DAC Trajectory Workshop
Trajectory Format 3.1

M. Scanderbeg, J. Gilson

December 2017
Hamburg, Germany

How have we reached this point......

Fitting the Trajectory file into 'standard' Argo practice

Includes more cycle timing information and other measurements during float's trajectory

Trajectory Format 3.1

ANDRO DEP files

1) Flexible format to indicate where in trajectory measurements occurred
2) Retains 'Raw' Data
3) Logic that is familiar

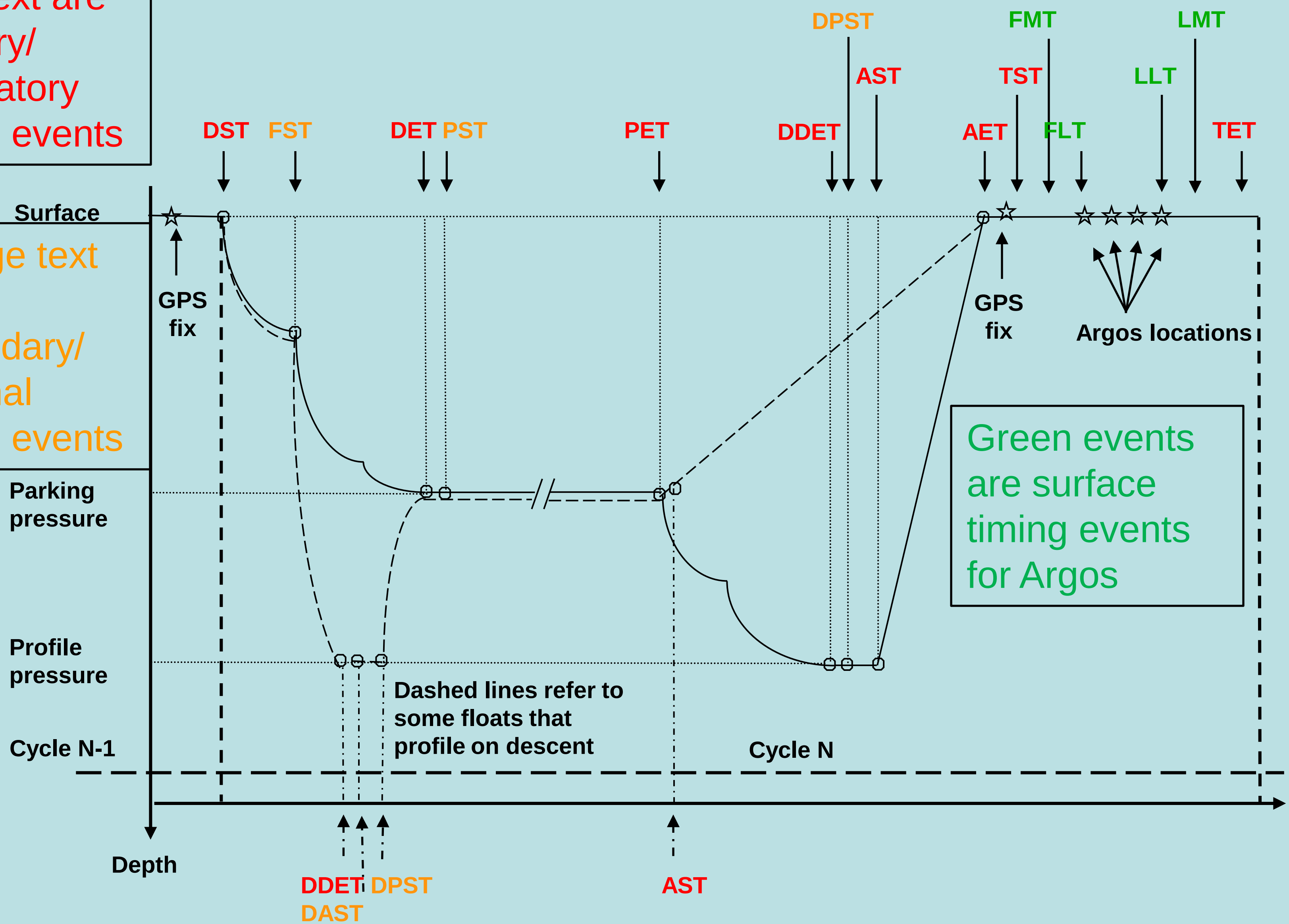
Overview of Changes

- Identified key cycle timing events that Argo would like measured (primary and secondary mcodes)
