THE DATA LOGGING SYSTEM OF R.V. "G.O.SARS"

DESCRIPTION OF SOFTWARE

by

Johan Blindheim and Per Eide
Fiskeridirektoratets Havforskningsinstitutt
Boks 2906, 5011 Bergen-Nordnes

Editor
Erling Bratberg

SERIE B NR. 8
1973
<table>
<thead>
<tr>
<th>CONTENTS</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>INTRODUCTION</td>
<td>1</td>
</tr>
<tr>
<td>SYSTEM CONFIGURATION, HARDWARE</td>
<td>1</td>
</tr>
<tr>
<td>DESCRIPTION OF THE SOFTWARE</td>
<td>3</td>
</tr>
<tr>
<td>DISCUSSION</td>
<td>14</td>
</tr>
<tr>
<td>REFERENCES</td>
<td>15</td>
</tr>
<tr>
<td>APPENDIX 1</td>
<td>1</td>
</tr>
<tr>
<td>APPENDIX 2</td>
<td>2</td>
</tr>
<tr>
<td>APPENDIX 2</td>
<td>2</td>
</tr>
</tbody>
</table>
INTRODUCTION

When the Institute of Marine Research planned its research vessel "G.O.Sars", it was decided to include a computer centered data acquisition system in the equipment of the ship. The intention with the system was to obtain automatic and accurate data collection from various instruments which should be connected on-line to the computer. A need for this was particularly felt in the use of echo sounders which had developed requirements for data acquisition at rates which were impossible by more conventional methods. The computer should also perform data reduction and necessary processing in real time. Reduced data and real-time products should be stored on paper tape for further management onboard or ashore.

The contract for planning the system, including choice of computer with peripherals and construction of interface units for the instruments to be logged, was placed with the Central Institute for Industrial Research, Oslo. The software of the system was made by this institute and the Institute of Marine Research in cooperation. The hardware, and also parts of the software, of the system is earlier presented by JAHR et al. 1970 and a paper on the use of the system is earlier presented by BLINDHEIM and EIDE 1971.

Since the system became operational and experience in practical use has been gained, several modifications have been made. This report describes the modified software of the system and summarizes briefly its hardware configuration.

SYSTEM CONFIGURATION, HARDWARE

The data logging system is built around a NORD-1 computer which has paper tape input/output (I/O) units. A block diagram of the system is shown in Fig.1 and specifications for the different peripheral devices are entered in Table 1. As indicated in Fig.1, the system is collecting data from a wide
variety of instruments within the fields of oceanography, meteorology, hydroacoustics and navigation. Many of the instruments give analog inputs and are connected to the computer via a multiplexer and an analog to digital converter (ADC). Other instruments produce digital outputs. These are given device numbers and connected directly to the I/O channels of the computer.

The data acquisition from these sources is governed by interrupts generated by the real-time clock and the ship's log. From the log an interrupt is given every tenth of a nautical mile outsailed distance and the interrupts from the real-time clock occur at a rate of one Hz. More details about the sampling which is performed on log interrupt is entered in Table 2, and in Table 3 is the sampling activities occurring on each clock interrupt summarized.

In some case it is necessary to sample the echo sounders at rates up to 50 kHz. For this reason the echo sounders can be connected to the computer via an acoustic interface unit (Simrad Computer Interface Unit) and a fast ADC. The interface unit gives interrupt to the computer when sampling shall start and stop, the interrupt occurring when the echoes are received from the upper and lower limits of the depth interval to be sampled. In the interface unit the timing of the interrupts is controlled by an adjustable electronic counter. The greatest possible depth to the upper limit is 400 m and the magnitude of the depth interval itself can not exceed 200 m. This corresponds to a time interval of approximately 0,3 sec. In order to avoid loss of interrupt from the lower priority real-time clock during these periods, the low interrupt rate of one Hz has been chosen for the clock.

One of the teleprinters which is involved in the system serves as operators keyboard terminal in the acoustic instrument room. The rest of the system is placed in one room where the computer, the papertape I/O units and the interface electronics occupy three instrument racks. Both these rooms are on the
bridge deck level adjacent to the Operations Centre from where the research activities of the ship are currently coordinated.

DESCRIPTION OF THE SOFTWARE

The standard software of the NORD-1 comprises the programming languages MAC Assembly and debugging system, FORTRAN II, FORTRAN IV and BASIC. Considering the requirements it was evident that assembly coding had to be applied for the programming of the system. During its further design two characteristic features had to be considered: Firstly that the hardware configuration made it necessary that most of the system software should be core resident, and secondly that some of the data collection should be done at a speed which required the full capacity of the computer. During the programming phase it was, therefore, thought more of compactness and speed than of generality. One of the means for obtaining this was to utilize the efficient priority interrupt system of the computer so that time shearing could be applied for the different data acquisition programs and processing routines which accordingly were placed on the different priority interrupt levels. Here the priority of the data collection and processing were given priority according to sampling rate and necessary speed. Slower procedures as for instance output were given low priority. Further a compact operation system was incorporated and re-entrant I/O routines were made specially for the system.

Since the system is supposed to be working constantly when the ship is at sea, it was considered important to have also the possibility to apply more casual programs when the system is working. This is achieved by placing the MAC assembler on Level 1. In retaliation some programs belonging to the system, but not in regular use, are read into the memory by means of the assembler only when they are needed. This is done in order to reserve a sufficient part of the core store for general use.

A listing of the system software is entered in Appendix 1 and a short description of the programming on each level is compiled in the following.
LEVEL 15

Start: Interrupts from the acoustic interface unit.

Operations: Identification of interrupts for start or stop of data collection from the echo sounder. Reading of counter indicating depth of acoustic sampling.

Products: Interrupts to Level 14.

Description: The fast data collection from the echo sounders which requires the full speed capacity of the computer, is given the highest priority. The acoustic interface unit gives interrupt on Level 15 to start or stop the sampling. These interrupts are identified and interrupts are given to Level 14 accordingly. Further is an electronic counter in the acoustic interface unit read when the interrupts are given. According to the interrupt these readings give the depth to, and the magnitude of the interval to be sampled. The maximum possible depth to the upper limit of the interval is 400 m and the magnitude of the interval can not exceed 200 m.

LEVEL 14

Start: Interrupt from Level 15.

Operations: Reading of echo signals from echo sounder via 10-bit ADC. Reduction and processing of the collected data.

Products: Characteristic parameters of echoes from single targets. Interrupt to Level 5 for output when adequate.

Description: The intention with the fast sampling of the echo signals which is done on this level is classification of echo traces based on reflection patterns of single targets. For this reason a procedure is applied which
utilizes all echoes received from a single fish when it passes through the acoustic beam from the echo sounder. The method is described by MIDTTUN and NAKKEN 1971 and here shall only be summarized what is done in the computer. The processing of the echo signals is split into three phases. About 20 micro seconds between each sample is utilized for the first phase. Here the program keeps record of when the fish comes into and fades out of the acoustic beam. Before the sample is considered valid and stored in memory, tests on threshold and pulse length are also performed. The sampling may go on up to about 0.3 seconds on each transmission (when the interval to be sampled is 200 m) and the rest of the time before the next pulse transmission is more than sufficient for the second phase of processing and data reduction. During this phase the maximum amplitude of the current echo signal is recorded, the echo integral and the pulse length at 50% of maximum amplitude is computed. This is done for each ping when the target passes through the acoustic beam. After the target has faded out of the beam the depth to the target is worked out and also the reflection angle of the target given by

$$\theta = 2 \arctan \frac{v(n+1)}{2DP}$$

where \(v\) is the speed of the ship in \(\text{cm/sec.}\), \(n\) is the number of echoes received from the target, \(D\) is the depth to the target in \(\text{cm}\) and \(P\) is the repetition rate of the echo sounder in number of transmission per sec. The echo strength of each echo is computed by

$$Es = k + 20 \log aD$$

where \(k\) is constant, \(a\) is the voltage sampled from the echo sounder and \(D\) is the depth to the target.

A program which optionally can be combined with the classification program plots the envelope curve of the echo
signals on the drum plotter. An example of this is shown in Fig. 2.

On Level 14 there is also an option for getting the specter of echo strengths received from the sampled interval in a chosen number of transmission. In that case the computer observes the peak values in all echoes received. Every signal exceeding the noise threshold is then considered as an echo.

An option for processing of bottom back scattering data is also included. It has, however, not been much applied as yet.

LEVEL 13

Start: Interrupt from the real-time clock.

Operations: Data acquisition from gyro compass, ships log, meteorological instruments, digital depth recorder, echo integrators and STD system.

Products: Position (updated by dead reckoning), N-S and E-W components of the ships speed, wind speed and direction, date and time. Interrupts to Level 12, Level 5, Level 4 and Level 3 for Decca position, punching of STD data, routine output and listing of STD data respectively.

Description: The signals from the gyro compass are digitized in the interface (coding disc.). Decoding of the digitized input is done by software and correction for the ship's speed and latitude is done by

\[ \theta = \arctan \left( \frac{v \cdot \sin C}{900 \cos \phi \sin C} \right) \]

where \( \theta \) is the correction of the course in radians, \( C \) is the uncorrected course of the ship obtained from the gyro compass and \( v \) is the ship's speed and \( \phi \) is the latitude.
From the course and speed of the ship N-S and E-W components of its velocity are worked out. These are used in the updating of the ship's position by dead reckoning and further when working out true wind speed and direction from the relative wind observed on the moving ship.

The interrupts from the real-time clock are also used for updating the time every second and the date when adequate.

The STD-data are received in the form of frequencies proportional to salinity, temperature and depth. These are fed into the computer via frequency counters and the parameters are derived by linear interpolation within the frequency interval defined for each parameter. Instrument calibration constants are supplied to the computer by the operator and are applied in the program. Computations of $\varrho_t$ (Density of the sea water), $\Delta \alpha$ (specific volume anomaly) and $\Delta D$ (dynamic depth anomaly) are done in real time and can be printed out according to program, normally every 5 second. An example of STD-listing with depth, temperature and $\varrho_t$ is shown in Fig. 6. The computation of $\varrho_t$ is done as given by KNUDSEN (1901) and rearranged by SÆLEN 1958.

$$\varrho_t = D + (C+0.1324)(1-AT+BT(C-0.1324))$$

where

$$D = - \frac{(t-3.98)^2(t+283)}{503.57(t+67.26)}$$

$$C = 28.1263+0.806(S-35)+(S-35)^2 \cdot 0.23 \cdot 10^{-3}+(S-35)^3 0.68 \cdot 10^{-5}$$

$$AT = (4.7867 \cdot t-0.098185 \cdot t^2+0.0010843 \cdot t^3) \cdot 10^{-3}$$

$$BT = (18.03 \cdot t-0.8164 \cdot t^2+0.01667 \cdot t^3) \cdot 10^{-6}$$

$t$ is the temperature in °C and $S$ is the salinity.
The computation of $\omega$ is done as given by BJERKNES (1910) and SVERDRUP (1933).

$$10^5\omega = (\alpha(s,t)-0.972643) \cdot 10^5 \cdot (1-4.66 \cdot 10^{-6} \cdot p) + \alpha(s,t) \cdot p \cdot 10^{-4} (G(t,P)+ \frac{C-28}{10} \cdot H(t,P)-1.85+p \cdot 4 \cdot 10^{-5})$$

where

$$\alpha(s,t) = \frac{1}{1+\gamma \cdot 10^{-3}}$$

and

$$G(t,P) = 28.33 \cdot t - 0.551 \cdot t^2 + 0.004 \cdot t^3 - 10^{-4} \cdot (9.5-0.158 \cdot t) + 1.5 \cdot 10^{-8} \cdot p^2$$

$$H(t,P) = 147.3-2.72 \cdot t + 0.04 \cdot t^2 - 10^{-4} \cdot (32.4-0.87 \cdot t + 0.02 \cdot t^2)$$

$\gamma$ is the same as in the computation of $\eta$, $t$ is temperature, $S$ is salinity and $P$ is the pressure in d bar. $10^4 \Delta D$ is worked out by integration of $\Delta \omega$.

$$10^4 \Delta D = \int_{P=0}^{P=P_1} 10^5 \Delta \alpha \, dp$$

Here $P$ is simply exchanged by the depth in metres which introduces only a minor error.

LEVEL 12

Start: Interrupt from Level 13.

Operations: Data acquisition from DECCA navigator.

Products: DECCA position.
Description: The signals from the DECCA navigator are read every second and the position is computed in degrees and minutes of latitude and longitude. The applied geometry is explained more in detail by J. CASPERSEN in a note which is entered in Appendix 2 (in Norwegian). When initiating this program the operator has to feed the current DECCA coordinates into the computer together with identification number of the DECCA chain to be applied as explained more in detail in Appendix 3. Fixed constants for the various chains must be kept in program.

LEVEL 11.

Start: Interrupt from teleprinter, paper tape reader or the ship's log.

Operations: Identification of interrupts.

Products: Interrupt to Level 10 on interrupt from the log and interrupt to Level 9 on interrupt from paper tape reader or teleprinters.

Description: When input shall come from any of the teleprinters or from the paper tape reader, an interrupt is wired in hardware to occur on this level. The ship's log also give an interrupt on this level for every tenth of a nautical mile outsailed distance.

The programming identifies the interrupts and gives interrupt to other levels accordingly. The activities which are governed by interrupts from the log are placed on Level 10, and Level 9 is activated if input from the other devices generates interrupt.

LEVEL 10

Start: Interrupt from Level 11.
Operations: Data acquisition of sea surface temperature, sea surface salinity and transparency of the sea water.

Products: Average of the ship's speed, temperature, salinity and transparency in the surface layer of the sea for each nautical mile outsailed distance. Echo abundance per nautical mile. Interrupt to Level 4 and Level 3 for output when adequate.

Description: Activities governed by interrupts from the ship's log are placed on Level 10. The log gives an interrupt every tenth of a nautical mile outsailed distance and initiates data collection routines for the parameters to be observed in the surface layer of the sea. Associated data reduction and averaging, for instance per nautical mile, is further done according to program. The program also give interrupt to Level 4 for routine output on paper tape and teleprinter, normally for every nautical mile outsailed distance.

Also the echo abundance value is accumulated over an adequate distance, normally 5 nautical miles, which can be set in program by the operator. Interrupt is given to Level 3 for output.

LEVEL 9

Start: Interrupt from teleprinter keyboard.

Operations: Initiation and termination of programs as adequate. Printout of current value of any parameter on request.

Description: The programming for the operation of the system occupies Level 9. This comprises initiation of the different data acquisition programs including incertion of initial values of some parameters which are used in associated processing. In some of the routines it is also allowed for correction of current values when necessary. This is for instance the case in the routine
for the dead reckoning position.

More details about this operation system are entered in Appendix 3.

LEVEL 8

This Level is not used.

LEVEL 7

Start: Interrupt from teleprinters or paper tape punches.

Operations: Identifications of interrupts.

Products: Interrupts to the level on which the interrupting device was triggered.

Description: If output is performed when the interrupt system is on, the output devices are programmed to give interrupt to Level 7 when each output instruction is executed. The program on Level 7 identifies the interrupting device and gives interrupt to the level where the device is working.

LEVEL 6

This Level is not used.

LEVEL 5

Start: Interrupt from Level 14 or Level 13.

Operations: Output of acoustic data and STD-data.

Products: The programming on this level comprises routines for printout of echo classification data and echo strength data, and further for output of STD-data on
paper tape. For the acoustic data only data lists are produced, and examples of these are shown in Figs. 3 and 4.

The output rate of the STD-data on paper tape is one set of parameters per second.

LEVEL 4

Start: Interrupt from Level 13 or Level 10.

Operations: Output on punch and teleprinter every nautical mile and every hour.

Products: Data record on paper tape and lists. Interrupt to Level 3 for output of echo abundance.

Description: This level holds the programs for output on teleprinter and punch every whole nautical mile outsailed distance and every whole hour, triggered by interrupt from the ship's log and the real-time clock respectively. An example of such routine listings are shown in Fig. 5.

The data are simultaneously punched on paper tape together with identification characters for each parameter.

LEVEL 3

Start: Interrupt from Level 13 of Level 3.

Operations: Output on teleprinter.

Products: Data lists for STD-data and echo abundance.

Description: This level holds output program for the echo abundance data obtained from the echo integrators. These data are produced only on data lists since the
next step in their management has to be done manually.

The program is triggered by interrupts originally coming from the ship's log and the distance travelled between each printout may be set by program, normally a distance of 5 nautical miles is chosen.

The program for listing of STD-data is also placed on this level. It is governed by the real-time clock and the printout rate may be chosen by program.

The printout of the three derived parameters, $\omega$, $\Delta \alpha$ and $\Delta D$ is optional. The maximal printout rate allowed by the capacity of the teleprinter is one set of parameters every fifth to seventh second depending on the parameters printed. An example of an STD-listing is shown in Fig.6.

LEVEL 2

This Level is not used.

LEVEL 1

Start: Interrupt from the operation system on Level 9.

Operations: Application of assembler when the system is running.

Description: The MAC-assembler, the assembly language of the computer, is placed on Level 1. This facilitates the application of the system by enabling programs to be assembled and executed in time shearing with the rest of the system. Further the operating procedures become more flexible since it gives access to all debugging facilities in the assembler. It also offers possibility of leaving out of the system programs that only occasionally are used in real time. This is for instance done with the voluminous routines for the drum plotter.
LEVEL 0

Start: When priority allows.

Operations: Waiting mode or operations according to background program.

Products: According to background program, normally none.

Description: When no program on higher priority interrupt levels is busy, the control is given to Level 0. When the computer is working on Level 0 it is consequently in a waiting mode, but it may still be executing an application program. In this system Level 0 has been much utilized to hold programs for updating the numerical display which may be displaying any parameter as for instance the current position of the ship.

DISCUSSION

Introductorily the intentions with the system was outlined and it appears from this that the two main tasks for the system is firstly data acquisition and secondly storing of reduced and preliminary processed data on paper tape.

The data storage on paper tape has offered some problems. During the planning phase of the system paper tape was considered as a reliable data storage medium. The amount of data which is produced by the system described above is, however, somewhat too large for storage on paper tape. The resulting quantities of tape are very clumsy to handle and easily broken. This makes the further data management onboard rather time consuming and is to some degree a limiting factor.

It has also been difficult to keep the paper tape punches working satisfactorily over periods like a research cruise of about a month. The punches which originally belonged to the equipment were very problematic and had to be exchanged. The substitutes
are far better, but even with these it occurs that erroneous characters are punched, particularly just after the start of the punch.

As to the data logging from the different instruments the system has worked quite satisfactorily. Particularly in connection with data collection from the STD-system, and also from the acoustic instruments, it has proved a valuable tool. The advantage here is that the computer presents digital data in real time (as far as the STD-system concerns also derivatives) where the instruments present only an analog paper record. To digitize these data would otherwise be a cumbersome and lengthy job, and it would not be possible to obtain the accuracy of the computer.

In the present and previous use of the system it's flexibility has been sufficient. In particular has the possibility of using the assembler on Level 1 been useful as well in the use of external programs as in on-line control of the system.

REFERENCES


SVERDRUP, H.U. 1933. Vereinfachtes Verfahren zur Berechnung der

SÆLEN, 0.H. 1958. Punched-card treatment of hydrographic observ-
Table 1. Specifications of the computer and peripheral devices connected to it.

<table>
<thead>
<tr>
<th>UNIT</th>
<th>SPECIFICATIONS</th>
</tr>
</thead>
<tbody>
<tr>
<td>PAPER TAPE PUNCH (two units)</td>
<td>Type: FACIT 6040</td>
</tr>
</tbody>
</table>
| PAPER TAPE READER          | Type: Elliot TRM 250  
                            | Speed: 250 characters/S  
                            | Rack mounted |
| KEYBOARD TERMINAL (tree units) | Type: Data Dynamics ASR - 33 teletypes  
                                | Speed: 10 characters/S |
| DRUM PLOTTER               | Type: Calcomp 563  
                            | Speed: 300 steps/S  
                            | Resolution: 0.1 mm  
                            | Plotting width 30 inches |
| MULTIPLEXER AND A/D CONVERTER | Type: Raytheon Miniverter  
                                | Number of channels: 24 (extendable to 64)  
                                | Number of bits: 12  
                                | Through-put rate: 35 kHz  
                                | Input voltage: ± 10 Volts |
| A/D CONVERTER              | Type: Raytheon Miniverter  
                            | Number of bits: 10  
                            | Through-put rate: 50 kHz  
                            | Input voltage: ± 10 Volts |
| NUMERIC DISPLAY            | Type: LED display made at Central Institute for Industrial Research.  
                            | 12 digits in four groups. |
| NUMERIC INPUT              | Type: Thumbwheel switch input made at Central Institute for Industrial Research.  
<pre><code>                        | 12 digits in four groups. |
</code></pre>
<p>| BOUY DATA RECORDER         | Type: TANDBERG 1600x, modified by Chr. Michelsens institute. For input from magnetic data tapes produced in oceanographic buoy instruments. |</p>
<table>
<thead>
<tr>
<th>UNIT</th>
<th>SPECIFICATIONS</th>
</tr>
</thead>
<tbody>
<tr>
<td>COMPUTER, NORD-1</td>
<td>CORE STORE: 16 K (extendable up to 64 K)</td>
</tr>
<tr>
<td></td>
<td>WORDSIZE: 16 bits</td>
</tr>
<tr>
<td></td>
<td>CYCLE TIME: 1.7 μs</td>
</tr>
<tr>
<td></td>
<td>CPU: Hardware floating point arithmetic, 7 programmable registers, control flip-</td>
</tr>
<tr>
<td></td>
<td>flops and full parallel operation.</td>
</tr>
<tr>
<td></td>
<td>INTERRUPT SYSTEM: 16 priority interrupt levels with multiprogramming system.</td>
</tr>
<tr>
<td></td>
<td>OPTIONS: Two direct memory access channels with a total transfer rate of</td>
</tr>
<tr>
<td></td>
<td>588 K words/s.</td>
</tr>
<tr>
<td></td>
<td>Two I/O channels with capacity of up to 256 device numbers in four groups.</td>
</tr>
<tr>
<td></td>
<td>Real time clock.</td>
</tr>
<tr>
<td></td>
<td>Memory protect system.</td>
</tr>
</tbody>
</table>
Table 2. Sampling activities on log interrupt every tenth of a nautical mile. Real time processing on log interrupt.

