# Parameter Reference Manual

SCHOOL OF OCEANOGRAPHY
and
APPLIED PHYSICS LABORATORY
UNIVERSITY OF WASHINGTON
Version 66.12 November 2014

# Chapter 1
# Conventions and Introduction

## 1.1 Conventions

All parameters have a leading **$** in their name. Parameters will be given in bold font, file names in italic. Example (not default) values follow each parameter's description in square brackets.

## 1.2 Introduction

This manual provides an alphabetical reference to the parameters that govern the operation of the Seaglider. All parameters described in this manual are changeable by the pilot via the command file, using the **name,value** (example: **$SM_CC,475**) convention. Clicking a parameter name jumps to that parameter's description. The "Back" button returns to the previous location in the document.

Values reported by the Seaglider in the log file associated with a dive include all the pilot-changeable parameters described in this document. The log file also contains other values that are given parameter-like names for consistency of parsing during post-dive data processing.

# Chapter 2
# Seaglider Command File (*cmdfile*) Directives

## 2.1 Introduction

Command file (*cmdfile*) directives control the fundamental state of autonomous Seaglider operations. The two fundamental states of autonomous Seaglider operation are diving and recovery. The diving state is the normal repeating sequence of canonical dives. The recovery state is when the Seaglider is held at the surface, calling to the basestation at pilot-specified intervals with an updated position and receiving instructions.

Command file (*cmdfile*) directives are given as the last (and perhaps only) line of the command file, as stored on the basestation and transferred to the Seaglider during its communication session.

## 2.2 Directives

**$GO**
This command will cause the glider to continue in its current mode of operation. If in an autonomous run, doing repeated dives, it will continue to dive according to its current set of parameters. If a **$GO** command is received while the Seaglider is in the recovery phase, the Seaglider will stay in the recovery phase.

**$RESUME**
This command will cause the Seaglider to resume diving from within the recovery phase, using its current set of parameters. It acts just like a **$GO** directive if the Seaglider is diving.

**$QUIT**
This command will cause the Seaglider to go immediately to the recovery phase. The Seaglider will hold at the surface, sleeping **$T_RSLEEP** minutes between the end of one communications session and the start of the next.

There are about two minutes of communications overhead associated with each session, so that sessions are about (**$T_RSLEEP** + 2) minutes apart. Replacing the **$QUIT** directive with a **$RESUME** directive will cause the Seaglider to initiate a new dive with the existing set of parameters.

The state table below outlines the effect of each directive on the glider in each of the two autonomous run states: diving and recovery.

| *State* *Directive* | **$GO** | **$RESUME** | **$QUIT** |
|---|---|---|---|
| Diving | Diving | Diving | Recovery |
| Recovery | Recovery | Diving | Recovery |

# Chapter 3
# Seaglider Parameters

Example (not default) values follow each parameter's description in square brackets.

**$AD7714Ch0Gain**
The gain assigned to the pressure sensor channel on the AD7714 analog-to-digital converter. The parameter takes two values: 128 for normal Seaglider operations with the installed Paine pressure sensor, and 1 for bench testing where a synthetic voltage is injected in place of the pressure sensor output to simulate diving. The advent of the parameters **$SIM_W** and **$SIM_PITCH** have obviated this parameter. [128]

**$AH0_10V**
The capacity of the 10V (low-voltage) primary battery pack (AmpHr). There is some safety factor in this number, but its accuracy has been verified in post-recovery depletion testing of Seaglider battery packs. The Seaglider goes into the recovery state if the total 10V battery pack amp-hours used on a mission equals or exceeds this value. [61.2]

**$AH0_24V**
The capacity of the 24V (high-voltage) battery pack (AmpHr). There is some safety factor in this number, but its accuracy has been verified in post-recovery depletion testing of Seaglider battery packs. The Seaglider goes into the recovery state if the total 24V battery pack amp-hours used on a mission equals or exceeds this value. [91.8]

**$ALTIM_BOTTOM_PING_RANGE**
The range (in meters) from the presumed apogee depth (the nominal depth at which the glider will begin its apogee maneuver) to ping for the bottom. With this parameter set the altimeter will only ping a maximum of once per dive. A value of 0 disables pinging. If the configured transponder device does not have a ping function, these parameters are ignored. See also **$ALTIM_PING_DEPTH**. [0]

**$ALTIM_BOTTOM_TURN_MARGIN**
The distance (in meters) from the altimeter-detected seafloor (or an obstacle) at which to initiate the apogee maneuver (bottom turn). A value of 0 disables use of the altimeter to determine the start of the apogee maneuver. [0]

**$ALTIM_FREQUENCY**
The frequency (kHz) to use for altimeter pings. Must be an integer between 10 and 25. [15]

**$ALTIM_PING_DELTA**
If the altimeter does not get a successful return and confirmation ping return at **$ALTIM_PING_DEPTH**, it will continue to issue pings at depth intervals of **$ALTIM_PING_DELTA** meters. See **$ALTIM_PING_DEPTH**.

**$ALTIM_PING_DEPTH**
The depth of the first altimeter ping (meters), if non-zero. If the altimeter gets a return, and a return to an immediate second confirmation ping, it will set the bottom depth equal to the current depth plus the altimeter range to the bottom. The apogee maneuver will be initiated at **$ALTIM_BOTTOM_TURN_MARGIN** meters above the bottom. If **$ALTIM_PING_DEPTH** is non-zero, the altimeter timeout will be set so that the maximum range is the larger of 0.75***$ALTIM_PING_DEPTH** and 1.2***$ALTIM_TOP_PING_RANGE** , if set. The first test is meant to exclude surface returns. Note that **$ALTIM_PING_DEPTH** and the **$ALTIM_BOTTOM_PING_RANGE** modes are

mutually exclusive. If **$ALTIM_BOTTOM_PING_RANGE** is set, it will be honored to the exclusion of **$ALTIM_PING_DEPTH**.

**$ALTIM_PULSE**
Pulse width (ms) of altimeter pings. Must be an integer between 1 and 9. [5]

**$ALTIM_SENSITIVITY**
Sensitivity of the envelope detector on the altimeter. Must be an integer between 0 and 5. A value of 0 disables the envelope detector, causing the altimeter to trigger on any return at the right frequency. Values between 1 and 5 require that the return signal be above that voltage for the duration of a pulse width before triggering. [1]

**$ALTIM_TOP_MIN_OBSTACLE**
The minimum obstacle depth (m) to honor in initiating a subsurface finish. [2]

**$ALTIM_TOP_PING_RANGE**
The range (in meters) from the surface (this is simply depth) at which to ping the altimeter. A value of 0 disables a ping. If the configured transponder device does not have a ping function these parameters are ignored. [0]

**$ALTIM_TOP_TURN_MARGIN**
The distance (in meters) from an altimeter-detected obstacle at which to initiate the sub-surface finish. A value of 0 disables use of the altimeter to determine the start of the sub-surface finish. [0]

**$APOGEE_PITCH**
The intermediate pitch (position of the pitch mass) the glider will pitch to between the dive and climb phases, to prevent stalling. The glider assumes **$APOGEE_PITCH** when it observes a depth of greater than the apogee depth (whether by altimeter, **$D_TGT** or **$D_GRID**). The apogee maneuver also includes pumping the VBD to 0cc. After the apogee maneuver, the climb is initiated by changing both pitch and VBD to the opposite-signed values from the dive. [-5]

