



INTERNAL DOCUMENT No. 332

SWALES Sonic Buoy - telemetered data report

K G Birch, C H Clayson & R W Pascal

1994

**INSTITUTE OF OCEANOGRAPHIC SCIENCES
DEACON LABORATORY**

INTERNAL DOCUMENT No. 332

SWALES Sonic Buoy - telemetered data report

K G Birch, C H Clayson & R W Pascal

1994

Wormley
Godalming
Surrey GU8 5UB UK
Tel +44-(0)428 684141
Telex 858833 OCEANS G
Telefax +44-(0)428 683066

DOCUMENT DATA SHEET

<i>AUTHOR</i> BIRCH, K G, CLAYSON, C H & PASCAL, R W		<i>PUBLICATION DATE</i> 1994
<i>TITLE</i> SWALES Sonic Buoy - telemetered data report.		
<i>REFERENCE</i> Institute of Oceanographic Sciences Deacon Laboratory, Internal Document, No. 332, 89pp. (Unpublished manuscript)		
<i>ABSTRACT</i> The objective of the telemetry data system was the collection of continuous raw data from the Solent Sonic anemometer, whilst the sensor was deployed at sea, on the Sonic Buoy. The success of the system exceeded that expected, with a 98% data blocks received without errors. The report details the data collected together with file formats. Software developed for file format conversion and error analysis of the data is included. Arrangements for the shore station erected for SWALES are detailed, also included is copy of the licence obtained from Department of Trade and Industry Radiocommunications Agency.		
<i>KEYWORDS</i> SONIC ANEMOMETER SWALES VHF TELEMETRY		
<i>ISSUING ORGANISATION</i> Institute of Oceanographic Sciences Deacon Laboratory Wormley, Godalming Surrey GU8 5UB. UK. Director: Colin Summerhayes DSc Telephone Wormley (0428) 684141 Telex 858833 OCEANS G. Facsimile (0428) 683066		
Copies of this report are available from: <i>The Library,</i>		<i>PRICE</i> £0.00

1.	OVERVIEW	7
2.	INTRODUCTION	7
3.	EXPERIMENT LOCATION	7
3.1.	Equipment	8
3.2.	System Diagram	9
3.3.	Map of the Area	10
4.	DATA SOURCES AND PROCESSING	11
4.1.	Data Quality Checking	12
4.2.	Summary of Data Produced	12
4.2.1.	Raw Data Files	12
5.	ACKNOWLEDGEMENTS	13
6.	APPENDIX A DATA FORMATS	14
6.1.	Appendix A.1 Raw Data Files	14
6.2.	Appendix A.2 FASTCOM Files	15
6.3.	Appendix A.3 Error Files	15
6.3.1.	.PDE and .UDE files	15
6.3.2.	SUMMARY.RDE and SUMMARY.NDE files	16
7.	APPENDIX B RADIO LICIENCE	17
8.	APPENDIX C SHORE STATION SPECIFICATION	22
9.	APPENDIX D RAW DATA FILES	24
10.	APPENDIX E SOFTWARE LISTINGS	46
10.1.	Appendix E.1 THORREAD.C	46
10.2.	Appendix E.2 THORSTAT.C	50
10.3.	Appendix E.3 THORSUMM.C	54
10.4.	Appendix E.4 REPLAY2.C	58

1. OVERVIEW

The objective of the Telemetry Data system was the collection of continuous raw data from the Solent Sonic anemometer, whilst the sensor was deployed at sea, on the Sonic Buoy.

The transmitter sub-system operations are transparent to the other buoy sub-systems, requiring only the power from the primary battery packs. Once a second serial data blocks with, on average, 20.83 sets of measurements are encoded by a radio modem and transmitted via an omni-directional aerial to a shore receiver. The decoded data are error checked and written to time stamped data files on a dedicated personal computer (PC).

Off line processing, checks for short records and gaps in the data. The data files are re-formatted by the removal of the 'block counter' and block length words, which are encoded within the saved dataset, to the standard Sonic data file format, with header information derived from the file time stamp.

The success of the system exceeded that expected, with a 98% data blocks received without errors.

The report details the data collected together with file formats. Software developed for file format conversion and error analysis of the data is included.

Arrangements for the shore station erected for SWALES are detailed, also included is copy of the licence obtained from Department of Trade and Industry Radiocommunications Agency

2. INTRODUCTION

The development of the Sonic Buoy has provided the opportunity for spectral wind data measurements without the air flow distortion associated with a ship's superstructure. However the change in the sensor platform from a ship to a buoy, requires consideration of the effect of the waves on the air flow and the energy induced by the motion of the buoy.

To quantify possible platform effects, the collection of raw data was considered necessary over a range of wind and sea conditions. But as the data rate of the Sonic anemometer is ~750K Bytes per hour, onboard data storage was not considered viable. The solution adopted was to transmit the data in real time to an on shore recording system.

The Sonic Buoy has a 3m diameter discus surface following hull with a 2.5m high tower on which the sonic anemometer is mounted. The buoy is orientated into the wind by a 'V' shaped vane with the anemometer mounted on the buoy's windward leading edge. This deployment technique places the anemometer sensor head within ~3m of the wave

3. EXPERIMENT LOCATION

The mooring site for the Sonic Buoy was 51° 29.5' N and 4° 45.0' W, with the receiving Land Station at Hill Farm, Manorbier, Nr Tenby, 51° 47.0' N and 4° 47.0' W.

Details of the shore station building and electrical specification at Hill Farm Manobier are in Appendix C

3.1. Equipment

The buoy hardware is a 0.5 Watt RF transmitter with a 3db co-linear omni-directional whip aerial. It is powered by a 12V supply derived in the buoy raw data logger module.

The operation of the Sonic anemometer, on buoy, is under the control of the Sonic processor, which configures the baud rate, sets the mode of operation, and polls the sensor for data when in the prompted mode. At the start of each quarter hour the Sonic processor re-configures the anemometer from unprompted mode into prompted mode with the control characters "PP". At the completion of collecting $12 * 1024$ data samples the characters "UU" are used to reset the anemometer into the unprompted mode.

To allow the shore station PC to correlate the recorded raw data with the buoy processed data these "PP" and "UU" characters are also transmitted in between the anemometer data blocks. Both serial data lines between the anemometer and the Sonic processor are 'OR' gated together for input into the radio modem. After 'power on' of the transmitter and the anemometer, firmware within the modem detects the transition between modes. Transmission from the buoy commences immediately these criteria have been reached. The mode switch commands are used by the receiving station to synchronise datasets with the buoy. Each occurrence of a mode change initiates the opening a new file. The duty cycle between prompted and unprompted data is approximately 2:1.

The system configuration as used in the buoy is shown in the diagram below.

At the Shore receiving station a 6 element VHF yagi directional aerial with 8.5db gain was mounted with clear line of sight to the buoy. A high gain mast head amplifier, which was powered from the receiving station via the co-ax cable, enabled the recording system to be mounted away from the aerial position.

Within the receiving station the signal is decoded by the customised receiver and recorded to hard disc. The logging PC opens a new data file when transmitted data contains the control characters "PP" or "UU". Each file is named with the time that the file is opened suffixed by ".PDT" prompted or ".UDT" unprompted.

Details of the radio licence are in Appendix B

Thorcom Ltd, Unit4,96B Blackpole Trading Estate West, Worcester, WR3 8TJ

Telephone Number 0905 756700

3.2. System Diagram

Base/Shore Station
Receive Only Equipment

Buoy
Transmitter

6 element Yagi
8.5dB gain
eg. Jaybeam 7042



Pre-amp
Mast-head mounted
RF pre-amplifier

RG213 co-ax feeder

In-door equipment

Receiver Unit
153MHz
LMW TR17 or PMR
radio transceiver
configured for
receive only operation

RLC220
Radio
Modem

+12V

+12V PSU

PC running data
gathering program

RS232

ANTENNA
3dB co-linear
eg. Jaybeam 7074



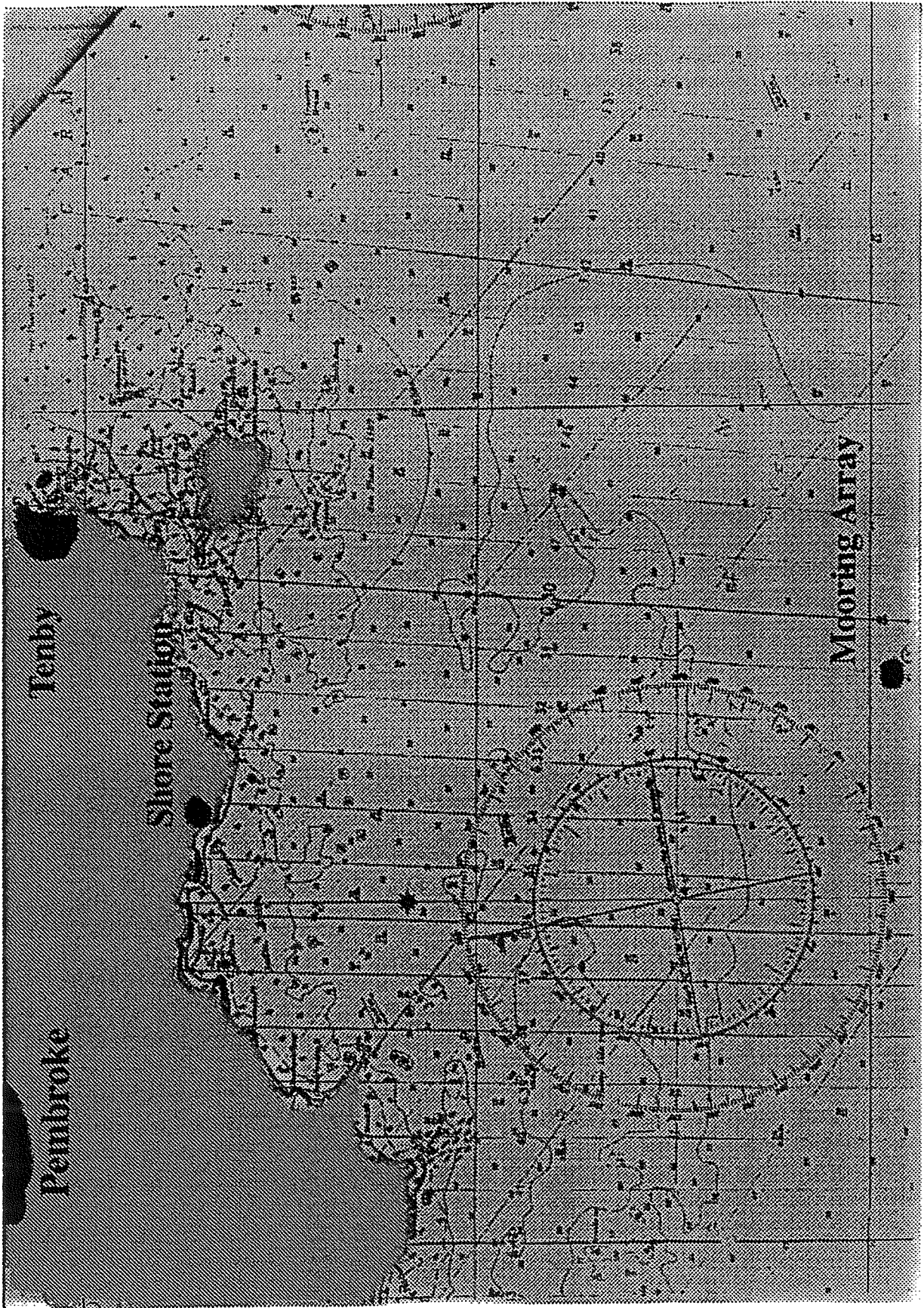
RF17
153MHz
500mW

RLC220
Radio
Modem

RS232 Data Messages

Thorcom Systems Limited		
Unit 4, 96B Blackpole Trading Estate West, Worcester, WR5 1PW, England, U.K. Tel: 0905 756700 Fax: 0905 755777		
Title Institute of Oceanographic Sciences		
Size	Document Number	REV
A4	Data logger link	1
Date:	July 23, 1993	Sheet 1 of 1

3.3. Map of the Area



4. DATA SOURCES AND PROCESSING

The application THORREAD.EXE was developed to convert .PDT and .UDT files produced by the THORCOM system into files of the standard binary FASTCOM format produced by the shipboard logging systems. If an error is encountered in the data, it deals with it in the manner described in the header of the source file listing THORREAD.C (see Appendix E.1). It also produces an ASCII error file, suffix .PDE or .UDE, listing the missing and bad blocks in the input file.

The program can be used under batch files TXPDR.BAT and TXUDR.BAT, suitably modified, to process a number of files with an appropriate wild card selection:

TXPDR.BAT

```
for %%f in (logdata\*.pdt) do thorread %%f
rem translates Tubby format to FASTCOM format and generates error files
rem alter path\file (in brackets) to use wildcards, etc., as required
```

TXUDR.BAT

```
for %%f in (logdata\*.udt) do thorread %%f
rem translates Tubby format to FASTCOM format and generates error files
rem alter path\file (in brackets) to use wildcards, etc., as required
```

Since the quantity of data collected is large and the requirements for using it are, as yet, undefined, the transfer to FASTCOM format files has only been carried out on a small number of files, for test purposes. However, there was an immediate interest in the reliability of the system, so the application THORSTAT.EXE was produced. This is a cut down version of THORREAD which has been used to analyse the .PDT and .UDT files for errors, producing an ASCII error file for each file processed.

Since a very large number of error files were produced, the application THORSUMM.EXE was developed to combine the errors listed in the error files produced by the above two applications into a tabular ASCII error statistics file.

The program can be used under batch files SUMPDE.BAT and SUMUDE.BAT, suitably modified, to process a number of files with an appropriate wild card selection:

SUMPDE.BAT

```
for %%f in (c:\data\*.pde) do thorsumm %%f
rem produces summary file of errors
rem alter path\file (in brackets) to use wildcards, etc., as required
```

SUMUDE.BAT

```
for %%f in (c:\data\*.ude) do thorsumm %%f
rem produces a summary file of errors
rem alter path\file (in brackets) to use wildcards, etc., as required
```

For test purposes and for eventual spectral processing, the application REPLAY2.EXE was produced; this is a form of the shipboard processing application FFTSETSW, used on "Warden". It is used to re-process a FASTCOM format file produced by THORREAD but, where bad data is encountered (as shown by a zero velocity of sound value), it substitutes the previous section's data accumulator values and flags the section of 1024 points of data so that it is not used.

The program can be used under batch files REPPDR.BAT and RERUDR.BAT, suitably modified, to process a number of files with an appropriate wild card selection:

REPPDR.BAT

```
for %%f in (c:\data\*.pdr) do replay2 12 %%f
rem displays time series graphically and does spectral processing as per FFTSETSW
rem alter path\file (in brackets) to use wildcards, etc., as required
```

REPUDR.BAT

```
for %%f in (c:\data\*.udr) do replay2 6 %%f
rem displays time series graphically and does spectral processing as per FFTSETSW
rem alter path\file (in brackets) to use wildcards, etc., as required
```

4.1. Data Quality Checking

Quality checking and a limited amount of error correction are carried out during the translation process, as described above. The net result of missing blocks of raw data is that the spectral analysis is performed on a lesser number of (1024 sample) sections, resulting in an increase in the confidence limits for the spectral estimates. Other schemes of processing could be implemented, such as the use of different length sections and the discarding of only the missing blocks, rather than the complete sections.

4.2. Summary of Data Produced

4.2.1. Raw Data Files

The complete list of data files collected during SWALES are in Appendix D. These are listed by Julian Day number, the listing details the location of the files on the Optical Disc which contains the complete SWALES telemetered data set.

The format of the data files is described in Appendix A. The data is written to disc in DOS format.

All the files have been analysed for errors by THORSTAT and THORSUM resulting in overall good data rate of 98%.

5. ACKNOWLEDGEMENTS

Without the advice and technical support of Thorcom this development would not have reached fruition, with particular thanks to Mike Tubby who directed the design of the system within Thorcom

Our thanks must also be extended to Mr Morgan at Hill Farm Manorbier for allowing the erection of our land station and aerial mast on his farm land.

The SWALES data set was the result of the concerted efforts of many, including the IOSDL Centre for Ocean Technology Development members of the Met Team, the IOSDL Moorings Team and the JRC members of the Met Team. The experimental work was funded by the MAFF Flood and Coastal Defence Division under commission FD0603; analysis of the data will be under commission FD0601.

6. APPENDIX A DATA FORMATS

6.1. Appendix A.1 Raw Data Files

The THORCOM receiving station produces files with suffices .PDT (prompted raw Sonic data) and .UDT (unprompted raw Sonic data); these share the same format.

The filenames have the form:

MMddhhmm.PDT

MMddhhmm.UDT

where MM = month (range 01 to 12)

dd = day (range 01 to 31)

hh = hour (range 00 to 23)

mm = minute (range 00 to 59)

Typical filelengths are 129000 bytes for .PDT files (approximately 10 minutes of raw data) and 68000 bytes for .UDT bytes (approximately 5 minutes of rawdata).

Each file consists of a number of blocks of received data, each preceded by a 1 byte block length, range 0 to 255, normally 202, 212 or 222; the block length does not include this byte.

The block of received data consists of

Sonic record number (2 bytes, binary, range 0 to 65535)

a number (normally 20, 21 or 22) of Sonic samples, each consisting of the 10 bytes:

uuvvwwcchh

where U = uu

V = vv

W = ww are 3 velocitycomponents each of 2 bytes (16 bit binary integers),

C = cc is a 2byte velocity of sound (16 bit binary integer),

H = hh is a 2byte compass reading (16 bit binary integer)

U, V and W normally have the range -6000 to +6000 for -60 m/s to +60 m/s, with a value of -10000 being used if there is a fault condition

C normally has the range 0 to +18500 for 0 m/s to 370 m/s, with a value of -10000 being used if there is a fault condition

H has the nominal range 2048 to 4088 for a compass output of 0 to 255 (0° to 358.6° clockwise relative to magnetic North)

6.2. Appendix A.2 FASTCOM Files

The standard FASTCOM file format consists of a header and a number of samples of data, i.e.

Header (44 bytes):

Mode<sp>1<LF>

Analog<sp>1<LF>

Time<sp>hh:mm:ss<sp>Date<sp>mm/dd/yy<LF>

a number of Sonic samples (about 12200 or 6400, depending upon whether the file was derived from a .PDT or a .UDT file, respectively), each consisting of the 10 bytes:

uuvvwwcchh

defined as in Appendix B.1, above.

6.3. Appendix A.3 Error Files

6.3.1. .PDE and .UDE files

The .PDE and .UDE files produced by either THORREAD or THORSTAT, are ASCII text files which begin with the line:

Start record no. sssss<LF>

- where sssss is the record number of the first block of data and <LF> is the line feed character (10).

In the event of errors being detected, this is followed by lines of the form:

Missing record at mmmmm<LF>

- where a non-sequential record number is detected, mmmmm being the expected record number.

and/or

Bad Block Length lll at rrrrr<LF>

- where lll is the block length minus 2 (the 2 record number bytes) and rrrrr is the record number.

Usually the block length is found to be one less than expected, i.e. 199, 209 or 219; this is not too serious, since it represents only the loss of the final compass reading. Bad length blocks are padded out to the nominal length with bytes of zero, the nominal length being defined as 210 for prompted data and 200 for unprompted data. Data with more than 2 bytes missing will be rejected by the application REPLAY2, since the padding will result in (one or more) zero velocity of sound values.

The length of a .PDE or .UDE file will, therefore, depend upon the number of errors encountered, with a good record resulting in a length of about 23 bytes and a defective record resulting in a length of at least 36 bytes.

6.3.2. SUMMARY.RDE and SUMMARY.NDE files

The SUMMARY.RDE and .NDE files produced by THORSUMM from .PDE and .UDE files, respectively, are ASCII tabular files which consist of a number of lines of the format:

```
ddd.ddddd<tab>missing<tab>bad<LF>
```

where ddd.ddddd is the (decimal) day number derived from the , e.g. 273.50694 for day 273 1210 hrs

missing is the number of missing blocks in the source file (.PDE or .UDE)

bad is the number of blocks with incorrect length (not a multiple of 20 bytes) in the source file (.PDE or .UDE)

<tab> is the tab character (9)

<LF> is the line feed character (10)

The day numbers will be in the order of processing and further use of the application THORSUMM will result in summary data being appended to the output files. The data can be sorted into sequential day number order by loading the table into CricketGraph and then using the Menu⇒DATA⇒SORT function.

7. APPENDIX B RADIO LICENCE

Department of Trade and Industry Radiocommunications Agency
Wireless Telegraphy Act 1949 Section 1



LIC. NO. TMP/33

DEPARTMENT OF TRADE AND INDUSTRY

RADIOCOMMUNICATIONS AGENCY

Wireless Telegraphy Act 1949 Section 1

TEMPORARY USE LICENCE

UNITED KINGDOM OF GREAT BRITAIN AND NORTHERN IRELAND,
THE CHANNEL ISLANDS AND THE ISLE OF MAN

ESTABLISHMENT

1 This Licence ("the Licence") granted under section 1(1) of the Wireless Telegraphy Act 1949 ("the 1949 Act") on 23 August 1993 ("the Date of Issue") by the Secretary of State for Trade and Industry ("the Secretary of State") to the Institute Of Oceanographic Sciences ("the Licensee") [whose company number is] authorises the Licensee as from 1 October 1993 ("the Commencement Date") to establish and use the stations at the locations set out in the Schedule to this Licence ("the Schedule") for wireless telegraphy, FOR THE PURPOSES SPECIFIED IN AND SUBJECT TO THE TERMS, PROVISIONS AND LIMITATIONS CONTAINED IN THIS LICENCE.

PURPOSE OF USE

2 The Licensee may establish and use sending and receiving stations for wireless telegraphy at the locations specified in the Schedule for the purposes specified hereinafter:

TERMS, PROVISIONS AND LIMITATIONS

LIMITATIONS ON USE

3 The Licensee shall operate the Stations in accordance with the requirements set out in the Schedule.

APPARATUS

4 the Licensee shall ensure that the apparatus comprised in the Stations ("the Apparatus") is so designed, constructed, maintained and used that it does not cause any undue interference with any wireless telegraphy apparatus or stations for wireless telegraphy.

USERS OF THE STATIONS

5 The Licensee shall not permit or suffer any person to use the Stations unless that person is:

- (a) under the control of the Licensee, or

Department of Trade and Industry Radiocommunications Agency
Wireless Telegraphy Act 1949 Section 1



- (b) authorised by the Licensee in writing to use the Stations.

6 The Licensee shall ensure that:

- (a) all persons authorised under Clause 5 above are made aware of the terms, provisions and limitations of this Licence; and
- (b) all such persons comply with the terms, provisions and limitations of this Licence.

CALL SIGN

7 During transmission, the Licensee shall transmit the call sign (if any) specified in the Schedule:-

- (a) at the beginning and at the end of each period of transmission and when the period of transmission is longer than 15 minutes, at the end of each interval of 15 minutes;
- (b) at the beginning of transmission on a new frequency (whenever the frequency of transmission is changed);
- (c) in the same format that is being used for the transmission of the information; and
- (d) on the same carrier frequency that is being used for the transmission.

NOTICE OF VARIATION OR REVOCATION

8 Where the Secretary of State exercises his power to revoke or vary this licence in accordance with section 1(4) of the 1949 Act, a written notice will be served on the Licensee.

INSPECTION

9 The Licensee shall permit a person authorised by the Secretary of State to:

- (a) have access to the Stations;
- (b) inspect the Licence; and
- (c) inspect and test the Apparatus,

at any reasonable time, or when, in the opinion of the Secretary of State, an urgent situation exists, at any time, for the purpose of verifying compliance with the terms, provisions and limitations of the Licence, or investigating a radio interference problem.

Department of Trade and Industry Radiocommunications Agency
Wireless Telegraphy Act 1949 Section 1



RESTRICTION, SUSPENSION OR CLOSEDOWN

10 When, in the opinion of the Secretary of State, or of a person authorised by him in that behalf:

- (a) the Licensee or a person authorised by the Licensee under Clause 5 above is in breach of the Licence and in the circumstances such breach justifies immediate restriction or closedown; or
- (b) exceptional circumstances beyond the control of the Licensee have arisen such that any Station or Apparatus, although not operated in breach of the Licence, is causing or contributing to or aggravating undue interference with other wireless telegraphy,

the Licensee shall restrict the operation of, or closedown and cease to operate, the Station, or any Apparatus, immediately, either permanently or for a specified temporary period, in accordance with the demand of a person authorised by the Secretary of State.

PERIOD OF VALIDITY OF LICENCE

11 This Licence shall be valid from the commencement date to midnight of 17 December 1993 unless revoked earlier by the Secretary of State.

12 Any Licence which the Secretary of State has previously granted to the Licensee under the 1949 Act for any of the Stations is hereby revoked.

INTERPRETATION

13 In this Licence, unless the context otherwise requires:

- (a) the Interpretation Act 1978 shall apply to this Licence as it applies to an Act of Parliament; and
- (b) "inspect" includes examine and test.

14 The headings are for ease of reference only and shall not affect the interpretation of this Licence.

15 The Schedule is incorporated in and forms part of this Licence.

SIGNED Mrs DeFreitas

Department of Trade and Industry Radiocommunications Agency
Wireless Telegraphy Act 1949 Section 1



SCHEDULE

RA Ref: TMP/33

Call sign (if applicable):

Location of Station: Carmarthen Bay

Purpose of Station: Transmittiion of environmental data from a moored platform to a shore based logger

Frequency: 153.3 MHz

Class of Emission: 16K0F1D

Maximum Power: -3 dBW

Antenna Characteristics: Co-Linear (omni directional)

Department of Trade and Industry Radiocommunications Agency
Wireless Telegraphy Act 1949 Section 1

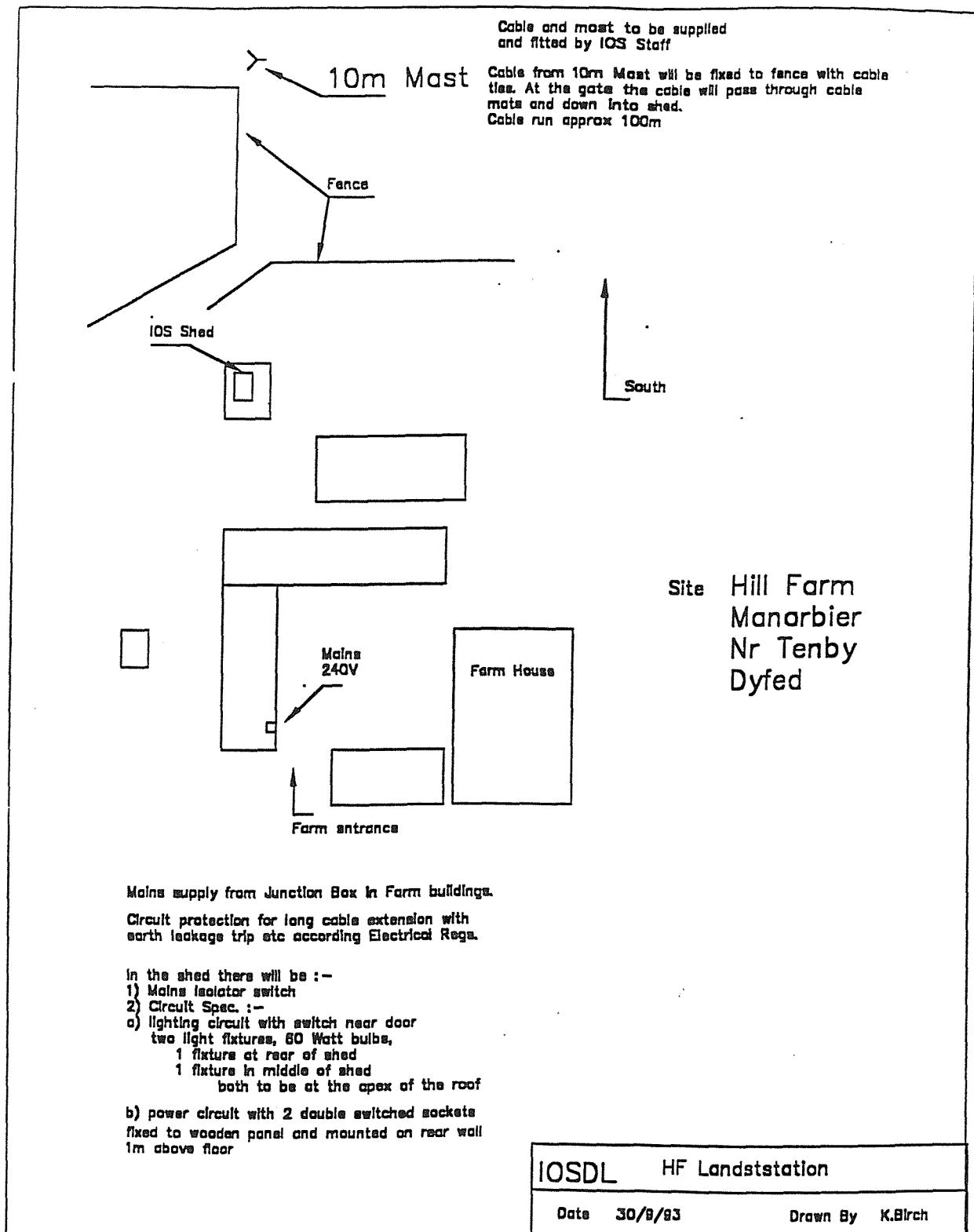


NOTES

- 1 In the event of a demand to close down or restrict the operation of any Station or Apparatus under clause 9 of the Licence, the Licensee must close down or restrict the operation of the Station or Apparatus immediately. The Licensee will be given oral reasons for the demand and will have an opportunity to provide reasons why the demand should not be met. If the demand is affirmed then it will be confirmed to the Licensee in writing as soon as practicable. If the Licensee does not comply with the demand or if the breach resulting in the demand is not rectified within a reasonable period of time to the satisfaction of the Secretary of State, then revocation or variation of Licence procedures may be commenced under section 1(4) of the Act or a prosecution may be initiated (depending on the circumstances of each case).
- 2 The Licensee must apply for a variation of the Licence from the Secretary of State before making any changes which may contravene the Licence.
- 3 The Licence is not transferable.
- 4 Radiocommunications Agency is an Executive Agency of the Department of Trade and Industry acting on behalf of the Secretary of State.

