Slocum G3S Glider

Operator Manual

P/N M311172-NFC, Rev. A

Teledyne Webb Research
49 Edgerton Drive
North Falmouth, MA 02556
U.S.A.
Tel: (508) 548-2077
Fax: (508) 540-1686

https://www.teledynemarine.com/webb-research

Notices

Proprietary Information

The information contained herein is the property of Teledyne Webb Research and is considered proprietary. This information may not be used for any other purpose, copied, transferred or disclosed to third parties, reverse engineered, modified or improved without written consent from Teledyne Webb Research.

Terms and Conditions

Teledyne Webb Research's General Terms and Conditions of Sale are hereby incorporated by reference and are located at https://www.teledynemarine.com/webb-research under the *Terms & Conditions* link.

The Seller objects to any different or additional terms except, in connection with an order in furtherance of a government prime contract, those government contract clauses that are required by law to be incorporated herein.

Changes

Teledyne Webb Research reserves the right to make changes to the design or specifications at any time without incurring any obligation to modify previously installed units. In addition, while considerable effort has been made to ensure that the information in this manual is accurate and complete, Teledyne Webb Research assumes no liability for any errors or omissions.

Other Copyrights

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.

Revision History

Revision	Description	Author	Approval	Date
A	Initial Release.	JCC	SQ	7/18/2025

Contents

List of Figures

List of Tables

Preface

This manual provides the information required to operate the Teledyne Webb Research Slocum Glider System. Use in conjunction with the *SFMC User Manual* [P/N M313476-NFC] and the *Slocum G3S Glider Maintenance Manual* [P/N M313476-NFC].

Conventions Used in This Publication

Where applicable, safety information is presented as follows:

WARNING
Identifies a potential hazard that could result in injury or death to the operator or to other personnel.

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

Note
Identifies information of particular interest the reader must be aware of.

Tip
Identifies information the reader might find helpful to: achieve better results, be aware of how an action affects the system, or work more efficiently.

Menu Options and Paths

Menu options are in bold type. Rather than writing out "From the **Admin** menu, select **User Administration**, then select **Users**," angle brackets are used to show the next menu level down or menu option:

Select **Admin > User Administration > Users**.

File Types and Extensions

File names are written as `file.typ`, where **`file`** is the descriptor and **`typ`** is the extension.

- When the text mentions a specific file name, it is written as `file.txt` or something similar.
- When the text mentions file types in general, it is written using the extension in all caps without the period before it; for example, `INI` files or `XML` files.

Typographical Conventions

Font	Description
Bold	The name of a folder, node, path, menu option, icon, or command button that acts like a menu option or command (Open, Save, etc.).
Italic	The name of a window, page, tab, dialog box, panel, area, field, button, or drop-down list within the software interface.
[blue square brackets]	A physical key on the computer's keyboard or device's keypad.
`Monospace` `Monospace & yellow`	A system value or text displayed by the screen or computer.
`Monospace Bold` **`Monospace Bold & green`**	A user value or text the user enters.

Manual Revision Notice

Always confirm you are working with the latest revision of the guide. To verify this guide is the most current, contact glidersupport@teledyne.com.

Protected Documents

Protected documents are accessible by user account through Glider Support.

Many 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.

Reference Documents

The G3S glider is supported by multiple documents:

- The *Slocum G3S Glider Operator Manual* (this publication) outlines the glider's functionality and piloting.
- The *Slocum G3S Glider Maintenance Manual* [`P/N M313476-NFC`] focuses on the hardware and how to assemble and disassemble the vehicle and Lowest Replaceable Units (LRUs); e.g., a GPS board.
- The *SFMC User Guide* [`P/N M313476-NFC`] to explain the Dock Server interface and piloting tools.
- The *G3S Glider Training Slides* serves as a reference during Slocum training sessions.
- The glider spreadsheets that were explained during training

Note
Before operating a glider, it is recommend that all personnel who pilot or work with the gliders: • Participate in a TWR-hosted training class • Become familiar with the material in all these documents

This manual and other materials can be attained by contacting glidersupport@teledyne.com.

Customer Service

We welcome your comments and suggestions for improving our products and documentation as well as developing better ways of serving you. Should you require service or support, contact Teledyne Webb Research customer service using any of the following means:

TELEDYNE WEBB RESEARCH
Attention: Customer Service
49 Edgerton Drive
North Falmouth, MA 02556
U.S.A.

Telephone: (508) 548-2077
Fax: (508) 540-1686

E-mail: webbresearch@teledyne.com

https://www.teledynemarine.com/webb-research

E-mail: glidersupport@teledyne.com

https://www.teledynemarine.com/en-us/support/Pages/Webb-Research.aspx

Part 1: General Information

1 System Overview

Slocum gliders are autonomous underwater vehicles available with optimized engines for the following application types:

- Coastal (shallow)
- Deep sea

Each glider type 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 the following properties of the water:

- Currents
- Sea surface height
- Temperature
- Acoustic
- Optical

These systems enable 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.

1.1 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 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.

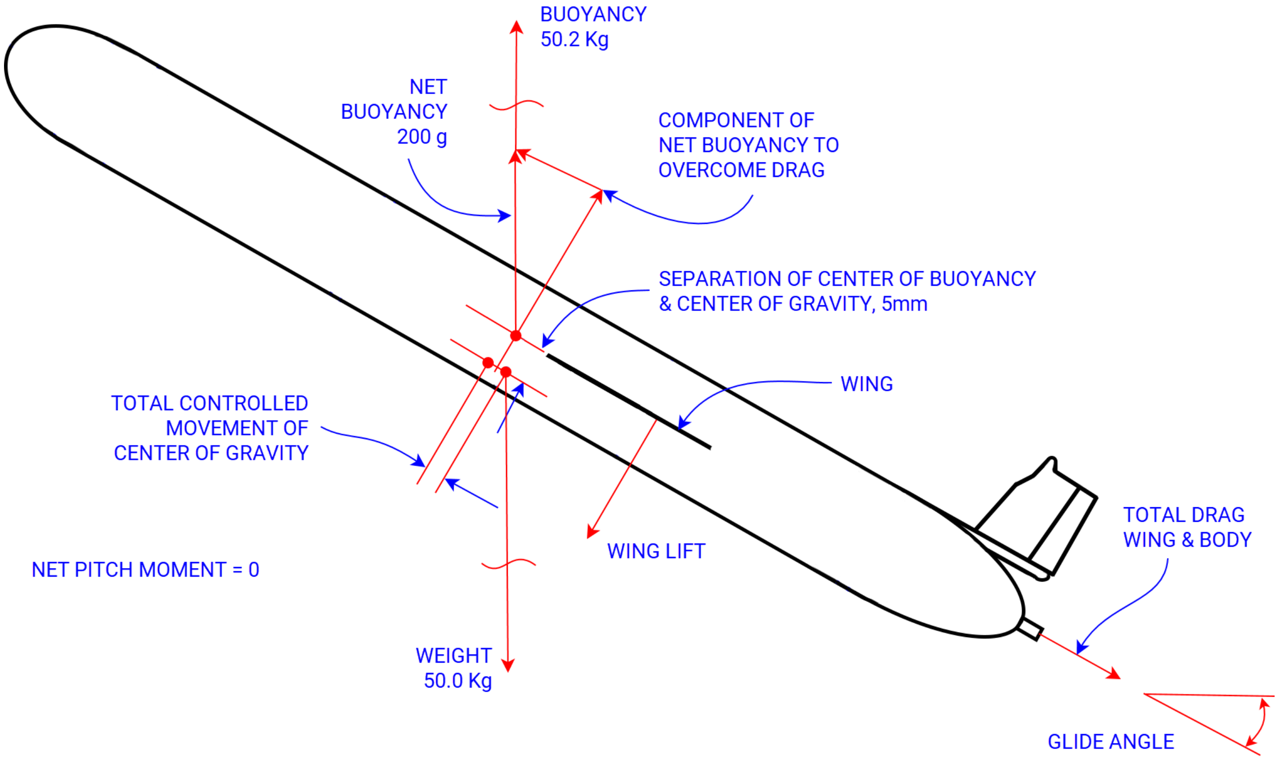

Figure 1-1: Force balance diagram (forces acting on the glider, angle of attack not included)

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.

1.2 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.

1.3 Safety and Handling Procedures

For information on safety and handling procedures for the Slocum glider, refer to the *Slocum G3S Glider Maintenance Manual* [P/N M313476-NFC].

1.4 Sections

The Slocum glider comprises three sections:

- Forward Section - Forward hull with pump assembly, forward batteries, and a "wet" nose dome. A recovery system in the nose dome is optional.
- Mid Section - Mid hull with payload (science) bay; a "wet" payload bay and an extended energy bay are optional.
- Aft Section - Aft hull with main electronics assembly, aft batteries, and a "wet" tail assembly.

This design was chosen for its simplicity, economy, and expandability.

- Main hull sections are made of carbon fiber wound to a specific angle to provide similar compressibility to ocean water
- Nose end cap is a machined pressure resistant elliptical shape
- Tail cap is a truncated cone
- Composite wings are swept backward at 45° and easily replaced using a quick release system.

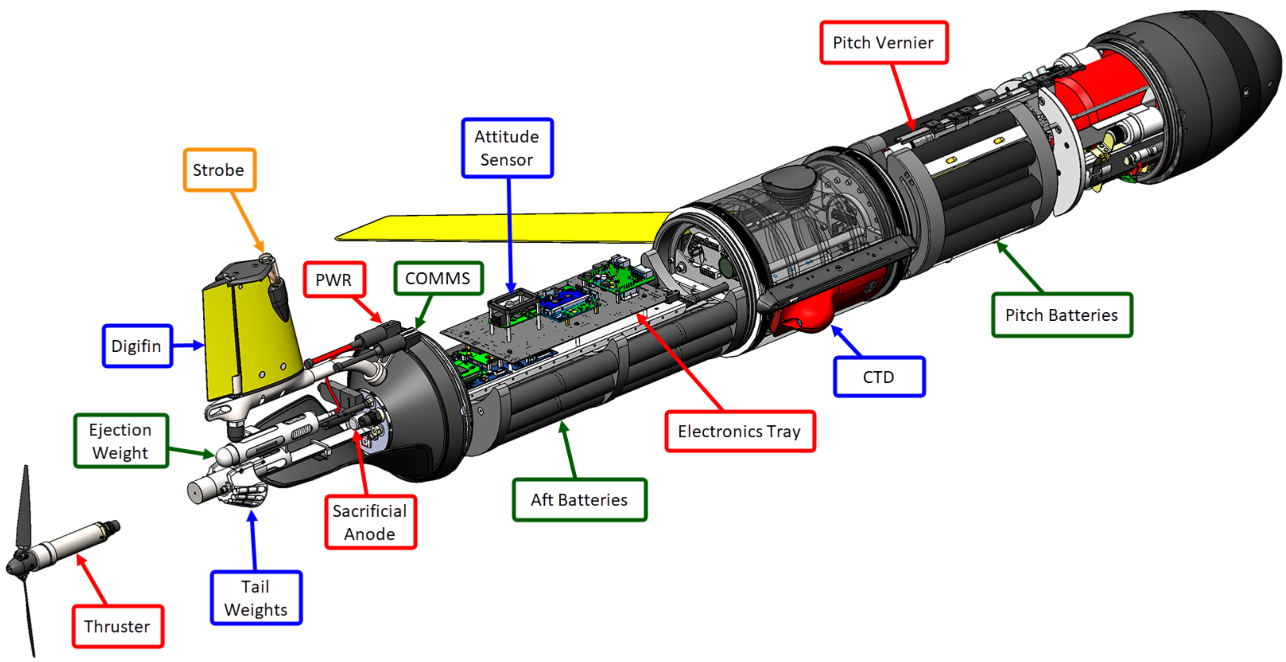

Figure 1-2: Sections and components of the Slocum G3S glider

1.4.1 Forward Section

1.4.1.1 Nose Dome

The nose dome is an acoustically transparent plastic cap located at the front of the glider. It 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.

1.4.1.2 Forward Hull Section

This section houses the ballast pump assembly, pitch motor, forward batteries, and ballast weights.

The pitch motor is used to move the batteries forward or backward to adjust glider pitch in the water.

Internal wiring connectors that attach to the payload bay are located at the aft end of the ballast pump assembly.

1.4.2 Mid Section

1.4.2.1 Payload (Science) Bay

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 3–4 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 Slocum glider has a payload bay computer that is connected to the glider's main computer. This computer controls the sensor packages and collects and stores data via **proglets.dat**, an ASCII file that resides in the config directory.

Each glider and glider type has a unique **proglets.dat** file that depends on the current configuration of scientific instruments. The original source file, which comes in the most recent software release, is found in your memory stick and onboard the glider. See Chapter 2, "Software Architecture" and Chapter 11, "Editing a Proglets.dat File"for more **proglets.dat** file information.

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 the person ballasting the glider 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 "1.4.2.2 Wings" on page 1-4).

1.4.2.2 Wings

The standard carbon fiber wings for the Slocum glider are designed with a 45° sweep angle that is suitable for both shallow and deep gliders. The Slocum's wings are located aft of the center of buoyancy and provide pitch stability during flight.

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

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 available for purchase. These wing rails are slotted to accommodate up to ten 14g stainless steel *bullet weights*. Bullet weights are 13g in water, 15.4g in air.

Bullet weights are typically half-populated when delivered to a customer and the non-used weights are provided separately. Designed for making small ballasting adjustments "in the field," you can add or remove them without opening the sealed parts of the glider. Bullet weights are located in the nose cone, wing rails, and aft section.

1.4.3 Aft Section

1.4.3.1 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 processor) 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 that recombines hydrogen gas and oxygen into water. The catalyst reduces the risk of explosion in the event that alkaline batteries are shorted or come into contact with fresh 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 comms & power ports at the top of the cap.

1.4.3.2 Tail Cone

Above the aft tail cone are the 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 ejection weight and optional thruster are attached.

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
- Iridium system
- Argos transmitter
- radio frequency modem

The strobe light is integrated directly to the fin.

CAUTION

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.

The fin is sturdy enough:

- To guide the glider while it is underway
- When the glider is 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.

1.4.4 External Ballast

G3S gliders are equipped with external ballast in four locations: the buoyancy pump, the port and starboard wing rails, and the tail cowling. Refer to the *Slocum G3S Glider Maintenance Manual* [`P/N M315357-NFC`] for glider ballasting information.

2 Software Architecture

2.1 Slocum Software

Slocum gliders are supported by software release `11.0` or later.

Slocum gliders that have a science bay have the Science Bay Motherboard 3 (SBMB3). Use the Slocum `flash-science.gex` application to update code on the science processor.

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

To acquire the Slocum glider code update procedures, contact glidersupport@teledyne.com.

2.2 Software Control and Hierarchy

The controller code is written in C and architecturally 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 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 its 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* to reduce surface radio transmission time, allowing near real-time data collection.

The glider can have pre-written missions in memory on a CF card. You can:

- Call those missions or create a new mission
- Download the mission(s) to the glider via the RF modem or Iridium
- Run the mission(s)

Mission changes can include:

- Different inflect depths
- New GPS waypoints
- 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** (mission acquisition) files. Those files allow you to transfer very small files to modify the most commonly adjusted mission sensors.

Table 2-1: Software Control Hierarchy

Sensor	Description
`GliderShell`	Glider's shell
`GliderDOS`	Glider's operating system
`masterdata`	Defines the sensors, also known as the glider variables (~2,500)
`longterm.dat`	Maintains the sensors/variables on a power cycle
`autoexec.mi`	Defines glider-specific variables
`*.mi`	Mission files: define mission variables, mission behaviors, & arguments
`*.ma`	Mission acquisition file: define mission behavior variables

2.3 GliderShell

`GliderShell` is the low-level[1] glider shell environment. Typing **`help`** displays the available commands. The navigation devices may be tested in `GliderShell` using the talk program. The talk program syntax is shown in Table 2-2:

Table 2-2: Talk Program Syntax

Command Name	Description
`talk gps`	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.
`talk att`	Turns on the "talk attitude" sensor and displays the pitch, roll, heading, and temperature output.
`talk iridium`	Turns on the Iridium modem and allows you to manually place or receive a call.

`GliderShell` contains the following folder and file structure:

```
Volume in drive C is NONAME
Volume Serial Number is 75E1-51B6
Directory of C:\
       CONFIG
       BIN
       LOG
       MISSION
       SENTLOGS
       STATE
```

1. Contact to the computer only; will not engage physical movement.

2.3.1 CONFIG Folder

The files in the **CONFIG** folder are described in Table 2-3:

Table 2-3: **CONFIG** Folder Files

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

2.3.2 BIN Folder

`GliderShell` executable programs are stored in the **BIN** folder. The programs listed in Table 2-4 run in `GliderShell`:

Table 2-4: **BIN** Folder Files

File Name	Description
`flash-flight.gex`	Re-flashes the flight processor with the version of flight software associated with this "flash-flight.gex" file.
`rekey-flight.gex`	Re-encrypts the flight processor files using a new key. WARNING: Do ***not*** run this program while the glider is out at sea.

2.3.3 LOG Folder

Mission-derived data is stored in the **LOG** folder. The data types are shown by file extension and described in Table 2-5:

Table 2-5: **LOG** Folder File Types

Flight Side Extensions		Science Side Extensions		Description
Regular	Compressed	Regular	Compressed	
`.dbd`	`.dcd`	`.ebd`	`.ecd`	**All binary data** — All sensors turned on for recording are stored in this type file.
`.sbd`	`.scd`	`.tbd`	`.tcd`	**Short binary data** (user-defined) — Records only those sensors specified in sbdlist.dat to reduce file size and thus communication time.

Table 2-5: **LOG** Folder File Types (continued)

Flight Side Extensions		Science Side Extensions		Description
Regular	Compressed	Regular	Compressed	
.mbd	.mcd	.nbd	.ncd	**Medium binary data** (user-defined) — Records only those sensors specified in **mbdlist.dat**.
.mlg	.mcg	.nlg	.ncg	**Mission log** — Tracks the calls for behaviors and device drives.
.log				Stores the process of opening and closing files and operations.

2.3.4 MISSION Folder

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

2.3.5 SENTLOGS Folder

This folder stores the file types listed in the first four rows of Table 2-5 above from log folders that were sent successfully.

2.4 GliderDOS

The operating shell `GliderDOS` is:

- A superset of `GliderShell`
- An application that performs most `GliderShell` functions and has knowledge of the glider

To list the functions available within `GliderDOS`, type one of the following:

```
help  -or-
?
```

All variables used in `GliderDOS` are referred to as *sensors* and are defined in the Masterdata file (see “2.5 Masterdata” on page 2-5).

To list all sensor names of a glider, type:

```
list
```

Behaviors use these values to operate the vehicle.

Note

To acquire the production copy of **masterdata**, please contact glidersupport@teledyne.com.

`GliderDOS` is an application that is loaded onto the processor (**flash-flight.gex**). When 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
- FreeWave on
- GPS on

It is necessary to be in `GliderShell` 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 `GliderShell` or displayed while set to **`boot shell`**.

For a detailed description of how to load a glider and payload bay with a new release of glider production code, contact glidersupport@teledyne.com.

When in `GliderDOS`, use the basic control commands in Table 2-6:

Table 2-6: Basic Glider Control Commands

Command Name	Description
[Ctrl]-**[C]**	Takes control of the glider.
`exit shell`	Sends the glider to `GliderShell` from `GliderDOS`.
`exit reset`	Return to `GliderDOS` (as long as processor is set to boot app).
`boot shell`	Commands the glider to start in `GliderShell` when reset or the power is cycled. Use this when loading an application. Never deploy a glider left to **`boot shell`**.
`boot app`	Commands the glider to start in `GliderShell` when reset or the power is cycled. Use this after loading an application.

2.5 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. To obtain a copy of the **masterdata** file, contact glidersupport@teledyne.com.

2.5.1 Prefixes

The prefixes in **masterdata** are described in Table 2-7:

Table 2-7: **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

2.5.2 Sensor Commands

Sensors can be changed on the `GliderDOS` command line. The sensor commands are described in Table 2-8:

Table 2-8: Glider Sensor Commands

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

2.5.3 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 2-9: Use Command Options

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

Note

The science processor must always be set to boot app so the science application runs the proglets.

Note

In Table 2-10 below, commands listed in all caps are executable during a mission when preceded by an exclamation point (!, *Exe* column). The commands in lower case cannot be used during mission (*NDM* column).

When you enter them in the glider software, commands are not case-sensitive.

Table 2-10: Device Commands

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

Table 2-10: Device Commands (continued)

<table>
<tr><th>Exe</th><th>Command Name</th><th>NDM</th><th>Syntax and/or Description</th></tr>
<tr><td>!</td><td>DEL</td><td></td><td>del [drv:][pth][name] [/p]</td></tr>
<tr><td>!</td><td>DEVICES?</td><td></td><td>Prints device driver information</td></tr>
<tr><td>!</td><td>DF</td><td></td><td>Prints disk space used and disk space free</td></tr>
<tr><td></td><td>digifin</td><td>×</td><td>digifin [dc|fc|lr|pr|rc|rr|rs|sa|sc|st|wr]</td></tr>
<tr><td>!</td><td>DIR</td><td></td><td>dir [d:][p][fn] [/pwblv4a:a]</td></tr>
<tr><td>!</td><td>DUMP</td><td></td><td>dump file[start[,end]] *</td></tr>
<tr><td>!</td><td>ERASE</td><td></td><td>erase [drv:][pth][name] [/P] *</td></tr>
<tr><td></td><td>exit</td><td>×</td><td>exit [-nofin] [poweroff|reset|shell|pbm]</td></tr>
<tr><td>!</td><td>GET</td><td></td><td>get <sensor name></td></tr>
<tr><td>!</td><td>HARDWARE?</td><td></td><td>hardware? [-v]; hardware configuration</td></tr>
<tr><td>!</td><td>HEAP</td><td></td><td>Reports free memory</td></tr>
<tr><td>!</td><td>HELP</td><td></td><td>Prints help for commands</td></tr>
<tr><td>!</td><td>HIGHDENSITY</td><td></td><td>highdensity ?; for help</td></tr>
<tr><td>!</td><td>LAB_MODE</td><td></td><td>lab_mode [on|off]</td></tr>
<tr><td>!</td><td>LIST</td><td></td><td>Displays all sensor names</td></tr>
<tr><td></td><td>loadmission</td><td>×</td><td>Loads mission file</td></tr>
<tr><td></td><td>logging</td><td>×</td><td>logging on|off; during GliderDOS</td></tr>
<tr><td>!</td><td>LONGTERM_PUT</td><td></td><td>longterm_put <sensor name> <new value></td></tr>
<tr><td>!</td><td>LONGTERM</td><td></td><td>longterm ?; for help</td></tr>
<tr><td>!</td><td>LS</td><td></td><td>ls [path] ; list a file system branch</td></tr>
<tr><td>!</td><td>MBD</td><td></td><td>mbd ?; for help</td></tr>
<tr><td>!</td><td>MKDIR</td><td></td><td>mkdir [drive:][path]</td></tr>
<tr><td>!</td><td>MV</td><td></td><td>mv <src_path> <dest_path>; copy a file system branch</td></tr>
<tr><td>!</td><td>PATH</td><td></td><td>path—Shows search path * path [[d:]path[;...]] [/p] *</td></tr>
<tr><td></td><td>prompt</td><td>×</td><td>prompt [text] [/p] *</td></tr>
<tr><td>!</td><td>PRUNEDISK</td><td></td><td>Prunes expendable files to free space on disk</td></tr>
<tr><td>!</td><td>PURGELOGS</td><td></td><td>Deletes sent log files</td></tr>
<tr><td>!</td><td>PUT</td><td></td><td>put <sensor name> <value></td></tr>
<tr><td>!</td><td>RENAME</td><td></td><td>rename [d:][p]oldname newname</td></tr>
<tr><td>!</td><td>REPORT</td><td></td><td>report ?; for help</td></tr>
<tr><td>!</td><td>RMDIR</td><td></td><td>rmdir [drive:][path]</td></tr>
<tr><td>!</td><td>RM</td><td></td><td>rm <path>; deletes a file system branch *</td></tr>
<tr><td></td><td>run</td><td>×</td><td>run [mission_file]; runs the mission file</td></tr>
<tr><td>!</td><td>SBD</td><td></td><td>sbd ?; ? for help</td></tr>
<tr><td>!</td><td>SEND</td><td></td><td>send [-f={rf}|{irid] [-num=<n>] [-t=<s>] [filespec ...]</td></tr>
<tr><td></td><td>sequence</td><td>×</td><td>sequence ?; do this for help</td></tr>
<tr><td>!</td><td>SETDEVLIMIT</td><td></td><td>setdevlimit devicename os w/s w/m</td></tr>
<tr><td>!</td><td>SETNUMWARN</td><td></td><td>setnumwarn [x]; sets max dev warnings to x</td></tr>
<tr><td>!</td><td>SET</td><td></td><td>set [var=[str]] [/slfe?] *</td></tr>
</table>

Table 2-10: Device Commands (continued)

Exe	Command Name	NDM	Syntax and/or Description
!	SIMUL?		Displays a print description of what is simulated
!	SRF_DISPLAY		srf_display ?; for help
	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

2.6 Science Computer

To communicate with the science processor:

- From either `GliderShell` or `GliderDOS`, type:

```
consci
```

- From a mission, press: **[Ctrl]-[T]**

The science processor's folder structure is shown in Figure 2-1:

Figure 2-1: STM32 processor folder structure

Note

When in communication with science, a loss of carrier detection automatically switches control and communications back to the glider controller.