**$C_PITCH**
The center (neutral, or flat) position (A/D counts) for pitch. [2590]

**$C_ROLL_CLIMB**
The center (neutral, or straight flight) position (A/D counts) for roll during the climb (positive pitch control) phase. Note that the two roll centers will probably be different. This was implemented to correct for roll biases induced by physical asymmetries in the glider. [2150]

**$C_ROLL_DIVE**
The center (neutral, or straight flight) position (A/D counts) for roll during the dive (negative pitch control) phase. [2150]

**$C_VBD**
The center (neutrally buoyant at a specified density) position (A/D counts) for VBD. [2735]

**$CALL_NDIVES**
The number of profiles (dive/climb cycles) to perform before attempting communications. The Seaglider will surface normally following each dive, and GPS fixes 1 and 2 will be obtained at each surfacing, independent of the value of **$CALL_NDIVES**. [1]

**$CALL_TRIES**
The maximum number of phone calls to attempt during a surfacing between dives. [5]

**$CALL_WAIT**
The wait time (seconds) between call attempts during a communication session. This is time to allow the Iridium satellite geometry to change and perhaps improve the connection. [60]

**$CAPMAXSIZE**
The maximum size (in bytes, prior to compression) of the capture file to upload. If the capture file is to be uploaded (either due to critical output, **$CAPUPLOAD** set to 1, or a completed self-test) and the capture file size exceeds this value, the glider will create a new capture file that is of the size requested, per the following strategy.

If there are no critical lines of output, then the first **$CAPMAXSIZE** bytes will be sent. If there are critical lines of output, then the new capture file will consist of the first 20 of the critical output lines, with a window of output lines surrounding each critical line.

**$CAPUPLOAD**
A Boolean value that determines if the capture file from the current dive should be uploaded or not: 0 means do not upload the capture file, 1 means upload the capture file. If critical output is captured during a dive, or a self-test has been completed, then the capture file is uploaded regardless of the value of **$CAPUPLOAD**. [0]

**$CF8_MAXERRORS**
The maximum number of CF8 errors allowed before the Seaglider goes into recovery state. A CF8 error is counted against the **$CF8_MAXERRORS** limit when a CF8 open or write call continues to fail (returns an error code) after three retries.

**$COMM_SEQ**
The specification of the sequence of file transfer to use. A value of zero indicates the standard communication file transfer sequence: command, targets, science, current dive log file, current dive data file, earlier un-transferred log and data files, pdoscmds.bat, the results of the pdoscmds.bat commands, and any other files as commanded in pdoscmds.bat. A value of one indicates skipping the normal log and data file transmission step, and going directly to pdoscmds.bat: command, targets, science, pdoscmds.bat, the results of pdoscmds.bat, including and any other files as commanded in pdoscmds.bat. This prioritizes pdoscmds.bat transfers over data files in the event that communications problems or file corruptions prevented data transfers. It is a control mode to be used only when communications or other glider problems exist. [0]

**$COMPASS_DEVICE**
Configuration flags specifying the specific model and port for the compass and transponder/altimeter devices. This integer value is equal to port_number + 16*type_number. For example, for a TCM2-50 (type 0) on general purpose port 1, **$COMPASS_DEVICE** = 1 + 16*0 = 1. The array of available models is specific to each device. The compass device must be defined (parameter cannot be -1). For transponders not connected to a serial port (e.g., Benthos ENT-380) the null port (generally index 0) can be specified. See **$DEVICE**[**1/2/3/4/ 5/6**]. [1]

**$COMPASS_USE**
This parameter was used in development to assess compass problems. It allows for faking inputs and outputs in compass calibration and reporting. For normal operations, it should be set to 0, which yields current behavior of trusting inputs from the compass and using and reporting fully calibrated values.

- Bit 0 Heading source:
  - 0 = cal'd mag field w/PR corrections
  - 1 = direct from compass (effectively ignores PR bits)
- Bit 1 Autocal mode (experimental):
  - 0 = no autocalibration
  - 1 = leave compass on throughout flight and run autocalibration
- Bit 2 Offboard cal mode (experimental):
  - 0 = do not report mag values in data file
  - 1 = report mag values in data file for use in offboard cal
- Bits 3-4 pitch source for heading correction:
  - 0 = calibrated from sensor inputs per normal routine
  - 1 = use value direct from compass
  - 2 = calculate based on pitch mass position and gain
- Bits 5-6 roll source for heading correction
  - 0 = calibrated from sensor inputs per normal routine
  - 1 = use value direct from compass
  - 2 = calculate based on pitch mass position and gain
- Bits 7-8 pitch to report in data stream
  - 0 = calibrated from sensor inputs per normal routine
  - 1 = use value direct from compass
  - 2 = calculate based on pitch mass position and gain
- Bits 9-10 roll to report in data stream
  - 0 = calibrated from sensor inputs per normal routine

    - 1 = use value direct from compass
    - 2 = calculate based on pitch mass position and gain
- Bit 11 which compass device to use
    - 0 = COMPASS_DEVICE
    - 1 = COMPASS2_DEVICE

A value of 40 (0x28) will calculate a pitch and roll from control positions for use in the heading calibration but will report calibrated pitch and roll in the data stream. A value of 680 (40 + 640) will use the control position calculated attitudes for reporting as well. [0]

**$COMPASS2_DEVICE**
Same as **$COMPASS_DEVICE** for an optional second compass (used in glider development).

**$COURSE_BIAS**
A heading bias (degrees) to compensate for a glider's observed tendency to veer to one side. This value is subtracted from the desired heading to produce the target heading. [0]

**$D_ABORT**
The maximum depth (meters) for Seaglider operations. If this depth is reached, the dive is aborted and the glider immediately enters the recovery phase. [1090]

**$D_BOOST**
The depth (meters) above which only the boost pump will be run.[0]

**$D_CALL**
For non-zero values, require a depth shallower than this value before initiating the GPS and phone call portions of a surface maneuver. If this depth is not reached, a subsurface finish is executed.
[0]

**$D_FINISH**
Depth (meters) at which to consider a dive completed. Normally, this is 0, but may be a number greater than zero to specify the depth at which subsurface finish maneuvers should be started. Used only when an additional trigger to initiate a subsurface finish is present (see **$N_NOSURFACE**). [0]

**$D_FLARE**
The depth (meters) at which the glider flares to the computed desired pitch angle following the initial dive from the surface. The guidance and control (G&C) action at the start of the dive phase maintains full pitch forward as VBD bleeding takes place. A new G&C action is initiated as soon as a glider reaches **$D_FLARE**: pitch is adjusted first (the flare), then VBD is adjusted (continued bleed to the target VBD), then roll is actuated to turn the glider to the desired heading. [3]

**$D_NO_BLEED**
The depth (meters) below which the Seaglider will not bleed (move oil from the bladder into the internal reservoir, making the Seaglider less buoyant) on dives. This parameter also defines the depth at which the **$T_NO_W** parameter takes effect. [50]

**$D_OFFGRID**
The depth (meters) that the bathymetry map look-up routine returns in the event the Seaglider's position is outside an area for which the Seaglider carries a map. [1001]

