argo data management

Argo data management
DOI: http://dx.doi.org/10.13155/47998

Processing BGC-Argo measurement timing information at the DAC level

Version 1.0

March 28th 2017

ARGO

part of the integrated global observation strategy

Argo data management
Processing Argo measurement timing information at the DAC level

Authors: Henry C. BITTIG, Catherine SCHMECHTIG, Jean-Philippe RANNOU, Antoine POTEAU

How to cite this document

Henry C. BITTIG, Catherine SCHMECHTIG, Jean-Philippe RANNOU, Antoine POTEAU (2017). **Processing BGC-Argo measurement timing information at the DAC level, v1.0**. http://dx.doi.org/10.13155/47998

Table of contents

History of the document

Version	Date	Authors	Modification
1.0	March 2017	H. Bittig, C. Schmechtig, J.-P. Rannou, A. Poteau	Initial version with feedback from bio-argo list

1 Introduction

It is optional to the individual DAC to implement and use the MTIME parameter.

Thermal lag or sensor time response corrections (e.g., [1]) require knowledge of the time of each individual sensor observation. In the Argo data system, the float's trajectory file is the natural place to put measurement timing information. Historically, only few levels of a float's profile have been timed, and DACs stored these sparse timed levels of the float profile in the trajectory already. Depending on the DAC, all measured parameters are stored with the timing information, or just the PRES variable is stored together with the time. With the need to have all observations timed, this approach leads to the following problems:

a) With all parameters stored together with the timing information and put into the trajectory file, the trajectory file in effect duplicates all profiles. This contradicts the Argo data system's design to split profile and trajectory files and may pose a file size issue.

b) With only PRES stored together with the timing information and put into the trajectory file, the only link between trajectory timing information and profile parameter observations is through the PRES variable. Due to the potential occurrence of equal PRES values for different parameter observations, this link may be ambiguous.

To keep the structure of the Argo data system and to resolve these problems, Biogeochemical-Argo saw a need to be able to store timing information in the float's profile file. Following discussions at ADMT16, AST17, and ADMT17 it was concluded that timing information associated with each measured parameter of the profile is scientifically useful for Biogeochemical-Argo, but does not seem necessary for core Argo. Sparse timing data should remain in the trajectory file – it was designed to handle this data and keeps QC of timing information mostly to one file. Abundant timing information (i.e., timing information for all profile observations), however, are stored in the i-parameter "MTIME" in the b-profile files.

As an i-parameter, MTIME is accessible to the bio-sandbox (expert / biogeochemistry) user and to the data system for corrections to other parameters. This was considered as being the relevant user group and the main scope of this variable. If defined as a b-parameter, it would have to be carried through to the merged file intended for non-expert, merged data end-users, without having adequate procedures for real-time and delayed-mode QC of MTIME in place. Moreover, a b-parameter would impose extra processing and DMQC requirements. Thus, MTIME is defined as an i-parameter containing only the raw timing information (with potential real-time adjustments). All further processing needs to be done by the user.

2 Recommendations for the storage of timing informatiom

Following feedback from AST17 and ADMT17, abundant profile timing information data are stored in the i-parameter MTIME in the b-profile file with the following specifications:

parameter name	MTIME
parameter sensor	FLOATCLOCK_MTIME
standard name	measurement time
long name	fractional day of the individual measurement relative to JULD of the station
units	days since JULD
conventions	relative days
resolution	(at least) 1e-06
range	[-3;+3]

Different float types can transmit timing information in different formats, e.g., as UNIX Epoch, which is seconds since January 1, 1970, as a string like “03/15/2017 172710", or as string “2017-Mar-15

17:27:10” for some of the data (the first of a package) and as integer seconds (relative to the first time of a package) for the remaining data. The raw data are the same and need to be expressed in a common, universal way, independent of float type. Therefore, profile timing information is not stored as transmitted by the float but with unit conversion to a common format as specified above.

The motivation to use a relative time instead of an absolute time is to reduce potential efforts for QC of MTIME, e.g., due to clock drift. Clock drift can be an issue over the float lifetime, but it is negligible over the duration of a profile. By using JULD as a reference date, only JULD would have to be adjusted for clock drift. Otherwise, if storing timing information as an absolute time in MTIME, both JULD and MTIME would require adjustment – by the same offset. This was seen as unnecessary QC load by the ADMT and preference was given to use JULD as reference time.

As for the unit, seconds since JULD are an obvious start, but feedback at ADMT17 proposed to use the same unit as JULD for convenience. Those who wanted absolute measurement times would thus easily add MTIME and JULD without the need for an extra conversion factor.

Bin-averaged data are a particular case with respect to assigning a measurement time. It has to be specified in the float type section below whether timestamps are applicable to bin averaged data and whether the timestamp is associated with the first (or last) sample of the bin, or whether the timestamp itself is somehow averaged.

3 Float types

3.1 Navis BGCi floats

Some Navis Biogeochemical floats with iridium communication provide a timestamp for each profile observation labelled "UNIX Epoch" in the float .msg file. The "UNIX Epoch" gives the timing information in seconds since January 1, 1970, 00:00:00, and can easily be converted to MTIME accordingly.

