FLG

Institute of
Oceanographic Sciences
Deacon Laboratory

INTERNAL DOCUMENT No. 341

SWALES Directional Waverider data report

C H Clayson

1994

Natural Environment Research Council

INSTITUTE OF OCEANOGRAPHIC SCIENCES
DEACON LABORATORY

INTERNAL DOCUMENT No. 341

SWALES Directional Waverider data report

C H Clayson

1994

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

DOCUMENT DATA SHEET

AUTHOR	PUBLICATION DATE
CLAYSON, C H	1994

TITLE

SWALES Directional Waverider data report.

REFERENCE

Institute of Oceanographic Sciences Deacon Laboratory, Internal Document, No. 341, 36pp.
(Unpublished manuscript)

ABSTRACT

During the SWALES experiment in the autumn of 1993, a Directional Waverider Buoy (DWR) was deployed as part of an array of moored instrumentation. Directional spectra were logged on the buoy's EEPROM logger at half-hourly intervals; the same data were telemetered in near real time via the polar orbiting ARGOS satellite data collection system.

This data report briefly describes the processes employed in acquisition of the data. It then describes the processes for the recovery of the data from the two source media, the quality control procedures applied and, finally, the resulting output data files.

Appendices include comprehensive details of the software developed for the above processes and of the formats used for the input and output data.

KEYWORDS

ISSUING ORGANISATION

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: **The Library,** *PRICE* *£0.00*

Index

SWALES Directional Waverider Data Report

Equipment

Buoy

The buoy used during the SWALES experiment (ref. 1) was a standard Datawell Directional Waverider (DWR), equipped with ARGOS transmitter and EEPROM logger. The buoy serial number was 30015 and the ARGOS ID number was 7276.

Mooring

The DWR was used with a standard Datawell mooring as described in ref. 2.

Data Sources and Processing

For an overall view of the data sources and processing, see Figure 1. Although the buoy operated continuously between day 288/1993 and day 006/1994, periods of meaningful data (buoy deployed and on or near station) are as follows:

Day 293.6667	(1st deployed)	to Day 316.5000	(recovered adrift)
Day 326.6042	(re-deployed)	to Day 006.7917	(final recovery)

ARGOS

During the deployments, data were received from the DWR via the ARGOS system, which was regularly interrogated during the experiment to allow checks on both buoy position and data quality. The ARGOS messages, downloaded from the CLS ARGOS computer at Toulouse via the Public Switched System were decoded and sorted by the QuickBasic applications ARGDWRFILE and SORT SPECS (Appendices A.1, A.2). The former decoded all DWR ARGOS messages within an ARGOS dump into directional spectra; the latter sorted the spectra into chronological order, selecting the best choice from duplicated spectra, and produced chronologically ordered tabular files of the parameters Hs, Tz and Temperature in addition to the complete spectral files. File formats are given in Appendices B.1 - B.4

The ARGOS system could not acquire all of the half-hourly data messages produced by the buoy; there were also occasional transmission errors in the messages, which do not incorporate parity checking. The ARGOS data terminated with the recovery of the buoy on day 006/1994.

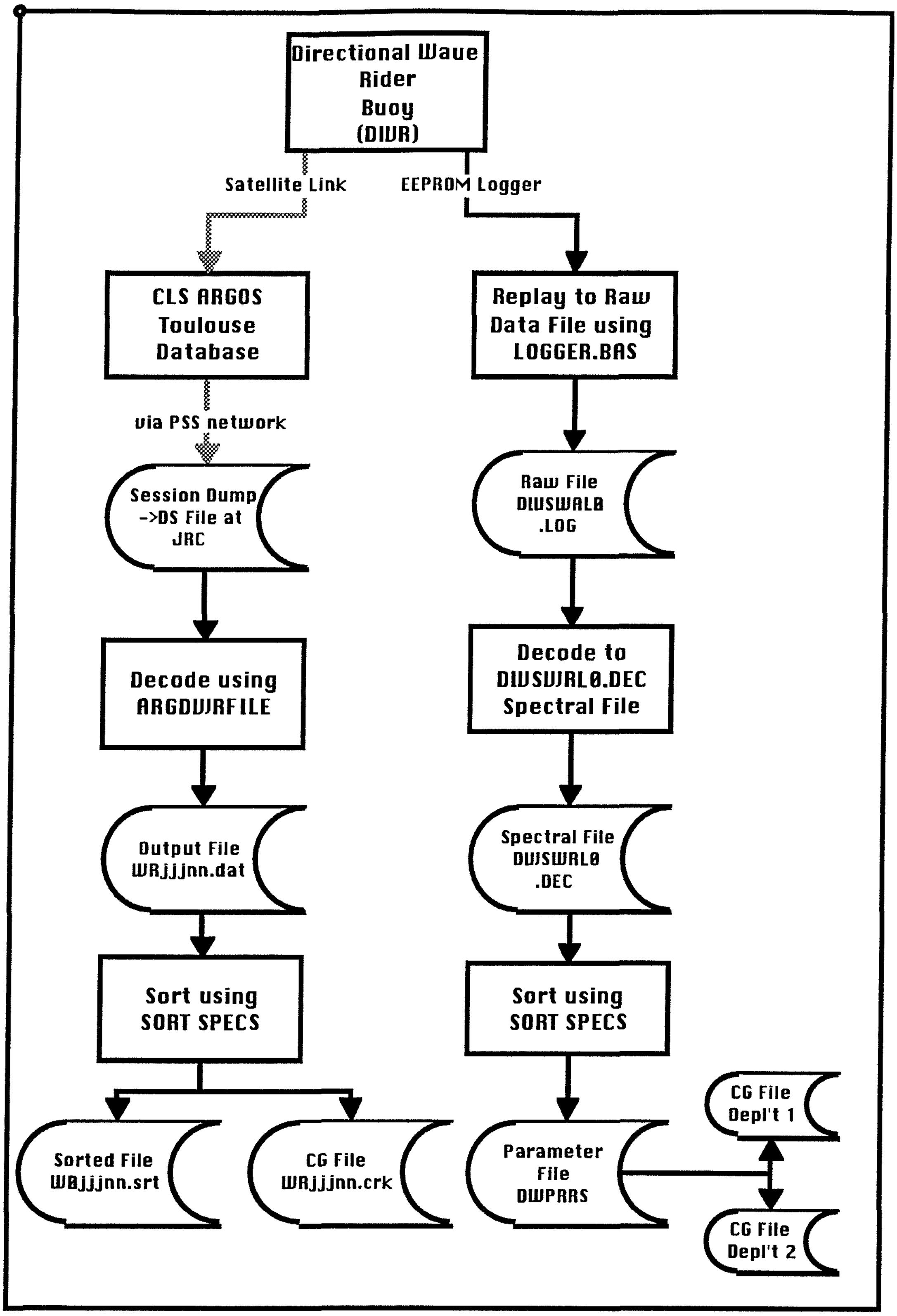


Figure 1 Data Acquisition/Processing

Internal EEPROM Logger

The complete error-free data set up to day 364/1993 was eventually recovered from the internal EEPROM logger. The logger contents were dumped to a PC disk file, using the Datawell application LOGGER.BAS. The resulting file was then decoded by the QuickBasic application DWDUMPDEC (Appendix A.3) to a file of the format produced by ARGDWRFILE; SORT SPECS was then run to produce a file of the concise parameters (the logger data were already in correct chronological order, so the sorting process was trivial).

The complete set of spectra and parameters for the period day 288.4583 to day 364.4167 are in files DWSWAL0.DEC and DWPARS. These have the formats given in Appendices B.2 - B.3

Data Quality Checking

The parameters for the two Sonic Buoy deployments were loaded into CricketGraph and plots made of Hs vs Day (figures 2 and 3), Temperature vs Day (figures 4 and 5) and Hs vs Tz (figures 6 and 7). Before quality control, the Hs plots showed, in addition to a period of high Hs (ca. 15m) during day 342, a few spikes during the following week. In order to investigate these, a simple QuickBasic plotting application PLOT DWR (Appendix A.4) was used to display plots of the energy spectrum, mean wave direction and directional spread against frequency; this program had been written to examine the spectra received via ARGOS for data quality. It was found that, during the suspect periods, there was a very large peak in the energy spectrum at a frequency of about 0.025 Hz (figure 9); this corresponds to the natural period of the stable platform used in the Datawell sensor. A normal spectrum is shown for comparison in figure 8. Unrealistic values of mean direction and directional spread were also present in these records. The very low frequency energy is symptomatic of a disturbance of the stable platform which normally results from an unusually large rotation or rotational rate of the buoy, e.g. if the buoy is spun when passing along the waterline of a ship, during a recovery operation or a collision. One might also expect the hull to be subjected to excess rotation in conditions of very high breaking waves, as were present during most of the suspect periods. During the first deployment, there were 2 occasions were bad Hs, Tz values occurred and these were followed in both cases by a record with a bad temperature value and a minor amount of very low frequency energy. The values of Hs, Tz or Temperature for the affected records were deleted from the CricketGraph data files and the plots included in this report are for the edited data.

The plots of Hs vs Tz showed that almost all Hs values lay below the $\lambda z/14$ limit calculated from the dispersion relationship without correction for current, i.e.

mean zero-crossing wavelength, $\lambda z = g.Tz^2/2\pi$ where $g = 9.81\ ms^{-2}$

Two other spectral plots are included to show well resolved combinations of swell from the West and a wind sea from the South-East (figure 10) and swell from the West and a wind sea from the North-West (figure 11). The spectral plots were prepared by running a simple QuickBasic application (SELECT DWR FOR PLOT, Appendix A.5) to convert the processed spectral file into tabular data suitable for entry into CricketGraph.