**$D_PITCH**
If non-zero, depth (meters) that must be reached before surface pitch maneuver is executed. If a depth shallower than this value is not reached then execute a subsurface finish.
[0]

**$D_SAFE**
If non-zero, target depth to use when flying an escape route and limiting dive depth for VBD safety reasons. Used

on escape routes when the escape was triggered by VBD max errors exceeded or uncommanded bleed. [0]

**$D_SURF**
The depth (meters) at which the glider begins its approach to the surface. In order to collect data all the way to the surface, at **$D_SURF** the Seaglider computes how many more data samples to take, based on the observed vehicle vertical speed, depth, and the data sample interval. The number of additional points is limited to 50. The Seaglider then goes into passive G&C mode and collects that number of data points at the appropriate sample interval for the depth range. When complete, the Seaglider enters the surface phase. Note that this approach will occasionally result in the last few data samples being taken when the conductivity sensor is actually in air, giving unrealistic conductivity values. These samples can be removed in shore-side processing, if necessary.[3]

**$D_TGT**
The nominal depth (meters) at which the Seaglider begins the apogee phase, the transition from the negatively buoyant, pitch-down dive to positively buoyant, pitch-up climb. This depth is also used in conjunction with **$T_DIVE** to determine the specified vertical velocity for the dive and climb.

The actual depth of the starting point of the apogee maneuver may be determined by reading a digital bathymetric map (**$D_GRID**), or by using the altimeter. In either case, the vertical velocity specified by the combination of **$D_TGT** and **$T_DIVE** is retained by appropriate scaling of **$T_DIVE**. The apogee maneuver is not started until a depth greater than **$D_TGT** (or other depth trigger) is detected. In scenario runs, **$D_TGT** is also the depth at which the Seaglider enters the recovery phase. [990]

**$DBDW**
Overrides the hydro model calculated derivative of buoyancy with respect to w (vertical speed) used as a gain term in adjusting buoyancy to achieve the desired vertical speed. If parameter is zero then the onboard calculated value is used. Units are grams per m/s. Typical onboard calculated values are between 2000 and 4000. [0]

**$DEEPGLIDER**
Indicates whether the glider is a standard Seaglider (0), or a Deepglider (1). [0]

**$DEVICE**[**1/2/3/4/5/6**]
Configuration flags specifying device type and port for each of the six possible attached science sensors. Empty device slots are indicated with a parameter value of -1. Non-negative entries indicate that a device is attached. The encoding is specific to the version of the Seaglider software. These entries are set through the Seaglider menu system by the builder/assemblers.

**$DIVE**
The number of the next dive the Seaglider will start. [314]

**$ESCAPE_HEADING**
The base heading the Seaglider will steer in an escape recovery situation when either no position fix is available or no escape target was supplied in the targets file. [0]

**$ESCAPE_HEADING_DELTA**
The actual heading steered by the Seaglider in an escape recovery situation will always be **$ESCAPE_HEADING** +/- **$ESCAPE_HEADING_DELTA**. The sign will switch (and thus the heading will toggle) when the bottom depth (as detected by altimetry or **$T_NO_W**) shallows by 5% relative to the depth at the last toggle. [10]

**$FERRY_MAX**
The maximum correction (in degrees) to apply to the rhumb line to the active (next) waypoint when **$NAV_MODE** = 2. This is a safety limit to prevent spurious depth-averaged current calculations from giving the Seaglider a heading in the wrong direction. [45]

**$FG_AHR_10V**
Cumulative A-hr consumed from the 10V pack as tracked by the supervisor fuel gauge. This parameter is automatically updated whenever the fuel gauge is read and the supervisor's onboard accumulators are cleared. When reported in the log file it reflects the fuel gauge state at the start of the dive. This parameter is only relevant for RevC and later Seaglider motherboards. For RevB and earlier, this parameter is always zero. [0]

**$FG_AHR_24V**
Cumulative A-hr consumed from the 24V pack as tracked by the supervisor fuel gauge. This parameter is automatically updated whenever the fuel gauge is read and the supervisor's onboard accumulators are cleared. When reported in the log file it reflects the fuel gauge state at the start of the dive. This parameter is only relevant for RevC and later Seaglider motherboards. For RevB and earlier, this parameter is always zero. [0]

**$FILEMGR**
An integer parameter that specifies how aggressively to manage the onboard file system, according to the following table. [0]

0: none
1: only store compressed files
2: delete splits on failed phone call

**$FIX_MISSING_TIMEOUT**
Time in days to tolerate lack of any valid navigation fix (GPS, RAFOS, Iridium geolocation) before triggering recovery.
[0]

**$GLIDE_SLOPE**
The absolute value of the maximum glide slope (degrees) allowed for the Seaglider. The desired glide slope is calculated on board the Seaglider to best achieve the goals of the next dive. The stall angle provides the lower limit; this parameter is the upper limit. [30]

**$GPS_DEVICE**
Configuration value specifying the model of the attached device. These devices have dedicated hardware ports on all motherboard revisions and as such a port specification is not necessary. A GPS device must be defined (cannot be -1). [32]

**$HD_A**
The hydrodynamic parameter representing the lift coefficient, determined empirically and used in the glider's on-board performance prediction and guidance calculations. [0.0029217]

**$HD_B**
The hydrodynamic parameter representing the drag coefficient, determined empirically and used in the glider's on-board performance prediction and guidance calculations.[0.010758]

**$HD_C**
The hydrodynamic parameter representing the induced drag coefficient, determined empirically and used in the glider's on-board performance prediction and guidance calculations. [2.8991001e-06]

**$HEAD_ERRBAND**
Deadband for heading (degrees). This value is used to determine if a correction to heading is required during an active G&C mode. If the absolute value of the difference between the actual heading and the desired heading is less than or equal to **$HEAD_ERRBAND**, no heading correction is made. If the difference is greater than **$HEAD_ERRBAND**, then a turn is performed until the desired heading is passed, or until the amount of time **$T_TURN** has elapsed. [10]

In a scenario run, **$HEAD_ERRBAND** is the deadband for roll angle. If the absolute value of the specified position of the mass shifter (in degrees) is less than or equal to **$HEAD_ERRBAND**, no roll is performed. [10]

**$HEADING**
A floating point value between -1.0 and 360.0 (true degrees, 0.0 and 360.0 are equivalent values) used in conjunction with certain navigation modes (controlled by the **$NAV_MODE** parameter) to determine the course steered by the Seaglider.

-If **$NAV_MODE** is 0, 1, or 2 and the value of **$HEADING** is between 0.0 and 360.0, the glider will use this value to synthesize a waypoint 20 km distant on the specified bearing from the current location.

