated that there is a message waiting. If there is no data to be read, the following NAK is returned:

Description	Bytes	Explanation
Header	9	Start-of-Frame + Header data
Msg Type	2	**0x0000**
Message Data	2	Original source Msg Type number
	1	**0x02** (Invalid Data)
	2	Length = **0x00**
Error Chk	2	CRC-16

ACK Response to Request Queued Message

If there is a message waiting, the response format is:

Description	Bytes	Explanation
Header	9	Start-of-Frame + 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

The format of the data message is dependent on the target address.

ACK/NAK Queued Message (0x0041)

When a payload executes a command by sending the appropriate message to a target device, this message carries back the result of that command. For example, a payload could:

1. Queue up a Set Rudder Position command, targeting address 1 (the Float C&C command interpreter). The format to target the command interpreter (address 1) is:

Description	Bytes	Explanation
Header	9	Start-of-Frame + Header data
Msg Type	2	**0x0001**
Message Data	2	Original source Msg Type number (bit 15 set if additional messages available)
	2	Message format (1)
	≤ **540**	Level 2 data to be executed
Error Chk	2	CRC-16

2. Respond to Request Status and indicate that there are messages waiting to be read.
3. Respond to Request Queued Message with the queued Set Rudder Position message.
4. When the command is executed by the Float C&C, it sends ACK/NAK Queued Message with the result of the command execution.

Description	Bytes	Explanation
Header	9	Start-of-Frame + 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 with no payload, or a NAK with no accompanying data.

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

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		

Start-up Identification Broadcast

PIB		Device
System start-up or device power up, PIB waits for a specified/assumed time after start-up before attempting to communicate. Perhaps multiple		

PIB		Device
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		

Device Receives File from WGMS

PIB		Device
WGMS 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.)		
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

PIB		Device
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 WGMS		*File is now resident on device; this event can be used to interpret the file if it is actually a command or parameter set*

Send File from Device to WGMS

(Send File from Device to WGMS)		
PIB		**Device**
		Device finishes creating a new data set to be sent to WGMS
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 WGMS Task ID)*
Float C&C sends New File Available message to shore side via the configured method		

(Send File from Device to WGMS)		
PIB		**Device**
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		
WGMS 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		
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		

(Send File from Device to WGMS)		
PIB		**Device**
of the device's file from its file system.		

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		

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*

Chapter 12. Payload Communications Hardware

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. Liquid Robotics has developed a common generic communications protocol, described below. Using this protocol allows Liquid Robotics and its customers to work independently on a project, yet retain communication interoperability.

Payload communications hardware can be both of Liquid Robotics-supplied and customer-supplied or created electronic payload boards and devices.

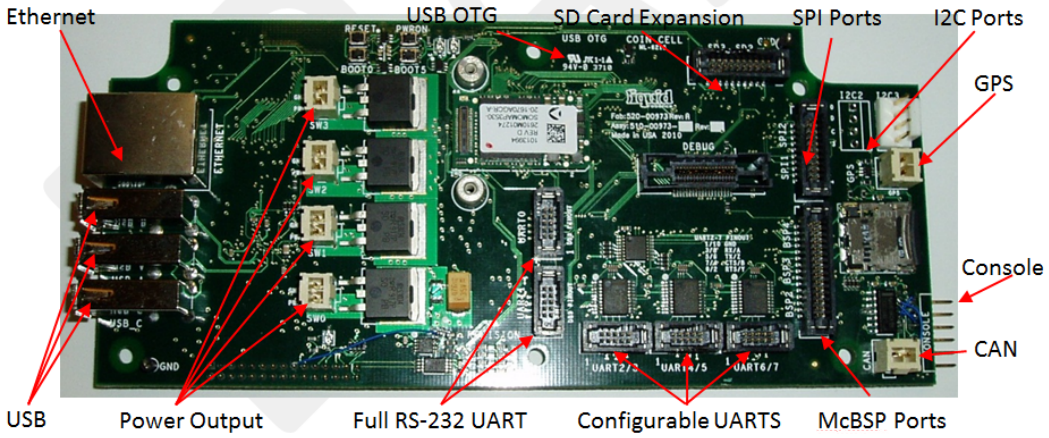
Sensor Management Computer (SMC)

Insert a brief description of the SMCs role in the WG system.

Overview

This section covers the various connectors and interfaces to the Sensor Management Computer (parts 510-00973 and 510-01232). Each connection is described with a pin-out, part numbers for the mating connector/crimp pins and an electrical specification.

Figure 12.1. SMC Connector Locations



Console Port

The console port is the main interface to the processor module. This port is used to login to the Sensor Management Computer Linux console and is the primary debug point. After deployment, this port is typically unavailable. This connection is intended to connect directly to an FTDI TTL-232R-3V3. The TTL-232R-3V3 should be oriented such that pin 1 (black lead) is to the left when reading CONSOLE.

Console Port Pin Out

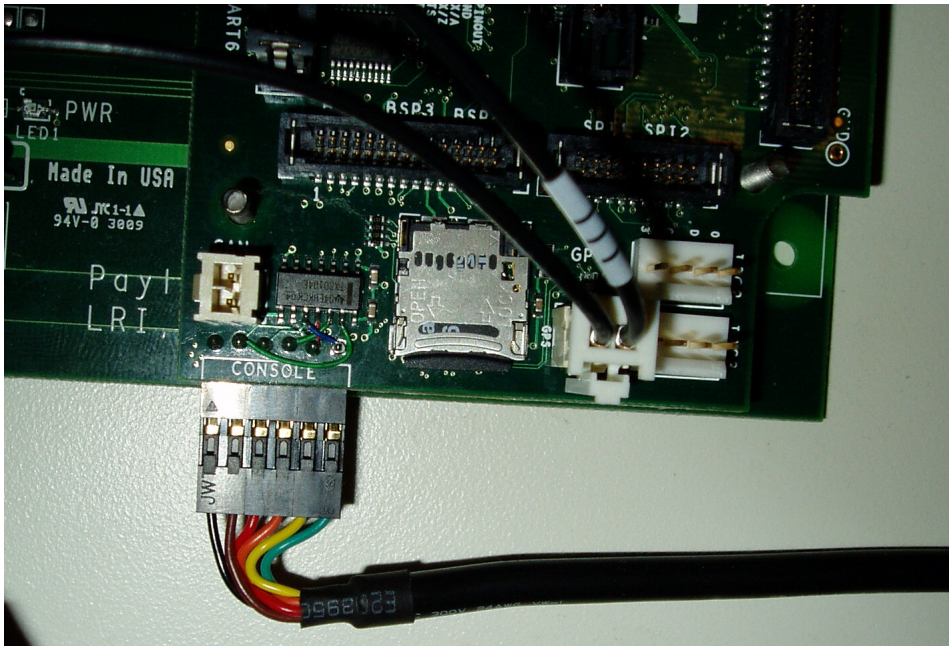
The pin out on the console port is from left to right.

Pin #	SMC Name	Notes
1	GND	System Ground
2	RTS#	Not currently used, but available to support hardware flow control
3	NC	5V signal from USB converter
4	RXD	Data into the SMC
5	TXD	Data out of the SMC
6	CTS#	Not currently used, but available to support hardware flow control

Console Port Electrical Interface

The console port is a 3.3 volt level CMOS serial port intended to be interfaced to a level converter to support either a USB interface or an RS-232 style interface. For a detailed specification of the I/O levels, see the data sheet for the TI TXS0104E.

Figure 12.2. Console Port Connection



Full RS-232 Ports (UART0 and UART1)

The full RS-232 ports implement a complete RS-232 interface that is compatible with a standard DB-9 pin out. One option for connecting up to this port is to use the Liquid Robotics Inc. cable assembly 140-01233 that converts the connector directly to a DB9. The connector is keyed and latching for robustness.

Full RS-232 Port Pin Out

Pin #	SMC Name	Notes
1	CD	Carrier Detect output

Pin #	SMC Name	Notes
2	RX	Data into the UART
3	TX	Data out of the UART
4	DTR	Data Terminal Ready output
5	GND	Ground
6	DSR	Data Set Ready input
7	RTS	Request To Send output
8	CTS	Clear To Send input
9	RI	Ring Indicator input
10	GND	Ground, but intended as a no connect

Full RS-232 Electrical Interface

These connections are compatible with all standard RS-232 drivers.

Full RS-232 Mating Connector

There are two options for a mating connector from Samtec. The preferred option is to order a pre-built cable assembly such as SFSD-05-28-G-05.00-SR. The length can be adjusted as needed by changing the 4-digit number between the G and SR (i.e. the XX.XX in SFSD-05-28-G-XX.XX-SR). The second option is to use a crimp housing and crimp pins, part numbers ISDF-05-D-M and CC03R/L.

Configurable UART Ports (UART2/3, UART4/5, and UART6/7)

The configurable UART ports implement a dual, software configurable UART port that supports any of the following three options, RS-232 with RTS/CTS support, RS-485 full/half duplex and RS-422. One option for connecting to this port is to use the Liquid Robotics Inc. cable assembly 140-01234 that converts the connector directly to two DB9s. The connector is keyed and latching for robustness.

Configurable UART Port Pin Out

Pin #	SMC Name	Notes
1	GND-A	Ground for even numbered port
2	RTS-B	Request To Send output for odd numbered port
3	RX-A	Data into the UART for the even numbered port
4	CTS-B	Clear To Send input for even numbered port
5	TX-A	Data out of the UART for the even numbered port
6	TX-B	Data out of the UART for the odd numbered port

Pin #	SMC Name	Notes
7	CTS-A	Clear To Send input for even numbered port
8	RX-B	Data into the UART for the odd numbered port
9	RTS-A	Request To Send output for even numbered port
10	GND-B	Ground for odd numbered port

Configurable UART Electrical Interface

These connections are supported by the Maxim driver IC MAX3160EEAP+

Configurable UART Mating Connector

There are two options for a mating connector from Samtec. The preferred option is to order a pre-built cable assembly such as SFSD-05-28-G-05.00-SR. The length can be adjusted as needed by changing the 4-digit number between the G and SR (i.e. the XX.XX in SFSD-05-28-G-XX.XX-SR). The second option is to use a crimp housing and crimp pins, part numbers ISDF-05-D-M and CC03R/L.

GPS/PPS Port

The GPS/PPS port is the interface to the GPS/PPS data coming in from the Float. This connection connects directly to the incoming connection or to the header on the PIB. The SMC is expecting the GPS string to be at 4800 Baud and contain the following three NEMA strings, GGA, ZDA, and VTG, using RS-232 levels. The connector is keyed and latching for robustness.

GPS/PPS Port Pin Out

Pin #	SMC Name	Notes
1	GPS	GPS Data into the system – Connects to pin 8 of the PIB header
2	PPS	PPS timing pulse into the system – connects to pin 1 of the PIB header

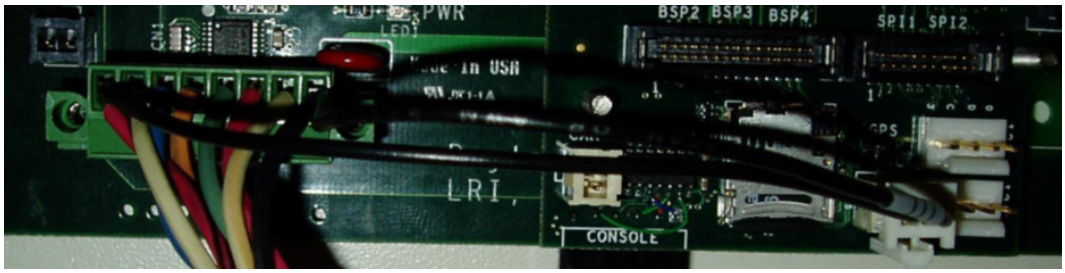
GPS/PPS Electrical Interface

These inputs are capable of handling +/-15 volt signals. They are intended to level shift RS-232 signals down to 1.8 volt CMOS logic. These inputs are simple N-FET level shifts/inverters.

GPS/PPS Mating Connector

There are two options for a mating connector from Samtec. The preferred option is to order a pre-built cable assembly such as MMSS-02-24-L-05.00-S-K. The length can be adjusted as needed by changing the 4-digit number between the L and S (i.e. the XX.XX in MMSS-02-24-L-XX.XX-S-K). The second option is to use a crimp housing and crimp pins, part numbers IPD1-02-S-K and CC79R/L.

Figure 12.3. GPS/PPS Connections to a PIB



Power Ports

The power ports are software controllable switches to enable/disable sensors as needed. The connector is keyed and latching for robustness.

Power Port Pin Out

Pin #	SMC Name	Notes
1	PWR	12 Volt +/-10% output at up to 2 A
2	GND	Ground

Power Electrical Interface

These outputs are capable of providing 12 volts +/-10% at up to 2 amps. The total system current only supports up to 3 amps into the SMAC so the sum of currents out is the limiting factor, not the switches. The current through the switch can be read in software.

Power Mating Connector

There are two options for a mating connector from Samtec. The preferred option is to order a pre-built cable assembly such as MMSS-02-24-L-05.00-S-K. The length can be adjusted as needed by changing the 4-digit number between the L and S (i.e. the XX.XX in MMSS-02-24-L-XX.XX-S-K). The second option is to use a crimp housing and crimp pins, part numbers IPD1-02-S-K and CC79R/L.

CAN Port

The CAN port is currently not supported, but is intended for inter-SMC communication and/or for specific sensors that require a CAN interface. The information provided here is for future use only. The connector is keyed and latching for robustness.

CAN Port Pin Out

Pin #	SMC Name	Notes
1	CANH	CAN non-inverting output
2	CANL	CAN inverting output

CAN Electrical Interface

This output is supported by a TI SN65HVD230DR transceiver. One thing to note is that this interface is running at 3.3 volts, and although this meets the ISO 11898 standard (CAN specifications), it may not be compatible with older 5.0 volt CAN bus systems.

CAN Mating Connector

There are two options for a mating connector from Samtec. The recommended option is to order a pre-built cable assembly such as MMSS-02-24-L-05.00-S-K. The length can be adjusted as needed by changing the 4-digit number between the L and S (i.e. the XX.XX in MMSS-02-24-L-XX.XX-S-K). The second option is to use a crimp housing and crimp pins, part numbers IPD1-02-S-K and CC79R/L.

USB Ports (not USB OTG)

These ports are not supported.

USB OTG Port

The USB OTG port meets the USB OTG specification. The USB OTG interface is currently not supported in the Linux kernel. To use the USB OTG port as the host a special mini A to B (or mini B) cable is required. A standard A to mini B cable will turn the SMC into a slave device.

USB OTG Port Pin Out

See the USB OTG specification.

USB OTG Electrical Interface

See the USB OTG specification.

USB OTG Mating Connector

See the USB OTG specification.

Ethernet Port

The Ethernet port meets the 10/100 Base T Ethernet specification. The connector is specified as a latching RJ-45.

Ethernet Port Pin Out

See the Ethernet specification.

Ethernet Electrical Interface

See the Ethernet specification.

Ethernet Mating Connector

See the Ethernet specification.

SD Card Expansion Port

The SD Card Expansion port supports additional data storage. This interface connects directly to the Torpedo module/OMAP processor so the signals must be level shifted to 1.8 volts. This port is untested.

SD Card Expansion Port Pin Out

Pin #	SMC Name	Notes
1	1V8	1.8 Volt output for supplying a logic level reference, limit draw to less than 10mA
2	3V3	3.3 volts for powering any needed circuits
3	SD2_CLK	SD card interface #2 clock output
4	SD2_CDn	SD card interface #2 card detect input
5	SD2_DAT3	SD card interface #2 data bit 3 I/O
6	SD2_CMD	SD card interface #2 command output
7	SD2_DAT1	SD card interface #2 data bit 1 I/O
8	SD2_DAT2	SD card interface #2 data bit 2 I/O
9	GND	Ground
10	SD2_DAT0	SD card interface #2 data bit 0 I/O
11	1V8	1.8 Volt output for supplying a logic level reference, limit draw to less than 10mA
12	3V3	3.3 volts for powering any needed circuits
13	SD3_CLK	SD card interface #3 clock output
14	SD3_CDn	SD card interface #3 card detect input
15	SD3_DAT3	SD card interface #3 data bit 3 I/O
16	SD3_CMD	SD card interface #3 command output
17	SD3_DAT1	SD card interface #3 data bit 1 I/O
18	SD3_DAT2	SD card interface #3 data bit 2 I/O
19	GND	Ground
20	SD3_DAT0	SD card interface #3 data bit 0 I/O

SD Card Expansion Electrical Interface

The inputs and outputs here are referenced to the OMAP 1.8 volt logic levels. The 1.8 volt output is intended only for level shifters as needed and the rest of the circuit should be run from 3.3 volts.

SD Card Expansion Mating Connector

There are two options for a mating connector from Samtec. The recommended option is to order a pre-built cable assembly such as SFSD-10-28-G-05.00-SR. The length can be adjusted as needed by changing the 4-digit number between the G and SR (i.e. the XX.XX in SFSD-10-28-G-XX.XX-SR). The second option is to use a crimp housing and crimp pins, part numbers ISDF-10-D-M and CC03R/L.

SPI Expansion Port

The SPI Expansion port supports additional circuitry for expanded operation. This interface connects directly to the Torpedo module/OMAP processor so the signals will need to be level shifted to 1.8 volts. This port is untested.

SPI Expansion Port Pin Out>

Pin #	SMC Name	Notes
1	1V8	1.8 Volt output for supplying a logic level reference, limit draw to less than 10mA
2	3V3	3.3 volts for powering any needed circuits
3	SPI1_CS2	SPI interface #1 chip select 2 output
4	SPI1_CS3	SPI interface #1 chip select 3 output
5	SPI1_CS0	SPI interface #1 chip select 0 output
6	SPI1_CS1	SPI interface #1 chip select 1 output
7	SPI1_SIMO	SPI interface #1 slave in master out output
8	SPI1_CLK	SPI interface #1 clock output
9	GND	Ground
10	SPI1_SOMI	SPI interface #1 slave out master in input
11	1V8	1.8 Volt output for supplying a logic level reference, limit draw to less than 10mA
12	3V3	3.3 volts for powering any needed circuits
13	SPI2_CS2	SPI interface #2 chip select 2 output
14	SPI2_CS3	SPI interface #2 chip select 3 output
15	SPI2_CS0	SPI interface #2 chip select 0 output
16	SPI2_CS1	SPI interface #2 chip select 1 output
17	SPI2_SIMO	SPI interface #2 slave in master out output
18	SPI2_CLK	SPI interface #2 clock output
19	GND	Ground

Pin #	SMC Name	Notes
20	SPI2_SOMI	SPI interface #2 slave out master in input

SPI Expansion Electrical Interface

The inputs and outputs here are referenced the OMAP 1.8 volt logic levels. The 1.8 volt output is intended only for level shifters as needed and the rest of the circuit should be run from 3.3 volts.

