GJSF

GeoMLib (β)

Marine engineering geophysical data processing toolbox

Ivan V. Dmitriev
17.02.2020

Contents

1 Jsf files read and write....4
1.1 16-Byte Message Header....7
1.2 0080=Sonar Data Message....8
1.3 0082=Side Scan Data Message....11
1.4 0086=4400-SAS Processed Data....12
1.5 0182=System Information Message....13
1.6 0426=File Timestamp Message....14
1.7 0428=File Padding Message....15
1.8 2000=Equip Serial Ports Raw Data....16
1.9 2002=NMEA String....17
1.10 2020=Pitch Roll Data....18
1.11 2060=Pressure Sensor Reading....19
1.12 2080=Doppler Velocity Log Data (DVL)....20
1.13 2090=Situation Message....21
1.14 2091=Situation Comprehensive Message (version 2)....22
1.15 2100=Cable Counter Data Message....23
1.16 2101=Kilometer of Pipe Data....24
1.17 2111=Container Timestamp Message....25
1.18 3000=Bathymetric Data Message Type (rev.4 only)....26
1.19 3001=Attitude Message Type....27
1.20 3002=Pressure Message Type....28
1.21 3003=Altitude Message Type....29
1.22 3004=Position Message Type....30
1.23 3005=Status Message Type....31
1.24 9001=Discover-2 General Prefix Message....32
1.25 9002=Discover-2 Situation Data....33
1.26 9003=Discover-2 Acoustic Prefix Message....34
1.27 Private messages: 40, 181, 1009, 1011, 1065, 2040, 3060, 3061, 3062....35
1.28 Examples....36
2 Jsf additional functions....39
2.1 Create DTEN fields (coordinates transformation)....39
2.2 Create Dataset from Jsf-files with Message Type 0080....39
2.3 Create Poly-line from files with MessageType 0080....40
Citation....42

Figures list

Figure 1 Jsf records structure 4
Figure 2 gJsf using examples 38

Tables list

Table 1 Jsf files read and write functions.....4

1 Jsf files read and write

MatLab functions set for reading and writing Records/Messages from Jsf-files. Functions are based on EdgeTech's documents (see ***Table 1***).

Jsf-file includes several numbers of Jsf Records. Each Jsf Record consists of 16-byte Jsf Header and Message (Data Block; ***Figure 1***). The Message (Data Block) was divided in two parts: Head (structured data with fields) and Data (the element of big numeric matrix the same sonar ping, as part of channel data). The Message Types (types of Data Block) were included in Ge0MLib-library showed in ***Table 1***. The JsfHead (16-byte Jsf Header's vector), Head (Message Header's vector) and Data (matrix with SSS, SBP, Bathy or same data) are formed Jsf-variables set.

The original Jsf-file (some "Message was read") is used as "parents" (source for not-read Jsf Messages). When Message writing to file, another messages (not read) are copied to new file from "parent's" file without any changes (in accordance with JsfHead.ROnFlag's value).

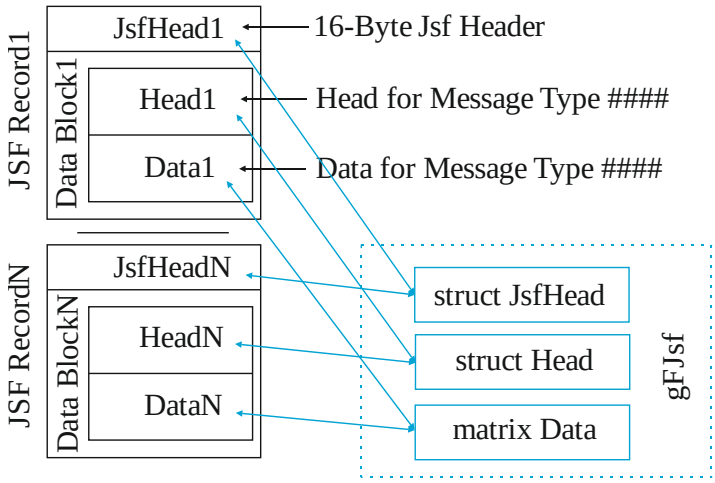

Figure 1 Jsf records structure

Table 1 Jsf files read and write functions

Read/Write function name	Message Type and Description revision
gJsfHeaderRead, gJsfHeaderWrite	16-Byte Message Header. 0004824_REV_1.20 Copyright 2014 by EdgeTech.
gJsf0000Read	Read bytes vectors for Private Messages Types.
gJsf0080Read, gJsf0080Write	80=Sonar Data Message. 0004824_REV_1.20 Copyright 2014 by EdgeTech.
gJsf0082Read	82=Side Scan Data Message. 0004824_REV_1.18 Copyright 2014 by EdgeTech. Warning: NOT TESTED.
gJsf0086Read	86=4400-SAS Processed Data. 990-0000048-1000_Revision:1.7/Nov2006 Warning: NOT TESTED.
gJsf0182Read, gJsf0182Write	182=System Information Message. 0004824_REV_1.20 Copyright 2014 by EdgeTech.

Read/Write function name	Message Type and Description revision
gJsf0426Read	426=File Timestamp Message. 0004824_REV_1.20 Copyright 2016 by EdgeTech. Warning: NOT TESTED.
gJsf0428Read	428=File Padding Message. 0004824_REV_1.20 Copyright 2016 by EdgeTech. Warning: NOT TESTED.
gJsf2000Read, gJsf2000Write	2000=Equip Serial Ports Raw Data. Not annotated / 0004824_REV_1.20 Copyright 2016 by EdgeTech.
gJsf2002Read, gJsf2002Write	2002=NMEA String. 0004824_REV_1.20 Copyright 2016 by EdgeTech.
gJsf2020Read, gJsf2020Write	2020=Pitch Roll Data. 0004824_REV_1.20 Copyright 2016 by EdgeTech.
gJsf2060Read, gJsf2060Write	2060=Pressure Sensor Reading. 0004824_REV_1.20 Copyright 2016 by EdgeTech.
gJsf2080Read	2080=Doppler Velocity Log Data (DVL). 0004824_REV_1.20 Copyright 2016 by EdgeTech.
gJsf2090Read	2090=Situation Message. 0004824_REV_1.20 Copyright 2016 by EdgeTech.
gJsf2091Read	2091=Situation Comprehensive Message (version 2). 0004824_REV_1.20 Copyright 2016 by EdgeTech. Warning: NOT TESTED.
gJsf2100Read	2100=Cable Counter Data Message. 0004824_REV_1.20 Copyright 2016 by EdgeTech. Warning: NOT TESTED.
gJsf2101Read	2101=Kilometer of Pipe Data. 0004824_REV_1.20 Copyright 2016 by EdgeTech. Warning: NOT TESTED.
gJsf2111Read	2111=Container Timestamp Message. 0004824_REV_1.20 Copyright 2016 by EdgeTech. Warning: NOT TESTED.
gJsf3000Read ~~gJsf3000Write~~	3000=BathymetricDataMessageType. 0014932_REV_D, March 2016, Copyright by EdgeTech.
gJsf3001Read ~~gJsf3001Write~~	3001=AttitudeMessageType. 0014932_REV_D, March 2016, Copyright by EdgeTech.
gJsf3002Read ~~gJsf3002Write~~	3002=PressureMessage. 0014932_REV_D, March 2016, Copyright by EdgeTech.
gJsf3003Read ~~gJsf3003Write~~	3003=AltitudeMessageType. 0014932_REV_D, March 2016, Copyright by EdgeTech.
gJsf3004Read ~~gJsf3004Write~~	3004=PositionMessageType. 0014932_REV_D, March 2016, Copyright by EdgeTech.
gJsf3005Read ~~gJsf3005Write~~	3005=StatusMessageType. 0014932_REV_D, March 2016, Copyright by EdgeTech.
gJsf9001Read	9001=Discover-2 General Prefix Message. 0004824_REV_1.18 Copyright 2014 by EdgeTech. Warning: NOT TESTED.
gJsf9002Read	9002=Discover-2 Situation Data. 0004824_REV_1.18 Copyright 2014 by EdgeTech. Warning: NOT TESTED.

