Am Luneort 15 a
27572 Bremerhaven
Germany

Operating Manual

NEMO
(Navigating European Marine Observer)

Part Number: 160 0 01 04 003

Initial Issue: June 26th, 2007

Revision No:	G00
Revision Date:	2009-05-27

THIS PAGE IS INTENTIONALLY LEFT BLANK

RECORD OF REVISIONS

This page lists the manual revisions. When you receive revision pages, replace the old pages with the new and log the information in this table.

Manual Title	Operating Manual, NEMO
Part Number	16000104003

REVISION NUMBER	**ISSUE DATE**	**CHANGE**	**INCORPORATION DATE**	**INCORPORATED BY**
E00	Sep 30/08			
F00	Feb 19/09			
G00	May 26/09			

THIS PAGE IS INTENTIONALLY LEFT BLANK

Table of Contents

Table of Figures

1 DEPLOYMENT

1.1 General

All NEMO-Floats are delivered in 'Mission Mode'. The power consumption in Mission Mode is extremely low. In Mission Mode, the menu and the mission profile can be initialized via the magnetic switch. The mission-startup contains a self-test and the transmission of test messages until first descent. The first descent begins six hours after reset of the float.

CAUTION: DO NOT DEPLOY THE FLOAT WITHOUT SUCCESSFUL COMPLETION OF THE START-UP PROCEDURE INCLUDING THE SUCCESSFUL TRANSMISSION OF THE TEST-MESSAGES.

CAUTION: DO NOT DEPLOY THE FLOAT IN HIGH-TRAFFIC ZONES.

DO NOT DEPLOY THE FLOAT IN REGIONS WITH WATERDEPTHS CLOSE OR LESS THAN THE PROFILE DEPTH + 200 m.

DO NOT DEPLOY THE FLOAT IN REGIONS WITH PHYSICAL PARAMETERS THAT ARE NOT SPECIFIED IN THE BALLASTING SHEETS.

CAUTION: FOR DEPLOYMENT IN ICE-COVERED REGIONS OR OTHER DEPLOYMENT MODES WHICH ARE NOT COVERED BY THIS MANUAL PLEASE CONTACT OPTIMARE IN ADVANCE FOR SPECIFIC INSTRUCTION..

The operator is warned and must accept and deal with the risks in order to use the NEMO-float.

Only trained operators shall deploy the instrument.

OPTIMARE disclaims liability for any consequences of incorrect deployment. Furthermore, warranty by OPTIMARE expires immediately upon the attempt to open the instrument.

1.2 Standard Start-up Procedure

1. Be aware of temperatures below 0 °C since this may cause freezing of the conductivity cell. Also do not place the float in the blazing sun otherwise it may be damaged.
2. Remove plugs and plastic bag which protects the CTD.

CAUTION: THIS IS VERY IMPORTANT SINCE THE CTD WILL NOT FUNCTION PROPERLY WITH THE PLUGS BEING STILL ATTACHED.

3. Reset the float by holding the magnet close to the float surface on and around the marked location (location marked with a cross; duration 2 seconds).
4. The internal vacuum and the battery voltages are tested. The vacuum is factory set at 10.5 psi (19 ^{0}C). Ten seconds after reset the air pump begins to run.
5. In case of failure of any of the tested parameters (limits: vacuum ≤ 12.7 psi; CPU battery ≥ 6.7 V; Pump battery ≥ 13.3 V) the float switches back to sleep mode.
6. During the next approximately 90 minutes, the oil bladder will be filled. Then it takes about 90 seconds for the air (sleeve) bladder to fill.

   ARGOS telemetry: The WildCAT PTT will transmit 6 times in specified transmission rate the 'ID number' before the bladders are filled. It is helpful to place the NEMO beeper close to the antenna to detect these transmissions.

   Iridium telemetry: No signal is available before the bladders are filled.
7. Check that both bladders are filled.
8. **ARGOS telemetry:** The WildCAT PTT will transmit the 'Test Message' until the mission starts (see 10.). It is helpful to place the NEMO beeper close to the antenna to detect these transmissions..
9. **Iridium telemetry:** After the oil and the sleeve bladders are filled, the float will try to receive a GPS signal. Depending on the signal strength this can vary between a few minutes or up to one hour. After the reception of the GPS signal, the 'Test Message' is sent via Iridium (SBD) every 30 minutes. This will continue until mission start (6 h after Reset).

CAUTION: IT IS IMPORTANT THAT THE FLOAT HAS UNOBSTRUCTED VIEW TO THE SKY IN ORDER TO MAXIMIZE THE SIGNAL STRENGTH FOR GPS AND IRIDIUM COMMUNICATIONS.

10. The mission starts six hours after the Reset of the float.
11. Reset with magnet during start-up procedure forces a return to step 4 including emptying the air or oil bladder. Six hours later the float will descend to park depth.
12. Complete interrupt of the start-up procedure has to be conducted via the terminal connection (procedure described in Chapter 10.1).

NOTE: A QUICK GUIDE CAN BE FOUND IN CHAPTER 12.

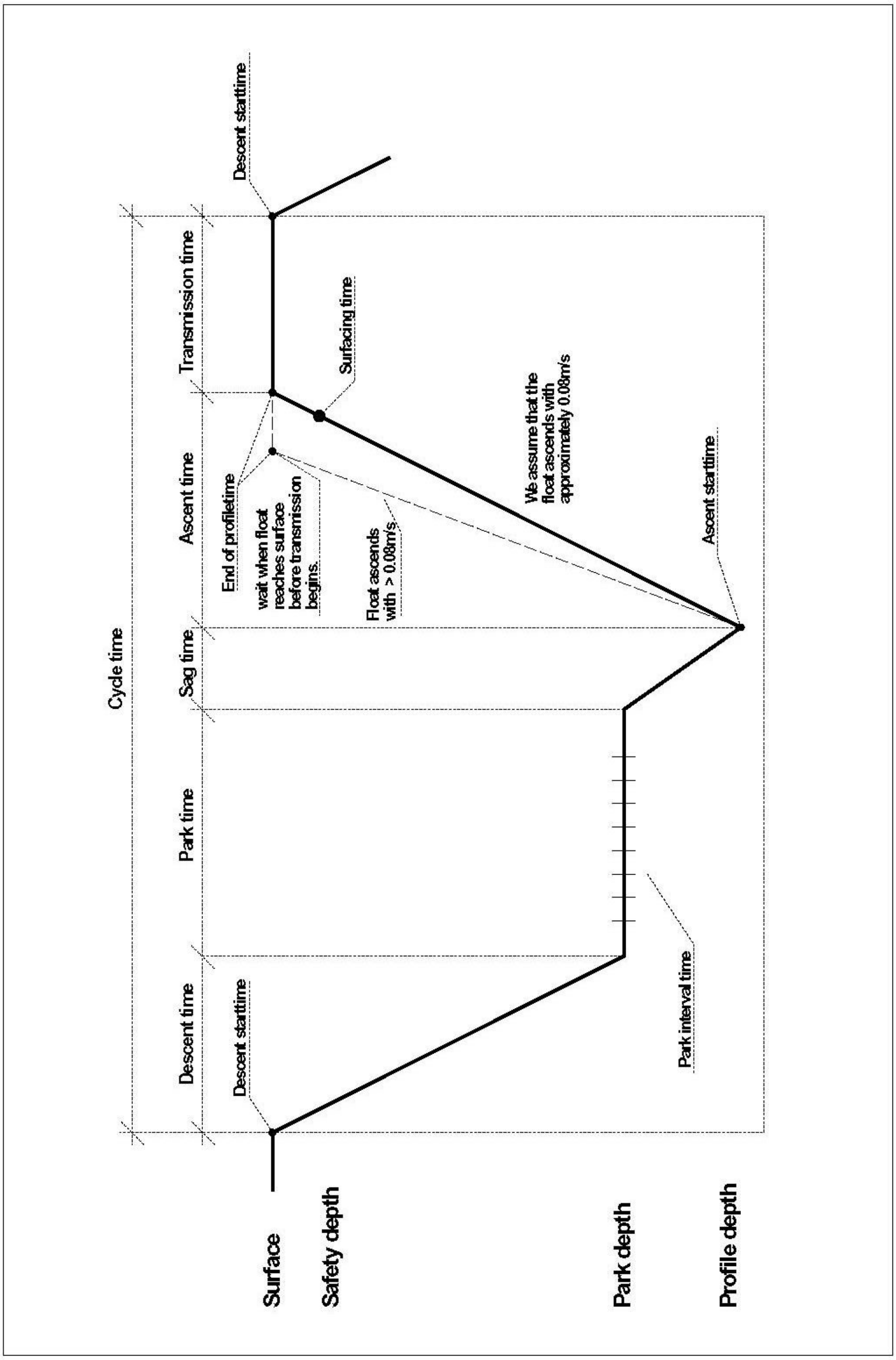
2 PARK-AND-PROFILE FEATURE

The NEMO-float is equipped with a park-and-profile feature. This feature allows the float to descend initially to a specified park depth (intermediate/shallower depth). At the end of the park mode, the float then descends to the specified target depth before the ascent to surface starts.

