



RVDAS

General info here about the RVDAS logger system.

Table of Contents:

RVDAS.....	1
Table of Contents:	1
Modules	7
RVDAS RAM	7
Version.....	7
Changes.....	7
Build 2.7 (10/2020).....	7
Build 2.6 (01/2019).....	7
RVDAS RAM UserInterface	7
Version.....	7
RVDAS Ingestor	8
Built With.....	8
RVDAS Netcdf Converter	8
Sensor NMEA tableName(s) and fieldnames:	8
Applanix POS MV320 V5 Position/attitude Primary scientific GPS	8
POSMV	9
\$GPGGA – Global Positioning Fix Data.....	9
Table: posmv_gpgga	9
\$GPHDT – Heading – True Data	10
Table: posmv_gphdt	10
\$GPVTG – Course Over Ground and Ground Speed Data	10
Table: posmv_gpvtg.....	10
\$GPRMC – RMC navigation data	11
Table: posmv_gprmc	11



\$GPZDA – Time and Date	11
Table: posmv_gpzda.....	11
\$PASHR – Attitude Data.....	12
Table: posmv_pashr	12
\$GPGLL – Position data: Position fix, time of position fix and status.....	12
Table: posmv_gpgll.....	12
\$GPGST – Pseudorange Noise Statistics.....	13
Table: posmv_gpgst.....	13
Kongsberg Seatex Seapath 330 Position/attitude Secondary scientific GPS	13
SEAPATHPOS	13
\$INGGA – Global Positioning Fix Data	14
Table: seapathpos_ingga.....	14
\$INHDT – Heading – True Data.....	14
Table: seapathpos_inhdt.....	14
\$INVTG – Course Over Ground and Ground Speed Data	15
Table: seapathpos_invtg.....	15
\$INRMC – RMC navigation data.....	15
Table: seapathpos_inrmc.....	15
\$INZDA – Time and Date.....	16
Table: seapathpos_inzda.....	16
\$GNGST – GPS Pseudorange Noise Statistics.....	16
Table: seapathpos_gngst, seapathpos_gpgst	17
SEAPATHATT	17
\$INZDA – Time and Date.....	17
Table: seapathatt_inzda.....	17
\$INGGA – Global Positioning Fix Data	18
Table: seapathpos_ingga.....	18
\$PSXN,23 – Roll, Pitch, Heading and Heave observations	18
Table: seapathatt_psx23.....	18
\$PSXN,20 – Quality for Roll, Pitch, Heading and Heave observations	19
Table: seapathatt_psx20.....	19
C-Nav3050 DGNSS (L2 correction to Applanix PosMV)	19
CNAV	19



\$GNGGA – Global Positioning Fix Data	20
Table: cnav_gngga	20
\$GNVTG – Course Over Ground and Ground Speed Data	20
Table: cnav_gnvtg	20
\$GNDTM – Datum being used	21
Table: cnav_gndtm	21
\$GNGST – GPS Pseudorange Noise Statistics	22
Table: cnav_gpgst, cnav_gngst	22
iXSea PHINSIII Inertial Navigation System	22
PHINS	22
\$PASHR – Attitude Data	23
Table: phins_pashr	23
\$PRDID – attitude data* replaced with PIXSE.ATITUD and HEHDT	23
Table: phins_prdid	24
\$HEHDT – Heading – True Data	24
Table: phins_hehdt	24
\$HEHTS – Heading – Data	24
Table: phins_hehts	24
\$PIXSE,ATITUD Roll Pitch Attitude Data	24
Table: phins_pixseattitud	24
\$PIXSE,POSITI Latitude, longitude, altitude Data	25
Table: phins_pixsepositi	25
\$PIXSE,SPEED_ East, North, Up Speed Data	25
Table: phins_pixsespeed0	25
\$PIXSE,UTMWGS UTM Zone Related Data	25
Table: phins_pixseutmwgs	25
\$PIXSE,HEAVE_ surge, sway and heave Data	26
Table: phins_pixseheave_	26
\$PIXSE,TIME__ UTC time reference Data	26
Table: phins_pixsetime00	26
\$PIXSE,STDHRP Heading, Roll and Pitch Standard Deviation Data	26
Table: phins_pixsestdhrp	26
\$PIXSE,STDPOS Latitude, Longitude and Altitude Standard Deviation Data	27



Table: phins_pixsestdpos	27
\$PIXSE,STDSPD North Speed, East Speed and Vertical Speed Standard Deviation Data	27
Table: phins_pixsestdspd	27
\$PIXSE,UTCIN_ UTC time received Data	27
Table: phins_pixseutcin0	27
\$PIXSE,GPSIN_ Latitude, Longitude, Altitude, UTCTime and quality Data Received from the Manual GPS	27
Table: phins_pixsegpsin0	28
\$PIXSE,GP2IN_ Latitude, Longitude, Altitude, UTCTime and quality Data Received from the GPS2 Sensor.....	28
Table: phins_pixsegp2in0	28
\$PIXSE,ALGSTS INS Algo Status	28
Table: phins_pixsealgsts.....	28
\$PIXSE,STATUS INS System Status	29
Table: phins_pixsestatus	29
\$PIXSE,HT_STS INS High Level Status.....	29
Table: phins_pixseht0sts	29
Fugro 9205 DGNSS Seastar (L2 correction for Seapath330)	29
FUGRO	29
\$GPGGA – Global Positioning Fix Data.....	30
Table: fugro_gpgga.....	30
\$GNVTG – Course Over Ground and Ground Speed Data	31
Table: fugro_gpvtg	31
\$GNDTM – Datum being used.....	31
Table: fugro_gpdtm.....	31
\$GPRMC – RMC navigation data	32
Table: fugro_gprmc.....	32
\$GPGSV – Number of SVs in view, PRN, elevation, azimuth and SNR	32
Table: fugro_gpgsv, fugro_glgsv.....	32
\$GPGLL – Position data: Position fix, time of position fix and status.....	33
Table: fugro_gppll	33
\$GNGSA – GPS DOP and active satellites	34
Table: fugro_gngsa	34
Sonardyne Fusion USBL	35



RANGER2USBL.....	35
\$GPGGA – Global Positioning Fix Data.....	35
Table: ranger2usbl_gpgga	35
USBLPSON.....	36
\$PSONLLD – Ranger 2 PSONLLD message	36
Table: usblpson_psonlld	36
Sperry Marine NAVITVIN IV Ship gyrocompasses	36
SHIPSGYRO	37
\$HEHDT – Heading – True Data	37
Table: shipsgyro_hehdt.....	37
\$TIROT – Rate of Turn	37
Table: shipsgyro_tirotd.....	37
\$PPLAN – Nav Status Data	38
Table: shipsgyro_pplan.....	38
\$GPGGA – Global Positioning Fix Data (limited to lat/lon and quality)	38
Table: shipsgyro_gpgga.....	38
\$GPVTG – Course Over Ground and Ground Speed Data (limited to speed).....	39
Table: shipsgyro_gpvtg.....	39
SKIPPER DL850 Skipper log (ship's velocity).....	39
SKIPPERLOG	39
\$VDVBW –Dual ground/water speed.....	40
Table: skipperlog_vdvw	40
\$VDVHW – Water speed & heading	40
Table: skipperlog_vdvhw	40
\$IIDPT – Depth of water	41
Table: skipperlog_iidpt.....	41
\$VDMTW – Water Temperature (not valid).....	41
Table: skipperlog_vdmtw	41
\$GPVTG – Course Over Ground and Ground Speed Data (limited to speed).....	41
Table: shipsgyro_gpvtg.....	42
Kongsberg Maritime Simrad EM122 Multibeam echo sounder (deep).....	42
EM122.....	42
\$KIDPT – Depth of water	42



Table: em122_kidpt	42
Kongsberg Maritime Simrad EA640 Single beam echo sounder 10 & 12KHz	43
EA640	43
\$SDDPT – Depth of water	43
Table: ea640_sddpt	43
\$SDDBS – Depth below surface	44
Table: ea640_sddbd	44
Cable Logging And Monitoring system.....	44
WINCH	44
\$WINCH – Cable logging system data.....	44
Table: winch_winch.....	44
NMF SURFMET Meteorology and Surface Hydrography Suite.....	45
SURFMET	45
\$GPXSM – Surfmet output message	46
Table: surfmet_gpxsm	46
SBE45	47
NOT NMEA ASCII OUTPUTS: 1	47
t1= – Thermosalinograph data (NOT NMEA LIKE)	47
Table: sbe45_nanan	47
Lambrecht THpro Humidity and Temperature Probe	48
ENVTEMP	48
\$WIMTA – Temperature data	48
Table: envtemp_wimta	48
\$WIMHU – Environment Humidity	48
Table: envtemp_wimhu	49
OceanWaves WaMoS II Wave Radar	49
WAMOS	49
\$PWAM - WAMOS WAVERADAR Output Data	49
Table: wamos_pwam	49
Appendix:	50
Current appconfig.json used by the RAM module:.....	50
Ingester configuration file example:	53
Netcdf Converter JSON configuration file example:.....	54



Sensorconfig file example:.....	59
Interpolate frequency options:.....	62
Crontab schedule example:	63
Talker Identifiers the first two character from MessageID	64
"Message" "Function"	64
How to:	65
How to use Grafana for visualisation or data export?	65

Modules

RVDAS RAM

The Raw Acquisition Module (RAM) is a component of the Research Vessel Data Acquisition System which receives records and re-transmits UDP ASCII data.

Version

RVDAS Build: 2019, RAM Major Version: 2 (2018), RAM Build: 7 (2020)

Changes

Build 2.7 (10/2020)

Overhauled Configuration menus. Removed experimental features. Added ability to configure filenames and output formats. Start acquisition now creates a copy of the RAM appconfig file in the data directory. Timestamp function overhauled and now zero-padded by default. Integration with Chrony for time synchronisation implemented, whilst integration for ntpd has been removed, as it has been removed from CentOS.

Build 2.6 (01/2019)

Added ability to set timeouts per channel. Added ability to broadcast individual channels to multiple targets. Enabled customisation of watchkeeper tick and filestream buffer size.

RVDAS RAM UserInterface

A web-based user interface for interacting with the raw acquisition module.

Version



2020-1b

RVDAS Ingestor

input: r2i-postgres.ini, Raw Acquisition Module broadcasted UDP messages from sensors (described in appconfig and Metadata), [sensor.json] output: PostgreSQL/TimescaleDB database, daily created file include all SQL sentences to PostgreSQL server

external modules: nmeacase.py - change input messages to nmea like messages zipdaily.py - compress the daily logfiles (scheduled with crontab)

Built With

Python3.7, (json 2.0.9 psutil 5.6.3 pycopg2 2.8.3 (dt dec pq3 ext lo64) pytz 2019.1 twisted 19.2.1)

RVDAS Netcdf Converter

netcdf_converter - Combine PostgreSQL (v10-12) database data with metadata and convert to RAW NetCDF4 format and clean interpolated NetCDF4, SQL and ascii CSV format with QC.

Sensor NMEA tableName(s) and fieldnames:

The tablename are decided by the following schema: sensorname+nmea eg. posmv sensor gpgga message ingested in table: posmv_gpgga. If the NMEA name had one or more '_' character it is replaced with '0'.

The first 3 fields are common in all table (time, sensorid, messageid) as you can see in the following example:

```
Table "public.shipsgyro_hehdt"
  Column          |          Type          | Collation | Nullable |          Default
-----+-----+-----+-----+-----
time              | timestamp with time zone |           | not null |
sensorid          | character varying      |           |          |
messageid         | character varying      |           |          |
headingtrue       | double precision       |           |          |
trueheading       | text                   |           |          |
flag_headingtrue  | smallint               |           |          | '-10'::integer
flag_trueheading  | smallint               |           |          | '-10'::integer
Indexes:
    "shipsgyro_hehdt_time_idx" btree ("time" DESC)
```

Applanix POS MV320 V5 Position/attitudePrimary scientific GPS

A combined GNSS receiver, gyrocompass, and conventional motion sensor. The GPS aspect is for use with Multibeam Echosounder systems. It provides data about: attitude; heave; position; and velocity



POSMV

NMEA ASCII OUTPUTS: 6

\$GPGGA
 \$GPHDT
 \$GPVTG
 \$GPRMC
 \$GPZDA
 \$PASHR
 \$GPGLL
 \$GPGST

Example:

21/07/2021 00:00:01.031	\$PASHR,000000.917,144.99,T,0.65,0.34,0.08,0.015,0.015,0.028,2,1*1B
21/07/2021 00:00:01.145	\$GPGLL,5416.38,N,01203.69,W,000000.92,A,D,*5B
21/07/2021 00:00:01.145	\$GPGST,000000.917,,0.1,0.1,0.0,0.1,0.1,0.2*44
21/07/2021 00:00:01.249	\$GPGGA,000000.917,5416.38135,N,01203.68961,W,5,02,0.8,57.88,M,,,5,0005*24
21/07/2021 00:00:01.249	\$GPHDT,145.0,T*35
21/07/2021 00:00:01.249	\$GPVTG,136.1,T,,M,1.0,N,1.8,K,D*05
21/07/2021 00:00:01.344	\$GPRMC,000000.92,A,5416.38135,N,01203.68961,W,1.0,145.0,210721,,,d*6E
21/07/2021 00:00:01.347	\$GPZDA,000001.0047,21,07,2021,,*61

\$GPGGA – Global Positioning Fix Data

Table: posmv_gpgga

example:

\$GPGGA,131156.543,4035.12170,N,01755.51529,W,2,08,1.0,8.27,M,,,0,0444*16

fieldnames:

\$GPGGA,utcTime, latitude, latDir, longitude, lonDir, ggaQual, numSat, hdop, altitude, unitsOfMeasureAntenna, geoidAltitude, unitsOfMeasureGeoid, diffcAge, dgnssRefId*checksum

utcTime: hhmmss.sss UTC of position fix - hours 2|minutes 2|seconds 2|decimal_seconds 3 digits

latitude: ddmm.mmmmm 0 to 90 degrees 2|minutes 2|decimal_minutes 5 digits

latDir: N or S – Latitude Cardinal Direction

longitude: dddmm.mmmmm 0 to 180 degrees 3|minutes 2|decimal_minutes 5 digits

lonDir: E or W – Longitude cardinal direction

ggaQual: 0-6 – GNNS quality indicator

numSat: 0-32 – Number of Satellites used in the fix

hdop: Horizontal Dillution of precision

altitude: Altitude above or below mean sea level unit=metres

unitsOfMeasureAntenna: null or M

geoidAltitude: Height of Geoid above WGS84 in meters

unitsOfMeasureGeoid: null or M

diffcAge: 0 to 999 – Age of differential correction in seconds since last RTCM-104

message unit: seconds

dgnssRefId: 0000 to 1023 – DGNSS reference station identity

Checksum: XOR checksum after *



\$GPHDT – Heading – True Data

Table: posmv_gphdt

example:

\$GPHDT,260.7,T36

fieldnames:

\$GPHDT,heading,true_heading*checksum

heading: 0 to 359.999 – True Vessel Heading in the vessel frame degrees

3|decimal_degrees 1 digit

trueHeading: T -True

Checksum: XOR checksum after *

\$GPVTG – Course Over Ground and Ground Speed Data

Table: posmv_gpvtg

example:

\$GPVTG,258.9,T,,M,10.4,N,19.2,K,D*01

fieldnames:

\$GPVTG, courseTrue, TrueCourse, magneticTrack, mFlag, speedKnots, n_flag,
speed_kmph, k_flag, positioningMode*checksum

courseTrue: 000.0 to 359.999 – True Vessel track in the vessel frame degrees

3|decimal_degrees 1 digit

trueCourse: T -True designation

magneticTrack: 000.0 to 359.999 – Course Over Ground Magnetic degrees

mFlag: M – Magnetic designation

speedKnots: Speed in the vessel frame unit: knots

nFlag: N – Knots designation

speedKmph: Speed in the vessel frame unit: km/h

kFlag: K – km/h designation

positioningMode: A=GPS used, D=DGPS used, E=dead reckoning, M=Manual Input Mode,

S=Simulated Mode, N=invalid

checksum: XOR checksum after *



\$GPRMC – RMC navigation data

Table: posmv_gprmc

example:

\$GPRMC,131156.54,A,4035.12170,N,01755.51529,W,10.4,260.7,251217,,d*56

fieldnames:

\$GPRMC, utcTime, vFlag, latitude, latDir, longitude, lonDir, speedKnots, trackMadeGood, navDate, magvar, magvarDirpositioningMode*checksum

utcTime: hhmmss.ssss UTC time of navigation data - hours 2|minutes 2|seconds 2|decimal_seconds 4 digits (max)

vFlag: A (valid) or V (Not valid) – Receiver Status

latitude: ddmm.mmmmm 0 to 90 degrees 2|minutes 2|decimal_minutes 5 digits.

latDir: N or S – Latitude Cardinal Direction

longitude: dddmm.mmmmm 0 to 180 degrees 3|minutes 2|decimal_minutes 5 digits.

lonDir: E or W – Longitude Cardinal Direction

speedKnots: dd.dd Speed Over Ground unit: knots

trackMadeGood: hhh.h – 0 to 359.9 – Track Made Good – unit: degrees

navDate: ddmmyy – UTC date of navigation data – days 2|months 2|year 2 digits.

magvar: d.d Magnetic Variation unit: degrees

magvarDir: E or W - direction of magnetic Variation

positioningMode: s Mode Indicator (A Autonomous or D Differential or E Dead reckoning or N Not valid)

Checksum: XOR checksum after *

\$GPZDA – Time and Date

Table: posmv_gpzda

example:

\$GPZDA,131157.008,25,12,2017,,*6E

fieldnames:

\$GPZDA, utcTime, day, month, year, zoneHour, zoneMinutes*checksum

utcTime: hhmmss.ssss UTC time hours 2|minutes 2|seconds 2|decimal_seconds 4 digits (max)

day: 01 to 31 Day of Month

month: 01 to 12 Month of year

year: YYYY Year



zoneHour: -13 to 13 Local zone hours
zoneMinutes: 00 to 59 Local zone minutes
Checksum: XOR checksum after *

\$PASHR – Attitude Data

Table: posmv_pashr

example:

\$PASHR,131156.543,260.67,T,-2.09,-1.55,-0.70,0.022,0.032,2,1*34

fieldnames:

\$PASHR, utcTime, heading, trueFlag, roll, pitch, heave, accuracy_roll, pitchAccuracy, accuracy_heading, flag_accuracy_heading, flag_IMU*checksum
utcTime: hhmmss.ssss UTC time hours 2|minutes 2|seconds 2|decimal_seconds 3 digits (max)
heading: 0 to 359.99 – Course Over Ground True – unit: degrees
trueFlag: T - True Designation
roll: -90.00 to 90.00 – Roll unit: degrees
pitch: -90.00 to 90.00 – Pitch unit: degrees
heave: -99.00 to 99.00 – Heave unit: metres
rollAccuracy: 0 to 9.999 – Accuracy of roll unit: degrees
pitchAccuracy: 0 to 9.999 – Accuracy of pitch unit: degrees
headingAccuracy: 0 to 9.999 – Accuracy of heading unit: degrees
headingAccuracyFlag: 0, 1, 2 – Flag Accuracy heading (0 = no aiding, 1 = GNSS aiding, 2 = GNSS & GAMS aiding)
imuFlag: 0, 1 – Flag IMU (0 = IMU out, 1 = satisfactory)
Checksum: XOR checksum after *

\$GPGLL – Position data: Position fix, time of position fix and status

Table: posmv_gpgll

example:

\$GPGLL,4034.176963,N,01800.760675,W,133628.23,A,D*61

fieldnames:

\$GPGLL, latitude, latDir, longitude, lonDir, utcTime, gllQual, positioningMode*checksum
latitude: ddmm.mmmmm 0 to 90 degrees 2|minutes 2|decimal_minutes 5 digits
latDir: N or S - Latitude Cardinal Direction
longitude: dddmm.mmmmm 0 to 180 degrees 3|minutes 2|decimal_minutes 5 digits



lonDir: E or W – Longitude Cardinal Direction

utcTime: hhmmss.ss UTC time hours 2|minutes 2|seconds 2|decimal_seconds 2 digits (max)

gllQual: A=valid, D=DGPS used, V=dead reckoning or invalid - Geographic Position Quality

positioningMode: s - Mode Indicator (A: Autonomous(GPS) or D: Differential (DGPS) or E: Dead reckoning or M: Manual Input Mode or S: Simulated Mode or N: Data Not Valid)

Checksum: XOR checksum after *

\$GPGST – Pseudorange Noise Statistics

Table: posmv_gpgst

example:

\$GPGST,000003.315,,0.1,0.1,0.0,0.1,0.1,0.2*4F

fieldnames:

\$GPGST, utcTime, rms, semiMajor, semiMinor, ellipseOrient, standardDeviationOfLatitude, standardDeviationOfLongitude, standardDeviationOfHeight*checksum

utcTime: hhmmss.sss UTC time hours 2|minutes 2|seconds 2|decimal_seconds 3 digits (max)

rms: d.d - RMS

semiMajor: d.ddd - Standard deviation of semi-major axis of error ellipse unit: metres

semiMinor: d.ddd - Standard deviation of semi-minor axis of error ellipse unit: metres

ellipseOrient: ddd.d - Orientation of semi major axis of error ellipse unit: degrees

standardDeviationOfLatitude: ddd.d - Standard Deviation of Latitude unit: metres

standardDeviationOfLongitude: ddd.d - Standard Deviation of Longitude unit: metres

standardDeviationOfHeight: d.ddd – Standard Deviation of Altitude unit: metres

Checksum: XOR checksum after *

Kongsberg Seatex Seapath 330 Position/attitude Secondary scientific GPS

This is a secondary Science GPS and attitude sensor. The position output is the position of the ship's common reference point (the cross on the top of the POSMV MRU in the Gravity Meter Room).

SEAPATHPOS

NMEA ASCII OUTPUTS: 7

\$INGGA

\$INHDT

\$INVTG



\$INRMC
 \$INZDA
 \$GNGST, \$GPGST

Example:

21/07/2021 09:07:49.711	\$INGGA,090749.00,5410.770649,N,01150.353142,W,2,08,1.1,0.04,M,58.72,M,17.2,0100*7D
21/07/2021 09:07:49.711	\$INVTG,146.89,T,,M,0.1,N,0.1,K,D*2A
21/07/2021 09:07:49.820	\$INRMC,090749.00,A,5410.770649,N,01150.353142,W,0.1,146.89,210721,,,D*62
21/07/2021 09:07:49.820	\$GPGST,090749.00,0.50,0.385,0.231,18,0.372,0.250,0.677*4F
21/07/2021 09:07:49.820	\$INHDT,74.72,T*23
21/07/2021 09:07:50.636	\$INZDA,090750.49,21,07,2021,,,75

\$INGGA – Global Positioning Fix Data

Table: seapathpos_ingga

example:

\$INGGA,000000.21,2138.561570,S,00930.646268,E,1,12,0.7,-1.73,M,23.37,M,,*41

fieldnames:

\$INGGA, utcTime, latitude, latDir, longitude, lonDir, ggaQual, numSat, hdop, altitude, unitsOfMeasureAntenna, geoidAltitude, unitsOfMeasureGeoid, diffcAge, dgnssRefId*checksum
 utcTime: hhmmss.sss UTC of position fix - hours 2|minutes 2|seconds 2|decimal_seconds 3 digits
 latitude: ddmm.mmmmm 0 to 90 degrees 2|minutes 2|decimal_minutes 5 digits
 latDir: N or S – Latitude Cardinal Direction
 longitude: dddmm.mmmmm 0 to 180 degrees 3|minutes 2|decimal_minutes 5 digits
 lonDir: E or W – Longitude cardinal direction
 ggaQual: 0-6 – GNNS quality indicator
 numSat: 0-32 – Number of Satellites used in the fix
 hdop: Horizontal Dillution of precision
 altitude: Altitude above or below mean sea level unit=metres
 unitsOfMeasureAntenna: null or M
 geoidAltitude: Height of Geoid above WGS84 in meters
 unitsOfMeasureGeoid: null or M
 diffcAge: 0 to 999 – Age of differential correction in seconds since last RTCM-104 message unit: seconds
 dgnssRefId: 0000 to 1023 – DGNS reference station identity
 Checksum: XOR checksum after *

\$INHDT – Heading – True Data

Table: seapathpos_inhdt



example:

\$INHDT,144.23,T*15

fieldnames:

\$INHDT,headingTrue,trueHeading*checksum

headingTrue: 0 to 359.99 – True Vessel Heading in the vessel frame degrees

3|decimal_degrees 1 digit

trueHeading: T -True

Checksum: XOR checksum after *

\$INVTG – Course Over Ground and Ground Speed Data

Table: seapathpos_invtg

example:

\$INVTG,194.87,T,,M,0.6,N,1.2,K,A*2B

fieldnames:

\$INVTG, courseTrue, TrueCourse, magneticTrack, mFlag, speedKnots, n_flag, speed_kmph, k_flag, positioningMode*checksum

courseTrue: 000.0 to 359.999 – True Vessel track in the vessel frame degrees

3|decimal_degrees 1 digit

trueCourse: T –True designation

magneticTrack: 000.0 to 359.999 – Course Over Ground Magnetic degrees

mFlag: M – Magnetic designation

speedKnots: Speed in the vessel frame unit: knots

nFlag: N – Knots designation

speedKmph: Speed in the vessel frame unit: km/h

kFlag: K – km/h designation

positioningMode: A=GPS used, D=DGPS used, E=dead reckoning, M=Manual Input Mode, S=Simulated Mode, N=invalid

checksum: XOR checksum after *

\$INRMC – RMC navigation data

Table: seapathpos_inrmc

example:



\$INRMC,000002.21,A,2138.561892,S,00930.646142,E,0.9,187.81,270518,,A*65

fieldnames:

\$INRMC, utcTime, vFlag, latitude, latDir, longitude, lonDir, speedKnots, heading, navDate, magvar, magvarDir, positioningMode*checksum

utcTime: hhmmss.ssss UTC time of navigation data - hours 2|minutes 2|seconds 2|decimal_seconds 4 digits (max)

vFlag: A (valid) or V (Not valid) – Receiver Status

latitude: ddmm.mmmmm 0 to 90 degrees 2|minutes 2|decimal_minutes 5 digits.

latDir: N or S – Latitude Cardinal Direction

longitude: dddmm.mmmmm 0 to 180 degrees 3|minutes 2|decimal_minutes 5 digits.

lonDir: E or W – Longitude Cardinal Direction

speedKnots: dd.dd Speed Over Ground unit: knots

heading: hhh.h – 0 to 359.9 – Heading – unit: degrees

navDate: ddmmyy – UTC date of navigation data – days 2|months 2|year 2 digits.

magvar: d.d Magnetic Variation unit: degrees

magvarDir: E or W - direction of magnetic Variation

positioningMode: s Mode Indicator (A Autonomous or D Differential or E Dead reckoning or N Not valid)

Checksum: XOR checksum after *

\$INZDA – Time and Date

Table: seapathpos_inzda

example:

\$INZDA,000003.21,27,05,2018,,*7D

fieldnames:

\$INZDA,utcTime,day,month,year,zoneHour,zoneMinutes*checksum

utcTime: hhmmss.ssss UTC time hours 2|minutes 2|seconds 2|decimal_seconds 4 digits (max)

day: 01 to 31 Day of Month

month: 01 to 12 Month of year

year: YYYY Year

zoneHour: -13 to 13 Local zone hours

zoneMinutes: 00 to 59 Local zone minutes

Checksum: XOR checksum after *

\$GNGST – GPS Pseudorange Noise Statistics



Table: seapathpos_gngst, seapathpos_gpgst

example:

\$GNGST,000001.21,3.14,0.680,0.576,47,0.627,0.634,1.26*6D

fieldnames:

\$GPGST, utcTime, rms, semiMajor, semiMinor, ellipseOrient,
standardDeviationOfLatitude, standarDeviationofLongitude,
standardDeviationOfHeight*checksum

utcTime: hhmmss.sss UTC time hours 2|minutes 2|seconds 2|decimal_seconds 3 digits
(max)

rms: d.d - RMS

semiMajor: d.ddd - Standard deviation of semi-major axis of error ellipse unit: metres

semiMinor: d.ddd - Standard deviation of semi-minor axis of error ellipse unit: metres

ellipseOrient: ddd.d - Orientation of semi major axis of error ellipse unit: degrees

standardDeviationOfLatitude: ddd.d - Standard Deviation of Latitude unit: metres

standardDeviationOfLongitude: ddd.d - Standard Deviation of Longitude unit: metres

standardDeviationOfHeight: d.ddd - Standard Deviation of Altitude unit: metres

Checksum: XOR checksum after *

SEAPATHATT

NMEA ASCII OUTPUTS: 4

\$INZDA

\$INGGA

\$PSXN,20

\$PSXN,23

Example:

21/07/2021 10:35:14.548 \$INZDA,103514.55,21,07,2021,,*71

21/07/2021 10:35:14.548 \$INGGA,103514.55,5410.770984,N,01150.353275,W,2,08,1.0,0.90,M,58.72,M,8.6,0100*43

21/07/2021 10:35:14.548 \$PSXN,20,0,0,0,0*3B

21/07/2021 10:35:14.548 \$PSXN,23,-0.81,0.13,74.71,-0.02*04

\$INZDA – Time and Date

Table: seapathatt_inzda

example:

\$INZDA,000003.21,27,05,2018,,*7D

fieldnames:

\$INZDA,utcTime,day,month,year,zoneHour,zoneMinutes*checksum

utcTime: hhmmss.ssss UTC time hours 2|minutes 2|seconds 2|decimal_seconds



4 digits (max)
 day: 01 to 31 Day of Month
 month: 01 to 12 Month of year
 year: YYYY Year
 zoneHour: -13 to 13 Local zone hours
 zoneMinutes: 00 to 59 Local zone minutes
 Checksum: XOR checksum after *

\$INGGA – Global Positioning Fix Data

Table: seapathpos_ingga

example:

\$INGGA,000000.21,2138.561570,S,00930.646268,E,1,12,0.7,-1.73,M,23.37,M,,*41

fieldnames:

\$INGGA, utcTime, latitude, latDir, longitude, lonDir, ggaQual, numSat, hdop, altitude, unitsOfMeasureAntenna, geoidAltitude, unitsOfMeasureGeoid, diffcAge, dgnssRefId*checksum
 utcTime: hhmmss.sss UTC of position fix - hours 2|minutes 2|seconds 2|decimal_seconds 3 digits
 latitude: ddmm.mmmmm 0 to 90 degrees 2|minutes 2|decimal_minutes 5 digits
 latDir: N or S – Latitude Cardinal Direction
 longitude:dddmm.mmmmm 0 to 180 degrees 3|minutes 2|decimal_minutes 5 digits
 lonDir: E or W – Longitude cardinal direction
 ggaQual: 0-6 – GNNS quality indicator
 numSat: 0-32 – Number of Satellites used in the fix
 hdop: Horizontal Dillution of precision
 altitude: Altitude above or below mean sea level unit=metres
 unitsOfMeasureAntenna: null or M
 geoidAltitude: Height of Geoid above WGS84 in meters
 unitsOfMeasureGeoid: null or M
 diffcAge: 0 to 999 – Age of differential correction in seconds since last RTCM-104 message unit: seconds
 dgnssRefId: 0000 to 1023 – DGNSS reference station identity
 Checksum: XOR checksum after *

\$PSXN,23 – Roll, Pitch, Heading and Heave observations

Table: seapathatt_psx23



example:

\$PSXN,23,1.02,-1.70,133.25,-0.42*3D

fieldnames:

\$PSXN, mruFlag, roll, pitch, heading, heave*checksum

mruFlag: always 23! It is part of the message ID, but separated because of the comma delimiter.

roll: dd.dd - roll positive port side up unit: degrees

pitch: dd.dd - pitch positive with bow up unit: degrees

heading: ddd.dd 000.00-359.99 – Sensor Heading unit degrees

heave: ddd.dd Heave positive down unit: metres

Checksum: XOR checksum after *

\$PSXN,20 – Quality for Roll, Pitch, Heading and Heave observations

Table: seapathatt_psx20

example:

\$PSXN,20,0,0,0,0*3B

fieldnames:

\$PSXN, mruFlag, rollPitchQuality, headingQuality, heightQuality, horizontalPositionQuality*checksum

mruFlag: always 20! It is part of the message ID, but separated because of the comma delimiter.

rollPitchQuality: s - rollPitchQuality (0: normal; 1: reducedPerformance; 2: invalidData)

headingPitchQuality: s - headingQuality (0: normal; 1: reducedPerformance; 2: invalidData)

heightQuality: s - heightQuality (0: normal; 1: reducedPerformance; 2: invalidData)

horizontalPositionQuality: s - horizontalPositionQuality (0: normal; 1: reducedPerformance; 2: invalidData)

Checksum: XOR checksum after *

C-Nav3050 DGNSS (L2 correction to Applanix PosMV)

GPS and RTCM Satellite Corrections Receiver. The position output is the position of the antenna. This GPS is not referenced to any other systems. It is primarily used to provide RTCM differential corrections to the Applanix PosMV GPS system. Please note that the position output is the position of the antenna.

