

Slocum G3 Glider

Operators Manual

P/N M311172-NFC, Rev. 1



Teledyne Webb Research,
a Business Unit of Teledyne Instruments, Inc.
49 Edgerton Drive
North Falmouth, MA 02556
U.S.A.

Tel: (508) 563-1000

<http://www.teledynemarine.com/webb-research/>

Use and Disclosure of Data

Information contained herein is classified as EAR99 under the U.S. Export Administration Regulations.
Export, reexport or diversion contrary to U.S. law is prohibited.

© Copyright 2017–2019
Teledyne Webb Research

The material in this document is for information purposes only and is subject to change without notice. Teledyne Webb Research assumes no responsibility for any errors or for consequential damages that may result from the use or misrepresentation of any of the material in this publication.

AquaShield is a trademark of AquaShield, Inc. Drakeol is a registered trademark of Calumet Specialty Products Partners, L.P. Loctite is a registered trademark of Henkel Corp., USA. NatraSorb is a registered trademark of Multisorb Technologies. Royal Purple is a registered trademark of Royal Purple, Inc.

Contents

1 Introduction	1-1
New Features with the G3	1-1
External Ballast Locations	1-2
Forward Propulsion	1-2
Navigation and Flight	1-3
Safety and Handling Procedures	1-3
Architecture and Components	1-3
Architecture	1-3
Nose Dome	1-3
Forward Hull Section	1-4
Payload Bay Mid-Hull Section	1-4
Aft Hull Section	1-5
Aft Tail Cone	1-5
Wings	1-5
Slocum Glider Components	1-6
HD Pumps	1-6
Shallow Ballast Pump Assembly	1-7
1000-meter Ballast Pump Assembly	1-9
Pitch Vernier	1-9
Altimeter	1-10
Carbon Fiber Hulls	1-13
Payload Bay	1-14
Payload Bay Connectors	1-17
Payload Bay Computers and Sensors	1-17
Conductivity, Temperature, and Depth (CTD)	1-18
Argos Satellite Platform Transmitter Terminal (PTT)	1-19
Catalyst	1-20
Air Pump System	1-20
Vehicle Controller	1-21
Hardware Interface Board	1-22
Attitude Sensor	1-23
Global Positioning System (GPS)	1-24
Iridium Satellite Telemetry	1-25
Radio Frequency (RF) Modem Telemetry	1-26
Communications	1-26
Pressure Transducer	1-27
Leak Detect Sensor Boards	1-27
Air Bladder	1-28
Burn Wire and Jettison Weight	1-29
Power Port	1-30
Battery Management System (BMS)	1-30
Batteries	1-32
Emergency Battery	1-32
Rechargeable Battery Packs	1-33
Battery Charger	1-34
Pitch Battery Ports	1-35

Battery Charge Indicator	1-36
Battery Charging Instructions	1-36
Optional Comms	1-37
G3 Primary Batteries	1-37
Installing the Battery Packs	1-38
Rechargeable Battery Operational Considerations	1-40
Shipping Requirements	1-40
Fin	1-40
Tail Fin	1-41
Communications Subsystem	1-42
Steering Subsystem	1-42
Strobe Illumination	1-42
Fin Leak Detect	1-42
Tail Fin Maintenance	1-43
Sacrificial Anode	1-44
Batteries/Coulomb Counter	1-45
Desiccant	1-46
Lifting Point	1-47
Recovery System (optional)	1-47
Strobe	1-48
Spare Ports and Modem Readiness	1-48
Optional 10-Watt Thruster	1-49
Operational Usage	1-49
Post Deployment Care	1-50
2 Pre-mission Testing	2-1
On the Beach, Boat, or Bench	2-1
In the Water	2-2
3 Deployment and Recovery	3-1
Deploying the Glider	3-1
Large Ship Deployment	3-4
Recovering the Glider	3-5
4 Emergency Recovery	4-1
5 Glider Communications	5-1
Communicating with the RF Modem	5-1
Communicating with Argos	5-1
Communicating with Iridium	5-1
Communicating with RUDICS	5-2
6 Software Architecture	6-1
G3 Software	6-1
G3 Missions	6-2
PicoDOS	6-3
Config Folder	6-3
Bin Folder	6-4

Logs Folder	6-5
Missions Folder	6-5
Sentlogs Folder	6-5
Autoexec.bat	6-5
GliderDOS	6-6
Masterdata	6-7
Prefixes	6-8
Sensor Commands	6-8
Device Commands	6-9
Science Computer	6-12
Bin Folder	6-13
Config Folder	6-14
Autoexec.bat	6-14
7 Science Data Logging	7-1
Logfile Types	7-1
Configuration Files	7-2
Transparency	7-2
Control	7-3
8 Flight Data Retrieval	8-1
9 Missions	9-1
.mi Files	9-1
.ma Files	9-1
Running Missions	9-1
A Abbreviations and Acronyms	A-1
B Code Theory and Operation	B-1
Operating Systems	B-1
Code Design	B-1
Control Levels	B-2
Control: Stacks, States, and Abort Sequences	B-2
General Control Structure	B-3
Synchronous Abort	B-3
Out of Band Abort	B-4
Hardware Generated Abort	B-4
Sample Mission and Comments	B-4
C Abort Codes	C-1
D Glider Repositories and Forum	D-1
E Ancillary Equipment	E-1
F FreeWave Configuration	F-1
About FreeWave Transceivers	F-1
Setting up the Glider Shoreside FreeWave	F-1

Setting up the Glider FreeWave Slave (internal to the glider) F-2

Choosing a Location for the Transceivers F-3

G Iridium Service and the SIM Card G-1

H Argos Satellite Service and IDH-1

 Instructions for Completing the Argos Technical Information Form.H-1

 Argos Data FormatH-2

I How to Determine Mission Battery Longevity I-1

J How to Edit a Proglets.dat FileJ-1

K Quick Reference and ChecklistsK-1

L Glider Wiring DiagramL-1

Preface

This manual provides the information required to operate the Slocum Glider G3 System. This manual is used in conjunction with the *Slocum Glider Maintenance Manual*. It is divided into the following sections:

- **Section 1—Introduction** provides a general description of the Slocum G3 glider and an overview of its architecture and components.
- **Section 2—Pre-mission Testing** describes the in- and out-of-water tests that should be conducted to qualify a glider before delivery or with new or modified software.
- **Section 3—Deployment and Recovery** provides tips and reference material to assist in deploying and recovering gliders.
- **Section 4—Emergency Recovery** provides tips for recovering a glider during an emergency.
- **Section 5—Glider Communications** includes steps for setting up communications with the glider from various sources, such as a radio frequency (RF) modem, Argos, Iridium, and RUDICS. Steps for setting up direct communication with the glider are also provided.
- **Section 6—Software Architecture** describes the structure of the glider's software: software control hierarchy, folder structure, commands, and masterdata prefixes.
- **Section 7—Science Data Logging** provides an introduction to science data logging and how this functionality works.
- **Section 8—Flight Data Retrieval** is a guide for retrieving data from the glider during a mission surfacing and when a mission has ended.
- **Section 9—Missions** provides instructions for loading mission files onto a glider and running missions.
- **Appendix A—Abbreviations and Acronyms** provides a list of abbreviations and acronyms related to glider maintenance and operations.
- **Appendix B—Code Theory and Operation** provides details on the software used to operate the glider. Topics include a description of the operating systems used, the code design, abort sequences, general control structure for glider deployments, and sample code for a mission.
- **Appendix C—Abort Codes** contains a list of codes used to abort glider tasks and operations.
- **Appendix D—Glider Repositories and Forum** provides instructions for downloading the glider code from the repository.
- **Appendix E—Ancillary Equipment**
- **Appendix F—FreeWave Configuration** includes excerpts from FreeWave Technologies' *Spread Spectrum Users Manual* on how to set up FreeWave transceiver communications on a glider.
- **Appendix G—Iridium Service and the SIM Card** contains a list of companies who sell the Iridium SIM card, general billing and activation information, and how to determine which usage plan is the best option.
- **Appendix H—Argos Satellite Service and ID** provides instructions on how to set up Argos satellite service and an ID number.

- **Appendix I—How to Determine Mission Battery Longevity** provides information about how the length of a mission battery is determined and details on how to monitor the battery voltage.
- **Appendix J—How to Edit a Proglets.dat File** provides instructions for making changes to proglets.dat.
- **Appendix K—Quick Reference and Checklists** provides a link to the latest copy of the *Slocum G3 Glider Operators Manual*.
- **Appendix L—Glider Wiring Diagram**

Notes and Warnings

Where applicable, special notes and warnings are presented as follows:



NOTE A referral to another part of this manual or to another reference; a recommendation to check that certain criteria are met before proceeding further in a step or sequence; or general information applicable to the setup and operation of the Teledyne Webb Research Slocum G3 Glider.



CAUTION Identifies a potential hazard that could be damaging to equipment or could result in the loss of data.



WARNING Identifies a potential hazard that could cause personal injury or death to yourself or to others.

Format Notes

Glider sensors and commands will be denoted in the Courier font throughout this document, as shown in the example below:

Typing `Report ++ m_roll` will report measured roll (`m_roll`) every four seconds.

When displayed on a PC, some areas will be hyperlinked to information available on the Internet, such as:

<http://www.webbresearch.com>

Protected documents are accessible by user account at the TWR Glider User Forum.

Many of the links and the code mentioned in this manual require access by prior arrangement. Please contact glidersupport@teledyne.com to inquire about access to these protected documents.

Manual Revision Notice

This manual could be out of date. Always confirm you are working with the latest revision of the manual. To verify that this manual is the most current, visit the TWR Glider User Forum and download the manual or contact glidersupport@teledyne.com.

Other Manuals and a the TWR Glider User Forum

The G3 glider is supported my multiple documents and the TWR Glider User Forum. The Maintenance Manual focuses on the hardware and how to assemble and disassemble the vehicle and LRU's. This Operators manual outlines the glider's functionality and piloting. Additionally, there is a SFMC Manual to explain the Dockserver interface and piloting tools, and there is a Training guide used as reference during Slocum Glider Training sessions. Before operating a glider, it is recommend that all personnel who come in contact the gliders participate in a TWR hosted Training session, be familiar with the material in all these documents and be a member of the TWR Glider User Forum. This manual and many other operations and support material can be found at the TWR Glider User Forum at <https://datahost.webbresearch.com>.

Customer Service

We welcome your comments and suggestions for improving our products, documentation, and service of the glider system. Please contact glidersupport@teledyne.com should you have any comments or suggestions about this manual, the glider system, or if you require service or support.

Please contact us at:

Site URL: <http://www.teledynemarine.com/webb-research/support>

E-mail: glidersupport@teledyne.com

Mail: 49 Edgerton Drive
North Falmouth, MA 02556

Telephone: (508) 563-1000

1 Introduction

Conceived by Douglas C. Webb and supported by Henry Stommel and others, the class of Slocum gliders is named after Joshua Slocum, the first man to single-handedly sail around the world. These gliders are innovative autonomous underwater vehicles that are available with optimized engines for either coastal (shallow) or deep applications. Each of the two existing glider types is specifically designed to maximize littoral or deep ocean capabilities with ranges from 4 to 1000 meters. These platforms are a uniquely mobile network component capable of moving to specific locations and depths, and occupying controlled spatial and temporal grids. The gliders are driven in a sawtooth vertical profile by variable buoyancy and can move horizontally and vertically.

Long-range and satellite remote sensing systems are being realized in the ocean measurement field. These systems are being used to quantify currents, sea surface height, temperature, acoustic, and optical properties of the water, enabling modeling and prediction of ocean state variables in the littoral zone. A similar nested grid of subsurface observations is required to maximize the impact and ground-truth of the more extensive surface remote sensing observations.

The long-range capabilities of the Slocum gliders make them ideally suited for subsurface sampling at a regional or larger scale. These gliders can be programmed to patrol for weeks, months or even years at a time, surfacing to transmit their data to shore while downloading new instructions at regular intervals, at a substantial cost savings compared to traditional ship-based research methods.

The small relative cost and the ability to operate multiple vehicles with minimal personnel and infrastructure enable fleets of gliders to study and map the dynamic (temporal and spatial) features of our subsurface coastal or deep ocean waters 24 hours daily, 365 days a year.

For users familiar with the G1 and G2 glider systems, this section will detail the new features provided by the G3 system and the required reading for operating a G3 system for the first time.

New Features with the G3

Below you will find the section and page numbers to the content that will detail the new features and required reading for existing Slocum users who are operating a G3 system for the first time. Additionally, legacy manual content has been updated for G3 throughout the manual.

- G3 Software (page 6-2)
- G3 Missions (page 6-1)
- Battery Management System (BMS) (page 1-30)
- Rechargeable Battery Packs (page 1-33)
- HD Pumps (page 1-6)
- SBMB2 (page 1-14)
- Payload Bay Connectors (page 1-17)

- Double O-ring (refer to the *Slocum G3 Glider Maintenance Manual*)
- Power port (page 1-30)
- Optional Comms (page 1-37)
- Spare Ports and Modem Readiness (page 1-48)
- Tail Fin (page 1-41)
- External Ballast Locations (page 1-2)

External Ballast Locations

For the locations of the external ballast, refer to the *Slocum G3 Glider Maintenance Manual*.

Forward Propulsion

Gliders are unique in the autonomous underwater vehicle (AUV) world, because their varying vehicle buoyancy creates forward propulsion. Wings and control surfaces convert the vertical velocity into forward velocity so that the vehicle glides downward when denser than water and glides upward when buoyant (see Figure 1-1 below), which is representative of a shallow engine glider. Gliders require no propeller and operate in a vertical sawtooth trajectory. Optionally, thrusters are available for speed, horizontal flight, surface lens penetration and additional operational capability.

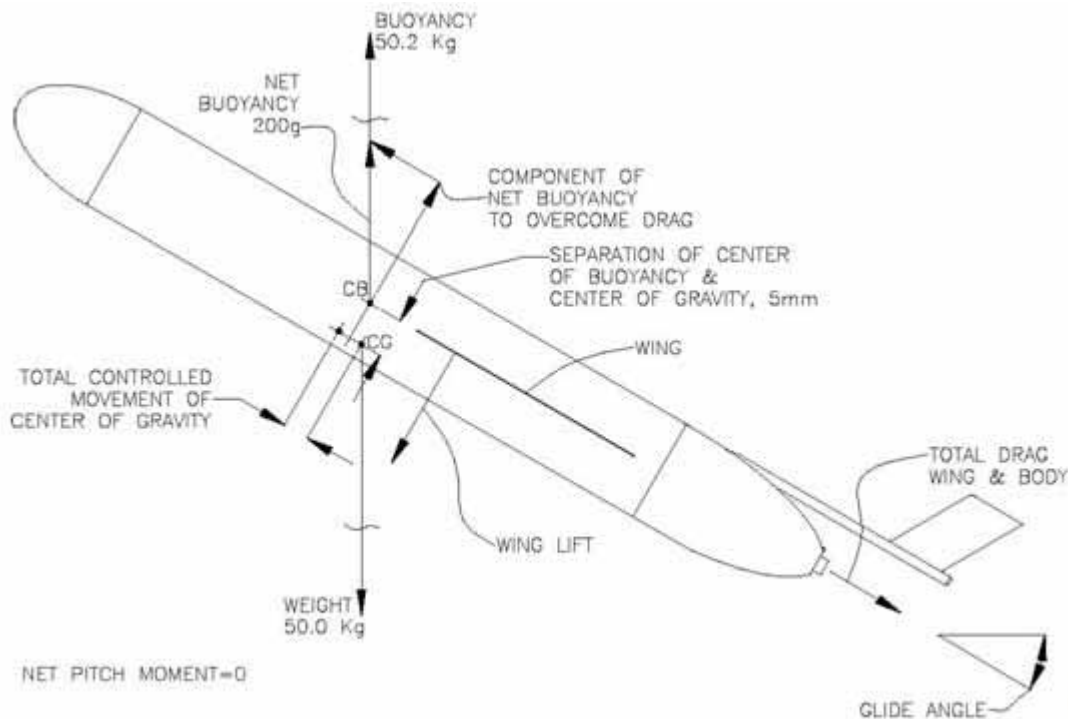


Figure 1-1 Force Balance Diagram of Forces Acting on the Glider (angle of attack not included).

Navigation and Flight

The Slocum Glider navigates to waypoints via dead reckoning, inflecting at depths and altitudes as prescribed in a text mission file. As set by the mission, the glider periodically surfaces to obtain a GPS location fix and to communicate data and instructions with glider operators via local RF or satellite communication. Differences between its estimated dead reckoning position and its GPS position are attributed to currents, which are accounted for in subsequent mission segments via set and drift calculations.

For legacy information about the glider's flight dynamics, contact Glider Support regarding "how it works" documentation.

Safety and Handling Procedures

For information on safety and handling procedures for the Slocum Glider, see the "Warnings and Precautions" section of the *Slocum G3 Glider Maintenance Manual*.

Architecture and Components

The Slocum Glider comprises five total sections (three main hull sections and two wet sections located fore and aft), a design which was chosen for its simplicity, economy, and expandability.

Architecture

The glider's main hull sections are made of carbon fiber wound to a specific angle to provide similar compressibility to ocean water. The nose end cap is a machined pressure resistant elliptical shape, and the tail cap is a truncated cone. Composite wings are swept backward at 45 degrees and are easily replaced using a quick release system.



WARNING Take care while removing and installing wings as they are not buoyant and will sink if dropped.

Nose Dome

The nose dome is an acoustically transparent plastic cap located at the front of the glider that protects the altimeter and ballast pump assemblies. The hole in the center of the nose dome allows the ballast pump to draw water in or push water out to change the vehicle's buoyancy. Some gliders are also outfitted with an optional nose dome recovery assembly.

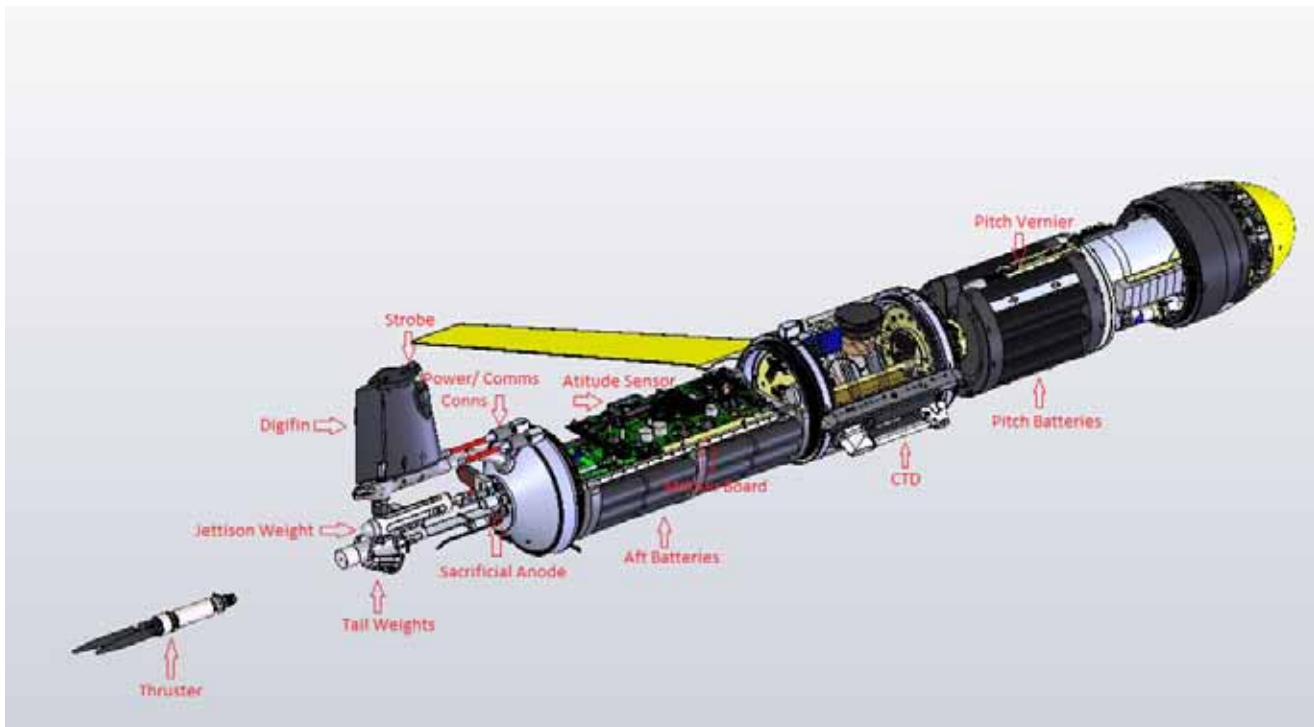


Figure 1-2 Components of the Slocum G3 Glider.