The records affected by spurious low frequency energy were as follows:

First Deployment - CG file SWALES DWR 1st Deploy't Params

308.6458	Hs and Tz	(? fishing Activities ?)
308.6667	Hs and Tz	(? fishing Activities ?)
308.6875	Temperature	(? fishing Activities ?)
310.6667	Hs and Tz	(re-deployment by Warden)
310.6875	Temperature	(re-deployment by Warden)

Second Deployment - CG file SWALES DWR 2nd Deploy't Params

342.6250	Hs and Tz
342.7708 - 342.8542 inclusive	Hs and Tz
344.9375	Hs and Tz
349.8750	Hs and Tz
349.9792	Hs and Tz
353.4167	Hs and Tz

Figure 2. Hs vs Day Number for 1st Deployment period

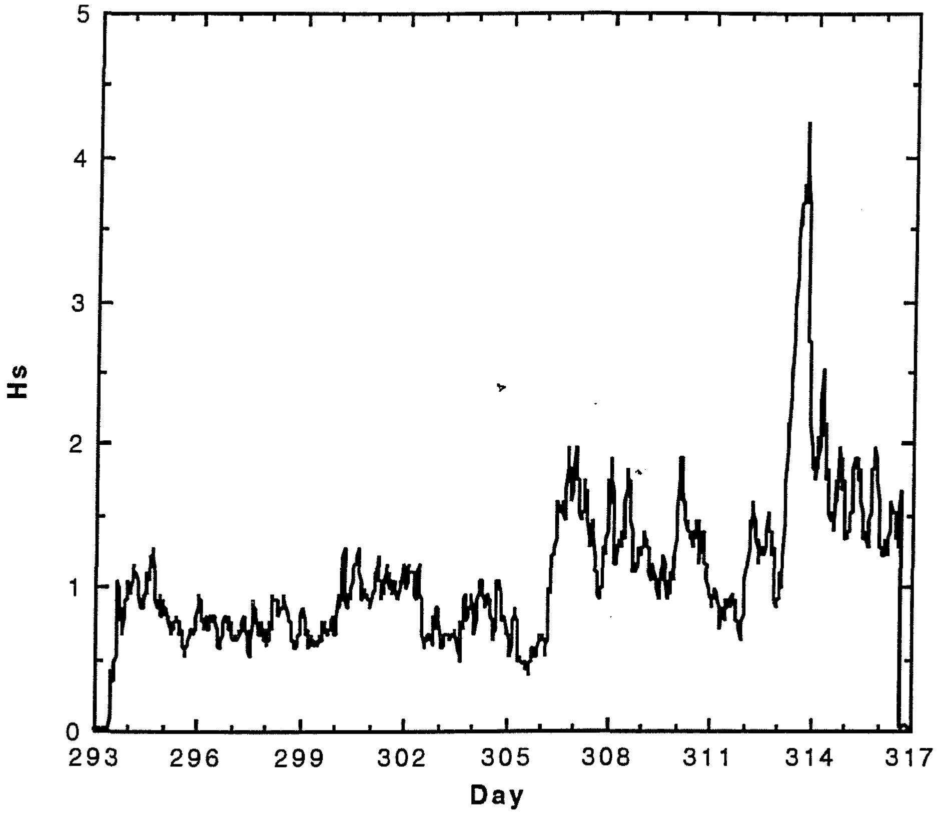


Figure 3. Hs vs Day Number for 2nd Deployment period

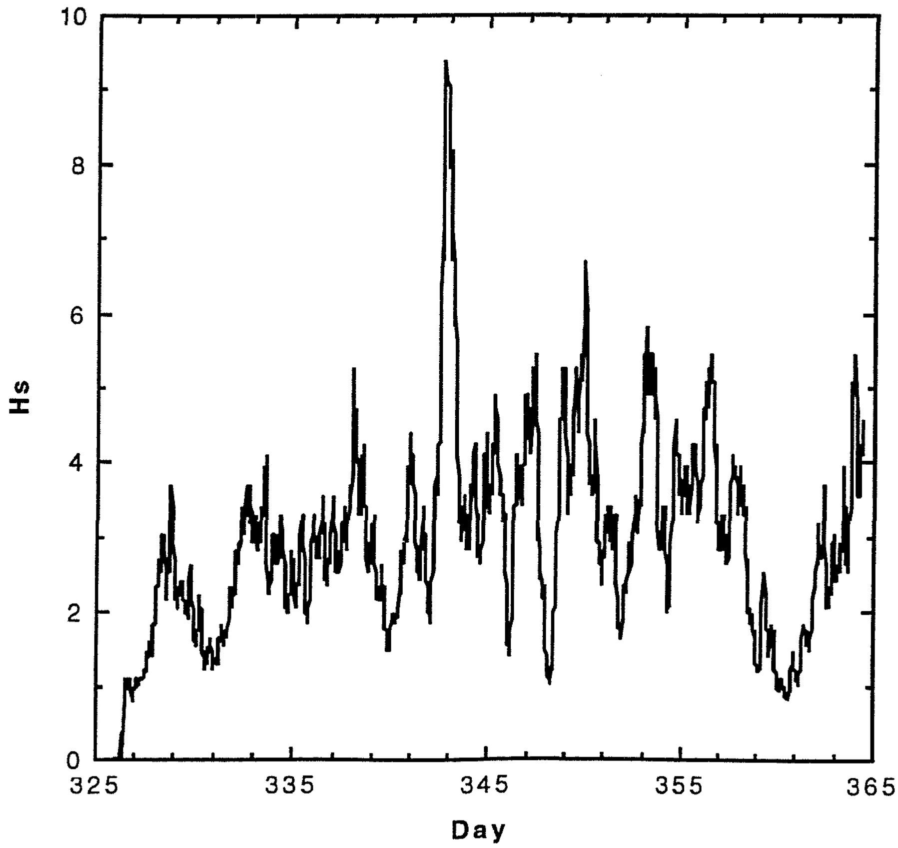


Figure 4. Temperature vs Day Number for 1st Deployment period

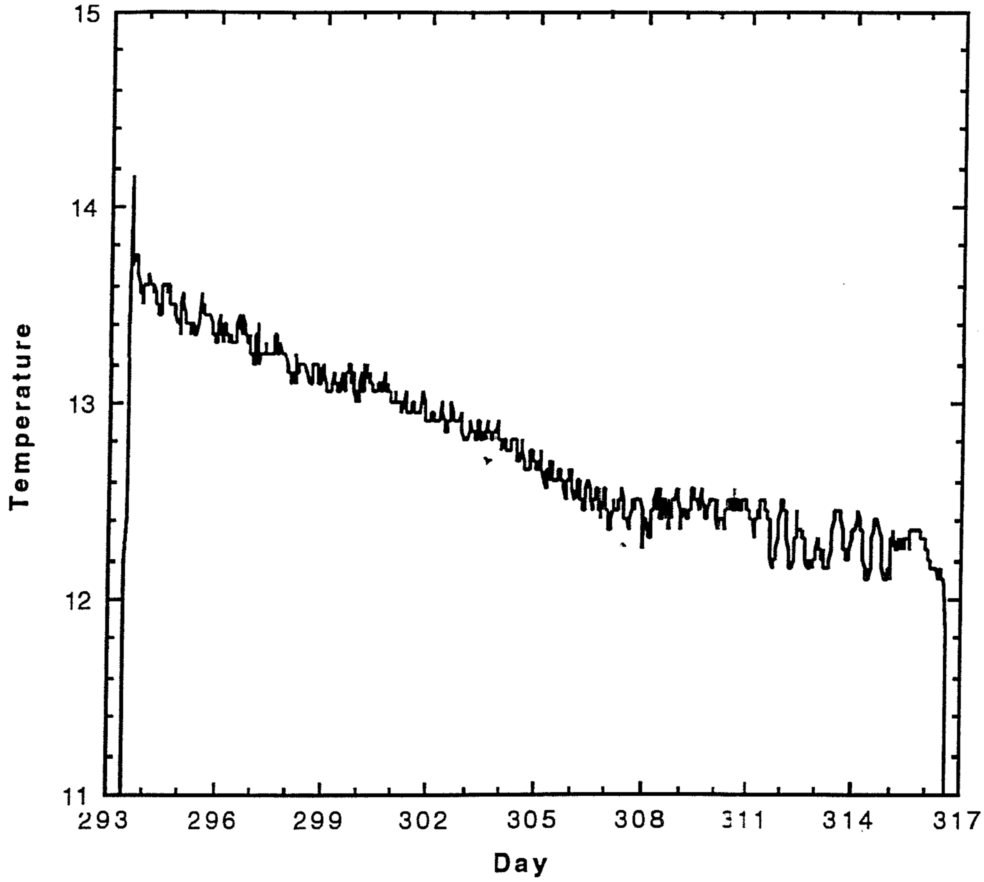


Figure 5. Temperature vs Day Number for 2nd Deployment period

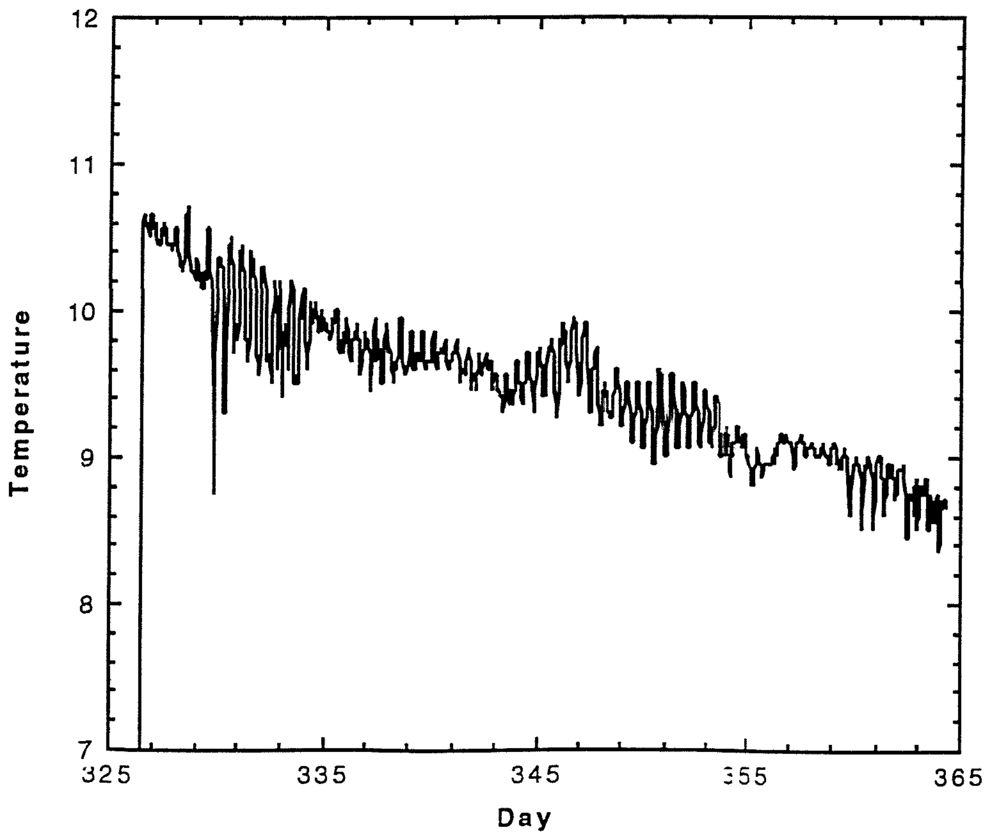


Figure 6. Hs vs Tz for 1st Deployment period

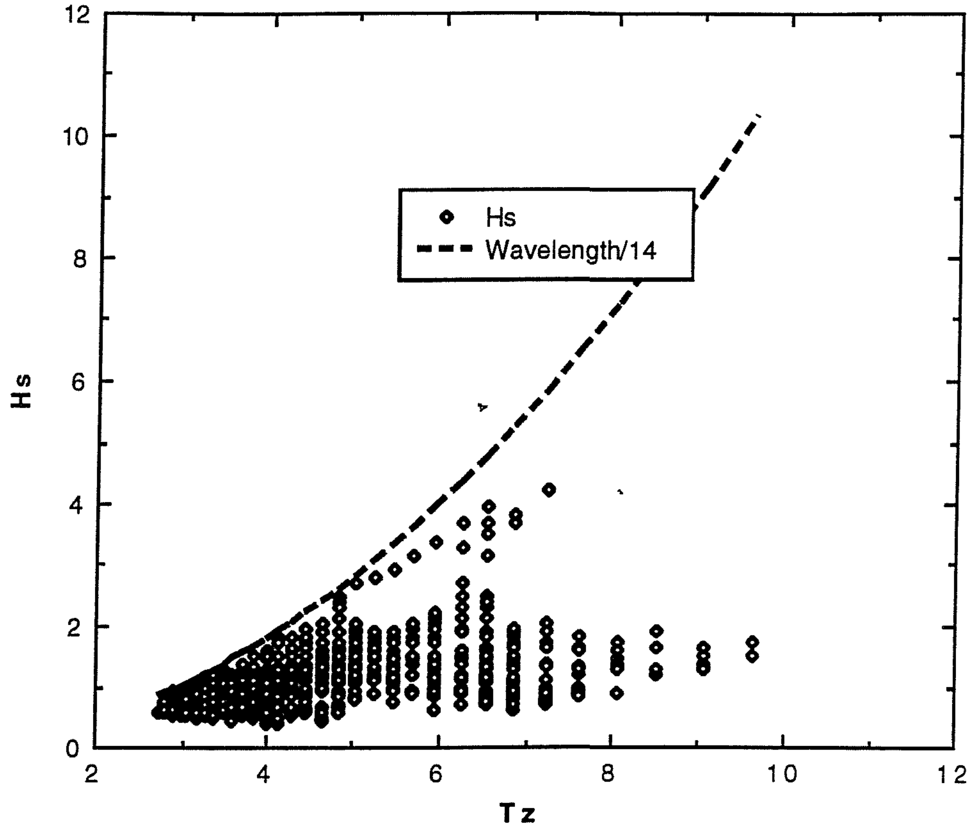


Figure 7. Hs vs Tz for 2nd Deployment period

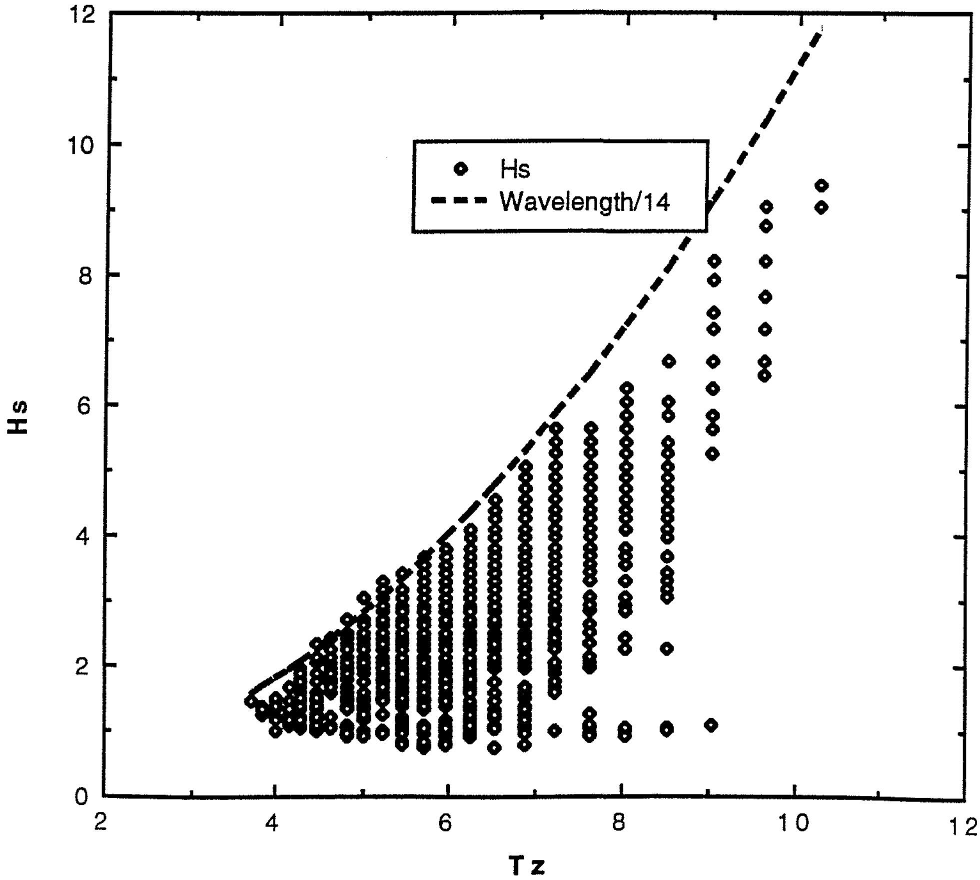


Figure 8. SWALES Directional Waverider Data Day 342.7500

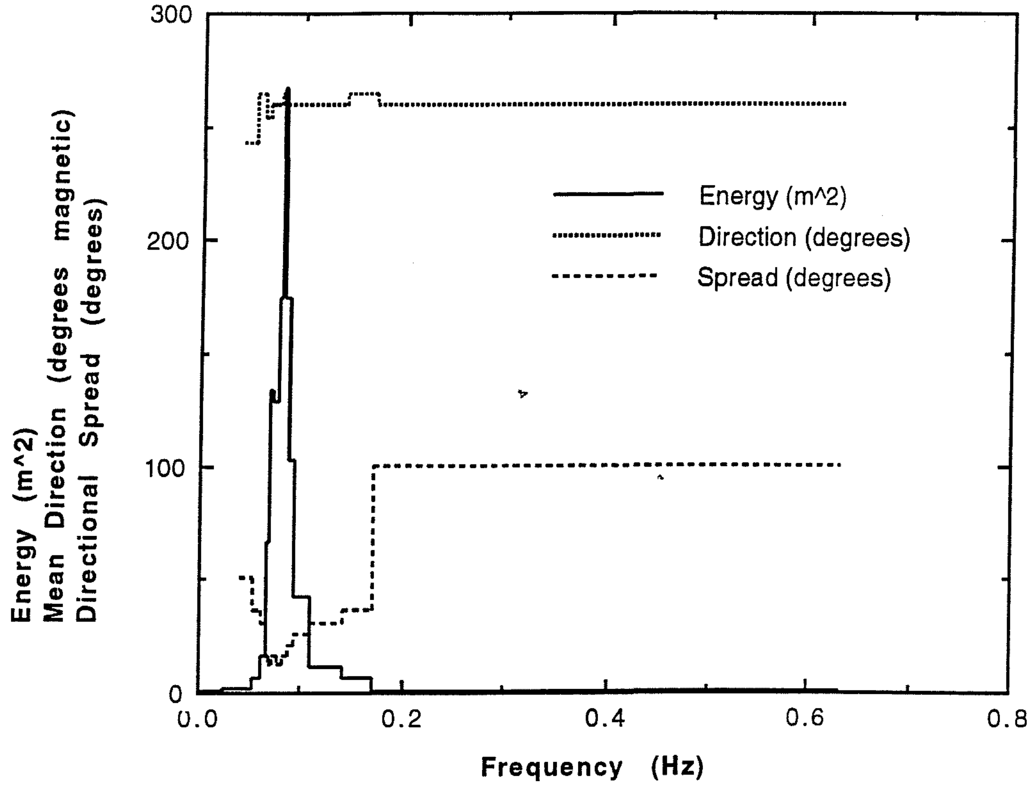


Figure 9. SWALES Directional Waverider Data Day 342.8333

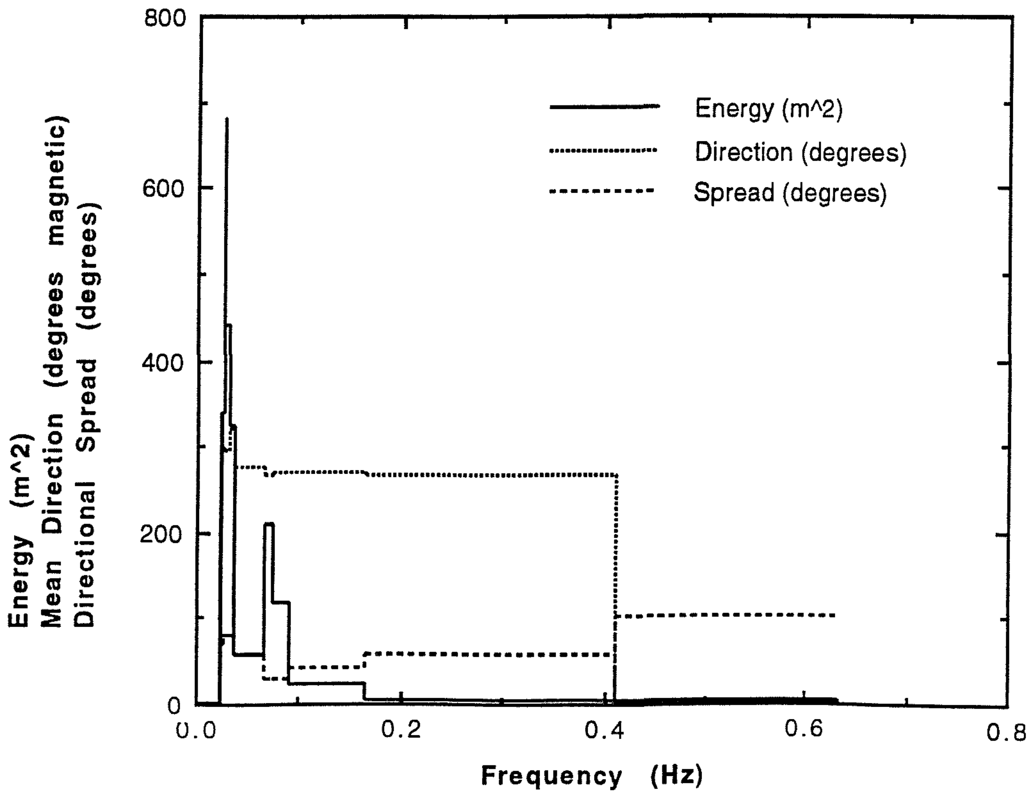


Figure 10. SWALES Directional Waverider Data Day 303.5208

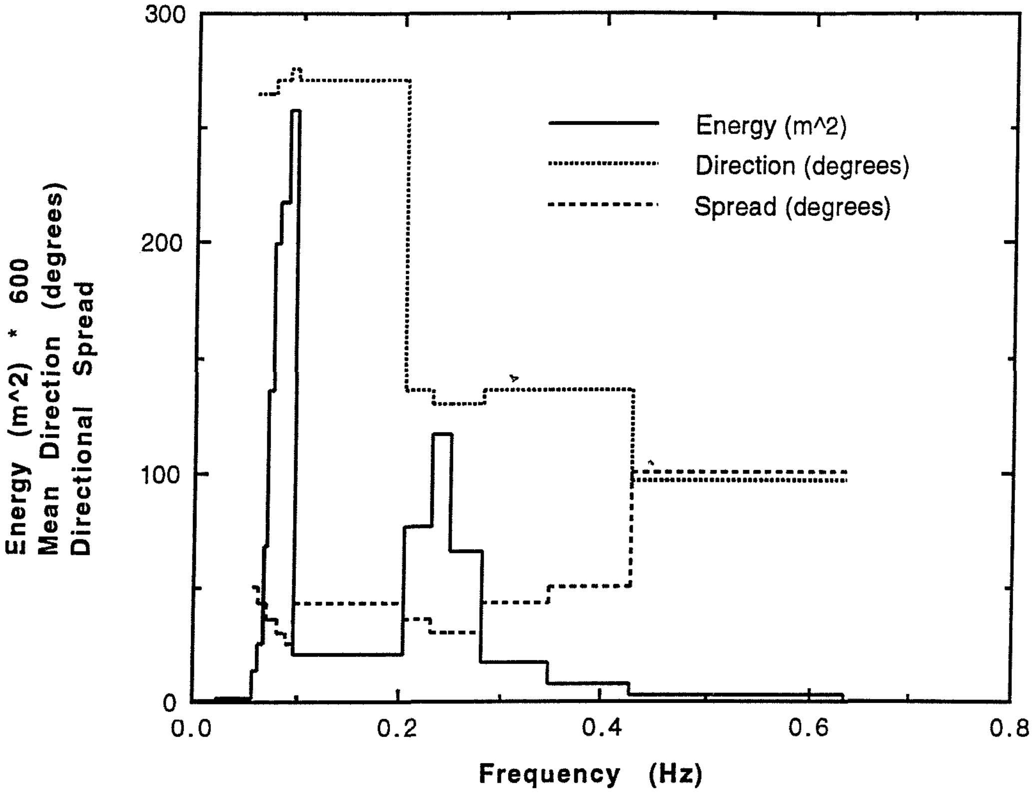
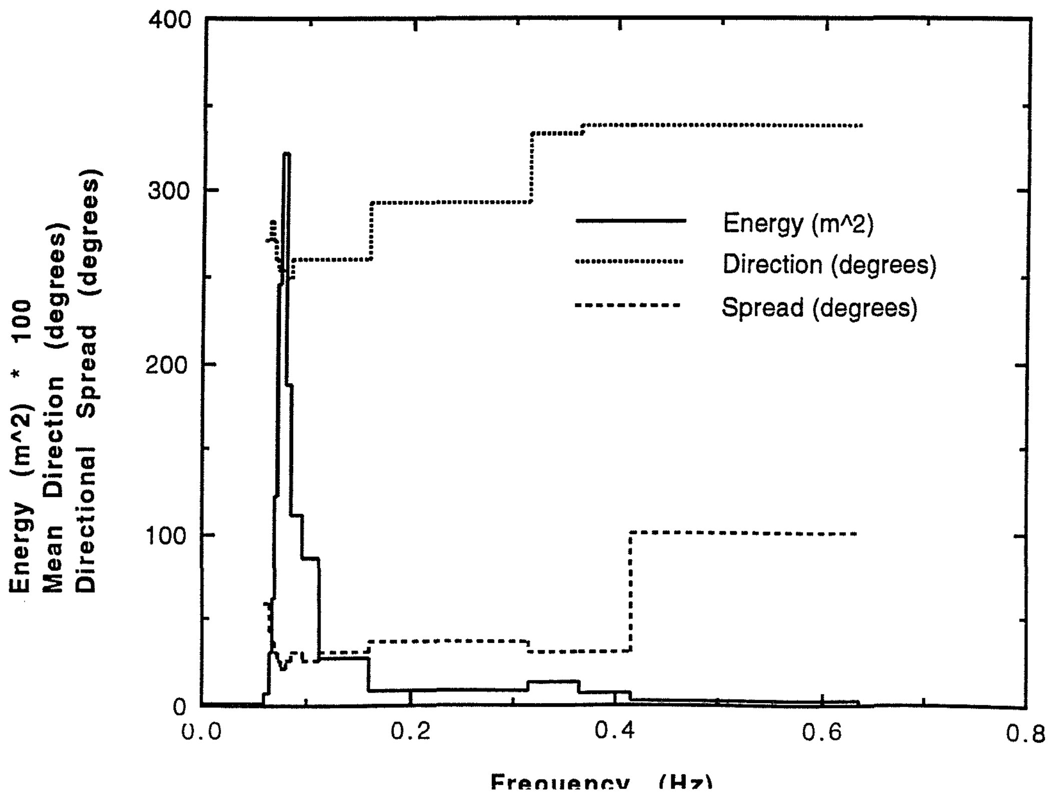


