A SAIL Compatible Three Channel Acoustic Navigation Interrogator

by

Stephen P. Liberatore

September 1990

Funding was provided by the Office of Naval Research under Contract Nos. N00014-82-C-0152 and N00014-85-C-0379.

Approved for public release; distribution unlimited.
A SAIL Compatible Three Channel
Acoustic Navigation Interrogator

by

Stephen P. Liberatore

Woods Hole Oceanographic Institution
Woods Hole, Massachusetts 02543

September 1990

Technical Report

Funding was provided by the Office of Naval Research
under Contract Nos. N00014-82-C-0152 and N00014-85-C-0379.

Reproduction in whole or in part is permitted for any purpose of the
United States Government. This report should be cited as Woods Hole Oceanographic Inst.

Approved for public release; distribution unlimited.

Approved for Distribution:

Albert J. Williams 3rd, Chairman
Department of Applied Ocean Physics & Engineering
ABSTRACT

Ocean Acoustic Tomography data are significantly degraded if mooring motion is unknown. An autonomous instrument employing a solid state data logger designed to track and record mooring motion is described.

Navigation is accomplished by simultaneously interrogating each of three bottom mounted transponders positioned in an equilateral triangle around the mooring's anchor at a range approximately equal to the depth of the tracked instrument. The three round-trip travel times thus obtained having a resolution of 125uS and a SNR dependent jitter of less than 1.5mS, define a unique instrument position and are recorded along with the time of day and day of year.

The measurement period, the system clock and the program start time are set via a 20mA SAIL. Since the standby power requirement is negligible compared to the battery capacity, the instrument may be programmed months in advance of the deployment.

System endurance varies with the measurement period, however, typical programs permit navigation for up to 21 months at 12 points per day.

Upon recovery, the navigator data may be downloaded via SAIL directly to the storage medium of a suitable computer.
# Table of Contents

**GENERAL DESCRIPTION** .................................................... Section 1.0

- Introduction ................................................................. 1.1
- System Components ......................................................... 1.2

**SPECIFICATIONS** ............................................................. Section 2.0

- Interrogator ........................................................................ 2.1
- Power .................................................................................. 2.2
- Schedule ............................................................................. 2.3
- Data Format .......................................................................... 2.4
- Transponder .......................................................................... 2.5

**OPERATION** ........................................................................ Section 3.0

- Power On Reset ..................................................................... 3.1
- Connect to SAIL .................................................................... 3.2
- Monitor Current ..................................................................... 3.3
- Address ................................................................................. 3.4
- Entering Commands ............................................................... 3.5
- Correcting Errors ................................................................. 3.6
- PROM Test ........................................................................... 3.7
- RAM Test ............................................................................. 3.8
- Clock Set ............................................................................... 3.9
- Schedule ............................................................................... 3.10
- Verify Schedule ................................................................. 3.11
- Test Pinger .......................................................................... 3.12
- Final Test ............................................................................ 3.13
- Data Recovery ..................................................................... 3.14
THEORY OF OPERATION ............................. Section 4.0

- Acoustic Electronics 4.1
- Power Supply 4.2
- System Control 4.3
- Memory Control 4.4
- 64 K Memory 4.5

MODIFIED BENTHOS ELECTRONICS .................. Section 5.0

- Logic Board 5.1
- Back-Plane 5.2
- Battery Stack 5.3

ACKNOWLEDGEMENTS ................................. Section 6.0

REFERENCES ........................................ Section 7.0

APPENDIX ........................................... Section 8.0

- Deployment History 8.1
- PNAVGR Program 8.2
1.0 GENERAL DESCRIPTION

1.1 Introduction

The requirement to spatially track acoustic transceivers moored as part of an Ocean Acoustic Tomography experiment has led the Woods Hole Oceanographic Institution and Benthos Inc. of Falmouth, Ma, to develop an acoustic mooring navigation system.

The electronics module designed at W.H.O.I. and described in this manual is used with the BENTHOS model (ES) 210-TCSSA acoustic transceiver. Together they form a Mooring Motion Monitoring Module (QUAD M) Interrogator.

This document serves as a system hardware reference manual for the technical, but uninitiated user. It references other hardware manuals where appropriate and provides system-oriented information unavailable elsewhere. A copy of the interrogator control program (PNAVLGR) is included as an addendum to this manual.

1.2 System Components

Tracking is accomplished by measuring round-trip travel time from the interrogator to three transponders. The transponders are moored about three meters above the ocean floor and approximately one water depth away from the mooring anchor.
Figure 1 is a block diagram of a mooring equipped to monitor the motion of an instrument mounted near a sub-surface float. "A", "B", and "C", are bottom-mounted acoustic transponders, either Benthos model 210-TR17A-GF which are recoverable or model XT-6000 which are not. The interrogator is mounted as near as practical to the instrument tracked. The frequencies depicted are those which were originally employed. To remain compatible with as many tomography instruments as possible, the 13.5kHz channel has been retuned to 12.0kHz.

The interrogator pings to all three transponders simultaneously at a predetermined time and at a predetermined rate. The time required to receive a response from each transponder, along with the time of day and date, are stored in CMOS static RAM.

The operating parameters are set via the Serial ASCII Instrumentation Loop (SAIL). Pre-deployment checks and data retrieval are also accomplished over the SAIL. A formal description of the SAIL standard is presented in U.N.O.L.S. Ref. TAC-81-1 Aug. 1981, "Serial ASCII Instrumentation Loop (SAIL)" or IEEE standard 997-1985.
Figure 1: Typical Tomography Mooring
2.0 SPECIFICATIONS

2.1 Interrogator

The transceiver specifications, except the electrical power source and operating life, are as listed in the Benthos operating manual for the (ES)210-TCSSA. These two exceptions are the result of replacing a MICRO tape recorder and its associated control electronics with a solid state memory and a power-switched, microprocessor-based controller. The transceiver configured in this manner will henceforth be referred to as an interrogator.

2.2 Power

Twenty-one 1.5 volt "D" size alkaline cells supply power for the interrogator. The DURACELL B1300-T2, with spot welded solder tabs on both terminals is the preferred cell.

The cells are configured as follows: Two diode-isolated parallel strings, each consisting of 9 cells are wired in series yielding 12 volts, then 3 cells are wired in series with the 12 volt stack to yield 16 volts. The battery thus formed is tapped at 12 volts to power the acoustic receiver and the digital electronics, while the 16 volt tap supplies the pinger's power amplifier.

De-rating for temperature and storage, and assuming an average cell voltage of 1 volt, each cell will yield approximately 10 watt hours. The above stack is therefore rated at 210 watt hours.
Making one measurement per hour, the interrogator requires fewer than 0.0045 watt hours. This yields an operating life in excess of 5 years, which exceeds the nominal self discharge time of an alkaline cell. It is however, recommended that the battery be replaced before each deployment.

2.3 Schedule

A measurement may be made as often as every three minutes, or as seldom as once every 999 minutes. The time-of-day clock must be set to the nearest whole minute. Assuming that the clock's oscillator was adjusted to 32.768kHz with the interrogator at the same temperature encountered while deployed, its time will be accurate to within +/- 5 minutes after 365 days, i.e., the clock will lose or gain about 1 second per day. The start of a measurement sequence may be scheduled on any whole minute of the year. Leap years are not accounted for so the clock will reset to day 1 on day 366 of a leap year. Note: Interrogator S/N 005 has an alternate program allowing it to make measurements as often as every 3 seconds or as seldom as every 999 seconds. This system is typically employed as a recording acoustic range finder for towed instruments.

2.4 Data Format

The 60K RAM (Random Access Memory) allows space for 7648 measurements which, at 12 measurements per day, yields a system endurance in excess of twenty months. After the 7649th
measurement, which will be made but not stored, the system will enter the "idle" mode, and no further measurements will be made.

Each measurement consists of a 16 bit time-of-day word, and three 16 bit two-way travel time words. The time of day is recorded with a resolution of one hour. An LSB of travel time is equal to 250 μs. Measurement data, stored beginning at RAM address 1000H, are ordered as follows:

Time of day, Travel time A, Travel time B, and Travel time C.

The time of day is encoded as follows:

<table>
<thead>
<tr>
<th>BIT #</th>
<th>HD</th>
<th>HD</th>
<th>TD</th>
<th>TD</th>
<th>TD</th>
<th>UD</th>
<th>UD</th>
<th>UD</th>
<th>UD</th>
<th>TH</th>
<th>TH</th>
<th>UH</th>
<th>UH</th>
<th>UH</th>
</tr>
</thead>
<tbody>
<tr>
<td>15</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>8</td>
<td>4</td>
<td>2</td>
<td>1</td>
<td>8</td>
<td>4</td>
<td>2</td>
<td>1</td>
<td>8</td>
</tr>
<tr>
<td>14</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>13</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>12</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>11</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>10</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>9</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>8</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>7</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>6</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Where HD is hundreds of days, TD is tens of days, UD is units of days, TH is tens of hours, and UH is units of hours.

As an example, a time code word of 11D6H would convert to day 047 hour 16 as follows:

```
1 1 D 6
00 0100 0111 01 0110
HD TD UD TH UH
0 4 7 1 6
```

### 2.5 Transponder

The transponder specifications may be found in BENTHOS report 0-210-TR17A-GF or the XT-6000 Technical Manual.
3.0 OPERATION

3.1 Power On / Reset

Following the instructions in Benthos manual 0-210-TCSSA, section 2.1, remove the electronics from the pressure housing. Position the electronics with the back-plane wiring facing away from you and with the transducer on your left. Locate the power switch near the transducer end of the instrument and ensure that it is in the "on" position. Locate the reset pins on the opposite end of the instrument and short them together for at least five seconds. This will reset the digital electronics and start the microprocessor.

3.2 Connect to SAIL

Connect to the SAIL via the banana jacks on the controller electronics card. Insure that the loop is closed and connect a terminal to the SAIL / RS-232 converter. Set the terminal for seven data bits, even parity, 1 stop bit, and 300 baud.

3.3 Monitor Current

Connect a digital voltmeter between test points 1 and 2 which are located on either side of R1 on the System Control card. The meter will read total system current scaled at 100uA/mV.
Once the SAIL loop is closed and a full minute has elapsed, the voltmeter will read between 60 and 80 mV. If less than a minute has elapsed the reading may be between .3 and .6 mV. Wait for the higher reading which indicates that the processor is awake and ready for SAIL control.

**Note:** Most of the interrogators are now equipped with a LED to monitor the switched power. With these instruments there is no need to monitor the voltage across R1. Simply wait for the LED to light before attempting to address the interrogator.

### 3.4 Address

Once the microprocessor has detected the presence of a closed SAIL and applied power to the rest of the system, the instrument may be addressed by typing `#In` where `n` is the interrogator's serial number. A correctly addressed instrument will respond with:

```
#In READY
```

**EXAMPLE**

```
I3 READY <<< Interrogator
```

The ":" in the above example is the system prompt and signifies that the interrogator is awaiting commands. Type an H and the interrogator will print a list of the available commands.
EXAMPLE

: H

INTERROGATOR PROGRAM     Ver. 1.1     Jan. 1985

SYSTEM COMMANDS

!Maaaa dddd     LOAD MEMORY
?M             DISPLAY MEMORY
?Paaaa         RUN PROGRAM
?C             CALCULATE CRC
M              MOVE MEMORY
R              TEST RAM
?S             DISPLAY SCHEDULE
!SCHEDULE       PROGRAM SCHEDULE
!TIME          SET CLOCK
?T             DISPLAY TIME
!LOCK          PROTECT MEMORY
!UNLOCK        UNPROTECT MEMORY
!IDLE          INHIBIT SCHEDULER
!PING          TRANSMIT A 10mS PULSE

3.5 Entering Commands

To initiate a command, simply type it exactly as it is listed in the "HELP" file. An error message will be printed in response to an unrecognized command. Usually this message will be followed by the "prompt", at which time you may try re-entering the command. NOTE: Commands are NOT terminated with a "Carriage
Return", but ALL numeric entries in response to system prompts MUST be terminated with a "Space".

3.6 Correcting Errors

Numeric entries are expected to be a certain number of digits in length. For example, when entering the start hour, a two digit figure is expected; but when entering the measurement interval, a three digit figure is expected. Only the last \( n \) digits typed prior to a "Space" are entered (\( n \) is the number of digits expected). Because of this, typing errors may be corrected by simply typing the correct figure immediately after the error. For example, when entering the measurement interval, if you mistakenly type 20 when what you really wanted was 120, the corrected entry would look like this: 20120. Similarly, an hour entry of 2314234121 would be accepted as hour 21.

3.7 PROM Test

Test the system program memory by typing ?C and answering the questions with 0 over 800, and 800 over 800. Verify the correct response by comparing the calculated CRC with the values recorded on the PROMS, IC 4 and 5.

EXAMPLE

: ?CRC From 000 Over 800 = 994C
: ?CRC From 800 Over 800 = EF9A
:  
3.8 RAM Test

Test the system RAM by typing !UNLOCK. The system will respond with OK. Then type an R. The system will respond by typing a cosmetic "am" and the words "Test From". You answer with 1000, and the system will then type Over, to which you answer FOOO. A RAM test over this much memory requires about one minute and seven seconds. After each successful pass, the system will type a *. Ten such passes would indicate good memory. Reset and address the system as in 3.1 and 3.4 respectively.

EXAMPLE

: !UNLOCK  OK
  : Ram Test From 1000 Over FOOO  OK (Y/N) ? Y

**********

The !UNLOCK command is required since RAM test will overwrite any measurements previously stored. The program will automatically execute the !LOCK command when the RAM test is terminated.

3.9 Clock Set

Set the system clock by typing !TIME DDD HH MM 00 where DDD is the year day, HH is hours and MM is minutes. Since the interrogator clock has a one minute resolution, seconds must always be entered as 00 and the clock must be started on the minute. When real time is equal to the time entered, type an @. This will start the clock. To verify that the correct time was entered and that the clock is running, re-address the instrument (Section 3.4) and after the prompt, type ?T. The interrogator will
respond with the current time plus one minute, wait for the real time to equal the time just printed and, on the mark, printing an @.

EXAMPLE

!TIME 123 21 35 00 @
#In
#In READY
: ?T 123 21:36 00 Z...@

3.10 Schedule

Set the operating schedule by typing !SCHEDULE. The Interrogator will ask you for Start day, hour, minute, and the measurement interval. Terminate all entries with a SPACE. When all parameters have been entered, the interrogator will ask permission before activating the scheduler.

EXAMPLE : !SCHEDULE
Start on day = 115 Hour = 18 Minute = 30
Measurement interval, minutes = 060 OK (Y/N) ? Y

3.11 Verify Schedule

Verify that the schedule has been accepted as entered by typing ?S. The interrogator will respond by typing the current time and schedule in addition to the system status (ARMED, not ARMED, or ACTIVE). If the system is ACTIVE, the number of minutes remaining to the next measurement (in HEX) and the current data
address pointer will also be shown.

EXAMPLE : ?S

At 115 18:10
Start on day = 115 hour = 18 minute = 30
Measurement interval = 060 minutes
Scheduler is ARMED BUT NOT ACTIVE

3.12 Test Pinger

Test the pinger by typing !PING. The interrogator will respond by typing OK (Y/N) ? If you next type a Y you should hear the transmit pulse.

EXAMPLE : !PING OK (Y/N) Y :

3.13 Final Test

Disconnect the SAIL cable and observe the system current immediately drop to some value below 100uA. At the next one minute mark, the current will rise to a level near 7mA and stay at that level for about 70mS. If the interrogator is equipped with a LED, it will dimly flash. These observations indicate that the interrogator is functioning correctly and the instrument may be encased in its pressure housing. Refer to section 2.1 of
BENTHOS manual (ES) 210-TCSSA and, following instructions there, place the electronics within the pressure housing. At this point the interrogator is ready for deployment.

### 3.14 Data Recovery (fast)

When the instrument is recovered, the data which are stored in RAM may be downloaded at a high baud rate directly to the storage medium of a suitably equipped computer. **Be careful not to interrupt power to the system in any way as this WILL result in lost data.** Proceed as follows:

1. Remove the electronics from the pressure housing (3.1)
2. Connect the SAIL to RS-232 converter box. (3.2)
3. Monitor current, and wait for the high reading. (3.3)
4. Replace the jumper plug located on the control card (P3) with the cable from the 5 VOLT BAUD RATE GENERATOR. Set the baud rate generator for 9600 baud. (see Figure 8.)
5. Connect the auxiliary I/O port of the computer to the RS-232 connector on the SAIL to RS-232 converter.
6. Set this port for 9600 baud, seven data bits, one stop bit, and even parity.
g. Using the computer terminal (and the appropriate communications program) address the interrogator. (3.4).

h. Type ?S to verify that the system is still "ACTIVE", that the clock is still running, and to obtain the data address pointer. Subtract 1000H from the current address pointer, and make note of the result.

i. Type !IDLE to inhibit further measurements.

j. Prepare the computer to receive an ASCII data file, and type ?M. The system will respond by printing From. You respond by typing 1000. The system will then print Over, and you respond by typing the result of the calculation done in 3.14 (h.) followed by a carriage return.

The interrogator downloads two measurements per line. A full memory (7648 measurements) requires approximately three minutes to download.
4.0 THEORY OF OPERATION

4.1 Acoustic Electronics

Section 5 of Benthos report 0-210-TCSSA explains the operation of the acoustic electronics.

4.2 Power Supply

Refer to Figure 2, which is a simplified block diagram of the interrogator. The capacitor board, the 5 volt regulator, and the low voltage detector are the only blocks which receive power directly from the battery. The 5 volt regulator supplies power on a continuous basis to two other blocks, the clock, and the 60K CMOS static RAM. All other blocks are powered intermittently.

Refer to Figure 3, which is a schematic drawing of the interrogator power supply. These components are located on the SYSTEM CONTROL PC card. R1 is in series with the 12 volt stack, and is used as a current sense resistor for the entire electronics package. A voltmeter placed across this resistor will display current scaled at 100 uA/mV. The ICL 7663 is a micro-power voltage regulator with over-current sense. The output of this regulator is set to 5.5 volts by adjusting P1. The 2N3643 is a series pass transistor used to supply surge current during the power-up sequence.

The ICL7665 is a micro-power under voltage detector. Its
Figure 2: Interrogator Block Diagram
purpose is to monitor the battery and at a preset voltage inhibit further measurements in order to conserve battery power for data retention. When the battery voltage drops below 6.6 volts, LV NOT goes true (logic 0). This will stop a measurement in progress, and inhibit any further measurements from being initiated. LV NOT will remain true until the input voltage on PI-12 rises above 7.8 volts. The 1.2 volt hysteresis prevents the switch from oscillating between true and false, which could occur due to the difference between the open circuit voltage of the battery and the battery voltage while the system is enabled.

4.3 System Control

Refer to Figure 4. This is a schematic of the interrogator system control. These components are located on the same card as the power supply. 5 volt logic power enters through diode D1. This diode drops approximately .5 volts so that VCC and VDD to all components on this card will equal about 5 volts. If this is not the case, check the adjustment of R24.

IC 5, 6, and 7 provide a once-per-minute pulse. If the rest of the system is already powered, this pulse simply generates an interrupt for the microprocessor (IC9). If the rest of the system was not already powered, the once-per-minute pulse will clock a HIGH to pin 13 of IC 4. This causes pin 4 of IC 2 to go LOW which enables system memory and turns on Q2.

VDD is applied to the remaining unpowered ICs on this card when Q2 is on. IC 11, which was already powered, now has VDD on input pins 3 and 6. VDD is level shifted via this IC to 12 volts and fed through P1 directly to the BENTHOS electronics.
When VDD first goes high, a reset pulse is generated via C4 charging through R8. The reset pulse is applied directly to pin 11 of IC 15 which inhibits this IC and prevents inadvertent I/O operations. The reset pulse is also inverted via IC 2 and 12. The inverted reset (CLR NOT) is level shifted via IC 11 and routed to the BENTHOS transmitter through P1. This signal, along with a slight modification to the BENTHOS electronics, prevents the transmitter from pinging upon power up. CLR NOT is also connected to IC 9 and 10. IC 9 is the microprocessor, and when CLR NOT goes HIGH, program execution begins at address 0000. The software clock is updated once the program has been initialized, and the UART (IC 10) is examined to determine if the SAIL is open or closed. If the loop is found to be open, a test is made to determine if it is time to begin a measurement cycle. If the loop is closed, interrupts are enabled and take over the function of updating the clock. If the loop is open and it is not time to begin a measurement the microprocessor generates a signal which appears on IC 15 pin 2. This signal is then gated to the reset pin of IC 4 via the OR gate composed of IC 2 and 3. Resetting IC 4 causes a HIGH to appear on pin 4 of IC 2 which will disable the memory select circuits and cause Q2 to turn off. The disable signal is inverted by IC 3, and the LOW thus produced is connected to VDD. Since Q2 is no longer conducting, this LOW will cause VDD to drop rapidly.

NOTE: It is important to remember that the microprocessor reacts to a manual reset in exactly the same fashion that it reacts to the once-per-minute tick. For this reason, the interrogator clock, which resides only in software, will be advanced one minute with each manual reset, regardless of how much time has actually elapsed.
IC 14 and 13 divide the 1MHz clock by 250 to produce a 4 kHz square wave which is applied to pin 21 of IC 9. During a measurement sequence, the microprocessor will increment three separate counters on each rising edge of this signal. The action begins immediately after a ping is transmitted, and continues until either all three transponders reply or the counters overflow. The reply detected signals (f1, f2, and f3) from the BENTHOS electronics enter through P1, are level shifted by IC 12, and latched by IC 16. The output of the latch is connected to pins 22, 23, and 24 of the microprocessor; these are three of the flag lines. When the microprocessor detects one of these flags, it stops incrementing the counter associated with that reply channel. The number remaining in the counter represents the two-way travel time. A counter which contains all zeros has overflowed and indicates no reply on that channel.

IC 18 converts the 20 mA SAIL levels to 5 volt CMOS levels for the UART, and provides an output which indicates an open loop. IC 17 divides the TPA clock signal from IC 9 by 26 to provide the 16X clock rate the UART requires to run at 300 baud.

The Q4 output of IC 15 and the DO output of IC 18 synchronize the clock. Once the time has been entered, the microprocessor generates a signal which causes Q4 of IC 15 to go HIGH. This is the SET signal and is applied to the set input of IC 6. Pin 1 of this IC goes HIGH and is gated by the OR gate formed with IC 2 and 3 to the reset inputs of IC 5,6, 7. This stops the clock’s oscillator and resets its down counters. The start bit of any character typed over the loop will be inverted by IC 3 and used to clock IC 6. This will remove the reset and allow the clock’s oscillator and down counters to operate. If the character was not an "@", the microprocessor will again
generate the signal which causes Q4 of IC 15 to go HIGH, and the cycle repeats.

4.4 Memory Control

Refer to Figure 5. This is a schematic of the memory control electronics. These components are located on the 64K memory card.

IC 17 gates the buffered MWR NOT and MRD NOT signals with the DISABLE signal generated on the system control card. This signal will go true just before power is removed from the microprocessor. When disable is true, both XMWR NOT and XMRD NOT are false (logic "1"). XMRD NOT being HIGH holds IC 21 reset. The Q4 output of IC 21 is applied to pin 8 of IC 13; and since pin 9 of this IC is also HIGH, its output, pin 10, is LOW. This is the memory on (or enable memory bus) signal, and when LOW, inhibits all memory operations by de-selecting the memory chips and by turning off the memory bus drivers.

IC 16, 18, and the remaining NAND gates of IC 13 decode the address lines to produce the 8K selects which enable the HM6264 RAM chips on this card. IC 12 decodes the proper address lines to produce the 2K selects which are required by the 27C16 PROM chips, and the HM6116 RAM chips. Since the PROM is power switched, the 2K selects used by these chips are buffered by IC 19. IC 21 is a counter and, with IC 13, is used to truncate the memory cycle and thus conserve power. It is recommended that the jumper from pin 8 of IC 13 to pin 11 of IC 21 be moved to pin 13 of IC 21, thereby increasing the memory enabled time by 1uS. This modification, although not essential and causing a slight increase in power consumption, will improve the system’s reliability.
Figure 5: Interrogator Memory Control (64K) Schematic

[Diagram of memory control circuit with labels and waveforms for CLK, TPA, TP B, MA, A1, A0, XRWR, XMWR, and MEMOR, indicating read and write cycles.]
4.5 64K Memory

Refer to Figure 6. This is a schematic of the system memory. These components are located on the same card as the memory control electronics. A 24 pin ribbon cable connects the memory card to the system control card. The memory is fully buffered by IC 18 and 22 which buffer the data lines and IC 14 and 15 which buffer the address and clock lines. Since IC 4 and 5 are power switched the MRD NOT signal is buffered by IC 19.

