SBE SOLO DOCUMENTATION
(ROM version SBE524)

The following manual covers the algorithms required to post-process the ARGOS data from a SBE SOLO. This version is specific for ROM version SBE524 and later which allow the ARGO PTT ID to be either a 20 bits or 28 bits. The data message is necessarily reduced from 256 bits to 248 bits since ARGOS gets the extra 8 bits of ID information from the space previously allotted to the sensor data.

Section	**Contents**	**page**

Appendices

Rev. 1.3 : 13Nov,2001 Update Notes:

1) Clarified description of surface pressure data in profile and engineering messages. Updated to reflect changes due to resetting pressure offset in SBE 41CP if drifts become excessive.

Rev. 1.2 : 28May,2001 Update Notes:

1) Changed format of ARGOS data message to reduce length to 31 bytes so can use 28 bit ARGOS PTT Ids. The format type is **18** and it can be used with either 20 bit or 28 bit ARGOS PTT IDs.

Rev. 1.1 : 20 November, 2000 Update Notes:

1) Erratum regarding calculation of Savg2 (p. 7) is corrected.
2) Argos Diagnostic message is described in detail (p. 10).

1. Description of the typical SBE SOLO cycle

(Starting with the SOLO on the surface with the piston fully extended and the air sleeve bladder inflated).

1) The air valve is opened, emptying the air sleeve bladder, and the piston is retracted to its parked position determined from the last dive cycle.

1) The SOLO descends for a pre-programmed X hours. At the end of X hours it will have settled out at its neutral depth.
2) The SOLO does multiple seeks, comprised of moving the piston to get closer to the desired parked depth, and then waiting to settle out. The final piston position is then used as the starting point for the next cycle in step 1.
3) The SOLO waits for N hours (programmable) at its neutral depth.
   a) It measures P , T and S every Y hrs (programmable) during this period.
   b) It averages P , T and S for the first N/2 hrs (values Pavg1, Tavg1, Savg1).
   c) And averages P , T and S for the last N/2 hrs (Pavg2, Tavg2, Savg2).

d) The ascent profile is performed (the piston is fully extended and data sampled as the SOLO rises). If it is a SBE-CP (continuous-profiler), data are acquired at a 1 Hz rate, and then averaged over the set depth range bins. If it is a SBE-DP(discrete-profiler), P, T and S are acquired at the set depth points only. The ascent lasts a set time (programmable) to ensure SOLO gets to the surface.
e) The air sleeve bladder is inflated. This ensures the antenna is comfortably out of the water.
f) The data is further processed and compacted into the ARGOS messages.
g) SOLO transmits the ARGOS messages through a rotating buffer over a 24 hr period (also programmable).
h) The SOLO returns to 1) for the start of the next cycle.

2. Description of the Calibration file LOGxxxx (xxxx = SOLO S/N)

All of the pre-deployment calibration information is collated into the LOGxxxx file. A sample file is given here (10May00 refers to when the LOG file was written):

```
SN1108 ID 23457  -99 SB5.20  03Dec1998
MISSION 10May00    18   24.00  166.90
MULTCYC 10May00    10  166.9   20  100.0    15 200.0     800  -1   600
CALIBRT 10May00  0.5000   -10.0   0.065  0.26   0.027  0.03
THRMCAL 10May00    8.00    0.00 0.0 0.0 0.0 0.0
PROFORM 10May00      5    2    4   20   40
SBE_CAL 10May00      78  8.000    32000.00
END DATA
```

Each line has the following meaning:

```
SN1108     ID 23457     -99          SB5.20        03DEC1998
SOLO s/n  ARGOS ID   2nd ARGOS ID   ROM ver.    ROM creation date
```

```
MISSION 10MAY00    17   24.00  166.90
                   ID   TMSRF  TMCYCLE
```

`ID` = i.d. type.