Forward Hull Section

This section houses the ballast pump assembly, pitch vernier, forward batteries, and ballast weights. The pitch vernier is used to move the batteries forward or backward to adjust the glider's pitch. Internal wiring connectors that attach to the payload bay are located at the aft end of the ballast pump assembly.

Payload Bay Mid-Hull Section

The payload bay mid-hull section is capable of accommodating a flexible science payload where a variety of instruments can be easily removed and replaced. This section is composed of two rings and a hull section, with a nominal capacity of 3 to 4 kg. The front ring is typically fitted with a penetrating connector to accommodate a variety of externally mounted science sensors, such as the Rockland MicroRider.

The standard G3 glider has a payload bay computer ("Persistor") that is connected to the glider's main computer. This Persistor controls the sensor packages and collects and stores data via `proglts.dat`, an ASCII file that resides in the `config` directory. Each glider and glider type has a unique `proglts.dat` file that depends on the current configuration of scientific instruments. The original source document can be found at the [glider forum production/target-science/config/proglts.dat](http://gliderforum.org/production/target-science/config/proglts.dat).

Depending on the configuration of science instruments in the payload bay, ballast weight can be added to this section. H-moment adjustments can also be made by moving weight, high or low, in this section of the vehicle.

External wing rails provide external ballast adjustment in the form of wing rail weights, where each weight can add or subtract approximately 14 grams of mass, for fine tuning ballast adjustments quickly in the field (see "Wings" on page 1-5).

Aft Hull Section

The aft hull section houses the main glider electronics and the chassis that ties the vessel together. The main controller board (upper electronics board) carries the flight computer (flight Persistor) and attitude sensor. The communications board, which is located beneath the main controller board, contains the Argos transmitter, GPS, Iridium modem, and radio frequency (RF) modem. The air pump system is attached to the underside of the chassis. This section also contains a catalyst which recombines hydrogen gas and oxygen into water to reduce the risk of explosion in the event that alkaline batteries are shorted or come into contact with water or sea water. The aft battery pack, which is not attached to the chassis, is positioned below the chassis in the bottom half of the aft hull section. Unlike the forward battery pack, the aft battery pack is fixed in place and is not used to control the glider's pitch or roll. Behind the aft battery pack, the "flight" pressure transducer is ported through the aft end cap. In addition, fitted through the aft cap are ports for fin, external air bladder, rechargeable battery connector, spare MS plug, vacuum, and new to the G3, comms and power ports at the top of the cap.

Aft Tail Cone

Above the aft tail cone are the new comms and power ports. Inside there is a wet area that houses the assorted wet mate connectors, air bladder and tail boom, to which the jettison weight and optional thruster are located. The tail boom attaches to the center of the aft end cap. This wet area can be modified to include additional devices, such as science sensors or the optional thruster. The fin, which protrudes from the aft end cap through the aft tail cone, contains antennas for the GPS and Iridium systems, the Argos transmitter, and the radio frequency transmitter. New with the G3 is a strobe light which is now integrated directly to the fin. The fin is sturdy enough so that the glider can be guided by it and drawn onto small boats in conjunction with use of the glider cart. The fin additionally has a tie off point for attaching a neutral line. However, it should be noted that the aft tail boom, which is located just below the fin, is not sturdy and can be bent if grasped while manipulating the glider in the water. The fin and line attachment point should not be used to lift the glider with no other support

Wings

The standard carbon fiber wings for the G3 Slocum Glider are designed with a sweep angle of 45° which is suitable for both shallow and deep gliders. The G3's wings are located aft of the center of buoyancy and provide pitch stability during flight. Wings are attached to the glider via a quick release system that clicks into place. The quick release mechanism is located near the aft

end of the wing rail. Adjustable-ballast wing rails, designed to expedite the field ballasting process, are now available for purchase. These wing rails are slotted to accommodate up to ten 14g stainless steel masses.

Slocum Glider Components



CAUTION Some devices and equipment cannot or should not be tested while the glider is deployed and can only be tested in the lab. The glider should never be put into lab_mode while deployed. Properly trained personnel or glider pilots should be consulted before any of the testing below is attempted. Contact glidersupport@teledyne.com regarding questions, for clarification or to troubleshoot unexpected test results. When glider sensors are not defined below and for default values, refer to the masterdata. Available at the glider forum [glider/windoze/production/masterdata](#) link.

HD Pumps

G3 Gliders continue to provide two optimized capabilities to use either a hydraulic pump for deep applications or a piston driven pump for shallower applications, however both designs have been substantially overhauled to allow for greater performance and reliability.

Both pump designs share a new external nose geometry which allows for the wet section of the nose to be extended for use as a wet payload bay in custom applications. This feature also includes an external nose transition piece that allows for 16 external ballast weights to be added or removed for ballasting, reducing the need to open the glider to make fine adjustments.

The deep, hydraulic, pump was completely redesigned for the G3 making it more robust and easier to maintain. This pump has a new device name, `hd_pump`, that needs to be in the list of installed devices in the `Autoexec.mi`. The HD Pump includes a 3 way ball valve that controls the flow of oil from the external bladder back into the internal reservoir in the glider. During normal operation the glider will open this valve when it's at the surface to retract oil from the external bladder and begin the dive. It will remain closed for the duration of the dive, even through the inflection at the bottom of the Yo where the glider pumps oil out from the internal reservoir to become positively buoyant again. When the glider approaches the top of the yo, it will either open the valve to dive again or continue on to the surface depending on how the mission is configured. In this typical use case, the valve never needs to get opened against high pressure. However, if the mission is configured in a way that requires it to become positively and then negatively buoyant at depth (like a "drift at depth mission"), the valve can experience higher pressure pushing in on it. In the past, this has been dealt with by using a flow restrictor to ensure that the internal plumbing never experiences that higher pressure. Through the use of the 3 way ball valve in the G3 we were able to add the option to use an unrestricted port to retract oil through when the glider is inflecting at a depth that will not damage the internal plumbing on the

pump. This is important because it allows for much faster inflections in the unrestricted position, which allows deep gliders to be substantially more efficient in shallow water. The depth at which the glider decides to use the restricted vs the unrestricted valve position is controlled by the sensor `u_valve_open_max_depth`, but this should not be changed from the default without specific instructions from Teledyne.

G3 gliders have standardized to a high displacement pump of 960 cc or greater depending on the glider type, shallow or deep. If speed or greater operational density range are required the full volume should be used. If energy consumption is the greater concern, it should be noted that the glider will fly well with as little as 300 total cc of drive. Reduced drive can be set empirically by the user or by use of the autoballast software feature.

The pumps are the following:

- Shallow piston driven pumps
- Deep pumps

During ballasting with HD pumps there is significant risk of incorrect ballasting if care is not taken to ensure all of the air is removed from the buoyancy pump.

Shallow Ballast Pump Assembly

A shallow ballast pump assembly is a single-stroke piston design that uses a 90-watt motor and a rolling diaphragm seal to move 1000 cc of seawater directly into and out of a short port on the nose centerline (the stagnation point). The pumps for different motor types (50 and 150-meter) are rated for different pressures based on the gearbox associated with the motor. The mechanical gear drive is not the limiting factor; it is the maximum amount of energy that must be pulled from the battery source. The selection of gearbox/motor assembly should be optimized for the working depth to allow for quick inflections (more important in shallow water) and to minimize energy used on the return stroke.



NOTE Shallow pumps (rolling diaphragm or bellowphragm) have a 10000-cycle service life or 20000 `m_tot_num_inflections` when profiling to full depth. When not profiling to full depth, contact Glider Support regarding the use of the sensor `m_pump_effective_num_cycles`.



NOTE The shallow displacement pump should not be run without either external pressure or internal vacuum on the rolling diaphragm. The vacuum inside the glider should be drawn to 6 inHg less from the external atmosphere to ensure that the diaphragm folds smoothly as it rolls, otherwise damage may result. To eliminate back drive of the pump at pressure, a latching brake is used to hold the motor when at rest.

How to Configure

Factory configured. Please contact glidersupport@teledyne.com for assistance.

How to Test

From lab_mode:

1. Type `wiggle on`.
2. Type `report ++ m_ballast_pumped`.
3. Make sure that the ballast pump successfully completes a full extension (`m_ballast_pumped = +400 cc`) and full retraction (`m_ballast_pumped = -400 cc`) without errors.
4. Type `wiggle off`.

How to Evaluate Data

With a properly ballasted glider, positive cc's should result in a positive buoyancy and a climb. Negative cc's will result in a negative buoyancy and a dive.

Relevant Sensors

Sensor Name	Description
<code>m_ballast_pumped</code>	Measured volume of buoyancy pumped in cc's
<code>c_ballast_pumped</code>	Commanded volume of buoyancy pumped in cc's

1000-meter Ballast Pump Assembly

The deep 350 and 1000-meter 900 cc ballast pump is a rotary displacement design that moves oil from an internal reservoir to an external bladder to change the vehicle's buoyancy. It is necessary that the glider be under vacuum while operating the ballast pump, because the force of the internal vacuum is used to draw oil from the bladder back into the body of the glider. An operator will not damage the pump by running it without a vacuum. A rotary valve is used to control the flow of oil from the bladder to the internal reservoir.

How to Configure

Factory configured. Please contact glidersupport@teledyne.com for assistance.

How to Test

From lab_mode:

1. Type `wiggle on`.
2. Type `report ++ m_de_oil_vol`.
3. Make sure that the ballast pump successfully completes a full extension (`m_ballast_pumped = +430 cc`) and full retraction (`m_ballast_pumped = -430 cc`) without errors.
4. Type `wiggle off`.

How to Evaluate Data

With a properly ballasted glider, positive cc's should result in a positive buoyancy and a climb. Negative cc's will result in a negative buoyancy and a dive.

Relevant Sensors

Sensor Name	Description
<code>m_de_oil_vol</code>	Measured volume of buoyancy pumped in cc's
<code>c_de_oil_vol</code>	Commanded volume of buoyancy pumped in cc's

Pitch Vernier

Provided that the H moment is $6 \text{ mm} \pm 1$ (see "H Moment Calculation" and "Adjusting the H Moment" in section 3 of the *Slocum Glider Maintenance Manual*), the fluid movement from the ballast pump assembly provides the moment for changing pitch. (Water moves into the nose making the vehicle nose heavy when diving, similarly making the nose buoyant when rising.) To trim to the desired dive and climb angles, a lead screw drives the forward ~10 kg battery pack fore or aft as a vernier. The vehicle is designed to climb and descend at an angle of 26 degrees. During surfacing, the battery pack is moved all the way forward to better raise the tail out of the water for communications.

How to Configure

Factory configured. Please contact glidersupport@teledyne.com for assistance.

How to Test

From `lab_mode`:

1. Type `wiggle on`.
2. Type `report ++ m_battpos`.
3. Make sure that the pitch battery successfully completes a full extension and retraction without errors.
4. Type `wiggle off` and `report clearall`.



NOTE The length of a full extension and retraction depends on the type of buoyancy pump (shallow or deep) and battery (alkaline or lithium) as shown in the table below. The value of `f_battpos_safety_max` is specified in the glider's `autoexec.mi` file.

How to Evaluate Data

With a properly ballasted glider, positive pitch battery movement will result in decreased angle of attack and negative pitch battery movement will result in increased angle of attack (i.e., moving the battery toward the aft of the vehicle will lift the nose and moving the battery forward will bring the nose down).

Relevant Sensors

Sensor Name	Description
<code>m_battpos</code>	Measured position of the battery in inches
<code>c_battpos</code>	Commanded position of the battery in inches
<code>m_pitch</code>	Measured vehicle pitch
<code>c_pitch</code>	Commanded vehicle pitch

Altimeter

The Airmar altimeter, with a 0-100 m range transducer, is mounted on the front of the ballast pump assembly, and its electronics are supported on the cylinder of the ballast pump assembly. The transducer leads feed through a bulkhead connector on the front end cap. The transducer is oriented so that it is parallel to a flat sea bottom at a nominal dive angle of 26 degrees.

How to Configure

Factory configured. Please contact glidersupport@teledyne.com for assistance.

How to Test

In the lab:

1. Type `put c_alt_time 0` to ensure that the altimeter is updating as frequently as possible.
2. Listen closely to the altimeter to check for audible clicks similar to the ticking of a watch.
3. Type `report ++ m_altimeter_voltage`, and confirm that the altimeter voltage is not a fixed value.
4. Type `report clearall`.

How to Evaluate Data

The altimeter is normally used only while diving to prevent the glider from hitting the sea floor. When the altimeter is in use, the glider dives until it reaches a set distance from the bottom; then it begins to climb. This distance from the bottom is specified in either the glider's mission or ma files as an argument for the `d_target_altitude` command. Each buoyancy pump design and mission requirement will dictate the necessary value to achieve successful bottom avoidance. The suggested minimum values for `d_target_altitude` are:

Pump Type	Suggested minimum value for <code>d_target_altitude</code>
1000 m	12 m
350 m	6 m
150 m	4 m
50 m	2 m

Relevant Sensors

Sensor Name	Description
<code>m_altitude(m)</code>	Height above the bottom
<code>m_raw_altitude(m)</code>	Height above bottom, unfiltered
<code>u_alt_min_depth(m)</code>	<p>How deep the vehicle must be before the altimeter turns on</p> <p>A glider pilot may choose to increase <code>u_alt_min_depth</code> so that the altimeter will not turn on until it is below the thermocline or to reduce energy usage when the minimum water depth for the mission is known.</p>
<code>u_alt_reqd_good_in_a_row(nodim)</code>	How many <code>m_raw_altitude</code> readings are used to calculate the processed <code>m_altitude</code> measurement
<code>u_alt_filter_enabled(bool)</code>	Enable median filter depth for altitude
<code>u_alt_reduced_usage_mode(bool)</code>	<p>The glider calculates and uses the altimeter only when necessary.</p> <p>A glider pilot might decide to activate or deactivate reduce usage mode. Reduced usage will save energy; however, it may result in ineffective results in shallow water deployments.</p>
<code>m_water_depth(m)</code>	Calculates the depth of the water by adding <code>m_depth</code> and <code>m_altitude</code>
<code>u_max_water_depth_lifetime(yos)</code>	How long to use <code>m_depth</code> only in absence of new altimeter fixes

Carbon Fiber Hulls

Carbon fiber hulls are custom wound, compressible, engineered parts and should be handled with care. Any nicks in the paint finish should be examined to ensure that damage does not extend into the fiber winding. Careful inspection of interior and exterior finishes should be performed regularly, especially in the area of the sealing surface to the o-ring.

How to Test

Pull and maintain the recommended vacuum on the vehicle (6 inHg for shallow gliders and 7 inHg for deep gliders). For more details on adjusting the vacuum, see “Checking and Setting the Vacuum” in Section 4 of the *Slocum G3 Glider Maintenance Manual*.



NOTE The vacuum will fluctuate with the temperature.

How to Evaluate Data

The vacuum should be monitored. Note that the vacuum is affected by temperature (increases with cooling and decreases with warmth).

Relevant Sensors

m_vacuum—Measured internal vacuum

Payload Bay

The payload bay, also referred to as the "science bay," includes the Mother board and the science Persistor. They are mounted on a tray inside the payload bay in the central section of the G3 glider. There are three revisions of the Mother board: revision 1 (G1), included on all gliders shipped prior to November 2014 and shown in Figure 1-3; revision 2 (G2), included on some gliders shipped from November 2014 to September 2017 and shown in Figure 1-4; and revision 3 (G3), included on all gliders shipped since September 2017 and shown in Figure 1-5. The three revisions are different enough to require separate programs for each board.



Figure 1-3 Mother Board (G1)



Figure 1-4 Mother Board (G2)



Figure 1-5 Mother Board, Gen 3

In the release/target-science directory, there are now separate directories for G1 and G2. Copy the appropriate one of these to your science CF card. It is important to get all the changed files; just re-burning the app will not suffice.

The following files are different between rev 1 and 2:

- app/supersci.app vs. app/sbm2sci.app
- bin/sbmbloop.run
- bin/sbmbspew.run
- bin/u4stalk.run

The file bin/burnapp.run is the same for both revisions, but must be version 7.16 or newer in order to properly distinguish between the hardware revisions and program the correct file.

The information in proglets.dat is different between revisions:

- The "uart" for G2 is the same as the connector ID on the board, i.e. "uart = 1" refers to J1 on the board. Valid ports are 0–8 and 14.
- Power control is handled automatically for most devices. G2 values for the power bit control the power source that will be switched on automatically when any sensor using that supply is powered on:

For the power bit supply:

- 0: Vbatt (default, No-op)
- 1: 12 V regulated supply
- 2: 8.8 V regulated supply
- 3: SMB2 auxiliary power output
- 4: SMB2 high power output

The actual power supply used by a given sensor is determined by its wiring. If the setting in `proglots.dat` is incorrect, it will not power the sensor instead of applying wrong voltage. The default `proglots.dat` for both versions has example settings for that version. In general, you should not have to change the power bits setting for the G2 board.

Powering sensors with the Mother board G2 is quite different than for G1 and quite a bit more flexible. The following explains how to wire a sensor for a desired power output and how to configure `proglots.dat` to get the desired power and comms for that sensor.

Sensor power on SBMBii are the 10-pin, 3-mm connectors labeled J0–J6. For each port, Pin 1 is marked, on the opposite side of the latch.

The pinout is as follows:

Pin 1:	Ground
Pin 2:	Tx
Pin 3:	Rx
Pin 4:	RTS (ready to send)
Pin 5:	CTS (clear to send)
Pin 6:	Switched Power Out
Pin 7:	Switched Power In
Pin 8:	+8.8V regulated (Power bit 2)
Pin 9:	+15V regulated (power bit 1)
Pin 10:	VBatt (power bit 0)

To get the desired power to pin 6, the power source (one of pins 8, 9 or 10) must be connected to pin 7. When the associated power bit (2, 1 or 0 respectively) is enabled for that connector's UART (0-6), voltage from pin 7 becomes available at pin 6, powering the sensor. For example, for a generic data logger connected to J6 that requires battery power to trigger:

```
# generic data logger
proklet = logger
uart = 6
bit = 0 # power control
start_snsr = c_logger_on(sec)
```

On the J6 male connector, Pin 10 will be jumped to Pin 7, Pin 6 will be wired to the sensor power, and pin 1 will be wired to the sensor ground. If communications using `u4stalk` are desired, you may additionally wire the Rx, Tx, Cts, Rts lines.

Payload Bay Connectors

The forward plate and connectors to the payload bay will be familiar to glider operators and are covered in the *Slocum G3 Glider Maintenance Manual*.

New with the G3 glider is a large CPC connector at the aft of the payload bay. No longer does the operator need to reach inside the hull to attempt to mate the connectors. Additionally the aft plate has a cutout allowing for easy access to remove the CF card from the science Persistor if required.

Access to the inside of the payload bay can be gained by removing the forward guard plate. There is interchangeability between the G2 and G3 glider payload bays.

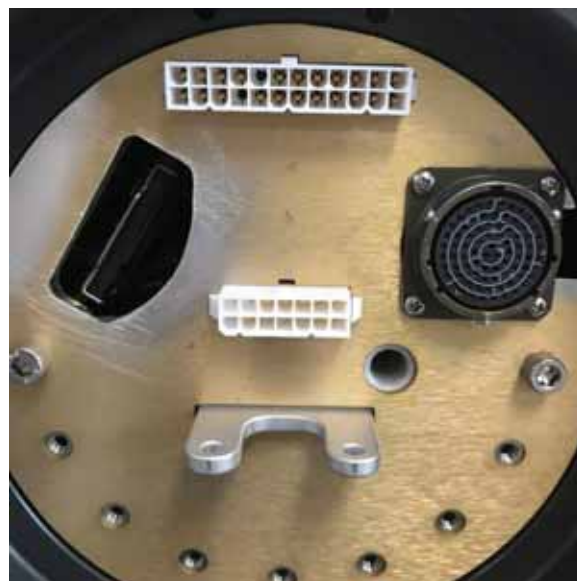


Figure 1-6 G3 Payload Bay Connectors

Payload Bay Computers and Sensors

A proklet controls each science sensor. The proklet sets the power bit and uart on the payload bay controller and expects formatted data from each sensor.

How to Test

From GliderDOS, type `loadmission sci_on.mi`. All payload bay sensors will be turned on and their data will be printed to the terminal. When it has been determined that all sensors are reporting data properly, type `loadmission sci_off.mi` to turn off the science sensors.

Each sensor can be addressed directly using “u4stalk.” Contact glidersupport@teledyne.com for details.

How to Evaluate Data

The method of evaluating the data is unique to the sensors contained within each payload bay.

Relevant Sensors

The relevant sensors are unique to each payload bay.

Conductivity, Temperature, and Depth (CTD)

A Slocum glider does not require a CTD; however, a typical sensor package contained in the payload bay on the glider is a Sea Bird (SBE) pumped conductivity, temperature, and depth package. The CTD sensor is delicate and should be protected from abuse.