Power for this card is supplied via a disconnect through two diodes which isolate the logic power from the memory back-up battery. The back-up battery is composed of three AAA cells wired in series and, if used, is mounted on the rail over the system control card.
5.0 MODIFIED BENTHOS ELECTRONICS

Slight modifications were made to the electronics supplied by BENTHOS. The effects of these modifications are as follows:

a. A six-volt tap from the battery stack is eliminated.

b. Transmitting on every power-up sequence is prevented.

5.1 Logic Board

Refer to BENTHOS drawing B-210-248. This is a schematic for the LOGIC board which must be modified to make provision for a power-up reset pulse. The power-up reset pulse originates on the system control card and inhibits the pinger during the power on cycles which occur at the rate of one per minute. Remove the LOGIC board from the chassis and locate IC 2, a CD4098B. Remove the etch between pins 3, 16, and 13 of IC 2. Connect pin 13 to pin 16 with a short jumper. Connect pin 3 to board I/O pin 10 with another short jumper. Clean the board of flux, and re-coat the patched area with a clear acrylic.

5.2 Back-Plane

Clip the white wire from the pin 5 end of the 10K ohm resistor located on the CAPACITOR card connector between pins 5 and 7. Connect this wire to pin 10 of the LOGIC card connector.
5.3 Battery Stack

Locate the 12 pin female MOLEX connector which exits the battery housing. Remove the orange wire from pin 1 of this connector, and discard it. Remove the red wire from pin 2 and place it in pin 1. Remove the white/red trace wire from pin 7 and place it in pin 2.

Refer to Figure 7. This is a schematic of the modified stack. Using twenty-one B1300-T2 alkaline cells and two 1N4002 diodes, construct such a stack and connect it to the molex connector as illustrated.
Figure 7: Interrogator Battery Pack
Figure 8  Interrogator UART Clock Generator
6.0 ACKNOWLEDGEMENTS

As a general rule, many hands are involved in the development of an oceanographic instrument and the interrogator was no exception. The author wishes to gratefully acknowledge contributions to this endeavor made by the following people and organizations: Benthos, Inc. of N. Falmouth Mass. for their support during the entire program, Scripps Institution of Oceanography at the University of California for funding the publication of this document, Fred Schuler for his many helpful comments and his aid in de-bugging the prototype, Dick Nowak who developed the measurement synchronization algorithm, Bob Spindel for his encouragement, without which the project would not have been undertaken, and finally John Kemp and Paul Boutin for their assistance during the "wet" tests both at Woods Hole and from the deck of the R/V ERLINE.

Funding was provided by the Office of Naval Research under contract Numbers N00014-C-82-0152 and N00014-85-C-0379.
7.0 REFERENCES

1. The RCA COS/MOS Integrated Circuits Manual SSD-250B

2. The RCA CMOS-LSI Circuits Manual SSD-260A

3. RCA ICAN-6581 "Power-on Reset/Run circuits for the RCA CDP1802 COSMAC microprocessor"

4. RCA ICAN-6304 "Power Supplies for COS/MOS"

5. RCA ICAN-6525 "Guide to Better Handling and Operation of CMOS Integrated Circuits:

6. RCA ICAN-6576 "Power-Supply Considerations for COS/MOS Devices"

7. The RCA User Manual for the CDP1802 COSMAC Microprocessor MPM-201B

8. Benthos report 0-210-TR17A-GF, "Instructions for the installation, operation, and maintenance of the model 210-TR17A-GF combination commandable transponder and glow flash"

9. Benthos report 0-210-TCSSA, "Instructions for installation, operation and maintenance of the model (ES) 210-TCSSA acoustic transceiver"

10. The Benthos XT-6000 Technical Manual

11. The Motorola CMOS Data Manual

12. The MAXIM Data Acquisition Catalog
8.0 APPENDIX

8.1 Deployment History

During the past five years, the interrogator has been successfully employed to navigate more than twenty moorings set as part of five major Tomography experiments fielded in the North Atlantic, North Pacific, Gulf of Mexico, the Greenland Sea and the Mediterranean.

Twice during the course of these experiments an interrogator has failed. One system recovered from the RTE-88 experiment failed after three months of operation. Interrogator S/N 008 was recovered from the Greenland Sea in 1989 with a completely depleted battery. On inspection a leaky cell in the battery stack was discovered and may have caused the problem. However, both of these failures might also be attributed to a marginal memory component forcing the program to "hang", which in turn would disable power switching and cause the battery to drain at a 6 to 10mA rate. The modification recommended at the end of section 4.4 should help to eliminate this type of failure.
Figure 9  Interrogator Program Flow Chart
### 8.2 PNAVLR Program

**INTERROGATOR GLOBAL PAGE**

<table>
<thead>
<tr>
<th>LABEL</th>
<th>ADDRESS</th>
<th>FUNCTION</th>
<th>LABEL</th>
<th>ADDRESS</th>
<th>FUNCTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>GLOBAL</td>
<td>FF00</td>
<td>UART STAT. OR CHAR</td>
<td>DSHD</td>
<td>FF30</td>
<td>DEC. STRT.H.D.</td>
</tr>
<tr>
<td>1</td>
<td></td>
<td>SYSTEM ERROR FLAG</td>
<td>1</td>
<td></td>
<td>&quot; &quot; T.D.</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>&quot; &quot; U.D.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>&quot; &quot; T.H.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>&quot; &quot; U.H.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SCRACH</td>
<td>5</td>
<td>USED BY SALTY</td>
<td>5</td>
<td></td>
<td>&quot; &quot; T.M.</td>
</tr>
<tr>
<td>6</td>
<td>&quot; &quot; U.M.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>&quot; &quot; T.D.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>&quot; &quot; U.D.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CRCHI</td>
<td>B</td>
<td>CRC HIGH BYTE</td>
<td>8</td>
<td></td>
<td>ASCII STRT.H.D.</td>
</tr>
<tr>
<td>9</td>
<td>&quot; &quot; T.H.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CRCLO</td>
<td>C</td>
<td>CRC LOW BYTE</td>
<td>9</td>
<td></td>
<td>&quot; &quot; T.M.</td>
</tr>
<tr>
<td>D</td>
<td>F &quot; &quot; U.H.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>STRADD</td>
<td>E</td>
<td>STORE ADDRESS HI</td>
<td>F</td>
<td>&quot; &quot; &quot; STOP CHAR.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>&quot; &quot; &quot; STRT T.M.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>HD</td>
<td>FF10</td>
<td>DEC. H. DAYS</td>
<td>ASHT</td>
<td>FF40</td>
<td>ASCII STRT.U.M.</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>&quot; &quot; T.M.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>&quot; &quot; U.M.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>&quot; &quot; T.M.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>&quot; &quot; U.M.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>5</td>
<td>&quot; &quot; T.M.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>6</td>
<td>&quot; &quot; U.M.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TICK</td>
<td>7</td>
<td>TICK FLAG</td>
<td>ASCII INT.H.M.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>8</td>
<td>&quot; &quot; T.M.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>9</td>
<td>&quot; &quot; U.M.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NXTM</td>
<td>8</td>
<td>ASCII H. DAYS</td>
<td>&quot; &quot; &quot; STOP CHAR.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>A</td>
<td>&quot; &quot; T.D.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>B</td>
<td>&quot; &quot; U.D.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>C</td>
<td>&quot; &quot; T.H.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>D</td>
<td>&quot; &quot; U.H.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>E</td>
<td>&quot; &quot; T.M.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>F</td>
<td>&quot; &quot; U.M.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FF20</td>
<td>ASCII U. MINUTES</td>
<td>FF50</td>
<td>1</td>
<td>&quot; STOP CHARACTER</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>2</td>
<td></td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>5</td>
<td>5</td>
<td>6</td>
<td>6</td>
</tr>
<tr>
<td>7</td>
<td>7</td>
<td>8</td>
<td>8</td>
<td>9</td>
<td>9</td>
</tr>
<tr>
<td>HEXMI</td>
<td>4</td>
<td>HEX MEAS. INT. HI</td>
<td>9</td>
<td>A</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>5</td>
<td>6</td>
<td>6</td>
<td>7</td>
<td>7</td>
</tr>
<tr>
<td>8</td>
<td>8</td>
<td>9</td>
<td>9</td>
<td>A</td>
<td></td>
</tr>
<tr>
<td>MINOW</td>
<td>6</td>
<td>HEX MINS.TO NEXT HI</td>
<td>B</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>7</td>
<td>8</td>
<td>8</td>
<td>C</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>9</td>
<td>10</td>
<td>10</td>
<td>D</td>
<td></td>
</tr>
<tr>
<td>S1HD</td>
<td>9</td>
<td>DEC. TIME + 1 MIN.</td>
<td>E</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>10</td>
<td>11</td>
<td>11</td>
<td>F</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>11</td>
<td>12</td>
<td>12</td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>12</td>
<td>13</td>
<td>13</td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>13</td>
<td>14</td>
<td>14</td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>14</td>
<td>15</td>
<td>15</td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>15</td>
<td>16</td>
<td>16</td>
<td></td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>16</td>
<td>17</td>
<td>17</td>
<td></td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>17</td>
<td>18</td>
<td>18</td>
<td></td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>18</td>
<td>19</td>
<td>19</td>
<td></td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>19</td>
<td>20</td>
<td>20</td>
<td></td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>20</td>
<td>21</td>
<td>21</td>
<td></td>
<td></td>
</tr>
<tr>
<td>21</td>
<td>21</td>
<td>22</td>
<td>22</td>
<td></td>
<td></td>
</tr>
<tr>
<td>22</td>
<td>22</td>
<td>23</td>
<td>23</td>
<td></td>
<td></td>
</tr>
<tr>
<td>23</td>
<td>23</td>
<td>24</td>
<td>24</td>
<td></td>
<td></td>
</tr>
<tr>
<td>24</td>
<td>24</td>
<td>25</td>
<td>25</td>
<td></td>
<td></td>
</tr>
<tr>
<td>25</td>
<td>25</td>
<td>26</td>
<td>26</td>
<td></td>
<td></td>
</tr>
<tr>
<td>26</td>
<td>26</td>
<td>27</td>
<td>27</td>
<td></td>
<td></td>
</tr>
<tr>
<td>27</td>
<td>27</td>
<td>28</td>
<td>28</td>
<td></td>
<td></td>
</tr>
<tr>
<td>28</td>
<td>28</td>
<td>29</td>
<td>29</td>
<td></td>
<td></td>
</tr>
<tr>
<td>29</td>
<td>29</td>
<td>30</td>
<td>30</td>
<td></td>
<td></td>
</tr>
<tr>
<td>30</td>
<td>30</td>
<td>31</td>
<td>31</td>
<td></td>
<td></td>
</tr>
<tr>
<td>31</td>
<td>31</td>
<td>32</td>
<td>32</td>
<td></td>
<td></td>
</tr>
<tr>
<td>32</td>
<td>32</td>
<td>33</td>
<td>33</td>
<td></td>
<td></td>
</tr>
<tr>
<td>33</td>
<td>33</td>
<td>34</td>
<td>34</td>
<td></td>
<td></td>
</tr>
<tr>
<td>34</td>
<td>34</td>
<td>35</td>
<td>35</td>
<td></td>
<td></td>
</tr>
<tr>
<td>35</td>
<td>35</td>
<td>36</td>
<td>36</td>
<td></td>
<td></td>
</tr>
<tr>
<td>36</td>
<td>36</td>
<td>37</td>
<td>37</td>
<td></td>
<td></td>
</tr>
<tr>
<td>37</td>
<td>37</td>
<td>38</td>
<td>38</td>
<td></td>
<td></td>
</tr>
<tr>
<td>38</td>
<td>38</td>
<td>39</td>
<td>39</td>
<td></td>
<td></td>
</tr>
<tr>
<td>39</td>
<td>39</td>
<td>40</td>
<td>40</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Note:** The table above represents a list of labels, addresses, and functions for various parts of the PNAVLR program. Each row corresponds to a specific directive, with the label, address, and function indicated.
TITLE  INTERROGATOR CONTROL/DATA LOGGER (PNVLGR.MAC)
SUBTIL  WOODS HOLE OCEANOGRAPHIC INST. OCEAN ENGINEERING

; Ver. 1.1  27 Feb. 1985     By Steve Liberatore
;
; Copyright (c) 1985 Woods Hole Oceanographic Institution.
; All rights reserved.
;
; A SAIL compatible micro-power Controller / Data Logger for the
; Benthos Quad-M Transceiver. All standard 1902 monitor functions
; are implemented along with extensive self test capabilities.
; The 60k (FO00) RAM memory allows space for 7648 measurements.
; The 7649th measurement will not be stored, and will cause the
; system to idle. A measurement will consist of a 16 bit time
; code word, and three 16 bit two way travel time words. When
; output to disk there will be two measurements per line. An LSB
; of travel time will be equivalent to 250 usec., and time will be
; encoded as follows:
;
; BIT #  15 14 13 12 11 10  9  8  7  6  5  4  3  2  1  0
; UNITS  HD HD TD TD TD UD UD UD TH TH HH HH HH HH HH
; WEIGHT 2 1 8 4 2 1 8 4 2 1 2 1 8 4 2 1
;
; Measurement data will be stored in memory beginning at address
; 1000H and ordered as follows:
; TIME_CODE, TRAVEL TIME A, TRAVEL TIME B, AND TRAVEL TIME C.
;
; To assemble this program using the IBM PC/AT system:
; First execute the 280MU command to enter the CP/M SHELL
;
; 1) Type M18 =PNVLGR.MAC
; 2) After the system prompt type L18.
; 3) After the LINK prompt type /P:0000
; 4) Next type PNVLGR,PNVLGR/N/X/E
; 5) Answer the MOVE question with N
; 6) At the system prompt re-enter DOS by typing E
; 7) Use a word processor to divide PNVLGR.HEX. The
;    two new files will be named PROM1.HEX and PROM2.HEX.
;    The PROM1 file contains addresses 0 thru 817 and the
;    PROM2 file contains addresses 7FC thru FFF.
; 8) Next type MOVEHEX.
; 9) At the system prompt type UDLINT.
; 10) To burn the first PROM type PROM1 and to burn the
;    second PROM type PROM2.
; 11) Return to the system by typing BYE
;
; INCLUDE LIMIT.MAC
; **************
; * LIMIT.MAC *
; **************

; THIS SEGMENT OF CODE WILL INITIALIZE ALL REGISTERS
; At this point, all registers are preset, so execution begins in register R3.
Cette macro execute la routine CALL.

CALL MACRO SUB BEGIN MACRO CALL
.SALL NO LISTING
SEP R4 CALL
DIW SUB SUBROUTINE
ENDM END MACRO CALL

Cette macro execute la routine RETURN.

RETURN MACRO BEGIN MACRO RETURN
.SALL NO LISTING
SEP R5 RETURN
ENDM END MACRO RETURN

Cette macro recherche des erreurs de status UART.

ERROR? MACRO BEGIN MACRO ERROR
.SALL NO LISTING
GET RC RECOVER STATUS WORD
LENZ ERVEC BRANCH ON ERROR FLAG
ENDM

Voici une macro qui, lorsqu'elle est appelée, lance successivement
les caractères d'entrée et les compare avec un caractère
string stocké dans la mémoire permanente. Si les comparaisons
sont échouées, la macro quitte avec un résultat non nul
restant dans l'ACCUMULATEUR.

WORD? MACRO WORD BEGIN MACRO WORD
.SALL NO LISTING
CALL COMPAR CALL SUBROUTINE
DIW WORD PASS WORD
ERROR? REACT TO FLAGS
GET COMPAR RESULT
ENDM END OF MACRO WORD

Voici une macro qui lorsqu'elle est appelée, lit un
ASCII character puis quitte avec ce caractère
restant dans l'ACCUMULATEUR.

CHAR? MACRO BEGIN MACRO CHAR
.SALL NO LISTING
CALL INCHAR CALL SUBROUTINE INCHAR
HERE IS A MACRO WHICH WILL CALL SALITY, PASS THE MESSAGE ADDRESS, AND React TO ERRORS UPON EXITING.

;BEGIN MACRO TYPMSG
.SALL
CALL SALITY
DW MSG
ERROR?
;END MACRO TYPMSG

;BEGIN MACRO GETFLG
.SALL
LDI 01H
PLO R7
LIN R7
ENM
;END MACRO GETFLG

* SCRT.MAC *

**** THESE ARE THE RCA STANDARD CALL AND RETURN ROUTINES. +

;THIS IS THE CALL ROUTINE, IT RUNS IN R4

;R3 IS POINTING AT THE FIRST INSTRUCTION IN THE SUBROUTINE
;THIS IS A "PUBLIC ROUTINE"

SEX R2
GHI R6
STXD
GLO R6
STXD
GHI R3
PHI R6
GLO R3
PLO R6
LDA R6
LDA R6
PLO R3
0047' CO 0039' C LBR EXITC ;RUN THE SUBROUTINE IN R3 C
C ;THIS IS THE RETURN ROUTINE, IT RUNS IN R5 C
C
0048' D3 C EXTR: SEP R3 ;RETURN TO MAIN PROGRAM C
C
0048' B3 C RETURN: ;THIS IS A "PUBLIC ROUTINE" C
0048' G6 C R3 ;COPY R6 INTO R3 C
0048' 86 C R3 ;R3 CONTAINS THE RETURN C
0048' A3 C GLO R6 ;ADDRESS C
0048' E2 C FLO R3 C
0050' 12 C SEX R2 ;POINT TO THE STACK C
0051' 72 C INC R2 ;GET OLD VALUE OF R6 C
0052' A6 C LEXA ;AND RESTORE IT TO R6 C
0053' F0 C FLO R6 C
0054' B6 C LEX C
0055' CO 004A' C LBR EXITR ;RUN MAIN PROGRAM C
C
; INCLUDE ATCH.MAC
**************
C
* ATCH.MAC *
**************
C
; * ASCII TO HEXADECIMAL CONVERTER *
C
C
; This sub-routine converts the ASCII character in the
C ; low half of RC to a HEX digit, shifts this hex digit
C ; four (4) places to the left and returns it in
C ; the low half of RC.
C
C
0058' 8C C ATCH: = GLO RC ;GET THE ASCII CHAR.
0059' FF 30 C SM1 "0" ;TOO SMALL ?
005B' CB 0081' C LENF AERROR ;IF SO GOTO ERROR
005E' BC C PHI RC ;SAVE RESULT
005F' 8C C GLO RC ;RESTORE
0060' FF 47 C SM1 "9" ;TOO LARGE ?
0062' C3 0081' C LEFD AERROR ;IF SO GOTO ERROR
0065' 9C C GHI RC ;CHAR MINUS ASCII BIAS
0066' FF 0A C SM1 OAH ;IS IT 0 THROUGH 9 ?
0068' CB 0077' C LEFD HDONE ;IF SO CONVERT IS DONE
006B' 9C C GHI RC ;RESTORE
006C' FF 11 C SM1 11H ;IS IT ASCII ?
006E' CB 0081' C LENF AERROR ;IF NOT GOTO ERROR
0071' 9C C GHI RC ;RESTORE
0072' FF 07 C SM1 O7H ;REMOVE ALPHA BIAS
0074' CO 0078' C LER SHIFT ;GOTO SHIFT
0077' 9C C HDONE: = GHI RC ;RESTORE
0078' FE C SHIFT: = SHI ;SHIFT LEFT 4 TIMES
0079' FE C SHI
007A' FE C SHI
007B' FE C SHI
007C' AC C FLO RC ;HEX DIGIT TO RC LOW
007F'  BC  C  PHI RC  ;AND NON-HEX FLAG
0080'  D5  C+  RETURN  ;BACK TO MAIN
0081'  F8  01  C  ERROR: LD¥  01H  ;SET NON-HEX FLAG
0083'  BC  C  PHI RC  ;BACK TO MAIN
0084'  D5  C+  ;

;INCLUDE HTOA.MAC
***************
* HTOA.MAC *
***************

+ HEXADECIMAL TO ASCII CONVERTER +

;This sub-routine converts the HEX digit in the
;low half of RC to an ASCII character, and returns
;with this character in the high half of RC.

0085'  RC  C  GLO RC  ;GET THE HEX DIGIT
0086'  FA  0F  C  ANI OFH  ;MASK HIGH BYTE
0088'  FC  30  C  ADI 3OH  ;ADD ASCII BIAS
008A'  BC  C  PHI RC  ;SAVE RESULT
008B'  FF  3A  C  SM# 3AH  ;IS IT NUMERIC ?
008D'  CD  0094'  C  LOCAL ASCONE  ;IF SO CONVERT IS DONE
0090'  9C  C  GHI RC  ;OTHERWISE,
0091'  FC  07  C  ADI 07H  ;ADD ALPHA BIAS
0093'  BC  C  PHI RC  ;SAVE RESULT
0094'  C+  ADONE: RETURN  ;RETURN TO MAIN
0094'  D5  C+  ;

;INCLUDE DTOA.MAC
***************
* DTOA.MAC *
***************

+ ADD ASCII BIAS AND STORE +

;This subroutine will add 30 hex to the byte
;pointed at by R7, store the result using RA
;as a pointer, then increment the pointers.
;This operation will be repeated n times as
;specified by the in-line byte following the
;call instruction.

0095'  46  C  DTOA: LDA R6  ;GET REPEAT VALUE
0096'  AC  C  PLO RC  ;SET COUNTER
0097'  47  C  ADBIAS: LDA R7  ;GET DIGIT
0098' FC 30 C ADI 30H ;ADD ASCII DEC. BIAS
0099' 5A C STR RA ;STORE RESULT
0099' 1A C INC RA ;ADVANCE POINTER
009C' 2C C DEC RC ;COUNT OPERATION
009D' 8C C GLO RC ;TEST COUNTER
009E' CA 0097' C LNZ ADDIAS ;EXIT IF DONE
009D' 8C C RETURN ;OTHERWISE CONTINUE

INCLUDE DELAY.MAC

**************

* DELAY.MAC *

**************

(RC)

+ DELAY PROGRAM EXECUTION +

;This subroutine will delay program execution by an amount
;of time equivalent to (120N +168)/Fc where Fc is the system
;clock frequency and N is a two byte value specified by the
;bytes following the call instruction. SRT is expected.

00A2' 46 C DELAY: LDA R6 ;LOAD DELAY CONSTANT
00A3' BC C PHI RC ;USE RC AS A DOWN COUNTER
00A4' 46 C LDA R6
00A5' AC C PLO RC
00A6' 9C C TSTHIC: GHI RC ;CONTINUE TESTING RC
00A7' CA 00A2' C LNZ WST5 ;WHEN EQUAL TO ZERO
00AA' 8C C GLO RC ;SPECIFIED TIME HAS
00AB' C2 00A6' C LBE EXOLY ;ELAPSED, OTHERWISE
00AE' 2C C DECC: DEC RC ;DEC COUNTER AND TEST FOR
00AF' CO 00A6' C LBR TSTHIC ;NON ZERO VALUE
00B2' 8C C WST5: GLO RC ;EQUALIZE COUNTER LOOPS
00B3' CO 00AE' C LBR DECC ;AND CONTINUE
00B6' D5 C EXOLY: RETURN ;RETURN TO MAIN WHEN COUNT = 0000

INCLUDE RSB2A.MAC

**************

* RSB2A.MAC *

**************

+ RIGHT SHIFT n BITS FROM RB TO RA +

(RA, RB, RC)

;This subroutine will right shift a single bit
;from register B to register A. This operation
;will be repeated a number of times as specified
;by the byte following the call instruction.
INTERROGATOR CONTROL/DATA LOGGER (PNSAVGR.MAC) PAGE 1-7
Woods Hole Oceanographic Inst. OCEAN ENGINEERING

0087' 46 C RSB2A: LDA R6 ; GET REPEATED VALUE
0088' AC C PLO RC ; USE RC AS A COUNTER
0089' 9B C SHR: GHI RB ;RIGHT SHIFT A BIT
008A' F6 C SHR ; FROM R8 TO RA
008B' BB C PHI RB
008C' 8B C GLO RB
008D' 76 C RSHR
008E' AB C PLO RB
008F' 9A C GHI RA
0090' 76 C RSHR
0091' BA C PHI RA
0092' 8A C GLO RA
0093' 76 C RSHR
0094' AA C PLO RA
0095' 2C C DEC RC ; DECREMENT COUNTER
0096' 8C C GLO RC ; TEST FOR ZERO AND
0097' CA 0089' C LNZ SHRB ; IF FOUND
          C RETURN ; RETURN TO MAIN

00CA' D5 C+ ;

INCLUDE SALITY.MAC
***************
* SALITY.MAC *
***************

; TYPES MESSAGE NAMED AFTER CALL +

; (RC)

; This subroutine will type the message indicated
; by the call. The loop will be continually monitored,
; Any error condition will cause this routine to
; exit. The status word and the last character typed
; will be available in RC upon exiting this routine.