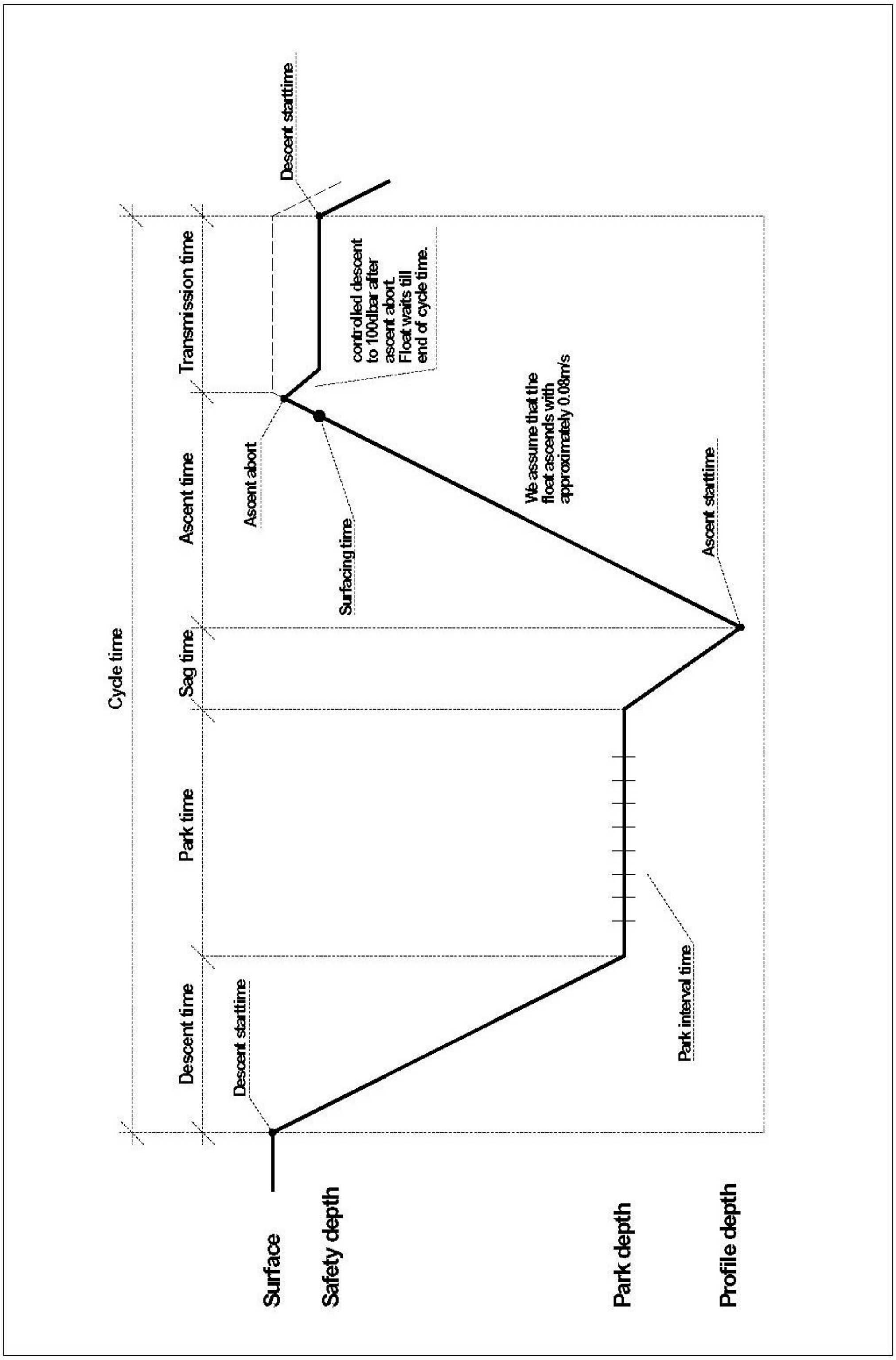
2.1 Charts

The following charts indicate several scenarios of the park-and-profile options:

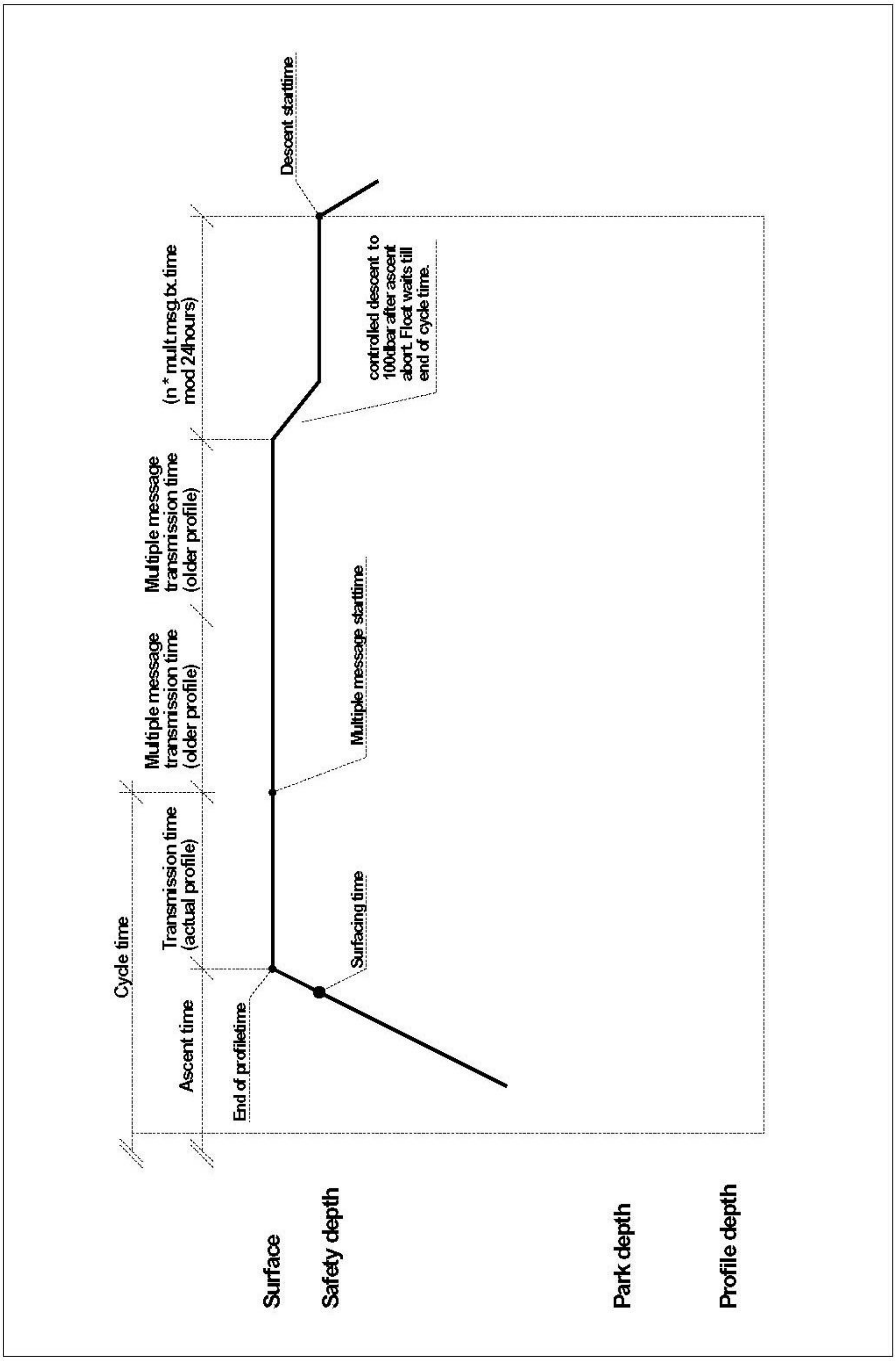
A. Regular profile: Regular ascent (solid line), fast ascent (broken line).



B. Aborted profile: Regular ascent (broken line), aborted ascent due to ice detection or water mass stratification (solid line).



C. Transmission of stored data: Regular ascent (broken line), aborted ascent due to ice or stratification (solid line).



3 ARGOS DATA PROCESSING

3.1 Mission Data

For software release see APP I; Example: Data transmission see 5.5.

The mission data of the NEMO float consist of the following main parts:

- Profile header
- Data set 1
- Data set 2
- Data set 3
- Data set 4
- Data set 5

The profile header is at the top of the mission data. It contains the basic information of the complete profile. There are important information of the profile and the different data sets. The profile header has a fixed length of 56 bytes (see chapter 3.2 for more details).

The five data sets have a variable length and contain the proper information of the profile. Each data set consists of a data record with a fixed length. The quantity of data records inside a data set is stored in the profile header. There can be maximal 65535 data records in one data set.

Currently only two of the five data sets are used for the profile data.

Data set 1: unused

Data set 2: CTD data during parking.

Data set 3: unused

Data set 4: CTD data during ascent.

Data set 5: unused

The length of the different data records is listed below.

Data record 1: unused

Data record 2: 6 bytes CTD data in sequence during parking.

- 2 bytes temperature in degrees with three decimal places (low_byte/high_byte).
- 2 bytes salinity in PSU with three decimal places (low_byte/high_byte).
- 2 bytes pressure in dbar with one decimal place (low_byte/high_byte).

Data record 3: unused

Data record 4: 6 bytes CTD data in sequence during parking.

- 2 bytes temperature in degrees with three decimal places (low_byte/high_byte).
- 2 bytes salinity in PSU with three decimal places (low_byte/high_byte).
- 2 bytes pressure in dbar with one decimal place (low_byte/high_byte).

Data record 5: unused

Table: Overview mission data.

Profile header (fixed length of 56 bytes)
Data set 1 (variable length) Data record 1 (fixed length) Data record 2 (fixed length) “ “ “ “ Data record 65535 (fixed length)
Data set 2 (variable length) Data record 1 (fixed length) Data record 2 (fixed length) “ “ “ “ Data record 65535 (fixed length)
Data set 3 (variable length) Data record 1 (fixed length) Data record 2 (fixed length) “ “ “ “ Data record 65535 (fixed length)
Data set 4 (variable length) Data record 1 (fixed length) Data record 2 (fixed length) “ “ “ “ Data record 65535 (fixed length)
Data set 5 (variable length) Data record 1 (fixed length) Data record 2 (fixed length) “ “ “ “ Data record 65535 (fixed length)

3.2 Format of Profile Header

NOTE: THE DATA STORAGE IS IN LITTLE ENDIAN FORMAT.

For 16Bit data (2 bytes) => low_byte / high_byte
For 32Bit data (4 bytes) => lowest_byte / low_byte / high_byte / highest_byte
The byte numeration begins here with '1'. In the source files the variables are stored in arrays and beginning at index '0'.