Tempuse.11c

8. APPENDIX C SHORE STATION SPECIFICATION



Cable supplied by IOS, and installed by contractor

Electrical fittings to be supplied by contractor

Mains supply from Junction Box in Farm buildings.

Circuit protection for long cable extension with earth leakage trip etc according Electrical Regs.

In the shed there will be :-

1) Mains isolator switch

2) Circuit Spec. :-

a) lighting circuit with switch near door

two light fixtures, 60 Watt bulbs,

1 fixture at rear of shed

1 fixture in middle of shed

both to be at the apex of the roof

b) power circuit with 2 double switched sockets

fixed to wooden panel and mounted on rear wall

1m above floor

IOSDL	Landstation
Date 1/10/83	Drawn by K.Birch

9. APPENDIX D RAW DATA FILES

SONIC RAW DATA FROM VHF LANDSTATION

Volume in drive D is SWALES HUT
Volume Serial Number is 2629-12CB
Directory of D:\HUTDATA

MO DRIVE SIDE A BUOY DEPLOYMENT 1
JDAY 279

10060500.PDT	10060510.UDT	10060515.PDT	10060525.UDT	10060530.PDT
10060540.UDT	10060545.PDT	10060555.UDT	10060600.PDT	10060610.UDT
10060615.PDT	10060625.UDT	10060630.PDT	10060640.UDT	10060655.UDT

JDAY 294

10211430.UDT	10211432.PDT	10211442.UDT	10211447.PDT	10211457.UDT
10211502.PDT	10211512.UDT	10211517.PDT	10211527.UDT	10211532.PDT
10211542.UDT	10211547.PDT	10211557.UDT	10211602.PDT	10211612.UDT
10211617.PDT	10211627.UDT	10211632.PDT	10211642.UDT	10211647.PDT
10211657.UDT	10211702.PDT	10211712.UDT	10211717.PDT	10211727.UDT
10211732.PDT	10211742.UDT	10211747.PDT	10211757.UDT	10211802.PDT
10211812.UDT	10211817.PDT	10211827.UDT	10211832.PDT	10211842.UDT
10211847.PDT	10211857.UDT	10211902.PDT	10211912.UDT	10211917.PDT
10211927.UDT	10211932.PDT	10211942.UDT	10211947.PDT	10211957.UDT
10212002.PDT	10212012.UDT	10212017.PDT	10212027.UDT	10212032.PDT
10212042.UDT	10212047.PDT	10212057.UDT	10212102.PDT	10212112.UDT
10212117.PDT	10212127.UDT	10212132.PDT	10212142.UDT	10212147.PDT
10212157.UDT	10212202.PDT	10212212.UDT	10212217.PDT	10212227.UDT
10212232.PDT	10212242.UDT	10212302.PDT	10212312.UDT	10212317.PDT
10212327.UDT	10212332.PDT	10212342.UDT	10212348.PDT	10212358.UDT

JDAY 295

10220003.PDT	10220013.UDT	10220018.PDT	10220028.UDT	10220033.PDT
10220043.UDT	10220048.PDT	10220058.UDT	10220103.PDT	10220113.UDT
10220118.PDT	10220128.UDT	10220133.PDT	10220143.UDT	10220148.PDT
10220158.UDT	10220203.PDT	10220213.UDT	10220218.PDT	10220228.UDT
10220233.PDT	10220243.UDT	10220248.PDT	10220258.UDT	10220303.PDT
10220313.UDT	10220318.PDT	10220328.UDT	10220333.PDT	10220343.UDT
10220348.PDT	10220358.UDT	10220403.PDT	10220413.UDT	10220418.PDT
10220428.UDT	10220433.PDT	10220443.UDT	10220448.PDT	10220458.UDT
10220503.PDT	10220513.UDT	10220518.PDT	10220528.UDT	10220533.PDT
10220543.UDT	10220548.PDT	10220558.UDT	10220603.PDT	10220613.UDT
10220618.PDT	10220628.UDT	10220633.PDT	10220643.UDT	10220648.PDT
10220658.UDT	10220703.PDT	10220713.UDT	10220718.PDT	10220728.UDT
10220733.PDT	10220743.UDT	10220748.PDT	10220758.UDT	10220803.PDT
10220813.UDT	10220818.PDT	10220828.UDT	10220833.PDT	10220843.UDT
10220848.PDT	10220858.UDT	10220903.PDT	10220913.UDT	10220918.PDT
10220928.UDT	10220933.PDT	10220943.UDT	10220948.PDT	10220958.UDT
10221003.PDT	10221026.UDT	10221029.PDT	10221039.UDT	10221044.PDT
10221054.UDT	10221059.PDT	10221109.UDT	10221114.PDT	10221124.UDT
10221129.PDT	10221139.UDT	10221144.PDT	10221154.UDT	10221159.PDT
10221209.UDT	10221214.PDT	10221224.UDT	10221229.PDT	10221239.UDT
10221244.PDT	10221254.UDT	10221259.PDT	10221309.UDT	10221314.PDT
10221324.UDT	10221329.PDT	10221339.UDT	10221344.PDT	10221354.UDT
10221359.PDT	10221409.UDT	10221414.PDT	10221424.UDT	10221429.PDT
10221439.UDT	10221444.PDT	10221454.UDT	10221459.PDT	10221509.UDT
10221514.PDT	10221524.UDT	10221529.PDT	10221539.UDT	10221544.PDT
10221554.UDT	10221559.PDT	10221609.UDT	10221614.PDT	10221624.UDT
10221629.PDT	10221639.UDT	10221644.PDT	10221654.UDT	10221659.PDT
10221709.UDT	10221714.PDT	10221724.UDT	10221729.PDT	10221739.UDT

10221744.PDT	10221754.UDT	10221759.PDT	10221809.UDT	10221814.PDT
10221824.UDT	10221829.PDT	10221839.UDT	10221844.PDT	10221854.UDT
10221859.PDT	10221909.UDT	10221914.PDT	10221924.UDT	10221929.PDT
10221939.UDT	10221944.PDT	10221954.UDT	10221959.PDT	10222009.UDT
10222014.PDT	10222024.UDT	10222029.PDT	10222039.UDT	10222044.PDT
10222054.UDT	10222059.PDT	10222109.UDT	10222114.PDT	10222124.UDT
10222129.PDT	10222139.UDT	10222144.PDT	10222154.UDT	10222159.PDT
10222209.UDT	10222214.PDT	10222224.UDT	10222229.PDT	10222239.UDT
10222244.PDT	10222254.UDT	10222259.PDT	10222309.UDT	10222314.PDT
10222324.UDT	10222329.PDT	10222339.UDT	10222344.PDT	10222354.UDT

JDAY 296

10230000.PDT	10230009.UDT	10230015.PDT	10230024.UDT	10230030.PDT
10230039.UDT	10230045.PDT	10230054.UDT	10230100.PDT	10230109.UDT
10230115.PDT	10230124.UDT	10230130.PDT	10230139.UDT	10230145.PDT
10230154.UDT	10230200.PDT	10230209.UDT	10230215.PDT	10230224.UDT
10230230.PDT	10230239.UDT	10230245.PDT	10230254.UDT	10230300.PDT
10230309.UDT	10230315.PDT	10230324.UDT	10230330.PDT	10230339.UDT
10230345.PDT	10230354.UDT	10230400.PDT	10230409.UDT	10230415.PDT
10230424.UDT	10230430.PDT	10230439.UDT	10230445.PDT	10230454.UDT
10230500.PDT	10230509.UDT	10230515.PDT	10230524.UDT	10230530.PDT
10230539.UDT	10230545.PDT	10230554.UDT	10230559.PDT	10230609.UDT
10230614.PDT	10230624.UDT	10230630.PDT	10230639.UDT	10230644.PDT
10230654.UDT	10230659.PDT	10230709.UDT	10230715.PDT	10230724.UDT
10230730.PDT	10230739.UDT	10230744.PDT	10230754.UDT	10230800.PDT
10230809.UDT	10230814.PDT	10230824.UDT	10230830.PDT	10230839.UDT
10230845.PDT	10230854.UDT	10230900.PDT	10230909.UDT	10230914.PDT
10230924.UDT	10230929.PDT	10230939.UDT	10230945.PDT	10230954.UDT
10231000.PDT	10231009.UDT	10231014.PDT	10231024.UDT	10231029.PDT
10231039.UDT	10231044.PDT	10231054.UDT	10231100.PDT	10231109.UDT
10231114.PDT	10231124.UDT	10231129.PDT	10231139.UDT	10231144.PDT
10231154.UDT	10231159.PDT	10231209.UDT	10231214.PDT	10231224.UDT
10231229.PDT	10231239.UDT	10231244.PDT	10231254.UDT	10231259.PDT
10231309.UDT	10231314.PDT	10231324.UDT	10231329.PDT	10231339.UDT
10231344.PDT	10231354.UDT	10231359.PDT	10231409.UDT	10231414.PDT
10231424.UDT	10231429.PDT	10231439.UDT	10231444.PDT	10231454.UDT
10231459.PDT	10231509.UDT	10231514.PDT	10231524.UDT	10231529.PDT
10231539.UDT	10231544.PDT	10231554.UDT	10231559.PDT	10231609.UDT
10231614.PDT	10231624.UDT	10231629.PDT	10231639.UDT	10231644.PDT
10231654.UDT	10231659.PDT	10231709.UDT	10231714.PDT	10231724.UDT
10231729.PDT	10231739.UDT	10231744.PDT	10231754.UDT	10231759.PDT
10231809.UDT	10231814.PDT	10231824.UDT	10231829.PDT	10231839.UDT
10231844.PDT	10231854.UDT	10231859.PDT	10231909.UDT	10231914.PDT
10231924.UDT	10231929.PDT	10231939.UDT	10231944.PDT	10231954.UDT
10231959.PDT	10232009.UDT	10232014.PDT	10232024.UDT	10232029.PDT
10232039.UDT	10232044.PDT	10232054.UDT	10232059.PDT	10232109.UDT
10232114.PDT	10232124.UDT	10232129.PDT	10232139.UDT	10232144.PDT
10232154.UDT	10232159.PDT	10232209.UDT	10232214.PDT	10232224.UDT
10232229.PDT	10232239.UDT	10232244.PDT	10232254.UDT	10232259.PDT
10232309.UDT	10232314.PDT	10232324.UDT	10232329.PDT	10232339.UDT
10232344.PDT	10232354.UDT	10240000.PDT	10240010.UDT	10240015.PDT
10240025.UDT	10240030.PDT	10240040.UDT	10240045.PDT	10240055.UDT

JDAY 297

10240100.PDT	10240110.UDT	10240115.PDT	10240125.UDT	10240130.PDT
10240140.UDT	10240145.PDT	10240155.UDT	10240200.PDT	10240210.UDT
10240215.PDT	10240225.UDT	10240230.PDT	10240240.UDT	10240245.PDT
10240255.UDT	10240300.PDT	10240310.UDT	10240315.PDT	10240325.UDT
10240330.PDT	10240340.UDT	10240345.PDT	10240355.UDT	10240400.PDT
10240410.UDT	10240415.PDT	10240425.UDT	10240430.PDT	10240440.UDT

10240445.PDT	10240455.UDT	10240500.PDT	10240510.UDT	10240515.PDT
10240525.UDT	10240530.PDT	10240540.UDT	10240545.PDT	10240555.UDT
10240600.PDT	10240610.UDT	10240615.PDT	10240625.UDT	10240630.PDT
10240640.UDT	10240645.PDT	10240655.UDT	10240700.PDT	10240710.UDT
10240715.PDT	10240725.UDT	10240730.PDT	10240740.UDT	10240745.PDT
10240755.UDT	10240800.PDT	10240810.UDT	10240815.PDT	10240825.UDT
10240830.PDT	10240840.UDT	10240845.PDT	10240855.UDT	10240900.PDT
10240910.UDT	10240915.PDT	10240925.UDT	10240930.PDT	10240940.UDT
10240945.PDT	10240955.UDT	10241000.PDT	10241010.UDT	10241015.PDT
10241025.UDT	10241030.PDT	10241040.UDT	10241045.PDT	10241055.UDT
10241100.PDT	10241110.UDT	10241115.PDT	10241125.UDT	10241130.PDT
10241140.UDT	10241145.PDT	10241155.UDT	10241200.PDT	10241210.UDT
10241215.PDT	10241225.UDT	10241230.PDT	10241240.UDT	10241245.PDT
10241255.UDT	10241300.PDT	10241310.UDT	10241315.PDT	10241325.UDT
10241330.PDT	10241340.UDT	10241345.PDT	10241355.UDT	10241400.PDT
10241410.UDT	10241415.PDT	10241425.UDT	10241430.PDT	10241440.UDT
10241445.PDT	10241455.UDT	10241500.PDT	10241510.UDT	10241515.PDT
10241525.UDT	10241530.PDT	10241540.UDT	10241545.PDT	10241555.UDT
10241600.PDT	10241610.UDT	10241615.PDT	10241625.UDT	10241630.PDT
10241640.UDT	10241645.PDT	10241655.UDT	10241700.PDT	10241710.UDT
10241715.PDT	10241725.UDT	10241730.PDT	10241740.UDT	10241745.PDT
10241755.UDT	10241800.PDT	10241810.UDT	10241815.PDT	10241825.UDT
10241830.PDT	10241840.UDT	10241845.PDT	10241855.UDT	10241900.PDT
10241910.UDT	10241915.PDT	10241925.UDT	10241930.PDT	10241940.UDT
10241945.PDT	10241955.UDT	10242000.PDT	10242010.UDT	10242015.PDT
10242025.UDT	10242030.PDT	10242040.UDT	10242045.PDT	10242055.UDT
10242100.PDT	10242110.UDT	10242115.PDT	10242125.UDT	10242130.PDT
10242140.UDT	10242145.PDT	10242155.UDT	10242200.PDT	10242210.UDT
10242215.PDT	10242225.UDT	10242230.PDT	10242240.UDT	10242245.PDT
10242255.UDT	10242300.PDT	10242310.UDT	10242315.PDT	10242325.UDT
10242330.PDT	10242340.UDT	10242345.PDT	10242355.UDT	10250000.PDT
JDAY 298				
10250010.UDT	10250015.PDT	10250025.UDT	10250030.PDT	10250040.UDT
10250045.PDT	10250055.UDT	10250100.PDT	10250110.UDT	10250115.PDT
10250125.UDT	10250130.PDT	10250140.UDT	10250145.PDT	10250155.UDT
10250200.PDT	10250210.UDT	10250215.PDT	10250225.UDT	10250230.PDT
10250240.UDT	10250245.PDT	10250255.UDT	10250300.PDT	10250310.UDT
10250315.PDT	10250325.UDT	10250330.PDT	10250340.UDT	10250345.PDT
10250355.UDT	10250400.PDT	10250410.UDT	10250415.PDT	10250425.UDT
10250430.PDT	10250440.UDT	10250445.PDT	10250455.UDT	10250500.PDT
10250510.UDT	10250515.PDT	10250525.UDT	10250530.PDT	10250540.UDT
10250545.PDT	10250555.UDT	10250600.PDT	10250610.UDT	10250615.PDT
10250625.UDT	10250630.PDT	10250640.UDT	10250645.PDT	10250655.UDT
10250700.PDT	10250710.UDT	10250715.PDT	10250725.UDT	10250730.PDT
10250740.UDT	10250745.PDT	10250755.UDT	10250800.PDT	10250810.UDT
10250815.PDT	10250825.UDT	10250830.PDT	10250840.UDT	10250845.PDT
10250855.UDT	10250900.PDT	10250910.UDT	10250915.PDT	10250925.UDT
10250930.PDT	10250940.UDT	10250945.PDT	10250955.UDT	10251000.PDT
10251010.UDT	10251015.PDT	10251025.UDT	10251030.PDT	10251040.UDT
10251045.PDT	10251055.UDT	10251100.PDT	10251110.UDT	10251115.PDT
10251125.UDT	10251130.PDT	10251140.UDT	10251145.PDT	10251155.UDT
10251200.PDT	10251210.UDT	10251215.PDT	10251225.UDT	10251230.PDT
10251240.UDT	10251245.PDT	10251255.UDT	10251300.PDT	10251310.UDT
10251315.PDT	10251325.UDT	10251330.PDT	10251340.UDT	10251345.PDT
10251355.UDT	10251400.PDT	10251410.UDT	10251415.PDT	10251425.UDT
10251430.PDT	10251440.UDT	10251445.PDT	10251455.UDT	10251500.PDT
10251510.UDT	10251515.PDT	10251525.UDT	10251530.PDT	10251540.UDT
10251545.PDT	10251555.UDT	10251600.PDT	10251610.UDT	10251615.PDT

10251625.UDT	10251630.PDT	10251640.UDT	10251645.PDT	10251655.UDT
10251700.PDT	10251710.UDT	10251715.PDT	10251725.UDT	10251730.PDT
10251740.UDT	10251745.PDT	10251755.UDT	10251800.PDT	10251810.UDT
10251815.PDT	10251825.UDT	10251830.PDT	10251840.UDT	10251845.PDT
10251855.UDT	10251900.PDT	10251910.UDT	10251915.PDT	10251925.UDT
10251930.PDT	10251940.UDT	10251945.PDT	10251955.UDT	10252000.PDT
10252010.UDT	10252015.PDT	10252025.UDT	10252030.PDT	10252040.UDT
10252045.PDT	10252055.UDT	10252100.PDT	10252110.UDT	10252115.PDT
10252125.UDT	10252130.PDT	10252140.UDT	10252145.PDT	10252155.UDT
10252215.PDT	10252225.UDT	10252230.PDT	10252240.UDT	10252245.PDT
10252255.UDT	10252300.PDT	10252310.UDT	10252315.PDT	10252325.UDT
10252330.PDT	10252340.UDT	10252345.PDT	10252355.UDT	

JDAY 299

				10260001.PDT
10260011.UDT	10260016.PDT	10260026.UDT	10260031.PDT	10260041.UDT
10260046.PDT	10260056.UDT	10260101.PDT	10260111.UDT	10260116.PDT
10260126.UDT	10260131.PDT	10260141.UDT	10260146.PDT	10260156.UDT
10260201.PDT	10260211.UDT	10260216.PDT	10260226.UDT	10260231.PDT
10260241.UDT	10260246.PDT	10260256.UDT	10260301.PDT	10260311.UDT
10260316.PDT	10260326.UDT	10260331.PDT	10260341.UDT	10260346.PDT
10260356.UDT	10260401.PDT	10260411.UDT	10260416.PDT	10260426.UDT
10260431.PDT	10260441.UDT	10260446.PDT	10260456.UDT	10260501.PDT
10260511.UDT	10260516.PDT	10260526.UDT	10260531.PDT	10260541.UDT
10260546.PDT	10260556.UDT	10260601.PDT	10260611.UDT	10260616.PDT
10260626.UDT	10260631.PDT	10260641.UDT	10260646.PDT	10260656.UDT
10260701.PDT	10260711.UDT	10260716.PDT	10260726.UDT	10260731.PDT
10260741.UDT	10260746.PDT	10260756.UDT	10260801.PDT	10260811.UDT
10260816.PDT	10260826.UDT	10260831.PDT	10260841.UDT	10260846.PDT
10260856.UDT	10260901.PDT	10260911.UDT	10260916.PDT	10260926.UDT
10260931.PDT	10260941.UDT	10260946.PDT	10260956.UDT	10261001.PDT
10261011.UDT	10261016.PDT	10261026.UDT	10261031.PDT	10261041.UDT
10261046.PDT	10261056.UDT	10261101.PDT	10261111.UDT	10261116.PDT
10261126.UDT	10261131.PDT	10261141.UDT	10261146.PDT	10261156.UDT
10261201.PDT	10261211.UDT	10261216.PDT	10261226.UDT	10261231.PDT
10261241.UDT	10261246.PDT	10261256.UDT	10261301.PDT	10261311.UDT
10261316.PDT	10261326.UDT	10261331.PDT	10261341.UDT	10261346.PDT
10261356.UDT	10261401.PDT	10261411.UDT	10261416.PDT	10261426.UDT
10261431.PDT	10261441.UDT	10261446.PDT	10261456.UDT	10261501.PDT
10261511.UDT	10261516.PDT	10261526.UDT	10261531.PDT	10261541.UDT
10261546.PDT	10261556.UDT	10261601.PDT	10261611.UDT	10261616.PDT
10261626.UDT	10261631.PDT	10261641.UDT	10261646.PDT	10261656.UDT
10261701.PDT	10261711.UDT	10261716.PDT	10261726.UDT	10261731.PDT
10261741.UDT	10261746.PDT	10261756.UDT	10261801.PDT	10261811.UDT
10261816.PDT	10261826.UDT	10261831.PDT	10261841.UDT	10261846.PDT
10261856.UDT	10261901.PDT	10261911.UDT	10261916.PDT	10261926.UDT
10261931.PDT	10261941.UDT	10261946.PDT	10261956.UDT	10262001.PDT
10262011.UDT	10262016.PDT	10262026.UDT	10262031.PDT	10262041.UDT
10262046.PDT	10262056.UDT	10262101.PDT	10262111.UDT	10262116.PDT
10262126.UDT	10262131.PDT	10262141.UDT	10262146.PDT	10262156.UDT
10262201.PDT	10262211.UDT	10262216.PDT	10262226.UDT	10262231.PDT
10262241.UDT	10262246.PDT	10262256.UDT	10262301.PDT	10262311.UDT
10262316.PDT	10262326.UDT	10262331.PDT	10262341.UDT	10262346.PDT
10262356.UDT				

JDAY 300

	10270001.PDT	10270011.UDT	10270016.PDT	10270026.UDT
10270031.PDT	10270041.UDT	10270046.PDT	10270056.UDT	10270101.PDT
10270111.UDT	10270116.PDT	10270126.UDT	10270131.PDT	10270141.UDT
10270146.PDT	10270156.UDT	10270201.PDT	10270211.UDT	10270216.PDT

10270226.UDT	10270231.PDT	10270241.UDT	10270246.PDT	10270256.UDT
10270301.PDT	10270311.UDT	10270316.PDT	10270326.UDT	10270331.PDT
10270341.UDT	10270346.PDT	10270356.UDT	10270401.PDT	10270411.UDT
10270416.PDT	10270426.UDT	10270431.PDT	10270441.UDT	10270446.PDT
10270456.UDT	10270501.PDT	10270511.UDT	10270516.PDT	10270526.UDT
10270531.PDT	10270541.UDT	10270546.PDT	10270556.UDT	10270601.PDT
10270611.UDT	10270616.PDT	10270626.UDT	10270631.PDT	10270641.UDT
10270646.PDT	10270656.UDT	10270701.PDT	10270711.UDT	10270716.PDT
10270726.UDT	10270731.PDT	10270741.UDT	10270746.PDT	10270756.UDT
10270801.PDT	10270811.UDT	10270816.PDT	10270826.UDT	10270831.PDT
10270841.UDT	10270846.PDT	10270856.UDT	10270901.PDT	10270911.UDT
10270916.PDT	10270926.UDT	10270931.PDT	10270941.UDT	10270946.PDT
10270956.UDT	10271001.PDT	10271011.UDT	10271016.PDT	10271026.UDT
10271031.PDT	10271041.UDT	10271046.PDT	10271056.UDT	10271116.PDT
10271126.UDT	10271131.PDT	10271141.UDT	10271146.PDT	10271156.UDT
10271201.PDT	10271211.UDT	10271216.PDT	10271226.UDT	10271231.PDT
10271241.UDT	10271246.PDT	10271256.UDT	10271301.PDT	10271311.UDT
10271316.PDT	10271326.UDT	10271331.PDT	10271341.UDT	10271346.PDT
10271356.UDT	10271401.PDT	10271411.UDT	10271416.PDT	10271426.UDT
10271431.PDT	10271441.UDT	10271446.PDT	10271456.UDT	10271501.PDT
10271511.UDT	10271516.PDT	10271526.UDT	10271531.PDT	10271541.UDT
10271546.PDT	10271556.UDT	10271601.PDT	10271611.UDT	10271616.PDT
10271626.UDT	10271631.PDT	10271641.UDT	10271646.PDT	10271656.UDT
10271701.PDT	10271711.UDT	10271716.PDT	10271726.UDT	10271731.PDT
10271741.UDT	10271746.PDT	10271756.UDT	10271801.PDT	10271811.UDT
10271816.PDT	10271826.UDT	10271831.PDT	10271841.UDT	10271846.PDT
10271856.UDT	10271901.PDT	10271911.UDT	10271916.PDT	10271926.UDT
10271931.PDT	10271941.UDT	10271946.PDT	10271956.UDT	10272001.PDT
10272011.UDT	10272016.PDT	10272026.UDT	10272031.PDT	10272041.UDT
10272046.PDT	10272056.UDT	10272101.PDT	10272111.UDT	10272116.PDT
10272126.UDT	10272131.PDT	10272141.UDT	10272146.PDT	10272156.UDT
10272201.PDT	10272211.UDT	10272216.PDT	10272226.UDT	10272231.PDT
10272241.UDT	10272246.PDT	10272256.UDT	10272301.PDT	10272311.UDT
10272316.PDT	10272326.UDT	10272331.PDT	10272341.UDT	10272346.PDT
10272356.UDT				
JDAY 301	10280002.PDT	10280012.UDT	10280017.PDT	10280027.UDT
10280032.PDT	10280042.UDT	10280047.PDT	10280057.UDT	10280102.PDT
10280112.UDT	10280117.PDT	10280127.UDT	10280132.PDT	10280142.UDT
10280147.PDT	10280157.UDT	10280202.PDT	10280212.UDT	10280217.PDT
10280227.UDT	10280232.PDT	10280242.UDT	10280247.PDT	10280257.UDT
10280302.PDT	10280312.UDT	10280317.PDT	10280327.UDT	10280332.PDT
10280342.UDT	10280347.PDT	10280357.UDT	10280402.PDT	10280412.UDT
10280417.PDT	10280427.UDT	10280432.PDT	10280442.UDT	10280447.PDT
10280457.UDT	10280502.PDT	10280512.UDT	10280517.PDT	10280527.UDT
10280532.PDT	10280542.UDT	10280547.PDT	10280557.UDT	10280602.PDT
10280612.UDT	10280617.PDT	10280627.UDT	10280632.PDT	10280642.UDT
10280647.PDT	10280657.UDT	10280702.PDT	10280712.UDT	10280717.PDT
10280727.UDT	10280732.PDT	10280742.UDT	10280747.PDT	10280757.UDT
10280802.PDT	10280812.UDT	10280817.PDT	10280827.UDT	10280832.PDT
10280842.UDT	10280847.PDT	10280857.UDT	10280902.PDT	10280912.UDT
10280917.PDT	10280927.UDT	10280932.PDT	10280942.UDT	10280947.PDT
10280957.UDT	10281002.PDT	10281012.UDT	10281017.PDT	10281027.UDT
10281032.PDT	10281042.UDT	10281047.PDT	10281057.UDT	10281102.PDT
10281112.UDT	10281117.PDT	10281127.UDT	10281132.PDT	10281142.UDT
10281147.PDT	10281157.UDT	10281202.PDT	10281212.UDT	10281217.PDT
10281227.UDT	10281232.PDT	10281242.UDT	10281247.PDT	10281257.UDT
10281302.PDT	10281312.UDT	10281317.PDT	10281327.UDT	10281332.PDT