NOTE To ensure that the pumped CTD does not get damaged in lab or simulation SBE ensures the impeller pump does not operate if a minimum conductivity is not present. If operation is going to be done in fresh water lake or pond and accurate CTD readings are required this setting can be adjusted. Contact glidersupport@teledyne.com for details. Note gliders configured for fresh water do not have protection from damage when run “dry”.

The CTD is not used for flight, because the glider has an independent pressure sensor for dynamic flight control. It can be used in an emergency by changing the sensor `u_use_ctd_depth_for_flying`.



WARNING The Seabird Electronics (SBE) pumped CTD should not be run dry for more 30 seconds at a time.

The Slocum Glider data page is located at <http://www.seabird.com/products/profilers.htm>. Scroll down to the bottom of this page to access the link to the data page.

How to Configure

Manufacturer and factory configured

How to Test

The steps to turn on the science sensors listed above can be used to evaluate the CTD data. When dry and in the lab, the measured conductivity will be 0 and can sometimes go negative. Pressure will fluctuate near 0 dbar. Temperature should closely match ambient temperature.



NOTE The CTD manufacturer provides plugs to keep contamination out and moisture in the cell. These plugs should be used during storage, but they must be removed before deployment.

How to Evaluate Data

See manufacturer's documentation.

Relevant Sensors

Sensor Name	Description
sci_water_cond	units (S/m)
sci_water_temp	units (degrees C)
sci_water_pressure	units (bar)

Argos Satellite Platform Transmitter Terminal (PTT)

The Seimac X-Cat PTT is used in recovery situations to transmit last-known GPS positions, when available. The Argos service also provides periodic surface locations accurate to approximately 100 meters. See "Argos Data Format" on page H-2.

How to Configure

Factory configured

How to Test

A full system test can be performed by putting the glider outside with a clear view of the entire sky for three hours in `lab_mode`. The Argos account for the vehicle can then be checked for data. Locally, a detection device or receiver can be used to confirm transmissions.

How to Evaluate Data

A utility to decode Argos data can be found at:

Forum /glider/windoze/production/windoze-bin/

Relevant Sensors

```
c_argos_on(enum) 0  # <0 PTT is always turned off, even at surface
                   # 0 PTT powered off, but can be auto turned on at surface
                   # >0 PTT is powered on and transmitting:
                   # 1 no diagnostic output
                   # 2 output xmitted chars to MLOG/TERM
                   # 3 output xmitted/recvd chars to MLOG/TERM
```

Catalyst

When alkaline batteries are used. A catalyst is used to recombine hydrogen and oxygen into water to reduce the risk of explosion should the batteries begin to outgas. The reaction is exothermic and the catalyst may become hot. This item does not need periodic replacement.



WARNING If the glider contains alkaline batteries, there is a small but finite possibility that batteries of alkaline cells will release a combustible gas mixture, especially if the batteries are exposed to water or sea water and/or shorted. This gas release generally is not evident when batteries are exposed to the atmosphere, as the gases are dispersed and diluted to a safe level. When the batteries are confined in a sealed instrument, the gases can accumulate and an explosion is possible. Teledyne Webb Research has added a catalyst inside of the glider to recombine hydrogen and oxygen into water, and the glider has been designed to relieve excessive internal pressure buildup by having the hull sections separate under internal pressure.

Teledyne Webb Research knows of no way to completely eliminate this hazard. The user is warned, and must accept and deal with this risk in order to use this instrument safely as so provided. Personnel with knowledge and training to deal with this risk should seal or operate the instrument.

Air Pump System

The air bladder in the flooded tail cone is used to provide additional buoyancy on the surface for assisting in lifting the tail from the water line for communications. It is inflated using air from the hull interior and can provide 1400 ml of reserve buoyancy. The air pump is mechanically switched off when the differential pressure (between the air bladder and the internal hull pressure) becomes 6.25 PSI. This has been factory set. When surfaced, the glider equilibrates with the tail elevated, and the boom holds the antenna clear of the water. This air is vented back into the vehicle via a latching solenoid valve for descent.

How to Configure

Factory configured. The air systems differential pressure switch is set to 3 in/Hg during manufacture.

How to Test

To activate the air pump and inflate the external air bladder in the lab, type put c_air_pump 1. To open the solenoid and deflate the external air bladder, type put c_air_pump 0. Full inflation should only be done with the cowling installed and will take between 3-10 minutes. The switch will stop the air pump when the pressure differential reaches 1.5 to 2.5 in/Hg. The glider's

internal vacuum will increase by 1.5 to 2.5 in/Hg as atmosphere from the vehicle is moved into the external bladder. The pump inflation typically should be complete in under 5 minutes. Longer duration inflation times may be an indication of system trouble.

How to Evaluate Data

The air bladder will be activated per the above sensors at the surface. `m_vacuum` will increase when functioning properly.

Relevant Sensors

- `c_air_pump`
- `m_air_pump`
- `m_vacuum`

Vehicle Controller

The glider's functions are controlled by a Persistor CF1 single-board miniature computer that is based on a Motorola 68338 processor. This board has low current consumption capability and supports the use of compact flash cards and miniature hard drives that are able to store large amounts of data. Controller code is written in C and architecturally is based on a layered single thread approach where each task is coded into a behavior. Behaviors can be combined in almost any order to achieve flexible, unique missions. Each device is labeled as a sensor and is logged every time that the value changes during a mission. Data is retrieved from the glider as a binary file and is post-parsed into a matrix that allows the user to easily construct graphical views of vehicle performance or scientific data. A subset of the sensors can be chosen as a science data package so as to reduce surface radio transmission time. The Persistor can have any number of pre-written missions (text files) in its memory that can be called, or new missions can be written, transmitted to the glider, and run. Missions can be altered to change the depth of inflections, send new GPS waypoints to the glider, or for many other reasons. For additional reading regarding the construction of binary data or the .dbd family of files and their contents, visit:

Forum glider/windoze/production/src/doco/specifications/

Forum glider/windoze/production/src/doco/how-to-operate/

How to Configure

G3 gliders are factory configured with production code. Production code is updated frequently and should be maintained. The pilot should review the updates and features added during each release:

Forum glider/windoze/production/readme.txt

Instructions on updating the glider to the most recent production release can be found here:

<ftp://ftp.glider.webbresearch.com/glider/windoze/production/src/doco/software-howto/updating-all-glider-software.txt>

How to Test

Activate glider with green plug and from GliderDOS prompt, type `ver` to confirm release revision.

How to Evaluate Data

The G3 glider monitors and records data from hundreds of glider sensors or variables, along with hundreds of potential science sensor variables. The user controls the vehicle by manipulating some of these sensors and by writing missions that define hundreds of arguments within well-structured behaviors. Data from the glider often interact in complicated and sometimes non-intuitive fashions and need thoroughly investigated before the vehicle's behavior is understood.

Teledyne Webb Research provides a number of tools for viewing and processing data. Each dockserver also hosts a datasever, which transfers raw glider binary data into a database. This database can be viewed using the a data visualizer, and any available glider sensor can then be plotted against time. Teledyne Webb Research also provides the tools to convert raw data to ASCII so that a user can format and publish data as desired. These tools can be found at:

Forum /glider/windoze/production/windoze-bin/

or

Forum /glider/windoze/production/linux-bin/

Relevant Sensors

See masterdata at:

Forum /glider/windoze/production/masterdata

Hardware Interface Board

The flight Persistor is mated to the main driver board that interfaces with all of the sensors, communications, and drive mechanisms. The board nominally runs on 11–15 volts DC. A section of the board is dedicated to a hardware abort mechanism. As a recovery precaution for errant events, a timer (set to 32 hours in the factory) is reset (`COP_tickled`) every time there is either a GPS fix or a keystroke while in GliderDOS, which would indicate that the glider is safely on the surface. If the timer elapses, however, the following circuits will be energized from the emergency circuit, forcing the glider to surface:

- Air pump
- Argos PTT
- Burn wire for the jettison weight

How to Configure

Factory configured. Pitch and buoyancy motors can be activated via controls located on the top of the board.

How to Test

See the glider's wiring diagram. To receive the latest version of the Slocum G3 Glider's wiring diagram, contact glidersupport@teledyne.com.

The cop tickle circuit can be changed by moving a jumper. Please contact Glider Support if testing is needed.

For further information regarding vehicle aborts and the function of the cop tickle, see the following support documentation:

Forum /glider/windoze/production/src/doco/how-it-works/abort- sequences.txt

Attitude Sensor

A precision navigation compass and attitude sensor monitors the bearing, pitch, and roll of the glider. These inputs are used for dead reckoning the vehicle while under water. Recalibrating the compass may be necessary at times (depending on the magnetic anomalies of the usage area). See the *Slocum Glider Maintenance Manual* for compass calibration instructions.

How to Configure

A calibration utility and its instructions can be found here:

<ftp://ftp.glider.webbresearch.com/clients/compass-calibrator/files-for-windows/>

How to Test

The compass is very susceptible to interference caused by ferrous materials and magnetic fields that can be present in electronics, vehicles, and buildings; for this reason, the calibration procedure should be performed as far as possible from any of these sources of interference. Type `report ++ m_heading m_pitch m_roll`, rotate the vehicle, and independently verify output.

How to Evaluate Data

The relevant sensors should reflect the orientation of the vehicle at all times.

Relevant Sensors

- `c_att_time(sec)`—0 turns on as fast as possible if off (-1)
- `m_heading`
- `m_pitch`
- `m_roll`

Global Positioning System (GPS)

The glider's GPS updates the unit's position every five seconds while the vehicle is on the surface. Output from the GPS is in the RMC NMEA 0183 format. The GPS can also be used to update the system's internal clock, if necessary.

How to Configure

Factory configured

How to Test

The glider must be outside with a clear view of the sky. If the vehicle has moved a great distance or has not been turned on for some time, a new GPS fix may require 5-10 minutes to be acquired. Type `put c_gps_on 3` to print GPS data to the screen for verification. Optionally an operator can type `where` to see the output of the "where dialog."

How to Evaluate Data

At each surfacing or when the command "where" is issued, the glider will print the surface display. Four types of locations are printed with the current time:

```
Current Time: Tue May 31 12:52:37 2011 MT: 307186
```

```
DR Location: 4116.681 N -7037.206 E measured 183.928 secs ago
```

```
Dead Reckoned or calculated position and age.
```

```
GPS TooFar: 69696969.000 N 69696969.000 E measured 1e+308 secs ago
```

```
GPS Invalid: 4116.696 N -7037.202 E measured 293.261 secs ago
```

```
Possible good location based on the sensor u_gps_reqd_valid_fixes (default 6)
```

```
GPS Location: 4116.681 N -7037.206 E measured 185.342 secs ago
```

```
This is the last good recorded GPS location.
```

Note that each of these values has a time stamp to their age.

The vehicle calculates a maximum distance it can travel based on the sensor `u_max_water_speed`. This GPS location can be valuable if the glider is retrieved from the water and is moving with a vessel.

Relevant Sensors

There are many sensors for the GPS. The two most important are:

- `m_gps_lat`
- `m_gps_lon`

To view the other sensors relevant to the GPS, refer to the masterdata at:

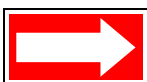
Forum glider/windoze/production/masterdata

Iridium Satellite Telemetry

The Iridium 9522b new modem type bidirectional satellite modem is on the lower electronics tray with a low-noise amplifier (LNA) switching board for the antenna, which is shared with the GPS. The LNA switch allows the IR modem to share its antenna with the GPS.

How to Configure

The glider's Iridium phone and SIM card are configured at the factory using the PicoDOS command `talk iridium`. Contact glidersupport@teledyne.com for assistance depinning SIM cards or checking phone configuration.



NOTE Depinning SIM cards for the Iridium phone is normally a factory configuration and is only provided to users installing their own card or changing services. For more information on depinning, see the *Slocum G3 Glider Maintenance Manual*.

The primary and alternate phone number to dial are entered into the glider's `autoexec.mi` file in the config directory. The `put` command can be used to change and test a phone number temporarily. For Rudics connections, the dockserver must also be configured for network.

How to Test

Place the glider outside, with a clear view of the sky. Type `callback 0 0` to dial the primary number immediately. Type `callback 1 1` to call the alternate number in one minute. When satisfied, type `callback 30 0` to call back the primary number in 30 minutes. Thirty minutes is the maximum allowable callback time.

How to Evaluate Data

Connection time can vary depending on cloud cover and availability of satellites. A utility can be downloaded to confirm satellite coverage at:

<http://gpredict.oz9aec.net/>

Relevant Sensors

`m_iridium_signal_strength(nodim)` # iridium received signal where 0 equals no signal and 5 is the maximum signal strength.

See each vehicle's `autoexec.mi` file for the primary and alternate numbers to be dialed by the Iridium phone.

To view the other sensors relevant to the Iridium phone, refer to the masterdata at:

Forum `glider/windoze/production/masterdata`

Radio Frequency (RF) Modem Telemetry

When connected to the flight Persistor, the FreeWave 900 MHz radio modem is used for local high-speed communications with the glider. Due to its very high baud rate, no cable is required for communication and all in lab communications are facilitated through RF comms. See "FreeWave Configuration" on page F-1.

How to Configure

The FreeWave slave in the Slocum glider is factory configured to call all master FreeWave transmitters. The FreeWave Master shipped with each glider is configured to communicate with only one glider.

To configure a master to another glider connect the master unit to a computer with a terminal program, such as ProComm or HyperTerminal, configured for 19200 N-8-1. Power on the master, and press the button on the back of the master. From the FreeWave menu, edit the call book (option 2), and enter the seven-digit slave number from the `autoexec.mi` file from the desired glider into one of the 10 available slots. From the FreeWave menu, assign the call book entry to `call (C 0-9)`; do not use `call all(A)`.

How to Test

When the glider and master are powered, the red carrier detect (CD) light on the front of the master unit will turn green.

Communications

When first powered on, the FreeWave radio modem is set to communicate with the flight Persistor. Communications can be established with the science Persistor from PicoDOS and GliderDOS.

While in PicoDOS, typing `consci` enables direct communication with the science Persistor via a hardware controlled connection. In order to resume communication with the flight Persistor, the carrier must be disconnected for three seconds, which can be accomplished by disconnecting power to the FreeWave master for about 10 seconds.

While in GliderDOS, typing `consci` enables direct communication with the science Persistor via a software-controlled connection known as the *clothesline*. In order to resume communication with the flight Persistor, enter the `quit` command.

How to Configure

Contact Glider Support.

How to Test

Type `consci` from PicoDOS and establish communication with science. The prompt should change from `(GPICO)C:\>` to `(SCI)C:\>`.

Pressure Transducer

G3 gliders are outfitted with Micron 2000 PSIA strain gage transducers, which are used for vehicle control and dead reckoning. The stainless steel transducer is ported through the aft cap and is isolated from the aluminum aft cap by a PEEK fitting.

How to Configure

Factory calibrated. Surface pressure values can fluctuate, but the `zero_ocean_pressure` command can be used to reset the zero pressure voltage. It is recommended that the through port be examined and cleaned of contamination or blockage between deployments.

How to Test

In the field, CTD pressure can be compared against flight pressure transducer values. `m_depth` can be reported in the lab and will fluctuate.

How to Evaluate Data

Plot `m_depth` against `sci_water_pressure`.

Relevant Sensors

- `m_depth`
- `m_depth_state`
 - 99 ignore
 - 3 hover
 - 2 climbing
 - 1 diving
 - 0 surface
 - 1 none

Leak Detect Sensor Boards

Each glider is equipped with three leak detect sensor boards. The aft leak detect sensor is located on the bottom of the aft cap. The forward leak detect sensor is attached to the bottom of the front cap. The payload bay leak detect is tied to the forward leak detect sensor circuit. These sensors normally report 2.5 volts. If exposed to moisture, the circuit is shorted, and any value below the masterdata default entry of 2 volts will cause an abort for leak detect.

New with the G3 glider payload bay is a leak detect sensor. This sensor is wired in parallel to the forward leak detect circuit:

```
sensor: m_leakdetect_voltage_forward(volts) 0.0
```

How to Test

Type `report ++ m_leakdetect_voltage m_leakdetect_voltage_forward`. Both values should be greater than 2.3. The glider will abort missions for leak detect with any value less than 2.

How to Evaluate Data

If the leak detect sensors are less than 2.3, there is likely water in the vehicle as the result of a leak. Vehicle recovery and careful handling is recommended.

Relevant Sensors

Sensor Name	Description
<code>f_leakdetect_threshold(volts)</code>	Any value of <code>m_leakdetect_voltage</code> below this threshold is considered a leak.
<code>m_leakdetect_voltage(volts)</code>	Voltage that was reported by the aft leak detect sensor. The lower the voltage, the worse the leak.
<code>m_leakdetect_voltage_forward(volts)</code>	Voltage that was reported by the forward leak detect sensor. The lower the voltage, the worse the leak.
<code>m_leakdetect_voltage_science(volts)</code>	Voltage that was reported by the payload bay detect sensor. The lower the voltage, the worse the leak.
<code>m_leakdetect_voltage_stack_on(volts)</code>	Voltage that was reported by the stack on bay leak detect sensor. The lower the voltage, the worse the leak.

Air Bladder

The glider's 1400 cc bladder, which is inflated by the air pump system, provides buoyancy and stability and lifts the antenna support out of the water while the glider is surfaced. Although the bladder is ruggedly constructed, care should be taken to have the aft tail cowling in place when the bladder is filling to prevent it from over-inflating. With the cowling in place, the bladder is supported as it inflates until the pressure switch shuts off the air pump. Likewise, it is important to deflate the air bladder when removing the aft tail cowling, as it will be hard up against the cowling.

How to Test

See the air pump system test. The bladder should be inspected for damage before and after each deployment.

How to Evaluate Data

See the air pump system test.

Relevant Sensors

See the air pump system test.

Burn Wire and Jettison Weight

The glider is equipped with an emergency abort system. In the event that the vehicle is unable to surface during a mission, a battery-activated corrosive link will release the ~500-g stainless steel spring-loaded jettison weight, forcing the glider to surface. This burn process lasts for a few seconds in salt water and approximately four hours in fresh water. The jettison weight is positioned beneath the tail fin assembly and is held in place by a 20 AWG Inconel burn wire that is mated and sealed to a single-pin Mecca connector on the aft end cap.



NOTE Activating the burn wire in air will have no effect, as it takes ions in the water to complete the return path to ground. The burn wire may be compromised if the glider is still wet but not submerged and the ejection weight is activated. See the *Slocum Glider Maintenance Manual* for more details.

How to Test

The ejection weight burn wire electronics can be tested in the lab by measuring voltage to the burn wire while the mecca connector is unplugged.

1. While in GliderDOS, type `lab_mode off`.
2. Disconnect the supply lead to the drop weight at the Mecca connector.
3. Connect the digital voltmeter between the supply lead and the tail boom.
4. Type `put c_weight_drop 1`.
5. Verify that the voltage is at least 5 volts.
6. Put `c_weight_drop 0`
7. Put `m_weight_drop 0`
8. Type `exit pico`.
9. Verify that the voltage is 0.
10. Reconnect the drop weight supply lead.

Relevant Sensors

Sensor Name	Description
<code>c_weight_drop(bool)</code>	Values greater than 0 will drop the weight.
<code>m_weight_drop(bool)</code>	The active state for the weight drop which will be held through power cycle until manually set back to 0.
<code>u_abort_min_burn_time(sec)</code>	During the abort sequence when a glider is having trouble getting to the surface, never drop the weight before this time.
<code>u_abort_max_burn_time(sec)</code>	During the abort sequence when a glider is having trouble getting to the surface, always drop the weight after this time.
<code>f_crush_depth(m)</code>	When the glider gets crushed, this is used to determine when to eject the weight.

Power Port

A subcon connector is used to switch or supply power to the glider. When the stop plug (red) is inserted or the connector end is empty, there is no power applied to the vehicle. This is done so that power can be removed from the system without special tooling. By powering the glider via the umbilical, there is no need for an internal switch, which could generate a spark. To power the glider on either, use the provided external power cable or insert the go plug (green). Do not exceed 16 volts. The umbilical is accessible external to the glider aft tail cone. The G3 glider should always be powered down properly by using the exit command. The G3 glider battery management board will indicate an improper shut down by emitting a beep.

How to Test

The glider is activated by inserting either wall power or the green plug. Removing the green plug and installing the red plug powers off the glider and should only be done after the exit command is issued and accepted by the glider. Only in an emergency should, the green plug or wall power may be removed without a software command. This can result in corrupted files. A `chkdsk` should be performed on the compact flash (CF) cards and, if necessary, the cards should be reformatted. When improperly shut down the emergency circuit will ultimately activate the burn wire and emergency circuit.

