



Сдвиг отметок времени в jsf-файлах

Скрипт JsfTimestampShift. Сдвиг отметок
времени в jsf-файлах (записи 0080,
0181,0182, 2000, 2020, 2043, 2060)

Rev. 1.1

Содержание

1. Введение	4
2. Методика устранения сдвига в отметках времени	8
3. Работа со скриптом	10
3.1 Параметры команд Shift01 и Shift01H	10
3.2 Обработка записей 0040, 0080, 0181, 0182, 2000, 2020, 2043, 2060	10
3.3 Результаты работы команд Shift01 и Shift01H	13
3.4 Параметры команды InterpShift	14
4. Методика высокоточной привязки сонограмм ко времени	15
Приложение 1. Содержимое записи 0182	17
Приложение 2. Код скрипта	20

История изменений

Версия	Дата	Описание изменений
1.0	2023-08-27	Документ создан // Иван Д.
1.1	2024-06-07	Документ отредактирован // Иван Д.

Список рисунков

Figure 1.1 Окно исполнения скрипта5

Figure 1.2 Меню “Tools-JSF-JSF Packet Info”5

Figure 1.3 Описание «записей» созданное SonarWiz5

Figure 1.4 Настройки шаблона для CNV-файлов в NavInjecto.....6

Figure 1.5 Неправильные настройки для шаблона NavInjector без десятичных знаков для секунд...7

Figure 1.6 Зоны сжатия и растяжения на сонограмме при использовании времени с усеченной
дробной частью секунд..... 7

Figure 3.1 Пример содержимого файла “files_map.txt”10

Figure 3.2 Исходный вид корневой папки (только входные данные, слева) и вид корневой папки с
обработанными файлами (справа)13

Figure 3.3 Особенность морского дна, отмаркированная для исходного jsf-файла и для файлов с
введенным сдвигом по времени на -3.0 и -3.4 секунды.....14

Figure 3.4 Пример файла “interp.txt”14

Список таблиц

Table 3.1 Корректируемые поля для записи 008011

Table 3.2 Корректируемые поля для записей типа 20xx.....12

1. Введение

Скрипт выполняет сдвиг отметок времени в jsf-файле на заданную величину с точностью до миллисекунд. Каждый jsf-файл содержит последовательность «записей» или «сообщений», приходящих с различных датчиков аппаратуры и записанных в файл в порядке их поступления. Регистрация jsf-файлов, для модификации которых написан скрипт, выполнялась при помощи SSS ET4205; файлы содержат следующие типы записей 0040, 0080, 0181, 0182, 2000, 2020, 2043, 2060. Сдвиг отметок времени выполняется для полей записей 0080, 2000, 2020, 2043, 2060.

Скрипт выполняет пакетную обработку jsf-файлов, расположенных в заданной папке. Для работы скрипта необходим текстовый файл, содержащий имена файлов и значения сдвига по времени для каждого файла. Окно с логом работы скрипта показано на [Figure 1.1](#); при обработке, в окне выводится следующая информация (показана стрелками на [Figure 1.1](#)): имя обрабатываемого файла, название типа обработанной записи, время обработки файла.

Для запуска скрипта необходимы установленные MatLab 2018a (или более поздняя версия) и библиотека geOmlib.

```
Command Window
>> ['Shift01','c:\temp\000\','files_map.txt');JsfTimestampShift;
=== File: F_1824_test1.jsf
Mess: 0040=System Status Message (private), Num: 177 [ Subs: 0, Num: 177; Chan: 0, Num: 177 ]
Mess: 0080=Sonar Data Message, Num: 45032 [ Subs: 20, Num: 22516; Chan: 0 1, Num: 11258 11258 ][ Subs:
Mess: 0181=Navigation Offset, Num: 1 [ Subs: 0, Num: 1; Chan: 0, Num: 1 ]
Mess: 0182=System Information Message / Configuration, Num: 15 [ Subs: 0, Num: 15; Chan: 0, Num: 15 ]
Mess: 2000=Equip Serial Ports Raw Data, Num: 4010 [ Subs: 100, Num: 4010; Chan: 1, Num: 4010 ]
Mess: 2020=Pitch Roll Data, Num: 4010 [ Subs: 101, Num: 4010; Chan: 1, Num: 4010 ]
Mess: 2043=AnalogIO Data (not public), Num: 2732 [ Subs: 102, Num: 2732; Chan: 0, Num: 2732 ]
Mess: 2060=Pressure Sensor Data, Num: 2732 [ Subs: 101, Num: 2732; Chan: 0, Num: 2732 ]
Processing...
Mess 0040_0_000 did nothing
Trace: 5000
Trace: 10000
Trace: 5000
Trace: 10000
Mess 0080_0_020 completed
Trace: 5000
Trace: 10000
Trace: 5000
Trace: 10000
Mess 0080_1_020 completed
Trace: 5000
Trace: 10000
Trace: 5000
Trace: 10000
Mess 0080_0_022 completed
Trace: 5000
Trace: 10000
Trace: 5000
Trace: 10000
Mess 0080_1_022 completed
Mess 0181_0_000 did nothing
Mess 0182_0_000 did nothing
Mess 2000_1_100 completed
Mess 2020_1_101 completed
Mess 2043_0_102 completed
Mess 2060_0_101 completed
Elapsed time is 211.825809 seconds.
=== File: F_1876_test2.jsf
Mess: 0040=System Status Message (private), Num: 224 [ Subs: 0, Num: 224; Chan: 0, Num: 224 ]
Mess: 0080=Sonar Data Message, Num: 57228 [ Subs: 20, Num: 28614; Chan: 0 1, Num: 14307 14307 ][ Subs:
Mess: 0181=Navigation Offset, Num: 1 [ Subs: 0, Num: 1; Chan: 0, Num: 1 ]
Mess: 0182=System Information Message / Configuration, Num: 19 [ Subs: 0, Num: 19; Chan: 0, Num: 19 ]
Mess: 2000=Equip Serial Ports Raw Data, Num: 5097 [ Subs: 100, Num: 5097; Chan: 1, Num: 5097 ]
Mess: 2020=Pitch Roll Data, Num: 5097 [ Subs: 101, Num: 5097; Chan: 1, Num: 5097 ]
Mess: 2043=AnalogIO Data (not public), Num: 3471 [ Subs: 102, Num: 3471; Chan: 0, Num: 3471 ]
Mess: 2060=Pressure Sensor Data, Num: 3471 [ Subs: 101, Num: 3471; Chan: 0, Num: 3471 ]
fx Processing...
```

Figure 1.1 Окно исполнения скрипта

Заметки:

1) Без использования MatLab информацию о типах записей в jsf-файле можно получить при помощи SonarWiz, меню “Tools–JSF–JSF Packet Info” (Figure 1.2). Пример информации о типах записей приведен на Figure 1.3.

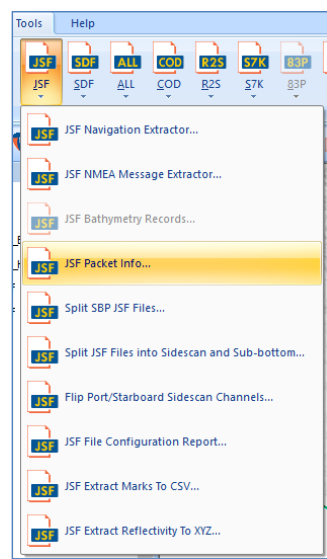


Figure 1.2 Меню “Tools–JSF–JSF Packet Info”

	A	B	C
1	RecordID	RecordName	Count
2	40	SONAR_MESSAGE_SYSTEM_STATUS	525
3	80	SONAR_MESSAGE_DATA	134376
4	181	Unknown Message	1
5	182	SYSTEM_MESSAGE_SONAR_CONFIGURATION	41
6	2000	DEV_SERIAL_DATA	11966
7	2002	DEV_SERIAL_NMEA_DATA	1197
8	2020	DEV_PITCHROLL_DATA	11966
9	2043	DEV_ANALOGIO_DATA_V2	8151
10	2060	DEV_PRESSURE_DATA	8151
11			

Figure 1.3 Описание «записей» созданное SonarWiz

2) В SonarWiz осуществить «сдвиг временных отметок» для координат можно в NavIvjector, введя поправку в поле “Offset in seconds” (Figure 1.4). При этом будет выполнен сдвиг «времени координат» под временные отметки в файле (без коррекции времени внутри файла). Значение сдвига вводится в целых секундах.