- Added MEASUREMENT_CODE variable to indicate what type of measurement was taken and when during the cycle
- JULD_STATUS introduced to indicate how the JULD timing information was obtained
- JULD_ADJUSTED introduced to hold estimates and clock drift adjustments
- CLOCK_DRIFT(N_CYCLE) introduced
- Modified variables dealing with cycle number
- Positioning System Errors/Satellite Name (ARGOS centric)
- Format contains redundant N_MEASUREMENT and N_CYCLE variable groups

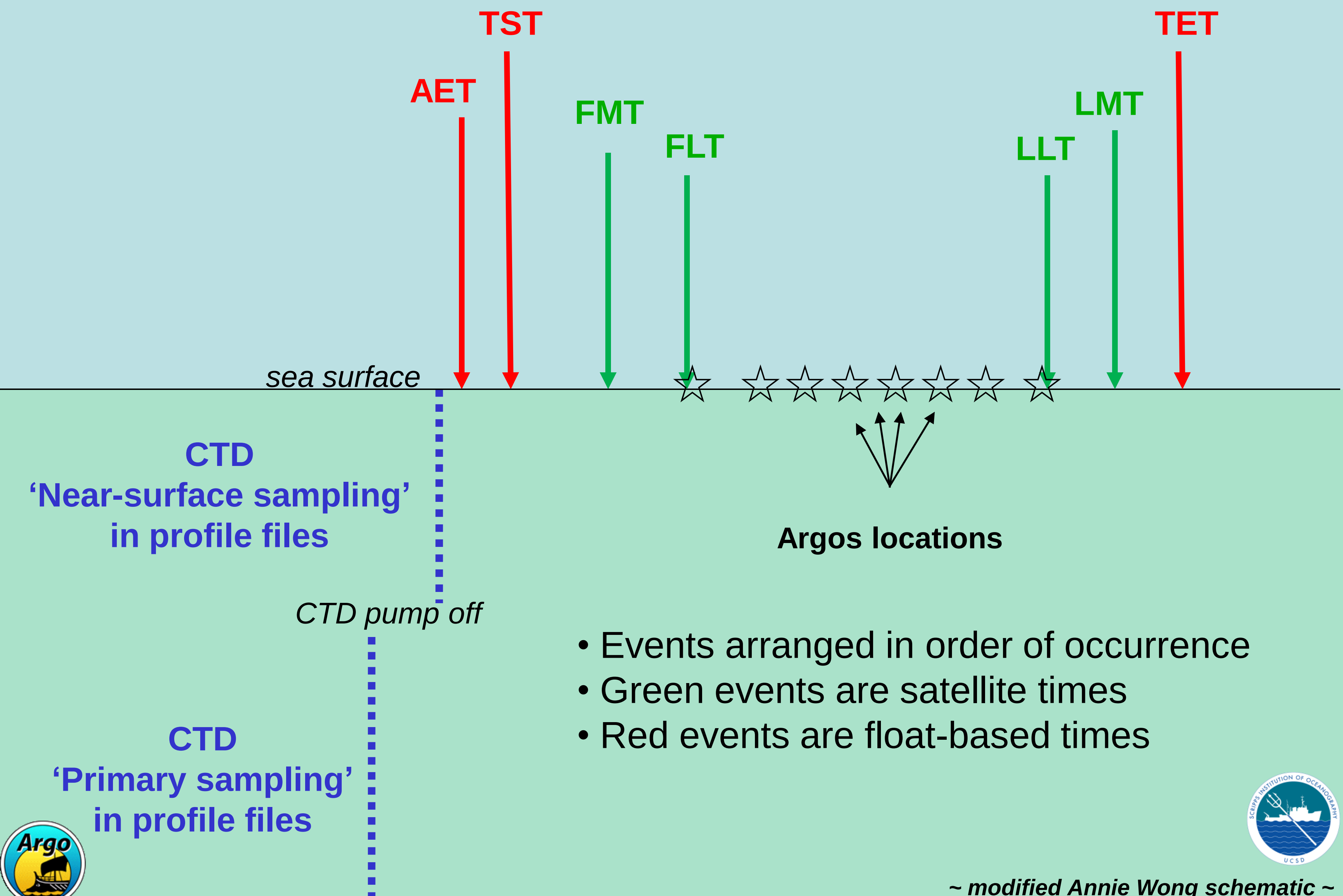
Primary and Secondary Measurement Code (redundant with N_CYCLE variables)

