

Format of SOLOII X messages: Argo Version Manual/Decoder V1.5
latest update: 07 January 2015
[For ROM SBE602 17May2013, 11Sep2013, 20Sep2013, 29Oct2013,10Dec2013]

An X message is used to transfer data from ISU to GS or from GS to ISU. The data is assumed to be binary and each byte can have any value from 0x00 to 0xff. The format of the message is the same regardless of direction of transmission:

Xnnmddp<data>**\$**cc>

- X** = the character **X**
- nn = number of data characters in the message following after nn. The count does not include **X**, nn, or anything from **\$** to the end >. The count is in 2 binary bytes with MSB first and LSB second.
- mm = serial number of SOLOII. The SN is in 2 binary bytes with MSB first and LSB second.
- dd = the dive number in 2 binary bytes with MSB first and LSB second.
: Dive number begins at -1 for the start-up, increments to 0 for the test dive, increments +1 for all normal' (0xE2) dives.
- p = one-byte packet ID index, range 0 to 255. Used to identify multiple X messages within a dive cycle. The data for each dive cycle starts with p=0.
- <data> = binary data characters. The length of <data> = nn -5. The contents of the <data> section is described below.
- \$** = a dollar sign delimiter at start of the checksum
- cc = the 8 bit byte-wise checksum from **X** to the byte preceding the **\$**. The 8 bit sum is coded as 2 4bit nibbles. The binary value of a nibble is converted to a visible character by adding 0x30. Thus a value of 0x0 -> 0x30 = character '0', 0x1 -> 0x31 = '1', 0xe -> 0x3e = '>', and 0xf -> 0x3f = '?'.
> = a > delimiter at end of checksum which also serves as a prompt to GS that the ISU is done transmitting and that the GS may now transmit to ISU.

The remainder of this document describes the format of the <data> portion of the message sent from SOLOII to the ground station (**GS**). The format of commands from **GS** sent to SOLOII will be described in another document.

Highlights in document

Fields that are moved relative to the previous float version are highlighted in cyan

New fields relative to the previous version are highlighted in yellow

NOTE on firmware V1.5: An ice detection algorithm was introduced with this firmware. However, for the float to be able to use the algorithm, the Seabird CTD must be a SBE41CP-V3.0C. SBE41CP-V3.0A, does not contain the necessary commands. Thus not all V1.5 floats may have operational ice detection due to hardware.

The <data> section contains information from multiple sensors. Data from successive sensors are separated by a semicolon (';' = 0x 3b); the final sensor is terminated by a ';' (immediately preceding the \$ delimiter).

IDjj<sensor_data>;
ID = one-byte sensor ID code.
 jj = Number of bytes for this sensor. The count includes **ID**, jj, and the trailing ;.
 The count is in 2 binary bytes with MSB first and LSB second.
 <sensor_data> = binary data characters. The length of <sensor_data> = jj-4 bytes, and its contents are described below for each sensor.
 ; = delimiter at the end of each sensor's data.

The **ID** byte is divided into two 4-bit nibbles. The MS nibble identifies the sensor and the second nibble specifies the message number for that sensor. For example, the ID for first Pressure message is 0x10, the second is 0x11, the third 0x12, etc. For a 1000 sample profile, there will be 6 messages for each of the pressure, salinity and temperature sensors.

Sensor	ID byte(hex)	
GPS	00	fix at end of first diagnostic dive at start of mission
GPS	01	fix at before leaving surface
GPS	02	fix at end of normal profiling ascent
GPS	03	fix following mission abort
GPS	05	fix during BITest
Pressure	1x	depths of CTD readings (scaled 1st difference)
Temperature	2x	depth series of temperature (scaled 1st difference)
Salinity	3x	depth series of salinity (scaled 1st difference)
Fall Rate	40	series of time,depth during SOLO II downward profile
Rise Rate	50	series of time,depth from drift depth to surface
Pump Series	60	pressure,time, voltage,current,vacuum for each pump
High Resolution Pressure	9x	High Resolution Pressure (scaled 1 st difference)
High Resolution Temperature	ax	High Resolution Temperature (scaled 1 st difference)
High Resolution Salinity	bx	High Resolution Salinity (scaled 1 st difference)
Mission EEPROM	dx	ASCII dump of mission parameters in EEPROM
Engineering	e0	diagnostic data in first diagnostic dive
Engineering	e2	engineering data in normal profiling dive
Engineering	e3	engineering data following mission abort
Engineering	e5	engineering data BIT test pass
Engineering	e6	engineering data BIT test failure
Argo Data	f0	Mission parameter list
Test pattern	f1	<i>ID reserved, format not yet defined</i>