SPI Expansion Mating Connector

There are two options for a mating connector from Samtec. The recommended option is to order a pre-built cable assembly such as SFSD-10-28-G-05.00-SR. The length can be adjusted as needed by changing the 4-digit number between the G and SR (i.e. the XX.XX in SFSD-10-28-G-XX.XX-SR). The second option is to use a crimp housing and crimp pins, part numbers ISDF-10-D-M and CC03R/L.

McBSP Expansion Port

The McBSP Expansion port supports additional circuitry for expanded operation. This interface connects directly to the Torpedo module/OMAP processor so the signals will need to be level shifted to 1.8 volts. This port is untested.

McBSP Expansion Port Pin Out

Pin #	SMC Name	Notes
1	1V8	1.8 Volt output for supplying a logic level reference, limit draw to less than 10mA
2	3V3	3.3 volts for powering any needed circuits
3	BSP2_I/O1	McBSP interface #2 general purpose I/O 1
4	BSP2_I/O0	McBSP interface #2 general purpose I/O 0
5	BSP2_DR	McBSP interface #2 DR
6	BSP2_I/O2	McBSP interface #2 general purpose I/O 2
7	BSP2_FSX	McBSP interface #2 FSX
8	BSP2_DX	McBSP interface #2 DX
9	GND	Ground
10	BSP2_CLKX	McBSP interface #2 CLKX
11	1V8	1.8 Volt output for supplying a logic level reference, limit draw to less than 10mA
12	3V3	3.3 volts for powering any needed circuits
13	BSP3_I/O1	McBSP interface #3 general purpose I/O 1

Pin #	SMC Name	Notes
14	BSP3_I/O0	McBSP interface #3 general purpose I/O 0
15	BSP3_DR	McBSP interface #3 DR
16	BSP3_I/O2	McBSP interface #3 general purpose I/O 2
17	BSP3_FSX	McBSP interface #3 FSX
18	BSP3_DX	McBSP interface #3 DX
19	GND	Ground
20	BSP3_CLKX	McBSP interface #3 CLKX
21	1V8	1.8 Volt output for supplying a logic level reference, limit draw to less than 10mA
22	3V3	3.3 volts for powering any needed circuits
23	BSP4_I/O1	McBSP interface #4 general purpose I/O 1
24	BSP4_I/O0	McBSP interface #4 general purpose I/O 0
25	BSP4_DR	McBSP interface #4 DR
26	BSP4_I/O2	McBSP interface #4 general purpose I/O 2
27	BSP4_FSX	McBSP interface #4 FSX
28	BSP4_DX	McBSP interface #4 DX
29	GND	Ground
30	BSP4_CLKX	McBSP interface #4 CLKX

McBSP Expansion Electrical Interface

The inputs and outputs here are referenced to the OMAP 1.8 volt logic levels. The 1.8 volt output is intended only for level shifters as needed and the rest of the circuit should be run from 3.3 volts.

McBSP Expansion Mating Connector

There are two options for a mating connector from Samtec. The recommended option is to order a pre-built cable assembly such as SFSD-15-28-G-05.00-SR. The length can be adjusted as needed by changing the 4-digit number between the G and SR (i.e. the XX.XX in SFSD-15-28-G-XX.XX-SR). The second option is to use a crimp housing and crimp pins, part numbers ISDF-15-D-M and CC03R/L.I2C Ports (I2C2 and I2C3)

I2C Ports

The I2C ports are for debug only in the 510-00973 design. In the 510-01232 design, they have an expansion capability as specified below. The 510-00973 connectors are keyed. The 510-01232 PCBA connector is keyed and latching for robustness.

I2C Ports 510-01232 Pin Out

Pin #	SMC Name	Notes
1	3V3	3.3 volt output to power external circuits
2	3V3	3.3 volt output to power external circuits
3	I2C2_SCL	I2C2 serial clock output
4	I2C3_SCL	I2C3 serial clock output
5	I2C2_SDA	I2C2 serial data I/O
6	I2C3_SDA	I2C3 serial data I/O
7	GND	Ground
8	GND	Ground
9	GND	Ground
10	GND	Ground

I2C Electrical Interface

This interface conforms to the I2C specification for 3.3 volt logic levels and supports up to 400 KHz operation.

I2C Ports 510-01232 Mating Connector

There are two options for a mating connector from Samtec. The recommended option is to order a pre-built cable assembly such as SFSD-05-28-G-05.00-SR. The length can be adjusted as needed by changing the 4-digit number between the G and SR (i.e. the XX.XX in SFSD-05-28-G-XX.XX-SR). The second option is to use a crimp housing and crimp pins, part numbers ISDF-05-D-M and CC03R/L.

Payload Interface Board (PIB)

This section covers Payload Interface Board, including how it fits into the system architecture and how to interface to it.

Payload Interface Board Architecture

The Payload Interface Board provides multiple sensors for leak detection and environmental monitoring. It also provides a common interface for Personality Modules, which may be quickly modified to adapt to end user equipment.

Figure 12.4. Simple Single PIB with 3 Personality Modules Connected to 3 Devices

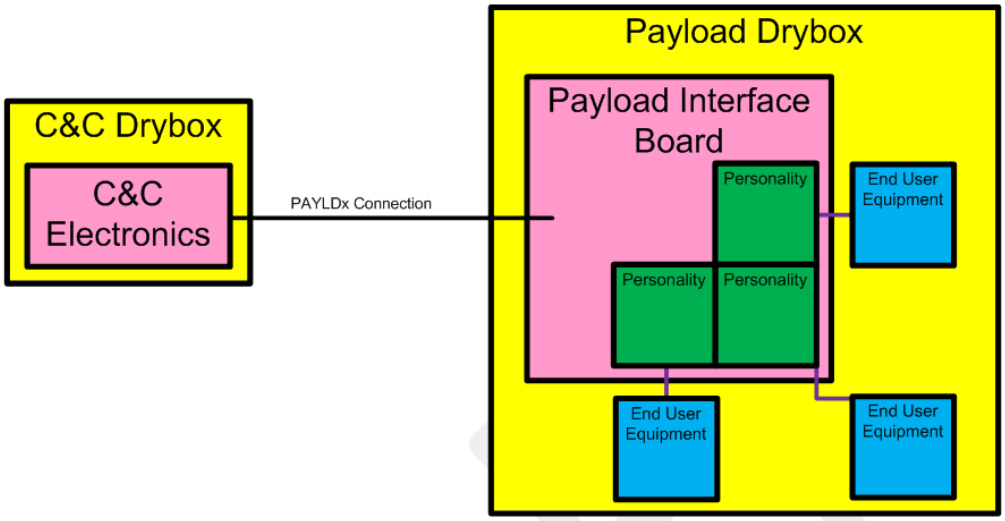


Figure 12.5. Simple Double PIB Connected to 6 Devices

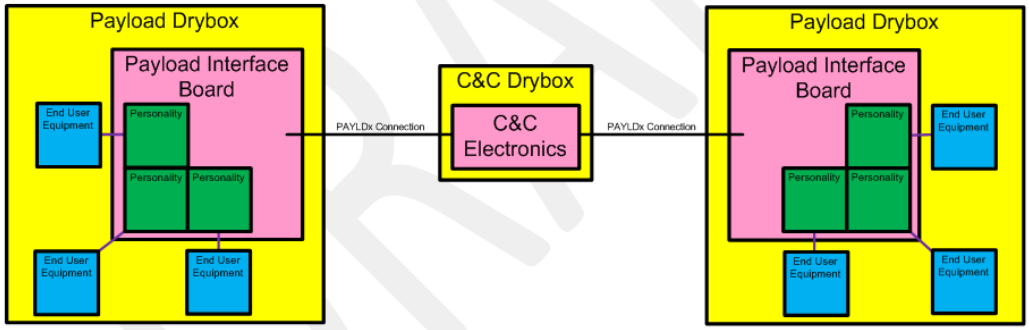
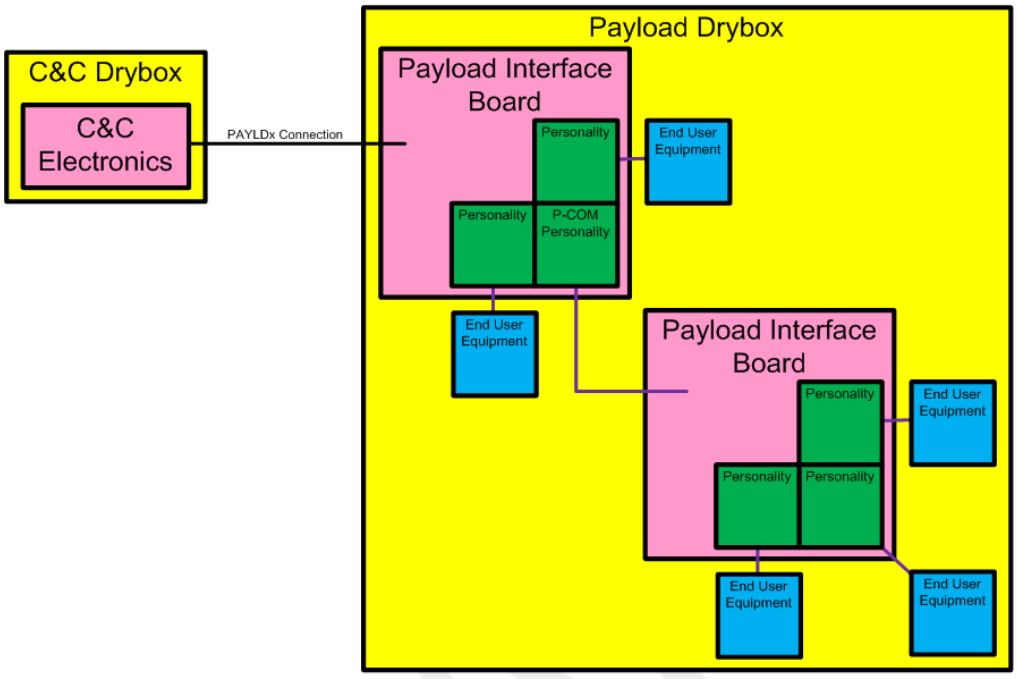


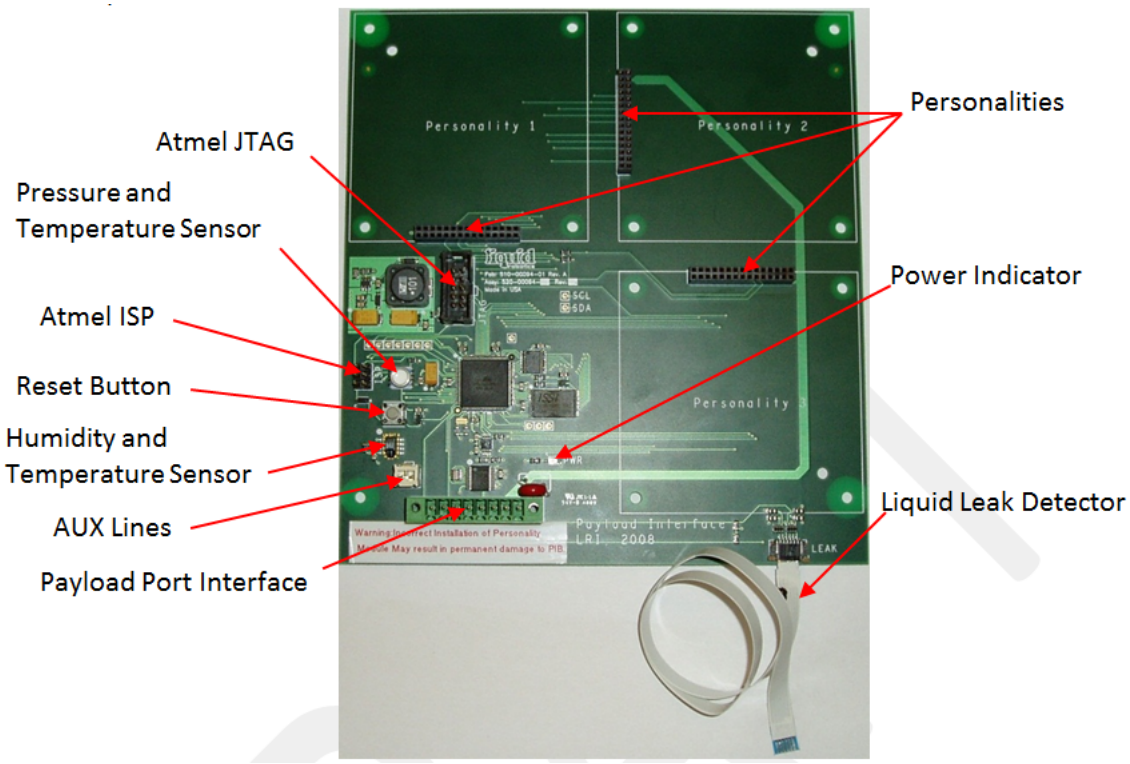
Figure 12.6. Daisy Chained PIBs Used to Expand Device Count in a Single Dry Box



Payload Interface Board Interfaces

This section covers the various connectors and interfaces to the Payload Interface Board (510-00094). Each connection is described with a pin-out, part numbers for the mating connector/crimp pins and an electrical specification.

Figure 12.7. PIB Connector and Sensor Locations



Pressure and Temperature Sensor

The pressure and temperature sensor provides feedback to WGMS on the state of the dry box. The pressure is temperature compensated. It may be used as an early detection mechanism that the seal has failed.

Maximum pressure on the sensor is 14 bars, with a resolution of 1.2 millibar and an accuracy of +/-200 millibar. WGMS reports this value to the nearest kilo-Pascal. The temperature range is -40°C to 85 C with a resolution of 0.015 C. and an accuracy of -2/+6 C. WGMS reports this with a resolution of 1 C.

Reset Button

The reset button is provided for development, testing and debugging. It resets the micro controller when pressed.

Power Indicator

The power indicator is provided for development, testing and debugging. It notifies the user when the 3.3 volt regulator is functioning, which means a minimum of 6 volts is coming from J7 pin 2 to the regulator. It is not an indicator of the micro controller health.

Humidity and Temperature Sensor

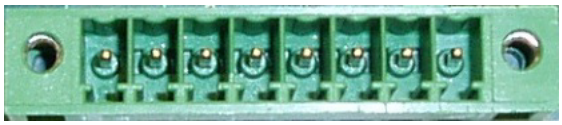
The humidity sensor provides feedback to the user on the state of the dry box. The humidity is temperature compensated. It may be used as an early detection mechanism that the seal has failed. The humidity range is 0 to 100% with an accuracy of +/-3% between 20 and 80% relative humidity. The sensor has a resolution

of 0.05% relative humidity. The reading is temperature compensated. WGMS reports this with a resolution of 1% relative humidity. The temperature range is -40°C to 85 C with a resolution of 0.4 C. and an accuracy of +/-2.5 C. WGMS reports this with a resolution of 1 C.

Payload Port Interface – J7

The payload port interface connector typically is the interface to the Float C&C. This is a disconnect able screw terminal to allow for easy removal of the PIB or the standard dry box penetrator cable. The connector is manufactured by Tyco and is part number 284516-8.

Figure 12.8. Payload Port Pin Number Orientation



Payload Port Interface - J7 Pin Out

Pin #	Signal Name	Description	Wire Color
1	PPS	GPS pulse-per-second, using RS-232 levels	Black
2	PWR	13.2 Volts regulated at 3.0A continuous	White
3	TX	RS-232 Transmit Line	Red
4	RX	RS-232 Receive Line	Green
5	AUXA	Glider Connections, exact connections are determined by which payload port it is	Orange
6	AUXB	Glider Connections, exact connections are determined by which payload port it is	Blue
7	GND	Ground reference for all pins	Yellow
8	GPS Data	4800 Baud GGA, ZDA, and VTG, using RS-232 levels, 1 second report rate	White/Black

Payload Port Interface - J7 Electrical Interface

These connections are compatible with all the specification for the PAYLDx ports on the Float C&C dry box.

PWR and GND

The power pin requires 12 volts +/-10% at up to 3.0A continuous draw.

TX and RX

These communication lines connect between Float C&C and PIB processors. This connection is RS-232 at 115,200 baud rate. These signals are provided via a MAX232 compatible circuit.

PPS and GPS Signals

The PIB does not directly connect to these pins. If the end user equipment needs this information, connect directly to the wires on the penetrator or add additional connections into the screw terminals.

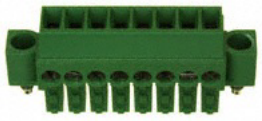
Auxiliary Lines – AUXA and AUXB

The auxiliary lines are provided to give direct access to other payloads on the Float or to connect to the Glider assembly. These pins connect to J6 (AUX lines) pins 1 and 2 respectively.

Payload Port Interface - J7 Mating Connector

The mating connector is manufactured by Tyco, part number 284510-8. Below is an example photo. It is recommended that ferrules be used with the loose wires as well.

Figure 12.9. PIB to Penetrator Example Connector



AUX Lines - J6

The AUX lines connector is intended to simplify cabling from the auxiliary lines to a personality module. This is a keyed and latching connector for robustness. The connector is manufactured by Samtec and is part number IPL1-102-01-S-S.

Figure 12.10. AUX Lines J6 Pin Number Orientation



AUX Lines - J6 Pin Out

Pin #	Signal Name	Description
1	AUXA	Connects to Payload Port Interface pin 5
2	AUXB	Connects to Payload Port Interface pin 6

AUX Lines – J6 Electrical Interface

These connections are direct shorts back to the Payload Port Interface. It is recommended that the voltage between the two pins not exceed 50 volts or 2.5 amps.

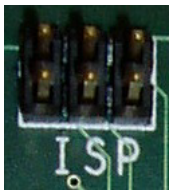
AUX Lines – J6 Mating Connector

There are two options for a mating connector from Samtec. The first and preferred option is to order pre-built cable assembly like MMSS-02-24-L-05.00-S-K. The length can be adjusted as needed by changing the 4-digit number between the L and S (i.e. the XX.XX in MMSS-02-24-L-XX.XX-S-K). The second option is to use a crimp housing and crimp pins, part numbers IPD1-02-S-K and CC79R/L.

Atmel ISP – J5

The Atmel ISP connector is intended to be the production programming interface port. This dual row 0.100” pitch header is unshrouded for quick connection and disconnection in a production environment. This connection is intended for an Atmel ATAVRISP2, but will work with other ATMEL programmers.