Read/Write function name	Message Type and Description revision
gJsf9003Read	9003=Discover-2 Acoustic Prefix Message. 0004824_REV_1.18 Copyright 2014 by EdgeTech. Warning: NOT TESTED.

JsfHead and Head structures

JsfHead structure includes the follow fields from jsf-file:

JsfHead.HMarkerForStart – Marker for the Start of Header = 0x1601;

JsfHead.HVersionOfProtocol – Version of Protocol used;

JsfHead.HSessionIdentifier – Session Identifier;

JsfHead.HMessageType – Message Type;

JsfHead.HCommandType – Command Type;

JsfHead.HSubsystem – Subsystem for a Multi-System Device;

JsfHead.HChannelMulti – Channel for a Multi-Channel Subsystem;

JsfHead.HSequenceNumber – Sequence Number;

JsfHead.HReserved – Reserved;

JsfHead.HSizeFollowingMessage – Size of following Message in Bytes.

JsfHead structure includes the next addition fields:

JsfHead.fName – name of Jsf-file for that structure;

JsfHead.Descript – description for some types of structure fields;

JsfHead.RSeek – file pointer to each Data Block in Jsf-file;

JsfHead.ROnFlag – flag WriteOn for each JsfHead or [Head,Data] (depending on context).

Head structure includes the follow addition fields:

Head.HMessageType – Message Type (scalar) for current Head;

Head.HMessageNum – Number Data Block in JsfHead structure;

Head.HSubsystem – Subsystem for current Head (for example, Message Types 80, 82);

Head.HChannelMulti – Channel for current Head (for example, Message Types 80, 82, 2000, 2002).

Each field is vector (raw data). The row length equals Records number. If Record's field includes more than one number, the numbers come to column. If Record's field is length-changed, then create zero-matrix with column length correspond to maximum field-data length (all "not used" data set to zero). Head structure used same principle.

1.1 16-Byte Message Header

function JsfHead=gJsfHeaderRead (fName,flStat)

Read JsfHead structure from *.jsf file.

Parameters:

JsfHead – JsfHead structure;

fName – the target file name;

flStat – flag for statistics display (1 or 0).

JsfHead include the addition fields:

JsfHead.Descript – description for some fields values;

JsfHead.fName – the name of “parent’s” file;

JsfHead.RSeek – the seeker to Jsf-records in “parent’s” file;

JsfHead.ROnFlag – the flag for on/off records when data were copied from “parent” to new file.

Example: JsfHead=gJsfHeaderRead ('c:\temp\1.jsf',1);

function gJsfHeaderWrite (JsfHead,fNameNew)

Write JsfHead structure to New *.jsf file.

Parameters:

JsfHead – structure with parent jsf-file description;

fNameNew – string; the target file for writing.

gFJsfWriteHeader used file JsfHead.fName as data source. The file JsfHead.fName must be presented; names JsfHead.fName and fNameNew must be different.

You can use JsfHead.ROnFlag to except any record from new Jsf-file. The gJsfHeaderWrite will be write file with “excepted data”.

Example: gJsfHeaderWrite(JsfHead,'c:\temp\1new.xtf');

1.2 0080=Sonar Data Message

function [Head,Data]=gJsf0080Read (JsfHead,ChN,SubSys)

Read [Head,Data] from JsfHead.fName (*.jsf) file for Message Type 0080 (SBP or SSS Data Message). Warning: one sample in packet is used.

Parameters:

JsfHead – Jsf Header structure;

Head – Header structure;

ChN – channel number;

SubSys – subsystem number;

Data – Data Body for channel number ChN, subsystem number SubSys.

Head include the addition fields:

HMessageType – scalar; current message type (e.g. 80);

HChannelMulti = ChN – channel number

HSubsystem = SubSys – subsystem number;

HMessageNum – message number in JsfHead for "parent's" file.

The Sonar Data Message consists of a single ping (receiver sounding period) of data for a single channel (such as Port Side Low Frequency Side-Scan). Usually, Side Scan Systems have two channels of data, port and starboard. Usually, Sub-Bottom Profiling systems have a single channel of data.

Example: [Head,Data]=gJsf0080Read (JsfHead,20,1);

Use gJsfDTEN function to calculate Nav's fields: GpsDay, GpsTime, CompDay, CompTime, GpsE, GpsN, GpsH. The message ID-field is Head.HMessageType = 80.

GpsDay,GpsTime source is NMEA-string; this time is "Position Fix Time". There are follow functions:

Head.GpsDay = gNTime2Time('YDy2Dx',Head.NmeaYear,Head.NmeaDay);

Head.GpsTime = gNTime2Time('HMS32Sd',Head.NmeaHour,Head.NmeaMinutes,Head.NmeaSeconds);

CompDay, CompTime source is "software"; this time is "Data Recorded Time". There are follow functions:

Head.CompDay = gNTime2Time('YDy2Dx',Head.Year,Head.Day);

Head.CompTime = Head.MillisecondsToday./1000;

GpsE, GpsN, GpsH source is NMEA-string; this coordinates are measured in "Position Fix Time". The source is follow fields: Head.X, Head.Y, Head.CoordinateUnits and geodetic parameters for ellipsoid and projection.

function gJsf0080Write(JsfHead,Head,Data,fNameNew,flTraceLenChanged)

Write [JsfHead,Head,Data] to *.jsf file for Message Type 0080 (SBP or SSS Data Message). Warning: one sample in packet is used.

All another messages (not 0080) are copied to new file from “parent’s” file without any changes, in accordance with JsfHead.ROnFlag’s value.

Parameters:

JsfHead – Jsf Header structure;

Head – Header structure;

Data – Data Body for sonar channels;

fNameNew – string, the target file for writing;

flTraceLenChanged – set to one, if Data-Trace Length (Head.NumberDataSamples) was changed and JsfHead.HSizeFollowingMessage field need to correct.

gFJsf0080Write used file JsfHead.fName as data source for JsfHead.RHeaderType~=0. The file XtfHead.fName must be presented; names XtfHead.fName and fNameNew must be different.

If you change Data-Trace Length (Head.NumberDataSamples), the JsfHead.HSizeFollowingMessage field must be correct. Use flTraceLenChanged==1 for auto-correction.

Example: gJsf0080Write(JsfHead,Head,Data,'c:\temp\1new.jsf',0);

Use gJsfDTEN_off function to convert Nav's fields: GpsDay, GpsTime, CompDay, CompTime, GpsE, GpsN, GpsH to original Jsf-fields. There are:

Head.X – 80-83// X in millimeters or decimeters or Longitude in Minutes of Arc / 10000 (see bytes 30-31 and 88-89)

Head.Y – 84-87// Y in millimeters or decimeters or Latitude in 0.0001 Minutes of Arc (see bytes 30-31 and 88-89)

Head.CoordinateUnits – 88-89// Coordinate Units: 1 = X,Y in millimeters; 2 = Longitude, Latitude in minutes of arc times 10000; 3 = X,Y in decimeters

Head.PingTime – 0-3// Ping Time in seconds [since the start of time based on time() function] (1/1/1970) (added in protocol version 8, this field is zero in prior protocol versions)

Head.Year – 156-157// Year Data Recorded (e.g. 2009) (See Bytes 0-3 these 2 time stamps are equivalent and identical). For higher resolution (milliseconds) use the Year, and Day values of bytes 156 to 159, and then use the milliSecondsToday value of bytes 200-203 to complete the timestamp.

Head.Day – 158-159// Day Data Recorded (1–366) (Should not be used)

Head.Hour – 160-161// Hour Data Recorded (see Bytes 200-203) (Should not be used)

Head.Minute – 162-163// Minute Data Recorded (Should not be used)

Head.Second – 164-165// Second Data Recorded (should not be used)

Head.NmeaHour – 186-187// Position Fix Hour (0–23). NOTE: the NAV time is the time of the latitude and longitude fix

Head.NmeaMinutes – 188-189// Position Fix Minutes (0–59). NOTE: the NAV time is the time of the latitude and longitude fix

Head.NmeaSeconds – 190-191// Position Fix Seconds (0–59). NOTE: the NAV time is the time of the latitude and longitude fix.