The **proglets.dat** file 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 proglet.

Each proglet within the **proglets.dat** file can be modified to reflect changes to the science payload's configuration and/or instrumentation. Some of the available instruments are listed below:

Table 2-11: Selected Devices Connected to the Science Computer (1 of 3)

Device Name	Description
ad2cp	Nortek, AD2CP Acoustic Doppler Current Profiler
amar	JASCO Autonomous Multichannel Acoustic Recorder sensor
auvb	Wet Labs AUV-B Fluorometer
azfp	ASL Acoustic Zooplankton Fish Profiler (AZFP), echosounder
badd	Benthos Acoustic Data Delivery system
badd_mmp	Benthos Acoustic Data Delivery system using Modem Management Protocol (MMP) API
bam	Benthos Acoustic Modem
bb2c	Wet Labs, bb2c sensor
bb2f	Wet Labs bb2f fluorometer / backscatter sensor
bb2fls	Wet Labs bb2flslk: custom 3-param scattering meter/fluorometer
bb2flsV2	Wet Labs bb2flslk: custom 3-param scattering meter/fluorometer
bb2flsV3	Wet Labs bb2flslk: custom 3-param scattering meter/fluorometer
bb2flsV4	Wet Labs bb2flslk: custom 3-param scattering meter/fluorometer
bb2flsV5	Wet Labs bb2flslk: custom 3-param scattering meter/fluorometer
bb2flsV6	Wet Labs bb2flslk: custom 3-param scattering meter/fluorometer
bb2flsV7	Wet Labs bb2flslk: custom 3-param scattering meter/fluorometer
bb2flsV8	Wet Labs bb2flslk: custom 3-param scattering meter/fluorometer
bb2flsV9	Wet Labs bb2flslc: custom 3-param scattering meter/fluorometer
bb2fV2	Wet Labs bb2f fluorometer / backscatter sensor
bb2lss	Wet Labs, Light Scatter Sensor
bb3slo	Wet Labs bb3slo: backscatter triplet sensor
bb3sloV2	Wet Labs bb3slo: backscatter triplet sensor, 2nd configuration
bb3sloV3	Wet Labs bb3slo: backscatter triplet sensor, 3nd configuration
bbfl2s	Wet Labs bbfl2slo: custom 3-param fluorescence/scattering meter
bbfl2sV2	Wet Labs bbfl2slo: custom 3-param fluorescence/scattering meter
bsipar	Biospherical Instruments PAR sensor
c3sfl	Turner Designs C3 Submersible Fluorometer
ctd41	Sea-bird CTD(SBE-41) "old" pulsed style
ctd41cp	Sea-bird CTD(SBE-41CP) continuous profiling
ctd41cp2	CTD41 Continuous Profile 2 (allows to operate 2 CTD pumps at the same time)
dmon	W.H.O.I. Digital Monitor, a passive acoustic monitor.
dvl	Teledyne RDI ExplorerDVL
echodroid	ODROID Echo Sounder sensor

Table 2-11: Selected Devices Connected to the Science Computer (2 of 3)

Device Name	Description
echosndr853	Imaginext Technology Modem 853 Echo Sounder with Data Logger
extctl	External Controller (Backseat Driver)
FIRe	Fluorescence Induction and Relaxation electronics
fl2PeCdom	Wet Labs fl2slc: Phycoerythrin and CDOM sensor
fl2UrRh	Wet Labs fl2slc: Uranine and Rhodamine sensor
fl3slo	Wet Labs fl3slo: fluorescence triplet sensor
fl3sloV2	Wet Labs fl3slo: fluorescence triplet sensor, 2nd configuration
flbb	Wet Labs fluorometer and scattering meter
flbbbbV1	Wet Labs flbbbbslc: Fluorometer (ug/l), scatter, scatter sensor
flbbbbV2	Wet Labs flbbbbslc: Fluorometer (ppb), scatter, scatter sensor
flbbcd	Wet Labs fluorometer, scattering meter, and CDOM sensor
flbbrh	Wet Labs fluorometer, scattering meter, and rhodamine sensor
flntu	Wet Labs FLNTU: fluorescence and turbidity sensor
flrh	Wet Labs Rhodamine Fluorometer sensor
flur	Wet Labs uranine sensor
glbps	ASL GLBPS SONAR Device
hydrophone	DREA hydrophone sampler
lisst	Sequoia Laser In-Situ Scattering and Transmissometry (LISST)
lms	Franatech Laser Methane Sensor (LMS)
logger	generic data logger
microRider	Rockland Scientific MicroRider MR-1000 (MICRORIDER)
miniProCO2	Pro-Oceanus Mini-Pro CO2 sensor
MoteOPD	Mote Marine Laboratory Optical Phytoplankton Discriminator
nbctd	Neil Brown CTD (W.H.O.I.)
obsvr	JASCO Observer Hydrophone sensor
ocr504I	Satlantic OCR-504 Irrandiance Sensor
ocr504R	Satlantic OCR-504 Radiance Sensor
ocr507I	Satlantic OCR-507 Irrandiance Sensor
ocr507R	Satlantic OCR-507 Radiance Sensor
oxy3835	Aanderaa Oxygen Optode 3835
oxy3835_wphase	Aanderaa Oxygen Optode 3835
oxy4	Aanderaa Oxygen Optode 4330F or 4831
pCO2	Aanderaa pCO2 sensor
rbrctd	RBR logger CTD sensor
rbrodo	RBRcoda^3 T.ODO sensor, Temperature & Dissolved Oxygen Sensor
rbrtridente	RBRtridente Eco Puck
rinkoII	JFE ALEC CO., RINKO-II dissolved oxygen and temperature sensor
sam	Wet Labs, Scattering Attenuation Meter

Table 2-11: Selected Devices Connected to the Science Computer (3 of 3)

Device Name	Description
satpar	Satlantic PAR sensor
sbe41n_ph	Sea-bird SBE41N-pH sensor
seaOWL	Wet Labs SeaOWL sensor, Sea Oil-in-Water Locator
solocam	Williamson camera
sscsd	SPAWAR Acoustic array
suna	Satlantic Submersible Ultraviolet Nitrate Analyzer
svs603	SeaView Systems Inertial Wave Sensor SVS-603 (SVS603)
tarr	OASIS Towed Array Receiver/DSP
tau	Sequoia LISST-Tau, Beam Attenuation Meter sensor
ubat	Wet Labs Underwater Bioluminescence Assessment Tool (UBAT)
uModem	W.H.O.I. Acoustic Micro-Modem
uviluxPAH	Chelsea Technologies, UviLux PAH sensor
viper	MDA Viper Processor
vr2c	Vemco VR2C Submersible, Dual Single-Channel Acoustic Receivers
vro	InnovaSea Vemco Receiver Offload module (VRO) sensor
vsf	Wet Labs Volume Scattering Function meter
wetlabsA	Wet Labs generic proglet A
wetlabsB	Wet Labs generic proglet B
wetlabsC	Wet Labs generic proglet C
whfctd	W.H.O.I. Fast CTD
wispr	Embedded Oceans Systems Passive Acoustic Monitoring

2.6.1 Bin Folder

GliderShell executable programs are stored in the **bin** folder. All of these programs run in ScienceShell.

These files are described in the table below:

Table 2-12: Files in the **bin** Folder

File Name	Description
flash-science.gex	Re-flashes the science processor with the version of science software associated with this flash-science.gex file.
rekey-science.gex	Re-encrypts the science processor files using a new key. WARNING: Do NOT run this program while the glider is out at sea.

2.6.2 Config Folder

Note

You must delete the **appcmd.dat** file before an actual flight.

The files in the **config** folder are described in the table below:

Table 2-13: 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.

Part 2: Glider Components

3 Forward Section Components

3.1 HD Pumps Overview

CAUTION
Before ballasting with HD pumps, you *must* ensure ***all*** air is removed from the buoyancy pump. Otherwise, there is significant risk of incorrect ballasting.

Slocum gliders provide two optimized capabilities to use either a:

- Hydraulic pump for deep applications, –or–
- Piston-driven pump for shallower applications

Both pump designs share an external nose geometry that 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. The external weights reduce the need to open the glider to make fine adjustments.

The hydraulic deep (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 opens this valve when it is at the surface to retract oil from the external bladder and begin the dive. It remains 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—depending on how the mission is configured—either open the valve to dive again or remain closed to continue to the surface.

  In this typical use case, the valve never needs to be 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, the high pressure push has been dealt with by using a flow restricter to ensure that the internal plumbing never experiences that higher pressure.

  The 3-way ball valve is important because it allows for much faster inflections in the unrestricted position, which in turn 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.

Slocum gliders are 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, use the full volume.
- 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

3.2 Shallow Ballast Pump Assembly

Note

Shallow pumps (rolling diaphragm or bellophragm) 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 in/Hg less from the external atmosphere to ensure that the diaphragm folds smoothly as it rolls; otherwise, damage may result.

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-, 100-, and 200-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.

3.2.1 How to Configure

The shallow ballast pump assembly is factory configured. Please contact Glider Support at glidersupport@teledyne.com for assistance.

3.2.2 How to Test

From `lab_mode`:

1. Type:

```
wiggle on
```

2. Type:

```
report ++ m_ballast_pumped
```

3. Ensure the ballast pump successfully completes the following without errors:
   - a full extension (`m_ballast_pumped= +400 cc`) –and–
   - a full retraction (`m_ballast_pumped= -400 cc`)
4. Type:

```
wiggle off
```

3.2.3 How to Evaluate Data

With a properly ballasted glider:

- Positive cubic centimeters (cc) result in positive buoyancy and a climb.
- Negative cc result in negative buoyancy and a dive.

3.2.4 Relevant Sensors

Table 3-1: Relevant Sensors: Shallow Ballast Pump Assembly

Sensor Name	Description
m_ballast_pumped	Measured volume of buoyancy pumped in cc
c_ballast_pumped	Commanded volume of buoyancy pumped in cc

3.3 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.

3.3.1 How to Configure

The 1000-meter ballast pump assembly is factory configured. Please contact Support at glidersupport@teledyne.com for assistance.

3.3.2 How to Test

From `lab_mode`:

1. Type:

```
wiggle on
```

2. Type:

```
report ++ m_de_oil_vol
```

3. Ensure the ballast pump successfully completes the following without errors:
   - a full extension (`m_ballast_pumped= +430 cc`) –and–
   - a full retraction (`m_ballast_pumped= -430 cc`)

4. Type:

```
wiggle off
```

3.3.3 How to Evaluate Data

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

3.3.4 Relevant Sensors

Table 3-2: Relevant Sensors: Ballast Pump Assembly

Sensor Name	Description
de_oil_vol	Deep pumps
m_de_oil_vol	Measured volume of buoyancy pumped in cc
c_de_oil_vol	Commanded volume of buoyancy pumped in cc
ballast_pumped	Shallow pumps
m_ballast_pumped	Measured volume of buoyancy pumped in cc
c_ballast_pumped	Commanded volume of buoyancy pumped in cc

3.4 Pitch

The pitch system works in hand with the buoyancy system to set the dive and climb angle.

Provided that the H-moment is 6 ±1 mm, the fluid movement from the ballast pump assembly provides the moment for changing pitch.

As the vehicle's volume is changed via the buoyancy engine, the pitch mechanism allows for an efficient 26° angle to achieve a dive or climb.

To trim to the desired dive and climb angles, a lead screw drives the forward ~10 kg battery pack fore or aft.

For more information, see the *Slocum G3S Glider Maintenance Manual* [P/N M315357-NFC].

During surfacing, the battery pack is moved all the way forward to better raise the tail out of the water for communications.

3.4.1 How to Configure

The pitch vernier is factory configured. Please contact Support at glidersupport@teledyne.com for assistance.

3.4.2 How to Test

> **Note**
>
> The length of a full extension and retraction depends on the type of buoyancy pump (shallow or deep) and battery (alkaline or lithium). The value of *f_battpos_safety_max* is specified in the glider's **autoexec.mi** file.

From `lab_mode`:

1. Type:

```
wiggle on
report ++ m_battpos
```

2. Ensure the pitch battery successfully completes a full extension and retraction without errors.

3. Type:

```
wiggle off
report clearall
```

3.4.3 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.

For example, moving the battery toward the aft of the vehicle will lift the nose and moving the battery forward will bring the nose down.

3.4.4 Relevant Sensors

Table 3-3: Relevant Sensors: Pitch Vernier

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

3.5 Altimeter

The altimeter is mounted on the front of the ballast pump assembly. Its electronics are supported on the cylinder of the ballast pump assembly. The transducer leads go 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.

3.5.1 How to Configure

The altimeter is factory configured. Please contact Support at glidersupport@teledyne.com for assistance.

3.5.2 How to Test

Concerning the standard altimeter (known simply as "the altimeter") in the lab:

1. Ensure the altimeter is updating as frequently as possible, type:

```
put c_alt_time0
```

2. Listen closely to the altimeter to check for audible clicks similar to the ticking of a watch.
3. Type:

```
report ++ m_altimeter_voltage
```

4. Confirm the altimeter voltage is not a fixed value.
5. Type:

```
report clearall
```

Concerning altimeter 232 (also known simply as "the 232") in the lab:

1. Ensure the 232 is updating as frequently as possible, type:

```
put c_alt_time0
```

2. Listen closely to the 232 to check for audible clicks similar to the ticking of a watch.
3. Type:

```
report ++ m_altimeter_voltage
```

4. Confirm the 232 voltage is not a fixed value.
5. Type:

```
report clearall
```

3.5.3 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 dictates the necessary value to achieve successful bottom avoidance. The suggested minimum values for *d_target_altitude* are:

Table 3-4: How to Evaluate Data: Altimeter

Pump Type	Suggested minimum value for *d_target_altitude*
1000 m	12 m
350 m	6 m
200 m	6 m
100 m	4 m
50 m	2 m

3.5.4 Relevant Sensors

Table 3-5: Relevant Sensors: Altimeter

Sensor Name (Units)	Description
m_altitude (m)	Height above the bottom.
m_raw_altitude (m)	Height above bottom, unfiltered.
u_alt_min_depth (m)	How deep the vehicle must be before the altimeter turns on. A glider pilot may choose to increase *u_alt_min_depth* 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.
u_alt_reqd_good_in_a_row (nodim)	How many *m_raw_altitude* readings are used to calculate the processed *m_altitude* measurement.
u_alt_filter_enabled (bool)	Enable median filter depth for altitude.
u_alt_reduced_usage_mode (bool)	The glider calculates and uses the altimeter only when necessary. A glider pilot might decide to activate or deactivate Reduced Usage mode. Reduced usage saves energy; however, it may result in ineffective results in shallow water deployments.
m_water_depth (m)	Calculates the depth of the water by adding *m_depth* and *m_altitude*.
u_max_water_depth_lifetime (yos)	How long to use *m_depth* only in absence of new altimeter fixes.

3.6 Recovery System (optional)

3.6.1 How to Configure

See the *Slocum Glider Maintenance Manual* [P/N M313476-NFC] for installation and configuration.

3.6.2 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.

3.6.3 Relevant Sensors

Table 3-6: Relevant Sensor: Recovery System

Sensor	Description
c_recovery_on	**0** — Not triggered. **1** — Recovery system activated. Unlike the ejection weight, it requires several minutes to release.

3.7 Carbon Fiber Hulls

CAUTION

Any nicks in the paint finish must be examined to ensure the damage does not extend into the fiber winding.

Note

Careful inspection of interior and exterior finishes must be performed regularly—especially in the area of the sealing surface to the O-ring.

Carbon fiber hulls are custom wound, compressible, engineered parts and must be handled with care.

3.7.1 How to Test

Note

The vacuum will fluctuate with the temperature.

Pull and maintain the recommended vacuum on the vehicle (6 in/Hg for shallow gliders and 7 in/Hg for deep gliders), see *Slocum Glider Maintenance Manual* [P/N M315357-NFC], Chapter 4, "Checking and Setting the Vacuum".

3.7.2 How to Evaluate Data

Monitor the vacuum. Vacuum is affected by temperature; increases with cooling and decreases with warmth.

3.7.3 Relevant Sensors

Table 3-7: Relevant Sensor: Carbon Fiber Hulls

Sensor	Description
m_vacuum	Measured internal vacuum

3.8 Spare Ports and Modem Readiness

Slocum gliders can be equipped with a number of through-the-hull external connectors that can be customized based on operator requirements.

Figure 3-1: Example of an external connector

The Slocum 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.

Concerning external connectors:

1. 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.
2. Before connecting any external connectors, flush those connectors with fresh water to ensure that any particulates are rinsed from conductive surfaces.
3. Before 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.

The Slocum glider has additional wet payload space available in the nose fairing that can be extended, pushing the nose cone out to provide significant wet volume for external sensors.

Note

This optional expanded volume does not support the nose recovery system. Please contact glidersupport@teledyne.com if you are interested in using this volume.

The nose fairing has a check valve. It is designed to prevent silt from entering and collecting inside the nose cap.

4 Payload (Science) Bay Components

The payload bay—also referred to as the "science bay"—is located in the glider mid section and includes:

- Science bay motherboard
- Expansion board
- Science processor board

These boards are mounted on a tray inside the payload bay (Figure 4-1).

CAUTION
Copy the Slocum **science** folder to your science MicroSD card. You must get *all* the files; just re-flashing the app will not suffice.

There are three revisions of the motherboard included on all gliders. The three revisions are different enough to require separate programs for each board.

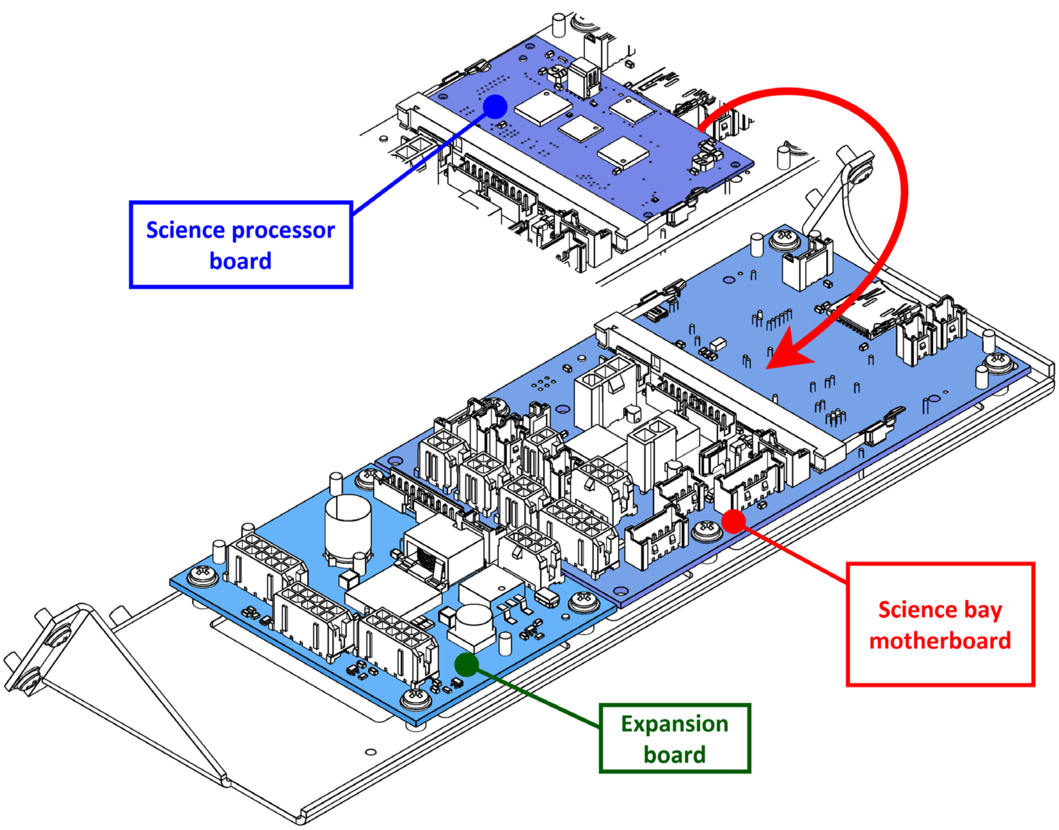

Figure 4-1: Slocum G3S motherboard, expansion board, and science processor board

4.1 Payload Bay Connectors

The forward plate and connectors to the payload bay are covered in the *Slocum Glider Maintenance Manual* [P/N M315357-NFC].

Note

Not all payload bays have the same connectors. Connectors will depend on payload sensor/device configuration.

Connectors shown in Figure 4-2:

- Pass-through
- Battery, Forward Pump, and switched power

  Molex P/N: 0039012246, Mating P/N: 0039012245
- Signals

  Amphenol P/N: MS27513E 20 B35S
- More battery and switched power options

  These vary, depending on the configuration. For example, this may be a 20-pin, 14-pin, or 10-pin connector.

The Slocum glider includes a large CPC connector at the aft of the payload bay. The operator does not need to reach inside the hull to attempt to mate the connectors.

However, if you want to access the inside of the payload bay, you must remove the forward guard plate.

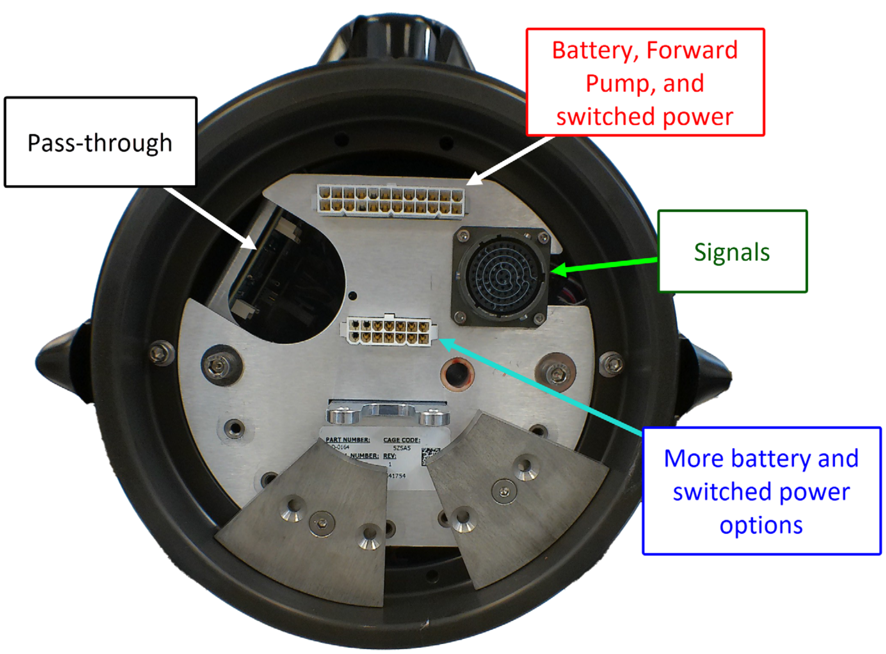

Figure 4-2: G3S payload bay aft bulkhead connectors

4.2 Payload Bay Sensors

Each science sensor is controlled by a proglet (a small program to control science sensors). The proglet:

- Sets the power bit
- Sets the **UART** command on the payload bay controller
- Expects formatted data from its associated sensor

A **UART** is a command that powers on a specific port and allows the user to interact with one sensor at a time.

4.2.1 How to Test

1. From `GliderDOS`, type:

```
load mission sci_on.mi
```

   All payload bay sensors are turned on and their data is printed to the terminal.

2. When you have determined that all sensors are reporting data properly, type the following to turn off the science sensors:

```
load mission sci_off.mi
```

Each sensor can be addressed directly using a UART. Contact glidersupport@teledyne.com for details.

4.2.2 How to Evaluate Data

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

4.2.3 Relevant Sensors

The relevant sensors are unique to each payload bay.

4.3 Conductivity, Temperature, and Depth (CTD)

A Slocum glider does not require a CTD sensor; however, a typical sensor package contained in the payload bay on the glider includes a CTD pumped conductivity/temperature/depth sensor.

The CTD sensor is delicate and should be protected from abuse.

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

CAUTION
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 a fresh water lake or pond and accurate CTD readings are required, this setting can be adjusted. For more details, contact glidersupport@teledyne.com. 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. The CTD can be used in an emergency by changing the sensor *u_use_ctd_depth_for_flying*.

CTD data and informationSlocum is located at:

`https://www.seabird.com/glider-payload-ctd-gpctd/product?id=60762467712`

Select the **Downloads** tab on that page to access data sheets, user manuals, and more.

4.3.1 How to Configure

Manufacturer and factory configured.

4.3.2 How to Test

Note

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

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.

4.3.3 How to Evaluate Data

See manufacturer's documentation.

4.3.4 Relevant Sensors

Table 4-1: Relevant Sensors: CTD

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

4.4 Desiccant

CAUTION
Install desiccant packs to absorb internal moisture and replace them for each deployment.

Note
If possible, open and seal the glider in a controlled, dry environment.

When the glider is open for long periods, store the desiccant in sealed plastic bags to prevent it from being saturated by atmospheric moisture.

A fully saturated desiccant bag can increase in mass (by ~40 grams) and, therefore, affect ballasting.

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

4.4.1 How to Configure

Keep dry. When the glider is open, desiccant gradually gains mass from the atmosphere. Desiccant mass increase affects glider ballasting and can render the desiccant ineffective.