Figure 12.11. Atmel ISP - J5 Pin Number Orientation



Atmel ISP – J5 Pin Out

Pin #	Signal Name	Description
1	MISO-ISP	Master In – Slave Out
2	VCC	+3.3 VDC
3	SCLK	Clock for synchronizing data
4	MOSI	Master Out – Slave In
5	RESET	Pin 5 Grounded → Reset
6	GND	Ground reference for all pins

Atmel ISP – J5 Electrical Interface

These connections are compatible with the Atmel ATmega ISP programming interface running at 3.3 volts.

PWR and GND

The power pin is used to provide a reference voltage to the programming adapter. No more than 30mA should be sourced from this pin during programming.

MISO-ISP, MOSI, SCLK and RESET

These are the communication lines that connect between the programmer and the micro controller. This link should only be used with an Atmel approved programming device.

Atmel ISP – J5 Mating Connector

The mating connector is either an Atmel ATJTAGICE2 with 10 pin to 6 pin adapter cable or an Atmel ATAVRISP2. Other Atmel programmers should work as well, but only these two have been tested over this interface.

Figure 12.12. Atmel ATJTAGICE2 without the adapter cable

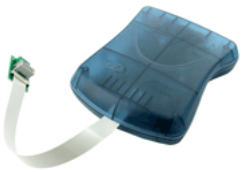


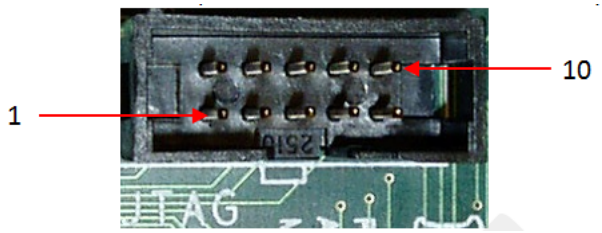
Figure 12.13. Atmel ATAVRISP2



Atmel JTAG – J4

The Atmel JTAG connector is intended to be the development programming and debug port. This dual row 0.100” pitch header is shrouded with a key for reliable orientation in a development environment. This connection is for an Atmel ATJTAGICE2, but will work with other ATMEL programmers.

Figure 12.14. Atmel JTAG - J4 Pin Number Orientation



Atmel JTAG – J4 Pin Out

Pin #	Signal Name	Description
1	TCK	Clock
2	GND	Ground reference for all pins
3	TDO	Data Out
4	VCC	+3.3 VDC
5	TMS	Test Mode Select
6	RESET	Pin 6 Grounded → Reset
7	VCC	+3.3 VDC
8	N/C	No Connection
9	TDI	Data In
10	GND	Ground reference for all pins

Atmel JTAG – J4 Electrical Interface

These connections are compatible with the Atmel ATmega JTAG programming and debugging interface running at 3.3 volts.

PWR and GND

The power pin is used to provide a reference voltage to the programming adapter. No more than 30mA should be sourced from this pin during programming.

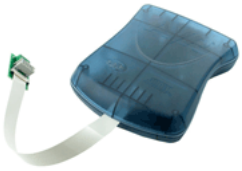
TCK, TDO, TMS, TDI and RESET

These are the communication lines that connect between the programmer and the micro controller. This link should only be used with an Atmel approved programming device.

Atmel JTAG – J4 Mating Connector

The mating connector is an Atmel ATJTAGICE2. Other Atmel programmers should work as well, but this is the only one that has been tested over this interface.

Figure 12.15. Atmel ATJTAGICE2



Personality Interface - J1, J2, J3

The personality interface connector is intended to be the interface to a variety of personality modules. This interface was designed around flexibility and the ability to identify the specific location. This is a 2mm, dual row, 30 pin header. The connectors are not keyed or shrouded. Warning incorrect installation of Personality Modules will result in permanent damage to the PIB. When installing a personality please remove the connector at J7 to guarantee that there is no power present. The connector is manufactured by Samtec and is part number SQT-115-01-L-D.

Figure 12.16. Personality Connector Locations

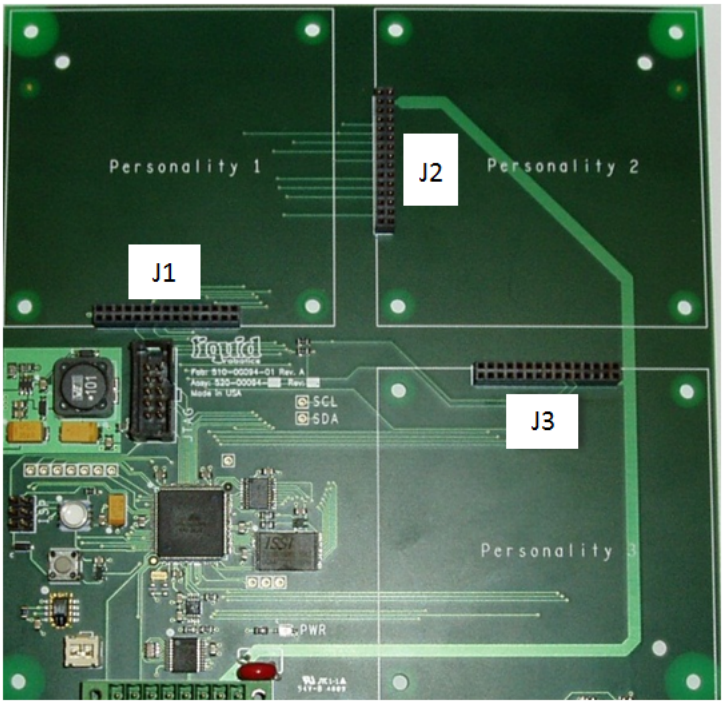


Figure 12.17. Pin Number Orientation



Personality Interface – J1, J2, J3 Pin Out

Pin #	Signal Name	Description
1	PWR	Power from C&C
2	GND	Ground
3	PWR	Power from C&C
4	GND	Ground
5	VCC	Regulated +3.3V
6	GND	Ground
7	SDA	I2C Data Line
8	GND	Ground
9	SCL	I2C Clock Line
10	GND	Ground
11	TXD	UART Transmit to Personality Module
12	GND	Ground

Pin #	Signal Name	Description
13	RXD	UART Receive from Personality Module
14	GND	Ground
15	XCK GPIO	Synchronous Serial Clock GPIO (optional)
16	GND	Ground
17	PON	Personality Module Power-On/ Enable
18	GND	Ground
19	POK	Personality Module Power OK Response
20	GND	Ground
21	GPIO-0	Connection to micro controller
22	GND	Ground
23	GPIO-1	Connection to micro controller
24	GND	Ground
25	GPIO-2	Connection to micro controller
26	GND	Ground
27	GPIO-3	Connection to micro controller
28	GND	Ground
29	ADDR2	Personality Module address 2nd bit
30	ADDR1	Personality Module address 1st bit

Personality Interface – J1, J2, J3 Electrical Interface

These connections are 3.3 volt CMOS connections directly to the ATmega2560 on the PIB

PWR

The power pins are a direct connection to J7 pin 2. They should be limited to no more than 1.0 amps per personality module.

VCC

The regulated 3.3 volt supply can provide up to 250 milliamps for the personality modules. This is a sum of all three ports, not a per port limitation. If more current is drawn by the personality modules, the PIB processor can reset.

GND

The ground pins are a direct connection to J7 pin 7 through an internal plane on the PIB. These should all be connected together on the personality module for robustness.

SCL and SDA

These communication lines are common to all of the personality modules. They conform to the I2C specification for a 3.3 volt bus at up to 400 KHz. The intention of this bus is to allow inter-personality communication and/or a common interface for automatically detecting personality modules. I2C is available to the personality module. It is also intended to be connected to an I2C serial EEPROM (Atmel AT24CxxB [except for the AT24C08B]) on each personality module, but could be used to control the personality as well. I2C addresses 7'b1010_xxx should be reserved for EEPROM storage on personality modules. Pull-ups are already provided on the PIB.

TX, RX and XCK

These communication lines provide for an interface to a USART on the micro controller. If the USART is used in asynchronous mode, the XCK line may be used as a general purpose I/O pin from the micro controller.

Connector	**USART**
J7 (for reference only)	USART0
J1/Personality 1	USART1
J2/Personality 2	USART2
J3/Personality 3	USART3

PON and POK

The PON and POK signals are used to provide an enable/power on pin and feedback that the system is on and running. If the PON pin will not be used, tie it directly to POK so that it is compatible with all other personality modules. These pins are 3.3 volt CMOS compatible. PON should be brought high and sometime later POK should come high to indicate that the personality module is functional.

GPIO-X

The GPIO lines are provided to give direct access to I/O on the micro controller. Refer to the ATmega2560 data sheet for specific pin requirements. The table below shows the specific connections made from each personality module to the micro controller.

Personality Module Interface	**J1**	**J2**	**J3**
21/GPIO-0	PK0	PH4	PF0
23/GPIO-1	PK1	PH5	PF1
25/GPIO-2	PK2	PH6	PF2
27/GPIO-3	PK3	PH7	PF3

ADDR1 and ADDR2

The address 1 and address 2 pins are used to identify which specific personality location a personality has been inserted into. These pins should be tied to the I2C EEPROM, but can also be read by the personality module to determine location. The table below shows the specific addressing of each location on the PIB.

Location	**ADDR1/Pin 30**	**ADDR2/Pin 29**
J1	GND	GND

Location	ADDR1/Pin 30	ADDR2/Pin 29
J2	GND	VCC
J3	VCC	GND

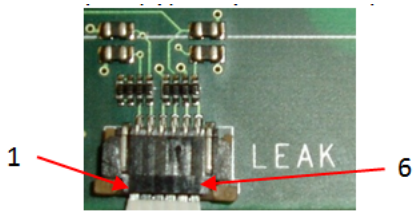
Personality Interface – J1, J2, J3 Mating Connector

The mating connector is manufactured by Samtec, and is part number TMM-115-03-G-D.

Liquid Leak Detector - J8

The liquid leak detector connector is designed to interface to a custom cable assembly from Liquid Robotics, part number 140-00277-012. This cable should be mounted at the lowest point in the dry box to detect the presence of a conductive liquid (typically salt water) leaking into the housing.

Figure 12.18. Liquid Leak Detector - J8 Pin Number Orientation



Liquid Leak Detector – J8 Pin Out

Pin #	Signal Name	Description
1	LEAKCO	Leak cable Continuity Out
2	LEAKCI	Leak cable Continuity In
3	WEAK PULL DOWN	Grounded thru 1K
4	LEAK1	Connection to weak pull down indicates a leak
5	WEAK PULL DOWN	Grounded thru 1K
6	LEAK0	Connection to weak pull down indicates a leak

Liquid Leak Detector – J8 Electrical Interface

These connections are 3.3 volt CMOS connections with ESD protection.

LEAKCO and LEAKCI

The leak cable continuity out and in pins are used to detect the presence of the leak detector cable. The continuity out pin drives both a high and low signal and checks to make sure the continuity in pin follows it.

LEAK0, LEAK1 and WEAK PULL DOWN

The LeakX pins have weak internal pull-ups and the micro controller watches for these pins to be pulled low indicating a short to the weak pull down pins. These pins are pulled up to 3.3 volts via 47 kilo-ohms resistors.

Liquid Leak Detector – J8 Mating Connector

The mating connector is manufactured by Liquid Robotics and is part number 140-00277-012.

Glider (Sub) Payload Interface Board (SubPIB)

This section covers the Sub Payload Interface Board, both 510-01581 and 510-01086, including how it fits into the system architecture and how to interface to it. If anything is not fully described here, refer to the PIB section as an alternate reference. All of the connector reference designators are the same and this design is completely based on that design.

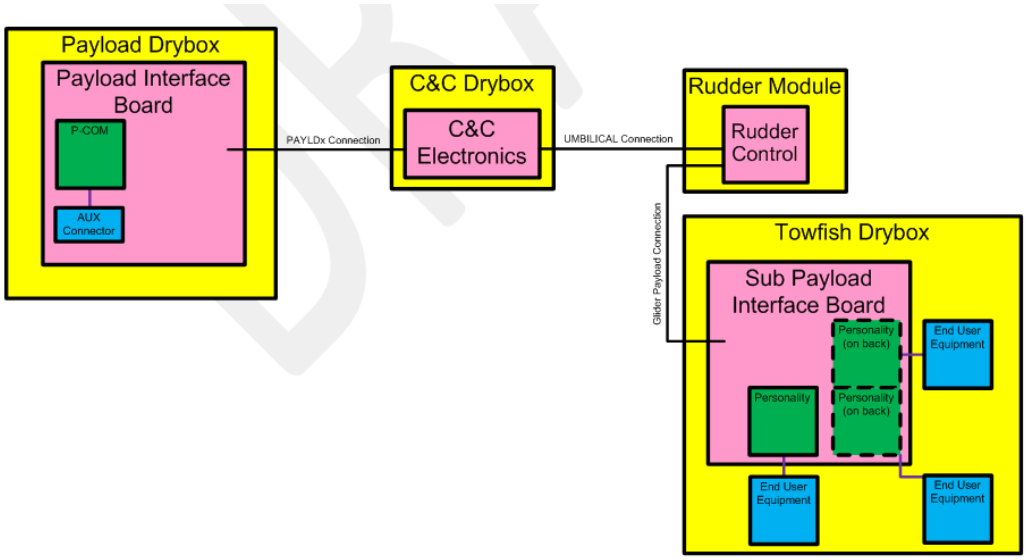
Sub Payload Interface Board Architecture

The Sub Payload Interface Board is similar to the Payload Interface Board in functionality, but is built around a double-sided assembly architecture that allows for the finished assembly to fit into a towfish assembly or where the smaller form factor is desired. It provides multiple sensors for leak detection and environmental monitoring. It also provides a common interface for Personality Modules, which are intended to be quickly and easily modified to adapt to the End User Equipment.

The main difference for the Sub PIB is that it is intended to be installed in a towfish that does not have a direct connection to the C&C. The figure below shows a basic setup for any towfish design that needs to connect to the C&C. The auxiliary lines are not shown, but it is assumed that Sub PIB is communicating on one of the two pairs of auxiliary lines (which pair depends on the PAYLDx port used on the C&C) to a PIB with a P-COM in a Float payload dry box. The power is still delivered via the Rudder Module controls.

The example below is a very basic example and can be combined with or used in scenarios similar to a standard PIB. This includes multiple towfish assemblies and/or payload boxes that have their own PIB/ Sub PIB or daisy chaining sub-PIBs within a single towfish or payload box.

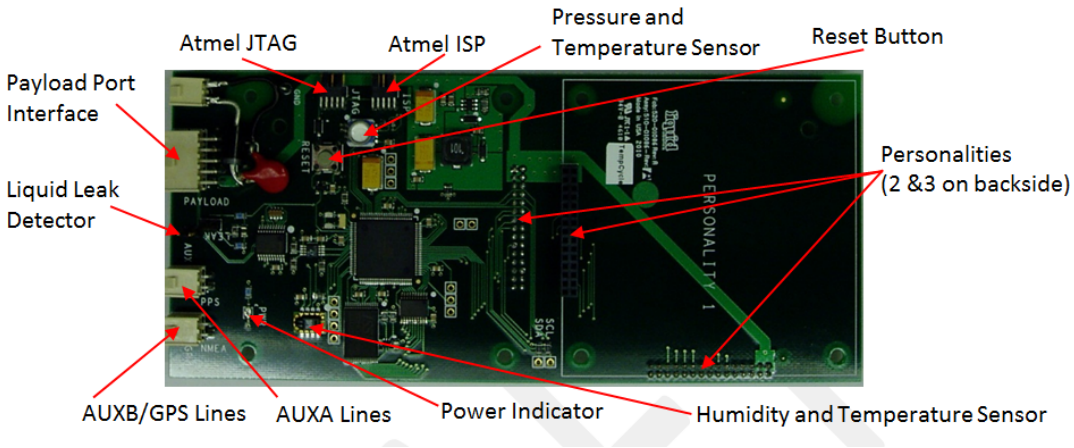
Figure 12.19. Single Sub PIB in a Towfish Connected to 3 Personality Modules and 3 Devices



Sub Payload Interface Board Interfaces

This section covers the various connectors and interfaces to the Sub Payload Interface Board (510-01086 and 510-01581). Each connection is described with a pin-out, part numbers for the mating connector/crimp pins and an electrical specification.

Figure 12.20. Sub Payload Interface Board 510-01086 Connector and Sensor Locations



Pressure and Temperature Sensor

The pressure and temperature sensor provides feedback to the user on the state of the dry box or towfish. The pressure is temperature compensated. It can be used as an early detection mechanism that the seal has failed by pressurizing the dry box or towfish and watching for a decrease over time. The PIB does not provide an alarm for a failure, only the data from the sensor. The user must monitor the data to detect the failure.

Maximum pressure on the sensor is 14 bars, with a resolution of 1.2 millibar and an accuracy of +/-200 millibar. WGMS reports this value as to the nearest kilo-Pascal

The temperature range is -40°C to 85 C with a resolution of 0.015 C. and an accuracy of -2/+6 C. WGMS reports this with a resolution of 1 C.

Reset Button

The reset button is provided for development, testing and debugging. It resets the micro controller when pressed.

Power Indicator

The power indicator is provided for development, testing and debugging. It notifies the user when the 3.3 volt regulator is functioning, which means a minimum of 6 volts is coming from J7 pin 2 to the regulator. It is not an indicator of the micro controller health.

Humidity and Temperature Sensor

The humidity sensor provides feedback to the user on the state of the dry box or towfish. The humidity is temperature compensated. It can be used as an early detection mechanism that the seal has failed by

watching for an increase in the dry box or towfish humidity over time. The PIB does not provide an alarm for a failure, only the data from the sensor. The user must monitor the data to detect the failure.

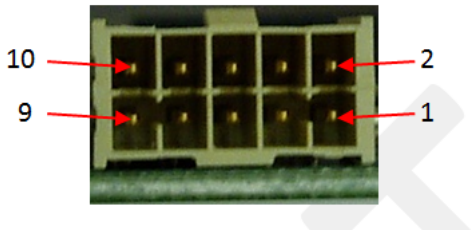
The humidity range is 0 to 100% with the best accuracy of +/-3% between 20 and 80% relative humidity. The sensor has a resolution of 0.05% relative humidity. The reading is temperature compensated. WGMS reports this with a resolution of 1% relative humidity.

The temperature range is -40°C to 85 C with a resolution of 0.4 C. and an accuracy of +/-2.5 C. WGMS reports this with a resolution of 1 C.

Payload Port Interface - J7

The payload port interface connector is intended to be the interface to a payload port, from either the Float C&C or the Rudder module. This is a latching 0.100” pitch connector from Samtec, part number IPL1-105-01-S-D-RA-K for robustness. The intention of this connection is to interface to either a penetrator or bulkhead connector that exits the payload dry box or towfish, but it can be used to daisy chain from another PIB or Sub PIB.