Head.NmeaDay – 196-197// Position Fix Day (1–366).

Head.NmeaYear – 198-199// Position Fix Year

Head.MillisecondsToday – 200-203// Milliseconds today (since midnight) (use in conjunction with Year/Day to get time of Ping)

1.3 0082=Side Scan Data Message

function [Head,Data]=gJsf0082Read(JsfHead,ChN,SubSys)

Read [Head,Data] from JsfHead.fName (*.jsf) file for Message Type 0082 (Side Scan Data Message). Warning: NOT TESTED.

Parameters:

JsfHead – Jsf Header structure;

Head – Header structure;

ChN – channel number;

SubSys – subsystem number;

Data – Data Body for sonar channel number ChN, subsystem number SubSys.

Head include the addition fields: HMessageType, HChannelMulti, HSubsystem, HMessageNum.

Side-Scan Data Messages are no longer used, and are only described here for historical reasons.

Example: [Head,Data]=gJsf0082Read(JsfHead,20,1);

Use gJsfDTEN function to calculate Nav's fields: CompDay, CompTime. The message ID-field is Head.HMessageType = 82.

CompDay, CompTime source is “software”; this time is “Data Recorded Time”. There are follow functions:

Head.CompDay = gNTime2Time('YDy2Dx',Head.Year,Head.Day);

Head.CompTime = Head.MillisecondsToday./1000;

Use gJsfDTEN_off function to convert Nav's fields: CompDay, CompTime to original Jsf-fields. There are:

Head.MillisecondsToday – 40-43// Milliseconds today;

Head.Year – 44-45// Year;

Head.Day – 46-47// Day (1–366);

Head.Hour – 48-49// Hour of day (0–23);

Head.Minute – 50-51// Minute (0–59);

Head.Second – 52-53// Second (0–59).

1.4 0086=4400-SAS Processed Data

function [Head,Data]=gJsf0086Read(JsfHead,ChN,SubSys)

Read [Head,Data] from JsfHead.fName (*.jsf) file for Message Type 0086 (4400-SAS Processed Data; 990-0000048-1000_Revision:1.7/Nov2006 used). Warning: NOT TESTED.

Parameters:

JsfHead – Xsf Header structure;

Head – Header structure;

ChN – channel number;

SubSys – subsystem number;

Data – Data Body for sonar channel number ChN, subsystem number SubSys.

Head include the addition fields: HMessageType, HChannelMulti, HSubsystem, HMessageNum.

SAS (Synthetic Aperture Sonar) processed data consists of a 152 byte header, followed by port and starboard data. This message is only present if there is SAS image data present. Unused fields are set to 0.

Example: [Head,Data]=gJsf0086Read(JsfHead,0,0);

Use gJsfDTEN function to calculate Nav's fields: CompDay, CompTime, GpsE, GpsN, GpsH. The message ID-field is Head.HMessageType = 0086.

CompDay, CompTime source is “software”; this time is “Data Recorded Time”. There are follow functions:

Head.CompDay = gNTime2Time('YMD32Dx',Head.Year,Head.Month,Head.Day);

Head.CompTime = gNTime2Time('HMS32Sd',Head.Hour,Head.Minute, Head.Second);

GpsE, GpsN, GpsH source is NMEA-string (?); this coordinates are measured in “Position Fix Time”. The source is follow fields: Head.Latitude, Head.Longitude and geodetic parameters for ellipsoid and projection.

Use gJsfDTEN_off function to convert Nav's fields: CompDay, CompTime, GpsE, GpsN, GpsH to original Jsf-fields. There are:

Head.Year – 4-7// Year;

Head.Month – 8-11// Month: 1-12;

Head.Day – 12-15// Day: 1-31;

Head.Hour – 16-19// Hour: 0-23;

Head.Minute – 20-23// Minute: 0-59;

Head.Second – 24-27// Second: 0.0-59.999;

Head.Latitude – 28-35// Latitude in degrees;

Head.Longitude – 36-43// Longitude in degrees.

1.5 0182=System Information Message

function Head=gJsf0182Read(JsfHead)

Read Head from JsfHead.fName (*.jsf) file for Message Type 0182 (System Information Message).

Parameters:

JsfHead – Jsf Header structure;

Head – Message Header structure.

Head include the addition fields: HMessageType, HMessageNum.

The system information message contains details of the system used to acquire data. This message is normally present at the beginning of a JSF file, and may be repeated if configuration parameters change.

Example: Head=gJsf0182Read(JsfHead);

function gJsf0182Write(JsfHead,Head,fNameNew,flTraceLenChanged)

Write [JsfHead,Head] to *.jsf file for Message Type 0182 (System Information Message).

All another messages (not 0182) are copied to new file from “parent’s” file without any changes, in accordance with JsfHead.ROnFlag’s value.

Parameters:

JsfHead – structure with parent jsf-file description;

Head – Message Header structure.

fNameNew – string, the target file for writing;

flTraceLenChanged – set to one, if Reserved3 field length was changed and JsfHead.HSizeFollowingMessage field need to correct.

If you change Reserved3 field length, the JsfHead.HSizeFollowingMessage field must be correct. Use flTraceLenChanged==1 for auto-correction.

Example: gJsf0182Write(JsfHead,Head,'c:\temp\1new.jsf',0);

1.6 0426=File Timestamp Message

function Head=gJsf0426Read(JsfHead)

Read Head from JsfHead.fName (*.jsf) file for Message Type 0426 (File Timestamp Message; 0004824_REV_1.18 used). Warning: NOT TESTED.

Parameters:

JsfHead – Jsf Header structure;

Head – Message Header structure.

Head include the addition fields: HMessageType, HMessageNum.

File timestamp messages are often found at the beginning and end of a file.

Example: Head=gJsf0426Read(JsfHead);

Use gJsfDTEN function to calculate Nav's fields: CompDay, CompTime. The message ID-field is Head.HMessageType = 0426. CompTime source is “software”; this time is “Data Recorded Time”. There is follow function:

Head.CompDay = fix(Head.TimeInSeconds./3600./24)+datenum(1970,1,1);

Head.CompTime = Head.TimeInSeconds-fix(Head.TimeInSeconds./3600./24).*3600.*24 +
Head.MillisecondsCurrentSecond./1000;

Use gJsfDTEN_off function to convert Nav's fields: CompDay, CompTime to original Jsf-fields. There are:

Head.TimeInSeconds – 0-3// Time in seconds (since the start of time based on time() func.) (1/1/1970);

Head.MillisecondsCurrentSecond – 4-7// Milliseconds in the current second.

1.7 0428=File Padding Message

function Head=gJsf0428Read(JsfHead)

Read Head from JsfHead.fName (*.jsf) file for Message Type 0428 (File Padding Message; 0004824_REV_1.18 used). Warning: NOT TESTED.

Parameters:

JsfHead – Jsf Header structure;

Head – Message Header structure.

Head include the addition fields: HMessageType, HMessageNum.

A file padding message is sometimes found at the end of the file. In some implementations files are padded to optimize the write process. These messages should be ignored.

Example: Head=gJsf0428Read(JsfHead);

1.8 2000=Equip Serial Ports Raw Data

function Head=gJsf2000Read(JsfHead,ChN)

Read Head from JsfHead.fName (*.jsf) file for Message Type 2000 (Sonar Virtual Ports Data).

Parameters:

JsfHead – Jsf Header structure;

ChN – channel number;

Head – Message Header structure.

Head include the addition fields: HMessageType, HChannelMulti, HMessageNum.

Each message contains the data flow piece from one virtual serial port.