Profile header:
Byte #
01&02 Profile number, begins with 1 and increases by one for every float profile.
03&04 Serial number, identification of the float controller board.
05 Status byte, used for different flag information.
Bit 0 : ARGOS_ID => Used ID for transmission ('0'=>20bit / '1'=>28bit)
Bit 1 : MULTIPLE_MSG => Indicates the transmission of multiple messages.
Bit 2-7: UNUSED
06 RTC - year, immediately before transmission.
07 RTC - month, immediately before transmission.
08 RTC - day, immediately before transmission.
09 RTC - hour, immediately before transmission.
10 RTC - minute, immediately before transmission.
11 RTC – second, immediately before transmission
12 Ice detect count, increments by one for every time float aborts profile due to ice detection.
13&14 Piston position at surface.
15&16 Piston position in park depth.
17&18 Piston position in profile depth.
19&20 Piston position in at end of profile (end of ascent).
21 CPU battery voltage at surface.
22 Pump battery voltage in profile depth (during initialization of motor).
23 Hydraulic current during first ascent (during initialization of motor).
24 Internal tube pressure [PSI] at surface.
25&26 CTD-pressure at surface after ascent (5 digits, one decimal place, in dbar).
27&28 CTD-pressure at surface before descent (5 digits, one decimal place, in dbar).
29&30 Pressure at end of parking (median).
31 – 34 Descent start time, just before descent (in seconds).
35 – 38 Ascent start time, just before ascent (in seconds).
39 - 42 Surfacing time (at 100 dbar) (in seconds).
43 - 46 End of profile time (after last sample or abortion due to ice) (in seconds).
47&48 Quantity of data record 1 (max 65535).
49&50 Quantity of data record 2 (max 65535).
51&52 Quantity of data record 3 (max 65535).
53&54 Quantity of data record 4 (max 65535).
55&56 Quantity of data record 5 (max 65535).

3.3 Format of Test-Message

NOTE: THE DATA STORAGE IS IN LITTLE ENDIAN FORMAT.

For 16Bit data (2 bytes) => low_byte / high_byte
For 32Bit data (4 bytes) => lowest_byte / low_byte / high_byte / highest_byte
The byte numeration begins here with '1'. In the source files the variables are stored in arrays and beginning at index '0'.

TEST-MESSAGE:

Byte #
01 CRC, calculated CRC (bytes 2 to 32), for more information look at CRC routine.
02 Message number is always zero for test message.
03&04 Serial number, identification of the float controller board (low_byte/high_byte).
05 Software version year (in decimal).
06 Software version month (in decimal).
07 Software version day (in decimal).
08 Status byte, used for different flag information.
Bit 0 : ARGOS_ID => Used ID for transmission ('0'=>20bit / '1'=>28bit)
Bit 1 : MULTIPLE_MSG => Indicates the transmission of multiple messages.
Bit 2-7: unused
09 RTC - year info from float startup (reset).
10 RTC - month info from float startup (reset).
11 RTC - date info from float startup (reset).
12 RTC - hour info from float startup (reset).
13 RTC - minute info from float startup (reset).
14 RTC – second, immediately before transmission.
15&16 Max piston count value at min piston position.
17&18 Min piston count value at max piston position.
19 Actual CPU battery voltage during startup.
20 Actual Pump. battery voltage during startup.
21 Actual tube pressure during startup.
22&23 Specified profile pressure.
24&25 Specified park pressure.
26&27 Specified cycle time.
28&29 Specified transmission time.
30&31 Specified park interval time.
32 Fill byte reserved for 28bit ARGOS ID option (always zero).

3.4 Data Conversion

Battery voltage:

The battery voltage is measured with 10bit resolution. But only 8bit are used for the ARGOS message. Therefore the value is converted from 10bit to an 8bit value.

The 8bit value has the following format:

0-255 => 0-25.5V (e.g. value=143 => 14.3V)

(max. voltage is 25.5V / min. voltage is 0.1V)

Hydraulic current:

The hydraulic current is measured with 10bit resolution. but only 8bit are used for the ARGOS message. Therefore the value is converted from 10bit to an 8bit value.

The 8bit value has the following format:

0-255 => 0-2.55A (e.g. Value=80 => 0.80A)

(max. current is 2.55A / min. current is 0.01A)

Internal tube pressure:

The tube pressure is measured with 10bit resolution. but only 8bit are used for the ARGOS message. Therefore the value is converted from 10bit to an 8bit value.

The 8bit value has the following format:

0-255 => 0-25.5psi (e.g. value=112 => 11.2psi)

(max. voltage is 25.5psi / min. voltage is 0.1psi)

Piston position / piston count min, max:

The piston count value is measured with 10bit-resolution which is transmitted in 2 bytes. The number of piston counts depends on the mechanical configuration of the specific float.

CTD data:

Each CTD data (salinity, temperature and pressure) for park and profile are stored with two bytes.

CTD-temperature: 5 digits, three decimal places (1milli-degree resolution).

CTD-salinity: 5 digits, three decimal places (1milli-psu resolution).

CTD-pressure: 5 digits, one decimal place (10cm resolution).

	Hex-value	Dec-value	Converted	Unit
Temperature:	3EA6	16038	16,038	°C
Temperature*:	F58B	2677	-2,677	°C
Salinity:	8FDD	36829	36,829	PSU
Pressure:	1D4C	7500	750,0	dbar

*Note regarding negative temperatures (T °C < 0).

Positive temperature range is 0 to 62.535°C (0 to F447 hex). Negative temperature range is -0.001 to -3.000°C (FFFF to F448 hex). If(hex value) >= F447, then calculate FFFF – (hex value) + 1= Y_DEG.

Convert Y from hex to decimal, divide by 1000, and multiply by (-1), for degree C.

3.5 Checksum

During ARGOS transmission of the mission data, errors can occur. Therefore the first byte of each 32byte message block contains a Cyclic Redundancy Check (CRC). This CRC (byte 1) is calculated as a function of the message content (byte 2 to 32).

The 8bit CRC calculation is accomplished by the function below.

```
// 8bit CRC look-up-table
// ----------------------
unsigned int8 const crc_table[256] = {
        0, 94,188,226, 97, 63,221,131,194,156,126, 32,163,253, 31, 65,
      157,195, 33,127,252,162, 64, 30, 95,  1,227,189, 62, 96,130,220,
       35,125,159,193, 66, 28,254,160,225,191, 93,  3,128,222, 60, 98,
      190,224,  2, 92,223,129, 99, 61,124, 34,192,158, 29, 67,161,255,
       70, 24,250,164, 39,121,155,197,132,218, 56,102,229,187, 89,  7,
      219,133,103, 57,186,228,  6, 88, 25, 71,165,251,120, 38,196,154,
      101, 59,217,135,  4, 90,184,230,167,249, 27, 69,198,152,122, 36,
      248,166, 68, 26,153,199, 37,123, 58,100,134,216, 91,  5,231,185,
      140,210, 48,110,237,179, 81, 15, 78, 16,242,172, 47,113,147,205,
       17, 79,173,243,112, 46,204,146,211,141,111, 49,178,236, 14, 80,
      175,241, 19, 77,206,144,114, 44,109, 51,209,143, 12, 82,176,238,
       50,108,142,208, 83, 13,239,177,240,174, 76, 18,145,207, 45,115,
      202,148,118, 40,171,245, 23, 73,  8, 86,180,234,105, 55,213,139,
       87,  9,235,181, 54,104,138,212,149,203, 41,119,244,170, 72, 22,
      233,183, 85, 11,136,214, 52,106, 43,117,151,201, 74, 20,246,168,
      116, 42,200,150, 21, 75,169,247,182,232, 10, 84,215,137,107, 53
};
// This routine calculates the checksum of the ARGOS data message.
// The checksum is calculated from a 'look-up-table' (see above).
// The return value is the 8bit checksum.
unsigned int8 wildcat_get_crc(unsigned int8 *data, unsigned int8
array_size) {
      unsigned int8 index;    // Variable for loop operation.
      unsigned int8 crc;      // Variable for checksum calculation.
      crc = 0;

      for (index=1; index<array_size; index++)
            crc = crc_table[crc ^ data[index]];
      return(crc);
}
```

How to use the CRC:

- calculate the CRC value for each message (byte2-32).

- compare the calculated CRC to the transmitted CRC (byte 1).

- if the calculated and transmitted CRC bytes are equal, the 32byte ARGOS message is ok.

- Otherwise one or more bytes of the message are corrupted. In this case the message should be deleted and not be used for further data processing.

4 ARGOS DATA PROCESSING INCLUDING RAFOS

4.1 Mission Data

For software release see APP I; Example: Data transmission see 5.5.

The mission data of the NEMO float consist of the following main parts:

- Profile header
- Data set 1
- Data set 2
- Data set 3
- Data set 4
- Data set 5

The profile header is at the top of the mission data. It contains the basic information of the complete profile. There are important information of the profile and the different data sets. The profile header has a fixed length of 56 bytes (see chapter 4.2 for more details).

The five data sets have a variable length and contain the proper information of the profile. Each data set consists of a data record with a fixed length. The quantity of data records inside a data set is stored in the profile header. There can be maximal 65535 data records in one data set.

Currently only two of the five data sets are used for the profile data.

Data set 1: unused

Data set 2: RAFOS and CTD data during parking.

Data set 3: unused

Data set 4: CTD data during ascent.

Data set 5: unused

The length of the different data records is listed below.

Data record 1: unused

Data record 2: 24 bytes RAFOS and CTD data in sequence during parking.

- 1 byte RAFOS amplitude 1.
- 2 bytes RAFOS rank 1 (low_byte/high_byte).
- 1 byte RAFOS amplitude 2.
- 2 bytes RAFOS rank 2 (low_byte/high_byte).
- 1 byte RAFOS amplitude 3.
- 2 bytes RAFOS rank 3 (low_byte/high_byte).
- 1 byte RAFOS amplitude 4.
- 2 bytes RAFOS rank 4 (low_byte/high_byte).
- 1 byte RAFOS amplitude 5.

- 2 bytes RAFOS rank 5 (low_byte/high_byte).
- 1 byte RAFOS amplitude 6.
- 2 bytes RAFOS rank 6 (low_byte/high_byte).

- 2 bytes temperature in degrees with three decimal places (low_byte/high_byte).
- 2 bytes salinity in PSU with three decimal places (low_byte/high_byte).
- 2 bytes pressure in dbar with one decimal place (low_byte/high_byte).