Battery Management System (BMS)

The BMS board provides several new power management features:

- A latch circuit that keeps the emergency circuits powered in the event of an enable circuit failure at sea. When the glider application launches, it enables the latch. Once enabled,

the emergency battery will provide power in the event of the main batteries turning off. To properly turn the vehicle off, the battery must be unlatched. Properly exiting the glider app will accomplish this. In the event that the emergency battery is left latched when the main battery is disconnected, a buzzer will sound a warning <pattern> to indicate to the user that the emergency battery is still active. This is intended to avoid draining the emergency battery in the lab if the user fails to exit properly.



NOTE If the BMS is emitting the buzzer warning, the glider must be powered on and exited properly to turn off the buzzer.

- Support for the measuring of the main battery and emergency battery voltages separately. This allows for the emergency battery health to be better estimated by the user.
- An auxiliary high power switch allows for custom user applications.

```
# G3 specific coulomb Battery Management System(BMS)
sensor: m_bms_pitch_current(amp)           0.0 # instantaneous current
sensor: m_bms_ebay_current(amp)           0.0 # instantaneous current
sensor: m_bms_aft_current(amp)           0.0 # instantaneous current

sensor: m_bms_main_battery_voltage(volts) 0.0 # main battery voltage
sensor: m_bms_emergency_battery_voltage(volts) 0.0 # emergency battery
voltage
sensor: m_bms_battery_in_use(bool)        0.0 # 0-main battery in use, 1-
emergency battery in use
sensor: m_bms_emergency_battery_latch(bool) 0.0 # 1-emergency battery
latch enabled, operating state
```

Batteries

The G3 glider supports the following batteries:

- Legacy alkaline batteries, nominal 15 V
- Legacy 3s lithium batteries, nominal 12 V
- 4s lithium primary batteries, nominal 15 V, new with the G3 glider
- Rechargeable 15 V batteries, new with the G3 glider

Contact glidersupport@teledyne.com for assistance in estimating glider endurance and which chemistry is appropriate for your application.

Emergency Battery

The addition of the BMS system has given the G3 glider the ability to individually draw power from each pack. This has allowed for the switch to a standardized Alkaline emergency battery that can be used with any type of primary battery (Alkaline, Lithium primary and lithium secondary). The switch to an alkaline emergency battery makes the system more robust because it removes the need for a safety circuit on that pack as well as giving the user the ability to monitor the charge state of that pack by monitoring the voltage of the pack. This alkaline emergency battery nominal full capacity voltage of 12.8V and deplete down to 9.6V as it is used. This starting voltage is lower than all of the G3 battery options for the main packs so it will not be consumed until the main packs are depleted below the nominal full capacity voltage of the emergency pack.

The emergency battery can be installed in either the nose of a G3 glider or in the payload bay depending on ballasting needs specific to different glider configurations.

Rechargeable Battery Packs

There are two rechargeable battery packs, the pitch pack shown in Figure 1-7 and the aft pack shown in Figure 1-8. Each set provides 10.0–16.8 V nominal, a capacity of 3 kW·h total for two packs and a charge time of 12 hours.



Figure 1-7 Pitch Rechargeable Battery Pack, Forward Side Up



Figure 1-8 Aft Rechargeable Battery Pack, Aft Side to Left

Battery Charger

The Teledyne Webb supplied battery charger is shown in Figure 1-9.



Figure 1-9 Battery Charger with Charging Cable

The battery charger display and connectors are the following:

- Red #1: USB comms (115200, no flow control)
- Red #2: Charging current display (amps)
- Gray #0,1: Aft pack charge and comms channel 0 & 1
- Gray #2: 1st Pitch Pack wircomms channel 2
- Gray #3: 2nd Pitch Pack comms channel 3

Pitch Battery Ports

The pitch battery port is shown in Figure 1-10.



Figure 1-10 Pitch Battery Ports

The pitch battery ports are the following:

- Red #1: Battery out (not used for charging)
- Red #2: Comms (not required for charging)
- Red #3: Charging port and cooling fan

Battery Charge Indicator

Battery charge indicator bars are available to view on the pitch pack only as shown in Figure 1-11.

Battery Charging Instructions

When charging the batteries, use only the Teledyne Webb supplied battery charger. The battery charger is not GFCI protected. If using the battery charger near water, ensure that the charger is connected to a GFCI outlet or use the GFCI cable.



Figure 1-11 Battery Charge Indicator Bars

1. Switch the battery charger Power switch to Off.
2. Connect the AC cable to the battery charger.
3. If the batteries are already installed in the glider, connect the LPIL-5 cable to the glider end cap and go to Step 8.
4. If charging the batteries outside of the glider, the aft, pitch and extended pitch battery packs can be charged simultaneously. Connect the LPIL-5 cable to the charger adapter box.
5. Connect the 14-pin cable from charger adapter box to the aft battery pack.
6. There are two identical pairs of cables for charging the pitch and extended pitch battery packs. They each consist of a 2-pin charge cable and a 6-pin comms cable. Plug the 2-pin charge cable from one pair into the pitch battery pack charging port (see Figure 1-10 on page 1-35). Plug the 6-pin comms cable into the pitch battery pack comms port if comms are desired (remove the loopback cable during charging if installed).
7. If charging the extended pitch battery pack, plug the 2-pin charge cable from the other cable pair into the extended pitch battery pack charging port. Plug the 6-pin comms cable into the extended pitch battery pack comms port if comms are desired.
8. Verify that the air vents on the battery charger are not obstructed.
9. Switch the battery charger Power switch to On. The fans on the battery packs should begin operating.

While the battery packs are charging, the battery charger will display the charge current which will vary as communications is established with the battery packs. When charging, an initial 1-amp draw should soon increase to 12 amps. Within a minute or two the current should stabilize. The current will be slightly lower on a nearly discharged battery than a full one. During charging the charging cables may feel warm to touch. As each individual pack finishes charging, the charger current will drop. Once all the battery packs are charged, the current should be around 0.5 amp.

Optional Comms

If the optional comms are to be used when charging the batteries, the baud rate is 115200 with no flow control. One command is available:

\$R,n

where "n" is the channel and can be 0–3. Channels 0 and 1 are for the aft pack which contains 6 cells each. Channel 2 is typically the first pitch pack and channel 3 is the extended pitch pack. Each pitch pack contains 10 cells.

While charging the battery packs, the following data are available:

- Batt #
- Status
- LTCO
- Voltage (mv)
- Current (mA)
- Temperature (K)
- Serial number
- Remaining capacity (maH)
- Full charge capacity (mAh)
- Time to full charge (min)
- Cycle count
- Thermistor (raw)
- Thermistor (V)
- Manufacturer name
- Device name
- Device chemistry

G3 Primary Batteries

A change has been made to the lithium primary batteries on the G3 Gliders to go from a 3S configuration to a 4S configuration. This means that the nominal voltage of the packs has gone from 10.8V to 14.8V. The higher voltage means that the glider will be able to continue to fly after it hits the knee on the battery packs where the voltage drops to 11.5V near the end of their life. This allows the glider to use more of the total energy in the packs, but also allows the user to get much closer to the true end of life of the packs. This knee typically happens at approximately 85% of the total energy available in the packs. For safety, we suggest that the users use this energy after the knee as a reserve tank.



CAUTION The 3s battery design cannot support glider operations after the voltage drop – The “shelf energy” cannot be used for reserve with this design.

Putting 4 cells in series on the 4S batteries results a higher voltage but also results in a lower total amphr rating than the 3S packs despite actually having more energy. The sensor `f_coulomb_battery_capacity` and undervolts behavior argument should be adjusted to match the type of battery that is installed. The figure below shows a comparison between the different types of packs available and uses Watt Hours to normalize the total energy.

	Total Non-derated Amphrs (amphr)	Total non-derated Watt Hours (WH)	Recommended <code>f_coulomb_battery_capacity</code>	Recommended Undervolts (V)
Alkaline Standard Pack	168	2520	120	10
3S Primary Standard	780	8424	720	9.8
3S Primary Extended	1140	12312	1050	9.8
4S Primary Standard	600	8600	550	12
4S Primary Extended	870	12500	800	12
Lithium Rechargeable Standard	234.6	3300	215	12
Lithium Rechargeable Extended	326.4	4700	300	12



NOTE The figure above lists the total theoretical energy in each pack without any derating. It also lists a suggestion for `f_coulomb_battery_capacity` based on average deratings for temp, age, and storage as well as real world flight data. Please see battery manufacturers derating instructions for specific applications.

Installing the Battery Packs

Connect the pitch battery pack to the FWD harness. Connect the aft battery pack to the AFT harness 14-pin connector. Install the emergency battery pack in the nose of the glider.



NOTE Rechargeable battery packs can be installed in G2 gliders. To do so, refer to the applicable G2 battery pack installation instructions and ensure that the proper adapter cables are used. Figure 1-12 shows a G2 battery connection on the Main board. Rechargeable batteries are treated the same as alkaline batteries in a G2 and do not require the in-line connection used for lithium batteries.

The Main board has two main power supplies, one for alkaline batteries and one for lithium batteries as shown in Figure 1-12 and Figure 1-13 for the G2 and the G3, respectively. To change the battery chemistries, swap the location of the connector at the starboard aft end of the Main board. No harm will result if the connections are improperly made; however, the glider will not operate properly.



Figure 1-12 Connector at Starboard Aft End of the Main Board (G2)



Figure 1-13 Connector at Starboard Aft End of the Main Board (G3)

Rechargeable Battery Operational Considerations

Some rechargeable battery operational considerations should be noted as follows:

- The rechargeable battery packs will turn themselves off at 10 V.
- It will take 32 hours for the emergency circuit to become active when a battery pack turns off, despite the alarm sounding on the BMS.
- Fully charged battery packs will provide 3 kW·h or 232 coulomb amp/hr.

In addition, when fully charged and installed, the following sensor should be set:

```
sensor: f_coulomb_battery_capacity(amp-hrs) 232.0 # nominal battery capacity
(standard set)
```

And missions should be set to abort at:

```
behavior: abend
b arg: undervolts (volts)          12.1    #Critical Rechargeable battery
                                      #abort (Not determined)
                                      #batteries will turn off at 10v
```

Shipping Requirements

When shipping the battery packs, only a certified carrier should be used. In addition, the batteries must be discharged to below 30%. Class 9 HAZ, in equipment UN3481 and outside of equipment UN3480 also apply.

Fin

The self-calibrating tail fin assembly contains a strobe, the vehicle's rudder and its three antennas:

- ARGOS 401 MHz
- FreeWave RF modem 900 MHz
- Combined GPS 1575 MHz and Iridium 1626 MHz

This tail fin is rugged enough so that it can be used to handle and manipulate the glider.

How to Configure

Factory configured

How to Test

From `lab_mode`, wiggle on/wiggle off. Range ± 25 degrees.

How to Evaluate Data

With a properly ballasted glider, positive fin movement will result in increased heading values.

Relevant Sensors

- m_fin
- c_fin
- m_heading
- c_heading
- m_roll

Tail Fin

The G3 glider fin, which is shown in Figure 1-14, has been designed for robust, trouble free operation, strength, and resilience. It is composed of a titanium boom, a motor housing, an antenna feed through, and a plate which is bolted to a peek mast using a barrel nut fastener system. Field serviceable items may be quickly and simply replaced. The strobe is integrated into the leading edge of the mast encased in the orange noryl housing that is shared with the quad helix GPS/Iridium Antenna.

Four subsystems are integrated into the tail fin and should be independently tested:

- Communications
- Steering
- Strobe illumination
- Leak detect

All these subsystems should be tested as detailed in the 4095-FCP Functional Checkout Procedure and briefly described below.



Figure 1-14 G3 Glider Fin

Communications Subsystem

The communications subsystem consists of three modules: an Iridium module, an Argos module and a Freewave module.

The Freewave module can be tested by powering the vehicle and ensuring that a carrier is detected by the shore side Freewave modem (indicated by a green CD light).

The Argos module can be tested by providing the glider with a clear view of sky and allowing it to transmit for 1-3 hours. Due to the polar orbits of the Argos satellite network, time between passes at any particular location on the earth may vary. Log into the CLS America website using the login credentials created during the account generation process and verify that the Argos ID corresponding to the glider has broadcast hits successfully. If you would like to test Argos communications independently of the Argos network, an RF detector or receiver sensitive to 401.650 MHz \pm 30 kHz may be used. Detectors range in cost and performance from a small, inexpensive passive transducer (chirper), to the hand held TSUR-400 Argos Receiver which will show the transmitter unit ID, data bytes received and some additional diagnostic information.

The Iridium module can be tested by placing the glider in an outdoor environment with a clear view of the sky. Attempt to make phone calls and verify that the calls are successfully connecting to the base station/dockserver. It may take several attempts for the glider to acquire a satellite. While the glider is making phone calls, take note of the value displayed after AT+CSQ during the dialing attempts. This number is equivalent to the bars on a cell phone, and with a clear view of the satellite, it should be either 4 or 5.

Steering Subsystem

To verify that the steering subsystem is performing properly, boot the vehicle into LabMode. Put the glider into wiggle (wiggle on) and report commanded and measured fin position to the screen (report ++ m_fin c_fin). Verify that c_fin drives m_fin, and that no oddities or warnings are displayed from the fin. Further verify that when m_fin is greater than 0, the rudder is deflected toward the starboard side of the vehicle, and that when m_fin is less than 0, the rudder is deflected toward the port side of the vehicle.

Strobe Illumination

To test strobe illumination, boot the glider in LabMode, activate the strobe (strobe on) and verify that the strobe periodically flashes (every 3–5 seconds). Once verified, deactivate the strobe (strobe off).

Fin Leak Detect

The leak detect reading measured in the fin can be read by polling the value of m_digifin_leakdetect_reading from GliderDOS or LabMode. The value should hold near 1025 when no leak has occurred. When a leak occurs and water is present, the value will drop. A value below 1018 is considered to indicate a leak.

Tail Fin Maintenance

After deployment, the tail fin should be rinsed with fresh water to prevent salt crystals from inhibiting motion during the next power up. In the event that the rudder has become damaged, it can be easily replaced.

To replace the rudder:

1. Remove the two Phillips head screws securing the mast top to the mast as shown in Figure 1-15.
2. Hold on to the rudder with one hand and lift the mast top away from the fin with the other as shown in Figure 1-16.



Figure 1-15 Removing the two Phillips Head Screws



Figure 1-16 Removing the Mast Top

3. Lift the rudder off the magnetic coupling as shown in Figure 1-17.
4. Install a new rudder and reassemble in reverse of disassembly.



Figure 1-17 The Rudder Removed from Tail Fin



CAUTION The loop at the aft end of the boom beneath the rudder is intended as a small load tie off point. It may be used for tying a buoy for trial dives or for manipulating the vehicle using a boat hook or gaff while it is in the water. It should never be used for lifting or suspending the vehicle out of the water. It is not a lifting point.



CAUTION The four titanium socket head cap screws at the base of the fin boom should not be loosened or removed without explicit instruction from TWR. These screws provide access to the feed-through plate which connects communications subsystem antennas to the vehicle communications board. Leak testing is typically required upon gaining access to the feed through plate.

Sacrificial Anode

The outside of the aft and forward end caps are fitted with sacrificial zinc anodes to prevent corrosion of the glider's exposed aluminum and stainless steel components. The anodes should be checked for continuity to ground on a regular basis to ensure proper protection against

corrosion, and they need to be replaced periodically. It is important to take note of any scratches to the glider's anodizing, as exposed aluminum parts can be corroded. Scratches should be touched up with paint, but nail polish is effective in an emergency. The glider should be rinsed with fresh water every time it is exposed to salt water.

Several sizes of anodes are available. Contact glidersupport@teledyne.com for assistance in determining the appropriate size for your deployment.

How to Test

1. Probe between the forward anode and top pump flange screw using a digital voltmeter on ohms setting.
2. Verify that the resistance is less than 10 ohms.
3. Probe between the aft anode and ejection weight tube using a digital voltmeter on ohms setting.
4. Verify that the resistance is less than 10 ohms.

Batteries/Coulomb Counter

Alkaline battery packs, which are nominally at 15 volts, consist of 10 diode-protected Duracell C-cells in series. As indicated below, the number of packs can be adjusted, depending on reserve buoyancy after payload considerations. Given 26 packs (260 C-cells), the total battery weight is ~18.2 kg with 7,800 kJoules of available energy.

How to Configure

Factory configured. When installing new batteries, the following commands should be entered at a GliderDOS prompt. These will reset the coulomb counter to 0.

```
Put m_coulomb_amphr_total 0
```

```
Exit reset
```

From the GliderDOS prompt, type `get m_coulomb_amphr_total` and confirm that the value is close to zero and increasing when powered by battery. This value should not increase when powered by a power supply.

How to Test

When predicting mission longevity, see the electric mission spreadsheet:

Forum [glider/windoze/production/src/doco/how-to-calibrate/](#)

Battery voltage and coulomb measurements should be monitored during lab testing as well as during deployments.



WARNING If the glider contains alkaline batteries, there is a small but finite possibility that batteries of alkaline cells will release a combustible gas mixture, especially if the batteries are exposed to water or sea water and/or shorted. This gas release generally is not evident when batteries are exposed to the atmosphere, as the gases are dispersed and diluted to a safe level. When the batteries are confined in a sealed instrument, the gases can accumulate and an explosion is possible. Teledyne Webb Research has added a catalyst inside of the glider to recombine hydrogen and oxygen into water, and the glider has been designed to relieve excessive internal pressure buildup by having the hull sections separate under internal pressure.

Teledyne Webb Research knows of no way to completely eliminate this hazard. The user is warned, and must accept and deal with this risk in order to use this instrument safely as so provided. Personnel with knowledge and training to deal with this risk should seal or operate the instrument.

How to Evaluate Data

See "How to Determine Mission Battery Longevity" on page I-1.

Relevant Sensors

- m_battery
- m_battery_inst
- m_coulomb_amphr_total
- m_coulomb_current
- m_lithium_battery_relative_charge

Desiccant

The glider should be opened and sealed in a controlled, dry environment, if possible. Desiccant packs should be installed to absorb internal moisture and should be replaced for each deployment. When the glider is open for long periods, the desiccant should be stored in sealed plastic bags to prevent it from being saturated by atmospheric moisture. A fully saturated dessicant bag can increase in mass by ~40 grams and can therefore also affect ballasting.

Desiccants are often installed in a location of opportunity inside the vehicle depending on the configuration. Often this is in the payload bay.

How to Configure

Keep dry. When the glider is open desiccant will gradually gain mass from the atmosphere. Increase in mass can effect glider ballasting and cause desiccant to become ineffective.

How to Test

Desiccant weight can be monitored. The desiccant pouches provide by Teledyne Webb research weigh ~114 grams when fully dried.

Lifting Point

The payload or payload bay can be outfitted with a lifting point.



NOTE The single point pick point should not be used with gliders with stack on bays.

Recovery System (optional)

How to Configure

See the *Slocum G3 Glider Maintenance Manual* for installation and configuration.

How to Test

1. Disconnect the supply lead to the recovery system at the Mecca connector.
2. Connect the digital voltmeter between the supply lead and the forward anode.
3. Type `put c_recovery_on 1`.
4. Verify that the voltage is at least 5 volts.
5. Type `put c_recovery_on 0`.
6. Reconnect the recovery system supply lead.

Relevant Sensors

`c_recovery_on`—A value of 1 will activate the recovery system. Unlike the jettison weight, it will require several minutes to release.

Strobe

How to Configure

Factory configured

How to Test

From GliderDOS or `lab_mode` or during a mission preceded by `!`, type `strobe on` to activate and `strobe off` to deactivate. The strobe can also be controlled autonomously during a glider surfacing behavior.

Relevant Sensors

- `c_strobe_ctrl`
- `m_strobe_ctrl`

Spare Ports and Modem Readiness

G3 gliders are equipped with a number of through the hull external connectors which can be customized based on operator requirements. The G3 glider is equipped with a spare IE55 connector on the forward section to accommodate a transducer that provides acoustic modem and altimeter ranging functions. Payload bay stiffener rings may be additionally accommodated with an impulse connector (typically 8-pin) to facilitate communications and power provisioning for externally mounted sensors.

Before connecting any external connectors, they should be flushed with fresh water to ensure that any particulates are rinsed from conductive surfaces. When plugging in an external connector, apply a small amount of dielectric grease or silicone lubricant to the connector surface to provide lubrication to the high friction interfaces. Inspect bulkhead connector bodies to identify any potential corrosion and contact glidersupport@teledyne.com with photos and concerns in the event that a connector is suspicious and may be failing.

The G3 glider has additional wet payload space available in the nose fairing, which can be extended, pushing the nose cone out to provide significant wet volume for external sensors. This optional expanded volume will not support the nose recovery system. Please contact glidersupport@teledyne.com if use of this volume is of interest.