GPS data (ID=0x00, 0x01, 0x02, 0x03, 0x05)

The LS nibble of the ID indicates in what phase of the mission the fix was taken. The remainder of the data is the same for all mission phases. The length of GPS data is in bytes 1 and 2. GPS fix data starts in byte 3:

Byte	Contents
0	Mission phase: 0 = 1st diagnostic dive at the start of a mission 1 = beginning of normal dive cycle (just before leaving surface) 2 = end of a normal dive cycle 3 = following mission abort 5 = during BITest
1-2	Number of bytes in the message, 24 = 0x18 with the format as described here
3	0 if fix is invalid, +2 if longitude is East, -2 if longitude is West
4-7	Signed latitude degrees * 1e7
8-11	Signed longitude degrees * 1e7 range (+180 to -180 degrees)
12-13	GPS week (traditional GPS week =0 to 1023 in LS 10 bits; rollover fix in MS 6 bits)
14	GPS day of week, 0=Sunday, 6=Saturday
15	UTC hour
16	UTC minutes
17	Time to get fix = (seconds to get fix)/10 , range 0 to 255 = 0 to 2550 seconds
18	Number of satellites used in fix
19	Minimum signal level
20	Average signal level
21	Maximum signal level
22	10*Horiz. dilution of precision
23	; terminator (0x3B)

Pressure data (ID=0x10)

Temperature data (ID=0x20)

Salinity data (ID=0x30)

Profile data from the pressure, temperature, and salinity sensors are all processed in the same way and the message format differs only in the ID code. The SeaBird CTD takes a profile as the SOLOII ascends and stores the values internally. When SOLOII reaches the surface, it takes the data from the CTD and block averages it in depth into **PRO_BINS** (= 1000) bins.

The size of depth bins can vary with depth. The averaging scheme is determined by 5 parameters: **BLOK**, **PB1**, **PB2**, **AV1**, and **AV2**. The smallest bin size is **BLOK** decibars. Bins 0 thru **PB1**-1 have a vertical extent of **BLOK** decibars. Bins **PB1** thru **PB2**-1 are **AV1*BLOK** decibars tall while bins **PB2** thru **PRO_BINS**-1 are **AV2*BLOK** decibars. In the special case that **PB1** >= **PRO_BINS**, then all of the bins are **BLOK** decibars in extent, and the values of **PB2**, **AV1**, and **AV2** are ignored.

The data series from all channels are processed in the same way and are synchronous with each other. Each depth series is broken into sub-blocks of 25 samples, and a first-differencing method is applied to each sub-block to reduce the number of bytes required to transmit the data. Because the data series will generally be longer than the 189 bytes available in a 9601 SBD message, it is divided into multiple messages. Each message has an integral number of sub-blocks in it. The final sub-block of the time series may have fewer than 25 samples in it. The data message looks like:

High Resolution Pressure data (ID=0x90)
High Resolution Temperature data (ID=0xa0)
High Resolution Salinity data (ID=0xb0)

The float can be set to return a high resolution P,T,S profile spanning a subsection of the primary binned profile. Data is packed and decoded similarly to the binned profile (ID=0x10, 0x20, 0x30). The High Resolution profile can return every scan of the CTD (1 Hz) or every other scan (1/2 Hz). The data is limited to 1024 values. [Note: When the High Resolution data is requested, the averaging of the primary binned profile must be done by the float (not within the CTD). Typical SOLO II averaging uses every other CTD scan. However if the High Resolution profile includes every scan, the bin averages will also use every scan. Thus the averaging of the primary binned profile may differ between the subsection with High Resolution data and all other spans.

Fall Rate data (ID=0x40)

As it falls from the surface to its drift depth, SOLOII periodically interrogates the SeaBird for a depth reading. This time series is sent back in this data message.

The data message looks like:

```
IDjj<start_time><time(1),depth(1)> . . . <time(m),depth(m)>;
  ID = one-byte sensor ID code = 0x40.
  jj = Number of bytes in the message. The count includes ID, jj, the data, and the
      trailing ;. The count is in 2 binary bytes with MSB first and LSB second.
start_time = SOLO time at start of fall (seconds since 1Jan2000) in 4 bytes (MSB first).
time(i) = seconds since start_time in 2 bytes, i=1, ..., m
depth(i) = depth (LSB=0.04 db) at time(i) in 2 bytes, i=1, ..., m.
          dBar = .04 * depth(i) -10
          depth(i) = 0xffff if the pressure reading is invalid
```

Each depth observation takes 4 bytes. The first time is taken when the valve is opened to leave the surface. The next two times are when the float passes 50m and 100m. After 100 m, pressures are logged every 30 minutes. Typically we allow for 500 (**FallIn**) minutes for the SOLOII to fall 1000 meters so there will be about 16 more measurements. The last record is at the 500 minute mark.