CNAV

NMEA ASCII OUTPUTS: 4

\$GNGGA



\$GNVTG
 \$GNDTM
 \$GNGST

Example:

21/07/2021 09:12:08.220	\$GNGGA,091208.00,5410.772757,N,01150.341031,W,2,16,0.7,89.367,M,0.0,M,4.0,0525*4C
21/07/2021 09:12:08.220	\$GNVTG,135.8,T,M,0.35,N,0.65,K,D*1C
21/07/2021 09:12:08.235	\$GNGST,091208.00,0.3259,0.0444,0.0318,018.5518,0.0433,0.0333,0.0805*72
21/07/2021 09:12:09.111	\$GNDTM,999,,,,,999*54

\$GNGGA – Global Positioning Fix Data

Table: *cnav_gngga*

example:

\$GNGGA,000000.00,2138.567643,S,00930.651093,E,2,18,0.7,55.321,M,0.0,M,5.0,0525*44

fieldnames:

\$GNGGA, utcTime, latitude, latDir, longitude, lonDir, ggaQual, numSat, hdop, altitude, unitsOfMeasureAntenna, geoidAltitude, unitsOfMeasureGeoid, diffcAge, dgnssRefId*checksum

utcTime: hhmmss.sss UTC of position fix - hours 2|minutes 2|seconds

2|decimal_seconds 3 digits

latitude: ddmm.mmmmm 0 to 90 degrees 2|minutes 2|decimal_minutes 5 digits

latDir: N or S – Latitude Cardinal Direction

longitude: dddmm.mmmmm 0 to 180 degrees 3|minutes 2|decimal_minutes 5 digits

lonDir: E or W – Longitude cardinal direction

ggaQual: 0-6 – GNNS quality indicator

numSat: 0-32 – Number of Satellites used in the fix

hdop: Horizontal Dilution of precision

altitude: Altitude above or below mean sea level unit=metres

unitsOfMeasureAntenna: null or M

geoidAltitude: Height of Geoid above WGS84 in meters

unitsOfMeasureGeoid: null or M

diffcAge: 0 to 999 – Age of differential correction in seconds since last RTCM-104

message unit: seconds

dgnssRefId: 0000 to 1023 – DGNSS reference station identity

Checksum: XOR checksum after *

\$GNVTG – Course Over Ground and Ground Speed Data

Table: *cnav_gnvtg*



example:

\$GNVTG,293.9,T,,M,1.18,N,2.19,K,D*15

fieldnames:

\$GNVTG, courseOverGround, TrueCourse, magneticTrack, mFlag, speedKnots, n_flag, speed_kmph, k_flag, positioningMode*checksum

courseOverGround: 000.0 to 359.999 – True Vessel track in the vessel frame degrees
3|decimal_degrees 1 digit

trueCourse: T –True designation

magneticTrack: 000.0 to 359.999 – Course Over Ground Magnetic degrees

mFlag: M – Magnetic designation

speedKnots: Speed in the vessel frame unit: knots

nFlag: N – Knots designation

speedKmph: Speed in the vessel frame unit: km/h

kFlag: K – km/h designation

positioningMode: FAA Mode Indicator A=GPS used, D=DGPS used, E=dead reckoning,
M=Manual Input Mode, S=Simulated Mode, N=invalid

checksum: XOR checksum after *

\$GNDTM – Datum being used

Table: cnav_gndtm

example:

\$GNDTM,999,,,,,,999*54

fieldnames:

\$GNDTM, datumCode, subdatumCode, latOffset, latDir, lonOffset, lonDir, altitudeOffset,
referenceDatumCode *checksum

datumCode: Local Datum Code - W84 or W72 or S85 or PE90 or 999=user defined

subdatumCode: Local datum subdivision code if available unit: minutes

latOffset: Latitude offset from reference position in minutes

latDir: N or S Cardinal Direction of latitude

lonOffset: Longitude offset from reference position in minutes

lonDir: E or W Cardinal Direction of longitude

altitudeOffset: Altitude offset from reference position in meters

referenceDatumCode: W84 or W72 or S85 or PE90 or 999 user defined

Checksum: XOR checksum after *



\$GNGST – GPS Pseudorange Noise Statistics

Table: *cnav_gpgst, cnav_gngst*

example:

\$GPGST,000001.21,3.14,0.680,0.576,47,0.627,0.634,1.26*6D

\$GNGST,000001.21,0.3856,0.0412,0.0312,359.7027,0.0412,0.0312,0.0691*72

fieldnames:

\$GNGST, utcTime, rms, semiMajor, semiMinor, ellipseOrient,
standardDeviationOfLatitude, standarDeviationofLongitude,
standardDeviationOfHeight*checksum

utcTime: hhmmss.sss UTC time hours 2|minutes 2|seconds 2|decimal_seconds 3 digits
(max)

rms: d.d - RMS

semiMajor: d.ddd - Standard deviation of semi-major axis of error ellipse unit: metres

semiMinor: d.ddd - Standard deviation of semi-minor axis of error ellipse unit: metres

ellipseOrient: ddd.d - Orientation of semi major axis of error ellipse unit: degrees

standardDeviationOfLatitude: ddd.d - Standard Deviation of Latitude unit: metres

standardDeviationOfLongitude: ddd.d - Standard Deviation of Longitude unit: metres

standardDeviationOfHeight: d.ddd – Standard Deviation of Altitude unit: metres

Checksum: XOR checksum after *

iXSea PHINSIII Inertial Navigation System

A surface inertial navigation system that uses a FOG (Fibre-Optic Gyro) to output accurate position, attitude, and velocity data.

PHINS

NMEA ASCII OUTPUTS: 2

\$PASHR

\$PRDID* replaced with PIXSE.ATITUD and HEHDT

\$HEHDT

\$HETHS

\$PIXSE.ATITUD

\$PIXSE.POSITI

\$PIXSE.SPEED_

\$PIXSE.UTMWGS

\$PIXSE.HEAVE_

\$PIXSE.TIME__

\$PIXSE.STDHRP

\$PIXSE.STDPOS

\$PIXSE.STDSPD

\$PIXSE.UTCIN_

\$PIXSE.GPSIN_



\$PIXSE.GP2IN_
\$PIXSE.ALGTS
\$PIXSE.STATUS
\$PIXSE.HT_STS

Example NMEA (all sentences with HEHDT timestamp!!!):

```
21/07/2021 06:50:01.700    $HEHDT,74.21,T*2F
$HETHS,74.21,A*2D
$PIXSE,ATITUD,-0.082,-0.174*6A
$PIXSE,POSITI,54.17951609,348.16078205,0.666*52
$PIXSE,SPEED_,0.000,0.023,-0.139*46
$PIXSE,UTMWGS,U,29,314716.052,6007218.730,0.666*11
$PIXSE,HEAVE_,0.000,0.000,-0.111*55
$PIXSE,TIME_,065001.691466*64
$PIXSE,STDHRP,0.015,0.000,0.000*74
$PIXSE,STDPOS,0.03,0.03,4.78*4D
$PIXSE,STDSPD,0.002,0.002,0.928*7E
$PIXSE,UTCIN_,065001.490000*6C
$PIXSE,GPSIN_,54.17951111,348.16079298,-1.419,065000.000003,2*67
$PIXSE,GP2IN_,54.17951617,348.16078166,58.029,065000.896003,5*12
$PIXSE,ALGTS,00003041,00000030*60
$PIXSE,STATUS,00000600,0000060C*1C
$PIXSE,HT_STS,E7F15551*41
```

\$PASHR – Attitude Data

Table: phins_pashr

example:

\$PASHR,075742.076,133.25,T,+2.24,+0.33,+0.71,0.006,0.006,0.066,1,0*3C

fieldnames:

\$PASHR, utcTime, heading, trueFlag, roll, pitch, heave, rollAccuracy, pitchAccuracy, headingAccuracy, headingAccuracyFlag, imuFlag*checksum

utcTime: hhmmss.ssss UTC time hours 2|minutes 2|seconds 2|decimal_seconds
3 digits (max)

heading: 0 to 359.99 – True vessel heading – unit: degrees

trueFlag: T True

roll: -90.00 to 90.00 – Roll unit: degrees

pitch: -90.00 to 90.00 – Pitch unit: degrees

heave: -99.00 to 99.00 – Heave unit: metres

rollAccuracy: 0 to 9.999 – Accuracy roll unit: degrees

pitchAccuracy: 0 to 9.999 – Accuracy pitch unit:degrees

headingAccuracy: 0 to 9.999 – Accuracy heading unit: degrees

headingAccuracyFlag: 0, 1, 2 – Flag Accuracy heading (0 = no aiding, 1 = GNSS
aiding, 2 = GNSS & GAMS aiding)

imuFlag: 0, 1 – Flag IMU (0 = IMU out, 1 = satisfactory)

Checksum: XOR checksum after *

\$PRDID – attitude data* replaced with PIXSE.ATITUD and HEHDT



Table: phins_prdid

example:

\$PRDID,-1.55,-2.09,260.67*76

fieldnames:

\$PRDID,pitch,roll,heading*checksum

pitch: Pitch unit=degrees

roll: Roll unit=degrees

heading: Heading 0 to 359.99 unit=degrees

Checksum: XOR checksum after *

\$HEHDT – Heading – True Data

Table: phins_hehdt

example:

\$HEHDT,74.21,T*2F

fieldnames:

\$HEHDT, headingTrue, trueHeading*checksum

headingTrue: 0 to 359.99 – True Vessel Heading in the vessel frame degrees

3|decimal_degrees 1 digit

trueHeading: T=True

Checksum: XOR checksum after *

\$HEHTS – Heading – Data

Table: phins_hehts

example:

\$HEHTS,74.21,A*2D

fieldnames:

\$HEHTS, headingTrue, trueHeading*checksum

headingTrue: 0 to 359.99 – True Vessel Heading in the vessel frame degrees

3|decimal_degrees 1 digit

trueHeading: T=True

Checksum: XOR checksum after *

\$PIXSE,ATITUD Roll Pitch Attitude Data

Table: phins_pixseattitud

example:



\$PIXSE,ATITUD,-0.082,-0.174*6A

fieldnames:

\$ PIXSE,ATITUD, roll, pitch*checksum

roll: -180 to 180 – is the roll in degrees 3|decimal_degrees 3 digit

pitch: -90 to 90 – is the pitch in degrees 3|decimal_degrees 3 digit

Checksum: XOR checksum after *

\$PIXSE,POSITI Latitude, longitude, altitude Data

Table: phins_pixsepositi

example:

\$PIXSE,POSITI,54.17951609,348.16078205,0.666*52

fieldnames:

\$ PIXSE,POSITI, latitudeDD, longitudeDD, altitude*checksum

latitudeDD: is the latitude in decimal degrees, 8 digits after decimal point

longitudeDD: is the longitude in decimal degrees, 8 digits after decimal point

altitude: is the altitude in meters, 3 digits after decimal point

Checksum: XOR checksum after *

\$PIXSE,SPEED_ East, North, Up Speed Data

Table: phins_pixsespeed0

example:

\$PIXSE,SPEED_,0.000,0.023,-0.139*46

fieldnames:

\$ PIXSE,SPEED_, xEast, xNorth, xUp*checksum

xEast: is the East speed in m/s, 3 digits after decimal point

xNorth: is the North speed in m/s, 3 digits after decimal point

xUp: is the vertical speed in m/s (+ towards up side), 3 digits after decimal point

Checksum: XOR checksum after *

\$PIXSE,UTMWGS UTM Zone Related Data

Table: phins_pixseutmwgs

example:

\$PIXSE,UTMWGS,U,29,314716.052,6007218.730,0.666*11

fieldnames:

\$ PIXSE,UTMWGS, latitudeUTMZone, longitudeUTMZone, eastPosition, northPosition, altitude*checksum

latitudeUTMZone: is the latitude UTM zone (char)



longitudeUTMZone: is the longitude UTM Zone (integer)
eastPosition: is the East UTM position in meter, 3 digits after decimal point
northPosition: is the North UTM position in meter, 3 digits after decimal point
altitude: is the altitude in meter, 3 digits after decimal point
Checksum: XOR checksum after *

\$PIXSE,HEAVE_ surge, sway and heave Data

Table: phins_pixseheave_

example:

\$PIXSE,HEAVE_,0.000,0.000,-0.111*55

fieldnames:

\$ PIXSE,HEAVE_ surge, sway, heave*checksum

surge: is the surge in meters, 3 digits after decimal point

sway: is the sway in meters, 3 digits after decimal point

heave: is the heave in meters, 3 digits after decimal point

Checksum: XOR checksum after *

\$PIXSE,TIME__ UTC time reference Data

Table: phins_pixsetime00

example:

\$PIXSE,TIME__,065001.691466*64

fieldnames:

\$ PIXSE,TIME__, *checksum

UTCtime: is the validity time of the computed data transmitted in the UTC time

reference frame if available otherwise in the system time reference frame. 6 digits after decimal point.

Checksum: XOR checksum after *

\$PIXSE,STDHRP Heading, Roll and Pitch Standard Deviation Data

Table: phins_pixsestdhrp

example:

\$PIXSE,STDHRP,0.015,0.000,0.000*74

fieldnames:

\$ PIXSE,STDHRP, headingStd, rollStd, pitchStd*checksum

headingStd: is the heading standard deviation in degrees, 3 digits after decimal point

rollStd: is the roll standard deviation in degrees, 3 digits after decimal point

pitchStd: is the pitch standard deviation in degrees, 3 digits after decimal point



Checksum: XOR checksum after *

\$PIXSE,STDPOS Latitude, Longitude and Altitude Standard Deviation Data

Table: phins_pixsestdpos

example:

\$PIXSE,STDPOS,0.03,0.03,4.78*4D

fieldnames:

\$ PIXSE,STDPOS, latitudeStd, longitudeStd, altitudeStd*checksum

latitudeStd: is the latitude standard deviation in meters, 2 digits after decimal point

longitudeStd: is the longitude standard deviation in meters, 2 digits after decimal point

altitudeStd: is the altitude standard deviation in meters, 2 digits after decimal point

Checksum: XOR checksum after *

\$PIXSE,STDSPD North Speed, East Speed and Vertical Speed Standard Deviation Data

Table: phins_pixsestdspd

example:

\$PIXSE,STDSPD,0.002,0.002,0.928*7E

fieldnames:

\$ PIXSE,STDPOS, northSpeedStd, eastSpeedStd, verticalSpeedStd*checksum

northSpeedStd: is the North Speed standard deviation in m/s, 3 digits after decimal point

eastSpeedStd: is the East Speed standard deviation in m/s, 3 digits after decimal point

verticalSpeedStd: is the Vertical Speed standard deviation in m/s, 3 digits after decimal point

Checksum: XOR checksum after *

\$PIXSE,UTCIN_ UTC time received Data

Table: phins_pixseutcin0

example:

\$PIXSE,UTCIN_,065001.490000*6C

fieldnames:

\$ PIXSE,UTCIN_, UTCTime *checksum

UTCTime: is the UTC time received. 6 digits after decimal point.