; SALITY: ; A PUBLIC ROUTINE
          ; USE R7 AS THE POINTER
00CA' E7 C SEX R7
00CC' F8 04 C LD1 LOW (SCRATCH-1) ; POINT TO A SCRATCH
00CE' A7 C PLO R7 ; LOCATION IN RAM
00CF' 8A C GLO RA ; SAVE OLD ADDRESS
00D0' 73 C STXD ; POINTER
00D1' 9A C GHI RA
00D2' 73 C STXD
00D3' 46 C LDA R6 ; GET HIGH HALF OF
00D4' BA C PHI RA ; MESSAGE ADDRESS
00D5' 46 C LDA R6 ; GET LOW HALF OF
00D6' AA C PLO RA ; MESSAGE ADDRESS

; Enter the subroutine here if the address of the data
; to type is already in register RA.

; TYPE:: SEX R7 ; USE R7 AS THE POINTER
00D7' E7 C LOH 00H ; POINT TO A SCRATCH
00D8' F8 00 C PLO R7 ; LOCATION IN RAM

48
INTERROGATOR CONTROL/DATA LOGGER (PNAVGR.MAC) MACRO-18 3.36
WOODS HOLE OCEANOGRAPHIC INST. OCEAN ENGINEERING

008' 6F C C INP DATA ;CLEAR UART DATA BIT
008C' E7 C C TREE? SEX K7 ;RESET POINTER TO K7
008D' 6F C C INP STATUS ;GET UART STATUS
008E' FA 10 C C ANI 10H ;IS THE LOOP OPEN?
008F' C2 008' C C LEZ TSTR ;IF NOT TEST FOR TREE
009' F3 80 C C LDI 80H ;OTHERWISE,
0095' BC C C PFI RC ;SET FLAG
0095' C0 010B' C C LBR TXT ;AND RETURN
0098' 6F C C TSTR? INP STATUS ;GET UART STATUS
009A' FE C C SRS ;IS THE TREE?
009B' CB 00DC' C C LBMF TREE? ;IF NOT, KEEP TRYING
009E' 0A C C LNI RA ;GET NEXT CHARACTER
009F' FB 7E C C XRI STOP ;MESSAGE OVER?
00F1' C2 010B' C C LEZ TXT ;IF SO EXIT
00F4' EA C C SEX RA ;OTHERWISE, TYPE THE
00F5' 66 C C OUT DATA ;CHARACTER
00F6' D4 C+ CALL INCAR ;MONITOR THE LOOP FOR
00F7' 0145' C+ GHI RC ;ANY ERRORS?
00F9' 9C C C GHI RC ;ANY ERRORS?
00FA' CA 010B' C C LBNZ TXT ;IF SO, EXIT
00FD' 2A C C DEB RA ;WAS THE LAST CHAR.
00FE' EA C C SEX RA ;TYPED THE SAME AS
00FF' 8C C C GLO RC ;THE CHARACTER JUST
0100' F3 C C XOR ;RECEIVED?
0101' CA 0108' C C LBNZ BADCHR ;IF SO,
0104' 1A C C INC RA ;REPOSITION POINTER
0105' C0 00DC' C C LBR TREE? ;AND CONTINUE
0108' F8 02 C C BADCHR: LDI 02H ;OTHERWISE, SET FLAG
010A' BC C C PFI RC
010B' F8 03 C C TXT: LDI LOW SCRATCH 2 ;RESTORE OLD ADDRESS
010D' A7 C C PLO K7 ;POINTER
010E' 47 C C LDA K7
010F' EA C C PFI RA
0110' 47 C C LDA K7
0111' AA C C PLO RA
0112' D5 C+ RETURN ;AND RETURN

; INCLUDE ASKOK.MAC
**************
; * ASKOK.MAC *
**************

; + ASK FOR FINAL PERMISSION TO CARRY OUT A COMMAND +

; Type OK? (Y/M) and input a response. Set RC.0 to 00
; upon detecting a "Y". Exit upon detecting any error
; with the UART status word remaining in RC.1.

0113' D4 C+ ASKOK: CALL SALTY ;ASK OK?
INTERROGATOR CONTROL/DATA LOGGER (PRWLR.MAC) MACRO-18 3.36
WOODS HOLE OCEANOGRAPHIC INST. OCEAN ENGINEERING

0116' 0423' C
0118' 9C C
0119' CA 012A' C
011C' D4 C+
011D' 0145' C+
011F' 9C C
0120' CA 012A' C
0123' 8C
0124' FB 59 C
0126' CA 012A' C
0129' AC C
012A' C+
012A' D5 C+

C
C
C
C
C
C
C
C
C
C
C
C
C
C
C
C
C
C
C
C
C
C
C

012B' 46 C
012C' BA C
012D' 46 C
012E' 4A C
012F' D4 C+
0130' 0145' C+
0132' 9C C
0133' CA 0144' C
0136' 8C C
0137' EA C
0138' F3 C
0139' CA 0143' C
013C' IA C
013D' F8 7E C
013F' F3 C
0140' CA 012F' C
0143' AC C
0144' C
0144' D5 C+

INCH: RETURN ;TO SAIL

; INCLUDE COMPAR.MAC
; ***************
; * COMPAR.MAC *
; ***************

;+ COMPARE RECEIVED STRING WITH STORED STRING +
; (RA + RC)
; This subroutine will sequentially input characters
; and compare them with a character string stored in
; permanent memory. Unsuccessful comparisons will
; cause the subroutine to exit leaving a non-zero
; result in the low half of register C
;
012B' 46 C
012C' BA C
012D' 46 C
012E' 4A C
012F' D4 C+
0130' 0145' C+
0132' 9C C
0133' CA 0144' C
0136' 8C C
0137' EA C
0138' F3 C
0139' CA 0143' C
013C' IA C
013D' F8 7E C
013F' F3 C
0140' CA 012F' C
0143' AC C
0144' C
0144' D5 C+

; INCLUDE INCHAR.MAC
; ***************
; * INCHAR.MAC *
**INPUT CHARACTER AND UART STATUS FLAG TO REGISTER C**

```assembly
.INCHR:
  DIL 00 ; POINT TO A SCRATCH
  PLO R7 ; LOCATION
  PHI RC ; RESET STATUS FLAGS
  SEX R7 ; X POINTS TO SCRATCH
  NDA: INP STATUS ; GET UART STATUS
  ANX 10H ; IS THE LOOP CLOSED ?
  LEZ TSTD ; IF SO TEST FOR DA
  LDI 00H ; OTHERWISE, USE
  PHI RC ; SET ERROR FLAG
  RETURN ; AND RETURN

.TSTD: LDM R7 ; RESTORE STATUS
  SBR ; IS DATA AVAILABLE ?
  LDN NDA ; IF NOT, TRY AGAIN
  ANX 05H ; OTHERWISE, LOOK FOR ERRORS
  LEZ DATAIN ; IF FOUND SET FLAG
  PHI RC ; AND
  RETURN ; EXIT. OTHERWISE,

.DATAIN: INP DATA ; GET THE CHARACTER
  PLO RC ; AND PLACE IN RC LOW
  XRI "#" ; IS IT A "#"?
  LBNZ DADONE ; IF NOT RETURN
  LDI 40H ; OTHERWISE SET
  PHI RC ; FLAG THEN
  DADONE: RETURN ; RETURN TO MAIN

INCLUDE GETHEX.MAC

******

* GETHEX.MAC *

******
```

: This subroutine will load RB with a four digit hex number. Only the last four hex digits typed are
:entered. Non hex entries will cause this routine
:to exit.

GETHEX:;

;THIS WILL BE A PUBLIC
;CLEAR REGISTER B

GETCHR: CALL INCHR

;GET A CHARACTER

;TEST UART FOR ERRORS

;IF FOUND EXIT

;CONVERT ASCII TO HEX

;LOOK FOR NON-HEX ENTRY

;IF FOUND EXIT, ELSE

;TRANSFER HEX DIGIT

;TO HIGH HALF OF RC

;PREPARE TO SHIFT

;HEX CHARACTER TO RB

;BEGIN SHIFT

;IS THIS THE FOURTH

;SHIFT ?

;IF NOT SHIFT AGAIN

;ELSE GET NEXT DIGIT

;EXIT

;INCLUDE INDEC.MAC

**************

* INDEC.MAC *

**************

+ INPUT AND STORE DECIMAL NUMBERS +

(RS,RA,RB,RC, RD)

;This subroutine will input and store n decimal digits
;beginning at the address specified by the two bytes
;following the call instruction. The number of bytes
;to store is specified by the single byte following
;the call. Only the last n digits typed will be stored.
;Errors and non-decimal entries cause an exit which
;leaves the status word in RC.1 and the last digit
:type in RC.0. NOTE: n may not be greater than 4.
0196' 46 C INDEX: LDA R6 ;GET STORE ADDRESS
0197' BD C PHI R6 ;AND PLACE IN RD
0198' 46 C LDA R6
0199' AD C PLO RD
019A' F8 00 C LDI 00H ;ZERO REGISTER B
019C' AB C PLO RB
019D' BB C PHI RB
019E' C GETDEC: CALL INCHR ;GET A CHARACTER
019E' D4 C+
019E' 0145' C+
01A1' 9C C GHI RC ;TEST FOR ERRORS
01A2' CA 01C9' C LENZ XINDEC ;EXIT IF ERROR IS FOUND
01A5' D4 C+
01A6' 0058' C+
01A8' 9C C GHI RC ;EXIT IF NOT HEX
01A9' CA 01C9' C LENZ XINDEC
01AC' 8C C GLO RC ;TEST FOR DECIMAL
01AD' FF AD C SHI QA0H ;AND SET ERROR FLAG
01AF' C3 01C5' C LDIF XINE ;IF NOT DECIMAL
01B2' F8 04 C LDI 04 ;OTHERWISE, USING R9
01B4' A9 C PLO R9 ;AS A COUNTER, SHIFT
01BD' 8C C SHFCT: GLO RC ;THE DIGIT A BIT AT
01BD' FE C SHL
01BE' AC C PLO RC
01B8' 8B C GLO RB
01B9' 7E C RSHL
01BA' AB C PLO RB
01BB' 9B C GHI RB
01BC' 7E C RSHL
01BD' BB C PHI RB
01BE' 29 C DEC R9
01BF' 89 C GLO R9 ;TEST FOR FOURTH SHIFT
01CO' CA 01BS' C LENZ SHFCT ;SHIFT AGAIN IF NOT DONE
01C3' CO 019E' C LBR GETDEC ;OTHERWISE, GET NEXT DIGIT
01C5' F8 01 C XINE: LDI 01H ;INDICATE NON-DECIMAL
01C8' BC C PHI RC ;AND RETURN TO SAIL
01C9' 22 C XINDEC: DEC R2 ;SAVE THE CONTENTS
01CA' 8C C GLO RC ;OF REGISTER C
01CB' 73 C STXD
01CC' 9C C GHI RC
01CD' 73 C STXD
01CE' 46 C LDA R6 ;GET NUMBER OF DIGITS
01CF' A9 C PLO R9 ;TO STORE
01D0' D4 C STRDEC: CALL R5B2A ;SHIFT A DIGIT TO RA
01D0' 00B7' C+
01D3' 04 C DB 04H ;SHIFT TO LSB
01D4' 9A C GHI RA
01D5' F6 C SHR
01D6' F6 C SHR
01D7' F6 C SHR
01D8' F6 C SHR
01D9' 5D C STR RD ;STORE DECIMAL DIGIT
INCLUDE MACIN.MAC

**************

**************

; + PROMPTS FOR AND LOADS RB WITH A HEX NUMBER +

; (RC)

;This subroutine will prompt the operator for
;a hex number using the prompt message addressed
;by the in-line code after the call instruction.
;Only the last four hex digits typed are entered.
;UART errors and non-hex entries cause this routine
;to exit with a non-zero value remaining in RC.1

SEX R7
LUI LOY ((SCRATCH+4))
SAVE CONTENTS OF
REGISTER RA
FIRST THE LOW HALF
GET THE HIGH HALF
GET LOW HALF OF
MESSAGE ADDRESS
TYPE THE MESSAGE

GEHE

;A PUBLIC ROUTINE

;USING R7 AS A POINTER

;SAVE THE CONTENTS OF

PLO R7
REGISTR RA

;FIRST THE LOW HALF

;THEN THE HIGH HALF

;GET THE HEX NUMBER

;GET THE STATUS WORD

;MASK NON-HEX FLAG

;EXIT ON UART ERRORS

;GET THE NON-HEX ENTRY

;A SPACE

;IF SO RESET ERROR FLAG

;A CARRIAGE RETURN

;IF SO RESET ERROR FLAG

54
020E'  F8 10  C  LDI  10H ; OTHERWISE INDICATE
0210'  BC  C  PHI  RC ; A NON-HEX ENTRY AND
0211'  F8 08  C  EXPAND:  LDI  LOW  (SCRATCH+3) ; PREPARE TO RESTORE RA
0213'  A7  C  PLO  R7 ; GET OLD RA HI
0214'  47  C  LDA  R7 ; PUT IT BACK
0215'  BA  C  PHI  RA ; GET OLD RA LO
0216'  47  C  LDA  R7 ; PUT IT BACK
0217'  AA  C  PLO  RA ; GO BACK TO SAIL
0218'  D5  C+  C
0219'  F8 00  C  CLRCL:  LDI  00 ; CLEAR ERROR FLAGS
021B'  BC  C  PHI  RC
021C'  00 0211'  C  LDR  EXPAND ; RETURN TO SAIL:

; INCLUDE TYPEC.MAC
***************
; * TYPEC.MAC *
***************

; CONVERT THE CONTENT OF RC TO ASCII AND TYPE +

; (RC)
 ; This subroutine converts the hex contents of
; the low half of RC to two ASCII characters, stores
; the characters at "ASCII", and types them via SALITY.

021F'  D4  C+  C
0220'  0085'  C+  C
0222'  F8 07  C  LDI  LOW  (SCRATCH+2) ; PREPARE TO STORE
0224'  A7  C  PLO  R7 ; RESULT OF CONVERT
0225'  E7  C  SEX  R7
0226'  F8 7E  C  LDI  STOP ; STORE TTY STOP
0228'  73  C   STXD
0229'  9C  C   GHI  RC ; GET RESULT OF CONVERT
022A'  73  C   STXD ; AND STORE IT
022B'  8C  C   GLO  RC ; GET READY TO CONVERT
022C'  F6  C   SHR ; LOW ORDER BYTE
022D'  F6  C   SHR
022E'  F6  C   SHR
022F'  F6  C   SHR
0230'  AC  C   PLO  RC ; CONVERT HIGH ORDER
0231'  D4  C+  C
0232'  0085'  C+  C
0234'  9C  C   GHI  RC ; BYTE, GET RESULT
0235'  57  C   STR  R7 ; AND STORE IT
0236'  D4  C+  C
0237'  00CB'  C+  C
0239'  FF05  C   DW  SCRATCH
                      C   RETURN

55
INTERROGATOR CONTROL/DATA LOGGER (PNAVLOG.MAC) MACRO-18 3.36
Woods Hole Oceanographic Inst. Ocean Engineering

023B' D5 C+ C
C;
C;
C INCLUDE GET2DH.MAC
C;
C ****************
C;
C * GET2DH.MAC *
C;
C ****************;
C;
C;
C + GET TWO FOUR DIGIT HEX NUMBERS +
C;
C;
C (RA + RB + RC)
C;
C;
C ;This subroutine obtains two four digit hex numbers.
C;
C ;The first number is placed in RA, the second in RB.
C;
C;
C GET2DH:
C CALL PHXN ;THIS IS A PUBLIC
C PROMPT FOR FIRST
C
023C' D4 C+ C
023D' 0185' C+ C
023E' 03DD' C
0241' 9C C
0242' CA 0252' C
0245' 9B C
0246' BA C
0247' 8B C
0248' AA C
0249' D4 C+ C
024A' 0185' C+ C
024C' 03BB' C
024E' 9C C
024F' CA 0252' C
0252' C
0252' D5 C+ C

C INCLUDE CALCRC.MAC
C;
C ****************
C;
C * CALCRC.MAC *
C;
C ****************;
C;
C + CALCULATE A NEW CRC VALUE +
C;
C;
C ;This subroutine will calculate a new value CRC each
C ;time it is called. The old value will be over
C ;written, the address pointer (RA) will be
C ;incremented, and the block counter (RB) will
C ;be decremented
C;
C 0253' F8 0B C CALCRC: LDI LOW (CRCHI) ;POINT TO CRC HI
0255' AC C PLO RC ;USE RC AS THE POINTER
0256' F8 FF C LDH HIGH (CRCHI)
0258' BC C PHI RC

56
0259' F8 00 C LDJ 00 ;POINT TO A SCRATCH
025A' A7 C PLO R7 ;LOCATION WITH GLOBAL
025C' 4A C SEX RC ;POINT TO CRC HI BYTE
025E' F3 C XOR RC ;GET MEMORY BYTE
025F' 57 C STR R7 ;XOR WITH MEMORY BYTE
0260' F6 C SHR ;SAVE RESULT
0261' F6 C SHR ;DIVIDE RESULT
0262' F6 C SHR ;BY 16
0263' F6 C SHR
0264' E7 C SEX R7 ;POINT TO RESULT
0265' F3 C XOR ;OF FIRST XOR AND XOR
0266' 57 C STR R7 ;WITH RESULT OF DIVIDE
0267' FE C SHL ;MULTIPLY BY 16
0268' FE C SHL
0269' FE C SHL
026A' FE C SHL
026B' 1C C INC RC ;POINT AT CRC LO BYTE
026C' BC C SEX RC
026D' F3 C XOR ;XOR WITH RESULT OF
026E' 2C C DEC RC ;MULTIPLY, AND STORE
026F' 5C C STR RC ;RESULT AT CRC HI BYTE
0270' 07 C LIR R7 ;GET RESULT OF SECOND
0271' F6 C SHR ;XOR, AND DIVIDE
0272' F6 C SHR ;IT BY 8
0273' F6 C SHR
0274' F3 C XOR ;XOR WITH CRC HI BYTE
0275' 5C C STR RC ;RESULT IS NEW CRC HI
0276' 07 C LIR R7 ;GET RESULT OF SECOND
0277' FE C SHL ;XOR AND
0278' FE C SHL ;MULTIPLY IT BY 32
0279' FE C SHL
027A' FE C SHL
027B' FE C SHL
027C' E7 C SEX R7 ;XOR THIS PRODUCT WITH
027D' F3 C XOR ;THE PRODUCT OF THE
027E' 1C C INC RC ;FIRST MULTIPLY
027F' 5C C STR RC ;RESULT IS NEW CRC LO
0280' 17 C INC R7 ;POINT AT SYSTEM FLAG
0281' D5 C+ C RETURN ;EXIT

INCLUDE MCLK.MAC
***************
* MCLK.MAC *
***************

; + INCREMENT THE CLOCK BY ONE MINUTE +
; ---------------------------------------
; (RA + RC)
;
; This subroutine will increment the software
; clock by one minute. The year day will be
; reset to 001 one day after year day 365,
;i.e., leap year not allowed! A second clock
;which is always one minute ahead of the system
;clock will also be incremented. If the "GO" flag
;is set, the value at MINOW will be decremented.
;If the new value at MINOW is equal to 1, Q will be
;set indicating the start of a measurement sequence.
;
;Reserve seven locations in RAM to hold decimal
;time code data of the system clock.

FF10
HD EQU GLOBAL+10H ;HUNDRED OF DAYS

FF16
UM EQU GLOBAL+16H ;UNITS OF MINUTES

FF29
SHD EQU GLOBAL+29H ;HUNDRED OF DAYS

FF2F
SIM EQU GLOBAL+2FH ;UNITS OF MINUTES (+1)

FF43
GOFIQ EQU GLOBAL+43H ;GO FLAG

0282' 7B
MLCLK: SEQ ;USE Q AS A LOOP COUNTER

0283' E7
SEX R7 ;USE R7 AS A POINTER

0284' F8 16
LMX LOW (UM) ;POINT AT SYSTEM TIME

0286' A7
MLCLK: PLO R7 ;START HERE FOR SECOND PASS

0287' 07
LMN R7 ;GET UNITS OF MINUTES

0288' AA
PLO RA

0289' 1A
INC RA ;ADD 1 MINUTE

028A' 8A
GLO RA ;GET NEW MIN. COUNT

028B' FB 0A
XR1 OAH ;IS IT NOW 10 ?

028D' CA 034E'
LENZ STNEN ;IF NOT, STORE NEW MIN.

0290' 73
STKD ;STORE A 0 AT U.M.

0291' 07
LMN R7 ;GET TENS OF MINUTES

0292' AA
PLO RA

0293' 1A
INC RA ;ADD 1 TO TEN MIN. CNT.

0294' 8A
GLO RA ;GET NEW TEN MIN. CNT.

0295' FB 06
XR1 06H ;IS IT NOW MINUTE 60 ?

0297' CA 034E'
LENZ STNEN ;IF NOT STORE NEW T.M.C.

029A' 73
STKD ;STORE 0 AT TM

029B' 07
LMN R7 ;GET UNITS OF HOURS

029C' AA
PLO RA

029D' 1A
INC RA ;ADD 1 TO UNITS OF HRS.

029E' 8A
GLO RA ;GET NEW U.H. COUNT

029F' FB 0A
XR1 OAH ;IS IT NOW 10 ?

02A1' C2 034A'
LENZ INCH ;IF SO INC. T.H.

02A4' 8A
GLO RA ;RESTORE NEW U.H. CNT.

02A5' FB 04
XR1 04H ;IS IT NOW 4 ?
02A7' CA 034E' C LENZ STRNEW ;IF NOT STORE NEW U.H.C.

;The units of hour counter is now 4. If the tens of hour
counter is now 2, both these counters will be reset, and
the units of days counter will be incremented.

02A7' 27 C DEC R7 ;POINT AT TENS OF HOURS
02A7' 47 C LDA R7 ;GET TENS OF HOURS
02AC' FB 02 C XRI 02H ;IS IT NOW 2 ?
02AE' CA 034E' C LENZ STRNEW ;IF NOT STORE A 4 AT UH
02B1' 73 C STED ;ZERO TO UNITS OF HOURS
02B2' 73 C STED ;ZERO TO TENS OF HOURS

;This loop will update the days counter.

02B3' F8 03 C
02B5' AC C
02B6' 07 C UPDATE: LEN R7 ;GET A DAY DIGIT
02B7' AA C
02B8' 1A C INC RA ;ADD 1 DAY COUNT
02B9' 8A C GLO RA ;GET NEW DAY COUNT
02BA' FB 0A C XRI 04H ;IS IT NOW 10 ?
02BC' CA 034E' C LENZ STRNEW ;IF NOT STORE NEW DAY COUNT
02BF' 73 C STED ;ZERO THIS DIGIT
02C0' 2C C
02C1' 8C C
02C2' CA 02B6' C LENZ UPDATE ;ZERO EXIT THE LOOP

;If it is year day 366, reset to day 001

02C5' C9 0353' C
02CA' A7 C
02CB' 47 C
02CC' FB 03 C
02CD' CA 02E4' C
02D1' 47 C
02D2' FB 06 C
02D4' CA 02E4' C
02D7' 07 C
02D8' FB 06 C
02DA' CA 02E4' C
02DD' F8 01 C
02DF' 73 C
02E0' FB 00 C
02E2' 73 C
02E3' 57 C

;If Q is set this is the first time through the loop. Point
;at time plus 1 minute, copy current time to this local, reset
;Q and go through the loop a second time. If Q is not set, test
;the condition of the "Q" flag.