Data record 3: unused

Data record 4: 6 bytes CTD data in sequence during parking.

- 2 bytes temperature in degrees with three decimal places (low_byte/high_byte).
- 2 bytes salinity in PSU with three decimal places (low_byte/high_byte).
- 2 bytes pressure in dbar with one decimal place (low_byte/high_byte).

Data record 5: unused

Table: Overview mission data.

Profile header (fixed length of 55 bytes)	
Data set 1 (variable length)	
	Data record 1 (fixed length)
	Data record 2 (fixed length)
	“ “
	“ “
	Data record 65535 (fixed length)
Data set 2 (variable length)	
	Data record 1 (fixed length)
	Data record 2 (fixed length)
	“ “
	“ “
	Data record 65535 (fixed length)
Data set 3 (variable length)	
	Data record 1 (fixed length)
	Data record 2 (fixed length)
	“ “
	“ “
	Data record 65535 (fixed length)
Data set 4 (variable length)	
	Data record 1 (fixed length)
	Data record 2 (fixed length)
	“ “
	“ “
	Data record 65535 (fixed length)
Data set 5 (variable length)	
	Data record 1 (fixed length)
	Data record 2 (fixed length)
	“ “
	“ “
	Data record 65535 (fixed length)

4.2 Format of Profile Header

NOTE: THE DATA STORAGE IS IN LITTLE ENDIAN FORMAT.

For 16Bit data (2 bytes) => low_byte / high_byte
For 32Bit data (4 bytes) => lowest_byte / low_byte / high_byte / highest_byte
The byte numeration begins here with '1'. In the source files the variables are stored in arrays and beginning at index '0'.

Profile header:
Byte #
01&02 Profile number, begins with 1 and increases by one for every float profile.
03&04 Serial number, identification of the float controller board.
05 Status byte, used for different flag information.
Bit 0 : ARGOS_ID => Used ID for transmission ('0'=>20bit / '1'=>28bit)
Bit 1 : MULTIPLE_MSG => Indicates the transmission of multiple messages.
Bit 2-7 : UNUSED
06 RTC - year, immediately before transmission.
07 RTC - month, immediately before transmission.
08 RTC - day, immediately before transmission.
09 RTC - hour, immediately before transmission.
10 RTC - minute, immediately before transmission.
11 RTC - second, immediately before transmission.
12 Ice detect count, increments by one for every time float aborts profile due to ice detection.
13&14 Piston position at surface.
15&16 Piston position in park depth.
17&18 Piston position in profile depth.
19&20 Piston position in at end of profile (end of ascent).
21 CPU battery voltage at surface.
22 Pump battery voltage in profile depth (during initialization of motor).
23 Hydraulic current during first ascent (during initialization of motor).
24 Internal tube pressure [PSI] at surface.
25&26 CTD-pressure at surface after ascent (5 digits, one decimal place, in dbar).
27&28 CTD-pressure at surface before descent (5 digits, one decimal place, in dbar).
29&30 Pressure at end of parking (median).
31 - 34 Descent start time, just before descent (in seconds).
35 - 38 Ascent start time, just before ascent (in seconds).
39 - 42 Surfacing time (at 100 dbar) (in seconds).
43 - 46 End of profile time (after last sample or abortion due to ice) (in seconds).
47&48 Quantity of data record 1 (max 65535).
49&50 Quantity of data record 2 (max 65535).
51&52 Quantity of data record 3 (max 65535).
53&54 Quantity of data record 4 (max 65535).
55&56 Quantity of data record 5 (max 65535).

4.3 Format of Test-Message

NOTE: THE DATA STORAGE IS IN LITTLE ENDIAN FORMAT.

For 16Bit data (2 bytes) => low_byte / high_byte
For 32Bit data (4 bytes) => lowest_byte / low_byte / high_byte / highest_byte
The byte numeration begins here with '1'. In the source files the variables are stored in arrays and beginning at index '0'.

TEST-MESSAGE:

Byte #
01 CRC, calculated CRC (bytes 2 to 32), for more information look at CRC routine.
02 Message number is always zero for test message.
03&04 Serial number, identification of the float controller board (low_byte/high_byte).
05 Software version year (in decimal).
06 Software version month (in decimal).
07 Software version day (in decimal).
08 Status byte, used for different flag information.
Bit 0 : ARGOS_ID => Used ID for transmission ('0'=>20bit / '1'=>28bit)
Bit 1 : MULTIPLE_MSG => Indicates the transmission of multiple
Bit 2-7: unused
09 RTC - year info from float startup (reset).
10 RTC - month info from float startup (reset).
11 RTC - date info from float startup (reset).
12 RTC - hour info from float startup (reset).
13 RTC - minute info from float startup (reset).
14 RTC - second info from float startup (reset).
15&16 Max piston count value at min piston position.
17&18 Min piston count value at max piston position.
19 Actual CPU battery voltage during startup.
20 Actual Pump. battery voltage during startup.
21 Actual tube pressure during startup.
22&23 Specified profile pressure.
24&25 Specified park pressure.
26&27 Specified cycle time.
28&29 Specified transmission time.
30&31 Specified park interval time.
32 Fill byte reserved for 28bit ARGOS ID option (always zero).

4.4 Data Conversion

Battery voltage:

The battery voltage is measured with 10bit resolution. But only 8bit are used for the ARGOS message. Therefore the value is converted from 10bit to an 8bit value.

The 8bit value has the following format:

0-255 => 0-25.5V (e.g. value=143 => 14.3V)

(max. voltage is 25.5V / min. voltage is 0.1V)

Hydraulic current:

The hydraulic current is measured with 10bit resolution. but only 8bit are used for the ARGOS message. Therefore the value is converted from 10bit to an 8bit value.

The 8bit value has the following format:

0-255 => 0-2.55A (e.g. Value=80 => 0.80A)

(max. current is 2.55A / min. current is 0.01A)

Internal tube pressure:

The tube pressure is measured with 10bit resolution. but only 8bit are used for the ARGOS message. Therefore the value is converted from 10bit to an 8bit value.

The 8bit value has the following format:

0-255 => 0-25.5psi (e.g. value=112 => 11.2psi)

(max. voltage is 25.5psi / min. voltage is 0.1psi)

Piston position / piston count min, max:

The piston count value is measured with 10bit-resolution which is transmitted in 2 bytes. The number of piston counts depends on the mechanical configuration of the specific float.

CTD data:

All CTD data (salinity, temperature and pressure) for park and profile are stored with two bytes.

CTD-temperature: 5 digits, three decimal places (1milli-degree resolution).

CTD-salinity: 5 digits, three decimal places (1milli-psu resolution).

CTD-pressure: 5 digits, one decimal place (10cm resolution).

Table: Example for value conversions.

	Hex-value	*Dec-value*	*Converted*	*Unit*
Temperature:	3EA6	16038	16,038	°C
Temperature*:	F58B	2677	-2,677	°C
Salinity:	8FDD	36829	36,829	PSU
Pressure:	1D4C	7500	750,0	dbar

*Note regarding negative temperatures (T °C < 0).

Positive temperature range is 0 to 62.535°C (0 to F447 hex).

Negative temperature range is -0.001 to -3.000°C (FFFF to F448 hex).

If(hex value) >= F447, then calculate FFFF – (hex value) + 1= Y_DEG.

Convert Y from hex to decimal, divide by 1000, and multiply by (-1), for degree C.

RAFOS data:

RAFOS amplitude: Amplitude of the correlation (8 bit value). The max value is $F4_{HEX}$ (244_{DEC}).

RAFOS rank: Number of signal samples in interval of 0.3075 seconds (16 bit value).

All values of RAFOS amplitude and RAFOS rank are HEX-values.

The RAFOS receiver can collect six correlations (RAFOS amplitude1-6 / RAFOS rank1-6).

The RAFOS values are ordered by decreasing magnitudes of the amplitude.

The runtime of the acoustic signal since start of the RAFOS receiver can be calculated with following formula: runtime = RAFOS rank * 0.3075 seconds

The RAFOS sound sources are sending the acoustic signal for 80 seconds three times, 00:30 am; 01:00 am and at 01:30 am.

The RAFOS receiver in the float is activated at 00:25 am and stops at 01:55 am.

4.5 Checksum

During ARGOS transmission of the mission data, errors can occur. Therefore the first byte of each 32byte message block contains a Cyclic Redundancy Check (CRC). This CRC (byte 1) is calculated as a function of the message content (byte 2 to 32).

The 8bit CRC calculation is accomplished by the function below.

```
// 8bit CRC look-up-table
// ----------------------
unsigned int8 const crc_table[256] = {
        0, 94,188,226, 97, 63,221,131,194,156,126, 32,163,253, 31, 65,
      157,195, 33,127,252,162, 64, 30, 95,  1,227,189, 62, 96,130,220,
       35,125,159,193, 66, 28,254,160,225,191, 93,  3,128,222, 60, 98,
      190,224,  2, 92,223,129, 99, 61,124, 34,192,158, 29, 67,161,255,
       70, 24,250,164, 39,121,155,197,132,218, 56,102,229,187, 89,  7,
      219,133,103, 57,186,228,  6, 88, 25, 71,165,251,120, 38,196,154,
      101, 59,217,135,  4, 90,184,230,167,249, 27, 69,198,152,122, 36,
      248,166, 68, 26,153,199, 37,123, 58,100,134,216, 91,  5,231,185,
      140,210, 48,110,237,179, 81, 15, 78, 16,242,172, 47,113,147,205,
       17, 79,173,243,112, 46,204,146,211,141,111, 49,178,236, 14, 80,
      175,241, 19, 77,206,144,114, 44,109, 51,209,143, 12, 82,176,238,
       50,108,142,208, 83, 13,239,177,240,174, 76, 18,145,207, 45,115,
      202,148,118, 40,171,245, 23, 73,  8, 86,180,234,105, 55,213,139,
       87,  9,235,181, 54,104,138,212,149,203, 41,119,244,170, 72, 22,
      233,183, 85, 11,136,214, 52,106, 43,117,151,201, 74, 20,246,168,
      116, 42,200,150, 21, 75,169,247,182,232, 10, 84,215,137,107, 53
};
// This routine calculates the checksum of the ARGOS data message.
// The checksum is calculated from a 'look-up-table' (see above).
// The return value is the 8bit checksum.
unsigned int8 wildcat_get_crc(unsigned int8 *data, unsigned int8 array_size) {
      unsigned int8 index;    // Variable for loop operation.
      unsigned int8 crc;      // Variable for checksum calculation.
      crc = 0;

      for (index=1; index<array_size; index++)
            crc = crc_table[crc ^ data[index]];
      return(crc);
}
```

How to use the CRC:
- calculate the CRC value for each message (byte2-32).
- compare the calculated CRC to the transmitted CRC (byte 1).
- if the calculated and transmitted CRC bytes are equal, the 32byte ARGOS message is ok.
- Otherwise one or more bytes of the message are corrupted. In this case the message should be deleted and not be used for further data processing.