Checksum: XOR checksum after *

\$PIXSE,GPSIN_ Latitude, Longitude, Altitude, UTCTime and quality Data Received from the Manual GPS



Table: phins_pixsegpsin0

example:

\$PIXSE,GPSIN_,54.17951111,348.16079298,-1.419,065000.000003,2*67

fieldnames:

\$ PIXSE,GPSIN_, latitudeDD, longitudeDD, altitude, UTCtime, qualityIndicator*checksum

latitudeDD: is the latitude in decimal degrees, 8 digits after decimal point

longitudeDD: is the longitude in decimal degrees, 8 digits after decimal point

altitude: is the altitude in meters, 3 digits after decimal point

UTCtime: is the validity time of the GPS data received in the UTC time reference frame if available otherwise in the system time reference frame. 6 digits after decimal point.

qualityIndicator: is the GPS quality indicator (0 and >=5 - Fix not valid; 1 - GPS SPS Mode Fix not valid; 2 - Differential Mode, SPS Mode, Fix not valid; 3 - GPS PPS Mode, Fix not valid; 4 - GPS RTK Mode)

Checksum: XOR checksum after *

\$PIXSE,GP2IN_ Latitude, Longitude, Altitude, UTCtime and quality Data Received from the GPS2 Sensor

Table: phins_pixsegp2in0

example:

\$PIXSE,GP2IN_,54.17951617,348.16078166,58.029,065000.896003,5*12

fieldnames:

\$ PIXSE,GP2IN_, latitudeDD, longitudeDD, altitude, UTCtime,

qualityIndicator*checksum

latitudeDD: is the latitude in decimal degrees, 8 digits after decimal point

longitudeDD: is the longitude in decimal degrees, 8 digits after decimal point

altitude: is the altitude in meters, 3 digits after decimal point

UTCtime: is the validity time of the GPS 2 data received in the UTC time reference frame if available otherwise in the system time reference frame. 6 digits after decimal point.

qualityIndicator: is the GPS quality indicator (0 and >=5 - Fix not valid; 1 - GPS SPS Mode Fix not valid; 2 - Differential Mode, SPS Mode, Fix not valid; 3 - GPS PPS Mode, Fix not valid; 4 - GPS RTK Mode)

Checksum: XOR checksum after *

\$PIXSE,ALGSTS INS Algo Status

Table: phins_pixsealgsts

example:

\$PIXSE,ALGSTS,00003041,00000030*60

fieldnames:

\$ PIXSE,ALGSTS, status1LSB, status2MSB*checksum



Status1LSB: is the hexadecimal value of INS Algo status1 (LSB)
Status2MSB: is the hexadecimal value of INS Algo status2 (MSB)
Checksum: XOR checksum after *

\$PIXSE,STATUS INS System Status

Table: phins_pixsestatus

example:

\$PIXSE,STATUS,00000600,0000060C*1C

fieldnames:

\$ PIXSE,ALGSTS, status1LSB, status2MSB*checksum

Status1LSB: is the hexadecimal value of INS system status1 (LSB)

Status2MSB: is the hexadecimal value of INS system status2 (MSB)

Checksum: XOR checksum after *

\$PIXSE,HT_STS INS High Level Status

Table: phins_pixseht0sts

example:

\$PIXSE,HT_STS,E7F15551*41

fieldnames:

\$ PIXSE,HT_STS, status1HighLevel*checksum

Status1HighLevel: is the hexadecimal value of INS High Level Status

Checksum: XOR checksum after *

Fugro 9205 DGNS S Seastar (L2 correction for Seapath330)

FUGRO 9205 Integrated GNSS Receiver with built-in L-Band correction receiver

Fully operational GNSS systems GPS is complemented by GLONASS(GLObalnaya NAVigatsionnaya Sputnikovaya Sistema, operated by the Russian Federation) as well as BeiDou and Galileo (see below). Most GNSS systems will offer global coverage and similar degrees of accuracy, ultimately their combined constellations are expected to number more than 100 in-service satellites.

This system gives L2 correction to the Seapath330 GPS.

FUGRO

NMEA ASCII OUTPUTS: 4

\$GPGGA

\$GPVTG

\$GPD TM

\$GPRMC



\$GPGSV
\$GLGSV
\$GPGLL
\$GNGSA

Example: NMEA (all sentences with GPGBA timestamp!!!):

```
21/07/2021 10:39:56.020 $GPGBA,103956.0,5410.77293,N,01150.34024,W,5,13,0.8,32.0,M,57.8,M,23.0,1015*65
$GPGLL,5410.77292616,N,01150.34024047,W,103956.00,A,D*79
$GNGSA,A,3,27,7,8,10,16,21,23,30,,,,,1.5,0.8,1.3*2F
$GNGSA,A,3,84,68,73,83,85,,,,,1.5,0.8,1.3*22
$GPGSV,4,1,15,7,22,275,48,27,73,101,50,30,18,312,37,8,71,284,53*7F
$GPGSV,4,2,15,10,46,098,48,16,32,163,45,21,35,233,47,23,33,055,45*72
$GLGSV,4,3,15,68,29,086,43,83,35,062,38,73,14,193,39,84,72,303,41*64
$GLGSV,4,4,15,67,24,029,44,74,48,235,47,85,17,261,44*5B
$GPRMC,103956.00,A,5410.77293,N,01150.34024,W,0.190,179.2,210721,4.8,W,D*3A
$GPVTG,179.16,T,183.97,M,0.19,N,0.35,K,D*24
$GPDTM,W84,,0.0,N,0.0,W,0.0,W84*7D
```

\$GPGBA – Global Positioning Fix Data

Table: *fugro_gpgga*

example:

```
$GPGBA,131156.543,4035.12170,N,01755.51529,W,2,08,1.0,8.27,M,,,0,0444*16
```

fieldnames:

\$GPGBA,utcTime, latitude, latDir, longitude, lonDir, ggaQual, numSat, hdop, altitude,
unitsOfMeasureAntenna, geoidAltitude, unitsOfMeasureGeoid, diffcAge,
dgnssRefId*checksum

utcTime: hhmmss.sss UTC of position fix - hours 2|minutes 2|seconds
2|decimal_seconds 3 digits

latitude: ddmm.mmmmm 0 to 90 degrees 2|minutes 2|decimal_minutes 5 digits

latDir: N or S – Latitude Cardinal Direction

longitude: dddmm.mmmmm 0 to 180 degrees 3|minutes 2|decimal_minutes 5 digits

lonDir: E or W – Longitude cardinal direction

ggaQual: 0-6 – GNNS quality indicator

numSat: 0-32 – Number of Satellites used in the fix

hdop: Horizontal Dillution of precision

altitude: Altitude above or below mean sea level unit=metres

unitsOfMeasureAntenna: null or M

geoidAltitude: Height of Geoid above WGS84 in meters

unitsOfMeasureGeoid: null or M

diffcAge: 0 to 999 – Age of differential correction in seconds since last RTCM-104

message unit: seconds

dgnssRefId: 0000 to 1023 – DGNSS reference station identity

Checksum: XOR checksum after *



\$GNVTG – Course Over Ground and Ground Speed Data

Table: fugro_gpvtg

example:

\$GPVTG,,,,,-00.1,N,-00.2,K,A*39

fieldnames:

\$GNVTG, courseOverGround, trueCourse, magneticTrack, mFlag, speedKnots, nFlag, speedKmph, kFlag, positioningMode*checksum

courseOverGround: 000.0 to 359.9 – True Vessel track in the vessel frame degrees 3|decimal_degrees 1 digit

trueCourse: T -True

magneticTrack: 0-359.9 Magnetic track unit: degrees

mFlag: M

speedKnots: 0-1000 Speed in the vessel frame unit: knots

nFlag: N

speedKmph: 0-1852 Speed in the vessel frame unit: km/h

kFlag: K Speed over ground unit K=km/h

positioningMode: A=GPS used, D=DGPS used, E=dead reckoning, N=invalid, S=simulator, P=precise (P NMEA4.1 only)

checksum: XOR checksum after *

\$GNDTM – Datum being used

Table: fugro_gpdtm

example:

\$GPDTM,999,,,,,,999*54

fieldnames:

\$GPDTM, datumCode, subdatumCode, latOffset, latDir, lonOffset, lonDir, altitudeOffset, referenceDatumCode *checksum

datumCode: Local Datum Code - W84 or W72 or S85 or PE90 or 999=user defined

subdatumCode: Local datum subdivision code if available unit: minutes

latOffset: Latitude offset from reference position in minutes

latDir: N or S Cardinal Direction of latitude

lonOffset: Longitude offset from reference position in minutes

lonDir: E or W Cardinal Direction of longitude

altitudeOffset: Altitude offset from reference position in meters

referenceDatumCode: W84 or W72 or S85 or PE90 or 999 user defined

Checksum: XOR checksum after *



\$GPRMC – RMC navigation data

Table: fugro_gprmc

example:

\$GPRMC,131156.54,A,4035.12170,N,01755.51529,W,10.4,260.7,251217,,d*56

fieldnames:

\$GPRMC, utcTime, vFlag, latitude, latDir, longitude, lonDir, speedKnots, trackMadeGood, navDate, magvar, magvarDirpositioningMode*checksum

utcTime: hhmmss.ssss UTC time of navigation data - hours 2|minutes 2|seconds 2|decimal_seconds 4 digits (max)

vFlag: A (valid) or V (Not valid) – Receiver Status

latitude: ddmm.mmmmm 0 to 90 degrees 2|minutes 2|decimal_minutes 5 digits.

latDir: N or S – Latitude Cardinal Direction

longitude: dddmm.mmmmm 0 to 180 degrees 3|minutes 2|decimal_minutes 5 digits.

lonDir: E or W – Longitude Cardinal Direction

speedKnots: dd.dd Speed Over Ground unit: knots

trackMadeGood: hhh.h – 0 to 359.9 – Track Made Good – unit: degrees

navDate: ddmmyy – UTC date of navigation data – days 2|months 2|year 2 digits.

magvar: d.d Magnetic Variation unit: degrees

magvarDir: E or W - direction of magnetic Variation

positioningMode: s Mode Indicator (A Autonomous or D Differential or E Dead reckoning or N Not valid)

Checksum: XOR checksum after *

\$GPGSV – Number of SVs in view, PRN, elevation, azimuth and SNR

Table: fugro_gpgsv, fugro_glgsv

GSV - Satellites in View shows data about the satellites that the unit might be able to find based on its viewing mask and almanac data. It also shows current ability to track this data. Note that one GSV sentence only can provide data for up to 4 satellites and thus there may need to be 9 sentences for the full information. It is reasonable for the GSV sentence to contain more satellites than GGA might indicate since GSV may include satellites that are not used as part of the solution. It is not a requirement that the GSV sentences all appear in sequence. To avoid overloading the data bandwidth some receivers may place the various sentences in totally different samples since each sentence identifies which one it is.

The field called SNR (Signal to Noise Ratio) in the NMEA standard is often referred to as signal strength. SNR is an indirect but more useful value than raw signal strength. It can range from 0 to 99 and has units of dB according to the NMEA standard, but the various manufacturers send different ranges of numbers with different starting numbers so the values themselves cannot necessarily be used to evaluate different units. The range of working values in a given GPS will usually



show a difference of about 25 to 35 between the lowest and highest values, however 0 is a special case and may be shown on satellites that are in view but not being tracked.

example:

\$GPGSV,7,3,23,28,07,125,39,29,07,286,44,30,61,090,51*44

\$GLGSV,7,4,23,69,30,080,51,70,59,357,44,71,23,300,51,73,15,170,45*6A

note: Inside one cycle the first two char in the message Talker ID (randomly) changing from/to GPGSV to GLGSV.

fieldnames:

\$GP(L)GSV, messageTotalNo, messageNo,nsatView,sv1Id,sv1Elevation,sv1Azimuth,sv1Snr, sv2Id,sv2Elevation,sv2Azimuth,sv2Snr
sv3Id,sv3Elevation,sv3Azimuth,sv3Snr sv4Id,sv4Elevation,sv4Azimuth,sv4Snr
*checksum

messageTotalNo: 1-9 Total number of sentences for full information

messageNo: 1-9 Sentence number

nsatView: Total number of satellites in view that will be included in the messages

sv1Id: SV PRN number

sv1Elevation: 00-90 Elevation in degrees

sv1Azimuth: 000-359 Azimuth degrees from the True North

sv1Snr: 00-99 Signal Noise Ratio in decibel (null when not tracking)

sv2Id: SV PRN number

sv2Elevation: 00-90 Elevation in degrees

sv2Azimuth: 000-359 Azimuth degrees from the True North

sv2Snr: 00-99 Signal Noise Ratio in decibel (null when not tracking)

sv3Id: SV PRN number

sv3Elevation: 00-90 Elevation in degrees

sv3Azimuth: 000-359 Azimuth degrees from the True North

sv3Snr: 00-99 Signal Noise Ratio in decibel (null when not tracking)

sv4Id: SV PRN number

sv4Elevation: 00-90 Elevation in degrees

sv4Azimuth: 000-359 Azimuth degrees from the True North

sv4Snr: 00-99 Signal Noise Ratio in decibel (null when not tracking)

Checksum: XOR checksum after *

\$GPGLL – Position data: Position fix, time of position fix and status

Table: fugro_gpgll

example:

\$GPGLL,4034.176963,N,01800.760675,W,133628.23,A,D*61

fieldnames:



\$GPGLL, latitude, latDir, longitude, lonDir, utcTime, gllQual, positioningMode*checksum
 latitude: ddmm.mmmmm 0 to 90 degrees 2|minutes 2|decimal_minutes 5 digits
 latDir: N or S - Latitude Cardinal Direction
 longitude: dddmm.mmmmm 0 to 180 degrees 3|minutes 2|decimal_minutes 5 digits
 lonDir: E or W - Longitude Cardinal Direction
 utcTime: hhmmss.ss UTC time hours 2|minutes 2|seconds 2|decimal_seconds 2 digits
 (max)
 gllQual: A=valid, D=DGPS used, V=dead reckoning or invalid - Geographic Position
 Quality
 positioningMode: s - Mode Indicator (A: Autonomous(GPS) or D: Differential
 (DGPS) or E: Dead reckoning or M: Manual Input Mode or S: Simulated Mode or N: Data
 Not Valid)
 Checksum: XOR checksum after *

\$GNGSA – GPS DOP and active satellites

Table: fugro_gngsa

example:

\$GNGSA,A,3,15,5,10,13,14,23,24,30,,,,,1.3,0.7,1.1*1A

fieldnames:

\$GNGSA, gsaMode,gsaStatus,sId1, sId2, sId3, sId4, sId5, sId6, sId7, sId8, sId9, sId10,
 sId11, sId12, pdop, hdop, vdop, gsId*checksum

gsaMode: M or A – Manual or Automatic selection of 2D or 3D fix

gsaStatus: - solution (1=fix not available; 2=2D; 3=3D)

sId1: - PRN of satellite used for fix.

sId2: - PRN of satellite used for fix.

sId3: - PRN of satellite used for fix.

sId4: - PRN of satellite used for fix.

sId5: - PRN of satellite used for fix.

sId6: - PRN of satellite used for fix.

sId7: - PRN of satellite used for fix.

sId8: - PRN of satellite used for fix.

sId9: - PRN of satellite used for fix.

sId10: - PRN of satellite used for fix.

sId11: - PRN of satellite used for fix.

sId12: - PRN of satellite used for fix.

pdop: - Dilution of position

hdop: Horizontal dilution of position

vdop: Vertical dilution of position

gsId: GNSS System ID

Checksum: XOR checksum after *



Sonardyne Fusion USBL

All USBL systems calculate position by measuring the range and bearing from a vessel-mounted transceiver to an acoustic transponder fitted to a moving target or placed on the seabed. But not all USBL systems do it with the accuracy and precision offered by Ranger 2. It can track your equipment to beyond 7,000 metres and update its position every second. It's engineered for shallow water, deep water, high elevation and multi-user operating scenarios.