4.4.2 How to Test

Monitor desiccant weight. The desiccant pouches provide by Teledyne Webb Research weigh approximately 114 grams when fully dried.

4.5 Lifting Point

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

The payload bay can be outfitted with a lifting point.

5 Aft Section Components

5.1 Catalyst

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 fresh water, 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 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. You must accept and deal with this risk to use this instrument safely as provided. Only personnel with knowledge and training to deal with this risk should seal or operate the instrument.

When a glider uses alkaline batteries, a catalyst is used to recombine hydrogen and oxygen into water.

The catalyst reduces the risk of explosion should the batteries begin to “outgas” (release or give off a substance as a gas or vapor). The reaction is exothermic and the catalyst may become hot.

This item does not need periodic replacement.

5.2 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 (4.309223 kPa). 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.

5.2.1 How to Configure

Factory configured. The air systems differential pressure switch is set to 3.0 in/Hg (76.2mmHg, 101.5917mbar) during manufacturing.

5.2.2 How to Test

Note
Do not fully inflate unless the cowling is installed.

1. To activate the air pump and inflate the external air bladder in the lab, type:

```
put c_air_pump 1
```

   Full inflation completes in 3–10 minutes.

2. To open the solenoid and deflate the external air bladder, type:

```
put c_air_pump 0
```

   The glider's internal vacuum increases by 1.5–2.5 in/Hg (1.03421–1.72369 kPa) as atmosphere from the vehicle is moved into the external bladder.

3. The switch stops the air pump when the pressure differential reaches 1.5–2.5 in/Hg.

Note
Longer duration inflation times may be an indication of system trouble.

   Pump inflation typically completes in under 5 minutes.

5.2.3 How to Evaluate Data

The air bladder activates per the above sensors at the surface. The *m_vacuum* sensor increases when functioning properly.

5.2.4 Relevant Sensors

Table 5-1: Relevant Sensors: Air Pump System

Sensor	Description
c_air_pump	units (in/Hg)
m_air_pump	units (in/Hg)
m_vacuum	units (in/Hg)

5.3 Flight Controller

The glider's functions are controlled by an STM32 processor. Controller code is written in C and architecturally 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, then post-parsed into a matrix that allows the user to easily construct graphical views of vehicle performance or scientific data.

Sensor subsets can be selected as a "science data package" to reduce surface radio transmission time.

Missions can be altered to change the depth of inflections, send new GPS waypoints to the glider, and so on. For additional reading regarding the construction of binary data or the family of **.dbd** files and their contents, visit:

For more information, contact glidersupport@teledyne.com.

5.3.1 How to Configure

Slocum gliders are factory configured with production code. Production code is updated frequently and should be maintained.

1. The pilot must review the updates and features added during each release.

   For more information, contact glidersupport@teledyne.com.

2. Open **README_UPGRADE.pdf** and follow the instructions.

5.3.2 How to Evaluate Data

The Slocum 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 nonintuitive fashion. These data need to be thoroughly investigated before the vehicle’s behavior is understood.

Teledyne Webb Research provides a number of tools for viewing and processing data. Each Dock Server hosts a data server, which transfers raw glider binary data into a database. You can view this database using the data visualizer, where you can plot any available glider sensor against time.

Webb Research also provides tools to convert raw data to ASCII so a user can format and publish data, as desired. For more information, contact glidersupport@teledyne.com.

5.3.3 Relevant Sensors

To view **masterdata**, contact glidersupport@teledyne.com.

5.4 Hardware Interface Board

The flight STM32 processor is mated to a carrier board that connects to a motherboard that, in turn, 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
- Burn wire for the ejection weight

5.4.1 How to Configure

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

5.4.2 How to Test

Contact glidersupport@teledyne.com for the following:

- Refer to the glider wiring diagram. To receive the latest version of the wiring diagram, contact Glider Support.
- 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, contact Glider Support.

5.5 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 document number 4095-CC, *Glider Compass Calibration*, for instructions.

5.5.1 How to Configure

A calibration utility and its instructions can be attained by contacting glidersupport@teledyne.com.

5.5.2 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, perform the calibration procedure as far away as possible from any of these sources of interference.

1. Type:

```
report ++ m_heading m_pitch m_roll
```

2. Rotate the vehicle.
3. Independently verify output.

5.5.3 How to Evaluate Data

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

5.5.4 Relevant Sensors

Table 5-2: Relevant Sensors: Air Pump System

Sensor Name	Description
c_att_time	units (sec) 0 = turns on as fast as possible -1 = off
m_heading	units (radians)
m_pitch	unit (radians or degrees)
m_roll	unit (radians or degrees)

5.6 Iridium Satellite Telemetry

The Iridium 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.

Phone numbers for the Iridium modem can be different for each customer. You must configure the number when purchase the SIM card.

5.6.1 How to Configure

Note

De-PINning 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 de-PINning information, see the *Slocum Glider Maintenance Manual* [P/N M313476-NFC].

The glider's Iridium phone and SIM card are configured at the factory. Contact glidersupport@teledyne.com for assistance de-PINning SIM cards or checking phone configuration.

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 Dock Server must also be configured for network.

5.6.2 How to Test

1. Place the glider outside, with a clear view of the sky.
2. To dial the primary number immediately, type:

```
callback 0 0
```

3. To call the alternate number in one minute, type:

```
callback 1 1
```

4. To call back the primary number in 30 minutes, type:

```
callback 30 0
```

Thirty minutes is the maximum allowable callback time.

5.6.3 How to Evaluate Data

Connection time can vary depending on cloud cover and availability of satellites. Use this utility that provides a map to confirm satellite coverage:

`https://oz9aec.dk/gpredict/`

5.6.4 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**. For more information, contact glidersupport@teledyne.com

5.7 Radio Frequency (RF) Modem Telemetry

When connected to the flight processor, 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 communications.

For more information, see "9.1.1 FreeWave Configuration" on page 9-1.

5.7.1 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.

1. 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 baud rate.
2. Power on the master.
3. Press the **SETUP** button on the back of the master.
4. 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.

Note
From the FreeWave menu, do not use: **`call all(A)`**

5. From the FreeWave menu, assign the call book entry to: **`call (C 09)`**

5.7.2 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.

5.8 Communications

When first powered on, the FreeWave radio modem is set to communicate with the flight processor. You can establish communications with the science processor from `Glidershell` and `GliderDOS`.

1. To enable direct communication with the science processor via a hardware controlled connection in `Glidershell`, type:

```
consci
```

   The science processor asks two questions:

   a. When it asks if you want to use the clothesline, answer: **no**

   b. When it asks if you want to go to shell, answer: **s** (to enter `SciShell>`)

2. To resume communication with the flight processor, go from `SciShell>` to `Glider Shell>`; in other words:

   - Disconnect FreeWave for **10 seconds** (known as "power cycle the FreeWave").

3. To enable direct communication with the science processor via a software-controlled connection known as "the clothesline" in `GliderDOS`, type:

```
consci
```

4. To resume communication with the flight processor, type:

```
quit
```

5.8.1 How to Configure

Contact Glider Support.

5.8.2 How to Test

1. From `Glidershell`, type:

```
consci
```

2. Establish communication with science.

   For STM32 processors, the prompt changes from:

   `Glider Shell>` to `SciShell>`

5.9 Pressure Transducer

Slocum 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
- Isolated from the aluminum aft cap by a PEEK fitting

5.9.1 How to Configure

Factory calibrated. Surface pressure valves can fluctuate, but you can use the **zero_ocean_pressure** command to reset the zero pressure voltage.

It is recommended that you examine the through port and clean it between deployments to remove any contamination or blockage.

5.9.2 How to Test

In the field, CTD pressure can be compared against flight pressure transducer values.

In the lab, *m_depth* can be reported and will fluctuate.

5.9.3 How to Evaluate Data

Plot *m_depth* against *sci_water_pressure*.

5.9.4 Relevant Sensors

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

5.10 Leak Detect Sensor Boards

Each glider is equipped with three leak detect sensor boards:

- 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.
- The aft leak detect sensor is located on the bottom of the aft cap.

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.

The Slocum glider payload bay sensor is wired in parallel to the forward leak detect circuit:

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

5.10.1 How to Test Leak Detect Sensor Boards

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.

5.10.2 How to Evaluate Data: Leak Detect Sensor Boards

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.

5.10.3 Relevant Sensors: Leak Detect Sensor Boards

Sensor Name	Description
f_leakdetect_threshold (volts)	Any value of *m_leakdetect_voltage* below this threshold is considered a leak.
m_digifin_leakdetect_reading (nodim)	Any value below the 1019 nodims threshold indicates a leak in the fin.
m_leakdetect_voltage (volts)	Voltage that was reported by the aft leak detect sensor. The lower the voltage, the worse the leak.
m_leakdetect_voltage_forward (volts)	Voltage that was reported by the forward leak detect sensor. The lower the voltage, the worse the leak.
m_leakdetect_voltage_science (volts)	Voltage that was reported by the payload bay detect sensor. The lower the voltage, the worse the leak.
m_leakdetect_voltage_stack_on (volts)	Voltage that was reported by the stack on bay leak detect sensor. The lower the voltage, the worse the leak.

5.11 Air Bladder

The glider's air 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.

5.11.1 How to Test

See "5.2.2 How to Test" on page 5-2.

Inpsect air bladder for damage before and after each deployment.

5.11.2 How to Evaluate Data

See "5.2.3 How to Evaluate Data" on page 5-2.

5.11.3 Relevant Sensors

See "5.2.4 Relevant Sensors" on page 5-2.

5.12 Burn Wire and Ejection 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 releases the ~500g stainless steel spring-loaded ejection 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 ejection weight is:

- Positioned beneath the tail fin assembly
- 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 has 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 wet but not submerged and the ejection weight is activated.

See the *Slocum Glider Maintenance Manual* [P/N M313476-NFC] for more details.

5.12.1 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. Type:

```
put c_weight_drop 0
put m_weight_drop 0
exit shell
```

7. Verify that the voltage is **0**.
8. Reconnect the drop weight supply lead.

5.12.2 Relevant Sensors

Figure 5-1: Relevant Sensors: Burn Wire and Ejection Weight

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

5.13 Power Port

Power to the glider is supplied by a SubConn connector located externally via the glider aft tail cone.

When the RED Stop plug is inserted in that connector or the connector has nothing attached, no power is being applied to the vehicle. This design allows power to be removed from the system without special tooling.

CAUTION
Do not exceed 16 volts.

By powering the glider via the umbilical, there is no need for an internal switch that could generate a spark.

To power the glider ON, insert one of the following:

- The provided external power cable, –or–
- The GREEN Go plug

The Slocum glider should always be powered down properly by using the **exit** command. If the Slocum glider battery management board detects an improper shutdown, it emits a beep.

5.13.1 How to Test

The glider is activated by inserting either wall power or the GREEN plug.

CAUTION
The GREEN Go plug or wall power must never be removed without a software command—*except in an emergency situation*. Doing so results in corrupted files.

Removing the GREEN plug powers off the glider. Personnel must do this ***only*** after issuing the **exit** command and verifying the glider accepted the command.

When improperly shut down, the emergency circuit ultimately activates the burn wire and emergency circuit.

5.14 10-Watt Thruster (Optional)

An optional accessory for the Slocum glider is a high efficiency thruster with larger blades that sweep back to lower drag when not in use.

However, while additional speed or operational capability is provided by employing this thruster, it can increase a significant penalty in consumed energy.

Watch energy budgets carefully when using the thruster.

5.14.1 Operational Modes

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

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

After a surfacing, the thruster is turned on for a short burst right before diving to remove buildup. This is turned on by default and can be disabled.

For complete thruster documentation and recommended thruster mission (`thrstock.mi`), contact glidersupport@teledyne.com.

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

To run a thruster in air:

1. Ensure the propeller blades are clear of the cart.
2. Type:

```
Report ++ m_thruster_current m_thruster_power
Put c_thruster_on 30
```

   Let this run for less than a minute or so.

3. Confirm the blades are spinning clockwise when viewed from the aft end.
4. Type:

```
Put c_thruster_on 0
```

5.15 Tail Fin

The tail fin, shown in Figure 5-1, is field serviceable equipment designed for robust, trouble free operation, strength, and resilience.

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

- 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.

Figure 5-2: Slocum Glider Fin

The tail fin is composed of:

- Titanium boom
- Motor housing
- Antenna feed-through
- Plate bolted to a peek mast using a barrel nut fastener system

Field serviceable item replacements are quick and simple. 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.

5.15.1 Tail Fin Subsystems

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

- Communications
- Steering
- Strobe illumination
- Leak detect

Test all these subsystems as explained in the 4095-FCP Functional Checkout Procedure. They are briefly described in the sections that follow.

5.15.1.1 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 detect by the shore side Freewave modem (indicated by a green CD light).

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/Dock Server. 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**.

5.15.1.2 Steering Subsystem

To verify that the steering subsystem is performing properly:

5. Boot the vehicle into Lab Mode.
6. Put the glider into wiggle (**`wiggle on`**).
7. Report commanded and measured fin position to the screen (**`report ++ m_fin c_fin`**).
8. Verify that `c_fin` drives `m_fin`, and that no oddities or warnings are displayed from the fin.
9. Verify when:
    - `m_fin` is *greater* than **0**, the rudder is deflected toward the starboard side of the vehicle.
    - `m_fin` is *less* than **0**, the rudder is deflected toward the port side of the vehicle.

5.15.1.3 Strobe Illumination Subsystem

To test strobe illumination, perform the following steps:

1. Boot the glider in Lab Mode.
2. Activate the strobe by typing:

```
strobe on
```

3. Verify the strobe periodically flashes every 3–5 seconds.
4. Deactivate the strobe by typing:

```
strobe off
```

5.15.1.4 Fin Leak Detect Subsystem

The leak detect reading measured in the fin can be read by polling the value of *m_digifin_leakdetect_reading* from `GliderDOS` or Lab Mode. 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.

5.15.2 How to Configure

Factory configured.

5.15.3 How to Test

From **`lab_mode`**, `wiggle on/wiggle off`. Range ±~25 degrees (or radians on terminal).

5.15.4 How to Evaluate Data

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

5.15.5 Relevant Sensors

Table 5-3: Relevant Sensors: Tail Fin

Sensor Name	Description	Units
m_fin	Fin measurement	Units are given in: • radians in terminal • degrees in data vis[a]
c_fin	Fin control	radians
m_heading	Heading measurement	radians
c_heading	Heading control	radians
m_roll	Roll measurement	radians

a. These variables can be seen in degrees or radians in the data visualizer that is built into SFMC.

6 Batteries

The Slocum 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
- Rechargeable 15 V batteries

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

6.1 Battery Management System (BMS)

The BMS board provides several 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 provides 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 accomplishes this.

Note

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

  In the event that the emergency battery is left latched when the main battery is disconnected, a buzzer sounds a warning ***pattern*** to indicate 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.

- Ability to measure the main battery and emergency battery voltages separately.

  This allows the user to be better estimate the emergency battery's health.

- An auxiliary high power switch that allows custom user applications.

The BMS parameters are as follows:

```
# Slocum 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
```

6.2 Emergency Battery

The Battery Management System (BMS) gives the Slocum glider the ability to individually draw power from each pack. This feature allows 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
- Gives 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 Slocum 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 Slocum glider or in the payload bay, depending on ballasting needs specific to different glider configurations.

6.3 Rechargeable Battery Packs

There are two rechargeable battery packs, the pitch pack shown in Figure 6-1 and the aft pack shown in Figure 6-2. Each set provides 10.0–16.8 V nominal, a capacity of 3 kWh total for two packs and a charge time of 12 hours.

Figure 6-1: Pitch Rechargeable Battery Pack, Forward Side Up

Figure 6-2: Aft Rechargeable Battery Pack, Aft Side to Left

6.4 Battery Charger

The Teledyne Webb supplied battery charger is shown in Figure 6-3:

Figure 6-3: 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 wire comms channel 2
Gray #3:	2nd Pitch Pack comms channel 3

6.5 Pitch Battery Ports

The pitch battery port is shown in Figure 6-4:

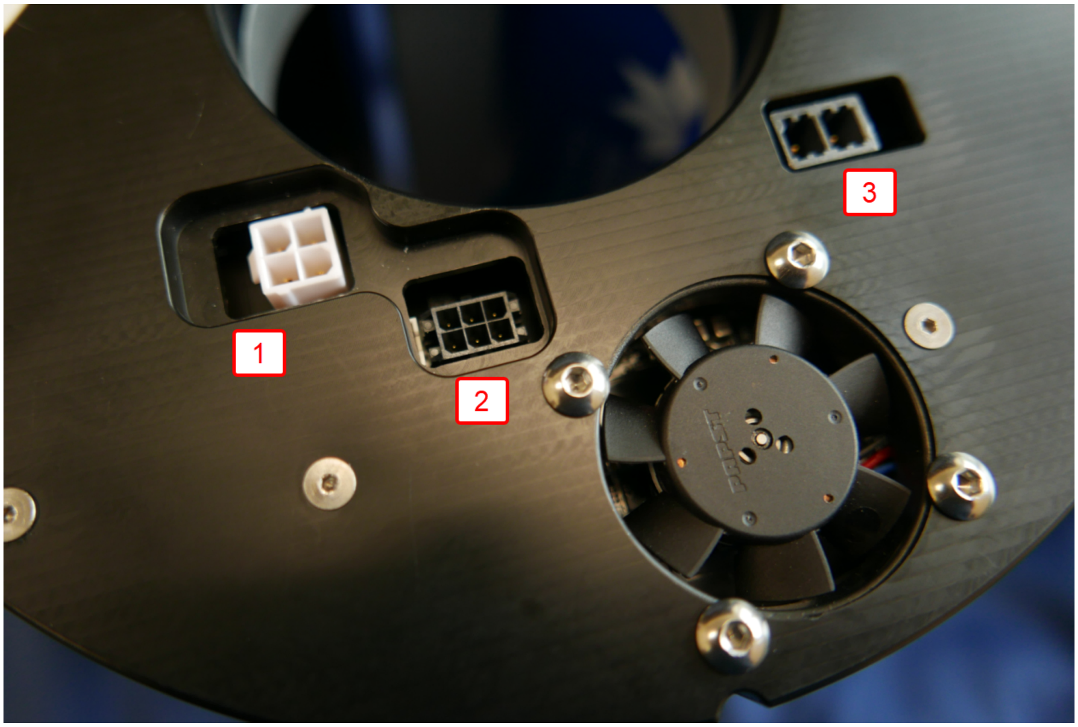

Figure 6-4: Pitch Battery Ports

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

6.6 Battery Charge Indicator

Battery charge indicator bars are available to view on the pitch pack only as shown in Figure 6-5:

Figure 6-5: Battery Charge Indicator Bars

6.7 Battery Charging Instructions

When charging the batteries, use only the Teledyne Webb Research supplied battery charger.

Note
The battery charger is not GFCI protected. If you use the battery charger near water, ensure one of the following: • You connect the charger to a GFCI outlet, –or– • You use a GFCI cable.

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:
   a. Connect the LPIL-5 cable to the glider end cap.
   b. Proceed to Step 8.
4. If you are charging the batteries outside of the glider, you can charge the aft, pitch, and extended pitch battery packs 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.
   a. Plug the 2-pin charge cable from one pair into the pitch battery pack charging port (see Figure 6-4 on page 6-4).
   b. If comms are desired:
      i. If the loopback cable is installed, remove it during charging.
      ii. Plug the 6-pin comms cable into the pitch battery pack comms port.
7. If charging the extended pitch battery pack, plug the:
   a. 2-pin charge cable from the other cable pair into the extended pitch battery pack charging port
   b. 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.

When the battery packs are charging:

- The charging cables may feel warm to touch.
- The battery charger displays the charge current that varies as communications are established with the battery packs.
- An initial 1 amp draw should soon increase to 12 amps. Within a minute or two, the current should stabilize. The current is slightly lower on a nearly discharged battery than a full one.
- 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.

6.8 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 its value 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.
- Channel **3** is the extended pitch pack.
  Each pitch pack contains 10 cells.

While the battery packs are charging, 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

6.9 Slocum Primary Batteries

A change has been made to the lithium primary batteries on the Slocum 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.

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

The 4S configuration 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.

Note

Table 6-1 on page 6-7 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 temperature, age, and storage, as well as real world flight data.

Please see battery manufacturers derating instructions for specific applications.

Putting four cells in series on the 4S batteries results a higher voltage; however, it 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 table below shows a comparison between the different types of packs available and uses Watt Hours to normalize the total energy:

Table 6-1: Comparison between types of packs

	Total Non-derated Amphrs (amphr)	Total non-derated Watt Hours (WH)	Recommended f_coulomb_ battery_capacity	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

6.10 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.

The Main board has two main power supplies, one for alkaline batteries and one for lithium batteries as shown below:

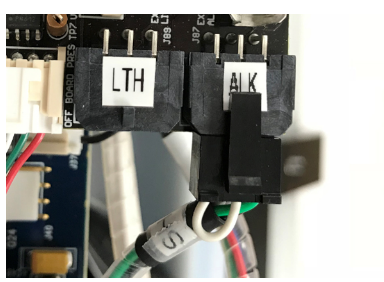

Figure 6-6: Connector at Starboard Aft End of the Main Board

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.

6.11 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:

1. Ensure the following sensor is set:

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

2. Ensure missions are set to abort at:

```
behavior:  abend

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

6.12 Batteries/Coulomb Counter

Alkaline battery packs, which are nominally at 15 volts, consist of 10 diode-protected Duracell C-cells in series.

You can adjust the number of packs, 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.

6.12.1 How to Configure

Factory configured. When you install new, unused batteries or fully recharged rechargeable batteries, enter the following commands at a `GliderDOS` prompt.

1. These reset the coulomb counter to **0**:

```
Put m_coulomb_amphr_total 0
Exit reset
```

2. From the `GliderDOS` prompt, type:

```
get m_coulomb_amphr_total
```

3. Confirm the value is close to zero and increasing when powered by battery.

   This value should not increase when powered by a power supply.

6.12.2 How to Test

WARNING

If the glider contains alkaline batteries, there is a small but finite possibility that alkaline battery cells will release a combustible gas mixture—especially if the batteries are exposed to fresh or sea water and/or shorted.

A gas release is generally not evident when batteries are exposed to the atmosphere, as the gases are dispersed and diluted to a safe level.

However, *when the batteries are confined in a sealed instrument, the gases can accumulate and an explosion is possible*.

Teledyne Webb Research adds a catalyst inside of the glider to recombine hydrogen and oxygen into water. The glider has been designed to relieve excessive internal pressure buildup by separating the hull sections under internal pressure.

Teledyne Webb Research knows of no way to completely eliminate this hazard. *Now that you have been warned, you 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***.

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

6.12.3 How to Evaluate Data

See Appendix 12, “Determining Battery Longevity.”

6.12.4 Relevant Sensors

- *m_battery* (volts)
- *m_battery_inst* (volts)
- *m_coulomb_amphr_total* (amp-hrs)
- *m_coulomb_current* (amp-hrs)
- *m_lithium_battery_relative_charge* (percentage)

6.13 Shipping Requirements

When shipping the battery packs:

- Only a certified carrier should be used.
- The batteries must be discharged to below 30%.

Class 9 HAZ, in equipment UN3481 and outside of equipment UN3480, also apply.

Part 3: Operations

7 Pre-mission Testing

Teledyne Webb Research provides a document called the *Functional Checkout Procedure* (FCP). Follow it to perform thorough testing.

Note

Some, not all, tests must be performed outside because the gliders need a clear view of the sky to get a GPS fix and to make Iridium communications.

We recommend you perform this checkout procedure on the glider after:

- Final seal in the lab
- Storage
- Shipping

The following minimal procedures should be followed to qualify a glider before deployment, or with new or modified software.

7.1 On Shore (Beach, Boat, or Bench)

1. Power on the glider and enter GliderDOS by pressing **[Ctrl]-[C]** when instructed.
2. To stop the air pump from running, type:

```
put c_air_pump 0
```

3. To hang up the Iridium phone, type:

```
callback 30
```

4. To test the GPS, type:

```
put c_gps_on 3
```

   This command places GPS communications into "verbose" mode. When the GPS signal is confirmed, the data stream output display changes 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.

5. Wait until you are satisfied with the GPS location.
6. To return to the non-verbose mode, type:

```
putc_gps_on 1
```

WARNING

Never deploy a glider in lab_mode.

7. Test the motors by typing two commands:

```
lab_mode on
wiggle on
```

8. For shallows, type one of the following:

```
m_de_oil_vol      -or-
m_ballast_pumped
```

9. To confirm the pump and pitch battery are moving as expected: **m_battpos**
10. Run for 3–5 minutes to check for any device errors or other abnormalities.
11. To stop wiggling, type:

```
wiggle off
```

12. If there is a deep pump, you can run the motor longer and report the pump's location to the screen.

Note
Ensure the glider is set to boot app before deploying it in the water.

13. If you find no errors, return to the GliderDOS prompt by typing:

```
lab_mode off
```

    Always ensure the glider is not in Glidershell or Lab Mode before deploying it in the water.

14. Type:

```
run status.mi
```

15. Confirm all sensors are being read.

    The mission ends with this confirmation message:

```
mission completed normally
```

16. Load the glider into the boat and head out toward the first waypoint or deployment location.

7.2 In Water

1. If possible, attach a line with flotation to the glider before putting the glider in the water.

   If you are performing this mission on a line with flotation, ensure the line length is sufficient or modify yo depth.

   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 this again:

```
run status.mi
```

3. If ballasting has already been confirmed, proceed to Step 4 or Step 5.

   If unsure of ballasting, most operators 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
```

4. When the glider returns to the surface, examine the data to evaluate whether it is alright to proceed.
5. If you have not removed the line from the glider, do it now.
6. [*optional*] From the `GliderDOS` prompt, type:

```
exit reset
```

This command forces all the sensor values to reinitialize. It is *advisable* to do an "exit reset" after removing the buoy but not it is not necessary.