The nose fairing has a check valve, which is new with the G3 glider. It is designed to prevent silt from entering and collecting inside the nose cap.



CAUTION Care should be taken during ballasting that all air has been removed from the buoyancy engine, nose cap and fairing.



Figure 1-18 External Connectors

Optional 10-Watt Thruster

A high efficiency thruster with larger blades, which sweep back to lower drag when not in use, is an optional add on accessory for the G3 glider. However, while additional speed or operational capability will be provided by employing this thruster, there can be a significant penalty in energy consumed. Energy budgets should be watched carefully when using the thruster.

Operational Usage

There are three modes in which you can use thruster-assist:

- yo-based sawtooth flight
- drift_at_depth horizontal flight
- surface assist/lens penetration

The thruster will also be turned on for a short burst right before diving after a surfacing in order to remove buildup. This is turned on by default, and can be disabled.

The full documentation for this thruster can be found at https://datahost.webbresearch.com/files.php?cwd=glider/RELEASE_8_0/doco/how-it-works/thruster.txt. And the recommended thruster mission, thrstock.mi can be found at <https://datahost.webbresearch.com/files.php?cwd=/glider/production/target-glider/electric-1000/missions/thrstock.mi>.

To run a thruster in air:

1. Make sure the propeller blades are clear of the cart.
2. Report ++ m_thruster_current m_thruster_power.
3. Put c_thruster_on 30 (only do this for < minute or so)
4. Confirm that the blades are spinning clockwise when viewed from the aft end.
5. Put c_thruster_on 0



NOTE Do not run the thruster for prolonged periods of time in air.

Post Deployment Care

To service a thruster after deployment:

1. Remove the thruster hub using a 7/64 hex wrench.
2. Rinse thoroughly with fresh water.
3. Apply molybdenum disulfide grease which comes with the thruster kit.
4. Replace the hub.

2 Pre-mission Testing

TWR provides a document called FCP Functional checkout procedure. It is recommended that this checkout procedure be performed on the glider after final seal in the lab, after storage, and after shipping. The following minimal procedures should be followed to qualify a glider before deployment, or with new or modified software.



NOTE These tests need to be done outside, because the gliders need a clear view of the sky to get a GPS fix and to make Iridium communications.

On the Beach, Boat, or Bench

1. Power on the glider and enter GliderDOS by typing `ctrl-c` when instructed.
2. Type `put c_air_pump 0` to stop the air pump from running.
3. Type `callback 30` to hang up the Iridium phone.
4. Test the GPS by typing `put c_gps_on 3`. This will place GPS communications into a verbose mode. You should see the data stream change from V to A. Generally, several minutes of GPS acquisition is all that is necessary. However, if large geographical distances have been moved since the last position was acquired, it is recommended to let the GPS run for some time to build a new almanac. When satisfied with the GPS location, type `put c_gps_on 1` to return to the non-verbose mode.



WARNING NEVER deploy a glider in `lab_mode`!

5. Test the motors by typing `lab_mode on` and then `wiggle on`. Run for 3-5 minutes to check for any device errors or other abnormalities. Type `wiggle off` to stop wiggling. If there is a deep pump, you might choose to run the motor longer and report the pump's location to the screen.



NOTE Make sure the glider is set to `boot app` before deploying it in the water.

6. If no errors are found, type `lab_mode off` to return to the GliderDOS prompt. (Always ensure that the glider is not in PicoDOS or lab mode before deploying it in the water.)

7. Type `run status.mi` and confirm that all sensors are being read. The mission should end with this confirmation message, `mission completed normally`.
8. Load the glider into the boat and head out toward the first waypoint or deployment location.

In the Water

1. If possible, attach a line with flotation to the glider before putting the glider in the water. (However, if you have great confidence in the ballasting and are an experienced user, proceed without a flotation device.)
2. Once the glider is deployed, type `run status.mi` again. If ballasting has already been confirmed proceed to step 3 or 4. If unsure of ballasting, most operators will then run one or several of the following missions to quantify the quality of the ballasting. An operator may choose to remove the buoy during any number of the missions below if results are satisfactory:

`run ini0.mi`

Does 1 yos, dive to 3 m
Fixed pitch and fin

If you are performing this mission on a line with flotation, ensure the line length is sufficient or modify yo depth.
3. When the glider returns to the surface, examine the data to evaluate whether it is OK to proceed.
4. If you have not removed the line from the glider, do it now.
5. From the GliderDOS prompt, type `exit reset`. This will force all of the sensor values to reinitialize. It is advisable to do an `exit reset` after removing the buoy but not necessary.
6. When the glider reboots, type `ctrl-c` to return to a GliderDOS prompt, and type `loadmission waterclr.mi` to zero any built-up water currents that are remembered long term.
7. Type `run stock.mi` to begin the stock mission or run the desired mission.

3 Deployment and Recovery



WARNING Deployment and recovery can be challenging and/or dangerous, especially in heavy seas. Plan accordingly to get the glider in and out of the water.

Deployment conditions and craft may vary. The glider can be deployed using the pick point. With smaller vessels, the glider cart can be used on the gunwale to allow the glider to slip into the water. During handling, hold the glider by the fin.

During recovery, the pick point or cart can be useful tools to manipulate and provide support while moving the glider aboard. A hook or lasso on a pole has also been used to manipulate the glider while in the water.

Photographs of deployment and recovery by cart can be found in the manual's link in the *Operators Handbook* at the glider forum.

Glider outfitted with a recovery system can be commanded to release the nose flotation. A vessel's overhead crane can then be used to lift the vehicle from the water line. To activate, type `put c_recovery_on 1`.

Deploying the Glider

Deployment at sea can be dangerous, and the welfare of crew and glider handlers should be considered while at the rail of a ship. From a small boat the glider cart can be used to let the glider slip easily into the water. Lower the nose ring and undue the strap when ready to release the glider and tip the cart to allow the glider to slide into the water.

For larger boats, the pick point affixed to the payload bay should be used to lower and raise the glider with a crane or winch from the vessel to the water.

Glider with the Buoy and Rope Ready for the First Deployment



NOTE In the deployment sequence below, the fin can be handled, do not suspend the entire glider by the fin.



Large Ship Deployment

A quick release system using the pick point can be fashioned from supplies found on most vessels, as illustrated in the following two images.



Recovering the Glider



NOTE A boat hook can be used to manipulate the glider in the water. Care should be taken with non-digifin gliders during deployment and recovery, because the fin can be knocked out of calibration or damaged if handled too aggressively. Handle non-digifin gliders by the tail boom or pick point only.



Lower the cart with nose ring into water, and manipulate the glider by the tail boom into position on the cart. Lift and tilt the glider onto the ship's deck.

4 Emergency Recovery

Contact glidersupport@teledyne.com for specifics for the particular emergency; however, some or all of the following may need to be done in an emergency recovery scenario.

- Callback time can be increased significantly by changing these glider sensors:
 - `u_iridium_max_time_til_callback(sec) 1800.0 #`. This is the maximum legal value for
`# c_iridium_time_til_callback`.
 - If the glider has blown the ejection weight or you feel safe, change `u_max_time_in_gliderdos` from GliderDOS from 900 to 3600. This will increase how often the glider cycles into a mission and tries to call in to save energy while it is stuck at the surface.
- Consider running special scripts to conserve energy.
- A pilot might change to a `callback 30` script on the Dockserver.
- Contact service Argos and turn on Argos ALP (all location processing).
- Determine the best-known position with the available data.

5 Glider Communications

The glider is intended to be used in conjunction with Dockserver. See the *Slocum Fleet Mission Control User Manual* for information on communications to the glider while using the Dockserver.

Communicating with the RF Modem

The RF modem transmits at a frequency of 900 MHz at 1 watt nominally, but it can be configured to transmit at 0.05 watts.

A FreeWave modem should be paired with the vehicle using the Point-to-Point protocol (with repeaters, if necessary), the correct IDs in Call Book, and matching frequency keys. See "FreeWave Configuration" on page F-1.

1. Connect the FreeWave to the Dockserver or to the computer running terminal emulation.
2. Once power is applied to the FreeWave and the glider, communication should begin.

Communicating with Argos

The Argos device transmits at a frequency of ~401 MHz and power of ~1 watt.

Argos messages are nominally transmitted from the glider while it is powered up on the surface. See "Argos Data Format" on page H-2.

Communicating with Iridium

The Iridium phone transmits at a frequency of ~1600 MHz and power of ~1.1 watt.

Iridium is generally used in the absence of the FreeWave. The primary and secondary phone numbers are configured in the autoexec.mi file. While in GliderDOS, Iridium is the primary route of communication; however, while running a mission, Iridium will call only if FreeWave communication is not available. Similarly, during data transfer, the data is transferred via Iridium, only if FreeWave is not available. The option to force transfer through Iridium while FreeWave is connected is also available.



WARNING Never enter PicoDOS outside of a lab environment!

- To test Iridium while in PicoDOS, change the Dockserver to serial perspective and type `talk iridium` (refer to "PicoDOS" on page 6-3).

- To leave talk, type `control-c`. Pause for a moment; then type `control-c` again.
- To dial a number using a commercial card, type `AT 001 number to be dialed`. To dial a number using a military card type `AT 00697 number to be dialed`.
- `ata` answers an Iridium phone call.
- `ath` hangs up the phone after an Iridium call. Alternately, `control-c` can be used to hang up the phone.

Communicating with RUDICS

Refer to section 12 of the *GMC Manual*, which is located at:

<ftp://ftp.glider.webbresearch.com/glider/windoze/production/src/gliderMissionControl/Documentation/gmcUserGuide.pdf>

6 Software Architecture

The controller code is written in C and architecturally is based on a layered single-thread approach where each task is coded into a behavior, and behaviors can be combined in almost any order to achieve flexible and unique missions. These behaviors can also be constructed to deal with more complex issues, such as dead reckoning navigation, current correction, and adaptive sampling.

Each device is labeled as a sensor and is logged every time that the value changes during a mission. This data is retrieved as a binary file and is post-parsed into a matrix that allows you to replay flight dynamics or easily construct graphical views of vehicle performance or scientific data. A subset of the sensors can be chosen as a science data package so as to reduce surface radio transmission time, allowing near real-time data collection.

The glider can have in memory of the CF card any number of pre-written missions that can be called or a new mission can be created, downloaded to the glider via the RF modem or Iridium and run. Mission changes might include different inflect depths, new GPS waypoints, or turning a behavior on or off, such as current correction. Mission files are small text files. To further decrease the size of mission particulars, portions of missions can be broken out into .ma or mission acquisition files. This allows for transferring very small files to modify the most commonly adjusted mission sensors.

Table 6-1 Software Control Hierarchy

Component Name	Description
PicoDOS	Persistor's operating system.
GliderDOS	Glider's operating system.
masterdata	Defines the sensors; better known as the glider variables. There are approximately 2500 variables.
longterm.dat	Maintains the sensors/variables on a power cycle.
autoexec.mi	Defines glider specific variables.
.mi files	Mission files; define mission variables, mission behaviors and argument.
.ma files	Mission acquisition file; define mission behavior variables.

G3 Software

G3 gliders are supported by software release 8.0 and greater, available at the TWR Glider User Forum <https://datahost.webbresearch.com>. G3 gliders will have SBMB2, and the G3 “sbm2sci.app” application should be used when updating code on the science Persistor. The

Slocum glider code update procedures can be found at <https://datahost.webbresearch.com/viewtopic.php?f=5&t=263&p=724&hilit=update#p724> and at <https://datahost.webbresearch.com/files.php?cwd=/glider/production/readme.txt>.



WARNING Always read the readme.txt before updating to the production release of glider code.

G3 Missions

While pilots do not have to use autoballast due to the introduction of High Displacement (HD) pumps with the G3 glider, TWR highly recommends the use of autoballast for a number of reasons. Of the greatest concern is the energy consumption/conservation and uncontrolled or unnecessary speed. All legacy missions and ma files should be reviewed and modified before being deployed to G3 vehicles.

With the release of 8.0 there is a significant masterdata change to default values for buoyancy drive used for flight arguments (not surface arguments or initial.mi and lastgasp.mi missions).

If no buoyancy is specified by the mission or ma file the new masterdata fault of +/-260 cc of drive will be used. If a pilot wants to use greater amounts of the HD drive, or speed derived from greater drive are desired, this must be done by using autoballast or manually overriding the following arguments in the yo behavior:

```
b_arg: d_bpump_value(X) -260.0      #! min = -1000; max = 1000
b_arg: c_bpump_value(X) 260.0       #! min = -1000.0; max = 1000.0
```



NOTE Note using the full HD drive capability will significantly increase the energy consumed.



CAUTION Diving in shallow waters with legacy mission setting of +/- 1000cc may result in fast steep dives that run the risk of uncontrolled bottoming.

The default TWR mission recommended is astock.mi which utilizes autoballast and can be found in the production release at <https://datahost.webbresearch.com/files.php?cwd=/glider/production/target-glider/electric-1000/missions>.

PicoDOS

PicoDOS is the operating system that ships with the Persistor CF1. Typing `help` will access the many DOS like functions and their command lines. The navigation devices may be tested in PicoDOS using the talk program. The syntax of the talk program is shown below:

Table 6-2 Syntax of the Talk Program

Command Name	Description
<code>talk gps</code>	Turns on the GPS and displays the NMEA output. This may also be used to acquire a full almanac. Leaving the GPS on for more than 15 minutes will refresh the almanac.
<code>talk att</code>	Turns on the attitude sensor and displays the pitch, roll, heading, and temperature output.
<code>talk iridium</code>	Turns on the Iridium modem and allows you to manually place or receive a call.
<code>talk arg</code>	Legacy: Turns on the ARGOS transmitter and displays the output.

PicoDOS contains the following folder and file structure:

Volume in drive C is NONAME
Volume Serial Number is 75E1-51B6

Directory of C:

APP
CONFIG
BIN
LOG
MISSION
SENTLOGS
STATE
AUTOEXEC.BAT

Config Folder

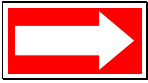
The files in the config folder are described in the table below.

Table 6-3 Files in the CONFIG Folder

File Name	Description
<code>autoexec.mi</code>	Configuration file for glider calibration constants and factory settings.
<code>config.sci</code>	Specify which sensors are sent to the glider and when.

Table 6-3 Files in the CONFIG Folder

File Name	Description
sbdlist.dat	Specify which sensors are recorded for a short binary data file.
mbdlist.dat	Specify which sensors are recorded for a medium binary data file.
simul.sim	Used to convert the glider into several versions of a simulator. This file must be deleted before an actual flight.
zmext.dat	Automatically places transferred files in the correct directory from any other directory.



NOTE The simul.sim file must be deleted before an actual flight.

Bin Folder

Pico executable programs are stored in the bin folder. All of these programs run in PicoDOS. These files are described in the table below.

Table 6-4 Files in the Bin Folder

File Name	Description
adtest.run	Displays and updates raw voltages for analog to digital devices. Used in calibration procedures.
alloff.run	Placed in autoexec.bat to start the world with all registers zeroed and turns on the RF modem for communication.
consci.run	Switches the RF modem to communicate with the payload/science computer for direct access and code loads. A loss of carrier detection on the glider side will automatically switch back to the glider controller.
srtest.run	Allows switching selected bits on and off. Used for hardware interface board testing.
talk.run	Ports to specific components, such as Argos, attitude sensor, and GPS, for direct access and setup. <code>talk help</code> displays a list of parameters.
uarttest.run	Tests uart drivers for programming.
zr.run, zs.run	Required for the Zmodem to send and receive transfers.

Logs Folder

Mission derived data is stored in the logs folder. The data types are shown by file extension and described in the table below.

Table 6-5 Files Types in the Logs Folder

File Extension	Description
.dbd	Dinkum binary data—All sensors turned on for recording are stored in this type file.
.sbd	Short binary data—Records only those sensors specified in sbdlist.dat to reduce file size and thus communication time.
.mbd	Medium binary data—Records only those sensors specified in mbdlist.dat.
.mlg	Mission log—Tracks the calls for behaviors and device drives.
.log	Stores the process of opening and closing files and operations.

Missions Folder

Missions are stored in this folder as .mi files. These are text files that, when run by the glider, determines the behavioral parameters.

Sentlogs Folder

This folder stores .dbd, .mbd, .sbd, and .mlg files from log folders that were sent successfully.

Autoexec.bat

The autoexec.bat file is a DOS system file whose function is to automatically execute commands on system startup.



NOTE Do not confuse autoexec.bat with autoexec.mi. (See the config directory for a description of autoexec.mi.)

Typically the autoexec.bat file contains:

```
path \bin
prompt (GPico) $P$G
alloff
```

GliderDOS

The operating shell GliderDOS is a superset of PicoDOS. GliderDOS is an application that performs most of the PicoDOS functions and has knowledge of the glider. Typing `help` or `?` lists the functions available within GliderDOS. All of the variables used in GliderDOS are referred to as sensors and are defined in the Masterdata file (see "Masterdata" on page 6-7). The glider lists all of its sensors names with the `list` command. Behaviors then use these values to operate the vehicle.



NOTE The production copy of masterdata can be found here:
<ftp://ftp.glider.webbresearch.com/glider/windoze/production/masterdata>

GliderDOS is an application that is loaded onto the Persistor (`glider.app`). When it is configured correctly, it boots up and calls `autoexec.mi`, a file containing all of the glider's calibration coefficients. Certain devices are set automatically to ensure the best possible surface expression:

- Ballast pump assembly full extension
- Pitch full forward
- Air pump on
- Argos on
- FreeWave on
- GPS on

It is necessary to be in PicoDOS in order to:

- Load new source code for GliderDOS.
- Work in the file structure without the device drivers being called.



WARNING The glider should never be deployed while in PicoDOS or displayed while set to `boot pico`.

For a detailed description of how to load a glider and payload bay with a new release of glider production code, visit:

Forum [glider/windoze/production/src/doco/software-howto/](#)

When in GliderDOS (or as noted from PicoDOS), use the basic control commands in the table below.

Table 6-6 Basic Glider Control Commands

Command Name	Description
<code>control-c</code>	Takes control of the glider.
<code>exit pico</code>	Sends the glider to PicoDOS from GliderDOS.
<code>exit reset</code>	Return to GliderDOS (as long as Persistor is set to boot app).
<code>boot pico</code>	Commands the glider to start in PicoDOS when reset or the power is cycled. Use this when loading an application (glider.app). Never deploy a glider left to <code>boot pico</code> .
<code>boot app</code>	Commands the glider to start in GliderDOS when reset or the power is cycled. Use this after loading an application (glider.app).

Masterdata

The Masterdata file contains all of the sensor definitions and their default values as used by the application. Because Masterdata is essentially just a list of the sensors, it cannot be edited. A copy of the Masterdata file is available at:

Forum [/glider/windoze/production](#)

Prefixes

The prefixes in masterdata are described in the table below.

Table 6-7 Masterdata Prefixes

Prefix Name	Description
m_	Measured
c_	Commanded
u_	User defined before run time
f_	Set in factory. Do not change unless you know what you are doing.
x_	Never set this; typically computed at run-time.
s_	Simulated state variables
sci_	Science variable

Sensor Commands

Sensors can be changed on the GliderDOS command line. The sensor commands are described in the table below.

Table 6-8 Glider Sensor Commands

Command Name	Description
list	Prints all of the sensor names and values.
get <i>sensor_name</i>	Returns the present value of the sensor requested.
put <i>sensor_name value</i>	Changes the value of the sensor.
report ?	Prints help.
report + <i>sensor_name</i>	Reports the sensor value every time it changes.
report ++ <i>sensor_name</i>	Reports the sensor value every cycle.
report - <i>sensor_name</i>	Removes sensor from reporting.
report all	Reports all changed sensors.
report clearall	Removes all sensors from reporting.
report list	Tells what is being reported.

Device Commands

The `use` command displays a list of all devices that are installed and in use. Installed devices are dictated by the glider's `autoexec.mi` file. During GliderDOS operation, an installed device is taken out of service if it receives two errors. During missions, this number is increased to 20 for most devices. Errors are primarily generated when the driven device is not moving. For this reason, the device is eventually taken out of service as a protective measure.

Table 6-9 Use Command Options

Command Option	Description
<code>use ?</code>	Prints help.
<code>use</code>	Lists all devices that are installed and in use.
<code>use + device_name</code>	Puts device(s) in service.
<code>use - device_name</code>	Takes device(s) out of service.
<code>use all</code>	Puts all installed devices in service.
<code>use none</code>	Takes all devices out of service.

From a GliderDOS prompt, the `help` command lists all commands available to the user. These commands are also listed in the table below.

Table 6-10 Device Commands

[In this table, all commands listed in all caps are executable during a mission when preceded by an exclamation point (!). The commands in lower case cannot be used during mission. Commands are not case-sensitive when you are entering them in the glider software.]