Figure 11. SWALES Directional Waverider Data Day 312.5000



Summary of Data Produced

Raw Data Files

EEPROM Logger File	DWSWAL0.LOG	Days 288.4583 - 364.4167

Processed Spectral Files

EEPROM Data	DWSWAL0.DEC	Days 288.4583 - 364.4167

Processed Parameter Files

EEPROM Data	DWPARS	Days 288.4583 - 364.4167

CricketGraph Data

ARGOS Data	(JRC) WRclean.crk.all.data.txt	Days 293.6812 - 1994/6.9037
EEPROM Data	SWALES DWR 1st Deploy't Params	Days 293.0000 - 316.9792
	SWALES DWR 2nd Deploy't Params	Days 326.0000 - 364.4167

Data Time Stamping

The raw EEPROM data file has no time information, but consists of sequential records starting at a known time which, according to the handbook, is 3 hours after power-up. The buoy was powered up on day 288 at 08:30 GMT, so that the first processing period should have started at 11:00 GMT (day 288.45833) and the first record written at 11:30 GMT. The EEPROM data files are based on this and are time stamped with the START of the sampling period; the sampling period is 1600 seconds, so that the middle of the sample is .01852 day after the time stamp.

As a check on buoy timing, the EEPROM logger data was aligned with the ARGOS data for two cases, a) near the start of the initial deployment and b) near the end of the fional deployment. From ARGOS satellite passes where the transmitted data was updated during the pass, a time check of a particular EEPROM logger record could be obtained from the satellite acquisition times with an uncertainty of a few minutes. The update time is nominally 30 minutes after the record start time. This gave the following results:

Case a)

EEPROM nominal record start time 296.75

DWR clock estimated to be 8.16 ± 3.89 minutes slow

Case b)

EEPROM nominal record start time 361.75

DWR clock estimated to be 6.08 ± 2.38 minutes slow

The mean correction is 7.12 minutes which, combined with the 800 seconds to the centre of the sampling period, means that 20.45 minutes (0.0142 day) must be added to the time stamps to obtain the middle-of-sample time,correct to about ±3 minutes.

This correction has not been made to the files listed above.

In the case of ARGOS data, the data are stamped with the time of acquisition by the satellite; normally this could be between 16.67 and 46.67 minutes after the middle-of-sample time, depending upon whether the data has just been updated or is just about to be updated. There will, of course, frequently be occasions where the data are updated within the satellite pass; however, the satellite may not acquire messages just before and just after the update, so some timing uncertainty will still result, as in cases a) and b) above.

Acknowledgements

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.

References

1. Taylor, P. K. (ed.) (1994) The SWALES experiment - Field phase summary,
JRC Internal Document (in preparation) .
2. Waddington, I. (1994) SWALES Moorings,
IOSDL Internal Document (in preparation)