02E4' C9 0300' C
02E7' F8 10 C
02E9' A7 C
INTO ccDATA UJ (PN\I.MA)
MACR18 3.36
PAGE 1-19

; List of instructions

02EA' F8 29 C LDI LOW (SIHD) ; Use RA as a pointer
02ED' 97 C FLO RA
02EE' BA C PHI RA
02EF' F8 07 C LDI OTH ; Use RC as a loop counter
02F1' AC C FLO RC ; Initially set to 7
02F2' 47 C CPTIM: LDA R7 ; Get a digit of current
02F3' 5A C STR RA ; Time and store it
02F4' 1A C INC RA ; Move pointer
02F5' 2C C DEC RC ; Test counter
02F6' 8C C GLO RC ; And if done
02F7' CA 02F2' C LENR CPTIM ; Point
02FA' F8 2F C LDI LOW (SIUM) ; At time +1 minute
02FC' 7A C SEQ ; Reset loop counter
02FD' CO 0286' C LER MCLK ; Go through again

; Since Q was not set this is the second pass through
; the loop. Decrement the measurement interval counter,
; if and only if the "GO" flag is equal to AAH. If
; after decrementing the measurement interval counter
; its new value is 01H, request a measurement by setting
; Q prior to exiting.

0300' F8 43 C TESTG: LDI LOW (GOFLG) ; Point at GO flag
0302' A7 C PLO R7
0303' 47 C LDA R7 ; And if not set exit
0304' FB AA C XRI QA9H
0306' CA 0328' C LENR TSTIME ; If set, test for
0309' 07 C LDN R7 ; Start time
030A' FB AA C XRI QA9H
030C' CA 0328' C LENR TSTIME
030F' F8 26 C LDI LOW (MIN HW) ; Get current interval
0311' A7 C PLO R7 ; Count and decrement
0312' 47 C LDA R7
0313' BA C PHI RA
0314' 07 C LDN R7
0315' AA C PLO RA
0316' 2A C DEC RA
0317' 8A C GLO RA ; Save new interval
0318' 73 C STD
0319' 9A C GHI RA
031A' 57 C STR R7
031B' CA 0358' C LENR TSTICK ; Exit if new interval is
031E' 8A C GLO RA ; Not equal to 1 minute
031F' FB 01 C XRI Q0H
0321' CA 0358' C LENR TSTICK
0324' 7B C SEQ ; Otherwise, set Q first
0325' CO 0358' C LER TSTICK ; Then exit

; Compare current time +1 minute with start time. If they
; are equal set the GO flag and request a branch to the
; measurement sequence by exiting with Q set.

0328' F8 30 C TSTIME: LDI LOW (DSHD) ; Point at start time
032A' AA C FLO RA ; Using RA as the
032B' 97 C GHI R7 ; Pointer
032C' RA C
032D' F8 29 C
032F' A7 C
0330' E7 C
0331' 8A C
0332' FB 37 C
0334' C2 0340' C
0337' 4A C
0338' F3 C
0339' CA 0358' C
033C' 17 C
033D' CO 0331' C
0340' F8 44 C
0343' F8 A9 C
0345' 57 C
0346' 7B C
0347' CO 0358' C
034A' 73 C
034B' 07 C
034C' A9 C
034D' 1A C
034F' 8A C
0350' CO 02CA' C
0353' F8 29 C
0355' CO 02CA' C
0358' F8 17 C
035A' A7 C
035B' 07 C
035C' C2 037A' C
035F' D5 C+

INCLUDE INTRPT.MAC
***************
* INTRPT.MAC *
***************

++; THIS IS THE INTERRUPT SERVICE ROUTINE +

; This routine handles interrupt requests. The only
; interrupt which can occur in this system is the
; one minute tick. The purpose of this routine is
; simply to update the clock. Two exits are possible,
; the normal exit, and the forced exit. A forced exit
; occurs when upon returning from the routine which
; advances the clock, Q is set indicating the start
; of a measurement sequence.

0360' E2 22 C EXINT: SEX R2 ;RESTORE STACK POINTER AND
0361' 70 C RET ;ENABLE FURTHER INTERRUPTS

; INTERRUPT: DEC R2 ;POINT TO A CLEAR LOCATION
0362' 78 C SAV ;SAVE OLD X AND P
0364' 76 C IN: IE R2 ;MOVE DF TO MSB OF D
0365' 73 C STXD ;SAVE DEF
0367' 73 C STXD
0368' 87 C GLO R7 ;SAVE THE CONTENTS OF R7
0369' 73 C STXD

;At this point we have preserved enough of the
;register data to safely test the tick flag and
;exit if it is set.

036A' F8 17 C LDI LOW (TICK)
036C' A7 C PLO R7
036D' 07 C LDN R7 ;GET TICK FLAG
036E' CA 038E' C LNIZ XINTF ;EXIT IF SET

;Tick flag was not set so continue saving registers

0371' 8A C GLO RA ;SAVE RA
0372' 73 C STXD
0373' 9A C GHI RA
0374' 73 C STXD
0375' 8C C GLO RC ;SAVE RC.0
0376' 73 C STXD

;With these registers preserved the clock may
;now be incremented.

0377' C0 0282' C LBR MICLK ;INCREMENT THE CLOCK
0379' C9 038A' C ENINT: LBRQ RSTRX ;IF Q IS SET, LOAD R3
037D' F8 4B' C LDI LOW (MSRSQ)
037F' A3 C PLO R3 ;WITH THE ADDRESS OF
0380' F8 0F' C LDI HIGH (MSRSQ)
0382' B3 C PHI R3 ;MEASUREMENT SEQUENCE
0383' F8 FF C LDI LOW (STACK)
0385' A2 C PLO R2 ;RESTORE STACK POINTER
0386' F8 FF C LDI HIGH (STACK)
0388' B2 C PHI R2 ;AND
0389' D3 C SEP R3 ;EXIT INTERRUPT, OTHERWISE
038A' E2 C RSTRX: SEX R2 ;RESTORE POINTER AND
038B' C0 039A' C LBR RESTR ;RESTORE ALL REGISTERS

;This is the fast interrupt exit

038E' 12 C XINTF: INC R2 ;POINT TO OLD R7.0
038E' 42 C EXCON: LDA R2
0390' A7 C PLO R7 ;RESTORE R7
0391' 42 C LDA R2

62
C; This is the slow interrupt exit.

C
C RESTR: INC R2 ; POINT TO OLD RC.0
C LDA R2 ; GET RC.0
C FLO RC ; RC.0
C LDA R2 ; RESTORE RA
C PHI RA
C LDA R2
C FLO RA
C LDR EXCON ; CONTINUE RESTORING DATA

INCLUDE ISAIL.MAC

C ; This module is designed to be a "KERNEL" around which 
C ; the operating system of any SAIL oriented instrument 
C ; may be based. The program expects the UART to be an 
C ; 1854 and the CPU to be an 1802. The UART should be 
C ; located at I/O ports 6 and 7, and have its BS (bar) 
C ; input connected to a loop status indicator. The RCA 
C ; Standard Call and Return Technique (SCRT) is used.
C ;
C ; 1. Define the NAME of the SAIL device.
C ; 2. Define the PROMPT character to be used.
C ; 3. Change the HELP file as required.
C ;
C ; Define a few RAM locations.
C ;
FF05 C SCRATCH EQU GLOBAL+5 ; A SCRATCH LOCATION
FF0B C CRCHI EQU SCRATCH+6 ; CRC HI BYTE
FF0C C CRCLO EQU CRCHI+1 ; CRC LO BYTE
C ;
C ; Note that GLOBAL will always be address m00 and 
C ; defines the start of a RAM page to be used by all 
C ; routines. The first location will usually contain 
C ; either the last character typed or the contents of 
C ; the UART status register. The second location is 
C ; reserved for the system error flag. Register R7 
C ; will always point to some GLOBAL location.
C ;
C ; *******************************
C ; * CUSTOMIZED FOR INTERROGATOR DATA LOGGER *
C ; *******************************
C ;
03A4' P8 17 C CLATTIC: LDI LOW (TICK)
03A6' A7 C FLO R7

63
03A7' F8 01 C LDI 01H ;SET THE TICK FLAG
03A9' 57 C STR R7 ;ADVANCE THE CLOCK
03AA' D4 C+ CALL MCLK
03AB' 0282' C+ ;IF TIME MEASURE, OTHERWISE
03AD' CI 0F4B' C LDQ MSRSEQ
03B0' F8 01 C SAIL: LDI 01H ;POINT TO THE FLAG
03B2' A7 C PLO K7 ;WORD AND
03B3' F8 00 C LDI 00H ;RESET ALL BITS
03B5' E7 C SEX K7
03B6' 73 C STD
03B7' 6E C IMP DATA ;CLEAR UART DA BIT
03B8' E3 C SEX R3 ;CONFIGURE UART
03B9' 67 C OUT STATUS
03BA' 12 C DB CONFIG
03BB' F8 17 C LDI LOW (TICK)
03BD' A7 C PLO K7
03BE' F8 00 C LDI 00H ;RESET TICK FLAG
03CO' 57 C STR K7
03C1' C0 07BC' C LBR ADDR? ;TEST FOR CLOSED LOOP
03C4' E3 C EXIT: SEX R3 ;CLEAR INTERRUPT LATCH
03C5' 65 C OUT CLRINT
03C6' 00 C DB 00H
03C7' 71 C DLS ;DISABLE INTERRUPTS
03C8' 33 C DB 33H
03C9' C0 0F48' C LBR MAIN ;EXIT THIS MODULE

;The device NAME may be any combination of alphabetic and numeric characters.

03CC' 49 32 7E C NAME: DB "T2",STOP ;DEVICE NAME
007E C STOP EQU "-" ;MESSAGE TERMINATOR
;Pick a character to be used as an instrument PROMPT
003A C PROMPT EQU "::" ;THIS IS THE PROMPT
;Define the hex equivalent of an ASCII carriage return, line feed, space, null, bell, and other characters.
000D C CR EQU 0DH
000A C LF EQU 0AH
0000 C NULL EQU 00H
0020 C SPACE EQU 20H
0003 C ETX EQU 03H
0007 C BEL EQU 07H
;The UART is configured for 7 data bits, even parity, and 1 stop bit. This configuration may be modified by changing the byte stored at CONFIG.
0012 C CONFIG EQU 12H ;UART CONFIGURATION
;Define the I/O
; Store a few often used messages

03CF' 3B    C EOL:  DB ";
03C0' 0A 0D 7E C CR LF:  DB LF,CR,STOP
03C1' 20    C SP SP:  DB SPACE
03C2' 20 7E C SP:  DB SPACE,STOP
03C6' 0A 0D 20 7E C CR LF SP:  DB LF,CR,SPACE,STOP
03C8' 52 43 7E C RCS:  DB "RC",STOP
03C9' 20 46 72 6F C FROM:  DB " From ",ETX,STOP
03C1' 6D 20 03 7E C
03C5' 20 54 5F 20 C TO:  DB " To ",ETX,STOP
03C9' 03 7E C
03C1' 20 4F 76 65 6C C LOCK:  DB "OCK",STOP
0411' 4E 4C 4F 43 C UNLOCK:  DB "UNLOCK",STOP
0415' 4B 7E C
0417' 44 4C 45 7E C DLE:  DB "DLE",STOP
041B' 49 4E 47 7E C IN1:  DB "IN1",STOP
041F' 20 4F 4B 7E C OR:  DB " OR",STOP
0423' 20 20 4F 4B C OR2:  DB " OR (Y/N) ? ",ETX,STOP
0427' 20 28 59 2F C
042B' 4E 29 20 3F C
042F' 20 20 03 7E C
0433' 0A 0D 20 3A C PRMT:  DB LF,CR,SPACE,PROMPT,SPACE,ETX,STOP
0437' 20 03 7E C
043A' 20 20 57 48 C ERROR:  DB SPACE,SPACE,"WHAT?",BEL,STOP
043E' 41 54 20 3F C
0442' 07 7E C
0444' 5F 76 65 7E C OVE:  DB "ove",STOP
0448' 20 52 45 41 C READY:  DB " READY",STOP
044C' 44 59 7E C
044F' 20 30 30 2E C SECS:  DB SPACE,"00... ",STOP
0453' 28 26 20 7E C
0457' 40 7E C AT:  DB "AT",STOP
0459' 49 4D 45 7E C TIME:  DB "TIME",STOP
045D' 43 48 45 44 C SCHED:  DB "SCHEDULE",STOP
0461' 55 4C 45 7E C
0465' 0D 0A 20 53 C STDAY:  DB CR,LF," Start on day = ",STOP
0469' 74 61 72 74 C
INTERROGATOR CONTROL/DATA LOGGER (PNAVLR.MAC) MACRO-18 3.36
Woods Hole Oceanographic Inst. Ocean Engineering

046D' 20 6F 6E 20 C
0471' 64 61 79 20 C
0475' 3D 20 7E C
0478' 20 20 66 6F C
047C' 75 72 20 3D C
0480' 20 7E C
0482' 20 20 6D 69 C
0486' 68 75 74 65 C
048A' 20 3D 20 7E C
048E' 0D 0A 20 4D C
0492' 65 61 73 75 C
0496' 72 65 69 65 C
049A' 6E 74 20 69 C
049E' 6E 74 65 72 C
04A2' 76 61 6C 2C C
04A6' 20 20 6D 69 C
04A8' 6E 75 74 65 C
04AE' 73 20 3D 20 C
04B2' 7E C
04B3' 0D 0A 20 53 C
04B7' 63 68 65 64 C
04B9' 75 6C 65 72 C
04BF' 20 69 73 20 C
04C3' 7E C
04C4' 41 43 54 49 C
04C8' 56 45 20 77 C
04CC' 69 74 68 20 C
04D0' 7E C
04D1' 48 20 6D 69 C
04D5' 6E 75 74 65 C
04D9' 73 20 72 65 C
04DD' 6D 61 69 65 C
04E1' 69 6E 67 20 C
04E5' 74 6F 20 74 C
04E9' 68 65 20 66 C
04ED' 65 78 74 20 C
04F1' 6D 65 61 73 C
04F5' 75 72 65 6D C
04F9' 65 6E 74 2E C
04FD' 7E C
04FE' 49 44 4C 45 C
0502' 2E 7E C
0504' 0D 0A 20 50 C
0508' 6F 69 6E 74 C
050C' 65 72 20 69 C
0510' 73 20 61 74 C
0514' 20 7E C
0516' 0D 0A 20 53 C
051A' 63 68 65 64 C
051E' 75 6C 65 72 C
0522' 20 77 61 73 C
0526' 20 4E 4F 54 C
052A' 20 61 63 74 C
052E' 69 76 65 20 C
0532' 21 07 7E C
0535' 0D 0A 20 54 C

STHOR: DB "hour = ",'STOP
STMIN: DB "minute = ",'STOP
MEAN: DB CR,LF, Measurement interval, minutes = ",'STOP
SCMSG: DB CR,LF, Scheduler is ",'STOP
ACTIVE: DB "ACTIVE with ",'STOP
MINREM: DB "H minutes remaining to the next measurement."
IDLE: DB "IDLE.",'STOP
PNTR: DB CR,LF," Pointer is at ",'STOP
NOTACT: DB CR,LF," Scheduler was NOT active !",'BEL,'STOP
MINMIN: DB CR,LF," This interval must be greater than "

66
; This is the HELP file. It contains an explanation
; of the common 1802 monitor functions accessible
; via the sail loop. Special functions that apply
; to the program in which this module is placed may
; be added here. The monitor functions included are:

HELP:   DB   CR,LF,LF
        DB   "INTERROGATOR PROGRAM"
        DB   "Ver. 1.1  Jan. 1985"
        DB   "SYSTEM COMMANDS",CR,LF,LF
        DB   !Maada ddd"
<table>
<thead>
<tr>
<th>Address</th>
<th>Value 1</th>
<th>Value 2</th>
<th>Value 3</th>
<th>Value 4</th>
<th>Type</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>05E9'</td>
<td>61 61 61 61</td>
<td>C</td>
<td></td>
<td></td>
<td>DB</td>
<td>&quot;LOAD MEMORY&quot;</td>
</tr>
<tr>
<td>05E9'</td>
<td>61 61 61 61</td>
<td>C</td>
<td></td>
<td></td>
<td>DB</td>
<td>&quot;DISPLAY MEMORY&quot;</td>
</tr>
<tr>
<td>05E9'</td>
<td>61 61 61 61</td>
<td>C</td>
<td></td>
<td></td>
<td>DB</td>
<td>&quot;RUN PROGRAM&quot;</td>
</tr>
<tr>
<td>05E9'</td>
<td>61 61 61 61</td>
<td>C</td>
<td></td>
<td></td>
<td>DB</td>
<td>&quot;CALCULATE CRC&quot;</td>
</tr>
<tr>
<td>05E9'</td>
<td>61 61 61 61</td>
<td>C</td>
<td></td>
<td></td>
<td>DB</td>
<td>&quot;MOVE MEMORY&quot;</td>
</tr>
<tr>
<td>05E9'</td>
<td>61 61 61 61</td>
<td>C</td>
<td></td>
<td></td>
<td>DB</td>
<td>&quot;TEST RAM&quot;</td>
</tr>
<tr>
<td>05E9'</td>
<td>61 61 61 61</td>
<td>C</td>
<td></td>
<td></td>
<td>DB</td>
<td>&quot;DISPLAY SCHEDULE&quot;</td>
</tr>
<tr>
<td>05E9'</td>
<td>61 61 61 61</td>
<td>C</td>
<td></td>
<td></td>
<td>DB</td>
<td>&quot;?S&quot;</td>
</tr>
<tr>
<td>05E9'</td>
<td>61 61 61 61</td>
<td>C</td>
<td></td>
<td></td>
<td>DB</td>
<td>&quot;?A&quot;</td>
</tr>
</tbody>
</table>
0681' 20 20 44 49 C
0685' 53 50 4C 41 C
0689' 59 20 53 43 C
068D' 48 45 44 55 C
06C1' 4C 45 C
06C3' 0A0D C
06C5' 20 20 21 53 C
06C9' 43 48 45 44 C
06CD' 55 4C 45 C
06D0' 20 20 20 20 C
06D4' 20 20 20 50 C
06D8' 52 4F 47 52 C
06DC' 41 4D 20 53 C
06E0' 43 48 45 44 C
06E4' 55 4C 45 C
06E7' 0A0D C
06E9' 20 20 21 54 C
06ED' 49 4D 45 C
06F0' 20 20 20 20 C
06F4' 20 20 20 20 C
06F8' 20 20 20 53 C
06FC' 45 54 20 43 C
0700' 4C 4F 43 4B C
0704' 0A0D C
0706' 20 20 3F 54 C
070A' 20 20 20 20 C
070E' 20 20 20 20 C
0712' 20 20 20 20 C
0716' 20 20 44 49 C
071A' 53 50 4C 41 C
071E' 59 20 54 49 C
0722' 4D 45 C
0724' 0A0D C
0726' 20 20 21 4C C
072A' 4F 43 4B 20 C
072E' 20 20 20 20 C
0732' 20 20 20 20 C
0736' 20 20 C
0738' 50 52 4F 54 C
073C' 45 43 54 20 C
0740' 4D 45 4D 4F C
0744' 52 59 C
0746' 0A0D C
0748' 20 20 21 55 C
074C' 4E 4C 4F 43 C
0750' 4B 20 20 20 C
0754' 20 20 20 20 C
0758' 20 20 C
075A' 55 4E 50 52 C
075E' 4F 54 45 43 C
0762' 54 20 4D 45 C
0766' 4D 4F 52 59 C
076A' 0A0D C
076C' 20 20 21 49 C
0770' 44 4C 45 20 C
0774' 20 20 20 20 C
INTERROGATOR CONTROL/DATA LOGGER (PMAGLR.MAC) MACRO-18 3.36
WOOS HOLE OCEANOGRAPHIC INST. OCEAN ENGINEERING

0778' 20 20 20 20 C
077C' 20 20 C
077E' 49 4E 48 49 C
0782' 42 49 54 20 C
0786' 53 43 48 45 C
078A' 44 55 4C 45 C
078E' 52 C
078F' 040D C
0791' 20 20 21 50 C
0795' 49 4E 47 20 C
0799' 20 20 20 20 C
079D' 20 20 20 20 C
07A1' 20 20 C
07A3' 54 52 41 4E C
07A7' 53 4D 49 54 C
07AB' 20 41 20 31 C
07AF' 30 20 6D 53 C
07B3' 20 50 55 4C C
07B7' 53 45 0D 0A C
07BB' 7B C

DB "INHIBIT SCHEDULER"

DW ORDDH

DB "PING",""

DB "TRANSMIT A 10 mS PULSE",CR,LF,STOP

;If data is available, the loop is closed.

;Enable interrupts which will take over the
;function of incrementing the real time clock,
;and look for the address sequence. (NAME)

;ADDR?: SEX R3 OUT CLINT ;RESET INTERRUPT

DB 00H ;HARDWARE

DB 33H ;ENABLE INTERRUPTS

CHAR? ;RECEIVED A "#" ?

;Since the # was received, set up to receive NAME

DEVICE: WORD? NAME ;LOOK FOR "NAME"

LBNZ ADDRS? ;TRY AGAIN

LBR IDENT ;IDENTIFY INSTRUMENT

;AT THIS POINT THE INSTRUMENT IS CORRECTLY ADDRESSED

INCLUDE ISCHDS.MAC

************

* SCHDS.MAC *
Test the first character received after a correct address sequence. It should be an H, M, R, ?, !, or $. If it is not, generate an error message.

07D9 FB 17 C
07DB A7 C
07DC FB 00 C
07DE 57 C
07DF D4 C+
07E0 0145' C+
07E2 9C C+
07E3 CA 0939' C+
07E5 8C C+
07E7 FB 48 C
07E9 C2 0905' C
07EC 8C C
07ED FB 3F C
07EF C2 080D' C
07F2 8C C
07F3 FB 21 C
07F5 C2 082F' C
07F8 8C C
07F9 FB 24 C
07FB C2 0BB1' C
07FE 8C C
07FF FB 4D C
0801 C2 0ADF' C
0804 8C C
0805 FB 52 C
0807 C2 0BD5' C

CMDIN: LD1 LOW (TICK)
FL0 R7
LDI OOH ;CLEAR TICK FLAG
STR R7
CHAR? ;GET A CHARACTER

XRI "H" ;IS IT AN "H"
LBZ HELPOUT ;IF SO TYPE THE HELP MESSAGE
GLO RC ;OTHERWISE, RESTORE CHARACTER
XRI "?" ;IS IT A "?"?
LBZ ComM ;IF SO TEST NEXT FOR C OR M
GLO RC ;OTHERWISE, RESTORE CHARACTER
XRI "!" ;IS IT A "!"?
LBZ NorL ;IF SO TEST NEXT FOR M OR LOCK
GLO RC ;OTHERWISE, RESTORE CHARACTER
XRI "S" ;IS IT AN "S"?
LBZ P? ;IF SO TEST NEXT FOR P
GLO RC ;OTHERWISE, RESTORE CHARACTER
XRI "M" ;IS IT AN "M"?
LBZ MOVE ;IF SO GOTO MOVE
GLO RC ;OTHERWISE, RESTORE CHARACTER
XRI "R" ;IS IT AN "R"?
LBZ RAMTST ;IF TEST RAM

This ends the test for the standard sail commands.

Enter any additional command tests after this comment.

***** ADDITIONAL COMMAND TESTS GO HERE *****

LB0 ERROUT ;NOT A RECOGNIZED COMMAND

Determine the character which follows "?". It should be either a "C", "M", "T" or an "S". If it is not, type the error message and go to PRMOUT.