RANGER2USBL

NMEA ASCII OUTPUTS: 1

\$GPGGA

Example:

```
18/07/2021 01:40:39.588 $GPGGA,014029.073,5417.52486,N,01201.91167,W,2,00,0.0,-2049.506,M,0.0,M,0.0,0023*49
18/07/2021 01:40:41.498 $GPGGA,014030.986,5417.52512,N,01201.91168,W,2,00,0.0,-2050.557,M,0.0,M,0.0,0023*4D
18/07/2021 01:40:43.417 $GPGGA,014032.895,5417.52529,N,01201.91296,W,2,00,0.0,-2051.696,M,0.0,M,0.0,0023*49
```

\$GPGGA – Global Positioning Fix Data

Table: ranger2usbl_gpgga

example:

```
$GPGGA,161628.493,2630.00058,N,07652.02388,W,2,00,0.0,-
239.793,M,0.0,M,0.0,0003*71
```

fieldnames:

\$GPGGA,utcTime, latitude, latDir, longitude, lonDir, ggaQual, numSat, hdop, altitude, unitsOfMeasureAntenna, geoidAltitude, unitsOfMeasureGeoid, diffcAge, dgnssRefId*checksum

utcTime: hhmmss.sss UTC of position fix - hours 2|minutes 2|seconds

2|decimal_seconds 3 digits

latitude: ddmm.mmmmm 0 to 90 degrees 2|minutes 2|decimal_minutes 5 digits

latDir: N or S – Latitude Cardinal Direction

longitude: dddmm.mmmmm 0 to 180 degrees 3|minutes 2|decimal_minutes 5 digits

lonDir: E or W – Longitude cardinal direction

ggaQual: 0-6 – GNNS quality indicator

numSat: 0-32 – Number of Satellites used in the fix

hdop: Horizontal Dillution of precision

altitude: Altitude above or below mean sea level unit=metres

unitsOfMeasureAntenna: null or M

geoidAltitude: Height of Geoid above WGS84 in meters

unitsOfMeasureGeoid: null or M

diffcAge: 0 to 999 – Age of differential correction in seconds since last RTCM-104

message unit: seconds



dgnssRefId: 0000 to 1023 – DGNSS reference station identity
Checksum: XOR checksum after *

USBLPSON

NMEA ASCII OUTPUTS: 1
\$GPGGA

Example:

```
08/08/2022 15:22:55.996 $PSONLLD,152255.72,3005,A,54.1948921,-11.86006306,24.211,0.68,0.68,0.00,0.13,,,*29
08/08/2022 15:22:57.960 $PSONLLD,152257.69,3005,A,54.1948926,-11.86006298,24.996,0.68,0.68,0.00,0.13,,,*24
```

\$PSONLLD – Ranger 2 PSNLLD message

Table: usblpson_psonlld

example:

```
$PSONLLD,152257.69,3005,A,54.1948926,-11.86006298,24.996,0.68,0.68,0.00,0.13,,,*24
```

fieldnames:

\$PSONLLD,TimeValid,id, status,lat, long, depth, horErrMajor, horrErrMinor, depthError, optionalSpec, opt1, opt2, opt3*checksum

TimeValid: hhmmss.ss UTC Time of Validity

id: ID of tracked object (integer 4 digit)

status: Status string [A or V]

lat: Latitude DD dd.mmmmmmm -90 to 90 decimal degrees

long: Longitude DD dd.mmmmmmm -180 to 180 degrees

depth: Depth in metres with respect to the water surface ffff.ff

horErrMajor: Horizontal error ellipse major axis in metres

horrErrMinor: Horizontal error ellipse minor axis in metres

depthError: Depth error in metres

optionalSpec: Options fields specifier

opt1: Optional field 1

opt2: Optional field 2

opt3: Optional field 3

Checksum: XOR checksum after *

Sperry Marine NAVITVIN IV Ship gyrocompasses

The ship heading management system is a Sperry Marine NAVITVIN IV multi-compass system connected to the 3x NAVIGAT X MK 1 gyros.



SHIPSGYRO

NMEA ASCII OUTPUTS: 5

\$HEHDT

\$TIROT

\$PPLAN

\$GPGGA

\$GPVTG

Example:

```
21/07/2021 05:44:19.713 $HEHDT,074.74,T*1F
21/07/2021 05:44:19.817 $TIROT,-0002.7,A*23
21/07/2021 05:44:19.913 $HEHDT,074.74,T*1F
21/07/2021 05:44:20.113 $GPGGA,,5410.773,N,01150.340,W,1,,,,,*4F
21/07/2021 05:44:20.113 $HEHDT,074.75,T*1E
21/07/2021 05:44:20.216 $TIROT,-0002.1,A*25
21/07/2021 05:44:20.313 $HEHDT,074.77,T*1C
21/07/2021 05:44:20.444 $GPVTG,,,,000.0,N,000.0,K,A*3A
21/07/2021 05:44:20.513 $HEHDT,074.81,T*15
21/07/2021 05:44:20.663 $TIROT,00005.6,A*38
21/07/2021 05:44:20.663 $PPLAN,,,,,1*72
21/07/2021 05:44:20.713 $HEHDT,074.86,T*12
21/07/2021 05:44:20.913 $GPGGA,,5410.773,N,01150.340,W,1,,,,,*4F
21/07/2021 05:44:20.913 $HEHDT,074.88,T*1C
21/07/2021 05:44:21.016 $TIROT,00006.2,A*3F
21/07/2021 05:44:21.112 $HEHDT,074.89,T*1D
21/07/2021 05:44:21.244 $GPVTG,,,,000.0,N,000.0,K,A*3A
```

\$HEHDT – Heading – True Data

Table: shipsgyro_hehdt

example:

\$HEHDT,134.79,T*17

fieldnames:

\$HEHDT, headingTrue, trueHeading*checksum

headingTrue: 0 to 359.99 – True Vessel Heading in the vessel frame degrees

3|decimal_degrees 1 digit

trueHeading: T=True

Checksum: XOR checksum after *

\$TIROT – Rate of Turn

Table: shipsgyro_tirotd

example:

\$TIROT,-0021.4,A*21



fieldnames:

\$TIROT, rot, rot_status*checksum

rot: Rate of Turn, degrees per minute negative means bow to port

rot_status: A=data is valid V=invalid

Checksum: XOR checksum after *

\$PPLAN – Nav Status Data

Table: shipsgyro_pplan

example:

\$PPLAN,,,,,,1*72

fieldnames:

\$PPLAN, pplan1, pplan2, pplan3, pplan4, pplan5, pplan6, pplan7, pplan8*checksum

Pplan1-8: nav status messages.

Checksum: XOR checksum after *

\$GPGGA – Global Positioning Fix Data (limited to lat/lon and quality)

Table: shipsgyro_gpgga

example:

\$GPGGA,,2133.298,S,00925.808,E,1,,,,,*4F

fieldnames:

\$GNGGA, utcTime, latitude, latDir, longitude, lonDir, ggaQual, numSat, hdop, altitude, unitsOfMeasureAntenna, geoidAltitude, unitsOfMeasureGeoid, diffcAge, dgnssRefId*checksum

utcTime: hhmmss.ss UTC of position fix - hours 2|minutes 2|seconds 2|decimal_seconds 3 digits

latitude: 0 to 90 degrees 2|minutes 2|decimal_minutes 5 digits

latDir: N or S Direction of latitude

longitude: 0 to 180 degrees 3|minutes 2|decimal_minutes 5 digits

lonDir: E or W

ggaQual: 0-6 – GPS quality indicator

numSat: 0-32 – Number of Satellites used in the fix

hdop: Horizontal Dillution of precision

altitude: Altitude above or below mean sea level unit=metres

unitsOfMeasureAntenna null or M

geoidAltitude: Height of Geoid above WGS84 in meters

unitsOfMeasureGeoid: null or M

diffcAge: 0 to 999 – Age of differential correction in seconds since last RTCM-104



message unit: seconds

dgnssRefId: 0000 to 1023 – DGNSS reference station identity

Checksum: XOR checksum after *

\$GPVTG – Course Over Ground and Ground Speed Data (limited to speed)

Table: shipsgyro_gpvtg

example:

\$GPVTG,,,,,-00.1,N,-00.2,K,A*39

fieldnames:

\$GNVTG,courseTrue, trueCourse, magneticTrack, mFlag, speedKnots, nFlag,
 speedKmph, kFlag, positioningMode*checksum

courseTrue: 000.0 to 359.9 – True Vessel track in the vessel frame degrees
 3|decimal_degrees 1 digit

trueCourse: T -True

magneticTrack: 0-359.9 Magnetic track unit: degrees

mFlag: M

speedKnots: 0-1000 Speed in the vessel frame unit: knots

nFlag: N

speedKmph: 0-1852 Speed in the vessel frame unit: km/h

kFlag: K Speed over ground unit K=km/h

positioningMode: A=GPS used, D=DGPS used, E=dead reckoning, N=invalid,
 S=simulator, P=precise (P NMEA4.1 only)

checksum: XOR checksum after *

SKIPPER DL850 Skipper log (ship's velocity)

The SKIPPER DL 850 is a dual axis speed log, working on the Doppler principle, providing longitudinal and fore/aft transversal ship speed relative to the sea bed or the water.

SKIPPERLOG

The \$IIDPT message source is the navigational echosounder

NMEA ASCII OUTPUTS: 4

\$VDVBW

\$VDVHW

\$IIDPT

\$VDMTW

\$VDVTG



Example:

```
21/07/2021 08:00:25.740 $VDVBW,0.02,-0.10,A,,V,,V*68
21/07/2021 08:00:25.740 $VDVHW,,,,,0.10,N,0.18,K*56
21/07/2021 08:00:25.740 $VDVTG,,,,,N,,K,N*30
21/07/2021 08:00:25.740 $IIDPT,,,*6C
21/07/2021 08:00:25.790 $VDMTW,-8.4,C*10
21/07/2021 08:00:26.716 $VDVBW,0.02,-0.08,A,,V,,V*61
21/07/2021 08:00:26.716 $VDVHW,,,,,0.08,N,0.14,K*53
21/07/2021 08:00:26.716 $VDVTG,,,,,N,,K,N*30
21/07/2021 08:00:26.716 $IIDPT,,,*6C
21/07/2021 08:00:26.767 $VDMTW,-8.4,C*10
```

\$VDVBW – Dual ground/water speed

Table: skipperlog_vdvbw

example:

```
$VDVBW,0.14,-0.54,A,0.12,-0.49,A,,V,,V*5B
```

fieldnames:

\$VMVBW, longitudinalWaterSpeed, TransverseWaterSpeed, status1,
longitudinalGroundSpeed, transverseGroundspeed, status2, vbw7, vbw8, vbw9,
vbw10*checksum

longitudinalWaterSpeed: dd.d - Longitudal WaterSpeed in Knots positive forward
TransverseWaterSpeed: dd.d - Transverse WaterSpeed in Knots positive port
Status1: A=Data valid|V=Data invalid
longitudinalGroundspeed: dd.d - Longitudal GroundSpeed in Knots positive forward
TransverseGroundSpeed: dd.d - Transverse GroundSpeed positive port
Status2: A=Data valid|V=Data invalid
vbw7: unknown
vbw8: unknown
vbw9: unknown
vbw10: unknown
checksum: XOR checksum after *

\$VDVHW – Water speed & heading

Table: skipperlog_vdvhw

example:

```
$VDVHW,,,,,0.52,N,0.97,K*57
```

fieldnames:

\$VDVHW, headingTrue, headingTrueFlag, headingMagnetic, headingMagneticFlag,



speedKnots, nFlag, speedKmph, kFlag

headingTrue: degrees true

headingTrueFlag: T True flag

headingMagnetic: degrees magnetic

headingMagneticFlag: M magnetic flag

speedKnots: Speed relative to water unit: knots

nFlag: N=knots

speedKmph: Speed relative to water unit: km/h

kFlag: K=km/h

\$IIDPT – Depth of water

Table: skipperlog_iidpt

example:

\$IIDPT,113.6,0.00,100*68

fieldnames:

\$SDDPT, waterDepthMeter, offsetT, maxRange*checksum

waterDepthMeter: Water depth relative to transducer unit=meters

offsetT: Offset of transducer from waterline in meters

maxRange: Maximum range Scale in Use

Checksum: XOR checksum after *

\$VDMTW – Water Temperature (not valid)

Table: skipperlog_vdmtw

example:

\$VDMTW,23.4,C*04

fieldnames:

\$VDMTW, waterTemperatureCelsius, celsiusFlag*checksum

waterTemperatureCelsius: Water Temperature unit=Celsius

cFlag C Celsius flag

Checksum: XOR checksum after *

\$GPVTG – Course Over Ground and Ground Speed Data (limited to speed)



Table: shipsgyro_gpvtg

example:

\$GPVTG,,,,,-00.1,N,-00.2,K,A*39

fieldnames:

\$GNVTG, courseOverGround, trueCourse, magneticTrack, mFlag, speedKnots, nFlag, speedKmph, kFlag, positioningMode*checksum

courseOverGround: 000.0 to 359.9 – True Vessel track in the vessel frame degrees
3|decimal_degrees 1 digit

trueCourse: T -True

magneticTrack: 0-359.9 Magnetic track unit: degrees

mFlag: M

speedKnots: 0-1000 Speed in the vessel frame unit: knots

nFlag: N

speedKmph: 0-1852 Speed in the vessel frame unit: km/h

kFlag: K Speed over ground unit K=km/h

positioningMode: A=GPS used, D=DGPS used, E=dead reckoning, N=invalid,
S=simulator, P=precise (P NMEA4.1 only)

checksum: XOR checksum after *

Kongsberg Maritime Simrad EM122 Multibeam echo sounder (deep)

EM122

This echosounder is rated to 11,000m, but probably up to 8,000m for good quality data.

The EM122 it is viewed and operated via SIS (Seafloor Information Service). A quickstart guide to common tasks is included in these notes, but the manuals for this software are extensive.

NMEA ASCII OUTPUTS: 1

\$KIDPT

Example:

20/07/2021 08:20:34.936	\$KIDPT,1388.34,6.01,12000.0*71
20/07/2021 08:20:41.534	\$KIDPT,1388.18,5.92,12000.0*76
20/07/2021 08:20:47.734	\$KIDPT,1388.59,5.83,12000.0*73
20/07/2021 08:20:54.334	\$KIDPT,1388.57,5.95,12000.0*7a
20/07/2021 08:21:00.162	\$KIDPT,1388.90,5.87,12000.0*72
20/07/2021 08:21:06.934	\$KIDPT,1388.39,6.07,12000.0*7a

\$KIDPT – Depth of water

Table: em122_kidpt



example:

\$KIDPT,1548.34,5.79,12000.0*77

fieldnames:

\$KIDPT, waterDepthMeter, transduceroffset, maxRange *checksum

waterDepthMeter: d.d Depth in meters from the transducer Centre Beam unit=meters

transduceroffset: Offset of transducer from waterline in meters unit=meters

maxRange: Maximum range scale in use (Always 12000.0)

Checksum: XOR checksum after *

Kongsberg Maritime Simrad EA640 Single beam echo sounder 10 & 12KHz

The EA640 is a special version of the EA600 commissioned for the *RRS Discovery*, pretty much identical to the EA600 and can operate at either 12kHz or 10kHz as required.

EA640

NMEA ASCII OUTPUTS: 2

\$SDDPT

\$SDDBS

Example:

21/07/2021 09:28:53.773	\$SDDPT,1428.25,6.00*69
21/07/2021 09:28:53.773	\$SDDBS,4705.53,f,1434.24,M,784.25,F*39
21/07/2021 09:28:58.774	\$SDDPT,1425.62,6.17*61
21/07/2021 09:28:58.774	\$SDDBS,4697.48,f,1431.79,M,782.91,F*3D
21/07/2021 09:29:03.741	\$SDDPT,1426.95,6.08*64
21/07/2021 09:29:03.741	\$SDDBS,4701.55,f,1433.03,M,783.59,F*35
21/07/2021 09:29:08.848	\$SDDPT,-6.06,6.06*7A
21/07/2021 09:29:08.848	\$SDDBS,0.00,f,0.00,M,0.00,F*31
21/07/2021 09:29:14.579	\$SDDPT,1428.80,6.16*61
21/07/2021 09:29:14.579	\$SDDBS,4707.90,f,1434.97,M,784.65,F*38

\$SDDPT – Depth of water

Table: ea640_sddpt

example:

\$SDDPT,3992.52,6.42*61

fieldnames:

\$SDDPT, waterDepthMeterFromTransducer, transduceroffset*checksum

waterDepthMeterFromTransducer: Depth in meters from the transducer unit=meters

transduceroffset: z-axis distance from the platform coordinate system origin to the

sonar transducer in meter unit: meters

Checksum: XOR checksum after *



\$SDDBS – Depth below surface

Table: ea640_sddbd

example:

\$SDDBS,13116.42,f,3997.88,M,2186.07,F*3D

fieldnames:

\$SDDBS,waterDepthFeetFromSurface, feetFlag, waterDepthMeterFromSurface, meterFlag, waterDepthFathomFromSurface, fathomFlag*checksum

waterDepthFeetFromSurface: Water depth relative to surface (waterlevel)

unit=feet

feetFlag: f feet

waterDepthMeterFromSurface: Water depth relative to surface (waterlevel)

unit=meter

meterFlag: M feet

waterDepthFathomFromSurface: Water depth relative to surface (waterlevel)

unit=fathom

fathomFlag: F fathom

Checksum: XOR checksum after *

Cable Logging And Monitoring system

The CLAM system is the Cable Logging And Monitoring system. The system displays and logs the cable usage and outputs data to Techsas.