Example: Head=gJsf2000Read(JsfHead,0);

Use gJsfDTEN function to calculate Nav's fields: CompDay, CompTime. The message ID-field is Head.HMessageType = 2000. CompTime source is “software”; this time is “Data Recorded Time”. There is follow function:

```
Head.CompDay = fix(Head.TimeInSeconds./3600./24)+datenum(1970,1,1);
Head.CompTime = Head.TimeInSeconds-fix(Head.TimeInSeconds./3600./24).*3600.*24 +
                    Head.MillisecondsCurrentSecond./1000;
```

function gJsf2000Write(JsfHead,Head,fNameNew,flTraceLenChanged)

Write [JsfHead,Head] to *.jsf file for Message Type 2000 (Sonar Virtual Ports Data).

All another messages (not 2000) are copied to new file from “parent’s” file without any changes, in accordance with JsfHead.ROnFlag’s value.

Parameters:

JsfHead – structure with parent jsf-file description;

Head – Message Header structure.

fNameNew – string, the target file for writing;

flTraceLenChanged – set to one, if String field length was changed and JsfHead.HSizeFollowingMessage field need to correct.

If you change String field length, the JsfHead.HSizeFollowingMessage field must be correct. Use flTraceLenChanged==1 for auto-correction.

Example: gJsf2000Write(JsfHead,Head,'c:\temp\1new.jsf',0);

Use gJsfDTEN_off function to convert Nav's fields: CompDay, CompTime to original Jsf-fields. There are:

Head.TimeInSeconds – 0-3// Time in seconds (since the start of time based on time() func.) (1/1/1970);

Head.MillisecondsCurrentSecond – 4-7// Milliseconds in the current second.

1.9 2002=NMEA String

function Head=gJsf2002Read(JsfHead,ChN)

Read Head from JsfHead.fName (*.jsf) file for Message Type 2002 (NMEA String).

Parameters:

JsfHead – Jsf Header structure;

ChN – channel number;

Head – Message Header structure.

Head include the addition fields: HMessageType, HChannelMulti, HMessageNum.

Each message contains the data flow piece from serial port.

NMEA String consists of a time stamp followed by a NMEA string as read from a GPS, Gyro or other device. Each message is a single NMEA string excluding the <CR>/<LF>.

Example: Head=gJsf2002Read(JsfHead,0);

Use gJsfDTEN function to calculate Nav's fields: CompDay, CompTime. The message ID-field is Head.HMessageType = 2002. CompTime source is “software”; this time is “Data Recorded Time”. There is follow function:

```
Head.CompDay = fix(Head.TimeInSeconds./3600./24)+datenum(1970,1,1);
Head.CompTime = Head.TimeInSeconds-fix(Head.TimeInSeconds./3600./24).*3600.*24 +
                    Head.MillisecondsCurrentSecond./1000;
```

function gJsf2002Write(JsfHead,Head,fNameNew,flTraceLenChanged)

Write [JsfHead,Head] to *.jsf file for Message Type 2002 (NMEA String).

All another messages (not 2002) are copied to new file from “parent’s” file without any changes, in accordance with JsfHead.ROnFlag’s value.

Parameters:

JsfHead – structure with parent jsf-file description;

Head – Message Header structure.

fNameNew – string, the target file for writing;

flTraceLenChanged – set to one, if String field length was changed and JsfHead.HSizeFollowingMessage field need to correct.

If you change String field length, the JsfHead.HSizeFollowingMessage field must be correct. Use flTraceLenChanged==1 for auto-correction.

Example: gJsf2002Write(JsfHead,Head,'c:\temp\1new.jsf',0);

Use gJsfDTEN_off function to convert Nav's fields: CompDay, CompTime to original Jsf-fields. There are:

Head.TimeInSeconds – 0-3// Time in seconds (since the start of time based on time() func.) (1/1/1970);

Head.MillisecondsCurrentSecond – 4-7// Milliseconds in the current second.

1.10 2020=Pitch Roll Data

function Head=gJsf2020Read(JsfHead)

Read Head from JsfHead.fName (*.jsf) file for Message Type 2020 (Pitch Roll Data).

Parameters:

JsfHead – Jsf Header structure;

Head – Message Header structure.

Head include the addition fields: HMessageType, HMessageNum.

A pitch roll message consists of a single reading from a pitch roll sensor such as a Seatex MRU, TSS or Octans device.

Example: Head=gJsf2020Read(JsfHead);

Use gJsfDTEN function to calculate Nav's fields: CompDay, CompTime. The message ID-field is Head.HMessageType = 2020. CompTime source is “software”; this time is “Data Recorded Time”. There is follow function:

```
Head.CompDay = fix(Head.TimeInSeconds./3600./24)+datenum(1970,1,1);
Head.CompTime = Head.TimeInSeconds-fix(Head.TimeInSeconds./3600./24).*3600.*24 +
                    Head.MillisecondsCurrentSecond./1000;
```

function gJsf2020Write(JsfHead,Head,fNameNew)

Write [JsfHead,Head] to *.jsf file for Message Type 2020 (Pitch Roll Data).

All another messages (not 2020) are copied to new file from “parent’s” file without any changes, in accordance with JsfHead.ROnFlag’s value.

Parameters:

JsfHead – structure with parent jsf-file description;

Head – Message Header structure.

fNameNew – string, the target file for writing.

Example: gJsf2020Write(JsfHead,Head,'c:\temp\1new.jsf');

Use gJsfDTEN_off function to convert Nav's fields: CompDay, CompTime to original Jsf-fields. There are:

Head.TimeInSeconds – 0-3// Time in seconds (since the start of time based on time() func.) (1/1/1970);

Head.MillisecondsCurrentSecond – 4-7// Milliseconds in the current second.

1.11 2060=Pressure Sensor Reading

function Head=gJsf2060Read(JsfHead)

Read Head from JsfHead.fName (*.jsf) file for Message Type 2060 (Pressure Sensor).

Parameters:

JsfHead – Jsf Header structure;

Head – Message Header structure.

Head include the addition fields: HMessageType, HMessageNum.

If a pressure sensor is present in the system these messages will be in the data stream.

Example: Head=gJsf2060Read(JsfHead);

Use gJsfDTEN function to calculate Nav's fields: CompDay, CompTime. The message ID-field is Head.HMessageType = 2060. CompTime source is “software”; this time is “Data Recorded Time”. There is follow function:

```
Head.CompDay = fix(Head.TimeInSeconds./3600./24)+datenum(1970,1,1);
Head.CompTime = Head.TimeInSeconds-fix(Head.TimeInSeconds./3600./24).*3600.*24 +
                    Head.MillisecondsCurrentSecond./1000;
```

function gJsf2060Write(JsfHead,Head,fNameNew)

Write [JsfHead,Head] to *.jsf file for Message Type 2060 (Pressure Sensor).

All another messages (not 2060) are copied to new file from “parent’s” file without any changes, in accordance with JsfHead.ROnFlag’s value.

Parameters:

JsfHead – structure with parent jsf-file description;

Head – Message Header structure.

fNameNew – string, the target file for writing.

Example: gJsf2060Write(JsfHead,Head,'c:\temp\1new.jsf');

Use gJsfDTEN_off function to convert Nav's fields: CompDay, CompTime to original Jsf-fields. There are:

Head.TimeInSeconds – 0-3// Time in seconds (since the start of time based on time() func.) (1/1/1970);

Head.MillisecondsCurrentSecond – 4-7// Milliseconds in the current second.

1.12 2080=Doppler Velocity Log Data (DVL)

function Head=gJsf2080Read(JsfHead)

Read Head from JsfHead.fName (*.jsf) file for Message Type 2080 (Doppler Velocity Log Data). Warning: NOT TESTED.

Parameters:

JsfHead – Jsf Header structure;

Head – Message Header structure.

Head include the addition fields: HMessageType, HMessageNum.

This is data from a DVL (if fitted) and often includes velocity and altitude readings.