NOTE: time(i) returned within the fall rate data is aliased past 18 hours. In a well-behaved cycle this is not an issue. However if the float does not hold pressure during the drift phase, a pressure time pair is recorded during each of the floats drift buoyancy adjustments. These drift phase times will be aliased.

Beginning in ROM 10Dec13, the float can reset its clock using Iridium time (if GPS time is unavailable). Any re-epoch of the Iridium system will shift the reported time (after reset by Iridium) by 226492400 seconds.

Rise Rate data (ID=0x50)

The rise rate message is identical in structure to the fall rate message. The rise rate time series begins when the SOLO II opens its valve to descent from the drift depth to the profile depth. It logs a pressure/time record 10 times during its descent to the profile depth (interval = **PwaitN**/10). At the bottom of dive, whether determined by timing out (exceeding **PwaitN**) or by reaching the target depth (**Zpron**), another pressure/time record is logged. At this point, the float pumps for **PmpBtm** seconds. A pressure/time record is logged every 30 minutes while the float is ascending.

Pump data (ID=0x60)

. The data message looks like:

```
IDjj< depth(1),time(1),voltage(1),current(1),vac0(1),vac1(1)> . . .
  < depth(m),time(m),voltage(m),current(m),vac0(m),vac1(m);
    ID = one-byte sensor ID code = 0x60.
    jj = Number of bytes in the message. The count includes ID, jj, the data, and the
        trailing ;. The count is in 2 binary bytes with MSB first and LSB second.
    depth(i) = depth (LSB=0.04 db) at time(i) in 2 bytes, i=1, ..., m.
              dBar = .04 * depth(i) -10
              depth(i) = 0xffff if the pressure reading is invalid
    time(i) = seconds the pump ran in 2 bytes (signed)
    voltage(i) = average pump battery counts while pumping in 2 bytes (0.01V)
    current(i) = average pump current at bottom in 2 bytes, LSB=1ma
    vac0(i) = vacuum counts after pump starts in 1 byte
    vac1(i) = vacuum counts before pump stops in 1 byte
```

NOTE: vac0 and vac1 are NOT set for SIO ARM controllers in ROM versions 10Dec13, 29Oct13, and 20Sep13;

Engineering data (ID=0xe0, 0xe2, 0xe3, 0xe5, 0xe6)

The engineering data is used to diagnose SOLOII anomalies. A different format is used in each of the 3 distinct phases of a SOLOII mission. The LS nibble of the ID indicates the phase of the mission.

Byte	Contents
0	ID/Mission phase: 0xe0 = 1st diagnostic dive at the start of a mission 0xe2 = end of a normal dive cycle 0xe3 = following mission abort 0xe5 = BITtest
1-2	Number of bytes in the message, depends on mission phase as described below
3 -> ??	Depends on mission phase as described below

ID=0xe0, Engineering message in 1st diagnostic dive at start of mission

Byte	Contents
0	ID/Mission phase = 0xe0
1-2	Number of bytes = 76= 0x4c
3	Engineering message version =4
4	#packets in current session
5-10	0 (dummy filler)
11-12	EP -> sattime
13-14	DP->Vcpu = CPU battery voltage counts 0.01V
15-16	DP->Vpmp = Pump battery counts at surface(0.01V)
17-18	DP->Vple = Pump battery counts at end of last pump(0.01V)
19-20	BTvac = pcase vacuum at beginning of BIT in 0.01 inHg
21-22	DP->Air[1] = vac before filling bladder at surface 0.01 inHg
23-24	DP->Air[2] = vac after filling bladder at surface 0.01 inHg
25-26	DP->ISRID = i.d. of last interrupt
27-28	DP->HPavgI = average pump current at bottom, LSB=1ma
29-30	DP->HPmaxI = maximum pump current at bottom, LSB=1ma
31-32	Total seconds pumped to surface
33-34	Seconds pumped at Surface
35-36	DP->P[5] = surf press counts @ end of ASCEND (LSB=.04dBar)
37-38	SPRX = Surf press before resetoffset (pertains to prev dive)
39-40	SPRXL = press after resetoffset (pertains to prev dive)