Red text are primary/ mandatory timing events

Orange text are secondary/ optional timing events



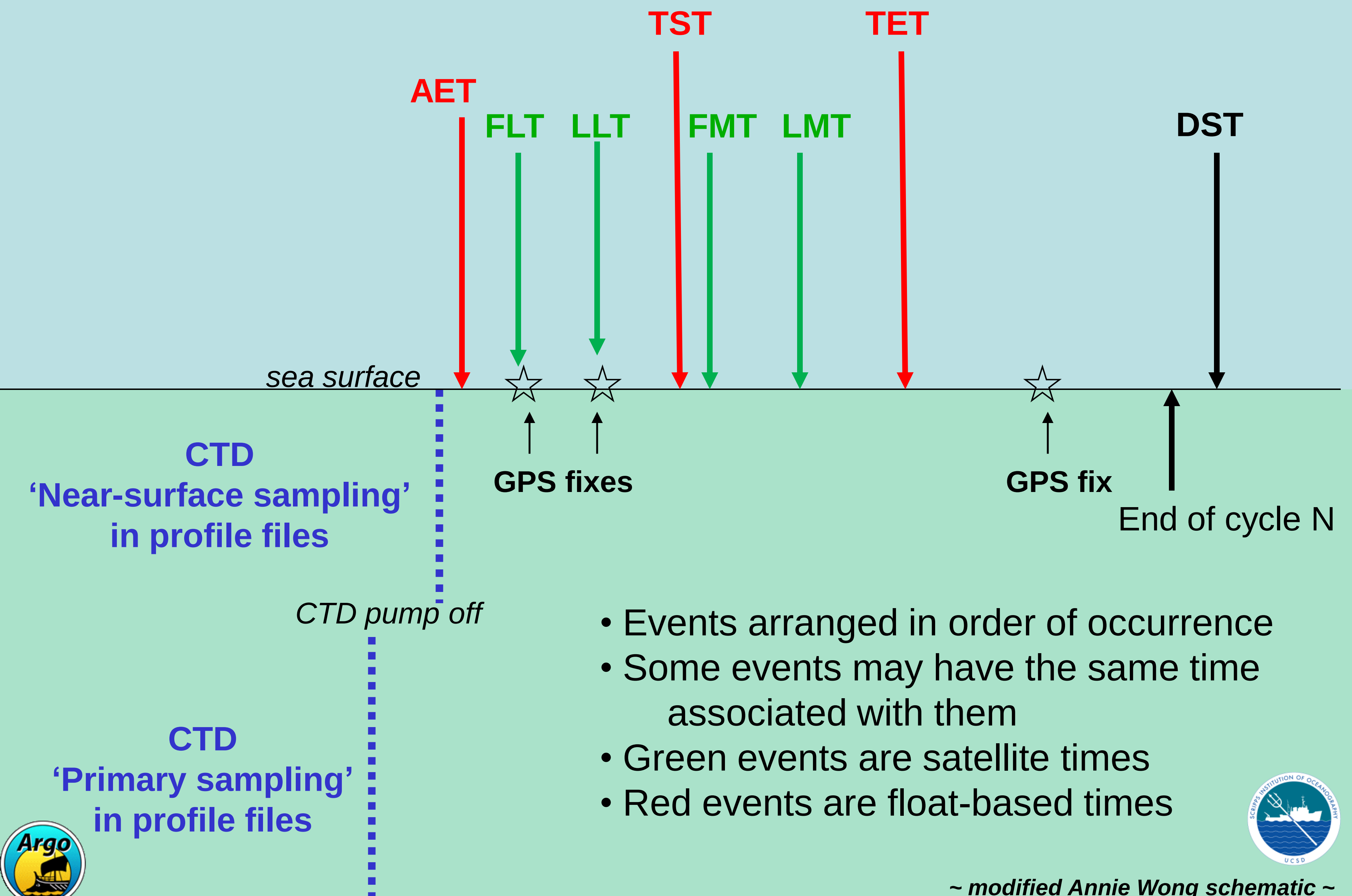
Close up of surface times for Argos floats