7. [*optional*] When the glider reboots, press **[Ctrl]**-**[C]** to return to a `GliderDOS` prompt.
8. [*optional*] To zero any built-up water currents that are remembered long term, type:

```
load mission waterclr.mi
```

9. To begin the stock mission or run the desired mission, type:

```
run stock.mi
```

8 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. You can deploy the glider using:

- The pick point
- On smaller vessels, the glider cart on the gunwale

Perform the following steps:

1. Allow the glider to slip into the water. During handling, hold the glider by the fin.
2. While moving the glider aboard during recovery, you can use the pick point or cart.
3. While in the water, you can use a hook or lasso on a pole to manipulate the glider.

The nose cone is attached to a rope and, once deployed, will sit at surface for a crane to hook. You can find photographs of deployment and recovery by cart in "8.1 Deploying the Glider" below and "8.3 Recovering the Glider" on page 8-3.

Gliders outfitted with a recovery system can be commanded to release the nose flotation.

If a vessel has an overhead crane, use that to lift the vehicle from the water line. To activate, type:

```
put c_recovery_on 1
```

8.1 Deploying the Glider

CAUTION

Be very careful with non-Digifin gliders during deployment and recovery. If handled too aggressively, you can knock the fin out of calibration or damaged it.

Note

Use the top of the tail fin to manipulate the glider in the water. Handle non-Digifin gliders by the tail boom or pick point only.

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.

Figure 8-1: Glider with the Buoy and Rope Ready for deployment

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

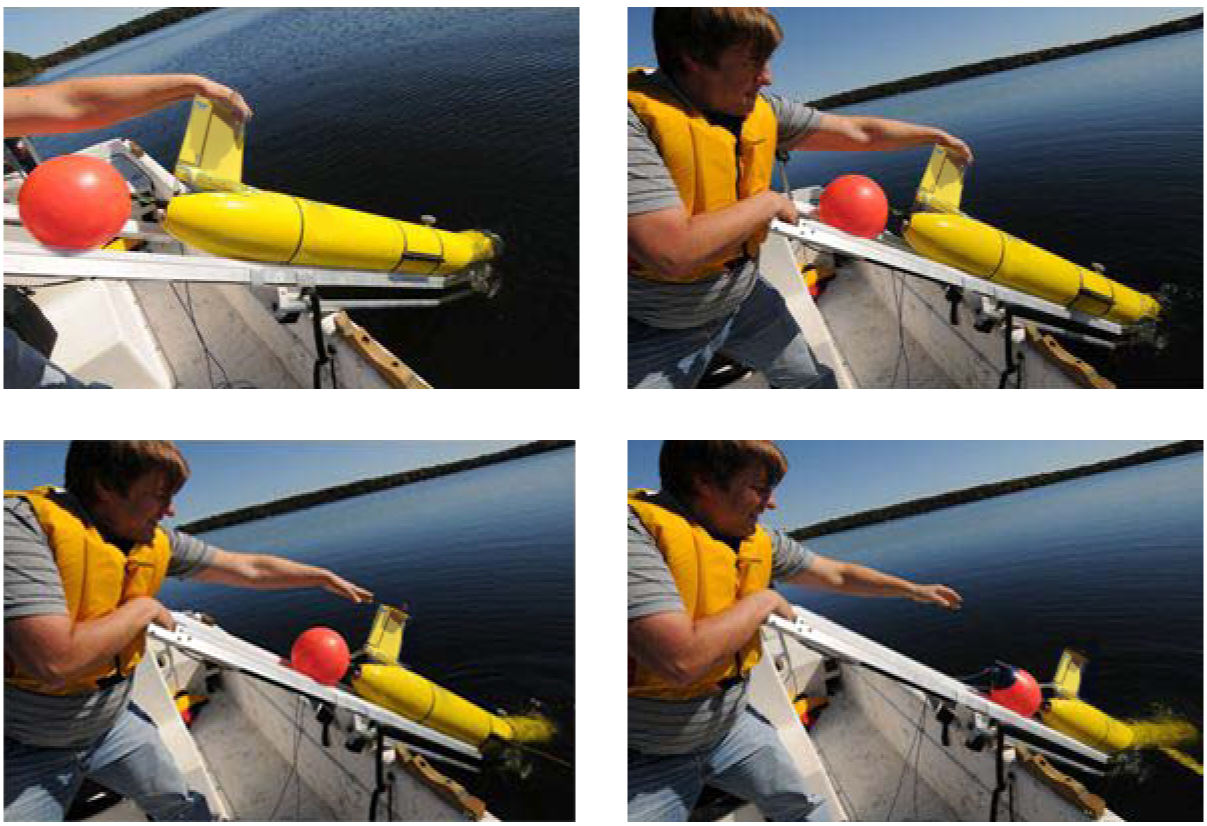

Figure 8-2: Launching the glider

8.2 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.

Figure 8-3: Pick point from supplies

Figure 8-4: Using pick point

8.3 Recovering the Glider

CAUTION

Be very careful with non-Digifin gliders during deployment and recovery. If handled too aggressively, you can knock the fin out of calibration or damaged it.

Note

Use the top of the tail fin to manipulate the glider in the water. Handle non-Digifin gliders by the tail boom or pick point only.

1. A boat hook can be used to recover a glider.
2. Lower the cart with nose ring into water.
3. Manipulate the glider by the tail boom into position on the cart.
4. Lift and tilt the glider onto the ship's deck.

Figure 8-5: Recovering the glider

8.4 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 Dock Server.
- Determine the best-known position with the available data.

9 Glider Communications

The glider is intended to be used in conjunction with Dock Server. See the *Slocum Fleet Mission Control User Manual* [P/N M313834-NFC] for information on communications to the glider while using the Dock Server.

9.1 Communicating with the RF Modem (FreeWave)

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 "9.1.1 FreeWave Configuration" on page 9-1 for more information.

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

9.1.1 FreeWave Configuration

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

9.1.1.1 About FreeWave Transceivers

FreeWave transceivers operate in virtually any environment where RS232 data communications occur. The transceivers functions 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, the FreeWave transceivers can be connected directly.
- If FreeWave transceivers are to be used to replace a straight-through RS232 cable, a null modem cable must be placed between the transceiver and the DCE instrument to which it is connected.

9.1.2 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:
    a. Connect to COMx (depending on which COM port your cable is connected to)
    b. Set the following to:
        - data rate: **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]** (zero key) 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 `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 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.
8. Press **[Esc]** to return to the *Main* menu.
9. Press **[Esc]** to exit set-up.

9.1.3 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, as shown in Figure 9-1:

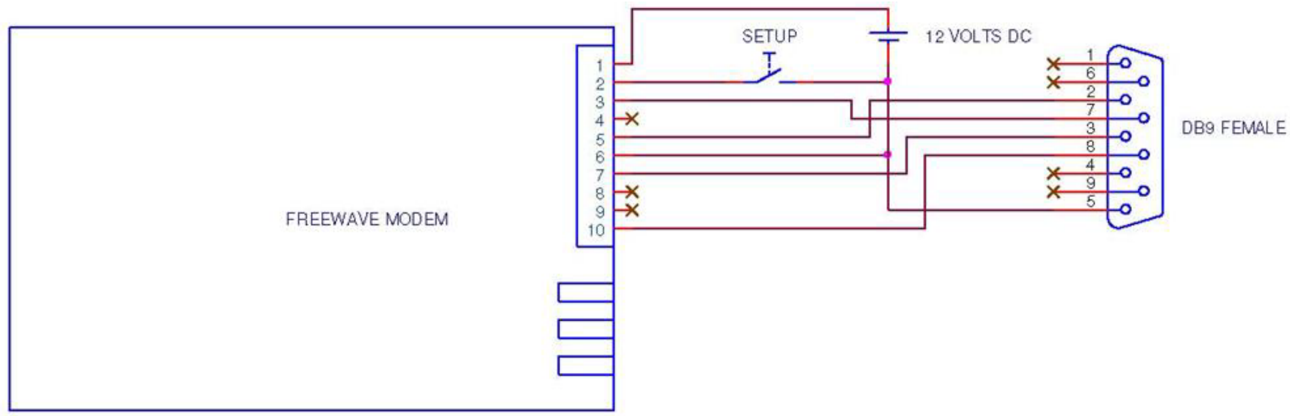

Figure 9-1: Connecting through a serial cable.

2. Open a HyperTerminal session. Use the following settings to connect with HyperTerminal:
   a. Connect to COMx (depending on which COM port your cable is connected to)
   b. Set the following to:
      - data rate: **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]** (zero key) 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.
8. Press **[Esc]** to return to the *Main* menu.
9. Press **[Esc]** to exit set-up.

9.1.4 Choosing FreeWave Transceiver Location

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.

9.2 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 Glidershell outside of a lab environment.

- To test Iridium while in Glidershell, change the Dock Server to serial perspective and type talk iridium (refer to “2.3 GliderShell” on page 2-2).
- To leave talk, press **[Ctrl]**-**[C]**. Pause for a moment; then press **[Ctrl]**-**[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, **[Ctrl]-[C]** can be used to hang up the phone.

9.2.1 Iridium Service and the SIM Card

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

9.2.1.1 Providers

Teledyne Webb Research uses Joubeh.

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/

9.2.1.2 Billing

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

De-PINning 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 de-PINning, see the *Slocum G3S Maintenance Manual* [P/N M315357-NFC].

Billing for the service is monthly.

The SIM card is required during the manufacturing process and must be activated 30 days prior to shipment.

Teledyne Webb Research needs the SIM card and the unlocking PIN to unlock the card and permanently deactivate the PIN code.

9.3 Communicating with RUDICS

For more information, contact glidersupport@teledyne.com.

10 Science Data Logging (SDL)

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

10.1 Log File Types

Science has a parallel log file type for each log file type on the glider, as shown in the table below. Each pair is formatted the same (for example, EBD files are formatted the same as DBD files).

Table 10-1: Log File Types on the Glider and Science Processors

Log File Type on the Glider Processor	Equivalent Log File Type on the Science Processor		Compressed Log File Type on the Glider Processor	Compressed Equivalent Log File Type on the Science Processor
.dbd	.ebd		.dcd	.ecd
.mbd	.nbd		.mcd	.ncd
.sbd	.tbd		.scd	.tcd
.mlg	.nlg		.mcg	.ncg

- The *m_present_time* should be present in **mbdlist.dat** and **sbdlist.dat** on the glider processor.
- The *sci_m_present_time* should be present in **nbdlist.dat** and **tbdlist.dat** on the science processor.

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

```
\logs
\sentlogs
\state\cache (contains the header cache files)
```

The following `SciDOS` commands support SDL 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`**

10.2 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 10-2: 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

10.3 Transparency

When the logging on command is issued to the glider processor, a message is sent 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 share the same file name root (i.e., **12345678.mlg** and **12345678.nlg** represent the glider, respectively).

Log file 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 Dock Server is transparent. Issuing the send command from GliderDOS or the **s** command in a mission file 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, the following command is typed in GliderDOS:

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

The science processor sends 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 the example above, 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. For more information, contact glidersupport@teledyne.com.

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).

10.4 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.

To send all science sensors, a new sensor must be used (see *c_science_send_all* below).

Table 10-3: Sensors Added to the Glider

Sensor	Processor	Default Setting	Description
`c_science_send_all (bool)`	Glider	0	Tells the science processor whether to send all variables or just a few
`m_mission_start_time (timestamp)`	Glider	N/A	Propagates to science processor
`m_science_readiness_for_consci (enum)`	Glider	N/A	Tells if ready or, if not, why not
`sci_m_disk_free (Mbytes)`	Science	N/A	How much space is currently free on science
`sci_m_disk_usage (Mbytes)`	Science	N/A	How much space is currently used on science
`sci_m_present_secs_into_mission (sec)`	Science	N/A	Analog of *m_present_secs_into_mission* on science
`sci_m_free_heap (bytes)`	Science	N/A	Analog of *m_free_heap* on science
`sci_m_min_free_heap (bytes)`	Science	N/A	Analog of *m_min_free_heap* on science
`sci_m_min_spare_heap (bytes)`	Science	N/A	Analog of *m_min_spare_heap* on science
`sci_m_spare_heap (bytes)`	Science	N/A	Analog of *m_spare_heapon* science
`sci_x_disk_files_removed (nodim)`	Science	N/A	Count of files removed by last science processor's prune command
`sci_x_sent_data_files(nodim)`	Science	N/A	Count of files successfully transmitted by last science processor's sent command
`u_sci_cycle_time (secs)`	Glider	1.0	Tells the science processor how fast to run
`u_sci_dbd_sensor_list_xmit_control (enum)`	Glider	0	Always transmit the header to tell the science processor what to do
`x_science_logging_state (enum)`	Glider	N/A	Indicates the science processor's logging state
`u_science_send_time_limit_adjustment_factor (nodim)`	Glider	0.5	Puts an absolute limit on how long the glider can spend sending science files during a single `consci` session

10.4.1 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:

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.

- *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 sending 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.

11 Editing a *Proglets.dat* File

To communicate with the processor:

- From `GliderShell`, type: **`consci`**
- from `GliderDOS`, type: **`consci`**
- From a mission, type: **[Ctrl]-[T]**

The processor folder structure is shown in Figure 11-1:

Figure 11-1: STM32 processor folder structure

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 the **config** directory called **proglets.dat** controls the wiring and configuration of the science computer.

There is a proglet 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 Dock Server 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 Dock Server.
8. Confirm the glider is still in `SciDOS`. (There is a 1200-second default timeout value.)
9. If necessary, type:

```
cd config
```

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. To exit the science computer, type:

```
quit
```

14. Do one of the following:
    - If in a mission, type:

      ```
      !use - science_super
      ```

      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)
#-------------------------------------------------------------------------
```

12 Determining Battery Longevity

Battery longevity is estimated by considering several factors:

- Type of battery
- Style of pump
- Science sensor types and sampling strategy
- Surface time required for real time data transmission
- Starting battery voltage

12.1 Using Alkaline Batteries

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
```

12.2 Using Lithium Batteries

Monitoring the following sensors in the surface dialog or included in the files is recommended when using lithium batteries:

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.

```
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
```

Table 12-1 is an actual alkaline battery voltage data plot from a 27-day glider deployment:

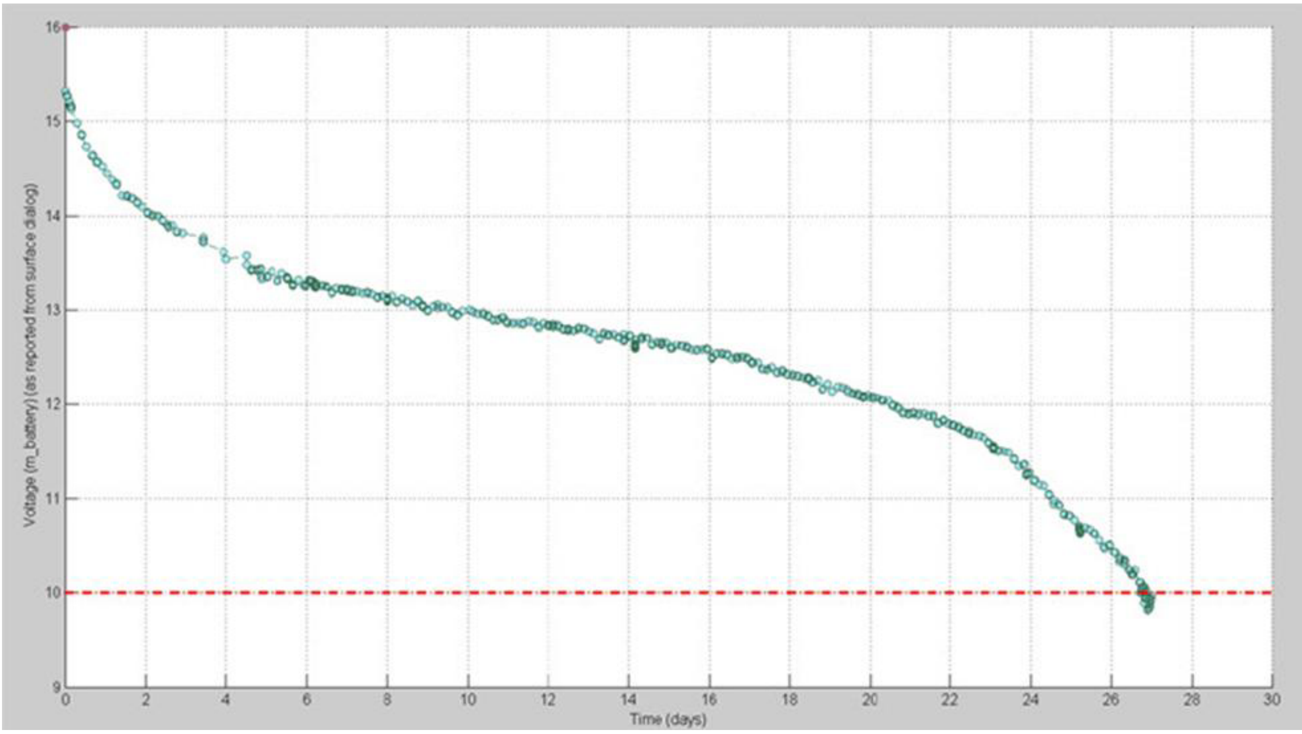

Figure 12-1: Alkaline battery voltage plot

Monitoring available recovery scenarios is typically initiated as the voltage approaches 12 volts. In the deployment above, recovery became critical as voltage dropped below 11 volts on day 25. While using factory settings, the glider began to abort all missions for under voltage beginning at day 27.

If recovery is not possible as the voltage drops below 11.5 volts, contact glidersupport@teledyne.com for battery preservation recommendations until recovery is possible, including any/all of the following:

- Termination of science sampling
- Limiting dive depth
- Limiting communication
- Drifting at a shallow depth

13 Missions

13.1 Mission Files

Note

Refer to the *Slocum Fleet Mission Control User Manual* [P/N M313834-NFC] for more information regarding mission planning.

Mission (**.mi**) files and **.ma** files can be loaded to or drawn off the glider at any time by using SFMC software and **zs** and **dockzr** commands.

13.1.1 .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 Dock Server is stored, and from where the glider draws its name.

13.1.2 .ma Files

The mission acquisition (***.ma**) files called by mission (***.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, if the behavior used is *goto_list*, and it calls a file reference number of 07, the ***.ma** file should:

- Be named **goto_l07.ma** and
- Contain just a list of latitudes and longitudes.

13.2 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 Glider Terminal.
3. Power the glider by inserting the go plug (green).
4. Ensure the glider is not in Simulation Mode. The following files must be deleted:
    - **simul.sim** in the glider processor flash card config directory
    - **appcmd.dat** in the processor flash card config directory

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 the number of times to run that mission.

13.3 Using Autoballast

- Autonomously decrease the amount of drive used on dives and climbs. Conserve energy and equalize dive and climb speeds.
- Avoid unnecessary full extension of pump at the surface. Increase surface stability and conserve energy.

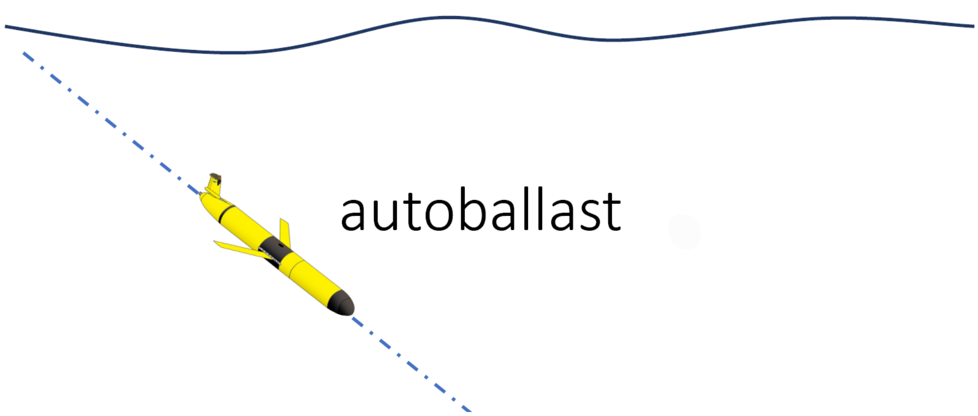

Figure 13-1: Autoballast concept

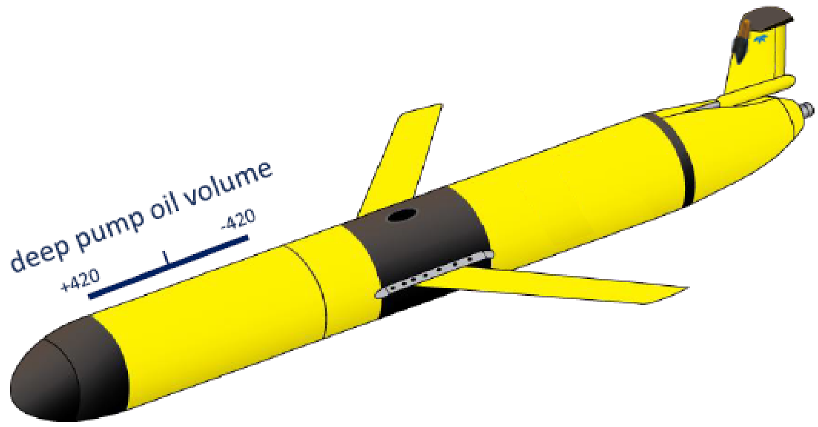

Figure 13-2: Deep pump oil volume

13.3.1 Important Variables to Monitor

Typically, these variables are monitored in the **Surface** dialog:

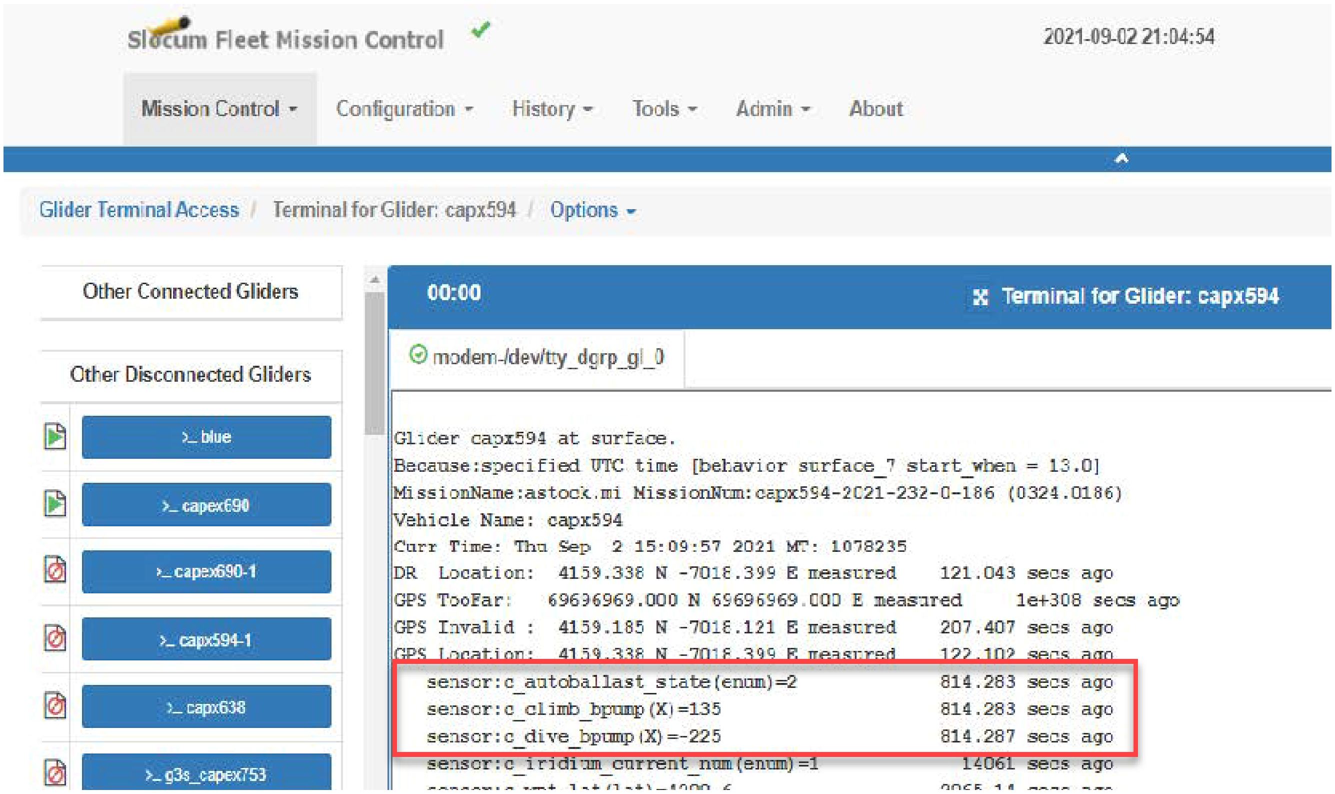

Figure 13-3: Important variables to monitor

- **`c_autoballast_state`**
  This provides the current status of the autoballast process. The user may also change its value to force autoballast to change state. Look in the **masterdata** file for more information.
  - 0 = uninitialized
  - 1 = initialized, still converging
  - 2 = converged successfully
- **`c_dive_bpump`**
- **`c_climb_bpump`**
  These determine the amount of drive being used on dive and climb, respectively.