<table>
<thead>
<tr>
<th>Instrument</th>
<th>Parameter</th>
<th>Type of input</th>
<th>Data processing and reduction in real time.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Thermosalinograph</td>
<td>SST</td>
<td>Analog</td>
<td>Mean sea surface temperature per n. mile.</td>
</tr>
<tr>
<td></td>
<td>SSS</td>
<td>Analog</td>
<td>Mean sea surface salinity per n. mile.</td>
</tr>
<tr>
<td>Transparencymeter</td>
<td>Transparency</td>
<td>Analog</td>
<td>Mean transparency per n. mile.</td>
</tr>
<tr>
<td>Echo-integrators</td>
<td>Echo-abundance</td>
<td>Analog</td>
<td>Echo abundance integrated over latest n. mile.</td>
</tr>
<tr>
<td>Real-time clock</td>
<td>Mean speed</td>
<td></td>
<td>Mean speed through latest n. mile.</td>
</tr>
</tbody>
</table>
Table 3. Sampling activities and real-time processing every second on interrupt from the real-time clock.

<table>
<thead>
<tr>
<th>Instrument</th>
<th>Parameters</th>
<th>Type of input</th>
<th>Processing and data reduction in real time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gyro</td>
<td>Course</td>
<td>Digital</td>
<td>N-S and E-W components from course and speed. Dead reckoning position.</td>
</tr>
<tr>
<td>Ships log</td>
<td>Speed</td>
<td>Analog</td>
<td>Decca position</td>
</tr>
<tr>
<td>Decca navigator</td>
<td>Red, green and purple lane</td>
<td>Analog</td>
<td></td>
</tr>
<tr>
<td>STD-system</td>
<td>Salinity, Temperature, Depth</td>
<td>Digital &quot;</td>
<td>Values obtained from instrument frequencies and adjusted according to instrument calibrations, products computed (σ_t, Δa and ΔD).</td>
</tr>
<tr>
<td>Depth indicator</td>
<td>Echo depth</td>
<td>Digital</td>
<td></td>
</tr>
<tr>
<td>Echo integrator (6 channels)</td>
<td>Echo abundance</td>
<td>Analog</td>
<td>Accumulation</td>
</tr>
<tr>
<td>Air-thermometer</td>
<td>Air temp.</td>
<td>Analog</td>
<td>Mean for last ten minutes</td>
</tr>
<tr>
<td>Psychrometer</td>
<td>Dew point temperature</td>
<td>Analog</td>
<td>Mean for last ten minutes</td>
</tr>
<tr>
<td>Barometer</td>
<td>Air pressure</td>
<td>Analog</td>
<td>Mean for last ten minutes</td>
</tr>
<tr>
<td>Anemometer</td>
<td>Wind speed</td>
<td>Analog</td>
<td>Mean for last ten minutes</td>
</tr>
<tr>
<td>Wind wane</td>
<td>Wind direction</td>
<td>Analog</td>
<td>Mean for last ten minutes</td>
</tr>
</tbody>
</table>
Fig. 1. Block diagram of the data logging system on R.V. "G.O.Sars".
Fig. 2. Plot of the envelope of five echoes from a single fish as it passed through the acoustic beam. The echo sounder was sampled at a rate close to 50 kHz.

Fig. 3. Print out from the echo classification program.
YES?
BG
GAIN -32
RATE 48
TOTAL 200
FRAME 0
INTEGR. -1
MAX DB -20

| DATE 71 05 07
| TIME 22 38 61 5.6 | LAT | LONG | LOG | SPD | STMP | SSAL | TRSP | DPT | CRS |
|-----|----------------|-----|------|-----|-----|------|------|------|-----|-----|
| 1   | -34            | 71  | 21.7 | 325.5 | 2.2 | 9.5 | 32.3 | 662 | 691 | 311 |
| 2   | -35            |     |      |       |     |     |      |     |     |     |
| 4   | -36            | 6   | 37   |       |     |     |      |     |     |     |
| 6   | -37            | 14  | 38   |       |     |     |      |     |     |     |
| 57  | -39            | 60  | -40  |       |     |     |      |     |     |     |
| 65  | -41            | 85  | -42  |       |     |     |      |     |     |     |
| 105 | -43            | 92  | -44  |       |     |     |      |     |     |     |
| 112 | -45            | 96  | -46  |       |     |     |      |     |     |     |
| 97  | -47            | 123 | -48  |       |     |     |      |     |     |     |
| 116 | -49            | 90  | -50  |       |     |     |      |     |     |     |
| 102 | -51            | 23  | -52  |       |     |     |      |     |     |     |
| 6   | -53            | 2   | -54  |       |     |     |      |     |     |     |
| 3   | -58            | 18  | -59  |       |     |     |      |     |     |     |
| 58  | -60            | 46  | -61  |       |     |     |      |     |     |     |
| 31  | -62            | 24  | -63  |       |     |     |      |     |     |     |
| 13  | -64            | 61  | -65  |       |     |     |      |     |     |     |
| 8   | -66            | 3   | -67  |       |     |     |      |     |     |     |

Fig. 4. Printout of the echo strength distribution.
YES?
BG
GAIN -82
RATE 48
TOTAL 200
FRAME 0
INTEGR. -1
MAX DB -20

DATE 71 05 07
TIME LAT LONG LOG SPD STMP SSAL TRSF LPT CHS
22 38 61 5.6 -7 21.7 325.5 2.2 9.5 32.3 66.2 691 311
1 -34
2 -35
4 -36
6 -37
14 -38
57 -39
60 -40
65 -41
85 -42
105 -43
92 -44
112 -45
96 -46
97 -47
123 -48
116 -49
93 -50
102 -51
23 -52
6 -53
2 -54
3 -58
18 -59
58 -60
46 -61
31 -62
24 -63
13 -64
61 -65
8 -66
3 -67

Fig. 4. Printout of the echo strength distribution.
DATE: 73 05 04
TIME LAT LONG LOG SPD STMP SSAL TRSP DPT CRS
17 27 59 22.0 3 35.6 747.0 3.7 2.8 13.8 20.3 242 358
17 33 59 23.1 3 35.5 748.0 10.2 7.1 34.4 40.9 241 356
17 39 59 24.1 3 35.4 749.0 9.8 7.1 34.5 41.6 242 358
17 46 59 25.2 3 35.2 750.0 8.8 7.1 34.5 41.1 243 356
17 51 59 26.2 3 35.1 751.0 10.7 7.0 34.4 40.8 244 359

TIME LAT LONG LOG PRES AIRT DEWP LUXM WF WDIR
18 00 59 26.3 3 35.2 751.0 01010.6 7.3 6.0 0.0 16 72

TIME LAT LONG LOG SPD STMP SSAL TRSP DPT CRS
18 20 59 27.3 3 35.0 752.0 2.1 6.9 34.2 40.3 242 356
18 26 59 28.3 3 34.9 753.0 9.9 6.9 34.2 39.7 242 357
18 33 59 29.4 3 34.8 754.0 8.7 6.9 34.1 39.6 241 356

TIME LAT LONG LOG PRES AIRT DEWP LUXM WF WDIR
19 00 59 30.2 3 34.8 754.8 1010.3 7.7 5.0 0.0 13 84

Fig. 5. Routine printout on interrupt from the real-time clock and from the ship's log.

ST NO: 431
OBSERVATIONS: 6
KORREKSJON FOR DYP: 0.0 TEMPERATUR: 0.00 SALTHOLDIGHET: -0.060

DATE: 73 05 27
TIME LAT LONG LOG SPD STMP SSAL TRSP DPT CRS
14 45 76 47.0 14 13.0 653.2 11.5 -0.7 34.1 31.8 79 235

DEPTH TEMP° SAL° SIGMA-T D-ALFA DELTA-D
4 -0.71 34.10 27.44 65.32 5160.90
4 -0.71 34.11 27.44 64.79 5160.90
4 -0.75 34.11 27.45 64.40 5160.90
8 -0.73 34.09 27.43 66.03 5186.05
13 -0.76 34.06 27.40 68.51 5223.85
19 -0.82 34.10 27.43 65.39 5261.45
23 -0.84 34.10 27.44 65.31 5286.48
29 -0.84 34.10 27.44 65.03 5323.84
36 -0.85 34.16 27.48 60.72 5372.19
42 -0.80 34.24 27.55 54.06 5404.26
48 -0.75 34.31 27.61 48.99 5433.77
53 -0.16 34.45 27.69 41.31 5456.05
61 1.51 34.76 27.84 27.15 5469.22
65 2.07 34.91 27.91 20.61 5479.31
69 2.64 35.02 27.95 16.73 5486.00

Fig. 6. Listing of STD-data. Sigma-t, D-alfa and Delta-D are computed simultaneously with the lowering of the probe.
PROGRAM LISTING
<table>
<thead>
<tr>
<th>COMMON AREA</th>
<th>POINTERS TO SUB-ROUTINES</th>
</tr>
</thead>
<tbody>
<tr>
<td>EXP.EXP1</td>
<td>% EXPONENTIAL FUNCTION</td>
</tr>
<tr>
<td>SQRT.ROT1</td>
<td>% SQUARE ROOT</td>
</tr>
<tr>
<td>S H S 1</td>
<td>% SINE FUNCTION (RADIANS)</td>
</tr>
<tr>
<td>COS.COS1</td>
<td>% COSINE FUNCTION (RADIANS)</td>
</tr>
<tr>
<td>TAN.TAN1</td>
<td>% TANGENT FUNCTION (RADIANS)</td>
</tr>
<tr>
<td>ATAN.ATAN1</td>
<td>% ARCCOS TANGENT (RADIANS)</td>
</tr>
<tr>
<td>ALOG.ALOG1</td>
<td>% NATURAL LOGARITHM FUNCTION</td>
</tr>
<tr>
<td>ASCI.ASCI1</td>
<td>% TWO DIGIT NUMBER OUTPUT</td>
</tr>
<tr>
<td>AV1.AVF</td>
<td>% DATA REDUCTION, AVERAGING</td>
</tr>
<tr>
<td>CLF.CLF1</td>
<td>% CVLF OUTPUT</td>
</tr>
<tr>
<td>DATE.DTE1</td>
<td>% DATE OUTPUT</td>
</tr>
<tr>
<td>DORT.DORTE1</td>
<td>% FLOATING OUTPUT</td>
</tr>
<tr>
<td>FLIN.FLINT</td>
<td>% FLOATING INPUT</td>
</tr>
<tr>
<td>HT1.FRI</td>
<td>% INTEGER INPUT</td>
</tr>
<tr>
<td>HTD.MELT*</td>
<td>% INTEGER OUTPUT</td>
</tr>
<tr>
<td>INP2.DINF</td>
<td>% POSITION OUTPUT (DEGREES AND MINUTES)</td>
</tr>
<tr>
<td>KONV.KONV1</td>
<td>% TWO CHARACTERS INPUT</td>
</tr>
<tr>
<td>MPY.MPY1</td>
<td>% READING MULTIPLEXER</td>
</tr>
<tr>
<td>OR100.FLOTS</td>
<td>% CALCOMP FLOTTED, SETTING OR100</td>
</tr>
<tr>
<td>PLFLT.NUMBER</td>
<td>% &quot;&quot; &quot;&quot; FLOATING OUTPUT</td>
</tr>
<tr>
<td>FLYTT.SYMBL</td>
<td>% &quot;&quot; &quot;&quot; TEXT OUTPUT</td>
</tr>
<tr>
<td>PIPosition</td>
<td>% &quot;&quot; &quot;&quot; PEN POSITION</td>
</tr>
<tr>
<td>RIO.WRA</td>
<td>% INPUT OUTPUT ROUTINE</td>
</tr>
<tr>
<td>SCALE.FACT</td>
<td>% CALCOMP FLOTTED, SCALING ROUTINE</td>
</tr>
<tr>
<td>TIME.TIM1</td>
<td>% TIME OUTPUT</td>
</tr>
<tr>
<td>TPAP.PARCH</td>
<td>% PARITY CHECK</td>
</tr>
<tr>
<td>TXT.TXT1</td>
<td>% TEXT OUTPUT</td>
</tr>
<tr>
<td>TRAC.ELOC</td>
<td>% CALCOMP FLOTTED, LINE OUTPUT</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>% CONSTANTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>DS. #</td>
</tr>
<tr>
<td>DAD.C102.3</td>
</tr>
<tr>
<td>HFP.C204.7</td>
</tr>
<tr>
<td>BLEXE.CO.524.9.4.483</td>
</tr>
<tr>
<td>DTR.CO.6175427825</td>
</tr>
<tr>
<td>PI.C3.14159265</td>
</tr>
<tr>
<td>TP1.C6.285288</td>
</tr>
<tr>
<td>P2.C1.5707963266</td>
</tr>
<tr>
<td>EX1.CO.696592131</td>
</tr>
<tr>
<td>EX2.CO.60672267</td>
</tr>
<tr>
<td>ER.x607595958</td>
</tr>
<tr>
<td>ELR.x6376228</td>
</tr>
<tr>
<td>TR.x0.002</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>% FLOATING POINT NUMBERS</th>
</tr>
</thead>
<tbody>
<tr>
<td>F10.C10</td>
</tr>
<tr>
<td>F11.C1</td>
</tr>
<tr>
<td>F2.C2</td>
</tr>
<tr>
<td>F3.C3</td>
</tr>
<tr>
<td>F5.C5</td>
</tr>
<tr>
<td>F7.C70</td>
</tr>
<tr>
<td>F9.C240</td>
</tr>
<tr>
<td>F10.C360</td>
</tr>
<tr>
<td>F11.C420</td>
</tr>
<tr>
<td>F12.C500</td>
</tr>
<tr>
<td>F13.C1000</td>
</tr>
<tr>
<td>MIN05.C-0.5</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>% INTEGERS</th>
</tr>
</thead>
<tbody>
<tr>
<td>HI5.3</td>
</tr>
<tr>
<td>HT2.0244</td>
</tr>
<tr>
<td>HT5.012</td>
</tr>
<tr>
<td>HT6.0144</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>% INTERRUPT CONTROLS, INPUT-OUTPUT DEVICES</th>
</tr>
</thead>
<tbody>
<tr>
<td>DU3.0</td>
</tr>
<tr>
<td>DU7.0</td>
</tr>
<tr>
<td>DTV.0</td>
</tr>
<tr>
<td>DTV.0</td>
</tr>
<tr>
<td>DV17.0</td>
</tr>
<tr>
<td>DV22.0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>K.O</th>
<th>% BASE ADDRESS, USUALLY HOLD IN D-REG.</th>
</tr>
</thead>
<tbody>
<tr>
<td>SMD.CO</td>
<td>% CORR. FACTOR, DEPTH S-T-O</td>
</tr>
<tr>
<td>SKT.CO</td>
<td>% CORR. FACTOR, TEMPERATURE S-T-O</td>
</tr>
<tr>
<td>SN1.CO</td>
<td>% CORR. FACTOR, SALINITY S-T-O</td>
</tr>
<tr>
<td>C131.0</td>
<td>% TEMPORARY STORAGE, LEVEL 13</td>
</tr>
<tr>
<td>0</td>
<td>% &quot;&quot; &quot;&quot; &quot;&quot; &quot;&quot;</td>
</tr>
<tr>
<td>0</td>
<td>% &quot;&quot; &quot;&quot; &quot;&quot; &quot;&quot;</td>
</tr>
<tr>
<td>0</td>
<td>% &quot;&quot; &quot;&quot; &quot;&quot; &quot;&quot;</td>
</tr>
<tr>
<td>10</td>
<td>% &quot;&quot; &quot;&quot; &quot;&quot; &quot;&quot;</td>
</tr>
<tr>
<td>10</td>
<td>% &quot;&quot; &quot;&quot; &quot;&quot; &quot;&quot;</td>
</tr>
</tbody>
</table>
% FLOATING OUTPUT. THIS ROUTINE PERFORMS OUTPUT OF FLOATING ACCUMULATOR.
% THE OUTPUT IS RIGHT HAND ORIENTATED. DEVICE NUMBER 5.5.7.11 OR 17 CAN
% BE USED. THE ROUTINE IS RE-ENTRANT. X REGISTER WILL BE SAVED AND THE
% BASE-ADDRESS (K) WILL BE LOADED INTO B-REGISTER BEFORE RETURN.

DORTEL.COPY 5A DB
THA .STB
SHR 4
MPY X RE
STO I SAVEX .X
COPY SB DX
COPY ADD SL DD
COPY ADI SB DL
BLDA 0170 DT
BSET IRO 0170 DT
SAA 3
MPY .1 .B
ADD SIFF -1
SWAP SA DX
RIUC DT
PAD OES .X
RDR DT
COPY SA DX
SAA -2
MPY .B
ADD SIFF -2
BSEP ONE SSK
AAA -2
SWAP OLD SA DD
SPACE.AAS 3
LDA SIFF -2
SKE IF DA UEG 5B
JMP = 012
COPY ST DA
SUB .E
JAF + A
COPY SX DA
SUB 1 .B
JAF + 3
COPY SD DA
SUB 2 .B
JAN BACK -011
ROLR DA
ESTA 00 DA

% CONSTANTS COMPUTED AND USED BY DECCA PROGRAM

ORF .CO
FA .CO
FB .CO
FC .CO
FD .CO
STG .CO
OA .CO
OB .CO
OC .CO
CH1 .0
CH2 .0
0
CH3 .0
0
T1 .0
T2 .0
T4 .0
TS .0
S2 .0
S3 .0
S4 .0

% FLAG-WORD, INDICATING WHETHER DECCA IS RUNNING

% FLAG-WORD, ONE BIT FOR EACH ROUTINE
JAF       BACK -2  % NEO. SIGN TO BE PRINTED OUT
BSHP      ZRO 0160 DT % FLOATING ACCUMULATOR LESS THEN ONE?
JMP       FORB 1  % NO. COMPUTE DIGIT
JMP       BACK
SWAP      SL  DB
LDA       1 -2  B  % DEVICE NUMBER
SWAP      SL  DB
BSET      ZRO 050 DA
AAA        -7  %
JAZ SPACE  % SKIP OUTPUT OF SPACE ON FAST PUNCH
SAA       0164  % ENTRY FOR MAKING ASCII-CODE FOR SPACE
AAA        -2  % ENTRY FOR MAKING ASCII-CODE FOR NEO. SIGN
AAA        -2  % ENTRY FOR MAKING ASCII-CODE FOR DEC. POINT
BACK AAA  040  % ENTRY FOR MAKING ASCII-CODE FOR ZERO
SWAP SA DB
TJR MPR   % USING MPR-REG. FOR TEMPORARY STORAGE OF D-REG
LDA       1 -2  B  % DEVICE-NUMBER
AAA       -017
JAZ        #2
AAA       016  %
SWAP SA DL
RADD SL DP
IOT ACT PIN 017  % JUMP TO CORRECT OUTPUT-INSTRUCTION
JMP       #010  % OUTPUT ON FAST PUNCH II
IOT ACT PIN 03  % OUTPUT ON TELETYPEx I
JMP       #6  %
IOT ACT PIN 05  % OUTPUT ON TELETYPEx II
JMP       #4  %
IOT ACT PIN 07  % OUTPUT ON FAST PUNCH I
JMP       #2  %
IOT ACT PIN 011  % OUTPUT ON TELETYPEx III
WAIT      % GIVE UP PRIORITY. WAIT FOR INTERRUPT FROM DEVICE
COPY SA DL
SWAP SL DB
TRX MPR
SWAP SA DB
AAA       -055  %
JAZ SPACE  020  % JUMP IF OUTPUT WAS NEO. SIGN
JAZ FORB 1  % JUMP IF OUTPUT WAS DEC. POINT
AAA       -0162
JAZ SPACE  % JUMP IF OUTPUT WAS SPACE
SAA        3
SWAP SX DL
MPY       .X  % NUMBER OF DIGITS AFTER DEC. POINT
SWAP SX DL
ADD       $IFF -2  % OUTPUT FINISHED?
SKP       IF DA UEQ $B  % YES
LDA       $IFF -2
RSUB       $B DA
JAZ        BACK -1  % TIME FOR DEC. POINT?
FOREI, AND 2  % START COMPUTING NEXT DIGIT
SWAP OLD SX DA
PB .B
BSHP ONE 0170 DT
JPC       # -2  % COUNT THE SUBTRACTION LOOP
FAD .B
SWAP SA DX
BLDA 00 DA
BSHP BAC 010 DA
BSET ECM 070 DA
BLDA 020 DA
BSHP EBC 030 DA
BSET ECM 070 DA
JMP       BACK  % CORRECT SETTING OF PARITY BIT
TRE .3
SAVEX, DLX 1  % ADDRESS TO STORAGE BLOCK. 3 LOC. FOR EACH INT. LEV.
UT, TRA STS
SMA .4  % CURRENT INTERRUPT-LEVEL
SMR ZIN SRC 14
MPY TRE
COPY SA DX
LDA 1 SAVEX .X
COPY SA DX
LDA BASE1
SWAP OLD SX DA
COPY AD1 SL DP
% BASE-ADDRESS TO D-REG.
EN
$IFF
$IFF, C1000000000.
C1000000000.
C1000000000.
C1000000000.
C1000000000.
C1000000000.
C1000000000.
EN .X.
% FLOATING INPUT. THIS ROUTINE PERFORMS INPUT OF DECIMAL NUMBERS INTO % FLOATING ACCUMULATOR FROM DEVICE NUMBER 2, 4, 10 OR 22. THE ROUTINE IS % RE-ENTRANT. X-REGISTER WILL BE SAVED AND THE BASE-ADDRESS (K) WILL BE % LOADED INTO B-REGISTER BEFORE RETURN.