- Events arranged in order of occurrence
- Green events are satellite times
- Red events are float-based times

~ modified Annie Wong schematic ~

Close up of surfaces time for GPS floats



- Events arranged in order of occurrence
- Some events may have the same time associated with them
- Green events are satellite times
- Red events are float-based times

~ modified Annie Wong schematic ~

Order of data in trajectory file

- Events are to be arranged in the order they occurred; not arranged by increasing MC or increasing JULD
- JULD times come from two sources: floats and satellites

  JULD may be out of chronological order depending on internal clock drift as compared to GPS or Iridium satellite clocks

  - When clock drift corrections are applied and put into JULD_ADJUSTED, that will be in chronological order
- Some events have the same times (especially Iridium floats)
- If no time is associated with an event, it should be placed between the event prior and the event following it with FillValue in JULD

Guidelines for inclusion of timing data/variables in netCDF file

All Primary and Secondary MC events that **are experienced by the float** are required to be present in the N_MEASUREMENT array and redundantly in the N_CYCLE variables. All other codes are voluntary.

For example, many float types have a test, shallow cycle after deployment. Typically they do not have a drift phase. Thus it is not necessary to record PST (Park Start Time) or PET (Park End Time).

Status values assigned to times: Defined in variable definition...

- If the float experiences an event but the time is not able to be determined, then use status='9'. This indicates that it might be possible to estimate in the future, it is a placeholder.
- In the N_CYCLE variables, if the float does not experience an event then use status='fillvalue'. Only events that are experienced by a float are recorded in the N_MEASUREMENT array so status='fillvalue' is not used in those variables.

Measurement Code Table/Key

MC Code values range from 0-1099

Defined catagories of MC codes

- Primary MC : Mandatory cycle timing variables (100,200,300,etc)
- Secondary MC : Suggested, float specific (150, 250, 350, etc)
- Relative Generic Codes (MC-24 to MC-1)
  - Generic definitions allow application to wide range of floats/data
  - Every Primary and Secondary MC has a 'set' of 'Relative' codes associated
- Relative Specific Codes (MC+1 to MC+25)

Relative Generic MC Table Key

Relative Measurement code	Meaning
MC minus 1	Any single measurement transitioning towards MC (see MC-10 for a 'series' of measurements)
MC minus 2	Maximum value while float is transitioning towards an MC (e.g. pressure)
MC minus 3	Minimum value while float is transitioning towards an MC (e.g. pressure)
MC minus 4	Any averaged measurements made during transition to MC
MC minus 5	Median value while float is transitioning towards an MC
MC minus 6	Standard deviation of measurements taken during transition towards an MC
MC minus 7 to MC minus 9	currently unassigned
MC minus 10	Any “series” of measurements recorded while transitioning towards MC. (e.g. Provor 'spy' measurements, SOLOII pressure-time pairs, etc)
MC minus 11	Active adjustment to buoyancy made at this time
MC minus 12	Any supporting measurements for the maximum value (MC minus 2)
MC minus 13	Any supporting measurements for the minimum value (MC minus 3)
MC minus 14	Any supporting measurements for the averaged value (MC minus 4)
MC minus 15	Any supporting measurements for the median value (MC minus 5)
MC minus 16 to MC minus 24	currently unassigned

Subset of Measurement Code Table

Relative Generic MC

Primary MC

Relative Specific MC's

476-499	see above table	Any measurement recorded during transition towards AST	
500	**AST**	**All measurements made at the start of the float's ascent to the surface** Time (JULD_ASCENT_START)	Time: APEX APF9, PROVOR, ARVOR, SOLO-II, NEMO, NEMOIR, POPS
501		Down-time end date: end date of the down-time parameter reported by APEX floats	APEX
502		Ascent start date directly transmitted by APEX floats	APEX
503		Deepest bin reached during ascending profile	
504-525	unassigned	Reserved for specific timing events around AST.	