13.3.2 Setting Up Autoballast

- Must be specified in **yo** behavior or **drift** behavior: **`yoXX.ma`** file or **`driftXX.ma`** file
- Must be specified in surface behaviors: **`surfaceXX.ma`** files
- In the Webb Research software release, the default **`astock.mi`** file is a standard autoballast mission.

The default **astock.mi** file calls the following files:

- **surfac21.ma**
- **surfac22.ma**
- **surfac23.ma**
- **surfac24.ma**
- **surfac25.ma**
- **surfac26.ma**
- **yo14.ma**
- **goto_l10.ma**

13.3.3 Yo Behavior

The figures below illustrate autoballast behavior arguments from a **yo** behavior file; for example, **yo14.ma** (or given file name).

13.3.3.1 Turning Autoballast ON

```
behavior_name=yo

<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)       30
   b_arg: d_target_altitude(m)    5

   b_arg: d_use_bpump(enum) 0             # 0  Autoballast/Speed control.
   b_arg: d_bpump_value(X) 300.0  # use_bpump == 0   Total amt of ballast, stored as C_AUTOBALLAST_VOLUME

   b_arg: d_use_pitch(enum)       3       # 1:battpos  2:setonce  3:servo
                                          #   in          rad         rad, <0 dive
   b_arg: d_pitch_value(X)              -0.4538    # -26 deg
   b_arg: d_stop_when_hover_for(sec)     600.0     # increased to accomod                  t
   b_arg: d_stop_when_stalled_for(sec)   660.0     # increased to accomod                  t
   b_arg: d_speed_min(m/s) 0.06           # minimum depth rate for dive

   # arguments for climb_to
   b_arg: c_target_depth(m)       7
   b_arg: c_target_altitude(m)   -1

   b_arg: c_use_bpump(enum)       0       # 0  Autoballast/Speed control.

   b_arg: c_use_pitch(enum)       3       # 1:battpos  2:setonce  3:servo
                                          #   in          rad         rad, >0 climb
   b_arg: c_pitch_value(X)               0.4538    # 26 deg
   b_arg: c_stop_when_hover_for(sec)     600.0     # increased to accomodate slower depth rate for autoballast
   b_arg: c_stop_when_stalled_for(sec)   660.0     # increased to accomodate slower depth rate for autoballast
   b_arg: c_speed_min(m/s) -0.06          # minimum depth rate for climb

   b_arg: end_action(enum) 2              # 0-quit, 2 resume
<end:b_arg>
```

Turns autoballast ON

Figure 13-4: Behavior arguments that turn **autoballast** ON

13.3.3.2 Specifying Total Drive

Behavior argument that specifies how much total drive to use is illustrated in Figure 13-5.

- If using **d_bpump_value** = 300 and the glider is perfectly ballasted, it will use 150cc on the dives and +150cc on the climbs.
- If setting this to something smaller than the minimum **sensor: f_min_ballast (X)**, the mission will abort. The f_min_ballast default is **250**. Review masterdata description before changing.
- There is no corresponding "climb" value for this argument in an autoballast mission. This includes the **surfacXX.ma** file's "climb" arguments.

This is saved as variable **c_autoballast_volume** in the DBD files.

```
behavior_name=yo

<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
                                             # <0 is inf

   # arguments for dive_to
   b_arg: d_target_depth(m)      30
   b_arg: d_target_altitude(m)   5

   b_arg: d_use_bpump(enum) 0            # 0  Autoballast/Speed control.
   b_arg: d_bpump_value(X) 300.0  # use_bpump == 0   Total amt of ballast, stored as C_AUTOBALLAST_VOLUME

   b_arg: d_use_pitch(enum)      3       # 1:battpos  2:setonce  3:servo
                                         #   in          rad         rad, <0 dive
   b_arg: d_pitch_value(X)             -0.4538   # -26 deg
   b_arg: d_stop_when_hover_for(sec)    600.0    # increased to accomodate slower depth rate for autoballast
   b_arg: d_stop_when_stalled_for(sec)  660.0    # increased to accomodate slower depth rate for autoballast
   b_arg: d_speed_min(m/s) 0.06          # minimum depth rate for dive

   # arguments for climb_to
   b_arg: c_target_depth(m)      7
   b_arg: c_target_altitude(m)  -1

   b_arg: c_use_bpump(enum)      0       # 0  Autoballast/Speed control.

   b_arg: c_use_pitch(enum)      3       # 1:battpos  2:setonce  3:servo
                                         #   in          rad         rad, >0 climb
   b_arg: c_pitch_value(X)              0.4538   # 26 deg
   b_arg: c_stop_when_hover_for(sec)    600.0    # increased to accomodate slower depth rate for autoballast
   b_arg: c_stop_when_stalled_for(sec)  660.0    # increased to accomodate slower depth rate for autoballast
   b_arg: c_speed_min(m/s) -0.06         # minimum depth rate for climb

   b_arg: end_action(enum) 2             # 0-quit, 2 resume
<end:b_arg>
```

Specifies how much total drive to use

Figure 13-5: Total drive behavior argument

13.3.3.3 Specifying Speed Control

Speed control is illustrated in Figure 13-6.

- Slowest glider should go before increasing drive used.
- **m_depth_rate_avg_final** is the default variable for determining the vertical speed.

```
behavior_name=yo

<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)      30
   b_arg: d_target_altitude(m)   5

   b_arg: d_use_bpump(enum) 0           # 0  Autoballast/Speed control.
   b_arg: d_bpump_value(X) 300.0  # use_bpump == 0   Total amt of ballast, stored as C_AUTOBALLAST_VOLUME

   b_arg: d_use_pitch(enum)      3       # 1:battpos  2:setonce  3:servo
                                         #   in          rad         rad, <0 dive
   b_arg: d_pitch_value(X)             -0.4538   # -26 deg
   b_arg: d_stop_when_hover_for(sec)    600.0    # increased to accomodate slower depth rate for autoballast
   b_arg: d_stop_when_stalled_for(sec)  660.0    # increased to accomodate slower depth rate for autoballast
   b_arg: d_speed_min(m/s) 0.06         # minimum depth rate for dive

   # arguments for climb_to
   b_arg: c_target_depth(m)      7
   b_arg: c_target_altitude(m)  -1

   b_arg: c_use_bpump(enum)      0       # 0  Autoballast/Speed control

   b_arg: c_use_pitch(enum)      3       # 1:battpos  2:setonce  3:serv
                                         #   in          rad         rad, >0 climb
   b_arg: c_pitch_value(X)              0.4538   # 26 deg
   b_arg: c_stop_when_hover_for(sec)    600.0    # increased to accomodate slower depth rate for autoballast
   b_arg: c_stop_when_stalled_for(sec)  660.0    # increased to accomodate slower depth rate for autoballast
   b_arg: c_speed_min(m/s) -0.06        # minimum depth rate for climb

   b_arg: end_action(enum) 2            # 0-quit, 2 resume
<end:b_arg>
```

Speed control

Figure 13-6: Speed control

13.3.4 Surface Behavior

Figure 13-7 illustrates autoballast behavior arguments from a **surface** behavior file; for example, **surfac21.ma** (or given file name).

Autoballast climb values must be specified in each surface behavior or **surfacXX.ma** file.

```
behavior_name=surface

<start: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 # Surface every 20 min for no comms

   b_arg: end_action(enum)   1    # 0-quit, 1-wait for ^C quit/resume, 2-resume, 3-drift til "end_wpt_dist"
   b_arg: gps_wait_time(s)          300      # how long to wait for gps
   b_arg: keystroke_wait_time(sec)  300      # how long to wait for control-C
   b_arg: when_wpt_dist(m)          10       # how close to waypoint before surface, only if start_when==7
   b_arg: c_use_bpump(enum)         0        # use autoballast on surface climb (requires autoballast yo)
   b_arg: c_use_pitch(enum)         3        # 3:servo
   b_arg: c_pitch_value(X)          0.4538   # 26 deg
   b_arg: printout_cycle_time(sec)  60.0     # How often to print dialog

<end:b_arg>
```

Specifies use of autoballast climb values for surface behaviors

Figure 13-7: Autoballast climb values for surface behaviors

13.3.5 Getting Started

1. Start with a ***full*** drive on dives and climbs.

   `masterdata` defaults:

   `c_dive_bpump = -1000`

   `c_climb_bpump = 1000`

2. For each yo, decrease the amount of drive used.

13.3.5.1 Considerations

- Be careful if flying in shallow water with a smaller **`d_target_altitude`** value. Consider reducing **`c_dive_bpump`** before initial dive to avoid bottom sampling.
- If the glider is reset during the deployment, these values automatically return to the **`masterdata`** defaults.

Amount of drive used is stepped down with every yo as autoballast converges:

Figure 13-8: Stepping down the amount of drive with every yo

Amount of drive used increases if vertical speed falls below **d_speed_min** or **c_speed_min**:

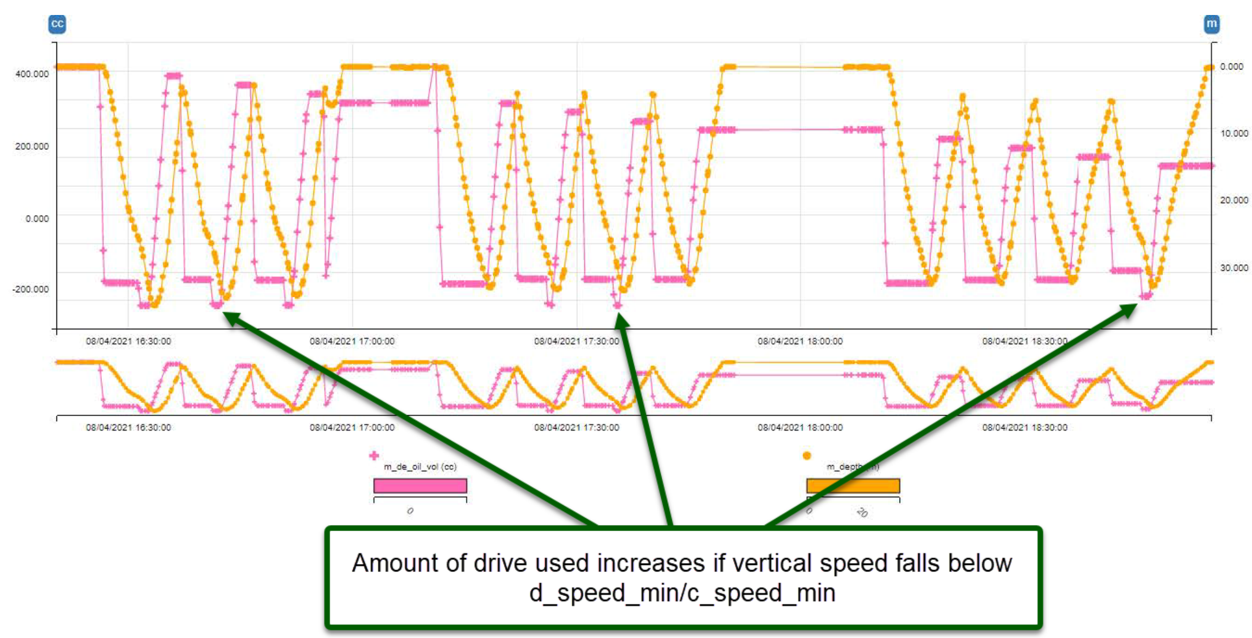

Figure 13-9: Increasing the amount of drive if vertical speed falls below limits

Reduced drive is maintained at surface:

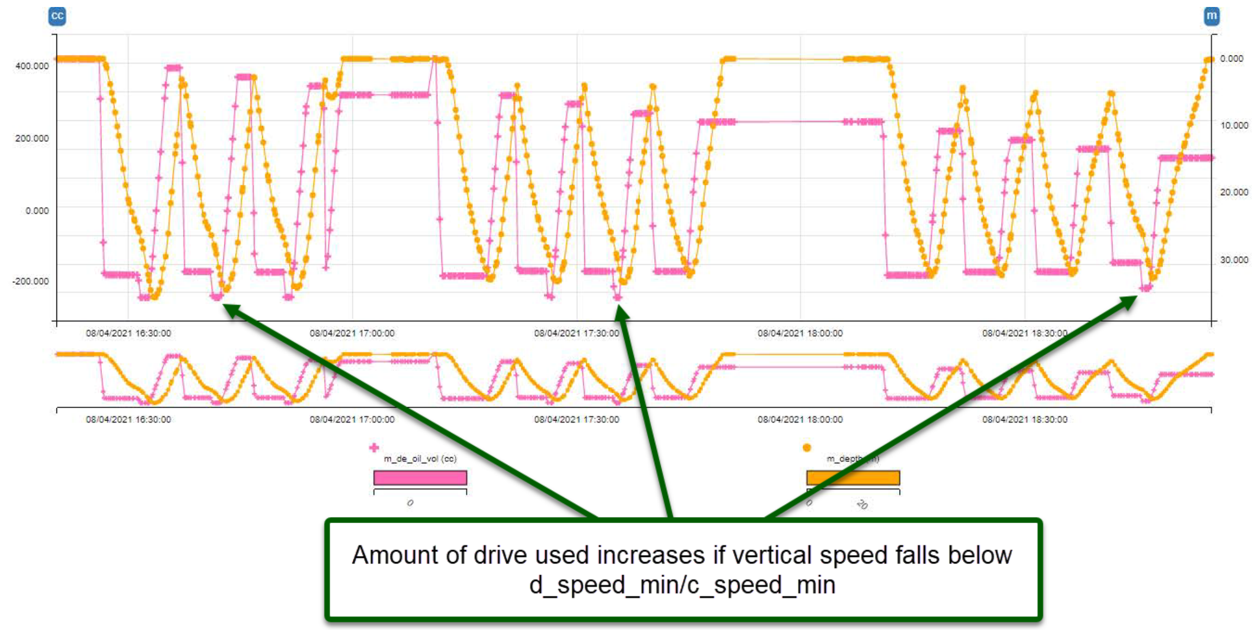

Figure 13-10: Reduced drive maintained at surface

13.3.6 Other Autoballast Uses

While pilots do not have to use autoballast (a setting that commands the oil volume to adjust to match the diving and climbing speed) due to the High Displacement (HD) pumps, TWR highly recommends the use of autoballast for a number of reasons, including:

- Optimizes efficiency resulting in longer battery life
- Reaches ideal speed and pitch angle better

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 by the user before being deployed to Slocum vehicles.

If no buoyancy is specified by the **.mi** or **.ma** file, the new masterdata fault of +/-480 cc (with a deadband of 40) of drive will be used. The glider usually operates with +/-420 cc.

CAUTION

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

Note

Using the full HD drive capability will significantly increase the energy consumed.

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) -420.0 #! min = -1000; max = 1000

b_arg: c_bpump_value(X)  420.0 #! min = -1000.0; max = 1000.0
```

The default TWR mission recommended is `astock.mi`, which utilizes autoballast. For more information, contact glidersupport@teledyne.com.

13.4 Correcting for Waypoint Drift

This chapter provides a generalized solution for determining heading and time needed for reaching a desired position while experiencing a constant drift.

13.4.1 Heading Mode Settings

There are three separate heading modes that can be set using the sensor *u_use_current_correction*.

Heading Mode values are described in Table 13-1.

Using **2** is recommended, especially in strong current situations. The remainder of this chapter explains the reason in detail.

Table 13-1: Heading Mode values

Value	Definition	Description
0	No current correction	The G3S glider does not take current into account when determining its heading. The glider sets its heading towards the waypoint without any further corrections.
1	Weak Current correction	Uses the methodology discussed in the "13.4.3 Current Correction – Method 1" on page 13-11.
2	Strong Current correction	• If a valid heading can be determined for the glider to achieve the commanded waypoint, the glider uses the methodology discussed in the "13.4.4 Current Correction – Method 2" on page 13-13. • If the current is too strong and no valid heading (no positive time solutions) can be determined, the glider aims its heading toward the waypoint and does not take current into account.

13.4.2 Assumptions

1. The initial assumption is the object is at a position within an XY plane and moving with some forward speed.
2. The next assumption is the object must reach a desired location.
3. The last assumption is there are external properties causing a constant drift (such as wind, water currents, or other circumstances).

Therefore, the desired equation results in a heading the object must take in order to arrive at the desired destination under the circumstances. This heading is in relation to the y-axis.

13.4.3 Current Correction – Method 1

For a Slocum G3S Glider in weak currents, the steps for calculating the desired equation are:

1. Determine the desired waypoint.
2. Calculate the distance to that waypoint. The result is the **waypoint vector**.
3. Divide the distance by the horizontal speed of the glider. The quotient provides a **time**.
4. Multiply the time by the current. The product results in a **drift vector**.
5. Add the drift vector to the waypoint vector.
6. Subtract the drift vector from the direction of the waypoint. The result is the **target vector**.
7. The angle of this vector becomes the glider's commanded heading, *c_heading*.

Figure 13-11 is an example with the glider assumed to be at the origin.

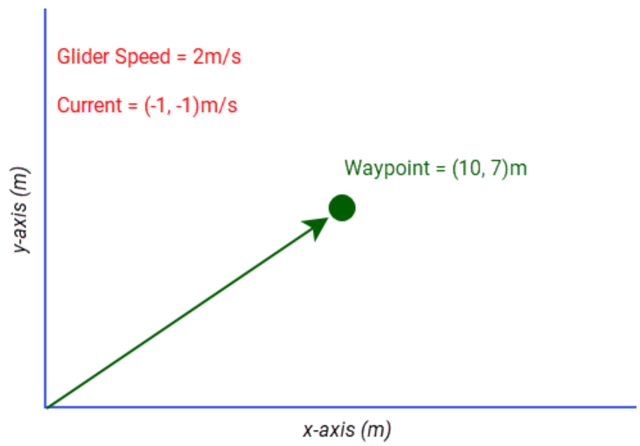

Figure 13-11: Plot 1 — Determining the drift vector, current correction method 1

In this example, the equations for the four variables mentioned above (**waypoint vector**, **time**, **drift vector**, and **target vector**) are shown in Figure 13-12.

$$time = Distance\,To\,Waypoint = \frac{\sqrt{x^2 + y^2}}{Speed} = \frac{\sqrt{10^2 + 7^2}}{2} = 6.1s$$

$$Drift\,Vector = Current * time = \frac{(-1,-1)m}{s} * 6.1s = (-6.1,-6.1)m$$

$$Target\,Vector = Waypoint\,Vector - Drift\,Vector = (10 + 6.1, 7 + 6.1)m = (16.1, 13.1)m$$

$$Heading = \tan^{-1}(16.1/13.1) = 0.88779rad$$

Figure 13-12: Old method equations to determine the drift vector

In this example, the heading would be set to **0.88779**rad, measured from the y-axis.

In Figure 13-13, to reach the green dot, the glider heading is set at the red dot.

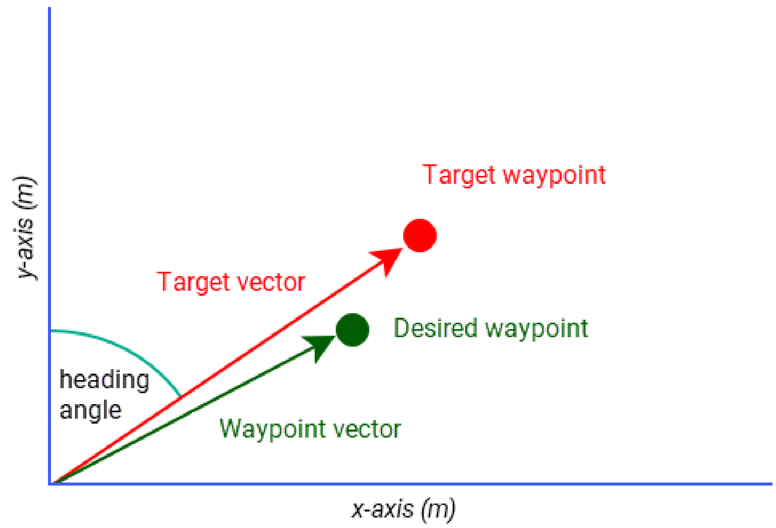

Figure 13-13: Plot 2

Initially, this method seems totally acceptable. However, on inspection of the distance to the waypoint vector (green) and the target vector (red), a problem becomes evident: these two vectors are different lengths.

In the time used to determine these values, the glider can only move a distance from the origin equal to the length of the green vector.

This vector mismatch between green and red results in the glider being at the wrong location.

Using the equations in Figure 13-14, where V_{dx} and V_{dy} are the water currents, V_v is the glider's horizontal speed, and θ is the heading angle, we get final position of the glider:

$x = V_{dx}t + V_v sin(\theta)t$	$y = V_{dy}t + V_v cos(\theta)t$

Figure 13-14: Calculations for final position of the glider

Plugging in the values of this example case, the final position of the glider will be (3.36328, 1.59976)m—that misses the desired waypoint of (10, 7)m by a large amount.

The cause of the inaccuracy is the previously described vector mismatch. Therefore, we can expect a few resulting behaviors and trends:

1. An object following this determined heading veers away from the desired waypoint towards this "phantom point."

   However, as time progresses, the phantom point's distance from the actual waypoint should decrease as the object adjusts its position and performs another heading determination.

   Therefore, arching trajectories are expected for objects using this method.

2. This arching path becomes larger with increasing current, and:

a. Takes significantly longer to get to the desired waypoint
b. May lead to the glider hitting land in tight areas

3. This arching path becomes larger with increased distance to a waypoint.

As the Slocum glider checks its position and recalculates its heading every four seconds, the glider progressively gets closer to the desired point, even though the method is incorrect. The glider inevitably reaches the target waypoint—but in a non-optimized path. For more information, see "13.4.5 Pure Math Comparison" on page 13-17.

13.4.4 Current Correction – Method 2

To solve the issue of the previous method, two vectors are implemented:

- Vehicle velocity vector
- Drift vector

Both vectors were constrained by having the end of the vehicle vector be the beginning of the drift vector (end-to-end). Each vector can grow at their own separate rates based on the magnitude of the speed associated with that vector (vehicle speed and drift speed).

If the drift is constant, its angle with respect to the vehicle velocity vector is fixed—but the same cannot be said for the vehicle velocity vector as its heading is an unknown. In addition, the total length of these vectors is currently an unknown because the time for each vector to grow at their respective speeds is unknown. Thus, two variables that must be solved are heading angle and time.

Before solving this problem, the final constraint is the sum of these two vectors must equal the desired position. Normally, this would leave two unknowns and one equation.

However, by simply breaking the desired position into x and y components, there are now two unknowns and two constraint equations that can be solved, illustrated in Figure 13-15:

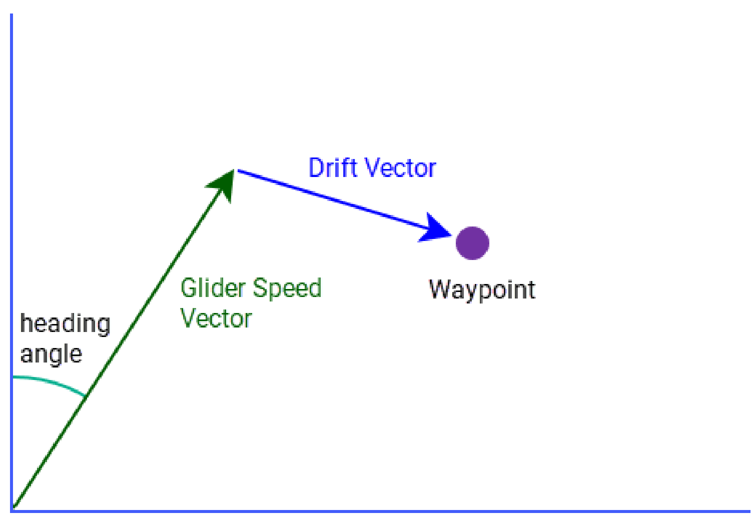

Figure 13-15: Plot 3

Figure 13-15 above shows what we want to attain. The drift vector direction is constant, but its size varies with time and is always attached to the end of the glider speed vector. The glider speed vector is allowed to grow with time and its direction is determined by the heading angle.

In the equations below, the solution is for *heading angle* and *time* for these conjoined vectors to grow so their sum is the waypoint coordinates.

The first step begins with the general form of the equations in Figure 13-16:

$x = V_{dx}t + V_v sin(\theta)t$	$y = V_{dy}t + V_v cos(\theta)t$

Figure 13-16: General form of the equations

Next, divide the equation into two separate portions for x and y, as shown in Figure 13-17:

$x = V_{dx}t + V_v sin(\theta)t$	$y = V_{dy}t + V_v cos(\theta)t$

Figure 13-17: General form separated into two equations

As the drift is constant, it has constant x and y components. However, the vehicle will be dependent on the heading angle, θ, and will thus be multiplied by sin(θ) and cos(θ).

In this situation, two unknowns must be solved:

- t for time
- theta (θ) for the heading angle

Fortunately, two equations can solve the two unknowns. The resulting equations are shown in Figure 13-18:

$t = \frac{x}{W_x + Vsin(\theta)}$	$\cos(\theta) - \frac{y}{x}\sin(\theta) = \frac{1}{V}(\frac{yW_x}{x} - W_y) = c$

Figure 13-18: Two equations for two unknowns

The answer c in the second equation is:

- For neatness of the equations that follow, and
- Equal to the constants on the right side of the equal sign.

The equation on the right in Figure 13-18 must be solved for theta. Given the equation is a polynomial, multiple solution cases are listed in Figure 13-19:

CASE 1

$$Given: (c+1)x \neq 0 \; and \; (c+1)x^2 + y^2 \neq y\sqrt{-c^2x^2 + x^2 + y^2}$$

$$\theta = 2(\pi n - \tan^{-1}\left(\frac{y - \sqrt{-c^2x^2 + x^2 + y^2}}{cx + x}\right))$$

CASE 2

$$Given: (c+1)x \neq 0 \; and \; y\left(\sqrt{-c^2x^2 + x^2 + y^2} + y\right) + (c+1)x^2 \neq 0$$

$$\theta = 2(\pi n - \tan^{-1}\left(\frac{\sqrt{-c^2x^2 + x^2 + y^2} + y}{cx + x}\right))$$

CASE 3

$$Given: y \neq 0 \; and \; x(x^2 + y^2) \neq 0 \; amd \; c = -1$$

$$\theta = 2(\pi n - \tan^{-1}(\frac{x}{y}))$$

CASE 4

$$Given: x \neq 0 \; and \; c = -1$$

$$\theta = 2\pi n + \pi$$

Figure 13-19: Four cases (1)

The variable n in all four cases is any integer from negative infinity to positive infinity. This mathematically gives us an infinite number of solutions for all four cases.

Practically, the $n = 0$ solution is one that is of interest because adding a factor of 2π or -2π (360 degrees or negative 360 degrees instead of radians) to any angle produces the same angle.

For example:

0.2rad = (0.2 + 2π)rad = (0.2 +4π)rad = (0.2 +6π)rad

and the same holds true for negative 2π values.

So, in coding, ignore the $2\pi n$ terms entirely and the four cases reduce to the ones shown in Figure 13-20:

CASE 1

$$Given: (c+1)x \neq 0 \; and \; (c+1)x^2 + y^2 \neq y\sqrt{-c^2x^2 + x^2 + y^2}$$

$$\theta = -2\tan^{-1}\left(\frac{y - \sqrt{-c^2x^2 + x^2 + y^2}}{cx + x}\right)$$

CASE 2

$$Given: (c+1)x \neq 0 \; and \; y\left(\sqrt{-c^2x^2 + x^2 + y^2} + y\right) + (c+1)x^2 \neq 0$$

$$\theta = -2\tan^{-1}\left(\frac{\sqrt{-c^2x^2 + x^2 + y^2} + y}{cx + x}\right)$$

CASE 3

$$Given: y \neq 0 \; and \; x(x^2 + y^2) \neq 0 \; amd \; c = -1$$

$$\theta = -2\tan^{-1}\left(\frac{x}{y}\right)$$

CASE 4

$$Given: x \neq 0 \; and \; c = -1$$

$$\theta = \pi$$

Figure 13-20: Four cases (2)

13.4.4.1 Time Check and Meaning

Once these cases are used to determine a value for theta, the calculated value can be inserted into the equation for time, as shown in Figure 13-21:

$$t = \frac{x}{W_x + V sin(\theta)}$$

Figure 13-21: Equation for time

Any solution that yields a negative time value is invalid (mathematically, time can be negative; practically, obviously not). This is an issue in cases 1 and 2 only, which can have overlap and have their given statements both satisfied.

In the situation where cases 1 and 2 are satisfied, whichever case result yields a positive time value is the result that should be used for the heading. The software checks for this and selects the result with non-negative time.

Note
If the software detects both cases 1 and 2 are satisfied and yield a negative time value, the glider sets its heading directly towards the waypoint and proceeds (but fails).

If both cases 1 and 2 are satisfied and yield a negative time value, this indicates the:

- Current is too strong to overcome
- Desired waypoint is unreachable under present current conditions and speed.

13.4.5 Pure Math Comparison

The figures in this section are visual representations of the differences between the two methods. Using the different heading calculations, the expected positions of a glider are plotted with the following parameters:

- *Waypoint* [200,100]
- *Glider horizontal speed* 1m/s
- *Current* [0.2, 0.2]m/s

For this example:

- Just like an actual glider, the heading was recalculated every 4 seconds.
- The glider reaches the waypoint within a 3m radius.

Figure 13-22 on page 13-18 is a comparison of a glider positions using the two different heading methods under the conditions stated above.

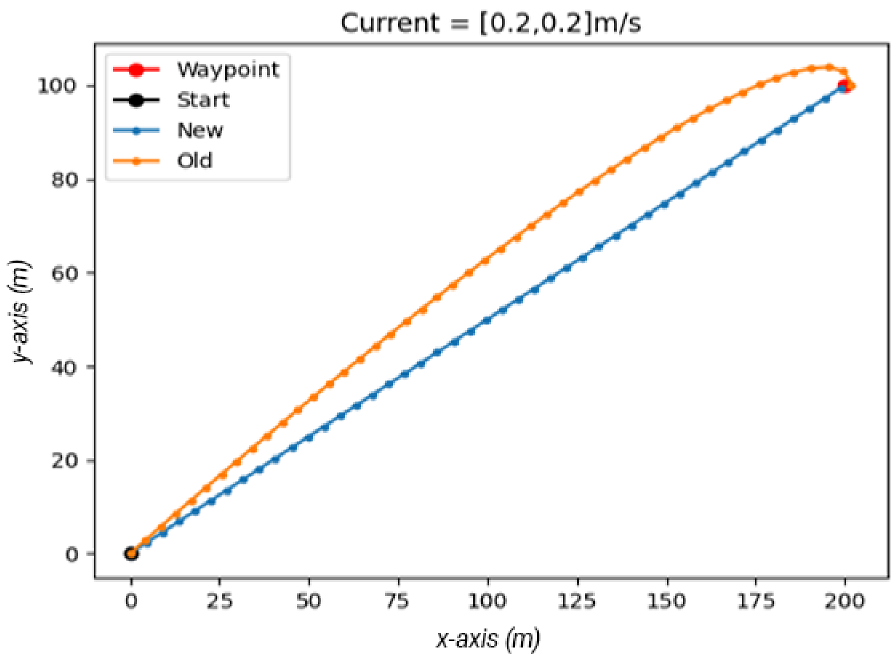

Figure 13-22: Heading comparison 1

The y-axis and x-axis numbers are the glider's coordinates in meters:

- The previous method produced the arched path.
  The better method produces a straight line.
- The previous method took 184 seconds to reach the waypoint.
  The better method took 176 seconds, saving 8 seconds.

The savings do not appear large because this trip was only 224m long.

Over larger distances, this time savings grows; the arched path becomes more pronounced.

In stronger currents, the time saved by the better method grows even larger.

Figure 13-23 on page 13-19 is a graph of the heading values of the glider comparing the previous and better methods.

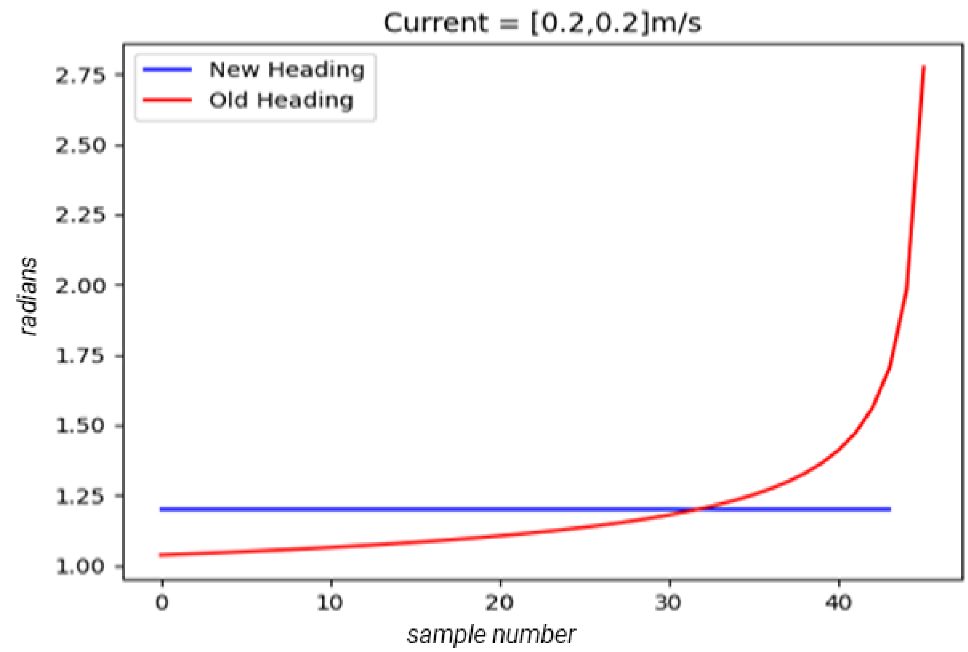

Figure 13-23: Heading comparison 2

The y-axis is in radians and the x-axis is the sample number:

- While under a constant current, the better method produces a flatline heading plot. The previous method gradually changing over the course.
- The better method reduces the fin movement of the gliders with this new, more stable heading.

14 Flight Data Retrieval

14.1 Overview

The *Dock Server* can automate file retrieval and data storage, and display data and glider locations for easy viewing. Dock Server 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 chapter explains how to retrieve 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 (or their compressed versions) by using the FreeWave modem.

Table 14-1: Data File Types

Data File Extension	Compressed File Extension	Description
.dbd	.dcd	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	.scd	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	.mcd	Medium binary data — A second user specified data set specified in mbdlist.dat.
.mlg	.mcg	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.

A separate document, *SFMC User Manual* [P/N M313834-NFC], provides instructions for using Glider Terminal (a Java application) and one of the Dock Server applications. To obtain a copy of that publication, contact glidersupport@teledyne.com.

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

14.2 Retrieving Data When at the Surface

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

1. Type ***one*** of the following, either regular or compressed:
   - Flight Side:

     ```
     s *.dbd
     s *.sbd
     s *.mbd
     s *.mlg
     ```

- Compressed Flight Side:

```
s *.dcd
s *.scd
s *.mcd
s *.mcg
```

Note
Commands listed in lower case in the *Help* menu are not available ***during*** a mission.

Note
The **send *.*** command sends all files of type: **.sbd**, **.mbd**, **.dbd**, **.mlg**, **.tbd**, **.nbd**, **.ebd**, and **.nlg**.

2. The 30 most recent files of the type specified in Step 1 are transmitted. Remember, sending DBD files over Iridium is not recommended:

To retrieve ***glider*** data while the glider is not running a mission from a GliderDOS prompt, type one of the following (regular, then compressed):

```
send *.dbd
send *.sbd
send *.mbd
send *.mlg
send *.*
send *.dcd
send *.scd
send *.mcd
send *.mcg
```

To retrieve ***science*** data while the glider is not running a mission from a GliderDOS prompt, type one of the following (regular, then compressed):

```
send *.ebd
send *.tbd
send *.nbd
send *.nlg
send *.*
send *.ecd
send *.tcd
send *.ncd
send *.ncg
```

3. To send specific files in GliderShell or GliderDOS, use **zmodem** to send the files desired by typing: **zs <path><filename>**

Table 14-2: Immediate Commands Available During Surface Mission Paused Dialog

Command	Description
ctrl-r	Resumes glider mission, if so programmed.
ctrl-c	Aborts the mission, closes files, and remains on the surface in GliderDOS.
ctrl-e	Extends surface time by five minutes before resuming the mission.
ctrl-p	Starts the mission immediately.
ctrl-f	Reads the **.ma** files again.
ctrl-t	Switches to communications with the science processor.
!	Followed by an allowable GliderDOS command to run a subset of GliderDOS commands. The **!** command is also referred to as "bang."

15 Post-Deployment Care

15.1 G3S Glider

After each deployment:

1. Strap glider in cart and rinse thoroughly with fresh water.
2. Visually inspect for damage, paying special attention to seals, connectors, and external components.
3. Replace damaged parts as necessary. See *Slocum Glider Maintenance Manual* [P/N M315357-NFC] for parts requiring replacement at the lab/depot.

15.2 Tail Fin

CAUTION

The loop at the aft end of the boom, beneath the rudder, is intended as a small load tie-off point.

Never use the loop to lift or suspend the vehicle out of the water. ***It is not a lifting point***.

You can, however, use it to tie a buoy for trial dives or for manipulating the vehicle using a boat hook or gaff while it is in the water.

CAUTION

Do not loosen or remove the four titanium socket head cap screws at the base of the fin boom *without explicit instruction from TWR*.

These screws provide access to the feed-through plate that connects communications subsystem antennas to the vehicle communications board.

Once you gain access to the feed through plate, you must perform leak testing.

After each deployment:

1. Rinse tail fin with fresh water to prevent salt crystals from inhibiting motion during the next power up.
2. Inspect tail fin components for damage, and replace as necessary. See *Slocum Glider Maintenance Manual* [P/N M315357-NFC] for parts requiring replacement at the lab/depot.
3. If the rudder is damaged, this can be replaced in the field:
   a. Remove the two Phillips head screws securing the mast top to the mast as shown in Figure 15-1.

Figure 15-1: Removing the two Phillips Head Screws

4. Hold on to the rudder with one hand and lift the mast top away from the fin with the other as shown in Figure 15-2:

Figure 15-2: Removing the Mast Top

5. Lift the rudder off the magnetic coupling as shown in Figure 15-3:

Figure 15-3: The Rudder Removed from Tail Fin

6. Install a new rudder and reassemble in reverse of disassembly.

15.3 Strobe

15.3.1 How to Configure

Factory configured.

15.3.2 How to Test

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

15.3.3 Relevant Sensors

Use the following commands:

```
strobe on    -or-
strobe off
```

15.4 10-Watt Thruster (Optional)

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 (included in thruster kit).
4. Replace the hub.

15.5 Sacrificial Anode

CAUTION
To ensure proper protection against corrosion: • Check anodes for continuity to ground on a regular basis • Replace periodically

CAUTION
Take note of any scratches to the glider's anodizing, as exposed aluminum parts can be corroded. Touch up scratches with paint—nail polish is effective in an emergency.

CAUTION
Rinse the glider with fresh water every time it is exposed to salt water.

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.

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

15.5.1 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.

Part 4: Appendices

A Abbreviations, Acronyms, and Terms

Table A-1 lists abbreviations, acronyms, and terms that may be used in this manual.

Table A-1: Abbreviations, Acronyms, and Terms

Item	Description
AC or ac	Alternating Current
ASSY	Assembly
AUV	Autonomous Underwater Vehicle
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
MS Plug	Military Standard vacuum seal plug
OC	Operations Center
OEM	Original Equipment Manufacturer
outgas	Release or give off (a substance) as a gas or vapor
QCP	Quality Control Process
PEEK	PolyEther Ether Ketone
P/N	Part Number
PPE	Personal Protective Equipment
proglet	A small program that meets an transient need
RF	Radio Frequency
RHEL	Red Hat Enterprise Linux
RHN	Red Hat Network
RUDICS	Router-based Unrestricted Digital Internetworking Connectivity System
SDL	Software Data Logging
SE	Systems Engineering
SHCS	Socket Head Cap Screw
SN	Serial Number
SOP	Standard Operating Procedure
SSL	Secure Sockets Layer
STE	Secure Telephone Equipment
TWR	Teledyne Webb Research

Table A-1: Abbreviations, Acronyms, and Terms

Item	Description
U.S.	United States
USB	Universal Serial Bus
UUV	Unmanned Undersea Vehicle
VAC	Volts Alternating Current

B GliderShell Commands

CAUTION

Command names in lower case are ***not*** executable during a mission via an exclamation point or "bang" (**!**). These commands are shown below in red.

?	[<cmd>] [-full] [-desc] [-usage] ; Show help on one or all commands, just names or full
ballast	[<cc>] ; prepare glider for manually ballasting
ballvalve	pos|settings ; get ballvalve position or configuration
boot	[app | shell] ; show or set boot mode... use EXIT to boot
callback	<minutes til callback> <num to dial 0=PRI 1=ALT> ; hang up iridium, call back in spec'd minutes at spec'd number
capture	[<pathspec>]<filename> [/Dx/B/N/E] ; create file from user input
CAT	[<pathspec>]<filename> ; display contents of a file
CD	[<pathspec>] ; change directory
CHDIR	[<pathspec>] ; change directory
CLRDEVERRS	; zero device errs
compass_cal	get_offsets | get_mode | set_offsets <XXX> <YYY> <ZZZ> | cancel ; get/set attitude_rev parameters
consci	[-f rf|irid] ; console to science
COPY	<source> <dest> ; copy file
core	[clear] ; show/clear core dump data
CP	<source> <dest> ; copy file
date	[<yyyy-mm-dd> [<h:m:s>]] ; set/get date & time
DEL	[<pathspec>]<filename...> ; delete file(s), wildcards supported
DELLOG	[ALL|MLG|DBD|SBD|MBD] ; delete log files
DEVICES?	; print device driver info
DF	; Print disk space used and disk space free
digifin	<many options> ; digifin command, ? for help
DIR	[-l][<pathspec>][<filename>] ; list directory contents
drift_table	[<many options>] ; Set/show neutral ballast vs depth table used by drift_at_depth behavior
echo	[<args...>] ; echo args on teminal
EXECUTE	<filename> ; Execute a Glider Batch File (*.GBF)
exit	[-nofin] [poweroff | reset | shell] (default is poweroff) ; completely shut down glider, and reset if not 'poweroff'
freport	+|++|- <sensor list filename> ; read file to get list of sensors to show on cmd line
GET	<sensor name> ; show sensor value and units
glider_test	[indoor|outdoor] [manual|automatic] ; glider checkout (formerly FCP)
HARDWARE?	[-v] ; Show glider hardware info, -v for more
HEAP	; Report Free Memory

```
HELP            [<cmd>] [-full] [-desc] [-usage] ; Show help on some or all commands, just names or full
hs              on | off ; Turn on or off high speed serial mode
LAB_MODE        [on|off] ; set or show lab mode state
LIST            [<partial or full sensor name>] ; list some or all sensor names and values.
                    can use ctl-C
loadmission     <mission filename> ; loads mission file
logging         [on|off] ; turn on/off data logging (show state if no arg)
LONGTERM        + <sensor_name ..> | - <sensor_name ..> | clearall | list ; set (or list) longterm
                    sensors config file
LONGTERM_PUT    <sensor name> <new value> ; set sensor to value and add sensor to longterm config file
LS              [-1][<pathspec>][<filename>] ; list directory contents
MBD             + <sensor_name ..> | - <sensor_name ..> | clearall | load <filename> | list ;
                    edit/set/list sensors to be logged in MBD
MKDIR           <pathspec> ; create directory
MV              [<pathspec>]<oldname> <newname> ; move/rename file
PRUNEDISK       ; disk space recovery: Prune expendable files to free space on disk
PURGELOGS       ; Deletes sent log files
PUT             <sensor name> <value> ; set sensor value
RENAME          [<pathspec>]<oldname> <newname> ; rename/move file
REPORT          + <sensor_name ..> | ++ <sensor_name ..> | - <sensor_name ..> | all | clearall | list ;
                    show sensor(s) status on cmd line
RM              [<pathspec>]<filename> ; delete file
RMDIR           <pathspec> ; delete directory
run             [<mission_filename>] ; runs mission specified, or initial.mi if no param
SBD             + <sensor_name ..> | - <sensor_name ..> | clearall | load <filename> | list ;
                    edit/set/list sensors to be logged in SBD
SEND            [-f rf|irid] [-t=<max seconds>] [-num=<max files>] [<send_filespec ...>] ;
                    send logfiles from glider
sequence        <mission-specifier> ... | -report | -resume | -resume-next ; sequence mission(s), or
                    print last mission and where it terminated
SETDEVLIMIT     <devicename> <oos> <w/s> <w/m> ; set warning limits for device
SETNUMWARN      [<number>] ; set max dev warnings to <number>
SIMUL?          ; print desc of what is simulated
SRF_DISPLAY     + <sensor_name ..> | - <sensor_name ..> | clearall | list ; config (or list) which
                    sensors to display in surface dialog
strobe          [on|off] ; shows or sets strobe light flashing
sync_time       ; Syncs system time with gps time
SZR             [-f rf|irid] [-v <level>] ; Zmodem Receive file(s) to science proc. 'szr ?' for details
SZS             [-f rf|irid] [-v <level>] <filename>... ; Zmodem Send file(s) from science proc.
                    'szs ?' for details
talk            [-f rf|irid] [<other args>] ; connect user input to a serial port
tvalve          [up|charge|down [backward]] ; shows and optionally sets thermal valve settings
TYPE            [<pathspec>]<filename> ; display contents of a file
```

```
USE              [+ <device> |- <device> | all | none] ; lists devices and if in use, or takes devices
                     in or out of service
VER              ; Show firmware versions
WHERE            ; prints lat/lon
whoami           ; current vehicle name (who am I)
whoru            ; current vehicle name (who are you)
WHY?             [<abort number>] ; Tells the reason for an abort
wiggle           [on|off [percent]] ; set or show wiggle settings (motor test). lab mode only
ZERO_OCEAN_PRESSURE   ; re-calibrate(zero) ocean pressure sensor
ZR               [-f rf|irid] [-v <level>] [-nosmartdir] ; Zmodem Receive file(s). 'zr ?' for details
ZS               [-f rf|irid] [-v <level>] <filename>... ; Zmodem Send file(s). 'zs ?' for details

       HELP                  list all commands in alphabetical order
       HELP -full            list all commands with their usage and descriptions
       HELP -desc            list all commands with their descriptions
       HELP -usage           list all commands with their usage

       HELP <cmd> ... <cmd>  prints the full help msgs for listed commands.
           Note that -desc and -usage can also be used
           command names in lower case are NOT executable in mission via !
```

C GliderDOS Commands

This is the flight DOS command list.