WINCH

NMEA ASCII OUTPUTS: 1

\$WINCH

Example:

20/07/2021 05:15:07.265	\$WINCH,21 201 05:15:04,1,0.93,830.58,-0.60,0.94,0.00,
20/07/2021 05:15:08.264	\$WINCH,21 201 05:15:05,1,0.93,829.91,-0.60,0.94,0.00,
20/07/2021 05:15:09.265	\$WINCH,21 201 05:15:06,1,0.94,829.29,-0.61,0.94,0.00,
20/07/2021 05:15:10.265	\$WINCH,21 201 05:15:07,1,0.94,828.69,-0.60,0.94,0.00,
20/07/2021 05:15:11.265	\$WINCH,21 201 05:15:08,1,0.94,828.08,-0.60,0.95,0.00,
20/07/2021 05:15:12.265	\$WINCH,21 201 05:15:09,1,0.92,827.48,-0.60,0.95,0.00,
20/07/2021 05:15:13.265	\$WINCH,21 201 05:15:10,1,0.91,826.81,-0.57,0.95,0.00,
20/07/2021 05:15:14.265	\$WINCH,21 201 05:15:11,1,0.84,826.38,-0.29,0.94,0.00,
20/07/2021 05:15:15.265	\$WINCH,21 201 05:15:12,1,0.89,826.04,-0.34,0.95,0.00,
20/07/2021 05:15:16.265	\$WINCH,21 201 05:15:13,1,0.94,825.71,-0.34,0.95,0.00,
20/07/2021 05:15:17.265	\$WINCH,21 201 05:15:14,1,0.92,825.37,-0.34,0.95,0.00,

\$WINCH – Cable logging system data

Table: winch_winch

example:



\$WINCH,18 148 16:09:20,1,0.48,161.07,1.03,1.13,0.00,

fieldnames:

\$WINCH,winchDatum,cableType,tension,cableOut,rate,btension,angle

winchDatum: YY jjj hh:mm:ss (year, Julian day time) - DateTime from CLAM

cableType: selected winch cable (0: nothing; 1: CTD1; 2: CTD2; 3: Coring; 4: Deep Tow; 5: Trawl; 6: General Purpose

tension: d.d Tension - Load on the Cable (outboard) unit: tonne

cableOut: d.d - Length of Deployed Cable unit: meter

rate: d.d speed of the Cable unit: m/s

backTension: back tension on cable unit: tonne

rollAngle: roll angle (not used)

undefined: d.d undefined extra comma

NMF SURFMET Meteorology and Surface Hydrography Suite

SurfMet comprises two sets of scientific instruments: Meteorological; and Surface Water Sampling, along with ADCs and a PC hosting SurfMet data conversion software that passes data to the Data Systems for event logging.

The Meteorological part of the system comprises a range of instruments located near the forward mast on the Met-Platform about 10 metres above sea level.

The instrument called the...	...measures...	...in...	...to calculate...
<i>Vaisala HMP155</i> Temperature & Humidity Sensor	Thermal radiation and water vapour	Sunlight; Air	Ambient air temperature and Relative humidity
<i>Gill Windsonic</i> Anemometer	Ultrasonic sound waves via ultrasound transceivers	Air	Wind speed and direction
<i>Vaisala BaroCap PTB210</i> Barometric Pressure sensor	Change in electrical resistance via a deflectable diaphragm strain gauge within a pressure transducer	Air	Air pressure
<i>Kipp & Zonen CM6B</i> Pyranometer	Electromagnetic radiation flux density by converting solar radiation into heat, and thence into a voltage	Sunlight	Total Irradiance (Solar energy)
<i>Skye Instruments SKE510</i> PAR (Photosynthetic Active Radiation) Pyranometer	Electromagnetic radiation flux density by converting solar radiation into heat, and thence into a voltage, passed through a bandpass filter	Sunlight	Total Irradiance (Solar energy) within a fixed range of wavelengths for photosynthesis
<i>SeaBird 38</i> Digital Oceanographic Thermometer	Change in resistance via a thermistor	Seawater	Particulate density

SURFMET

NMEA ASCII OUTPUTS: 1

\$GPXSM



Example:

```
21/07/2021 10:40:18.222 $GPXSM,1.710411,-0.010879,-0.011094,0.9012,1.1742,4.869,28.0512,16.5,93.2,1018.4884,,,,*4a
21/07/2021 10:40:19.242 $GPXSM,1.746144,-0.01075,-0.010879,0.9024,1.174,5.008,28.0512,16.5,93.19,1018.4884,,,,*71
21/07/2021 10:40:20.277 $GPXSM,1.728428,-0.010965,-
0.011008,0.9,1.1746,5.051,26.1144,16.49,93.19,1018.4936,327.6,328.5,787.4,819.1*74
21/07/2021 10:40:21.293 $GPXSM,1.759861,-0.01075,-0.010965,0.8997,1.1749,,,,,327.6,326.2,787.4,819.4*72
21/07/2021 10:40:22.309 $GPXSM,1.732298,-0.010836,-0.010965,0.8995,1.1694,4.779,25.1064,16.49,93.17,1018.5404,,,,*42
21/07/2021 10:40:23.367 $GPXSM,1.72344,-0.011094,-0.011094,0.8998,1.1694,4.778,28.0512,16.5,93.17,1018.5352,,,,*4e
21/07/2021 10:40:24.710 $GPXSM,1.719527,-0.01075,-
0.011008,0.9009,1.1766,4.839,28.0512,16.5,93.17,1018.504,324.1,325.6,789.5,821.2*76
21/07/2021 10:40:25.743 $GPXSM,1.728428,-0.011137,-0.011266,0.9007,1.1772,4.84,28.0512,16.5,93.14,1018.504,,,,*7c
21/07/2021 10:40:26.790 $GPXSM,1.737501,-0.010965,-0.010879,0.899,1.1775,5.023,28.0512,16.5,93.17,1018.5248,,,,*47
21/07/2021 10:40:27.812 $GPXSM,1.736856,-0.010836,-0.011137,0.8998,1.1765,,,,,,*6e
21/07/2021 10:40:28.934 $GPXSM,1.746402,-0.010836,-0.011008,0.8978,1.1743,5.112,26.0784,16.5,93.19,1018.5404,,,,*78
21/07/2021 10:40:29.943 $GPXSM,1.737759,-0.010965,-
0.010965,0.8986,1.1771,4.825,27.0576,16.5,93.15,1018.556,322.4,328.1,789.6,821.8*46
21/07/2021 10:40:31.167 $GPXSM,1.728643,-0.010836,-0.011094,0.8994,1.1777,4.733,27.0576,16.5,93.17,1018.5248,,,,*7d
21/07/2021 10:40:32.198 $GPXSM,1.719312,-0.010965,-
0.011094,0.8994,1.1771,4.733,27.0576,16.5,93.15,1018.4936,323.5,325.6,788.7,820.5*74
21/07/2021 10:40:33.269 $GPXSM,1.746359,-0.010707,-0.010879,0.8989,1.1769,,,,,327.6,326.4,788.4,820.6*4e
21/07/2021 10:40:34.325 $GPXSM,1.727138,-0.010879,-
0.011223,0.8992,1.1781,4.99,25.1064,16.49,93.12,1018.478,322.3,327.3,788.6,820.6*44
21/07/2021 10:40:35.341 $GPXSM,1.720903,-0.010836,-
0.010836,0.9001,1.1774,5.067,25.1064,16.5,93.14,1018.5404,327.0,325.6,788.9,820.9*7e
21/07/2021 10:40:36.385 $GPXSM,,,,,4.762,27.0576,16.5,93.12,1018.5404,327.1,325.6,789.3,821.1*61
21/07/2021 10:40:37.401 $GPXSM,1.719441,-0.011137,-
0.011438,0.8995,1.1783,4.764,30.0168,16.5,93.14,1018.5404,320.4,325.6,789.5,821.4*77
```

\$GPXSM – Surfmet output message

Table: surfmet_gpxsm

example:

```
$GPXSM,26.5291,26.2810,5.6775,0.0659,4.5805,7.5960,358.2072,25.1600,81.77
00,1014.3908,281.1000,277.7000,524.1000,583.0000*77
$GPXSM,25.0788,24.7506,0.0015,0.1007,3.7695,5.1220,217.6056,21.9200,50.22
00,1014.6716,398.6000,392.6000,973.4000,789.9000*73
```

fieldnames:

\$GPXSM,flow1,watertemperature,flow3,fluo,trans,windSpeed,windDirection,airTemperature,
humidity,airPressure,parPort,parStarboard,tirPort,tirStarboard*checksum

flow1: flowmeter1 – outflow from the SBE45 TSG unit: l/m

watertemperature: SBE38 drop-keel sensor unit:degrees Celsius

flow3: flowmeter3 unit: l/m (not installed)

fluo: fluorometer instrument output unit: Volts

trans: transmissometer instrument data output unit: Volt

windSpeed: 0 to 50 Relative Wind Velocity unit: m/s

windDirection: 0 to 360 Relative Wind Direction unit: degrees

airTemperature: -40 to 60 Air Temperature unit: degrees Celsius

humidity: 0 to 100 Relative Humidity of the Air unit: percentage

airPressure: The atmospheric pressure unit: hPa



parPort: Photosynthetically Active Radiation (PAR) from portside
 parStarboard: Photosynthetically Active Radiation (PAR) from starboard side
 tirPort: Total Irradiance (TIR) from portside
 tirStarboard: Total Irradiance (TIR) from starboard side
 Checksum: XOR checksum after *

SBE45

The Surface Water part of the system collects seawater (known as “non-toxic water”) from the upper 5.3 metres of the ocean, and passes it through the following instruments:

The instrument called the...	...measures...	...in...	...to calculate...
SeaBird 45 Thermosalinograph	Temperature and conductivity	Seawater	Salinity
SeaBird 38 Digital Oceanographic Thermometer	Change in resistance via a thermistor	Seawater	Temperature
WetLabs WetStar WS3S Fluorometer	Reflected light frequency difference between beams of light passed through water	Seawater	Marine floral density
WetLabs WetStar CST Transmissometer	Photon quanta (received light)	Seawater	Particulate density

TSG flow is approx 1.6 litres per minute whilst fluorometer and transmissometer flow is approx 3 l/min. Flow to instruments is degassed using a debubbler with 10 l/min inflow and 10/l min waste flow.

You can find the fluourometer and transmissometer data in the [nfm_surfmet_gpxsm](#) table.

NOT NMEA ASCII OUTPUTS: 1

t1= tt.tttt c1= c.ccccc s= ss.ssss sv= vvvv.vvv t2= tt.tttt

t1= – Thermosalinograph data (NOT NMEA LIKE)

Example:

21/07/2021 09:31:24.435	t1= 15.4864, c1= 4.36386, s= 35.2185, sv=1508.499, t2= 15.5005
21/07/2021 09:31:25.478	t1= 15.4890, c1= 4.36413, s= 35.2185, sv=1508.472, t2= 15.4919
21/07/2021 09:31:26.521	t1= 15.4916, c1= 4.36437, s= 35.2184, sv=1508.451, t2= 15.4854
21/07/2021 09:31:27.566	t1= 15.4939, c1= 4.36467, s= 35.2191, sv=1508.385, t2= 15.4640
21/07/2021 09:31:28.607	t1= 15.4968, c1= 4.36498, s= 35.2192, sv=1508.294, t2= 15.4347
21/07/2021 09:31:29.648	t1= 15.4996, c1= 4.36531, s= 35.2196, sv=1508.268, t2= 15.4261
21/07/2021 09:31:30.690	t1= 15.5023, c1= 4.36548, s= 35.2188, sv=1508.250, t2= 15.4207

Table: sbe45_nanan

example:



t1= 20.9661, c1= 4.96654, s= 35.6140, sv=1524.241, t2= 20.7603

fieldnames:

housingWaterTemperature, conductivity, salinity, soundVelocity,
remoteWaterTemperature

housingWaterTemperature: t1= tt.tttt SBE45 housing temperature unit: degrees Celsius (ITS-90)

conductivity: c1= c.ccccc SBE45 conductivity unit: S/m

salinity: s=ss.ssss calculated salinity unit: PSU

soundVelocity: sv=vvvv.vvv SBE45 sea surface sound velocity (chen-millero) m/s

remoteWaterTemperature: t2= tt.tttt SBE38 remote water temperature unit: degrees Celsius (ITS-90)

Lambrecht THpro Humidity and Temperature Probe

ENVTEMP

NMEA ASCII OUTPUTS: 2

\$WIMTA

\$WIMHU

Example:

21/07/2021 10:17:31.175	\$WIMTA,005.5,C*2B
21/07/2021 10:17:31.175	\$WIMHU,058.2,-02.0,C*1D
21/07/2021 10:17:32.170	\$WIMTA,005.5,C*2B
21/07/2021 10:17:32.170	\$WIMHU,058.2,-02.0,C*1D
21/07/2021 10:17:33.175	\$WIMTA,005.5,C*2B
21/07/2021 10:17:33.175	\$WIMHU,058.2,-02.0,C*1D

\$WIMTA – Temperature data

Table: envtemp_wimta

example:

\$WIMTA,005.6,C*28

fieldnames:

\$WIMTA, airTemperature, celsiusFlag*checksum

airTemperature: -40 to 60 Air Temperature unit: degrees Celsius

celsiusFlag: C Celsius Flag

Checksum: XOR checksum after *

\$WIMHU – Environment Humidity



Table: envtemp_wimhu

example:

\$WIMHU,051.0,-03.7,C*10

fieldnames:

\$WIMTA, humidity, flag, temperatureDewPoint*checksum

humidity: Relative humidity of the atmosphere

flag: validity Flag

temperatureDewPoint: Dew point temperature of the atmosphere

Checksum: XOR checksum after *

OceanWaves WaMoS II Wave Radar

WaMoS is an X-Band nautical RADAR with a range of 100m to 4km. It can only generate data in above a minimum wind speed of 3ms^{-1} . It detects open wave spectra. Sea state is calculated from detected backscatter of μ wave "sea clutter" in real time. The system can detect wavelengths from 15 m – 600 m and covers periods from 4 sec-20 seconds. At coastal sites, WaMoS II can only measure the spatial wave field beyond the wave breaking zone. There is a WaMoS computer in the Met Lab, where it stores processed radar images. Data is logged in WaMoS's own format and with the TechSAS and RVDAS logger. Summary wave information is available in one of the ASCII files generated.

WAMOS

\$PWAM - WAMOS WAVERADAR Output Data

NMEA ASCII OUTPUT: 1

\$PWAM

Example:

```
21/07/2021 08:07:49.531    $PWAM,000.79,006.58,344.00,008.19,105.00,-09.00,-09.00,-
09.00,062.00,005.38,045.00,220.00,000.24,2021-07-21 08:05:00,001.31,000274
21/07/2021 08:10:48.985    $PWAM,000.80,006.56,344.00,008.18,104.00,-09.00,-
09.00,062.00,005.38,045.00,219.00,000.26,2021-07-21 08:08:00,001.31,000234
21/07/2021 08:13:50.920
    $PWAM,000.78,006.56,343.00,008.02,100.00,170.00,011.43,204.00,062.00,005.38,045.00,222.00,000.27,2021-07-21
08:11:00,001.19,000274
21/07/2021 08:16:50.381
    $PWAM,000.80,006.55,349.00,008.01,100.00,162.00,010.76,181.00,054.00,005.54,048.00,220.00,000.28,2021-07-21
08:14:00,001.30,000274
21/07/2021 08:19:50.302
    $PWAM,000.81,006.55,351.00,008.01,100.00,162.00,010.76,181.00,066.00,005.54,048.00,222.00,000.28,2021-07-21
08:17:00,001.30,000274
```

Table: wamos_pwam

example:

\$PWAM,000.99,006.63,259.00,009.16,131.00,-09.00,-09.00,-



09.00,266.00,009.14,131.00,110.00,000.20,2021-07-11 00:20:00,001.67,000010

fieldnames:

\$PWAM, hs, tm2, pdir, tp, lp, dp1, tp1, lp1, dp2, tp2, lp2, currentdir, currentspeed,

hs: Significant wave height of waves (Hs) on the water body

tm2: Period of waves (swell) on the water body

pdir: Direction (from) of waves on the water body

tp: Period at spectral maximum of waves (peak period Tp) on the water body

lp: Peak Wave length of waves (lp) on the water body

dp1: Direction (from) of waves on the water body

tp1: Period at spectral maximum of waves (swell period Tp1) on the water body

lp1: Swell Wave length of waves (lp1) on the water body

dp2: Direction (from) of waves on the water body

tp2: Period at spectral maximum of waves (swell period Tp2) on the water body

lp2: Swell Wave length of waves (lp2) on the water body

currentdir: current direction unit: degrees

currentspeed: Current speed unit: m/s

No Seismic Sensors Gravimeters and Magnetometer and Rex Waveradar documentations yet!