Figure 12.21. Payload Port Interface - J7 Pin Number Orientation



Payload Port Interface - J7 Pin Out

Pin #	Signal Name	Description	Wire Color
1	PPS	GPS pulse-per-second, using RS-232 levels	Black
2	PWR	13.2 Volts regulated at 3.0A continuous	White
3	TX	RS-232 Transmit Line	Red
4	RX	RS-232 Receive Line	Green
5	AUXA	Glider Connections, exact connections are determined by which payload port it is	Orange
6	AUXB	Glider Connections, exact connections are determined by which payload port it is	Blue
7	GND	Ground reference for all pins	Yellow
8	GPS Data	4800 Baud GGA, ZDA, and VTG, using RS-232	White/Black

Pin #	Signal Name	Description	Wire Color
		levels, 1 second report rate	
9	GND	Ground reference for all pins – Typically not used	
10	GND	Ground reference for all pins – Typically not used	

Payload Port Interface - J7 Electrical Interface

These connections are compatible with all the specification for the PAYLDx ports on the Float C&C dry box.

PWR and GND

The power pin requires 12 volts +/-10% at up to 3.0A continuous draw.

TX and RX

These are the communication lines that connect between the Float C&C and PIB processors. This link should be used by equipment to send data to other payloads and between the vehicle and shore. This connection is RS-232 at 115,200 baud rate. These signals are provided via a MAX232 compatible circuit.

PPS and GPS Signals

The PPS and GPS lines are not used by the PIB, but are provided via connector J9 for end user equipment use or specific personality modules that require this data. Please see the specification for J9 for more information.

Auxiliary Lines – AUXA & AUXB

The auxiliary lines are provided to give direct access to other payloads on the Float or to connect to the Glider assembly. These pins connect to J6 (AUX lines) pins 1 and 2 respectively.

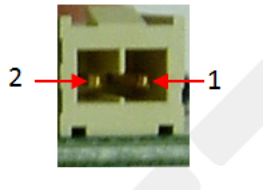
Payload Port Interface - J7 Mating Connector

There are two options for a mating connector from Samtec. The first and preferred option is to order pre-built cable assembly like MMSD-05-24-L-06.00-S-K. The length can be adjusted as needed by changing the 4-digit number between the L and S (i.e. the XX.XX in MMSD-05-24-L-XX.XX-S-K). The second option is to use a crimp housing and crimp pins, part numbers IPD1-05-D-K and CC79R/L.

Optional Power Input Port Interface - J10 (510-01581 only)

The optional power input port interface connector is intended to be the interface to a power supply external to the dry box or towfish. This is a latching 0.100” pitch connector from Samtec, part number IPL1-102-01-S-S-RA-K for robustness. The intention of this connection is to interface to either a penetrator or bulkhead connector that exits the payload dry box or towfish to provide power to the SUB-PIB and its personalities without the use of the Rudder module connection. An example of a use for this is to power an SMC (Liquid Robotics part number 510-01232) on to retrieve data from it without powering on the Float C&C.

Figure 12.22. Optional Power Port Interface - J10 Pin Number Orientation



Optional Power Input Port Interface – J10 Pin Out

Pin #	Signal Name	Description
1	GND	Ground reference for all pins
2	PWR	13.2 Volts regulated at 3.0A continuous

Optional Power Input Port Interface – J10 Electrical Interface

These connections are compatible with all the specification for the PAYLDx ports power connections on the Float C&C dry box.

PWR and GND

The power pin requires 12 volts +/-10% at up to 3.0A continuous draw.

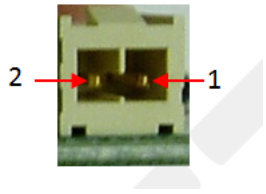
Optional Power Input Port Interface – J10 Mating Connector

There are two options for a mating connector from Samtec. The first and preferred option is to order pre-built cable assembly like MMSS-02-24-L-05.00-S-K. The length can be adjusted as needed by changing the 4-digit number between the L and S (i.e. the XX.XX in MMSS-02-24-L-XX.XX-S-K). The second option is to use a crimp housing and crimp pins, part numbers IPD1-02-S-K and CC79R/L.

AUX Lines - J6

The AUX lines connector is intended to simplify cabling from the auxiliary lines to a personality module. This is a keyed and latching connector for robustness. The connector is manufactured by Samtec and is part number IPL1-102-01-L-S-RA-K.

Figure 12.23. Pin Number Orientation



AUX Lines – J6 Pin Out

Pin #	Signal Name	Description
1	AUXA	Connects to Payload Port Interface pin 5

Pin #	Signal Name	Description
2	AUXB	Connects to Payload Port Interface pin 6

AUX Lines – J6 Electrical Interface

These connections are direct shorts back to the Payload Port Interface. It is recommended that the voltage between the two pins not exceed 50 volts or 2.5 amps.

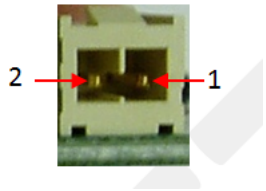
AUX Lines – J6 Mating Connector

There are two options for a mating connector from Samtec. The first and preferred option is to order pre-built cable assembly like MMSS-02-24-L-05.00-S-K. The length can be adjusted as needed by changing the 4-digit number between the L and S (i.e. the XX.XX in MMSS-02-24-L-XX.XX-S-K). The second option is to use a crimp housing and crimp pins, part numbers IPD1-02-S-K and CC79R/L.

GPS and Secondary AUX Lines - J9

The GPS lines connector is intended to simplify cabling from the GPS and PPS signals to a personality module. In a towfish configuration, GPS and PPS are not present on this connector; instead, it is a secondary set of AUX lines that connects to the PAYLD2 connector. This is a keyed and latching connector for robustness. The connector is manufactured by Samtec and is part number IPL1-102-01-L-S-RA-K.

Figure 12.24. GPS/Secondary AUX Lines – J9 Pin Number Orientation



GPS Lines – J9 Pin Out

Pin #	Signal Name	Description
1	PPS	Connects to Payload Port Interface pin 1
2	NMEA	Connects to Payload Port Interface pin 8

Secondary AUX Lines – J9 Pin Out

Pin #	Signal Name	Description
1	AUXC	Connects to Payload Port 2 pin 5
2	AUXD	Connects to Payload Port 2 pin 6

GPS Lines and Secondary AUX – J9 Electrical Interface

These connections are direct shorts back to the Payload Port Interface. These signals are both RS-232 compatible outputs.

As Secondary AUX lines these are just bare wires that connect to the PAYLD2 port via the Rudder Module and Umbilical.

GPS Lines and Secondary AUX – J9 Mating Connector

There are two options for a mating connector from Samtec. The first and preferred option is to order pre-built cable assembly like MMSS-02-24-L-05.00-S-K. The length can be adjusted as needed by changing the 4-digit number between the L and S (i.e. the XX.XX in MMSS-02-24-L-XX.XX-S-K). The second option is to use a crimp housing and crimp pins, part numbers IPD1-02-S-K and CC79R/L.

Atmel ISP - J5

The Atmel ISP connector is intended to be the production programming interface port. This dual row 0.050” pitch header is shrouded with a key for reliable orientation and quick connection/disconnection in a production environment. This connection is for an Atmel ATAVRISP2 via a Liquid Robotics programming adapter, part number 510-00537, but will work with other ATMEL programmers via the same adapter. The connector itself is manufactured by Samtec and is part number FTSH-105-01-F-D-RA-K.

Figure 12.25. ISP and JTAG Connector Location

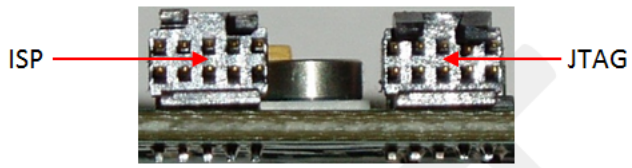
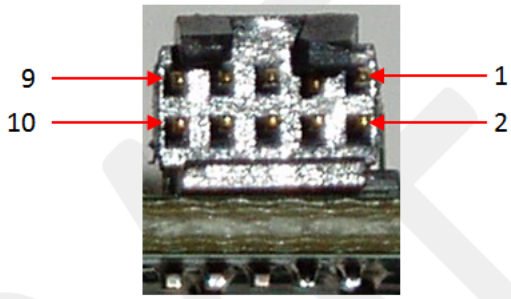


Figure 12.26. ISP Pin Number Orientation



Atmel ISP – J5 Pin Out

Pin #	Signal Name	Description
1	VCC	+3.3 VDC
2	NC	No Connection
3	GND	Ground reference for all pins
4	SCLK	Clock for synchronizing data
5	GND	Ground reference for all pins

Pin #	Signal Name	Description
6	MISO-ISP	Master In – Slave Out
7	NC	No Connection
8	MOSI	Master Out – Slave In
9	GND	Ground reference for all pins
10	RESET	Pin 10 Grounded → Reset

Atmel ISP – J5 Electrical Interface

These connections are compatible with the Atmel ATmega ISP programming interface running at 3.3 volts.

PWR and GND

The power pin is used to provide a reference voltage to the programming adapter. No more than 30mA should be sourced from this pin during programming.

MISO-ISP, MOSI, SCLK and RESET

These are the communication lines that connect between the programmer and the micro controller. This link should only be used with an Atmel approved programming device.

Atmel ISP – J5 Mating Connector

The mating connector is a cable assembly manufactured by Samtec, part number FFSD-05-D-06.00-01-N. This cable then connects to the Liquid Robotics programming adapter, part number 510-00537 and then either to an Atmel ATJTAGICE2 with a 10 pin to 6 pin adapter cable or an Atmel ATAVRISP2. Both of these setup options need to be connected into the ISP connector port on the Liquid Robotics programming adapter. Other Atmel programmers should work as well, but only these two have been tested over this interface.

Figure 12.27. Liquid Robotics Programming Adapter, Part Number 510-00537

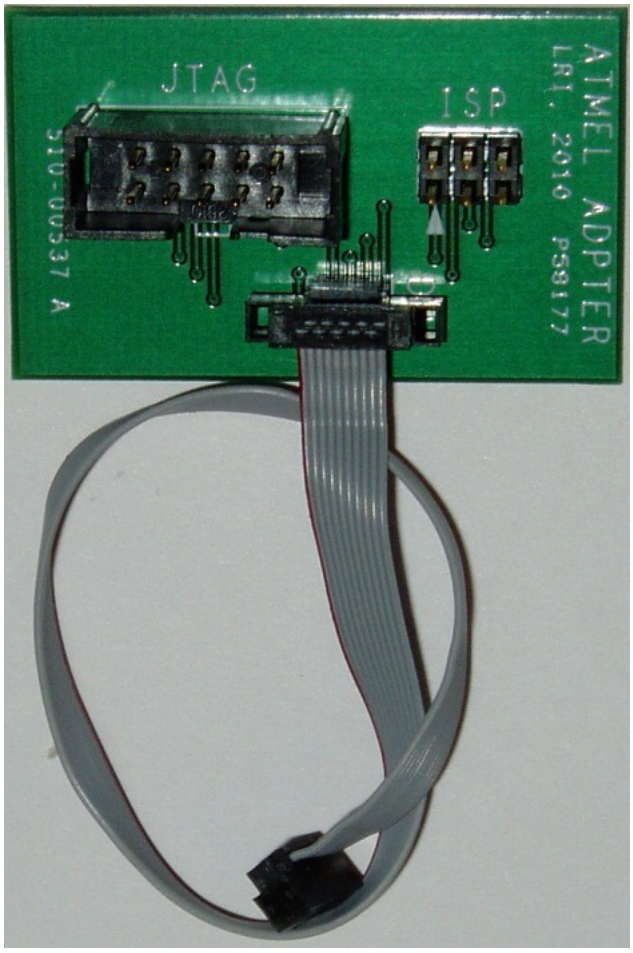


Figure 12.28. Atmel ATJTAGICE2 without the Programming Adapter or the 10 Pin to 6 Pin Adapter

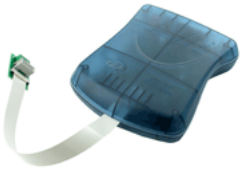


Figure 12.29. Atmel ATAVRISP2 without the Programming Adapter



Atmel JTAG - J4

The JTAG connector is intended to be the development programming and debug port. This dual row 0.050" pitch header is shrouded with a key for reliable orientation in a development environment. This connection is for an Atmel ATJTAGICE2 via a Liquid Robotics programming adapter, part number 510-00537, but will work with other ATMEL programmers via the same adapter. The connector itself is manufactured by Samtec and is part number FTSH-105-01-F-D-RA-K.

Figure 12.30. ISP and JTAG Connector Location

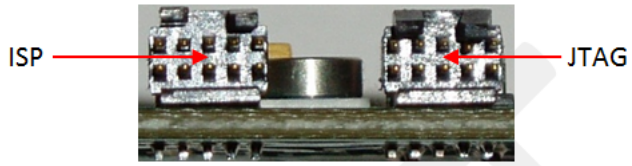
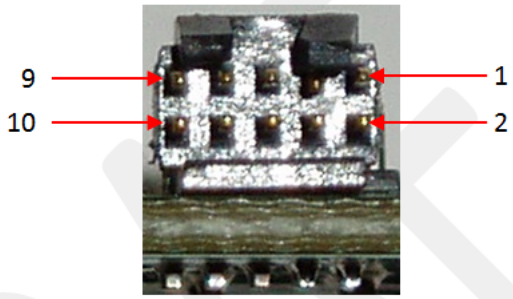


Figure 12.31. JTAG Pin Number Orientation



Atmel JTAG – J4 Pin Out

Pin #	Signal Name	Description
1	VCC	+3.3 VDC
2	TMS	Test Mode Select
3	GND	Ground reference for all pins
4	TCK	Clock
5	GND	Ground reference for all pins
6	TDO	Data Out
7	NC	No connection
8	TDI	Data In
9	GND	Ground reference for all pins
10	RESET	Pin 10 Grounded → Reset

Atmel JTAG – J4 Electrical Interface

These connections are compatible with the Atmel ATmega ISP programming interface running at 3.3 volts.

PWR and GND

The power pin is used to provide a reference voltage to the programming adapter. No more than 30mA should be sourced from this pin during programming.

TCK, TDO, TMS, TDI and RESET

These are the communication lines that connect between the programmer and the micro controller. This link should only be used with an Atmel approved programming device.

Atmel JTAG – J4 Mating Connector

The mating connector is a cable assembly manufactured by Samtec, part number FFSD-05-D-06.00-01-N. The setup is intended to connect to the Liquid Robotics programming adapter, part number 510-00537 and then to an Atmel ATJTAGICE2 via the JTAG port. Other Atmel programmers should work as well, but this is the only one that has been tested over this interface.

Figure 12.32. Liquid Robotics Programming Adapter, Part Number 510-00537

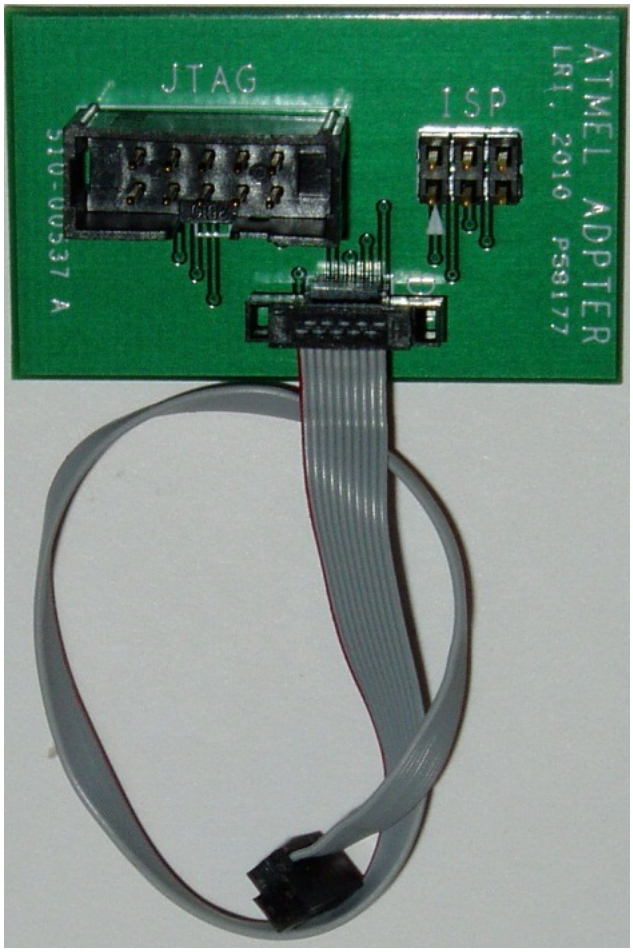
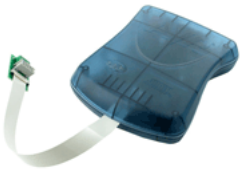


Figure 12.33. Atmel ATJTAGICE2 without the Programming Adapter



Personality Interface - J1, J2, J3

The personality interface connector is intended to be the interface to a variety of personality modules. This interface was designed around flexibility and the ability to identify the specific location. This is a 2mm, dual row, 30 pin header. The connectors are not keyed or shrouded. Personality 1 is on the top side of the board, while personalities 2 and 3 are on the bottom side of the assembly. Because Personality 3 is directly behind the PIB electronics, please be aware of the personality module stack height, verifying that none of the module components touch the through-hole component leads. **Warning incorrect installation of Personality Modules will result in permanent damage to the PIB.** When installing a personality please remove the connector at J7 to guarantee that there is no power present. The connector is manufactured by Samtec and is part number SQT-115-01-L-D.