```
BOOT          [app | shell] ; show or set boot mode... use EXIT to boot
CAPTURE       [<pathspec>]<filename> [/Dx/B/N/E] ; create file from user input
CAT           [<pathspec>]<filename> ; display contents of a file  //unix
TYPE          [<pathspec>]<filename> ; display contents of a file //dos
CD            [<pathspec>] ; change directory //unix
CHDIR         [<pathspec>] ; change directory //dos
CP            <source> <dest> ; copy file //unix
COPY          <source> <dest> ; copy file //dos
CORE          [clear] ; show/clear core dump data
DATE          [<yyyy-mm-dd> [<h:m:s>]] ; set/get date & time
DEL           [<pathspec>]<filename...> ; delete file(s), wildcards supported
DF            ; Print disk space used and disk space free
DIR           [-l][<pathspec>][<filename>] ; list directory contents //dos
ECHO          [<args...>] ; echo args on terminal
HELP          [<cmd>] [-full] [-desc] [-usage] ; Show help on some or all commands, just names or full
HS            on | off ; Turn on or off high speed serial mode
LS            [-l][<pathspec>][<filename>] ; list directory contents //unix
MKDIR         <pathspec> ; create directory
MV            [<pathspec>]<oldname> <newname> ; move/rename file //unix
PURGELOGS     ; Deletes sent log files
RENAME        [<pathspec>]<oldname> <newname> ; rename/move file //dos
RM            [<pathspec>]<filename> ; delete file //unix
RMDIR         <pathspec> ; delete directory
VER           ; Show firmware versions
HARDWARE?     [-v] ; Show glider hardware info, -v for more
RUN           [<mission_filename>] ; runs mission specified, or initial.mi if no param
LOADMISSION   <mission filename> ; loads mission file
SEQUENCE      <mission-specifier> ... | -report | -resume | -resume-next ; sequence mission(s), or
                 print last mission and where it terminated
USE           [+ <device> |- <device> | all | none] ; lists devices and if in use, or takes devices
              in or out of service
DEVICES?      ; print device driver info
CLRDEVERRS    ; zero device errs
SETNUMWARN    [<number>] ; set max dev warnings to <number>
EXIT          [-nofin] [poweroff | reset | shell] (default is poweroff) ; completely shut down glider,
                 and reset if not 'poweroff'  // reset, restart, and shell have same result
GET           <sensor name> ; show sensor value and units
PUT           <sensor name> <value> ; set sensor value
```

```
LIST           [<partial or full sensor name>] ; list some or all sensor names and values.
                   Can use ctl-C
REPORT         + <sensor_name ..> | ++ <sensor_name ..> | - <sensor_name ..> | all | clearall | list ;
                   show sensor(s) status on cmd line
FREPORT        +|++|- <sensor list filename> ; read file to get list of sensors to show on cmd line
SRF_DISPLAY    + <sensor_name ..> | - <sensor_name ..> | clearall | list ; config (or list) which
                   sensors to display in surface dialog
LONGTERM       + <sensor_name ..> | - <sensor_name ..> | clearall | list ; set (or list) longterm
                   sensors config file
SBD            + <sensor_name ..> | - <sensor_name ..> | clearall | load <filename> | list ;
                   edit/set/list sensors to be logged in SBD
DELLOG         [ALL|MLG|DBD|SBD|MBD] ; delete log files
LOGGING        [on|off] ; turn on/off data logging (show state if no arg)
SYNC_TIME      ; Syncs system time with gps time
WIGGLE         [on|off [percent]] ; set or show wiggle settings (motor test). lab mode only
LAB_MODE       [on|off] ; set or show lab mode state
SETDEVLIMIT    <devicename> <oos> <w/s> <w/m> ; set warning limits for device // ; args include
                   out-of-service-count, warnings-per-segment, warnings-per-minute
CALLBACK       <minutes til callback> <num to dial 0=PRI 1=ALT> ; hang up iridium, call back in spec'd
                   minutes at spec'd number
ZS             [-f rf|irid] [-v <level>] <filename>... ; Zmodem Send file(s). \'zs ?\' for details
ZR             [-f rf|irid] [-v <level>] [-nosmartdir] ; Zmodem Receive file(s). \'zr ?\' for details
WHORU          ; current vehicle name (who are you)
WHOAMI         ; current vehicle name (who am I)
HEAP           ; Report Free Memory
WHERE          ; prints lat/lon
TVALVE         [up|charge|down [backward]] ; shows and optionally sets thermal valve settings
                   // actually, optionally operates it

#if defined(USE_GCMDDF)
DF             ; Print disk space used and disk space free
#endif // USE_GCMDF

PRUNEDISK      ; disk space recovery: Prune expendable files to free space on disk
SEND           [-f rf|irid] [-t=<max seconds>] [-num=<max files>] [<send_filespec ...>] ;
                   send logfiles from glider
CONSCI         [-f rf|irid] ; console to science
WHY?           [<abort number>] ; Tells the reason for an abort
LONGTERM_PUT   <sensor name> <new value> ; set sensor to value and add sensor to longterm config file
BALLAST        [<cc>] ; prepare glider for manually ballasting
ZERO_OCEAN_PRESSURE   ; re-calibrate(zero) ocean pressure sensor
SIMUL?         ; print desc of what is simulated
MBD            + <sensor_name ..> | - <sensor_name ..> | clearall | load <filename> | list ;
                   edit/set/list sensors to be logged in MBD
```

DIGIFIN	<many options> ; digifin command, ? for help
STROBE	[on|off] ; shows or sets strobe light flashing
DRIFT_TABLE	[<many options>] ; Set/show neutral ballast vs depth table used by drift_at_depth behavior
EXECUTE	<filename> ; Execute a Glider Batch File (*.GBF)
SZR	[-f rf|irid] [-v <level>] ; Zmodem Receive file(s) to science proc. \'szr ?\' for details
SZS	[-f rf|irid] [-v <level>] <filename>... ; Zmodem Send file(s) from science proc. \'szs ?\' for details
COMPASS_CAL	get_offsets | get_mode | set_offsets <XXX> <YYY> <ZZZ> | cancel ; get/set attitude_rev parameters
TALK	[-f rf|irid] [<other args>] ; connect user input to a serial port
?	[<cmd>] [-full] [-desc] [-usage] ; Show help on some or all commands, just names or full

D SciShell & SciDOS Commands

```
?             [<cmd>] [-full] [-desc] [-usage] ; Show help on one or all commands, just names or full
BIT           <bit number> [<action>] ; show or set/clear/pulse gpio bit
BOOT          [app | shell] ; show or set boot mode. Use REBOOT to boot
CAPTURE       [<pathspec>]<filename> ; create file from user input
CAT           [<pathspec>]<filename> ; display contents of a file  //unix
CD            [<pathspec>] ; change directory //unix
CHDIR         [<pathspec>] ; change directory //dos
CP            <source> <dest> ; copy file //unix
COPY          <source> <dest> ; copy file //dos
CORE          [clear] ; show/clear core dump data
DATE          [<yyyy-mm-dd> [<h:m:s>]] ; set/get date & time
DEL           [<pathspec>]<filename> ; delete file(s), wildcards supported
DF            ; Print disk space used and disk space free
DIR           [-l] [<pathspec>][<filename>] ; list directory contents //unix
ECHO          [<args...>] ; echo args on terminal
HELP          [<cmd>] [-full] [-desc] [-usage] ; Show help on one or all commands, just names or full
LS            [-l] [<pathspec>][<filename>] ; list directory contents //dos
MKDIR         <pathspec> ; create directory  //unix
MV            [<pathspec>]<oldname> <newname> ; move/rename file //unix
REBOOT        ; Reboot the processor
RENAME        [<pathspec>]<oldname> <newname> ; rename/move file //dos
RM            [<pathspec>]<filename> ; delete file //unix
RMDIR         <pathspec> ; delete directory
TYPE          [<pathspec>]<filename> ; display contents of a file //dos
UART          <port-name> <baudrate> [<control char>] [<power/control bit>...] ; write user
                  input to uart port
VER           ; Show science FW version, debug level, opt level

// App specific commands
CLOSELOG;     close log files
DELLOG        [ALL|NLG|EBD|TBD|NBD] ; delete log files

#if defined(USE_SCMDDF)
DF            ; Print disk space used and disk space free
#endif // USE_SCMDDF

EXIT          ; Exit SciDos
GET           <sensor name> ; show sensor value and units
HEAP          ; Report Free Memory
```

```
LIST          ; List currently known sensors
PRUNEDISK;    Prune expendable files to free space on disk
PURGELOGS;    Delete sent log files
PUT           <sensor name> <value> ; set sensor value
QUIT          ; Exit SciDos
SEND          [-t=<max seconds>] [-num=<max files>] [<send_filespec ...>] ; send logfiles from science
ZR            [-v <level>] [-nosmartdir] <filename>... ; Receive file(s) using Zmodem,
                  zr ? for details
ZS            [-v <level>] <filename>... ; Send file(s) using Zmodem, zs ? for details
TBD           + <sensor_name ..> | - <sensor_name ..> | clearall | load <filename> | list
                  ; edit/set/list sensors to be logged in TBD
NBD           + <sensor_name ..> | - <sensor_name ..> | clearall | load <filename> | list
                  ; edit/set/list sensors to be logged in NBD
```

E Abort Codes

The following is the list of mission status codes (MS_*) and includes all the abort codes (MS_ABORT_*).

```
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,            // Missing control for buoyancy, pitch, or heading
MS_ABORT_HEADING_IS_IDLE = 2,          // No one at the helm
MS_ABORT_PITCH_IS_IDLE = 3,            // Missing pitch control
MS_ABORT_BPUMP_IS_IDLE = 4,            // Missing (shallow) buoyancy pump control
MS_ABORT_THRENG_IS_IDLE = 5,           // Missing thermal engine control
MS_ABORT_BEH_ERROR = 6,                // behavior entered error state, usually bad b_args
MS_ABORT_OVERDEPTH = 7,                // went too deep - depth > abend:overdepth
MS_ABORT_OVERTIME = 8,                 // went too long - time > abend:overtime
MS_ABORT_UNDERVOLTS = 9,               // batteries getting low - voltage < abend:undervolts
MS_ABORT_SAMEDEPTH_FOR = 10,           // not moving vertically for > abend:samedepth_for
MS_ABORT_USER_INTERRUPT = 11,          // hit control-C
MS_ABORT_NOINPUT = 12,                 // some required device not producing data
MS_ABORT_INFLECTION = 13,              // inflection took too long
MS_ABORT_NO_TICKLE = 14,               // watchdog not serviced, abort before we blow the weight
MS_ABORT_ENG_PRESSURE = 15,            // Thermal engine pressure too low
MS_ABORT_DEVICE_ERROR = 16,            // a required device produced too many / severe errors
MS_ABORT_DEV_NOT_INSTALLED = 17,       // a required device is not there
MS_ABORT_WPT_TOOFAR = 18,              // distance to waypoint > abend:max_wpt_distance
MS_ABORT_UNREASONABLE_SETTINGS = 19,   // global sensors are inconsistent
MS_ABORT_LMC_NOT_FIXED = 20,           // internal navigation error
MS_ABORT_NO_HEAP = 21,                 // out of memory
MS_ABORT_LOG_DATA_ERROR = 22,          // error logging data
MS_ABORT_THERMAL_NOT_ENABLED = 23,     // thermal mission without thermal engine
MS_ABORT_LEAK = 24,                    // keeping the ocean out is Job One...
MS_ABORT_VACUUM = 25,                  // vacuum < abend:vacuum_min or > abend:vacuum_max
MS_ABORT_NO_HEADING_MEASUREMENT = 26,  // compass is busted
MS_ABORT_STALLED = 27,                 // not moving horizontally for > abend:stalled_for
MS_ABORT_DE_PUMP_IS_IDLE = 28,         // Missing (shallow) pump control
MS_ABORT_DE_PUMP_NOT_ENABLED = 29,     // deep mission without deep engine
MS_ABORT_CPU_LOADED = 30,              // computer too busy
MS_ABORT_NO_ABEND_BEHAVIOR = 31,       // no abend behavior in mission
```

```
MS_ABORT_LOW_REL_CHARGE = 32,            // batteries getting low on energy
MS_ABORT_WEIGHT_DROPPED = 33,            // weight has been dropped, not going to fly
MS_ABORT_INITIALIZATION_ERROR = 34,      // empty mission or bad mission sensors
MS_ABORT_CRITICAL_ABORT_ACTIVE = 35,     // critical abort has occurred, not going to fly
MS_ABORT_INVALID_GPS = 36,               // too many segments in a row without a valid gps fix
MS_ABORT_NO_COMMS_TICKLE = 37,           // no comms for this long
MS_ABORT_EMERGENCY_BATTERY_ACTIVE = 38,  // 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
```

F Sensor Codes

This is the list of sensor codes. The code number is on the right.

The sensor's name is the first entry if you remove the **C_** and **_ON**. For example, C_BAM_ON is for the BAM sensor that has the code number of 15.

```
{-1, -1, -1}, // index 0 reserved

{C_PROFILE_ON, SCI_CTD_IS_INSTALLED, SCI_WATER_PRESSURE        },        // 1
{C_OBSOLETE_ON, SCI_OBSOLETE_IS_INSTALLED, SCI_OBSOLETE_VAR          },  // 2
{C_BB2F_ON, SCI_BB2F_IS_INSTALLED, SCI_BB2F_B470             },          // 3
{C_BB2C_ON, SCI_BB2C_IS_INSTALLED, SCI_BB2C_BETA532          },          // 4
{C_BB2LSS_ON, SCI_BB2LSS_IS_INSTALLED, SCI_BB2LSS_BETA880        },      // 5
{C_SAM_ON, SCI_SAM_IS_INSTALLED, SCI_SAM_S1                },            // 6
{C_OBSOLETE_ON, SCI_OBSOLETE_IS_INSTALLED, SCI_OBSOLETE_VAR          },  // 7
{C_OBSOLETE_ON, SCI_OBSOLETE_IS_INSTALLED, SCI_OBSOLETE_VAR          },  // 8
{C_MOTEOPD_ON, SCI_MOTEOPD_IS_INSTALLED, SCI_MOTEOPD_SN           },      // 9
{C_BBFL2S_ON, SCI_BBFL2S_IS_INSTALLED, SCI_BBFL2S_BB_SIG         },      // 10
{C_FL3SLO_ON, SCI_FL3SLO_IS_INSTALLED, SCI_FL3SLO_CHLOR_SIG      },      // 11
{C_BB3SLO_ON, SCI_BB3SLO_IS_INSTALLED, SCI_BB3SLO_B470_SIG       },      // 12
{C_OXY3835_ON, SCI_OXY3835_IS_INSTALLED, SCI_OXY3835_OXYGEN       },     // 13
{C_WHFCTD_ON, SCI_WHFCTD_IS_INSTALLED, SCI_WHFCTD_REF_HI         },      // 14
{C_BAM_ON, SCI_BAM_IS_INSTALLED, SCI_BAM_SCIENCE_ON        },            // 15
{C_OCR504R_ON, SCI_OCR504R_IS_INSTALLED, SCI_OCR504R_RAD1         },     // 16
{C_OCR504I_ON, SCI_OCR504I_IS_INSTALLED, SCI_OCR504I_IRRAD1       },     // 17
{C_BADD_ON, SCI_BADD_MMP_IS_INSTALLED, SCI_BADD_POWER_ON         },      // 18
{C_FLNTU_ON, SCI_FLNTU_IS_INSTALLED, SCI_FLNTU_CHLOR_SIG       },        // 19
{C_FL3SLOV2_ON, SCI_FL3SLOV2_IS_INSTALLED, SCI_FL3SLOV2_CHLOR_SIG    },  // 20
{C_BB3SLOV2_ON, SCI_BB3SLOV2_IS_INSTALLED, SCI_BB3SLOV2_B532_SIG     },  // 21
{C_OCR507R_ON, SCI_OCR507R_IS_INSTALLED, SCI_OCR507R_RAD1         },     // 22
{C_OCR507I_ON, SCI_OCR507I_IS_INSTALLED, SCI_OCR507I_IRRAD1       },     // 23
{C_BB3SLOV3_ON, SCI_BB3SLOV3_IS_INSTALLED, SCI_BB3SLOV3_B532_SIG     },  // 24
{C_BB2FLS_ON, SCI_BB2FLS_IS_INSTALLED, SCI_BB2FLS_B660_SIG       },      // 25
{C_BB2FLSV2_ON, SCI_BB2FLSV2_IS_INSTALLED, SCI_BB2FLSV2_B470_SIG     },  // 26
{C_OXY3835_WPHASE_ON, SCI_OXY3835_WPHASE_IS_INSTALLED, SCI_OXY3835_WPHASE_OXYGEN },  // 27
{C_AUVB_ON, SCI_AUVB_IS_INSTALLED, SCI_AUVB_SIG              },          // 28
{C_BB2FV2_ON, SCI_BB2FV2_IS_INSTALLED, SCI_BB2FV2_B470_SIG       },      // 29
{C_TARR_ON, SCI_TARR_IS_INSTALLED, SCI_TARR_TRACK_COUNT      },          // 30
{C_BBFL2SV2_ON, SCI_BBFL2SV2_IS_INSTALLED, SCI_BBFL2SV2_BB_SIG       },  // 31
{C_GLBPS_ON, SCI_GLBPS_IS_INSTALLED, SCI_GLBPS_ROUND_TRIP_TIME },        // 32
{C_SSCSD_ON, SCI_SSCSD_IS_INSTALLED, SCI_SSCSD_TEST            },        // 33
```

```
{C_BB2FLSV3_ON, SCI_BB2FLSV3_IS_INSTALLED, SCI_BB2FLSV3_B715_SIG     },          // 34
{C_FIRE_ON, SCI_FIRE_IS_INSTALLED, SCI_FIRE_FO             },                    // 35
{C_OBSOLETE_ON, SCI_OBSOLETE_IS_INSTALLED, SCI_OBSOLETE_VAR          },          // 36
{C_BB2FLSV4_ON, SCI_BB2FLSV4_IS_INSTALLED, SCI_BB2FLSV4_B412_SIG     },          // 37
{C_BB2FLSV5_ON, SCI_BB2FLSV5_IS_INSTALLED, SCI_BB2FLSV5_B532_SIG     },          // 38
{C_LOGGER_ON, SCI_LOGGER_IS_INSTALLED, SCI_LOGGER_STATUS         },              // 39
{C_BBAM_ON, SCI_BBAM_IS_INSTALLED, SCI_BBAM_BEAM_C             },                // 40
{C_UMODEM_ON, SCI_UMODEM_IS_INSTALLED, SCI_UMODEM_ERROR          },              // 41
{C_RINKOII_ON, SCI_RINKOII_IS_INSTALLED, SCI_RINKOII_TEMP          },            // 42
{C_DVL_ON, SCI_DVL_IS_INSTALLED, SCI_DVL_SA_PITCH           },                   // 43
{C_BB2FLSV6_ON, SCI_BB2FLSV6_IS_INSTALLED, SCI_BB2FLSV6_B532_SIG     },          // 44
{C_FLBBRH_ON, SCI_FLBBRH_IS_INSTALLED, SCI_FLBBRH_BB_SIG         },              // 45
{C_FLUR_ON, SCI_FLUR_IS_INSTALLED, SCI_FLUR_SIG              },                  // 46
{C_BB2FLSV7_ON, SCI_BB2FLSV7_IS_INSTALLED, SCI_BB2FLSV7_B532_SIG     },          // 47
{C_FLBBCD_ON, SCI_FLBBCD_IS_INSTALLED, SCI_FLBBCD_BB_SIG         },              // 48
{C_DMON_ON, SCI_DMON_IS_INSTALLED, SCI_DMON_MSG_BYTE_COUNT   },                  // 49
{C_C3SFL_ON, SCI_C3SFL_IS_INSTALLED, SCI_C3SFL_CH1_SIG         },                // 50
{C_SUNA_ON, SCI_SUNA_IS_INSTALLED, SCI_SUNA_RECORD_OFFSET    },                  // 51
{C_SATPAR_ON, SCI_SATPAR_IS_INSTALLED, SCI_SATPAR_PAR            },              // 52
{C_VSF_ON, SCI_VSF_IS_INSTALLED, SCI_VSF_100_SCALED        },                    // 53
{C_OXY4_ON, SCI_OXY4_IS_INSTALLED, SCI_OXY4_OXYGEN           },                  // 54
{C_OBSOLETE_ON, SCI_OBSOLETE_IS_INSTALLED, SCI_OBSOLETE_VAR          },          // 55
{C_BSIPAR_ON, SCI_BSIPAR_IS_INSTALLED, SCI_BSIPAR_PAR            },              // 56
{C_FLBB_ON, SCI_FLBB_IS_INSTALLED, SCI_FLBB_CHLOR_SIG        },                  // 57
{C_VR2C_ON, SCI_VR2C_IS_INSTALLED, SCI_OBSOLETE_VAR          },                  // 58
{C_CTD41CP2_ON, SCI_CTD41CP2_IS_INSTALLED, SCI_WATER_PRESSURE2       },          // 59
{C_ECHOSNDR853_ON, SCI_ECHOSNDR853_IS_INSTALLED, SCI_ECHOSNDR853_PING_COUNT},    // 60
{C_FLRH_ON, SCI_FLRH_IS_INSTALLED, SCI_FLRH_SIG              },                  // 61
{C_BB2FLSV8_ON, SCI_BB2FLSV8_IS_INSTALLED, SCI_BB2FLSV8_B470_SIG     },          // 62
{C_UVILUXPAH_ON, SCI_UVILUXPAH_IS_INSTALLED, SCI_UVILUXPAH_SIG         },        // 63
{C_AD2CP_ON, SCI_AD2CP_IS_INSTALLED, SCI_AD2CP_RUN_STATE       },                // 64
{C_MINIPROCO2_ON, SCI_MINIPROCO2_IS_INSTALLED, SCI_MINIPROCO2_RAWCO2     },      // 65
{C_PCO2_ON, SCI_PCO2_IS_INSTALLED, SCI_PCO2_PCO2             },                  // 66
{C_SEAOWL_ON, SCI_SEAOWL_IS_INSTALLED, SCI_SEAOWL_CHL_SCALED     },              // 67
{C_AZFP_ON, SCI_AZFP_IS_INSTALLED, SCI_AZFP_FILE_OFFSET      },                  // 68
{C_UBAT_ON, SCI_UBAT_IS_INSTALLED, SCI_UBAT_BL_POTENTIAL     },                  // 69
{C_LISST_ON, SCI_LISST_IS_INSTALLED, SCI_LISST_TOTVOL          },                // 70
{C_LMS_ON, SCI_LMS_IS_INSTALLED, SCI_LMS_METHANE           },                    // 71
{C_SVS603_ON, SCI_SVS603_IS_INSTALLED, SCI_SVS603_HEADING        },              // 72
{C_MICRORIDER_ON, SCI_MICRORIDER_IS_INSTALLED, SCI_MICRORIDER_LAST       },      // 73
```