5 ARGOS TRANSMISSION

5.1 ARGOS Message and ARGOS ID Format

Data is sent via the ARGOS WildCAT in message blocks with fixed length. The length of the message block depends on the used ARGOS ID. There are two different ARGOS IDs, the 20bit and the 28bit ID. For 20bit IDs the message block length is 32bytes. For 28bit IDs the length is 31 byte.

The used mode is defined in the NEMO setup parameters.

5.2 Transmission of Test Message

The test message includes all important information of the float (setup-parameters, start-up data).

The test message is only sent during start-up (self test). The format of the test messages are described in corresponding chapter. The length of the message is set to 32 bytes.

5.3 Transmission of Profile Data (single profile)

The single profile data are transmitted in the following format:

Byte #1 CRC

Byte #2 Message number

Bytes #3 to 32 Profile data (20bit ID)

alternatively Bytes #3 to 31 Profile data (28bit ID)

5.4 Transmission of Profile Data (long-term data storage)

The long-term storage profile data are transmitted in the following format:

Byte #1 CRC

Byte #2 Message number

Byte #3 Profile number

Bytes #4 to 32 Profile data (20bit ID)

alternatively Bytes #4 to 31 Profile data (28bit ID)

5.5 Example: Data Transmission

Data are stored on the memory card as described under section 3.1.

They are stored without message number and CRC.

Before the data are transmitted, the first 30 bytes (1-30) of the memory card are read and the first message number (#1) is added before the 30 bytes. Now the total length of the message is 31 bytes. The checksum calculated from all 31 bytes and the result added again as the 1st byte of a now total of 32 bytes. A total of 32 bytes is then transmitted.

In the next step bytes (31 to 60 bytes) are read from the memory card. #2 is added in byte number two as a message number, the CRC is calculated and added in byte number one and the data with 32 bytes are transmitted.

The process continues until all data are transmitted from the memory card to the receiving station. The last message is filled up with FF (or 255) to a total of 32 bytes.

Once all data are transmitted the process starts from the beginning with the creation of message #1.

Message numbers and CRCs are created for every sent message and not stored on the memory card and the first 2 bytes must be removed during the reassembly of the complete data set with the data structure described in this manual (first 56 bytes of header followed by data).

6 IRIDIUM DATA PROCESSING INCLUDING O_2-SENSOR

6.1 Mission Data

For software release see APP I; Example: Data transmission see 8.5.

The mission data of the NEMO float consist of the following main parts:

- Profile header
- Data set 1
- Data set 2
- Data set 3
- Data set 4
- Data set 5

The profile header is at the top of the mission data. It contains the basic information of the complete profile. There are important information of the profile and the different data sets. The profile header has a fixed length of 56 bytes (see chapter 6.2 for more details).

The five data sets have a variable length and contain the proper information of the profile. Each data set consists of a data record with a fixed length. The quantity of data records inside a data set is stored in the profile header. There can be maximal 65535 data records in one data set.

Currently only two of the five data sets are used for the profile data.

Data set 1: unused

Data set 2: unused

Data set 3: unused

Data set 4: O_2-Sensor and CTD data during ascent

Data set 5: GPS data at surface

The length of the different data records is listed below.

Data record 1: unused

Data record 2: unused

Data record 3: unused

Data record 4: 10 bytes O2-sensor data and CTD data in sequence during parking.

- 2 bytes O2-oxygen data in µM with two decimal places (low_byte/high_byte).
- 2 bytes O2-temperature data in degrees with two decimal places (low_byte/high_byte).
- 2 bytes temperature in degrees with three decimal places (low_byte/high_byte).
- 2 bytes salinity in PSU with three decimal places (low_byte/high_byte).
- 2 bytes pressure in dbar with one decimal place (low_byte/high_byte).

Data record 5: 14 bytes GPS data at surface

- 1 byte GPS year
- 1 byte GPS month
- 1 byte GPS day
- 1 byte GPS hour
- 1 byte GPS minute
- 1 byte GPS second
- 4 bytes GPS latitude in degrees with seven decimal places (lowest_byte/low_byte/high_byte/highest_byte).
- 4 bytes GPS longitude in degrees with seven decimal places (lowest_byte/low_byte/high_byte/highest_byte).

Table: Overview mission data.

Profile header (fixed length of 56 bytes)
Data set 1 (variable length) Data record 1 (fixed length) Data record 2 (fixed length) “ “ “ “ Data record 65535 (fixed length)
Data set 2 (variable length) Data record 1 (fixed length) Data record 2 (fixed length) “ “ “ “ Data record 65535 (fixed length)
Data set 3 (variable length) Data record 1 (fixed length) Data record 2 (fixed length) “ “ “ “ Data record 65535 (fixed length)
Data set 4 (variable length) Data record 1 (fixed length) Data record 2 (fixed length) “ “ “ “ Data record 65535 (fixed length)
Data set 5 (variable length) Data record 1 (fixed length) Data record 2 (fixed length) “ “ “ “ Data record 65535 (fixed length)

6.2 Format of Profile Header

NOTE: THE DATA STORAGE IS IN LITTLE ENDIAN FORMAT.

For 16Bit data (2 bytes) => low_byte / high_byte
For 32Bit data (4 bytes) => lowest_byte / low_byte / high_byte / highest_byte
The byte numeration begins here with '1'. In the source files the variables are stored in arrays and beginning at index '0'.

Profile header:
Byte #
01&02 Profile number, begins with 1 and increases by one for every float profile.
03&04 Serial number, identification of the float controller board.
05 Status byte, used for different flag information.
Bit 0 : MULTIPLE_MSG => Indicates the transmission of multiple messages.
Bit 1 : ARGOS_ID => Used ID for transmission ('0'=>20bit / '1'=>28bit)
Bit 2-7 : UNUSED
06 RTC - year, immediately before Iridium transmission.
07 RTC - month, immediately before Iridium transmission.
08 RTC - day, immediately before Iridium transmission.
09 RTC - hour, immediately before Iridium transmission.
10 RTC - minute, immediately before Iridium transmission.
11 RTC - second, before Iridium transmission.
12 Ice detect count, increments by one for every time float aborts profile due to ice detection.
13&14 Piston position at surface.
15&16 Piston position in park depth.
17&18 Piston position in profile depth.
19&20 Piston position in at end of profile (end of ascent).
21 CPU battery voltage at surface.
22 Pump battery voltage in profile depth (during initialization of motor).
23 Hydraulic current during first ascent (during initialization of motor).
24 Internal tube pressure [PSI] at surface.
25&26 CTD-pressure at surface after ascent (5 digits, one decimal place, in dbar).
27&28 CTD-pressure at surface before descent (5 digits, one decimal place, in dbar).
29&30 Pressure at end of parking (median).
31 - 34 Descent start time, just before descent (in seconds).
35 - 38 Ascent start time, just before ascent (in seconds).
39 - 42 Surfacing time (at 100 dbar) (in seconds).
43 - 46 End of profile time (after last sample or abortion due to ice) (in seconds).
47&48 Quantity of data record 1 (max 65535).
49&50 Quantity of data record 2 (max 65535).
51&52 Quantity of data record 3 (max 65535).
53&54 Quantity of data record 4 (max 65535).
55&56 Quantity of data record 5 (max 65535).

6.3 Format of Test-Message

NOTE: THE DATA STORAGE IS IN LITTLE ENDEAN FORMAT.

For 16Bit data (2 bytes) => low_byte / high_byte
For 32Bit data (4 bytes) => lowest_byte / low_byte / high_byte / highest_byte
The byte numeration begins here with '1'. In the source files the variables are stored in arrays and beginning at index '0'.

TEST-MESSAGE:
Byte #
01 CRC, calculated CRC (bytes 2 to 32), for more information look at CRC routine.
02 Message number is always zero for test message.
03&04 Serial number, identification of the float controller board (low_byte/high_byte).
05 Software version year (in decimal).
06 Software version month (in decimal).
07 Software version day (in decimal).
08 Status byte, used for different flag information.
Bit 0 : MULTIPLE_MSG => Indicates the transmission of multiple messages.
Bit 1 : ARGOS_ID => Used ID for transmission ('0'=>20bit / '1'=>28bit)
Bit 2-7: unused
09 RTC - year info from float startup (reset).
10 RTC - month info from float startup (reset).
11 RTC - date info from float startup (reset).
12 RTC - hour info from float startup (reset).
13 RTC - minute info from float startup (reset).
14 RTC - second info from float startup (reset).
15&16 Max piston count value at min piston position.
17&18 Min piston count value at max piston position.
19 Actual CPU battery voltage during startup.
20 Actual pump battery voltage during startup.
21 Actual tube pressure during startup.
22&23 Specified profile pressure.
24&25 Specified park pressure.
26&27 Specified cycle time.
28&29 Specified transmission time.
30&31 Specified park interval time.
32 GPS year.
33 GPS month.
34 GPS day.
35 GPS hour.
36 GPS minute.
37 GPS seconds.
38-41 GPS latitude.
42-45 GPS longitude.

6.4 Data Conversion

Battery voltage:

The battery voltage is measured with 10bit resolution. But only 8bit are used and transmitted.

Therefore the value is converted from 10bit to an 8bit value.