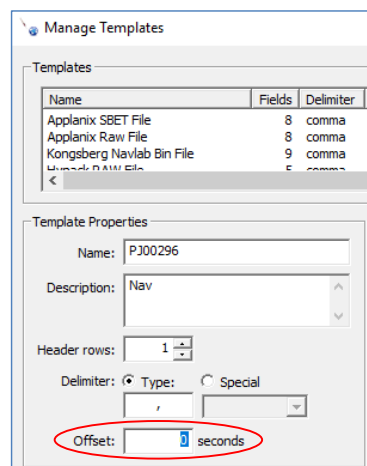


Figure 1.4 Настройки шаблона для CNV-файлов в NavInject

3) Иногда при работе с SonarWiz используются CNV-файлы, которые содержат время, координаты и курс SSS (до загрузки jsf-файлов в SonarWiz в них добавляются данные из CNV-файлов при помощи NavInjector). Частой ошибкой при этом является запись времени (которое выдается QINSy или иной навигационной системой) в формате HH:MM:SS без десятичных знаков для секунд. При этом происходит усечение дробной части, и вместо записи “00:00:40.25” в CNV-файле присутствует “00:00:40”. При этом время в CNV-файле оказывается сдвинутым относительно «реального» времени UTC. Теоретически, такой сдвиг может достигать 1 секунды (что даст погрешность 2.5 метра для скорости буксировки SSS 5 узлов); на практике сдвиг оказывается обычно менее 0.1 секунды (в зависимости от конфигурации QINSy). Проблемы начинаются, если время изменяется в области 00.00 секунд. Например, последовательность секунд [40.01 41.01 41.99 43.01] будет записана как [40 41 41 43]. Для 41-41 мы получим «зону растяжения» на сонограмме (небольшое количество пингов в интервале времени), для 41-43 мы получим «зону сжатия» (относительно большое количество пингов в интервале времени). Пример «неправильного» шаблона для NavInjector без десятичных знаков для секунд показан на Figure 1.5. Пример зон сжатия и растяжения показан на Figure 1.6.

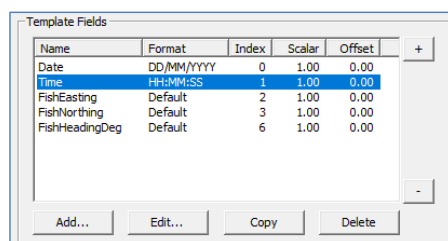


Figure 1.5 Неправильные настройки для шаблона NavInjector без десятичных знаков для секунд

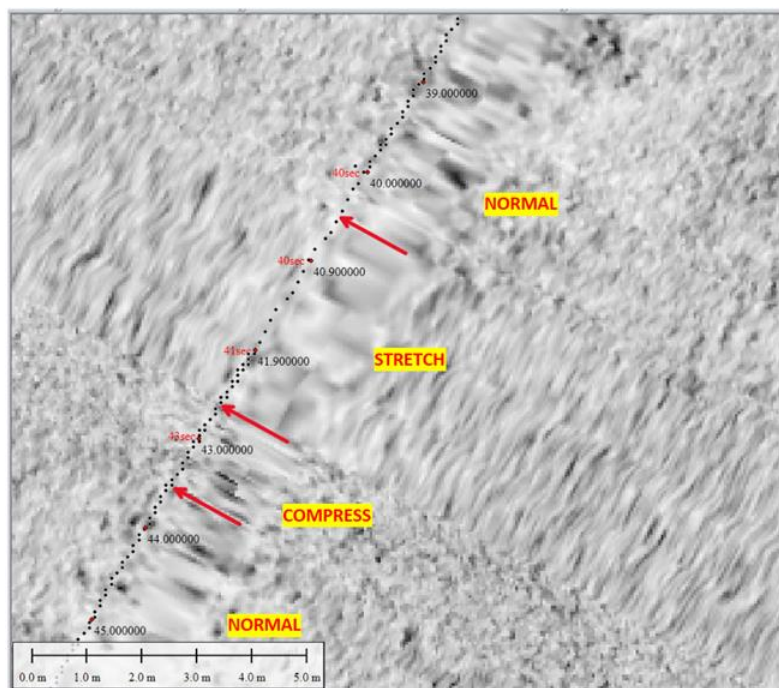


Figure 1.6 Зоны сжатия и растяжения на сонограмме при использовании времени с усеченной дробной частью секунд

2. Методика устранения сдвига в отметках времени

Рассмотрим некоторые характерные особенности для сбоя в привязке ко времени в jsf-файле.

- 1) При отсутствии «синхронизации» программное обеспечение Discoverer помещает внутрь jsf-файла время, определенное по часам компьютера. Для часов современных компьютеров характерно линейное накопление ошибки по времени (1-2 секунды в день) с небольшими осцилляциями (вероятно, осцилляции вызваны изменением температуры печатной платы, питающего напряжения и иными факторами). Таким образом, если на компьютере не включена служба коррекции времени, то уход часов компьютера в первом приближении можно аппроксимировать линейной функцией.
- 2) Величину сдвига по времени можно определять при помощи сопоставления мозаики SSS с данными батиметрии в области зоны надира на сонограмме. Положение характерных элементов морского дна в зоне надира слабо зависит от ошибки, присутствующей в углах поворота пингов SSS.
- 3) Устранение сдвига во времени путем смещения мозаики в горизонтальной плоскости (коррекции координат сонограммы) может привести к неравномерному «расползанию» целей (SSS и MBES) вдоль профиля, поскольку будут проявляться «нелинейные эффекты» от:
 - изменения скорости буксировки вдоль профиля;
 - кривизны линии съемки (для «отклонений от прямой линии» при коррекции координат производится параллельный перенос, а при коррекции времени «смещение вдоль трека»);
 - смещения привязки курса SSS (направления пингов SSS).

Однако, в первом приближении, такое смещение может дать удовлетворительный результат.

- 4) Если смещение SSS вызвано не ошибками времени, а, например, неправильным определением расстояния до SSS по USBL, то компенсация этой ошибки «смещением во времени» также может вызвать «расползание» целей (SSS и MBES) вдоль профиля. Под «расползанием» здесь понимается изменение величины смещения между целями SSS и MBES вдоль профиля на различную величину, в результате чего совмещение одних целей приводит к увеличению расхождения для других целей.

Если в jsf-файлах обнаружена «рассинхронизация» со временем UTC, то оптимальными представляются следующие шаги по определению величины невязок по времени и их удалению.

- 1) Приближенное определение величины сдвига по времени с точностью до секунды при помощи офсета в NavInjector (*Figure 1.4*).
- 2) Подготовка набора файлов при помощи описываемого скрипта с различным сдвигом в пределах половины секунды относительно «приближенно определенной величины сдвига во времени», полученной на предыдущем шаге. Например, со слагаемыми к сдвигу [-0.4 -0.2 0 +0.2 +0.4] секунды. На данном шаге «прошивать значение времени» можно только в запись с кодом 0080 в jsf-файле (и только в высокочастотный канал данных SSS).

3) Загрузка jsf-файлов в SonarWiz и выбор файла (сдвига), для которого SSS и MBES совпадают наилучшим образом.