ComM: CHAR? ;GET THE NEXT CHARACTER
0814' 8C  C+  XRI  "C" ;IS IT A "C" ?  
0815' FB 43  C  LBZ  CRC ;IF SO GO TO CRC  
0817' C2 0058'  C  GLO  RC ;OTHERWISE, RESTORE CHARACTER  
0818' 8C  C  XR1  "M" ;IS IT AN "M" ?  
081B' FB 4D  C  LBZ  QUERY ;IF SO GO TO QUERY MEMORY  
081D' C2 009D'  C  GLO  RC ;OTHERWISE, RESTORE CHARACTER  
0820' 8C  C  XR1  "T" ;IS IT A "T"  
0821' FB 54  C  LBZ  QUESTIM ;IF SO GO TO QUESTION TIME  
0823' C2 0B35'  C  GLO  RC ;OTHERWISE, RESTORE CHARACTER  
0827' FB 53  C  XR1  "S" ;IS IT AN "S" ?  
0828' C2 00D6'  C  LBZ  QRUSCE ;IF SO TYPE SCHEDULE  
082C' CO 08F9'  C  
082F' D4  C+ 
0830' 0145'  C+ 
0832' 9C  C+ 
0833' CA 0939'  C+ 
0836' 8C  C+ 
0837' FB 4D  C  XRI  "M" ;IS IT AN "M"  
0839' C2 094A'  C  LBZ  LOAD ;IF SO GO TO LOAD  
083C' 8C  C  GLO  RC ;RESTORE CHARACTER  
083D' FB 4C  C  XR1  "L" ;IS IT AN "L" ?  
083F' CA 084F'  C  LBZ  UP ;IF SO LOOK FOR "OCK"  
0842' D4  C+ 
0843' 012B'  C+ 
0845' 040D'  C+ 
0847' 9C  C+ 
0848' CA 0939'  C+ 
084B' 8C  C+ 
084C' C2 0911'  C  U?;  GLO  RC ;RESTORE CHARACTER  
084F' 8C  C  XR1  "U" ;IS IT A "U" ?  
0850' FB 55  C  LBZ  T? ;IF SO LOOK FOR "LOCK"  
0852' CA 0862'  C  WORD?  UNLOCk ;AND IF FOUND GOTO  
0855' D4  C+ 
0856' 012B'  C+ 
0858' 0411'  C+ 
085A' 9C  C+ 
085B' CA 0939'  C+ 
085E' 8C  C+ 
085F' C2 0925'  C  LBZ  OPEN ;OPEN, OTHERWISE  
0862' 8C  C  T?;  GLO  RC ;RESTORE CHARACTER  
0863' FB 54  C  XR1  "T" ;IS IT A "T" ?
0865' CA 0875' C LENZ S? ;IF SO LOOK FOR "IME"
               C WORD? TIME ;IF FOUND
0868' D4 C+
0869' 0128' C+
086A' 0459' C+
086D' 9C C+
086E' CA 0939' C+
0871' 9C C+
0872' C2 0890' C LEZ LDTIM ;SET THE CLOCK
0875' 8C C S?; GLO RC ;IS IT AN "S"
0876' FB 53 C XRI "S" ;IF SO LOOK FOR
0878' CA 0888' C LENZ IT? ;"SCHEDULE"
               C WORD? SCED ;AND IF FOUND
087B' D4 C+
087C' 0128' C+
087E' 045D' C+
0880' 9C C+
0881' CA 0939' C+
0884' 8C C+
0885' C2 088E' C LEZ LDSCED ;LOAD THE SCHEDULE
0888' 8C C IT?; GLO RC ;RESTORE CHARACTER
0889' FB 49 C XRI "T" ;IS IT AN "T" ?
088B' CA 089B' C LENZ PN? ;IF SO LOOK FOR DLE
               C WORD? DLE ;AND IF FOUND RESET
088E' D4 C+
088F' 0128' C+
0891' 0417' C+
0893' 9C C+
0894' CA 0939' C+
0897' 8C C+
0898' C2 0F11' C LEZ GFT00 ;THE GO FLAG
089B' 8C C PM?; GLO RC ;OTHERWISE, IS IT
089C' FB 50 C XRI "P" ;A "P" ?
089E' CA 08AE' C LENZ NOCHD ;IF SO, LOOK FOR
               C WORD? ING ;"ING", AND IF FOUND
08A1' D4 C+
08A2' 0128' C+
08A4' 041B' C+
08A6' 9C C+
08A7' CA 0939' C+
08A8' 8C C+
08AB' C2 0F37' C LEZ TEXIT ;SEND A PING
               C C C
               ;**** ENTER ADDITIONAL "!" COMMANDS HERE ****
08AE' C0 08F9' C NOCHD: LBR ERRUT ;GOTO ERRUT
               C C C
               ;Determine the character which follows the $,
               ;it should be a "P".
               C C C
               ; P?; CHAR? ;GET THE NEXT CHARACTER
08B1' C
08B1' D4 C+
08B2' 0145' C+
08B4' 9C C+
08B5' CA 0939' C+
08B8' 8C C+
**INTERROGATOR CONTROL/DATA LOGGER (PNAVLOG.MAC) MACRO-18**

**PAGE 1-33**

**WOODS HOLE OCEANOGRAPHIC INST. OCEAN ENGINEERING**

089' FB 50 C XRI "F" ; IS IT A "F" ? C
C ; **** ENTER ADDITIONAL "S" COMMANDS HERE **** C

088' CA 089' C LBNZ ERROUT ; IF NOT GOTO ERROUT C
C GETFLG ; IS THE SYSTEM OPEN ? C

088' F8 01 C+ 08C' A7 C+ 08C1' 07 C+ 08C2' F6 C SHR C

08C3' C3 0A0C' C LEDE RUN ; IF IT IS GOTO RUN C

08C5' C NORUN: TYPMSG NO ; OTHERWISE, TYPE THE NO MSG.

08C6' C+ 08C7' 00CB' C+ 08C8' 03FE' C+ 08C9' 9C C+ 08CC' CA 0939' C+ 08CF' CO 08D'E' C LBR PRNOUT ; AND GOTO PRNOUT C
C ; At this point all "command" tests have been made. C
C ;
C INCLUDE ISACTON.MAC C
C *************** C
C * SACTION.MAC * C
C *************** C

08D0' C; THIS BLOCK OF CODE DEFINES THE ACTION TO BE TAKEN C
C; UPON THE RECEIPT OF STANDARD SAIL COMMANDS C
C IDENT: TYPMSG CR LFSP ; IDENTIFY BY TYPING C

08D2' D4 C+ 08D3' 00CB' C+ 08D5' 03D6' C+ 08D7' 9C C+ 08D8' CA 0939' C+ 08D9' D4 C+ 08DC' 00CB' C+ 08DE' 03CC' C+ 08DO' 9C C+ 08EL' CA 0939' C+ 08E4' D4 C+ 08E5' 00CB' C+ 08E7' 044B' C+ 08E9' 9C C+ 08EA' CA 0939' C+ 08ED' C TYPMSG NAME ; INSTRUMENT NAME AND C

08ED' D4 C+ 08EE' 00CB' C+ 08F0' 0433' C+ 08F2' 9C C+ 08F3' CA 0939' C+ 08F6' CO 07D9' C LBR CMDIN ; GET ANOTHER COMMAND
0829'  C   ; ERROR: TYPMSG ERROR ; TYPE ERROR SEQUENCE
0829'  D4  C+
087A'  00CB'  C+
08FC'  043A'  C+
08E'  9C  C+
08F'  CA 0939'  C+
  0902'  CO 091A'  C
  0905'  C   LBR RSTFLG ; GOTO PRMT VIA RESER-
  0906'  D4  C+
  0909'  00CB'  C+
  0909'  053E'  C+
  090A'  9C  C+
  090B'  CA 0939'  C+
  090E'  CO 08ED'  C
  0911'  C   LBR PRMOUT ; GET NEXT COMMAND
  0911'  D4  C+
  0912'  00CB'  C+
  0914'  0413'  C+
  0916'  9C  C+
  0917'  CA 0939'  C+
  091A'  F8 01  C   RSTFLG: LDI 01H ; POINT AT SYSTEM FLAG
  091C'  A7  C   FLO R7
  091D'  E7  C   SEX R7
  091E'  F8 FE  C   LDI OFEH ; MASK ALL BUT OPEN BIT
  0920'  F2  C   AND ; RESET THIS BIT
  0921'  57  C   STR R7 ; STORE FLAG
  0922'  CO 08ED'  C
  0925'  F8 01  C   OPEN: LDI 01H ; POINT AT SYSTEM FLAG
  0927'  E7  C   FLO R7
  0928'  E7  C   SEX R7
  0929'  F8 01  C   LDI 01H ; MASK ALL BUT OPEN BIT
  092B'  F1  C   OR ; SET THIS BIT
  092C'  57  C   STR R7 ; STORE FLAG
  092D'  D4  C+
  092E'  00CB'  C+
  0930'  041F'  C+
  0932'  9C  C+
  0933'  CA 0939'  C+
  0936'  CO 08ED'  C
  0939'  C   LBR PRMOUT ; GET NEXT COMMAND
  0940'  C   CLOSE: TYPMSG OK ; SAY OK THEN RESET FLAG
  0941'  D4  C+
  0942'  00CB'  C+
  0944'  043A'  C+
  0946'  9C  C+
  0947'  CA 0939'  C+
  094A'  F8 01  C   RSTFLG: LDI 01H ; POINT AT SYSTEM FLAG
  094C'  A7  C   FLO R7
  094D'  E7  C   SEX R7
  094E'  F8 FE  C   LDI OFEH ; MASK ALL BUT OPEN BIT
  0950'  F2  C   AND ; RESET THIS BIT
  0951'  57  C   STR R7 ; STORE FLAG
  0952'  CO 08ED'  C
  0955'  F8 01  C   OPEN: LDI 01H ; POINT AT SYSTEM FLAG
  0957'  E7  C   FLO R7
  0958'  E7  C   SEX R7
  0959'  F8 01  C   LDI 01H ; MASK ALL BUT OPEN BIT
  095B'  F1  C   OR ; SET THIS BIT
  095C'  57  C   STR R7 ; STORE FLAG
  095D'  D4  C+
  095E'  00CB'  C+
  0960'  043A'  C+
  0962'  9C  C+
  0963'  CA 0939'  C+
  0966'  CO 08ED'  C
  0969'  C   LBR PRMOUT ; GET NEXT COMMAND
  0970'  C   ; Depending on the state of an error flag set
  0971'  C   ; after reading the UART status, the program

75
C ;will either branch to MAIN, un-address, or
C ;output the error message.
C ;
 0939' 9C  C  ERVEC: GHI  RC  ;RECOVER STATUS FLAG
 093A' FE  C  SHL  ;LOOP OPEN ?
 093B' C3 03C4' C  LADF  EXIT  ;IF SO EXIT
 093E' FE  C  SHL  ;RECEIVED A "$" ?
 093F' C3 07C9' C  LADF  DEVICE  ;IF SO LOOK FOR NAME
 0942' FE  C  SHL  ;UART ERROR ?
 0943' C3 03B0' C  LADF  SAIL  ;IF SO DE-ADDRESS
 0946' FE  C  SHL  ;OPERATOR ERROR ?
 0947' C3 08F9' C  LADF  ERRUT  ;IF SO SEND ERROR MSG.
 094A' CO 03C4' C  LSR  EXIT  ;NONE OF ABOVE, EXIT
C ;The response to a "?M" is to type the contents of a
C ;specified number of memory locations starting at
C ;a specified address.
C ;
 094D' C  C  QUERY: CALL  GET2B  ;GET START AND END
 094D' D4  C+  C  
 094E' 023C' C+  C  
 0950' 9C  C+  ERROR?  ;REACT TO ERRORS
 0951' CA 0939' C+  C  TYPMSG  CRF  ;TYPE A CR/LF
 0954' D4  C+  
 0955' 00CB' C+  
 0957' 03D0' C+  
 0959' 9C  C+  
 095A' CA 0939' C+  
 095D' 9A  C  TYPADD: GHI  RA  ;TYPE MSB OF ADDRESS
 095E' AC  C  FLO  RC  ;CALL: TYPEC
 095F' D4  C+  C  
 0960' 021F' C+  C  
 0962' 9C  C+  ERROR?  ;REACT TO ERRORS
 0963' CA 0939' C+  C  
 0966' 8A  C  GLO  RA  ;TYPE LSB OF ADDRESS
 0967' AC  C  FLO  RC  ;CALL: TYPEC
 0968' D4  C+  C  
 0969' 021F' C+  C  
 096B' 9C  C+  ERROR?  ;REACT TO ERRORS
 096C' CA 0939' C+  C  
 096F' C  C  
 096F' D4  C+  C  
 0970' 00CB' C+  C  
 0972' 03D4' C+  C  
 0974' 9C  C+  
 0975' CA 0939' C+  
 0978' 4A  C  TYPOUT: TYPMSG  SP  ;TYPE A SPACE
 0979' AC  C  FLO  RC  ;CALL: TYPEC
 097A' D4  C+  C  

76
INTO CATA UX (PNVL.MAC) PAG. 18 3.36

ME.18 3.36 PAGE 1-36

WC ID ~c INS'.
OC mG
097B' 021F' C+ ERROR? ;REACT TO ERRORS
097D' 9C C+ 097E' CA 0939' C+
0981' 2B C DEC RB ;TEST FOR LAST LOCATION
0982' SB C GHI RB
0983' CA 098A' C LENZ LEIST 0986' SB C GLO RB ;IF DONE PROMPT AND
0987' C2 08ED' C LEZ PRMUT ;GET NEXT COMMAND
098A' 8A C LEIST: GLO RA ;OTHERWISE, TEST FOR
098B' FA OF C ANI OFH ;END OF LINE
098D' CA 099C' C LENZ TSTSP ;IF FOUND, TYPE :
098' C TYPMSG EXL ;CONTINUE
0990' D4 C+ 0991' 00CB' C+
0993' 03CE' C+
0995' 9C C+ 0996' CA 0939' C+
0999' CO 095D' C LBR TYPADD ;CONTINUE
099C' 8A C TSTSP: GLO RA ;NEXT LOCATION EVEN ?
099D' F6 C SIR ;IF SO, FIRST TYPE A
099E' CD 096F' C LENF SPOUT ;SPACE, OTHERWISE,
09A1' CO 0978' C LBR BYTOUT ;TYPE NEXT MEMORY BYTE
09A4' C ;The response to a "!M" is to load memory with data
09B4' F8 01 C+ ;as it is input beginning at a specified address, and
09A6' A7 C+ ;continuing until a carriage return is encountered.
09A7' 07 C+
09A8' F6 C SIR ;IF SO, TYPE THE ERROR
09A9' C3 09AF' C LENF LDADD ;MESSAGE THEN PROMPT
09AC' CO 0805' C LBR NOKIN ;OTHERWISE,
09AF' E7 C LDADD: SEX R7 ;RESET THE SYSTEM
09B0' F8 7E C LDI 7EH ;LOCK FLAG AND THE
09B2' F2 C AND ;COMPLETE BYTE FLAG
09B3' 57 C STR R7;
09B4' D4 C+ CALL GETHEX ;GET START ADDRESS
09B5' 016D' C+
09B7' 9C C GHI RC ;TEST FOR UART ERRORS
09B8' FA FE C ANI OFEH ;IF FOUND GOTO ERVEC
09BA' CA 0939' C LENZ ERVEC
09BD' 8C C GLO RC ;WAS THE LAST CHARACTER
09BE' FB 20 C XRI SPACE ;A SPACE?
09C0' C2 09D2' C LEZ NXTD ;IF SO LOAD DATA
09C3' 8C C GLO RC ;WAS THE CHARACTER A
09C4' FB OD C XRI CR ;CARRIAGE RETURN ?
09C6' C2 0A39' C LEZ MODFLG ;IF SO MODIFY COL. FLAG
09C9' 8C C GLO RC ;WAS THE CHARACTER A
09CA' FB OA C XRI LF ;LINE FEED ?
09CC' C2 0A39' C LEZ MODFLG ;IF SO MODIFY COL. FLAG
09CF' CO 08F3' C LBR ERROUT :OTHERWISE, INDICATE AN ERROR
09D2' F8 01 C NXTD: LDI 01H ;POINT AT SYSTEM FLAG
09d4' A7 C  FLO R7 ;RESET COL. FLAG BITS
09d5' E7 C  SEX R7
09d6' F8 9F C  LDI 9FH
09d7' F2 C  AND
09d9' 57 C  STR R7 ;STORE SYSTEM FLAG
09da' D4 C+  CALL INCHAR ;GET NEXT CHARACTER
09db' 0145' C+  ERROR? ;REACT TO ERRORS
09dc' 9C C+  CALL ATRN ;CONVERT TO HEX
09dd' D4 C+  CALL ATRN
09de' 0058' C+  CALL ATRN
09e0' 9C C  GHL RC ;ZERO FOR GOOD CONVERT
09e1' CA 0A06' C  LBNZ CRST ;REACT TO NON-HEX ENTRY
09e2' F8 04 C  LDI 04H ;PREPARE TO ASSEMBLE
09e3' AA C  PLO RA ;AN EIGHT BIT BYTE
09e5' 8C C  DSHFT: GLO RC ;SHIFT HEX DIGIT FROM
09e6' FE C  SHL ;RA LOW TO ED HIGH
09e7' AC C  PLO RC
09e8' 9A C  GHL RA
09e9' 7E C  RSHL
09ea' BA C  PHI RA
09eb' 2A C  DEC RA
09ec' 8A C  GLO RA
09ed' CA 09EB' C  LENZ DSHFT
09ee' D4 C+  GETFGLG ;EIGHT BIT BYTE ASSEMBLED ?
09ef' F8 01 C+  GETFGLG
09f0' A7 C+  GETFGLG
09f1' 07 C+  GETFGLG
09f2' FE C  SHL
09f3' C3 0A23' C  LEDF STORE ;IF SO STORE IT, OTHERWISE
09f5' F8 80 C  LDI 80H ;SET THE COMPLETE BYTE FLAG
09f7' E7 C  SEX R7
09f8' F1 C  OR
09f9' 57 C  STR R7
09fa' C0 09D2' C  LBR NXTD ;AND GET THE NEXT HEX DIGIT
09fb' C CRST: GETFGLG ;THERE WAS A NON-HEX ENTRY
09fc' F8 01 C+  GETFGLG
09fd' A7 C+  GETFGLG
09fe' 07 C+  GETFGLG
09ff' FE C  SHL
0a00' C3 08F9' C  LEDF ERROUT ;IF THE BYTE WAS NOT COMPLETE
0a01' 8C C  GLO RC ;THIS IS AN ERROR. IF THE BYTE
0a02' FB 0D C  XRI CR ;WAS COMPLETE AND THE ENTRY
0a03' C2 08ED' C  LEZ FRNOUT ;HAS A CARRIAGE RETURN, EXIT
0a04' 8C C  GLO RC ;HAS THE ENTRY
0a05' FB 20 C  XRI SPACE ;A SPACE ?
0a06' C2 08D2' C  LEZ NXTD ;IF SO CONTINUE
0a07' 8C C  GLO RC ;HAS THE ENTRY A ";" ?
0a08' FB 3B C  XRI "" ;IF SO SET THE COL. FLAG
0a09' C2 0A2E' C  LEZ COLSET ;AND CONTINUE. OTHERWISE
0a0a' C0 08F9' C  LBR ERROUT ;INDICATE AN ERROR AND EXIT
0a0b' 9A C  STORE: GHL RA ;GET THE BYTE TO BE STORED
0a0c' 5B C  STR RB ;STORE IT
<table>
<thead>
<tr>
<th>Address</th>
<th>Instruction</th>
<th>Operation</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>0A25'</td>
<td>F8 7E</td>
<td>C</td>
<td>LDI 7BH</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>;RESET THE COMPLETE</td>
</tr>
<tr>
<td>0A27'</td>
<td>E7</td>
<td>C</td>
<td>SEX R7</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>;BYTE FLAG</td>
</tr>
<tr>
<td>0A28'</td>
<td>F2</td>
<td>C</td>
<td>AND</td>
</tr>
<tr>
<td>0A29'</td>
<td>57</td>
<td>C</td>
<td>STR R7</td>
</tr>
<tr>
<td>0A38'</td>
<td>1B</td>
<td>C</td>
<td>INC R8</td>
</tr>
<tr>
<td></td>
<td></td>
<td>;INCREMENT ADDRESS POINTER</td>
<td></td>
</tr>
<tr>
<td>0A32'</td>
<td>09D2'</td>
<td>C</td>
<td>LER NXTD</td>
</tr>
<tr>
<td></td>
<td></td>
<td>;GET THE NEXT BYTE</td>
<td></td>
</tr>
<tr>
<td>0A3E'</td>
<td>F8 01</td>
<td>C</td>
<td>COLSET: LDI 01H</td>
</tr>
<tr>
<td></td>
<td></td>
<td>;POINT AT SYSTEM FLAG</td>
<td></td>
</tr>
<tr>
<td>0A30'</td>
<td>A7</td>
<td>C</td>
<td>FLO R7</td>
</tr>
<tr>
<td>0A31'</td>
<td>E7</td>
<td>C</td>
<td>SEX R7</td>
</tr>
<tr>
<td>0A32'</td>
<td>F8 60</td>
<td>C</td>
<td>LDI 60H</td>
</tr>
<tr>
<td></td>
<td></td>
<td>;INDICATE RECEPTION OF</td>
<td></td>
</tr>
<tr>
<td>0A34'</td>
<td>F1</td>
<td>C</td>
<td>OR</td>
</tr>
<tr>
<td></td>
<td></td>
<td>;&quot;.&quot; BY SETTING COL. FLAG</td>
<td></td>
</tr>
<tr>
<td>0A35'</td>
<td>57</td>
<td>C</td>
<td>STR R7</td>
</tr>
<tr>
<td></td>
<td></td>
<td>;SET THE FLAG</td>
<td></td>
</tr>
<tr>
<td>0A36'</td>
<td>C0 09AF'</td>
<td>C</td>
<td>LER LOADD</td>
</tr>
<tr>
<td></td>
<td></td>
<td>;GET NEXT ADDRESS</td>
<td></td>
</tr>
<tr>
<td>0A39'</td>
<td>F8 01</td>
<td>C+</td>
<td>MODEL: GETFLG</td>
</tr>
<tr>
<td></td>
<td></td>
<td>;IF COL. FLAG IS SET</td>
<td></td>
</tr>
<tr>
<td>0A3B'</td>
<td>A7</td>
<td>C+</td>
<td></td>
</tr>
<tr>
<td>0A3C'</td>
<td>07</td>
<td>C+</td>
<td></td>
</tr>
<tr>
<td>0A3D'</td>
<td>E7</td>
<td>C</td>
<td>SEX R7</td>
</tr>
<tr>
<td></td>
<td></td>
<td>;MODIFY IT TO REFLECT THE</td>
<td></td>
</tr>
<tr>
<td>0A3E'</td>
<td>FE</td>
<td>C</td>
<td>SEL</td>
</tr>
<tr>
<td></td>
<td></td>
<td>;RECEPTION OF EITHER A</td>
<td></td>
</tr>
<tr>
<td>0A3F'</td>
<td>FE</td>
<td>C</td>
<td>SEL</td>
</tr>
<tr>
<td></td>
<td></td>
<td>;CARRIAGE RETURN OR LINE FEED</td>
<td></td>
</tr>
<tr>
<td>0A40'</td>
<td>C3 0A4A'</td>
<td>C</td>
<td>LERD NOCR</td>
</tr>
<tr>
<td>0A43'</td>
<td>FE</td>
<td>C</td>
<td>SEL</td>
</tr>
<tr>
<td>0A44'</td>
<td>C3 0A51'</td>
<td>C</td>
<td>LERD NOLF</td>
</tr>
<tr>
<td></td>
<td></td>
<td>;IF THIS FLAG WAS NOT SET</td>
<td></td>
</tr>
<tr>
<td>0A47'</td>
<td>C0 08F9'</td>
<td>C</td>
<td>LER EBCUT</td>
</tr>
<tr>
<td></td>
<td></td>
<td>;THERE IS AN ERROR</td>
<td></td>
</tr>
<tr>
<td>0A4A'</td>
<td>F8 BF</td>
<td>C</td>
<td>NOCR: LDI 0B8H</td>
</tr>
<tr>
<td></td>
<td></td>
<td>;RESET CARRIAGE RETURN BIT</td>
<td></td>
</tr>
<tr>
<td>0A4C'</td>
<td>F2</td>
<td>C</td>
<td>AND</td>
</tr>
<tr>
<td>0A4D'</td>
<td>57</td>
<td>C</td>
<td>STR R7</td>
</tr>
<tr>
<td></td>
<td></td>
<td>;STORE MODIFIED FLAG</td>
<td></td>
</tr>
<tr>
<td>0A4E'</td>
<td>C0 09AF'</td>
<td>C</td>
<td>LER LOADD</td>
</tr>
<tr>
<td></td>
<td></td>
<td>;GET NEW ADDRESS</td>
<td></td>
</tr>
<tr>
<td>0A51'</td>
<td>F8 DF</td>
<td>C</td>
<td>NOLF: LDI 0DFH</td>
</tr>
<tr>
<td></td>
<td></td>
<td>;RESET THE LINE FEED BIT</td>
<td></td>
</tr>
<tr>
<td>0A53'</td>
<td>F2</td>
<td>C</td>
<td>AND</td>
</tr>
<tr>
<td>0A54'</td>
<td>57</td>
<td>C</td>
<td>STR R7</td>
</tr>
<tr>
<td></td>
<td></td>
<td>;STORE MODIFIED FLAG</td>
<td></td>
</tr>
<tr>
<td>0A55'</td>
<td>C0 09AF'</td>
<td>C</td>
<td>LER LOADD</td>
</tr>
<tr>
<td></td>
<td></td>
<td>;GET NEW ADDRESS</td>
<td></td>
</tr>
<tr>
<td>0A58'</td>
<td>D4</td>
<td>C+</td>
<td></td>
</tr>
<tr>
<td>0A59'</td>
<td>00CB'</td>
<td>C+</td>
<td></td>
</tr>
<tr>
<td>0A5A'</td>
<td>03DA'</td>
<td>C+</td>
<td></td>
</tr>
<tr>
<td>0A5B'</td>
<td>9C</td>
<td>C+</td>
<td></td>
</tr>
<tr>
<td>0A5E'</td>
<td>CA 0939'</td>
<td>C+</td>
<td></td>
</tr>
<tr>
<td>0A61'</td>
<td>F8 OC</td>
<td>C</td>
<td>LDI LOW (CRC60)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>;SET RA TO POINT AT</td>
<td></td>
</tr>
<tr>
<td>0A63'</td>
<td>AA</td>
<td>C</td>
<td>PLO RA</td>
</tr>
<tr>
<td></td>
<td></td>
<td>;THE 16 BIT CRC</td>
<td></td>
</tr>
<tr>
<td>0A64'</td>
<td>F8 FF</td>
<td>C</td>
<td>LDI HIGH (CRC60)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>;CONSTANT</td>
<td></td>
</tr>
<tr>
<td>0A66'</td>
<td>EA</td>
<td>C</td>
<td>PHI RA</td>
</tr>
<tr>
<td>0A67'</td>
<td>F8 00</td>
<td>C</td>
<td>LDI 00H</td>
</tr>
<tr>
<td>0A69'</td>
<td>EA</td>
<td>C</td>
<td>SEX RA</td>
</tr>
<tr>
<td></td>
<td></td>
<td>;SET THE CRC CONSTANT</td>
<td></td>
</tr>
<tr>
<td>0A6A'</td>
<td>73</td>
<td>C</td>
<td>STXD RA</td>
</tr>
<tr>
<td></td>
<td></td>
<td>;TO ZERO</td>
<td></td>
</tr>
<tr>
<td>0A6B'</td>
<td>5A</td>
<td>C</td>
<td>STR RA</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0A6C'</td>
<td>D4</td>
<td>C+</td>
<td></td>
</tr>
<tr>
<td>0A6D'</td>
<td>023C'</td>
<td>C+</td>
<td></td>
</tr>
</tbody>
</table>

;The response to a PC is to calculate the CRC of a
;specified block of memory, beginning at a specified
;location. An erased block will cause the CLEAR
;to be typed.