41-42 diagP[0] = Press when "in water" sensed
 43-44 diagT[0] = Temp when "in water" sensed
 45-46 diagS[0] = Salinity when "in water" sensed
 47-48 SBnscan = # scans recorded by SBE
 // -1 (0xffff) indicates unable to get scan count from SBE
 // -2 (0xfffe) indicates SBE never started so SBE didn't reset
 // scan count before returning an old value
 49-50 Compacted SBntry,SBstr,SBstop status (see misspec.h):
 ((DP->SBntry&0xf)<<4) | ((DP->SBstr&0x3)<<2) | (DP->SBstop&0x3))
 51-52 diagP[1] = Shallowest press in profile
 53-54 diagT[1] = Shallowest Temp in profile
 55-56 diagS[1] = Shallowest Salinity in profile
 NOTE: The shallow PTS triplet uses the standard scaling until firmware 10Dec13.
 57-58 BTvac = BIT vacuum in 0.01 inHg
 59-60 BTPcur = BIT motor current OUT, LSB=1mA
 61-62 BTPsec = BIT Pump seconds
 63 BTPvac[0] = BIT Pump vacuum at beginning of test, before pumping
 64 BTPvac[1] = BIT Pump vacuum after pumping
 65-66 BTVple = BIT pump batt 0.01V
 67-68 BTVcpu= BIT CPU batt 0.01V
 69-70 exception flags (not set for ROM 17May2013, set for all others)
 71 vent data; MSB=#0.1 seconds vent motor ran
 72 LSB LLD status before/after vent ran
 73-74 AbrtCd = code for what caused abort_miss
 75 ; terminator

ID=0xe2, Engineering message in normal dive cycle

Byte	Contents
0	ID/Mission phase = 0xe2
1-2	Number of bytes = 98 = 0x62
3	Engineering message version = 4
4	#packets sent in current surface session
5-6	#tries to connect in previous surface session
7-8	parse_X_reply low order byte number of messages: upper byte bit field of errors
9-10	ATSBD return status in previous surface session
11-12	EP->sattime Seconds taken in previous surface session to send all SBD messages
13-14	DP->Vcpu = CPU battery voltage counts 0.01V
15-16	DP->Vpmp = Pump battery counts at surface(0.01V)
17-18	DP->Vple = Pump battery counts at end of last pump(0.01V)
19-20	DP->Air[0] = pcase vac during sinking @50db with oil all inside pcase ,0.01 inHg
21-22	DP->Air[1] = pcase vac before filling oil bladder at surface 0.01 inHg
23-24	DP->Air[2] = pcase vac after filling bladder at surface 0.01 inHg
25-26	DP->ISRID = i.d. of last interrupt
27-28	DP->HPavgI = average pump current at bottom, LSB=1ma
29-30	DP->HPmaxI = maximum pump current at bottom, LSB=1ma
31-32	Total seconds pumped to surface
33-34	Seconds pumped at Surface
35-36	SPRX = Surf press before resetoffset (pertains to prev dive)
37-38	SPRXL = press after resetoffset (pertains to prev dive)
39-40	diagP[0] = Pressure before pumping for ascent
41-42	diagT[0] = Temp before pumping for ascent
43-44	diagS[0] = Salinity before pumping for ascent
45-46	diagP[1] = Last (shallowest) Pressure scan on ascent
47-48	diagT[1] = Last (shallowest) Temperature scan on ascent
49-50	diagS[1] = Last (shallowest) Salinity scan on ascent.

NOTE: The shallow PTS triplet uses the standard scaling until firmware 10Dec13.