10281342.UDT	10281347.PDT	10281356.UDT	10281402.PDT	10281411.UDT
10281417.PDT	10281427.UDT	10281432.PDT	10281441.UDT	10281447.PDT
10281456.UDT	10281502.PDT	10281512.UDT	10281714.UDT	10281715.PDT
10281725.UDT	10281730.PDT	10281740.UDT	10281745.PDT	10281755.UDT
10281800.PDT	10281810.UDT	10281815.PDT	10281825.UDT	10281830.PDT
10281840.UDT	10281845.PDT	10281855.UDT	10281900.PDT	10281910.UDT
10281915.PDT	10281925.UDT	10281930.PDT	10281940.UDT	10281945.PDT
10281955.UDT	10282000.PDT	10282010.UDT	10282015.PDT	10282025.UDT
10282030.PDT	10282040.UDT	10282045.PDT	10282055.UDT	10282100.PDT
10282110.UDT	10282115.PDT	10282125.UDT	10282130.PDT	10282140.UDT
10282145.PDT	10282155.UDT	10282200.PDT	10282210.UDT	10282215.PDT
10282225.UDT	10282230.PDT	10282240.UDT	10282245.PDT	10282255.UDT
10282300.PDT	10282310.UDT	10282315.PDT	10282325.UDT	10282330.PDT
10282340.UDT	10282345.PDT	10282355.UDT		

JDAY 302

			10290001.PDT	10290011.UDT
10290016.PDT	10290026.UDT	10290031.PDT	10290041.UDT	10290046.PDT
10290056.UDT	10290101.PDT	10290111.UDT	10290116.PDT	10290126.UDT
10290131.PDT	10290141.UDT	10290146.PDT	10290156.UDT	10290201.PDT
10290211.UDT	10290216.PDT	10290226.UDT	10290231.PDT	10290241.UDT
10290246.PDT	10290256.UDT	10290301.PDT	10290311.UDT	10290316.PDT
10290326.UDT	10290331.PDT	10290341.UDT	10290346.PDT	10290356.UDT
10290401.PDT	10290411.UDT	10290416.PDT	10290426.UDT	10290431.PDT
10290441.UDT	10290446.PDT	10290456.UDT	10290501.PDT	10290511.UDT
10290516.PDT	10290526.UDT	10290531.PDT	10290541.UDT	10290546.PDT
10290556.UDT	10290601.PDT	10290611.UDT	10290616.PDT	10290626.UDT
10290631.PDT	10290641.UDT	10290646.PDT	10290656.UDT	10290701.PDT
10290711.UDT	10290716.PDT	10290726.UDT	10290731.PDT	10290741.UDT
10290746.PDT	10290756.UDT	10290801.PDT	10290811.UDT	10290816.PDT
10290826.UDT	10290831.PDT	10290841.UDT	10290846.PDT	10290856.UDT
10290901.PDT	10290911.UDT	10290916.PDT	10290926.UDT	10290931.PDT
10290941.UDT	10290946.PDT	10290956.UDT	10291001.PDT	10291011.UDT
10291016.PDT	10291026.UDT	10291031.PDT	10291041.UDT	10291046.PDT
10291056.UDT	10291101.PDT	10291111.UDT	10291116.PDT	10291126.UDT
10291131.PDT	10291141.UDT	10291146.PDT	10291156.UDT	10291201.PDT
10291211.UDT	10291216.PDT	10291226.UDT	10291231.PDT	10291241.UDT
10291246.PDT	10291256.UDT	10291301.PDT	10291311.UDT	10291316.PDT
10291326.UDT	10291331.PDT	10291341.UDT	10291346.PDT	10291356.UDT
10291401.PDT	10291411.UDT	10291416.PDT	10291426.UDT	10291431.PDT
10291441.UDT	10291446.PDT	10291456.UDT	10291501.PDT	10291511.UDT
10291516.PDT	10291526.UDT	10291531.PDT	10291541.UDT	10291546.PDT
10291556.UDT	10291601.PDT	10291611.UDT	10291616.PDT	10291626.UDT
10291631.PDT	10291641.UDT	10291646.PDT	10291656.UDT	10291701.PDT
10291711.UDT	10291716.PDT	10291726.UDT	10291731.PDT	10291741.UDT
10291746.PDT	10291756.UDT	10291801.PDT	10291811.UDT	10291816.PDT
10291826.UDT	10291831.PDT	10291841.UDT	10291846.PDT	10291856.UDT
10291901.PDT	10291911.UDT	10291916.PDT	10291926.UDT	10291931.PDT
10291941.UDT	10291946.PDT	10291956.UDT	10292001.PDT	10292011.UDT
10292016.PDT	10292026.UDT	10292031.PDT	10292041.UDT	10292046.PDT
10292056.UDT	10292101.PDT	10292111.UDT	10292116.PDT	10292126.UDT
10292131.PDT	10292141.UDT	10292146.PDT	10292156.UDT	10292201.PDT
10292211.UDT	10292216.PDT	10292226.UDT	10292231.PDT	10292241.UDT
10292246.PDT	10292256.UDT	10292301.PDT	10292311.UDT	10292316.PDT
10292326.UDT	10292331.PDT	10292341.UDT	10292346.PDT	10292356.UDT

JDAY 303

10300002.PDT	10300011.UDT	10300016.PDT	10300026.UDT	10300032.PDT
10300041.UDT	10300047.PDT	10300056.UDT	10300101.PDT	10300111.UDT
10300116.PDT	10300126.UDT	10300131.PDT	10300141.UDT	10300147.PDT

10300156.UDT	10300201.PDT	10300211.UDT	10300216.PDT	10300226.UDT
10300231.PDT	10300241.UDT	10300246.PDT	10300256.UDT	10300301.PDT
10300311.UDT	10300316.PDT	10300326.UDT	10300331.PDT	10300341.UDT
10300346.PDT	10300356.UDT	10300401.PDT	10300411.UDT	10300416.PDT
10300426.UDT	10300431.PDT	10300441.UDT	10300446.PDT	10300456.UDT
10300501.PDT	10300511.UDT	10300516.PDT	10300526.UDT	10300531.PDT
10300541.UDT	10300546.PDT	10300556.UDT	10300601.PDT	10300611.UDT
10300616.PDT	10300626.UDT	10300631.PDT	10300641.UDT	10300646.PDT
10300656.UDT	10300701.PDT	10300711.UDT	10300716.PDT	10300726.UDT
10300731.PDT	10300741.UDT	10300746.PDT	10300756.UDT	10300801.PDT
10300811.UDT	10300816.PDT	10300826.UDT	10300831.PDT	10300841.UDT
10300846.PDT	10300856.UDT	10300901.PDT	10300911.UDT	10300916.PDT
10300926.UDT	10300931.PDT	10300941.UDT	10300946.PDT	10300956.UDT
10301001.PDT	10301011.UDT	10301016.PDT	10301026.UDT	10301031.PDT
10301041.UDT	10301046.PDT	10301056.UDT	10301101.PDT	10301111.UDT
10301116.PDT	10301126.UDT	10301131.PDT	10301141.UDT	10301146.PDT
10301156.UDT	10301201.PDT	10301211.UDT	10301216.PDT	10301226.UDT
10301231.PDT	10301241.UDT	10301246.PDT	10301256.UDT	10301301.PDT
10301311.UDT	10301316.PDT	10301326.UDT	10301331.PDT	10301341.UDT
10301346.PDT	10301356.UDT	10301401.PDT	10301411.UDT	10301416.PDT
10301426.UDT	10301431.PDT	10301441.UDT	10301446.PDT	10301456.UDT
10301501.PDT	10301511.UDT	10301516.PDT	10301526.UDT	10301531.PDT
10301541.UDT	10301546.PDT	10301556.UDT	10301601.PDT	10301611.UDT
10301616.PDT	10301626.UDT	10301631.PDT	10301641.UDT	10301646.PDT
10301656.UDT	10301701.PDT	10301711.UDT	10301716.PDT	10301726.UDT
10301731.PDT	10301741.UDT	10301746.PDT	10301756.UDT	10301801.PDT
10301811.UDT	10301816.PDT	10301826.UDT	10301831.PDT	10301841.UDT
10301846.PDT	10301856.UDT	10301901.PDT	10301911.UDT	10301916.PDT
10301926.UDT	10301931.PDT	10301941.UDT	10301946.PDT	10301956.UDT
10302001.PDT	10302011.UDT	10302016.PDT	10302026.UDT	10302031.PDT
10302041.UDT	10302046.PDT	10302056.UDT	10302101.PDT	10302111.UDT
10302116.PDT	10302126.UDT	10302131.PDT	10302141.UDT	10302146.PDT
10302156.UDT	10302201.PDT	10302211.UDT	10302216.PDT	10302226.UDT
10302231.PDT	10302241.UDT	10302246.PDT	10302256.UDT	10302301.PDT
10302311.UDT	10302316.PDT	10302326.UDT	10302331.PDT	10302341.UDT
10302346.PDT	10302356.UDT			

JDAY 304

10310012.UDT	10310017.PDT	10310027.UDT	10310032.PDT	10310002.PDT
10310047.PDT	10310057.UDT	10310102.PDT	10310112.UDT	10310042.UDT
10310127.UDT	10310132.PDT	10310142.UDT	10310147.PDT	10310117.PDT
10310202.PDT	10310212.UDT	10310217.PDT	10310227.UDT	10310157.UDT
10310242.UDT	10310247.PDT	10310257.UDT	10310302.PDT	10310232.PDT
10310317.PDT	10310327.UDT	10310332.PDT	10310342.UDT	10310312.UDT
10310357.UDT	10310402.PDT	10310412.UDT	10310417.PDT	10310347.PDT
10310432.PDT	10310442.UDT	10310447.PDT	10310457.UDT	10310427.UDT
10310512.UDT	10310517.PDT	10310527.UDT	10310532.PDT	10310502.PDT
10310547.PDT	10310557.UDT	10310602.PDT	10310612.UDT	10310542.UDT
10310627.UDT	10310632.PDT	10310642.UDT	10310647.PDT	10310617.PDT
10310702.PDT	10310712.UDT	10310717.PDT	10310727.UDT	10310657.UDT
10310742.UDT	10310747.PDT	10310757.UDT	10310802.PDT	10310732.PDT
10310817.PDT	10310827.UDT	10310832.PDT	10310842.UDT	10310812.UDT
10310857.UDT	10310902.PDT	10310912.UDT	10310917.PDT	10310847.PDT
10310932.PDT	10310942.UDT	10310947.PDT	10310957.UDT	10310927.UDT
10311012.UDT	10311017.PDT	10311027.UDT	10311032.PDT	10311002.PDT
10311047.PDT	10311057.UDT	10311102.PDT	10311112.UDT	10311042.UDT
10311127.UDT	10311132.PDT	10311142.UDT	10311147.PDT	10311117.PDT
10311202.PDT	10311212.UDT	10311217.PDT	10311227.UDT	10311157.UDT
				10311232.PDT

10311242.UDT	10311247.PDT	10311257.UDT	10311302.PDT	10311312.UDT
10311317.PDT	10311327.UDT	10311332.PDT	10311342.UDT	10311347.PDT
10311357.UDT	10311402.PDT	10311412.UDT	10311417.PDT	10311427.UDT
10311432.PDT	10311442.UDT	10311447.PDT	10311457.UDT	10311502.PDT
10311512.UDT	10311517.PDT	10311527.UDT	10311532.PDT	10311542.UDT
10311547.PDT	10311557.UDT	10311602.PDT	10311612.UDT	10311617.PDT
10311627.UDT	10311632.PDT	10311642.UDT	10311647.PDT	10311657.UDT
10311702.PDT	10311712.UDT	10311717.PDT	10311727.UDT	10311732.PDT
10311742.UDT	10311747.PDT	10311757.UDT	10311802.PDT	10311812.UDT
10311817.PDT	10311827.UDT	10311832.PDT	10311842.UDT	10311847.PDT
10311857.UDT	10311902.PDT	10311912.UDT	10311917.PDT	10311927.UDT
10311932.PDT	10311942.UDT	10311947.PDT	10311957.UDT	10312002.PDT
10312012.UDT	10312017.PDT	10312027.UDT	10312032.PDT	10312042.UDT
10312047.PDT	10312057.UDT	10312102.PDT	10312112.UDT	10312117.PDT
10312127.UDT	10312132.PDT	10312142.UDT	10312147.PDT	10312157.UDT
10312202.PDT	10312212.UDT	10312217.PDT	10312227.UDT	10312232.PDT
10312242.UDT	10312247.PDT	10312257.UDT	10312302.PDT	10312312.UDT
10312317.PDT	10312327.UDT	10312332.PDT	10312342.UDT	10312347.PDT
10312357.UDT				

JDAY 305	11010002.PDT	11010012.UDT	11010017.PDT	11010027.UDT
11010032.PDT	11010042.UDT	11010047.PDT	11010057.UDT	11010102.PDT
11010112.UDT	11010117.PDT	11010127.UDT	11010132.PDT	11010142.UDT
11010147.PDT	11010157.UDT	11010202.PDT	11010212.UDT	11010217.PDT
11010227.UDT	11010232.PDT	11010242.UDT	11010247.PDT	11010257.UDT
11010302.PDT	11010312.UDT	11010317.PDT	11010327.UDT	11010332.PDT
11010342.UDT	11010347.PDT	11010357.UDT	11010402.PDT	11010412.UDT
11010417.PDT	11010427.UDT	11010432.PDT	11010442.UDT	11010447.PDT
11010457.UDT	11010502.PDT	11010512.UDT	11010517.PDT	11010527.UDT
11010532.PDT	11010542.UDT	11010547.PDT	11010557.UDT	11010602.PDT
11010612.UDT	11010617.PDT	11010627.UDT	11010632.PDT	11010642.UDT
11010647.PDT	11010657.UDT	11010702.PDT	11010712.UDT	11010717.PDT
11010727.UDT	11010732.PDT	11010742.UDT	11010747.PDT	11010757.UDT
11010802.PDT	11010812.UDT	11010817.PDT	11010827.UDT	11010832.PDT
11010842.UDT	11010847.PDT	11010857.UDT	11010902.PDT	11010912.UDT
11010917.PDT	11010927.UDT	11010932.PDT	11010942.UDT	11010947.PDT
11010957.UDT	11011002.PDT	11011012.UDT	11011017.PDT	11011027.UDT
11011032.PDT	11011042.UDT	11011047.PDT	11011057.UDT	11011102.PDT
11011112.UDT	11011117.PDT	11011127.UDT	11011132.PDT	11011142.UDT
11011147.PDT	11011157.UDT	11011202.PDT	11011212.UDT	11011217.PDT
11011227.UDT	11011232.PDT	11011242.UDT	11011247.PDT	11011257.UDT
11011302.PDT	11011312.UDT	11011317.PDT	11011327.UDT	11011332.PDT
11011342.UDT	11011347.PDT	11011357.UDT	11011402.PDT	11011412.UDT
11011417.PDT	11011427.UDT	11011432.PDT	11011442.UDT	11011447.PDT
11011457.UDT	11011502.PDT	11011512.UDT	11011517.PDT	11011527.UDT
11011532.PDT	11011542.UDT	11011547.PDT	11011557.UDT	11011602.PDT
11011612.UDT	11011617.PDT	11011627.UDT	11011632.PDT	11011642.UDT
11011647.PDT	11011657.UDT	11011702.PDT	11011712.UDT	11011717.PDT
11011727.UDT	11011732.PDT	11011742.UDT	11011747.PDT	11011757.UDT
11011802.PDT	11011812.UDT	11011817.PDT	11011827.UDT	11011832.PDT
11011842.UDT	11011847.PDT	11011857.UDT	11011902.PDT	11011912.UDT
11011917.PDT	11011927.UDT	11011932.PDT	11011942.UDT	11011947.PDT
11011957.UDT	11012002.PDT	11012012.UDT	11012017.PDT	11012027.UDT
11012032.PDT	11012042.UDT	11012047.PDT	11012057.UDT	11012117.PDT
11012127.UDT	11012132.PDT	11012142.UDT	11012147.PDT	11012157.UDT
11012202.PDT	11012212.UDT	11012217.PDT	11012227.UDT	11012232.PDT
11012242.UDT	11012247.PDT	11012257.UDT	11012302.PDT	11012312.UDT
11012317.PDT	11012327.UDT	11012332.PDT	11012342.UDT	11012347.PDT

11012357.UDT

JDAY 306	11020003.PDT	11020013.UDT	11020018.PDT	11020028.UDT
11020033.PDT	11020043.UDT	11020048.PDT	11020058.UDT	11020103.PDT
11020113.UDT	11020118.PDT	11020128.UDT	11020133.PDT	11020143.UDT
11020148.PDT	11020158.UDT	11020203.PDT	11020213.UDT	11020218.PDT
11020228.UDT	11020233.PDT	11020243.UDT	11020248.PDT	11020258.UDT
11020303.PDT	11020313.UDT	11020318.PDT	11020328.UDT	11020333.PDT
11020343.UDT	11020348.PDT	11020358.UDT	11020403.PDT	11020413.UDT
11020418.PDT	11020428.UDT	11020433.PDT	11020443.UDT	11020448.PDT
11020458.UDT	11020503.PDT	11020513.UDT	11020518.PDT	11020528.UDT
11020533.PDT	11020543.UDT	11020548.PDT	11020558.UDT	11020603.PDT
11020613.UDT	11020618.PDT	11020628.UDT	11020633.PDT	11020643.UDT
11020648.PDT	11020658.UDT	11020703.PDT	11020713.UDT	11020718.PDT
11020728.UDT	11020733.PDT	11020743.UDT	11020748.PDT	11020758.UDT
11020803.PDT	11020813.UDT	11020818.PDT	11020828.UDT	11020833.PDT
11020843.UDT	11020848.PDT	11020858.UDT	11020903.PDT	11020913.UDT
11020918.PDT	11020928.UDT	11020933.PDT	11020943.UDT	11020948.PDT
11020958.UDT	11021003.PDT	11021013.UDT	11021018.PDT	11021028.UDT
11021033.PDT	11021043.UDT	11021048.PDT	11021058.UDT	11021103.PDT
11021113.UDT	11021118.PDT	11021128.UDT	11021133.PDT	11021143.UDT
11021148.PDT	11021158.UDT	11021203.PDT	11021213.UDT	11021218.PDT
11021228.UDT	11021233.PDT	11021243.UDT	11021248.PDT	11021258.UDT
11021303.PDT	11021313.UDT	11021318.PDT	11021328.UDT	11021333.PDT
11021343.UDT	11021348.PDT	11021358.UDT	11021403.PDT	11021413.UDT
11021418.PDT	11021428.UDT	11021433.PDT	11021443.UDT	11021448.PDT
11021458.UDT	11021503.PDT	11021513.UDT	11021518.PDT	11021528.UDT
11021533.PDT	11021543.UDT	11021548.PDT	11021558.UDT	11021603.PDT
11021613.UDT	11021618.PDT	11021628.UDT	11021633.PDT	11021643.UDT
11021648.PDT	11021658.UDT	11021703.PDT	11021713.UDT	11021718.PDT
11021728.UDT	11021733.PDT	11021743.UDT	11021748.PDT	11021758.UDT
11021803.PDT	11021813.UDT	11021818.PDT	11021828.UDT	11021833.PDT
11021843.UDT	11021848.PDT	11021857.UDT	11021903.PDT	11021913.UDT
11021918.PDT	11021928.UDT	11021933.PDT	11021942.UDT	11021948.PDT
11021957.UDT	11022003.PDT	11022013.UDT	11022018.PDT	11022027.UDT
11022033.PDT	11022043.UDT	11022048.PDT	11022058.UDT	11022103.PDT
11022112.UDT	11022118.PDT	11022127.UDT	11022133.PDT	11022142.UDT
11022148.PDT	11022157.UDT	11022203.PDT	11022213.UDT	11022218.PDT
11022227.UDT	11022233.PDT	11022242.UDT	11022248.PDT	11022258.UDT
11022303.PDT	11022312.UDT	11022318.PDT	11022327.UDT	11022333.PDT
11022342.UDT	11022348.PDT	11022357.UDT		

JDAY 307

			11030003.PDT	11030013.UDT
11030018.PDT	11030028.UDT	11030033.PDT	11030043.UDT	11030048.PDT
11030058.UDT	11030103.PDT	11030113.UDT	11030118.PDT	11030128.UDT
11030133.PDT	11030143.UDT	11030148.PDT	11030158.UDT	11030203.PDT
11030213.UDT	11030218.PDT	11030228.UDT	11030233.PDT	11030243.UDT
11030248.PDT	11030258.UDT	11030303.PDT	11030313.UDT	11030318.PDT
11030328.UDT	11030333.PDT	11030343.UDT	11030348.PDT	11030358.UDT
11030403.PDT	11030413.UDT	11030418.PDT	11030428.UDT	11030433.PDT
11030443.UDT	11030448.PDT	11030458.UDT	11030503.PDT	11030513.UDT
11030518.PDT	11030528.UDT	11030533.PDT	11030543.UDT	11030548.PDT
11030558.UDT	11030603.PDT	11030613.UDT	11030618.PDT	11030628.UDT
11030633.PDT	11030643.UDT	11030648.PDT	11030658.UDT	11030703.PDT
11030713.UDT	11030718.PDT	11030728.UDT	11030733.PDT	11030743.UDT
11030748.PDT	11030758.UDT	11030803.PDT	11030813.UDT	11030818.PDT
11030828.UDT	11030833.PDT	11030843.UDT	11030848.PDT	11030858.UDT
11030903.PDT	11030913.UDT	11030918.PDT	11030928.UDT	11030933.PDT

11030943.UDT	11030948.PDT	11030958.UDT	11031003.PDT	11031013.UDT
11031018.PDT	11031028.UDT	11031033.PDT	11031043.UDT	11031048.PDT
11031058.UDT	11031103.PDT	11031113.UDT	11031118.PDT	11031128.UDT
11031133.PDT	11031143.UDT	11031148.PDT	11031158.UDT	11031203.PDT
11031213.UDT	11031218.PDT	11031228.UDT	11031233.PDT	11031243.UDT
11031248.PDT	11031258.UDT	11031303.PDT	11031313.UDT	11031318.PDT
11031328.UDT	11031333.PDT	11031343.UDT	11031348.PDT	11031358.UDT
11031403.PDT	11031413.UDT	11031418.PDT	11031428.UDT	11031433.PDT
11031443.UDT	11031448.PDT	11031458.UDT	11031503.PDT	11031513.UDT
11031518.PDT	11031528.UDT	11031533.PDT	11031543.UDT	11031548.PDT
11031558.UDT	11031603.PDT	11031613.UDT	11031618.PDT	11031628.UDT
11031633.PDT	11031643.UDT	11031648.PDT	11031658.UDT	11031703.PDT
11031713.UDT	11031718.PDT	11031728.UDT	11031733.PDT	11031743.UDT
11031748.PDT	11031758.UDT	11031803.PDT	11031813.UDT	11031818.PDT
11031828.UDT	11031833.PDT	11031843.UDT	11031848.PDT	11031858.UDT
11031903.PDT	11031913.UDT	11031918.PDT	11031928.UDT	11031933.PDT
11031943.UDT	11031948.PDT	11031958.UDT	11032003.PDT	11032013.UDT
11032018.PDT	11032028.UDT	11032033.PDT	11032043.UDT	11032048.PDT
11032058.UDT	11032103.PDT	11032113.UDT	11032118.PDT	11032128.UDT
11032133.PDT	11032143.UDT	11032148.PDT	11032158.UDT	11032203.PDT
11032213.UDT	11032218.PDT	11032228.UDT	11032233.PDT	11032243.UDT
11032248.PDT	11032258.UDT	11032303.PDT	11032313.UDT	11032318.PDT
11032328.UDT	11032333.PDT	11032343.UDT	11032348.PDT	11032358.UDT
JDAY 308				
11040004.PDT	11040014.UDT	11040019.PDT	11040029.UDT	11040034.PDT
11040044.UDT	11040049.PDT	11040059.UDT	11040104.PDT	11040114.UDT
11040119.PDT	11040129.UDT	11040134.PDT	11040144.UDT	11040149.PDT
11040159.UDT	11040204.PDT	11040214.UDT	11040219.PDT	11040229.UDT
11040234.PDT	11040244.UDT	11040249.PDT	11040259.UDT	11040304.PDT
11040314.UDT	11040319.PDT	11040329.UDT	11040334.PDT	11040344.UDT
11040349.PDT	11040358.UDT	11040404.PDT	11040414.UDT	11040419.PDT
11040428.UDT	11040434.PDT	11040443.UDT	11040449.PDT	11040458.UDT
11040504.PDT	11040513.UDT	11040519.PDT	11040528.UDT	11040534.PDT
11040544.UDT	11040549.PDT	11040559.UDT	11040604.PDT	11040613.UDT
11040619.PDT	11040628.UDT	11040634.PDT	11040643.UDT	11040649.PDT
11040658.UDT	11040704.PDT	11040714.UDT	11040719.PDT	11040729.UDT
11040734.PDT	11040743.UDT	11040749.PDT	11040758.UDT	11040804.PDT
11040813.UDT	11040819.PDT	11040828.UDT	11040834.PDT	11040844.UDT
11040849.PDT	11040858.UDT	11040904.PDT	11040914.UDT	11040919.PDT
11040928.UDT	11040934.PDT	11040943.UDT	11040949.PDT	11040958.UDT
11041004.PDT	11041013.UDT	11041019.PDT	11041028.UDT	11041034.PDT
11041043.UDT	11041049.PDT	11041058.UDT	11041104.PDT	11041113.UDT
11041119.PDT	11041128.UDT	11041134.PDT	11041143.UDT	11041149.PDT
11041158.UDT	11041204.PDT	11041213.UDT	11041219.PDT	11041228.UDT
11041234.PDT	11041243.UDT	11041249.PDT	11041258.UDT	11041304.PDT
11041313.UDT	11041319.PDT	11041328.UDT	11041334.PDT	11041343.UDT
11041349.PDT	11041358.UDT	11041404.PDT	11041413.UDT	11041419.PDT
11041428.UDT	11041434.PDT	11041443.UDT	11041449.PDT	11041458.UDT
11041504.PDT	11041513.UDT	11041519.PDT	11041528.UDT	11041534.PDT
11041543.UDT	11041549.PDT	11041558.UDT	11041604.PDT	11041613.UDT
11041619.PDT	11041628.UDT	11041634.PDT	11041643.UDT	11041649.PDT
11041658.UDT	11041704.PDT	11041713.UDT	11041719.PDT	11041728.UDT
11041734.PDT	11041743.UDT	11041749.PDT	11041758.UDT	11041804.PDT
11041813.UDT	11041819.PDT	11041828.UDT	11041834.PDT	11041843.UDT
11041849.PDT	11041858.UDT	11041904.PDT	11041913.UDT	11041919.PDT
11041928.UDT	11041934.PDT	11041943.UDT	11041949.PDT	11041958.UDT
11042004.PDT	11042013.UDT	11042019.PDT	11042028.UDT	11042034.PDT
11042043.UDT	11042049.PDT	11042058.UDT	11042104.PDT	11042113.UDT

```

11042119.PDT  11042128.UDT  11042134.PDT  11042143.UDT  11042149.PDT
11042158.UDT  11042204.PDT  11042213.UDT  11042218.PDT  11042228.UDT
11042234.PDT  11042243.UDT  11042249.PDT  11042258.UDT  11042304.PDT
11042313.UDT  11042318.PDT  11042328.UDT  11042334.PDT  11042343.UDT
11042349.PDT  11042358.UDT
[.]           [..]
      2757 file(s)  258898846 bytes
                  25952256 bytes free