Command Name	Syntax and/or Description
ATTRIB	<code>attrib [+ - rash] [d:][p][name]</code>
ballast	<code>ballast ?</code> ; for help
boot	<code>boot [pico][pbm][app]</code>
callback	<code>callback <minutes til callback></code>
capture	<code>capture [d:][p]fn [/Dx/B/N/E]</code>
CD	Change directory
CHKDSK	<code>chkdsk [d:][p][fn] [/F]/[I] *</code>
CLR_DRIFT_TABLE	Re-initializes neutral ballast vs depth table
CLRDEVERRS	Zero device errors
consci	<code>consci [-f rf irid]</code> ; console to science
COPY	<code>copy source dest [/V]</code>

Table 6-10 Device Commands

[In this table, all commands listed in all caps are executable during a mission when preceded by an exclamation point (!). The commands in lower case cannot be used during mission. Commands are not case-sensitive when you are entering them in the glider software.]

Command Name	Syntax and/or Description
CP	cp <src_path> <dest_path>; copies a file system branch
CRC	Computes CRC on memory
date	date [mdy hms[a p]] /IEUMCP]
DELLOG	dellog all mlg dbd sbd
DEL	del [drv:][pth][name] [/p]
DEVICES?	Prints device driver information
DF	Prints disk space used and disk space free
digifin	digifin [dc fc lr pr rc rr rs sa sc st wr]
DIR	dir [d:][p][fn] [/pwbv4a:a]
DUMP	dump file[start[,end]] *
ERASE	erase [drv:][pth][name] [/P] *
exit	exit [-nofin] [poweroff reset pico pbm]
GET	get <sensor name>
HARDWARE?	hardware? [-v]; hardware configuration
HEAP	Reports free memory
HELP	Prints help for commands
HIGHDENSITY	highdensity ?; for help
LAB_MODE	lab_mode [on off]
li_ion	Need description here
LIST	Displays all sensor names
loadmission	Loads mission file
logging	logging on off; during GliderDOS
LONGTERM_PUT	longterm_put <sensor name> <new value>
LONGTERM	longterm ?; for help

Table 6-10 Device Commands

[In this table, all commands listed in all caps are executable during a mission when preceded by an exclamation point (!). The commands in lower case cannot be used during mission. Commands are not case-sensitive when you are entering them in the glider software.]

Command Name	Syntax and/or Description
lpstop	lpstop [0 - 60 secs] ; puts the Persistor into low power mode for the specified amount of time. Will use 30 seconds as the low power time if none is specified.
LS	ls [path] ; list a file system branch
MBD	mbd ?; for help
MKDIR	mkdir [drive:][path]
MV	mv <src_path> <dest_path>; copy a file system branch
PATH	path—Shows search path * path [[d:]path[;...]] [/p] *
prompt	prompt [text] [/p] *
PRUNEDISK	Prunes expendable files to free space on disk
PURGELOGS	Deletes sent log files
PUT	put <sensor name> <value>
RENAME	rename [d:][p]oldname newname
REPORT	report ?; for help
RMDIR	rmdir [drive:][path]
RM	rm <path>; deletes a file system branch *
run	run [mission_file]; runs the mission file
SBD	sbd ?; ? for help
SEND	send [-f={rf}]{irid} [-num=<n>] [-t=<s>] [filespec ...]
sequence	sequence ?; do this for help
SETDEVLIMIT	setdevlimit devicename os w/s w/m
SETNUMWARN	setnumwarn [x]; sets max dev warnings to x
SET	set [var=[str]] [/slfe?] *
SIMUL?	Displays a print description of what is simulated
SRF_DISPLAY	srf_display ?; for help

Table 6-10 Device Commands

[In this table, all commands listed in all caps are executable during a mission when preceded by an exclamation point (!). The commands in lower case cannot be used during mission. Commands are not case-sensitive when you are entering them in the glider software.]

Command Name	Syntax and/or Description
strobe	strobe [on off] ; flashes strobe light
sync_time	sync_time [offset]; syncs system time with gps time
tcm3	tcm3 ?; for help
time	time [hh:mm:ss [a p]] [/m/c]
tvalve	tvalve [up charge down][backward] *
TYPE	type [drv:][pth][name]
USE	use ?; do this for help
VER	Displays firmware versions
WHERE	Prints latitude/longitude
whoru	whoru vehicle name;; displays vehicle name
WHY?	why? [abort#]; displays the reason for an abort
wiggle	wiggle [on off] [fraction]; moves motor
ZERO_OCEAN_PRESSURE	Recalibrates zero ocean pressure
ZR	zmodem rec: zr ? for help
ZS	zmodem send: zs ? for help

* not often used by average user

Science Computer

In order to communicate with the Persistor:

- From PicoDOS, type `consci`.
- from GliderDOS, type `consci`.
- From a mission, type `ctrl-t`.

The Persistor's folder/file structure is shown below.

```
app
bin
config
autoexec.bat
```



NOTE A loss of carrier detection when in communication with science will automatically switch control and communications back to the glider controller.

Proglets.dat is the file that contains the standard configuration values for all of the glider's science instruments. Each instrument is addressed in a section of code known as a *progle*t. Each progle

Table 6-11 Selected Devices Connected to the Science Computer

Device Name	Description
ctd41cp	Sea-bird CTD (SBE-41CP) continuous profiling
bb2f	Wet labs bb2f fluorometer/backscatter sensor
bb2c	Wet labs, bb2c sensor
bb2lss	Wet labs, light scatter sensor
sam	Wet labs, Scattering Attenuation Meter
whpar	WHOI PAR Photosynthetically Active Radiation
whgpbm	WHOI Glider Bathy-Photometer
hs2	HobiLab HydroScat2 Spectral Backscattering Sensor
bam	Benthos Acoustic Modem



NOTE The Persistor must always be set to `boot app` so that the science application runs the proglets.

Bin Folder

Pico executable programs are stored in the bin folder. All of these programs run in PicoDOS. These files are described in the table below.

Table 6-12 Files in the Bin Folder

File Name	Description
amconnct.run	Port to acoustic modem that allows communication for testing.
blast.run	Blasts characters to all ports for testing.
ctd_ctrl.run	Runs the CTD program for sampling. In <code>autoexec.bat</code> typed with a number ≤ 3 for fastest sampling, or > 3 for sampling at that rate in seconds. (No longer runs on gliders with software version 5.0 or greater.)
mcmd.run	Runs the acoustic modem.

Table 6-12 Files in the Bin Folder

File Name	Description
mdatacol.run	Sets the acoustic modem to listen mode until the buffer is full, then tells the glider to surface.
u4talk.run	Testing of uart drivers for programming.
ctd_hs2.run	Runs either CTD, HydroScat 2, or both simultaneously. Type <code>help ctd_hs2</code> in the science computer for information on usage and options. (No longer runs on gliders with software version 5.0 or greater.)
zr.run, zs.run	Required for Zmodem send and receive transfers.

Config Folder

The files in the config folder are described in the table below.

Table 6-13 Files in the Config Folder

File Name	Description
appcmd.dat	Simulates functions in the payload bay. This file must be deleted before an actual flight.
zmext.dat	Automatically places transferred files in the correct directory from any other directory.
proglets.dat	This contains configuration of the science computer sensors and defines which sensors are installed in the payload bay.



NOTE The appcmd.dat file must be deleted before an actual flight.

Autoexec.bat

This file contains the path to the executable and the PicoDOS prompt.

```
path \bin /p
prompt (sci)$p$g
```

7 Science Data Logging

In the G1 and early G2 models of the Slocum glider, all science sensor data had to be passed over the serial data connection (also known as the clothesline) in real time to the main flight processor, where it was logged with glider sensor data. However, the clothesline imposes a limit on the number of science sensors per second that can be transferred between the glider and science processors. Science data logging (SDL) is an architectural change that addresses the clothesline's limited throughput. In the new system using SDL, the science sensors do not have to be transmitted over the clothesline, which eliminated the bottleneck.

With SDL, the science processor primarily collects and logs data, and although it can maintain a high cycle rate, it runs at reduced CPU speed by default. Of course, the actual attainable throughput depends on the installed sensor load and sensor data stream parsing overhead.

Logfile Types

Science now has a parallel logfile type for each logfile type on the glider, as shown in the table below. Each pair is formatted the same (i.e., .ebd files are formatted the same as .dbd files).

Table 7-1 Logfile Types on the Glider and Science Processors

Logfile Type on the Glider Processor	Equivalent Logfile Type on the Science Processor
.dbd	.ebd
.mbd	.nbd
.sbd	.tbd
.mlg	.nlg

Just as `m_present_time` should be present in `mbdlist.dat` and `sbdlist.dat` on the glider processor, `sci_m_present_time` should be present in `nbdlist.dat` and `tbdlist.dat` on the science processor.

The logfiles on the science processor are stored in the same directories as the glider processor:

`\logs` and `\sentlogs`, with the header cache files in `\state\cache`

To support SDL, some new Sc iD OS commands parallel GliderDOS commands on the glider:

`dellog`

`df`

`get` (any variable known to science, as seen by science)

`heap`

`list` (all variables known to science, with their values)

prunedisk

put (Think about whether it propagates to glider side or not.)

send

Configuration Files

There are also some new configuration files on science that are parallel to the glider's configuration files (see table below). Each of these pairs is also formatted the same.

Table 7-2 Configuration File Comparison on the Glider and Science Processors

Configuration File on the Glider Processor	Configuration File Equivalent on the Science Processor
\config\mbdlist.dat	\config\nbdlist.dat
\config\sbdlist.dat	\config\tbdlist.dat
\config\highdens.dat	\config\highdens.dat

Transparency

When the `logging on` command is issued to the glider processor, a message is passed over the clothesline that enables logging for the science processor. If science is not running at the time, a flag is set to turn on science logging when science is started. When glider logging is turned off, a clothesline message is sent to turn off science logging. If science is not running at the time, a flag is set that cancels the startlogging flag.

The log files will share the same file name root (i.e., 12345678.mlg and 12345678.nlg will represent the glider, respectively). Logfile names on science and glider are kept synchronized, but the science processor is unaware of these; the glider processor furnishes the names when it tells science to start logging each time.

Sending data files to the Dockserver is transparent. Issuing the `send` command from GliderDOS or the `s` command from the surface dialog causes log files to be sent first from science and then from the glider. The same command line is processed by each processor in turn. For example, if the command as typed is

```
send -num=3 *.sbd *.tbd
```

then science will send three .tbd files (while finding no match for *.sbd), and the glider will send three .sbd files (finding no match for *.tbd). The lack of matching files for some of the file specs is not considered an error. The `-num=3` command limits the number of files sent by each processor. In this case, the total number of files sent by two processors together is six. Sites using Data Server and Data Visualizer can view data as before. Other shoreside software tools will likely require changes to view science data. A merge tool has been developed to combine ASCII files to appear as they did before science data logging was introduced and can be found at:

Forum glider/windoze/production/windoze-bin/

The dellog and prunedisk commands are local to either science or the glider. The send command issued directly from SciDOS is local to science (not recommended).

Control

In normal operation, only certain key science sensors are sent to the glider. For setup purposes, `c_science_on` is generally set to 2 or 3, and all sensor values are visible as they are being sent. In order to send all science sensors, as was the case before this release, a new sensor must be used (see `c_science_send_all` below).

Table 7-3 Sensors Added to the Glider

Sensor Name	Processor	Default Setting	Description
<code>c_science_send_all(bool)</code>	Glider	0	Tells the science processor whether to send all variables or just a few
<code>m_mission_start_time(timestamp)</code>	Glider	N/A	Propagates to science processor
<code>m_science_readiness_for_consci(enum)</code>	Glider	N/A	Tells if ready or, if not, why not
<code>sci_m_disk_free(Mbytes)</code>	Science	N/A	How much space is currently free on science
<code>sci_m_disk_usage(Mbytes)</code>	Science	N/A	How much space is currently used on science
<code>sci_m_present_secs_into_mission(sec)</code>	Science	N/A	Analog of <code>m_present_secs_into_mission</code> on science
<code>sci_m_free_heap(bytes)</code>	Science	N/A	Analog of <code>m_free_heap</code> on science
<code>sci_m_min_free_heap(bytes)</code>	Science	N/A	Analog of <code>m_min_free_heap</code> on science
<code>sci_m_min_spare_heap(bytes)</code>	Science	N/A	Analog of <code>m_min_spare_heap</code> on science
<code>sci_m_spare_heap(bytes)</code>	Science	N/A	Analog of <code>m_spare_heap</code> on science
<code>sci_x_disk_files_removed(nodim)</code>	Science	N/A	Count of files removed by last science processor's <code>prune</code> command
<code>sci_x_sent_data_files(nodim)</code>	Science	N/A	Count of files successfully transmitted by last science processor's <code>sent</code> command

Table 7-3 Sensors Added to the Glider

Sensor Name	Processor	Default Setting	Description
<code>u_sci_cycle_time(secs)</code>	Glider	1.0	Tells the science processor how fast to run
<code>u_sci_dbd_sensor_list_xmit_control(enum)</code>	Glider	0	Always transmit the header to tell the science processor what to do
<code>x_science_logging_state(enum)</code>	Glider	N/A	Indicates the science processor's logging state
<code>u_science_send_time_limit_adjustment_factor(nodim)</code>	Glider	0.5	Puts an absolute limit on how long the glider can spend sending science files during a single consci session

Sending Science Data

The science `send` command is implemented as a pre-programmed consci batch command. It may take a long time to complete a single send command if a large number of files is to be transferred, so two sensors are used to limit the length of a consci session:

- `u_sci_cmd_max_consci_time` (seconds) establishes a limit for the length of a single consci session. The default value for this sensor is factory set to 3600 seconds (1 hour).
- `u_science_send_time_limit_adjustment_factor` (nodim) specifies the fraction of `u_sci_cmd_max_consci_time` that can be used for the sending of science data. This value is set conservatively to account for extra time spent enumerating files before the send, shuffling files after the send, and possible less-than-optimal communication conditions during the send. This makes it nearly (but not absolutely) certain that the send on science will not be aborted in the middle of a file by the consci timeout.



CAUTION If you need to run a huge `send` command for a large number of files, temporarily raise `u_sci_cmd_max_consci_time` to a potentially huge number for this purpose. (There are 3600 seconds in an hour, 86,400 seconds in a day.). Exercise caution before doing this in the water, and take care to restore the normal setting after the long send. Alternatively, send the files in small batches until all of the files have been transferred.

8 Flight Data Retrieval

The Dockserver can automate file retrieval and data storage, and display data and glider locations for easy viewing. Dockserver applications also allow data transmission via FTP. Flight data can be recorded and recovered in a number of ways. There are a number of nomenclatures described below for the way in which data is stored. Depending on the nature of the mission, the amount of data being transmitted will be customized by users to suit their particular mission needs.

This section is a guide to retrieving data from the glider during a mission surfacing, as well as when the mission has ended. Due to low transmission speed and expensive transmission rates, it is not recommended to send large files, such as .dbd, over the Iridium service. You may inexpensively and relatively quickly retrieve all *.dbd, *.mlg, and *.sbd files by using the FreeWave modem.

A separate document, *GMC User Guide*, provides instructions for using Glider Terminal, a Java application, and one of the Dockserver applications. This user guide can be found at the following location:

<ftp://ftp.glider.webbresearch.com/glider/windoze/production/src/gliderMissionControl/Documentation/gmcUserGuide.pdf>

Table 8-1 Data File Types

Data File Extension	Description
.dbd	Dinkum binary data—All sensors (or variables) are recorded and stored in this type file. These are large files that, in most cases, would be undesirable to transmit during a mission—especially over Iridium.
.sbd	Short binary data—Records only those sensors specified in SBDLIST.DAT to reduce communication time. Customize this list of sensors to receive limited amounts of data during a mission.
.mbd	Medium binary data—A second user specified data set specified in mbdlist.dat.
.mlg	Mission log—Tracks the calls for behaviors and device drives. This is the dialog seen in communication sessions with the vehicle.
.log	Stores the process of opening and closing files and operations.

Data can be accessed and transmitted while using a terminal program while in communication with a vehicle by following the steps outlined below.

To retrieve glider data when the glider is at the surface during a mission:

1. Type `s *.dbd` or `s *.sbd`, or `s *.mlg` or `s *.mbd`.
2. The 30 most recent files of the type specified in step 1 will transmit. Remember, it is not desirable to send .dbd files over Iridium.

To retrieve glider data while the glider is not running a mission from a GliderDOS prompt, type `send *.dbd`, `send *.sbd`, `send *.mbd`, `send *.mlg`, or `send *.*`.

To retrieve science data while the glider is not running a mission from a GliderDOS prompt, type `send *.ebd`, `send *.tbd`, `send *.nbd`, `send *.nlg`, or `send *.*`



NOTE `send *.*` sends all of the files of type `.sbd`, `.mbd`, `.dbd`, `.mlg`, `.tbd`, `.nbd`, `.ebd`, and `.nlg`.

To send specific files in PicoDOS or GliderDOS, use `zmodem` to send the files desired by typing `zs <path><filename>`.

Table 8-2 Immediate Commands Available During Surface Mission Paused Dialog

Command Name	Description
<code>ctrl-r</code>	Resumes glider mission, if so programmed.
<code>ctrl-c</code>	Aborts the mission, closes files, and remains on the surface in GliderDOS.
<code>ctrl-e</code>	Extends surface time by five minutes before resuming the mission.
<code>ctrl-p</code>	Starts the mission immediately.
<code>ctrl-f</code>	Reads the <code>.ma</code> files again.
<code>ctrl-t</code>	Switches to communications with the science Persistor.
<code>!</code>	Followed by an allowable GliderDOS command to run a subset of GliderDOS commands. This <code>!</code> command is also referred to a <i>bang</i> . (Commands listed as lower case in the help menu are not available during a mission.)

9 Missions



NOTE The SFMC glider interface used for mission planning has its own manual which is the *Slocum Fleet Mission Control User Manual*.

Missions and mafile can be loaded to or drawn off the glider at any time by using SFMC software and zs and dockzr commands..

.mi Files

The main mission files that are run are *.mi files. These files contain all the main behaviors and sensor values for the glider. These files can be run independently or they can call mission acquisition (*.ma) files.

The autoexec.mi file is an important and unique .mi file that resides in the config directory. This file controls many of the individual settings particular to each glider. This is where calibrations for motors are stored, where the hardware expected to initialize is drawn from, where the phone number for Iridium dialing to a Dockserver is stored, and from where the glider draws its name.

.ma Files

The mission acquisition (*.ma) files called by *.mi files can contain modified behaviors, waypoint lists, surface instructions, or even sensor values. These files cannot be run independently and are always called from *.mi files. Typically, they are referenced by a number in the *.mi files, and the behavior calling.

For example, for a list of waypoints, the behavior used is goto_list, and it calls a file reference number of 07. Then the *.ma file should be called goto_107.ma and should contain just a list of latitudes and longitudes.

Running Missions

Before running a mission, there are a few steps to follow to ensure that everything is functioning properly.

1. Connect the FreeWave radio to the antenna and a computer.
2. Be prepared to open a terminal connection in SFMC.
3. Power the glider by inserting the go plug (green).

4. Check that the glider is not in Simulation Mode. (simul.sim in the glider Persistor flash card config directory and appcmd.dat in the Persistor flash card config directory must be deleted.)
5. Run `Status.mi` and check the following:
 - Everything is working properly.
 - There is a GPS fix.
 - The batteries are at an acceptable level.
 - No errors appear on the computer screen.

Missions can be run singly or sequenced by typing:

- Run `mission.mi`
- Sequence `mission.mi mission2.mi mission3.mi missionX.mi`
- Sequence `mission.mi(n)`, where n is number of time to run that mission.

A Abbreviations and Acronyms

ABBREVIATION OR ACRONYM	DESCRIPTION
AC or ac	Alternating Current
ASSY	Assembly
BAM	Beam Attenuation Meter
CTD	Conductivity/Temperature/Depth
COTS	Commercial Off-The-Shelf
DC or dc	Direct Current
DG	Dangerous Goods
GLMPC	Glider Mission Planning and Control
GMC	Glider Mission Control
GPS	Global Positioning System
IR	Infrared
ISO	International Organization for Standardization
ISU	Iridium Subscriber Unit
LNA	Low Noise Amplifier
MSDS	Material Safety Data Sheet
OC	Operations Center
OEM	Original Equipment Manufacturer
QCP	Quality Control Process
PPE	Personal Protective Equipment
PTT	Platform Transit Terminal (for Argos)
RHEL	Red Hat Enterprise Linux
RHN	Red Hat Network
RUDICS	Router-based Unrestricted Digital Internetworking Connectivity System
SE	Systems Engineering
SHCS	Socket Head Cap Screw
SN	Serial Number
SOP	Standard Operating Procedure
SSL	Secure Sockets Layer

ABBREVIATION OR ACRONYM	DESCRIPTION
STE	Secure Telephone Equipment
TWR	Teledyne Webb Research
U.S.	United States
USB	Universal Serial Bus
UUV	Unmanned Undersea Vehicle
VAC	Volts Alternating Current

B Code Theory and Operation

The brain of the G3 glider is a Persistor Instruments Inc. CF1 computer chip based on Motorola's MC68CK338 design. The disk space is provided via a removable compact flash card. The Slocum glider also contains a separate Persistor for logging and collecting scientific data (science computer).