4) Использование сдвига, определенного на предыдущем шаге, для прошивки времени во все записи jsf-файла (коды 0080, 2000, 2020, 2043, 2060). Шаг 4 вынесен отдельно, поскольку для шага 2 подготовка файлов выполняется в 3.5 раза быстрее (за счет работы только с одним типом записи).

Исходя из того, что уход часов компьютера осуществляется относительно равномерно, можно выполнять пункты 1 или 2 не для всех файлов, а с шагом 3-6 часов. Для остальных файлов можно попробовать выполнить линейную интерполяцию величины сдвига относительно времени записи файла. Контролировать результат интерполяции можно на шаге 2 или шаге 4.

При регистрации файлов длительностью в несколько часов хорошим решением будет от системы «одна невязка для одного файла» перейти к системе «величина невязки во времени». Тогда «сдвиг для привязки ко времени» будет вводиться не для файла целиком, а для каждого отдельного измерения (пинга SSS, измерения датчика глубины и т.д.) внутри jsf-файла. В рассматриваемом скрипте данный подход не реализован.

3. Работа со скриптом

Для работы со скриптом используется система команд следующего формата:

```
{CommandName,Parameter1,...,ParameterN};ScriptName;
```

где

CommandName – имя команды скрипта;

Parameter1,...,ParameterN – параметры для команды скрипта;

ScriptName – имя скрипта.

Для сдвига отметок времени в jsf-файле используется скрипт с именем JsfTimestampShift.

Скрипт может исполнять три команды:

'Shift01H' – сдвиг времени внутри jsf-файла для записи HF (код 0080, подсистема 22, каналы 0 и 1);

'Shift01' – сдвиг времени внутри jsf-файла для записей с кодами 0080, 2000, 2020, 2043, 2060;

'InterpShift' – линейная интерполяция сдвига, рассчитанного для нескольких файлов (нескольких отметок времени) на все остальные файлы (отметки времени).

3.1 Параметры команд Shift01 и Shift01H

Пример строк для запуска скрипта:

```
{'Shift01H','c:\temp\000\','files_map.txt'};JsfTimestampShift;
```

```
{'Shift01','c:\temp\000\','files_map.txt'};JsfTimestampShift;
```

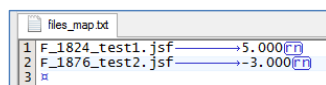
где

-- Shift01 – сдвиг времени внутри jsf-файла для записи HF (код 0080, подсистема 22, каналы 0 и 1);

-- Shift01H – сдвиг времени внутри jsf-файла для записей с кодами 0080, 2000, 2020, 2043, 2060;

-- 'c:\temp\000\' – путь к корневой папке скрипта, которая содержит jsf-файлы для обработки и файл files_map.txt;

-- files_map.txt – файл в корневой папке скрипта, содержащий два столбца. Первый столбец содержит имена обрабатываемых файлов (которые расположены в корневой папке скрипта). Второй столбец содержит величину сдвига для отметки времени в секундах с точностью до миллисекунд. Пример файла приведен на [Figure 3.1](#).



1	F_1824_test1.jsf	5.000
2	F_1876_test2.jsf	-3.000

[Figure 3.1](#) Пример содержимого файла “files_map.txt”

Скриншоты с примером обработанных файлов показаны на [Figure 3.2](#).

3.2 Обработка записей 0040, 0080, 0181, 0182, 2000, 2020, 2043, 2060

Скрипт обрабатывает следующие типы записей в jsf-файле: 0040, 0080, 0181, 0182, 2000, 2020, 2043, 2060. Типы каналов и подсистем, содержащихся в записях, жестко прописаны в теле скрипта. То есть исполняемый код не является универсальным (не определяет самостоятельно

типы записей, номера каналов и подсистем), а привязан к определенной конфигурации аппаратуры (при изменении конфигурации требуется корректировка кода скрипта).

Корректируемые поля/атрибуты записей описаны ниже.

Mess: 0040=System Status Message (private)

Формат записи не опубликован EdgeTech. Действий не производится.

Mess: 0080=Sonar Data Message

Описание формата записи приведено в “0023492_REV_H, JSF file and message descriptions // EdgeTech, April 27, 2023”. Поля/атрибуты, корректируемые для записи, приведены в [Table 3.1](#).

Table 3.1 Корректируемые поля для записи 0080

Name of field	Description
PingTime	Bytes 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)
Year	Bytes 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.
Day	Bytes 158-159// Day Data Recorded (1–366) (Should not be used)
Hour	Bytes 160-161// Hour Data Recorded (see Bytes 200-203) (Should not be used)
Minute	Bytes 162-163// Minute Data Recorded (Should not be used)
Second	Bytes 164-165// Second Data Recorded (should not be used)
MillisecondsToday	Bytes 200-203// Milliseconds today (since midnight) (use in conjunction with Year/Day to get time of Ping)

Заметки:

- 1) «PingTime» соответствует «MillisecondsToday» с усеченной десятичной частью (аналог исполнения функции “fix”). Поля Year, Day, Hour, Minute, Second содержат время, соответствующее «PingTime».
- 2) В jsf-файлах, записанных с помощью ранних версий Discoverer, встречалась рассинхронизация [Year, Day] с [Hour, Minute, Sec]; при этом смена даты выполнялась позже 00:00:00. Данная ошибка была исправлена в последующих версиях.
- 3) Вероятно, время в отмеченных полях измеряется по показаниям часов компьютера, для которых вычисляется сдвиг относительно UTC времени, получаемого из навигационных сообщений ZDA или GGA.
- 4) Поскольку навигационные сообщения характеризуются задержкой (которая может достигать 0.1-0.3 секунд), то «сдвиг относительно UTC времени», вероятно, вычисляется при помощи **осреднения** на относительно длинном отрезке времени с последующей корректировкой. Наименьшей задержкой в отправке навигационных сообщений GGA и GLL характеризовалась GPS Trimble, работающая без поправок (менее 0.1 секунды при работе с поправками); Veripos характеризуется большей задержкой (0.1-0.2 сек при работе с поправками); наибольшей задержкой характеризуется C-Nav (до 0.4 секунд при работе с поправками). Чтобы исключить влияние таких

Commented [Ab1]: осреднения чего? (думаю, нужно уточнить в тексте)

«задержек», желательно использовать Trimble или тот или иной метод регистрации PPS сигнала по часам компьютера с последующей корректировкой времени внутри jsf-файла.

Mess: 0181=Navigation Offset

Формат записи не опубликован EdgeTech. Запись встречается один раз в начале файла и практически всегда содержит последовательность нулевых значений (то есть, не содержит отметок времени). Действий не производится.

Mess: 0182=System Information Message

Частичное описание формата записи приведено в “0023492_REV_H, JSF file and message descriptions // EdgeTech, April 27, 2023”. Запись содержит информацию о конфигурации аппаратуры (вероятно, в основном акустических каналов). С большой вероятностью запись не содержит отметок времени. Действий не производится.

В документе “0023492_REV_H, JSF file and message descriptions // EdgeTech, April 27, 2023” приведено описание первых 6-ти полей записи. Дальнейшие байты записи (около килобайта) обозначены как “Reserved”. Более полную расшифровку записи можно получить при помощи SonarWiz, меню “Tools–JSF–JSF File Configuration Report”. Пример расшифровки приведен в Приложении 1.

Mess: 2000=Equip Serial Ports Raw Data

Формат записи не опубликован EdgeTech. Эмпирически установлено, что запись содержит данные, поступающие на буксируемое устройство с «внешнего» com-порта (например, магнитометр, подключенный в piggy-back). Формат полностью соответствует записи 2020, описание которой приведено в “0023492_REV_H, JSF file and message descriptions // EdgeTech, April 27, 2023”. Корректируемые поля/атрибуты приведены в [Table 3.2](#).

Для рассматриваемой конфигурации ET4205 запись 2000 содержит строку формата TSS1 с показаниями внутреннего компаса-наклонометра. Запись 2000 всегда следует за 2020.