APPENDIX A SOFTWARE LISTINGS

A.1 ARGDWRFILE

```
REM QuickBasic program ARGDWRFILE
REM  - decodeS ARGOS Dispose File DATA
REM (copied from Telnet) into engineering data
REM
REM ?Use program SORT RECS to further process into final data
REM
REM Author CHC Date 24-09-1993
REM modified by PKT 13/10/93

DIM b$(8)
DIM W(32)
DIM day%(5), hrs%(5), mins%(5), mdays%(12)

ON ERROR GOTO Handler    'for opening new output file

REM load days of month array (used to find Julian day)
FOR N% = 1 TO 12:READ mdays%(N%):NEXT
DATA 0, 31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30

INPUT"Filename containing ARGOS session",F$

F$=":.SWALES_data:..Telnet_files:"+F$
PRINT "Using input file...  ";F$
OPEN F$ FOR INPUT AS #1

Outfile:
cflag% = 0
INPUT"Enter filename for output data (WRjjjnn) :";g$
IF (MID$(g$,1,2)<>"WR")THEN
 PRINT "Filename must start with WR"
 GOTO Outfile
 END IF
g$=":.SWALES_data:.dat_files:"+g$+".dat"

PRINT "Using output file...  ";g$
10 FILES g$   ' results in error if not existing already, handled by Handler
IF cflag% = 0 THEN
BEEP
INPUT "This file exists.  Overwrite data";r$
IF (r$ <> "y") AND (r$ <> "Y") THEN GOTO Outfile
END IF
OPEN g$ FOR OUTPUT AS #2

WHILE NOT EOF(1)
Readheader:
IF EOF(1) THEN END
LINE INPUT#1, h$
REM Check for start line and correct PTT
IF (LEFT$(h$,11) <> "00296 07276") THEN GOTO Readheader

CLS
PRINT "PTT: ";MID$(h$, 7, 5);
```

```
REM line will be over 30 chars if it contains a fix
IF LEN(h$) < 30 THEN fixflag% = 0 ELSE fixflag% = 1

REM nlines%-1 of data, 8 lines per frame
nlines% = VAL(MID$(h$, 14, 2)):nframes% = (nlines% - 1)/8
PRINT " Lines:";nlines%;" Frames:";nframes%

IF fixflag% = 1 THEN
  fixtime$= MID$(h$, 24, 19)
  lat$=MID$(h$, 45, 6):long$=MID$(h$, 53, 7)
  PRINT fixtime$
  PRINT "Latitude:";lat$;" Longditude:";long$
END IF

FOR frame% = 1 TO nframes%
  FOR m% = 1 TO 8
    LINE INPUT#1, b$(m%)
    IF m% = 1 THEN acqtime$= MID$(b$(m%), 7, 19):PRINT acqtime$
      FOR N% = 1 TO 4
        REM 1st line is decimal, others are hex
        IF (m% > 9) THEN   'temporary fix for decimal data
          b$(m%) = RIGHT$(b$(m%), 41)
          p% = 1+13*(N%-1):N$=MID$(b$(m%), p%, 2)
          n1% = ASC(LEFT$(N$,1))
          IF (n1% < 58) THEN n1% = n1% - 48 ELSE n1% = n1% - 55
          n2% = ASC(RIGHT$(N$, 1))
          IF (n2% < 58) THEN n2% = n2% - 48 ELSE n2% = n2% - 55
          W(4 * (m% - 1) + N%) = 16 * n1% + n2%
        ELSE
          b$(m%) = RIGHT$(b$(m%),42)
          p% = 1+13*(N% - 1):N$=MID$(b$(m%), p%, 3)
          W(4 * (m% - 1) + N%) = VAL(N$)
        END IF
      NEXT N%
  NEXT m%

  RESTORE 100
REM FOR Z%=1 TO 8:FOR Y%=1 TO 4:PRINT W(Y%+4*(Z%-1));"
";:NEXT:PRINT:NEXT:INPUT R$
  ayear% = VAL(LEFT$(acqtime$, 4))
  aday% = VAL(MID$(acqtime$, 9, 2))
  amonth% = VAL(MID$(acqtime$, 6, 2))
  ahr% = VAL(MID$(acqtime$, 12, 2))
  amin% = VAL(MID$(acqtime$, 15, 2))
  asec% = VAL(MID$(acqtime$, 18, 2))
  ahr = ahr% + amin%/60 + asec%/3600

  jday% = 0
  FOR N% = 1 TO amonth%
    jday% = jday% + mdays%(N%)
    IF (N% = 3) AND (INT(ayear%/4) = 0) THEN jday% = jday% + 1
  NEXT
  jday% = jday% + aday%

  PRINT "Day:"; jday%;" Hr:";
  PRINT USING "##.###"; ahr

  PRINT #2, USING "###.#####,";jday% + ahr%/24 +amin%/1440 + asec%/86400&;
  IF (fixflag% = 1) THEN
    PRINT #2, USING "\   \,"; lat$;
    PRINT #2, USING "\   \";long$
  ELSE
    PRINT #2,"99.999,999.999"
```

```
END IF

V=8.5+2*INT(W(1)/32)
IF V>22 THEN
   PRINT " Batt. Voltage > 22.5 V, ";
ELSE
   PRINT " Batt. Voltage = ";V;" V, ";
END IF
g=W(2) MOD 128
RMS=16*(EXP(g/32)-1)
PRINT "    R.M.S. Ht = ";RMS;" cm"
M0=RMS*RMS
PRINT#2,USING "######.##";M0
FU=.025
g=2*(W(1) MOD 32)+INT(W(2)/128)
PRINT"  Band (Hz)       Energy  Direction (deg)   Spreading (deg)"
PRINT USING "#.######_-";FU;
PRINT#2,USING "#.######_,";FU;
W=(EXP(g/32)-1)/25
FU=FU+W
PRINT USING "#.######       ";FU;
PRINT#2,USING "#.######_,";FU;
READ E$:PRINT USING "\        \_---.---          --.---";E$
PRINT#2,"999.999,99.999"
FOR N=1 TO 13
   XA=W(2*N + 1):XB=W(2*N+2):g=INT(XA/4)
   PRINT USING"#.######_-";FU;
   PRINT#2,USING"#.######_,";FU;
   W=8*(EXP(g/32)-1)/200
   FU=FU+W
   PRINT USING "#.######      ";FU;
   PRINT#2,USING "#.######_,";FU;
   READ E$:PRINT USING "\        \";E$;
   dirn = 5.625*(16*(XA MOD 4)+INT(XB/16))
   PRINT USING "###.###          ";dirn;
   PRINT#2,USING "###.###_,";dirn;
   spread = 14.32*(EXP((XB MOD 16)/8)-1)
   PRINT USING "##.###";spread
   PRINT#2,USING "##.###";spread
NEXT
XA=W(29):XB=W(30)
PRINT USING "#.######_-0.63500      ";FU;
PRINT#2,USING "#.######_,0.63500,";FU;
READ E$:PRINT USING "\        \";E$;
PRINT USING "###.###          _--.---";5.625*INT(XA/4)
PRINT#2,USING "###.###_,99.999";5.625*INT(XA/4)

g=16*(XA MOD 4)+INT(XB/16)
HS = .04*RMS:TZ=1/(.32*(EXP(g/64)-1))
PRINT USING "_Hs = ##.##"; HS;
PRINT USING "_ metres: Tz = ##.##_ seconds";TZ
PRINT #2, USING "##.##_,"; HS;
PRINT#2, USING "##.##_,";TZ;
SB=INT((XB MOD 16)/8)
IF SB=1 THEN
  PRINT "Memory Error":PRINT#2,"0,";
ELSE
   PRINT "Memory Check OK":PRINT#2,"1,";
END IF
PRINT "Acceleration offset compensation (cm/sec/sec)"
AV = 32*(XB MOD 8)
PRINT USING "_Vertical = ###"; AV;
PRINT#2, USING "###_,"; AV;
```

```
    XA=W(31):XB=W(32):AN = 32*INT(XA/32):AW = 32*(INT(XA/4) MOD 8)
    PRINT USING "_ North = ###,"; AN;
    PRINT#2, USING "###_,"; AN;
    PRINT USING "_ West = ###,"; AW
    PRINT #2, USING "###_,"; AW;
    TEMP = (256*(XA MOD 4)+XB)/20-5
    PRINT USING "_Temperature = ##.##_ deg C"; TEMP
    PRINT#2, USING "+##.##"; TEMP

  NEXT
  WEND

  100 DATA
  "1/512","1/256","1/128","1/64","1/32","1/16","1/8","1/8","1/8","1/8","1/8","
  1/8","1/16","1/32","1/30.12"


  Endprog:
  CLOSE#1
  CLOSE#2
  END

  REM Subroutines

  Handler:
    IF (ERL = 10) AND (ERR = 53) THEN
      OPEN g$ FOR OUTPUT AS #2
      cflag% = 1
      CLOSE#2
    END IF
    RESUME NEXT
```

A.2 SORT SPECS

```
REM QuickBasic Program SORT SPECS
REM - this sorts DWR ARGOS data (which has already been
REM decoded from DS format by the program ARGDWRFILE)
REM  into chronological order.
REM
REM Author CHC Date 24-09-1993
REM Modified by PKT 18/10/93

REM Can process a file containing up to 1000 messages
DIM day(1000), flag%(1000),indx%(1000)

ON ERROR GOTO Handler    'for opening new output file

REM home lat and lon positions and distance km that buoy can move from that
position
REM before a warning is given
REM hlat=51.695:hlon=355 'positions for dock
hlat=51.494
hlon=355.246
hdist=5

Infile:
INPUT "Enter name of file to be sorted (WRjjjnn) :";f$
IF MID$(f$,1,2)<>"WR" THEN
BEEP
PRINT "Must start with WR"
GOTO Infile
```

```
END IF

ff$=":.SWALES_data:.dat_files:"+f$+".dat"
PRINT "Using input file...  ";ff$
OPEN ff$ FOR INPUT AS #1

Outfile:
cflag% = 0
REM INPUT "Enter filename for output data:";g$
g$=":.SWALES_data:.crk_files:"+f$+".crk"
PRINT "Using output file...  ";g$
10 FILES g$   ' results in error if not existing already, handled by Handler
IF cflag% = 0 THEN
BEEP
INPUT "This file exists. Overwrite data";r$
IF (r$<> "y") AND (r$ <> "Y") THEN GOTO Endprog
END IF
OPEN g$ FOR OUTPUT AS #3

REM write cricket graph header
PRINT#3, "*"
PRINT#3,
"day";CHR$(9);"hhmm$";CHR$(9);"lat";CHR$(9);"dlat";CHR$(9);"long";CHR$(9);"d
lon";
PRINT#3, CHR$(9);"Hs";CHR$(9);"Tz";CHR$(9);"Temp"

REM file for sorted data
Outfile1:
cflag% = 0
REM INPUT "Enter filename for output data:";hh$
hh$=":.SWALES_data:.srt_files:"+f$+".srt"
PRINT "Using output file...  ";hh$
20 FILES hh$   ' results in error if not existing already, handled by Handler
IF cflag% = 0 THEN
BEEP
INPUT "This file exists. Overwrite data";r$
IF (r$<> "y") AND (r$ <> "Y") THEN GOTO Endprog
END IF
OPEN hh$ FOR OUTPUT AS #2


PRINT "Reading File..."
REM First find the number of messages, l%,
l% = 1:lin% = 1
WHILE NOT EOF(1)
  LINE INPUT#1, h$
  IF (MID$(h$,4,1) = ".") THEN
    day(l%) = VAL(LEFT$(h$,8))
    flag%(l%) = lin%
    l% = l% + 1
  END IF
  lin%=lin%+1
WEND
CLOSE#1

n% = l% - 1
PRINT "File contains ";n%;" spectra"
PRINT "Sorting file..."

REM Now sort into chronological order by producing an index table
REM Method from Press, Flannery et al. "The Art of Scientific Computing"
FOR j% = 1 TO n%:indx%(j%) = j%:NEXT
IF (n% = 1) THEN GOTO Datasave
```

```
l% = n%/2 + 1
ir% = n%

WHILE (ir% > 1)
  IF (l% > 1) THEN
    l% = l% - 1
    indxt% = indx%(l%)
    Q = day(indxt%)
  ELSE
    indxt% = indx%(ir%)
    Q = day(indxt%)
    indx%(ir%) = indx%(1)
    ir% = ir%-1
    IF (ir% = 1) THEN indx%(1) = indxt%
  END IF
  i% = l%:j% = 2*l%
  WHILE (j% <= ir%)
    IF (j% < ir%) AND (day(indx%(j%)) < day(indx%(j%+1))) THEN j% = j%+1
    IF (Q < day(indx%(j%))) THEN
      indx%(i%) = indx%(j%)
      i% = j%
      j% = j% + i%
    ELSE
      j% = ir% + 1
    END IF
  WEND
indx%(i%) = indxt%
WEND

Datasave:
REM Now use the index table to identify groups of messages having
REM the same day/time. The first one is selected and
REM this message  is then written to the output file

OPEN ff$ AS #1 LEN = 532
FIELD#1, 8 AS day$, 524 AS rest$

lastday$ = "": stflag% = 0
FOR n% = 1 TO lin%
  dayn = day(indx%(n%))
  IF (n% = 1) THEN
    lastday = dayn
    n1% = n%
  ELSE
    IF (dayn <> lastday) THEN
      n2% = n% - 1
      PRINT n% - n1%;" duplicate(s)  ";
      GET#1, indx%(n1%)
      PRINT day$;
      PRINT#2, day$;rest$;

      IF VAL(day$)<999 THEN
       PRINT#3, day$;CHR$(9);
       time=VAL(day$)
       day=INT(time)
       amin=((time-day)*24)
       hr=INT(amin)
       amin=(amin-hr)*60
       time=INT(hr*100+amin)
       PRINT#3, time;CHR$(9);
       ELSE
       PRINT#3,CHR$(9);CHR$(9);
       END IF
```

```
          lat$=MID$(rest$,2,6)
          lon$=MID$(rest$,9,7)
          PRINT " ";lat$;" ";lon$
          IF VAL(lat$)<90 THEN
          dlat=(VAL(lat$)-hlat)*111.262
          PRINT#3, lat$;CHR$(9);dlat;CHR$(9);
          ELSE
          PRINT#3,CHR$(9);CHR$(9);
          dlat=0
          END IF
            IF VAL(lon$)<900 THEN
            dlon=(VAL(lon$)-hlon)*70.201
            PRINT#3, lon$;CHR$(9);dlon;CHR$(9);
            ELSE
            PRINT#3,CHR$(9);CHR$(9);
            dlon=0
            END IF
          dist=SQR(dlat*dlat+dlon*dlon)
          IF dist>hdist THEN
          BEEP
          PRINT "Wave Rider OFF POSITION"
          END IF

         PRINT #3,
 MID$(rest$,492,5);CHR$(9);MID$(rest$,498,5);CHR$(9);MID$(rest$,518,6)
         lastday = dayn
         n1% = n%
       END IF
    END IF
    lastday$ = day$
Endprog:
NEXT n%
CLOSE#1
CLOSE#2
CLOSE#3

END

REM Subroutines

Handler:
  IF (ERL = 10) AND (ERR = 53) THEN
     OPEN g$ FOR OUTPUT AS #3
     cflag% = 1
     CLOSE#3
  END IF
  IF (ERL = 20) AND (ERR = 53) THEN
     OPEN hh$ FOR OUTPUT AS #2
     cflag% = 1
     CLOSE#2
  END IF

  RESUME NEXT
```