51-52 SBnbad = # bad bins from SBE
 53-54 SBnscan = # scans recorded by SBE
 // -1 (0xffff) indicates unable to get scan count from SBE
 // -2 (0xfffe) indicates SBE never started so SBE didn't reset
 // scan count before returning an old value
 55-56 Compacted SBntry,SBstr,SBstop status (see misspec.h):
 ((DP->SBntry&0xf)<<4) | ((DP->SBstr&0x3)<<2) | (DP->SBstop&0x3))
 57-58 DP->P[0] = press counts before begin of FALL (LSB=.04dBar)
 59-60 DP->P[1] = press counts at end of FALL (LSB=.04dBar)
 NOTE: P[1] will not be valid if the float fall exceeds profile depth
 61-62 DP->P[2] = press counts at beginning of DRIFT (LSB=.04dBar)
 63-64 DP->P[3] = press counts at end of DRIFT (LSB=.04dBar)
 65-66 DP->P[5] = surf press counts @ end of ASCEND (LSB=.04dBar)
 67-68 DP->PAVG[0]=average pressure over first half of DRIFT
 69-70 DP->TAVG[0]=average temperature over first half of DRIFT
 71-72 DP->SAVG[0]=average salinity over first half of DRIFT
 73-74 DP->PAVG[1]=average pressure over second half of DRIFT
 75-76 DP->TAVG[1]=average temperature over second half of DRIFT
 77-78 DP->SAVG[1]=average salinity over second half of DRIFT
 79-80 DP->fall_time = seconds from open air valve to end of settle
 81-82 DP->fall_rate = avg mm/sec while sinking
 83-84 DP-> SeekT = seconds pumped in 1st settle to drift
 85-86 DP-> SeekP = change of depth (signed 0.1 dbar in 1st settle)
 87-88 exception flags (can be added)
 0x0001 Valve failed to open
 0x0002 Valve failed to close
 0x0004 Questionable pressure
 0x0008 Antenna was toggled
 0x0010 Antenna switch failure. (no satellites even after toggling)
 0x0020 GPS communication error (cannot talk to GPS unit)
 0x0080 Float took too long to leave the surface. (toggled valve)
 0x0400 Leaky valve returned to surface (no drift nor seek executed)
 0x1000 Valve failure during Sink phase of mission
 0x2000 Valve failure during Ascend phase of mission
 89 vent data; # 0.1 seconds vent motor ran
 90 vent data; LLD status before and after vent ran
 91-92 SBE P offset(*800)
 93-94 PP->SeekSc; tenths of seconds pumped to target depth
 95-96 Number of Packets sent in previous cycle
 97 ; terminator

NOTE: SBE P offset (Bytes 91-92) is NOT set for SIO ARM controllers in ROM versions 10Dec13, 29Oct13, and 20Sep13

ID=0xe3, Engineering message following mission abort

Byte	Contents
0	ID/Mission phase = 0xe3
1-2	Number of bytes = 30 = 0x1e
3	Engineering message version = 4
4	#packets sent in current surface session
5-6	#tries to connect in last surface session
7-8	parse_X_reply: low order byte number of messages: upper byte bit field of errors
9-10	ATSBD return status in last surface session
11-12	Seconds taken in sending last SBD message
13-14	current CPU battery voltage counts 0.01V
15-16	current pump battery counts 0.01V
17-18	DP->Air[1] = pcase vacuum at beginning of abort 0.01inHg
19-20	DP->Air[0] = pcase vacuum at end of last xmit (previous cycle) 0.01 inHg
23-24	DP->ISRID = i.d. of last interrupt
25-26	AbtrCd = code for what caused abort_miss 0 = no error 1 = current time is later than RTCabort 2 = unable to WakeOST 3 = unable to send Dive number to SOLO II (LOdiveNo) 4 = Iridium ground station commanded to go to abort 5 = FnIDiv was completed. Mission is done 6 = Diagnostic dive failed to get GPS fix, pressure never>dBarGo, or unable to send message to Iridium 7 = pressure sensor failure
27-28	Empty
29	; terminator

ID=0xe5, Engineering message following BITest

Byte	Contents
0	ID/Mission phase = 0xe5
1-2	Number of bytes = 58 = 0x3a
3	Engineering message version =4
4	#packets sent in this surface session
5-6	SBE P Offset(*800)
7-8	CPU battery voltage 0.01 V
9-10	no load pump battery voltage 0.01 V
11-12	pump battery voltage counts at end of last pump (0.01V)
13-14	DP->HPavgI = average pump current at bottom, LSB=1ma
15-16	seconds pumped out during test
17	Oil vacuum before filling bladder 0.01inHG
18	Oil vacuum after filling bladder 0.01 inHG
19-20	DP-> Air[0] = Pcase Vacuum at beginning of BIT. (Oil Bladder Empty) 0.01 inHg
21-22	DP → Air[1] = Pcase Vacuum at end of BIT with air bladder inflated. 0.01 inHg
23	Number of tries needed to open valve
24	Number of tries to close valve
25-26	i.d. of last interrupt
27-56	string returned from SBE pt command
57	; terminator