-If **$NAV_MODE** is 3, **$HEADING** is added to the depth-averaged current calculated for the previous dive, to give a Seaglider heading that is the specified amount to the right of the current. [-1]

**$HEAPDBG**
A boolean value that toggles extended heap debugging. When set to 1, the glider maintained heap is checked for integrity before each memory allocation and free operation. 0 disables this checking. Note: due to the extensive output, it is not recommended that this parameter be used in conjunction with a DEBUG level of output on the SGLMALLOC service during field operations. [0]

**$ICE_FREEZE_MARGIN**
Temperature margin (degrees C) to apply to the freezing point calculation, weighted by ice condition, for surfacing decisions. For an in situ freezing point Tf and temperature T, the glider will surface if T > Tf + fm*(ic - 1), where fm is the freeze margin and ic is the ice condition. [0.3]

**$ID**
Seaglider identification (serial) number. This is a number between 1 and 999. Leading zeros are not required. This number is used in many ways, including creating the Seaglider's login on the basestation, in file naming conventions, and as a serial number for manufacturing purposes. [1]

**$INT_PRESSURE_SLOPE**
The slope (psia per A/D count) calibration of the internal pressure sensor. The sensor has a span of 0 to 30 psia, with a 90mV output at full-scale at 12V excitation. The output is proportional to supply. The Seaglider excitation is 4.096V and the gain is 100, so 30 psia = 90*4.096/12*100*1. Full-scale = 90*4.096*gain*counts/mVolt, so the nominal slope is 0.009766 psia per A/D count. [0.0097660]

**$INT_PRESSURE_YINT**
The y-intercept of the linear calibration of the internal pressure sensor. [0]

**$KALMAN_USE**
The control parameter for the run state of the Kalman filter navigation program. The **$NAV_MODE** parameter controls whether the Kalman filter output heading is used to control the Seaglider. This separation of functions allows the Kalman filter to be run, but not used, while it "learns" the currents. Bits 0 and 1 together indicate the command mode of the filter and control its operation as described below. [1]

0: Reset the Kalman filter. Set Kalman state vector and origin of local Kalman coordinate system to 0 and restart the filter.
1: Run the Kalman filter.
2: Do not run the Kalman filter.

If the command mode of **$KALMAN_USE** are 0 or 1, and **$NAV_MODE** is 1, the Kalman filter results will be used to determine the Seaglider's heading.

If **$NAV_MODE** is 1, but the command mode of **$KALMAN_USE** is 2 (not being run), the Seaglider acts as though **$NAV_MODE** is 0.

Bits 2 and 3 of **$KALMAN_USE** control whether to employ (0) or disable (1) the dirunal and semidiurnal components of the model, respectively. The mean component is always determined. For example, a **$KALMAN_USE** of 9 (= 8 + 1) runs the Kalman filter but disables the semidiurnal component of the model. A **$KALMAN_USE** value of 1 runs the Kalman filter with all components enabled.

**$KERMIT**
Controls the use of kermit (vs xmodem) for file transfers. This parameter was experimental, and has never been used operationally. The value controls what files kermit is used for: 0 =none, 1= data only, 2 =data only, batch mode transfer, 3= data and control, 4= data and control - data via batch mode.

**$LOGGERS**
Which logger devices to enable or disable. [0]

**$LOGGERDEVICE[1/2/3/4]**
Configuration flags specifying logger device type and port for each of the found possible attached logger devices. Empty device slots are indicated with a parameter value of -1. Non-negative entries indicate that a device is attached. The encoding is specific to the version of the Seaglider software. These entries are set through the Seaglider menu system by the builder/assemblers.

**$MASS**
The mass of the glider in grams. Used in on-board buoyancy and current estimation calculations. [52000]

**$MASS_COMP**
The mass of any compressee in grams. Used in on-board buoyancy and current estimation calculations. [0]

**$MAX_BUOY**
The absolute value of the maximum negative buoyant force (grams) to which the glider is allowed to bleed during the dive phase. There is no restriction on positive buoyancy during the climb phase. [150]

**$MAXI_10V**
The maximum allowable observed current draw in amperes from the 10 volt battery pack before the glider will stop diving and go into recovery. [0.6]

**MAXI_24V**
The maximum allowable observed current draw in amperes from the 24 volt battery pack before the glider will stop diving and go into recovery. [0.8]

**$MINV_10V**
The minimum allowable observed voltage on the 10 volt battery pack before the glider will stop diving and go into recovery. A value of 0 disables the check. [8]

**$MINV_24V**
The minimum allowable observed voltage on the 24 volt battery pack before the glider will stop diving and go into recovery. A value of 0 disables the check.[19]

**$MISSION**
The current Seaglider mission number. This value is intended to be unique for each of a particular Seaglider's various deployments. It has no effect on glider operations, but is reported back in data files for the purpose of data file distinction from other missions a glider may have performed. [0]

**$MOTHERBOARD**
An integer value indicating the motherboard revision carried by the glider and set by the builder or assembler. 1 = NOPP, 2 = A.1, 3 = AB, 4 = B.1, etc. [4]

**$NAV_MODE**
An integer value specifying the method used to choose a heading for the Seaglider to maintain on the next dive, according to the following table [1].

0: Steer constant heading (**$HEADING**)
1: Kalman filter (see **$KALMAN_USE**)
2: Ferry angle correction with respect to calculated depth-averaged current
3: Steer relative to depth-averaged current

Navigation flow proceeds as follows.

1. If we have calculated a valid depth-averaged-current (DAC) and **$NAV_MODE** = 3 (steer relative to DAC), synthesize a heading = currentdirection + **$HEADING**.

2. If a synthesized heading exists from (1) or **$HEADING** >= 0, synthesize a target 20 km away at a true bearing given by that heading. Otherwise,choose a target based on the usual selection process. Calculate range and bearing to the target.

3. Modify dive time and depth as needed to honor safeties and grids. Calculate speed limits - Seaglider minimum and maximum possible horizontalspeeds through the water. The speed limits and dive times fix the range of possibilities for the Seaglider's distance through the water.

4. Choose a glide slope that gets the Seaglider closest to the target (maximum speed if the target is distant, minimum speed if the target is too close).

5. At this point the Seaglider has a bearing and glide slope in hand. If **$NAV_MODE** = 0 or **$NAV_MODE** = 3, then these will be the heading and slope for this dive.

6. If **$NAV_MODE** = 2 (calculate set/ferry angle correction) and we have calculated a valid DAC, we apply **$SPEED_FACTOR** to the speed limits to account for the fact that the Seaglider actually won't achieve its ideal speed over the entire dive. We then iteratively calculate set corrections as a function of our horizontal speed through the water so that we optimize the Seaglider's travel toward the target (just as we chose our slope above). At each speed setting, the code uses a nonlinear solver to solve for ferry angle. First guess at speed is maximum. If the predicted distance over ground (DOG) with the set correction is less than the range to target (typical case) then the computation is complete. Otherwise we try the minimum speed. If the resulting DOG is greater than the range to target then the computation can't do any better and it is complete. If neither limit applies, we iterate via bisection to settle on the best speed. At convergence, the Seaglider has a ferry angle to steer and a horizontal speed to apply. The predicted horizontal speed is used to calculate a new value for glide slope.