Example: Head=gJsf2080Read(JsfHead);

Use gJsfDTEN function to calculate Nav's fields: CompDay, CompTime. The message ID-field is Head.HMessageType = 2080. CompTime source is “software”; this time is “Data Recorded Time”. There is follow function:

```
Head.CompDay = fix(Head.TimeInSeconds./3600./24)+datenum(1970,1,1);
Head.CompTime = Head.TimeInSeconds-fix(Head.TimeInSeconds./3600./24).*3600.*24 +
                    Head.MillisecondsCurrentSecond./1000;
```

1.13 2090=Situation Message

function Head=gJsf2090Read(JsfHead)

Read Head from JsfHead.fName (*.jsf) file for Message Type 2090 (Situation Message). Warning: NOT TESTED.

Parameters:

JsfHead – Jsf Header structure;

Head – Message Header structure.

Head include the addition fields: HMessageType, HMessageNum.

A situation message is a composite of several motion / position sensors. This message is not commonly used.

Example: Head=gJsf2090Read(JsfHead);

Use gJsfDTEN function to calculate Nav's fields: CompDay, CompTime. The message ID-field is Head.HMessageType = 2090. CompTime source is “software”; this time is “Data Recorded Time”. There is follow function:

Head.CompDay = fix(Head.MicrosecondTimestamp./1e7./3600./24)+datenum(1970,1,1);

Head.CompTime = Head.MicrosecondTimestamp./1e7-

fix(Head.MicrosecondTimestamp./1e7./3600./24).*3600.*24.

Use gJsfDTEN_off function to convert Nav's fields: CompDay, CompTime to original Jsf-fields. There are:

Head.TimeInSeconds – 0-3// Time in seconds (since the start of time based on time() func.) (1/1/1970);

Head.MillisecondsCurrentSecond – 4-7// Milliseconds in the current second.

Head.Timestamp – 20-27// Microsecond timestamp (0.01 of a microsecond), us since 12:00:00 am GMT, January 1, 1970

1.14 2091=Situation Comprehensive Message (version 2)

function Head=gJsf2091Read(JsfHead)

Read Head from JsfHead.fName (*.jsf) file for Message Type 2091 (Situation Comprehensive Message, version 2; 0004824_REV_1.18 used). Warning: NOT TESTED.

Parameters:

JsfHead – Jsf Header structure;

Head – Message Header structure.

Head include the addition fields: HMessageType, HMessageNum.

This message contains of a device header followed by a data area. The data area is a composite of several motion / position sensors.

Example: Head=gJsf2091Read(JsfHead);

Use gJsfDTEN function to calculate Nav's fields: CompDay, CompTime. The message ID-field is Head.HMessageType = 2091. CompTime source is “software”; this time is “Data Recorded Time”. There is follow function:

Head.CompDay = fix(Head.MicrosecondTimestamp./1e7./3600./24)+datenum(1970,1,1);

Head.CompTime = Head.MicrosecondTimestamp./1e7-

fix(Head.MicrosecondTimestamp./1e7./3600./24).*3600.*24.

Use gJsfDTEN_off function to convert Nav's fields: CompDay, CompTime to original Jsf-fields. There are:

Head.TimeInSeconds – 0-3// Time in seconds (since the start of time based on time() func.) (1/1/1970);

Head.MillisecondsCurrentSecond – 4-7// Milliseconds in the current second.

Head.Timestamp – 20-27// Microsecond timestamp (0.01 of a microsecond), us since 12:00:00 am GMT, January 1, 1970

1.15 2100=Cable Counter Data Message

function Head=gJsf2100Read(JsfHead)

Read Head from JsfHead.fName (*.jsf) file for Message Type 2100 (Cable Counter Data Message). Warning: NOT TESTED.

Parameters:

JsfHead – Jsf Header structure;

Head – Message Header structure.

Head include the next addition fields: HMessageType, HMessageNum.

Example: Head=gJsf2100Read(JsfHead);

Use gJsfDTEN function to calculate Nav's fields: CompDay, CompTime. The message ID-field is Head.HMessageType = 2100. CompTime source is “software”; this time is “Data Recorded Time”. There is follow function:

Head.CompDay = fix(Head.TimeInSeconds./3600./24)+datenum(1970,1,1);

Head.CompTime = Head.TimeInSeconds-fix(Head.TimeInSeconds./3600./24).*3600.*24 +
Head.MillisecondsCurrentSecond./1000;

Use gJsfDTEN_off function to convert Nav's fields: CompDay, CompTime to original Jsf-fields. There are:

Head.TimeInSeconds – 0-3// Time in seconds (since the start of time based on time() func.) (1/1/1970);

Head.MillisecondsCurrentSecond – 4-7// Milliseconds in the current second.

1.16 2101=Kilometer of Pipe Data

function Head=gJsf2101Read(JsfHead)

Read Head from JsfHead.fName (*.jsf) file for Message Type 2101 (Kilometer of Pipe Data; 0004824_REV_1.18 used). Warning: NOT TESTED.

Parameters:

JsfHead – Jsf Header structure;

Head – Message Header structure.

Head include the addition fields: HMessageType, HMessageNum.

Example: Head=gJsf2101Read(JsfHead);

Use gJsfDTEN function to calculate Nav's fields: CompDay, CompTime. The message ID-field is Head.HMessageType = 2101. CompTime source is “software”; this time is “Data Recorded Time”. There is follow function:

```
Head.CompDay = fix(Head.TimeInSeconds./3600./24)+datenum(1970,1,1);
Head.CompTime = Head.TimeInSeconds-fix(Head.TimeInSeconds./3600./24).*3600.*24 +
                    Head.MillisecondsCurrentSecond./1000;
```

Use gJsfDTEN_off function to convert Nav's fields: CompDay, CompTime to original Jsf-fields. There are:

Head.TimeInSeconds – 0-3// Time in seconds (since the start of time based on time() func.) (1/1/1970);

Head.MillisecondsCurrentSecond – 4-7// Milliseconds in the current second.

1.17 2111=Container Timestamp Message

function Head=gJsf2111Read(JsfHead)

Read Head from JsfHead.fName (*.jsf) file for Message Type 2111 (Container Timestamp Message). Warning: NOT TESTED.

Parameters:

JsfHead – Jsf Header structure;

Head – Message Header structure.

Head include the next addition fields: HMessageType, HMessageNum.

Some messages in a JSF file are generated by external entities then passed to the recording system in containers. These messages are only checked to see if their length matches that specified in the message header, no other validation is performed. These contained messages are always preceded by a container timestamp message. This message contains the receipt timestamp of the container message.

Example: Head=gJsf2111Read(JsfHead);

Use gJsfDTEN function to calculate Nav's fields: CompDay, CompTime. The message ID-field is Head.HMessageType = 2111. CompTime source is “software”; this time is “Data Recorded Time”. There is follow function:

```
Head.CompDay = fix(Head.TimeInSeconds./3600./24)+datenum(1970,1,1);
Head.CompTime = Head.TimeInSeconds-fix(Head.TimeInSeconds./3600./24).*3600.*24 +
                    Head.MillisecondsCurrentSecond./1000;
```

Use gJsfDTEN_off function to convert Nav's fields: CompDay, CompTime to original Jsf-fields. There are:

Head.TimeInSeconds – 0-3// Time in seconds (since the start of time based on time() func.) (1/1/1970);

Head.MillisecondsCurrentSecond – 4-7// Milliseconds in the current second.

1.18 3000=Bathymetric Data Message Type (rev.4 only)

function [Head,Data]=gJsf3000Read(JsfHead,ChN,SubSys)

Read Head from JsfHead.fName (*.jsf) file for Message Type 3000 (Bathymetric Data Message Type; 0014932_REV_D March 2016 used).

Parameters:

JsfHead – Jsf Header structure;

Head – Message Header structure.

Head include the addition fields: HMessageType, HMessageNum.

This is the primary message sent from the Bathymetry System. For each ping, there is one message for the port side, and one for the starboard side.

This message contains the time delay, angle and amplitude of each assumed seafloor echo. Multiple messages of this type are interspersed throughout the data file or data stream.

This message consists of a header, followed by a number of bathymetric samples (numberOfSamples), one corresponding to each sounding point.