The 8bit value has the following format:

0-255 => 0-25.5V (e.g. value=143 => 14.3V)

(max. voltage is 25.5V / min. voltage is 0.1V)

Hydraulic current:

The hydraulic current is measured with 10bit resolution. but only 8bit are used and transmitted.

Therefore the value is converted from 10bit to an 8bit value.

The 8bit value has the following format:

0-255 => 0-2.55A (e.g. Value=80 => 0.80A)

(max. current is 2.55A / min. current is 0.01A)

Internal tube pressure:

The tube pressure is measured with 10bit resolution. but only 8bit are used and transmitted.

Therefore the value is converted from 10bit to an 8bit value.

The 8bit value has the following format:

0-255 => 0-25.5psi (e.g. value=112 => 11.2psi)

(max. pressure is 25.5psi / min. pressure is 0.1psi)

Piston position / piston count min, max:

The piston count value is measured with 10bit-resolution which is transmitted in 2 bytes. The number of piston counts depends on the mechanical configuration of the specific float.

CTD data:

All CTD data (salinity, temperature and pressure) for park and profile are stored with two bytes.

CTD-temperature: 5 digits, three decimal places (1milli-degree resolution).

CTD-salinity: 5 digits, three decimal places (1milli-psu resolution).

CTD-pressure: 5 digits, one decimal place (10cm resolution).

Table: Example for value conversions of CTD-data.

	Hex-value	Dec-value	Converted	Unit
Temperature:	3EA6	16038	16,038	°C
Temperature*:	F58B	2677	-2,677	°C
Salinity:	8FDD	36829	36,829	PSU
Pressure:	1D4C	7500	750,0	dbar

*Note regarding negative temperatures (T °C < 0).

Positive temperature range is 0 to 62.535°C (0 to F447 hex).

Negative temperature range is -0.001 to -3.000°C (FFFF to F448 hex).

If(hex value) >= F447, then calculate FFFF – (hex value) + 1= Y_DEG.

Convert Y from hex to decimal, divide by 1000, and multiply by (-1), for degree C.

O_2-Sensor data:

O_2 oxygen data: 16 bit value with three decimal places.

O_2 temperature data: 16 bit value with three decimal places.

Table: Example for value conversions of O2-Sensor data.

	Hex-value	Dec-value	Converted	Unit
O_2 oxygen data:	69E1	27105	271,05	µM
O_2 temperature:	0762	1890	18,90	°C

GPS data:

GPS time and date information is stored with one byte each.

GPS geographic position information (latitude/longitude) is stored with 4 bytes each.

Table: Example for value conversions of GPS data.

	Hex-value	Dec-value	Converted	Unit
GPS year:	06	6	6	
GPS month:	09	9	9	
GPS day:	18	24	24	
GPS hour:	7	7	7	
GPS minute:	20	32	32	
GPS second:	1E	30	30	
GPS latitude:	1FE3E461	535028833	53,5028833	degree
GPS longitude:	051C2F2E	85733166	8,5733166	degree

6.5 Checksum

Every Iridium SBD message contains a checksum. This assures additional data integrity during Iridium transmissions.

Therefore the first byte of each SBD message block contains a Cyclic Redundancy Check (CRC). This CRC (byte 1) is calculated as a function of the message content (byte 2 to message end).

The 8-bit CRC is calculated using the function below.

```
// 8bit CRC look-up-table
// ----------------------
unsigned int8 const crc_table[256] = {
        0, 94,188,226, 97, 63,221,131,194,156,126, 32,163,253, 31, 65,
      157,195, 33,127,252,162, 64, 30, 95,  1,227,189, 62, 96,130,220,
       35,125,159,193, 66, 28,254,160,225,191, 93,  3,128,222, 60, 98,
      190,224,  2, 92,223,129, 99, 61,124, 34,192,158, 29, 67,161,255,
       70, 24,250,164, 39,121,155,197,132,218, 56,102,229,187, 89,  7,
      219,133,103, 57,186,228,  6, 88, 25, 71,165,251,120, 38,196,154,
      101, 59,217,135,  4, 90,184,230,167,249, 27, 69,198,152,122, 36,
      248,166, 68, 26,153,199, 37,123, 58,100,134,216, 91,  5,231,185,
      140,210, 48,110,237,179, 81, 15, 78, 16,242,172, 47,113,147,205,
       17, 79,173,243,112, 46,204,146,211,141,111, 49,178,236, 14, 80,
      175,241, 19, 77,206,144,114, 44,109, 51,209,143, 12, 82,176,238,
       50,108,142,208, 83, 13,239,177,240,174, 76, 18,145,207, 45,115,
      202,148,118, 40,171,245, 23, 73,  8, 86,180,234,105, 55,213,139,
       87,  9,235,181, 54,104,138,212,149,203, 41,119,244,170, 72, 22,
      233,183, 85, 11,136,214, 52,106, 43,117,151,201, 74, 20,246,168,
      116, 42,200,150, 21, 75,169,247,182,232, 10, 84,215,137,107, 53
};

//-------------------------------------------------------------------------
------
// This routine calculates the checksum of the Iridium SBD data message.
// The checksum is calculated from a 'look-up-table'
// The return value is the 8bit checksum.
//-------------------------------------------------------------------------
------
unsigned int8 iridium9601_get_crc(unsigned int8 *data, unsigned int8
array_size) {

unsigned int8 index; // Variable for loop operation.
unsigned int8 crc; // Variable for checksum calculation.

crc = 0;
for (index=1; index<array_size; index++) {
crc = crc_table[crc ^ data[index]];
}
return(crc);
}
```

How to use the CRC:

- calculate the CRC value for each message (byte 2 to last message byte).

- compare the calculated CRC to the transmitted CRC (byte 1).

- if the calculated and transmitted CRC bytes are equal, the SBD message is ok. Otherwise one or more bytes of the message are corrupted. In this case the message should be deleted and not be used for further data processing.

7 IRIDIUM DATA PROCESSING

7.1 Mission Data

For software release see APP I; Example: Data transmission see 8.5.

The mission data of the NEMO float consist of the following main parts:

- Profile header
- Data set 1
- Data set 2
- Data set 3
- Data set 4
- Data set 5

The profile header is at the top of the mission data. It contains the basic information of the complete profile. There are important information of the profile and the different data sets. The profile header has a fixed length of 56 bytes (see chapter 7.2 for more details).

The five data sets have a variable length and contain the proper information of the profile. Each data set consists of a data record with a fixed length. The quantity of data records inside a data set is stored in the profile header. There can be maximal 65535 data records in one data set.

Currently only two of the five data sets are used for the profile data.

Data set 1: unused

Data set 2: CTD data during parking.

Data set 3: unused

Data set 4: CTD data during ascent.

Data set 5: GPS data at surface

The length of the different data records is listed below.

Data record 1: unused

Data record 2: 6 bytes CTD data in sequence during parking.

- 2 bytes temperature in degrees with three decimal places (low_byte/high_byte).
- 2 bytes salinity in PSU with three decimal places (low_byte/high_byte).
- 2 bytes pressure in dbar with one decimal place (low_byte/high_byte).

Data record 3: unused

Data record 4: 6 bytes CTD data in sequence during parking.

- 2 bytes temperature in degrees with three decimal places (low_byte/high_byte).
- 2 bytes salinity in PSU with three decimal places (low_byte/high_byte).
- 2 bytes pressure in dbar with one decimal place (low_byte/high_byte).

Data record 5: 14 bytes GPS data at surface

- 1 byte GPS year

- 1 byte GPS month
- 1 byte GPS day
- 1 byte GPS hour
- 1 byte GPS minute
- 1 byte GPS second
- 4 bytes GPS latitude in degrees with seven decimal places (lowest_byte/low_byte/high_byte/highest_byte).
- 4 bytes GPS longitude in degrees with seven decimal places (lowest_byte/low_byte/high_byte/highest_byte).

Table: Overview mission data.

Profile header (fixed length of 56 bytes)
Data set 1 (variable length) Data record 1 (fixed length) Data record 2 (fixed length) " " " " Data record 65535 (fixed length)
Data set 2 (variable length) Data record 1 (fixed length) Data record 2 (fixed length) " " " " Data record 65535 (fixed length)
Data set 3 (variable length) Data record 1 (fixed length) Data record 2 (fixed length) " " " " Data record 65535 (fixed length)
Data set 4 (variable length) Data record 1 (fixed length) Data record 2 (fixed length) " " " " Data record 65535 (fixed length)
Data set 5 (variable length) Data record 1 (fixed length) Data record 2 (fixed length) " " " " Data record 65535 (fixed length)

7.2 Format of Profile Header

NOTE: THE DATA STORAGE IS IN LITTLE ENDIAN FORMAT.

For 16Bit data (2 bytes) => low_byte / high_byte
For 32Bit data (4 bytes) => lowest_byte / low_byte / high_byte / highest_byte
The byte numeration begins here with '1'. In the source files the variables are stored in arrays and beginning at index '0'.

Profile header:
Byte #
01&02 Profile number, begins with 1 and increases by one for every float profile.
03&04 Serial number, identification of the float controller board.
05 Status byte, used for different flag information.
Bit 0 : ARGOS_ID => Used ID for transmission ('0'=>20bit / '1'=>28bit)
Bit 1 : MULTIPLE_MSG => Indicates the transmission of multiple messages.
Bit 2-7 : UNUSED
06 RTC - year, immediately before Iridium transmission.
07 RTC - month, immediately before Iridium transmission.
08 RTC - day, immediately before Iridium transmission.
09 RTC - hour, immediately before Iridium transmission.
10 RTC - minute, immediately before Iridium transmission.
11 RTC - second, before Iridium transmission.
12 Ice detect count, increments by one for every time float aborts profile due to ice detection.
13&14 Piston position at surface.
15&16 Piston position in park depth.
17&18 Piston position in profile depth.
19&20 Piston position in at end of profile (end of ascent).
21 CPU battery voltage at surface.
22 Pump battery voltage in profile depth (during initialization of motor).
23 Hydraulic current during first ascent (during initialization of motor).
24 Internal tube pressure [PSI] at surface.
25&26 CTD-pressure at surface after ascent (5 digits, one decimal place, in dbar).
27&28 CTD-pressure at surface before descent (5 digits, one decimal place, in dbar).
29&30 Pressure at end of parking (median).
31 - 34 Descent start time, just before descent (in seconds).
35 - 38 Ascent start time, just before ascent (in seconds).
39 - 42 Surfacing time (at 100 dbar) (in seconds).
43 - 46 End of profile time (after last sample or abortion due to ice) (in seconds).
47&48 Quantity of data record 1 (max 65535).
49&50 Quantity of data record 2 (max 65535).
51&52 Quantity of data record 3 (max 65535).
53&54 Quantity of data record 4 (max 65535).
55&56 Quantity of data record 5 (max 65535).