CRC: TYPMSG RSC ;TYPE RC
INTERROGATOR CONTROL/DATA LOGGER (PIMNLG Mac) MACRO-18 3.36
WOODS HOLE OCEANOGRAPHIC INST. OCEAN ENGINEERING

OA6F'9C  CERROR?;AND BLOCK SIZE
OA70'CA 0939' C+  
OA73'F8 01' CLDI 01H;POINT TO SYSTEM FLAG
OA75'A7' C PLO R7  
OA76'E7' C SEX R7  
OA77'F8 80' CLDI 80H;SET THE CLEAR
OA79'F1' C OR;MEMORY FLAG
OA7A'57' C STR R7  
OA78'EA' C CLOOP;SEX RA;TEST A BYTE OF
OA7C'F8 FF' CLDI OFFH;MEMORY FOR THE
OA7F'F3' CXOR;CLEAR CONDITION
OA7F'C2 0A57' CLEZ CALCLOC;IF CLEAR, CALCULATE
OA82'E7' C SEX R7;OTHERWISE, RESET
OA83'F8 7F' CLDI 7FH;CLEAR MEMORY FLAG
OA85'F2' C AND
OA86'57' C STR R7  
OA87'CALCLOC; CALL CALCRC;CALCULATE CRC
OA87'D4' C+  
OA88'0253' C+  
OA8A'2B' CDEC RB;DECREMENT BYTE COUNT
OA8B'8B' CGLO RB;TEST FOR CRC
OA8C'CA 0A7B' CLNZ CLOOP;CALCULATION COMPLETE
OA8F'9B' CGHI RB;IF SO TEST FOR
OA90'CA 0A7B' CLNZ CLOOP;CLEAR MEMORY
OA93'O7' CLDN R7;GET SYSTEM FLAG
OA94'FE' CSHL;LOOK AT CLEAR MEMORY
OA95'CB 0A44' CLEAF CRCOUT;IF SET, TYPE THE
OA99'AC' CTYPMSG CLEAR;CLEAR MESSAGE AND
OA9A'D4' C+  
OA9B'OCB' C+  
OA9B'03F7' C+  
OA9D'9C' C+  
OA9E'CA 0939' C+  
OAAL'CO 08ED' C LBR FRMOUT;GET NEXT COMMAND
OAAL'CRCOUT; TYPMSG EQS;TYPE =
OAAL'D4' C+  
OAB0'00CB' C+  
OAB2'03F3' C+  
OAB4'9C' C+  
OAB8'CA 0939' C+  
OABD'F8 0B' CLDI LOW (CRCH);OTHERWISE, POINT
OABA'AA' CPLO RA;TO FINAL CRC
OABF'F8 FF' CLDI HIGH (CRCH);CONSTANT AND TYPE IT
OAB2'BA' CPHI RA  
OAB3'AA' CLDA RA;GET HI HALF OF CRC
OAB4'AC' CPLO RC  
OAB5'D4' C CALL TYPBC;TYPE IT
OAB6'021F' C+  
OABA'OA' CLN RA;GET LO HALF OF CRC
OAB9'AC' CPLO RC  
OABA'D4' C CALL TYPBC;TYPE IT
OABD'CO 08ED' C LBR FRMOUT;GET NEXT COMMAND
; The response to a SP command is to run a program
; beginning at a specified address. Prior to executing
; this command, the X and P registers will be set to RO

0AC0: F8 01 C
RUN: LDI O1H ;RESET THE SYSTEM
0AC2: A7 C
PLO K7 ;LOCK FLAG
0AC3: F8 FE C
LDI OFEH
0AC5: F2 C
AND
0AC6: 57 C
STR K7
0AC7: D4 C+ CALL GETHEX ;GET START ADDRESS
0AC8: 016D' C+ OAC0
0AC9: 9C C
gti RC ;REACT TO UART ERRORS
0ACB: FA FE C
ANI OFEH ;MASK NON-HEX FLAG
0ACD: CA 08F9' C LNZ ERROUT
0AEO: 8C C GLO RC ;LOCK FOR
0AE1: FB OD C XRI CR ;CARRIAGE RETURN
0AE3: CA 08F9' C LNZ ERROUT ;ERROR IF NOT FOUND
0AE5: 8B C GLO RB ;TRANSFER RUN ADDRESS
0AE7: A0 C PLO RO ;TO RO
0AE8: 9B C GHI RB
0AE9: B0 C PHI RO
0AEE: ED C ;SET X TO RO
0AEB: DO C LBR PRMOUT ;R3 LEFT POINTING HERE
0AEC: CO 08ED' C MOVE: GETFLG ;IS THE SYSTEM LOCKED ?
0AEF: F8 01 C+ OREF
0A1: A7 C+ OARE
0A2: 07 C+ SHA
0A3: F6 C
LEOF SPEC ;IF SO, TYPE THE ERROR
0A4: C3 OREA' C LBR NORM ;MESSAGE THEN PROMPT
0A5: CO 08C6' C FOR NEXT COMMAND
; SPEC: SEX K7 ;OTHERWISE, RESET THE
0A6: E7 C
OAE: F8 FE C LDI OFEH ;LOCK FLAG
0AED: F2 C AND
0AEE: 57 C STR K7
0AEF: D4 C+ TYPASS OVE ;PROMPT FOR SOURCE
0AF0: 00CB' C+ OAF
0AF2: 0444' C+ CALL PHIXN ;GET SOURCE ADDRESS
0AF4: 9C C+ OAF8: D4 C+ ERROR? ;LOOK FOR ERRORS
0AF9: 0165' C+ OAFB: 03DD' C
0AFD: 9C C+
INTERROGATOR CONTROL/DATA LOGGER (PINGLGR.MAC) MACRO-18 3.36
WOODS HOLE OCEANOGRAPHIC INST. OCEAN ENGINEERING

0AFE' CA 0939' C+ ; PLACE SOURCE ADDRESS
0B01' 8B C GLO RB
0B02' AA C PLO RA ; IN RA
0B03' 9B C GHI RB
0B04' BA C PHI RA

CALL FEXIN ; GET DESTINATION ADDRESS

0B05' D4 C+ ; LOOK FOR ERRORS
0B06' 01E6' C+ ERROR?
0B08' 03E5' C+ ; LOOK FOR ERRORS
0B0A' 9C C+ ; GET BLOCK SIZE
0B0B' CA 0939' C+ ; PLACE DESTINATION ADDRESS
0B0C' 9B C GHI RB ; IN RD
0B10' 9B C GHI RB
0B11' BD C PHI RD

CALL FEXIN ; ASK FINAL PERMISSION

0B12' D4 C+ ; LOOK FOR UART ERRORS
0B13' 01E6' C+ ERROR?
0B15' 03E5' C+ ; GET ANSWER
0B17' 9C C+ ; IF NOT YES EXIT
0B18' CA 0939' C+ ; GET A BYTE
0B1B' D4 C+ ; STORE IT
0B1C' 0113' C+ ; MOVE TO NEXT DESTINATION
0B1E' 9C C+ ; COUNT STORE OPERATION
0B1F' CA 0939' C+ ; TEST FOR BLOCK END
0B22' 8C C ; IF NOT AT END
0B23' CA 08ED' C ; CONTINUE, OTHERWISE

MOVIT: LDA RA ; GET NEXT COMMAND
0B26' 4A C+ ; IF NOT YES EXIT
0B27' 5D C STR RD
0B28' 1D C INC RD
0B29' 2B C DEC RB ; INCLUDE INTERVAL MAC
0B2A' 9B C GHI RB
0B2B' CA 0826' C LBNZ MOVIT
0B2E' 8B C GLO RB ; CONTINUE, OTHERWISE
0B2F' CA 0826' C LBNZ MOVIT
0B32' CO 08ED' C LBZ FRMOUT

INCLUDE INTERVAL MAC

************
* INTERVAL.MAC *
************

+ SET AND READ THE SOFTWARE CLOCK +
; The response to a ?T command is to first disable
; further interrupts, then advance the clock. This
; "future" time is then typed and a flag is set for
; the interrupt service routine prior to re-enabling
; interrupts and executing an idle instruction.
; This flag will cause the interrupt routine to quickly
; exit rather than advancing the clock. The interrupt
; re-activates the system, and an ' is typed as an
; immediate byte. This is the time tick.
; Define a byte of ram to be used as a flag word.
; Define the start of ASCII time message.

; Advance clock by one minute and convert this "future"
; time to an ASCII message string.

LDI  LOW  (TICK) ;POINT TO TICK FLAG
FLO  R7     ;USING REGISTER R7
LDI  01H     ;SET TICK FLAG
STR  R7     ;USING K7
CALL  MCLK  ;ADVANCE TIME BY 1 MIN.

DB  33H

LDI  LOW  (TICK+1)
FLO  RA     ;USING RA, POINT TO
LDI  RA     ;ASCII HUNDREDS OF DAYS
LDI  LOW  (HD) ;POINT AT "FUTURE" TIME
FLO  R7     ;USING R7
CALL  DTOA  ;CONVERT DAYS TO ASCII

DB  03H
LDI  SPACE  ;STORE A SPACE
STR  RA
INC  RA
CALL  DTOA  ;CONVERT HOURS TO ASCII

DB  02H
LDI  ";"    ;STORE A SEMI COLON
STR  RA
INC  RA
CALL  DTOA  ;CONVERT MINS. TO ASCII

DB  02H
LDI  STOP   ;STORE A STOP CHARACTER
| OB5F' | 5A | STR | RA | TYPMSG | SPSP | ;TYPE TWO SPACES |
| OB60' | D4 | C+  |     |        |      |                |
| OB61' | 00CB' | C+ |     |        |      |                |
| OB63' | 03D3' | C+ |     |        |      |                |
| OB65' | 9C  | C+  |     |        |      |                |
| OB66' | CA 0939' | C+ |     |        |      |                |
| OB69' | D4  | C+  |     | TYPMSG | NKTM | ;OUTPUT TIME AT NEXT @ |
| OB6A' | 00CB' | C+ |     |        |      |                |
| OB6C' | FF18 | C+ |     |        |      |                |
| OB6E' | 9C  | C+  |     |        |      |                |
| OB6F' | CA 0939' | C+ |     | TYPMSG | SECS |                |
| OB72' | D4  | C+  |     |        |      |                |
| OB73' | 00CB' | C+ |     |        |      |                |
| OB75' | 044F' | C+ |     |        |      |                |
| OB77' | 9C  | C+  |     |        |      |                |
| OB78' | CA 0939' | C+ |     |        |      |                |
| OB7B' | E3  | C   |     | SEX   | R3   | ;RESET INTERRUPT HARDWARE |
| OB7C' | 65  | C   |     | CUT   | CLINT |                |
| OB7D' | 00  | C   |     | DB    | OOH   |                |
| OB7E' | 70  | C   |     | RET   |       | ;ENABLE INTERRUPTS |
| OB7F' | 33  | C   |     | DB    | 33H   |                |
| OB80' | 00  | C   |     | IDL   |       | ;GO TO SLEEP    |
| OB81' | D4  | C+  |     | TYPMSG | AT   | ;SEND THE @     |
| OB82' | 00CB' | C+ |     |        |      |                |
| OB84' | 0457' | C+ |     |        |      |                |
| OB86' | 9C  | C+  |     |        |      |                |
| OB87' | CA 0939' | C+ |     |        |      |                |
| OB8A' | Cl 0F4B' | C |     | LDIQ | MRSSEQ | ;MAKE A MEASUREMENT OR |
| OB8D' | CO 08ED' | C |     | LBR   | PRMCUT | ;GET THE NEXT COMMAND |
| OB90' | E3  | C   |     | LUTIM | SEX | ;DISABLE INTERRUPTS |
| OB91' | 71  | C   |     | DIS   |      |                |
| OB92' | 33  | C   |     | DB    | 33H   | ;POINT AT TIME DATA |
| OB93' | F8 10 | C |     | LDIX  | LOW  (HD) |                |
| OB95' | AA  | C   |     | FLO   | RA    | ;SET THE DIGIT COUNTER |
| OB96' | F8 FF | C |     | LDIX  | HIGH  (HD) |                |
| OB98' | BA  | C   |     | FHI   | RA    | ;FOR SEVEN DIGITS |
| OB99' | F8 07 | C |     | LDIX  | 0TH   | ;STOP THE CLOCK  |
| OB9B' | AD  | C   |     | FLO   | RD    |                |
| OB9C' | E3  | C   | INTIM: | SEX | R3  |                |
| OB9D' | 64  | C   |     | OUT   | STCLK |                |
| OB9E' | 00  | C   |     | DB    | OOH   |                |
CALL INCHAR

; GET A CHARACTER

; MASK "#" BIT

; MASK FOR UART ERRORS

; BRANCH TO ERVEC IF

; FOUND, OTHERWISE

; IS IT A HEX NUMBER ?

; IF NOT, GET ANOTHER

; NUMBER.

; IS IT A DECIMAL NUMBER ?

; IF NOT, GET ANOTHER

; NUMBER. IF IT WAS A

; DECIMAL NUMBER, MOVE

; TO THE LEAST

; SIGNIFICANT HALF OF

; THE ACCUMULATOR AND

; STORE IT.

; POINT TO NEXT LOCATION

; COUNT THE OPERATION

; ENTERED SEVEN DIGITS YET ?

; IF NOT GET NEXT NUMBER

; OTHERWISE, STOP THE CLOCK

; AGAIN AND BEGIN LOOKING

; FOR @

; GET A CHARACTER

; IS IT AN "W" ?

; IF NOT KEEP LOOKING

; IF SO, DE-ADDRESS

; The response to an "R" command is to first test the system
; flag, then, if this flag is set, to prompt for a start address
; and block size. If the system flag was not set, exit and
; indicate an operator error since the memory was protected.
; After the start address and block size have been input, type
; "OK ? (Y/N)". If the operator types a "Y" in response to this
; question proceed with the RAM TEST. Load the entire specified
; block of RAM with a random number and verify a byte at a time.
: Type the address of each compare failure and its XOR data.
: Repeat the tests changing the random number with each pass.
: Program will exit upon detecting a UART error, but since
: interrupts have been disabled, the system MUST be reset.

OEED5' C+ RAMST: TYPMSG REMST ; TYPE "am test"
OEED5' D4 C+
OEED6' 00CB' C+
OEED8' 0596' C+
OEED' 9C C+
OEED' CA 0939' C+
OEED' F8 01 C+
OEED' A7 C+
OEED' 07 C+
OEED' F6 C+ SHR
OEED' C3 00CB' C+ LDRF RSPEC
OEED' CO 00CB' C+ LDR NONUN ; OTHERWISE, RESET THE
OEED' E7 C+ RSPEC: SEX R7 ; SYSTEM LOCK FLAG
OEED' F8 FE C+ LDI OFEH
OEED' F2 C+ AND
OEED' 57 C+ STR R7 ; THEN PROMPT FOR
OEED' CALL GET2EH ; START ADDRESS AND BLOCK SIZE
OEED' D4 C+
OEED' 023C' C+
OEED' 9C C+
OEED' CA 0939' C+
OEED' 9A C+ GHI RA ; SAVE START ADDRESS
OEED' B8 C+ PHI R8 ; USING REGISTER R8
OEED' 8A C+ GLO R8
OEED' A8 C+ FLO R8
OEED' 9B C+ GHI R8 ; SAVE BLOCK SIZE
OEED' B9 C+ PHI R9 ; USING REGISTER R9
OEED' 8B C+ GLO R9
OEED' 9C C+ FLO R9
OEED' CALL ASKOK ; ASK FINAL PERMISSION
OEED' D4 C+
OEED' 0113' C+
OEED' 9C C+
OEED' CA 0939' C+
OEED' 8C C+ GLO RC ; GET ANSWER
OEED' CA 08ED' C+ LENV PRMOUT ; EXIT IF NOT YES
OEED' TYPMSG CRLF ; OTHERWISE, TYPE A CR/LF
OEED' D4 C+
OEED' 00CB' C+
OEED' 03D0' C+
OEED' 9C C+
OEED' CA 0939' C+
OEED' E3 C+ SEX R3 ; DISABLE INTERRUPTS
OEED' 71 C+ DIS ; TO STOP THE CLOCK
OEED' 33 C+ DB 33H ; AND FALSE RAM ERRORS
OEED' F8 73 C+ LDI 73H ; AND SET RANDOM KEY =
OEED' EF C+ PHI RF ; 01110011
OEED' F8 0C' C+ LDI HIGH (LREGS)
OC19' R0 C PHI RO ;SET UP RO TO BE A
OC1A' F8 27' C LDI LOW (LREGS)
OC1C' A0 C FLO RO ;SUBROUTINE POINTER
OC1D' F8 0C' C LDI HIGH (RAND)
OC1E' BE C PHI RE ;SET UP RE TO BE A
OC20' F8 35' C LDI LOW (RAND)
OC22' AE C FLO RE ;SUBROUTINE POINTER
OC23' CO 0C5C' C LBR NCYCLE ;EXECUTE RAM TEST

;This is a subroutine which will load the next random
;key to the high half of register RD, the start address
;to register RA, and the block size to register RB.
;Using a subroutine here slows execution, but saves ROM.

OC26' D3 C LTOP: SEP R3 ;BACK TO RAM TEST
OC27' 9F C LREGS: GHI RF ;GET NEW KEY
OC28' 6D C PHI RD ;PASS TO RD
OC29' 98 C GHI R8 ;GET START ADDRESS
OC2A' 6A C PHI RA
OC2B' 88 C GLO R8 ;PASS TO RA
OC2C' AA C FLO RA
OC2D' 99 C GHI R9 ;GET BLOCK SIZE
OC2E' 6B C PHI RB
OC2F' 89 C GLO R9 ;PASS TO RB
OC30' AB C FLO RB
OC31' CO 0C26' C LBR LTOP ;RETURN

;This is a subroutine which will return with a random
;number in the high half of register RD. The random
;number is generated by right shifting the modulo 2 sum
;of bits 0, 2, 3, and 4 to bit 7.

OC34' D3 C RTOP: SEP R3 ;BACK TO RAM TEST
OC35' F8 00 C RAND: LDI 00H ;KEY IS IN RD HIGH
OC37' AD C FLO ED
OC38' 9D C GHI ED ;SET TO FFH IF KEY
OC39' CA 0C3F' C LBNZ OIIN00 ;IS NOW EQUAL TO 00
OC3C' F8 FF C LDI OFFH
OC3E' 6D C PHI ED
OC3F' F6 C OIIN00: SBR ;TEST BIT 1
OC40' CB 0C44' C LBNF OIIN01 ;ADD ONE IF SET
OC43' 1D C INC RD
OC44' F6 C OIIN01: SBR ;SKIP BIT 1
OC45' F6 C SBR ;TEST BIT 2
OC46' CB 0C4A' C LBNF OIIN02 ;ADD ONE IF SET
OC49' 1D C INC RD
OC4A' F6 C OIIN02: SBR ;TEST BIT 3
OC4B' CB 0C4F' C LBNF OIIN03 ;ADD ONE IF SET
OC4E' 1D C INC RD
OC4F' F6 C OIIN03: SBR ;TEST BIT 4
OC50' CB 0C54' C LBNF OIIN04 ;ADD ONE IF SET
OC53' 1D C INC RD
OC54' 6D C OIIN04: GLO RD ;GET RESULT OF SUM
OC55' F6 C SBR
OC56' 9D C GHI RD ;SHIFT IT TO RD HIGH
OC57' 76 C
OC58' BD C PHI RD
OC59' CO OC34' C LBR RTOP

OC5B' DO C NCYCLE: SEP RO
OC5D' DE C WRITE: SEP RE
OC5E' 9D C GHI RD
OC5F' 5A C STR RA

OC50' 1A C INC RA
OC51' 2B C DEC RB
OC52' 9B C GHI RB

OC53' CA OC5D' C LNZ WRITE
OC56' 8B C GLO RB
OC57' CA OC5D' C LNZ WRITE

OC6A' DO C VERIFY: SEP RO
OC6B' DE C VERCYC: SEP RE

OC6C' 9D C GHI RD
OC6D' EA C SEX RA
OC6E' F3 C XOR

OC6F' CA OC38' C LNZ WRERR
OC72' 1A C NEXTLOC: INC RA
OC73' 2B C DEC RB
OC74' 9B C GHI RB

OC75' CA OC6B' C LNZ VERCYC
OC78' 8B C GLO RB
OC79' CA OC6B' C LNZ VERCYC

OC7C' D4 C+ TYPMSG ASTK
OC7D' 00CB' C+ TYPMSG CRLESP
OC7F' 0594' C+ TYPMSG CRLESP
OC81' 9C C+ TYPMSG CRLESP
OC82' CA 0939' C+ TYPMSG CRLESP

OC85' 9F C GHI RF
OC86' BD C PHI RD
OC87' DE C SEP RE
OC88' 9D C GHI RD
OC89' BF C PHI RF

OC8A' CO OC5C' C WRERR: LBR NCYCLE
OC8D' AD C SAVE RESULT OF XOR
OC8E' D4 C+ MOVE TO NEXT LINE
OC8F' 00CB' C+ MOVE TO NEXT LINE
OC91' 0366' C+ MOVE TO NEXT LINE
OC93' 9C C+ MOVE TO NEXT LINE
OC94' CA 0939' C+ MOVE TO NEXT LINE

OC97' 9A C GHI RA
OC98' AC C PLO RC
OC99' D4 C+ CALL TYPEC
OC9A' 021F' C+ CALL TYPEC

OC9C' 9C C ERROR?
OC9D' CA 0939' C+ REACT TO UART ERRORS
OC9E' 8A C GLO RA
OC9F' AC C PLO RC
OCA0' AC C CALL TYPEC
OCA1' AC C CALL TYPEC
OCA2' D4 C+
INTERROGATOR CONTROL/DATA LOGGER (PNAVLGR.MAC) MACRO-18 3.36
Woods Hole Oceanographic Inst. Ocean Engineering

OCA3' 021F' C+ ERROR? ;REACT TO ERRORS
OCA5' 9C C+ C
OCA6' C9 0939' C+ C
OCA9' D4 C+ C
OCA' 00CB' C+ C
OCA' 03D3' C+ C
OCAE' 9C C+ C
OCF' CA 0939' C+ C
OC8' 8D C GLO RD ;THE RESULT OF THE XOR
OCB' AC C PLO RC C
OCB4' D4 C+ C
OCB5' 021F' C+ C
OCB7' 9C C+ C
OCB8' CA 0939' C+ C
OCB' CO 0C72' C LBR NOTLOC ;TEST THE NEXT LOCATION
;
;
; INCLUDE ISCEDUL.MAC
;
;
***************
C C
***************
;
;
; + SET THE INTERROGATOR SCHEDULE +
;
;
; This block of code will set the operating schedule
; of the interrogator. Two parameters are set via prompts,
; the start time, and the measurement interval.
;
; Define decimal and ASCII start times in RAM
;
FF30  C DSID EQU GLOBAL+30H
FF36  C DSUM EQU DSID+06H
FF38  C ASID EQU GLOBAL+38H
FF3C  C ASTH EQU ASID+04H
FF3F  C ASTM EQU ASTH+03H
FF3E  C ASUM EQU ASUM+06H
;
; Define decimal and ASCII measurement interval in RAM
;
FF45  C DTIM EQU GLOBAL+45H
FF4A  C ATIM EQU DTIM+05H
;
; Define a location in ram to hold the hex equivalent
; of the measurement interval.
;
;
FF24  C HEMI EQU GLOBAL+24H
;
; Define another location for the number of minutes
; in hex to the next measurement. This number is only
;valid if the schedule is active.