A.3 DWDUMPDEC

```
REM QuickBasic Program DWDUMPDEC
REM
REM Decodes file produced by memory dump
REM of Directional Waverider wave data
```

```
REM using Datawell LOGGER.BAS program

REM CHC 17-1-94

DIM W(32)

REM set disp% to 1 for screen output of spectra
disp%=0

OPEN "Wooig8-CHC:CHC-mac:Swales:DWSWAL0.LOG" AS #1 LEN=40
OPEN "Wooig8-CHC:CHC-mac:Swales:DWSWAL0.DEC" FOR OUTPUT AS #2
FIELD #1, 32 AS REC$, 8 AS SP$

PRINT "Starting Decoding"
REM first 3 fields are header data
FOR I%=4 TO 3650
  PRINT I%;" ";
  RESTORE 100
  GET #1, I%
  FOR B%=1 TO 32
    W(B%)=ASC(MID$(REC$,B%,1))
  NEXT

  REM I%=4 is 1st record which has start time
  REM of 1100 hrs on day 288
  jday=288+(I%+18)/48
  PRINT USING "###.####";jday;
  PRINT #2, USING "###.####,";jday;
  REM lat and long
  PRINT #2,"99.999,999.999"

  V=8.5+2*INT(W(1)/32)
  IF (V>22) AND (disp%>0) THEN
    PRINT " Batt. Voltage > 22.5 V, ";
  ELSE
    PRINT " Batt. Voltage = ";V;" V, ";
  END IF
  G=W(2) MOD 128
  RMS=16*(EXP(G/32)-1)
  PRINT "    R.M.S. Ht = ";RMS;" cm"
  M0=RMS*RMS
  PRINT#2,USING "######.##";M0
  FU=.025
  G=2*(W(1) MOD 32)+INT(W(2)/128)
  IF disp%>0 THEN PRINT"   Band (Hz)       Energy   Direction (deg)   Spreading (deg)"
  IF disp%>0 THEN PRINT USING "#.#####_-";FU;
  PRINT#2,USING "#.#####_,";FU;
  W=(EXP(G/32)-1)/25
  FU=FU+W
  IF disp%>0 THEN PRINT USING "#.#####      ";FU;
  PRINT#2,USING "#.#####_,";FU;
  READ E$:IF disp%>0 THEN PRINT USING "\        \_---.---          --.---";E$
  PRINT#2,"999.999,99.999"
  FOR N=1 TO 13
    XA=W(2*N + 1):XB=W(2*N+2):G=INT(XA/4)
    IF disp%>0 THEN PRINT USING"#.#####_-";FU;
    PRINT#2,USING"#.#####_,";FU;
    W=8*(EXP(G/32)-1)/200
    FU=FU+W
    IF disp%>0 THEN PRINT USING "#.#####      ";FU;
    PRINT#2,USING "#.#####_,";FU;
    READ E$:IF disp%>0 THEN PRINT USING "\        \";E$;
    dirn = 5.625*(16*(XA MOD 4)+INT(XB/16))
```

```
      IF disp%>0 THEN PRINT USING "###.###          ";dim;
      PRINT#2,USING "###.###_,";dim;
      spread = 14.32*(EXP((XB MOD 16)/8)-1)
      IF disp%>0 THEN PRINT USING "##.###";spread
      PRINT#2,USING "##.###";spread
    NEXT
    XA=W(29):XB=W(30)
    IF disp%>0 THEN PRINT USING "#.#####_-0.63500       ";FU;
    PRINT#2,USING "#.#####_,0.63500,";FU;
    READ E$:IF disp%>0 THEN PRINT USING "\        \";E$;
    IF disp%>0 THEN PRINT USING "###.###          _--.---";5.625*INT(XA/4)
    PRINT#2,USING "###.###_,99.999";5.625*INT(XA/4)

    G=16*(XA MOD 4)+INT(XB/16)
    HS = .04*RMS:TZ=1/(.32*(EXP(G/64)-1))
    IF disp%>0 THEN PRINT USING "_Hs = ##.##"; HS;
    IF disp%>0 THEN PRINT USING "_ metres: Tz = ##.##_ seconds";TZ
    PRINT #2, USING "##.##_,"; HS;
    PRINT#2, USING "##.##_,";TZ;
    SB=INT((XB MOD 16)/8)
    IF SB=1 THEN
      IF disp%>0 THEN PRINT "Memory Error"
      PRINT#2,"0,";
    ELSE
      IF disp%>0 THEN PRINT "Memory Check OK"
      PRINT#2,"1,";
    END IF
    IF disp%>0 THEN PRINT "Acceleration offset compensation (cm/sec/sec)"
    AV = 32*(XB MOD 8)
    IF disp%>0 THEN PRINT USING "_Vertical = ###"; AV;
    PRINT#2, USING "###_,"; AV;
    XA=W(31):XB=W(32):AN = 32*INT(XA/32):AW = 32*(INT(XA/4) MOD 8)
    IF disp%>0 THEN PRINT USING "_ North = ###,"; AN;
    PRINT#2, USING "###_,"; AN;
    IF disp%>0 THEN PRINT USING "_ West = ###,"; AW
    PRINT #2, USING "###_,"; AW;
    TEMP = (256*(XA MOD 4)+XB)/20-5
    IF disp%>0 THEN PRINT USING "_Temperature = ##.##_ deg C"; TEMP
    PRINT#2, USING "+##.##"; TEMP

NEXT


100 DATA "1/512  ","1/256  ","1/128  ","1/64   ","1/32   ","1/16  ","1/8   "
DATA "1/8   ","1/8   ","1/8   ","1/8   ","1/8   ","1/16   ","1/32   ","1/30.12"



CLOSE#1
CLOSE#2

END
```

A.4 PLOT DWR

```
REM QuickBasic Program PLOT DWR
REM
REM Plots spectra of Directional Waverider
REM Energy, Mean Direction and Directional Spread
REM from a spectral file produced by SORT SPECS
REM
```

```
REM CHC    10th Nov 1993

DIM fl(15),fu(15),dir(15),spr(15),wt(15),en(15)

INPUT "Enter filename of sorted DWR spectra:",f$
f$="Wooig8-CHC:Applications:QuickBASIC:DWR SWALES data:" + f$
OPEN f$ FOR INPUT AS #1
FOR n% = 1 TO 15
  READ wt(n%)
NEXT
WHILE NOT EOF(1)
INPUT#1,d$,lat$,long$
INPUT#1,var
maxen = 0
FOR n% = 1 TO 15
  INPUT#1,fl(n%),fu(n%),dir(n%),spr(n%)
  IF (fu(n%)=fl(n%)) THEN
    en(n%)=35
  ELSE
    en(n%) = .0001*var/(wt(n%)*(fu(n%)-fl(n%)))
    IF en(n%) > maxen THEN maxen = en(n%)
  END IF
  REM PRINT en(n%)
NEXT
INPUT#1,hs,tz,memcheck,accx,accy,accz,temperature

LINE (50,350) - (200,50),,B

FOR n% = 1 TO 15
  IF n% = 1 THEN PSET (200*fl(n%)+50,350)
  LINE - (200*fl(n%)+50,350 -300*en(n%)/maxen )
  LINE - (200*fu(n%)+50,350- 300*en(n%)/maxen )
NEXT
REM INPUT r$
LOCATE 24,5: PRINT d$;" Energy Density"
FOR n%=1 TO 1000:NEXT
REM INPUT r$

LINE (250,350) - (400,50),,B

k=300/360
FOR n% = 1 TO 15
  IF n% = 2 THEN PSET (200*fl(n%)+250,350- k*dir(n%))
  IF dir(n%) < 999 THEN
    LINE - (200*fl(n%)+250,350-k*dir(n%) )
    LINE - (200*fu(n%)+250,350-k*dir(n%) )
  END IF
NEXT
REM INPUT r$
LOCATE 24,35: PRINT d$;" Direction"
FOR n%=1 TO 1000:NEXT
REM INPUT r$

LINE (450,350) - (600,50),,B

k = 300/90
FOR n% = 1 TO 15
  IF n% = 2 THEN PSET (200*fl(n%)+450,350-k*spr(n%))
  IF spr(n%) < 99 THEN
    LINE - (200*fl(n%)+450,350-k*spr(n%) )
    LINE - (200*fu(n%)+450,350-k*spr(n%) )
  END IF
NEXT
```

```
REM INPUT r$
LOCATE 24,60: PRINT d$;" Spread"
REM FOR n&=1 TO 3000&:NEXT

INPUT r$
CLS

WEND
CLOSE#1
END
DATA 512,256,128,64,32,16,8,8,8,8,8,8,16,32,30.12
```