Figure 12.34. Top Side Personality Connector Locations

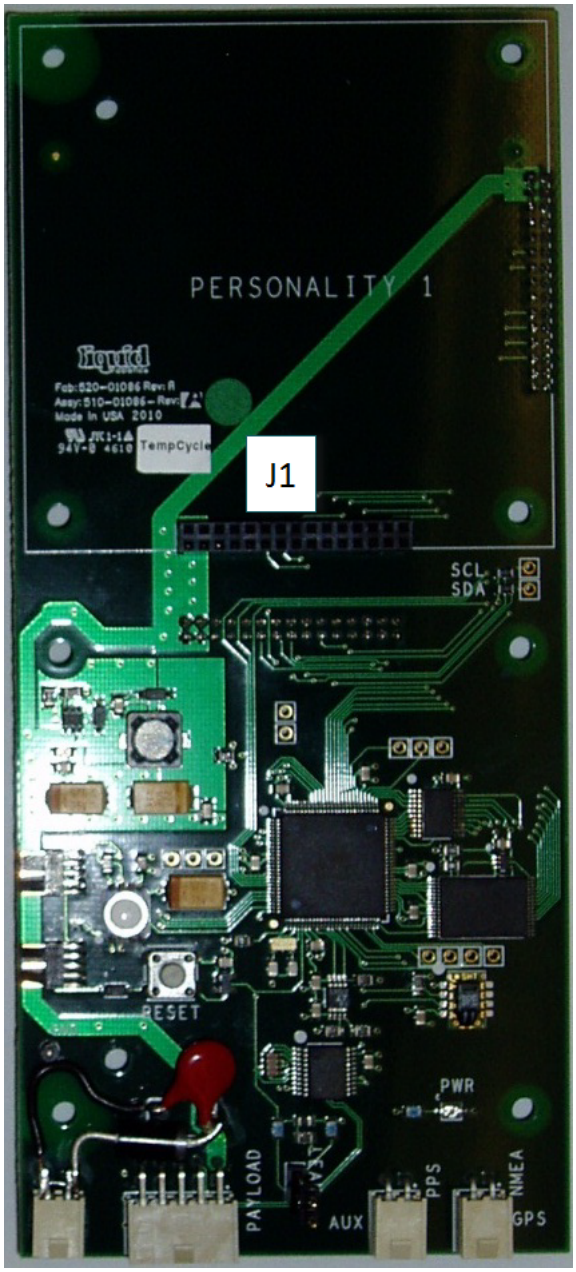


Figure 12.35. Bottom Side Personality Connector Locations



Figure 12.36. Pin Number Orientation



Personality Interface – J1, J2, J3 Pin Out

Pin #	Signal Name	Description
1	PWR	Power from C&C

Pin #	Signal Name	Description
2	GND	Ground
3	PWR	Power from C&C
4	GND	Ground
5	VCC	Regulated +3.3V
6	GND	Ground
7	SDA	I²C Data Line
8	GND	Ground
9	SCL	I²C Clock Line
10	GND	Ground
11	TXD	UART Transmit to Personality Module
12	GND	Ground
13	RXD	UART Receive from Personality Module
14	GND	Ground
15	XCK GPIO	Synchronous Serial Clock GPIO (optional)
16	GND	Ground
17	PON	Personality Module Power-On/ Enable
18	GND	Ground
19	POK	Personality Module Power OK Response
20	GND	Ground
21	GPIO-0	Connection to micro controller
22	GND	Ground
23	GPIO-1	Connection to micro controller
24	GND	Ground
25	GPIO-2	Connection to micro controller
26	GND	Ground
27	GPIO-3	Connection to micro controller
28	GND	Ground
29	ADDR2	Personality Module address 2nd bit
30	ADDR1	Personality Module address 1st bit

Personality Interface – J1, J2, J3 Electrical Interface

These connections are 3.3 volt CMOS connections directly to the ATmega2560 on the PIB.

PWR

The power pins are a direct connection to J7 pin 2. They should be limited to no more than 1.0 amps per personality module.

VCC

The regulated 3.3 volt supply can provide up to 250 milliamps for the personality modules. This is a sum of all three ports, not a per port limitation. If more current is drawn by the personality modules the PIB processor can reset.

GND

The ground pins are a direct connection to J7 pin 7 through an internal plane on the PIB. These should all be connected together on the personality module for robustness.

SCL and SDA

These communication lines are common to all of the personality modules. They conform to the I2C specification for a 3.3 volt bus at up to 400 KHz. The intention of this bus is to allow inter-personality communication and/or a common interface for automatically detecting personality modules. I2C is available to the personality module. It is also intended to be connected to an I2C serial EEPROM (Atmel AT24CxxB [except the AT24C08B]) on each personality module, but could be used to control the personality as well. I2C addresses 7'b1010_xxx should be reserved for EEPROM storage on personality modules. Pull-ups are provided on the PIB.

TX, RX and XCK

These communication lines provide for an interface to a USART on the micro controller. If the USART is used in asynchronous mode, the XCK line may be used as a general purpose I/O pin from the micro controller.

Connector	**USART**
J7 (for reference only)	USART0
J1/Personality 1	USART1
J2/Personality 2	USART2
J3/Personality 3	USART3

PON and POK

The PON and POK signals are used to provide an enable/power on pin and feedback that the system is on and running. If the PON pin will not be used, tie it directly to POK so that it is compatible with all other personality modules. These pins are 3.3 volt CMOS compatible. PON should be brought high and sometime later POK should come high to indicate that the personality module is functional.

GPIO-X

The GPIO lines are provided to give direct access to I/O on the micro controller. Refer to the ATmega2560 data sheet for specific pin requirements. The table below shows the specific connections made from each personality module to the micro controller.

Personality Module Interface	J1	J2	J3
21/GPIO-0	PK0	PH4	PF0
23/GPIO-1	PK1	PH5	PF1
25/GPIO-2	PK2	PH6	PF2
27/GPIO-3	PK3	PH7	PF3

ADDR1 and ADDR2

The address 1 and address 2 pins are used to identify which specific personality location a personality has been connected to. These pins should be tied to the I2C EEPROM, but can also be read by the personality module to determine location. The table below shows the specific addressing of each location on the PIB.

Location	ADDR1/Pin 30	ADDR2/Pin 29
J1	GND	GND
J2	GND	VCC
J3	VCC	GND

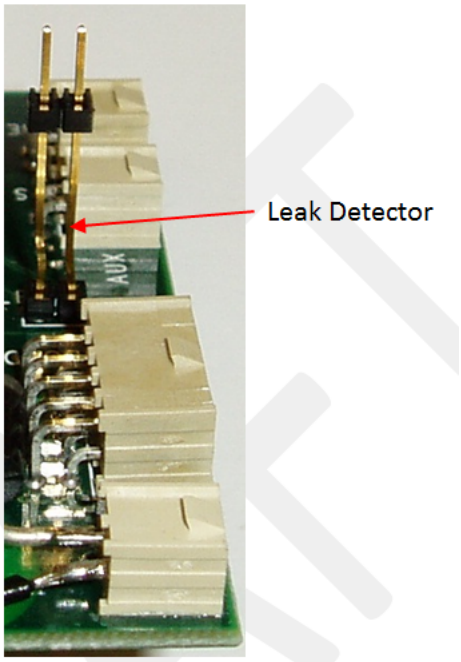
Personality Interface – J1, J2, J3 Mating Connector

The mating connector is manufactured by Samtec, and is part number TMM-115-03-G-D.

Liquid Leak Detector - J8

The liquid leak detector connector is intended to sit in an orientation such that the leads protrude to the lowest position in the payload box or towfish. The header is manufactured by Samtec and is part number DW-02-20-G-S-750. If necessary this header can be replaced by a connector to cable the leak detector leads to a lower location. The height of the leads is designed around installing the Sub-PIB into a 3” diameter towfish assembly.

Figure 12.37. Leak Detector Location



Liquid Leak Detector – J8 Pin Out

Pin #	Signal Name	Description
1	LEAK0	Connection to weak pull down indicates a leak
2	WEAK PULL DOWN	Grounded thru 1K

Liquid Leak Detector – J8 Electrical Interface

These connections are 3.3 volt CMOS connections with ESD protection. The SUB-PIB leak detector looks like a PIB leak detector to the firmware with LEAKCO connected directly to LEAKCI pin of the micro controller and LEAK1 always pulled high as if there is no leak.

LEAKCO and LEAKCI

The leak cable continuity out and in pins are used to detect the presence of the leak detector cable. These pins are not present on this connector, but are connected together on the micro controller.

LEAK0, LEAK1 and WEAK PULL DOWN

The LeakX pins have weak internal pull-ups and the micro controller watches for these pins to be pulled low indicating a short to the weak pull down pins. These pins are pulled up to 3.3 volts via 47 kilo-ohms resistors. The LEAK1 pin does not connect to the leak detector header, but is present for compatibility with the PIB firmware.

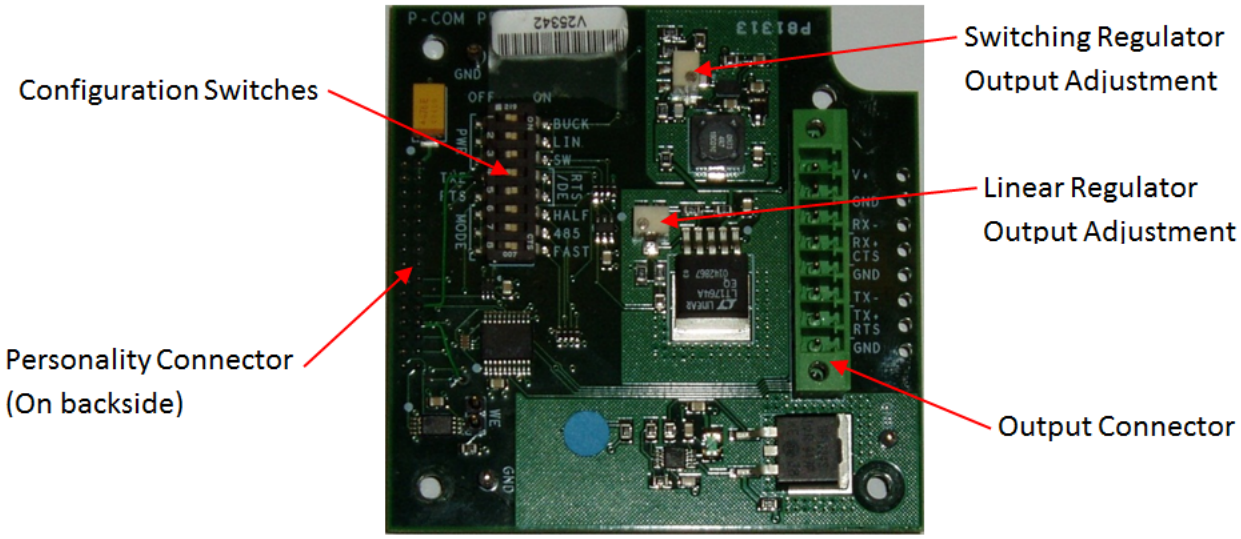
Liquid Leak Detector – J8 Mating Connector

There is no mating connector for the Sub PIB leak detector.

P-COM Personality Boards

The P-COM personality provides both a switchable power point and a serial communications end point. The power point can be direct power from the Float C&C port, a linear regulator for low noise, or a switching regulator for higher efficiency at lower voltages. The serial communications path supports RS-232 with RTS/CTS flow control as needed, RS-485, or RS-422. There are two released versions of this design, revision A and revision C. This document only covers revision C. If you have a revision A PCBA, contact Liquid Robotics Inc. for specific instructions on your setup.

Figure 12.38. P-COM Personality Board 510-00097 Rev C



Personality Interface - J1

The personality interface connector is the interface to a PIB in one of the three personality slots. Not all of the PIB pins are used. It is a 2mm, dual row, 30 pin header. The connectors are not keyed or shrouded. Warning incorrect installation of Personality Modules will result in permanent damage to the PIB. When installing a personality please remove power from the PIB. The connector is manufactured by Samtec and is part number TMM-115-03-G-D.

Figure 12.39. J1 As Viewed From The Bottom



Personality Interface – J1 Pin Out

Pin #	Signal Name	Description
1	PWR	Power from C&C

Pin #	Signal Name	Description
2	GND	Ground
3	PWR	Power from C&C
4	GND	Ground
5	VCC	Regulated +3.3V Input
6	GND	Ground
7	SDA	I2C Data Line to EEPROM
8	GND	Ground
9	SCL	I2C Clock Line to EEPROM
10	GND	Ground
11	TXD	Data from the PIB to the end user equipment
12	GND	Ground
13	RXD	Data to the PIB from the end user equipment
14	GND	Ground
15	NC	Unused
16	GND	Ground
17	PON	Power-On/Enable for switch selected output power
18	GND	Ground
19	POK	Power OK Response for switch selected output power
20	GND	Ground
21	RXEn	Used to disable the receiver when operating in half duplex mode
22	GND	Ground
23	TXE	Used to enable the receiver when operating in half duplex mode
24	GND	Ground
25	RTS	Ready to Send output from the PIB
26	GND	Ground
27	CTS	Clear to send input to the PIB
28	GND	Ground
29	ADDR2	EEPROM address line A2
30	ADDR1	EEPROM address line A1

Personality Interface – J1 Electrical Interface

These connections are 3.3 volt CMOS compatible.

PWR

The power pins connect to the inputs of the two regulators and the power switch. They are limited to no more than 1.0 amps per regulator input.

VCC

The regulated 3.3 volt provides up to 50 milliamps for the personality module.

GND

The ground pins are a common ground point for the entire personality module through an internal plane.

SCL and SDA

These communication lines connect only to the EEPROM. They conform to the I2C specification for a 3.3 volt bus at up to 400 KHz. The intention of this bus is to allow for automatically detecting the personality module. The EEPROM is currently not programmed at the factory.

TX and RX

These communication lines provide a path for data to/from the PIB. The transceiver does the appropriate physical layer translations based on the SW1 settings.

PON and POK

The PON and POK signals provide an enable/power on pin and feedback that the system is on. When PON goes above the 3.3 volt CMOS threshold, the output that is selected by the switch setting is turned on. Once the output stabilizes, the POK pin is pulled high to indicate that the power is stable. If the output goes below its regulation point POK will go low as well to indicate a brown out condition.

RXEn and TXE

These pins are used to control the direction of the transceiver. They also control whether data is echoed back when configured as either RS-485 or RS-422. For RS-485 mode with no echo RXEn and TXE should always be the same level, with low being receive mode and high being transmit mode. For RS-422 mode RXEn must be low and TXE must be high to enable both the driver and receiver at the same time.

RXEn	**TXE**	**Description**
Low	Low	Receiver enabled, Transmitter disabled – used for listening a on RS-485 bus
High	Low	Both the receiver and transmitter are disabled – good for low power
Low	High	Both the receiver and transmitter are enabled – used for RS-422 bus
High	High	Receiver disabled, Transmitter enabled – used for talking a on RS-485 bus

RTS and CTS

The RTS and CTS pins can handle a device that uses hardware flow control. RTS is an output stating that the PIB is ready to send data. CTS is an input to the PIB that should trigger when the data is actually sent

from the PIB. These pins are not required to be in the correct state for the communications channel to work, unless the end user equipment requires it.

RTS	CTS	Description
Low	Low	PIB not ready to send, end user equipment not ready to receive – PIB has no data to send but is gated
High	Low	PIB ready to send, , end user equipment not ready to receive – PIB has data to send but is gated
Low	High	PIB not ready to send, end user equipment ready to receive – This is an uncommon condition
High	High	PIB ready to send, end user equipment ready to receive – Send the data now

ADDR1 and ADDR2

The address 1 and address 2 pins identify to which specific personality location the personality has been connected. These pins are tied to the I²C EEPROM only.

Personality Interface - J1 Mating Connector

The mating connector is manufactured by Samtec, and is part number SQT-115-01-L-D.

Output Connector - J4

The output connector is the interface to the end user equipment. This is a disconnect able screw terminal to allow for easy removal of the end user equipment. The connector is manufactured by Tyco and is part number 284516-8.

Figure 12.40. Output Connector - J4 Pin Number Orientation



Output Connector – J4 Pin Out>

Pin #	Signal Name	Description
1	V+	Power Output based on switch selection and potentiometer settings
2	GND	Ground
3	RX-	RS-232 Receive or RS-422 Receive –

Pin #	Signal Name	Description
4	RX+/CTS	RS-232 RTS or RS-422 Receive +
5	GND	Ground
6	TX+/RTS	RS-232 RTS or RS-485+/RS-422 Transmit +
7	TX-	RS-232 Transmit or RS-485-/ RS-422 Transmit -
8	GND	Ground

Figure 12.41. RS-232 based device connection diagram

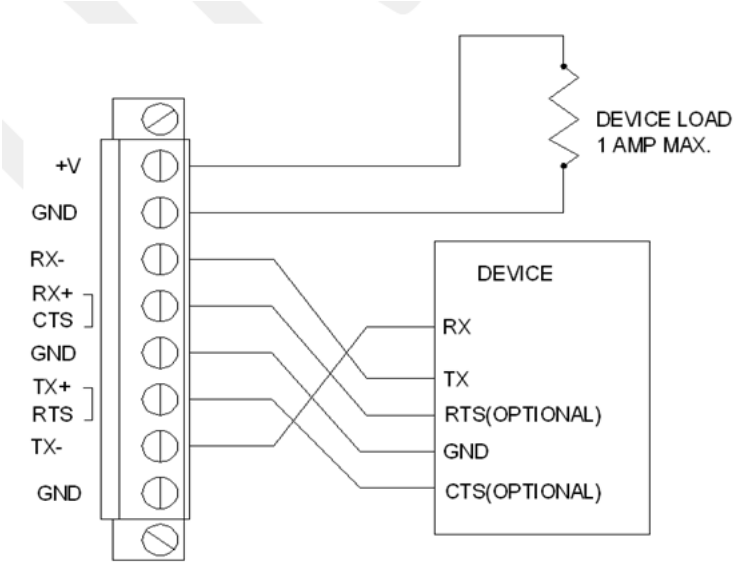
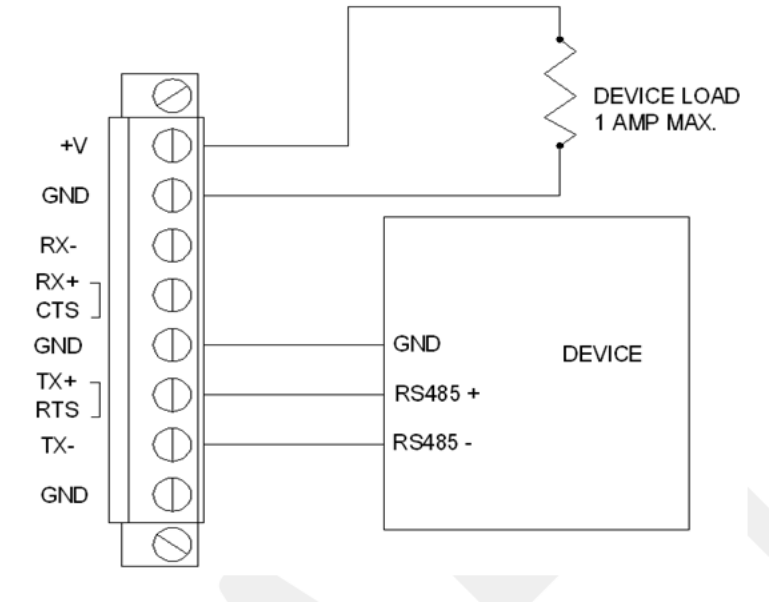


Figure 12.42. RS-485 half duplex based device connection diagram



Output Connector – J4 Electrical Interface

The communication connections are electrically compatible with the switch-selected protocol being driven by a MAX3160 transceiver. The power connections support the selected power output at up to 1.0 amps.

PWR and GND

The power output voltage is based on the settings of the switch and the regulator potentiometers. This pin is limited to 1.0 amp out, independent of the switch settings.