```

SONIC BUOY RAW DATA FROM THE VHF LANDSTATION

Volume in drive D is SWALES HUT
Volume Serial Number is 1F4B-12D6
Directory of D:\HUTDATA
MO DRIVE SIDE B

BUOY DEPLOYMENT 1 CONT.

```

JDAY 309
11050019.PDT  11050029.UDT  11050034.PDT  11050044.UDT  11050014.UDT
11050059.UDT  11050104.PDT  11050114.UDT  11050119.PDT  11050129.UDT
11050134.PDT  11050144.UDT  11050149.PDT  11050159.UDT  11050204.PDT
11050214.UDT  11050219.PDT  11050229.UDT  11050234.PDT  11050244.UDT
11050249.PDT  11050259.UDT  11050304.PDT  11050314.UDT  11050319.PDT
11050329.UDT  11050334.PDT  11050344.UDT  11050349.PDT  11050359.UDT
11050404.PDT  11050414.UDT  11050419.PDT  11050429.UDT  11050434.PDT
11050444.UDT  11050449.PDT  11050459.UDT  11050504.PDT  11050514.UDT
11050519.PDT  11050529.UDT  11050534.PDT  11050544.UDT  11050549.PDT
11050559.UDT  11050604.PDT  11050614.UDT  11050619.PDT  11050629.UDT
11050634.PDT  11050644.UDT  11050649.PDT  11050659.UDT  11050704.PDT
11050714.UDT  11050719.PDT  11050729.UDT  11050734.PDT  11050744.UDT
11050749.PDT  11050759.UDT  11050804.PDT  11050814.UDT  11050819.PDT
11050829.UDT  11050834.PDT  11050844.UDT  11050849.PDT  11050859.UDT
11050904.PDT  11050914.UDT  11050919.PDT  11050929.UDT  11051150.UDT
11051151.PDT  11051253.PDT  11051256.UDT  11051301.PDT  11051311.UDT
11051316.PDT  11051326.UDT  11051331.PDT  11051341.UDT  11051346.PDT
11051356.UDT  11051401.PDT  11051411.UDT  11051416.PDT  11051426.UDT
11051431.PDT  11051441.UDT  11051446.PDT  11051456.UDT  11051501.PDT
11051511.UDT  11051516.PDT  11051526.UDT  11051531.PDT  11051541.UDT
11051546.PDT  11051556.UDT  11051601.PDT  11051611.UDT  11051616.PDT
11051626.UDT  11051631.PDT  11051641.UDT  11051646.PDT  11051656.UDT
11051701.PDT  11051711.UDT  11051716.PDT  11051726.UDT  11051731.PDT
11051741.UDT  11051746.PDT  11051756.UDT  11051801.PDT  11051811.UDT
11051816.PDT  11051826.UDT  11051831.PDT  11051841.UDT  11051846.PDT
11051856.UDT  11051901.PDT  11051911.UDT  11051916.PDT  11051926.UDT
11051931.PDT  11051941.UDT  11051946.PDT  11051956.UDT  11052001.PDT
11052011.UDT  11052016.PDT  11052026.UDT  11052031.PDT  11052041.UDT
11052046.PDT  11052056.UDT  11052101.PDT  11052111.UDT  11052116.PDT
11052126.UDT  11052131.PDT  11052141.UDT  11052146.PDT  11052156.UDT
11052201.PDT  11052211.UDT  11052216.PDT  11052226.UDT  11052231.PDT
11052241.UDT  11052246.PDT  11052256.UDT  11052301.PDT  11052311.UDT
11052316.PDT  11052326.UDT  11052331.PDT  11052341.UDT  11052346.PDT
11052356.UDT

```

```

JDAY 310
11060032.PDT  11060042.UDT  11060047.PDT  11060057.UDT  11060102.PDT
11060112.UDT  11060117.PDT  11060127.UDT  11060132.PDT  11060142.UDT
11060147.PDT  11060157.UDT  11060202.PDT  11060212.UDT  11060217.PDT
11060227.UDT  11060232.PDT  11060242.UDT  11060247.PDT  11060257.UDT

```

11060302.PDT	11060312.UDT	11060317.PDT	11060327.UDT	11060332.PDT
11060342.UDT	11060347.PDT	11060357.UDT	11060402.PDT	11060412.UDT
11060417.PDT	11060427.UDT	11060432.PDT	11060442.UDT	11060447.PDT
11060457.UDT	11060502.PDT	11060512.UDT	11060517.PDT	11060527.UDT
11060532.PDT	11060542.UDT	11060547.PDT	11060557.UDT	11060602.PDT
11060612.UDT	11060617.PDT	11060627.UDT	11060632.PDT	11060642.UDT
11060647.PDT	11060657.UDT	11060702.PDT	11060712.UDT	11060717.PDT
11060727.UDT	11060732.PDT	11060742.UDT	11060747.PDT	11060757.UDT
11060802.PDT	11060812.UDT	11060817.PDT	11060827.UDT	11060832.PDT
11060842.UDT	11060847.PDT	11060857.UDT	11060902.PDT	11060912.UDT
11060917.PDT	11060927.UDT	11060932.PDT	11060942.UDT	11060947.PDT
11060957.UDT	11061002.PDT	11061012.UDT	11061017.PDT	11061027.UDT
11061032.PDT	11061042.UDT	11061047.PDT	11061057.UDT	11061102.PDT
11061112.UDT	11061117.PDT	11061127.UDT	11061132.PDT	11061142.UDT
11061147.PDT	11061157.UDT	11061202.PDT	11061212.UDT	11061217.PDT
11061227.UDT	11061232.PDT	11061242.UDT	11061247.PDT	11061257.UDT
11061302.PDT	11061312.UDT	11061317.PDT	11061327.UDT	11061332.PDT
11061342.UDT	11061347.PDT	11061357.UDT	11061402.PDT	11061412.UDT
11061417.PDT	11061427.UDT	11061432.PDT	11061442.UDT	11061447.PDT
11061457.UDT	11061502.PDT	11061512.UDT	11061517.PDT	11061527.UDT
11061532.PDT	11061542.UDT	11061547.PDT	11061557.UDT	11061602.PDT
11061612.UDT	11061617.PDT	11061627.UDT	11061632.PDT	11061642.UDT
11061647.PDT	11061657.UDT	11061702.PDT	11061712.UDT	11061717.PDT
11061727.UDT	11061732.PDT	11061742.UDT	11061747.PDT	11061757.UDT
11061802.PDT	11061812.UDT	11061817.PDT	11061827.UDT	11061832.PDT
11061842.UDT	11061847.PDT	11061857.UDT	11061902.PDT	11061912.UDT
11061917.PDT	11061927.UDT	11061932.PDT	11061942.UDT	11061947.PDT
11061957.UDT	11062002.PDT	11062012.UDT	11062017.PDT	11062027.UDT
11062032.PDT	11062042.UDT	11062047.PDT	11062057.UDT	11062102.PDT
11062112.UDT	11062117.PDT	11062127.UDT	11062132.PDT	11062142.UDT
11062147.PDT	11062157.UDT	11062202.PDT	11062212.UDT	11062217.PDT
11062227.UDT	11062232.PDT	11062242.UDT	11062247.PDT	11062257.UDT
11062302.PDT	11062312.UDT	11062317.PDT	11062327.UDT	11062332.PDT
11062342.UDT	11062347.PDT	11062357.UDT		

JDAY 311

			11070002.PDT	11070012.UDT
11070017.PDT	11070027.UDT	11070032.PDT	11070042.UDT	11070047.PDT
11070057.UDT	11070102.PDT	11070112.UDT	11070117.PDT	11070127.UDT
11070132.PDT	11070142.UDT	11070147.PDT	11070157.UDT	11070202.PDT
11070212.UDT	11070217.PDT	11070227.UDT	11070232.PDT	11070242.UDT
11070247.PDT	11070257.UDT	11070302.PDT	11070312.UDT	11070317.PDT
11070327.UDT	11070332.PDT	11070342.UDT	11070347.PDT	11070357.UDT
11070402.PDT	11070412.UDT	11070417.PDT	11070427.UDT	11070432.PDT
11070442.UDT	11070447.PDT	11070457.UDT	11070502.PDT	11070512.UDT
11070517.PDT	11070527.UDT	11070532.PDT	11070542.UDT	11070547.PDT
11070557.UDT	11070602.PDT	11070612.UDT	11070617.PDT	11070627.UDT
11070632.PDT	11070642.UDT	11070647.PDT	11070657.UDT	11070702.PDT
11070712.UDT	11070717.PDT	11070727.UDT	11070732.PDT	11070742.UDT
11070747.PDT	11070757.UDT	11070802.PDT	11070812.UDT	11070817.PDT
11070827.UDT	11070832.PDT	11070842.UDT	11070847.PDT	11070857.UDT
11070902.PDT	11070912.UDT	11070917.PDT	11070927.UDT	11070932.PDT
11070942.UDT	11070947.PDT	11070957.UDT	11071002.PDT	11071012.UDT
11071017.PDT	11071027.UDT	11071032.PDT	11071042.UDT	11071047.PDT
11071057.UDT	11071102.PDT	11071112.UDT	11071117.PDT	11071127.UDT
11071132.PDT	11071142.UDT	11071147.PDT	11071157.UDT	11071202.PDT
11071212.UDT	11071217.PDT	11071227.UDT	11071232.PDT	11071242.UDT
11071247.PDT	11071257.UDT	11071302.PDT	11071312.UDT	11071317.PDT
11071327.UDT	11071404.PDT	11071412.UDT	11071417.PDT	11071427.UDT

11071432.PDT	11071442.UDT	11071447.PDT	11071457.UDT	11071502.PDT
11071510.PDT	11071512.UDT	11071558.UDT	11071601.PDT	11071611.UDT
11071616.PDT	11071626.UDT	11071631.PDT	11071641.UDT	11071646.PDT
11071656.UDT	11071701.PDT	11071711.UDT	11071716.PDT	11071726.UDT
11071731.PDT	11071741.UDT	11071746.PDT	11071756.UDT	11071801.PDT
11071811.UDT	11071816.PDT	11071826.UDT	11071831.PDT	11071841.UDT
11071846.PDT	11071856.UDT	11071901.PDT	11071911.UDT	11071916.PDT
11071926.UDT	11071931.PDT	11071941.UDT	11071946.PDT	11071956.UDT
11072001.PDT	11072011.UDT	11072016.PDT	11072026.UDT	11072031.PDT
11072041.UDT	11072046.PDT	11072056.UDT	11072101.PDT	11072111.UDT
11072116.PDT	11072126.UDT	11072131.PDT	11072141.UDT	11072146.PDT
11072156.UDT	11072201.PDT	11072211.UDT	11072216.PDT	11072226.UDT
11072231.PDT	11072241.UDT	11072246.PDT	11072256.UDT	11072301.PDT
11072311.UDT	11072316.PDT	11072326.UDT	11072331.PDT	11072341.UDT
11072346.PDT	11072356.UDT			

JDAY 312

		11080002.PDT	11080012.UDT	11080017.PDT
11080027.UDT	11080032.PDT	11080042.UDT	11080047.PDT	11080057.UDT
11080102.PDT	11080112.UDT	11080117.PDT	11080127.UDT	11080132.PDT
11080142.UDT	11080147.PDT	11080157.UDT	11080202.PDT	11080212.UDT
11080217.PDT	11080227.UDT	11080232.PDT	11080242.UDT	11080247.PDT
11080257.UDT	11080302.PDT	11080312.UDT	11080317.PDT	11080327.UDT
11080332.PDT	11080342.UDT	11080347.PDT	11080356.UDT	11080402.PDT
11080411.UDT	11080417.PDT	11080426.UDT	11080432.PDT	11080441.UDT
11080447.PDT	11080457.UDT	11080502.PDT	11080511.UDT	11080517.PDT
11080527.UDT	11080532.PDT	11080542.UDT	11080547.PDT	11080557.UDT
11080602.PDT	11080612.UDT	11080617.PDT	11080627.UDT	11080632.PDT
11080642.UDT	11080647.PDT	11080656.UDT	11080702.PDT	11080712.UDT
11080717.PDT	11080726.UDT	11080732.PDT	11080742.UDT	11080747.PDT
11080757.UDT	11080802.PDT	11080812.UDT	11080817.PDT	11080827.UDT
11080832.PDT	11080842.UDT	11080847.PDT	11080856.UDT	11080902.PDT
11080912.UDT	11080917.PDT	11080927.UDT	11080932.PDT	11080942.UDT
11080947.PDT	11080956.UDT	11081002.PDT	11081011.UDT	11081017.PDT
11081026.UDT	11081032.PDT	11081041.UDT	11081047.PDT	11081056.UDT
11081102.PDT	11081111.UDT	11081117.PDT	11081126.UDT	11081132.PDT
11081141.UDT	11081147.PDT	11081156.UDT	11081202.PDT	11081211.UDT
11081217.PDT	11081226.UDT	11081232.PDT	11081241.UDT	11081247.PDT
11081256.UDT	11081302.PDT	11081311.UDT	11081317.PDT	11081326.UDT
11081332.PDT	11081341.UDT	11081347.PDT	11081356.UDT	11081402.PDT
11081411.UDT	11081417.PDT	11081426.UDT	11081432.PDT	11081441.UDT
11081447.PDT	11081456.UDT	11081502.PDT	11081511.UDT	11081517.PDT
11081526.UDT	11081532.PDT	11081541.UDT	11081547.PDT	11081556.UDT
11081602.PDT	11081611.UDT	11081617.PDT	11081626.UDT	11081632.PDT
11081641.UDT	11081647.PDT	11081656.UDT	11081702.PDT	11081711.UDT
11081717.PDT	11081726.UDT	11081732.PDT	11081741.UDT	11081747.PDT
11081756.UDT	11081802.PDT	11081811.UDT	11081817.PDT	11081826.UDT
11081832.PDT	11081841.UDT	11081847.PDT	11081856.UDT	11081902.PDT
11081911.UDT	11081917.PDT	11081926.UDT	11081932.PDT	11081941.UDT
11081947.PDT	11081956.UDT	11082002.PDT	11082011.UDT	11082017.PDT
11082026.UDT	11082032.PDT	11082041.UDT	11082047.PDT	11082056.UDT
11082102.PDT	11082111.UDT	11082117.PDT	11082126.UDT	11082132.PDT
11082141.UDT	11082147.PDT	11082156.UDT	11082202.PDT	11082211.UDT
11082217.PDT	11082226.UDT	11082231.PDT	11082241.UDT	11082246.PDT
11082256.UDT	11082302.PDT	11082311.UDT	11082317.PDT	11082326.UDT
11082332.PDT	11082341.UDT	11082346.PDT	11082356.UDT	

JDAY 313

				11090002.PDT
11090012.UDT	11090017.PDT	11090027.UDT	11090032.PDT	11090042.UDT

11090047.PDT	11090057.UDT	11090102.PDT	11090112.UDT	11090117.PDT
11090127.UDT	11090132.PDT	11090142.UDT	11090147.PDT	11090157.UDT
11090202.PDT	11090212.UDT	11090217.PDT	11090227.UDT	11090232.PDT
11090242.UDT	11090247.PDT	11090257.UDT	11090302.PDT	11090312.UDT
11090317.PDT	11090327.UDT	11090332.PDT	11090342.UDT	11090347.PDT
11090357.UDT	11090402.PDT	11090412.UDT	11090417.PDT	11090427.UDT
11090432.PDT	11090442.UDT	11090447.PDT	11090457.UDT	11090502.PDT
11090512.UDT	11090517.PDT	11090527.UDT	11090532.PDT	11090542.UDT
11090547.PDT	11090557.UDT	11090602.PDT	11090612.UDT	11090617.PDT
11090627.UDT	11090632.PDT	11090642.UDT	11090647.PDT	11090657.UDT
11090702.PDT	11090712.UDT	11090717.PDT	11090727.UDT	11090732.PDT
11090742.UDT	11090747.PDT	11090757.UDT	11090802.PDT	11090812.UDT
11090817.PDT	11090827.UDT	11090832.PDT	11090842.UDT	11090847.PDT
11090857.UDT	11090902.PDT	11090912.UDT	11090917.PDT	11090927.UDT
11090932.PDT	11090942.UDT	11090947.PDT	11090957.UDT	11091002.PDT
11091012.UDT	11091017.PDT	11091027.UDT	11091032.PDT	11091042.UDT
11091047.PDT	11091057.UDT	11091102.PDT	11091112.UDT	11091117.PDT
11091127.UDT	11091132.PDT	11091142.UDT	11091147.PDT	11091157.UDT
11091202.PDT	11091212.UDT	11091217.PDT	11091227.UDT	11091232.PDT
11091242.UDT	11091247.PDT	11091257.UDT	11091302.PDT	11091312.UDT
11091317.PDT	11091327.UDT	11091332.PDT	11091342.UDT	11091347.PDT
11091357.UDT	11091402.PDT			

BUOY DEPLOYMENT 2

JDAY 327		11230924.PDT	11230930.UDT	11230935.PDT
11230944.UDT	11230950.PDT	11230959.UDT	11231005.PDT	11231014.UDT
11231020.PDT	11231030.UDT	11231035.PDT	11231045.UDT	11231050.PDT
11231059.UDT	11231106.PDT	11231114.PDT	11231116.UDT	11231121.PDT
11231131.UDT	11231136.PDT	11231146.UDT	11231151.PDT	11231201.UDT
11231206.PDT	11231216.UDT	11231221.PDT	11231231.UDT	11231236.PDT
11231246.UDT	11231251.PDT	11231301.UDT	11231306.PDT	11231316.UDT
11231321.PDT	11231331.UDT	11231336.PDT	11231346.UDT	11231357.PDT
11231407.UDT	11231412.PDT	11231422.UDT	11231427.PDT	11231437.UDT
11231442.PDT	11231452.UDT	11231457.PDT	11231507.UDT	11231512.PDT
11231522.UDT	11231527.PDT	11231537.UDT	11231542.PDT	11231552.UDT
11231557.PDT	11231607.UDT	11231612.PDT	11231622.UDT	11231627.PDT
11231637.UDT	11231642.PDT	11231652.UDT	11231657.PDT	11231707.UDT
11231712.PDT	11231722.UDT	11231727.PDT	11231737.UDT	11231742.PDT
11231752.UDT	11231757.PDT	11231807.UDT	11231812.PDT	11231822.UDT
11231827.PDT	11231837.UDT	11231842.PDT	11231852.UDT	11231857.PDT
11231907.UDT	11231912.PDT	11231922.UDT	11231927.PDT	11231937.UDT
11231942.PDT	11231952.UDT	11231957.PDT	11232007.UDT	11232012.PDT
11232022.UDT	11232027.PDT	11232037.UDT	11232042.PDT	11232052.UDT
11232057.PDT	11232107.UDT	11232112.PDT	11232122.UDT	11232127.PDT
11232137.UDT	11232142.PDT	11232152.UDT	11232157.PDT	11232207.UDT
11232212.PDT	11232222.UDT	11232227.PDT	11232237.UDT	11232242.PDT
11232252.UDT	11232257.PDT	11232307.UDT	11232312.PDT	11232322.UDT
11232327.PDT	11232337.UDT	11232342.PDT	11232352.UDT	11232357.PDT
JDAY 328				
11240007.UDT	11240012.PDT	11240022.UDT	11240027.PDT	11240037.UDT
11240042.PDT	11240052.UDT	11240057.PDT	11240107.UDT	11240112.PDT
11240122.UDT	11240127.PDT	11240137.UDT	11240142.PDT	11240152.UDT
11240157.PDT	11240207.UDT	11240212.PDT	11240222.UDT	11240227.PDT
11240237.UDT	11240242.PDT	11240252.UDT	11240257.PDT	11240307.UDT
11240312.PDT	11240322.UDT	11240327.PDT	11240337.UDT	11240342.PDT
11240352.UDT	11240357.PDT	11240407.UDT	11240412.PDT	11240422.UDT
11240427.PDT	11240437.UDT	11240442.PDT	11240452.UDT	11240457.PDT

11240507.UDT	11240512.PDT	11240522.UDT	11240527.PDT	11240537.UDT
11240542.PDT	11240552.UDT	11240557.PDT	11240607.UDT	11240612.PDT
11240622.UDT	11240627.PDT	11240637.UDT	11240642.PDT	11240652.UDT
11240657.PDT	11240707.UDT	11240712.PDT	11240722.UDT	11240727.PDT
11240737.UDT	11240742.PDT	11240752.UDT	11240757.PDT	11240807.UDT
11240812.PDT	11240822.UDT	11240827.PDT	11240837.UDT	11240842.PDT
11240852.UDT	11240857.PDT	11240907.UDT	11240912.PDT	11240922.UDT
11240927.PDT	11240937.UDT	11240942.PDT	11240952.UDT	11241002.PDT
11241012.UDT	11241017.PDT	11241027.UDT	11241032.PDT	11241042.UDT
11241047.PDT	11241057.UDT	11241102.PDT	11241112.UDT	11241117.PDT
11241126.UDT	11241132.PDT	11241141.UDT	11241147.PDT	11241156.UDT
11241202.PDT	11241211.UDT	11241217.PDT	11241226.UDT	11241232.PDT
11241241.UDT	11241247.PDT	11241256.UDT	11241302.PDT	11241311.UDT
11241317.PDT	11241326.UDT	11241332.PDT	11241341.UDT	11241347.PDT
11241356.UDT	11241402.PDT	11241411.UDT	11241416.PDT	11241426.UDT
11241432.PDT	11241441.UDT	11241447.PDT	11241456.UDT	11241501.PDT
11241511.UDT	11241516.PDT	11241526.UDT	11241531.PDT	11241541.UDT
11241546.PDT	11241556.UDT	11241601.PDT	11241611.UDT	11241617.PDT
11241626.UDT	11241632.PDT	11241641.UDT	11241646.PDT	11241656.UDT
11241702.PDT	11241711.UDT	11241716.PDT	11241726.UDT	11241732.PDT
11241741.UDT	11241746.PDT	11241756.UDT	11241801.PDT	11241811.UDT
11241816.PDT	11241826.UDT	11241832.PDT	11241841.UDT	11241846.PDT
11241856.UDT	11241901.PDT	11241911.UDT	11241916.PDT	11241926.UDT
11241931.PDT	11241941.UDT	11241946.PDT	11241956.UDT	11242001.PDT
11242011.UDT	11242016.PDT	11242026.UDT	11242031.PDT	11242041.UDT
11242046.PDT	11242056.UDT	11242101.PDT	11242111.UDT	11242116.PDT
11242126.UDT	11242131.PDT	11242141.UDT	11242146.PDT	11242156.UDT
11242201.PDT	11242211.UDT	11242216.PDT	11242226.UDT	11242231.PDT
11242241.UDT	11242246.PDT	11242256.UDT	11242301.PDT	11242311.UDT
11242316.PDT	11242326.UDT	11242331.PDT	11242341.UDT	11242346.PDT
11242356.UDT				
JDAY 329	11250001.PDT	11250011.UDT	11250016.PDT	11250026.UDT
11250031.PDT	11250041.UDT	11250046.PDT	11250056.UDT	11250101.PDT
11250111.UDT	11250116.PDT	11250126.UDT	11250131.PDT	11250141.UDT
11250146.PDT	11250156.UDT	11250201.PDT	11250211.UDT	11250216.PDT
11250226.UDT	11250231.PDT	11250241.UDT	11250246.PDT	11250256.UDT
11250301.PDT	11250311.UDT	11250316.PDT	11250326.UDT	11250331.PDT
11250341.UDT	11250346.PDT	11250356.UDT	11250401.PDT	11250411.UDT
11250416.PDT	11250426.UDT	11250431.PDT	11250441.UDT	11250446.PDT
11250456.UDT	11250501.PDT	11250511.UDT	11250516.PDT	11250526.UDT
11250531.PDT	11250541.UDT	11250546.PDT	11250556.UDT	11250601.PDT
11250611.UDT	11250616.PDT	11250626.UDT	11250631.PDT	11250641.UDT
11250646.PDT	11250656.UDT	11250701.PDT	11250711.UDT	11250716.PDT
11250726.UDT	11250731.PDT	11250741.UDT	11250746.PDT	11250756.UDT
11250801.PDT	11250811.UDT	11250816.PDT	11250826.UDT	11250831.PDT
11250841.UDT	11250846.PDT	11250856.UDT	11250901.PDT	11250911.UDT
11250916.PDT	11250926.UDT	11250937.PDT	11250947.UDT	11250952.PDT
11251002.UDT	11251007.PDT	11251017.UDT	11251022.PDT	11251032.UDT
11251037.PDT	11251047.UDT	11251052.PDT	11251102.UDT	11251107.PDT
11251117.UDT	11251122.PDT	11251132.UDT	11251137.PDT	11251147.UDT
11251152.PDT	11251202.UDT	11251207.PDT	11251217.UDT	11251222.PDT
11251232.UDT	11251237.PDT	11251247.UDT	11251252.PDT	11251302.UDT
11251307.PDT	11251317.UDT	11251322.PDT	11251332.UDT	11251337.PDT
11251347.UDT	11251352.PDT	11251402.UDT	11251407.PDT	11251417.UDT
11251422.PDT	11251432.UDT	11251437.PDT	11251447.UDT	11251452.PDT
11251502.UDT	11251507.PDT	11251517.UDT	11251522.PDT	11251532.UDT
11251537.PDT	11251547.UDT	11251552.PDT	11251602.UDT	11251607.PDT