Table 3.2 Корректируемые поля для записей типа 20xx

Name of field	Description
TimeInSeconds	Bytes 0-3 // Ping Time in seconds [since the start of time based on time() function] (1/1/1970))
MillisecondsCurrentSecond	Bytes 4-7 // Milliseconds in the current second

Mess: 2020=Pitch Roll Data

Описание формата записи приведено в “0023492_REV_H, JSF file and message descriptions // EdgeTech, April 27, 2023”. Поля/атрибуты, корректируемые для записи, приведены в [Table 3.2](#).

Mess: 2043=AnalogIO Data

Формат записи не опубликован EdgeTech. SonarWiz выдает имя для записи «AnalogIO Data» ([Figure 1.3](#)). Количество записей в файле равно количеству записей 2060, также они следуют внутри файла после записи 2060. По аналогии с записями 2020&2000 можно предположить, что

2043 содержит сырые данные с датчика давления, которые в «расшифрованном виде» содержатся в 2060. Метка времени была обнаружена в байтах, приведенных в [Table 3.2](#); для данных байтов выполняется коррекция отметок времени.

Mess: 2060=Pressure Sensor Data

Описание формата записи приведено в “0023492_REV_H, JSF file and message descriptions // EdgeTech, April 27, 2023”. Поля/атрибуты, корректируемые для записи, приведены в [Table 3.2](#).

3.3 Результаты работы команд Shift01 и Shift01H

В результате работы скрипта в корневой папке (указанной в параметрах при запуске скрипта) создаются jsf-файлы с введенным сдвигом по времени (согласно значениям сдвига, указанным в файле “files_map.txt”).

Для команды Shift01H к имени обработанного файла добавляется постфикс с величиной сдвига в секундах. Пример имени исходного файла – “F_1824_test1.jsf”, пример имени файла после сдвига отметок времени – “F_1824_test1_+5.200.jsf” (сдвиг на +5.200 секунд). На [Figure 3.2](#) показаны два скриншота. Первый скриншот (слева) – вид корневой папки до запуска скрипта. Второй скриншот – вид корневой папки после запуска скрипта; папка содержит временные файлы (расширение *.tmp) и обработанные файлы. К каждому исходному файлу были применены два различных сдвига для отметок времени ([Figure 3.3](#)).

Для команды Shift01 к имени обработанного файла добавляется постфикс “_shift”. То есть, предполагается, что для одного имени файла будет применен один сдвиг по времени. Также сдвигается время в атрибутах обработанного файла.

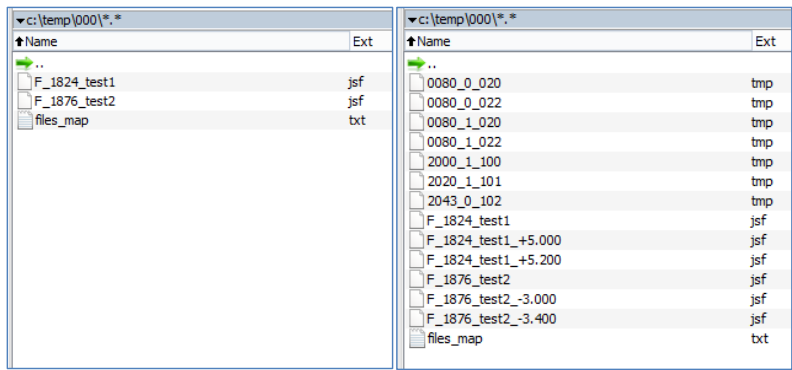


Figure 3.2 Исходный вид корневой папки (только входные данные, слева) и вид корневой папки с обработанными файлами (справа)

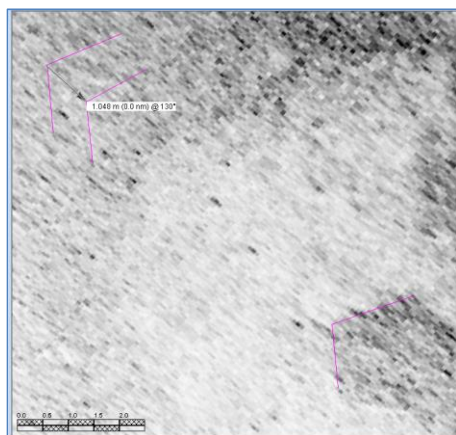


Figure 3.3 Особенность морского дна, отмаркированная для исходного jsf-файла и для файлов с введенным сдвигом по времени на -3.0 и -3.4 секунды

3.4 Параметры команды InterpShift

Команда выполняет линейную интерполяция сдвига, рассчитанного для нескольких файлов (нескольких отметок времени), на все остальные файлы (отметки времени).

Пример строк для запуска скрипта:

```
{'InterpShift','c:\temp\000\interp.txt'};JsfTimestampShift;
```

где

-- InterpShift – имя команды интерполяции;

-- 'c:\temp\000\interp.txt' – путь к текстовому файлу, содержащему четыре столбца: дату, время, имя файла, величину сдвига для отметки времени в секундах с точностью до миллисекунд (или значение 'nan', если величина сдвига неизвестна). Пример файла приведен на [Figure 3.1](#).

Line	Date	Time	File Name	Shift
1	01/07/2022	00:32:00	0956_F065.jsf	-8.4
2	01/07/2022	00:32:00	0956_F065.001.jsf	nan
3	01/07/2022	00:32:00	0956_F065.002.jsf	nan
4	01/07/2022	01:46:29	0957_F060.jsf	-8.6
5	01/07/2022	01:46:29	0957_F060.001.jsf	nan
6	01/07/2022	01:46:29	0957_F060.002.jsf	nan
7	01/07/2022	02:42:35	0958_F160.jsf	-8.8
8	01/07/2022	02:42:35	0958_F160.001.jsf	nan
9	01/07/2022	03:42:34	0959_F155.jsf	nan
10	01/07/2022	03:42:34	0959_F155.001.jsf	nan
11				

Figure 3.4 Пример файла “interp.txt”

Колонка с результатом интерполяции записывается в файл с расширением “*.txt01”. Результат может быть скопирован в Excel и использован для подготовки файла files_map.txt и последующего выполнения команды Shift01.

4. Методика высокоточной привязки сонограмм ко времени

Во время съемки при помощи программы ET Discoverer, для привязки по времени может использоваться три «источника времени»: (1) часы компьютера, (2) метка времени, передаваемая в строке ZDA, (3) метка времени, передаваемая с координатами SSS в строке типа GGA или GLL.

Алгоритм взаимодействия между тремя источниками времени производителем не описывается. Можно предположить, что выполняются следующие **правила для синхронизации по времени**:

- 1) При отсутствии строк ZDA, GGA, GLL, используется время часов компьютера;
- 2) При поступлении строки ZDA часы компьютера периодически синхронизируются со временем, поступающим в строке ZDA. Периодичность этой синхронизации неизвестна; также неизвестно, синхронизация выполняется по данным в единичной строке ZDA, или выполняется сбор статистики с данных строки ZDA за некоторый промежуток времени. Разумным представляется построение алгоритма, при котором синхронизация связана со службой времени Windows. Тогда синхронизация выполняется по следующим правилам (<https://learn.microsoft.com/en-us/windows-server/networking/windows-time-service/windows-time-service-tools-and-settings?tabs=config>):

-- проверяется смещение CurrentTimeOffset (текущее время и время компьютера), чтобы определить меньше ли смещение порогового значения MaxAllowedPhaseOffset;

-- если CurrentTimeOffset < MaxAllowedPhaseOffset, то выполняется постепенная подстройка часов компьютера с помощью «скорости часов» (значение PhaseCorrection, алгоритм отличается в разных версиях Windows);

-- если CurrentTimeOffset > MaxAllowedPhaseOffset, то выполняется немедленное выставление часов компьютера.