Operating Systems

The operating system for the Persistor CF1 is PicoDOS (Persistor Instruments Card Or Disk Operating System). PicoDOS is smaller than, but very similar to DOS, and many DOS commands will work in PicoDOS. Uploads of new glider controller code and mission files, and retrievals of data files are all done in PicoDOS.

GliderDOS is an application that is loaded into the Persistor. This resident operating system is a superset of PicoDOS, which is used to run the glider controller code. Missions are executed from within GliderDOS. It is written in C/C++, compiled using Metrowerks CodeWarrior, and then post linked and uploaded to the glider via Motocross.

Code Design

The user commands the glider by writing *mission files* (text files with an .mi extension) and loading them onto the Persistor. The mission is then executed from within GliderDOS. Mission files are based on a layered single thread approach where tasks are coded into *behaviors* (behavior:_), which are composed of *behavior arguments* (b_arg:_).

During a mission, the glider is continually updating a large number (~2500) of variables. These variables are referred to as *sensors* (sensor:_). In the simplest terms, sensors are defined as any variable whose value is changing or set over the duration of a mission. Some examples of sensor values are GPS fixes, piston displacement, pump position, and CTD values.

All sensor definitions, behavior arguments and behaviors are defined in a file called *masterdata*, which is located at:

<ftp://ftp.glider.webbresearch.com/glider/windoze/production/masterdata>

Only experienced users should manipulate sensors found in masterdata. These sensors can be manipulated by the commands `put`, `longterm_put`, in the headers of missions, or by the interaction of the glider with its missions and environs. Masterdata itself is only edited by compiling a new release of code.

All lines beginning with # are comments. Any line not beginning with # will be acted upon during glider operation.

When a mission is run, GliderDOS checks the sensor values against those in masterdata. If a sensor value is set in the autoexec.mi file, the default value in the masterdata file is overwritten at glider startup. Sensor values that are set in the mission file supersede the sensor values specified in masterdata and autoexec.mi, and any sensor value can be specified in an .ma file, which supersedes all previous value entries. Users can also put sensors to any value from a GliderDOS prompt by typing the put command. Otherwise, the materdata value is used to determine the resulting physical outcome of the glider for that specific mission. Care should be taken when changing any of the masterdata values as they can/will adversely affect glider performance.

Control Levels

The glider controller software contains five layers, or levels, of control. These levels can be visualized in the following hierarchal structure:

- Layered control—Determines which behavior is controlling the glider's movement.
- Dynamic control—Once the layered control determines the behavior, dynamic control moves the motors, inflates and deflates the bladders, etc., to initiate the movement of behavior commanded under the layered control. In order to achieve this, dynamic control uses a specific set of device drivers to perform the movements.
- Device drivers/device scheduler
- Sensor processing—Involves the algorithm calculations to assign values (1 or 0) to the **sensors**.
- Data logging (.dbd, .mbd, .sbd, and .mlg files)—Self explanatory, except that the values of all sensors (variables), instruments, GPS fixes, etc., are actually logged.

Control: Stacks, States, and Abort Sequences

In order to understand this controlled flow structure, it is useful to look at how various types of aborts are initiated and which layers are used to execute the aborts. Much of this information is taken from a text file called **abort-sequences** in

<http://www.glider-doco.webbresearch.com/how-it-works/abort-sequences.txt>

First, the general structure of a mission file and the assignment of priority levels to a particular behavior will be explained. Then, the three types of aborts will be discussed. Following this discussion, the structure of an actual mission will be explained in further detail.

Once the glider has been instructed to execute a mission, GliderDOS reads the mission and assigns a priority to each behavior. The priority is denoted by a numerical assignment that is determined by the physical location of the behavior in the mission text file. The assigned priority list is called a *log_c_stack*, or *stack*, and takes on the following generic form. (The particular behaviors following each number will be explained in further detail later.)

```
log_c_stack():  
1 - abend 2 - surface 3 - set_heading 4 - yo 5 - prepare_to_dive  
6 - sensors_in
```

When one behavior is assigned priority over another and a behavior argument (*b_arg*) is satisfied in both of the behaviors (i.e., if two or more surface behaviors have been written into the mission), the behavior with the higher priority is acted upon.

After the *log_c_stack()* has been created, GliderDOS begins to scroll through the mission in order to activate various sensors and/or behaviors by changing the state of a particular behavior.

Whether a particular behavior changes from one state to another depends upon numerous sensor values.

While a mission is being executed, all behaviors are in one of the following states:

- (1) Uninitiated
- (2) Waiting for Activation
- (3) Active
- (4) Complete

Toward the end of MASTERDATA is a list of numbers and their assigned actions. This list is named *beh_args.h* and can be found in the C code. This list primarily deals with the two *b_arg* statements: *b_arg: start_when* and *b_arg: stop_when*, and determines when a particular behavior becomes active. Only one behavior can be active at a time.

General Control Structure

For a typical glider deployment, mission files are created by layering behaviors in a predetermined sequence to obtain the desired glider path and vertical movement. The glider mission is being executed through the layered control as displayed above. All motor settings are controlled through the behaviors.

There are three types of aborts, which the glider can trigger, to get to the surface:

- Synchronous abort
- Out of band abort
- Hardware generated abort

Each type of abort utilizes different control layers to perform the abort.

Synchronous Abort

A *synchronous abort* occurs when the glider mission does not specify completely what action to take for the current vehicle state. When all of the behaviors listed in the mission file have been called, and the glider actuators (ballast pump, pitch and heading) have not been given a commanded value, dynamic control takes over and brings the glider to the surface. At this point, the actuators are set to: ballast pump full positive buoyancy, pitch nose up, and fin straight forward. This condition continues until the glider detects that it is on the surface, or the user

interrupts with a keystroke (CTRL + C), after which control is returned to GliderDOS. While in the abort condition, all communication and location devices are also turned on (GPS, Argos, etc.), and all system log files are enabled to trace the cause of the abort. Once the abort terminates, control is returned to GliderDOS. You will see a prompt similar to:

GliderDos A # >.

GliderDos A # >. The A stands for abort.

GliderDos N # >. N indicates that the mission ended normally.

GliderDos I # >. I stands for initial, i.e., no mission has been run.

refers to a specific abort code. The abort codes are listed in Appendix C.

For more information about the control levels, see "Control Levels" on page B-2.

Out of Band Abort

During the second type of abort, referred to as an *out of band* abort, the glider assumes that the software is no longer reliable. For this reason, the upper two layers of software control are no longer utilized. Instead, only the device drivers/schedulers are utilized. As with the synchronous abort, the device drivers will be used to achieve positive buoyancy, the last known GPS fix is output and communication devices are turned on. The abort can only be terminated by user keystroke, even though the glider is on the surface. The following dialog is presented:

Want to reset the system? (Y or N)

Want to exit to the operating system?

Make sure you know what you are doing before answering Y

Want to exit to the operating system? (Y or N).

In general, you want to reset the system. When the system is reset, you will be returned to the GliderDOS prompt.

Hardware Generated Abort

The third type of abort is a *hardware generated* abort. The glider hardware is capable of autonomously generating an abort sequence and getting the glider to the surface by triggering the burn wire and dropping the weight. There is a watchdog circuit in the hardware with a time constant of either two hours or 16 hours, set by a jumper in the glider control board and referred to as COP_TICKLE (COP = Computer Operating Properly).

Sample Mission and Comments

The following is from an actual glider mission called *gy10v001.mi*. Black text denotes mission behaviors and behavior arguments (b_args) and is what appears in the mission text and/or masterdata. Blue text denotes comments regarding what each b_arg actually does. Red text denotes .ma files, which are discussed as they are presented.

The following behavior describes the conditions under which the glider must abort. Any text preceded by # is a comment and will not be recognized by the glider code.

behavior: abend

b_arg: overdepth(m)	1000.0	# <0 disables, # clipped to F_MAX_WORKING_DEPTH
b_arg: overdepth_sample_time(s)	10.0	# how often to check
b_arg: overtime(s)	-1.0	# MS_ABORT_OVERTIME # < 0 disables
b_arg: samedepth_for(s)	120.0	# <0 disables
b_arg: samedepth_for_sample_time(s)	30.0	# how often to check
b_arg: no_cop_tickler_for(s)	7000.0	# secs, abort mission if watchdog not tickled #this often, <0 disables

The following *behavior: surface* instructs the glider to come up if it hasn't had communication for a given period of time, in this case, 20 minutes.

behavior: surface

b_arg: args_from_file(enum)	10	# read from mafiles/surfac10.ma.
-----------------------------	----	----------------------------------

The corresponding .ma file is displayed below.

```

behavior_name=surface
# surfac10.ma
# climb to surface with ballast pump full out
# pitch servo'ed to 20 degrees
# Hand Written
# 08-Apr-02 tc@DinkumSoftware.com Initial
<start:b_arg>
# arguments for climb_to
    b_arg: c_use_bpump(enum)          2
    b_arg: c_bpump_value(X)          1000.0
    b_arg: c_use_pitch(enum)          3      # 1:battpos 2:setonce 3:servo
                                           #   in      rad      rad, >0 climb
    b_arg: c_pitch_value(X)          0.3491  # 20 deg
<end:b_arg>

b_arg: start_when(enum)              12      # BAW_NOCOMM_SECS 12, when have
                                           # not had comms for WHEN_SECS secs
b_arg: when_secs(sec)                1200    # 20 min, How long between surfacing, only
                                           # if start_when==6,9, or 12
b_arg: when_wpt_dist(m)              10      # how close to waypoint before surface, only
                                           # if start_when==7. In this example, this
                                           # b_arg is not active.
b_arg: end_action(enum)              1        # 0-quit, 1 wait for ^C quit/resume, 2
                                           # resume, 3 drift til "end_wpt_dist"
b_arg: report_all(bool)              1        # T->report all sensors once, F->just gps
b_arg: gps_wait_time(sec)            120     # how long to wait for gps

```

```

behavior_name=surface
# surfac10.ma
# climb to surface with ballast pump full out
# pitch servo'ed to 20 degrees
# Hand Written
# 08-Apr-02 tc@DinkumSoftware.com Initial
<start:b_arg>
# arguments for climb_to
b_arg: keystroke_wait_time(sec)      300      # how long to wait for control-C
b_arg: end_wpt_dist(m)               0         # end_action == 3 ==> stop when
                                           # m_dist_to_wpt > this arg

```

The following *behavior: surface* instructs the glider to come up when the mission is done. This is determined by a lack of waypoints to direct the glider to in the x-y plane.

behavior: surface

```

b_arg: args_from_file(enum)          10         # read from mfiles/surfac10.ma
b_arg: start_when(enum)              3           # 0-immediately, 1-stack idle, 2-pitch idle,
                                           # 3-heading idle, 6-when_secs,
                                           # 7-when_wpt_dist
b_arg: end_action(enum)              0           # 0-quit, 1 wait for ^C quit/resume,
                                           # 2 resume
b_arg: gps_wait_time(s)              300        # how long to wait for gps
b_arg: keystroke_wait_time(s)        180        # how long to wait for control-C

```

The following *behavior: surface* instructs the glider to come up briefly if yo finishes. This happens if a bad altimeter hit causes a dive and climb to complete in the same cycle. The glider surfaces and the yo restarts.

behavior: surface

```

b_arg: args_from_file(enum)          10         # read from mfiles/surfac10.ma
b_arg: start_when(enum)              2           # 0-immediately, 1-stack idle, 2-pitch idle,
                                           # 3-heading idle, 6-when_secs,
                                           # 7- when_wpt_dist
b_arg: end_action(enum)              1           # 0-quit, 1 wait for ^C quit/resume,
                                           # 2 resume
b_arg: gps_wait_time(s)              300        # how long to wait for gps
b_arg: keystroke_wait_time(s)        15         # how long to wait for control-C

```

The following *behavior: surface* instructs the glider to come up every waypoint.

behavior: surface

```

b_arg: args_from_file(enum)          10         # read from mfiles/surfac10.ma
b_arg: start_when(enum)              8           # 0-immediately, 1-stack idle, 2-depth idle,
                                           # 6-when_secs 7-when_wpt_dist,
                                           # 8-when hit waypoint, 9-every when_secs
b_arg: when_secs(s)                  2700       # How long between surfacing, only if
                                           # start_when==6 or 9
b_arg: when_wpt_dist(m)              10         # how close to waypoint before surface,

```

behavior: surface

b_arg: end_action(enum)	1	# 0-quit, 1 wait for ^C quit/resume, # 2 resume
b_arg: report_all(bool)	0	# T->report all sensors once, F->just gps
b_arg: gps_wait_time(s)	300	# how long to wait for gps
b_arg: keystroke_wait_time(s)	300	# how long to wait for control-C

The following *behavior: surface* instructs the glider to come up when a surface request is made by the science computer.

behavior: surface

b_arg: args_from_file(enum)	10	# read from mfiles/surfac10.ma
b_arg: start_when(enum)	11	# BAW_SCI_SURFACE
b_arg: end_action(enum)	1	# 0-quit, 1 wait for ^C quit/resume, # 2 resume
b_arg: report_all(bool)	0	# T->report all sensors once, F->just gps
b_arg: gps_wait_time(s)	300	# how long to wait for gps
b_arg: keystroke_wait_time(s)	300	# how long to wait for control-C

The following *behavior: surface* instructs the glider to come up every 10 minutes. In this particular case/mission, it is commented out, and therefore not active.

behavior: surface

# b_arg: args_from_file(enum)	10	# read from mfiles/surfac10.ma
# b_arg: start_when(enum)	9	# 0-immediately, 1-stack idle, 2-depth idle, # 6-when_secs, 7-when_wpt_dist, # 8-when hit waypoint, 9-every when_secs
# b_arg: when_secs(s)	600	# How long between surfacing, only if # start_when=6 or 9
# b_arg: when_wpt_dist(m)	10	# how close to waypoint before surface
# b_arg: end_action(enum)	1	# 0-quit, 1 wait for ^C quit/resume, # 2-resume
# b_arg: report_all(bool)	0	#T->report all sensors once, F->just gps
# b_arg: gps_wait_time(s)	300	# how long to wait for gps
# b_arg: keystroke_wait_time(s)	300	# how long to wait for control-C

The following *behavior: goto_list* tells the glider where its waypoints are. For this case, we have again used the .ma convention, which allows us to write a more general mission and insert the particular waypoints/coordinates.

behavior: goto_list

b_arg: args_from_file(enum)	10	# read from mfiles/goto_l10.ma
b_arg: start_when(enum)	0	# 0-immediately, 1-stack idle 2-heading idle

The corresponding .ma file is displayed below.

```

behavior_name=goto_list
# Written by gen-goto-list-ma ver 1.0 on GMT:Tue Feb 19 18:56:54 2002
# 07-Aug-02 tc@DinkumSoftware.com Manually edited for spawars 7aug02 op in buzzards bay
# 07-Aug-02 tc@DinkumSoftware.com Changed from decimal degrees to degrees, minutes,
# decimal minutes
# goto_l10.ma
# Flies a hexagon around R4
<start:b_arg>
    b_arg: num_legs_to_run(nodim)      -1      # loop
    b_arg: start_when(enum)            0      # BAW_IMMEDIATELY
    b_arg: list_stop_when(enum)        7      # BAW_WHEN_WPT_DIST
    b_arg: initial_wpt(enum)           -2      # closest
b_arg: num_waypoints(nodim)           6
<end:b_arg>
<start:waypoints>
-7040.271 4138.861
-7040.271 4138.807
-7040.333 4138.780
-7040.395 4138.807
-7040.395 4138.861
-7040.333 4138.888
<end:waypoints>

```

The following *behavior: yo* instructs the glider to perform a *yo* (i.e., a single up and down pattern through the water column). Again, the .ma convention is used to designate the depth and altitude (together, the range) over which the *yo* is to be performed.

behavior: surface

```

b_arg: args_from_file(enum)           10      # read from mfiles/yo10.ma
b_arg: start_when(enum)               2      # 0-immediately, 1-stack idle 2-depth idle
b_arg: end_action(enum)               2      # 0-quit, 2 resume

```

The corresponding .ma file is displayed below.

```

behavior_name=yo
# yo-c3x5-d20-a3-p20.ma
# climb 3.5m dive 10m alt 20m pitch 20 deg
# Hand Written
# 18-Feb-02 tc@DinkumSoftware.com Initial
# 13-Mar-02 tc@DinkumSoftware.com Bug fix, end_action from quit(0) to resume(2)
# 03-aug-02 tc@DinkumSoftware.com DREA01 at ashument, went to depth only
<start:b_arg>
b_arg: start_when(enum)               2      # pitch idle (see doco below)
b_arg: num_half_cycles_to_do(nodim)   -1      # Number of dive/climbs to perform
                                           # <0 is infinite, i.e. never finishes

# arguments for dive_to
    b_arg: d_target_depth(m)          5
    b_arg: d_target_altitude(m)       -1

```



```

b_arg: d_use_pitch(enum)          3      # 1:battpos 2:setonce 3:servo
                                     #      in      rad      rad
                                     # <0 dive
b_arg: d_pitch_value(X)          -0.3491 # -20 deg

# arguments for climb_to
b_arg: c_target_depth(m)         3.5
b_arg: c_target_altitude(m)      -1
b_arg: c_use_pitch(enum)         3      # 1:battpos 2:setonce 3:servo
                                     #      in      rad      rad
                                     # >0 climb
b_arg: c_pitch_value(X)          -0.3491 # 20 deg
b_arg: end_action(enum)          2      # 0-quit, 2 resume
<end:b_arg>

```

The following *behavior: prepare_to_dive* instructs the glider to get ready to dive after waiting for as long as 12 minutes for a gps fix.

behavior: prepare to dive

```

b_arg: start_when(enum)          1      # 0-immediately, 1-stack idle,
                                     # 2-depth idle
b_arg: wait_time(s)              720    # 12 minutes, how long to wait for
                                     # gps

```

The following 'behavior: sensors_in' turns on most of the input sensors.

behavior: sensors_in

C Abort Codes

```

MS_NONE = -3,
MS_COMPLETED_ABNORMALLY = -2,
MS_COMPLETED_NORMALLY = -1,
MS_IN_PROGRESS = 0,

// Actual aborts start here.
MS_ABORT_STACK_IS_IDLE = 1,
MS_ABORT_HEADING_IS_IDLE = 2,
MS_ABORT_PITCH_IS_IDLE = 3,
MS_ABORT_BPUMP_IS_IDLE = 4,
MS_ABORT_THRENG_IS_IDLE = 5,
MS_ABORT_BEH_ERROR = 6,
MS_ABORT_OVERDEPTH = 7,
MS_ABORT_OVERTIME = 8,
MS_ABORT_UNDERVOLTS = 9,
MS_ABORT_SAMEDEPTH_FOR = 10,
MS_ABORT_USER_INTERRUPT = 11,
MS_ABORT_NOINPUT = 12,
MS_ABORT_INFLECTION = 13,
MS_ABORT_NO_TICKLE = 14,
MS_ABORT_ENG_PRESSURE = 15,
MS_ABORT_DEVICE_ERROR = 16,
MS_ABORT_DEV_NOT_INSTALLED = 17,
MS_ABORT_WPT_TOOFAR = 18,
MS_ABORT_UNREASONABLE_SETTINGS = 19,
MS_ABORT_LMC_NOT_FIXED = 20,
MS_ABORT_NO_HEAP = 21,
MS_ABORT_LOG_DATA_ERROR = 22,
MS_ABORT_THERMAL_NOT_ENABLED = 23,
MS_ABORT_LEAK = 24,
MS_ABORT_VACUUM = 25,
MS_ABORT_NO_HEADING_MEASUREMENT = 26,
MS_ABORT_STALLED = 27,
MS_ABORT_DE_PUMP_IS_IDLE = 28,
MS_ABORT_DE_PUMP_NOT_ENABLED = 29,
MS_ABORT_CPU_LOADED = 30,
MS_ABORT_NO_ABEND_BEHAVIOR = 31,
MS_ABORT_LOW_REL_CHARGE = 32,
MS_ABORT_WEIGHT_DROPPED = 33,
MS_ABORT_INITIALIZATION_ERROR = 34,
MS_ABORT_CRITICAL_ABORT_ACTIVE = 35,
MS_ABORT_INVALID_GPS = 36,
MS_ABORT_NO_COMMS_TICKLE = 37,
MS_ABORT_EMERGENCY_BATTERY_ACTIVE = 38,

// Missing control for buoyancy, pitch, or heading
// No one at the helm
// Missing pitch control
// Missing (shallow) buoyancy pump control
// Missing thermal engine control
// behavior entered error state, usually bad b_args
// went too deep - depth > abend:overdepth
// went too long - time > abend:overtime
// batteries getting low - voltage < abend:undervolts
// not moving vertically for > abend:samedepth_for
// hit control-C
// some required device not producing data
// inflection took too long
// watchdog not serviced, abort before we blow the weight
// Thermal engine pressure too low
// a required device produced too many / severe errors
// a required device is not there
// distance to waypoint > abend:max_wpt_distance
// global sensors are inconsistent
// internal navigation error
// out of memory
// error logging data
// thermal mission without thermal engine
// keeping the ocean out is Job One...
// vacuum < abend:vacuum_min or > abend:vacuum_max
// compass is busted
// not moving horizontally for > abend:stalled_for
// Missing (shallow) pump control
// deep mission without deep engine
// computer too busy
// no abend behavior in mission
// batteries getting low on energy
// weight has been dropped, not going to fly
// empty mission or bad mission sensors
// critical abort has occurred, not going to fly
// too many segments in a row without a valid gps fix
// no comms for this long
// emergency battery is active

```

MS_ABORT_SURFACE_BLOCKED=39,	// expecting ice near surface and can't reach the surface
MS_ABORT_NO_TICKLE_ICE = 40,	// expecting ice near surface and watchdog not serviced
MS_COMPLETED_NORMALLY_UNDER_ICE = 41	// expecting ice near surface, mission completed normally, but no comms/GPS at surface

D Glider Repositories and Forum

In order to download the glider code from the TWR repository, you must arrange for access through the glider forum by registering a user name:

<https://datahost.webbresearch.com/>

A user name and password will be approved by TWR staff by request to glidersupport@teledyne.com. Teledyne Webb Research requires the organization, name, phone number, and email addresses and typically that the user has attended a glider training session.