FLINT, TRA STS
SHA 4
SHA ZIN SHR 014 % CURRENT INTERRUPT-LEVEL
MPY RES3, K, B
SNAP SA DX % INDEX FOR TEMPORARY STORAGE
TRR MPR
LDF FL0-K, B
STF I RES1, X
STF I RES2, X
COPY ADI SL DB % RETURN ADDRESS
RCRLA DL
JMP RUT
CIFER BSET ONE 0160 DB
NLZ 020 % DIGIT JUST READ
FAD I RES2, X % ADD EARLIER READINGS
FNU EN-5 % GIVE PLACE FOR NEXT DIGIT
STF I RES2, X
BSLP ZRO 0170 DX % AFTER DECIMAL POINT ?
RDCL DL % YES, COUNT DECIMALS
RUT, COPY SL DT
JPL I WRI
COPY ST DL
AAA -072
JAP DELMT
AAA 012
JAP CIFER

AAA 2 % DECIMAL POINT ?
JNF # 3 % YES, X-REG. WILL BECOME NEGATIV
ANX -000 % YES, X-REG. WILL BECOME NEGATIV
JMP RUT
AAA 1 % NEGATIV SIGN ?
JNF # 2 % YES, SET TEST-BIT
BSET ONE 0170 DB % DECIMAL NUMBER HAS BEEN READ ?
JMP RUT % NO, IGNORE DELIMITER
BSLP ZRO 0170 DX
AAA 060 % RESET INDEX
LDF EN
FNU EN 3 % MULTIPLY WITH 0.1 FOR EACH DIGIT AFTER DEC. POINT
RINC DL
SNP IF DL GRE 0
JMP # -3
FNU I RES1, X
FAD I RES2, X % SECOND PART OF DECIMAL NUMBER. AFTER DEC. POINT
FNU EN 3 % FIRST PART OF DECIMAL NUMBER. BEFORE DEC. POINT
BLDA 0170 DB
BSTA 0170 DT
COPY SA DL
TRA MPR
COPY SA DX % UNSAVE X-REG.
SNAP SL DB
LDA BASE
SNAP SA DB % BASE-ADDRESS TO B-REG.
BSET ZRO 0170 DL % RESET TEST-BIT
BSET ZRO 0160 DL % RESET TEST-BIT
EXIT

% INTEGER INPUT. THIS ROUTINE HANDLES INPUT OF INTEGER NUMBERS INTO % A-REGISTER FROM DEVICE-NUMBER 2, 4, 10 OR 22. THE ROUTINE IS RE-ENTRANT. % X-REG. WILL BE SAVED AND THE BASE-ADDRESS WILL BE LOADED INTO B REG. % BEFORE RETURN.

PRI, COPY ADI SL DB % RETURN ADDRESS
RCRLA DT
BSET ONE 0170 DT % NUMBER HAS NOT BEEN READ
JPL I WRI % READ ONE CHARACTER
AAA -072
JAP DELMT
AAA 012
JAP CIFER % JUMP IF LAST CHARACTER WAS A DIGIT
\texttt{\textbf{JAN} NEGP} \% JUMP IF DELIMITER OR NEG. SIGN
\texttt{BSET ZRO 0170 DT} \% NUMBER HAS BEEN READ
\texttt{SNAP SA DT} \% ADD TO LAST DIGIT
\texttt{MPY SIFF 3} \% MULTIPLY EARLIER READINGS WITH TEN
\texttt{RADD SA DT} \% ADD TO LAST DIGIT
\texttt{JMP PRI 3}
\texttt{NEGP AAA 3} \% NEG. SIGN?
\texttt{JAP UTP} \% YES. SET TEST-BIT
\texttt{BSET ONE 0170 DB} \% No. 10 WERE DELIMITER
\texttt{UTP BSHP ZRO 0170 DT} \% NUMBER HAS BEEN READ?
\texttt{JMP PRI 3} \% NO. IGNORE DELIMITER
\texttt{COPY ST DL} \% NEG. SIGN?
\texttt{BSHP CM SL DL} \% YES. MAKE TWO'S COMPLEMENT
\texttt{JMP PRI-6} \% BASE-ADDRESS TO B-REG. RESET TEST-BIT AND EXIT

\texttt{\% INTEGER OUTPUT. THIS ROUTINE GIVES A RIGHT HAND ORIENTATED DECIMAL OUTPUT OF AN INTEGER NUMBER IN A-REGISTER TO DEVICE NUMBER 3, 5, 7, 11 OR 17. THE ROUTINE IS RE-ENTRANT. X-REG. WILL BE SAVED AND THE BASE-ADDRESS WILL BE LOADED INTO B-REG. BEFORE RETURN.}

\texttt{HELTA DLSA 0170 DA} \% SIGN-BIT TO K (ONE BIT ACCUMULATOR)
\texttt{BSHP ZRO 0170 DA} \% ABS. VALUE OF A-REG.
\texttt{COPY CM2 SA DA} \% POINTER TO NUMBER OF DIGITS IN OUTPUT
\texttt{COPY SX DT} \% NUMBER OF DIGITS IN OUTPUT
\texttt{COPY CH2 SX DX} \% INDEX FOR ADDRESSING THE "SIFF"-TABLE
\texttt{TRR MPR} \% MPR-REG. USED FOR TEMPORARY STORAGE OF A-REG
\texttt{LUP1 TRA MPR}
\texttt{SUB I SIFF-2 X} \% SIGN=1 WHEN TIME FOR PRINTING SIGN
\texttt{BLDC 0170 DA} \% SIGN=1 WHEN TIME FOR PRINTING NEG. SIGN
\texttt{BSET DMC 0160 DB} \% DEVICE NUMBER
\texttt{BSET ZRO 050 DA} \% Device number
\texttt{AAA -7} \% TIME FOR PRINTING NEG. SIGN?
\texttt{BSHP ONE SSK} \% NO. SKIP OUTPUT OF SPACE ON FAST PUNCH
\texttt{JAI 5} \% ASCII-CODE FOR SPACE
\texttt{SKUP ZRO $8H} \% CHANGE TO ASCII CODE FOR NEG. SIGN
\texttt{AAA 015} \% PRINT OUT SPACE OR NEG. SIGN
\texttt{JPL I WR1} \% TIME FOR PRINTING DIGIT?
\texttt{BSHP ONE 0160 DB} \% JNC LUP1
\texttt{LUP2 COPY CH1 DD} \% LAST DIGIT TO BE PRINTED OUT?
\texttt{TRA MPR} \% NO
\texttt{JPL I SIFF-1 X} \% COUNT THE SUBTRACTION LOOP
\texttt{RINC DB} \% MPR-REG. USED AS TEMPORARY STORAGE OF A-REG.
\texttt{JAP I SIFF-1 X} \% MAKING ASCII-CODE
\texttt{COPY DD DA} \% PRINT OUT ONE DIGIT
\texttt{AAA 040} \% JUMP IF NOT LAST DIGIT
\texttt{JPL I WR1}
\texttt{JXN LUP2}
\texttt{COPY ST DX}
\texttt{COPY ADI 65 DL} \% RETURN ADDRESS
\texttt{JMP PRI-5} \% EXIT
\texttt{RESI 1 BLOK1} \% ADDRESS TO STORAGE BLOCK. 3 LOC. FOR EACH INT. LEV.
\texttt{RES2 1 BLOK2}
\texttt{SIFF 6}
\texttt{SIFF 020200}
\texttt{001769}
\texttt{000012}
\texttt{000000}
\texttt{SKILL EH FLINT RUT CIFER DELMT PRI NEGP UTP HELTA LUP1 LUP2 SIFF}
\texttt{SKILL WRI RESI RES2 BASE1}

\texttt{\% DATE OUTPUT. THIS ROUTINE GIVES OUTPUT OF DATE ON DEVICE NUMBER 3, 5, 7, 11 OR 17. THE ROUTINE IS RE-ENTRANT. X-REG. WILL BE SAVED AND THE BASE-ADDRESS WILL BE LOADED INTO B-REG. BEFORE RETURN.}

\texttt{DATE1 COPY 3X DA} \% MPR-REG. USED FOR SAVING X-REG.
\texttt{TRA MPR}
\texttt{LDA I DAT01} \% DATE
\texttt{SWAP OLD SA DD} \% A-REG. BIT 0-5 YEAR
\texttt{SHD ROT 4}
ESET IRO SSM
SUM LIA 012
COPY SD DT % T-REG. BIT 012-017 MONTH, BIT 4-O11 DAY
AAA 0100 % OUTPUT LOOP TO BE EXECUTED THREE TIMES
JMP BACKT-1

% TIME OUTPUT. THIS ROUTINE GIVES OUTPUT OF TIME ON DEVICE NUMBER 3,5,7,
% 11 OR 17. THE ROUTINE IS RE-ENTRANT. X-REG. WILL BE SAVED AND THE
% BASE-ADDRESS WILL BE LOADED INTO B-REG. BEFORE RETURN

TIME1, COPY SD DA
TRR MPR % MPR-REG. USED FOR SAVING X-REG.
LDD 1 DATO1 % DATE TO A-REG. TIME TO D-REG.
ELDA 00 DA
RLCRL DA
SAD 4 % A-REG. BIT 0-5 HOURS
COPY SD DT % T-REG. BIT 012-017 MINUTES, BIT 4-011 SECONDS
BKPZ IRO SSK % P.M. OR A.M. ?
AA 014 % A.M. ADD TWELVE TO HOURS
SAX -2 % OUTPUT OF HOURS AND MINUTES ONLY
COPY ADI SD DB % RETURN ADDRESS
BACKT, JPL ASCII % MAKING ASCII-CODE OF A-REG. ONLY TWO DIGITS
JPL WRA 7 % PRINT OUT A-REG. TWO DIGITS.
SAA 040 % PRINT OUT SPACE
JPL WRA 7 % PRINT OUT CHARACTER
SHT ROT 4
SAA 077 % NEXT INFORMATION IN A-REG.
JND BACKT % JMP IF OUTPUT OF DATE OR TIME IS NOT FINISHED
TRA MPR
COPY SAB DX % UNSAVE X-REG.
JMP WRA 2 % EXIT

% BASE-ADDRESS WILL BE LOADED INTO B-REG. BEFORE RETURN

TXT1, COPY ADI SD DB % RETURN ADDRESS
SWAP ST DB % TWO CHARACTERS INTO A-REG.
LDA 8
SWAP ST DB
COPY SA DD
SAA 0177 % ONE CHARACTER TO A-REG.
JAZ WRA 2 % A-REG. EMPTY. EXIT
AAA =47
JAZ = 5 % TEST FOR END OF TEXT.
COPY SD DA
JPL WRA 7 % PRINT OUT CHARACTEHS IN A-REG.
RINC DT % ADDRESS TO NEXT TWO CHARACTERS
JMP =-014
COPY SD DA
SAA ZIN SHR 010 % LAST CHARACTER
JAZ WRA 2 % A-REG. EMPTY. EXIT
JMP WRA 1 % PRINT LAST CHARACTER

% NUMBER OUTPUT. ROUTINE FOR MAKING ASCII-CODE AND PRINT OUT A NUMBER
% IN A-REG. (LESS THAN 100) ON DEVICE NUMBER 3.5.7.11 OR 17.
% THERE WILL ALWAYS BE A Digit IN OUTPUT (E.G., 1 OUTPUT: 01 ). THE
% ROUTINE IS RE-ENTRANT. ONLY A- AND D-REG. WILL BE LOST.

ASCII, COPY ADI SD DB % RETURN ADDRESS
JPL ASCII % MAKING ASCII-CODE OF THE NUMBER IN A-REG.
JPL WRA 1 % PRINT OUT A-REG.

% ASCII-CODE. THIS ROUTINE CHANGE THE A-REG. FROM A NUMBER TO
% ASCII-CODE FOR THE SAME NUMBER. THE NUMBER MUST BE LESS THEN 100 AND
% POSITIV. THE ROUTINE IS RE-ENTRANT. ONLY A- AND D-REG. WILL BE CHANGED
% MAKING ASCII-CODE OF THE ONES
MAA 010
AAA 060
RADD SD DA
SAA ROT 010
EXIT

% CR/LF OUTPUT. THIS ROUTINE PRINT OUT CARRIAGE RETURN, LINE FEED ON
% DEVICE NUMBER 3,5,7,11 OR 17. ONLY A- AND D-REG. WILL BE LOST.

CLF1, LDA CL
% LOAD CR/LF INTO A-REG. CONTINUE WITH NEXT ROUTINE

% SINGLE INPUT-OUTPUT. THIS ROUTINE READ ONE CHARACTER INTO A-REG. OR
% PRINT OUT A-REG. (ONE OR TWO CHARACTERS). A- AND D-REG. WILL BE LOST

WPA,COPY ADI SL DB % RETURN ADDRESS
JPL WRA 7 % INPUT-OUTPUT OF A-REG.
COPY SB DL
COPY SA DB
LDA BASE
SWAP SA DB
% BASE-ADDRESS TO B-REG.
EXIT

% INPUT-OUTPUT. THIS IS THE BASIC INPUT-OUTPUT ROUTINE AND ALL INPUT-
% OUTPUT, EXCEPT FLOATING OUTPUT, GOES THROUGH THIS ROUTINE. DEPENDING
% ON DEVICE NUMBER THIS ROUTINE HANDLES INPUT OF ONE CHARACTER TO A-
% REG. FROM DEVICE NUMBER 2,4,10 OR 22, OR GIVES OUTPUT FROM A-REG.
% ONE OR TWO CHARACTERS, TO DEVICE NUMBER 3,5,7,11 OR 17. THE ROUTINE
% IS RE-ENTRANT AND ONLY A- AND D-REG. AND THE ONE BIT ACCUMULATOR
% WILL BE LOST.

INOOUT,COPY SA DD
JAZ = 015
SHD ROT 010
SAA 177
BSET ONE 070 DA

% JUMP IF NO CHARACTER IN THIS POSITION
JAZ --0
BSET ONE 070 DA
BSET 170 SSK
MPH 1
BSET 160 070 DA
BSET BCH 88K
JAP =-2
BSET BAC 070 DD % SET CORRECT PARITY-BIT
ENTR, SNA 2
MPV -1 -1.2
AAR =-4
SLDA 060 DA
BSP NOT 050 DA
WAIT
SWAP BA DA
RADD SD DP
JOT ACT PIN 02
JMP PARCH
JOT ACT PIN 03
JMP TEST
JOT ACT PIN 04
JMP PARCH
JOT ACT PIN 05
JMP TEST
CL 04012
% ASCII-CODE FOR CARRIAGE RETURN, LINE FEED
BASE K
JOT ACT PIN 07
JMP TEST
JOT ACT PIN 010
JMP PARCH
JOT ACT PIN 011
JMP TEST
DAT01, DATO % DATE

% DOUBLE INPUT. THIS ROUTINE HANDLES INPUT OF TWO CHARACTERS IN A-REG.
% FROM DEVICE NUMBER 2,4,10 OR 22. THE ROUTINE IS RE-ENTRANT, X- AND
% T-REG. WILL BE SAVED AND THE BASE-ADDRESS WILL BE LOADED INTO B-REG.
% BEFORE RETURN

DINP,COPY ADI SL DB % RETURN ADDRESS
JPL WRA 7 % READ FIRST CHARACTER
% PARITY CHECK: THIS ROUTINE TESTS FOR CORRECT PARITY OF EACH CHARACTER IT RECEIVES. IF IT
% DETECTS AN ERROR, IT SENDS A MESSAGE TO THE TELETYPE I. THE ROUTINE IS RESTARTED.
% ALWAYS FIELD HEADED THE ROUTINE IF NOT CORRECT PARITY AT END.
% WAIT FOR INTERRUPT
% INPUT-OUTPUT CONTINUE
% OUTPUT TO FAST-PUNCH 11
% WAIT FOR DATA
% PRINT OUT MESSAGE 'IC' TO TELETYPE I.
% WAIT FOR INTERRUPT
% SET BIT IN TESTWORD FOR TELETYPE I
% PRINT OUT MESSAGE 'IC' TO TELETYPE I
% SET BIT IN TESTWORD FOR TELETYPE I
% PRINT OUT MESSAGE 'IC' TO TELETYPE I
% PRINT OUT MESSAGE 'IC' TO TELETYPE I
% PRINT OUT MESSAGE 'IC' TO TELETYPE I

% ROUTINE FOR POSITION OUTPUT (DEGREES AND MINUTES )

124375  % JMP 8-3
KONV1, LDF, X
SLDA 0170 DT
SET 2RO 0170 DT
D42 -020
SWAP SL DB
LDT 2.B
BSKP 2RO 0170 DT
JMP # 4
LDT KONV1-1
STT 2.B
SWAP SL DB
LDT HT0-K, B
COPY AD1 ST DP
BAT 1
STT 2.B
SWAP SL DB
NLZ 020
BSTA 0170 DT
PSD, X
FMU FLA-K, B
FMU FL0-K, B
SET 2RO 0170 DT
JMP I DORIE-K, B

% ROUTINE FOR DATA REDUCTION, AVERAGING

AVJF, LDA 8, X
SHA 010
SHA ZIN SHR 010
NLZ 020
STF 1.X
LDF 010.X
FDV 1.X
STF 1.X
LDF FLO-K, B
STF 010.X
EXIT

% EXPONENTIAL FUNCTION

EXPF, SWAP ST DX
SLDA 0170 DT
SET 2RO 0170 DT
TRR MP1
STA STS
SHA 4
SHA ZIN SHR 014
MPY NBS-K, B
ADD A1 020
COPY SA DB
COPY SX DA
FDV A1
STF . B
LDX A1 021
RSUB ST DX
RCLR DD
JPC # 0
SAD ROT 1
JNC #-1
COPY SD DA

% ROUTINE FOR READING MULTIPLEXER

MPX1, LDA 4, X
SHA ZIN SHR 010
COPY SA DB
IOT ACT SHA 050
JMP 8-1
COPY SD DA
IOT ACT SHA 050
JMP 8-1
SHA 4
SHA SHR 4
NLZ 020
FDV MP4-K, B
FMU 013.X
FAD 01A.X
STF 5.X
FAD 010.X
STF 010.X
EXIT
% NATURAL LOGARITHM FUNCTION
LN2: 040000
130562
013767

AK: 040000
132404
171465

AK2: 137777
130562
013767

AL: 040001
177777

B: 037775
107344
140333

C: 140000
142876
006345

D: 140001
121160
125007

NET: 143000
177777
177777

)KILL AK1 AK2 AA B C D NEG HK1 ARG1

% SINE FUNCTION (RADIANS)
SIM1: RCLR DB
SLDA 0170 DT
JMP + 3

% COSINE FUNCTION (RADIANS)
COS1: RCLR ADI DB
BSET ZRO SSK
BSET ZRO 0170 DT
FBB FIT
RING DB

SKP IF DT LST 0
JMP = -3
BUP ZRO 010 DB
BSET BCM SSK
BSET ZRO 00 DB
FAD FIT
COPY SA DB
COPY SX DB
TAR MPR
TRA STS
SHA 4
SHA ZIN SHR 014
MPY TALLS
ADD ARG
SWAP SA DB
STF 060 , B
FMU 060 : B
STF , D
SAX - 0DE
LDF 1 KONST , X
FMU , B
ANX 3
FAD 1 KONST , X
JXN * -3
FMU 060 , B
BSEP ZRO SSK
BSET BCM 0170 DT
COPY SA DX
LDN KK
COPY SA DB
TRN MPR
SWAP SA DX
EXIT

TALLS: 0
KONST, KADR
FIT, 40004, 114417, 125242
137733, 153477, 117470
037740, 130222, 030234
137747, 153462, 025477
037756, 153480, 016460
137744, 150015, 000031
03772, 104210, 104211
137776, 125222, 125228
KADR, 040001 100000, 000000
ARG. BLOK1