В этом случае, по первой строке ZDA будет выполняться корректировка часов компьютера при значительном уходе (действительно, при погрешности в несколько секунд Discoverer выполняет «мгновенную корректировку»). По дальнейшим строкам ZDA будет выполняться «плавная подстройка» часов компьютера при помощи корректировки значения PhaseCorrection (действительно, при поступлении ZDA с ошибкой в 1 секунду перевод часов компьютера не производится – время, записанное в теле сообщения 0080, различается со временем, поступившим в «строке ZDA с погрешностью»).

- 3) При поступлении времени в строках GGA или GLL, оно игнорируется, если поступает строка ZDA. Используется для коррекции часов компьютера как описано выше, если строка ZDA не поступает.

Возможные **источники погрешности привязки ко времени**:

- 1) Привязка выполняется по строке ZDA, поступающей с навигационной системы (а не с GPS-приемника). Например, при поступлении строки с QIN\$y наблюдаются эффекты «повторения времени» (когда две последовательные строки за разные секунды содержат одно и то же время) и «сползания во времени» (время в строке медленно «уходит» с последующей мгновенной

коррекцией). Описанные эффекты проявляются нерегулярно, однако они будут вносить вклад в «подстройку часов» при помощи PhaseCorrection. Вероятнее всего, это будет некоторое «статистическое отставание» часов компьютера от точного времени.

- 2) Задержка поступления строк ZDA, GGA, GLL с навигационного приемника. Для строки ZDA экспериментальная статистика нами не собиралась. Для навигационных сообщений GGA и GLL наименьшей задержкой характеризовалась GPS Trimble – менее 0.1 секунды при работе с поправками; Veripos характеризуется задержкой 0.1-0.2 сек при работе с поправками; наибольшей задержкой характеризуется C-Nav – до 0.4 секунд при работе с поправками. То есть, при получении времени со строк GGA или GLL с C-Nav (при работе с поправками) средняя величина задержки может быть примерно 0.3 секунды (примерно 0.6 метра при скорости 4 узла).

Таким образом, при заводе строки ZDA непосредственно с Trimble погрешность привязки пингов сонограммы будет составлять менее 0.1 секунды (0.2 метра при скорости буксировки 4 узла), что является достаточным для большинства видов инженерных исследований с буксируемым SSS.

Поскольку программа Discoverer установлена, как правило, на ином компьютере, чем навигационная система, то возможны случаи, когда требуется высокоточная привязка ко времени. Например, если съемка выполняется при помощи ROV, то погрешность синхронизации часов компьютера будет влиять на привязку данных MRU и точность привязки углов Heading-Pitch-Roll. С учетом «быстрых корректирующих поворотов», выполняемых автопилотом ROV (например, при влиянии бокового течения) погрешность привязки во времени 0.1-0.3 секунды может быть критической для определения угла направления пингов SSS. Например, теоретически, погрешность определения угла в 2 градуса за счет отставания во времени даст смещение в 0.9 метров на краях сонограммы с наклонной дальностью 25 метров. На практике, рассинхронизация времени SSS и MRU приводит к «раздвиганию» целей на сонограммах за счет рысканья (yaw) ROV, а также к «растягиванию» локальных объектов и неправильному определению их размеров.

Для выполнения высокоточной синхронизации возможно использование следующей методики:

- запись данных SSS выполняется при помощи Discoverer без навигационных строк;
- служба времени Windows отключена;
- выполняется регистрация строки ZDA при помощи программы gLog (записываются строки ZDA и время поступления первого байта строки по часам компьютера);
- выполняется регистрация PPS при помощи FixBox (байт со значением 0, поступающий на COM-порт компьютера) и программы gLog;
- выполняется обработка меток времени, при этом: оценивается стабильность прихода ZDA по часам компьютера (отсутствие высокочастотных флуктуаций); осуществляется привязка времени ZDA к моментам прихода PPS; осуществляется привязка часов компьютера ко времени ZDA-PPS; выполняется прошивка точного времени в файл Jsf.

Приложение 1. Содержимое записи 0182

JSF Configuration Report: D:\temp.jsf
System Type: (51): Unknown system type
Low Rate IO Enabled: 1
Software Version: 795
Number of SubSystems: 3
Number of Serial Ports: 2
System Serial Number: 50009
Flags and Options: 1

Detail for 3 SubSystem(s)...

1. SubSystem Number: 20

 Sonar On: 1

 NetworkWindowInfo (Channel 0 only)...

 Frame Decimation: 1

 Pixel Decimation: 1

 Skip Samples: 0

 Total Samples: 2136

 Max Packet Size: 4272

 Network Window Enables: 1

 Disk Window Info (Channel 0 only)...

 Frame Decimation: 0

 Pixel Decimation: 0

 Skip Samples: 0

 Total Samples: 0

 Max Packet Size: 0

 Disk Window Enables: 16

 Post-processor Mode: 0

 Data Type CH0: 0 [one short per sample - envelope data. No ADC required]

 Bit Flags: 30

 Ping Rate or Range: 40000 (kHz or mm)

 Coupling Parameters

 Subsystem number to trigger this subsystem on: 20

 Trigger on every Nth: 1

 Trigger delay (uS): 0

 Trigger Mode: 0 (0 for internal, 1 for external, 2 for coupled)

 Direct Path Removal: -2

 Direct Path Value: 0

 Transmit Gains: [1000,1000,0,0]

 Total Transmit Gains: [449,449,0,0]

 ADC Gains: [32000,32000,32000,32000]

 Pulse Info

 Pulse File: 4205_SS230Q__25BW__1_5MS.spf

 Pulse Type: 18

 Number of Pulses: 1

 Max Ping Rate: 20.00 (Hz)

 Data Rate: 40.06 (Hz)

 Min Frequency: [217500.00,217500.00] (Hz)

 Max Frequency: [242500.00,242500.00] (Hz)

 Pulse Duration: [1.50,1.50] (ms)

 Pulse ID: 50860 (Hz)

 Pulse Desc: SS230Q 230.0 kHz:BW 25.0 kHz: 1.5 ms:ID 50860:SR 40.1:OL 100%

 System Type Pulse Designed For: 76

 Option Bits: 0

2. SubSystem Number: 21

 Sonar On: 0

 NetworkWindowInfo (Channel 0 only)...

 Frame Decimation: 1

Pixel Decimation: 1
Skip Samples: 0
Total Samples: 4166
Max Packet Size: 8332
Network Window Enables: 1
Disk Window Info (Channel 0 only)...
Frame Decimation: 0
Pixel Decimation: 0
Skip Samples: 0
Total Samples: 0
Max Packet Size: 0
Disk Window Enables: 16
Post-processor Mode: 0
Data Type CH0: 0 [one short per sample - envelope data. No ADC required]
Bit Flags: 10
Ping Rate or Range: 40000 (kHz or mm)
Coupling Parameters
Subsystem number to trigger this subsystem on: 20
Trigger on every Nth: 1
Trigger delay (uS): 0
Trigger Mode: 2 (0 for internal, 1 for external, 2 for coupled)
Direct Path Removal: -2
Direct Path Value: 0
Transmit Gains: [1000,1000,0,0]
Total Transmit Gains: [600,600,0,0]
ADC Gains: [32000,32000,32000,32000]
Pulse Info
Pulse File: 4205_SS540Q__60BW__0_5MS.spf
Pulse Type: 18
Number of Pulses: 1
Max Ping Rate: 60.00 (Hz)
Data Rate: 78.13 (Hz)
Min Frequency: [510000.00,510000.00] (Hz)
Max Frequency: [570000.00,570000.00] (Hz)
Pulse Duration: [0.50,0.50] (ms)
Pulse ID: 50865 (Hz)
Pulse Desc: SS540Q 540.0 kHz:BW 60.0 kHz: 0.5 ms:ID 50865:SR 78.1:OL 100%
System Type Pulse Designed For: 73
Option Bits: 0