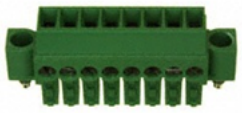
RX-, RX+/CTS, TX-, and TX+/RTS

These communication lines connect to the end user equipment. The function of the pin is determined by the switch settings. These pins are ESD protected. This path can be slew rate limited if needed for low EMI applications. These signals are provided via a MAX3160 compatible circuit.

Output Connector – J4 Mating Connector

The mating connector is manufactured by Tyco, part number 284510-8 or a connector manufactured by Phoenix contact, part number MCVR 1.5/8-STF-3.5. Below is an example photo. It is recommended that ferrules be used with the loose wires as well.

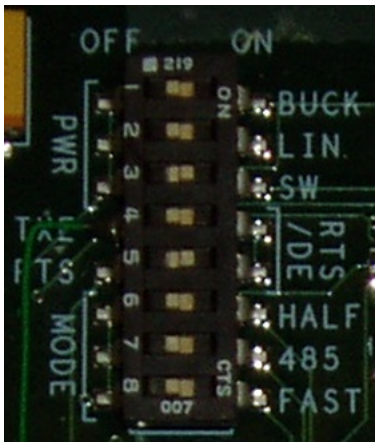
Figure 12.43. Output Example Connector



Configuration Switch - SW1

The configuration switch is used to select the power output type and the communication protocol type. There are three sections of selections on the switch. The first three switches are used to select which type of power output is used. The next two switches select whether the TXE or the RTS signal is passed through to the transceiver. The last three switches set the operating parameters of the transceiver. Switch 3 and switch 5 are on by default.

Figure 12.44. Configuration Switch SW1 With All Switches In The OFF Position



Switch Descriptions – SW1

Switch #	Signal Name	Description	Default Position
1	PWR – BUCK	Enables the buck switching regulator as the output enabled by PON – default output voltage of 5.0 volts	OFF
2	PWR – LIN	Enables the LDO linear regulator as the output enabled by PON – default output voltage of 12.0 volts	OFF
3	PWR – SW	Enables the switched output voltage from the Float C&C as the output enabled by PON – typical voltage of 13.2 volts	ON
4	TXE – RTS/DE	Enables the use of TXE for RS-422/RS-485 protocols	OFF
5	RTS – RTS/DE	Enables the use of RTS for RS-232 protocols	ON
6	MODE – HALF	Enables half duplex mode for the RS-485 protocol	OFF
7	MODE – 485	Enables the use of RS-422/RS-485 protocols instead of RS-232	OFF
8	MODE – FAST	Enables the use of data rates faster than 250 kbps	OFF

Switch Selection Examples – SW1

Examples of various switch selections. The power and communications switches can be set independent of each other.

The three tables below show the possible power output options. It is not recommended to enable more than one power output at a time as the POK feedback will only indicate when the lowest voltage has gone out of regulation. For more information on each of the selectable outputs, see the following three sections. The information here is only how to select each, not how to configure them.

To configure the output to use the linear regulator, turn SW1.2 on:

Switch #	Signal Name	Description	Default Position
1	PWR – BUCK	Enables the buck switching regulator as the output enabled by PON – default output voltage of 5.0 volts	ON

Switch #	Signal Name	Description	Default Position
2	PWR – LIN	Enables the LDO linear regulator as the output enabled by PON – default output voltage of 12.0 volts	OFF
3	PWR – SW	Enables the switched output voltage from the Float C&C as the output enabled by PON – typical voltage of 13.2 volts	OFF

To configure the output to use the switching regulator, turn SW1.1 on:

Switch #	Signal Name	Description	Default Position
1	PWR – BUCK	Enables the buck switching regulator as the output enabled by PON – default output voltage of 5.0 volts	ON
2	PWR – LIN	Enables the LDO linear regulator as the output enabled by PON – default output voltage of 12.0 volts	OFF
3	PWR – SW	Enables the switched output voltage from the Float C&C as the output enabled by PON – typical voltage of 13.2 volts	OFF

To configure the output to use the Float C&C voltage directly, turn SW1.3 on:

Switch #	Signal Name	Description	Default Position
1	PWR – BUCK	Enables the buck switching regulator as the output enabled by PON – default output voltage of 5.0 volts	OFF
2	PWR – LIN	Enables the LDO linear regulator as the output enabled by PON – default output voltage of 12.0 volts	OFF
3	PWR – SW	Enables the switched output voltage from the Float C&C as the output enabled by PON – typical voltage of 13.2 volts	ON

The three tables below show the most common communications configuration options.

To configure the output for RS-232 at 115,200 kbps with or without flow control make sure only SW1.5 is on:

Switch #	Signal Name	Description	Default Position
4	TXE – RTS/DE	Enables the use of TXE for RS-422/RS-485 protocols	OFF
5	RTS – RTS/DE	Enables the use of RTS for RS-232 protocols	ON
6	MODE – HALF	Enables half duplex mode for the RS-485 protocol	OFF
7	MODE – 485	Enables the use of RS-422/RS-485 protocols instead of RS-232	OFF
8	MODE – FAST	Enables the use of data rates faster than 250 kbps	OFF

Configuration for RS-485 half duplex communication protocol:

Switch #	Signal Name	Description	Default Position
4	TXE – RTS/DE	Enables the use of TXE for RS-422/RS-485 protocols	ON
5	RTS – RTS/DE	Enables the use of RTS for RS-232 protocols	OFF
6	MODE – HALF	Enables half duplex mode for the RS-485 protocol	ON
7	MODE – 485	Enables the use of RS-422/RS-485 protocols instead of RS-232	ON
8	MODE – FAST	Enables the use of data rates faster than 250 kbps	OFF

Configuration for RS-422 communication protocol:

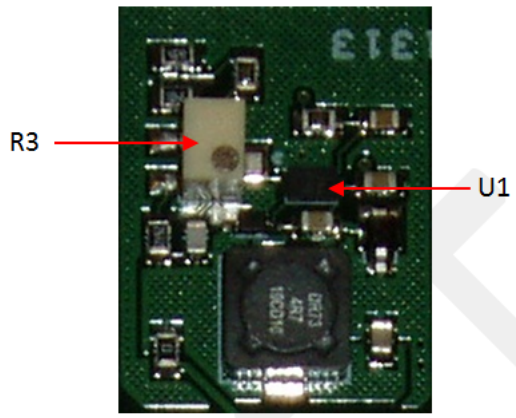
Switch #	Signal Name	Description	Default Position
4	TXE – RTS/DE	Enables the use of TXE for RS-422/RS-485 protocols	ON
5	RTS – RTS/DE	Enables the use of RTS for RS-232 protocols	OFF
6	MODE – HALF	Enables half duplex mode for the RS-485 protocol	OFF

Switch #	Signal Name	Description	Default Position
7	MODE – 485	Enables the use of RS-422/RS-485 protocols instead of RS-232	ON
8	MODE – FAST	Enables the use of data rates faster than 250 kbps	OFF

Switching Regulator - U1

The switching regulator is used when lower voltages are required at the output of the P-COM. The switching regulator is set to 5.0 volts out, but has an adjustable range of 3.3 to 10.0 volts. Adjust the voltage adjusted by changing the value of potentiometer R3.

Figure 12.45. Switching Regulator and Potentiometer



Switching Regulator Circuit Description – U1

The switching regulator design is based on Linear Technologies LT3503 data sheet and design example. The regulator design has been optimized around a 1.0 amp output current over the entire voltage range. The limiting factor that drove the design to a limited output current range is the heat dissipation from the inductor. The POK signal indicates when the feedback pin is above the reference voltage, but it does not indicate that the regulator has gone over voltage.

To select the switching regulator as the output power source, make sure that SW1.1 is on and both SW1.2 and SW1.3 are off.

Switching Regulator Adjustment – R3

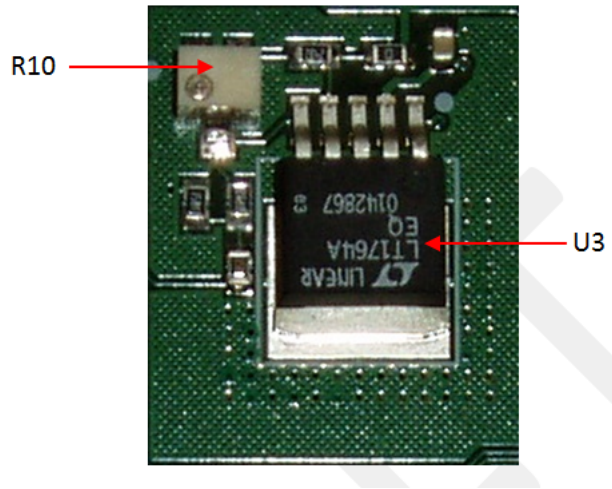
The regulator output is adjustable via the potentiometer R3. The range of the regulator is 3.3 to 10.0 volts. Turning the adjustment screw clockwise will decrease the voltage. Turning the adjustment screw counterclockwise will increase the voltage.

To adjust the voltage disconnect the end user equipment and turn on the P-COM. Measure the voltage across pins 1 and 2 of the output connector with a digital multi-meter. Adjust R3 until the desired voltage is reach. Turn off the P-COM and reconnect the end user equipment. Turn the P-COM on and recheck the voltage under load.

Linear Regulator - U3

Use the linear regulator when a low noise power source is required at the output of the P-COM. The linear regulator is set to 12.0 volts out, but has an adjustable range of 10.0 to 13.0 volts. Adjust the voltage by changing the value of potentiometer R10.

Figure 12.46. Linear Regulator and Potentiometer



Linear Regulator Circuit Description – U3

The linear regulator design is based on Linear Technologies LT1764 data sheet and design example. The regulator design has been optimized around a 1.0 amp output current over the entire voltage range. The limiting factor is the heat dissipation from the regulator itself. The POK signal indicates when the feedback pin is above the reference voltage, but it does not indicate that the regulator has not gone over voltage.

To select the linear regulator as the output power source, ensure that SW1.2 is on and both SW1.1 and SW1.3 are off.

Linear Regulator Adjustment – R10

The regulator output is adjustable via the potentiometer R10. The range of the regulator is 10.0 to 13.0 volts. Turning the adjustment screw clockwise will decrease the voltage. Turning the adjustment screw counterclockwise will increase the voltage.

To adjust the voltage disconnect the end user equipment and turn on the P-COM. Measure the voltage across pins 1 and 2 of the output connector with a digital multi-meter. Adjust R3 until the desired voltage is reach. Turn off the P-COM and reconnect the end user equipment. Turn the P-COM on and recheck the voltage under load.

Chapter 13. XBee Communications

This chapter provides information on using XBee communications to interact with Wave Gliders.

XBee Command Menu

The XBee console interface provides a method of interacting with the Wave Glider without using WGMS. XBee is primarily used in development, testing, verifying configuration, and debugging in a lab environment. In some cases it may also be used as part of the pre-launch check.

The XBee console window is available in the XBee Relay. It does not require an internet connection to WGMS in order to function. To launch the XBee console window:

1. Open up the XBee relay.

2. Select the vehicle you want to interact with from the list of *Connected Devices* (each vehicle is listed by its unique ID).

3. Click on the *Launch Debug Console for vehicle* button - the XBee console window will appear.

Figure 13.1. Launching the XBee Console from the XBee Relay

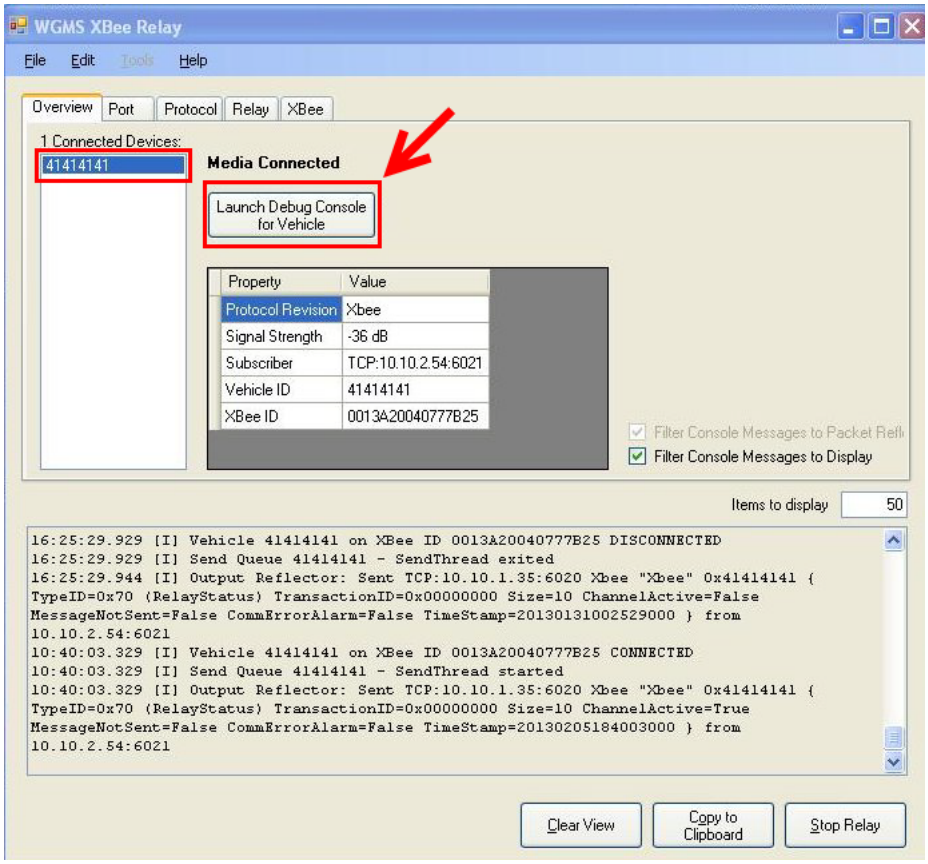
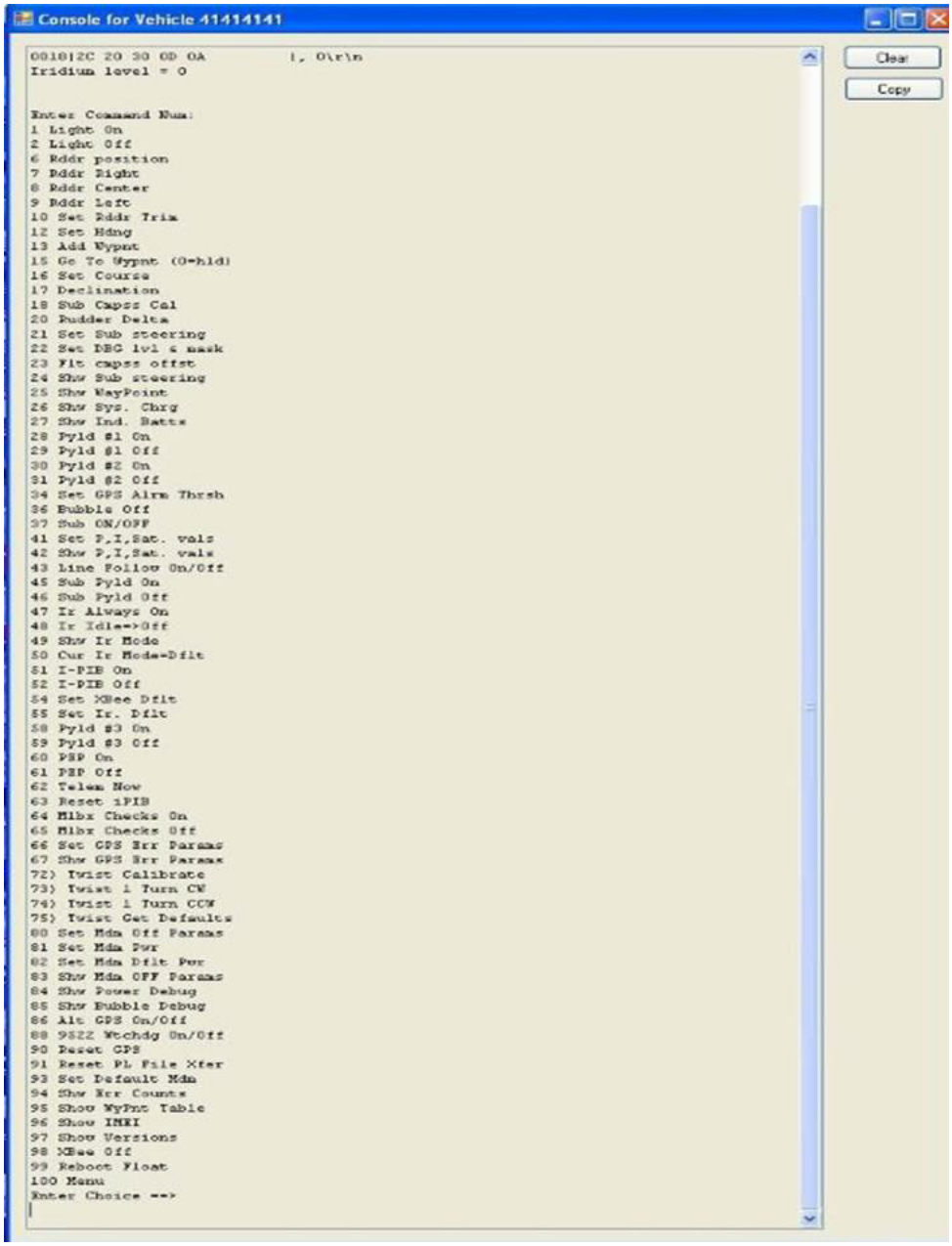


Figure 13.2. XBee Console Window for Vehicle 41414141



The XBee console functions like most serial terminals. Data from the vehicle is displayed and users can type in commands and data to send to the vehicle. There is only one console window per vehicle, but multiple vehicles can have console windows open simultaneously.

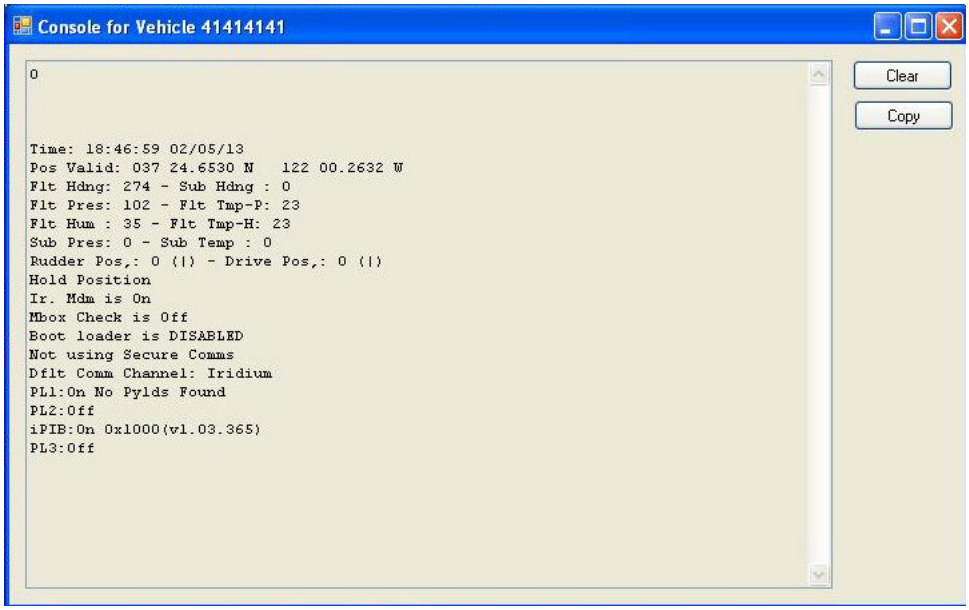
Each command has a number associated with it. When you enter a command number the command will be processed, (for example, a **Rudder Center** command), or you will be prompted for additional data (for example, when you enter a new way point).