3.2 PROVOR CTS4 Floats

Exact measurement timing information for Provor CTS4 floats can only be assigned to raw data, not to bin averaged data.

In essence, raw sensor data by a CTS4 float (a) are sampled at a set frequency, where (b) every x-th sample of a sensor has a proper timestamp, and (c) the order of samples is known. The combination of (a) – (c) can be used to assign a timestamp to every raw data sample: (a) can be found in the mission configuration; (b) is sparse timing information that would be available in the traj; (c) is only available at the decoding step, which is why this assignment of (full) timing information to raw data needs to be done during decoding of CTS4 floats.

Each sensor's data are transmitted sensor by sensor in separate rudics packages. Each sensor and thus each sensor's packages are identified by a unique "sensor data type" value. The following "sensor data type" values are used for the following types of data (from [2]):

Value	Sensor Data Type
0	CTD (Average)
1	CTD (Standard deviation+ Median)
2	CTD (Raw data)
3	Oxygen (Average)
4	Oxygen (Standard deviation+ Median)
5	Oxygen (Raw data)
6	Not used

7	Not used
8	Not used
9	ECO3 (Average)
10	ECO3 (Standard deviation + Median)
11	ECO3 (Raw data)
12	OCR (Average)
13	OCR (Standard deviation + Median)
14	OCR (Raw data)
15	FLNTU (Average)
16	FLNTU (Standard deviation + Median)
17	FLNTU (Raw data)
18	cROVER (Average)
19	cROVER (Standard deviation + Median)
20	cROVER (Raw data)
21	SUNA (Average)
22	SUNA (Standard deviation + Median)
23	SUNA (Raw data)
24	SUNA (APF Frame)
25	SUNA (APF Frame 2)

As mentioned above, the following description to assign MTIME is only valid for unaveraged, raw data (i.e., Sensor Data Types 2, 5, 11, 14, 17, 20, 23).

(a) Profile measurements of a sensor are split in up to 5 "measurement zones" that can be defined by the float operator. They are documented in the meta file under (LAUNCH_)CONFIG_PARAMETER with the key "CONFIG_XDepthZoneYDepthZoneY+1PressureThreshold_dbar" where X indicates the sensor (e.g., Ctd, Optode, Ocr, Eco, Suna) and Y=1..4 indicates the transition between zone Y and Y+1.

Within each zone, the sensor sampling frequency, data processing mode (raw, mean, or mean + standard deviation + median), and acquisition mode (sensor off, pulsed, continuous) are changed according to the mission presets, archived in the meta file. Measurements are not triggered by a pressure table. Instead, the measurements follow the sampling frequency defined for each sensor in the mission parameters (key "CONFIG_XDepthZoneYAscentSamplingPeriod_seconds" in the meta CONFIG_PARAMETER). Measurements might be done more often (e.g., acquisition mode continuous), however, they are only stored at the sampling frequency.

The high-frequency CTD pressure readings (typically every 2 s) are used to determine transitions between measurement zones for all sensors. A zone transition is triggered when (a) the sampling frequency for a sensor changes, (b) the data processing mode for a sensor changes, or (c) the acquisition mode changes (e.g., sensor switched on or off). In that case, the delay since the last sample of a given sensor is reset to zero and the first sample in the next zone occurs once the new sampling period elapsed. If no zone transition is triggered, the sensor continues its measurements without additional delay.

(b) Each rudics package provides only one timestamp, i.e., the timestamp of the first sensor observation of this package.

(c) The number of observations *n* inside an individual rudics package varies by sensor, depending on the precision of the data and how many data are transmitted per observation. The order of observations inside an individual package is in chronological order. Accordingly, an "IDtime" index (1..*n*) can be assigned to each observation of a package. The timestamp of the first observation (IDtime=1) is the package timestamp. Timing information for the other observations (IDtime>1) need to be reconstructed from the sequence of packages for each sensor and knowledge of the measurements zones.

An algorithm to assign MTIME for sensor raw data of CTS4 floats from the respective package times and sequence of acquisition is under development.

3.3 PROVOR CTS5 Floats

Provor CTS5 floats can transmit their data in two ways

- in standard binary format (compact):

(d) The first measurement of each measurement zone has an absolute timestamp. All following data do not have a time reference.

- in extended binary format:

(e) The first measurement of each measurement zone has an absolute timestamp. All other measurements in the respective zone are referenced on a relative axis with a delay (in seconds) with respect to its previous observation. This can be easily converted to MTIME.

Preference should be given to transmit data in extended binary format.

3.4 Others

Action: Find other floats and/or common feature that provide a "Webbtime" UNIX Epoch (or similar timestamp) for each profile measurement.

4 References

1. Bittig, H. C., and A. Körtzinger (2017): Technical note: Update on response times, in-air measurements, and in situ drift for oxygen optodes on profiling platforms. Ocean Sci., 13, 1–11, doi:10.5194/os-2016-75.
2. PROVOR ProvBioII – UTI USER MANUAL, rev. 9, NKE Instrumentation. 2015

5 Acknowledgements