ID=0xe6, Engineering message following FailedBITest

Byte	Contents
0	ID/Mission phase = 0xe6c
1-2	Number of bytes = 60 = 0x3c
3	Engineering message version =4
4	#packets sent in this surface session
5-6	BITest status register
7-8	SBE P Offset(*800)
9-10	CPU battery voltage 0.01 V
11-12	no load pump battery voltage 0. 01 V
13-14	pump battery voltage counts at end of last pump (0.01V)
15-16	DP->HPavgI = average pump current at bottom, LSB=1ma
17-18	seconds pumped out during test
19	Oil vacuum before filling bladder 0.01inHG
20	Oil vacuum after filling bladder 0.01 inHG
21-22	DP-> Air[0] = Pcase Vacuum at beginning of BIT. (Oil Bladder Empty) 0.01 inHg
23-24	DP → Air[1] = Pcase Vacuum at end of BIT with air bladder inflated. 0.01 inHg
25	Number of tries needed to open valve
26	Number of tries to close valve
27-28	i.d. of last interrupt
29-58	string returned from SBE pt command
59	; terminator

Mission EEPROM dump (ID=0xd0)

Most V1.5 floats use the following EEPROM format.....

Byte	Contents
0	ID/Mission phase = 0xd0,0xd1,0xd2,0xd3
1-2	len=Number of bytes (variable, see below)
3- (len-2)	ASCII listing of mission parameters Each EEPROM parameter has a 6 character name and 5 char value: NAMExx=vvvvv The = & signs are present in the listing of each parameter. (15 bytes/parameter) Successive parameters follow without gaps.
len-1	; terminator at the end of the dump

An example showing only the initial 3 and final 2 elements follows:

PchSec= -1|MaxHrs= 1440|dBarGo= -1|...| XP2= 0| XPdly= 0|;

The EEPROM dump message is sent only in response to a command "P" from the ground station. It is sent over 4 SBD messages (0xd0=328 bytes, 0xd1=328 bytes, 0xd2=328 bytes, 0xd3=7 bytes).

But format SBE602 29Oct2013 accidently used an older format which dropped some of the EEPROM values

Byte	Contents
0	ID/Mission phase = 0xd0,0xd1,0xd2
1-2	len=Number of bytes (variable, see below)
3- (len-2)	ASCII listing of mission parameters Each EEPROM parameter has a 6 character name and 5 char value: NAMExx=vvvvv The = & signs are present in the listing of each parameter. (15 bytes/parameter) Successive parameters follow without gaps.

len-1 ; terminator at the end of the dump

An example showing only the initial 3 and final 2 elements follows:

PchSec= -1|MaxHrs= 1440|dBarGo= -1|...| XP2= 0| XPdly= 0|;

The EEPROM dump message is sent only in response to a command "P" from the ground station. It is sent over 3 SBD messages (0xd0=328 bytes, 0xd1=328 bytes, 0xd2=279 bytes).

Argo Data ID=0xf0 Relayed in normal cycles

Byte	Contents
0	ID/Mission phase = 0xf0
1-2	Number of bytes = 37 = 0x25
3	Data Version (Minor version in high order nibble, major version in low order)
4-5	Target profile depth
6-7	Target parking depth
8-9	Maximum rise time in minutes
10-11	Target (maximum) fall to parking depth time in minutes
12-13	Maximum fall-from-parking-to-profile-depth time in second
14-15	Target drift time (units=5 minutes; To convert to minutes multiply packed value by 5)
16	Float version (0 SOLOII)
17	Target ascent rate while profiling
18-19	Number of seeks
20-21	Surface Time
22-23	Seek Interval in minutes
24-25	Pressure scaling gain; db=counts/gain-offset
26-27	Pressure scaling offset
28-29	Temperature scaling gain; degreesC=counts/gain-offset
30-31	Temperature scaling offset
32-33	Salinity scaling gain; PSU=counts/gain-offset
34-35	Salinity scaling offset
36	; terminator

The values of Gain/Offset are now sent back within the Argo Metafile message (0xf0) for data decoding purposes allowing a way to determine what Gain/offset is used in a given cycle. Beginning with ROM SBE 29Oct2013, the GAIN/OFFSET of Temperature/Salinity/Pressure can be modified via 2-way communication. Modifying these parameters will effect all variables returned, except for a bug resulting in the shallowest engineering PTS triplet to be returned with the standard scaling under all circumstances. This was fixed in firmware 10Dec2013.

Test Data (ID=0xf1)

Byte	Contents
0	ID/Mission phase = 0xf1
1-2	Number of bytes = variable
3	modulo
4-n	test data