`ID` = 18 = SBE SOLO 523 **(revised 18May2001)**

`TMSRF` = time spent on surface transmitting to ARGOS (hrs)

`TMCYCLE` = time to complete one cycle. If this is a multi-cycle float, the following line takes precedent:

```
MULTCYC 10May00    10  166.9   20  100.0    15 200.0     800  -1   600
                 CYC0  TIME0 CYC1  TIME1  CYC2 TIME2   PTAR1 DIR  PTAR2
```

See Section 4 for description of the multi-cycle variables.

```
CALIBRT 10May00  0.5000   -10.0   0.065  0.26   0.027  0.03
                 PGAIN    POFF   PMPGAIN PMPOFF CPUGAIN CPUOFF
```

where `xGAIN`, `xOFF` are the gain and offset calibration coefficients for the x sensor.
`x=P` refers to Pressure (dBar), `x=PMP` refers to the pump battery voltage (Volts), and `x=CPU` is the CPU battery voltage (Volts). To convert to engineering units:

`x = xCNTS * xGAIN + xOFF` (2.1)

where `xCNTS` are the a/d counts for the x sensor from the ARGOS message.

```
THRMCAL 10May00    8.00    0.00 0.0 0.0 0.0 0.0
                   TGAIN   TOFF  (…dummy variables…)
where T (degrees C) = (TCNTS * TGAIN  +  TOFF ) * 0.001
```
(2.2)

```
SBE_CAL 10May00      78  8.000    32000.00
                    s/n  SGAIN    SOFF
s/n is the SBE serial number
and S (PSU) = (SCNTS * SGAIN  +  SOFF ) * 0.001
```
(2.3)

```
PROFORM 10May00      5    2    4   20   40
                   BLOK  AV1  AV2  PB1  PB2
```

See Section 3 on how to convert the above to the profile depth bin values.

3. Converting the data bin # of the profile to pressure (dBar)

The profile is comprised of 56 bins, with varying resolution with depth. For instance, shallow bins are typically spaced 5 m apart, medium bins 10 m apart, and deep bins spaced 20-40 m apart. If the SBE-DP (discrete-profiler) is used, the CTD values are discrete samples at each of these depth values. If it is a SBE-CP (continuous-profiler) the data are averages centered on these depth values.

The depth bin parameters are found in the PROFORM line in the LOGxxxx file :

```
PROFORM 10May00       5    2    4   20   40
                   BLOK  AV1  AV2  PB1  PB2
```

BLOK = bin spacing for the shallow bins (I <= PB1)
AV1*BLOK = bin spacing for the medium bins (PB1 < I <= PB2)
AV2*BLOK = bin spacing for the deep bins (I > PB2)

For the above example,
there are 20 bins (=PB1) of 5 dBar spaced bins (=BLOK)
20 bins (=PB2-PB1) of 10 dBar spaced bins (=AV1*BLOK)
16 bins (=56 - PB2) of 20 dBar spaced bins (=AV2*BLOK)
giving a depth range of
20 bins from 0..100 dBar at 5 dBar resolution
20 bins from 100..300 dBar at 10 dBar resolution
16 bins from 300..620 dBar at 20 dBar resolution

The following is example fortran code to compute z(I)

```
      subroutine sbe_depth(z)
c ................................................................
c     ..compute P (dbars) over 56 bins using coeff. in common pro
      real z(56)
      common /pro/blok,av1,av2,pb1,pb2

c        ...compute depths
        do i=1,56
          if (i.le. pb1) then
            z(i) = i*blok
          else
            if (i.le. pb2) then
              z(i) = z(i-1) +  av1*blok
            else
              z(i) = z(i-1) +  av2*blok
            endif
          endif
        enddo

      return
      end
c ................................................................
```

4. Description of the Multiple Cycle Parameters

```
MULTCYC 10May00    10  166.9   20  100.0    15 200.0     800  -1   600
                 CYC0  TIME0 CYC1  TIME1  CYC2 TIME2   PTAR1 DIR  PTAR2

At the start of the mission the SOLO will do:
      CYC0 dives, each cycle taking TIME0 (hrs), seeking a depth of PTAR2.
      The above SOLO will do 10 dives, each dive taking 166.9 hrs,
      with the park depth of 600 dBar.

The SOLO will then alternate between CYC1 AND CYC2 for the remainder
of its life, i.e. in pseudo-code:
      do while(true)
            do CYC1 dives, taking TIME1 hrs for each dive, seeking PTAR1
            do CYC2 dives, taking TIME2 hrs for each dive, seeking PTAR2
      enddo
The above SOLO would first do 10 dives (CYC0), then alternate between:
      20 dives at 100.0 hrs each, seeking a park depth of 800 dBar, and
      15 dives at 200.0 hrs each, seeking a park depth of 600 dBar.

DIR   = profile direction.
      =  0 = profiles on the way down.
      = -1 = profiles on the way up.
This has no effect on the way the data are processed, but is included as a
descriptive parameter for the SOLO.
```

5. Description of the PTT messages and Header information

The PTT alternates between 4 messages, each containing 31 data bytes. Each message contains 3 bytes of header information, and 28 bytes of profile information. The format of the message depends upon whether the ARGOS ID is a 20 bit ID or the newer 28 bit ID.