11251617.UDT	11251622.PDT	11251632.UDT	11251637.PDT	11251647.UDT
11251652.PDT	11251702.UDT	11251707.PDT	11251717.UDT	11251722.PDT
11251732.UDT	11251737.PDT	11251747.UDT	11251752.PDT	11251802.UDT
11251807.PDT	11251817.UDT	11251822.PDT	11251832.UDT	11251837.PDT
11251847.UDT	11251852.PDT	11251902.UDT	11251907.PDT	11251917.UDT
11251922.PDT	11251932.UDT	11251937.PDT	11251947.UDT	11251952.PDT
11252002.UDT	11252007.PDT	11252017.UDT	11252022.PDT	11252032.UDT
11252037.PDT	11252047.UDT	11252052.PDT	11252102.UDT	11252107.PDT
11252117.UDT	11252122.PDT	11252132.UDT	11252137.PDT	11252147.UDT
11252152.PDT	11252202.UDT	11252207.PDT	11252217.UDT	11252222.PDT
11252232.UDT	11252237.PDT	11252247.UDT	11252307.PDT	11252317.UDT
11252322.PDT	11252332.UDT	11252337.PDT	11252347.UDT	11252352.PDT

JDAY 330

11260002.UDT	11260007.PDT	11260017.UDT	11260022.PDT	11260032.UDT
11260037.PDT	11260047.UDT	11260052.PDT	11260102.UDT	11260107.PDT
11260117.UDT	11260122.PDT	11260132.UDT	11260137.PDT	11260147.UDT
11260152.PDT	11260202.UDT	11260207.PDT	11260217.UDT	11260222.PDT
11260232.UDT	11260237.PDT	11260247.UDT	11260252.PDT	11260302.UDT
11260307.PDT	11260317.UDT	11260322.PDT	11260332.UDT	11260337.PDT
11260347.UDT	11260352.PDT	11260402.UDT	11260407.PDT	11260417.UDT
11260422.PDT	11260432.UDT	11260437.PDT	11260447.UDT	11260452.PDT
11260502.UDT	11260507.PDT	11260517.UDT	11260522.PDT	11260532.UDT
11260537.PDT	11260547.UDT	11260552.PDT	11260602.UDT	11260607.PDT
11260617.UDT	11260622.PDT	11260632.UDT	11260637.PDT	11260647.UDT
11260652.PDT	11260702.UDT	11260707.PDT	11260717.UDT	11260722.PDT
11260732.UDT	11260737.PDT	11260747.UDT	11260752.PDT	11260802.UDT
11260807.PDT	11260817.UDT	11260822.PDT	11260832.UDT	11260837.PDT
11260847.UDT	11260852.PDT	11260902.UDT	11260907.PDT	11260917.UDT
11260922.PDT	11260932.UDT	11260937.PDT	11260947.UDT	11260952.PDT
11261002.UDT	11261007.PDT	11261017.UDT	11261022.PDT	11261032.UDT
11261037.PDT	11261047.UDT	11261052.PDT	11261102.UDT	11261107.PDT
11261117.UDT	11261122.PDT	11261132.UDT	11261137.PDT	11261147.UDT
11261152.PDT	11261202.UDT	11261207.PDT	11261217.UDT	11261222.PDT
11261232.UDT	11261237.PDT	11261247.UDT	11261252.PDT	11261302.UDT
11261307.PDT	11261317.UDT	11261322.PDT	11261332.UDT	11261337.PDT
11261347.UDT	11261352.PDT	11261402.UDT	11261407.PDT	11261417.UDT
11261422.PDT	11261432.UDT	11261437.PDT	11261447.UDT	11261452.PDT
11261502.UDT	11261507.PDT	11261517.UDT	11261522.PDT	11261532.UDT
11261537.PDT	11261547.UDT	11261552.PDT	11261602.UDT	11261607.PDT
11261617.UDT	11261622.PDT	11261632.UDT	11261637.PDT	11261647.UDT
11261652.PDT	11261702.UDT	11261707.PDT	11261717.UDT	11261722.PDT
11261732.UDT	11261737.PDT	11261747.UDT	11261752.PDT	11261802.UDT
11261808.PDT	11261818.UDT	11261823.PDT	11261833.UDT	11261838.PDT
11261848.UDT	11261853.PDT	11261903.UDT	11261908.PDT	11261918.UDT
11261923.PDT	11261933.UDT	11261938.PDT	11261948.UDT	11261953.PDT
11262003.UDT	11262008.PDT	11262018.UDT	11262023.PDT	11262033.UDT
11262038.PDT	11262048.UDT	11262053.PDT	11262103.UDT	11262108.PDT
11262118.UDT	11262123.PDT	11262133.UDT	11262138.PDT	11262148.UDT
11262153.PDT	11262203.UDT	11262208.PDT	11262218.UDT	11262223.PDT
11262233.UDT	11262238.PDT	11262248.UDT	11262253.PDT	11262303.UDT
11262308.PDT	11262318.UDT	11262323.PDT	11262333.UDT	11262338.PDT
11262348.UDT	11262353.PDT			

JDAY 331

		11270003.UDT	11270008.PDT	11270018.UDT
11270023.PDT	11270033.UDT	11270038.PDT	11270048.UDT	11270053.PDT
11270103.UDT	11270108.PDT	11270118.UDT	11270123.PDT	11270133.UDT
11270138.PDT	11270148.UDT	11270153.PDT	11270203.UDT	11270208.PDT
11270218.UDT	11270223.PDT	11270233.UDT	11270238.PDT	11270248.UDT

11270253.PDT	11270303.UDT	11270308.PDT	11270318.UDT	11270323.PDT
11270333.UDT	11270338.PDT	11270348.UDT	11270353.PDT	11270403.UDT
11270408.PDT	11270418.UDT	11270423.PDT	11270433.UDT	11270438.PDT
11270448.UDT	11270453.PDT	11270503.UDT	11270508.PDT	11270518.UDT
11270523.PDT	11270533.UDT	11270538.PDT	11270548.UDT	11270553.PDT
11270603.UDT	11270608.PDT	11270618.UDT	11270623.PDT	11270633.UDT
11270638.PDT	11270648.UDT	11270653.PDT	11270703.UDT	11270708.PDT
11270717.UDT	11270723.PDT	11270733.UDT	11270738.PDT	11270747.UDT
11270753.PDT	11270802.UDT	11270808.PDT	11270818.UDT	11270823.PDT
11270833.UDT	11270838.PDT	11270848.UDT	11270853.PDT	11270903.UDT
11270908.PDT	11270917.UDT	11270923.PDT	11270932.UDT	11270938.PDT
11270948.UDT	11270953.PDT	11271003.UDT	11271008.PDT	11271017.UDT
11271023.PDT	11271032.UDT	11271038.PDT	11271048.UDT	11271053.PDT
11271103.UDT	11271108.PDT	11271117.UDT	11271123.PDT	11271132.UDT
11271138.PDT	11271147.UDT	11271153.PDT	11271202.UDT	11271208.PDT
11271218.UDT	11271223.PDT	11271233.UDT	11271238.PDT	11271247.UDT
11271253.PDT	11271302.UDT	11271308.PDT	11271318.UDT	11271323.PDT
11271333.UDT	11271338.PDT	11271347.UDT	11271353.PDT	11271402.UDT
11271408.PDT	11271417.UDT	11271423.PDT	11271432.UDT	11271438.PDT
11271447.UDT	11271453.PDT	11271502.UDT	11271508.PDT	11271517.UDT
11271523.PDT	11271532.UDT	11271538.PDT	11271547.UDT	11271553.PDT
11271602.UDT	11271608.PDT	11271617.UDT	11271623.PDT	11271632.UDT
11271638.PDT	11271647.UDT	11271653.PDT	11271702.UDT	11271708.PDT
11271717.UDT	11271723.PDT	11271732.UDT	11271738.PDT	11271747.UDT
11271753.PDT	11271802.UDT	11271808.PDT	11271818.UDT	11271823.PDT
11271833.UDT	11271838.PDT	11271848.UDT	11271853.PDT	11271903.UDT
11271908.PDT	11271918.UDT	11271923.PDT	11271933.UDT	11271938.PDT
11271948.UDT	11271953.PDT	11272003.UDT	11272008.PDT	11272018.UDT
11272023.PDT	11272033.UDT	11272038.PDT	11272048.UDT	11272053.PDT
11272103.UDT	11272108.PDT	11272118.UDT	11272123.PDT	11272133.UDT
11272138.PDT	11272148.UDT	11272153.PDT	11272203.UDT	11272208.PDT
11272218.UDT	11272223.PDT	11272233.UDT	11272238.PDT	11272248.UDT
11272253.PDT	11272303.UDT	11272308.PDT	11272318.UDT	11272323.PDT
11272333.UDT	11272338.PDT	11272348.UDT	11272353.PDT	

JDAY 332

11280008.PDT	11280018.UDT	11280023.PDT	11280033.UDT	11280038.PDT
11280048.UDT	11280053.PDT	11280103.UDT	11280108.PDT	11280118.UDT
11280123.PDT	11280133.UDT	11280138.PDT	11280148.UDT	11280153.PDT
11280203.UDT	11280208.PDT	11280218.UDT	11280223.PDT	11280233.UDT
11280238.PDT	11280248.UDT	11280253.PDT	11280303.UDT	11280308.PDT
11280318.UDT	11280323.PDT	11280333.UDT	11280338.PDT	11280348.UDT
11280353.PDT	11280403.UDT	11280408.PDT	11280418.UDT	11280423.PDT
11280433.UDT	11280438.PDT	11280448.UDT	11280453.PDT	11280503.UDT
11280508.PDT	11280518.UDT	11280523.PDT	11280533.UDT	11280538.PDT
11280548.UDT	11280553.PDT	11280603.UDT	11280608.PDT	11280618.UDT
11280623.PDT	11280633.UDT	11280638.PDT	11280648.UDT	11280653.PDT
11280703.UDT	11280708.PDT	11280718.UDT	11280723.PDT	11280733.UDT
11280738.PDT	11280748.UDT	11280753.PDT	11280803.UDT	11280808.PDT
11280818.UDT	11280823.PDT	11280833.UDT	11280838.PDT	11280848.UDT
11280853.PDT	11280903.UDT	11280908.PDT	11280918.UDT	11280923.PDT
11280933.UDT	11280938.PDT	11280948.UDT	11280953.PDT	11281003.UDT
11281008.PDT	11281018.UDT	11281023.PDT	11281033.UDT	11281038.PDT
11281048.UDT	11281053.PDT	11281103.UDT	11281108.PDT	11281118.UDT
11281123.PDT	11281133.UDT	11281138.PDT	11281148.UDT	11281153.PDT
11281203.UDT	11281208.PDT	11281218.UDT	11281223.PDT	11281233.UDT
11281238.PDT	11281248.UDT	11281253.PDT	11281303.UDT	11281308.PDT
11281318.UDT	11281323.PDT	11281333.UDT	11281338.PDT	11281348.UDT

11281353.PDT	11281403.UDT	11281408.PDT	11281418.UDT	11281423.PDT
11281433.UDT	11281438.PDT	11281448.UDT	11281453.PDT	11281503.UDT
11281508.PDT	11281518.UDT	11281523.PDT	11281533.UDT	11281538.PDT
11281548.UDT	11281553.PDT	11281603.UDT	11281608.PDT	11281618.UDT
11281623.PDT	11281633.UDT	11281638.PDT	11281648.UDT	11281653.PDT
11281703.UDT	11281708.PDT	11281718.UDT	11281723.PDT	11281733.UDT
11281738.PDT	11281748.UDT	11281753.PDT	11281803.UDT	11281809.PDT
11281818.UDT	11281824.PDT	11281833.UDT	11281839.PDT	11281848.UDT
11281854.PDT	11281903.UDT	11281909.PDT	11281918.UDT	11281924.PDT
11281933.UDT	11281939.PDT	11281948.UDT	11281954.PDT	11282003.UDT
11282009.PDT	11282018.UDT	11282024.PDT	11282033.UDT	11282039.PDT
11282048.UDT	11282054.PDT	11282103.UDT	11282109.PDT	11282118.UDT
11282124.PDT	11282133.UDT	11282139.PDT	11282148.UDT	11282154.PDT
11282203.UDT	11282209.PDT	11282218.UDT	11282224.PDT	11282233.UDT
11282239.PDT	11282248.UDT	11282254.PDT	11282303.UDT	11282309.PDT
11282318.UDT	11282324.PDT	11282333.UDT	11282339.PDT	11282348.UDT
11282354.PDT				

JDAY 333	11290003.UDT	11290009.PDT	11290018.UDT	11290024.PDT
11290033.UDT	11290039.PDT	11290048.UDT	11290054.PDT	11290103.UDT
11290109.PDT	11290118.UDT	11290124.PDT	11290133.UDT	11290139.PDT
11290148.UDT	11290154.PDT	11290203.UDT	11290209.PDT	11290218.UDT
11290224.PDT	11290233.UDT	11290239.PDT	11290248.UDT	11290254.PDT
11290303.UDT	11290309.PDT	11290318.UDT	11290324.PDT	11290333.UDT
11290339.PDT	11290348.UDT	11290354.PDT	11290403.UDT	11290409.PDT
11290418.UDT	11290424.PDT	11290433.UDT	11290439.PDT	11290448.UDT
11290454.PDT	11290503.UDT	11290509.PDT	11290518.UDT	11290524.PDT
11290533.UDT	11290539.PDT	11290548.UDT	11290554.PDT	11290603.UDT
11290609.PDT	11290618.UDT	11290624.PDT	11290633.UDT	11290639.PDT
11290648.UDT	11290654.PDT	11290703.UDT	11290709.PDT	11290718.UDT
11290724.PDT	11290733.UDT	11290738.PDT	11290748.UDT	11290753.PDT
11290803.UDT	11290809.PDT	11290818.UDT	11290824.PDT	11290833.UDT
11290839.PDT	11290848.UDT	11290854.PDT	11290903.UDT	11290909.PDT
11290918.UDT	11290924.PDT	11290933.UDT	11290938.PDT	11290948.UDT
11290954.PDT	11291003.UDT	11291008.PDT	11291018.UDT	11291023.PDT
11291033.UDT	11291039.PDT	11291048.UDT	11291053.PDT	11291103.UDT
11291108.PDT	11291118.UDT	11291123.PDT	11291133.UDT	11291138.PDT
11291148.UDT	11291154.PDT	11291203.UDT	11291208.PDT	11291218.UDT
11291223.PDT	11291233.UDT	11291239.PDT	11291248.UDT	11291253.PDT
11291303.UDT	11291308.PDT	11291318.UDT	11291323.PDT	11291333.UDT
11291338.PDT	11291348.UDT	11291353.PDT	11291403.UDT	11291408.PDT
11291418.UDT	11291423.PDT	11291433.UDT	11291438.PDT	11291448.UDT
11291453.PDT	11291503.UDT	11291508.PDT	11291518.UDT	11291523.PDT
11291533.UDT	11291538.PDT	11291548.UDT	11291553.PDT	11291603.UDT
11291608.PDT	11291618.UDT	11291623.PDT	11291633.UDT	11291638.PDT
11291648.UDT	11291653.PDT	11291703.UDT	11291708.PDT	11291718.UDT
11291723.PDT	11291733.UDT	11291738.PDT	11291748.UDT	11291753.PDT
11291803.UDT	11291809.PDT	11291819.UDT	11291824.PDT	11291834.UDT
11291839.PDT	11291849.UDT	11291854.PDT	11291904.UDT	11291909.PDT
11291919.UDT	11291924.PDT	11291934.UDT	11291939.PDT	11291949.UDT
11291954.PDT	11292004.UDT	11292009.PDT	11292019.UDT	11292024.PDT
11292034.UDT	11292039.PDT	11292049.UDT	11292054.PDT	11292104.UDT
11292109.PDT	11292119.UDT	11292124.PDT	11292134.UDT	11292139.PDT
11292149.UDT	11292154.PDT	11292204.UDT	11292209.PDT	11292219.UDT
11292224.PDT	11292234.UDT	11292239.PDT	11292249.UDT	11292254.PDT
11292304.UDT	11292309.PDT	11292319.UDT	11292324.PDT	11292334.UDT
11292339.PDT	11292349.UDT	11292354.PDT		

JDAY 334

11300019.UDT	11300024.PDT	11300034.UDT	11300004.UDT	11300009.PDT
11300054.PDT	11300104.UDT	11300109.PDT	11300039.PDT	11300049.UDT
11300134.UDT	11300139.PDT	11300149.UDT	11300119.UDT	11300124.PDT
11300209.PDT	11300219.UDT	11300224.PDT	11300154.PDT	11300204.UDT
11300249.UDT	11300254.PDT	11300304.UDT	11300224.UDT	11300239.PDT
11300324.PDT	11300334.UDT	11300339.PDT	11300309.PDT	11300319.UDT
11300404.UDT	11300409.PDT	11300419.UDT	11300349.UDT	11300354.PDT
11300439.PDT	11300449.UDT	11300454.PDT	11300424.PDT	11300434.UDT
11300519.UDT	11300524.PDT	11300534.UDT	11300504.UDT	11300509.PDT
11300554.PDT	11300604.UDT	11300609.PDT	11300539.PDT	11300549.UDT
11300634.UDT	11300639.PDT	11300649.UDT	11300619.UDT	11300624.PDT
11300709.PDT	11300719.UDT	11300724.PDT	11300654.PDT	11300704.UDT
11300749.UDT	11300754.PDT	11300804.UDT	11300734.UDT	11300739.PDT
11300824.PDT	11300834.UDT	11300839.PDT	11300809.PDT	11300819.UDT
11300904.UDT	11300909.PDT	11300919.UDT	11300849.UDT	11300854.PDT
11300939.PDT	11300949.UDT	11300954.PDT	11300924.PDT	11300934.UDT
11301019.UDT	11301024.PDT	11301034.UDT	11301004.UDT	11301009.PDT
11301054.PDT	11301104.UDT	11301109.PDT	11301039.PDT	11301049.UDT
11301134.UDT	11301139.PDT	11301149.UDT	11301119.UDT	11301124.PDT
11301209.PDT	11301219.UDT	11301224.PDT	11301154.PDT	11301204.UDT
11301249.UDT	11301254.PDT	11301304.UDT	11301234.UDT	11301239.PDT
11301324.PDT	11301334.UDT	11301339.PDT	11301309.PDT	11301319.UDT
11301404.UDT	11301409.PDT	11301419.UDT	11301349.UDT	11301354.PDT
11301439.PDT	11301449.UDT	11301454.PDT	11301424.PDT	11301434.UDT
11301519.UDT	11301524.PDT	11301534.UDT	11301504.UDT	11301509.PDT
11301554.PDT	11301604.UDT	11301609.PDT	11301539.PDT	11301549.UDT
11301634.UDT	11301639.PDT	11301649.UDT	11301619.UDT	11301624.PDT
11301709.PDT	11301719.UDT	11301724.PDT	11301654.PDT	11301704.UDT
11301749.UDT	11301754.PDT	11301804.UDT	11301734.UDT	11301739.PDT
11301825.PDT	11301834.UDT	11301839.PDT	11301809.PDT	11301819.UDT
11301904.UDT	11301909.PDT	11301919.UDT	11301849.UDT	11301855.PDT
11301939.PDT	11301949.UDT	11301954.PDT	11301925.PDT	11301934.UDT
11302019.UDT	11302024.PDT	11302034.UDT	11302004.UDT	11302009.PDT
11302054.PDT	11302104.UDT	11302109.PDT	11302039.PDT	11302049.UDT
11302134.UDT	11302139.PDT	11302149.UDT	11302119.UDT	11302124.PDT
11302209.PDT	11302219.UDT	11302224.PDT	11302154.PDT	11302204.UDT
11302249.UDT	11302254.PDT	11302304.UDT	11302234.UDT	11302239.PDT
11302324.PDT	11302334.UDT	11302339.PDT	11302309.PDT	11302319.UDT
			11302349.UDT	11302354.PDT

JDAY 335

12010004.UDT	12010009.PDT	12010019.UDT	12010024.PDT	12010034.UDT
12010039.PDT	12010049.UDT	12010054.PDT	12010104.UDT	12010109.PDT
12010119.UDT	12010124.PDT	12010134.UDT	12010139.PDT	12010149.UDT
12010154.PDT	12010204.UDT	12010209.PDT	12010219.UDT	12010224.PDT
12010234.UDT	12010239.PDT	12010249.UDT	12010254.PDT	12010304.UDT
12010309.PDT	12010319.UDT	12010324.PDT	12010334.UDT	12010339.PDT
12010349.UDT	12010354.PDT	12010404.UDT	12010409.PDT	12010419.UDT
12010424.PDT	12010434.UDT	12010439.PDT	12010449.UDT	12010454.PDT
12010504.UDT	12010509.PDT	12010519.UDT	12010524.PDT	12010534.UDT
12010539.PDT	12010549.UDT	12010554.PDT	12010604.UDT	12010609.PDT
12010619.UDT	12010624.PDT	12010634.UDT	12010639.PDT	12010649.UDT
12010654.PDT	12010704.UDT	12010709.PDT	12010719.UDT	12010724.PDT
12010734.UDT	12010739.PDT	12010749.UDT	12010754.PDT	12010804.UDT
12010809.PDT	12010819.UDT	12010824.PDT	12010834.UDT	12010839.PDT
12010849.UDT	12010854.PDT	12010904.UDT	12010909.PDT	12010919.UDT
12010924.PDT	12010934.UDT	12010939.PDT	12010949.UDT	12010954.PDT
12011004.UDT	12011009.PDT	12011019.UDT	12011024.PDT	12011034.UDT
12011039.PDT	12011049.UDT	12011054.PDT	12011104.UDT	12011109.PDT

12011119.UDT	12011124.PDT	12011134.UDT	12011139.PDT	12011149.UDT
12011154.PDT	12011204.UDT	12011209.PDT	12011219.UDT	12011224.PDT
12011234.UDT	12011239.PDT	12011249.UDT	12011254.PDT	12011304.UDT
12011309.PDT	12011319.UDT	12011324.PDT	12011334.UDT	12011339.PDT
12011349.UDT	12011354.PDT	12011404.UDT	12011409.PDT	12011419.UDT
12011424.PDT	12011434.UDT	12011439.PDT	12011449.UDT	12011454.PDT
12011504.UDT	12011509.PDT	12011519.UDT	12011524.PDT	12011534.UDT
12011539.PDT	12011549.UDT	12011554.PDT	12011604.UDT	12011609.PDT
12011619.UDT	12011624.PDT	12011634.UDT	12011639.PDT	12011649.UDT
12011654.PDT	12011704.UDT	12011709.PDT	12011719.UDT	12011724.PDT
12011734.UDT	12011739.PDT	12011749.UDT	12011754.PDT	12011804.UDT
12011810.PDT	12011820.UDT	12011825.PDT	12011835.UDT	12011840.PDT
12011850.UDT	12011855.PDT	12011905.UDT	12011910.PDT	12011920.UDT
12011925.PDT	12011935.UDT	12011940.PDT	12011950.UDT	12011955.PDT
12012005.UDT	12012010.PDT	12012020.UDT	12012025.PDT	12012035.UDT
12012040.PDT	12012050.UDT	12012055.PDT	12012105.UDT	12012110.PDT
12012120.UDT	12012125.PDT	12012135.UDT	12012140.PDT	12012150.UDT
12012155.PDT	12012205.UDT	12012210.PDT	12012220.UDT	12012225.PDT
12012235.UDT	12012240.PDT	12012250.UDT	12012255.PDT	12012305.UDT
12012310.PDT	12012320.UDT	12012325.PDT	12012335.UDT	12012340.PDT
12012350.UDT	12012355.PDT			

JDAY 336

		12020005.UDT	12020010.PDT	12020020.UDT
12020025.PDT	12020035.UDT	12020040.PDT	12020050.UDT	12020055.PDT
12020105.UDT	12020110.PDT	12020120.UDT	12020125.PDT	12020135.UDT
12020140.PDT	12020150.UDT	12020155.PDT	12020205.UDT	12020210.PDT
12020220.UDT	12020225.PDT	12020235.UDT	12020240.PDT	12020250.UDT
12020255.PDT	12020305.UDT	12020310.PDT	12020320.UDT	12020325.PDT
12020335.UDT	12020340.PDT	12020350.UDT	12020355.PDT	12020405.UDT
12020410.PDT	12020420.UDT	12020425.PDT	12020435.UDT	12020440.PDT
12020450.UDT	12020455.PDT	12020505.UDT	12020510.PDT	12020520.UDT
12020525.PDT	12020535.UDT	12020540.PDT	12020550.UDT	12020555.PDT
12020605.UDT	12020610.PDT	12020620.UDT	12020625.PDT	12020635.UDT
12020640.PDT	12020650.UDT	12020655.PDT	12020705.UDT	12020710.PDT
12020720.UDT	12020725.PDT	12020735.UDT	12020740.PDT	12020750.UDT
12020755.PDT	12020805.UDT	12020810.PDT	12020820.UDT	12020825.PDT
12020835.UDT	12020840.PDT	12020850.UDT	12020855.PDT	12020905.UDT
12020910.PDT	12020920.UDT	12020925.PDT	12020935.UDT	12020940.PDT
12020950.UDT	12020955.PDT	12021005.UDT	12021010.PDT	12021020.UDT
12021025.PDT	12021035.UDT	12021040.PDT	12021050.UDT	12021055.PDT
12021105.UDT	12021110.PDT	12021120.UDT	12021125.PDT	12021135.UDT
12021140.PDT	12021150.UDT	12021155.PDT	12021205.UDT	12021210.PDT
12021220.UDT	12021225.PDT	12021235.UDT	12021240.PDT	12021250.UDT
12021255.PDT	12021305.UDT	12021310.PDT	12021320.UDT	12021325.PDT
12021335.UDT	12021340.PDT	12021350.UDT	12021355.PDT	12021405.UDT
12021410.PDT	12021420.UDT	12021425.PDT	12021435.UDT	12021440.PDT
12021450.UDT	12021455.PDT	12021505.UDT	12021510.PDT	12021520.UDT
12021525.PDT	12021535.UDT	12021540.PDT	12021550.UDT	12021555.PDT
12021605.UDT	12021610.PDT	12021620.UDT	12021625.PDT	12021635.UDT
12021640.PDT	12021650.UDT	12021655.PDT	12021705.UDT	12021710.PDT
12021720.UDT	12021725.PDT	12021735.UDT	12021740.PDT	12021750.UDT
12021755.PDT	12021805.UDT	12021810.PDT	12021820.UDT	12021825.PDT
12021835.UDT	12021840.PDT	12021850.UDT	12021855.PDT	12021905.UDT
12021910.PDT	12021920.UDT	12021925.PDT	12021935.UDT	12021940.PDT
12021950.UDT	12021955.PDT	12022005.UDT	12022010.PDT	12022020.UDT
12022025.PDT	12022035.UDT	12022040.PDT	12022050.UDT	12022055.PDT
12022105.UDT	12022110.PDT	12022120.UDT	12022125.PDT	12022135.UDT
12022140.PDT	12022150.UDT	12022155.PDT	12022205.UDT	12022210.PDT

12022220.UDT	12022225.PDT	12022235.UDT	12022240.PDT	12022250.UDT
12022255.PDT	12022305.UDT	12022310.PDT	12022320.UDT	12022325.PDT
12022335.UDT	12022340.PDT	12022350.UDT	12022355.PDT	