A.5 SELECT DWR FOR PLOT

```
REM QuickBasic Program SELECT DWR FOR PLOT
REM
REM Plots particular spectrum of Directional Waverider
REM Energy, Mean Direction and Directional Spread
REM from a spectral file produced by SORT SPECS
REM
REM then produces a tabular file suitable for CG
REM
REM CHC   3rd March 1994

DIM fl(15),fu(15),dir(15),spr(15),wt(15),en(15)

INPUT "Enter filename of sorted DWR spectra:",f$
f$="Wooig8-CHC:CHC-mac:Swales:DWR:" + f$
OPEN f$ FOR INPUT AS #1
INPUT "Enter time of record for selection (jjj.jjjj)", DS$
FOR n% = 1 TO 15
  READ wt(n%)
NEXT
WHILE NOT EOF(1)
INPUT#1,d$,lat$,long$
INPUT#1,var
maxen = 0
FOR n% = 1 TO 15
  INPUT#1,fl(n%),fu(n%),dir(n%),spr(n%)
NEXT
INPUT#1,hs,tz,memcheck,accx,accy,accz,temperature

IF ABS(VAL(d$) - VAL(DS$)) > .001 THEN GOTO nextrec

FOR n% = 1 TO 15
  IF (fu(n%)=fl(n%)) THEN
    en(n%)=35
  ELSE
    en(n%) = .0001*var/(wt(n%)*(fu(n%)-fl(n%)))
    IF en(n%) > maxen THEN maxen = en(n%)
  END IF
NEXT


LINE (50,350) - (200,50),,B

FOR n% = 1 TO 15
  IF n% = 1 THEN PSET (200*fl(n%)+50,350)
  LINE - (200*fl(n%)+50,350 -300*en(n%)/maxen )
  LINE - (200*fu(n%)+50,350- 300*en(n%)/maxen )
NEXT
```

```
 REM INPUT r$
 LOCATE 24,5: PRINT d$;" Energy Density"
 FOR n%=1 TO 1000:NEXT
 REM INPUT r$

 LINE (250,350) - (400,50),,B

 k=300/360
 FOR n% = 1 TO 15
    IF n% = 2 THEN PSET (200*fl(n%)+250,350- k*dir(n%))
    IF dir(n%) < 999 THEN
       LINE - (200*fl(n%)+250,350-k*dir(n%) )
       LINE - (200*fu(n%)+250,350-k*dir(n%) )
    END IF
 NEXT
 REM INPUT r$
 LOCATE 24,35: PRINT d$;" Direction"
 FOR n%=1 TO 1000:NEXT
 REM INPUT r$

 LINE (450,350) - (600,50),,B

 k = 300/90
 FOR n% = 1 TO 15
    IF n% = 2 THEN PSET (200*fl(n%)+450,350-k*spr(n%))
    IF spr(n%) < 99 THEN
       LINE - (200*fl(n%)+450,350-k*spr(n%) )
       LINE - (200*fu(n%)+450,350-k*spr(n%) )
    END IF
 NEXT
 REM INPUT r$
 LOCATE 24,60: PRINT d$;" Spread"
 REM FOR n&=1 TO 3000&:NEXT

 INPUT "Enter P to produce CG file",R$
 IF R$ = "P" THEN
    SF$ = "Wooig8-CHC:CHC-mac:Swales:DWR:DWR" + DS$
    OPEN SF$ FOR OUTPUT AS #2
    PRINT#2,"*"
    PRINT#2,"Frequency (Hz)";CHR$(9);"Energy (m^2)";CHR$(9);"Direction
 (degrees)";CHR$(9);"Spread (degrees)"
    PRINT#2, 0;CHR$(9);0;CHR$(9);CHR$(9)
    PRINT#2, fl(1);CHR$(9);0;CHR$(9);CHR$(9)
    FOR f% = 1 TO 15
       IF f% > 1 THEN
          PRINT#2, fl(f%);CHR$(9);en(f%);CHR$(9);dir(f%);CHR$(9);spr(f%)
          PRINT#2, fu(f%);CHR$(9);en(f%);CHR$(9);dir(f%);CHR$(9);spr(f%)
       ELSE
          PRINT#2, fl(f%);CHR$(9);en(f%);CHR$(9);CHR$(9)
          PRINT#2, fu(f%);CHR$(9);en(f%);CHR$(9);CHR$(9)
    END IF
    NEXT
    PRINT#2, fu(15);CHR$(9);0;CHR$(9);CHR$(9)
    CLOSE#2

 END IF
 END
 nextrec:
 CLS

 WEND
 CLOSE#1
 END
```

```
DATA 512,256,128,64,32,16,8,8,8,8,8,8,16,32,30.12
```

APPENDIX B DATA FORMATS

Appendix B.1 Raw Data File

This file consists of binary data fields of 40 byte length; the first 3 fields are logger label messages, including the power-up time. Each of the remaining fields consists of 32 record bytes + 8 space bytes.

The 32 record bytes are the same as the 32 bytes transmitted in an ARGOS message and have the following significance:

Battery Voltage = 8.5 + Word1 / 16, range 8.5 to 22.5 Volts, where

Word1 = (byte1 & E0H)
(range 0 to 224)

R.M.S Wave height $\sqrt{m_o}$ = 16 * (exp(Word2 / 32) - 1), range 0 to 830.7 cm, where

Word2 = (byte2 & 7FH)
(range 0 to 127)

The directional spectrum is defined over 15 frequency bands, whose widths are defined by the data and are such that:

band 0 starts at 0.025 Hz, ends at $freq_0$ and contains 1/512 of the total energy

band 1 starts at $freq_0$, ends at $freq_1$ and contains 1/256 of the total energy

band 2 starts at $freq_1$, ends at $freq_2$ and contains 1/128 of the total energy

band 3 starts at $freq_2$, ends at $freq_3$ and contains 1/64 of the total energy

band 4 starts at $freq_3$, ends at $freq_4$ and contains 1/32 of the total energy

band 5 starts at $freq_4$, ends at $freq_5$ and contains 1/16 of the total energy

band 6 starts at $freq_5$, ends at $freq_6$ and contains 1/8 of the total energy

band 7 starts at $freq_6$, ends at $freq_7$ and contains 1/8 of the total energy

band 8 starts at $freq_7$, ends at $freq_8$ and contains 1/8 of the total energy

band 9 starts at $freq_8$, ends at $freq_9$ and contains 1/8 of the total energy

band 10 starts at $freq_9$, ends at $freq_{10}$ and contains 1/8 of the total energy

band 11 starts at $freq_{10}$, ends at $freq_{11}$ and contains 1/8 of the total energy

band 12 starts at $freq_{11}$, ends at $freq_{12}$ and contains 1/16 of the total energy

band 13 starts at $freq_{12}$, ends at $freq_{13}$ and contains 1/32 of the total energy

band 14 starts at $freq_{13}$, ends at 0.64 Hz and contains 1/30.12 of the total energy

$freq_0$ = 0.025 + 0.04 * (exp(Word3 / 32) - 1), range 0.025 to 0.2715, where

Word3 = 2 * (byte1 & 1FH) + (byte2 & 80H) / 128
(range 0 to 63)

and for band n, where n = 1 to 13

$freq_n - freq_{n-1}$ = 0.04 * (exp(Wordn+3 / 32) - 1), range 0 to 0.2465, where

Wordn+3 = (byte2n+1 & FCH) / 4
(range 0 to 63)

The above frequency bandwidths are used to calculate the spectral energy density, e.g. for band 5, the energy in the band is given by m_o / 16 and the bandwidth is ($freq_5 - freq_4$) so that the energy density is given by $m_o/(16 * (freq_5 - freq_4))$ in cm^2sec.

The mean direction is not given for band 0 but, for bands 1 to 13, it is given by:

Θ1n = 5.625 * Wordn+16, range 0 to 354.375 degrees, where

Wordn+16 = 16 * (byte2n+1 & 03H) + (byte2n+2 & F0H) / (16 range 0 to 63)

and, for band 14

Θ1 = 5.625 * Word30, range 0 to 354.375 degrees, where

Word30 = (byte28 & FCH) / 4
(range 0 to 63)

and the directional spread for bands 1 to 13 is given by:

Θ2n = 14.32 * (exp(Wordn+30 / 8) - 1), range , where

Wordn+30 = (byte2n+2 & 0FH)
(range 0 to 15)

The mean zero-crossing frequency Fz = $\sqrt{m_2 / m_o}$ is given by:

Fz = 0.32 * (exp(Word44 / 64) - 1), range 0 to 0.5364 Hz, where

Word44 = 16 * (byte28 & 03H) + (byte29 & F0H) / 16 (range 0 to 63)

Other data, more of a housekeeping nature, are derived as follows:

Status Bit = (byte29 & 08H), should be 0, if 1 this indicates a memory error

Vertical Acceleration offset compensation = 32 * (byte29 & 07H), range 0 to 224 cm sec^{-2}

North Acceleration offset compensation = (byte30 & E0H), range 0 to 224 cm sec^{-2}

West Acceleration offset compensation = 8 * (byte30 & 1CH), range 0 to 224 cm sec^{-2}

Hull Temperature = 0.05 * Word45 -5, range -5°C to 46.15 °C, where

Word5 = 256 * (byte30 & 03H) + byte31
(range 0 to 1023)

Summarising the data packing in tabular form we have, showing each byte as one line with the most significant bit to the left, from byte0 to byte31:

BATTERY	BATTERY	BATTERY	FREQ F0	FREQ F0	FREQ F0	FREQ F0	FREQ F0
FREQ F0	R.M.S.HT	R.M.S.HT	R.M.S.HT	R.M.S.HT	R.M.S.HT	R.M.S.HT	R.M.S.HT
F1-F0	F1-F0	F1-F0	F1-F0	F1-F0	F1-F0	Θ1-1	Θ1-1
Θ1-1	Θ1-1	Θ1-1	Θ1-1	Θ2-1	Θ2-1	Θ2-1	Θ2-1
F2-F1	F2-F1	F2-F1	F2-F1	F2-F1	F2-F1	Θ1-2	Θ1-2
Θ1-2	Θ1-2	Θ1-2	Θ1-2	Θ2-2	Θ2-2	Θ2-2	Θ2-2
F3-F2	F3-F2	F3-F2	F3-F2	F3-F2	F3-F2	Θ1-3	Θ1-3
Θ1-3	Θ1-3	Θ1-3	Θ1-3	Θ2-3	Θ2-3	Θ2-3	Θ2-3
F4-F3	F4-F3	F4-F3	F4-F3	F4-F3	F4-F3	Θ1-4	Θ1-4
Θ1-4	Θ1-4	Θ1-4	Θ1-4	Θ2-4	Θ2-4	Θ2-4	Θ2-4
F5-F4	F5-F4	F5-F4	F5-F4	F5-F4	F5-F4	Θ1-5	Θ1-5
Θ1-5	Θ1-5	Θ1-5	Θ1-5	Θ2-5	Θ2-5	Θ2-5	Θ2-5
F6-F5	F6-F5	F6-F5	F6-F5	F6-F5	F6-F5	Θ1-6	Θ1-6
Θ1-6	Θ1-6	Θ1-6	Θ1-6	Θ2-6	Θ2-6	Θ2-6	Θ2-6
F7-F6	F7-F6	F7-F6	F7-F6	F7-F6	F7-F6	Θ1-7	Θ1-7
Θ1-7	Θ1-7	Θ1-7	Θ1-7	Θ2-7	Θ2-7	Θ2-7	Θ2-7
F8-F7	F8-F7	F8-F7	F8-F7	F8-F7	F8-F7	Θ1-8	Θ1-8
Θ1-8	Θ1-8	Θ1-8	Θ1-8	Θ2-8	Θ2-8	Θ2-8	Θ2-8
F9-F8	F9-F8	F9-F8	F9-F8	F9-F8	F9-F8	Θ1-9	Θ1-9
Θ1-9	Θ1-9	Θ1-9	Θ1-9	Θ2-9	Θ2-9	Θ2-9	Θ2-9
F10-F9	F10-F9	F10-F9	F10-F9	F10-F9	F10-F9	Θ1-10	Θ1-10
Θ1-10	Θ1-10	Θ1-10	Θ1-10	Θ2-10	Θ2-10	Θ2-10	Θ2-10
F11-F10	F11-F10	F11-F10	F11-F10	F11-F10	F11-F10	Θ1-11	Θ1-11
Θ1-11	Θ1-11	Θ1-11	Θ1-11	Θ2-11	Θ2-11	Θ2-11	Θ2-11
F12-F11	F12-F11	F12-F11	F12-F11	F12-F11	F12-F11	Θ1-12	Θ1-12
Θ1-12	Θ1-12	Θ1-12	Θ1-12	Θ2-12	Θ2-12	Θ2-12	Θ2-12
F13-F12	F13-F12	F13-F12	F13-F12	F13-F12	F13-F12	Θ1-13	Θ1-13
Θ1-13	Θ1-13	Θ1-13	Θ1-13	Θ2-13	Θ2-13	Θ2-13	Θ2-13
Θ1-14	Θ1-14	Θ1-14	Θ1-14	Θ1-14	Θ1-14	FZ	FZ
FZ	FZ	FZ	FZ	MEM	ACC-V	ACC-V	ACC-V
ACC-N	ACC-N	ACC-N	ACC-W	ACC-W	ACC-W	TEMP	TEMP
TEMP	TEMP	TEMP	TEMP	TEMP	TEMP	TEMP	TEMP

Appendix B.2 Processed Spectral Files

These consist of a number of chronologically ordered spectra, each with accompanying time stamp, position and parameters; the format of each spectrum is as follows:

a) header information

```
jul.iand,la.tit,lon.gdi<CR>
varian.ce<CR>
```

where

`<CR>` = Carriage Return
`jul.iand` = Julian day, e.g. 305.7292
`la.tit` = Latitude if ARGOS fix available, else 99.999
`lon.gdi` = Longditude if ARGOS fix available, else 999.999
`varian.ce` = Variance of Wave Amplitude in cm^2

b) spectrum, consisting of 15 lines of the format

```
f.lower,f.upper,dir.ect,sp.red<CR>
```

where

`f.lower` = lower frequency of spectral band
`f.upper` = upper frequency of spectral band
`dir.ect` = mean wave direction for the band
`sp.red` = mean directional spread for the band

The bands contain, respectively, 1/512, 1/256, 1/128, 1/64, 1/32, 1/16, 1/8, 1/8, 1/8, 1/8, 1/8, 1/8, 1/16, 1/32, 1/30.12 of the total variance, so that the spectral energy density for a band is obtained bytaking the appropriate fraction of the total variance for the band and dividing by (`f.upper - f.lower`)

c) parameters

```
Hs.Hs,Tz.Tz,m,acv,acn,acw,+te.mp<CR>
```

where

`Hs.Hs` = the significant wave height in m
`Tz.Tz` = the mean zero crossing period in seconds
m = a memory check (normally 0)
`acv` = Vertical acceleration offset compensation (cm/sec^2)
`acn` = North acceleration offset compensation (cm/sec^2)
`acw` = West acceleration offset compensation (cm/sec^2)
`+te.mp` = sea surface temperature in ^{o}C

An example of a directional spectrum is given below:

```
313.3750,99.999,999.999
  3362.06
0.02500,0.06968,999.999,99.999
0.06968,0.08435,258.750,58.403
0.08435,0.08828,253.125,42.317
0.08828,0.09222,253.125,29.789
0.09222,0.10200,264.375,29.789
0.10200,0.12395,270.000,29.789
0.12395,0.14990,275.625,24.606
0.14990,0.16126,258.750,24.606
0.16126,0.16951,247.500,24.606
0.16951,0.18592,258.750,29.789
0.18592,0.20597,236.250,29.789
0.20597,0.27470,236.250,42.317
0.27470,0.33370,225.000,42.317
0.33370,0.39270,225.000,42.317
0.39270,0.63500,219.375,99.999
 2.32,  4.82,0,   0,   0,  32,+12.20
```

In the above example, taken from the EEPROM logger output, there was (naturally) no fix information available, so the latitude and longditude digits have been replaced with 9s. The direction and spread are not defined for the lowest band and the spread is not defined for the highest band; these values are, therefore, replaced with 9s.

As an example, energy density for the 6th band would be calculated as follows:

Energy density for band 0.10200 - 0.12395 Hz = 3362.06 / {16*(0.12395 - 0.10200)}

= 9573 cm^2/Hz

= 0.9573 m^{2}s

Appendix B.3 Processed Parameter Files

These files consist of a table consisting of chronologically ordered rows of the form:

```
jul.iand<tab>Hs.Hs<tab>Tz.Tz<tab>+te.mp<CR>
```

e.g.

```
.
.
316.2708(tab><sp>1.21<tab><sp>4.46<tab>+12.15<CR>
316.2917(tab><sp>1.27(tab><sp>4.63<tab>+12.15<CR>
316.3125(tab><sp>1.33(tab><sp>4.82<tab>+12.15<CR>
316.3333(tab><sp>1.39(tab><sp>4.82<tab>+12.15<CR>
316.3542(tab><sp>1.39(tab><sp>4.63<tab>+12.15<CR>
316.3750(tab><sp>1.52(tab><sp>4.82<tab>+12.10<CR>
316.3958(tab><sp>1.59(tab><sp>4.82<tab>+12.10<CR>
316.4167(tab><sp>1.52(tab><sp>4.63<tab>+12.15<CR>
.
.
```

Appendix B.4 CricketGraph Data

ARGOS Data

This is a tabular text file, suitable for entry into CricketGraph, produced by JRC; the header (labels) line consists of::

```
day<tab>hhmm$<tab>lat<tab>dlat<tab>long<tab>dlon<tab>
Hs<tab>Tz<tab>Temp<tab>jday<CR>          (all in one line)
```

followed by a number of rows of the above data in exponential format, e.g.

```
3.013584e+2<tab>8.36e+2<tab>5.1487e+1<tab>-7.788306e-1<tab>
3.5525e+2<tab>2.806497e-1<tab>9.9e-1<tab>3.37e+0<tab>
1.3e+1<tab>3.013584e+2<CR>          (all in one line)
```

In this example,

the Day (day) is 301.3584
the Time (hhmm$) is 0836
the Latitude (lat) is 51.487
the latitude difference (dlat) is -0.7788306 (km relative to 51.494 ^{O}N)
the Longditude (lon) is 355.25 (degrees East)
the longditude difference (dlon) is 0.2806497 (km relative to 355.246 ^{O}E)
the Significant Wave Height (Hs) is 0.99 (m)
the Mean Zero-crossing Period is 3.37 (seconds)
the Temperature (Temp) is 13 (degrees C)
the Julian Day (jday) is 301.3584 (incrementing over the New Year 1993/94)

EEPROM Data

These are tabular text files, suitable for entry into CricketGraph; the header (labels) line consists of:

```
Day<tab>Hs<tab>Tz<tab>Temperature<CR>
```

followed by a number of rows of the above data in exponential format, e.g.

2.999375e+2<tab>7.6e-1<tab>3.26e+0<tab>1.3e+1<CR>
2.999583e+2<tab>8e-1<tab>3.37e+0<tab>1.305e+1<CR>
2.999792e+2<tab>7.1e-1<tab>3.48e+0<tab>1.3e+1<CR>
3e+2<tab>7.1e-1<tab>3.72e+0<tab>1.305e+1<CR>
3.000208e+2<tab>6.7e-1<tab>3.6e+0<tab>1.31e+1<CR>
3.000417e+2<tab>6.7e-1<tab>3.72e+0<tab>1.315e+1<CR>
3.000625e+2<tab>8e-1<tab>3.72e+0<tab>1.315e+1<CR>
3.000833e+2<tab>8e-1<tab>3.6e+0<tab>1.305e+1<CR>
3.001042e+2<tab>9.4e-1<tab>3.6e+0<tab>1.305e+1<CR>
3.00125e+2<tab>8.5e-1<tab>3.48e+0<tab>1.305e+1<CR>