Example: Head=gJsf3000Read(JsfHead);

1.19 3001=Attitude Message Type

function Head=gJsf3001Read(JsfHead)

Read Head from JsfHead.fName (*.jsf) file for Message Type 3001 (Attitude Message Type; 0014932_REV_D March 2016 used).

Parameters:

JsfHead – Jsf Header structure;

Head – Message Header structure.

Head include the addition fields: HMessageType, HMessageNum.

AttitudeMessageType is a source for roll, pitch, heave, and heading data. Yaw is not used. Some or all of these fields may be valid (or set to 1) depending on which type(s) of sensor is (are) used.

Example: Head=gJsf3001Read(JsfHead);

1.20 3002=Pressure Message Type

function Head=gJsf3002Read(JsfHead)

Read Head from JsfHead.fName (*.jsf) file for Message Type 3002 (Pressure Message Type; 0014932_REV_D March 2016 used).

Parameters:

JsfHead – Jsf Header structure;

Head – Message Header structure.

Head include the addition fields: HMessageType, HMessageNum.

PressureMessageType is a source for sound velocity, and possibly water temperature, salinity, conductivity, and depth.

Example: Head=gJsf3002Read(JsfHead);

1.21 3003=Altitude Message Type

function Head=gJsf3003Read(JsfHead)

Read Head from JsfHead.fName (*.jsf) file for Message Type 3003 (Altitude Message Type; 0014932_REV_D March 2016 used).

Parameters:

JsfHead – Jsf Header structure;

Head – Message Header structure.

Head include the addition fields: HMessageType, HMessageNum.

AltitudeMessageType is a source for altitude and possibly speed, and heading.

Example: Head=gJsf3003Read(JsfHead);

1.22 3004=Position Message Type

function Head=gJsf3004Read(JsfHead)

Read Head from JsfHead.fName (*.jsf) file for Message Type 3004 (Position Message Type ; 0014932_REV_D March 2016 used).

Parameters:

JsfHead – Jsf Header structure;

Head – Message Header structure.

Head include the addition fields: HMessageType, HMessageNum.

PositionMessageType is a source for position (latitude/longitude), heading, speed, and antenna altitude. UTM Zone, Easting, and Northing fields are not typically used.

Example: Head=gJsf3004Read(JsfHead);

1.23 3005=Status Message Type

function Head=gJsf3005Read(JsfHead)

Read Head from JsfHead.fName (*.jsf) file for Message Type 3005 (Status Message Type; 0014932_REV_D March 2016 used).

Parameters:

JsfHead – Jsf Header structure;

Head – Message Header structure.

Head include the addition fields: HMessageType, HMessageNum.

StatusMessageType is a source for GPS status and quality (i.e. fixed, float, DGPS, etc.). The status includes information such as Number of Satellites and Horizontal Dilution of Precision. The quality indicator is given by its numerical code in the incoming GPS message.

EdgeTech reads these status codes and indicates which message structure provided the information. Currently, there are only two sources that can provide the necessary status information: GGA or GGK.

The Talker ID for the incoming GPS messages does not matter (e.g. $GPGGA, $PTNL,GGK, $INGGK, $GPGGK, etc).

Example: Head=gJsf3005Read(JsfHead);

1.24 9001=Discover-2 General Prefix Message

function Head=gJsf9001Read(JsfHead)

Read Head from JsfHead.fName (*.jsf) file for Message Type 9001 (DISCOVER II General Prefix Message; 0004824_REV_1.18 used). Warning: NOT TESTED.

Parameters:

JsfHead – Jsf Header structure;

Head – Message Header structure.

Head include the addition fields: HMessageType, HMessageNum.

The General Prefix Message proceeds most messages (such as NMEA strings, pitch/ roll, etc.) written by DISCOVER II and provides the supplementary context information for the message that follows it.

Example: Head=gJsf9001Read(JsfHead);

Use gJsfDTEN function to calculate Nav's fields: CompDay, CompTime. The message ID-field is Head.HMessageType = 9001. CompTime source is “software”; this time is “Data Recorded Time”. There is follow function:

Head.CompDay = fix(Head.MicrosecondTimestamp./1e7./3600./24)+datenum(1970,1,1);

Head.CompTime = Head.MicrosecondTimestamp./1e7-
fix(Head.MicrosecondTimestamp./1e7./3600./24).*3600.*24.

Use gJsfDTEN_off function to convert Nav's fields: CompDay, CompTime to original Jsf-fields. There are:

Head.Timestamp – 20-27// Microsecond timestamp (0.01 of a microsecond), us since 12:00:00 am GMT, January 1, 1970

1.25 9002=Discover-2 Situation Data

function Head=gJsf9002Read(JsfHead)

Read Head from JsfHead.fName (*.jsf) file for Message Type 9002 (DISCOVER II Situation Data Message; 0004824_REV_1.18 used). Warning: NOT TESTED.

Parameters:

JsfHead – Jsf Header structure;

Head – Message Header structure.

Head include the addition fields: HMessageType,HMessageNum.

A typical towed system often contains two Sensor Platforms - a boat doing the towing and towed sonar. This message is a summary of the "situation data" for a sensor platform.

This data is written for every defined sensor platform, normally at a 5Hz rate. A sensor platform may contain multiple sensors that provide the same type of data (e.g. Lat/Lon from RMC and GGL).

In this case, the configuration of the run time system contains a prioritized list situation data sources, and it only reports the highest priority available source data.

Example: Head=gJsf9002Read(JsfHead);

Use gJsfDTEN function to calculate Nav's fields: CompDay, CompTime. The message ID-field is Head.HMessageType = 9002. CompTime source is “software”; this time is “Data Recorded Time”. There is follow function:

Head.CompDay = fix(Head.MicrosecondTimestamp./1e7./3600./24)+datenum(1970,1,1);

Head.CompTime = Head.MicrosecondTimestamp./1e7-

fix(Head.MicrosecondTimestamp./1e7./3600./24).*3600.*24.

Use gJsfDTEN_off function to convert Nav's fields: CompDay, CompTime to original Jsf-fields. There are:

Head.Timestamp – 20-27// Microsecond timestamp (0.01 of a microsecond), us since 12:00:00 am GMT, January 1, 1970

1.26 9003=Discover-2 Acoustic Prefix Message

function Head=gJsf9003Read(JsfHead)

Read Head from JsfHead.fName (*.jsf) file for Message Type 9003 (DISCOVER II Acoustic Prefix Message; 0004824_REV_1.18 used). Warning: NOT TESTED.

Parameters:

JsfHead – Jsf Header structure;

Head – Message Header structure.

Head include the addition fields: HMessageType, HMessageNum.

This is a prefix message with supplementary data for the acoustic (type 80) message which follows it. This message consists of a fixed header containing supplementary acoustic information, followed by the situation information for that channel.

Example: Head=gJsf9003Read(JsfHead);

Use gJsfDTEN function to calculate Nav's fields: CompDay, CompTime. The message ID-field is Head.HMessageType = 9003. CompTime source is “software”; this time is “Data Recorded Time”. There is follow function:

Head.CompDay = fix(Head.MicrosecondTimestamp./1e7./3600./24)+datenum(1970,1,1);

Head.CompTime = Head.MicrosecondTimestamp./1e7-
fix(Head.MicrosecondTimestamp./1e7./3600./24).*3600.*24.

Use gJsfDTEN_off function to convert Nav's fields: CompDay, CompTime to original Jsf-fields. There are:

Head.Timestamp – 20-27// Microsecond timestamp (0.01 of a microsecond), us since 12:00:00 am GMT, January 1, 1970

1.27 Private messages: 40, 181, 1009, 1011, 1065, 2040, 3060, 3061, 3062

Message Type 0040 is System Status Message (private). To be ignored by 3rd party readers.

Message Type 0181 is Navigation Offset (private). To be ignored by 3rd party readers.

Message Type 1009 is the Array Information (private). To be ignored by 3rd party readers.

Message Type 1011 is Bathymetric Array Calibration (private). To be ignored by 3rd party readers.

Message Type 1065 is Sonar Performance Counter (private). To be ignored by 3rd party readers.