The list of available commands varies, depending on the version of firmware running on the vehicle C&C dry box. Only the commands that are supported with the firmware version running on the vehicle C&C dry box will be displayed in the XBee console window.

There are 2 commands that are supported with all firmware versions:

- Command 100 will display a list of all available commands.
- Command 0 will display the current status of the system.

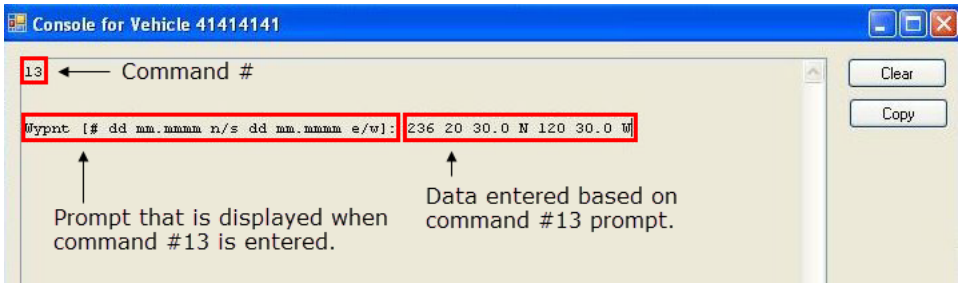
Figure 13.3. Example: Command 0 System Status Output for Vehicle 41414141



The specific data that is returned when command 0 is entered varies, depending on the firmware version that is running on the vehicle C&C drybox.

Most of the XBee commands are simple and self-explanatory. They will either require no data, for example entering a **Rudder Center** command, or a prompt will appear indicating what data is needed and how to format that data, for example, entering an **Add Waypoint** command. When the command is processed on the vehicle, there is no acknowledgement in the console window unless the processing of the command requires a response. For example, **Rudder Center** will not generate a text response, but **Show Versions** will, because displaying firmware versions is the result of the command.

Figure 13.4. Example: Prompt that Appears when the Add Waypoint Command is Entered



All XBee commands function in the regular system processing loop - the results may not be instantaneous. For example, entering the **Rudder Center** command will not immediately move the rudder, instead the command will set a flag that indicates to the vehicle C&C what the desired rudder position is. The next

time the C&C interacts with the rudder, it will be set to the desired position. There is often a 15 or 20 second delay between when the command is entered and when the result is visible to the user.

The following XBee commands require a more detailed explanation:

- **Command #15: Go To Waypoint (0=hold) --** here, the 0 means hold the vehicle at its current position. The vehicle generates a new temporary way point at the last good GPS position received and attempt to maintain that position as much as possible.

- **Command #21: Set Steering Params --** this command will change how the rudder behaves and can drastically affect vehicle behavior. There are multiple parameters that must be entered. If you must use this command, going through WGMS is recommended, since the data input and formatting is easier to see and mistakes are easier to catch.

- **Command #22: Set Debug Level and Mask --** This command will affect the data being displayed in the console window. The output will display all messages in the console window that are at a priority level the same or higher than the debug level. Valid values for the debug level range from 0 up to 5. At 0, All messages will be displayed and as the debug level is increased, the number of messages display will be filtered down until at a debug level of 5, only basic vehicle operation and critical error messages will be displayed in the console window. The Debug mask allows a user to selectively enable all debug messages for a given component regardless of the debug level. Mask values are treated as an OR mask and may be a sum of any of the following values:

  - 0 = no mask is used

  - 8 = compass/magnetometer

  - 16 = iridium/satellite communications

  - 32 = various messages used for bench debugging

  - 64 = communications with the rudder module

  - 128 = line following calculations and messages

- **Command #37/38 vs. #45/46:** These commands allow you to turn the rudder module, or the payload port on the rudder module, On or Off. They are easy to confuse with each other.

- #37/38 will provide power through the umbilical to the rudder module.

- #45/46 will turn on, or off, the switch inside the rudder module that provides power to the payload port on the rudder module.

- **Command #47/48:** These commands enable power duty cycling of the Iridium modem. If there is no data to transmit, when the duty cycle is enabled the modem will power down. When data is available to transmit, the modem will power up and transmit the data. These commands only change the behavior for the current run of the system. When the C&C reboots, these changes are no longer in place and the system will revert to default behavior.

- **Command #50: Cur Ir Mode=Dflt --** This command makes the current Iridium Duty Cycle mode the default mode when the system boots up. For example, if you want to make modem duty cycling the default behavior at start up, first enter command #48 to allow the modem to power down when not in use, and then enter command #50 to make this behavior the default behavior after the system reboots.

- **Commands #54/55:** Set Default Comm Channel to XBee (54) or Iridium (55) -- these commands set the communication path where unsolicited data from the vehicle will be sent. This will not affect the

functionality of the console window, and any command that the vehicle receives will be acknowledged over the same communication path where the command was received.

- **Command #62: Telem Now --** This command will queue up the current system status telemetry for transmission back to WGMS over the default communication channel, regardless of what the current schedule is for telemetry transmission. This is very useful for testing, especially over Iridium, since it can speed up the command response time by initiating a status telemetry transmission when you are aware of queued WGMS commands over Iridium.
- **Command #98: XBee Off --** This command will turn off the XBee transitter on the C&C dry box. If this command is used, the XBee console window will become inactive. The only ways to re-enable XBee at this point are through a command from WGMS over some other communication path (most likely Iridium) to turn XBee on, or to reboot the system.

Menu Error Responses

If the command cannot be processed by the vehicle C&C, an appropriate error message will be displayed in the console. The following example shows the messages that are displayed in response to some typical command menu entry errors:

```
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!
```

Chapter 14. Troubleshooting (Sensor Integration Issues)

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 Liquid Robotics User Manual for troubleshooting of on-water operational problems. (are there more issues we want to add to this chapter?)

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.

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 the Power Key on the C&C dry box is inserted. The Power Key is what turns on the electrical system. An LED light adjacent to the Power Key is activated when power is 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 5.7 “External Charging".

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 using an XBee command (see Chapter 13, *XBee Communications*). If you have Admin privileges, you can issue the Float Reboot command from the Dashboard. For more information, see the section called “Float Reboot (0xA000)”.

A “hard” re-boot can be initiated by powering the vehicle OFF (remove the Power Key), waiting 20 seconds, and then turning the vehicle back ON (insert the Power Key).

Contacting Liquid Robotics

If you've tried the solutions suggested here, and the problem persists, gather up the data and contact Liquid Robotics, as described in 13.4 “ Contacting Liquid Robotics Inc."

Problems and Responses

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 at 111,111 bps.

Responses:

1. Verify that the serial port speed is set to 111,111 bps on the XBee Relay.
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.
5. Verify that the PAN ID's are set to the same value in the vehicle parameters and on the XBee Relay.
6. Stop the relay, unplug the dongle and then reconnect the dongle, then restart the relay.

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 (see the Wave Glider User Manual for more information).
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.

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.

LED Light Will Not 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.

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 0 “LED Light Won't Turn ON/OFF".

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 Dry Box.
2. Check the connectors on C&C and Payload Dry Boxes.
3. Check for an over-current alarm.
4. Check the power status for low batteries.

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.

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 Dry Box.
2. Check that C&C has turned on power to the Payload.
3. Check for an over-current alarm.
4. Check payload electronics UART.

Float Continuously Re-Boots

This indicates a serious electronics problem. Contact Liquid Robotics.

Sub Continuously Re-Boots

This indicates a serious electronics problem. Contact Liquid Robotics.

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

Table 14.1. C&C Alarms

Alarm	Byte	bit	For More Information
Sub Leak Alarm	0	0	See the section called “Sub Leak Alarm (Byte 0 | bit 0)”
Float Leak Alarm	0	1	See the section called “Float Leak Alarm (Byte 0 | bit 1)”
Sub-to-Float Comms Alarm	0	2	See the section called “Sub-to-Float Comms Alarm (Byte 0 | bit 2)”
Payload Alarm	0	3	See the section called “Payload Alarm (Byte 0 | bit 3)”
Float Battery Low Alarm	0	4	See the section called “Float Battery Low Alarm (Byte 0 | bit 4)”
Umbilical Fault Alarm	0	5	See the section called “Umbilical Fault Alarm (Byte 0 | bit 5)”
GPS Not Functioning	0	7	See the section called “GPS Not Functioning (Byte 0 | bit 7)”
Sub Reboot Alarm	1	1	See the section called “Sub Reboot Alarm (Byte 1 | bit 1)”
Float-to-Sub Comms Error	1	2	See the section called “Float-to-Sub Comms Error (Byte 1 | bit 2)”

Alarm	Byte	bit	For More Information
Float Temperature Threshold Exceeded	1	4	See the section called “Float Temperature Threshold Exceeded (Byte 1 | bit 4)”
Float Pressure Threshold Exceeded	1	5	See the section called “Float Pressure Threshold Exceeded (Byte 1 | bit 5)”
EPROM Error	1	6	See the section called “EPROM Error (Byte 1 | bit 6)”
Float Reboot Alarm	1	7	See the section called “Float Reboot Alarm (Byte 1 | bit 7)”

The alarms are described below:

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.

Float Leak Alarm (Byte 0 | bit 1)

Water leak was detected in the C&C Dry Box. This should never occur unless the vehicle is in the water. The C&C Dry Box 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 Dry Box.

Sub-to-Float Comms Alarm (Byte 0 | bit 2)

The C&C computer detected a data comm error from the Glider.

Responses:

1. Check the electrical connectors on the C&C Dry Box and the Rudder Module.

2. Check the Umbilical for cracks or tears.>

3. Try re-booting. If the error occurs again, contact Liquid Robotics.

Payload Alarm (Byte 0 | bit 3)

General problem with a payload. It may be a communications, electrical or environmental issue.

Responses:

1. Check the connectors between the C&C Dry Box and each Payload Dry Box.

2. Verify that the Payload is not drawing more than 1 Amp of current.

3. Try re-booting. If the alarm occurs again, contact Liquid Robotics.

Float Battery Low Alarm (Byte 0 | bit 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 Liquid Robotics.

Umbilical Fault Alarm (Byte 0 | bit 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.

GPS Not Functioning (Byte 0 | bit 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.

Sub Reboot Alarm (Byte 1 | bit 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 Liquid Robotics.

Float-to-Sub Comms Error (Byte 1 | bit 2)

The Sub computer detected a data comm error from the C&C computer.

Responses:

1. Check the electrical connectors on the C&C Dry Box and the Rudder Module.

2. Check the Umbilical for cracks or tears.

3. Try re-booting. If the error occurs again, contact Liquid Robotics.

Float Temperature Threshold Exceeded (Byte 1 | bit 4)

The temperature in the C&C Dry Box 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 Liquid Robotics and do not re-power the unit.

Float Pressure Threshold Exceeded (Byte 1 | bit 5)

The pressure in the C&C Dry Box is above or below the "Float Pressure Threshold" values.

Responses:

1. Remove the valve cover on the top of the C&C Dry Box, 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.

EPROM Error (Byte 1 | bit 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 Liquid Robotics.

Float Reboot Alarm (Byte 1 | bit 7)

An error occurred on the C&C computer, causing a re-boot.

Responses:

1. If the error occurs continually, contact Liquid Robotics.

Contacting Liquid Robotics, Inc.

If you cannot determine or correct the cause of a problem, contact the Liquid Robotics Support Organization.

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

The Liquid Robotics Customer Support organization is available 24/7.

Telephone:

(US) 888-574-4574

(International) 1-408-636-4205

Email:

support@liquidr.com

Appendix A. Command Descriptions

Listed below are descriptions and explanations of the WGMS commands, and the expected response for each. The ACK/NAK format types are described in Chapter 9.

Note: In the following command descriptions all fields marked "reserved" or "not used" must have a value of zero, or the CRC will trigger an error.

Unrestricted Commands

No Command (0x0000)

- A "No Command" is not sent to the vehicle, 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**

Hold Current Position (0x0001)

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

ACK – Status Return (see the section called “ACK - Status Return (0x11)”)

NAK – Data Error (see the section called “NAK – Invalid Data (0x22)”) if Latitude or Longitude error

Field Name	Bytes	Value
Structure ID	1	**0x10**
Command Value	2	**0x0001**
(reserved)	5	0
Latitude	4	[see the section called “Waypoint Values”]
Longitude	4	[see the section called “Waypoint Values”]

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 (see the section called “ACK - Status with Waypoint Read-Back (0X10)”)

NAK – Data Error (see the section called “NAK – Invalid Data (0x22)”) 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)

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 (see the section called “ACK - No Payload Response (0x13)”)

NAK – Data Error (see the section called “NAK – Invalid Data (0x22)”) 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 (see the section called “NAK – Invalid Data (0x22)”) 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)

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 (see the section called “ACK - No Payload Response (0x13)”)

Field Name	Bytes	Value
Structure ID	1	**0x10**

Field Name	Bytes	Value
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)

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 (see the section called “ACK - No Payload Response (0x13)”)

NAK – Data Error (see the section called “NAK – Invalid Data (0x22)”) 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)

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 (see the section called “ACK - No Payload Response (0x13)”)

Field Name	Bytes	Value
Structure ID	1	**0x10**
Command Value	2	**0x000A**
Channel Number	1	See Channel Number options below.
(reserved)	4	0
Latitude	4	0 (not used)

Field Name	Bytes	Value
Longitude	4	0 (not used)

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

Set Unique ID (0x000F)

Set Unique Id sets the vehicle Unique Id programmed into Non-Volatile memory. It can only be issued by System Administrators. Typically this command is used only once in the vehicle lifetime when the vehicle is first brought online.

NOTE: This command can only be issued using Iridium.

The ACK for this command is structure type 0x1E containing the Unique Id.

Field Name	Bytes	Default Value
Structure Id	1	0x10
Command Value	2	0x000F
Hyperion Id	4	
Field 7	1	0
Latitude	4	0
Longitude	4	0

Twist Umbilical (0x1007)

Field Name	Bytes	Default Value
Structure Id	1	0x10
Command Value	2	0x1007
Number of Twists	1	Value (-7 – 7)
Field 4	1	0
Field 5	1	0
Field 6	1	0
Field 7	1	0
Latitude	4	0
Longitude	4	0

Retrieve Debug Buffer (0x7000)

System Admin/Developer roles have access to this command. The ACK for this command should have the debug buffer packet in it.

Field Name	Bytes	Default Value
Structure Id	1	0x10
Command Value	2	0x7000
Field 3	1	0
Field 4	1	0
Field 5	1	0
Field 6	1	0
Field 7	1	0
Latitude	4	0
Longitude	4	0

Retrieve Sub Heading Buffer (0x7001)

System Admin/Developer roles have access to this command. The ACK for this command should have the debug buffer packet in it.

Field Name	Bytes	Default Value
Structure Id	1	0x10
Command Value	2	0x7001
Field 3	1	0
Field 4	1	0
Field 5	1	0
Field 6	1	0
Field 7	1	0
Latitude	4	0
Longitude	4	0

Retrieve PID Integral Cache (0x7002)

System Admin/Developer roles have access to this command. The ACK for this command should have the debug buffer packet in it.

Field Name	Bytes	Default Value
Structure Id	1	0x10
Command Value	2	0x7002
Field 3	1	0
Field 4	1	0
Field 5	1	0
Field 6	1	0
Field 7	1	0
Latitude	4	0
Longitude	4	0

Retrieve PID Operating Params (0x7003)

System Admin/Developer roles have access to this command. The ACK this command should have the debug buffer packet in it.

Field Name	Bytes	Default Value
Structure Id	1	0x10
Command Value	2	0x7003
Field 3	1	0
Field 4	1	0
Field 5	1	0
Field 6	1	0
Field 7	1	0
Latitude	4	0
Longitude	4	0

New Waypoint (0x8000)

Waypoint numbers 0 and 255 are not valid for waypoint creation. The ACK should include the number of new waypoints created, the number of previous waypoints overwritten, and as long as at least 1 waypoint has been added or modified by the Wave Glider, it should respond with an ACK. The Wave Glider will NAK if no waypoints in the system have been added or modified.

Field Name	Bytes	Default Value
Structure Id	1	0x10
Command Value	2	0x8000
Field 3	1	0
Field 4	1	0
Field 5	1	0
Field 6	1	0
Waypoint Number	1	Waypoint number
Latitude	4	Latitude
Longitude	4	Longitude

Read Unique ID (0x000E)

- A vehicle unique 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).

ACK – Structure type 0x1E (see the section called “ACK – Unique ID (0x1E)”) containing the WGMS ID.

Field Name	Bytes	Value
Structure ID	1	**0x10**

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

Set Rudder Position (0x1006)

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

ACK – No Payload (see the section called “ACK - No Payload Response (0x13)”)

Field Name	Bytes	Value [range]	Notes
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)	

Notes:

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

Negative → Left Rudder

Positive → Right Rudder

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

- Turns On or Off the visible light.

ACK – No Payload (see the section called “ACK - No Payload Response (0x13)”)

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)

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

- Turns On or Off the Infrared (IR) light.

ACK – No Payload (see the section called “ACK - No Payload Response (0x13)”)

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)

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

- Turns On or Off the XBee communications channel.

ACK – No Payload (see the section called “ACK - No Payload Response (0x13)”)

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)

Retrieve Status (0x4001)

- Retrieves the Status Return data packet. Refer to section 0 for a detailed description of the Status Return packet.

ACK – Status Return (see the section called “ACK - Status Return (0x11)”)

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)

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 can be used instead (see the section called “Set Parameters (0x6000)”).

ACK – Status Return (see the section called “ACK - Status Return (0x11)”)

Field Name	Bytes	Value [range] (units)	Notes
Structure ID	1	**0x11**	

Field Name	Bytes	Value [range] (units)	Notes
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 (also see 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 Dry Box 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 Dry Box 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.