3. SubSystem Number: 22

Sonar On: 1
NetworkWindowInfo (Channel 0 only)...
Frame Decimation: 1
Pixel Decimation: 1
Skip Samples: 0
Total Samples: 4902
Max Packet Size: 9804
Network Window Enables: 1
Disk Window Info (Channel 0 only)...
Frame Decimation: 0
Pixel Decimation: 0
Skip Samples: 0
Total Samples: 0
Max Packet Size: 0
Disk Window Enables: 16
Post-processor Mode: 0
Data Type CH0: 0 [one short per sample - envelope data. No ADC required]
Bit Flags: 30
Ping Rate or Range: 40000 (kHz or mm)
Coupling Parameters

Subsystem number to trigger this subsystem on: 20
Trigger on every Nth: 1
Trigger delay (uS): 0
Trigger Mode: 2 (0 for internal, 1 for external, 2 for coupled)
Direct Path Removal: -2
Direct Path Value: 0
Transmit Gains: [1000,1000,0,0]
Total Transmit Gains: [540,540,0,0]
ADC Gains: [32000,32000,32000,32000]
Pulse Info
Pulse File: 4205_SS850__75BW__1_60MS.spf
Pulse Type: 18
Number of Pulses: 1
Max Ping Rate: 18.75 (Hz)
Data Rate: 91.91 (Hz)
Min Frequency: [812500.00,812500.00] (Hz)
Max Frequency: [887500.00,887500.00] (Hz)
Pulse Duration: [1.60,1.60] (ms)
Pulse ID: 50875 (Hz)
Pulse Desc: SS: 850KHz, Bandwidth:75KHz, Length:1.6ms, ID:50875
System Type Pulse Designed For: 75
Option Bits: 0

Detail for 2 Serial Port(s)...

1. Logical Channel: 1

Port Number: 4
Baud Rate: 19200
Parser Algorithm: (4): P
Port Flags: 3B
Port Type: 1
IP4: 0.0.0.0
Latency: 0.000 (s)

2. Logical Channel: 5

Port Number: 3
Baud Rate: 9600
Parser Algorithm: (0): N
Port Flags: 3B
Port Type: 1
IP4: 0.0.0.0
Latency: 0.000 (s)

Приложение 2. Код скрипта

```
%script JsftimeStampShift;
%The script is solving follow task: shift timestamp in a jsf-files. There are follow messages in file:
%0040=System Status Message (private), 0080=Sonar Data Message, 0181=Navigation Offset, 0182=System Information Message / Configuration,
%2000=Equip Serial Ports Raw Data, 2020=Pitch Roll Data, 2043=AnalogIO Data (not public), 2060=Pressure Sensor Data.
%Commands example:
%{'Shift01H','c:\temp\000\','files_map.txt'};JsftimeStampShift;
%{'InterpShift','c:\temp\000\interp.txt'};JsftimeStampShift;
%{'Shift01','c:\temp\000\','files_map.txt'};JsftimeStampShift;
%Input parameters for 'Shift01H' and 'Shift01' command:
% c:\temp\000\ - the name of folder with jsf-files;
% files_map.txt - ASCII file included two columns: file name and shift with 3 digits (for example 5.230), delimiter TAB.

gKey=ans;
if strcmp(gKey{1},'Shift01') %shift timestamp in a jsf-files
    %{'Shift01','c:\temp\000\','files_map.txt'};JsftimeStampShift;
    try dName=gKey{2};catch,dName=input('The name of folder with jsf-files=');end; try fNameMap=gKey{3};catch,fNameMap=input('The ASCII with jsf-file
names and shift=');end;
    fId=fopen([dName
fNameMap'],'r');C=textscan(fId,'%s%f','Delimiter','\t','MultipleDelimsAsOne',0,'EndOfLine','\r\n');fclose(fId);fName=C{1};LL=numel(fName);ddSec=C{2};
    for nn=1:LL,
        tic;dSec=round(ddSec(nn).*1000)./1000;disp(['== File: ' fName{nn} num2str(dSec,'%9.3f')]);gJsftimeHeaderRead([dName
fName{nn}],1);disp('Processing...');
        %Mess: 0040=System Status Message (private), Num: 304 [ Subs: 0, Num: 304; Chan: 0, Num: 304 ]
        disp('Mess 0040_0_000 did nothing');
        %== Message 0080 timestamp bytes:
        %Head.PingTime(m)=fread(fId,1,'int32'); %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(m)=fread(fId,1,'int16'); %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(m)=fread(fId,1,'int16'); %158-159// Day Data Recorded (1-366) (Should not be used)
        %Head.Hour(m)=fread(fId,1,'int16'); %160-161// Hour Data Recorded (see Bytes 200-203) (Should not be used)
        %Head.Minute(m)=fread(fId,1,'int16'); %162-163// Minute Data Recorded (Should not be used)
        %Head.Second(m)=fread(fId,1,'int16'); %164-165// Second Data Recorded (should not be used)
        %Head.MillisecondsToday(m)=fread(fId,1,'uint32'); %200-203// Milliseconds today (since midnight) (use in conjunction with Year/Day to get time
of Ping)
        %Mess: 0080=Sonar Data Message, Num: 77666 [ Subs: 20, Num: 38832; Chan: 0 1, Num: 19416 19416 ][ Subs: 22, Num: 38834; Chan: 0 1, Num: 19417
19417 ]
        JsftimeHead=gJsftimeHeaderRead([dName fName{nn}],0);[Head,Data]=gJsftime0080Read(JsftimeHead,0,20);
        [~,Sd]=gNavTime2Time('Ut2DxSd',Head.PingTime);if ~all(Sd==fix(Head.MillisecondsToday./1000)),error('Incorrect INITIAL time for
0080_0_020');end;
        tmp1=mod(Head.MillisecondsToday,1000)+round(dSec.*1000);tmp2=mod(tmp1,1000);MillisecondsCurrentSecond=tmp2;
        tmp3=(tmp1-tmp2)./1000;Head.PingTime=Head.PingTime+round(tmp3);if ~all(abs(tmp3-round(tmp3))<0.001),error('Incorrect time calculation for
0080_0_020');end;
        [Dx,Sd]=gNavTime2Time('Ut2DxSd',Head.PingTime);[Head.Year,~,~]=datevec(Dx);Head.Day=Dx-
datenum(Head.Year,ones(size(Head.Year)),ones(size(Head.Year)))+1;[Head.Hour,Head.Minute,Head.Second]=gNavTime2Time('Sd2HMS3',Sd);
```

```

Head.MillisecondsToday=Head.Hour.*3600000+Head.Minute.*60000+Head.Second.*1000+MillisecondsCurrentSecond;
gJsf0080Write(JsfHead,Head,Data,[dName '0080_0_020.tmp'],0);disp('Mess 0080_0_020 completed');
%Mess: 0080=Sonar Data Message, Num: 77666 [ Subs: 20, Num: 38832; Chan: 0 1, Num: 19416 19416 ][ Subs: 22, Num: 38834; Chan: 0 1, Num: 19417
19417 ]
JsfHead=gJsfHeaderRead([dName '0080_0_020.tmp'],0);[Head,Data]=gJsf0080Read(JsfHead,1,20);
[~,Sd]=gNavTime2Time('Ut2DxSd',Head.PingTime);if ~all(Sd==fix(Head.MillisecondsToday./1000)),error('Incorrect INITIAL time for
0080_1_020');end;
tmp1=mod(Head.MillisecondsToday,1000)+round(dSec.*1000);tmp2=mod(tmp1,1000);MillisecondsCurrentSecond=tmp2;
tmp3=(tmp1-tmp2)./1000;Head.PingTime=Head.PingTime+round(tmp3);if ~all(abs(tmp3-round(tmp3))<0.001),error('Incorrect time calculation for
0080_1_020');end;
[Dx,Sd]=gNavTime2Time('Ut2DxSd',Head.PingTime);[Head.Year,~,~]=datevec(Dx);Head.Day=Dx-
datenum(Head.Year,ones(size(Head.Year)),ones(size(Head.Year)))+1;[Head.Hour,Head.Minute,Head.Second]=gNavTime2Time('Sd2HMS3',Sd);
Head.MillisecondsToday=Head.Hour.*3600000+Head.Minute.*60000+Head.Second.*1000+MillisecondsCurrentSecond;
gJsf0080Write(JsfHead,Head,Data,[dName '0080_1_020.tmp'],0);disp('Mess 0080_1_020 completed');
%Mess: 0080=Sonar Data Message, Num: 77666 [ Subs: 20, Num: 38832; Chan: 0 1, Num: 19416 19416 ][ Subs: 22, Num: 38834; Chan: 0 1, Num: 19417
19417 ]
JsfHead=gJsfHeaderRead([dName '0080_1_020.tmp'],0);[Head,Data]=gJsf0080Read(JsfHead,0,22);
[~,Sd]=gNavTime2Time('Ut2DxSd',Head.PingTime);if ~all(Sd==fix(Head.MillisecondsToday./1000)),error('Incorrect INITIAL time for
0080_0_022');end;
tmp1=mod(Head.MillisecondsToday,1000)+round(dSec.*1000);tmp2=mod(tmp1,1000);MillisecondsCurrentSecond=tmp2;
tmp3=(tmp1-tmp2)./1000;Head.PingTime=Head.PingTime+round(tmp3);if ~all(abs(tmp3-round(tmp3))<0.001),error('Incorrect time calculation for
0080_0_022');end;
[Dx,Sd]=gNavTime2Time('Ut2DxSd',Head.PingTime);[Head.Year,~,~]=datevec(Dx);Head.Day=Dx-
datenum(Head.Year,ones(size(Head.Year)),ones(size(Head.Year)))+1;[Head.Hour,Head.Minute,Head.Second]=gNavTime2Time('Sd2HMS3',Sd);
Head.MillisecondsToday=Head.Hour.*3600000+Head.Minute.*60000+Head.Second.*1000+MillisecondsCurrentSecond;
gJsf0080Write(JsfHead,Head,Data,[dName '0080_0_022.tmp'],0);disp('Mess 0080_0_022 completed');
%Mess: 0080=Sonar Data Message, Num: 77666 [ Subs: 20, Num: 38832; Chan: 0 1, Num: 19416 19416 ][ Subs: 22, Num: 38834; Chan: 0 1, Num: 19417
19417 ]
JsfHead=gJsfHeaderRead([dName '0080_0_022.tmp'],0);[Head,Data]=gJsf0080Read(JsfHead,1,22);
[~,Sd]=gNavTime2Time('Ut2DxSd',Head.PingTime);if ~all(Sd==fix(Head.MillisecondsToday./1000)),error('Incorrect INITIAL time for
0080_1_022');end;
tmp1=mod(Head.MillisecondsToday,1000)+round(dSec.*1000);tmp2=mod(tmp1,1000);MillisecondsCurrentSecond=tmp2;
tmp3=(tmp1-tmp2)./1000;Head.PingTime=Head.PingTime+round(tmp3);if ~all(abs(tmp3-round(tmp3))<0.001),error('Incorrect time calculation for
0080_1_022');end;
[Dx,Sd]=gNavTime2Time('Ut2DxSd',Head.PingTime);[Head.Year,~,~]=datevec(Dx);Head.Day=Dx-
datenum(Head.Year,ones(size(Head.Year)),ones(size(Head.Year)))+1;[Head.Hour,Head.Minute,Head.Second]=gNavTime2Time('Sd2HMS3',Sd);
Head.MillisecondsToday=Head.Hour.*3600000+Head.Minute.*60000+Head.Second.*1000+MillisecondsCurrentSecond;
gJsf0080Write(JsfHead,Head,Data,[dName '0080_1_022.tmp'],0);disp('Mess 0080_1_022 completed');
%Mess: 0181=Navigation Offset, Num: 1 [ Subs: 0, Num: 1; Chan: 0, Num: 1 ]
disp('Mess 0181_0_000 did nothing');
%Mess: 0182=System Information Message / Configuration, Num: 23 [ Subs: 0, Num: 23; Chan: 0, Num: 23 ]
disp('Mess 0182_0_000 did nothing');
%=== Messages 20xx timestamp bytes:
%Head.TimeInSeconds(m)=fread(fId,1,'int32'); %0-3// Time in seconds (since the start of time based on time() function) (1/1/1970)
%Head.MillisecondsCurrentSecond(m)=fread(fId,1,'int32'); %4-7// Milliseconds in the current second
%Mess: 2000=Equip Serial Ports Raw Data, Num: 6917 [ Subs: 100, Num: 6917; Chan: 1, Num: 6917 ]
JsfHead=gJsfHeaderRead([dName '0080_1_022.tmp'],0);Head=gJsf20xxRead(JsfHead,2000,1,100);
tmp1=Head.MillisecondsCurrentSecond+round(dSec.*1000);tmp2=mod(tmp1,1000);Head.MillisecondsCurrentSecond=tmp2;