7.3 Format of Test-Message

NOTE: THE DATA STORAGE IS IN LITTLE ENDEAN FORMAT.

For 16Bit data (2 bytes) => low_byte / high_byte
For 32Bit data (4 bytes) => lowest_byte / low_byte / high_byte / highest_byte
The byte numeration begins here with '1'. In the source files the variables are stored in arrays and beginning at index '0'.

TEST-MESSAGE:
Byte #
01 CRC, calculated CRC (bytes 2 to 32), for more information look at CRC routine.
02 Message number is always zero for test message.
03&04 Serial number, identification of the float controller board (low_byte/high_byte).
05 Software version year (in decimal).
06 Software version month (in decimal).
07 Software version day (in decimal).
08 Status byte, used for different flag information.
Bit 0 : ARGOS_ID => Used ID for transmission ('0'=>20bit / '1'=>28bit)
Bit 1 : MULTIPLE_MSG => Indicates the transmission of multiple messages.
Bit 2-7: unused
09 RTC - year info from float startup (reset).
10 RTC - month info from float startup (reset).
11 RTC - date info from float startup (reset).
12 RTC - hour info from float startup (reset).
13 RTC - minute info from float startup (reset).
14 RTC - second info from float startup (reset).
15&16 Max piston count value at min piston position.
17&18 Min piston count value at max piston position.
19 Actual CPU battery voltage during startup.
20 Actual pump battery voltage during startup.
21 Actual tube pressure during startup.
22&23 Specified profile pressure.
24&25 Specified park pressure.
26&27 Specified cycle time.
28&29 Specified transmission time.
30&31 Specified park interval time.
32 GPS year.
33 GPS month.
34 GPS day.
35 GPS hour.
36 GPS minute.
37 GPS seconds.
38-41 GPS latitude.
42-45 GPS longitude.

7.4 Data Conversion

Battery voltage:

The battery voltage is measured with 10bit resolution. But only 8bit are used and transmitted.

Therefore the value is converted from 10bit to an 8bit value.

The 8bit value has the following format:

0-255 => 0-25.5V (e.g. value=143 => 14.3V)

(max. voltage is 25.5V / min. voltage is 0.1V)

Hydraulic current:

The hydraulic current is measured with 10bit resolution. but only 8bit are used and transmitted.

Therefore the value is converted from 10bit to an 8bit value.

The 8bit value has the following format:

0-255 => 0-2.55A (e.g. Value=80 => 0.80A)

(max. current is 2.55A / min. current is 0.01A)

Internal tube pressure:

The tube pressure is measured with 10bit resolution. but only 8bit are used and transmitted.

Therefore the value is converted from 10bit to an 8bit value.

The 8bit value has the following format:

0-255 => 0-25.5psi (e.g. value=112 => 11.2psi)

(max. pressure is 25.5psi / min. pressure is 0.1psi)

Piston position / piston count min, max:

The piston count value is measured with 10bit-resolution which is transmitted in 2 bytes. The number of piston counts depends on the mechanical configuration of the specific float.

CTD data:

All CTD data (salinity, temperature and pressure) for park and profile are stored with two bytes.

CTD-temperature: 5 digits, three decimal places (1milli-degree resolution).

CTD-salinity: 5 digits, three decimal places (1milli-psu resolution).

CTD-pressure: 5 digits, one decimal place (10cm resolution).

Table: Example for value conversions of CTD-data.

	Hex-value	Dec-value	Converted	Unit
Temperature:	3EA6	16038	16,038	°C
Temperature*:	F58B	2677	-2,677	°C
Salinity:	8FDD	36829	36,829	PSU
Pressure:	1D4C	7500	750,0	dbar

*Note regarding negative temperatures (T °C < 0).

Positive temperature range is 0 to 62.535°C (0 to F447 hex).

Negative temperature range is -0.001 to -3.000°C (FFFF to F448 hex).

If(hex value) >= F447, then calculate FFFF – (hex value) + 1= Y_DEG.

Convert Y from hex to decimal, divide by 1000, and multiply by (-1), for degree C.

GPS data:

GPS time and date information is stored with one byte each.

GPS geographic position information (latitude/longitude) is stored with 4 bytes each.

Table: Example for value conversions of GPS data.

	Hex-value	Dec-value	Converted	Unit
GPS year:	06	6	6	
GPS month:	09	9	9	
GPS day:	18	24	24	
GPS hour:	7	7	7	
GPS minute:	20	32	32	
GPS second:	1E	30	30	
GPS latitude:	1FE3E461	535028833	53,5028833	degree
GPS longitude:	051C2F2E	85733166	8,5733166	degree

7.5 Checksum

Every Iridium SBD message contains a checksum. This assures additional data integrity during Iridium transmissions.

Therefore the first byte of each SBD message block contains a Cyclic Redundancy Check (CRC). This CRC (byte 1) is calculated as a function of the message content (byte 2 to message end).

The 8-bit CRC is calculated using the function below.

```
// 8bit CRC look-up-table
// ----------------------
unsigned int8 const crc_table[256] = {
        0, 94,188,226, 97, 63,221,131,194,156,126, 32,163,253, 31, 65,
      157,195, 33,127,252,162, 64, 30, 95,  1,227,189, 62, 96,130,220,
       35,125,159,193, 66, 28,254,160,225,191, 93,  3,128,222, 60, 98,
      190,224,  2, 92,223,129, 99, 61,124, 34,192,158, 29, 67,161,255,
       70, 24,250,164, 39,121,155,197,132,218, 56,102,229,187, 89,  7,
      219,133,103, 57,186,228,  6, 88, 25, 71,165,251,120, 38,196,154,
      101, 59,217,135,  4, 90,184,230,167,249, 27, 69,198,152,122, 36,
      248,166, 68, 26,153,199, 37,123, 58,100,134,216, 91,  5,231,185,
      140,210, 48,110,237,179, 81, 15, 78, 16,242,172, 47,113,147,205,
       17, 79,173,243,112, 46,204,146,211,141,111, 49,178,236, 14, 80,
      175,241, 19, 77,206,144,114, 44,109, 51,209,143, 12, 82,176,238,
       50,108,142,208, 83, 13,239,177,240,174, 76, 18,145,207, 45,115,
      202,148,118, 40,171,245, 23, 73,  8, 86,180,234,105, 55,213,139,
       87,  9,235,181, 54,104,138,212,149,203, 41,119,244,170, 72, 22,
      233,183, 85, 11,136,214, 52,106, 43,117,151,201, 74, 20,246,168,
      116, 42,200,150, 21, 75,169,247,182,232, 10, 84,215,137,107, 53
};


//-------------------------------------------------------------------------
------
// This routine calculates the checksum of the Iridium SBD data message.
// The checksum is calculated from a 'look-up-table'
// The return value is the 8bit checksum.
//-------------------------------------------------------------------------
------
unsigned  int8  iridium9601_get_crc(unsigned  int8  *data,  unsigned  int8
array_size) {

unsigned int8 index; // Variable for loop operation.
unsigned int8 crc; // Variable for checksum calculation.

crc = 0;
for (index=1; index<array_size; index++) {
crc = crc_table[crc ^ data[index]];
}
return(crc);
}
```

How to use the CRC:

- calculate the CRC value for each message (byte 2 to last message byte).

- compare the calculated CRC to the transmitted CRC (byte 1).

- if the calculated and transmitted CRC bytes are equal, the SBD message is ok. Otherwise one or more bytes of the message are corrupted. In this case the message should be deleted and not be used for further data processing.

8 IRIDIUM TRANSMISSION

8.1 Iridium Message

Data are sent via the Iridium 9601SBD transceiver in variable message blocks. The maximum length of one message is 205 bytes. It is not necessary to transmit all 205 bytes in every message, e.g. shorter messages with a length <205 bytes are possible.

8.2 Transmission of Test Message

The test message includes all important information of the float (setup-parameters, start-up data).

The test message is only sent during start-up (self test). The format of the test message is described in corresponding chapter. The length of the message is set to 45 bytes. So the test message can be transmitted in only one SBD message.

8.3 Transmission of Profile Data (single profile)

A SBD message of a single profile transmission consists of the SBD header (byte 1 to 4) and the profile data (byte 5 to 205). The single profile data are transmitted in the following format:

Byte #1 CRC

Byte #2-3 Message number

Bytes #4 Data length of the SBD (only data, without header bytes 1 to 4)

Bytes #5 to 205 Profile data

8.4 Transmission of Profile Data (long-term data storage)

A SBD message of a long-term storage profile transmission consists of the SBD header (byte 1 to 6) and the profile data (byte 7 to 205). The long-term storage profile data are transmitted in the following format:

Byte #1 CRC

Bytes #2-3 Message number

Byte #4 Data length of the SBD (only data, without header bytes 1 to 6)

Bytes #5-6 Profile number

Byte #7 to 205 Profile data

8.5 Example: Data Transmission

Data are stored on the memory card as described under section 3.1.

They are stored without message number and CRC.

As described above one SBD message can have a maximum length of 205 bytes. First the profile data (up to 201 bytes) are read from the memory card before the data are transmitted. Then the SBD header (message number and data length) is updated. For the first message the message number is #1. The data length depend on how many bytes of the profile data were read. A checksum is calculated from bytes 2 to 205 and the result is added as first byte. A total of 205 bytes is then transmitted as a SBD message.

In the next step the next bytes of the profile data are read from the memory card. The message number is incremented to #2. After CRC calculation the next SBD message is transmitted.