Appendix:

Current appconfig.json used by the RAM module:

```
{
  "localDataDir": "/home/rvdas/ramdata/DY132",
  "createNewFile": "day",
  "shipId": "DY",
  "fsBufferSize": 32,
  "watchkeeperTick": 10,
  "channels": [
    {
      "port": 24008,
      "bcPort": 34508,
      "socketType": "udp4",
      "sensorType": "GYROCOMPASS",
      "id": "SHIPSGYRO",
      "timeout": 10,
      "eol": "\n"
    }
  ],
  {
```



```
"port": 24013,  
"bcPort": 34513,  
"socketType": "udp4",  
"sensorType": "USBL_BEACON",  
"id": "RANGER2USBL",  
"timeout": 120,  
"eol": "\n"  
},  
{  
  "port": 24001,  
  "bcPort": 34501,  
  "socketType": "udp4",  
  "sensorType": "GRAVITY",  
  "id": "AIRSEA2GRAVITY",  
  "timeout": 10,  
  "eol": "\n"  
},  
{  
  "port": 24016,  
  "bcPort": 34516,  
  "socketType": "udp4",  
  "sensorType": "TSS",  
  "id": "PHINS",  
  "timeout": 10,  
  "eol": "\n"  
},  
{  
  "port": 24017,  
  "bcPort": 34517,  
  "socketType": "udp4",  
  "sensorType": "GPS",  
  "id": "FUGRO",  
  "timeout": 10,  
  "eol": "\n"  
},  
{  
  "port": 24006,  
  "bcPort": 34506,  
  "socketType": "udp4",  
  "sensorType": "ECHOSOUNDER",  
  "id": "EM640",  
  "timeout": 180,  
  "eol": "\n"  
},  
{  
  "port": 24010,  
  "bcPort": 34510,  
  "socketType": "udp4",  
  "sensorType": "ECHOSOUNDER",  
  "id": "EM122",  
  "timeout": 180,  
  "eol": "\n"  
},  
{  
  "port": 24000,  
  "bcPort": 34500,  
  "socketType": "udp4",  
  "sensorType": "GPS",  
  "id": "POSMV",  
  "timeout": 10,  
  "eol": "\n"  
},  
}
```



```
{
  "port": 8001,
  "bcPort": 39501,
  "socketType": "udp4",
  "sensorType": "GPS",
  "id": "SEAPATHPOS",
  "timeout": 10,
  "eol": "\n"
},
{
  "port": 24015,
  "bcPort": 34515,
  "socketType": "udp4",
  "sensorType": "TSS",
  "id": "SEAPATHATT",
  "timeout": 10,
  "eol": "\n"
},
{
  "port": 24009,
  "bcPort": 34509,
  "socketType": "udp4",
  "sensorType": "WINCH",
  "id": "WINCH",
  "timeout": 10,
  "eol": "\n"
},
{
  "port": 24002,
  "bcPort": 34502,
  "socketType": "udp4",
  "sensorType": "SURFMET",
  "id": "SURFMET",
  "timeout": 10,
  "eol": "\n"
},
{
  "port": 24003,
  "bcPort": 34503,
  "socketType": "udp4",
  "sensorType": "TSG",
  "id": "SBE45",
  "timeout": 10,
  "eol": "\n"
},
{
  "port": 24007,
  "bcPort": 34507,
  "socketType": "udp4",
  "sensorType": "EMLOG",
  "id": "SKIPPERLOG",
  "timeout": 10,
  "eol": "\n"
},
{
  "port": 24014,
  "bcPort": 34514,
  "socketType": "udp4",
  "sensorType": "GPS",
  "id": "CNAV",
  "timeout": 10,
  "eol": "\n"
}
```



```
{
  {
    "port": 24020,
    "id": "ENVTEMP",
    "sensorType": "MET",
    "socketType": "udp4",
    "eol": "\n",
    "bcPort": 34520,
    "timeout": 10
  },
  {
    "port": 24012,
    "id": "RANGER2USBL2",
    "sensorType": "USBL_BEACON",
    "socketType": "udp4",
    "eol": "\n",
    "bcPort": 34512,
    "timeout": 120
  },
  {
    "port": 24019,
    "id": "SEASPY",
    "sensorType": "MAG",
    "socketType": "udp4",
    "eol": "\n",
    "bcPort": 34519,
    "timeout": 10
  },
  {
    "port": 24018,
    "id": "WAMOS",
    "sensorType": "WAVE_RADAR",
    "socketType": "udp4",
    "eol": "\n",
    "bcPort": 34518,
    "timeout": 300
  }
},
"broadcast": {
  "bufferIntervalS": 1,
  "resolution": 0.05,
  "compress": false,
  "multistreamEnabled": true,
  "aggregateBCPort": 30000,
  "targets": [
    "127.0.0.1"
  ]
},
"applicationInfo": {
  "applicationName": "NMF RVDAS",
  "build": 2.6,
  "publishDate": "01/2019",
  "framework": "NMFWAFR_2017",
  "author": "Ship Scientific Systems",
  "features": "MySQL Connector: No. \nAggregate Broadcaster: Yes.\nMultistream Broadcaster: Yes.\nCustom Channel Timeout: Yes.\nNTP Auto-Update: Yes.",
  "buildNotes": "Added ability to set timeouts per channel. Added ability to broadcast individual channels to multiple targets. Enabled customisation of watchkeeper tick and filestream buffer size."
}
}
```

Ingestor configuration file example:



```
[r2i]
postgres_host = 192.168.62.12
postgres_port = 5432
postgres_user = rvdas
postgres_password = *****
postgres_dbname = DY132
sensorfiles_folder = /home/rvdas/ingester/sensorfiles/dy
rvdas_appconfig_folder = /home/rvdas/ram/apps/rvdas/server
savefile_folder = /home/rvdas/ingesterdata
```

Netcdf Converter JSON configuration file example:

You can create your own JSON configuration file, following the example, modifying the highlighted values from the key – value pairs, do not remove or add new keys!

You can change anything in the table section but make sure there is the the right tableName and the field array values.

The schedule in minutes (max value is 1440 eg. 24h)

The prefix become the name of your created clean table and part of the generated filenames.

```
{
  "description": "NetCDF_Converter INI file DY132 daily",
  "version": 1.0,
  "createdBy": "zoltan.nemeth@noc.ac.uk",
  "creationDate": "2021-06-13",
  "shipID": "DY",
  "timeUnit": "seconds since 1970-01-01 00:00:00Z",
  "timeCalendar": "standard",
  "zscoreSigma": 3,
  "ingesterDatabase": {
    "host": "192.168.62.12",
    "port": 5432,
    "user": "rvdas",
    "password": "*****",
    "dbname": "DY132"
  },
  "metaDataDatabase": {
    "host": "192.168.62.12",
    "port": 5432,
    "user": "sciguest",
    "password": "sciguest",
    "dbname": "meta001"
  },
  "startDateTime": "2021-06-12 00:00:00",
  "endDateTime": "2021-08-08 00:00:00",
  "schedule": 1440,
  "netcdf": {
    "prefix": "D24H",
    "extension": "nc",
  }
}
```



```
"interpolateFrequency": "1S",
"outputLocationCSV": "/home/rvs/ncc_data/DY132/24h/ascii",
"outputLocationClean": "/home/rvs/ncc_data/DY132/24h/clean",
"outputLocationRaw": "/home/rvs/ncc_data/DY132/24h/raw",
"xy_tableName": "posmv_gpgga",
"xy_field": ["time", "latitude", "latdir", "longitude", "londir"],
"z_constant": true,
"z_tableName": "constantValue",
"z_field": 0,
"z_fieldUnit": "m",
"globalAttributes": [{
  "Deployment_id": "/deployments/DY132",
  "title": "Underway observations from RRS Discovery navigation, meteorology and sea surface hydrography sensor arrays",
  "Conventions": "CF-1.8 SeaDataNet-1.0 ACDD-1.3 IOcean-1.0",
  "source": "Underway observations from RRS Discovery navigation, meteorology and sea surface hydrography sensor arrays",
  "history": "",
  "references": "www.noc.ac.uk;https://github.com/I-Ocean/netcdf-specification",
  "date_created": "",
  "date_update": "",
  "processing_level": "RAW or CLEAN/Interpolated instrument data",
  "standard_name_vocabulary": "http://vocab.nerc.ac.uk/collection/P07/current/",
  "sdn_vocabulary": "http://vocab.nerc.ac.uk",
  "featureType": "trajectory",
  "data_interval": "1Hz",
  "owner": "TBC",
  "owner_id": "http://vocab.nerc.ac.uk/TBC",
  "campaign_name": "BLT",
  "comment": "\n2021-06-12T10:44:00Z: beta version last 24 hour",
  "platform_name": "RRS Discovery",
  "platform_model_name": "Discovery Research Vessel",
  "platform_model_id": "TBC",
  "platform_ices_id": "http://vocab.nerc.ac.uk/collection/C17/current/74EQ",
  "platform_imo_id": "9588029",
  "platform_callsign": "2FGX5"
}],
"table": [
  {
    "tableName": "cnav_gndtm",
    "field": ["time", "latoffset", "lonoffset", "altitudeoffset"]
  },
  {
    "tableName": "cnav_gngga",
    "field": ["time", "utctime", "latitude", "latdir", "longitude", "londir", "numsat", "hdop", "altitude", "geoidaltitude", "diffcage", "dgnssrefid"]
  },
  {
    "tableName": "cnav_gngst",
    "field": ["time", "rms", "semimajor", "semiminor", "ellipseorient", "standarddeviationoflatitude", "standarddeviationoflongitude", "standarddeviationofheight"]
  },
  {
    "tableName": "cnav_gnvtg",
    "field": ["time", "courseoverground", "magnetictrack", "speedknots", "speedkmph"]
  },
  {
    "tableName": "em122_kidpt",
    "field": ["time", "waterdepthmeter", "transduceroffset", "maxrange"],
    "noChangeAllowedLength": [0, 300]
  },
  {
    "tableName": "ea640_sddb",
    "field": ["time", "waterdepthfeetfromsurface", "waterdepthmeterfromsurface", "waterdepthfathomfromsurface"]
  }
]
```



```
        "noChangeAllowedLength": [0,300]
    },
    {
        "tableName": "ea640_sddpt",
        "field": ["time", "waterdepthmetertransducer", "transduceroffset"],
        "noChangeAllowedLength": [0,300]
    },
    {
        "tableName": "envtemp_wimhu",
        "field": ["time", "humidity", "temperaturedewpoint"]
    },
    {
        "tableName": "envtemp_wimta",
        "field": ["time", "airtemperature"]
    },
    {
        "tableName": "fugro_glgsv",
        "field":
["time", "messagetotalno", "messageno", "nsatview", "sv1elevation", "sv1azimuth", "sv1snr", "sv2elevation", "sv2azimuth", "sv2snr", "sv3elevation", "sv
3azimuth", "sv3snr", "sv4elevation", "sv4azimuth", "sv4snr", "signalid"]
    },
    {
        "tableName": "fugro_gngsa",
        "field": ["time", "pdop", "hdop", "vdop"]
    },
    {
        "tableName": "fugro_gpdtm",
        "field": ["time", "latoffset", "lonoffset", "altitudeoffset"]
    },
    {
        "tableName": "fugro_gpgga",
        "field": ["time", "utctime", "latitude", "latdir", "longitude", "londir", "numsat", "hdop", "altitude", "geoidaltitude", "diffcage", "dgnssrefid"]
    },
    {
        "tableName": "fugro_gpgll",
        "field": ["time", "latitude", "latdir", "longitude", "londir", "utctime"]
    },
    {
        "tableName": "fugro_gpgsv",
        "field":
["time", "messagetotalno", "messageno", "nsatview", "sv1elevation", "sv1azimuth", "sv1snr", "sv2elevation", "sv2azimuth", "sv2snr", "sv3elevation", "sv
3azimuth", "sv3snr", "sv4elevation", "sv4azimuth", "sv4snr", "signalid"]
    },
    {
        "tableName": "fugro_gprmc",
        "field": ["time", "utctime", "latitude", "latdir", "longitude", "londir", "speedknots", "trackmadegood", "magvar"]
    },
    {
        "tableName": "fugro_gpvtg",
        "field": ["time", "courseoverground", "magnetictrack", "speedknots", "speedkmph"]
    },
    {
        "tableName": "surfmet_gpxsm",
        "field":
["time", "flow1", "fluo", "trans", "windspeed", "winddirection", "airtemperature", "humidity", "airpressure", "parport", "parstarboard", "tirport", "tirstarboard"
]
    },
    {
        "tableName": "winch_winch",
        "field": ["time", "tension", "cableout", "rate", "backtension", "rollangle"]
    },
    {
```




```
"tableName": "phins_pashr",
"field": ["time", "utctime", "heading", "roll", "pitch", "heave", "rollaccuracy", "pitchaccuracy", "headingaccuracy"]
},
{
"tableName": "phins_prdid",
"field": ["time", "pitch", "roll", "heading"]
},
{
"tableName": "posmv_gpgga",
"field": ["time", "utctime", "latitude", "latdir", "longitude", "londir", "numsat", "hdop", "altitude", "geoidaltitude", "diffcage", "dgnssrefid"]
},
{
"tableName": "posmv_gpgll",
"field": ["time", "latitude", "latdir", "longitude", "londir", "utctime"]
},
{
"tableName": "posmv_gpgst",
"field": ["time", "utctime", "rms", "semimajor", "semiminor", "ellipseorient", "standarddeviationoflatitude", "standarddeviationoflongitude", "standarddeviationofheight"]
},
{
"tableName": "posmv_gphdt",
"field": ["time", "headingtrue"]
},
{
"tableName": "posmv_gprmc",
"field": ["time", "utctime", "latitude", "latdir", "longitude", "londir", "speedknots", "trackmadegood", "magvar"]
},
{
"tableName": "posmv_gpvtd",
"field": ["time", "coursetrue", "magnetictrack", "speedknots", "speedkmph"]
},
{
"tableName": "posmv_gpzda",
"field": ["time", "utctime", "day", "month", "year", "zonehour", "zoneminutes"]
},
{
"tableName": "posmv_pashr",
"field": ["time", "utctime", "heading", "roll", "pitch", "heave", "rollaccuracy", "pitchaccuracy", "headingaccuracy"]
},
{
"tableName": "ranger2usbl_gpgga",
"field": ["time", "utctime", "latitude", "latdir", "longitude", "londir", "numsat", "hdop", "altitude", "geoidaltitude", "diffcage", "dgnssrefid"]
},
{
"tableName": "sbe45_nanan",
"field": ["time", "housingwatertemperature", "conductivity", "remotewatertemperature", "salinity", "soundvelocity"],
"noChangeAllowedLength": [0,60,60,60,60,60]
},
{
"tableName": "seapathatt_ingga",
"field": ["time", "utctime", "latitude", "latdir", "longitude", "londir", "numsat", "hdop", "altitude", "geoidaltitude", "diffcage", "dgnssrefid"]
},
{
"tableName": "seapathatt_inzda",
"field": ["time", "utctime", "day", "month", "year", "zonehour", "zoneminutes"]
},
{
"tableName": "seapathatt_psn23",
"field": ["time", "roll", "pitch", "heading", "heave"]
},
}
```



```
{
  "tableName": "seapathpos_gngst",

  "field":["time","utctime","rms","semimajor","semiminor","ellipseorient","standarddeviationoflatitude","standarddeviationoflongitude","standarddeviationofheight"]
},
{
  "tableName": "seapathpos_ingga",
  "field":["time","utctime","latitude","latdir","longitude","lonDir","numsat","hdop","altitude","geoidaltitude","diffcage","dgnssrefid"]
},
{
  "tableName": "seapathpos_inhdt",
  "field":["time","headingtrue"]
},
{
  "tableName": "seapathpos_inrmc",
  "field":["time","utctime","latitude","latdir","longitude","lonDir","speedknots","heading","magvar"]
},
{
  "tableName": "seapathpos_invtg",
  "field":["time","courseoverground","magnetictrack","speedknots","speedkmph"]
},
{
  "tableName": "seapathpos_inzda",
  "field":["time","utctime","day","month","year","zonehour","zoneminutes"]
},
{
  "tableName": "shipsgyro_hehdt",
  "field":["time","headingtrue"]
},
{
  "tableName": "shipsgyro_tivot",
  "field":["time","rateofturn"]
},
{
  "tableName": "skipperlog_iidpt",
  "field":["time","waterdepthmeter","offset","maxrange"]
},
{
  "tableName": "skipperlog_vdmtw",
  "field":["time","watertemperaturecelsius"]
},
{
  "tableName": "skipperlog_vdvbw",
  "field":["time","longitudalwaterspeed","transversewaterspeed","longitudalgroundspeed","transversegroundspeed"]
},
{
  "tableName": "skipperlog_vdvhw",
  "field":["time","headingtrue","headingmagnetic","speedknots","speedkmph"]
},
{
  "tableName": "skipperlog_vdvtg",
  "field":["time","courseoverground","magnetictrack","speedknots","speedkmph"]
},
{
  "tableName": "wamos_pwam",
  "field":["time","hs","tm2","pdir","tp","lp","dp1","tp1","lp1","dp2","tp2","lp2","currentdir","currentspeed"]
}
]
}
```