**$N_DIVES**
The number of dives to complete before entering recovery. A value of zero (default) means dive until stopped by pilot command ($QUIT) or some other recovery condition occurs (battery exhaustion, error limit exceeded, etc.). [0]

**$N_FILEKB**
An integer value (-17 < **$N_FILEKB** < 17), which is the size (in kilobytes) and type (gzip-compressed or uncompressed) of file used for data uploading. Positive numbers direct the Seaglider to first use gzip to compress the data file, then split it into **$N_FILEKB**-sized pieces. Negative values for **$N_FILEKB** disable the gzip compression, but still cause the non-gzipped binary data file to be split into **$N_FILEKB**-sized pieces before transmission. A value of 0 means no splitting or compression is performed. The maximum allowed value of this parameter is currently set to 16. [4]

**$N_GPS**
The GPS termination criteria, encoded as *eeennff*, which, in addition to requiring horizontal dilution of precision (HDOP) to be less than 2.0, specifies the maximum estimated horizontal position error (HPE) permitted (*eee*, in meters), the minimum number of satellites required (*nn*) , and the maximum number of valid fixes to acquire (*ff*). If either *eee* or *nn* are zero, the corresponding criteria is ignored; if both are zero the GPS code terminates acquisition upon the first fix with HDOP <= 2.0 and after no more than *ff* fixes. Thus, the default value, 100840, requires a fix with HDOP <= 2.0 and an HPE of less than 10 meters acquired from at least 8 satellites but waiting no more than 40 valid fixes.
A negative value is treated as a specification of *ff* only (HPE and number of satellites criteria are ignored) and, when syncing time from GPS, forces a wait for that many valid fixes, independent of HDOP, HPE or number of satellites. A value of zero for all fields indicates that the NMEA data from the GPS should be copied to the glider's fix register at every update, regardless of the validity of the fix. Acquisition will cease on receipt of any valid fix with HDOD <= 2.0 or on timeout ( **$T_GPS**). This might be useful when debugging lack of fix information to see if any of the NMEA fields are updating. [100840]

**$N_NOCOMM**
The number of dives that are allowed to occur without a complete and successful data communication session before actions are taken to improve communications, enter recovery, or navigate to a rescue position according to the value of **$NOCOMM_ACTION**. The default (and traditional) behavior with a value for **$NOCOMM_ACTION** of 0 is for the surface buoyancy parameter **$SM_CC** to be set to the maximum allowed by the software limits. This is a safety provision in the event **$SM_CC** is not sufficient to allow for a good antenna position. [1]

**$N_NOSURFACE**
An integer value that determines when the glider will finish the dive at **$D_FINISH** and when it will finish at the actual surface. For values greater than 1, the rule is that when the remainder of **$DIVE** divided by **$N_NOSURFACE** is zero, the glider will finish the dive at depth **$D_FINISH**. Other dives will finish at the surface. For negative values less than -1, this logic is reversed, and values of **$DIVE** divisible by **$N_NOSURFACE** will finish at the surface; all others will be subsurface finishes. Values of 1 and -1 are not allowed. A value of 0 disables this behavior. [0]

**$NOCOMM_ACTION**
Bitmask parameter controlling behavior after **$N_NOCOMM** dives have passed without successful communication with the basestation. The parameter is a logical or of the bits described below:

- Bit 0 pump to max behavior:
  - 0 = pump to max after $N_NOCOMM dives
  - 1 = pump to max after 1 dive with no comms. Setting this bit allows for setting $N_NOCOMM to a larger value while still getting the typical behavior of pumping to max after a single dive with failed communications.

- Bit 1 recovery: Setting this bit causes the glider to enter recovery after $N_NOCOMM dives with no communications. This bit takes precedence over any values in bits 2-4.
- Bit 2 EPIRB mode: If set, the glider will loiter on the surface in low power sleep for **$T_EPIRB** seconds. The loiter period is immediately before GPS2 is acquired. The intent of this mode is to aid visual, acoustic, or ARGOS based backup recovery systems while still keeping the glider diving and navigating. This bit can be set independently or in conjunction with bits 3 and 4.
- Bit 3 escape: Setting this bit causes the glider to begin navigating to the escape target defined for the current target. If no escape target is defined then the glider navigates by heading according to the **$ESCAPE_HEADING** parameter.
- Bit 4 moor at position: This causes the glider to latch the current GPS position as a target and set that target as the current waypoint. The glider will try to virtually moor at that position.
- Bit 5 clear flow control bits on **$PHONE_DEVICE**.
- Bit 6 lengthen **$T_RSLEEP** by a factor of 30.
- Bit 7 send a SMS messages to the sms_email address configured in NVRAM. The messages is identical to the GPS status line that is emitted to the comm.log on the basestation.
- Bit 8 try both primary and alternate phone numbers before deciding communications are not successful.

As an example, parameter values for **$N_NOCOMM** of 10 and **$NOCOMM_ACTION** of 21 will cause the glider to reset $SM_CC to max any time one dive passes with no communications. If 10 dives pass with no communications, the glider will try to stay at its current position, with extra loitering on the surface between station keeping dives. [0]

**$P_OVSHOOT**
The distance (cm) by which the pitch mass overshoots its target after the pitch motor is turned off. [0.04]

**$P_OVSHOOT_WITHG**
The distance (cm) by which the pitch mass overshoots its target in the direction of gravity after the pitch motor is turned off. [0.08]

**$PHONE_DEVICE**
A configuration value specifying the model of the attached device, set by the assembler or builder. These devices have dedicated hardware ports on all motherboard revisions and, as such, a port specification is not necessary. [16]

**$PHONE_SUPPLY**
Controls the source of power for the Iridium phone: 1=10V battery pack, 2=24V battery pack. Intended for load balancing the energy consumption between the two packs. If negative, measures current draw of phone directly. [2]

**$PITCH_AD_RATE**
The pitch rate (A/D counts/second) used as the threshold for retries when pitching. If the observed rate is less than this number, the pitch will be stopped and restarted. The retries will continue until the pitch motor timeout limit is reached, then an error will be declared. [150]

**$PITCH_ADJ_DBAND**
This parameter, with **$PITCH_ADJ_GAIN**, enables and adjusts active (closed-loop) control on Seaglider pitch during a dive and climb. The Seaglider will automatically seek to maintain the desired pitch angle by moving the pitch mass when

$$| Pitch_{observed} - Pitch_{desired} | > \$PITCH\_ADJ\_DBAND$$

**$PITCH_ADJ_DBAND** has units of degrees. A value of 0 disables automatic pitch adjustment. [0.5]

**$PITCH_ADJ_GAIN**
This parameter, with **$PITCH_ADJ_DBAND**, enables and adjusts active (closed-loop) control on Seaglider pitch during a dive and climb. The amount of the adjustment is given by

$$(PitchDesired - PitchObserved) * \$PITCH\_ADJ\_GAIN$$

**$PITCH_ADJ_GAIN** has units of cm/degree. Adjustments are calculated at the beginning of the active guidance and control phase, based on the pitch observed over the same samples for which observed vertical speed is

calculated. Adjustments are not made during the first two active guidance and control phases following the start of a dive or climb. A value of zero disables automatic pitch adjustment. [0.03]

**$PITCH_CNV**
Pitch position conversion factor, from A/D counts to centimeters (cm/AD count). This is a constant determined by the pitch of the worm gear that drives the pitch motion, and is set by the builder or assembler. [0.0046]

**$PITCH_DBAND**
The pitch position deadband (cm). [0.05]

**$PITCH_GAIN**
The amount of vehicle pitch (degrees) change corresponding to 1 cm movement of the pitch mass. [15]

**$PITCH_MAX**
Pitch position software limit (A/D counts) aft. [3850]

**$PITCH_MAXERRORS**
The number of pitch motor errors allowed before the Seaglider goes into recovery state. An error occurs when the **$PITCH_TIMEOUT** expires prior to achieving the commanded pitch A/D position.