IF the ID is a 28 bit ID, Systeme ARGOS processing shifts the 31 bytes of data following the 8 bit ID extension to the left by one byte and appends a 0x00 byte to the end. This removes the ID extension from the data stream and restores the length to 32 bytes. The first 4 bits of the stream contain a message i.d. tag which will be either 0x0, 0x1, 0x2, 0x3, or 0xF.

If the ID is a 20 bit ID, Systeme ARGOS processing returns 32 bytes of data. SOLO places 0xDB in the first byte of data, followed by 31 bytes of profile data. A valid message from a 28 bit ID can never have this byte in the 8 bit ID extension. Nor can a valid message ever have 0xDB in the first byte of profile data because 0xD is not a valid message i.d. tag. Thus, an initial 0xDB byte is an unambiguous indicator that the ARGOS ID has 20 bits. Before being processed, the succeeding 31 bytes should be shifted to the left by one byte, covering up the 0xDB flag, and an 0x00 byte appended to the end. This will put the data into the same format received from SOLOs with 28 bit IDs.

The discussion below assumes that data from 20 bit IDs have been shifted in this fashion. Character 1 refers to the first character in this message; character 32 is the last character and will always be 0x00.

For a SBE-SOLO, each message contains 14 data bins:
Message #(1, 2, 3, 4) contain bins (1-14, 15-28, 29-42, 43-56) respectively.

Header information

The first three bytes of each message contains information about the SOLO health. Since these three bytes are represented by 6 HEXADECIMAL characters in the ARGOS ASCII file, it is more appropriate to discuss the data in terms of characters .

Let a generic 12-bit value be represented by **ABC** where **A** is the most-significant character and **C** is the least significant. Also let **BC** represent a generic 8-bit value (i.e. pump voltage and cpu voltage)

In the following let *123456* denote character placement in the 6-character header.

Message #1:

```
char placement    1     234         56
generic data      0     ABC         BC
description       i.d.  Pavg1       Tavg2

i.d.= 0 = message i.d.
Pavg1 = average Press counts over the first half of the down time. Use
        Eqn. 2.1 to convert to dBars.
Tavg2 = the 8 lsb of avg T counts over the 2nd half of down time.
        The upper bits must be taken from Tavg1 (message 2), i.e.
        Tavg2 = BC + A(Tavg1)*256.  Use 2.2 to convert to deg. C.
        If abs(Tavg2-Tavg1) > 128, correct value to minimize difference.
```

Message #2:

```
char placement    1     234         56
generic data      1     ABC         BC
description i.d.  Tavg1       Pavg2

i.d.= 1 = message i.d.
Tavg1 = average T counts over the first half of the down time.
        Use 2.2 to convert to deg. C.
Pavg2 = the 8 lsb of avg P counts over the 2nd half of down time.
        The upper bits must be taken from Pavg1 (message 1), i.e.
        Pavg2 = BC + A(Pavg1) * 256.  Use 2.1 to convert to dBars.
        If abs(Pavg2-Pavg1) > 128, correct value to minimize difference.
```

Message #3:

```
char placement    1     234         56
generic data2     ABC         AB
description       i.d.  SPRX        Savg1

i.d.= 2 = message i.d.
SPRX = average P counts at the surface at the end of transmitting in the previous
        cycle.  This is before any resetting of the pressure offset.  See PFS in the
        diagnostic message for surface pressure after the offset correction.
        In first cycle after deployment, SPRX is the pressure reading taken dockside
        during self test before the offset correction.
Savg1 = average S counts over the first half of the down time.  The MS 8 bits are in
        this message.  All 16 bits of Savg1 are repeated in the Diagnostic message
        described in the Diagnostics Appendix.  Use 2.3 to convert to PSU.
```