% TANDEMT FUNCTION (RADIANS)
TAN1, COPY SL DX
BLDC 0170 DT
BSET ZRO 0170 DT
FSB PIT
BSET ECM SSK
ESKP ONE 0170 DT
JMP * -5
FAD PIT
BSTA 0170 DX
JPL SIN1
COPY SX DL
BLDA 0170 DL
BSET ZRO 0170 DL
COPY SA DX
TRA STS
SHA 4
SHA ZIN SHR 014
MPY NBS-K ,B
ADD ARG
SWAP SA DX
STF ,X
FMU ,X
STF ,X
BSET ONE 0170 DT
JNZ UTAN
STF OAO ,X
LDF ,X
FAD 060 ,X
COPY SL DX
JPL ROT1
COPY SX DL
JMP * 2
UTAN,LDF BIO
BSTA 0170 DT
EXIT
BIO.077777
177777
177777

% ARCUS TANDEMT (RADIANS)
ATAN1, SWAP SA DX

TANR MPR
TRA STS
SHA 4
SHA ZIN SHR 014
MPY NBS-K ,B
ADD ARG
SWAP SA DX
BLDA 0170 DT
BSET ZRO 0170 DT
FSB FL1-K ,B
SKP IF DT LT <
JMP * 4
FAD FL1-K ,B
JMP * 4
FAD FL2-K ,B
STF ,X
FIB FL2-K ,B
FAD ,X
BSET ONE 0170 DL
STF O60 ,X
FMU O60 ,X
STF ,X
COPY SX DB
SAX -017
LDP 1 KOEFF ,X
FMU ,B
AAX 3
FAD 1 KOEFF ,X
JXN * -3
FMU O60 ,B
ESKP JNZ 0170 DL
FAD P14
BSET ZRO 0170 DL
RISUP ZRO SSK
BSET ECM 0170 DL
COPY SA DX
LDA KN
COPY SA DB
TRA MPR
SWAP SA DX
EXIT
KOEFF, KOADDR
KL.K
P14.<0.70535814
<0.01172120
< 0.055265332
LDX XI
SNAP CLE DX
ESET ONE 0150 DX
RESB SX DT
COPY SX DX
TRA MPR
SNAP SX DX
UTRO.EXIT

KOFF: C0 3448
1KILL: X1 KOFF KOFF KADR P14
1FCL XI

% INITIALIZATION OF INTERRUPT SYSTEM

INIT: IDP
SAA -1
MCL PID
MCL PIE
ION GR1 GRF2
WAIT
LDA (LEV15
STA I 10270
LDA (LEV14
STA I 10257
LDA (LEV13
STA I 10246
LDA (LEV12
STA I 10235
LDA (LEV11
STA I 10224
LDA (LEV10
STA I 10213
LDA (LEV9
STA I 10202
LDA (LEV8
STA I 10171
LDA (LEV7
STA I 10160
LDA (LEV6
STA I 10147
LDA (LEV5
STA I 10136
LDA (LEV4
STA I (0125)
LDA (LEV3)
STA I (0114)
LDA (LEV2)
STA I (0103)
LDA (LEV)
STA I (0172)
LDA (LEV0)
STA I (0061)
LDA (K)
STA I (0277)
STA I (0244)
STA I (0255)
STA I (0244)
STA I (0233)
STA I (0222)
STA I (0211)
STA I (0200)
STA I (0167)
STA I (0145)
STA I (0134)
STA I (0123)
STA I (0113)
STA I (0112)
STA I (0070)
IOT PIN 2
IOT PIN 4
IOT PIN 010
IOT PIN 0129
IOT PIN 0122
IOT PIN 0121
LDA (057277)
MST PIE
JMP I *
)FILL
)KILL LIMIT
% PROGRAM ON LEVEL 15
% IDENTIFICATION OF INTERRUPTS FROM ACOUSTIC INTERFACE UNIT

LEV15. IOT SNI 0121
JMP + 4
% INTERRUPT INDICATING UPPER LIMIT

IOT SNI 0122
JMP + 012
% INTERRUPT INDICATING LOWER LIMIT
JMP + 015
IOT PIN 0121
% PREPARE NEW INTERRUPT
IOT ACT 0130
% READ DEPTH COUNTER
SHA ZIN SHR 1
STA UPL
RCRL DA
BSET ONE 0160 DA
MST PIN
% INTERRUPT TO LEVEL 14
JMP + 5
IOT PIN 0122
% READ INTERVAL COUNTER
IOT ACT 0130
SHA ZIN SHR 1
STA UPL 2
WAIT
JMP LEV15
% EXIT

% PROGRAM ON LEVEL 14
% PROGRAM FOR ECHO STRENGTH SPECTER
TAB: WAIT
IOT ACT 052
% DUMMY READING
LDX UPL 2
COPY CH2 DX DX
LDT RASH
SNAP CLD ST DD
RCRL DB
JXI LEV14
JXI LEV14
JMP + 1
IOT ACT 52
% READING LOOP
SKP IF DA LST 8D
JMP + 020
SKP IF DT EOL 0
SKP IF DL LST 0
JMP + 7
STI .B
STX 1 .B
AAD 2
RCRL DL
JNC + 012
JMP + 022
STA -0144
AAAA -020
JMP LEV14  X INTERVAL CHANGED
LDA UPL
LDT UFL 1
STA UFL 1
RSUB ST DA
JAP # 2
COPY CH2 DA DA
AAA -020
JMP LEV14  X UPPER LIMIT CHANGED
WAIT
IOT ACT 052  X DUMMY READING
JXZ LEV14
JXZ LEV14
LDT RASK
RCLR DB
BSET ONE 070 DB
MIN DELAY
JMP = -7
BLDC 0120 DX
INLES. IOT ACT 052  X READING LOOP
SKP IF DA LST ST
JMP EKK0
BSKP EAC 0120 DX
JMP = 4
BSET EOM 00
SHD 1
JMP = 4
BSKP ZRO 00 DD
JMP = 3
JMP = 1
JMP = 3
BSKP ONE 0170 DD
JMP FISK:
RCLR DB
BSET ONE 0150 DB
JMP EKK0 010
BSKP ZRO 00 DD
BSKP ONE 00 DD
JMP EKK0 4
BSET ONE 0170 DD
JXZ LEV14
JMP EKK0 010
EKK0. BSKP ZRO 070 DB
JMP EKK0 -6
STA .B
RINC DE
RSUB ST DA
SKP IF DA ORE ST
SKP IF DA LST ST
BSET ONE 00 DD
JAP INLES
SKP IF DA U EQ 0
JMP LIMIT
BSKP ONE 00 DD
JMP ENDOP
JMP LEV14
LIMIT. RCLR DT
BSET ONE 0120 DT
SAX -020
LDA TELL1
RADD ST DA
RING DX
JAN = -2
SHD 1
JNC = -1
JMP START=3

FISK: LDA EOMS
ADD DLAY1
SKP IF DL U EQ 0
LDA TELL1
RSUB SX DA
COPY CH2 SA DX
AAA 0177
JAP LEV14
STA DELAY
STA DLAY1
SKP IF DL EQL 0
JMP = 013
ST2 CCC
ST1 CCC 1
ST2 CCC 2
COPY SX DA
ADD UFL
NLZ 020
FMU MPS
STF FDYP
DNS = -020
STA DYP
STT -051. B
LDA DVP
STA -055. S
LDI C00
FDV ($10. 004
DNZ -020
STA -055. B
COPY A00 BL DA
NLZ 020
FDV I P40 1
R002 DT
FDV PULSR
FDV FDVP
LDX PING 2
SNAP SX DB
JPL 1 ATAM-K. S
FDV DTR-K. B
COPY SX DB
DNZ -020
STA -054. B
STOKE. COPY SA DX
JXT = 014
LDA -052. B
SUB DUMX
JAP = 015
COPY CM1 SA DA
BSET ZRO 00 DA
SAT 094
SNIP IF DA ORE ST
SNP IF DX LST ST
JMP = 5
SHA 4
RADD SA DB
BAR -0177
MIN 055. X. B
SAA 040
MST PID
LDA PING 5
STA PING 7
JMP 1 PING 5

)FILL.

% PRELIMINARY PROGRAM FOR BOTTOM BACK SCAT TERING

TOM. LDA PING 2

% SAMPLING AND FILTERING

COPY SA DB
LDX 1 (UPL 2
COPY CM2 SX DX
R0L DT
JOT 00T 032
SNIP IF DA LST ST
SNP IF DX ORE ST
JMP = 4
COPY SA DT
COPY SX DD
JMP = 4
JMP = 1
JMP = 1
JMP = 1
JMP = 1
JMP = 1
JMP = 014

COPY ID DA
COPY ST DX
ADD I (UPL
ADD I (UPL 2
NLZ 020
FMU HPS
FSR (175
FDV (175. 167
DBF 020. 0170 DT
LDF FLO-K. B
STP S
COPY SX DA
NLZ 020
FDV SM-K. B
JXZ = 2
JPL 1 ATAM-K. S
FMU BLOE-K. S
FMU FL20-K. B
FAD S
STP S
LDA 1 IF
NLZ 020
FAD S
FAD S33
STP S33
MIN SC
WAIT
JMP I (LEV14. 
S. CO  % TEMPORARY STORAGE
ES. CO  % TEMPORARY STORAGE
BC. 0  % COUNTER

IFILL

SKILL TOM $ AFING SPING AVEL
SKILL BRT PLCTS FLOR WHERE FACT NRR2 SYMBOL SPECT NUMER KQHT

SKILL LEVI5 UFL LEVI4 START INLE3 EUKO LIMIT TELL1 DELAY PLAY1
SKILL FISH DEAMAX MAV DYP FDYP CCC VOLT HPS ENDOF STORE TARI TB14

ILINE

% PROGRAM ON LEVEL 13
% THIS LEVEL IS ACTIVATED EVERY SECOND BY THE REAL-TIME CLOCK

% CLOCK
LEVI5 LDA DATO 1
COPY ADI $A DL
SAT 074
SAX 074
COPY CM1 $X DD
SMB 1
BACK COPY ST DA
RAND $L DT
REDO ST DA
JAF UT 1  % EXIT IF NOT TWELVE O'CLOCK (AM. OR PM.)
RADD $D1 DL

SMT 6
SMB 6
JNC BACK

DATE
SAA 077
LIX DATO
RLD 0110 DX
BMP BAC 060 DX
JMP # 011
BMP BAC 070 DX
BMP BAC 0100 DX
JMP # 3  % MONTH WITH 30 DAYS.
BMP ONE 0130 DX
BMP ZRO 0120 DX  % SKIP NEXT STATEMENT IF LEAP-YEAR.
AAU 2
AAT 2
AXT 2
AXX 1
RAND $X DA
RAND ST DA
JMP UT   % EXIT IF NOT END OF MONTH.
BMP ZRO 000 DX
JMP UT
RAND ST DX
AXX 2
BMP ZRO 0110 DX
BMP ONE 0100 DX
JMP UT  % EXIT IF NOT END OF YEAR.
BESS ONE 0100 DA
RAND $4 DX
UT STX DATO
COPY $L $A
STA DATO 1

% VELOCITY
LDX TB13 015
SAA 6
JFL 1 $FXX
NLZ 0020
FIIU 010.X
FDV 010.X
STF 4.X
MIN 7.X

% CALLING SEQUENCE FOR RUNNING SUBROUTINES
% EVERY SUBROUTINE HAS ONE SPECIFIED BIT IN TESTWORD (TS).
% INDICATING WHETHER IT IS RUNNING
JPL I TB13 2
LDA TS-K, B
BSKP ZRO 0130 DA
JPL I TB13 11
LDA TS-K, B
BSKP ZRO 0140 DA
JPL I TB13 014
LDA TE13 015
MST PID
LDA TS-K, R
BSKP ZRO 00 DA
JPL I TB13 0
LDA TS-K, B
BSKP ZRO 0150 DA
JPL I TB13 010
LDA DATO 1
SHA 012
JAF UT13
JPL I TB13 7
LDA TS-K, B
BDX TB13 4
BSKP ZRO 0040 DA
JPL I MPX-K, B
LDA TS-K, B
BDX TB13 5
BSKP ZRO 0050 DA
JPL I MPX-K, B
LDA TS-K, B
BDX TB13 6
BSKP ZRO 0040 DA
JPL I MPX-K, B
MIN TB13 1
JMP UT13
SHA -012
STA TB13 1
JPL I TB13 12
LDA TS-K, B
BDX TB13 4
BSKP ZRO 0060 DA
JPL I AV1-K, B
LDA TS-K, B
BDX TB13 5
BSKP ZRO 0050 DA
JPL I AV1-K, B
LDA TS-K, B
LDX TB13 6
BSKP ZRO 0040 DA
JPL I AV1-K, B
LDA TS-K, B

% TEST FOR FULL-HOUR OUTPUT
UT13: LDA DATO 1
SHA 4
JAF + 05
SHA -1
STA TS-K, B
SHA 020
MST PID
WAIT
JMP LEV12

DATO, 0
% DATE
0
% COUNTER
-012
% POINT TO DEAD-RECKON SUBROUTINE
GYRO
% POINT TO COURSE SUBROUTINE
ECSP
% POINT TO ECHO-Sounder SUBROUTINE
ATEMD
% POINT TO AIR-TEMPERATURE DATA-BLOCK
ATEMIN
% POINT TO DEWPOINT-TEMPERATURE DATA-BLOCK
APRES
% POINT TO AIR-PRESSURE DATA-BLOCK
AVCL
% POINT TO MEAN-COURSE SUBROUTINE
TD32
% POINT TO STD SUBROUTINE
WIND
% POINT TO WIND-DATA SAMPLING SUBROUTINE
WINDAV
% POINT TO WIND-DATA AVERAGING SUBROUTINE
VELOC
% POINT TO VESSEL VELOCITY DATA-BLOCK
SMPL
% POINT TO BOTTOM-DEPTH SUBROUTINE
010000
MPXX, MPY2
%KILL LEV12 BACK UT UT13 TB13

% COURSE SUBROUTINE
% FIRST PART OF THE ROUTINE IS A DECODING TO BINARY CODE
% THE RESULTS OF THIS ROUTINE ARE THE VESSEL'S COURSE, AND THE
% N-S AND E-W COMPONENTS OF THE VESSEL'S VELOCITY.

GYRO 10T ACT 0139
SHA P CLS 54 DT
COPIY CML 07 DD
BLDA O120 DT
BSKP O2 0120 DT
AAA 044
BSKP BAC 0110 DT
AAA 022
BSKP BCM 0110 DT
COPY CM1 ST DT
BSKP ZRO 0100 DT
SLDA 010 DA
COPY 92 DT
BSKP BCM 0060 DT
JMP = 9
AAA 3
COPY CM1 ST DT
BSKP ZRO 0050 DT
AAA 1
BSKP ZRO 040 DT
AAA 1
SAT 0
BSKP ONE 010 DD
JMP = 6
BSKP ZRO DO DD
MAT = -1
BSKP ZRO 020 DD
MAT = 1
JMP = 2
RMSD SD DT
SLDA 00 DA
BSKP BCM 000 DD
JMP = 9
COPY CM2 ST DT
MAT = 011
MPY M75-K,D
RAAS ST DT;
COPY SL DT
STT LR
NIL 000
FDV FL5-K,D
FMU DTR-K,D
STF LR 1
JPL I SIN-K,D
STF LR 4
LDP DRP04 4
JPL I COS-K,D
STF C101-K,D
FDV FL6-K,D
FDV VELOC 4
FAD LR 4
STF LR 4
LDP LR 1
JPL I COS-K,D
FDV LR 4
JPL I ATAN-K,D
FMU VELOC 4
STF COURS 015
FAD COURS 7
STF COURS 7
LDP COURS 4
JPL I COS-K,D
FMU VELOC 4
STF COURS 020
FAD COURS 012
STF COURS 012
JMP = LR

% DEAD-RECKON SUBROUTINE
% THE RESULTS OF THIS ROUTINE ARE THE VESSEL'S POSITION
% AND THE EARTH-RADIUS AT THE ACTUAL POINT.

DRECF, LDP C101-K,D
FMU C101-K,D
FMU C2
FAD C2
FMU C101-K,D
FMU C101-K,D
FAD C1
FMU ER-K,D
STF ERL-K,D
LDP COURS 020
FMU CO
FDV ERL-K,D
FAD DRP01 4
STF DRP01 4
FDV DTR-K,D
STF DRP01 4
LDP COURS 015
FMU CO
FDV ERL-K,D
FAD DRP02 4
STF DRP02 4
FDV DTR-K,D
STF DRP02 4
EXIT
% WIND-DATA SAMPLING SUBROUTINE
% THE RESULTS OF THIS SUBROUTINE ARE THE
% N-S AND E-W COMPONENTS OF THE WIND-FORCE

% RADIAN
VELOC.041062
% DATA-BLOCK VEESSEL'S VELOCITY
% DEGREES (MEAN)
% DEGREES (INSTANTANEOUS)
% CUMULATIVE SUM OF E-W COMPONENT OF VELOCITY
% CUMULATIVE SUM OF N-S COMPONENT OF VELOCITY
% INSTANTANEOUS E-W COMPONENT OF VELOCITY
% INSTANTANEOUS N-S COMPONENT OF VELOCITY

% DATA-BLOCK WIND-FORCE
% WIND-FORCE (MEAN)
% CUMULATIVE SUM OF E-W COMPONENT OF WIND-FORCE
% CUMULATIVE SUM OF N-S COMPONENT OF WIND-FORCE

% NUMBER OF OBSERVATIONS IN CUMULATIVE SUM

% DATA-BLOCK BOTTOM-DEPTH
% BOTTOM-DEPTH (INSTANTANEOUS)

% WIND-DATA AVERAGING SUBROUTINE

% DATA-BLOCK LONGITUDE
% DEGREES
% RADIANS

% DATA-BLOCK LATITUDE
% DEGREES
BSKP ZRO 0170 DX
FAK PI-K, B
BSKP ZRO 0170 DT
FAK TP1-K, B
STF WDAD 4
FDV DTR-K, B
STF WDAD 1
LAD WDAD 020
NLZ 020
STF LR 1
LDF WDAD 4
FNU WDAD 4
STF LR 4
LDF WDAD 7
FNU WDAD 7
FAD LR 4
JFL 1 SOR-K, B
FDV LR 1
FNU VELOC 13
STF WDAD 1
LDF FLO-K, B
STF WDAD 4
STF WDAD 7
STZ WDAD 020
JMP 1 LR
% COURSE AVERAGING SUBROUTINE
AVC1 COPY SL DA
STA LR1
LDF COURS 7
COPY ST DL
LDX COURS 12
JXZ # 4
FDV COURS 12
JFL 1 ATAN-K, B
JMP # 4
LDF PI2-K, B
BSKP ZRO 0170 DL
BSET ONE 0170 DT
BSKP ZRO 0170 DX
FAK PI-K, B
BSKP ZRO 0170 DT
FAK TP1-K, B
FDV DTR-K, B
STF COURS 1
LDF FLO-K, B
STF COURS 7
STF COURS 12
JMP 1 LR1
LR1: 0
% BOTTOM-DEPTH SUBROUTINE
BSKP 10T OCT 0131
BSKP ZRO 0170 DX
JMP +2
% DEVICE BUSY
BSKP ZRO 0150 DA
EXIT
% DEVICE OVERFLOW
SHA ROT 4
SAD ZIN SHR 014
MPV HT6-K, B
SWAP CLD SL DT
SAD 4
MPV HT5-K, B
RADD BA DT
RCLR DA
SAD 4
RADD ST DA
BSKP ZRO 0140 DD
MPV HT5-K, B
BLDA 0160 DD
NLZ 020
BSKP ZRO 089
FDV FL10-K, B
STF BMMP 1
EXIT
1KILL MPXX OVTO DRECF C1 C2 C3 C4 WDAD LR SMKPT AVC1 WNDAV LR1
% ECHO-ABUNDANCE SAMPLING SUBROUTINE
ECSP, MIN LOG: 1
JMP # 2
EXIT
% AVOID READING WHEN_RESETTING INTEGRATOR
RTI SSK
LDX <ECMD1
LDA 7, X
COPY SL DT
JFL 1 MPXX
COPY ST DL
SHA SHR 1
COPY 5A DT
SUB 011, X
JAN = 3  
ADD 010.X  
STA 010.X  
AAA 1  
JA1 = 2  
STT 011.X  
AAX 015  
MIN LO01  
JMP ECSP 5  
SAA -6  
STA LO01  
LDX SX9  
EXIT  
SX9,0  
LO01, -6  
-1  
% SAVING OF ECHO-ABUNDANCE LAST NAUTICAL MILE SUBROUTINE  