**$PITCH_MIN**
Pitch position software limit (A/D counts) forward. This is also the value to which the pitch mass moves during the surface maneuver (fully forward for maximum pitch down). [490]

**$PITCH_TIMEOUT**
Pitch mass timeout (seconds). [20]

**$PITCH_VBD_SHIFT**
Vehicle pitch is controlled by the axial positions of the mass shifter and the hydraulic oil. **$PITCH_VBD_SHIFT** parameterizes volume control (VBD) displacement as an equivalent mass shifter displacement. During each GC maneuver, pitch control (cm) is computed as the sum of the pitch desired (in degrees, see the third field in **$MHEAD_RNG_PITCHd_Wd**) divided by pitch gain (**$PITCH_CNV**) plus the VDB control (cc) times **$PITCH_VBD_SHIFT** (cm/cc). [0.00167]

**$PITCH_W_DBAND**
Deadband on pitch adjustments used for vertical speed control. Units of cm/s. If the observed vertical speed is outside the deadband, but not beyond **$W_ADJ_DBAND**, the glider will adjust pitch according to **$PITCH_W_GAIN** try to maintain speed. Only valid when **$PITCH_W_GAIN** is also non-zero.[0]

**$PITCH_W_GAIN**
Gain on pitch adjustments used for vertical speed control. Units of cm per m/s. If the observed vertical speed is outside **$PITCH_W_DBAND** then this parameter controls the adjustment made to pitch control to try to maintain desired speed. If this parameter is negative adjustments will be made to both slow and speed the glider. If positive, adjustments will only be made if the glider is moving too slowly. This parameter is only used during climbs. This parameter cannot be used at the same time as **$PITCH_ADJ_GAIN**. Interesting values might be in the range 2--10.[0]

**$PRESSURE_SLOPE**
Slope of linear fit between psig and pressure sensor output (after digitization to A/D counts through AD7714). The fit is calculated from calibration data received with each pressure sensor, and converted to A/D counts knowing the configuration of the AD7714 and associated circuitry. This number is a constant for each pressure sensor and associated calibration. [1.159254e-04]

**$PRESSURE_YINT**
Y-intercept of linear fit between psig and pressure sensor output (after digitization to A/D counts through AD7714). This is the value that is adjusted in the field at launch to correct the pressure sensor relative to atmospheric pressure so that the seawater surface corresponds to 0m depth. [-19.65]

**$PROTOCOL**
The file transfer protocol to use. (bits 0-2 data protocol, bits 3-5 control protocol: 0 - xmodem, 1 - raw, 2 - kermit, 3

- kermit batch (data only), bit 6 - encrypt upload) [0]

**$R_PORT_OVSHOOT**
Roll mass overshoot (A/D counts) to port after motor is turned off. Overshoots are assumed to be positive, in the sense of past the desired position. The sign of the **$R_PORT_OVSHOOT** parameter indicates how the code handles the overshoots, not a direction. Positive values allow the Seaglider operating software to automatically compute the roll overshoots and apply them after each roll maneuver. Negative values allow the pilot to specify a static overshoot value to be applied uniformly to each roll maneuver. In the case where the Seaglider is computing and applying the roll overshoots, the value reported in the log file is the last value computed during a dive. [25]

**$R_STBD_OVSHOOT**
Roll mass overshoot (A/D counts) to starboard after motor is turned off. See comments under **$R_PORT_OVSHOOT** above regarding sign conventions. [25]

**$RAFOS_CORR_THRESH**
Correlation threshold to use when selecting RAFOS hits for navigation solutions.
[60]

**$RAFOS_DEVICE**
A configuration value specifying the model of the attached device, set by the builder or assembler. These devices have dedicated hardware ports on all motherboard revisions and, as such, a port specification is not necessary. A value of -1 for the RAFOS device specifies that RAFOS is not installed. [-1]

**$RAFOS_HIT_WINDOW**
Size of search window in seconds to use when clustering hits for navigation solutions.
[3600]

**$RAFOS_PEAK_OFFSET**
Offset in seconds of actual arrival time from the receiver reported arrival index due to receiver firmware artefacts.
[1.5]

**$RELAUNCH**
A Boolean value: 1 means the Seaglider automatically resumes diving in the event of a crash to TOM8 (or any watchdog timer reset), 0 means the Seaglider automatically enters recovery in the event of a crash to TOM8 (or other watchdog timer reset). Internally the Seagliders ors this value with 2 during a commanded reboot and 4 to indicate the glider is in recovery. If the internal indication for recovery is true then the glider will enter into recovery on reboot, regardless of the original value of the parameter. Internal indicator for commanded reboot is cleared after reboot is complete and at selftest and launch. Indicator for recovery is cleared upon exiting recovery and at selftest and launch. [1]

**$RHO**
The water density ($g/cm^3$) used for converting buoyancy force in grams to seawater displacement in $cm^3$. This parameter is also used in the on-board performance prediction computations. [1.023]

**$ROLL_AD_RATE**
The roll rate in A/D counts per second that the Seaglider operating code uses as the threshold for retries when rolling. If the observed rate is less than this number, the roll will be stopped and restarted. The retries will continue until the roll motor timeout limit is reached, then an error will be declared. [400]

**$ROLL_ADJ_DBAND**
This parameter, with **$ROLL_ADJ_GAIN**, controls the automatic adjustment of the Seaglider roll centers based on observed turn rate. At the end of a complete passive phase, a full guidance and control (GC) interval, the Seaglider will adjust the appropriate dive or climb roll center based on the turn rate over the last half of the passive phase if

$$|\text{turn rate}| > \textbf{\$ROLL\_ADJ\_DBAND}.$$

**$ROLL_ADJ_DBAND** has units of degrees/second. A value of zero disables automatic adjustment of the roll centers. [0.03]

**$ROLL_ADJ_GAIN**
This parameter, with **$ROLL_ADJ_DBAND**, controls the automatic adjustment of the Seaglider roll centers based on observed turn rate. The amount of the adjustment is

-(turn rate)***$ROLL_ADJ_DBAND**, if climbing, and,

(turn rate)***$ROLL_ADJ_DBAND**, if diving.

**$ROLL_ADJ_GAIN** has units of degrees/(degrees/second). This adjustment is only meaningful the next time the Seaglider turns, as the Seaglider will roll back to this new neutral position on the next turn. It will not immediately roll from the old neutral to the new neutral at the start of the next active guidance and control (GC) phase. Note that if **$HEAD_ERRBAND** is large, then several active/passive GC phases might pass before a turn is initiated and the new roll center is used. A new adjustment is computed only after a turn has been completed. [1.0]

**$ROLL_CNV**
Roll position conversion factor, from A/D counts to degrees. This is a constant determined by the particulars of the roll gear train, motor and potentiometer. [0.02827]

**$ROLL_DEG**
The number of degrees to roll the internal roll mass during a turn. Traditionally, this was hard-coded to $40^o$, and is used in all turns. With larger battery packs, different trim lead configurations, and the possibility of longer wings, this was changed to allow the pilot to specify other roll targets for turns, primarily to take advantage of more of the available roll capacity. The roll software limits will still provide the effective roll maximums. [45]

**$ROLL_MAX**
Roll position software limit (A/D counts) to starboard. The glider operating software will stop the mass shifter at this value when rolling to starboard. [4000]

**$ROLL_MAXERRORS**
The number of roll motor errors allowed before the Seaglider goes into recovery state. An error occurs when the **$ROLL_TIMEOUT** expires prior to achieving the commanded roll A/D position.