```

```

tmp3=(tmp1-tmp2)/1000;Head.TimeInSeconds=Head.TimeInSeconds+round(tmp3);if ~all(abs(tmp3-round(tmp3))<0.001),error('Incorrect time calculation
for 2000_1_100');end;
gJsF20xxWrite(JsfHead,Head,[dName '2000_1_100.tmp'],0);disp('Mess 2000_1_100 completed');
%Mess: 2020=Pitch Roll Data, Num: 6917 [ Subs: 101, Num: 6917; Chan: 1, Num: 6917 ]
JsfHead=gJsFHeaderRead([dName '2000_1_100.tmp'],0);Head=gJsF20xxRead(JsfHead,2020,1,101);
tmp1=Head.MillisecondsCurrentSecond+round(dSec.*1000);tmp2=mod(tmp1,1000);Head.MillisecondsCurrentSecond=tmp2;
tmp3=(tmp1-tmp2)/1000;Head.TimeInSeconds=Head.TimeInSeconds+round(tmp3);if ~all(abs(tmp3-round(tmp3))<0.001),error('Incorrect time calculation
for 2020_1_101');end;
gJsF20xxWrite(JsfHead,Head,[dName '2020_1_101.tmp'],0);disp('Mess 2020_1_101 completed');
%Mess: 2043=AnalogIO Data (not public), Num: 4711 [ Subs: 102, Num: 4711; Chan: 0, Num: 4711 ]
JsfHead=gJsFHeaderRead([dName '2020_1_101.tmp'],0);Head=gJsF20xxRead(JsfHead,2043,0,102);
tmp1=Head.MillisecondsCurrentSecond+round(dSec.*1000);tmp2=mod(tmp1,1000);Head.MillisecondsCurrentSecond=tmp2;
tmp3=(tmp1-tmp2)/1000;Head.TimeInSeconds=Head.TimeInSeconds+round(tmp3);if ~all(abs(tmp3-round(tmp3))<0.001),error('Incorrect time calculation
for 2043_0_102');end;
gJsF20xxWrite(JsfHead,Head,[dName '2043_0_102.tmp'],0);disp('Mess 2043_0_102 completed');
%Mess: 2060=Pressure Sensor Data, Num: 4711 [ Subs: 101, Num: 4711; Chan: 0, Num: 4711 ]
JsfHead=gJsFHeaderRead([dName '2043_0_102.tmp'],0);Head=gJsF20xxRead(JsfHead,2060,0,101);
tmp1=Head.MillisecondsCurrentSecond+round(dSec.*1000);tmp2=mod(tmp1,1000);Head.MillisecondsCurrentSecond=tmp2;
tmp3=(tmp1-tmp2)/1000;Head.TimeInSeconds=Head.TimeInSeconds+round(tmp3);if ~all(abs(tmp3-round(tmp3))<0.001),error('Incorrect time calculation
for 2060_0_101');end;
gJsF20xxWrite(JsfHead,Head,[dName fName{nn}(1:end-4) ' _shift.jsf'],0);disp('Mess 2060_0_101 completed'); % '_' num2str(dSec,'%+9.3f')
%Time for file in powershell (Windows) // to get time in 100uSec use 'powershell (Get-Item
c:\temp\000\01_2060_0_101.tmp).CreationTime.ToFileTime()
system(['powershell (Get-Item ' dName fName{nn}(1:end-4) ' _shift.jsf).CreationTime=(Get-Item ' dName fName{nn} ').CreationTime.AddSeconds('
num2str(fix(dSec),'%9.0f') ').AddMilliseconds(' num2str(rem(dSec.*1000,1000),'%9.0f') ') ']); % '_' num2str(dSec,'%+9.3f')
system(['powershell (Get-Item ' dName fName{nn}(1:end-4) ' _shift.jsf).LastWriteTime=(Get-Item ' dName fName{nn} ').LastWriteTime.AddSeconds('
num2str(fix(dSec),'%9.0f') ').AddMilliseconds(' num2str(rem(dSec.*1000,1000),'%9.0f') ') ']);
system(['powershell (Get-Item ' dName fName{nn}(1:end-4) ' _shift.jsf).LastAccessTime=(Get-Item ' dName fName{nn} ').LastAccessTime.AddSeconds('
num2str(fix(dSec),'%9.0f') ').AddMilliseconds(' num2str(rem(dSec.*1000,1000),'%9.0f') ') ']);
toc;
end;
clearvars dName fNameMap fId C fName LL ddSec nn dSec JsfHead Head Data Sd tmp1 tmp2 MillisecondsCurrentSecond tmp3 Dx Sd
end;
if strcmp(gKey{1},'Shift01H') %shift timestamp in a jsf-files for HF only
%{'Shift01H','c:\temp\000\','files_map.txt'};JsfTimestampShift;
try dName=gKey{2};catch,dName=input('The name of folder with jsf-files=');end; try fNameMap=gKey{3};catch,fNameMap=input('The ASCII with jsf-file
names and shift=');end;
fId=fopen([dName
fNameMap],'r');C=textscan(fId,'%s%f','Delimiter','\t','MultipleDelimsAsOne',0,'EndOfLine','\r\n');fclose(fId);fName=C{1};LL=numel(fName);ddSec=C{2};
for nn=1:LL,
tic;dSec=round(ddSec(nn).*1000)/1000;disp(['== File: ' fName{nn} num2str(dSec,'%+9.3f')]);disp('Processing...');
%Mess: 0080=Sonar Data Message, Num: 77666 [ Subs: 20, Num: 38832; Chan: 0 1, Num: 19416 19416 ][ Subs: 22, Num: 38834; Chan: 0 1, Num: 19417
19417 ]
JsfHead=gJsFHeaderRead([dName fName{nn}],1);[Head,Data]=gJsF0080Read(JsfHead,0,22);
[~,Sd]=gNavTime2Time('Ut2DxSd',Head.PingTime);if ~all(Sd==fix(Head.MillisecondsToday./1000)),error('Incorrect INITIAL time for
0080_0_022');end;
tmp1=mod(Head.MillisecondsToday,1000)+round(dSec.*1000);tmp2=mod(tmp1,1000);MillisecondsCurrentSecond=tmp2;
tmp3=(tmp1-tmp2)/1000;Head.PingTime=Head.PingTime+round(tmp3);if ~all(abs(tmp3-round(tmp3))<0.001),error('Incorrect time calculation for
0080_0_022');end;