```
{C_BB2FLSV9_ON, SCI_BB2FLSV9_IS_INSTALLED, SCI_BB2FLSV9_B532_SIG     },                    // 74
{C_SBE41N_PH_ON, SCI_SBE41N_PH_IS_INSTALLED, SCI_SBE41N_PH_REF_VOLTAGE },                  // 75
{C_FL2URRH_ON, SCI_FL2URRH_IS_INSTALLED, SCI_FL2URRH_URAN_UNITS    },                      // 76
{C_FLBBBBV1_ON, SCI_FLBBBBV1_IS_INSTALLED, SCI_FLBBBBV1_FL_SCALED    },                    // 77
{C_FLBBBBV2_ON, SCI_FLBBBBV2_IS_INSTALLED, SCI_FLBBBBV2_FL_SCALED    },                    // 78
{C_OBSVR_ON, SCI_OBSVR_IS_INSTALLED, SCI_OBSVR_PROCESSING_MODE },,                         // 79
{C_FL2PECDOM_ON, SCI_FL2PECDOM_IS_INSTALLED, SCI_FL2PECDOM_PE_UNITS    },                  // 80
{C_WETLABSA_ON, SCI_WETLABSA_IS_INSTALLED, SCI_WETLABSA_CH1_SCALED   },                    // 81
{C_WETLABSB_ON, SCI_WETLABSB_IS_INSTALLED, SCI_WETLABSB_CH1_SCALED   },                    // 82
{C_WETLABSC_ON, SCI_WETLABSC_IS_INSTALLED, SCI_WETLABSC_CH1_SCALED   },                    // 83
{C_ECHODROID_ON, SCI_ECHODROID_IS_INSTALLED, SCI_ECHODROID_SV},                            // 84
{C_TAU_ON, SCI_TAU_IS_INSTALLED, SCI_TAU_SERIALNUM},                                       // 85
{C_RBRODO_ON, SCI_RBRODO_IS_INSTALLED, SCI_RBRODO_O2_DOXY21},                              // 86
{C_SOLOCAM_ON, SCI_SOLOCAM_IS_INSTALLED, SCI_SOLOCAM_FREE_DISK_SPACE},                     // 87
{C_AMAR_ON, SCI_AMAR_IS_INSTALLED, SCI_AMAR_FW_MAJOR},                                     // 88
{C_VRO_ON, SCI_VRO_IS_INSTALLED, SCI_VRO_STATUS_STATE}                                     // 89
```

G Ancillary Equipment

For more information about ancillary equipment for the gliders, *see* the “Ancillary Glider Equipment” section in the *Slocum G3S Glider Maintenance Manual* [P/N M313476-NFC].

H Angle of Attack

Angle of Attack and Slocum Vehicle Model

Lauren Cooney
lcooney@teledyne.com

01 March 2011

1 Intro

While underwater, the glider dead reckons its position relative to the last GPS fix. For a glider without a DVL, the inputs are vehicle depth and pitch. Without the use of a DVL, any error between the final dead reckoned position and GPS surface fix is attributed to water currents (see /doco/how-it-works/water-velocity-calculations.txt). The horizontal position is the horizontal velocity integrated over time. It is clear that it is important to get the horizontal velocity as accurate as possible. We don't directly measure the horizontal velocity. It is a function of the measured vertical speed vs [m/s] (time derivative of pressure sensor output), measured pitch θ [rad], and estimated angle of attack α [rad]:

$$\tan(\theta + \alpha) = \frac{-vs}{hs} \tag{1}$$

for vehicle coordinate system shown in Fig. 1.

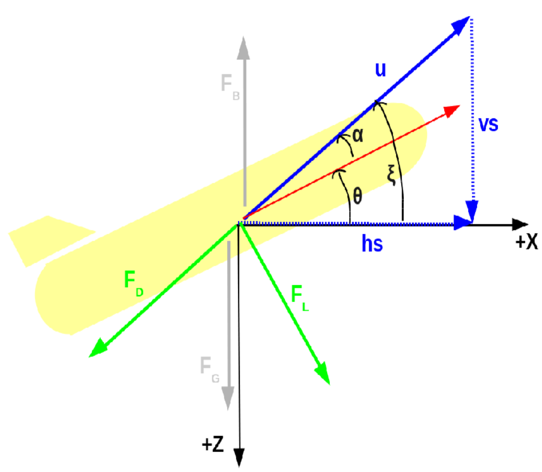

Figure 1: Slocum glider global reference frame.

2 Model Coefficients

2.1 Geometry

length	L	1.5	m
hull diameter	D	0.22	m
sweep angle	Λ	0.785	rad
wing area	S	0.0927	m^2
wing span	b	0.99	m
weight in air (shallow)	m_{Gs}	52	kg
weight in air (deep)	m_{Gd}	56	kg

Slocum Electric Geometry. Note that wing span b and area S includes both fins

2.2 Equations of Motion in Global Refence Frame

We create a simplified model of a glider in quasi-steady flight, following the methods of [4]:

$$\sum F_x = 0 = -F_D \cos\zeta - F_L \sin\zeta \tag{2}$$
$$\sum F_z = 0 = F_G - F_B + F_D \sin\zeta - F_L \cos\zeta \tag{3}$$

with drag force F_D, lift force F_L, buoyancy force F_B, force due to gravity F_G, and glide angle ζ, where $\zeta = \theta + \alpha$. Drag force and lift force are functions of surrounding fluid density ρ, reference area A_f, relative velocity v, and coefficients of drag and lift C_D and C_L respectively:

$$F_D = \frac{1}{2}\rho C_D A_f v|v| \tag{4}$$
$$F_L = \frac{1}{2}\rho C_L A_f v|v| \tag{5}$$

We substitute eqns. 4 and 5 into eqn. 2 to find:

$$C_D \cos\zeta = -C_L \sin\zeta \tag{6}$$

2.3 Lift

The total coefficient of lift C_L is the sum of lift due to the hull C_{Lh} and lift due to the wings C_{Lw}

$$C_L = C_{Lh} + C_{Lw} \tag{7}$$

2.3.1 Body Lift

For small angles of attack, Hoerner [2, pg. 13-2 and 19-13] defines non-dimensional lateral force coefficient due to the hull C_{Lh} as a function of angle of attackα.

$$\begin{aligned} C_{Lh} &= \frac{dC_{Lh}}{d\alpha}\alpha \\ &= \frac{L}{D}\frac{dC_{Lh}}{d\alpha}\alpha \end{aligned} \tag{8}$$

for which $\frac{dC_{Lh}}{d\alpha} = 0.003 \text{ deg}^{-1} = 0.172 \text{ rad}^{-1}$, and D and L are the diameter and length of the hull respectively. We find the coefficient of lift due to the hull C_{Lh}:

$$C_{Lh} = 1.172\beta \tag{9}$$

2.3.2 Wing Lift

The coefficient of lift due to the wings C_{Lw} is determined using lift theory of swept wings.

$$C_{Lw} = \frac{dC_{Lw}}{d\alpha}\alpha \tag{10}$$

The aspect ratio for the wings AR_w is

$$\begin{aligned} AR_w &= \frac{b^2}{S} && (11) \\ &= 10.573 && (12) \end{aligned}$$

For wings with aspect ratio >4, [2, pg. 15-7] gives

$$\begin{aligned} \frac{dC_{Lw}}{d\alpha^\circ} &= \frac{\cos\Lambda}{10 + 20/AR_w} && (13) \\ &= \frac{\cos 0.785}{10 + 20/10.573} && (14) \\ &= 0.0595 \text{ deg}^{-1} && (15) \end{aligned}$$

for which we find the coefficient of lift to be

$$\begin{aligned} C_{Lw} &= \frac{dC_{Lw}}{d\alpha}\alpha && (16) \\ &= (0.0595 * \frac{180}{\pi})\alpha && (17) \\ &= 3.40987\alpha && (18) \end{aligned}$$

2.4 Drag

The total coefficient of drag C_D is the sum of drag due to pressure C_{Do} and induced drag C_{Di}

$$C_L = C_{Lh} + CLw \tag{19}$$

Reynolds number R_e is the ratio of inertial to viscous forces, given in [1, Table A.2] as

$$R_e = \frac{uL}{\nu} \tag{20}$$

for speed u, characteristic length L, and fluid kinematic viscosity ν, where ν is taken to be $1.35\text{x}10^{-6} m^2/s$ at 10°deg C. Fig. 2 shows Reynolds Number as a function of characteristic glide speed u for a Slocum glider.

Hoerner [3, pg. 6-16] defines subcritical conditions to be $R_e < 10^5$, and the transitional range around R_e 10^6. Slocum gliders generally operate between these subcritical and transitional ranges, however it should be noted that turbulent flow may be tripped due to inconsistencies in the hull smoothness.

2.4.1 Pressure Drag

Hoerner [3, pg. 3-12] finds the coefficient of drag due to pressure C_{Do} to be:

$$C_{Do} = 0.44\frac{D}{L} + 4C_f(R_e)\frac{L}{D} + 4C_f(R_e)\frac{D}{L}^2 \tag{21}$$

where friction-drag coefficient C_f is a function of Reynolds Number (fig. 3).

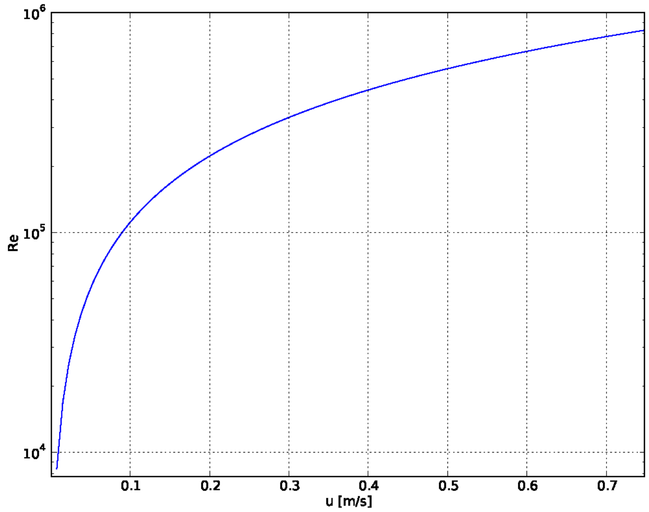

Figure 2: Reynolds Number as a function of glide speed u

2.4.2 Induced Drag

From [3, pg. 7-2] we find the induced drag coefficient, or drag due to lift C_{Di}

$$
\begin{aligned}
C_{Di} &= \frac{C_L^2}{AR\pi} && (22)\\
&= \frac{C_{Lh}^2}{AR_h\pi} + \frac{C_{Lw}^2}{AR_w\pi} && (23)\\
&= \frac{(1.172\alpha)^2}{0.147\pi} + \frac{(3.41\alpha)^2}{10.573\pi} && (24)\\
&= \frac{dC_{Di}}{\alpha^2}\alpha^2 && (25)\\
&= 3.331\alpha^2 && (26)
\end{aligned}
$$

where the aspect ratio of the hull $AR_h = D/L$

2.5 Putting it all together

$$
\begin{aligned}
C_L &= C_{Lh} + C_{Lw} && (27)\\
&= (\frac{dC_{Lh}}{d\alpha} + \frac{dC_{Lw}}{d\alpha})\alpha && (28)\\
&= (1.172 + 3.408)\alpha && (29)\\
&= 4.580\alpha && (30)
\end{aligned}
$$

$$
\begin{aligned}
C_D &= C_{Do} + C_{Di} && (31)\\
&= C_{Do} + \frac{dC_{Di}}{d\alpha^2}\alpha^2 && (32)\\
&= C_{Do}(R_e) + 3.33\alpha^2 && (33)
\end{aligned}
$$

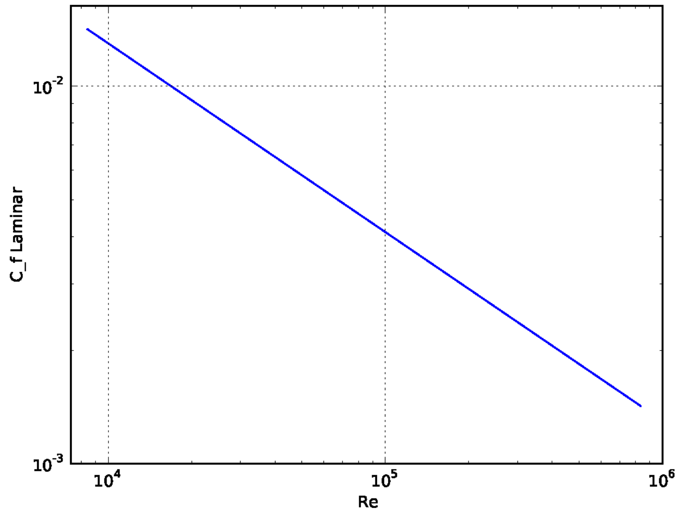

Figure 3: Laminar Skin-Friction coefficient C_F as a function of Reynolds Number. From [3, pg. 6-16]

Referring back to eq. 6:

$$C_D \cos\zeta = -C_L \sin\zeta \tag{34}$$

$$(C_{Do} + \frac{dC_{Di}}{d\alpha^2}\alpha^2)\cos\zeta = -(\frac{dC_{Lh}}{d\alpha} + \frac{dC_{Lw}}{d\alpha})\alpha)\sin\zeta \tag{35}$$

$$(C_{Do}(R_e) + 3.33\alpha^2)\cos(\theta+\alpha) = -(4.580)\alpha)\sin(\theta+\alpha) \tag{36}$$

For given pitch angle θ and glide speed u (which allows us to solve for R_e and therefore C_{Do}, we can solve for angle of attack α. It gets a bit messy, so as suggested by [5], we solve using an iterative approach. Figure 4 shows calculated angle of attack for a range of pitch and vehicle depth rates, for which vehicle vertical depth rate $vs = u\sin(\theta+\alpha)$.

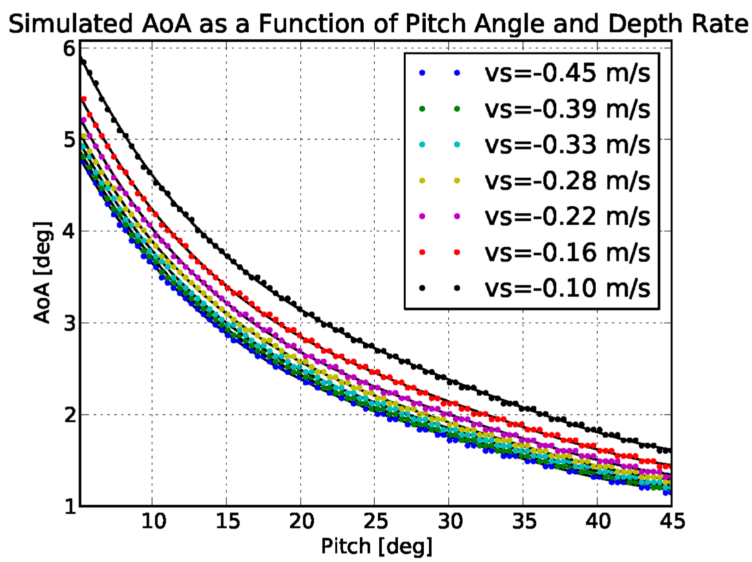

Figure 4: Angle of attack α as a function of pitch angle θ for several vehicle depth rates vs.

2.6 Universal Equation for Angle of Attack as a Function of Pitch and Depth Rate

In order to minimize computational cost of the angle of attack, we can characterize the angle of attack as a function of a given pitch angle and depth rate. The following equations were found to minimize the error between angle of attack computed using eq. 34 and angle of attack computed using the universal equation. For climbing ($\theta > 0$ and vertical speed $vs < 0$):

$$\alpha = 0.48330\theta^4 - 1.10730\theta^3 + 0.98334\theta^2 - 0.45210\theta + 0.1246 \quad (37)$$

$$- 0.00873 + 0.045 * 10^{-0.750} \frac{-0.220}{vs} \quad (38)$$

and for diving ($\theta < 0$ and vertical speed $vs > 0$): :

$$\alpha = - 0.48330\theta^4 - 1.10730\theta^3 - 0.98334\theta^2 - 0.45210\theta - 0.1246 \quad (39)$$

$$- 0.00873 + 0.045 * 10^{-0.750} \frac{-0.220}{vs} \quad (40)$$

Figure 5 shows how angle of attack solved using these two equations match up to simulated angle of attack.

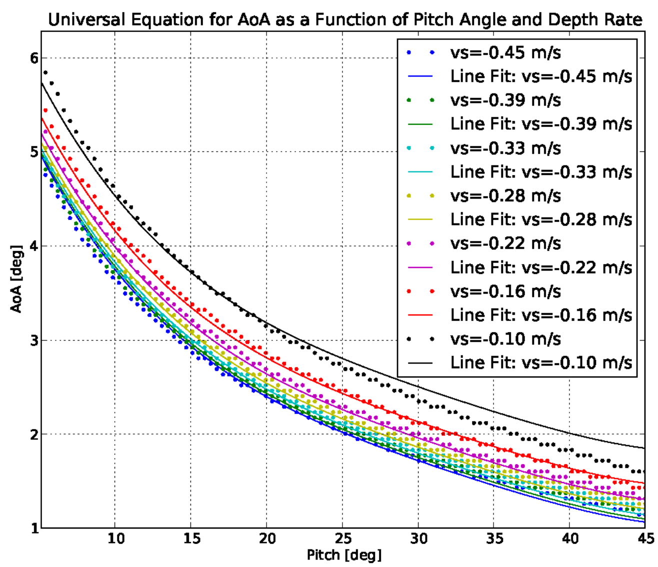

Figure 5: Fit line to AoA

3 Experimental Performance

TBW

References

[1] J. N. Newman, *Marine Hydrodynamics* MIT Press, August 1977.

[2] Sighard F. Hoerner, *Fluid Dynamic Lift*. Bakersfield, CA, 2nd Edition, 1985.

[3] Sighard F. Hoerner, *Fluid Dynamic Drag*. Bakersfield, CA, 2nd Edition, 1965.

[4] L. Merckelbach, D. Smeed, and G. Griffiths, *Vertical Water Velocities from Underwater Gliders*. Journal of Atmospheric and Oceanic Technology, 2009.

[5] L. Merckelbach, *Improved algorithm for deadreckoning for Slocum gliders*. January 29, 2010.

I Open Source License Acknowledgments

Teledyne Webb Research utilizes third-party software from various sources. Portions of this software are copyrighted by their respective owners as indicated in the copyright notices below.

This appendix provides regulatory open source licensing information for shoreside and onboard applications.

This appendix lists the licenses for software contained in Slocum Glider control software.

I.1 FreeRTOS

MIT License

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Reference: https://www.freertos.org/Documentation/02-Kernel/01-About-the-FreeRTOS-kernel/04-Licensing

I.2 Filesystem

Proprietary license.

Licensed from the manufacturer for use in Slocum Glider Products.

I.3 LZ4 Compression

LZ4 — Fast LZ compression algorithm

You can contact the author at:

- LZ4 homepage: https://lz4.org/
- LZ4 source repository: https://github.com/lz4/lz4

I.4 Backtrace

Reference: https://www.mozilla.org/en-US/MPL/2.0/

I.4.1 Mozilla Public License Version 2.0

1. **Definitions**
    1.1. **"Contributor"** means each individual or legal entity that creates, contributes to the creation of, or owns Covered Software.

    1.2. **"Contributor Version"** means the combination of the Contributions of others (if any) used by a Contributor and that particular Contributor's Contribution.

    1.3. **"Contribution"** means Covered Software of a particular Contributor.

    1.4. **"Covered Software"** means Source Code Form to which the initial Contributor has attached the notice in Exhibit A, the Executable Form of such Source Code Form, and Modifications of such Source Code Form, in each case including portions thereof.

1.5. **“Incompatible With Secondary Licenses”** means

a. that the initial Contributor has attached the notice described in Exhibit B to the Covered Software; or

b. that the Covered Software was made available under the terms of version 1.1 or earlier of the License, but not also under the terms of a Secondary License.

1.6. **“Executable Form”** means any form of the work other than Source Code Form.

1.7. **“Larger Work”** means a work that combines Covered Software with other material, in a separate file or files, that is not Covered Software.

1.8. **“License”** means this document.

1.9. **“Licensable”** means having the right to grant, to the maximum extent possible, whether at the time of the initial grant or subsequently, any and all of the rights conveyed by this License.

1.10. **“Modifications”** means any of the following:

a. any file in Source Code Form that results from an addition to, deletion from, or modification of the contents of Covered Software; or

b. any new file in Source Code Form that contains any Covered Software.

1.11. **“Patent Claims” of a Contributor** means any patent claim(s), including without limitation, method, process, and apparatus claims, in any patent Licensable by such Contributor that would be infringed, but for the grant of the License, by the making, using, selling, offering for sale, having made, import, or transfer of either its Contributions or its Contributor Version.

1.12. **“Secondary License”** means either the GNU General Public License, Version 2.0, the GNU Lesser General Public License, Version 2.1, the GNU Affero General Public License, Version 3.0, or any later versions of those licenses.

1.13. **“Source Code Form”** means the form of the work preferred for making modifications.

1.14. **“You” (or “Your”)** means an individual or a legal entity exercising rights under this License. For legal entities, “You” includes any entity that controls, is controlled by, or is under common control with You. For purposes of this definition, “control” means **(a)** the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or **(b)** ownership of more than fifty percent (50%) of the outstanding shares or beneficial ownership of such entity.

2. **License Grants and Conditions**

2.1. **Grants**

Each Contributor hereby grants You a world-wide, royalty-free, non-exclusive license:

a. under intellectual property rights (other than patent or trademark) Licensable by such Contributor to use, reproduce, make available, modify, display, perform, distribute, and otherwise exploit its Contributions, either on an unmodified basis, with Modifications, or as part of a Larger Work; and

b. under Patent Claims of such Contributor to make, use, sell, offer for sale, have made, import, and otherwise transfer either its Contributions or its Contributor Version.

2.2. **Effective Date**

The licenses granted in Section 2.1 with respect to any Contribution effective for each Contribution on the date the Contributor first such Contribution.

2.3. **Limitations on Grant Scope**

The licenses granted in this Section 2 are the only rights granted under this License. No additional rights or licenses will be implied from the distribution or licensing of Covered Software under this License. Notwithstanding Section 2.1(b) above, no patent license is granted by a Contributor:

a. for any code that a Contributor has removed from Covered Software; or

b. for infringements caused by: (i) Your and any other third party's modifications of Covered Software, or (ii) the combination of its Contributions with other software (except as part of its Contributor Version); or

c. under Patent Claims infringed by Covered Software in the absence of its Contributions.

This License does not grant any rights in the trademarks, service marks, or logos of any Contributor (except as may be necessary to comply with the notice requirements in Section 3.4).

2.4. **Subsequent Licenses**

No Contributor makes additional grants as a result of Your choice to distribute the Covered Software under a subsequent version of this License (see Section 10.2) or under the terms of a Secondary License (if permitted under the terms of Section 3.3).

2.5. **Representation**

Each Contributor represents that the Contributor believes its Contributions are its original creation(s) or it has sufficient rights to grant the rights to its Contributions conveyed by this License.

2.6. **Fair Use**

This License is not intended to limit any rights You have under applicable copyright doctrines of fair use, fair dealing, or other equivalents.

2.7. **Conditions**

Sections 3.1, 3.2, 3.3, and 3.4 are conditions of the licenses granted in Section 2.1.

3. **Responsibilities**

3.1. **Distribution of Source Form**

All distribution of Covered Software in Source Code Form, including any Modifications that You create or to which You contribute, must be under the terms of this License. You must inform recipients that the Source Code Form of the Covered Software is governed by the terms of this License, and how they can obtain a copy of this License. You may not attempt to alter or restrict the recipients' rights in the Source Code Form.

3.2. **Distribution of Executable Form**

If You distribute Covered Software in Executable Form then:

a. such Covered Software must also be made available in Source Code Form, as described in Section 3.1, and You must inform recipients of the Executable Form how they can obtain a copy of such Source Code Form by reasonable means in a timely manner, at a charge no more than the cost of distribution to the recipient; and

b. You may distribute such Executable Form under the terms of this License, or sublicense it under different terms, provided that the license for the Executable Form does not attempt to limit or alter the recipients' rights in the Source Code Form under this License.

3.3. **Distribution of a Larger Work**

You may create and distribute a Larger Work under terms of Your choice, provided that You also comply with the requirements of this License for the Covered Software. If the Larger Work is a combination of Covered Software with a work governed by one or more Secondary Licenses, and the Covered Software is not Incompatible With Secondary Licenses, this License permits You to additionally distribute such Covered Software under the terms of such Secondary License(s), so that the recipient of the Larger Work may, at their option, further distribute the Covered Software under the terms of either this License or such Secondary License(s).

3.4. **Notices**

You may not remove or alter the substance of any license notices (including copyright notices, patent notices, disclaimers of warranty, or limitations of liability) contained within the Source Code Form of the Covered Software, except that You may alter any license notices to the extent required to remedy known factual inaccuracies.

3.5. **Application of Additional Terms**

You may choose to offer, and to charge a fee for, warranty, support, indemnity or liability obligations to one or more recipients of Covered Software. However, You may do so only on Your own behalf, and not on behalf of any Contributor. You must make it absolutely clear that any such warranty, support, indemnity, or liability obligation is offered by You alone, and You hereby agree to indemnify every Contributor for any liability incurred by such Contributor as a result of warranty, support, indemnity or liability terms You offer. You may include additional disclaimers of warranty and limitations of liability specific to any jurisdiction.

4. **Inability to Comply Due to Statute or Regulation**

If it is impossible for You to comply with any of the terms of this License with respect to some or all of the Covered Software due to statute, judicial order, or regulation then You must: (a) comply with the terms of this License to the maximum extent possible; and (b) describe the limitations and the code they affect. Such description must be placed in a text file included with all distributions of the Covered Software under this License. Except to the extent prohibited by statute or regulation, such description must be sufficiently detailed for a recipient of ordinary skill to be able to understand it.

5. **Termination**

5.1. The rights granted under this License will terminate automatically if You fail to comply with any of its terms. However, if You become compliant, then the rights granted under this License from a particular Contributor are reinstated **(a)** provisionally, unless and until such Contributor explicitly and finally terminates Your grants, and **(b)** on an ongoing basis, if such Contributor fails to notify You of the non-compliance by some reasonable means prior to 60 days after You have come back into compliance. Moreover, Your grants from a particular Contributor are reinstated on an ongoing basis if such Contributor notifies You of the non-compliance by some reasonable means, this is the first time You have received notice of non-compliance with this License from such Contributor, and You become compliant prior to 30 days after Your receipt of the notice.

5.2. If You initiate litigation against any entity by asserting a patent infringement claim (excluding declaratory judgment actions, counter-claims, and cross-claims) alleging that a Contributor Version directly or indirectly infringes any patent, then the rights granted to You by any and all Contributors for the Covered Software under Section 2.1 of this License shall terminate.

5.3. In the event of termination under Sections 5.1 or 5.2 above, all end user license agreements (excluding distributors and resellers) which have been validly granted by You or Your distributors under this License prior to termination shall survive termination.

6. **Disclaimer of Warranty**

Covered Software is provided under this License on an “as is” basis, without warranty of any kind, either expressed, implied, or statutory, including, without limitation, warranties that the Covered Software is free of defects, merchantable, fit for a particular purpose or non-infringing. The entire risk as to the quality and performance of the Covered Software is with You. Should any Covered Software prove defective in any respect, You (not any Contributor) assume the cost of any necessary servicing, repair, or correction. This disclaimer of warranty constitutes an essential part of this License. No use of any Covered Software is authorized under this License except under this disclaimer.

7. **Limitation of Liability**

Under no circumstances and under no legal theory, whether tort (including negligence), contract, or otherwise, shall any Contributor, or anyone who distributes Covered Software as permitted above, be liable to You for any direct, indirect, special, incidental, or consequential damages of any character including, without limitation, damages for lost profits, loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses, even if such party shall have been informed of the possibility of such damages. This limitation of liability shall not apply to liability for death or personal injury resulting from such party's negligence to the extent applicable law prohibits such limitation. Some jurisdictions do not allow the exclusion or limitation of incidental or consequential damages, so this exclusion and limitation may not apply to You.

8. **Litigation**

Any litigation relating to this License may be brought only in the courts of a jurisdiction where the defendant maintains its principal place of business and such litigation shall be governed by laws of that jurisdiction, without reference to its conflict-of-law provisions. Nothing in this Section shall prevent a party's ability to bring cross-claims or counter-claims.

9. **Miscellaneous**

This License represents the complete agreement concerning the subject matter hereof. If any provision of this License is held to be unenforceable, such provision shall be reformed only to the extent necessary to make it enforceable. Any law or regulation which provides that the language of a contract shall be construed against the drafter shall not be used to construe this License against a Contributor.

10. **Versions of the License**

10.1. **New Versions**

Mozilla Foundation is the license steward. Except as provided in Section 10.3, no one other than the license steward has the right to modify or publish new versions of this License. Each version will be given a distinguishing version number.

10.2. **Effect of New Versions**

You may distribute the Covered Software under the terms of the version of the License under which You originally received the Covered Software, or under the terms of any subsequent version published by the license steward.

10.3. **Modified Versions**

If you create software not governed by this License, and you want to create a new license for such software, you may create and use a modified version of this License if you rename the license and remove any references to the name of the license steward (except to note that such modified license differs from this License).

10.4. **Distributing Source Code Form that is Incompatible With Secondary Licenses**

If You choose to distribute Source Code Form that is Incompatible With Secondary Licenses under the terms of this version of the License, the notice described in Exhibit B of this License must be attached.

I.4.1.1 Exhibit A - Source Code Form License Notice

This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at https://www.mozilla.org/en-US/MPL/2.0/ .

If it is not possible or desirable to put the notice in a particular file, then You may include the notice in a location (such as a LICENSE file in a relevant directory) where a recipient would be likely to look for such a notice.

You may add additional accurate notices of copyright ownership.

I.4.1.2 Exhibit B - “Incompatible With Secondary Licenses” Notice

This Source Code Form is “Incompatible With Secondary Licenses”, as defined by the Mozilla Public License, v. 2.0.

I.4.2 STM32 HAL

Apache License

Version 2.0, January 2004

Reference: https://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. **Definitions**.

“**License**” shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

“**Licensor**” shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

“**Legal Entity**” shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, “**control**” means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

“**You**” **(or** “**Your**”**)** shall mean an individual or Legal Entity exercising permissions granted by this License.

“**Source**” form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

“**Object**” form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

“**Work**” shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"**Derivative Works**" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"**Contribution**" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "**submitted**" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "**Not a Contribution.**"

"**Contributor**" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. **Grant of Copyright License**. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. **Grant of Patent License**. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
4. **Redistribution**. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:
    a. You must give any other recipients of the Work or Derivative Works a copy of this License; and
    b. You must cause any modified files to carry prominent notices stating that You changed the files; and
    c. You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
    d. If the Work includes a "**NOTICE**" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may

add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. **Submission of Contributions**. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.
6. **Trademarks**. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.
7. **Disclaimer of Warranty**. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
8. **Limitation of Liability**. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. **Accepting Warranty or Additional Liability**. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

(This page intentionally left blank.)