**$ROLL_MIN**
Roll position software limit (A/D counts) to port. The Seaglider operating software will stop the mass shifter at this value when rolling to port. [120]

**$ROLL_TIMEOUT**
Roll maneuver timeout (seconds). [15]

**$SEABIRD_[C_G/ C_H/ C_I/ C_J/ T_G/ T_H/ T_I/ T_J]**
Sea-Bird Electronics-provided calibration coefficients for their conductivity and temperature sensor on the Seaglider. These values are used to compute calibrated temperature and salinity for hardware test purposes and in situ density for self-trimming applications. Currently used for subsurface finish maneuvers in which the glider attempts to become neutral at a fixed depth below the surface or an obstacle on the surface.

**$SIM_PITCH**
Simulated Seaglider desired pitch angle (degrees) during the dive phase of a simulated run. If non-zero, this value is used in place of observed pitch on the dive phase of a simulated run. A value of 0 disables this feature. This parameter is automatically zeroed during the Sea Launch procedure. [-20]

**$SIM_W**
Simulated Seaglider desired vertical velocity (m/s). If non-zero, this value is used to generate depths so that the Seaglider can do simulated dives in the lab or on the deck. A value of 0 disables this feature. This parameter is automatically zeroed during the Sea Launch procedure. [0.1]

**$SM_CC**
The specified minimum-buoyancy position of the VBD ($cm^3$) the Seaglider attains at the surface. If the Seaglider enters the surface maneuver with less than this value of VBD, it pumps to this value. If the Seaglider enters the surface maneuver with more than this value of VBD, it does not change the value of VBD and continues to the next part of the surface maneuver. [350]

**$SPEED_FACTOR**
A factor to compensate for Seaglider's inability to dive at the desired horizontal velocity. This is a measure of the efficiency of the Seaglider's progress along a specified track. Factors that lower the Seaglider's efficiency are turns, leaving the surface at arbitrary headings, and reduced horizontal speed during the apogee maneuver. **$SPEED_LIMITS** are multiplied by this factor and **$KALMAN_CONTROL** components are divided by it. [0.9]

**$STOP_T**
Date and time after which the glider will stop diving and enter recovery. The stop time is only checked upon surfacing (the glider will not stop in the middle of a dive if this time is reached). Must be specified in the form mmddyyhh. For example, a value of 12151613 will cause the glider to enter recovery on the first surfacing after 15 December 2016 at 1300 hours. A value of zero turns off this check. [0]

**$STROBE**
Controls blinking of the strobe LED function available on gliders with Rev C motherboards. A value of zero disables all blinking, one turns the strobe on when the glider is in recovery, and two turns the strobe on whenever the glider is on the surface (between dives and recovery). [0]

**$SURFACE_URGENCY**
Number of dives to accumulate before trying extra surfacings. The $SURFACE_ parameters only affect under ice capable gliders. For non-under ice capable gliders, these parameters have no effect. [0]

**$SURFACE_URGENCY_FORCE**
Dive number modulo for forced extra surfacing attempts. For under ice only. [0]

**$SURFACE_URGENCY_TRY**
Dive number modulo for extra surfacing attempts. For under ice only. [0]

**$T_ABORT**
A safety time (minutes) such that if the current elapsed time on a given dive exceeds this time, the Seaglider will enter the recovery phase. [1440]

**$T_BOOST**
The desired time (seconds) to run the boost pump during VBD operations deeper than **$D_BOOST**. A value of 0 runs the boost pump continuously. If non-zero, the boost pump will run for a minimum of 3 seconds (to prime the main pump) and then shut off the boost pump after **$T_BOOST** total pump seconds have elapsed.

**$T_DIVE**
The desired time (minutes) for the glider to make one dive-climb cycle to the depth **$D_TGT** and back to the surface. This value does not include the time for pumping during the apogee phase. The value is used to calculate the Seaglider's desired vertical velocity in a particular dive, using the naive calculation w (cm/s) = 2*__$D_TGT__*100/(__$T_DIVE__*60). The desired w is combined with **$MAX_BUOY**, the range to the target, and the Seaglider hydrodynamic model to calculate the Seaglider's desired pitch angle on any given dive. [330]

**$T_EPIRB**
The amount of time in seconds to loiter on the surface when **$N_NOCOMM** is exceeded and bit 2 (EPIRB mode) of **$NOCOMM_ACTION** is set. [0]

**$T_GPS**
The maximum allowed time (minutes) to obtain a GPS position (GPS timeout). We often run with a **$T_GPS** of longer than 12.5 minutes, in order to ensure that the GPS receiver has time to receive a complete set of almanac entries in the event that lack of an almanac prevents getting a timely fix. [15]

**$T_GPS_ALMANAC**
Time to wait (in minutes) to acquire the GPS almanac. The wait will happen the next timethe GPS is turned on. After the wait, the parameter will be reset to zero and the regular GPS operation (presumably a fix) will proceed. If the parameter is greater than zero then the almanac sentences will be checked every minute. The wait will halt when the time has expired or at least ten satellites have recent almanac sentences. If the parameter is negative then the wait will only halt after the time has expired. A negative value also forces a complete NVRAM reset before the wait starts. [0]

**$T_GPS_CHARGE**
Time to wait (in seconds) before trickle-charging the GPS receiver (for Garmin GPS25 engines only). Negative values mean the GPS25 does not need charging. [-348.74615]

**$T_LOITER**
Time (seconds) to loiter after going neutral at apogee, before pitching up and becoming positively buoyant for climb. While in the loiter state the glider will attempt to maintain zero vertical velocity. It will pump, but not bleed (become heavier) to do this. The glider does not servo on depth in this state. All other timeouts and depths are honored in this state. $T_MISSION and $T_ABORT need to be adjusted manually to account for the additional dive duration. Total dive duration will be $T_DIVE + $T_LOITER. GC and sampling intervals during the loiter state are controlled by the appropriate depth bins in the science file.

**$T_MISSION**
The maximum mission time (minutes) allowed. After **$T_MISSION**/2 minutes, the Seaglider will transition from dive phase to apogee phase, then commence the climb phase. If **$T_MISSION** is reached prior to the Seaglider reaching **$D_SURF**, the Seaglider immediately enters the surface phase. This time includes the dive, apogee and climb phases. [375]

**$T_NO_W**
The time (seconds) for the Seaglider to wait with no significantly non-zero vertical velocity (less than 1 cm/s, as measured by dP/dt) before proceeding the next phase of a dive. This is primarily used to move from dive phase to climb phase (initiate an apogee maneuver) when the Seaglider unexpectedly encounters the bottom. Note that this protection is only in place at depths below **$D_NO_BLEED**. [120]

**$T_RSLEEP**
The sleep time interval (minutes) during the recovery phase. During the recovery phase, the Seaglider first gets a GPS fix, then calls the basestation up to **$CALL_TRIES** times to upload the GPS fix, then goes into low-power sleep for this length of time. The surface evolution has about two minutes of "overhead", so that the Seaglider calls are about (**#$T_RSLEEP** + 2) minutes apart in practice. [8]