N_MEASUREMENT versus N_CYCLE variables

Just like in previous trajectory formats we have N_MEASUREMENT and N_CYCLE variable groups

In N_MEASUREMENT have 12 important timing 'events' that receive highlighted 'codes' (for example Descent Start MC=100), flexibility for ALL timing 'events'

In N_CYCLE have 16 important timing 'events' that have their own variables (for example JULD_DESCENT_START)

	N_MEASUREMENT	N_CYCLE
dimension	Unlimited	N_CYCLE (defined)
timing events	8 Primary MC, 4 Secondary MC, other MC	16 timing variables (4 extra events are first and last message time and location on surface)
Data included	Raw float masurements, satellite/position details, best estimates, status	subset of best estimates and status

Redundant (↔ between N_MEASUREMENT and N_CYCLE timing events)

JULD, JULD_ADJUSTED and *_STATUS

- JULD group (R-mode)

holds the raw, unmodifiable float timing information

allowed STATUS = 2,3,4,9

no clock drift correction

- JULD_ADJUSTED group (A-mode, D-mode)

holds the 'best estimate'

allowed status =0,1,2,3,4,9

clock drift correction:

JULD + clock drift goes into JULD_ADJUSTED in N_MEASUREMENT array and corresponding times in N_CYCLE array

JULD/JULD_ADJUSTED STATUS flags

Flag	**Meaning**
0	Value is estimated from pre-deployement information found in the metadata
1	Value is estimated using information not transmitted by the float or by procedures that rely on typical float behavior
2	Value is transmitted by the float
3	Value is directly computed from relevant, transmitted float information
4	Value is determined by satellite
9	Value is not immediately known, but believe it can be estimated later

The clock drift goes in the CLOCK_DRIFT variable in N_CYCLE

Final Thoughts and Workshop goals

- Trajectory file designed to be flexible and allow events to be added with new MCs (seeing this with BGC near-surface measurements)
- Need float expert and/or manufacturer to help match raw data from floats to measurement codes – our work today! Will document results in Traj Cookbook
- Procedures to estimate times should be agreed upon and documented to be consistent (some estimates in real time, some in delayed mode)
- Need good communication between DACs and PIs to ensure all data is received and understood

Final Thoughts and Workshop goals

- Here about new float types
- Discuss the trajectory File Checker and when it should become active
- Discuss real time QC procedures to see if they can be improved
- Consider DM for trajectory files
- Find ways to help DACs create accurate, consistent trajectory files in a timely manner: code sharing?

- Other needs?

Working group activity

- Split into groups by float type
- Goal is to fill in blank MC tables for following float types:
    - APEX APF9 with Argos
    - APEX APF9 with Iridium
    - APEX APF11
    - NAVIS
    - Arvor
    - BGC APEX (if interested in this)
- Each group will report back & share their results on 1 – 2 floats
- If we run out of time today, we can finish tomorrow

Argo trajectory file measurement codes (MC) for				
Code (timing)	**Name in float data output**	**Description**	**Units and data profile number**	**JULD_STATUS**
0 (launch)				
During the drift phase, float				

- Table is split into sections
  - Descent start to drift start
  - Drift end to profile start
  - Profile start to profile end
  - Surface
- Use as many rows as necessary; add more if needed
- Highlight questions somehow

Working group instructions

- Feel free to use DAC Trajectory Cookbook, timing document, etc. to help you
- Goal is to make most useful trajectory files for users; including as much timing information as possible
- Looking to find agreement on what timing calculations/estimations will be expected in real time by DACs and what will be done in delayed mode
- Be flexible and open to finding the best way to assign codes and do small calculations, even if that means you will need to change how your trajectory are created at the DAC
- Be aware that you may need to look into more than one data file or cycle to find all information needed for one cycle

X – 10 = in-water samples, part of end of profile, shallower than nominal 10dbar
X – 20 = in-water samples, part of surface sequence
X – 30 = in-air samples, part of surface sequence
X – 1 = individual surface observations

Data to include should all be in PPOX_DOXY.