ECAD, SAA -1  
STA LO01 1  
LDF FL0-K.B  
STF SUM1 1  
STF SUM1 4  
STX EXP  
LDX ECND1  
LDA 010.X  
NLZ 020  
FMU 10, 04888  
FMU 012.X  
STF 1.X  
FAD 4.X  
STF 4.X  
FAD SUM1 4  
STF SUM1 4  
LDF 1.X  
FAD SUM1 1  
STF SUM1 1  
STZ 010.X  
STZ 011.X  
AAX 015  
MIN LO01  
JMP ECAD 7  
SAX -6  
STX LO01  
LDX SX9  
EXIT  
MFYY, NPY2  
DFILL  
ECND1, 142662  
% DATA BLOCK ECHO-ABUNDANCE (INTEGRATOR-CH. 1)  
% ECHO-ABUNDANCE LAST NAUTICAL MILE  
% CUMULATIVE SUM OF MILE VALUES OF ECHO-ABUNDANCE  
% MULTIPLEXER CHANNEL  
% ECHO-ABUNDANCE CURRENT NAUTICAL MILE  
% ECHO-ABUNDANCE SINCE LAST PEAK  
% INTEGRATOR GAIN  
ECMD2, 142462  
% DATA BLOCK ECHO-ABUNDANCE (INTEGRATOR-CH. 2)  
% 1  
% 2  
ECMD3, 142664  
% DATA BLOCK ECHO-ABUNDANCE (INTEGRATOR-CH. 3)  
% 3  
% 4  
ECMD4, 142465  
% DATA BLOCK ECHO-ABUNDANCE (INTEGRATOR-CH. 4)  
% 5  
% 6  
ECMD5, 142466  
% DATA BLOCK ECHO-ABUNDANCE (INTEGRATOR-CH. 5)  
% 7  
% 8  
ECMD6, 142467  
% DATA BLOCK ECHO-ABUNDANCE (INTEGRATOR-CH. 6)
KUDE
041
LEV12: LDA 71-10.H B
JSB F ONE 010 5A
JMP E12
LDX $X3
SAA 0
MPY LUP
ANX 030
JPL MPX2
COPY CMX SA 9X
LDT 013, X
STA 015, X
BLDA 0170 AA
REXO SN DT
COPY AD1 ED DA
JPL MPX2
COPY CMX SA 9A
JAF # 4
LEF P12-KK.B
BSET ZAC 0170 DT
JMP L12
JAN = 011
SNAP SA DT
JAP = 6
SAA 1
BEXP ZRO SHX
SHA = 1
ADD 014.X
STA 014.X
SNAP SA DT
NLZ 020
STF K1-KK.B
BLDA 0170 DT
LDA 015, X
NL2 020
FDV K1-KK.B
RESTA 0160 DX
JPL 1 ATAN-KK.B
BEXP ZRO 0160 DX
FAD P1-KK.B
L12: BEXP ZPO 0170 DT
FAD TP1-KK.B
FBV TP1-KK.B
STF 8.X
LDA 014, X
NLZ 020

% POINTER TO CHAIN-CONSTANTS
% NUMBER OF CONSTANTS FOR EACH CHAIN
% TEST WHETHER DECO-PROGRAM IS RUNNING
% EXIT

FAD 3, X
STF 0.X
ANX 016
MNI LUF
JMP LEV12 4
SHA = 9
STA LUP
JPL IMM
E12: WIT
JMP LEV12
% ROUTINE FOR READING OF MULTIPLEXER
MPX2: COPY SA DD
IOT ACT SHA 050
JMP #-1
COPY SD DA
IOT ACT SHA 050
JAF # 1
SHA 4
SHA 2HR 4
EXIT

$X3: VARIA
T20, T2X20
% POINTERS TO TEXT-BLOCKS
T2W, T2X2
% INITIALIZATION OF DECO-PROGRAM
INNLF: COPY $L DX
STV 02-H.B
LDT T20
JPL I TXT-H.B
L51
JPL I HTX-H.B
L52
MPY LEV12+1
ADD LEV12+2
STA CHI-KK.B
LDT T25
JPL I TXT-H.B
L51
LDA 3:X
DIN <PL I FLX-H.B
L52
FSB K1-K.B
FSV K2-K.B
STF OC-K.B
JMP INIT 2

% ROUTINE FOR CONVERTING LANE-READINGS TO HYPERBOLIC VARIABLES

SUE1,LDX SX3
LDA CH1-K.B
AAA 6
STA CH2-K.E
AAA 3
STA CH3-K.B
LDF .X
FSB 1 CH2-K.B
STF K1-K.B
LDF 3 .X
FSV K1-K.B
FMU FL2-K.B
FMU FL1-K.B
STF 4 .X
FMU 6 .X
FSB FL1-K.B
FMU 1 CH3-K.B
FMU MINUS-K.B
STF 011 .X
AAA 016
LDA CH3-K.B
MIN LUP
JMP SUE1 2
SAT -3
STT LUP
AAA -044
AAA -022
STA CH2-K.B
AAA 011
STA CH3-K.B

% TESTING SEQUENCE TO SELECT 2 SLAVES

TEST1:STX CHA 1-K.B
AAA 016
STX CHA 1-K.B
LDF -016.Y
BSET ZRO 0170 DT

STF K1-K.B
LDF .X
BSET ZRO 0170 DT
AAA 016
STF K2-K.B
FSB K1-K.B
PLDY 0170 DT
LDF .X
BSET ZRO 0170 DT
BSUP ZRO 28K
FSB K1-K.B
BSKP ONE 28K
FSB K2-K.B
BSKP ONE 0170 DT
JMP TEST2
LDA CH5-K.B
AAA 011
COPIE SX DD
BSKP ZRO 28K
STD CH2-K.B
BSKP ONE 28K
STD CH2-K.B
TEST2:LDA T1-K.B
BSKP ONE 020 DA
JMP NEXT

TEST3:LDP CH2-K.B
STD K1-K.B
LDO CH3-K.B
STD CH2-K.B
LDO K1-K.B
STD CH3-K.B
NEXT:STX 5
SRA -014
STA 1TH
LDA T1-K.B
BSKP ZRO 010 DA
JMP ITERA
LDA 1 CH2 1-K.B
PLKA 0170 DA
LDF TP1-K.B
BSKP ONE 28K
LDF FIX-K.B
PDV FL4-K.B
FSB PI-K.B
FMU FL2-K.B
FAD I CH2-K.B.X
BSHP ZRO 0170 DT
FAD TFI-K.B
STF TETA-K.B

% ITERATION-LOOP

ITERA. LDF TETA-K.B
FSE I CH2-K.B.X
JPL I SIN-K.B
FMU I CH2-K.B
FMU RR-K.B
FDV ERL-K.B
FDV MFL6
FAD TETA-K.B
FSE I CH2-K.B.X
JPL I COS-K.B
FAD I CH2 1-K.B
FDV I CH2 1-K.B.X
STF K3-K.B
LDF TETA-K.B.
FSE I CH2-K.B.X
JPL I SIN-K.B
FMU I CH2-K.B
FMU RR-K.B
FDV ERL-K.B
FDV MFL6
FAD TETA-K.B
FSE I CH2-K.B.X
JPL I COS-K.B
FAD I CH2 1-K.B
FDV I CH2 1-K.B.X
FSE K3-K.B
STF K1-K.B
FMU T
FAD TETA-K.B
STF TETA-K.B
LDF FL1-K.B
FDV K3-K.B
BSET ZRO 0170 DT
STF R1-K.B
FDV ERL-K.B
STF RR-K.B

LDF K1-K.B
BSET ZRO 0170 DT
FSE (G.00000000000001
BSHP ZRO 0170 DT
JMP UTP1

MIN'TI:
JMP ITERA
LDT I TEST7
JMP FEIL

ITL:O
% LOOP-COUNTER
MFL6. C=6
T:<500000. (USED IN ITERATION-LOOP)

% ROUTINE TO SELECT THE OTHER POINT OF INTERSECTION
SYTT LDLT T1-K.B
FAD TCH 200-v DA
STA T1-K.B
JMP TEST7

% ROUTINE TO CONVERT POINT OF INTERSECTION TO GEOGRAPHICAL COORDINATES
UTP1 LDF TETA-K.B
JPL I COS-K.B
STF K1-K.B
LDF TETA-K.B
JPL I SIN-K.B
STF K2-K.B
FMU K2-K.B
STF K2-K.B
FMU RC-K.B
FAD FC-K.B
FAD FR-K.B
STF K4-K.B
LDF K1-K.B
FMU K4-K.B
FMU RC-K.B
FAD K4-K.B
FAD RR-K.B
FMU RR-K.B
STF ORF-K.B
FAD I CH1-K.B
STF DPO8. 4
FDV DTR-K.B
STF DPO8. 1
LDF K2-K.B
FMU GC-K.B
FSE (G.00000000000001
BSHP ZRO 0170 DT
JMP UTP1

FMU RR-K.B
% ROUTINE TO SAVE LATEST POINT OF INTERSECTION

RESET, DM A -5
STA TTEST
LDF TETA-K, B
STF OTETA
LDF N1-K, B
STF N2-K, B
LDA T1-K, B
BSET ONE 010 DA
STA TI-K, B
JMP 10-K, B
% EXIT

% TESTING SEQUENCE FOR DISTANCE BETWEEN POINTS OF INTERSECTION

TESTA, LDF N1-K, B
FSB N2-K, B
BSET ZRO 0170 DT
FSB FLTUS-K, B
BSEP ONE 0170 DT
JMP = 7
LDF TETA-K, B
FSB OTETA
BSET ZRO 0170 DT
FSB DTR-K, B
BSEP ZRO 0170 DT
JMP RESET
LDF OTETA
STF TETA-K, B
MIN FTEST
JMP RESET 3
LDT [TEXTS]
% EXIT IF DISTANCE BETWEEN POINTS OF INTERSECTION IS TOO LARGE.
% OR IF NUMBER OF ITERATION EXCEEDS 100

FEIL, LDA T1-K, B
BSET ZRO 010 DA
STA T1-K, B
LDA DVS-K, B
BSET ONE 0140 DA
STA DVS-K, B
JPL I TXT-K, B
FEI
LDA DVS-K, B
BSET ZRO 0140 DA
STA DVS-K, B
JMP 12-K, B
% EXIT
FEI2% OUTPUT-DEVICE FOR ERROR-MESSAGE
FTEST,-5% NUMBER OF FAULTY READINGS BEFORE ERROR MESSAGE
OETEAL,0% ANGLE TO LATEST POINT OF INTERSECTION
TEXTS,2% KEEP=1, CHANGE=2.
WRONQ ITERATION

TEXTS, .
POS WR

)FILL

DPOS1.040661% DATA-BLOCK FOR DECCA-POSITION (LATITUDE)
0
DPOS2.040662% DATA-BLOCK FOR DECCA-POSITION (LONGITUDE)
0
T21.TXXZ% POINTERS TO TEXT-BLOCK
T22.TXXZ
VARIA,<0% DATA-BLOCK (RED SLAVE)
<24.0
<0
0
<300.0% DATA-BLOCK (GREEN SLAVE)
<18.0
<0
0
<500.0% DATA-BLOCK (PURPLE SLAVE)
<30.0
<0
0
0
% CONSTANT-BLOCKS FOR VARIOUS DECCA-CHAINS
KJYDE,CL 22655735% DECCA-CHAIN 0 (FINMARK)
00.44510244
0424.99

%.170246.
<1, 99771009
<632, 419
<2005,28, 0
<1, 50117017
<759, 44
<505379, 0
<0, 10547020
<1, 20622257
<0, 17777253
<642, 13
<182059, 0
<0, 50155096
<706, 778
<250744, 2
<8, 07671212
<727, 46
<27268, 1
<2, 61945777
<1, 19824424
<0, 21768300
<376, 61
<1649506, 0
<8, 16500413
<748, 98
<152409, 0
<8, 95461917
<718, 7
<74374, 0
<1, 41284461
<1, 10547259
<0, 14756007
<476, 99
<219802, 5
<0, 67161275
<641, 74
<212717, 6
<9, 09422632
<779, 25
<589327, 4
<2, 10521775
<41, 05129264
<0, 08762220
<444, 44
% PROGRAM ON LEVEL 11
% THIS LEVEL RECEIVES INTERRUPT-SIGNALS FROM TELETYPES (INPUT),
% PAPER-TAPE READER, AND SHIP'S NAVIGATIONAL LOG.
% PROGRAM-TRIGGERED INTERRUPT IS GIVEN TO OTHER LEVELS, DEPENDING ON
% INTERRUPTING DEVICE.

LEVL : IOT SNI 2
JMP T1U
IOT SNI 4
JMP T12U
IOT SNI O10
JMP T13U
IOT SNI 022
JMP TREAD
IOT SNI 0123
JMP TREAD 2
WAIT
JMP LEVL

T1U : SAT 3
LDA DV3-K-5
BSET ONE 0110 DA
STA DV3-K-5
JMP T1SU 4

T1SU : SAT 5
LDA DV5-K-5
BSET ONE 0110 DA
STA DV5-K-5
JMP T1SU 4

T3U : SAT 011
LDA DV11-K-5
BSET ONE 0110 DA
STA DV11-K-5
LDY I (LP1)
JXZ *0
BMP IP DX EQL ST
JMP *5
STT I (LP1)
MAT -1
STT I (LP1)
MST PID

)LINE

% DECCA-CHAIN 5 ( SKAGERRAK )
% DECCA-CHAIN 6 ( DANISH )
% DECCA-CHAIN 7 ( NORTH-SCOTTISH )

)KILL PNP SYS LUS MUX 14L 15L DECCA TEST1 TEST2 TEST3 NEXT1 ITEA
)KILL DECC1 LIST 14L 15L DECC1 LUS 14L 15L TEST1 TEST2 TEST3
)KILL SUBR1 LUP1
JMP LEVI1
TREAD LDA DW22-K, B
JMP TREAD-2
RCLR DA
BSET one 0120 DA
IOT PIN 0120
JMP TREAD-2

)FILL
)
KILL LEVI1 T71U T72U T73U TREAD

)CALLING SEQUENCE FOR ACTIVATED SUBROUTINES

)LEV10: LDA I TABLO 5
)ADD 2
)JNZ SEC
LDX TABIO
LDA TEK-K, 6
BSNP IRO 0160 DA
JPL I MPX-K, 2
LDX TABIO 1
LDA TEK-K, 6
BSNP IRO 0160 DA
JPL I MPX-K, 3
LDX TABIO 2
LDA TEK-K, 6
BSNP IRO 0160 DA
JPL I MPX-K, 3
LDX TABIO 4
LDF 1, X
FAD (0, 1)
BSNP IRO 0170 DA
FAD PLUS-K, B
STF 1, X
LDA 4, X
RAND DA
SAT 012
SHF IF DA LST IT
RCLR DA
STA 4, X
JAP UTL10
LDX TABIO 8
LDA 7, X
STI 7, X
NLZ 020
STF 016, X
LDF 013, X
PDP 016, X
STF 1, X
LDX TABIO
LDA TEK-K, 6
BSNP IRO 0160 DA
JPL I AVI-K, 8
LDX TABIO 1
LDA TEK-K, 6
BSNP IRO 0160 DA
JPL I AVI-K, 6

) PROGRAM ON LEVEL 10
) THIS LEVEL IS TRIGGERED BY INTERRUPT FROM THE SHIP'S NAVAIGATIONAL
) LOG EVERY CABLE'S LENGTH (0.1 NAUTICAL MILE)
) THIS SUBROUTINE PRODUCES TABLE OF BOTTOM BUCK SCATTERING VALUES
) SAMPLED ON LEVEL 14.
) SEC, LDMA TABIO 5
) BAA 3
) MPY 4, X
) COPY DA DX
) OCD I BECT
) NLZ 020
) STF BECT 2
) LDP I BECT 1
) PDP BECT 2
) STF I 1, X (BSCR
) LDP FLO-K, 5
) STF I BECT 1
) STZ I BECT
) JMP LEVI1 3
) BECT.BC
) DES
) BSCR. CO
) CO
) CO
) CO
) CO
) CO
) CO
) CO
) CO
)
LDX TAB10 2
LDA TS-K, B
BSEP ZERO 0120 DA
JPL I AV1-K, B
LDA TS-K, B
BSEP ZERO 00 DA
JPL I TAB10 5
SAA -1
STA TS-K, B
SAA 020
MST PID
UTL10. WAIT
JMP LEVI0

% INTERRUPT TO LEVEL 4 FOR OUTPUT

TAB10: STEMP
SSALI
TRANS
VELOC
LOG
ECAD
LOG0

% DATA BLOCK (SEA-SURFACE TEMPERATURE)
% DATA BLOCK (SEA-SURFACE SALINITY)
% DATA BLOCK (SEA-SURFACE TRANSPARENCY)
% DATA BLOCK (SHIP'S SPEED)
% POINTER TO DATA BLOCK (SHIPS NAVIGATIONAL LOG)
% POINTER TO ECHO-ABUNDANCE SUBROUTINE
% TESTWORD FOR ACTIVATED ECHO-SAMPLING ROUTINE

% DATA BLOCK (AIR-TEMPERATURE)
% MEAN TEMPERATURE LAST TEN MINUTES
% MULTIPLEXED CHANNEL (EIT 5-15)
% LAST OBSERVED VALUE
% CUMULATIVE SUM OF OBSERVED VALUES

% DATA BLOCK (AIR DENSITY POINT TEMPERATURE)
% DATA BLOCK (AIR PRESSURE)
% DATA BLOCK (SEA-SURFACE TRANSPARENCY)
% DATA BLOCK (SEA-SURFACE SALINITY)
% DATA BLOCK (AIR-TEMPERATURE)
% DATA BLOCK (SEA-SURFACE SALINITY)
% DATA BLOCK (SEA-SURFACE SALINITY)
% DATA BLOCK (SEA-SURFACE SALINITY)

% CONSTANTS FOR THE VARIOUS CHANNELS ON THE TERMO-SALINOGRAPH

TTAB, CR 2. 2471P
CR 3. 05
CR 1. 60514
CR 2. 2471P
CR 1. 60514
CR 1. 60514
CR 25.
STAB- 01 60514
01 60514
026
50 9257
081
00 52107
08A
00 52107
08B 96
00 52107
082 487
00 52107
08A

COPY CMC SA DA
STP CISI-K, S
STA CISI 1-4-H, S
LDT TXO 2
JPL I TXT-K, B
L91
LDF SRT-K, B
JPL I SORTE-K, B
L91
2
3
LDT TXO 3
JPL I TST-K, B
L91
LDF STK-K, S
JPL I DORTE-K, B
L91
1
3
LDT TXO 4
JPL I TXT-K, B
L91
LDF SYB-K, B
JPL I DORTE-K, B
L91
1
3
JPL I CLF-K, B
L91
LDX LVPA 2
LDF FLO-K, B
STF .X
STF -35 .X
STF -30 .X
STF 14 .X
SAN -6
STA CISI 1-K, B
SAN -1
STA SS-K, B
SAN 020
JMP * 2
SMB SA 2
ORA .B
STA .B
JMP I LEAV

% PROGROM ON LEVEL 8
% INITIATION OF STD SUBROUTINE
TDS: LDT TXO 1
JPL I TXT-K, B
L91
JPL I HTI-K, B
L92
COPY SA DX
LDA DVT-K, B
BESET ONE 0110 DA
STA DVT 1-H, B
LDA LVPA 3
JPL I RIO-K, B
LVPA 4
COPY SX DA
JPL I HTO-K, B
LVPA 4
4
JPL I CLF-K, B
LVPA 4
LDA DVT-K, B
BESET ZRO 0110 DA
STA DVT-K, B
LDT TXO 5
JPL I TXT-K, B
L91
JPL I HTI-K, B
L92

% INTERRUPT TO LEVEL 1 FOR USE OF MAC
ORI .B
STA .B
JMP I LEAV

% INPUT CALIBRATION DATA STD

KILL BSC LEV10 UTIL10 TAB10
JFL I TXT-K: B
L91
JFL I FLIN-K: B
L92
LDT TXO 3
JFL I TXT-K: B
L91
JFL I FLIN-K: B
L92
STF SRT-K: B
LDT TXO 4
JFL I TXT-K: B
L91
JFL I FLIN-K: B
L92
STF SNS-K: B
JMP I LEAV2
TXD, TXO % POINTERS TO TEXT BLOCK
TXO 33
TXO 36
TXO 37
TXO 39
KODE, 041505 % COUNTER FOR CODE IDENTIFICATION
NOTIN, JMP 1 LVPA 1 % EXIT IF ILLEGAL ABBREVIATION
LVPA, LEAV2
NITK
TDS1
041465
7
X TEST-SEQUENCE FOR DESIRED SUBROUTINE
LEAV2, LDT TXO
JFL I TXT-K: B
L91
JFL I INP2-K: B % INPUT ABBREVIATION
L92
SUB KODE
JAP NOTIN
COPY SA DX
JFL I CLF-K, B
L91
JPC STDS1 % JUMP TO STOP STD
JPC DSTOP % JUMP TO STOP DECCA
JPC STOP % JUMP TO STOP ECHO-CLASSIFYING
JPC ESTOP % JUMP TO STOP ECHO-ABUNDANCE
MAX 177
MAX 145
JPC NOTIN
JPC EMB % JUMP TO MAC
JPC KORR % JUMP TO CALIBRATION OF STD
JPC TDS % JUMP TO START STD
JPC DECIN % JUMP TO START DECCA
JPC CLASS % JUMP TO START ECHO-CLASSIFYING
JPC DRS % JUMP TO START DISPLAY
JPC LOG1 % JUMP TO START NAVIGATIONAL LOG
JPC DRIN % JUMP TO START DEAD-RECKON
JMP 1 LVPA