```

```

[Dx,Sd]=gNavTime2Time('Ut2DxSd',Head.PingTime);[Head.Year,~,~]=datevec(Dx);Head.Day=Dx-
datenum(Head.Year,ones(size(Head.Year)),ones(size(Head.Year)))+1;[Head.Hour,Head.Minute,Head.Second]=gNavTime2Time('Sd2HMS3',Sd);
Head.MillisecondsToday=Head.Hour.*3600000+Head.Minute.*60000+Head.Second.*1000+MillisecondsCurrentSecond;
gJsF0080Write(JsFHead,Head,Data,[dName '0080_0_022.tmp'],0);disp('Mess 0080_0_022 completed');
%Mess: 0080=Sonar Data Message, Num: 77666 [ Subs: 20, Num: 38832; Chan: 0 1, Num: 19416 19416 ][ Subs: 22, Num: 38834; Chan: 0 1, Num: 19417
19417 ]
JsFHead=gJsFHeaderRead([dName '0080_0_022.tmp'],0);[Head,Data]=gJsF0080Read(JsFHead,1,22);
[~,Sd]=gNavTime2Time('Ut2DxSd',Head.PingTime);if ~all(Sd==fix(Head.MillisecondsToday./1000)),error('Incorrect INITIAL time for
0080_1_022');end;
tmp1=mod(Head.MillisecondsToday,1000)+round(dSec.*1000);tmp2=mod(tmp1,1000);MillisecondsCurrentSecond=tmp2;
tmp3=(tmp1-tmp2)./1000;Head.PingTime=Head.PingTime+round(tmp3);if ~all(abs(tmp3-round(tmp3))<0.001),error('Incorrect time calculation for
0080_1_022');end;
[Dx,Sd]=gNavTime2Time('Ut2DxSd',Head.PingTime);[Head.Year,~,~]=datevec(Dx);Head.Day=Dx-
datenum(Head.Year,ones(size(Head.Year)),ones(size(Head.Year)))+1;[Head.Hour,Head.Minute,Head.Second]=gNavTime2Time('Sd2HMS3',Sd);
Head.MillisecondsToday=Head.Hour.*3600000+Head.Minute.*60000+Head.Second.*1000+MillisecondsCurrentSecond;
gJsF0080Write(JsFHead,Head,Data,[dName fName(nn)(1:end-4) '_' num2str(dSec,'%+9.3f') '.jsf'],0);disp('Mess 0080_1_022 completed');
toc;
end;
clearvars dName fNameMap fId C fName LL ddSec nn dSec JsFHead Head Data Sd tmp1 tmp2 MillisecondsCurrentSecond tmp3 Dx Sd
end;
if strcmp(gKey{1},'InterpShift') %
%{'InterpShift','c:\temp\000\interp.txt'};JsFTimestampShift;
try fName=gKey{2};catch,fName=input('File for interpolation=');end
%16/07/2023 21:12:41 GFO_0632_HDD_P195.jsf -2.6
fId=fopen(fName,'r');C=textscan(fId,'%f%f%f%f%f%f%f','Delimiter','\t','MultipleDelimsAsOne',0,'EndOfLine','\r\n');fclose(fId);
GpsDay=gNavTime2Time('YMD32Dx',C{3},C{2},C{1});GpsTime=gNavTime2Time('HMS32Sd',C{4},C{5},C{6});
F=struct('fName','','YY',C{3},'MM',C{2},'DD',C{1},'hh',C{4},'mm',C{5},'ss',C{6},'t',(GpsDay-GpsDay(1)).*3600.*24+(GpsTime-
GpsTime(1)),'GpsDay',GpsDay,'GpsTime',GpsTime,'LName',C(7),'Shift',C(8),'ShiftOut',zeros(size(C{8})));
L=isnan(F.Shift);
F.ShiftOut=interp1(F.t(~L),F.Shift(~L),F.t,'linear','extrap');
%plShift=polyfit(F.t(~L),F.Shift(~L),1);F.ShiftOut=polyval(plShift,F.t);
plot(F.t./3600,F.ShiftOut,'.-');hold on;plot(F.t(~L)./3600,F.Shift(~L),'*r');
dlmwrite([fName '01'],F.ShiftOut,'precision','%+9.3f','delimiter','\t','newline','pc');
clearvars rootD dz fName nn LL fId C
end;

%mail@ge0milib.com 03/08/2023

```