JDAY 337

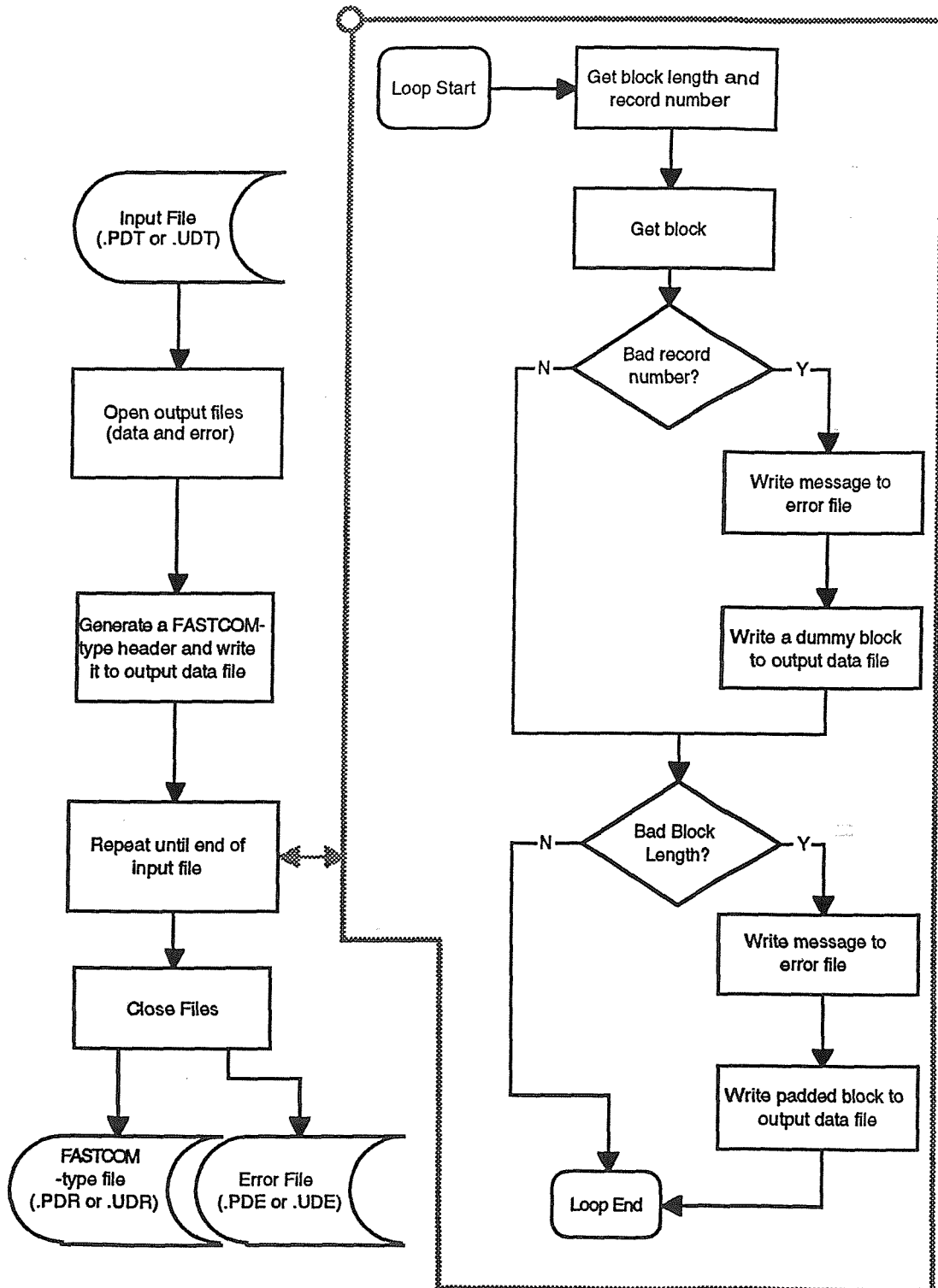
				12030005.UDT
12030010.PDT	12030020.UDT	12030025.PDT	12030035.UDT	12030040.PDT
12030050.UDT	12030055.PDT	12030105.UDT	12030110.PDT	12030120.UDT
12030125.PDT	12030135.UDT	12030140.PDT	12030150.UDT	12030155.PDT
12030205.UDT	12030210.PDT	12030220.UDT	12030225.PDT	12030235.UDT
12030240.PDT	12030250.UDT	12030255.PDT	12030305.UDT	12030310.PDT
12030320.UDT	12030325.PDT	12030335.UDT	12030340.PDT	12030350.UDT
12030355.PDT	12030405.UDT	12030410.PDT	12030420.UDT	12030425.PDT
12030435.UDT	12030440.PDT	12030450.UDT	12030455.PDT	12030505.UDT
12030510.PDT	12030520.UDT	12030525.PDT	12030535.UDT	12030540.PDT
12030550.UDT	12030555.PDT	12030605.UDT	12030610.PDT	12030620.UDT
12030625.PDT	12030635.UDT	12030640.PDT	12030650.UDT	12030655.PDT
12030705.UDT	12030710.PDT	12030720.UDT	12030725.PDT	12030735.UDT
12030740.PDT	12030750.UDT	12030755.PDT	12030805.UDT	12030810.PDT
12030820.UDT	12030825.PDT	12030835.UDT	12030840.PDT	12030850.UDT
12030855.PDT	12030905.UDT	12030910.PDT	12030920.UDT	12030925.PDT
12030935.UDT	12030940.PDT	12030950.UDT	12030955.PDT	12031005.UDT
12031010.PDT	12031020.UDT	12031025.PDT	12031035.UDT	12031040.PDT
12031050.UDT	12031055.PDT	12031105.UDT	12031110.PDT	12031120.UDT
12031125.PDT	12031135.UDT	12031140.PDT	12031150.UDT	12031155.PDT
12031205.UDT	12031210.PDT	12031220.UDT	12031225.PDT	12031235.UDT
12031240.PDT	12031250.UDT	12031255.PDT	12031305.UDT	12031310.PDT
12031320.UDT	12031325.PDT	12031335.UDT	12031340.PDT	12031350.UDT
12031355.PDT	12031405.UDT	12031410.PDT	12031420.UDT	12031425.PDT
12031435.UDT	12031440.PDT	12031450.UDT	12031455.PDT	12031505.UDT
12031510.PDT	12031520.UDT	12031525.PDT	12031535.UDT	12031540.PDT
12031550.UDT	12031555.PDT	12031605.UDT	12031610.PDT	12031620.UDT
12031625.PDT	12031635.UDT	12031640.PDT	12031650.UDT	12031655.PDT
12031705.UDT	12031710.PDT	12031720.UDT	12031725.PDT	12031735.UDT
12031740.PDT	12031750.UDT	12031755.PDT	12031805.UDT	12031811.PDT
12031821.UDT	12031826.PDT	12031836.UDT	12031841.PDT	12031851.UDT
12031856.PDT	12031906.UDT	12031911.PDT	12031921.UDT	12031926.PDT
12031936.UDT	12031941.PDT	12031951.UDT	12031956.PDT	12032006.UDT
12032011.PDT	12032021.UDT	12032026.PDT	12032036.UDT	12032041.PDT
12032051.UDT	12032056.PDT	12032106.UDT	12032111.PDT	12032121.UDT
12032126.PDT	12032136.UDT	12032141.PDT	12032151.UDT	12032156.PDT
12032206.UDT	12032211.PDT	12032221.UDT	12032226.PDT	12032236.UDT
12032241.PDT	12032251.UDT	12032256.PDT	12032306.UDT	12032311.PDT
12032321.UDT	12032326.PDT	12032336.UDT	12032341.PDT	12032351.UDT
12032356.PDT				

JDAY 338

	12040006.UDT	12040011.PDT	12040021.UDT	12040026.PDT
12040036.UDT	12040041.PDT	12040051.UDT	12040056.PDT	12040106.UDT
12040111.PDT	12040121.UDT	12040126.PDT	12040136.UDT	12040141.PDT
12040151.UDT	12040156.PDT	12040206.UDT	12040211.PDT	12040221.UDT
12040226.PDT	12040236.UDT	12040241.PDT	12040251.UDT	12040256.PDT
12040306.UDT	12040311.PDT	12040321.UDT	12040326.PDT	12040336.UDT
12040341.PDT	12040351.UDT	12040356.PDT	12040406.UDT	12040411.PDT
12040421.UDT	12040426.PDT	12040436.UDT	12040441.PDT	12040451.UDT
12040456.PDT	12040506.UDT	12040511.PDT	12040521.UDT	12040526.PDT
12040536.UDT	12040541.PDT	12040551.UDT	12040556.PDT	12040606.UDT
12040611.PDT	12040621.UDT	12040626.PDT	12040636.UDT	12040641.PDT
12040651.UDT	12040656.PDT	12040706.UDT	12040711.PDT	12040721.UDT
12040726.PDT	12040736.UDT	12040741.PDT	12040751.UDT	12040756.PDT

12040806.UDT	12040811.PDT	12040820.UDT	12040826.PDT	12040836.UDT
12040841.PDT	12040850.UDT	12040856.PDT	12040905.UDT	12040911.PDT
12040921.UDT	12040926.PDT	12040935.UDT	12040941.PDT	12041504.UDT
12041913.PDT	12042104.UDT	12042109.PDT		

2967 file(s) 282468414 bytes
1081344 bytes free

10. APPENDIX E SOFTWARE LISTINGS**10.1. Appendix E.1 THORREAD.C**

```

/*****THORREAD.C*****/
*
* Converts files produced by the THORCOM telemetry logging system
* to the standard file format produced by FASTCOM
* (as produced by the ship system)
*
* Deals with errors as follows:
*   if a data packet is missing, inserts a packet of all zero values
*   with length 210 bytes
*
*   if a data packet is short (non multiple of 10 bytes), adds zero
*   values to make it up to length 210 bytes
*
* NB packets are normally of length 200, 210 or 220 bytes,
*   210 is used as best guess for correct number for prompted data
*   200 is used as best guess for unprompted data
*
* NB first packet is discarded as it normally contains dud data
*   resulting from change from prompted to unprompted during fill
*   of the anemometer output buffer
*
* Writes description of each error to an individually named error file (.PDE or .UDE)
*
* Call with path/filename added, e.g. THORREAD C:\LOGDATA\11021015.PDT
* (or use suitably modified batch file TXPDR.BAT or TXUDR.BAT)
* This results in a FASTCOM-type file named      C:\DATA\11021015.PDR
* and an ASCII error file named                  C:\DATA\11021015.PDE
*
* CHC
* 25th October 1993
*
*****/

#include<stdio.h>
#include<stdlib.h>
#include<conio.h>
#include<string.h>

main(int argc, char *argv[])
{
char buffer[256], dum_rec[256];
char *filename, header[50], infile[32], outfile[32], thor_time[10], thor_date[10];

FILE * f_in;
FILE * f_out;
FILE * f_errs;

int ch, expected_rec_no, expected_length, first = 1, rec_no;

unsigned block_length, prompted;

unsigned long sum_bytes = 0L;

if (argc != 2)
{
printf("Wrong arguments\n");
exit(0);
}
for (ch = 0; ch < 210; ch++)
{
dum_rec[ch] = 0;
}

```

```

strcpy(infile, argv[1]);

if ( (filename = strrchr(infile, '\\') == NULL)
    {
        filename = infile;          /* path only contains name */
    }
else
    {
        filename++;                 /* set to name */
    }

if (strpbrk(filename, "Pp") != NULL)
    {
        prompted = 1;
        expected_length = 210;
    }
else
    {
        prompted = 0;
        expected_length = 200;
    }

strcpy(header, "Mode 1\nAnalog 1\nTime ");
thor_time[0] = filename[4];
thor_time[1] = filename[5];
thor_time[2] = 58; /* colon */
thor_time[3] = filename[6];
thor_time[4] = filename[7];
thor_time[5] = '\0';
strcat(thor_time, ":00");
thor_date[0] = filename[0];
thor_date[1] = filename[1];
thor_date[2] = 47; /* slash */
thor_date[3] = filename[2];
thor_date[4] = filename[3];
thor_date[5] = '\0';
strcat(thor_date, "/93\n");
strcat(header, thor_time);
strcat(header, "Date ");
strcat(header, thor_date);

if ( (f_in = fopen(infile, "rb")) == NULL)
    {
        printf("Could not open input file %s\n", infile);
        exit(0);
    }

strcpy(outfile, "c:\\data\\");
filename[11] = 'e';
strcat(outfile, filename);

if ( (f_errs = fopen(outfile, "w")) == NULL)
    {
        printf("Could not open errors file %s\n", outfile);
        exit(0);
    }

strcpy(outfile, "c:\\data\\");
filename[11] = 'r';
strcat(outfile, filename);

if ( (f_out = fopen(outfile, "wb")) == NULL)
    {
        printf("Could not open output file %s\n", outfile);
        exit(0);
    }

```

```

    }
else
{
    printf("Converting to FASTCOM format file %s\nWait a while please . .\n", outfile);
}

fwrite((void *) header, 1, 44, f_out);

while (feof(f_in) == 0)
{
    block_length = fgetc(f_in);
    rec_no = 256 * fgetc(f_in);
    rec_no += fgetc(f_in);
    if (first == 1)
    {
        fprintf(f_errs, "Start record no. %d\n", rec_no);
    }
    /* printf("%d %d\n", block_length, rec_no); */
    block_length -= 2;

    fread((void *) buffer, 1, block_length, f_in);

    if ( (rec_no != expected_rec_no) && (first == 0) && (feof(f_in) == 0) )
    {
        printf("Missing record at %d\n", expected_rec_no);
        fprintf(f_errs, "Missing record at %d\n", expected_rec_no);
        fwrite((void *) dum_rec, 1, expected_length, f_out);
        sum_bytes += expected_length;
    }

    if ( (div(block_length, 10).rem != 0) && (first == 0) && (feof(f_in) == 0) )
    {
        printf("Bad Block Length (%d) at %d\n", block_length, rec_no);
        fprintf(f_errs, "Bad Block Length (%d) at %d\n", block_length, rec_no);
        for (ch = block_length; ch < (10 * (div(block_length, 10).quot + 1)); ch++)
        {
            buffer[ch] = 0; /* pad out block with zeros to length multiple of 10 */
        }
        block_length = expected_length;
    }

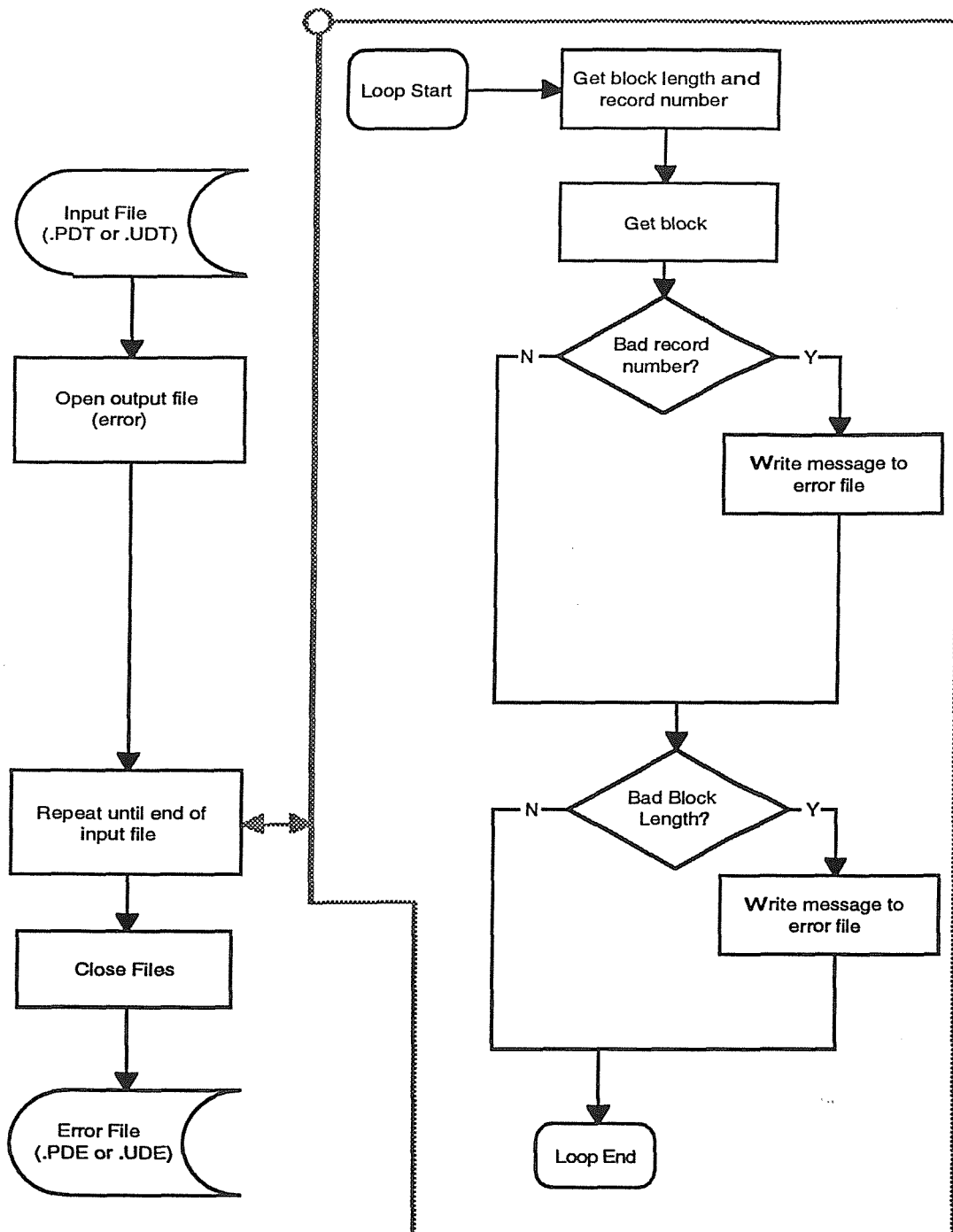
    if ( (first == 0) && (feof(f_in) == 0) )
    {
        fwrite((void *) buffer, 1, block_length, f_out);
        sum_bytes += block_length;
    }

    /* printf("%d %ld\n", rec_no, sum_bytes); */
    first = 0;
    expected_rec_no = rec_no + 1;
    if (expected_rec_no > 10000)
    {
        expected_rec_no = 1;
    }
    /* getch(); */
}

fclose(f_in);
fclose(f_out);
fclose(f_errs);
printf("Done\n");
return 0;
}

```

10.2. Appendix E.2 THORSTAT.C



```

/*****THORSTAT.C*****/
*
* Reads files produced by the THORCOM system and produces error files
*
* Writes description of each error to an individually named error file (.PDE or .UDE)
*
* Call with path/filename added, e.g. THORSTAT C:\LOGDATA\11021015.PDT
* (or use a modified form of TXPDR.BAT and TXUDR.BAT)
* This results in an ASCII error file named
*      C:\DATA\11021015.PDE
*
* CHC
* 19th January 1994
*
*****/

#include<stdio.h>
#include<stdlib.h>
#include<conio.h>
#include<string.h>

main(int argc, char *argv[])
{
    char buffer[256], dum_rec[256];
    char *filename, header[50], infile[50], outfile[50], thor_time[10], thor_date[10];

    FILE *f_in;
    FILE *f_errs;

    int ch, expected_rec_no, expected_length, first = 1, rec_no;

    unsigned block_length, prompted;

    unsigned long sum_bytes = 0L;

    if (argc != 2)
    {
        printf("Wrong arguments\n");
        exit(0);
    }
    for (ch = 0; ch < 210; ch++)
    {
        dum_rec[ch] = 0;
    }

    strcpy(infile, argv[1]);

    if ( (filename = strchr(infile, '\\')) == NULL)
    {
        filename = infile;                /* path only contains name */
    }
    else
    {
        filename++;                        /* set to name */
    }

    if (strpbrk(filename, "Pp") != NULL)
    {
        prompted = 1;
        expected_length = 210;
    }
    else
    {

```

```

        prompted = 0;
        expected_length = 200;
    }
    strcpy(header, "Mode 1\nAnalog 1\nTime ");
    thor_time[0] = filename[4];
    thor_time[1] = filename[5];
    thor_time[2] = 58; /* colon */
    thor_time[3] = filename[6];
    thor_time[4] = filename[7];
    thor_time[5] = '\0';
    strcat(thor_time, ":00");
    thor_date[0] = filename[0];
    thor_date[1] = filename[1];
    thor_date[2] = 47; /* slash */
    thor_date[3] = filename[2];
    thor_date[4] = filename[3];
    thor_date[5] = '\0';
    strcat(thor_date, "/93\n");
    strcat(header, thor_time);
    strcat(header, "Date ");
    strcat(header, thor_date);

    if ( (f_in = fopen(infile, "rb")) == NULL)
    {
        printf("Could not open input file %s\n", infile);
        exit(0);
    }
    strcpy(outfile, "c:\\data\\");
    filename[11] = 'e';
    strcat(outfile, filename);

    if ( (f_errs = fopen(outfile, "w")) == NULL)
    {
        printf("Could not open errors file %s\n", outfile);
        exit(0);
    }

    strcpy(outfile, "c:\\data\\");
    filename[11] = 'r';
    strcat(outfile, filename);

    while (feof(f_in) == 0)
    {
        block_length = fgetc(f_in);
        rec_no = 256 * fgetc(f_in);
        rec_no += fgetc(f_in);
        if (first == 1)
        {
            fprintf(f_errs, "Start record no. %d\n", rec_no);
        }
        /* printf("%d %d\n", block_length, rec_no); */
        block_length -= 2;

        fread((void *) buffer, 1, block_length, f_in);

        if ( (rec_no != expected_rec_no) && (first == 0) && (feof(f_in) == 0) )
        {
            printf("Missing record at %d\n", expected_rec_no);
            fprintf(f_errs, "Missing record at %d\n", expected_rec_no);
            sum_bytes += expected_length;
        }

        if ( (div(block_length, 10).rem != 0) && (first == 0) && (feof(f_in) == 0) )

```

```
    {
        printf("Bad Block Length (%d) at %d\n", block_length, rec_no);
        fprintf(f_errs, "Bad Block Length (%d) at %d\n", block_length, rec_no);
        for (ch = block_length; ch < (10 * (div(block_length, 10).quot + 1)); ch++)
        {
            buffer[ch] = 0; /* pad out block with zeros to length multiple of 10 */
        }
        block_length = expected_length;
    }

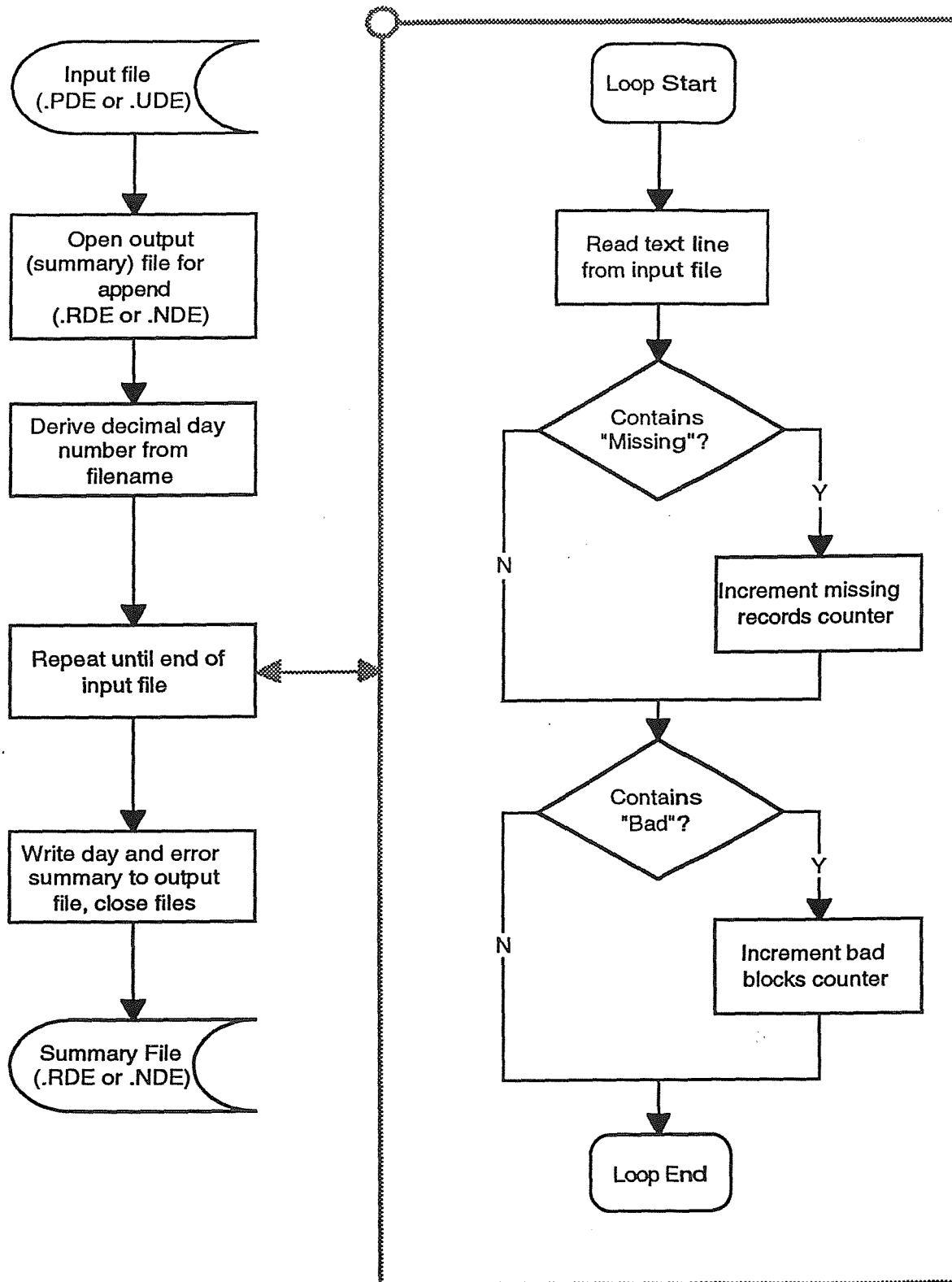
    if ( (first == 0) && (feof(f_in) == 0) )
    {
        sum_bytes += block_length;
    }

    /* printf("%d %ld\n", rec_no, sum_bytes); */
    first = 0;
    expected_rec_no = rec_no + 1;
    if (expected_rec_no > 10000)
    {
        expected_rec_no = 1;
    }
    /* getch(); */
}

fclose(f_in);
fclose(f_errs);
printf("Done\n");

return 0;
}
```

10.3. Appendix E.3 THORSUMM.C



```

/*****THORSUMM.C*****/
*
* Reads .PDE and .UDE error files produced by the THORSTAT program and appends to
* a .RDE or .NDE summary file
*
* Call with path/filename added, e.g. THORSUMM C:\DATA\1102???.PDE
* (or use a suitably modified form of SUMPDE.BAT or SUMUDE.BAT)
* This results in data statistics being appended to an ASCII error file named
*      C:\DATA\SUMMARY.RDE
*
* CHC
* 19th January 1994
*
*****/

#include<stdio.h>
#include<stdlib.h>
#include<conio.h>
#include<string.h>

main(int argc, char *argv[])
{
    char buffer[256];
    char *filename, header[50], infile[32], outfile[32], thor_time[10], thor_date[10];
    char *fdecode, *stop_at;

    FILE *f_in;

    FILE *f_errs;

    float jday = 0.;

    int ch, expected_rec_no, expected_length, first = 1, rec_no;
    int missing = 0, bad_block = 0;
    int month, day, hours, minutes;

    unsigned block_length, prompted;

    unsigned long sum_bytes = 0L;

    if (argc != 2)
    {
        printf("Wrong arguments\n");
        exit(0);
    }

    /* input file given in command line argument */
    strcpy(infile, argv[1]);

    if ( (filename = strrchr(infile, '\\')) == NULL)
    {
        filename = infile;          /* path only contains name */
    }
    else
    {
        filename++;                  /* set to name */
    }

    if ( (f_in = fopen(infile, "rb")) == NULL)
    {
        printf("Could not open input file %s\n", infile);
        exit(0);
    }

```

```

    }
    else
    {
        printf("%s\n", filename);
    }
    strcpy(outfile, "c:\\data\\summary.");
    if (strpbrk(filename, "Pp") != NULL)
    {
        strcat(outfile, "rde");
    }
    else
    {
        strcat(outfile, "nde");
    }

    if ( (f_errs = fopen(outfile, "a")) == NULL)
    {
        printf("Could not open errors file %s\n", outfile);
        exit(0);
    }

    /* need to convert mmddhhmm.xde filename to a decimal day value
    to put in the output file */
    fdecode = filename + 6;
    *(fdecode + 2) = 0;
    minutes = (int) strtol(fdecode, &stop_at, 10);
    fdecode = filename + 4;
    *(fdecode + 2) = 0;
    hours = (int) strtol(fdecode, &stop_at, 10);
    fdecode = filename + 2;
    *(fdecode + 2) = 0;
    day = (int) strtol(fdecode, &stop_at, 10);
    fdecode = filename;
    *(fdecode + 2) = 0;
    month = (int) strtol(fdecode, &stop_at, 10);

    switch(month)
    {
        case(10):
            jday = 273.;
            break;
        case(11):
            jday = 304.;
            break;
        case(12):
            jday = 334.;
            break;
        default:
            printf("Month out of range\n");
            exit(0);
    }
    jday += (float) day;
    jday += ( (float) hours / 24.);
    jday += ( (float) minutes / 1440.);

    while (feof(f_in) == 0)
    {
        if ( fgets(buffer, 256, f_in) != 0 )
        {
            if (strstr(buffer, "Start") != NULL)
            {

```

```
        printf("S");
    }
    if (strstr(buffer, "Missing") != NULL)
    {
        printf("M");
        missing++;
    }
    if (strstr(buffer, "Bad") != NULL)
    {
        printf("B");
        bad_block++;
    }
    }
    printf("\n");

    fprintf(f_errs, "%9.5f\t%d\t%d\n", jday, missing, bad_block);
    fclose(f_in);

    fclose(f_errs);
    printf("Done\n");

    return 0;
}
```

10.4. Appendix E.4 REPLAY2.C

```

/*****

```

```

Program REPLAY2.C

```

```

Version 1.1 28th October 1993

```

```

Author CHC

```

```

Compile using command line:

```

```

    qcl /AM /Zr /FPi87 replay2.c /F 9000 mlibc7.lib

```

Sonic processing program: use to replaying an existing Mode 1 raw data file, with 1 analogue channel for compass as used in sonic buoy, with save of spectral data .PRN and parameter .MWS files to floppy and hard disks

Requires fast 286 or 386 processor with coprocessor
Install in c:\sonic directory together with SETUP.SON (configuration file)

Throws out bad data

Call using command line REPLAY2 n path/name

where n is number of FFT sections expected (12 for a .PDR file, 6 for a .UDR file)
and path/name is the path and name of the .PDR or .UDR file, e.g. C:\DATA\11021015.PDR

Alternatively, use batch file REPPDR.BAT or REPUDR.BAT

Spectrum will be written to hard and floppy disk .PRN and .MWS files

e.g.