% C-Routines, Stop of Programs
ESTOP, LDA T5-K: B % STOP ECHO-ABUNDANCE
BSET IRO 00 DA
JMP * 00
STOP, LDA T5-K: B % STOP ECHO-CLASSIFYING
BSET IRO 0160 DA
STA T5-K, B
JMP I LEAV2
DSTOP, LDA T1-K: B % STOP DECCA
BSET IRO 0130 DA
STA T1-K, B
JMP I LEAV2
STDST, LDA T5-K: B % STOP STD
BSET IRO 0150 DA
STA T5-K, B
JMP I LEAV2
LEAV2, LEAV1 % EXIT
XKILL KODE NOTIN LVPA ESTOP DSTOP STDST

% B-Routines, Start of Programs
DECSN, LDA T1-K: B % INITIATION OF DECCA SUBROUTINE
BSET IRO 010 DA
STA T1-K, B
JPL I INIT
JMP I LEAV2

INIT., INBLUE

LDT ,X
JPL I TXT-K.B
L91
JPL I MTI-K.B
L92
SWAP SX DS
STA I 1.B
SWAP SX DS
AAX 2
LDA ,X
JMP *-012
LDA I ECT 4
MLI 020
FDV ECT 020
STF I ECT 4
LDA I ECT 6
COPY CM2 8A DA
STA I ECT 6
SAA -050
MFI I ECT 010
STA I ECT 010
LDX ECT 016
ST2 I ECT 017.X
JNC =1
SAA -1
STA B4-K.B
SAA 020
ORA .B
STA .B
LDA 75-K.B
BSET ONE 0160 DA
STA 75-K.B
JMP I LEAV2

DLR. LDT ECT -1

% INITIATION OF DISPLAY-SUBROUTINE
JPL I TXT-K.B
L91
JPL I MTI-K.B

L92
LDX T1-K.B
BCS O170 DA
ESTA 050 DX
STDA 00 DA
STDA 050 DX
STX T1-K.B
JMP I LEAV2

LOG1.LDT TX1 3

% INITIATION OF NAVIGATIONAL LOG SUBROUTINE
JPL I TXT-K.B
L91
JPL I FLIN-K.B
L92
STF LOG 1
INZ -020
COPY CM2 8A DA
MLI 020
FAD LOG 1
FMU FL10-K.B
INZ -020
STA LOG 4
JMP I LEAV2

LOG.O41065%

% DATA-BLOCK FOR NAVIGATIONAL LOG

0
% CABLE-LENGTH COUNTER IN CURRENT NAUTICAL MILE

DEMN.LDX TWO 1

% INITIATION OF DEAD RECKON SUBROUTINE
LDT TX1 1
JPL I TXT-K.B
L91
JPL I FLIN-K.B
L92
STF CP1-K.B
JPL I FLIN-K.B
L92
FDV ECT 020
FAD CP1-K.B
STF 1.X
FMU DTR-K.B
STF 4.X
AXX 7
LDT TX1 2
MI TWO
JMP DRIN 2
SAA -2
STA TNO
LDA TS-K,B
BSET ONE,C13 DA
STA TS-K,B
JMP I LEAV3
TNO,-2
DRP01
TXX24
ECT,ECT ;
TXX26
P
TXX27
PULSR
TXX28
ANTAL
TXX29
B0S
TXX30
LOG6
TXX31
ANTAL-2
0
-02251
022001
$60.

F1,6
TXX
STAB
STEPM 013
STEPM 016
SSA1 013
SSA1 014
CLT,DAT0
013770
TX1,TXX1
TXX2
TXX3
TXX4
TX17,TXX17
TX18

TXX19
TXX20
LEAV3,LEAV1 % EXIT

CLOCK,LDT TY17 % INITIATION OF DATE AND TIME SUBROUTINE
JPL I TX7-K,B
L91
JPL I HTK-K,B
L92
SHA 1
COPY SA DX
JPL I HT1-K,B
L92
SHA &
RADD SA DX
JPL I HT1-K,B
SHA 012
RADD SX P
STA I CLT
LDT TX1
JPL I TX7-K,B
L91
JPL I HT1-K,B
L92
ANA -014
JAP = 3
ANA 014
JMP + 2
MIN I CLT
SHA 014
COPY SA DX
JPL I HT1-K,B
L92
SHA &
RADD SA DX
JPL I HT1-K,B
L92
RADD SX LA
SAX 1
TA I CLT-X
LDA CT I
GRA TS-K,B
ST TS-K,B
SAA -01

% LOOP-COUNTER
% POINTER TO LATITUDE (DEAD RECKON POSITION)
% POINTER TO NEXT LOCATION
% POINTER TO TEXT BLOCK
% POINTER TO DETAILED ROUTINE (ECHO-CLASSIFYING)
% POINTER TO TEXT BLOCK
% POINTER TO NUMBER OF FISH TO SEE CLASSIFIED
% POINTER TO TEXT BLOCK
% POINTER TO DETAILED ROUTINE (ECHO-CLASSIFYING)
% POINTER TO TEXT BLOCK
% TABLE-SPACE FOR ECHO CLASSIFYING
% UPPER GROUND FOR TABLE (ECHO-CLASSIFYING)
% POINTER TO DATE AND TIME
% POINTERS TO TEXT BLOCK
STA C01 1-K,E
RCRL DA
BSET ONE 0150 DA
MST PIE
JMP I LEAV3

TMBAL: LDT TX17 1 % INITIATION OF THERMO-SALINOGRAPH SUBROUTINE
JPL I TXT-K,E
L01
JPL I HTI-K,E
L02
LDX F11
MPY F1
RADD 8A DX
LDF -6,X
STF I F1 5
LDF -3,X
STF I F1 4
LDT TX17 2
JPL I TXT-K,E
L03
JPL I HTI-K,E
L04
LDX F1 2
MPY F1
RADD 8A DX
LDF -6,X
STF I F1 5
LDF -3,X
STF I F1 6
JMP I LEAV3

ECST: LDA T5-K,E % INITIATION OF ECHO-ABUNDANCE SUBROUTINE
BSET ZRO 20 DA
SA T5-E,B
LDX ECH
LDT TX17 2
JPL I TXT-K,E
L01
LDA L004
ARH 667
JPL I RIO-K,E
L01
LDA BS-N,E

JPL I RIO-K,E
L01
LDT ECH 1
JPL I TXT-K,E
L01
JPL I HTI-K,E
L02
COPY SX DL
COPY 8A DX
LDF I ECH 4
JXZ #4
FPU F10-K,E
ANX -512
JMP #3
COPY SL DX
STF 012,X
ANX 015
JPL I CLF-K,E
L01
MIN LOOP4
JMP ECHST 4
SAA -A
STA LOOP4
LDT TX1: 2
JPL I TXT-K,E
L01
JPL I HTI-K,E
L02
COPY CM2 8A DT
STT I ECH 1
MLT 800
STF I ECH 2
SAA -1
STA I ECH 3
LDA T5-K,E
BSET ONE 00 DA
STA T5-K,E
JMP I LEAV3
JMP CLOCK

LOOP4,-A % LOOP COUNTER
ECH,ECHM1 % POINTER TO ECHO-ABUNDANCE DATA-BLOCK
E1,L3 2 % POINTER TO NO. OF MILES BETWEEN OUTPUT (ECHO-ABUND.)
E2,L3 3
E3,L3 6
TDS-6

% KILL TRK: DTRD 0017 CLAS ECH DTRD L001 BDRN TWD F1 CLT TX1 TX17 LEAV2
LEVPA, JPC LOOP4-1 % JUMP TO START DATE AND TIME
JPC THGAL % JUMP TO START THERMO-BLINDOGRAPH
JPC ECST % JUMP TO START ECHO-ABUNDANCE
AAX 177
AAX 166
JPC FATION
JPC CHECK % JUMP TO REQUEST MULTIFLEXER-CHANNELS
JPC MCT % JUMP TO REQUEST METEOROLOGICAL OBSERVATIONS
JPC OC % JUMP TO REQUEST SURFACE TEMP. AND SALINITY
JPC CUR % JUMP TO REQUEST SHIP'S COURSE
JPC SPEED % JUMP TO REQUEST SHIP'S SPEED
JPC EDEPR % JUMP TO REQUEST ECHO-DEPTH (BOTTOM)
JPC ULOG % JUMP TO REQUEST NAVIGATIONAL LOG
JPC CLPRI % JUMP TO REQUEST TIME
JPC DEPOS % JUMP TO REQUEST DECCA POSITION
INTN: JMP I = I

DEPOS: LDX PSADR % OUTPUT OF DECCA POSITION
JMP # 2

DRPOS: LDX PSADR 1 % OUTPUT OF DEAD RECKON POSITION
LDT TX2
JPL I TXT-K.B
L91
JPL I KONV-K.B
L91
3
1
LDT TX3
AAX 7
MIN PSADR 2
JMP -11
SAA -2
STA PSADR 2
JMP LEAV1

PSADR.DPOS1: % POINTER TO DECCA POSITION
DRPOS1 1 % POINTER TO DEAD RECKON POSITION
-2 % LOOP-COUNTER
BMIDP 1 % POINTER TO ECHO-DEPTH
VELOC 4 % POINTER TO CURRENT SHIP'S SPEED
COURS 1 % POINTER TO SHIP'S COURSE
MPX2 % POINTER TO INPUT ROUTINE FOR MULTIFLEXER
STEMP % POINTER TO SURFACE TEMPERATURE DATA-BLOCK

SSAIL % POINTER TO SURFACE SALINITY DATA-BLOCK
APRES 1 % POINTER TO AIR PRESSURE
ATEMD 1 % POINTER TO AIR TEMPERATURE
ATEMD 1 % POINTER TO AIR TEMPERATURE
NMDF 1 % POINTER TO WIND FORCE
NMD 1 % POINTER TO WIND DIRECTION
LOI 1 % POINTER TO NAVIGATIONAL LOG

CLPRI: LDT TX2-1 % OUTPUT OF TIME
JPL I TXT-K.B
L91
JPL I TIME-K.B
L91
JMP LEAV1

ULOG, LDX PSADR 16 % OUTPUT OF NAVIGATIONAL LOG
LDT TX4
JMP COUR 2

EDEPR, LDX PSADR 3 % OUTPUT OF ECHO-DEPTH
LDT TX9
JMP COUR 2

SPEED, LDX PSADR 4 % OUTPUT OF SHIP'S SPEED
LDT TX9
JPL I TXT-K.B
L91
LDF-X
FXU 7, X
JMP COUR 5

TX21 % POINTERS TO TEXT-BLOCK
TX22, TX23
TX24, TX25
TX26, TX27
TX28, TX29
TX30
TX31
TX32
TX33
TX34
TX35

COUR, LDX PSADR 5 % OUTPUT OF SHIP'S COURSE
LDT TX10
JPL I TXT-K.B
L91
LDF , X
JPL I DORTE-K.B
L91
4
D1
JMP LEAV1

DC, LDT TX10 1 % OUTPUT OF SEA SURFACE TEMPERATURE AND SALINITY
JPL I TXT-K.B
L91
LDX PSADR 7
JPL MPX5
JPL I DORTE-K.B
L91
5
2
LDT TX10 2
JPL I TXT-K.B
L91
LDX PSADR 10
JPL MPX3
JPL I DORTE-K.B
L91
3
JMP LEAV1

MET,LDT TX10 3 % OUTPUT OF METEOROLOGICAL OBSERVATIONS
JPL I TXT-K.B
L91
SAX -5
STA MET1 1
LDX MET1
SWAP SX DB
LDF 1,B
SWAP SX DB
JPL I DORTE-K.B
L91
5
1
MIN MET1 1
JPRC -010
JMP LEAV1
MET1,PSADR 011
0

CHK, IOT ACT 52 % OUTPUT OF MULTIPLEXER CHANNELS
SAX -030
COPY SX DA
AAA 30
JPL I PIAADR 3
JPL I HTG-K.B
L91
5
JPL I CLF-K.B
L91
JNC - -010
IOT ACT 52
JPL I HTG-K.B
L91
5
JMP LEAV1

MTKH,LDT TX32 % OUTPUT IF UNKNOWN ABBREVIATION IS USED
JPL I TXT-K.B
L91
LEAV1, JPL I CLF-K.B
L91
SAX 2
LDA L91
LDT DV$-K.B
BSKP ZRO 010 DA
BSET ZRO 0110 DT
BSKP ZRO 0110 DT
STX L91
STT DV$-K.B
AXX 2
LDT DV$5-K.B
BSKP ZRO 000 DA
BSET ZRO 0010 DT
BSKP ZRO 0010 DT
STX L91
STT DV$5-K.B
AXX 4
LDT DV$11-K.B
BSKP ZRO 000 DA
BSET ZRO 0010 DT
BSKP ZRO 0010 DT
STX L91

STT DV11-K: B
LDX LP1
AAA -1
STX LP2
AAA 1
SKP IF DA 80 3X
JMP # 6
STX LP1
LDA # 8
STX 8
MST PID
WAIT
JMP 1 LEV98
MPX3, LDA 4, X
SHA ZIN-SHR 010
COPY SA DD
IOT ACT SHA 050
JMP # 1
COPY SD DA
IOT ACT SHA 050
JMP # 1
SHA 4
SHA SHR 4
NLJ 020
FDV MPP-K, B
FMU 013, X
FAD 016, X
EXIT
TX92, TX92
% CURRENT OUTPUT DEVICE FOR THIS LEVEL
% CURRENT INPUIT DEVICE FOR THIS LEVEL
TX93, 'TRHF'
TX98, 'DPT'
TX10, 'CRI'
TX11, 'PREP'
TX12, 'HRT'
TX13, 'DEEP
TX14, 'LUX
TX15, 'HP'
TX16, 'WDR
TX17, 'DATE
TX18, 'TEMP-RANGE
TX19, 'SAL-RANGE
TX20, 'CH NO
TX21, OBSERVATIONS
TX24, 'IR=1 DEC=1 DIS=0
TX25, 'RED GREEN PURPLE
TX26, 'GAIN
TX27, 'PULS-RATE
TX28, 'TOTAL
TX29, 'FRAME
TX30, 'PROGRAM
TX31, 'MAX DB
TX32, NOT KNOWN
TX33, 'ST NO
TX34, 'DEEP
TX35, 'TEMP.
TX36, 'SAL.
TX37, 'SIGMA-T
TX38, 'ALFA
TX39, 'DELTA
TX40, 'CORRECTION FOR DEPTH.
TX41, 'TEMPERATURE.
TX42, 'SALINITY.
TX43, 'SEASURFACE TEMPERATURE.
TX44, 'SALINITY.
TX45, 'PRESSURE
TX46, 'TEMP.
TX47, 'DEW-P.
TX48, 'W-FORCE
TX49, 'W-DIR.
WILL LEAV3 CLOCK THSAL ECST LOOPS CHIO ECW DEPO3 DRFOS PADD TX2 TX3
WILL CLRPR LDRPZ SPEED TX5 CUR T1 P TR2 TX3 CHECK NHK LEAV3 TX33
WILL LEVEL F PULSR BOKS LEVEL KERR

% PROGRAM ON LEVEL 6. THIS LEVEL IS NOT USED.
Lev6: Wait
Jmp +1

% PROGRAM ON LEVEL 7

% IDENTIFICATION OF INTERRUPTING OUTPUT DEVICES
Lev7: Iot SNi 3
Jmp iTI
% INTERRUPT FROM TELLETYPE 1
IOT SNi 5
Jmp IT2
% INTERRUPT FROM TELLETYPE 2
IOT SNi 011
Jmp IT3
% INTERRUPT FROM TELLETYPE 3
IOT SNi 7
Jmp Tal1
% INTERRUPT FROM PAPER TAPE PUNCH 1
IOT SNi 017
Jmp Tal2
% INTERRUPT FROM PAPER TAPE PUNCH 2

Wait
Jmp Lev7

T1: LDA DV3-K.B
IOT PIN 2
MST PID
Jmp Lev7

T2: LDA DV5-K.B
IOT PIN 4
Jmp TTI 2

T3: LDA DV11-K.B
IOT PIN 010
Jmp TTI 2

Tal1: LDA DV7-K.B
Jmp TTI 2

Tal2: LDA DV17-K.B
Jmp TTI 2

)KILL Lev7 TTI TTS TAL1 TAL2

% PROGRAM ON LEVEL 6. THIS LEVEL IS NOT USED.

Lev6: Wait
Jmp +1

% LINE

% LEVEL 5

% OUTPUT OF STD-DATA ON PAPER TAPE
% LISTING AND PLOTTING OF ECHO CLASSIFICATION DATA

Tarox, Co
% X-COORDINATE FOR PLOTTING
Tarox, Co
% X-COORDINATE FOR PLOTTING
Tarox, 2
% PLOT PEN UP OR DOWN
Tadr, ADR 1
% POINTER TO HEADING
Tdata, TDS1 011
% POINTER TO STD-DATA, PUNCH
041444
% OUTPUT DEVICE

Std, LDA DV7-K.B
% IDENTIFICATION CODE, STD
BSET ONE OOS DA
Sta DV7-K.B
LDA TDATA 2
Jpl 1 R10-K.B
Tdata 1
Sna -011
Sty = 5
Ldp 1 TDATA X
Jpl 1 DORTE-K.2
Tdata 1
4
0
Sna 2
Sta =2
Sna 040
Jpl 1 R10-K.B
Tdata 1
Any 2
Jnc 8-013
Jpl 1 CLF-K.B
Tdata 1
Lda DV7-K.B
Bset IOR OOS DA
Sta DV7-K.B
Wait
LEVS LDA T5-K.B
BSKP ZRO 0150 DA
JMP STDFC
BSKP ONE 0160 DA
JMP LEVS -1
LDA I (LOGO
BSKP ZRO 0170 DA
JMP TAROU
LDA DV11-K.B
BSET ONE 050 DA
STA DV11-K.B
LDT TAROP 1

% IDENTIFICATION OF ENTRY
% JUMP TO PUNCH ROUTINE. STD
% EXIT
% JUMP TO LISTING. ECHO STRENGTH SPECTER
% LISTINO. ECHO CLASSIFICATION

AAA -030
STA ADR
LDA T5-K.B
BSKP ONE 0160 DA
JMP LEVS -1
LDA DV11-K.B
BSET ONE 050 DA
STA DV11-K.B
SAX -4
LDA I ADR \ X
JPL I HTO-K.B
DEK
3
JNC = -4
SAA 040
JPL I RTO-K.B
DEK
BSET ONE 0150 DX
STX ADR
LDX I PING
COPY CMZ DX DX
LDA I ADR X
JPL I HTO-K.B
DEK
2
RINC DX
JNC = -5
JPL I CLF-K.B
DEK
LDA I (LOGO
JNZ = 055
LDX (023000
LDA \ X
NLZ 020
FDP (102.2
STFP TAROV
JPL I TRACE-K.B
TAROV
TAROV
LDP TAROV
FAD (0.1
STFP TAROV
LDA I 'PING T
SKP IF DX ORE SA
JPC -015
LDP FLO-K.B
STFP TAROV