OCB7' F8 36
OCB0' A7
OCB1' E7
OCB2' F8 00
OCB4' 73
OCB5' 73
OCB6' 73
OCB7' 73
OCB8' 73
OCB9' 73
OCB1' 57
OCB2' F8 43
OCB3' A7
OCB4' F8 00
OCB5' 57
OCB6' 17
OCB7' 57
OCB3' D4
OCB4' 00CB'
OCB6' 0465'
OCB8' 9C
OCB9' CA 0939'
OCB0' D4
OCB1' 0196'
OCB2' FF32
OCB3' 03
OCB4' 9C
OCB5' FA FE
OCB6' CA 0939'
OCB7' 8C
OCB8' FB 20
OCB9' CA 08F9'
OCB3' D4
OCB4' 00CB'
OCB1' 0478'
OCB3' 9C
OCB4' CA 0939'
OCB0' D4
OCB1' 0196'
OCB2' FF34
OCB3' 02
OCB4' 9C
OCB5' FA FE
OCB6' CA 0939'
OCB7' 8C
OCB8' FB 20
OCB9' CA 08F9'
OCB3' D4
OCB4' 0196'
OCB1' FF34
OCB3' 02
OCB4' 9C
OCB5' FA FE
OCB6' CA 0939'
OCB7' 8C
OCB8' FB 20
OCB9' CA 08F9'

OCB7' F8 36
OCB0' A7
OCB1' E7
OCB2' F8 00
OCB4' 73
OCB5' 73
OCB6' 73
OCB7' 73
OCB8' 73
OCB9' 73
OCB1' 57
OCB2' F8 43
OCB3' A7
OCB4' F8 00
OCB5' 57
OCB6' 17
OCB7' 57
OCB3' D4
OCB4' 00CB'
OCB6' 0465'
OCB8' 9C
OCB9' CA 0939'
OCB0' D4
OCB1' 0196'
OCB2' FF32
OCB3' 03
OCB4' 9C
OCB5' FA FE
OCB6' CA 0939'
OCB7' 8C
OCB8' FB 20
OCB9' CA 08F9'
OCB3' D4
OCB4' 00CB'
OCB1' 0478'
OCB3' 9C
OCB4' CA 0939'
OCB0' D4
OCB1' 0196'
OCB2' FF34
OCB3' 02
OCB4' 9C
OCB5' FA FE
OCB6' CA 0939'
OCB7' 8C
OCB8' FB 20
OCB9' CA 08F9'

;POINT AT START TIME
;AND LOAD ZEROS
;TO BOTH HALVES OF
;OF THE GO FLAG
;PROMPT FOR START DAY
;GET START DAY AND
;STORE
;(THREE DIGITS)
;LOOK FOR ERRORS
;MASK NON-DECIMAL BIT
;EXIT ON ERROR
;TEST FOR SPACE
;CONTINUE IF FOUND
;OTHERWISE, INDICATE ERROR
;PROMPT FOR START HOUR
;GET START HOUR AND
;STORE
;(TWO DIGITS)
;LOOK FOR UART ERRORS
;BY MASKING NON-DECIMAL
;EXIT IF FOUND
;LOOK FOR A SPACE
;CONTINUE IF FOUND
;OTHERWISE INDICATE ERROR
;PROMPT FOR START MINUTE
Interrogator Control/Data Logger (PNVLGR..MAC) Macro-18 3.36 PAGE 1-50
Woods Hole Oceanographic Inst. Ocean Engineering

OD09' D4 C+ CALL INDEC ;GET START MINUTE AND
OD0A' 00CB' C+ ;STORE
OD0C' 0482' C+ ;TWO DIGITS
OD0E' 9C C+ ;LOOK FOR UART ERRORS
ODOF' CA 0939' C+ ;MASK NON-DECIMAL FLAG
OD12' D4 C+ ;EXIT ON ERROR
OD13' 0166' C+ ;LOOK FOR
OD15' 0F36 C ;SPACE
OD17' 02 C ;A SPACE
OD18' 9C C ;IF FOUND CONTINUE
OD1B' CA 0939' C ;OTHERWISE LOOK FOR A
OD1E' 8C C ;CARRIAGE RETURN
OD21' 0F2A' C ;ERROR IF NOT FOUND
OD24' 08 C
OD25' FB OD C
OD27' CA 08F9' C

;Start date and time are now in RAM. Measurement
;interval is next prompted for, input, converted
;to hex, and stored in two locations.

;INMIN: TYPMSG MEINT ;PROMPT FOR MEAS. INT.

OD2A' D4 C+ CALL INDEC ;GET MEASUREMENT INTERVAL
OD2B' 00CB' C+
OD2D' 048E' C+
OD2F' 9C C+
OD30' CA 0939' C+

OD33' D4 C+ CALL INDEC ;AND STORE
OD34' 0166' C+ ;(THREE DIGITS)
OD36' FF47 C ;LOOK FOR UART ERRORS
OD38' 03 C ;AND EXIT IF FOUND
OD39' 9C C ;INDICATE AN ERROR
OD3A' FA FE C ;LOOK FOR A SPACE
OD3C' CA 0939' C ;CONT. IF FOUND, OTHERWISE
OD3E' 8C C ;CONTINUE IF FOUND
OD40' FB 20 C ;OTHERWISE,;
OD42' C2 0DAE' C ;CARRIAGE RETURN
OD45' 8C C ;BCDHEX
OD46' FB OD C ;CARRIAGE RETURN
OD48' C2 0DAE' C ;INDICATE AN OPERATOR ERROR
OD4A' CO 08F9' C ;POINT AT DEC. INT. U.M.
OD4E' F7 C BCDHEX: SEX R7
OD4F' F8 47 C ;POINT AT DEC. INT. U.M.

0D51' A7 C LEI LOW (DIMP+2)
0D52' F8 00 C LEI 00H ;CONVERT MEASUREMENT
0D54' AA C LEI 00H ;INTERVAL TO HEX
0D55' BA C PLO RA ;ZERO REGISTER RA
0D56' 07 C PLO RA ;GET UNITS DIGIT
0D57' AA C PLO RA ;POINT AT TENS DIGIT
0D58' 27 C PLO RA ;SET ADD COUNTED
0D59' 07 C PLO RC ;AND TEST FOR ZERO
0D5A' AC C PLO RC

91
ODA9' CA 0939' C+  C+  TYPMSG OK?
ODAC' D4  C+  C+  
ODAD' 00CB' C+  C+  
ODAF' 0423' C+  C+  
ODAI' 9C  C+  C+  
ODAJ' CA 0939' C+  C+  
ODAJ' CHAR?  ;GET RESPONSE
ODAB' D4  C+  C+  
ODAC' 0145' C+  C+  
ODAD' 9C  C+  C+  
ODAE' CA 0939' C+  C+  
ODAF' 8C  C+  C+  
ODAG' FB 59  C  XRI "t"  ;IS IT YES ?
ODAH' CA 00CB' C  LENZ DEARM  ;IF NOT EXIT
ODAI' F8 43  C  LDI LOW (GOFILG) ;OTHERWISE, ARM
ODAJ' A7  C  FLO R7  ;THE SCHEDULER
ODAK' F8 AA  C  LDI OBAH  ;BY SETTING HI HALF
ODAL' 57  C  STR R7  ;GO FLAG THEN
ODAM' C0 08ED' C  LDR PRNOUT  ;GET NEXT COMMAND
ODAN' F8 43  C  DEARM: LDI LOW (GOFILG) ;ANSWER WAS NOT
ODAO' A7  C  FLO R7  ;YES, SO RESET
ODAP' F8 00  C  LDI OCH  ;GO FLAG AND EXIT
ODAQ' 57  C  STR R7  
ODAR' 17  C  INC R7  
ODAS' 57  C  STR R7  
ODAT' C0 08ED' C  LDR PRNOUT  
ODAU'  ;The response to a ?S command is to convert the
ODAV'  ;start time and transmission interval to an ASCII
ODAW'  ;message string, type the message, and return to CMD.
ODAX'  
ODAY' F8 30  C  QRYSCED:LDI LOW (DSHD) ;CONVERT START TIME
ODAZ' A7  C  FLO R7  ;TO ASCII
ODA0' F8 38  C  LDI LOW (ASHD) ;STORE AT ASHD
ODA1' AA  C  FLO RA  ;USING RA AS A POINTER
ODA2' 97  C  GHI R7  
ODA3' RA  C  PHI RA  
ODA4' D4  C+  C+  CALL DTOA  ;CONVERT DAYS
ODA5' 0095' C+  C+  
ODA6' 03  C  DB 03H  ;STORE A STOP
ODA7' F8 7E  C  LDI STOP  ;STORE A STOP
ODA8' 5A  C  STR RA  ;BETWEEN DAYS AND
ODA9' 1A  C  INC RA  ;HOURS
ODA0' D4  C+  C+  CALL DTOA  ;CONVERT HOURS
ODA1' 0095' C+  C+  
ODA2' 02  C  DB 02H  ;STORE A STOP
ODA3' F8 7E  C  LDI STOP  ;STORE A STOP
ODA4' 5A  C  STR RA  ;BETWEEN HOURS
ODA5' 1A  C  INC RA  ;AND MINUTES
ODA6' D4  C+  C+  CALL DTOA  ;CONVERT MINUTES
ODA7' 0095' C+  C+  
ODA8' 02  C  DB 02H  

93
<table>
<thead>
<tr>
<th>Location</th>
<th>Code</th>
<th>Operation</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>01F2'</td>
<td>F8 7E</td>
<td>LDI STOP</td>
<td>;STORE A STOP</td>
</tr>
<tr>
<td>01F4'</td>
<td>5A</td>
<td>STR RA</td>
<td>;BETWEEN MINUTES</td>
</tr>
<tr>
<td>01F5'</td>
<td>1A</td>
<td>INC RA</td>
<td>;AND TRAN. INT.</td>
</tr>
<tr>
<td>01F6'</td>
<td>F8 45</td>
<td>LDI LOW (DIHM)</td>
<td>;CONVERT TRANSM. INT.</td>
</tr>
<tr>
<td>01F8'</td>
<td>A7</td>
<td>FLO R7</td>
<td>;AND STORE AT A1BM</td>
</tr>
<tr>
<td>01F9'</td>
<td>F8 4A</td>
<td>LDI LOW (A1BM)</td>
<td>;USING RA AS A POINTER</td>
</tr>
<tr>
<td>01FB'</td>
<td>A4</td>
<td>FLO RA</td>
<td></td>
</tr>
<tr>
<td>01FC'</td>
<td>D4</td>
<td>CALL DTOA</td>
<td>;CONVERT TRANSMISSION</td>
</tr>
<tr>
<td>01FD'</td>
<td>0095'</td>
<td>C+</td>
<td></td>
</tr>
<tr>
<td>01FF'</td>
<td>03</td>
<td>DB 03H</td>
<td>;INTERVAL TO ASCII</td>
</tr>
<tr>
<td>0200'</td>
<td>F8 7E</td>
<td>LDI STOP</td>
<td>;STORE MESSAGE</td>
</tr>
<tr>
<td>0202'</td>
<td>5A</td>
<td>STR RA</td>
<td>;TERMINATION CHARACTER.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>;Type current time.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>;</td>
</tr>
<tr>
<td>0203'</td>
<td>D4</td>
<td>C+</td>
<td></td>
</tr>
<tr>
<td>0204'</td>
<td>00CB'</td>
<td>C+</td>
<td></td>
</tr>
<tr>
<td>0206'</td>
<td>03D3'</td>
<td>C+</td>
<td></td>
</tr>
<tr>
<td>0208'</td>
<td>9C</td>
<td>C+</td>
<td></td>
</tr>
<tr>
<td>0209'</td>
<td>CA 0939'</td>
<td>C+</td>
<td></td>
</tr>
<tr>
<td>020C'</td>
<td>F8 18</td>
<td>C</td>
<td></td>
</tr>
<tr>
<td>020E'</td>
<td>A4</td>
<td>LDI LOW (TICK+1)</td>
<td>;CONVERT TIME TO ASCII</td>
</tr>
<tr>
<td>020F'</td>
<td>97</td>
<td>GHL R7</td>
<td>;USING RA AS A POINTER</td>
</tr>
<tr>
<td>0210'</td>
<td>BA</td>
<td>PHI RA</td>
<td></td>
</tr>
<tr>
<td>0211'</td>
<td>F8 10</td>
<td>C</td>
<td></td>
</tr>
<tr>
<td>0213'</td>
<td>A7</td>
<td>LDI LOW (BD)</td>
<td></td>
</tr>
<tr>
<td>0214'</td>
<td>D4</td>
<td>C+</td>
<td></td>
</tr>
<tr>
<td>0215'</td>
<td>0095'</td>
<td>C+</td>
<td></td>
</tr>
<tr>
<td>0217'</td>
<td>03</td>
<td>DB 03H</td>
<td></td>
</tr>
<tr>
<td>0218'</td>
<td>F8 20</td>
<td>C</td>
<td></td>
</tr>
<tr>
<td>021A'</td>
<td>5A</td>
<td>STR RA</td>
<td></td>
</tr>
<tr>
<td>021B'</td>
<td>1A</td>
<td>INC RA</td>
<td></td>
</tr>
<tr>
<td>021C'</td>
<td>D4</td>
<td>C+</td>
<td></td>
</tr>
<tr>
<td>021D'</td>
<td>0095'</td>
<td>C+</td>
<td></td>
</tr>
<tr>
<td>021F'</td>
<td>02</td>
<td>DB 02H</td>
<td></td>
</tr>
<tr>
<td>0220'</td>
<td>F8 3A</td>
<td>C</td>
<td></td>
</tr>
<tr>
<td>0222'</td>
<td>5A</td>
<td>STR RA</td>
<td></td>
</tr>
<tr>
<td>0223'</td>
<td>1A</td>
<td>INC RA</td>
<td></td>
</tr>
<tr>
<td>0224'</td>
<td>D4</td>
<td>C+</td>
<td></td>
</tr>
<tr>
<td>0225'</td>
<td>0095'</td>
<td>C+</td>
<td></td>
</tr>
<tr>
<td>0227'</td>
<td>02</td>
<td>DB 02H</td>
<td></td>
</tr>
<tr>
<td>0228'</td>
<td>F8 7E</td>
<td>C</td>
<td></td>
</tr>
<tr>
<td>022A'</td>
<td>5A</td>
<td>STR RA</td>
<td></td>
</tr>
<tr>
<td>022B'</td>
<td>D4</td>
<td>C+</td>
<td></td>
</tr>
<tr>
<td>022C'</td>
<td>00CB'</td>
<td>C+</td>
<td></td>
</tr>
<tr>
<td>022E'</td>
<td>058C'</td>
<td>C+</td>
<td></td>
</tr>
<tr>
<td>0230'</td>
<td>9C</td>
<td>C+</td>
<td></td>
</tr>
<tr>
<td>0231'</td>
<td>CA 0939'</td>
<td>C+</td>
<td></td>
</tr>
<tr>
<td>0234'</td>
<td>D4</td>
<td>C+</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>TYPMSG RTE</td>
<td>;TYPE TIME</td>
</tr>
<tr>
<td>Code</td>
<td>Value</td>
<td>Type</td>
<td></td>
</tr>
<tr>
<td>-------</td>
<td>--------</td>
<td>------</td>
<td></td>
</tr>
<tr>
<td>0E35'</td>
<td>00CB'</td>
<td>C+</td>
<td></td>
</tr>
<tr>
<td>0E37'</td>
<td>FF18</td>
<td>C+</td>
<td></td>
</tr>
<tr>
<td>0E39'</td>
<td>9C</td>
<td>C+</td>
<td></td>
</tr>
<tr>
<td>0E3A'</td>
<td>CA 0939'</td>
<td>C+</td>
<td></td>
</tr>
<tr>
<td>0E3D'</td>
<td>D4</td>
<td>C+</td>
<td></td>
</tr>
<tr>
<td>0E3E'</td>
<td>00CB'</td>
<td>C+</td>
<td></td>
</tr>
<tr>
<td>0E40'</td>
<td>0465'</td>
<td>C+</td>
<td></td>
</tr>
<tr>
<td>0E42'</td>
<td>9C</td>
<td>C+</td>
<td></td>
</tr>
<tr>
<td>0E43'</td>
<td>CA 0939'</td>
<td>C+</td>
<td></td>
</tr>
<tr>
<td>0E46'</td>
<td>D4</td>
<td>C+</td>
<td></td>
</tr>
<tr>
<td>0E47'</td>
<td>00CB'</td>
<td>C+</td>
<td></td>
</tr>
<tr>
<td>0E49'</td>
<td>FF38</td>
<td>C+</td>
<td></td>
</tr>
<tr>
<td>0E4B'</td>
<td>9C</td>
<td>C+</td>
<td></td>
</tr>
<tr>
<td>0E4C'</td>
<td>CA 0939'</td>
<td>C+</td>
<td></td>
</tr>
<tr>
<td>0E4F'</td>
<td>D4</td>
<td>C+</td>
<td></td>
</tr>
<tr>
<td>0E50'</td>
<td>00CB'</td>
<td>C+</td>
<td></td>
</tr>
<tr>
<td>0E52'</td>
<td>0478'</td>
<td>C+</td>
<td></td>
</tr>
<tr>
<td>0E54'</td>
<td>9C</td>
<td>C+</td>
<td></td>
</tr>
<tr>
<td>0E55'</td>
<td>CA 0939'</td>
<td>C+</td>
<td></td>
</tr>
<tr>
<td>0E58'</td>
<td>D4</td>
<td>C+</td>
<td></td>
</tr>
<tr>
<td>0E59'</td>
<td>00CB'</td>
<td>C+</td>
<td></td>
</tr>
<tr>
<td>0E5B'</td>
<td>FF3C</td>
<td>C+</td>
<td></td>
</tr>
<tr>
<td>0E5D'</td>
<td>9C</td>
<td>C+</td>
<td></td>
</tr>
<tr>
<td>0E5E'</td>
<td>CA 0939'</td>
<td>C+</td>
<td></td>
</tr>
<tr>
<td>0E61'</td>
<td>D4</td>
<td>C+</td>
<td></td>
</tr>
<tr>
<td>0E62'</td>
<td>00CB'</td>
<td>C+</td>
<td></td>
</tr>
<tr>
<td>0E64'</td>
<td>0482'</td>
<td>C+</td>
<td></td>
</tr>
<tr>
<td>0E66'</td>
<td>9C</td>
<td>C+</td>
<td></td>
</tr>
<tr>
<td>0E67'</td>
<td>CA 0939'</td>
<td>C+</td>
<td></td>
</tr>
<tr>
<td>0E6A'</td>
<td>D4</td>
<td>C+</td>
<td></td>
</tr>
<tr>
<td>0E6B'</td>
<td>00CB'</td>
<td>C+</td>
<td></td>
</tr>
<tr>
<td>0E6D'</td>
<td>FF3F</td>
<td>C+</td>
<td></td>
</tr>
<tr>
<td>0E6F'</td>
<td>9C</td>
<td>C+</td>
<td></td>
</tr>
<tr>
<td>0E70'</td>
<td>CA 0939'</td>
<td>C+</td>
<td></td>
</tr>
<tr>
<td>0E73'</td>
<td>D4</td>
<td>C+</td>
<td></td>
</tr>
<tr>
<td>0E74'</td>
<td>00CB'</td>
<td>C+</td>
<td></td>
</tr>
<tr>
<td>0E76'</td>
<td>048E'</td>
<td>C+</td>
<td></td>
</tr>
<tr>
<td>0E78'</td>
<td>9C</td>
<td>C+</td>
<td></td>
</tr>
<tr>
<td>0E79'</td>
<td>CA 0939'</td>
<td>C+</td>
<td></td>
</tr>
<tr>
<td>0E7C'</td>
<td>D4</td>
<td>C+</td>
<td></td>
</tr>
<tr>
<td>0E7D'</td>
<td>00CB'</td>
<td>C+</td>
<td></td>
</tr>
<tr>
<td>0E7F'</td>
<td>FF4A</td>
<td>C+</td>
<td></td>
</tr>
<tr>
<td>0E81'</td>
<td>9C</td>
<td>C+</td>
<td></td>
</tr>
<tr>
<td>0E82'</td>
<td>CA 0939'</td>
<td>C+</td>
<td></td>
</tr>
<tr>
<td>0E85'</td>
<td>D4</td>
<td>C+</td>
<td></td>
</tr>
<tr>
<td>0E86'</td>
<td>00CB'</td>
<td>C+</td>
<td></td>
</tr>
<tr>
<td>0E88'</td>
<td>04B3'</td>
<td>C+</td>
<td></td>
</tr>
</tbody>
</table>

95
```
OE8A' 9C  C+  LDI  LOW  (GOFLG)
OE8B' CA 0939' C+  PLO  R7  ; IF GO FLAG IS SET
OE8C' F8 43  C  LDA  R7  ; SAY ACTIVE, AND
OE8D' 47  C  XRI  OAH  ; INDICATE THE NUMBER
OE8E' CA 09F9' C  LENZ  SNRM  ; OF MINUTES TO THE NEXT
OE8F' 07  C  LEN  R7  ; MEASUREMENT, OTHERWISE
OE8G' FB AA  C  XRI  OAH  ; TYPE THE CURRENT
OE8H' CA 0F05' C  LENZ  SARM  ; SYSTEM STATUS AND EXIT
    C  TYPMSG  ACTIVE
OE8I' D4  C+  
OE8J' 00CB' C+  
OE8K' 04C4' C+  
OE8L' 9C  C+  
OE8M' CA 0939' C+  
OE8N' F8 26  C  LDI  LOW  (MINOW)
OE8O' A7  C  PLO  R7  ; GET HEX MINOW
OE8P' 47  C  LDA  R7  ; AND PLACE IN RA
OE8Q' BA  C  PHI  RA  
OE8R' 07  C  LEN  R7  ; PLACE IN RA
OE8S' AA  C  PLO  RA  
OE8T' 9A  C  GHI  RA  
OE8U' AC  C  PLO  RC  
OE8V' D4  C+  CALL  TYPSC  ; TYPE HI BYTE
OE8W' 021F' C+  
OE8X' 9C  C+  
OE8Y' CA 0939' C+  
OE8Z' 8A  C  GLO  RA  ; GET HEX MINOW LO
OE8AA' AC  C  PLO  RC  
OE8AB' D4  C+  CALL  TYPSC  ; TYPE LO BYTE
OE8AC' 021F' C+  
OE8AD' 9C  C+  
OE8AE' CA 0939' C+  
OE8AF' D4  C+  TYPMSG  MINSEM
OE8AG' 00CB' C+  
OE8AH' 04D1' C+  
OE8AI' 9C  C+  
OE8AJ' CA 0939' C+  
OE8AK' D4  C+  TYPMSG  PMR  ; INDICATE POINTER LOCATION
OE8AL' 00CB' C+  
OE8AM' 0504' C+  
OE8AN' 9C  C+  
OE8AO' CA 0939' C+  
OE8AP' F8 0E  C  LDI  LOW  (STRADD)
OE8AQ' A7  C  PLO  R7  ; GET CURRENT STORE ADDRESS
OE8AR' 47  C  LDA  R7  
OE8AS' BA  C  PHI  RA  
OE8AT' 07  C  LEN  R7  
OE8AU' AA  C  PLO  RA  
```
CRI  RA
PLO  RC
CALL  TYPEC ;TYPE HI BYTE

ERROR? ;REACT TO ERRORS
GLO  RA
PLO  RC
CALL  TYPEC ;TYPE LO BYTE

ERROR? ;REACT TO ERRORS

GROUP: TYPMSG IDLE1 ;GET NEXT COMMAND
GO FLAG NOT SET

SAY IDLE AND EXIT
TYPMSG NOTARM ;SAY NOT ARMED

SAY ARMED BUT IDLE

LBR  PRMOUT ;AND EXIT

;The response to a "!IDLE" command is to test the go
;flag and if set, reset it. If the go flag is already
;reset, the message "Scheduler was NOT active !!" will be
;sent.