Message #4:

```
char placement    1     2     34    56
generic data3     C     BC    BC
description i.d.  err   Imin  Bmax  (Rev 1.1 update)

i.d.= 3 = message i.d.
err = 4-bit error code, signifying a spurious interrupt,
      stack overflow, or spurious reset (see Diagnostics Appendix).
      err = 0 =no error.  Any other value indicates a problem of some sort.
Imin: (Imin+1)=minimum depth bin with valid data.  Typically Imin=0
      for normal operation.  If for some reason the SOLO is ascending
      very slowly, the profile may time out, in which case Imin>0,
      and should be flagged for further inspection.
Bmax= maximum depth bin with valid data.  Since only 56 data bins are
      transmitted, if Bmax>56, all depth bins have data.  If Bmax<56,
      then the last data bins (I>Bmax) should be flagged as invalid.
```

Diagnostic Message: Every 13th message sent by the SBE SOLO is a diagnostic. The first character of this message is an **F**. See Appendix A for information.

6. Unpacking and Rescaling the ARGOS profile data

In general, T and S data are processed in the SOLO for each ARGOS message in the following way:

1) The first data bin in the message is left with its full resolution.
2) The rest of the profile is first-differenced (i.e DT(i+1) = bin(i+1) - bin(i)).
3) The minimum and maximum values of DT are found (=DTmin and DTmax).
4) A LOOKUP table is used to find indices Kmin and Kmax such that:
   Scalar ∗ LOOKUP(Kmin) < DTmin (Scalar = 256 for T, 64 for S)
   Scalar ∗ LOOKUP(Kmax) DTmax
5) An offset and gain are computed as:
   OFF = LOOKUP(Kmin) ∗ Scalar
   GAIN = LOOKUP(Kmax) - LOOKUP(Kmin)
5) DT is rescaled to form the output array ODT = (DT - OFF)/GAIN
6) The data are then packed into the ARGOS message, and the process is repeated for the next message.

The LOOKUP Table has 16 entries, and is the same for both T and S:

LOOKUP(1..16) =
[-4 -2.5 -1.5 -1 -.75 -0.5 -0.25 0 0.25 0.5 0.75 1 1.5 2.5 4 6.25]

<u>For the SBE-SOLO</u> characters 7-64 of each ARGOS message denote:

```
char# 7     8     9     10    11    12    13-14....   39-40
data  KTmin KTmax KSmin KSmax TMSB2 TMSB1 TLSB(1)..   TLSB(14)

char# 41    42-44       45-47       ..60-62
data  SMSB1 SLSB(1,2)   SLSB(3,4)   ..SLSB(13,14)  SLSB are 6-bit values

char# 63-64
data  0x0000
```

KTmin and KTmax are indices into LOOKUP for the T data.
KSmin and KSmax are indices into LOOKUP for the S data.
TMSB1, TMSB2 are the most-significant bits for the first & last T bin in this message.
TLSB(i) i=2..14 are the rescaled T data for the 14 bins (8 bits per bin).
SMSB1 are the most-significant bits for the first S bin in the message.
SLSB(i) i=2..14 are the rescaled S data for the 14 bins (6 bits per bin). An easy way to unpack the 6-bit values is to read in 3 characters at a time and then split it into the two 6-bit values.

Algorithm to rescale either T or S :

Define Tscale = 256 (use for T), Sscale = 64 (use for S),
and substitute the correct value for Scale in the below.
Let nbins=14 = # bins in one message

1) compute gain : GAIN = LOOKUP(Kmax+1) - LOOKUP(Kmin+1)
2) compute offset: OFF = LOOKUP(Kmin+1) ∗ Scale
3) compute counts for the first bin :
    cnts(1) = MSB1 ∗ Scale + LSB(1)
4) compute counts for i=2..nbins
    cnts(i) = cnts(i-1) + LSB(i) ∗ GAIN + OFF
5) use 2.2 or 2.3 to convert from counts to engineering units.

NOTE index values of Kmin+1, Kmax+1 are used for LOOKUP. This is because the SOLO processor uses k=0 as the first index value into an array, while Fortran uses k=1.