E Ancillary Equipment

For more information about ancillary equipment for the gliders, see “Ancillary Glider Equipment” in Section 4 of the *Slocum G3 Glider Maintenance Manual*.

F FreeWave Configuration

Following are excerpts from the FreeWave Technologies, Inc. *Spread Spectrum Users Manual*. Refer to www.FreeWave.com for complete details.

About FreeWave Transceivers

FreeWave transceivers operate in virtually any environment where RS232 data communications occur. The transceivers function on a nine-pin null modem cable. If the FreeWave transceivers are to be used in an application where a null modem cable is used, such as communication between two computers, then the FreeWave transceivers can be connected directly. If FreeWave transceivers are to be used to replace a straight-through RS232 cable, then a null modem cable must be placed between the transceiver and the DCE instrument to which it is connected.

Setting up the Glider Shoreside FreeWave

This mode allows a shoreside FreeWave to communicate with several slaves.

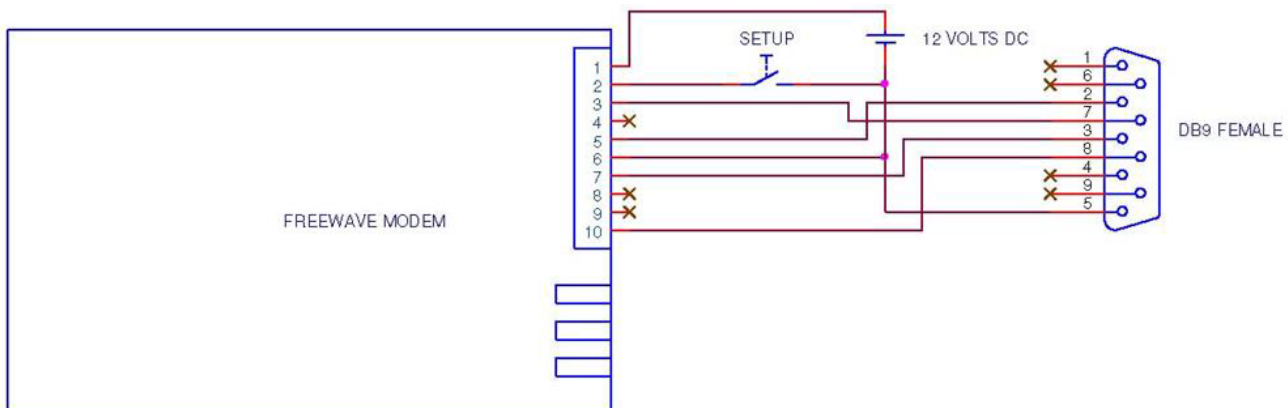
1. Connect the transceiver to the serial port of your computer through a serial cable.
2. Open a HyperTerminal session. Use the following settings to connect with HyperTerminal:
 - Connect to COMx (depending on which COM port your cable is connected to)
 - Set data rate to 19,200, data bits—8, parity—none, stop bits—1, flow control—none.
3. Press the setup button next to the serial port on the back of the radio.
 - The three lights on the board should all turn green, indicating setup mode.
 - The Main menu will appear on the screen.
4. Press 0 to access the Operation Mode menu.
 - Press 0 to set the radio as a point-to-point master.
 - Press Esc to return to the Main menu.
5. Press 1 in the Main menu to change the baud rate.
 - The baud rate in setup mode is always 19200.
 - The baud rate must be changed to match the baud rate of the device to which the radio is attached.
 - Press 1 to set the radio communication baud rate to 115,200.
 - Press Esc to return to the Main menu.
6. At the Main menu, press 3.
 - Set the frequency key.
 - Press 0 to set or change the frequency key.
 - Press 5 to set or change the frequency key to 5.
 - Press Esc to return to the Main menu.

7. At the Main menu, press 2.
 - Press 0 (0 through 9 may be used) to add the serial number of the FreeWave in the glider.
 - Press C to select the glider to which this master will communicate.
 - Press 0 to select entry 0, 1 to select entry 1, etc., or A for all radios on the list for this master to communicate with.
 - Press Esc to return to the Main menu.
 - Press Esc to exit set-up.

Setting up the Glider FreeWave Slave (internal to the glider)

This mode allows a slave to communicate with several shoreside FreeWaves.

1. Connect the transceiver to the serial port of your computer through a serial cable (see the drawing below).



2. Open a HyperTerminal session. Use the following settings to connect with HyperTerminal:
 - Connect to COMx (depending on which COM port your cable is connected to)
 - Set data rate to 19,200, data bits—8, parity—none, stop bits—1, flow control—none.
3. Press and release the Setup button. A 0 volt level on this pin will switch the radio into setup mode.
 - The three lights on the board should all turn green, indicating Setup mode.
 - The main menu will appear on the screen.
4. Press 0 to access the Operation Mode menu.
 - Press 1 to set the radio as a point-to-point slave.
 - Press Esc to return to the Main menu.
5. Press 1 in the Main menu to change the baud rate.

- The baud rate in setup mode is always 19,200.
 - The baud rate must be changed to match the baud rate of the device to which the radio is attached.
 - Press 1 to set the radio communication baud rate to 115,200.
 - Press Esc to return to the Main menu.
6. At the Main menu, press 3.
 - Set the frequency key.
 - Press 0 to set or change the frequency key.
 - Press 5 to set or change the frequency key to 5.
 - Press Esc to return to the Main menu.
 7. At the Main menu, press 2.
 - Press 0 (0 through 9 may be used) to add the serial number of the shoreside FreeWave.
 - Press C to select the glider to which this slave will communicate.
 - Press 0 to select entry 0, 1 to select entry 1, etc., or A for all radios on the list.
 - Press Esc to return to the Main menu.
 - Press Esc to exit set-up.

Choosing a Location for the Transceivers

Placement of the FreeWave transceiver is likely to have a significant impact on its performance. Height is key. In general, FreeWave units with a higher antenna placement will have a better communication link. In practice, the transceiver should be placed away from computers, telephones, answering machines, and other similar equipment. The six-foot RS232 cable included with the transceiver usually provides ample distance for placement away from other equipment. To improve the data link, FreeWave Technologies offers directional and omnidirectional antennas with cable lengths ranging from three to 200 feet. When using an external antenna, placement of that antenna is critical to a solid data link. Other antennas in close proximity are a potential source of interference; use the radio statistics to help identify potential problems. It is also possible that an adjustment as little as two feet in antenna placement can solve noise problems. In extreme cases, such as when the transceiver is located close to pager or cellphone transmission towers, the standard and cavity band pass filters that FreeWave offers can reduce the out-of-band noise.

G Iridium Service and the SIM Card

To obtain an Iridium SIM card, locate and choose a provider. Iridium charges can be a significant expense, so it is worth shopping for a good rate. There are many different plans and providers, and you can use any provider you wish.

Teledyne Webb Research uses Joubeh:

Paul Hill
Sales Manager
JouBeh Technologies Inc.
21 Thornhill Dr.
Dartmouth, Nova Scotia B3B 1R9
Canada
(902) 405-4428, x203
<http://www.joubeh.com/>

Stratos:

(800) 563-2255 (toll-free in North America)
(709) 748-4226 (worldwide)
support@stratosglobal.com
<http://www.stratosglobal.com/>

CLS America or Argos:

(301) 925-4411
userservices@clsamerica.com
<http://www.clsamerica.com/>

NAL Research (offers competitive rates):

(703) 392-1136
contact@nalresearch.com
<http://www.nalresearch.com/Airtime.html>

Infosat Communications:

(888) 524-3038
info@infosat.com
<http://www.infosat.com/>

Specify *data only* service. No equipment is needed, except for a commercial Iridium SIM card.

Billing for the service is monthly. The SIM card is required during the manufacturing process and must be activated 30 days before shipment. Teledyne Webb Research will need the actual card and the unlocking PIN so that the card can be unlocked and the PIN code deactivated permanently.



NOTE Iridium usage based on one of our users averaged 90 minutes per day per glider using some of the data reduction techniques that we provide. This cost is roughly \$108/day. Expect significantly larger usage for your first deployment, because you will be monitoring, testing, and learning. After that, plan on 75-90 minutes per day per glider.



NOTE Depinning SIM cards for the Iridium phone is normally a factory configuration and is only provided to users installing their own card or changing services. For more information on depinning, see the *Slocum G3 Maintenance Manual*.

H Argos Satellite Service and ID

Glider users must provide IDs in both decimal and hexadecimal. The standard repetition rate is 90 seconds. To establish Argos service, submit a program application form to the Argos User Office. They will respond with an acknowledgment and user manual.

To contact Service Argos:

<http://www.argosinc.com>

North American users:

useroffice@argosinc.com

Australia and New Zealand users:

clsargos@bom.gov.au

All other nations:

useroffice@cls.fr

Instructions for Completing the Argos Technical Information Form

Processing:	A2 (hex output)
Results format:	DS
Number of platforms:	Number of IDs
Lifetime of platform:	Number of years glider will be in use
Service required:	Location Platform
Type:	Other -Glider Message
Length:	256 bits, 20-bit IDS
Output power:	1 watt
Platform manufacturer:	Teledyne Webb Research
Platforms model:	Slocum Autonomous Glider
Transmitter manufacturer:	Seimac
Transmitters model:	X-Cat Transmission
Duty cycle:	Other -Period Surfacing/Day

It is helpful if you ask that we receive a copy of the IDs. Type of Argos application: Oceanography User Requirements: Global coverage, low transmitter power, platform compatibility, location, transmitter small size and weight, system access.

It is vital to ensure that the IDs are not in use. Please verify this.

Argos ID format change: Since 1999, CLS Argos has been preparing to change the format of the platform IDs. ~2003 Service Argos began offering two types of ID. The ID format will change from 20 bits to 28 bits. As a result, each 28-bit message will contain 31 data bytes instead of 32. This will change the format of the Argos data. For gliders with 20-bit IDs and X-cat transmitters, refer to the legacy Argos data format for decoding. For gliders with 20- or 28-bit IDs and smart cat transmitters, use rev1 format for decoding. You may need to revise their data processing software accordingly.

Teledyne Webb Research can provide data format information upon request. If your glider is missing and Argos transmissions are needed to find it, contact Service Argos and ask that ALP processing (All Location Processing) be activated.

Argos Data Format

The Argos transmitter in the glider will dictate the Argos data format. If your glider is equipped with the newer X-cat transmitter, the following data format will be transmitted at a 90-second repetition rate while the glider is on the surface. To determine the transmitter type, refer to autoexec.mi and the following sensor and sensor notes: sensor: f_argos_format(enum).

Below is the data format for gliders with X-cat ptts and 28-bit Argos IDs.

<http://www.glider-doco.webbresearch.com/specifications/index.html>

There are tools for automating the unpacking of the Argos data available at:

Forum /glider/windoze/production/windoze-bin

prntargo.exe and prntargo_rev1.exe should be put in the c drive and run from a command prompt to work properly. Please contact glidersupport@teledyne.com to if you are having trouble using this tool while searching for your glider.

I How to Determine Mission Battery Longevity

Mission battery longevity is estimated by considering a number of factors, including the type of battery, style of pump, science sensor types and sampling strategy, surface time required for real time data transmission, and the starting battery voltage.

During an alkaline deployment, monitor the battery voltage from the glider surface dialog or the .sbd data stream by using the masterdata sensor:

```
m_battery(volts)
```

When plotting `m_battery`, note that voltage drops during heavy current usage are expected and normal (i.e., where buoyancy pump adjustments occur at depth).

The glider will begin aborting missions when the voltage drops below 10 volts per the abort behavior below:

```
b_arg: undervolts(volts) 10.0 # < 0 disables
```

When using lithium batteries, it is recommended that the following sensors be monitored in the surface dialog or included in the files:

```
sensor: m_coulomb_amphr_total(amp-hrs) 0.0 # persistent amp-
hours total
```

```
sensor: f_coulomb_battery_capacity(amp-hrs) 720.0 # nominal
battery capacity
```

```
sensor: m_lithium_battery_relative_charge(%) 0 # relative
cumulative charge
```

The above sensors interact to cause a low battery abort behavior from the following `abend` argument:

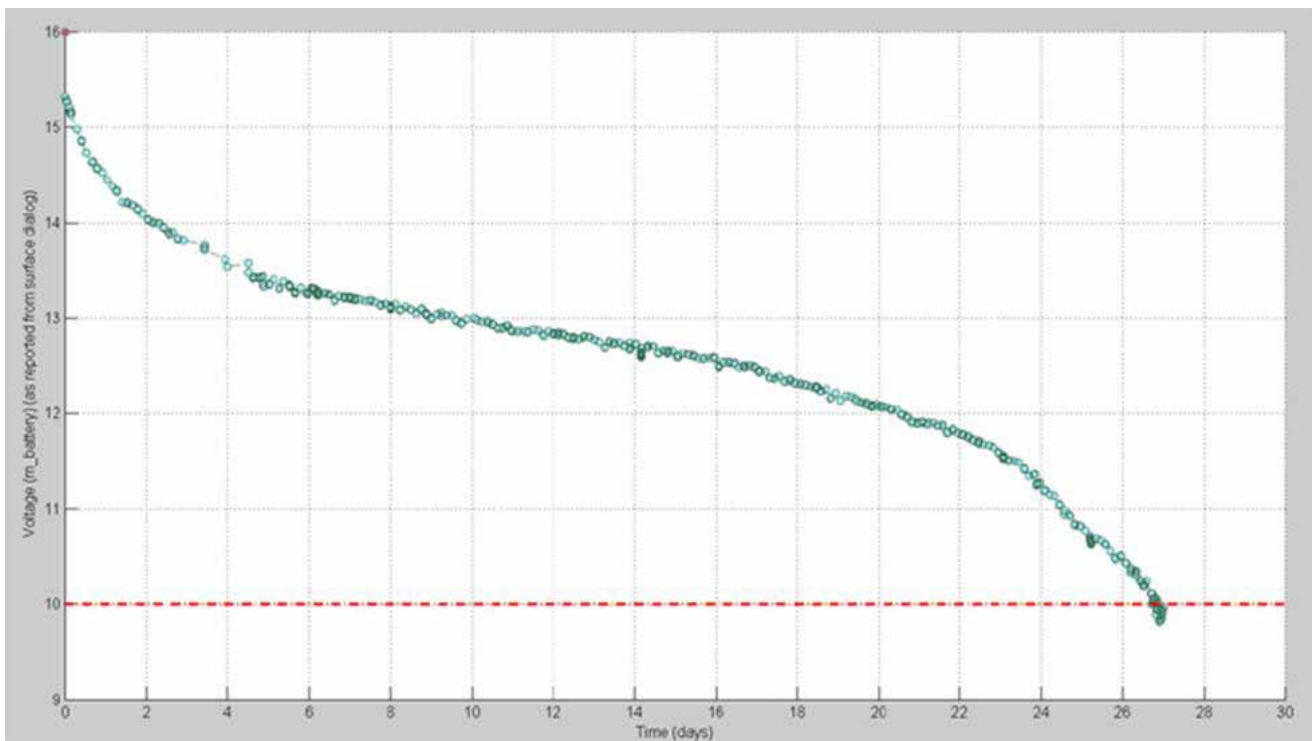
```
behavior: abend
```

```
b_arg: remaining_charge_min(%)10.0 # MS_ABORT_CHARGE_MIN out of
limits
```



NOTE Each time new batteries are installed, the `m_coulomb_amphr_total` sensor should be reset to zero by using the `put` command, followed by an `exit` reset.

To estimate battery longevity, use the 4347 Rev. 02 Glider Endurance Tool, which is found at:
<ftp://ftp.glider.webbresearch.com/glider/windoze/production/src/doco/how-to-calibrate/>



Above is an actual alkaline battery voltage data plot from a 27-day glider deployment. You would typically begin monitoring available recovery scenarios as the voltage approaches 12 volts. In this deployment, recovery became critical as voltage dropped below 11 volts on day 25. In this deployment, while using the factory settings, the glider began to abort all missions for under voltage beginning at day 27.

If you find yourself in a position where recovery is not possible as the voltage drops below 11.5 volts, contact glidersupport@teledyne.com for recommendations for battery preservation, including termination of science sampling, limiting dive depth, limiting communication, and/or drifting at a shallow depth until recovery is possible.

J How to Edit a Proglets.dat File

In order to communicate with the Persistor:

- From PicoDOS, type `consci`.
- from GliderDOS, type `consci`.
- From a mission, type `ctrl-t`.

The Persistor's folder/file structure is shown below.

```
bin
config
autoexec.bat
```

A single science program with a standard configuration for each instrument allows the run-time selection of which instruments are actually in a given glider. A file in config directory called `proglets.dat` controls the wiring and configuration of the science computer. There is a *progle* for each device connected to the science computer.

1. Type `cd config`.
2. Type `dir`.
3. Confirm that `proglets.dat` is in the **config** directory.
4. Transfer `proglets.dat` from the config folder to the Dockserver by typing:
`zs proglets.dat`
5. Preserve the original `proglets.dat` by changing its name. Type `rename proglets.dat proglets.org`.
6. Edit `proglets.dat` in the **from glider** directory.
7. Move the edited `proglets.dat` file to the **To glider** directory on the Dockserver.
8. Confirm that the glider is still in SciDOS. (There is a 1200-second default timeout value.)
9. Type `cd config`, if necessary.
10. Type `dockzr proglets.dat`.
11. When the transfer is complete, type `type proglets.dat`. The new file will now be displayed to screen.
12. Confirm the edits.
13. Type `quit` to exit the science computer.
14. Do one of the following:
 - If in a mission, type `!use - science_super`, and wait for science to power down. Then type `!use + science_super`.
 - If not in a mission, type `exit reset`.

Below is an excerpt from proglets.dat, in which the Aanderaa sensor is commented out to remove it from service.

Original document partial text:

```
#-----  
#Aanderaa Oxygen Optode 3835  
proglet = oxy3835  
    uart  = 3    # U4Soem Pins T-2,R-3 (we only use receive)  
    bit   = 34   # power control for sensor  
    start_snsr = c_oxy3835_on(sec)
```

```
#-----
```

The same partial text is edited to remove the Aanderaa sensor from service:

```
#-----  
#Aanderaa Oxygen Optode 3835  
#proglet = oxy3835  
#    uart  = 3    # U4Soem Pins T-2,R-3 (we only use receive)  
#    bit   = 34   # power control for sensor  
#    start_snsr = c_oxy3835_on(sec)
```

```
#-----
```

K Quick Reference and Checklists

To download or review the most recent *Slocum G3 Glider Operators Manual*, the quick reference, and checklists used to operate the glider, visit:

<ftp://ftp.glider.webbresearch.com/glider/windoze/production/src/doco/MANUAL/>

L Glider Wiring Diagram

To receive the latest version of the G3 glider's wiring diagram, contact
glidersupport@teledyne.com.