JPL I TXT-K.B
DEK
LDA DV11-K.B
BSET ZRO 050 DA
STA DV11-K.B
RCRD DA
BSET ONE 0170 DA
MST PIE
WAIT
MCL PIE
SMA ZIN SHR 2
<table>
<thead>
<tr>
<th>Line</th>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LBDX</td>
<td>LOADER</td>
<td></td>
</tr>
<tr>
<td>JPL</td>
<td>I TXYTH</td>
<td></td>
</tr>
<tr>
<td>JPL</td>
<td>I TTT5</td>
<td></td>
</tr>
<tr>
<td>LDX</td>
<td>C121 1-K. B</td>
<td></td>
</tr>
<tr>
<td>JXZ</td>
<td>* 5</td>
<td></td>
</tr>
<tr>
<td>LDT</td>
<td>DAO</td>
<td>1</td>
</tr>
<tr>
<td>JPL</td>
<td>I TXT-K. B</td>
<td></td>
</tr>
<tr>
<td>DEVTT</td>
<td></td>
<td></td>
</tr>
<tr>
<td>STZ</td>
<td>C121 1-K. B</td>
<td></td>
</tr>
<tr>
<td>SAA</td>
<td>17</td>
<td></td>
</tr>
<tr>
<td>STA</td>
<td>DEVTT</td>
<td></td>
</tr>
<tr>
<td>LDD</td>
<td>DUT-K. B</td>
<td></td>
</tr>
<tr>
<td>BSET</td>
<td>ZRO 040 DA</td>
<td></td>
</tr>
<tr>
<td>BSET</td>
<td>ZRO 040 DD</td>
<td></td>
</tr>
<tr>
<td>STD</td>
<td>DUT-K. B</td>
<td></td>
</tr>
<tr>
<td>JXZ</td>
<td>* 5</td>
<td></td>
</tr>
<tr>
<td>LDA</td>
<td>T5-K. B</td>
<td></td>
</tr>
<tr>
<td>BSET</td>
<td>ONE 0100 DA</td>
<td></td>
</tr>
<tr>
<td>STA</td>
<td>T5-K. B</td>
<td></td>
</tr>
<tr>
<td>JXN</td>
<td>* 2</td>
<td></td>
</tr>
<tr>
<td>SAA</td>
<td>040</td>
<td></td>
</tr>
<tr>
<td>MST</td>
<td>PID</td>
<td>% INTERRUPT TO LEVEL 5</td>
</tr>
<tr>
<td>JMP</td>
<td>LEV4</td>
<td></td>
</tr>
<tr>
<td>DAG</td>
<td>DATE</td>
<td>% POINTER TO OUTPUT ROUTINE FOR DATE</td>
</tr>
<tr>
<td>TXYTH</td>
<td>% POINTER TO TEXT</td>
<td></td>
</tr>
<tr>
<td>TOC</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LDX</td>
<td>T5-K. B</td>
<td></td>
</tr>
<tr>
<td>JXZ</td>
<td>* 5</td>
<td></td>
</tr>
<tr>
<td>BSET</td>
<td>ONE 0100 DX</td>
<td></td>
</tr>
<tr>
<td>STX</td>
<td>T5-K. B</td>
<td></td>
</tr>
<tr>
<td>JMP</td>
<td>* 5</td>
<td></td>
</tr>
<tr>
<td>L1</td>
<td>MIN T4-K. B</td>
<td></td>
</tr>
<tr>
<td>JMP</td>
<td>LOG TP</td>
<td></td>
</tr>
<tr>
<td>IOT</td>
<td>ACT</td>
<td>5</td>
</tr>
<tr>
<td>LDF</td>
<td>LOAD</td>
<td></td>
</tr>
<tr>
<td>MIN</td>
<td>I DEVTT</td>
<td>1</td>
</tr>
<tr>
<td>JMP</td>
<td>* 5</td>
<td></td>
</tr>
<tr>
<td>IOT</td>
<td>ACT</td>
<td>011</td>
</tr>
<tr>
<td>LDX</td>
<td>T5-K. B</td>
<td></td>
</tr>
<tr>
<td>BSET</td>
<td>ONE 0100 DX</td>
<td></td>
</tr>
<tr>
<td>STX</td>
<td>T5-K. B</td>
<td></td>
</tr>
<tr>
<td>JMP</td>
<td>* 5</td>
<td></td>
</tr>
<tr>
<td>L1</td>
<td>MIN T4-K. B</td>
<td></td>
</tr>
<tr>
<td>JMP</td>
<td>LOG TP</td>
<td></td>
</tr>
<tr>
<td>IOT</td>
<td>ACT</td>
<td>5</td>
</tr>
<tr>
<td>LDF</td>
<td>TIADD</td>
<td></td>
</tr>
<tr>
<td>STD</td>
<td>C42-K. B</td>
<td></td>
</tr>
<tr>
<td>STT</td>
<td>TEST</td>
<td></td>
</tr>
<tr>
<td>JPL</td>
<td>I HP17</td>
<td></td>
</tr>
<tr>
<td>LDF</td>
<td>LOAD</td>
<td></td>
</tr>
<tr>
<td>SUB</td>
<td>TEST</td>
<td></td>
</tr>
<tr>
<td>JAS</td>
<td>* 4</td>
<td></td>
</tr>
<tr>
<td>JMP</td>
<td>* 1</td>
<td></td>
</tr>
<tr>
<td>SFA</td>
<td>010</td>
<td></td>
</tr>
<tr>
<td>MST</td>
<td>PID</td>
<td>% INTERRUPT TO LEVEL 3</td>
</tr>
<tr>
<td>PAGE</td>
<td>SAA</td>
<td>012</td>
</tr>
<tr>
<td>ADD</td>
<td>C41 1-K. B</td>
<td></td>
</tr>
<tr>
<td>JMP</td>
<td>* 012</td>
<td></td>
</tr>
<tr>
<td>SAA</td>
<td>012</td>
<td></td>
</tr>
<tr>
<td>JPL</td>
<td>I SIO-K. B</td>
<td></td>
</tr>
<tr>
<td>DEVTT</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MIN</td>
<td>C41 1-K. B</td>
<td></td>
</tr>
<tr>
<td>JMP</td>
<td>* 4</td>
<td></td>
</tr>
<tr>
<td>JPL</td>
<td>I DAO</td>
<td></td>
</tr>
<tr>
<td>SAA</td>
<td>* 0100</td>
<td></td>
</tr>
<tr>
<td>STA</td>
<td>C41 1-K. B</td>
<td></td>
</tr>
<tr>
<td>STT</td>
<td>TEST</td>
<td>1</td>
</tr>
<tr>
<td>LCC</td>
<td>TEST</td>
<td></td>
</tr>
<tr>
<td>STT</td>
<td>TEST</td>
<td>1</td>
</tr>
<tr>
<td>EXO</td>
<td>END</td>
<td>DA</td>
</tr>
<tr>
<td>JMZ</td>
<td>* 5</td>
<td></td>
</tr>
<tr>
<td>LDX</td>
<td>TEST</td>
<td></td>
</tr>
<tr>
<td>JPL</td>
<td>I TXYTH</td>
<td></td>
</tr>
<tr>
<td>MIN</td>
<td>C41 1-K. B</td>
<td></td>
</tr>
<tr>
<td>MIN</td>
<td>C41 1-K. B</td>
<td></td>
</tr>
<tr>
<td>JPL</td>
<td>I TTT5</td>
<td></td>
</tr>
<tr>
<td>MIN</td>
<td>C41 1-K. B</td>
<td></td>
</tr>
<tr>
<td>JMP</td>
<td>LEV4</td>
<td></td>
</tr>
<tr>
<td>LOFDR</td>
<td>ADRL0</td>
<td>% POINTER TO HEADING, LOG PRINT OUT</td>
</tr>
<tr>
<td>LBLG</td>
<td></td>
<td>% POINTER TO ADDRESS BLOCK FOR DATA</td>
</tr>
<tr>
<td>LBLG</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TIDR</td>
<td>ADRT1</td>
<td>% POINTER TO HEADING, HOUR PRINT OUT</td>
</tr>
<tr>
<td>TIDC</td>
<td></td>
<td>% POINTER TO ADDRESS BLOCK FOR DATA</td>
</tr>
<tr>
<td>TIDG</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TEST</td>
<td>0</td>
<td>% DECISION OF OUTPUT DEVICE FOR HEADING</td>
</tr>
<tr>
<td>A</td>
<td></td>
<td></td>
</tr>
<tr>
<td>HP17-HP17</td>
<td></td>
<td>% POINTER TO OUTPUT, PUNCH</td>
</tr>
<tr>
<td>ASENC</td>
<td></td>
<td>% POINTER TO ROUTINE FOR OUTPUT OF ECHO ABUNDANCE</td>
</tr>
<tr>
<td>TXYTH, TXYTHD</td>
<td></td>
<td>% POINTER TO ROUTINE FOR OUTPUT OF ECHO ABUNDANCE</td>
</tr>
<tr>
<td>TTT5, TTT5</td>
<td></td>
<td>% POINTER TO OUTPUT OF DATA, TELETYP</td>
</tr>
<tr>
<td>DEV3, DEVF</td>
<td></td>
<td>% DEVICE NUMBER</td>
</tr>
<tr>
<td>Devices</td>
<td></td>
<td>% DEVICE NUMBER</td>
</tr>
<tr>
<td>Devices</td>
<td></td>
<td>% DEVICE NUMBER</td>
</tr>
<tr>
<td>Devices</td>
<td></td>
<td>% POINTER TO OUTPUT RATE, ECHO ABUNDANCE</td>
</tr>
</tbody>
</table>
BSET ZRO 040 DA
BSET ZRO 040 DD
STD DVS-K,E
WAIT % EXIT
JMP LOADER-1

HEADINGS FOR HOUR AND LOG PRINT OUT.

TXTHD, COPY SL DA
STA C41-K,B
JPL I CLF-K,B
DEVTT
LDT ,X
JPL I TXT-K,B
DEVTT
COPY AD1 SX DT
LDX 1,X
JXZ +7
DEVTT
JJC e-2
COPY AD1 ST DX
JMP TXTHD 1
JPL I CLF-K,B
DEVTT
JMP I C41-K,B

ADRLO.TXX1 % ADDRESS BLOCK LOG HEADING
-3
TXX2
-6
TXX3
-3
TXX4
-3
TXX5
-2
TXX6
-1
TXX7
-1
TXX8
-1
TXX9
-1
TXX10
0

ADRHI.TXX1 % ADDRESS BLOCK HOUR HEADING
-3
TXX2
-6
TXX3
-3
TXX4
-2
TXX5
-2
TXX11
-2
TXX12
-1
TXX13
-2
TXX14
-1
TXX15
0
TXX16

% HOUR AND LOG OUTPUT ON PAPER TAPE PUNCH.

MP17, COPY SL DA
STA C41-K,B
LDA 3KL
JPL I RIO-K,B
DEVHF
JPL I TIME-K,B
DEVHF
POS, LDX DP022
LDA TI-K,B
B8HF DNE 010 DA
LDX ER022
STA .-2
STA C41.2-K,B
LDA .X
JPL I RIO-K,B
DEVHF
LDH 1,X
JPL I DOTTE-K,B
DEVHF
-3
-3
SAA 040
JPL I RIO-K,B
DEVHF
AAX
MIN C41 2-K, 2
JMP POS &

LLD00: LDX 1, C42-K :B
MIN C42-K :B
JXZ #015
LDA .X
JPL I RIO-K :B
DEV:HP
LDF 1 :X
JPL I DORTE-K :B
DEV:HP
4
1
SAA 040
JPL I RIO-K :B
DEV:HP
JMP LLD00
LDX 1, C42-K :B
MIN C42-K :B
JXZ #015
LDA .X
JPL I RIO-K :B
DEV:HP
LDF 1 :X
JPL I DORTE-K :B
DEV:HP
5
0
SAA 040
JPL I RIO-K :B
DEV:HP
JMP #016
JPF I CLF-K :B
DEV:HP
JMP I C41-K :B

DPO22, DPO21 % POINTER TO DECCA POSITION
DPO202, DPO201 % POINTER TO DEAD RECKON POSITION
DPL 043461 % DATA IDENTIFICATION: TIME

% HOUR AND LOG PRINTOUT ON TELETYPES:

IT5, COPY 5L DA
SHT C41-K :B

JPL I TIME-K :B
DEV:PP
TPOS, LDX DPO22
LDA TI-K :B
ENP ONE 010 DA
LDX DPO202
AXZ I
JPL I KONV-K :B
DEV:PP
5
1
AXZ 7
LDA D8-K :B
JPL I RIO-K :B
DEV:PP
JPL I KONV-K :B
DEV:PP
5
1
SAA 040
JPL I RIO-K :B
DEV:PP
TLD00: LDX 1, C42 1-K :B
MIN C42 1-K :B
JXZ #07
LDF 1 :X
JPL I DORTE-K :B
DEV:PP
3
1
JMP TLD00
LDX 1, C42 1-K :B
JXZ #010
LDF 1 :X
JPL I DORTE-K :B
DEV:PP
4
0
MIN C42 1-K :B
JMP #010
JPF I CLF-K :B
DEV:PP
JMP I C41-K :B

% POINTERS TO DATACLIPLODS: HOUR PRINTOUT
TDLG, LOG
% NAVIGATIONAL LOG
APRES
% AIR PRESSURE
APTEM
% AIR TEMPERATURE
ATEMW
% DEW POINT
LUX
% RADIATION
O
% WIND FORCE
UND
% WIND DIRECTION
O
% POINTERS TO DATABLOCKS, LOG PRINTOUT

LDLG-LOG
% NAVIGATIONAL LOG
VELOC
% SHIP'S SPEED
STEMP
% SEA SURFACE TEMPERATURE
SSALI
% SEA SURFACE SALINITY
TRAN
% TRANSPARENCY
O
STMHP
% ECHO DEPTH
COURS
% SHIP'S COURSE
O

% ROUTINE FOR OUTPUT OF DATE ON TELEYTYPE AND PAPER TAPE

DATUM, COPY SL DA
STA C41-K, B
LDA DDT
JPL I MIN-K, B
DEVHP
JPL I DATE-K, B
DEVHP
JPL I CLF-K, B
DEVHP
LET (TXTX) 17
JPL I TXT-K, B
DEVTT
JPL I DATE-K, B
DEVTT
JMP I C41-K, B

DDT, 043642
% DATA IDENTIFICATION, DATE
O
ECMD1
% POINTER TO DATA BLOCK, ECHO ABUNDANCE

% ROUTINE FOR OUTPUT OF ECHO ABUNDANCE, PAPER TAPE

ADVC, COPY SL DA
STA C41-K, B
SAK -7
S'T DDT 1
LBY DDT 2
LBA X
JPL I RIO-K, B
DEVHP
LDF 1, X
JPL I DORTE-K, B
DEVHP
O
SAA 040
JPL I RIO-K, B
DEVHP
MAX 015
MIN DDT 1
JMP CLM6G 5
JPL I CLF-K, B
DEVHP
JMP I C41-K, B
FILL

% LISTING OF ECHO ABUNDANCE DATA
% LISTING OF STD DATA
% LISTING OF BOTTOM  BACK SCATTERING

LEV, LDA T5-K, B
SHIP ZRO O15 DA
JMP T55
% LISTING OF STD DATA
SHIP ZRO O10 DA
JMP ECOI
% LISTING OF ECHO ABUNDANCE DATA
LDA 1 (LOCO
AAA 2
LJI DISCO
% LISTING OF BOTTOM BACK SCATTERING
WAIT
% EXIT
JMP LEVS
ECHMD.BSET ZRO 010 DA % ECHO ABUNDANCE OUTPUT
JPL $3
LDF L3 3
DINZ .620
COPY CM2 EA DA
STA L3 2
MIN L3 6
JMP +2
JPL FD11
JPL I DORTE-K.B
L3
LDF 1 (LOG 1
JPL I DORTE-K.B
L3
5
0
LDX (ECHMD 4
LDF .2
STZ .1
STZ 1.X
STZ 2.X
FDV L3 2
JPL I DORTE-K.B
L3
5
0
AAX 015
MIN L3 1
JMP +13
SAA +7
STA L3 1
JPL I DORTE-K.B
L3
LDA DVII-K.B
BSET ZRO 050 DA
STA DVII-K.B
JMP LEVS 5
ESC0.LDA T5-K.B % BOTTOM BACK SCATTERING OUTPUT
JPL $3
SAX +S6
LDF 1.X (ESC0 066
JPL I DORTE-K.B
L3
0
0
AAX 2
JNC +6
JMP +020
TDS3, JPL $2
LAX TDS3
LDA C13I 2-K.B
STA TDS3 1
STZ +7
LDF .X
JPL I DORTE-K.B
L3
0
SAA 2
STA -2
AAX 2
MIN TDS3 1
JMP +011
JPL I DORTE-K.B
L3
LDA DVII-K.B
BSET ZRO 050 DA
STA DVII-K.B
JMP LEVS 010
L3
7
-1
0
PD11, COPY SL DA
STA L3 7
SAX -012
SAA 012
JPL I RIO-K.B
L3
JNC +9
JPL I DATE-K.B
L3
JPL I DORTE-K.B
SAA 012
JPL I RIO-K.B
L3
SAA -70
STA L3 6
JMP I L3 7
L3
LDSUB, BSET ZRO 0160 DA % WAIT IF TELETYP 65
STA T5-K.B
LDA DVII-K.B
%

% DEVICE NUMBER
% PARAMETER COUNTER FOR OUTPUT
% PRINTOUT RATE (NAUTICAL MILES)
% NUMBER OF NAUTICAL MILES BETWEEN OUTPUTS
% LINE COUNTER, PAGING
% TEMPORARY STORAGE
% ROUTINE FOR PAGING

% STR OUTPUT
BSHF J00 000 DA
JMP = -2
SET ONE 000 DA
STA DV1+K:K
EXIT

TD54.TD51 -030  % POINTER TO DATABLOCK, STD
0  % PARAMETER COUNTER FOR OUTPUT

;FILL

;KILL LEV1 ECMID 0000 TD54 L3 PD11 L3SUB TD54 LOGO ECM1 DSCR

% LEVEL 2. THIS LEVEL NOT USED

LEV2.WAIT
JMP = -1

% LEVEL 1

OFF.SAA -1  % EXIT FROM MAC (ASSEMBLER)

STA LEVI 1
WAIT
LEV1.SMIL
0

% LEVEL 0
% WAITING MODE

LEV0.LDA LEVI 1
J01 = 3
STI LEVI 1
IOT FIN 2
JMP = -4

;KILL LEVF1; LEVI 1FVO
;LINE
Beskrivelse av en regneprosedyre til å konvertere DECCA-avlesninger til geografiske koordinater.

Av

Jan Caspersen
Sentralinstitutt for industriell forskning.

INTRODUKSJON

DECCA Navigasjonssystem er nå fullt utbygget langs Norges kyst med i alt seks kjeder. Denne rapport presenterer rutiner for konvertering fra Decca-koordinater til geografiske koordinater. Bare selve verktøyet (lingingene) er tatt med, uten de teoretiske utledninger.
PARAMETERLISTE

\[ b = \text{basislengde fra master til slave (redusert).} \]
\[ r = \text{avstand fra master til punktet } P. \]
\[ r_r = \text{avstand fra master til punktet } P \text{ (redusert).} \]
\[ \theta = \text{vinkelen fra rett nord til linjen gjennom } P. \]
\[ A = \text{vinkelen fra rett nord til linjen gjennom slaven.} \]
\[ T = \text{Legendres vinkelkorreksjon.} \]
\[ R = \frac{1}{r} \]
\[ l = \text{linjetallet, index } M \text{ ved master og } S \text{ ved slave.} \]
\[ RLN = \text{riktig linjenummer.} \]
\[ ALS = \text{antall linjer i sonen.} \]
\[ AVK = \text{avlest kodetall.} \]
\[ Z = \text{tall som avløser kodedokstaven.} \]
\[ L_0 = \text{masters geografiske lengde.} \]
\[ \phi_0 = \text{masters geografiske bredde.} \]
\[ L_s = \text{slavens geografiske lengde.} \]
\[ \phi_s = \text{slavens geografiske bredde.} \]
\[ U = \text{redusert geografiske bredde.} \]
\[ E = \frac{c}{a} = 0.996659151, \text{ eksentrisiteten.} \]
\[ \Delta \phi = \text{breddekoordinat i forhold til master.} \]
\[ \Delta L = \text{lengdekoordinat i forhold til master.} \]
\[ a = \text{jordens store akse.} \]
\[ c = \text{jordens lille akse.} \]
\[ a' = \text{jordens radius avhengig av breddegraden, } a' = a \cos \phi. \]
1. GEOMETRIEN

Den bygger på et teorem av Legendre som sier at en sfærisk trekant kan behandles som en plan hvis vinklene reduseres med:

\[ T = \frac{1}{6} \cdot b \cdot r \cdot \sin (\theta - A) \]  

(1)

Dette er \( \frac{1}{3} \) av den sfæriske flate, og T kommer ut i radianer.

Fig. 1.

På fig. 1 er den sfæriske trekanten utspent av M (master), B (slave) og P som er det aktuelle punktet vi ønsker å bestemme.
Ved å bruke cosinus-satsen og hyperbelbetingelsene kan man stille opp følgende ligninger:

$$P_B^2 = M_B^2 + M_P^2 - 2M_B \cdot M_P \cdot \cos(PMB - T_B) \quad (2)$$

Ut fra hyperbelbetingelsen defineres en størrelse $X_B$,

$$X_B = \frac{P_B - P_M}{M_B} \quad (3)$$

som er konstant for vedkommende hyperbel.

Løses nå (2) med hensyn på $\cos(PM_B - T_B)$, fåes

$$\cos(PM_B - T_B) = \frac{M_B^2 + M_P^2 - P_B^2}{2 \cdot M_B \cdot M_P} \quad (4)$$

Av ligning (3) finnes $P_B$:

$$P_B = X_B \cdot M_B + P_M$$

$$P_B^2 = X_B^2 \cdot M_B^2 + M_P^2 + 2X_B \cdot M_B \cdot P_M$$

Dette settes inn i (4), og det hele reduseres til:

$$\cos(\theta - A_B - T_B) = \frac{M_B \cdot \frac{1 - X_B^2}{2}}{M_P} - X_B \quad (5)$$

Vi definerer to nye størrelser for å få ligningen litt penere:

$$y_B = \frac{M_B}{2} \cdot (1 - X_B^2)$$

$$R = \frac{1}{M_P}$$

Innsatt i (5) blir resultatet:

$$\cos(\theta - A_B - T_B) = y_B \cdot R - X_B \quad (6)$$
Tilsvarende ligninger fåes mot de andre slavene i kjeden, så ligningssettet blir følgende:

\[
\begin{align*}
\cos(\theta - A_B - T_B) &= y_B \cdot R - X_B \\
\cos(\theta - A_C - T_C) &= y_C \cdot R - X_C \\
\cos(\theta - A_D - T_D) &= y_D \cdot R - X_D
\end{align*}
\] (7)

Disse ligningene er transcendente, og ligningssettet over- bestemmer \( \theta \) og \( R \). Regneprosedyren baseres bare på to av dem.

2. BESTEMMELSE AV \( X \)

\( X \) er tidligere definert (lign.(3)) ut fra hyperbelbetingelsene slik at \( X \) er konstant for hver hyperbelgren. Fig.2 viser en slik hyperbelgren som går gjennom \( P \).