C:\DATA\11021015.PRN, C:\DATA\11021015.MWS, A:\11021015.PRN, A:\11021015.MWS

```

*****/

```

```

#include <stdio.h>

```

```

#include <stdlib.h>

```

```

#include <math.h>

```

```

#include <time.h>

```

```

#include <conio.h>

```

```

#include <graph.h>

```

```

#include <pgchart.h>

```

```

#include <string.h>

```

```

#include <process.h>

```

```

#include <dos.h>

```

```

#include "zcal.h"

```

```

#define LINES 256

```

```

/* = half the effective no. of samples per section */

```

```

#define pi 3.141592654

```

```

#define ramfile "testfile"

```

```

/* filename for raw data */

```

```

#define yoffset 32

```

```

#pragma check_stack(on)

```

```

#pragma check_pointer(on)

```

```

typedef enum {FALSE, TRUE} boolean;

```

```

/***** Function declarations *****/

```

```

float fitconsts(int, int, int *, int *, int *, float *, float *);

```

```

int getdat(int, int, int, char [], char [], float *, float *, float *, float *);

```

```

float dcfiler(int);
void window(int, float, float);
void fourl(int);
float regres(int, float *, float *, float *, float *, double *, double *);
void set_titles( char *, char *, char *, char * );
void set_axes( float, float, float, float, float, float , int);
void set_colours( short, short, short, short );
void do_cross( int, int, short );
char * wait_start(int, int *, int *);
void set_envconsts(int ypixels, int col0);
char * aform(int, int);
void waitl(void);
void harderror_handler(unsigned, unsigned, unsigned *);
void check_disks(int, char *, char *, char *, char *, char *, FILE *, FILE *, FILE *, FILE *);
void show_errors(char *);
int setup(char *);
int check_cache(char *, char *, FILE *);
void wipe_line(void);
/* float meanpsd(int); */ /* temporary addition */

/*****Global Variables*****/
char message[85] = "";
char julian[10];
char ser_no[12];

chartenv env;

FILE *fh;

double a[4 * LINES + 2], p[LINES + 2]; /* double necessary for precision */

float freq1, freq2, fu[LINES + 2], rate; /* fu[] == F0 in PKT FSprog */
float last_u = 0., last_v = 0., last_w = 0., last_c = 340.;

int q[4], raw_optical_save, recs_per_file, rows, sections;

long x_calibration_table[361];
long y_calibration_table[361];
long away_z_calibration_table[361];
long into_z_calibration_table[361];
unsigned char error_flag;
unsigned last_comp = 0;

/*****Start of Main*****/
main(int argc, char **argv)
{
char samples[10];
char gilltime[40];
char subhead[40] = "Section: ";
char fft_sec[10];
char head[40] = "Last Start";
char oi_comm[20];

char full_message[50] = "";
char spec_file[25], spec_file[25], spec_file[20], spec_file[20];
char raw_filename[20];
char baud[5], son_mode[5];
char back_up[30];
char sonic_id[6];
char infile[32];
char * fname_raw;

double a1, a2, b;

```

```

FILE *f_out, *ff_out, *f_cache, *fl_out, *ffl_out, *ps; /* ps TEMP INSERT */
/* hard .prn, hard .mws, hard .raw (cache), floppy .mws and floppy .prn, respectively */

float c_mean, den, dfr, dummy, east_mean, fact, fm, fp, fl, f2, fr1, fr2;
float invsw, mean, north_mean, psd, p1, p2;
float r, rms1, scale_x, scale_y, sea, seb, sumw, vert_mean;

float yvalue[1024];
float xvalue[LINES + 2];
float nvalue[256];
float psd_set[101];
float meanws_set[101];

int cflag, first = 1, fft, gflag, good_reads, i, j, j2;
int mode = _VRES16COLOR;
int nans4, nfft, nrec = 2 * LINES;
int nr2, nr23, nr24, nspec = LINES, qtr, re, sample;
int ypixels;
int xp, yp;

palettetype palette_struct;

short col0, col1, col2, col3, col4, col5, col6, col7, col8;

struct rccoord curpos;

unsigned good_total;

_harderr(harderror_handler);          /* set up hardware error handling */

if (argc != 3)
{
    printf("Wrong arguments\n");
    exit(0);
}

gflag = setup(sonic_id);
/* load system parameters and sensor calibration tables from setup.son file */

/* set timezone to GMT */
if (putenv("TZ=GMT") == -1)
{
    ("Error in setting TZ\n");
    return 0;
}
tzset();

if (gflag > 0)
{
    /* Initialise arrays for long term psd/meanws plot */

    for (i = 0; i < 100; i++)
    {
        psd_set[i] = 0.;
        meanws_set[i] = 0.;
    }

    for (i = 0; i < 256; i++)
    {
        nvalue[i] = (float) i / 20.83;
    }
}

```

```

/* printf("Enter file path/name for raw data\n");
scanf("%s", infile); */
strcpy(infile, argv[2]);

/***** Set up Graphics using pgchart functions *****/

while( !_setvideomode( mode ) )
{
    mode--;
}

if((mode != _ERESNOCOLOR) && (mode != _VRES2COLOR) && (mode !=
_VRES16COLOR))
{
    printf("Compatible Graphics Adaptor not installed\n");
    return( 0 );
}
switch (mode)
{
    case _ERESNOCOLOR:
    {
        ypixels = 350;
        rows = 25;
        col0 = 0; col1 = 0; col2 = 5;
        col3 = 5; col4 = 2; col5 = 2;
        col6 = 2; col7 = 4; col8 = 6;
        break;
    }
    case _VRES2COLOR:
    {
        ypixels = 480;
        rows = 30;
        col0 = 0; col1 = 12; col2 = 3;
        col3 = 4; col4 = 5; col5 = 6;
        col6 = 6; col7 = 11; col8 = 10; /* col6 was 7 */
        break;
    }
    case _VRES16COLOR:
    {
        ypixels = 480;
        rows = 30;
        col0 = 0; col1 = 12; col2 = 3;
        col3 = 4; col4 = 5; col5 = 6;
        col6 = 6; col7 = 11; col8 = 10; /* col6 was 7 */
        break;
    }
}
}

/**** end of graphics setup ****/

_clearscreen(_GCLEARSCREEN);

/***** Set up parameters for FFT, etc *****/

sumw = fitconsts(nrec, nspec, &nr2, &nr23, &nr24, &fm, &fp);
/* FSprog line 650 */
/* nr2 is total number of samples per section */

nans4 = nspec/4;

/* loop to do recs_per_file x 1/4 hr processes,
with results put in file spec_file */

```

```

for (re = 0; re < recs_per_file; re++)
{
    if ( (fname_raw = strchr(infile, '\\')) == NULL)
    {
        fname_raw = infile;                /* path only contains name */
    }
    else
    {
        fname_raw++;                      /* points to name, e.g. "10201344.pdr" */
    }

    strcpy(julian, fname_raw);
    julian[8] = '\0'; /* chop off the ".pdr" */
    sample = 1;
    qtr = 0;

    /* strcpy(julian, wait_start(rows, &sample, &qtr)); */

    if (q[qtr] == 1)
    {
        rate = 20.83;
        nfft = 12;
        strcpy(baud, "1");                /* 4800 baud */
        strcpy(son_mode, "1");
    }
    if (q[qtr] == 4)
    {
        rate = 55.55;
        nfft = 26;
        strcpy(baud, "2");                /* 9600 baud */
        strcpy(son_mode, "4");
    }
    if ( (q[qtr] != 1) && (q[qtr] != 4) && (q[qtr] != 0) )
    {
        printf("Incorrect Mode in setup file\n");
        return 0;
    }
    nfft = atoi(argv[1]);
    invsw = log10( (float) nrec / rate );

    /****** Initialise array for Graphics x-values *****/

    for (i = 0; i <= nspec; i++)
    {
        xvalue[i] = (float) i * rate / nrec;
    }
    yvalue[0] = 0.;

    if(raw_optical_save == 1)
    {
        cflag = check_cache(julian, raw_filename, f_cache);
    }
    else
    {
        cflag = 0;
    }

    /* Calculate frequencies for binning */
    fu[1] = 0.;

    for (i = 2; i <= nspec; i++)

```

```

    {
        fu[i] = 10. * log10( (float) rate * (i-1)/(2 * nspec) );
    }

    ltoa( (long) nfft * (long) nr2, samples, 10 );

    show_errors(full_message);

    /* delete disk retracted message */
    if ( (first == 1) || (gflag == 0) )
    {
        _settextposition(3,0);
        printf("          ");
    }
    else
    {
        _setcolor( _getpixel(env.chartwindow.x1 + 10, env.chartwindow.y1 + 30) );
        for (xp = env.chartwindow.x1; xp < env.chartwindow.x1 + 177; xp++)
        {
            for (yp = env.chartwindow.y1; yp < env.chartwindow.y1 + 17; yp++)
            {
                _setpixel(xp, yp);
            }
        }
    }
    strcpy(message, "");
    check_disks(first, full_message, spec_file, spec_ffile, spec_ffile,
               spec_fffile, f_out, ff_out, fl_out, fl_out);
    first = 0;

    mean = 0.;
    for (i = 1; i <= nspec; i++)          /* Initialise p[] array */
    {
        p[i] = 0.;
    }
    /* call fastcom.exe, datafile c:\sonic\testfile, mode 4, 9600 baud,
    nfft*nr2 records, 0 analogue inputs */

    wipe_line();
    printf("Record %d - getting %s samples from Sonic    (Wait)",
           sample, samples);

    wipe_line();
    printf("Samples acquired . . . Starting to Process");

    waitl();

    if (gflag > 0)
    {
        _pg_initchart();
        _pg_defaultchart( &env, _PG_SCATTERCHART, _PG_POINTANDLINE );
        set_envconsts(ypixels, col0);

        _pg_getpalette( palette_struct );
        palette_struct[1].plotchar = 0;          /* set to 0 so that no char is plotted */
        _pg_setpalette( palette_struct );
    }

    /***** Start actual calcs *****/

```

```

good_reads = 0;
north_mean = 0.;
east_mean = 0.;
vert_mean = 0.;
c_mean = 0.;

for (fft = 1; fft <= nfft; fft++)
{
    if ( kbhit() )
    {
        if (getch() == 125)          /* abort tidily if } key pressed */
        {
            exit(0);
        }
    }

    /* Get data section, calc mean, apply window, do fft */
    if ( getdat(q[qr], fft, nr2, infile, gilltime,
        &dummy, &north_mean, &east_mean, &vert_mean, &c_mean) == 0)
    {
        if (gflag > 0)
        {
            /****** Plot time series for this section of data *****/

            for (j = 0; j < 1024; j += 2)
            {
                yvalue[j/2] = a[j];
                yvalue[512 + j/2] = a[j + 1];
            }

            itoa(fft, fft_sec, 10);
            strcat(subhead, fft_sec);
            set_titles( "Time (s)", "Velocity (m/s)", "Time Series", subhead );

            set_axes(0, 13, 1, 0, 20., 2., ypixels);
            set_colours(col4, col0, col1, col0); /* last was 3 */

            for (j = 0; j < 1024; j += 256)
            {
                for (i = 0; i < 256; i++)
                {
                    yvalue[i] = yvalue[i + j];
                }
                if( _pg_chartscatter( &env, nvalue, yvalue, 256 ) )
                {
                    _setvideomode(_DEFAULTMODE);
                    _outtext( "Error: can't draw chart" );
                }
            }
            wait10;

            _settextposition(rows - 1, 10);
            _settextcolor(col7);
            _outtext("Sonic Record Start: ");
            _outtext( gilltime );
            _outtext(" ");
        }
    }
    wait10;

    good_reads++;

    mean += dcfiler(nr2);

```

```

window(nrec, fm, fp);

fourl(nrec);

/* convert complex estimates to power */
a[1] = a[1] * a[1] + a[2] * a[2];

for (j = 2; j <= nspec; j++)
{
    j2 = j * 2;

    a[j] = a[j2] * a[j2] + a[j2 - 1] * a[j2 - 1]
           + a[nr24 - j2] * a[nr24 - j2]
           + a[nr23 - j2] * a[nr23 - j2];
}

den = sumw * nr2;
/* corrected sumw 11/02/92 */

/* accumulate power estimates */
for (i = 1; i <= nspec; i++)
{
    p[i] += a[i];

    if (a[i] <= 0.)
    {
        printf("Error a[%d] %e\n", i, a[i]);
        a[i] = 0.;
    }
    else
    {
        a[i] = log10(a[i]/den);
    }
    yvalue[i] = (float) a[i];
}

if (gflag > 0)
{
    /****** Plot power spectrum for this section of data *****/

    /* itoa(fft, fft_sec, 10);
    strcat(subhead, fft_sec); */
    set_titles( "Frequency", "Energy", "Power Spectrum", subhead );
    set_axes(0, 0.5 * rate, 2, -8, 2, 2, ypixels);
    set_colours(col4, col0, col1, col0); /* last was 3 */

    if( _pg_chartscatter( &env, xvalue, yvalue, nspec ) )
    {
        _setvideomode( _DEFAULTMODE );
        _outtext( "Error: can't draw chart" );
    }

    _settextposition(rows - 1, 10);
    _settextcolor(col7);
    _outtext("Sonic Record Start: ");
    _outtext( gilltime );
    _outtext(" ");
}

```

```

/* Effectively multiply top part of spectrum by f^5/3 */
for (i = 2; i<=nspec; i++) /* i.e. i=96->256 */
{
    a[i] += 1.66666667 * log10((i-1) * rate / nrec) + invsw;
    yvalue[i] = (float) a[i];
}

/* temp insert
ps = fopen("c:\\qc2\\psds", "a+");
fprintf(stdprn, "%+8.5f ", meanpsd(nrec) );
fprintf(ps, "%+8.5f", meanpsd(nrec) );
if ( (fft == 8) || (fft == 15) )
{
    fprintf(stdprn, "\n");
}
fclose(ps);
end of temp insert */

if (gflag > 0)
{
    /****** Plot Spectrum converted to log{a[]*f^5/3} form *****/
    /* (should be near horiz. line) */

    set_axes(0, 0.5 * rate, 2, -8, 2, 2, ypixels); /* xmin was nans4*1.5 */
    strcat(subhead, " (-5/3 Region)");
    set_titles("Frequency", "Energy", "Power Spectrum", subhead);
    set_colours(col5, col0, col1, col0); /* last was 2 */

    if( _pg_chartscatter( &env, xvalue, yvalue, nspec ) )
    {
        _setvideomode( _DEFAULTMODE );
        _outtext( "Error: can't draw chart" );
    }

    show_errors(full_message);
    if (fft < nfft)
    {
        _settextposition(15,40);
        _outtext("Analysing next section");
    }

    strcpy( subhead, "Section: ");
}

/* end of if(getdat.....) block */

else
{
    printf("\a");
}

/* end of fft loop */

fclose(fh);
waitl0;

if (good_reads == 0)
{
    /* Leave graphics screen */
    _setvideomode( _DEFAULTMODE );
    _settextposition(rows / 2, 20);
    printf("FATAL ERROR:- BAD DATA FROM SONIC\n");
}

```

```

        _settextposition(2 + rows / 2, 25);
        printf("Please inform IOSDL\n");
        exit(0);
    }

    /* Correct power estimates for windowing, etc */

    den = sumw * nr2 * good_reads;          /* corrected sumw 11/02/92 */

    for (j = 1; j <= nspec; j++)
    {
        p[j] /= den;
    }

    mean /= good_reads;

    good_total = (unsigned) good_reads * nr2;

    north_mean /= good_total;
    east_mean /= good_total;
    vert_mean /= good_total;
    c_mean /= good_total;

    _settextposition(rows, 10);
    printf("                ");
    show_errors(full_message);
    waitl();

    for (i = 2; i <= nspec; i++)
    {
        fact = pow((float) (i - 1) * rate / nrec, 1.66666667) * nrec / rate;
        p[i] *= fact;
        a[i] = log10(p[i]);
        yvalue[i] = a[i];
    }

    /* wipe_line();
    printf("dc compt=%f press key to cont", p[1]);
    getch();
    above temp */

    if ( ((f_out = fopen(spec_file, "a+")) == NULL)
        || (fseek(f_out, 0L, SEEK_END) != NULL) )
    {
        wipe_line();
        printf("Failed to open C: .PRN File for full Spectrum\n");
        fclose(f_out);
    }
    else
    {
        wipe_line();
        printf("Writing Spectrum to File %s", spec_file);
        fprintf(f_out, "\nFrequency (Hz)\n\F%s : MWS %5.2f m/s\n", julian, mean);

        for (i = 2; i <= nspec; i++)
        {
            fprintf(f_out, "%5.2f,%6.3f\n",
                    (float) (i - 1) * rate / nrec, yvalue[i]);
        }
    }

```

```

        fclose(f_out);
    }

    if ( ((fll_out = fopen(spec_fllfile, "a+")) == NULL)
        || (fseek(fll_out, 0L, SEEK_END) != NULL) )
    {
        wipe_line();
        printf("Failed to open A: .PRN File for full Spectrum\n");
        fclose(fll_out);
    }
    else
    {
        wipe_line();
        printf("Writing Spectrum to File %s", spec_fllfile);
        fprintf(fll_out, "\nFrequency (Hz)\n", "F%s : MWS %5.2f m/s\n", julian, mean);

        for (i = 2; i <= nspec; i++)
        {
            fprintf(fll_out, "%5.2f,%6.3fn",
                    (float) (i - 1) * rate / nrec, yvalue[i]);
        }

        fclose(fll_out);
    }

    if (gflag > 0)
    {
        /***** Plot Mean Spectrum *****/

        set_axes(0, 0.5 * rate, 2, -8, 2, 2, ypixels);
        set_titles("Frequency", "Energy", "Mean Spectrum", "log(P(I%)*f^5/3)");
        set_colours(col2, col0, col1, col0); /* last was 7 */

        if( _pg_chartsdatter( &env, xvalue, yvalue, LINES ) )
        {
            _setvideomode( _DEFAULTMODE );
            _outtext( "Error: can't draw chart" );
        }
    }
    show_errors(full_message);
    waitl();

    /* Fit regression line */
    psd = regres(nrec, &r, &rmsl, &sea, &seb, &a1, &b);

    psd = log10(psd);
    sea = log10(fabs(sea));
    seb = log10(fabs(seb));

    if (a1 > 0.)
    {
        a1 = log10(a1);
    }
    else
    {
        a2 = a1;
        a1 = 9999.;
    }

    /* code for output of psd and mean to formatter here */
    /* code for storing fit to eprom logger here */

```

```

_settextposition(rows - 1,10);

if ( ( f_out = fopen(spec_file, "a+") == NULL )
      || ( fseek(f_out, 0L, SEEK_END) != NULL ) )
{
    wipe_line();
    printf("Failed to open Drive C: .PRN File for Parameters\n");
    fclose(f_out);
}
else
{
    wipe_line();
    printf("Writing PSD to File %s", spec_file);
    fprintf(f_out, "%sPSDSpd\n", julian);
    fprintf(f_out, "%s\n%.3d\n%.5.2f %.5.2f\n%s\n",
            sonic_id, good_reads, freq1, freq2, son_mode);
    fprintf(f_out,
            "%.5.2f\n%.6.2f\n%.6.2f\n%.6.2f\n%.8.5f\n%.8.5f\n%.11.5e\n",
            mean, north_mean, east_mean, vert_mean, c_mean, psd, a1,
b);
    fclose(f_out);
}

if ( ( ffil_out = fopen(spec_ffile, "a+") == NULL )
      || ( fseek(ffil_out, 0L, SEEK_END) != NULL ) )
{
    wipe_line();
    printf("Failed to open Drive A: .PRN File for Parameters\n");
    fclose(ffil_out);
}
else
{
    wipe_line();
    printf("Writing PSD to File %s", spec_ffile);
    fprintf(ffil_out, "%sPSDSpd\n", julian);
    fprintf(ffil_out, "%s\n%.3d\n%.5.2f %.5.2f\n%s\n",
            sonic_id, good_reads, freq1, freq2, son_mode);
    fprintf(ffil_out,
            "%.5.2f\n%.6.2f\n%.6.2f\n%.6.2f\n%.8.5f\n%.8.5f\n%.11.5e\n",
            mean, north_mean, east_mean, vert_mean, c_mean, psd, a1, b);
    fclose(ffil_out);
}

if ( ( fff_out = fopen(spec_fff_file, "a+") == NULL )
      || ( fseek(fff_out, 0L, SEEK_END) != NULL ) )
{
    wipe_line();
    printf("Failed to open Drive C: .MWS File for Parameters\n");
    fclose(fff_out);
}
else
{
    wipe_line();
    printf("Writing PSD to File %s", spec_fff_file);
    fprintf(fff_out, "%sPSDSpd\n", julian);
    fprintf(fff_out, "%s\n%.3d\n%.5.2f %.5.2f\n%s\n",
            sonic_id, good_reads, freq1, freq2, son_mode);
    fprintf(fff_out,
            "%.5.2f\n%.6.2f\n%.6.2f\n%.6.2f\n%.8.5f\n%.8.5f\n%.11.5e\n",
            mean, north_mean, east_mean, vert_mean, c_mean, psd, a1, b);
    fclose(fff_out);
}

```

```

if ( ( fl_out = fopen(spec_ffile, "a+") ) == NULL )
    || ( fseek(fl_out, 0L, SEEK_END) != NULL ) )
{
    wipe_line();
    printf("Failed to open Drive A: .MWS File for Parameters\n");
    fclose(fl_out);
}
else
{
    wipe_line();
    printf("Writing PSD to File %s", spec_ffile);
    fprintf(fl_out, "%sPSDSpd\n", julian);
    fprintf(fl_out, "%s\n%.3d\n%.5.2f %.5.2f\n%s\n",
            sonic_id, good_reads, freq1, freq2, son_mode);
    fprintf(fl_out,
            "%5.2f\n%.6.2f\n%.6.2f\n%.6.2f\n%.6.2f\n%.8.5f\n%.8.5f\n%.11.5e\n",
            mean, north_mean, east_mean, vert_mean, c_mean, psd, a1, b);

    fclose(fl_out);
}

if (gflag > 0)
{
    scale_x = (float) (env.datawindow.x2 - env.datawindow.x1) / LINES;
    scale_y = (float) (env.datawindow.y2 - env.datawindow.y1) / 10;

    fr1 = 1 + (freq1 * nrec / rate);
    fr2 = 1 + (freq2 * nrec / rate);

    if (a1 != 9999.)
    {
        p1 = log10(pow(10.,a1) + b * fu[(int) fr1]);
        p2 = log10(pow(10.,a1) + b * fu[(int) fr2]);
    }
    else
    {
        /* _settextposition(8, 35);
        _outtext("Can't plot regression line (a -ve)"); */
        p1 = a2 + b * fu[(int) fr1];
        p2 = a2 + b * fu[(int) fr2];
        if ( (p1 > 0.) && (p2 > 0.) )
        {
            p1 = log10(p1);
            p2 = log10(p2);
        }
        else
        {
            wipe_line();
            printf("Can't plot regression line (p -ve)");
        }
    }
}

/***** Plot fitted line over range freq1 - freq2 *****/

_setcolor(col8);
_moveto( env.datawindow.x1 + (short) (scale_x * fr1),
        yoffset + env.datawindow.y1 - (short) ((p1 - 2) * scale_y) );
_lineto( env.datawindow.x1 + (short) (scale_x * fr2),
        yoffset + env.datawindow.y1 - (short) ((p2 - 2) * scale_y) );

```

```

/* draw dashed lines to show region fr1->fr2 */
_setlinestyle( 0xAAAA);
_moveto( env.datawindow.x1 + (short) (scale_x * fr1),
        yoffset + env.datawindow.y1);
_lineto( env.datawindow.x1 + (short) (scale_x * fr1),
        yoffset + env.datawindow.y2);
_moveto( env.datawindow.x1 + (short) (scale_x * fr2),
        yoffset + env.datawindow.y1);
_lineto( env.datawindow.x1 + (short) (scale_x * fr2),
        yoffset + env.datawindow.y2);
_setlinestyle( 0xFFFF);

}

wait1();
wipe_line();

if (gflag > 0)
{
    meanws_set[sample] = mean;
    psd_set[sample] = psd;

    /* Reset Chart Format to Scatter Diagram */
    _pg_defaultchart( &env, _PG_SCATTERCHART, _PG_POINTONLY );
    set_envconsts(ypixels, col0);
    _pg_getpalette( palette_struct );
    palette_struct[1].plotchar = 249;    /* 249 plots ~ chars */
    _pg_setpalette( palette_struct );

    /***** Plot Scatter Diagram of PSD vs MWS *****/

    set_titles("Mean Wind Speed (m/s)", "PSD",
              "Scatter Plot of Last 100 Results", strcat(head, gilltime) );
    set_axes(0, 40, 10, -8, 2, 2, ypixels);
    set_colours(col6, col0, col1, col0);    /* last was 3 */

    if( _pg_charts( &env, meanws_set, psd_set, 100 ) )
    {
        _setvideomode( _DEFAULTMODE );
        _outtext( "Error: can't draw chart" );
    }

    strcpy(head, "Last Start");
    scale_x = (float) (env.datawindow.x2 - env.datawindow.x1) / 40;
    scale_y = (float) (env.datawindow.y2 - env.datawindow.y1) / 10;

    /***** Overplot Most Recent Point with a red cross *****/

    do_cross( env.datawindow.x1 + (short) (scale_x * mean),
              yoffset + env.datawindow.y1 - (short) ((psd - 2) * scale_y), col3 );
    show_errors(full_message);

}

_settextcolor(col7);
_settextposition(rows - 3, 0);
printf("Mean WS=%5.2fm/s, PSD*f^5/3=%+5.2f (%d-%dHz), Fit=%+5.2f %+10.2e*x ",
        mean, psd, (int) freq1, (int) freq2, a1, b);

_settextposition(rows - 2, 0);
printf("(N=%+6.2fE=%+6.2fV=%+6.2fm/s) (sea=%+4.2f,seb=%+4.2f), R=%+5.2f,\"