Message Type 2040 is Miscellaneous Analog Sensors (private; 990-0000048-1000_Revision:1.7/Nov2006 used). This message is from some miscellaneous sensors which are generally included for system diagnostic purposes. To be ignored by 3rd party readers.

Message Types 3060, 3061, 3062 are Internal Bathymetric (private) messages. To be ignored by 3rd party readers.

function [Head,Data]=gJsf0000Read(JsfHead,MessageType,ChN,SubSys)

Read Head and Data from JsfHead.fName (*.jsf) file for Private Messages Types.

Parameters:

JsfHead – Jsf Header structure;

Head – Message Header structure.

Head include the addition fields: HMessageType, HChannelMulti, HSubsystem, HMessageNum.

The bytes vector will be read to Data.

Example: [Head,Data]=gJsf0000Read(JsfHead,2040,0,0);

1.28 Examples

Example 1

%Read JsfHead for SSS ET4200 and show statistic

```
>> JsfHead=gJsfHeaderRead ('c:\SSS_ET4200_2.jsf',1);
Mess: 40=Not Defined, Num: 171 [ Subs: 0, Num: 171; Chan: 0, Num: 171 ]
Mess: 80=Sonar Data Message, Num: 40492 [ Subs: 20, Num: 20246; Chan: 0 1, Num: 10123 10123 ][
Subs: 21, Num: 20246; Chan: 0 1, Num: 10123 10123 ]
Mess: 181=Not Defined, Num: 1 [ Subs: 0, Num: 1; Chan: 0, Num: 1 ]
Mess: 182=System Information Message, Num: 1 [ Subs: 0, Num: 1; Chan: 0, Num: 1 ]
Mess: 2000=Equip Serial Ports Raw Data, Num: 20440 [ Subs: 100, Num: 20440; Chan: 1 5, Num: 9548
10892 ]
Mess: 2002=NMEA String, Num: 913 [ Subs: 100, Num: 913; Chan: 6, Num: 913 ]
Mess: 2020=Pitch Roll Data, Num: 7499 [ Subs: 101, Num: 7499; Chan: 1, Num: 7499 ]
Mess: 2040=Miscellaneous Analog Sensors, Num: 612 [ Subs: 102, Num: 612; Chan: 0, Num: 612 ]
Mess: 2060=Pressure Sensor Reading, Num: 6127 [ Subs: 101, Num: 6127; Chan: 0, Num: 6127 ]
```

% Set JsfHead.ROnFlag at 0 for Messages Type 2000

```
JsfHead.ROnFlag(JsfHead.HMessageType==2000)=0;
```

% Write Jsf Records to a new file

```
>> gJsfHeaderWrite(JsfHead,'c:\SSS_ET4200_3.jsf');
```

%Read JsfHead from new file and show statistic

```
JsfHeadNew=gJsfHeaderRead('c:\SSS_ET4200_3.jsf',1);
Mess: 40=Not Defined, Num: 171 [ Subs: 0, Num: 171; Chan: 0, Num: 171 ]
Mess: 80=Sonar Data Message, Num: 40492 [ Subs: 20, Num: 20246; Chan: 0 1, Num: 10123 10123 ][
Subs: 21, Num: 20246; Chan: 0 1, Num: 10123 10123 ]
Mess: 181=Not Defined, Num: 1 [ Subs: 0, Num: 1; Chan: 0, Num: 1 ]
Mess: 2002=NMEA String, Num: 913 [ Subs: 100, Num: 913; Chan: 6, Num: 913 ]
Mess: 2020=Pitch Roll Data, Num: 7499 [ Subs: 101, Num: 7499; Chan: 1, Num: 7499 ]
Mess: 2040=Miscellaneous Analog Sensors, Num: 612 [ Subs: 102, Num: 612; Chan: 0, Num: 612 ]
Mess: 2060=Pressure Sensor Reading, Num: 6127 [ Subs: 101, Num: 6127; Chan: 0, Num: 6127 ]
```

Example 2

%Read JsfHead without statistic

```
>> JsfHead=gJsfHeaderRead('c:\SSS_ET4200_2.jsf',0);
```

%Read sonar channel (Messages Type 80)

```
>> [Head,Data]=gJsf0080Read(JsfHead,0,20);
```

%Draw sonar channel data (***Figure 2, a***)

```
>> imagesc(Data); colormap('gray');
```

%Set part of Data to 20000 for testing

```
>> Data(:,1:4000)=2e4;
```

% Write Jsf Records with Messages Type 80 to a new file

```
>> gJsf0080Write(JsfHead,Head,Data,'c:\SSS_ET4200_3.jsf',0);
```

% Read new JsfHead without statistic

```
>> JsfHead2=gJsfHeaderRead('c:\SSS_ET4200_3.jsf',0);
```

%Read new sonar channel (Messages Type 80)

```
>> [Head2,Data2]=gJsf0080Read(JsfHead2,0,20);
```

%Draw new sonar channel data (***Figure 2, b***)

```
>> imagesc(Data2); colormap('gray');
```

%Draw sonar track plot (***Figure 2, c***)

```
>> plot(Head2.X,Head2.Y);
```

%Draw sonar depth (***Figure 2, d***)

```
>> plot(Head2.Depth);
```

Example 3

%Read JsfHead without statistic

```
>> JsfHead=gJsfHeaderRead('c:\05_Prog\gSpi\Sample\003_SSS_ET4200_2.jsf',0);
```

%Read raw serial data (Messages Type 2000)

```
>> [Head,Data]=gJsf0080Read(JsfHead,0,20);
```

%Display Messages Type 2000 String field (raw magnetometer integration data)

```
>> a=Head.String(:,1:10);a(a==0)=[];disp(a);
$ 55106.511,0896,0708,0662
$ 55103.279,0908,0708,0676
$ 55099.492,0920,0708,0676
$ 55095.
```