Alle punkter på hyperbelen gjennom \( P \) har samme \( X \) og følgelig samme \( Y \). Videre må \( X \) forandre seg lineært mellom master og slave. Det kan skrives:

\[
X = p \cdot l + q
\] (8)

der \( l = \) linjetallet for hyperbelen.
Tenker en seg hyperbelen gjennom $M$ og slik at $P$ ligger i $M$, blir $PM = 0$ og $PB = MB$ og følgelig $X_M = +1$. Legges $P$ i $B$, finner vi $X_B = -1$. Av (8) kan det dermed settes opp to ligninger for å bestemme $p$ og $q$:

I master: $+1 = p \cdot l_M + q$ \hspace{2cm} (9)

I slave: $-1 = p \cdot l_S + q$

Dette fører til: $p = \frac{2}{l_M - l_S}$ \hspace{2cm} (10)

$$q = 1 - \frac{2}{l_M - l_S}$$

3. BESTEMMELSE AV LINJETALLET 1

I en DECCA-avlesning er det en koding med en bokstav som angir sonenummer, og et tall som viser hvilken slave det leses mot og linjetallet i den bestemte sonen.

Sonene starter med master og vanligvis med bokstaven A. Det kan visstnok også begynne med andre bokstaver (Dansk kjede ?). Vanligvis er det 10 soner mellom master og slave, men det kan også være flere, og da begynnes det igjen på bokstaven A.

Masters linjetall mot de forskjellige slaver er som følger:

- Master mot slave B: $l_M = 0$
- " " " C: $l_M = 300$
- " " " D: $l_M = 500$

For å finne riktig linjenummer må følgende gjøres:

$$RLN = Z \cdot ALS_s + AVK - \frac{9}{10} l_M$$

$RLN = $ riktig linjenummer
$ALS = $ antall linjer i sonen (varierer)
$AVK = $ avlest kodetall
$Z = $ tall som avløser bokstaven i koden

92
Eks.: Avlesning B52

B viser at \( Z = 1 \), 52 sier at det leses mot slave D, da er ALS = 30 og \( l_M = 500 \). Dette gir:

\[
RLN = 1 \cdot 30 + 52 - \frac{500}{10} + 500 = 532
\]

Linjetallet i sonen varierer som følger:

\[
\begin{align*}
ALS_B &= 24 \\
ALS_C &= 18 \\
ALS_D &= 30
\end{align*}
\]

4. KORREKSJONER

Følgende korreksjoner er aktuelle:

a) DECCA-korreksjoner p.g.a. landlinjen

b) Natt- og dagkorreksjoner

Disse korreksjoner er det ikke mulig å fremstille matematisk og det kan derfor bare bli tale om å legge dem inn i maskinen i tabellform, som i sin tur vil bli uhyre plasskrevende.

5. KONVERTERING AV VINKELEN \( \theta \) OG AVSTANDEN \( r \) TIL GEOGRAFISKE KOORDINATER

Denne konverteringen foregår etter følgende formler:

\[
\Delta \varphi = F \cdot r_r (\cos \theta + f_1 \cdot r_r + f_2 \cdot r_r^2 + \ldots) \quad (11)
\]

\[
\Delta L = G \cdot r_r \cdot \sin \theta (1 + g_1 r_r + g_2 r_r^2 + \ldots) \quad (12)
\]

idet høyere ordens ledd sløyfes.

\[
\begin{align*}
& r_r = \text{redusert avstand): } \frac{r}{a}, \quad a = \text{jordradien} \\
& F = 3437,75 \frac{\sin^2(\phi_0) \cos(\phi_0)}{\sin^2(U_0) \cdot \cos(U_0)} \quad (13)
\end{align*}
\]
\[ f_1 = -\frac{1}{2} \tan(\phi_0) \left[ 3(1 - E^2) \cos^2(\phi_0) + \sin^2 \theta \right] \left[ 1 - \frac{1}{2} (1 - E^2)(1 + 5 \cos^2(\phi_0)) \right] \] (13)

\[ f_2 = -\frac{1}{6} \sin^2 \theta \cdot \cos \theta \cdot (1 + 3 \tan^2(\phi_0)) \] (13)

\[ G = 3437,75 \frac{1}{\cos U_0} \]

\[ g_1 = \frac{\sin(\phi_0)}{\cos U_0} \cos \theta \] (14)

\[ g_2 = \frac{(3 - 2 \cos^2(\phi_0)) - (4 - 3 \cos^2(\phi_0)) \cdot \sin \theta}{3 \cos^2 U_0} \]

De nye koordinatene er da gitt av:

\[ \phi = \phi_0 + \Delta \phi \] (Bredde) (15)

\[ L = L_0 + \Delta L \] (Lengde)
OPERATION OF THE SYSTEM

This appendix describes procedures for operation of the different application programs in the system.
ROUTINE FOR UPDATING OF HOUR AND DATE

This routine keeps record of the time (GMT) and the date. When starting the system the operator must give initial data for month, day in the month and year, and further for the time in hours, minutes and seconds. The seconds will be counted from the final press on the tabulator of the teleprinter when the routine is started. The following procedure must be followed (characters written by operator are underlined):

PRINTING

<table>
<thead>
<tr>
<th>&quot;TAB&quot;</th>
<th>YES?</th>
<th>COMMENTS</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Call to computer and response.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Code for the computer to identify this routine.</td>
</tr>
<tr>
<td>DATE</td>
<td>19&quot;TAB&quot; 2&quot;TAB&quot; 71&quot;TAB&quot;</td>
<td>Date (19 February 1971).</td>
</tr>
<tr>
<td>TIME</td>
<td>10&quot;TAB&quot; 55&quot;TAB&quot; 0&quot;TAB&quot;</td>
<td>TIME GMT e.g. 10.55.00 hours.</td>
</tr>
<tr>
<td>OK</td>
<td></td>
<td>Final response, program is started.</td>
</tr>
</tbody>
</table>

The time can be printed out any time a code is printed to ask for it:

PRINTING

<table>
<thead>
<tr>
<th>&quot;TAB&quot;</th>
<th>YES?</th>
<th>COMMENTS</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Call to computer and response.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Code for computer to identify routine.</td>
</tr>
<tr>
<td>TIME:</td>
<td>10 55</td>
<td>Time in hours and minutes is printed out.</td>
</tr>
</tbody>
</table>
ROUTINE FOR PRINTOUT OF ECHO ABUNDANCE

This is the routine which prints out values of echo abundance associated with echo integrator deflection. On start of the routine, constants indicating integrator gain in each of the six integrator channels must be printed into the computer via operators teleprinter. Each constant should be the same as the setting on the gain selector for the associated integrator channel (0, 10, 20, 30 or 40).

The operator must also give a constant to indicate the outsailed distance between outputs. If printout is wanted for instance on log 15, 20, 25 etc., the program must be started when the log is showing 10, and the constant should be 5.

Starting procedure is as follows (characters printed by operator are underlined):

**PRINTING**

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;TAB&quot;</td>
<td>YES?</td>
<td>Call to computer and response.</td>
</tr>
<tr>
<td>BA</td>
<td></td>
<td>Code for the computer to identify this routine.</td>
</tr>
<tr>
<td>CH NO 1</td>
<td>GAIN 10 &quot;TAB&quot;</td>
<td>Gain as set on integrator channel 1.</td>
</tr>
<tr>
<td>CH NO 2</td>
<td>GAIN 0 &quot;TAB&quot;</td>
<td>Gain as set on integrator channel 2.</td>
</tr>
<tr>
<td>CH NO 3</td>
<td>GAIN 10 &quot;TAB&quot;</td>
<td>Gain as set on integrator channel 3.</td>
</tr>
<tr>
<td>CH NO 4</td>
<td>GAIN 20 &quot;TAB&quot;</td>
<td>Gain as set on integrator channel 4.</td>
</tr>
<tr>
<td>CH NO 5</td>
<td>GAIN 30 &quot;TAB&quot;</td>
<td>Gain as set on integrator channel 5.</td>
</tr>
<tr>
<td>CH NO 6</td>
<td>GAIN 40 &quot;TAB&quot;</td>
<td>Gain as set on integrator channel 6.</td>
</tr>
<tr>
<td>LOG</td>
<td>5 &quot;TAB&quot;</td>
<td>Five n. miles between outputs.</td>
</tr>
<tr>
<td>OK</td>
<td></td>
<td>Final response, routine started.</td>
</tr>
</tbody>
</table>

There is also a procedure to stop this routine:

**PRINTING**

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;TAB&quot;</td>
<td>YES?</td>
<td>Call to computer and response.</td>
</tr>
<tr>
<td>CA</td>
<td></td>
<td>Identification code for computer.</td>
</tr>
<tr>
<td>OK</td>
<td></td>
<td>Final response before routine is stopped.</td>
</tr>
</tbody>
</table>
This is the routine for sampling of sea surface temperature and salinity from the thermo-salinograph. When starting the system the operator must indicate the setting of the temperature and salinity range selectors on the Thermo-salinograph. For this purpose the ranges are numbered from left to right consecutively from 1 to 5. The following procedure should be used for starting, and it should also be repeated if the setting of the range selectors is changed (characters printed by operator are underlined):

PRINTING

"TAB" YES?  
BB

TRANGE 3 "TAB"

SRANGE 2 "TAB"

OK

Call to computer and response.

Code for the computer to identify routine.

Constant according to range selector, temperature.

Constant according to range selector, salinity.

Final response, routine started.

There is also a procedure to ask for sea surface temperature and salinity.

PRINTING

"TAB" YES?  
AH

Example of printout below:

SEA SURFACE TEMPERATURE: 11.2 SALINITY: 35.1
ROUTINE GIVING DEAD RECKONING POSITION

When starting the system the operator must apply the procedure below to give initial values of the position. The same procedure can also be used if the position for some reason has to be corrected. For latitude south and longitude west the degrees as well as the minutes must have a leading minus.

PRINTING

<table>
<thead>
<tr>
<th>&quot;TAB&quot;</th>
<th>YES?</th>
</tr>
</thead>
<tbody>
<tr>
<td>BD</td>
<td></td>
</tr>
<tr>
<td>LAT</td>
<td>63&quot;TAB&quot; 25.5&quot;TAB&quot;</td>
</tr>
<tr>
<td>LONG</td>
<td>-5&quot;TAB&quot; -19.1&quot;TAB&quot;</td>
</tr>
<tr>
<td>OK</td>
<td></td>
</tr>
</tbody>
</table>

COMMENTS

Call to computer and response.
Code for the computer to identify routine.

Latitude 63° 25.5'N
Longitude 05°19.1'W
Final response, routine started.

The position may also be printed out any time a code is printed to ask for it:

PRINTING

<table>
<thead>
<tr>
<th>&quot;TAB&quot;</th>
<th>YES?</th>
</tr>
</thead>
<tbody>
<tr>
<td>AA</td>
<td></td>
</tr>
<tr>
<td>LAT 63 30.5 LONG 5 18</td>
<td></td>
</tr>
</tbody>
</table>

COMMENTS

Call to computer and response.
Code to identify the routine.

Position 63°30.5'N 5°18'E.
ROUTINE FOR THE SHIP'S LOG

When the system is started, initial values for the ship's log must be given by the operator so that the reading from the log meters are in agreement with the outputs from the computer. The following procedure for initiating the log routine can be repeated to give corrections if necessary. (Characters printed by operator are underlined)

PRINTING

"TAB" YES?
BE
LOG 716.4 "TAB"
OK

There is also a procedure to ask for the log reading:

PRINTING

"TAB" YES?
AE
LOG 718.3

COMMENTS

Call to computer and response.
Code for the computer to identify routine.
Log value entered to the first decimal place.
Final response, routine started.

Printout of log reading.
ROUTINE TO ENTER CALIBRATION DATA FOR STD-SYSTEM

Calibration constants for the STD-system must be entered when the data logging system is started. These data are printed on the listing for every station. The following procedure is used. (Characters printed by operator are underlined):

PRINTING:

"TAB"  YES?
BJ
CORRECTION FOR DEPTH  7
TEMPERATURE  0.02
SALINITY  -0.03
OK

COMMENTS:

Call to computer and response.
Code to identify procedure in computer.
Depth correction in metres.
Temperature correction in °C.
Salinity correction in %.
Final response.
ROUTINE FOR THE STD-SYSTEM ON STD-STATIONS

This routine must be started every time an STD station shall be worked and stopped when the station is completed. The operator must enter station number and amount of parameters to be printed out. Depth, temperature, salinity, sigma-t, D-alfa and delta-D are printed out from left to right. If 6 is entered after the legend "OBSERVATIONS" in the procedure below, a printout is produced every 6th second and all six parameters are printed. If 3 is entered only the 3 parameters from the left are printed and a printout is produced every third second. (Characters printed by operator are underlined):

PRINTING: COMMENTS:

"TAB" YES? Call to computer and response.
BI Code to identify routine in computer.
ST NO 186 Station number.
OBSERVATIONS 6 Print frequency and number of parameters to be included.

The routine is now started and headings are printed out.

When stopping the routine the following procedure should be used:

PRINTING: COMMENTS:

"TAB" YES? Call to computer and response.
CD Code to identify stop in computer.
OK Final response, routine stopped.
ROUTINE FOR ECHO CLASSIFICATION

Some of the software sequences in the echo classification routine are included in the routine giving echo strength distribution. Also the starting procedure is common for the two routines and the code with the legend "PROGRAM" indicates which program to be started. The complete start procedure for echo classification is entered below (characters printed by operator are underlined):

PRINTING

<table>
<thead>
<tr>
<th></th>
<th>COMMENTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>TAB</td>
<td>YES?</td>
</tr>
<tr>
<td>GAIN</td>
<td>-82&quot;TAB&quot;</td>
</tr>
<tr>
<td>RATE</td>
<td>48&quot;TAB&quot;</td>
</tr>
<tr>
<td>TOTAL</td>
<td>100&quot;TAB&quot;</td>
</tr>
<tr>
<td>FRAME</td>
<td>4&quot;TAB&quot;</td>
</tr>
<tr>
<td>PROGRAM</td>
<td>0&quot;TAB&quot;</td>
</tr>
<tr>
<td>MAX DB</td>
<td>-30&quot;TAB&quot;</td>
</tr>
</tbody>
</table>

Call to computer and response.
Code for computer to identify routine.
Gain depending on setting of echo sounder and interface unit.
Ping rate of echo sounder, 48 or 96.
Number of fishes to be classified.
Depth interval to be sampled around target e.g. 4 m.
Code for choice of option, classification.
Max. dB value in table according to fish size.
Final response, routine started.

The routine stops automatically when the indicated number of fishes is observed and the final table printed out. If it for some reason must be stopped before, the following procedure can be used:

PRINTING

<table>
<thead>
<tr>
<th></th>
<th>COMMENTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>TAB</td>
<td>YES?</td>
</tr>
<tr>
<td>CB</td>
<td></td>
</tr>
<tr>
<td>OK</td>
<td></td>
</tr>
</tbody>
</table>

Call to computer and response.
Code to identify stop routine.
Final response, routine stopped.
ROUTINE FOR ECHO STRENGTH DISTRIBUTION

This routine which gives the distribution of echo strengths observed in a given number of transmissions, can not be operated simultaneously with the echo classification routine. The start procedure entered below has many features in common with the start procedure of the echo classification. (Characters printed by operator are underlined):

**PRINTING**

<table>
<thead>
<tr>
<th>&quot;TAB&quot;</th>
<th>YES?</th>
</tr>
</thead>
<tbody>
<tr>
<td>BG</td>
<td></td>
</tr>
<tr>
<td>GAIN</td>
<td>-82 &quot;TAB&quot;</td>
</tr>
<tr>
<td>RATE</td>
<td>48 &quot;TAB&quot;</td>
</tr>
<tr>
<td>TOTAL</td>
<td>50 &quot;TAB&quot;</td>
</tr>
<tr>
<td>FRAME</td>
<td>0 &quot;TAB&quot;</td>
</tr>
<tr>
<td>PROGRAM</td>
<td>- 1 &quot;TAB&quot;</td>
</tr>
<tr>
<td>MAX DB</td>
<td>0 &quot;TAB&quot;</td>
</tr>
<tr>
<td>OK</td>
<td></td>
</tr>
</tbody>
</table>

**COMMENTS**

"TAB" YES? Call to computer and response.
BG Code for identification of procedure in computer.
GAIN -82 "TAB" Gain depending on setting of echo sounder and interface unit.
RATE 48 "TAB" Ping rate of echo sounder, 48 or 96.
TOTAL 50 "TAB" Number of ping to be processed.
FRAME 0 "TAB" Zero should be entered.
PROGRAM - 1 "TAB" Code for choice of option, echo strength
MAX DB 0 "TAB" Zero should be entered.
OK Final response, program started.

The routine stops automatically after printout of the echo strength distribution.
ROUTINE FOR DECCA NAVIGATION

When the program for DECCA-navigation shall be started it is necessary to enter initial DECCA-coordinates for the current position of the ship. This is also necessary when the DECCA-navigator is switched over from one DECCA-chain to another. In the initiation procedure below is also included code to identify DECCA-chain in use. (Characters printed by operator are underlined)

PRINTING

"TAB" YES?
CH NO 4
RED GREEN PURPLE

"TAB"

9.5 "TAB"
8 "TAB"
44.9 "TAB"
22 "TAB"
LAT 63 03.3 LONG -02 18.8
KEEP = 1, CHANGE = 2

Comments:

Call to computer and response.
Number to indicate DECCA-chain.
Indicates red, green and purple deccometer.
Letter on red deccometer, A=0, 10 or 20. B=1,11 or 21 etc.
Reading on red deccometer.
Letter on green deccometer, I=8.
Reading on green deccometer.
Letter on purple deccometer.

If position is correct, 1 is entered, otherwise 2.

The routine has two error messages POS WR and WRONG ITERATION. If one of these are printed out, the DECCA routine is automatically stopped.

The stop routine is as follows:

PRINTING

"TAB" YES?
CC
OK

Decca

The DECCA-position can be printed out when a code is printed to ask for it:

PRINTING

"TAB" YES?
AB
LAT 63 03.5
LONG -02 18.8

Comments:

Call to computer and response.
Code to identify routine.
DECCA position 63°03.5′N.
02°18.8′W.
NUMBERS OF DECCA CHAINS

When starting the routine for Decca navigation, an identification number of the chain to be applied has to be entered into program. Numbers for some chains are listed below.

<table>
<thead>
<tr>
<th>DECCA CHAIN</th>
<th>NUMBER</th>
</tr>
</thead>
<tbody>
<tr>
<td>Finnmark</td>
<td>0</td>
</tr>
<tr>
<td>Lofoten</td>
<td>1</td>
</tr>
<tr>
<td>Helgeland</td>
<td>2</td>
</tr>
<tr>
<td>Trøndelag</td>
<td>3</td>
</tr>
<tr>
<td>Vestlandet</td>
<td>4</td>
</tr>
<tr>
<td>Skagerak</td>
<td>5</td>
</tr>
<tr>
<td>Danish chain</td>
<td>6</td>
</tr>
<tr>
<td>North Scottish</td>
<td>7</td>
</tr>
</tbody>
</table>
ACCESS TO MAC ASSEMBLER

Access to the assembler is obtained in the same manner as the different application routines are started. (Characters printed by operator are underlined):

PRINTING: COMMENTS:

"TAB" YES? Call to computer and response.
BK Code to identify access to MAC.
OK Final response, MAC has control.

When the use of MAC-assembler is finished, the control is returned to Level 9 by the command ]OFF.
PRINTOUT OF PARAMETERS ON REQUEST

Besides the parameters already mentioned there is possibility to ask for the current values of some parameters.

For instance echo depth is obtained by the following procedure:

<table>
<thead>
<tr>
<th>PRINTING</th>
<th>COMMENTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;TAB&quot; YES?</td>
<td>Call to computer and response.</td>
</tr>
<tr>
<td>AE DEPTH</td>
<td>Code to ask for echo depth.</td>
</tr>
<tr>
<td>289</td>
<td>Echo depth in metres.</td>
</tr>
</tbody>
</table>

Codes for all parameters to be used in the same manner are:

<table>
<thead>
<tr>
<th>CODE</th>
<th>PARAMETER</th>
</tr>
</thead>
<tbody>
<tr>
<td>AA</td>
<td>Dead reckoning position.</td>
</tr>
<tr>
<td>AB</td>
<td>DECCA position.</td>
</tr>
<tr>
<td>AC</td>
<td>Time, GMT.</td>
</tr>
<tr>
<td>AD</td>
<td>Reading of ship's log.</td>
</tr>
<tr>
<td>AE</td>
<td>Echo depth.</td>
</tr>
<tr>
<td>AF</td>
<td>Ship's speed, knots.</td>
</tr>
<tr>
<td>AG</td>
<td>Ship's course, degrees.</td>
</tr>
<tr>
<td>AH</td>
<td>Surface temperature and salinity.</td>
</tr>
<tr>
<td>AI</td>
<td>Meteorological parameters, pressure, temperature, dew point, wind force, wind direction.</td>
</tr>
<tr>
<td>AJ</td>
<td>Numeric content in all ADC channels.</td>
</tr>
</tbody>
</table>