Sensorconfig file example:

```
{
  "shipId": "DY",
  "name": "EM122 Centre Beam Depth",
  "id": "EM122",
  "sentencesNo": 1,
  "sentences": [
    {
      "name": "KIDPT – Depth of water",
      "talkId": "KI",
      "messageId": "DPT",
      "fieldNo": 3,
      "field": [
        {
          "fieldNumber": "waterDepthMeter",
          "name": "Depth in meters from the transducer Centre Beam",
          "parseConfig": {
            "parseAs": "float",
            "format": "d.d"
          },
        },
      ],
    },
  ],
  "netcdf_attributes": {
    "device_name": "Kongsberg EM122 multibeam echosounder",
    "instrument": {
      "instrument_pid": "http://hdl.handle.net/TBC",
      "uuid": "TOOLnnnn_nnnn",
      "instrument_name": "Kongsberg EM122 multibeam echosounder",
      "serial_number": "123",
      "model_name": "EM122",
      "model_id": "http://vocab.nerc.ac.uk/TBC",
      "date_valid_from": "2013-01-01T00:00:00Z",
      "first_use_date": "2013-01-01T12:00:00Z",
      "comments": "\n2020-09-02T09:48:59Z: XXXX"
    },
  },
  "calibrations": {
    "calibration_function": "https://www.kongsberg.com/globalassets/maritime/km-products/product-documents/317669_em122_installation_manual_english.pdf",
    "input_units": "watt",
    "input_units_id": "http://vocab.nerc.ac.uk/TBC",
    "output_units": "meter",
    "output_units_id": "http://vocab.nerc.ac.uk/TBC",
    "calibration_coefficients": "N/A",
    "date_valid_from": "2013-01-01T00:00:00.000Z",
    "date_valid_to": "2199-01-01T00:00:00.000Z",
    "comments": "\n2020-09-02T09:48:59Z: XXXX"
  },
  "installations": {
    "position_datum": "CRP-E3",
    "orientation_datum": "horizontal",
    "sample_depth_datum": "centrePoint",
    "X": 61.619,
    "Y": -0.005,
    "Z": -0.004,
    "orientation": 0,
    "sample_depth": 0,
    "comments": "\n2020-09-30T09:48:59Z: sensor is hull mounted"
  },
  "coordinates": "TIME DEPTH LATITUDE LONGITUDE",
}
```



```
"_FillValue": -1,
"standard_name": "depth",
"long_name": "depth below surface",
"units": "metres",
"valid_min": 0,
"valid_max": 12000,
"ancillary_variables": "DEPTH_SEADATANET_QC",
"sdn_parameter_urn": "SDN:P01::ADEPZZ01",
"sdn_uom_urn": "SDN:P06::UAA",
"sdn_parameter_name": "Depth (spatial coordinate) relative to water body",
"sdn_uom_name": "Metres",
"atlantosEVId": "",
"atlantosEVname": "",

"axis": "z",
"positive": "down",
"grid_mapping": "",
"sample_rate": 1,
"data_precision": 0.01,

"device_sourcetype": "echosounder",
"source": "Acquisition of EM122 multibeam Echosounder",

"comments": "-"}
},
"options": {},
"unit": "metres"
},
{
  "fieldNumber": "transduceroffset",
  "name": "Offset of transducer from waterline in meters",
  "parseConfig": {
    "parseAs": "float",
    "format": "d.d"
  },
  "netcdf_attributes": {
    "device_name": "Kongsberg EM122 multibeam echosounder",
    "instrument": {
      "instrument_pid": "http://hdl.handle.net/TBC",
      "uuid": "TOOLnnnn_nnnn",
      "instrument_name": "Kongsberg EM122 multibeam echosounder",
      "serial_number": "123",
      "model_name": "EM122",
      "model_id": "http://vocab.nerc.ac.uk/TBC",
      "date_valid_from": "2013-01-01T00:00:00Z",
      "first_use_date": "2013-01-01T12:00:00Z",
      "comments": "\n2020-09-02T09:48:59Z: XXXX"
    },
    "calibrations": {
      "calibration_function": "https://www.kongsberg.com/globalassets/maritime/km-products/product-documents/317669_em122_installation_manual_english.pdf",
      "input_units": "watt",
      "input_units_id": "http://vocab.nerc.ac.uk/TBC",
      "output_units": "meter",
      "output_units_id": "http://vocab.nerc.ac.uk/TBC",
      "calibration_coefficients": "N/A",
      "date_valid_from": "2013-01-01T00:00:00.000Z",
      "date_valid_to": "2199-01-01T00:00:00.000Z",
      "comments": "\n2020-09-02T09:48:59Z: XXXX"
    },
    "installations": {
      "position_datum": "CRP-E3",
```



```
"orientation_datum": "horizontal",
"sample_depth_datum": "centrePoint",
"X": 61.619,
"Y": -0.005,
"Z": -0.004,
"orientation": 0,
"sample_depth": 0,
"comments": "\n2020-09-30T09:48:59Z: sensor is hull mounted"
},
"coordinates": "TIME DEPTH LATITUDE LONGITUDE",
"_FillValue": -1,
"standard_name": "transduceroffset",
"long_name": "Positive offset numbers provide the distance from the transducer to the water line",
"units": "metres",
"valid_min": -50,
"valid_max": 50,
"ancillary_variables": "DEPTH_SEADATANET_QC",
"sdn_parameter_um": "SDN:P01::ADEPZZ01",
"sdn_uom_um": "SDN:P06::UAA",
"sdn_parameter_name": "Depth (spatial coordinate) relative to water body",
"sdn_uom_name": "Metres",
"atlantosEVId": "",
"atlantosEVname": "",

"axis": "z",
"positive": "down",
"grid_mapping": "",
"sample_rate": 1,
"data_precision": 0.01,

"device_sourcetype": "echosounder",
"source": "Acquisition of EM122 multibeam Echosounder",

"comments": "- "
},
"options": {},
"unit": "metres"
},
{
  "fieldNumber": "maxRange",
  "name": "Maximum range Scale in Use",
  "parseConfig": {
    "parseAs": "float",
    "format": "d.d"
  },
},
"netcdf_attributes": {
  "device_name": "Kongsberg EM122 multibeam echosounder",
  "instrument": {
    "instrument_pid": "http://hdl.handle.net/TBC",
    "uuid": "TOOLnnnn_nnnn",
    "instrument_name": "Kongsberg EM122 multibeam echosounder",
    "serial_number": "123",
    "model_name": "EM122",
    "model_id": "http://vocab.nerc.ac.uk/TBC",
    "date_valid_from": "2013-01-01T00:00:00Z",
    "first_use_date": "2013-01-01T12:00:00Z",
    "comments": "\n2020-09-02T09:48:59Z: XXXX"
  },
},
"calibrations": {
  "calibration_function": "https://www.kongsberg.com/globalassets/maritime/km-products/product-
documents/317669_em122_installation_manual_english.pdf",
  "input_units": "watt",
```



```

    "input_units_id": "http://vocab.nerc.ac.uk/TBC",
    "output_units": "meter",
    "output_units_id": "http://vocab.nerc.ac.uk/TBC",
    "calibration_coefficients": "N/A",
    "date_valid_from": "2013-01-01T00:00:00.000Z",
    "date_valid_to": "2199-01-01T00:00:00.000Z",
    "comments": "\n2020-09-02T09:48:59Z: XXXX"
  },
  "installations":{
    "position_datum": "CRP-E3",
    "orientation_datum": "horizontal",
    "sample_depth_datum": "centrePoint",
    "X": 61.619,
    "Y": -0.005,
    "Z": -0.004,
    "orientation": 0,
    "sample_depth": 0,
    "comments": "\n2020-09-30T09:48:59Z: sensor is hull mounted"
  },
  "coordinates": "TIME DEPTH LATITUDE LONGITUDE",
  "_FillValue": -1,
  "standard_name": "maxrange",
  "long_name": "maximum range scale in use",
  "units": "metres",
  "valid_min": -5000,
  "valid_max": 50000,
  "ancillary_variables": "DEPTH_SEADATANET_QC",
  "sdn_parameter_um": "SDN:P01::ADEPZZ01",
  "sdn_uom_um": "SDN:P06::UAA",
  "sdn_parameter_name": "Depth (spatial coordinate) relative to water body",
  "sdn_uom_name": "Metres",
  "atlantosEVId": "",
  "atlantosEVname": "",

  "axis": "z",
  "positive": "down",
  "grid_mapping": "",
  "sample_rate": 1,
  "data_precision": 0.01,

  "device_sourcetype": "echosounder",
  "source": "Acquisition of EM122 multibeam Echosounder",

  "comments": "Always 12000m"
},
"options": {},
"unit": "metres"
}
}
}
}
}

```

Interpolate frequency options:

The general acquisition frequency: 1Hz. Default Interpolate Frequency: 1S

'''



B	business day frequency
C	custom business day frequency (experimental)
D	calendar day frequency
W	weekly frequency
M	month end frequency
SM	semi-month end frequency (15th and end of month)
BM	business month end frequency
CBM	custom business month end frequency
MS	month start frequency
SMS	semi-month start frequency (1st and 15th)
BMS	business month start frequency
CBMS	custom business month start frequency
Q	quarter end frequency
BQ	business quarter end frequency
QS	quarter start frequency
BQS	business quarter start frequency
A	year end frequency
BA, BY	business year end frequency
AS, YS	year start frequency
BAS, BYS	business year start frequency
BH	business hour frequency
H	hourly frequency
T, min	minutely frequency
S	secondly frequency
L, ms	milliseconds
U, us	microseconds
N	nanoseconds
""	

Crontab schedule example:

```
# Edit this file to introduce tasks to be run by cron.
#
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').
#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h dom mon dow  command
11 * * * /usr/bin/python3 /home/rvs/ncc/netcdf_converter/ncg.py /home/rvs/ncc/netcdf_converter/JSON/dy132-1h.json >
/home/rvs/ncc_data/lastrun-debug.txt
21 0 * * * /usr/bin/python3 /home/rvs/ncc/netcdf_converter/ncg.py /home/rvs/ncc/netcdf_converter/JSON/dy132-24h.json >
/home/rvs/ncc_data/lastrun24h.txt
```



```
*1 *** /usr/bin/python3 /home/rvs/ncc/netcdf_converter/ncg-truewind.py /home/rvs/ncc/netcdf_converter/JSON/dy132-truewind.json >
/home/rvs/ncc_data/lastrun-truewind.txt
2 0 *** /usr/bin/python3 /home/rvs/ncc/netcdf_converter/ncg.py /home/rvs/ncc/netcdf_converter/JSON/dy132-1hvary.json >
/home/rvs/ncc_data/lastrun-1hvary5.txt
2 6 *** /usr/bin/python3 /home/rvs/ncc/netcdf_converter/ncg.py /home/rvs/ncc/netcdf_converter/JSON/dy132-1hvary.json >
/home/rvs/ncc_data/lastrun-1hvary5.txt
2 12 *** /usr/bin/python3 /home/rvs/ncc/netcdf_converter/ncg.py /home/rvs/ncc/netcdf_converter/JSON/dy132-1hvary.json >
/home/rvs/ncc_data/lastrun-1hvary5.txt
2 18 *** /usr/bin/python3 /home/rvs/ncc/netcdf_converter/ncg.py /home/rvs/ncc/netcdf_converter/JSON/dy132-1hvary.json >
/home/rvs/ncc_data/lastrun-1hvary5.txt
```

Talker Identifiers the first two character from MessageID

AG Autopilot - General
AP Autopilot - Magnetic
CD Communications – Digital Selective Calling (DSC)
CR Communications – Receiver / Beacon Receiver
CS Communications – Satellite
CT Communications – Radio-Telephone (MF/HF)
CV Communications – Radio-Telephone (VHF)
CX Communications – Scanning Receiver
DF Direction Finder
EC Electronic Chart Display & Information System (ECDIS) EP Emergency
Position Indicating Beacon (EPIRB)
ER Engine Room Monitoring Systems
GP,GN,GL Global Positioning System (GPS)
HC Heading – Magnetic Compass
HE Heading – North Seeking Gyro
HN Heading – Non North Seeking Gyro
II Integrated Instrumentation
IN Integrated Navigation
LC Loran C
P Proprietary Code
RA RADAR and/or ARPA
SD Sounder, Depth
SN Electronic Positioning System, other/general
SS Sounder, Scanning
TI Turn Rate Indicator
VD Velocity Sensor, Doppler, other/general
DM Velocity Sensor, Speed Log, Water, Magnetic
VW Velocity Sensor, Speed Log, Water, Mechanical
WI ,WTWeather Instruments
YX Transducer
ZA Timekeeper – Atomic Clock
ZC Timekeeper – Chronometer
ZQ Timekeeper – Quartz
ZV Timekeeper – Radio Update, WWV or WWVH

"Message"

"AVR"

"Function"

"Trimble proprietary: Time, yaw, tilt, range, mode, PDOP and number of SVs for Moving Baseline RTK"



"ALM"	"Orbital data (almanac) for the specified GPS satellite"
"BPQ"	"Trimble proprietary: Base station position and position quality indicator"
"DG"	"Trimble proprietary: Fugro L-band receiver channel strength"
"DP"	"Fugro proprietary: Time, position and position error statistics"
"DPT"	"Depth & Draught"
"DBS"	"Depth Below Surface"
"DBT"	"Depth below Transducer"
"DBK"	"Depth Below Keel"
"DID"	"Attitude Data"
"VHW"	"Boat Speed and Heading"
"VLW"	"Distance Travelled Through Water"
"VBW"	"Dual Doppler Vector"
"DTM"	"Datum reference"
"GBS"	"GNSS satellite fault detection"
"GFA"	"Data quality check associated with a position solution"
"GGA"	"Time, position and fix related data"
"GGK"	"Trimble proprietary: Time, position, position type and DOP values"
"GLL"	"Latitude, longitude and time data"
"GNS"	"GNSS fix data."
"GRS"	"GNSS range residuals"
"GSA"	"GNSS DOP and active satellites"
"GST"	"Position error statistics"
"GSV"	"Number of SVs in view, PRN, elevation, azimuth and SNR"
"HDT"	"Heading from True North"
"HDG"	"Heading Magnetic"
"HDM"	"Magnetic heading, deviation and variation, from which true headings can be obtained"
"LLQ"	"Leica proprietary Local Grid and Position Quality Indicator"
"MLA"	"Orbital data (almanac) for the specified GLONASS satellite"
"MHU"	"Humidity"
"MTA"	"Air Temperature"
"NAN"	"SBE45 TSG + SBE38 data"
"NCH"	"WINCH data"
"PJK"	"Trimble proprietary: Local coordinate position output"
"PJT"	"Trimble proprietary: Receiver datum information"
"PPS"	"Time and Offset"
"RRE"	"Receiver Autonomous Integrity Monitoring (RAIM) data, reporting Range Residual Errors"
"RMC"	"Position, velocity and time. Note: the NMEA and IEC61162 standards for the RMC sentence differ. Both are shown"
"ROT"	"Rate of Turn"
"SHR"	"Attitude Data"
"PSXN,20-24"	"Attitude Data Seapath"
"TTM"	"Rover and displays baseline information including the baseline distance "
"UTC"	"Time and Date"
"UW"	"AT1M Gravimeter raw output"
"VGK"	"Trimble proprietary: Time, locator vector, type and DOP values"
"VHD"	"Trimble proprietary: Heading information"
"VTG"	"Actual track made good and speed over ground"
"XSE"	"Status Message"
"XSM"	"Surfmet output"
"ZDA"	"UTC day, month, year and local time zone offset"

How to:

How to use Grafana for visualisation or data export?

Grafana server running on the grafana.discovery.local machine also, in the postgresql.discovery.local machine.

The SST already preconfigured the grafana with the actual cruise database and you find some pre created dashboards.



You can access the grafana from your webbrowser using the grafana.discovery.local:3000 or postgresql.discovery.local:3000 url(s)
The SST can share with you the username and the password.

More information about grafana: <https://grafana.com/docs/grafana/latest/best-practices/best-practices-for-creating-dashboards/>