A. Diagnostics Appendix

This appendix is to help interpret some of the diagnostic messages not fully explained in the main section.

err : this variable is sent back in the ARGOS message #4 and should equal zero. It is mainly used to flag interrupt service routines that should never happen. In general, err>0 signifies a CPU or programming problem. Non-zero values are:

	err	Source (unexpected interrupt from:)
	1	SCI serial System
	2	SPI serial system
	3	Pulse Accum. Input
	4	Pulse Accum. Overflow
	5	Timer Overflow
	6	Timer Output Compare
	7	Timer Input Capture
	8	Real Time Interrupt
	9	PACE timer overflow
A	10	XIRQ
B	11	Software Interrupt
C	12	Illegal Op Code
D	13	COP failure
E	14	Clock Monitor Failure
F	15	FORTH STACK NOT EMPTY

If any non-zero values are observed they should be reported.

SBE SOLO DIAGNOSTIC MESSAGE

Every 13th message transmitted by the SBE SOLO is a diagnostic, containing both discrete samples from the SBE and other engineering parameters. The following describes the 62 character message, where column 'Char'=character placement, with ‘4,5,6,7 signifying characters 4,5,6,and 7 comprise the 16 bits for parameter P1. The number in the last column refers to the corresponding stage that the datum was taken, as referenced by the outline in the *Description of the typical SBE SOLO cycl*e (p.2).

Char.	Name	Description	
1	id	Diagnostic message identifier = 'F'	
2	BST	4-bit status of miscellaneous operations (see below)	
3,4,5,6	P1	Pressure counts before the start of ascent.	5
7,8,9,10	T1	Temperature counts at same time as P1	5
11,12,13,14	S1	Salinity counts at same time as P1	5
15,16	P2	8 Lsbits of Pressure counts taken ~2 s after P1	5
17,18	T2	8 Lsbits of Temperature counts at same time as P2	5
19,20	S2	8 Lsbits of Salinity counts at same time as P2	5
21,22	P3	8 Lsbits of Pressure counts taken ~2 s after P2	5
23,24	T3	8 Lsbits of Temperature counts at same time as P3	5
25,26	S3	8 Lsbits of Salinity counts at same time as P3	5
27,28	Vcpu	CPU Voltage counts. Use 2.1 to convert to Volts	
29,30	Vpmp	Pump battery counts. Use 2.3 to convert to Volts	
31,32,33,34	Savg1	Av S counts over the first half of the down time. Use 2.3 to convert to PSU.	
35,36	DS	DS = signed 8 lsb of Savg2 - Savg1. If DS>127, DS = DS - 256 gives the correct signed value Savg2 = Savg1 + DS.	
37,38	num_bad	Number of bins in the profile with invalid data	
39,40,41	ATE	Air pressure inside of SOLO at end of last surface time	8
42,43,44	ATS	Air pressure inside of SOLO at start of last surface time	8
45,46,47	PFS	Pressure counts at the start of the SOLO Fall time after any reset of pressure offset that was required.In first cycle after deployment, PFS is the pressure reading taken dockside during self test after the offset correction.	1
48,49,50	PFE	Pressure counts at the end of the SOLO Fall time	2
51,52,53	PRE	Pressure counts at the end of the SOLO Rise time	5
54,55,56	TSK	TSK*2 = time (s) piston ran during first SEEK cycle	3
57,58,59	PSK	(signed) change in pressure during first SEEK cycle	3
60,61,62	TIP	TIP*2 = new time (s) to run piston in to get to SEEK depth	3

The 4 bits of BST (bit 0 = lsb, bit 3 = msb) are assigned:

bit 3 = Alow = 1 if the air bladder was refilled during the surface transmit time (not assigned for the first dive cycle, otherwise signifies a potential leaky air valve).

bit 2 = PQUEST = 1 if the pressure counts is questionable at the end of the Fall time, true if P counts puts us deeper than 2000 dBar OR shallower than Ptol, a variable set during final programming. If set, the SOLO does no SEEKing, and pulls the piston all of the way in.

bit 1 = OUT = 1 if the piston OUT limit switch is detected with the SOLO at the surface (normally OUT=1)

bit 0 = IN = 1 if the piston IN limit switch is detected at the start of ascent. This will depend upon profile direction: if profiling on ascent, then normally IN=1.

Discrete Profiler Note (e.g., SOLO 1166, 1168)

P3, T3, S3 are not valid in the above message, and their values should be ignored.