Retrieve Identification (0x4003)

- Requests the vehicle to send back the ID Packet. Refer to 9.2.3 for a description of the ID Packet format.

ACK – Identification Packet (see the section called “ACK - Vehicle Identification Data (0x12)”)

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

Retrieve Power Status (0x5001)

- Requests the vehicle to send back the Power Controller Status. See the section called “ACK - Power Controller Status (0x14)” for a description of the Power Controller Status Packet format.

ACK – Power Controller Packet (see the section called “ACK - Power Controller Status (0x14)”)

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

Retrieve Rudder Control (0x5002)

- Requests the vehicle to send back the Rudder Control Packet. Refer to 9.2.6 for a description of the Rudder Control Packet format.

ACK – Rudder Control Packet (see the section called “ACK - Rudder Control Data (0x15)”)

Field Name	**Bytes**	**Value**
Structure ID	1	**0x10**
Command Value	2	**0x5002**

Field Name	Bytes	Value
(reserved)	5	0
Latitude	4	0
Longitude	4	0

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 (see the section called “ACK - Rudder Control Data (0x15)”)

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°)	

Field Name	Bytes	Value (units)	Default	Notes
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.

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 the section called “Set Restricted Parameters (0x6000)”.

ACK – No Payload (see the section called “ACK - No Payload Response (0x13)”)

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	

Field Name	Bytes	Value [range]
.....	...	…
.....	...	…
.....	...	…
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 not 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 dry box. 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.

Set Single Waypoint (0x8000)

- Used to set a single new Waypoint Lat/Long value. Refer to 0 for a description of the Lat/Long values.
- Waypoint numbers must be between 1 and 254.

ACK – No Payload (see the section called “ACK - No Payload Response (0x13)”)

NAK – Data Error (see the section called “NAK – Invalid Data (0x22)”) 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	[see the section called “Waypoint Values”]
Longitude	4	[see the section called “Waypoint Values”]

Note:

When this command is sent from the WGMS, the system will respond to the ACK by automatically issuing a “Read Waypoint” command, resulting in two ACKs being returned.

Read Waypoint (0x8003)

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

ACK – Status with Waypoints Read-Back (see the section called “ACK - Status with Waypoint Read-Back (0X10)”)

NAK – Data Error (see the section called “NAK – Invalid Data (0x22)”) 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.

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 (see the section called “ACK - No Payload Response (0x13)”)

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.

Follow Course With Options (0x8102)

- 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 (see the section called “ACK - No Payload Response (0x13)”)

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

Add MMSI to Watch List (0x8200)

Add MMSI to vehicle watch list to allow for increased reporting rate for the vessel under watch.

Field Name	Bytes	Default Value
Structure Id	1	0x10
Command Value	2	0x8200
MMSI	4	Any
Field 7	1	0
Latitude	4	0
Longitude	4	0

Initialize File Write (0x9000)

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

ACK – No Payload (see the section called “ACK - No Payload Response (0x13)”)

NAK:

- Command Error (see the section called “NAK - Invalid Command (0x22)”)

- Data Error (see the section called “NAK – Invalid Data (0x22)”)
- File Error (see the section called “NAK – File Error (0x24)”)

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 (see the section called “File Names”)	[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 (see the section called “Addresses”). 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. (see the section called “File Names”).

[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 file names 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.

Write File Block (0x9001)

- Send a single block of data from the file.

ACK – No Payload (see the section called “ACK - No Payload Response (0x13)”)

NAK:

- Command Error (see the section called “NAK - Invalid Command (0x22)”)
- Data Error (see the section called “NAK – Invalid Data (0x22)”)
- File Error (see the section called “NAK – File Error (0x24)”)

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 (see the section called “Addresses”). 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.

Delete File (0x9002)

- Delete a file from the selected storage address.

ACK – No Payload (see the section called “ACK - No Payload Response (0x13)”)

NAK:

- Command Error (see the section called “NAK - Invalid Command (0x22)”)
- Data Error (see the section called “NAK – Invalid Data (0x22)”)
- File Error (see the section called “NAK – File Error (0x24)”)

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. (see the section called “File Names”)	[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 (see the section called “Addresses”). The Float C&C has the reserved address value: 0x0000.

[2] The "File Name" is an ASCII string padded by spaces (see the section called “File Names”). 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

Read File Request (0x9003)

- Send a request to read a file from the vehicle.

ACK – Read File ACK

NAK:

- Command Error (see the section called “NAK - Invalid Command (0x22)”)
- Data Error (see the section called “NAK – Invalid Data (0x22)”)
- File Error (see the section called “NAK – File Error (0x24)”)

Field Name	Bytes	Value [range]	Note
Structure ID	1	**0x1B**	

Field Name	Bytes	Value [range]	Note
Command Value	2	**0x9003**	
Storage Controller Address	2	[1 to 256] + [1 to 256]	[1]
File Name	11	8+3 ASCII string. (see the section called “File Names”)	[2]
Transaction ID	4	See note [3]	[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. (see the section called “File Names”)

[3] The “Transaction ID” is -1 for the first read. For retries, it is the original Read File ID. (see the section called “Read File Transaction Closed (0x9007)”)

[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.

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 (see the section called “ACK - Detailed File List Data (0x1C)”)

ACK – File at Destination [2nd response if file transferred] (see the section called “ACK – File at Destination (0x1F)”)

NAK:

- Command Error (see the section called “NAK - Invalid Command (0x22)”)
- Data Error (see the section called “NAK – Invalid Data (0x22)”)
- File Error (see the section called “NAK – File Error (0x24)”)

Field Name	Bytes	Value [range]	Notes
Structure ID	1	**0x1C**	

Field Name	Bytes	Value [range]	Notes
Command Value	2	**0x9004**	
Storage Controller Address	2	[1 to 256] + [1 to 256]	[1]
File Name	11	8+3 ASCII string. (see the section called “File Names”)	[2]
File ID	4	**0xFFFFFFFF** (unless WGMS-generated)	[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 (see the section called “Addresses”). The Float C&C has the reserved address value: 0x0000

[2] The "File Name" is an ASCII string padded by spaces (see the section called “File Names”). 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.

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 (see the section called “ACK - No Payload Response (0x13)”)

ACK – File at Destination [2nd response if file transferred] (see the section called “ACK – File at Destination (0x1F)”)

NAK:

- Command Error (see the section called “NAK - Invalid Command (0x22)”)
- Data Error (see the section called “NAK – Invalid Data (0x22)”)

- File Error (see the section called “NAK – File Error (0x24)”)

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.

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 (see the section called “ACK - No Payload Response (0x13)”)

NAK – Command Error (see the section called “NAK - Invalid Command (0x22)”), Data Error (see the section called “NAK – Invalid Data (0x22)”)

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 (see the section called “Addresses”). 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.

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 (see the section called “ACK - No Payload Response (0x13)”)

NAK – Command Error (see the section called “NAK - Invalid Command (0x22)”), Data Error (see the section called “NAK – Invalid Data (0x22)”)

Field Name	Bytes	Default Value	Notes
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)	

Notes:

[1] The "Storage Controller Address" is the 1-byte Board Address, and the 1-byte Task ID (see the section called “Addresses”). The Float C&C has the reserved address value: 0x0000.

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 (see the section called “ACK - Detailed File List Data (0x1C)”)

ACK – File at Destination [2nd response if file transferred] (see the section called “ACK – File at Destination (0x1F)”)

NAK:

- Command Error (see the section called “NAK - Invalid Command (0x22)”)
- Data Error (see the section called “NAK – Invalid Data (0x22)”)
- File Error (see the section called “NAK – File Error (0x24)”)

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. (see the section called “File Names”)	[2]
File WGMS ID	4	See note [3]	[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 (see the section called “Addresses”). The Float C&C has the reserved address value: 0x0000

[2] The "File Name" is an ASCII string padded by spaces (see the section called “File Names”). 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.

Write Console Script (0x9100)

This is a single packet command to a particular 16-bit board address on a Wave Glider expecting either ASCII or binary data..

Field Name	Bytes	Value
Structure ID	1	**0x1E** (Write Console Input to Wave Glider)

Field Name	Bytes	Value
Command Value	2	**0x9100** (execute component command – ASCII format)
Destination Address	2	0 = Float C&C; otherwise, the address for the board to receive console input or command
Function	1	0 or destination specific value (user entered)
Number of Bytes in Data	2	The number of bytes in the Data field (at least 1)
Data	1..N	Data to send into Wave Glider Console input (ASCII)

Write Console Binary (0x9101)

This is a single packet command to a particular 16-bit board address on a Wave Glider expecting either ASCII or binary data.

Field Name	Bytes	Value
Structure ID	1	**0x1E** (Write Console Input to Wave Glider)
Command Value	2	**0x9101** (execute component command – binary format)
Destination Address	2	0 = Float C&C; otherwise, the address for the board to receive console input or command
Function	1	0 or destination specific value
Number of Bytes in Data	2	The number of bytes in the Data field (at least 1)
Data	1..N	Data to send into Wave Glider Console input (HEX)

Float Reboot (0xA000)

System Admin/Developer roles have access to this command to reset the float in the case of no-response or debug. This command is not expected to be ACKed by the vehicle and will be followed up by a Float Reboot alarm on the next telemetry transmission.

Field Name	Bytes	Default Value
Structure Id	1	**0x10**
Command Value	2	**0xA000**
Field 3	1	0
Field 4	1	0
Field 5	1	0
Field 6	1	0
Field 7	1	0

Field Name	Bytes	Default Value
Latitude	4	0
Longitude	4	0

Reserved (0xA51E)

The ACK for this command is structure type 0xA1.

Field Name	Bytes	Default Value
Structure Id	1	**0x64**
Command Value	2	**0xA51E**
Internal Use	Variable (68)	
Internal Use	32	

Secure On (0xA500)

Command to start the process of switching to a secure (encrypted) channel communication between vehicle and WGMS. *This command and its ACK is always sent in clear text.

Field Name	Bytes	Value
Structure ID	1	**0x64**
Command Value	2	**0xA500**
WGMS Ephemeral ECDH-P256 Public Key (in Vehicle Key Format**)	Variable (68)	
Client Channel Id	32	

*See Secure On ACK message format for response.

** Vehicle Key Format is 68 byte Big Endian Key Format:

Description	Value	Length (bytes)	
Bit Length X (for P256 and SHA256)	256	2	Count of bits (Big Endian)
X	…	32	
Bit Length Y (for P256 and SHA256)	256	2	Count of bits (Big Endian)
Y	…	32	

Secure Authenticate (0xA501)

Command to send the vehicle a pass-phrase to authenticate WGMS to the vehicle. This command is always sent in clear text, and the Pass-Phrase field is encrypted with a symmetric key.

Field Name	Bytes	Value
Structure ID	1	**0x65**

Field Name	Bytes	Value
Command Value	2	**0xA501**
Client Channel Id	32	
IV size	1	
Cipher text size	2	
IV	Variable	
Pass-Phrase (cipher text)	Variable	

Secure Register Client (0xA502)

Command to send the vehicle a client registration allowing the vehicle to recognize future secure clients.

Field Name	Bytes	Value
Structure ID	1	**0x66**
Command Value	2	**0xA502**
Client Permanent ECDsa-P256 Public Key (Vehicle Key Format)	68	
Client Channel Id	32	
Pass-Phrase	20	

Secure Off (0xA504)

Command to start the process of switching from a secure (encrypted) channel to clear text communication between the vehicle and WGMS.

Field Name	Bytes	Value
Structure ID	1	**0x10**
Command Value	2	**0xA504**
Field 3	1	0
Field 4	1	0
Field 5	1	0
Field 6	1	0
Field 7	1	0
Latitude	4	0
Longitude	4	0

Secure Enable Partition (0xA505)

This is a single packet command that will be intercepted by the Crypto Lib and be replaced by Write Debug Console with the appropriate encrypted partition mount command.

Field Name	Bytes	Value
Structure ID	1	**0x10**
Command Value	2	**0xA505**

Field Name	Bytes	Value
PartId	4	User entered
Field 7	1	0
Latitude	4	0
Longitude	4	0

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	Link to Command Description
Set WGMS ID	**0x000F**	0x10	the section called “Set Unique ID (0x000F)”
Set Restricted Parameters	**0x6000**	0x10	the section called “Set Restricted Parameters (0x6000)”
Retrieve Debug Buffer	**0x7000**	0x10	the section called “Retrieve Debug Buffer (0x7000)”
Retrieve Sub Heading Buffer	**0x7001**	0x10	the section called “Retrieve Sub Heading Buffer (0x7001)”
Retrieve PID Integral Cache	**0x7002**	0x10	the section called “Retrieve PID Integral Cache (0x7002)”
Retrieve PID Operating Parameters	**0x7003**	0x10	the section called “Retrieve PID Operating Parameters (0x7003)”
Float Reboot	**0xA000**	0x10	the section called “Float Reboot (0xA000)”

Set Unique ID (0x000F)

- Available to Admins only.
- Writes the vehicle unique ID into the C&C Non-Volatile Memory. This command is typically used only once in the vehicle lifetime - when the vehicle is first brought online.

Note: This command can only be issued by using 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 (see the section called “ACK – Unique ID (0x1E)”)

Field Name	Bytes	Value
Structure ID	1	**0x10**
Command Value	2	**0x000F**

Field Name	Bytes	Value
Vehicle Unique ID	4	See the section called “Vehicle Unique ID”
(reserved)	1	0
Latitude	4	0 (not used)
Longitude	4	0 (not used)

Set Restricted Parameters (0x6000)

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

ACK – No Payload (see the section called “ACK - No Payload Response (0x13)”)

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.

Retrieve Debug Buffer (0x7000)

- Requests retrieval of the Debug Buffer Packet.

ACK – Debug Buffer Packet (see the section called “ACK – Debug Buffer (0x16)”)

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)

Retrieve Sub Heading Buffer (0x7001)

- Requests retrieval of the Sub Heading Buffer.

ACK – Sub Heading Buffer (see the section called “ACK – Sub Heading Buffer (0x17)”)

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)

Retrieve PID Integral Cache (0x7002)

- Requests retrieval of the PID Integral Cache.

ACK – PID Integral Cache (see the section called “ACK – PID Integral Cache (0x18)”)

Field Name	**Bytes**	**Default Value**
Structure ID	1	**0x10**
Command Value	2	**0x7002**
(reserved)	5	0

Field Name	Bytes	Default Value
Latitude	4	0 (not used)
Longitude	4	0 (not used)

Retrieve PID Operating Parameters (0x7003)

- Requests retrieval of the PID Operating Parameters.

ACK – PID Operating Parameters (see the section called “ACK – PID Operating Parameters (0x19)”)

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)

Float Reboot (0xA000)

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

Appendix B. XML, Soap and Entities

XML

XML both defines and is the structure of the WGMS.

Entities

The primary components of 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.

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

Note: The time zone is the zone of the message originator.

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.

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

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/s
  <soap:Body>
   <EntityNameResponse xmlns="http://schemas.LiquidRobotics.com/WebServices">
     <EntityNameResult>string</EntityNameResult>
   </EntityNameResponse>
  </soap:Body>
</soap:Envelope>
```

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/
  <soap12:Body>
   <EntityNameResponse xmlns="http://schemas.LiquidRobotics.com/WebServices">
     <EntityNameResult>string</EntityNameResult>
   </EntityNameResponse>
  </soap12:Body>
</soap12:Envelope>
```

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	0
<CreateLoginSession>	Create and returns an encrypted session cookie	0
<DeleteEntity>	Delete a previously-created entity	0
<GetEntityColumnMetadata>	Retrieve from database a single column of metadata for a specified Entity type	0
<GetEntityColumnMetadataSet>	Retrieve from database a multi-column dataset for the specified entity type	0
<GetEntityColumnSelectMetadataSet>	Retrieve from database a named column dataset for the specified entity type	0
<GetEntityMetadata>	Retrieve from the database the complete metadata for the specified entity type	0
<GetEntitySet>	Retrieve from the database specific metadata for the entity type	0
<GetRelatedEntitySet>	Retrieve from the database related metadata for the entity type	0
<GetVehicleCommands>	Retrieve from the database date-specific vehicle command data	0
<GetVehicleCrumbs>	Retrieve from the database date-specific vehicle crumb data	0
<GetVehiclePowerCtlStatuses>	from the database date-specific vehicle crumb data	0
<ReadEntity>	Retrieve a single row of entity data from the database	0
<UpdateEntity>	Update a specific entity field in the database	0

Function Descriptions

Create 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)

Create 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)

Delete 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)

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)

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)

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)

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)

Get 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)

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)

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)

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)

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)

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)

Update 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 C. Iridium and Relay Header Data

Iridium Vehicle-to-WGMS Data Packets

Vehicle-to-WGMS 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 Vehicle-to-WGMS 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

Iridium Vehicle-to-WGMS Field Name Descriptions:

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 (see Chapter 9, *Vehicle Responses* and Chapter 10, *Vehicle Telemetry*)

Mobile Originated 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)

Relay Vehicle-to-WGMS Packets

Vehicle-to-WGMS Relay packets are similar (but not identical) to the Iridium Vehicle-to-WGMS 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 Vehicle-to-WGMS 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
Vehicle Unique ID	10-13	4	See the section called "Vehicle Unique ID"
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 not 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”

Vehicle Unique ID – The unique 4-byte identification number that is stored in the vehicle's Non-Volatile memory. See the section called “Vehicle Unique ID”.

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.

Iridium WGMS-to-Vehicle Data Packets

A WGMS-to-Vehicle 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 WGMS-to-Vehicle 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	$\leq$ 135		Little Endian

Iridium WGMS-to-Vehicle Field Name Descriptions:

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. (see Chapter 8, *Commands*)

Relay WGMS-to-Vehicle Packets

A WGMS-to-Vehicle 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 WGMS-to-Vehicle 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
Vehicle Unique ID	10-13	4	See the section called “Vehicle Unique ID”
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	

Relay WGMS-to-Vehicle Field Name Descriptions:

The **Protocol Version** and **CDR Reference** numbers are the same as used for Vehicle-to-WGMS Relay packets.

Protocol Version	Value
Iridium (plug-in)	1
XBee	2
Serial	3
802.11	4

Protocol Version	Value
Iridium (alternate)	5

CDR Reference	Value
Iridium (plug-in)	"Msg1"
Xbee	"Xbee"
Serial	"Ser1"
802.11	"WiFi"
Iridium (alternate)	"Irid"

Vehicle Unique ID – The unique 4-byte identification number that is stored in the vehicle's Non-Volatile memory - see the section called “Vehicle Unique ID”.

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 D. 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 on 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 seed)
{
    UInt16 crc = seed;

  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 = (UInt16)((crc >> 1) ^ 0xA001);
    }
    else
    {
       crc = (UInt16)(crc >> 1);
    }
  }

  return crc;
}
```