```

```

        RMS=%+10.2e\n", north_mean, east_mean, vert_mean, sea, seb, r, rms1);
/* printf("Sample No: %d\n", sample); */

/*****TEMPORARY INSERT*****/
fprintf(stdprn, "Mean %f\n", psd);
*****/

    waitl();

}          /* end of re loop */

fclose(f_out);
first = 1;

/* wait_start(rows, &sample); */

/* Leave graphics screen */
_setvideomode( _DEFAULTMODE );
_settextposition(rows - 1,0);

exit(0);

}          /* end of main function */

/***** START OF FUNCTION DEFINITIONS *****/

/***** FFTCONSTS sets parameters for fit *****/

float fitconsts(int nrec, int nspec, int *nr2, int *nr23, int *nr24,
               float *fm, float *fp)
{
    int j;
    float sumw, w;
    float alpha = 31.41592654/nrec;

    *nr2 = nrec * 2;
    *nr24 = *nr2 + 4;
    *nr23 = *nr2 + 3;

    *fm = nspec - .5;
    *fp = 1/(nspec + .5);

    sumw = 0.;
    /* Calculate sum of weights for Parsen window
    for (j = 1; j <= nrec; j++)
    {
        w = 1 - fabs(((j - 1) - *fm) * *fp);
        sumw += w * w;
    }
    */

    /* Calculate weights for partial cosine taper */

    for (j = 1; j <= LINES; j++)
    {
        if (j <= nrec/10)
        {
            sumw += 0.5 * pow(1 + cos(alpha * (LINES - j)), 2.);
        }
        else
        {
            sumw += 2.;
        }
    }
}

```

```

    }
}

return sumw;
}

/***** GETDAT loads data from diskfile *****/

int getdat(int son_mode, int ffit, int nr2, char ram_file[30], char ch[40],
          float *addr_dummy, float *addr_north_mean,
          float *addr_east_mean, float *addr_vert_mean, float *addr_c_mean)
{
double av1, av2, av3, c, cr, sr, u, v, w, res;
/* for ship system cr = cos(.5236), sr = sin(.5236) */
double lf_ck = 0.5 * 0.149 * 29491200;
/* = half path length ( in m ) * counting clock frequency */

float north_start, east_start, vert_start, c_start;

int dec, sign;
int e_flag = 0;
int i, j, k, theta;
int dbuff[12];
/* In mode1, dbuff holds 8 bytes velocity: 4 off (msbyte,lsbyte) */
/* These are 100*u, 100*v, 100*w, 50*c in m/s */
/* In mode4, dbuff holds 12 bytes transit times:6 of (msbyte,lsbyte) */

unsigned int t11, t12, t21, t22, t31, t32;
unsigned comp;

if ( ffit == 1 )
{
    if ( (fh = fopen(ram_file, "rb")) != 0 )
    {
        /* read header */
        for (i = 0; i < 5; i++)
        {
            fgetc(fh);
        }
        if ( fgetc(fh) != 48 + son_mode )
        {
            printf("\a"); /* beep */
            return(1); /* abort if incorrect mode */
        }

        for (i = 0; i < 8; i++) /* should be 7 if no CR */
        {
            fgetc(fh);
        }
        if (fgetc(fh) != '1')
        {
            return(1); /* abort if no analog channel present */
        }

        fgetc(fh);

        i=0;
        do
        {
            ch[i] = fgetc(fh);
            i++;
        }
    }
}

```

```

while ( (ch[i - 1] != 0x0a) && (i < 40) );

ch[i-1] = 0; /* replace LF with string terminator */
/* Resulting string is:
   "Time hh:mm:ss Date mm/dd/yy" */

}
else
{
wipe_line();
printf("Could not open file\n");
return(1);
}

} /* end of block for fft == 1 */

/* for all values of fft */

north_start = *addr_north_mean;
east_start = *addr_east_mean;
vert_start = *addr_vert_mean;
c_start = *addr_c_mean;

for (j = -1; j <= 0; j++)
{
for (i = j+2; i <= nr2; i += 2)
{
switch (son_mode)
{
case 1:
{
/* get 3 * 2byte vel compts plus 1 * 2byte vel of sound
   plus 1 * 2byte compass reading */
for (k = 0; k <= 9; k++)
{
dbuf[k] = fgetc(fh);
if (ferror(fh) != 0)
{
return(2);
}
}

if ( (dbuf[7] != 0) || (dbuf[6] != 0) ) /* vel of sound != 0 */
{
/* convert from motorola format to int format */
u = 0.01 * (int) (dbuf[1] + (dbuf[0] << 8));
v = 0.01 * (int) (dbuf[3] + (dbuf[2] << 8));
w = 0.01 * (int) (dbuf[5] + (dbuf[4] << 8));
c = 0.02 * (int) (dbuf[7] + (dbuf[6] << 8));
comp = (dbuf[9] >> 3) + (dbuf[8] << 5) - 256;
/* compass 8 bit value = (sonic count - 2048) / 8
   i.e. (sonic lsbyte / 8) + (sonic msbyte * 256 / 8) - 256 */

if (comp < 0)
{
comp = 0;
}
if (comp > 255)
{
comp = 255;
}
if ( (i > 1) && (abs(comp - last_comp) > 64)
    && (abs(comp - last_comp) < 192) )

```

```

        {
            comp = 0;
        }
        /* NB comp range 0-255 */

        last_u = u;
        last_v = v;
        last_w = w;
        last_c = c;
        last_comp = comp;
    }
    else
    {
        u = last_u;
        v = last_v;
        w = last_w;
        c = last_c;
        comp = last_comp;
        e_flag = 1;
    }

    if ((u == -100) || (v == -100) || (w == -100))
    {
        return(2);          * path was blocked on 1 or more axes */
    }

    sr = sin(.5236 - 0.0245437 * comp);
    cr = cos(.5236 - 0.0245437 * comp);
    break;
}                               /* end of case 1 */

case 4:
{
    /* get 6 * 2byte transit counts (not applicable any more) */
    for (k = 0; k <= 11; k++)
    {
        dbuff[k] = fgetc(fh);
        if (ferror(fh) != 0)
        {
            return(2);
        }
    }

    /* convert from motorola format to int format */
    t11 = dbuff[1] + (dbuff[0] << 8);
    t12 = dbuff[3] + (dbuff[2] << 8);
    t21 = dbuff[5] + (dbuff[4] << 8);
    t22 = dbuff[7] + (dbuff[6] << 8);
    t31 = dbuff[9] + (dbuff[8] << 8);
    t32 = dbuff[11] + (dbuff[10] << 8);

    /* calculate axis velocities */
    av1 = lf_ck / t11 - lf_ck / t12;
    av2 = lf_ck / t21 - lf_ck / t22;
    av3 = lf_ck / t31 - lf_ck / t32;

    /* convert to u, v, w velocities */
    u = .471409 * (2 * av1 - av2 - av3);
    v = .816527 * (av2 - av3);
    w = .471409 * (-av1 - av2 - av3);
    c = 0.333333 * (lf_ck / t11 + lf_ck / t12 + lf_ck / t21
                    + lf_ck / t22 + lf_ck / t31 + lf_ck / t32);

```

```

        /* apply calibrations */

        if (u == 0)
        {
            u += .001;
        }
        theta = (int) (57.3 * atan2(v, u) - 30);

        if (theta < 0)
        {
            theta += 360;
        }

        u = u * x_calibration_table[theta] / 65536;
        v = v * y_calibration_table[theta] / 65536;
        if (w > 0)
        {
            w = w * away_z_calibration_table[theta] / 65536;
        }
        else
        {
            w = w * into_z_calibration_table[theta] / 65536;
        }
    } /* end of case 4 */
    break;
} /* end of switch */

*addr_north_mean += (u * cr + v * sr);
*addr_east_mean += (v * cr - u * sr);
*addr_vert_mean += w;
/* above values are vector averaged north, east and vertical compts */

*addr_c_mean += c;

/* put resultant horiz vel. in array a[] (start address a_ptr) */
a[i] = sqrt(u * u + v * v + w * w); /* added w^2 term 5 7 93 */
    } /* end of i loop */
} /* end of j loop */
if (e_flag == 1)
{
    *addr_north_mean = north_start;
    *addr_east_mean = east_start;
    *addr_vert_mean = vert_start;
    *addr_c_mean = c_start;
    return(3);
}
else
{
    return(0);
}
}

/* returns 0 if ok
1 if failure to open file or header incorrect
2 if error during read
3 if vel of sound is zero (dummy data due to txn error)
(also array of nr2 resultant wind speeds at a_ptr) */

/***** DCFILTER removes mean from data *****/

```

```

float dcfilter(int nr2)
{
    int i;
    float tot = 0;

    for (i = 1; i <= nr2; i++)
    {
        tot += a[i];
    }
    tot = tot/nr2;
    for (i = 1; i <= nr2; i++)
    {
        a[i] -= tot;
    }

    return tot;
}

```

```

/***** WINDOW applies Parsen data window *****/
/*
void window(int nrec, float fm, float fp)
{
    int j, j2;
    float w;

    for (j = 1; j <= nrec; j++)
    {
        j2 = 2*j;
        w = 1 - fabs((j-1-fm)*fp);
        a[j2] *= w;
        a[j2-1] *= w;
    }
}
*/

```

```

/***** WINDOW applies partial cosine data window *****/

void window(int nrec, float fm, float fp)
{
    int j, j2, nr2 = 2 * nrec;
    float alpha = 31.41592654/nrec, w;

    for (j = 1; j <= nrec/2; j++)
    {
        if (j <= nrec/10)
        {
            j2 = 2 * j;
            w = 0.5 * (1 + cos(alpha * (LINES - j)));
            a[j2] *= w;
            a[j2 - 1] *= w;
            a[nr2 - j2 + 1] *= w;
            a[nr2 - j2 + 2] *= w;
        }
    }
}

```

```

/***** FOUR1 does fft *****/

```

```

void four1(int nrec)
{
    int i, j = 1, l, m, n = 2 * nrec, s;
    double tr, ti, te, t, wpr, wpi, wr, wi, wt;

    for (i = 1; i <= n; i += 2)
    {
        if (j > i)
        {
            tr = a[j];
            ti = a[j + 1];
            a[j] = a[i];
            a[j + 1] = a[i + 1];
            a[i] = tr;
            a[i + 1] = ti;
        }
        m = (int) n/2;

        while ((m >= 2) && (j > m))
        {
            j -= m;
            m /= 2;
        }
        j += m;
    }

    l = 2;
    while (n > l)
    {
        s = 2 * l;
        t = 2 * pi/l;
        te = sin(.5*t);
        wpr = -2 * te * te;
        wpi = sin(t);
        wr = 1;
        wi = 0;

        for (m = 1; m <= l; m += 2)
        {
            for (i = m; i <= n; i += s)
            {
                j = i + l;
                tr = wr * a[j] - wi * a[j + 1];
                ti = wr * a[j + 1] + wi * a[j];
                a[j] = a[i] - tr;
                a[j + 1] = a[i + 1] - ti;
                a[i] += tr;
                a[i + 1] += ti;
            }
            wt = wr;
            wr += wr * wpr - wi * wpi;
            wi += wi * wpr + wt * wpi;
        }
        l = s;
    }
}

```

```

/***** REGRES fits regression line *****/

```

```

float regres(int nrec, float *r, float *rmsl, float *sea, float *seb,
            double *a1, double *b)
{
    int i, i1, i2, n;
    float psd, xm, xn, ym, ynl;
    /* have to use ynl as yn appears to be in the include files */

    double sx = 0, sy = 0, sxx = 0, sxy = 0, syy = 0, ssa, ssb, ssr;

    i1 = 1 + (freq1 * nrec / rate);
    i2 = 1 + (freq2 * nrec / rate);
    psd = 0;
    n = 0;
    for (i = i1; i <= i2; i++)
    {
        psd += p[i];
        n++;
    }
    psd /= n;                                /* mean PSD over range freq1 to freq2 */

    xm = fu[1];                               /* tried [i1] */
    ym = p[1];                                /* ***** */
    n--;

    for (i = i1 + 1; i <= i2; i++)
    {
        xn = fu[i] - xm;
        ynl = p[i] - ym;
        sx = sx + xn;
        sy = sy + ynl;
        sxx = sxx + xn * xn;
        sxy = sxy + xn * ynl;
        syy = syy + ynl * ynl;
    }

    sxx = sxx - (sx * sx) / (double) n;
    sxy = sxy - (sx * sy) / (double) n;
    xm = xm + sx / (double) n;
    syy = syy - (sy * sy) / (double) n;

    *a1 = ym + sy / (double) n;
    *b = sxy / sxx;
    ssa = *a1 * *a1 * (double) n;
    ssb = *b * sxy;
    ssr = syy - ssb;
    *a1 = *a1 - *b * xm;
    *rmsl = ssr / (double) (n - 2);

    if (*rmsl < 0)
    {
        /* printf("RMS negative (%e)- is data OK?\n", *rmsl); */
        *rmsl = 0;
        *r = 0;
        *sea = 10000;
        *seb = 10000;
    }
    else
    {
        *r = (float) (sxy / sqrt(sxx * syy));
        *sea = (float) sqrt((double) *rmsl / (double) n);
    }
}

```

```

        *seb = (float) sqrt( (double) *rmsl / sxx );
        *rmsl = (float) sqrt( (double) *rmsl );
    }
    return psd;
}

```

/****** SET_TITLES sets titles for PGgraphics plot *****/

```

void set_titles( char *x_title, char *y_title,
                char *main_title, char *sub_title )
{
    strcpy(env.xaxis.axistitle.title, x_title);
    strcpy(env.yaxis.axistitle.title, y_title);
    strcpy(env.maintitle.title, main_title);
    strcpy(env.subtitle.title, sub_title);
}

```

/****** SET_AXES sets axes for PGgraphics plot *****/

```

void set_axes( float x_min, float x_max, float x_ticint,
              float y_min, float y_max, float y_ticint, int ypixels)
{
    _setviewport(0, 0, 639, ypixels - 65);
    _clearscreen(_GVIEWPORT);
    env.xaxis.scalemin = x_min;
    env.xaxis.scalemax = x_max;
    env.xaxis.ticinterval = x_ticint;
    env.yaxis.scalemin = y_min;
    env.yaxis.scalemax = y_max;
    env.yaxis.ticinterval = y_ticint;
}

```

/****** SET_COLOURS sets colours for PGgraphics plot *****/

```

void set_colours( short border_colour, short window_colour,
                 short maintitle_colour, short subtitle_colour )
{
    env.chartwindow.background = border_colour;
    env.datawindow.background = window_colour;
    env.maintitle.titlecolor = maintitle_colour;
    env.subtitle.titlecolor = subtitle_colour;
}

```

/****** DO_CROSS plots a cross *****/

```

void do_cross( int xc, int yc, short col )
{
    _setcolor(col);
    _moveto(xc - 10, yc);
    _lineto(xc + 10, yc);
    _moveto(xc, yc - 10);
    _lineto(xc, yc + 10);
}

```

/****** WAIT_START waits for start of next process *****/

```

char * wait_start(int rows, int * sample_no, int * qtr)

```

```

{
char cur_time[10], julian[10], last_time[10];
div_t quarters;
int sample;
time_t tnow;
struct tm *gmt;

_settextposition(rows - 1,10);
_outtext("Waiting for next Record Start . . .");
do
{
    if ( kbhit() )
    {
        if (getch() == 125)                /* abort tidily if } key pressed */
        {
            exit(0);
        }
    }

    time(&tnow);
    gmt = gmtime(&tnow);
    quarters = div(gmt->tm_min, 15);
    _strtime(cur_time);
    if (cur_time[7] != last_time[7])
    {
        _settextposition(rows - 1,46);
        printf("Day %d: %s", 1 + gmt->tm_yday, cur_time);
    }
    strcpy(last_time, cur_time);
}
while ( (gmt->tm_sec != 0) && (gmt->tm_sec != 15) && (gmt->tm_sec != 30) && (gmt->tm_sec != 45));
/* while (gmt->tm_sec != 0); */

strcpy( julian, aform((1 + gmt->tm_yday), 3) );
strcat( julian, aform((gmt->tm_hour), 2) );
strcat( julian, aform((gmt->tm_min), 2) );

/* tnow /= 60;   temp mod to allow consec sampling */
tnow /= 900;
/* current time in 1/4 hrs since 00:00:00 Jan 1, 1970 */
sample = (int) (tnow % (long) 100);
/* sample runs from 0 to 499 (cyclically) 1/4 hrly */

*sample_no = sample;
*qtr = quarters.quot;

return julian;
}

/***** SET_ENVCONSTS sets envt for PGgraphics plot *****/

void set_envconsts(int ypixels, int col0)
{
env.chartwindow.border = FALSE;
env.xaxis.autoscale = FALSE;
env.yaxis.autoscale = FALSE;
env.chartwindow.x1 = 0;
env.chartwindow.y1 = yoffset;
env.chartwindow.x2 = 639;
env.chartwindow.y2 = ypixels - 65;
env.xaxis.axistitle.titlecolor = col0;
env.yaxis.axistitle.titlecolor = col0;

```

```

}
```

```

/***** WAIT1 waits for 1/2 second *****/
```

```

void wait1()
{
    clock_t tnow, tnext;

    tnow = clock();

    do
    {
        tnext = clock();
    }
    while ( (tnext - tnow)/CLK_TCK <= 0.5 );
}

```

```

/***** AFORM formats a number in specified format *****/
```

```

char * aform(int i_var, int n_char)
{
    char asc_var[4] = "000", temp[3];
    int l_var;
    if (i_var <= 0 )
    {
        asc_var[n_char] = '\0';
        return asc_var;
    }
    else
    {
        l_var = (int) ( 1 + log10(i_var) );
        if ( ((n_char - l_var) < 4) && ((n_char - l_var) > -1) )
        {
            itoa(i_var, temp, 10);
            strcpy(asc_var + n_char - l_var, temp );
        }
    }
    return asc_var;
}

```

```

/***** HARDERROR_HANDLER handles hardware errors *****/
```

```

void harderror_handler(unsigned deverror, unsigned errcode, unsigned *devhdr)
{
    char cletter, num[5];
    error_flag = 1;

    if (strlen(message) > 40)
    {
        strcpy(message, "");
    }

    if ( (deverror & 0x8000) == 0)
    {
        switch(deverror & 0xff)
        {
            case 0:
                strcat(message, "Drive A ");
                break;
            case 1:

```

```

        strcat(message, "Drive B ");
        break;
    case 2:
        strcat(message, "Drive C ");
        break;
    }

    strcat(message, " ERROR:-");

    itoa(errcode & 0xff, num, 10);
    switch(errcode & 0xff)
    {
    case 0:
        strcat(message, " Write Prot'd");
        break;
    case 2:
        strcat(message, " Not Ready");
        break;
    case 10:
        strcat(message, " Write Fault");
        break;
    case 12:
        strcat(message, " Gen Failure");
        break;
    default:
        strcat(message, " Code ");
        strcat(message, num);
        break;
    }
    switch(deverror & 0x0600)
    {
    case 0:
        strcat(message, "-MSDOS: ");
        break;
    case 0x0200:
        strcat(message, "-FAT: ");
        break;
    case 0x0400:
        strcat(message, "-Directory: ");
        break;
    case 0x0600:
        strcat(message, "-Data Area: ");
        break;
    }
    }
else
{
    strcpy(message, "Non Disk I/O Error: ");
    if( (*(devhdr + 4) & 0x8000) == 0)
    {
        strcat(message, "Bad Image of FAT: ");
    }
    else
    {
        strcat(message, "Character Device: ");
    }
    }
_hardretn(_HARDERR_IGNORE);
}

/***** CHECK_DISKS checks for space for 1 more file *****/

void check_disks(int first, char *full_message, char *spec_file,

```

```

char *spec_ffile, char *spec_ffile, char *spec_fffile,
FILE *f_out, FILE *ff_out, FILE *fl_out, FILE *ffl_out)
{
char dletter[2];
int drive;
int disk_flag = 0;
struct diskfree_t dfinfo;
unsigned long free_space, safe_limit;

strcpy(full_message, "");

for (drive = 0; drive < 2; drive++)
{
    if (drive == 0)
    {
        strcpy(dletter, "C");
        safe_limit = 60000;
    }
    if (drive == 1)
    {
        strcpy(dletter, "A");
        safe_limit = 60000; /* now save .PRN to floppy also */
    }

    if (_dos_getdiskfree(drive, &dfinfo) != 0)
    {
        wipe_line();
        printf("Error in _dos_getdiskfree, Drive %s\n", dletter);
        exit(0);
    }
    free_space = (unsigned long) dfinfo.avail_clusters
                 * dfinfo.sectors_per_cluster
                 * dfinfo.bytes_per_sector;
    if ((free_space < safe_limit * 2) && (free_space >= safe_limit))
    {
        strcat(full_message, "Drive ");
        strcat(full_message, dletter);
        strcat(full_message, ": NEARLY FULL: ");
    }
    if (free_space < safe_limit)
    {
        strcat(full_message, "Drive ");
        strcat(full_message, dletter);
        strcat(full_message, ": NO FREE SPACE: ");

        disk_flag = disk_flag | (1 + drive);
    }
}

if ((disk_flag & 0x03) == 3)
{
    _clearscreen(_GOCLEARSCREEN);
    _settextposition(10, 12);
    printf("*****WARNING: No Disk Space free on A: or C:*****\n");
    printf("*****Program Terminated*****\n");
    exit(0);
}

if (first == 1)
{
    strcpy(spec_file, "c:\\data\\");
    strcat(spec_file, julian);
    /* hard disk file */
}

```

```

    strcpy(spec_ffile, spec_file);
    strcat(spec_file, ".pm");
    strcat(spec_ffile, ".mws");
    strcpy(spec_ffile, "a:\");
    strcat(spec_ffile, julian);
    strcpy(spec_fffile, spec_ffile);
    strcat(spec_ffile, ".mws");
    strcat(spec_fffile, ".pm");
}

if ( (f_out = fopen(spec_file, "a")) == NULL )
{
    disk_flag = disk_flag | 4;
}
else
{
    fclose(f_out);
}

if ( (ff_out = fopen(spec_ffile, "a")) == NULL )
{
    disk_flag = disk_flag | 4;
}
else
{
    fclose(ff_out);
}

if ( (fl_out = fopen(spec_ffile, "a")) == NULL )
{
    disk_flag = disk_flag | 8;
}
else
{
    fclose(fl_out);
}

if ( (fll_out = fopen(spec_fffile, "a")) == NULL )
{
    disk_flag = disk_flag | 8;
}
else
{
    fclose(fll_out);
}

if ( (disk_flag & 0x0c) == 12 )
{
    _clearscreen(_GCLLEARSCREEN);
    system("mode 80");
    printf("*****Cannot open Output Files on Drive C: or A:*****\n");
    printf("*****Program Terminated*****\n");
    exit(0);
}

}

/***** SHOW_ERRORS displays error messages *****/
void show_errors(char * full_message)
{
    int old_col = _gettextcolor();

    _settextcolor(15);
    _settextposition(0,0);

```

```

_outtext(full_message);
_outtext(message);
_settextcolor(old_col);
}

/***** SETUP loads configuration from setup.son file *****/
int setup(char *sonic_id)
{
char line[80], head[20], data[10];
FILE *fs;
int i, gflag, j;

if ( (fs = fopen("setup.son", "r")) == NULL)
{
wipe_line();
printf("Could not open SETUP.SON File \n");
return 0;
}

fscanf(fs, "%s %s\n", head, data);
if (strcmpi(data, "ON") == 0)                                /* case insensitive */
{
gflag = 1;
}
else
{
gflag = 0;
}

fscanf(fs, "%s %s\n", head, ser_no);
strcpy(sonic_id, ser_no);
fscanf(fs, "%s %s\n", head, data);
freq1 = (float) atof(data);
fscanf(fs, "%s %s\n", head, data);
freq2 = (float) atof(data);
fscanf(fs, "%s %s\n", head, data);
recs_per_file = atoi(data);
fscanf(fs, "%s %s\n", head, data);
q[0] = atoi(data);
fscanf(fs, "%s %s\n", head, data);
q[1] = atoi(data);
fscanf(fs, "%s %s\n", head, data);
q[2] = atoi(data);
fscanf(fs, "%s %s\n", head, data);
q[3] = atoi(data);
fscanf(fs, "%s %s\n", head, data);

if (strcmpi(data, "ON") == 0)                                /* case insensitive */
{
raw_optical_save = 1;
}
else
{
raw_optical_save = 0;
}

fclose(fs);

if( (q[0] == 4) || (q[1] == 4) || (q[2] == 4) || (q[3] == 4) )
{
strcat(ser_no, "rcal.h");
if ( (fs = fopen(ser_no, "r")) == NULL)

```

```

    {
        wipe_line();
        printf("Could not open Sensor Calibration File\n");
        exit(0);
    }
else
    {
        fgets(line, 80, fs);
        fgets(line, 80, fs);

        for (i = 0; i < 360; i += 10)
        {
            fgets(line, 80, fs);

            for (j = 0; j < 10; j++)
            {
                x_calibration_table[i + j] = atol(line + 6 * j);
                if ( (x_calibration_table[i+j] > 100000) || (x_calibration_table[i+j] < 50000) )
                {
                    wipe_line();
                    printf("Error in reading x_calibration table\n");
                    printf("x_cal=%ld\n", x_calibration_table[i+j]);
                    exit(0);
                }
            }
        }

        x_calibration_table[360] = x_calibration_table[0];
        fgets(line, 80, fs);
        fgets(line, 80, fs);
        fgets(line, 80, fs);
        for (i = 0; i < 360; i += 10)
        {
            fgets(line, 80, fs);

            for (j = 0; j < 10; j++)
            {
                y_calibration_table[i + j] = atol(line + 6 * j);
                if ( (y_calibration_table[i+j] > 100000) || (y_calibration_table[i+j] < 50000) )
                {
                    wipe_line();
                    printf("Error in reading y_calibration table\n");
                    printf("y_cal=%ld\n", y_calibration_table[i+j]);
                    exit(0);
                }
            }
        }

        y_calibration_table[360] = y_calibration_table[0];
        fclose(fs);
    }
} /* end of if( q[...] statements */
return gflag;
}

/***** CHECK_CACHE opens raw data copy file *****/
int check_cache(char *julian, char *raw_filename, FILE *f_cache)
{
    strcpy(raw_filename, "f.F");
    strcat(raw_filename, julian);
    strcat(raw_filename, ".raw");

    if ( (f_cache = fopen(raw_filename, "w+")) == NULL )
    {

```

Brook Road, Wormley, Godalming
Surrey, GU8 5UB,
United Kingdom
Telephone +44 (0) 428-684141
Facsimile +44 (0) 428-683066
Telex 858833 OCEANS G