**$T_TURN**
The maximum amount of time (seconds) allowed to complete a turn during the active G&C mode. If this timeout is reached before the desired heading is reached, the Seaglider rolls back to neutral and continues. [225]

**$T_TURN_SAMPINT**
The sample interval during active and passive G&C. This should be short enough so that the glider cannot pass entirely through the desired heading deadband without sampling. This was implemented to correct a bug in the guidance routine that allowed for complete circles to occur when trying simple course corrections. A negative value indicates that the "extra" samples taken during a turn should be recorded.[5]

**$T_WATCHDOG**
The value to which the watchdog timer is set (minutes). This is an information-only parameter; the watchdog timer itself is set with DIP-switches on its chip on the main board. [10]

**$TCM_PITCH_OFFSET**
Static offset in pitch axis (degrees) between the compass output and the actual Seaglider body, as measured in the lab. [1.3]

**$TCM_ROLL_OFFSET**
Static offset in roll axis (degrees) between the compass output and the actual Seaglider body, as measured in the lab. [-0.9]

**$TGT_AUTO_DEFAULT**
A Boolean parameter: if set to 1, automatically update the default target in NVRAM, if set to 0, do not. [0]

**$TGT_DEFAULT_LAT**
Floating point value (decimal degrees) between -90.0 and 90.0. Together with **$TGT_DEFAULT_LON** this parameter provides a default target location when the targets file cannot be read.

**$TGT_DEFAULT_LON**
Floating point value (decimal degrees) between -180.0 and 180.0. See **$TGT_DEFAULT_LAT**, above.

**$UNCOM_BLEED**
The uncommanded change in A/D counts of VBD bleed that triggers the following actions in an attempt to save the Seaglider: stop whatever motor is running (the assumption is that electrical noise from one of the motors causes the Skinner valve to open) and disable it, close the Skinner valve, and enter the recovery phase (go to the surface and call home). [20]

**$UPLOAD_DIVES_MAX**
The maximum number of dives to upload at one surfacing. A value of -1 means upload all available dives that have not been previously uploaded. [-1]

**$USE_BATHY**
If 0, do not use bathymetry grid file. Either dive to **$D_TGT**, or use the on-board altimeter to determine the appropriate depth at which to enter the apogee maneuver. If positive, use bathymetry file named *bathymap.nnn*, where *nnn* = **$USE_BATHY** (with leading zeros, if necessary), and dive to the shallower of the two parameters **$D_GRID** and **$D_TGT**. If negative, search for an on-board *bathymap.nnn* appropriate for the current position of the glider. This would be the standard usage in operating areas covered by more than one map. Example: If **$USE_BATHY** = 7, then use the bathymetry file called *bathymap.007* to get **$D_GRID**. No other map would be used. If **$USE_BATHY** = -4, then search all on-board *bathymap.nnn* files for one that covers the current glider position.

**$USE_ICE**
Same as **$USE_BATHY** above, except for the controlling the use of ice maps. When **$USE_ICE** is non-zero, the glider will use the appropriate maps to determine time-varying ice condition codes for surfacing decisions.

**$VBD_BLEED_AD_RATE**
The bleed rate in A/D counts per second that the code uses as the threshold for retries when bleeding. If the observed rate is less than this number, the bleed will be stopped and restarted. [10]

**$VBD_CNV**
VBD position conversion factor from A/D counts to $cm^3$. This is a constant determined by the geometry of the internal hydraulic fluid reservoir and the potentiometers. The sign is negative to mean that higher A/D counts means more oil in the internal reservoir, hence a less-inflated external bladder, hence a lower Seaglider displacement, and thus a lower Seaglider buoyancy. [-0.2453]

**$VBD_DBAND**
VBD position deadband ($cm^3$). [2]

**$VBD_LP_IGNORE**
Which VBD linpots to ignore when computing VBD position, if any. 0 - average both values; 1 - ignore A; 2 - ignore B [0]

**$VBD_MAX**
Variable Buoyancy Device (VBD) position (A/D counts) software limit when the internal reservoir is almost full (external bladder fully bled, minimum glider buoyancy). The Seaglider operating software will close the VBD main bleed valve (Skinner valve) when this value is reached. The builder or assembler typically sets this. [3800]

**$VBD_MAXERRORS**
Number of VBD errors allowed before entering recovery phase. This is an attempt to keep a Seaglider at the surface (prevent another dive) in the event it is reporting VBD errors. [1]

**$VBD_MIN**
Variable Buoyancy Device (VBD) position (A/D counts) software limit when the internal reservoir is almost empty (external bladder fully pumped). The Seaglider operating software will stop the VBD pump when this value is reached. The builder or assembler typically sets this. [300]

**$VBD_PUMP_AD_RATE_APOGEE**
The pump rate in A/D counts per second that the code uses as the threshold for retries when pumping at apogee. If the observed rate is less than this number, the pump will be stopped and restarted. The retries will continue until the pump timeout limit is reached, then an error will be declared. [4]

**$VBD_PUMP_AD_RATE_SURFACE**
The pump rate in A/D counts per second that the code uses as the threshold for retries when pumping at the surface. If the observed rate is less than this number, the pump will be stopped and restarted. The retries will continue until the pump timeout limit is reached, then an error will be declared. [6]

**$VBD_TIMEOUT**
VBD pump or bleed timeout (seconds). [720]

**$W_ADJ_DBAND**
Seaglider adjusts its buoyancy to maintain a desired vertical velocity (w) in the presence of strong density changes. In particular, if the observed w is too low, the glider may attempt to bleed on dives (subject to **$D_NO_BLEED** and **$MAX_BUOY**) or pump on climbs. However, in the presense of strong internal waves, the glider may appear to slow transiently because of upwelling on the dive or downwelling on the climb, triggering unneeded buoyancy adjustments. This parameter limits active control on VBD changes during a dive and climb. The Seaglider will automatically seek to maintain the desired vertical velocity by changing the VBD when

$$| W_{observed} | < | W_{desired} | - \$W\_ADJ\_DBAND$$

**$W_ADJ_DBAND** has units of cm/s, should be positive, and correspond to the typical RMS variance of observed w. A value of 0 ignores internal wave effects. [0]

**$XPDR_DEVICE**
A configuration value specifying the model of the attached device, set by the builder or assembler. These devices have dedicated hardware ports on all motherboard revisions and, as such, a port specification is not necessary. See **$COMPASS_DEVICE**. [21]

**$XPDR_INHIBIT**
A configuration value specifying the transponder inhibit time in hundreds of milliseconds. The inhibit time is the time after a transponder reply during which the transponder will not reply to subsequent interrogation. Shorter times mean the transponder can be interrogated more rapidly. Valid values are from 1 (100 milliseconds) to 99 (9.9 seconds). [90]

**$XPDR_VALID**
A configuration value specifying the transponder interrogation validation sensitivity in units of 0.5 ms. Valid values are from 0 (no validation) to 6 (3 ms). The validation value is the total time over a 10 ms window following initial triggering that the detector circuit must remain triggered. Validation of interrogation pulses is only available on AAE transponder boards with firmware version 3.061 or later. Longer validation times will reduce spurious interrogration replies, but could result in decreased range. A value of zero results in no validation which is equivalent to the behavior of firmware prior to 3.061. [0]