The process continues until all data are transmitted from the memory card to the receiving station. The last message can contain fewer profile data bytes. So the data length in the SBD header may be <205.

Once all data are transmitted, the process is finished.

Message numbers and CRCs are created for every sent message and not stored on the memory card and the first 4 bytes must be removed during the reassembly of the complete data set with the data structure described in this manual (first 56 bytes of header followed by data). If there is a failure during transmission of one SBD message, the actual profile will be stored on the memory card and transmitted during next transmission as a long-term storage profile.

9 LONG-TERM DATA STORAGE

Long-term data storage is useful in situations where the float cannot surface due to ice at the surface or strong water mass stratification below the surface. In both cases the data are stored in memory and transmitted at a later point in time when the transmission of stored profiles is possible. In general, the float descends to 100dbar immediately after abortion of the ascent and remains in this position until the cycle-time is completed. This assures that the decent to park-depth always occurs at the same time.

The program also allows determining a fixed period of time where long-term data storage is enabled (instead of 12 months per year).

9.1 Ice Detection Feature

During ascent the float controller performs ice detection using a mathematical algorithm. The temperature median of the seven near-surface sample points ≥ 20 dbar is computed. If the median is ≤ (-1.79) °C or the user-defined temperature, the probability of ice encounter at the surface is high and the ascent will be aborted.

9.2 Water Mass Stratification or Ice Detection

In general, the 'ASCENT TIME' is set to 200% of the time the float would ascend at 0.08m/s from the target depth.

If the 'ASCENT TIME' expires before the float has reached the surface and the temperature T ≤ -1.79°C (one sample) the ascent will be aborted (possibility of ice at the surface).

If the temperature T > -1.79°C , the float will initialize maximum volume settings for oil and air bladders and attempt to surface.

THIS PAGE IS INTENTIONALLY LEFT BLANK

10 TERMINAL CONNECTION

10.1 Communication with Float Controller

With the NEMO-specific terminal connector on the Top Cap the user has the possibility to communicate with the float controller inside the NEMO. The communication is done via RS232.

Settings: 57600, 8, N, 1 protocol: none

How to communicate with the float controller:

1. Unplug Subconn plug below the conductivity cell feeding pipe on the Top Cap (Figure 10.1).
2. Connect plug from terminal cable with terminal connector (Figure 10.2).
3. Start a terminal program (e.g. HyperTerminal) with Settings: 57600, 8, N, 1.
4. Reset NEMO by holding a magnet above the marked area letting it rest for a couple of seconds.
5. Press space bar within 5 seconds to output terminal menu on screen. Otherwise the NEMO starts with the mission or returns to sleep mode.

Figure 10.1: Location of the plug to be removed for the connection of the terminal connector.

Figure 10.2: Connection of terminal connector to the float.

The terminal menu has a main menu with eight sub menu items (self-explanatory).

The user can access the sub menu by pressing the corresponding characters.

Pressing the space bar will display the actual menu again.

With the 'ESC-button' the user can leave a sub menu and jump to the main menu.

For example a list of the main menu and sub-menus can be found below.

```
TERMINAL MENU:
--------------
>A<  MEMORY CONTROL
>B<  HYDRAULIC CONTROL
>C<  PNEUMATIC CONTROL
>D<  CTD CONTROL
>E<  ADC CONTROL
>F<  SYSTEM CONTROL
>G<  EXPERT CONTROL
>H<  WildCAT CONTROL
>I<  RAFOS CONTROL
>J<  TEST-ROUTINES


MEMORY CONTROL MENU:
--------------------
>a<  read single sector
>b<  read sector from/to
>c<  check cf card
>d<  get drive identity
>e<  write byte to sector
>f<  clear sector from/to
>g<  read sector from/to (tare dump)

press 'ESC' to return to main menu


HYDRAULIC CONTROL MENU:
-----------------------
>a<  move piston out by time
```

```
>b<  move piston in by time
>c<  move piston out max
>d<  move piston in max
>e<  open break
>f<  close break
press 'ESC' to return to main menu


PNEUMATIC CONTROL MENU:
-----------------------
>a<  turn air pump on by time
>b<  open air valve by time
>c<  turn on air pump
>d<  turn off air pump
>e<  open air valve
>f<  close air valve
press 'ESC' to return to main menu


CTD CONTROL MENU:
-----------------
>a<  get pressure
>b<  acquisition (with pump)
>c<  acquisition (without pump)
>d<  ctd pressure offset
press 'ESC' to return to main menu


ADC CONTROL MENU:
-----------------
>a<  get battery voltages
>b<  get tube pressure (vacuum)
>c<  get piston position
press 'ESC' to return to main menu


SYSTEM CONTROL MENU:
--------------------
>@<  Software Reset
>a<  read date/time
>b<  activate mission mode
>c<  deactivate mission mode
>d<  display mission parameter
>e<  display expert parameter
>f<  go to sleep mode
press 'ESC' to return to main menu


EXPERT CONTROL MENU:
--------------------
>*<  re-enter bootloader
>@<  Software Reset
>a<  set date/time
>b<  setup serial number
>c<  setup mission parameters
>d<  setup expert parameters
press 'ESC' to return to main menu
```

```
WILDCAT CONTROL MENU:
--------------------
>a<  send single testmessage
>b<  send multiple testmessage
press 'ESC' to return to main menu


RAFOS CONTROL MENU:
--------------------
>a<  Turn on RAFOS module
>b<  Turn off RAFOS module
>c<  Start measurement
>d<  Read correlation data
press 'ESC' to return to main menu


TEST-ROUTINE CONTROL MENU:
--------------------
>a<  tare test
>b<  hydraulic test
press 'ESC' to return to main menu
```

10.2 Selected Functions

This section describes some selected functions that are available in the menu.

10.2.1 Test of internal pressure (vacuum)

go to Terminal Menu

select >E< ADC CONTROL

select >b< get tube pressure (vacuum)

Now the display will show the internal pressure measured every 1.5 s. The internal pressure of the float should be approximately 10.5 psi @ 19 ^{0}C float temperature.

Attention: it takes several hours for the float to adjust to the ambient temperature!

10.2.2 Activation of mission mode

go to Terminal Menu

select >F< SYSTEM CONTROL

select >c< activate mission mode

These commands put the float into sleep mode. Disconnect the data cable. In order to activate the mission pass a magnet over the switch (location marked with a cross; duration 2 s).

After 10 sec the float begins to fill the pancake (oil; ca. 90 min) and sleeve (air; ca. 90 sec) bladders and the float is ready for deployment. Now the float begins to send the test message via Iridium. This will continue until mission start (6 h after Reset).

10.3 NEMO Final Test Procedure Before Delivery

This section describes the final test procedure which is done prior delivery.

Most important are the system parameters, date/time, and the status of the analogue values (battery voltage, tube pressure, piston position).

1. Connect float to terminal (see chapter 10.1)
2. Choose sub-menu "E" => ADC CONTROL MENU
3. In the ADC CONTROL MENU choose the items to get the analogue values.
4. Press 'ESC' to return to the main menu
5. Chose sub-menu "F" => SYSTEM CONTROL MENU
6. Check actual date/time => item "b".
7. If necessary setup new date/time in sub-menu "G" => EXPERT CONTROL MENU *(requires access code)*
8. Output actual float mission parameter => item "d".
9. Output actual float expert parameter => item "e".
10. If necessary setup new float serial number in sub-menu "G" => EXPERT CONTROL MENU *(requires access code)*
11. If necessary setup new float mission parameter in sub-menu "G" => EXPERT CONTROL MENU *(requires access code)*
12. If necessary setup new float expert parameter in sub-menu "G" => EXPERT CONTROL MENU *(requires access code)*
13. Chose sub-menu "F" => SYSTEM CONTROL MENU
14. Output actual float mission parameter again (verify) => item "d".
15. Output actual float expert parameter again (verify) => item "e".
16. Assure that the float is set to PISTON MAX IN and that the air bladder is empty.
17. Activate mission mode in sub-menu "F" => SYSTEM CONTROL MENU=> item "c".

    (NEMO goes to sleep mode and starts the mission on next reset).
18. Disconnect terminal menu and unplug terminal cable.
19. Put protective cap on terminal connector (Subconn).

THIS PAGE IS INTENTIONALLY LEFT BLANK

11 BATTERY SUPPLY

The “Float-Controller” is provided with 7.2V / 26Ah from batteries. For the hydraulic, pneumatic and Iridium modem batteries with 14.4V / 78Ah are used.

WARNING: DO NOT CRUSH, RECHARGE OR DISASSEMBLE THE BATTERIES. DO NOT HEAT THE BATTERIES ABOVE 100 °C (212 °F) OR INCINERATE THE BATTERIES. DO NOT SHORT-CIRCUIT THE BATTERIES. CRUSHING, RECHARGING, DISASSEMBLE, HEATING, AND INCINERATION MAY RESULT IN FIRE, EXPLOSION AND SEVERE BURN HAZARD.

The operator is warned and must accept and deal with the risks in order to use the NEMO-float according to safety standards.

Only trained operators shall deploy the instrument.

OPTIMARE disclaims liability for any consequences of combustion or explosion. Furthermore, warranty by OPTIMARE expires immediately upon the attempt to open the instrument.

THIS PAGE IS INTENTIONALLY LEFT BLANK

12 QUICK-START GUIDE NEMO FLOAT

1. Slowly swipe the magnet over “START”
2. Place the NEMO beeper next to the antenna.
3. Every 90 seconds the NEMO beeper will beep 6 times.
4. Wait about 90 minutes for the oil bladder to inflate.
5. Wait about 2 minutes for the air bladder to inflate.

It is safe to launch the NEMO 2 hours after “START” (see 1.). During launch the instrument has to be kept in an upright position with the antenna facing up.

NOTE: PLEASE BE ADVISED THAT THE **OPERATING MANUAL** IS APPLICABLE FOR HANDLING AND SAFETY INSTRUCTIONS OF THE NEMO FLOAT. FOR FURTHER INFORMATIONS PLEASE REFER TO THE **OPERATING MANUAL**.

THIS PAGE IS INTENTIONALLY LEFT BLANK