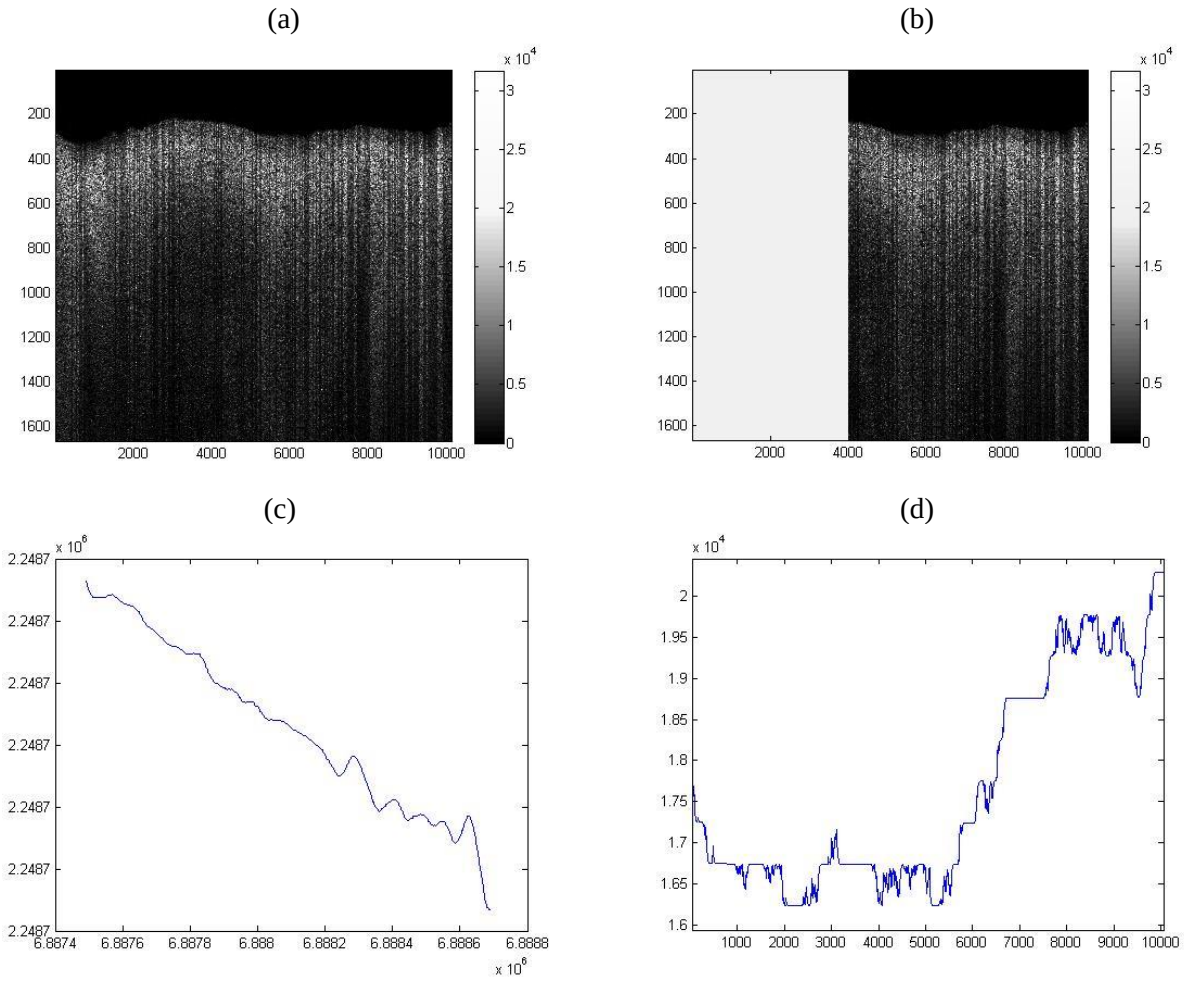

Figure 2 gJsf using examples

2 Jsf additional functions

2.1 Create DTEN fields (coordinates transformation)

function Head=gJsfDTEN(Head,FieldKP,NavS,NavP,varargin)

Create Nav's fields GpsDay,GpsTime,GpsE,GpsN,GpsH from Jsf-files fields (different for Messages Types)

Parameters:

Head – Jsf Header;

FieldKP – the name of Kp-field; there are

NavS – Sensor's Nav-structure (for segy-file);

NavP – Project's Nav-structure;

varargin=H – height in Sensor's Nav-structure datum;

Head – output Header fields: GpsDay,GpsTime,GpsE,GpsN,GpsH will created using fields MillisecondsToday, Year, Day, X, Y.

Used functions: gNTime2Time,gNavDayCheck,gNavCoord2Coord.

Example:

```
>> Head=gJsfDTEN(Head,st,NavS,NavP,H);
```

2.2 Create Dataset from Jsf-files with Message Type 0080

function [JsfHead,Head,Data]=gJsf0080DatasetImport(fName,tmpName,JsfHead,Ch,SubCh,Head,Data,FieldKP,NavS,NavP,PtsFileName,varargin)

Add jsf-files for Message Type 0080 (Sonar Data Message) from file-list or folder to Dataset.

Parameters:

fName – list with file's names or folder name with jsf (Message Type 0080 - Sonar Data Message) will be loaded;

tmpName – folder name for temporary files saving; if isempty, than Data will be empty;

JsfHead – input JsfHead(1..n) structure (can be empty);

SubCh – sub channel number;

Ch – channel number;

Head – input Head(1..n) structure (can be empty);

Data – input cells with Data-matrix or temporary file names;

FieldKP – the name of Kp-field; there are PingNumber or KilometerPipe (if exist);

NavS – sensor's navigation structure (see gNavCoord2Coord) for coordinates transformation;

NavP – project's navigation structure (see gNavCoord2Coord) for coordinates transformation;

PtsFileName – pts file for WaterDepth field calculation ('Xyz2Triang' method);

varargin{1} – if exist, then Head.CoordinateUnits(:)=varargin{1};

[JsfHead,Head,Data] – output variables with added data from files;

Additional fields:

JsfHead(n).fNameTmp – name of temporary file with Data matrix;

GpsDay,GpsTime,GpsE,GpsN,GpsH – fields created by function gJsfDTEN; WaterDepth- calculated using data from PtsFileName.

Example:

```
>> NavS=struct('TargCode',2);NavP=struct('EllipParam',[6378137 0.081819190842],'ProjParam',[0 142 0.9996 500000 0],'ProjForvFunc','gNavGeog2ProjUtm','ProjRevFunc','gNavProjUtm2Geog', 'TargCode',6);
>> [JsfHead,Head,Data]=gJsf0080DatasetImport('c:\jsf80in\','c:\jsf80in\tmp\',JsfHead,0,20,Head,Data, 'PingNumber',NavS,NavP,[],0);
>> [JsfHead,Head,Data]=gJsf0080DatasetImport('c:\jsf80in\','c:\jsf80in\tmp\',[],0,20,[],[],'PingNumber', [],[],[]);
```

2.3 Create Poly-line from files with MessageType 0080

function PL=gJsf0080Dir2PL
(fName,ChN,SubSys,KeyLineDraw,FieldKP,NavS,NavP,NavPTargCode)

Read coordinates form Directory with SSS files (MessageType 0080) to PL-structure. Used for quick Track-plot creation.

Parameters:

fName – reading file name or files name or folder name with files (last name's symbol must be '\');

ChN – channel number;

SubSys – subsystem number;

keyLineDraw – string key for line drawing: '-r','xb', etc;

FieldKP – field from Head structure will copied to PL(n).GpsKP; if empty, than PL(n).GpsKP=1:length(PL(n).GpsE).

NavS – (see gNavCoord2Coord) navigation datum for Sensor, fields: EllipParam, ProjParam, ProjForvFunc, ProjRevFunc, EllipTransParam, EllipForvTransFunc, EllipRevTransFunc, TargCode.

if ~isfield(NavS.EllipTransParam), then transformation Sensor's_Ellipsoid-to-Project's ellipsoid not calculate (fields EllipTransParam, EllipForvTransFunc, EllipRevTransFunc not used).

NavP – (see gNavCoord2Coord) navigation datum for Project, fields: EllipParam, ProjParam, ProjForvFunc, ProjRevFunc, TargCode.

NavPTargCodes= forced output_datum_code (see gNavCoord2Coord); there are: 1)sensor XY; 2)sensor geographic; 3)sensor geosentric; 4)project geocentric; 5)project geographic; 6)project XY.

if isempty(NavPTargCodes), than create NavPTargCodes=NavP.TargCode

if isempty(NavS)&&isempty(NavP)&&isempty(NavPTargCode), than there is no any coordinate transformation will applyed;

PL – output structure: PL(n).PLName; PL(n).Type; PL(n).KeyLineDraw; PL(n).GpsE; PL(n).GpsN; PL(n).GpsKP (to GpsKP write ping number in file)

Used functions: gJsfHeaderRead,gJsf0080Read,gNavCoord2Coord.

Example:

```
>> NavS=struct('TargCode',2);NavP=struct('EllipParam',[6378137 0.081819190842],'ProjParam',[0 141 0.9996 500000 0],'ProjForvFunc','gNavGeog2ProjUtm','ProjRevFunc','gNavProjUtm2Geog', 'TargCode',6);
>> PLJsf=gJsf0080Dir2PL('c:\temp\SSS\3\',0,21,'-b','PingNumber',NavS,NavP,6);
>> gMapPLDraw(100,PLJsf,'PLName');axis equal;
```

Citation

1) DESCRIPTION OF THE EDGETECH (.jsf) FILE FORMAT // Document No. 990-0000048-1000 // Revision: 1.7 / Nov 2006, by EdgeTech.
2) JSF DATA FILE FORMAT, USER SOFTWARE MANUAL // 0004824_REV_1.18 // Copyright 2014 by EdgeTech.
3) BATHYMETRIC DATA MESSAGES FILE FORMAT DESCRIPTION // 0014932_REV_D, March 2016 // Copyright 2016 by EdgeTech.
4) JSF DATA FILE DESCRIPTION // 0004824_REV_1.20, March 2016 // Copyright 2016 by EdgeTech.