GOFLAG:  LDY  LOW (GOFLAG)

PLO  R7 ;GET GO FLAG
LEN  R7 ;IS IT SET ?
AXI  GRAH ;IF SO RESET IT
LENZ  SAINOT ;OTHERWISE TYPE
LDI  OGH ;NOT ACTIVE MESSAGE
STR  R7 ;RESET HI AND LO
INC  R7 ;OF GO FLAG
STR  R7 ;THEN
TYPMSG OK ;SAY OK AND GET

CRI  RA
PLO  RC
CALL  TYPEC

HELP? ;REACT TO ERRORS
INTERROGATOR CONTROL/DATA LOGGER (PNAVGLR.MAC) MACRO-18 3.36
WODES HOLE OCEANOGRAPHIC INST. OCEAN ENGINEERING

OF22' 041F' C+ OF24' 9C C+ OF25' CA 0939' C+ OF27' C0 08ED' C SAYNOT: TYPMSG NOTACT ;FLAG WAS NOT SET
OF28' C+ OF2C' 00CB' C+ OF2E' 0516' C+ OF30' 9C C+ OF31' CA 0939' C+ OF34' C0 08ED' C LBR PRMOUT ;GET NEXT COMMAND

;The response to a "PING" command is to first ask "OK?"
;and if a "Y" is the answer to trigger the ping. Any other
;answer will cause an exit to CMD.

OF37' C+ OF37' D4 C+ OF38' 0113' C ERROR? ;REACT TO UART ERRORS
OF3A' 9C C+ OF3B' CA 0939' C+ OF3F' 8C C GLO RC ;IS IT YES?
OF3F' C0 08ED' C LENZ PRMOUT ;IF NOT EXIT
OF42' E3 C SAI R3
OF43' 63 C OUT PING ;SEND PING
OF44' 00 C DB OOH
OF45' C0 08ED' C LBR PRMOUT ;GET NEXT COMMAND

; INCLUDE MAIN.MAC
; ******************
; * MAIN.MAC *
; ******************
;
;+ THIS IS THE INTERROGATOR MAIN PROGRAM +
;
; Define the locations in RAM which hold the current
; data address.
;
FFOE C STRADD EQU GLOBAL+OEH ;STORE ADDRESS POINTER
;
OF48' C9 OFF7' C MAIN: LENZ SHTDN ;SHUT DOWN IF NO Q
;
;This is the measurement sequence. Since it is approximately
;one minute before the PING, enable interrupts to keep the
;clock running and stop processing for one minute.
;
OF4B' 7A C SSSEQ: REQ ;INSURE THAT Q IS RESET
OF4C' F8 17 C LDI LOW (TICK) ;RESET THE TICK FLAG
OF4E' A7 C LDI R07
OF4F' F8 00 C LDI OOH

98
OE51'  57  C  STR  R7
OE52'  E3  C  SEX  R3
OE53'  65  C  OUT  CLAIM ;RESET INTERRUPT
OE54'  00  C  DB  00H ;HARDWARE AND INSURE
OE55'  86 2' C  LDI  LOW  (INTRPT)
OE57'  A1  C  PLO  R1 ;THAT R1 IS POINTING
OE58'  5B  C  LDI  HIGH  (INTRPT)
OE5A'  B1  C  PHI  R1 ;AT INTERRUPT BEFORE...
OE5B'  70  C  RET  ;INTERRUPTS ARE ENABLED.
OE5C'  33  C  DB  33H ;CLOCK CAL. CONSTANT IS
OE5D'  D4  C+  CALL  DELAY
OE5E'  00A2'  C+  ;RESTORE MEASUREMENT INTERVAL COUNTER TO ITS ORIGINAL VALUE
OE60'  067D  C  DW  067DH ;200 ms. (OSC. START UP)
OE62'  00  C  ;WAIT ONE MINUTE
OE63'  8F 26  C  RSTMI:  LDI  LOW  (MINOW)
OE65'  A7  C  PLO  R7
OE66'  8F 24  C  LDI  LOW  (HEXMI)
OE68'  AA  C  PLO  RA  ;POINT AT MEAS. INT.
OE69'  97  C  GHI  R7  ;USING RA A
OE6A'  BA  C  PHI  RA  ;THE POINTER
OE6B'  4A  C  LDA  RA  ;GET OLD VALUE
OE6C'  57  C  STR  R7  ;AND DUPLICATE IT
OE6D'  17  C  INC  R7  ;AT MINOW
OE6E'  0A  C  LDN  RA
OE6F'  57  C  STR  R7

;CONVERT CURRENT TIME TO TIME CODE AND STORE.
OE70'  8F 14  C  LDI  LOW  (HD+4) ;POINT AT UNITS OF HOURS
OE72'  A7  C  PLO  R7
OE73'  07  C  LDN  R7  ;GET UNITS OF HOURS
OE74'  AB  C  PLO  RB  ;SHIFT 4 BITS TO RA
OE75'  D4  C+  CALL  8582A
OE76'  00B7'  C+  ;SHIFT 2 BITS TO RA
OE78'  04  C  DB  04H  ;GET TENS OF HOURS
OE79'  27  C  DEC  R7
OE7A'  07  C  LDN  R7
OE7B'  AB  C  PLO  RA
OE7C'  D4  C+  CALL  8582A
OE7D'  00B7'  C+  ;SHIFT 4 BITS TO RA
OE7E'  02  C  DB  02H  ;GET UNITS OF DAYS
OE80'  27  C  DEC  R7
OE81'  07  C  LDN  R7
OE82'  AB  C  PLO  RB
OE83'  D4  C+  CALL  8582A
OE84'  00B7'  C+  ;SHIFT 4 BITS TO RA
OE86'  04  C  DB  04H  ;GET TENS OF DAYS
OE87'  27  C  DEC  R7
OE88'  07  C  LDN  R7
```
INTERROGATOR CONTROL/DATA LOGGER (PRMLGR.MAC) MACRO-18 3.36
WOODS HOLE OCEANOGRAPHIC INST. OCEAN ENGINEERING

OF89' AB C PLO RB ;SHIFT 4 BITS TO RA
OF8B' D4 C+ CALL RSB2A
OF8B' 0087' C+ 3
OF8D' 04 C DB 04H 3
OF8E' 27 C DEC R7 ;GET HUNDREDS OF DAYS
OF8F' 07 C LIN R7 3
OF90' AB C PLO RB
OF91' D4 C+ CALL RSB2A ;SHIFT 2 BITS TO RA
OF92' 0087' C+ 3
OF94' 02 C DB 02H 3
OF95' 8A C GLO RA ;GET LOW BYTE
OF96' AF C PLO RF ;SAVE IT
OF97' 9A C GHI RA ;GET HI BYTE
OF98' BF C PHI RF ;SAVE IT

;This is the measurement sequence. RA, RB, and RC are
;used as travel time counters for Fl, F2, and F3. RD is
;used as a time out counter. The measurement sequence will
;terminate when a reply from all three transponders has been
;received, or RD rolls over to 0000. Since the counters are
;incremented at a 4 kHz rate, the maximum measurement time
;will not exceed 16.4 seconds.

OF99' F8 00 C SNDFMG: LDI 00 ;RESET ALL COUNTERS
OF9B' AA C PLO RA
OF9C' BA C PHI RA
OF9D' AB C PLO RB
OF9E' BB C PHI RB
OF9F' AC C PLO RC
OF9F' BC C PHI RC
OF9F' AD C PLO RD
OF9F' BD C PHI RD
OF9F' C4 C NOP
OF9F' C4 C NOP ;MOVE PROGRAM POINTER TO
OF9F' C4 C NOP ;TOP OF LAST PAGE.
OF9F' C4 C NOP

;Wait for the leading edge of the 4 kHz timing signal.

OFA7' 3F A7' C WAIT0: BH4 WAIT0
OFA9' 37 A9' C WAIT1: B4 WAIT1
OFA'B E3 C SEX R3
OFA'C 63 C OUT PING ;PING
OFA'D 00 C DB 00H

OFAE' 3F AE' C W0: BH4 W0 ;WAIT FOR THE NEXT
OF8B' 37 B0' C W1: B4 W1 ;RISING EDGE OF 4 KHZ
OF8B' 1D C INC RD ;COUNT IT

;Begin looking for reply to ping, and incrementing counters
;if the reply is not detected.

OF8B' 34 B6' C R1 TEST2 ;INCREMENT COUNTER IF
OF8B' 1A C INC RA ;NO RECEPTION, OTHERWISE
```
<table>
<thead>
<tr>
<th>Location</th>
<th>Instruction</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>OF6'</td>
<td>35 B9'</td>
<td>TEST2: B2</td>
</tr>
<tr>
<td>OF8'</td>
<td>1B</td>
<td>INC RB</td>
</tr>
<tr>
<td>OF9'</td>
<td>36 BC'</td>
<td>TEST3: B3</td>
</tr>
<tr>
<td>OFB'</td>
<td>1C</td>
<td>INC RC</td>
</tr>
<tr>
<td>OFC'</td>
<td>8D</td>
<td>TESTRD: GLO RD</td>
</tr>
<tr>
<td>OFD'</td>
<td>CA OFAE'</td>
<td>LENZ WO</td>
</tr>
<tr>
<td>OFE'</td>
<td>9D</td>
<td>GHI RD</td>
</tr>
<tr>
<td>OFF'</td>
<td>CA OFAE'</td>
<td>LENZ WO</td>
</tr>
<tr>
<td>OFC4'</td>
<td>E7</td>
<td>SAVIT: SEX R7</td>
</tr>
<tr>
<td>OFC5'</td>
<td>F8 0E</td>
<td>LDI LOW (STADD)</td>
</tr>
<tr>
<td>OFC7'</td>
<td>A7</td>
<td>PLO R7</td>
</tr>
<tr>
<td>OFC8'</td>
<td>47</td>
<td>LDA R7</td>
</tr>
<tr>
<td>OFC9'</td>
<td>BD</td>
<td>PHI RD</td>
</tr>
<tr>
<td>OFCA'</td>
<td>FB FF</td>
<td>XRI OFFH</td>
</tr>
<tr>
<td>OFCC'</td>
<td>C2 EFFE'</td>
<td>LBE ALSTOP</td>
</tr>
<tr>
<td>OFCF'</td>
<td>07</td>
<td>LDN R7</td>
</tr>
<tr>
<td>OFD0'</td>
<td>AD</td>
<td>PLO RD</td>
</tr>
<tr>
<td>OFD1'</td>
<td>9F</td>
<td>GHI RF</td>
</tr>
<tr>
<td>OFD2'</td>
<td>5D</td>
<td>STR RD</td>
</tr>
<tr>
<td>OFD3'</td>
<td>1D</td>
<td>INC RD</td>
</tr>
<tr>
<td>OFD4'</td>
<td>8F</td>
<td>GLO RF</td>
</tr>
<tr>
<td>OFD5'</td>
<td>5D</td>
<td>STR RD</td>
</tr>
<tr>
<td>OFD6'</td>
<td>1D</td>
<td>INC RD</td>
</tr>
<tr>
<td>OFD7'</td>
<td>9A</td>
<td>GHI RA</td>
</tr>
<tr>
<td>OFD8'</td>
<td>5D</td>
<td>STR RD</td>
</tr>
<tr>
<td>OFD9'</td>
<td>1D</td>
<td>INC RD</td>
</tr>
<tr>
<td>OFDA'</td>
<td>8A</td>
<td>GLO RA</td>
</tr>
<tr>
<td>OFDB'</td>
<td>5D</td>
<td>STR RD</td>
</tr>
<tr>
<td>OFDC'</td>
<td>1D</td>
<td>INC RD</td>
</tr>
<tr>
<td>OFD1'</td>
<td>9B</td>
<td>GHI RB</td>
</tr>
<tr>
<td>OFDE'</td>
<td>5D</td>
<td>STR RD</td>
</tr>
<tr>
<td>OFDF'</td>
<td>1D</td>
<td>INC RD</td>
</tr>
<tr>
<td>OFED'</td>
<td>8B</td>
<td>GLO RB</td>
</tr>
<tr>
<td>OFE1'</td>
<td>5D</td>
<td>STR RD</td>
</tr>
<tr>
<td>OFE2'</td>
<td>1D</td>
<td>INC RD</td>
</tr>
<tr>
<td>OFE3'</td>
<td>9C</td>
<td>GHI RC</td>
</tr>
<tr>
<td>OFE4'</td>
<td>5D</td>
<td>STR RD</td>
</tr>
<tr>
<td>OFE5'</td>
<td>1D</td>
<td>INC RD</td>
</tr>
<tr>
<td>OFE6'</td>
<td>8C</td>
<td>GLO RC</td>
</tr>
<tr>
<td>OFE7'</td>
<td>5D</td>
<td>STR RD</td>
</tr>
<tr>
<td>OFE8'</td>
<td>1D</td>
<td>INC RD</td>
</tr>
<tr>
<td>OFE9'</td>
<td>8D</td>
<td>GLO RD</td>
</tr>
<tr>
<td>OFEA'</td>
<td>73</td>
<td>STXO</td>
</tr>
<tr>
<td>OFEB'</td>
<td>9D</td>
<td>GHI RD</td>
</tr>
<tr>
<td>OFEC'</td>
<td>57</td>
<td>STR R7</td>
</tr>
<tr>
<td>OFED'</td>
<td>30 F7'</td>
<td>BR SHUTDOWN</td>
</tr>
<tr>
<td>OFEF'</td>
<td>F8 43</td>
<td>ALSTOP: LDI LOW (GOFLEG)</td>
</tr>
<tr>
<td>OFF1'</td>
<td>A7</td>
<td>PLO R7</td>
</tr>
<tr>
<td>OFF2'</td>
<td>F8 00</td>
<td>LDI 00H</td>
</tr>
<tr>
<td>OFF4'</td>
<td>57</td>
<td>STR R7</td>
</tr>
<tr>
<td>OFF5'</td>
<td>17</td>
<td>INC R7</td>
</tr>
<tr>
<td>OFF6'</td>
<td>57</td>
<td>STR R7</td>
</tr>
<tr>
<td>OFF7'</td>
<td>7B</td>
<td>SHUTDOWN: SEQ</td>
</tr>
<tr>
<td>OFF8'</td>
<td>7B</td>
<td>SHTDWN: SEQ</td>
</tr>
</tbody>
</table>

101
<table>
<thead>
<tr>
<th>Offs</th>
<th>Hex</th>
<th>C</th>
<th>Function</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>0FF8'</td>
<td>E3</td>
<td>C</td>
<td>PDLOOP: SEL R3</td>
<td>; resets power control</td>
</tr>
<tr>
<td>0FF9'</td>
<td>62</td>
<td>C</td>
<td>OUT        FFREST</td>
<td>; flip flop</td>
</tr>
<tr>
<td>0FFA'</td>
<td>00</td>
<td>C</td>
<td>DB          OOH</td>
<td></td>
</tr>
<tr>
<td>0FFB'</td>
<td>71</td>
<td>C</td>
<td>DIS         ; disable interrupts</td>
<td></td>
</tr>
<tr>
<td>0FBC'</td>
<td>33</td>
<td>C</td>
<td>DB          33H</td>
<td>; turn power off and</td>
</tr>
<tr>
<td>0FBD'</td>
<td>7A</td>
<td>C</td>
<td>REQ         ; wait till it drops</td>
<td></td>
</tr>
<tr>
<td>0FBE'</td>
<td>00</td>
<td>C</td>
<td>IDL         ;</td>
<td></td>
</tr>
</tbody>
</table>
Macros:
CALL CHAR? ERROR? GETFLG RETURN TYPE MSG WORD?

Symbols:
ACTIVE 04C4' ADIO 0D5E' AD100 0D67' ADIES 0097' 
ADRS? 07BC' ADONE 0094' AEDOR 0081' AEDM FF4A' 
ALSTOP 0EFE' ARMID 056D' ASHD FF38 ASKOR 0113' 
ASTH FF3C ASTK 0594' ASTM FF3F ASUM FF3E' 
AT 0457' AT 08C2' ATCH 00581' BADCH 0108' 
BченSX 0D4E' BEL 0007 BIDONE 007A' BYDOUT 097B' 
CALCRC 0253' CALL 003A1' CALLCC 0887' CLEAR 03E7' 
CLETIC 0384' CLOOP 0A7B' CLOSE 0911' CIRCLO 0219' 
CLINT 0005 CMDIN 07D9' CMPIXIT 0144' COLSET 0A2E' 
COMPAR 012B' CONFIG 0012 CORM 080D' CPYTIM 02F2' 
CR 000D CRC 0A58' CRCHL FF0B CRCLO FF0C' 
CROICUT 00A4' CRLS 03D0' CRLSPH 03D6' CRLST 0A06' 
CTST 012F' DADONE 016C' DATA 0006 DATAM 0162' 
DEARM 00AE' DECC 00AE' DELAY 00A2' DEVICE 07C9' 
DIFFER 0143' DIHM FF45 DLE 0417' DSID FF30' 
DSFET 09EB' DSHM FF36 DTOA 0095' ENTINT 037A' 
ECL 03CF' EQS 03F3' ERROR 043A' ERROUT 08F3' 
EREC 0939' ETX 0003 ESEND 012A' ERCON 0382' 
EXEC 000A' EXINT 0360' EXEC 03C4' EXEC 0039' 
EXTR 004A' EXPENV 0211' FROM 03DD' GETEX 0232' 
GETCHR 0171' GESEN 019E' GETHX 016D' GHT00 0F11' 
GlobaL FF00 GCPLG FF43 HD FF10 HDONE 0077' 
HELP 053E' HEXCH FF24 HLPOUT 0905' HTOA 00850' 
I? 0888' IDENT 08D2' IDLEL 048E' INCHAR 01451' 
INCH 034A' INDEX 0196' INS 0418' INIT 0000' 
INDEX 032A' INTIM 08BC' INTEPT 0362' ITYPE 0072' 
INDD 09AF' INDDS 0C27' LDSCED 0CB2' LDINT 0890' 
INSTR 098A' IP 000A' LOAD 09A4' LOCK 040D' 
IUP 0C26' IPYIR 02CA' IPYIR? 02C5' MICLE 0282' 
MCDE 028E' MAIN 0F48' MEINT 048E' MINDM 0535' 
MINIY 0F26 MINDM 04D1' MOUTL 0439' NREL 082F' 
MOVE 00AE' MOVIT 0B26' MSRSDB 0F4B' NAME 03CC' 
NCLE 005C' NDA 014A' NO 03FE' NOCMD 08AE' 
NOCR 00A1' NDF 05A1' NORM 08C6' NOTACT 0516' 
NOTARM 0582' NUL 0000 NOTADD 06D7' NOTD 09D2' 
NOTLOC 0C72' NOTM FF1B NOTOR 0331' OK 041F' 
OCT 0423' OCN00 0C3F' OCN01 0C44' OCN02 0C4A' 
OCON 004F' OCN04 0C54' OPEN 0925' OVE 0444' 
OVER 03B1' PO 08B1' POSSP 0FF8' PSIN 0151' 
PING 0003 PNP 089B' PNYR 0504' PNYUT 08ED' 
PPOUT 0433' PROMPT 003A' PSIH 0353' FWIRST 0002' 
PRGSC 00D6' PRG 054D' PREM 0B35' RAM 1000' 
RAMST 08BD' RAMO 0C35' RCS 03DA' READY 0448' 
RBST 039A' RETURN 044B1' RSTST 0596' RST2 0087' 
RGSC 08B9' RSTFLG 09A1' RSTM 0F63' RSTX 038A' 
RTOC 0354' RUC 0AC0' SHTD FF29 SIMUM FF2F' 
S? 0875' SAIL 0380' SALITY 00CE1' SARM 0F05' 
SAT 058C' SAVIT OPC4' SAYTDL 00ED' SAYNOT 0F2B' 
SCMSG 043S' SCLRD 045D' SCRAW FF05 SECS 044F' 
SELECT 0001 SETAD 0D9A' SGFLG 0340' SHFTC 03B5' 
SHIFT 007A' SHFTC 0184' SHR 0089' SHUTDOWN 0FF7' 
SIZE 0000 SNA 0FF9' SHIFTPG 0FF9' SP 03D4' 
SPACE 0020 SPC 096F' SPOUT 096F' SWOPL 03D3' 
STACK FF0F STATUS 0007 STDAY 0465' STHOUR 0478' 
STMD 0482' STOP 007E STORE 0A23' STPC 0004 
STRADD FFOE SROB 01DO' STRM 03A4' T? 0862'
NO FATAL ERROR(S)
Distribution List for Technical Report Exchange

Attn: Stella Sanchez-Wade
Documents Section
Scripps Institution of Oceanography
Library, Mail Code C-075C
La Jolla, CA 92093

Hancock Library of Biology & Oceanography
Alan Hancock Laboratory
University of Southern California
University Park
Los Angeles, CA 90089-0371

Gifts & Exchanges
Library
Bedford Institute of Oceanography
P.O. Box 1006
Dartmouth, NS, B2Y 4A2, CANADA

Office of the International Ice Patrol
c/o Coast Guard R & D Center
Avery Point
Groton, CT 06340

NOAA/EDIS Miami Library Center
4301 Rickenbacker Causeway
Miami, FL 33149

Library
Skidaway Institute of Oceanography
P.O. Box 13687
Savannah, GA 31416

Institute of Geophysics
University of Hawaii
Library Room 252
2525 Correa Road
Honolulu, HI 96822

Marine Resources Information Center
Building E38-320
MIT
Cambridge, MA 02139

Library
Lamont-Doherty Geological Observatory
Columbia University
Palisades, NY 10964

Library
Serials Department
Oregon State University
Corvallis, OR 97331

Pell Marine Science Library
University of Rhode Island
Narragansett Bay Campus
Narragansett, RI 02882

Working Collection
Texas A&M University
Dept. of Oceanography
College Station, TX 77843

Library
Virginia Institute of Marine Science
Gloucester Point, VA 23062

Fisheries-Oceanography Library
151 Oceanography Teaching Bldg.
University of Washington
Seattle, WA 98195

Library
R.S.M.A.S.
University of Miami
4600 Rickenbacker Causeway
Miami, FL 33149

Maury Oceanographic Library
Naval Oceanographic Office
Stennis Space Center
NSTL, MS 39522-5001

Marine Sciences Collection
Mayaguez Campus Library
University of Puerto Rico
Mayaguez, Puerto Rico 00708

Library
Institute of Oceanographic Sciences
Deacon Laboratory
Wormley, Godalming
Surrey GU8 8UB
UNITED KINGDOM

The Librarian
CSIRO Marine Laboratories
G.P.O. Box 1538
Hobart, Tasmania
AUSTRALIA 7001

Library
Proudman Oceanographic Laboratory
Bidston Observatory
Birkenhead
Merseyside L43 7 RA
UNITED KINGDOM
A SAIL Compatible Three Channel Acoustic Navigation Interrogator

Navigation is accomplished by simultaneously interrogating each of three bottom mounted transponders positioned in an equilateral triangle around the mooring's anchor at a range approximately equal to the depth of the tracked instrument. The three round-trip travel times thus obtained having a resolution of 125µS and a SNR dependent jitter of less than 1.5mS, define a unique instrument position and are recorded along with the time of day and day of year.

The measurement period, the system clock and the program start time are set via a 20mA SAIL. Since the standby power requirement is negligible compared to the battery capacity, the instrument may be programmed months in advance of the deployment.

System endurance varies with the measurement period, however, typical programs permit navigation for up to 21 months at 12 points per day.

Upon recovery, the navigator data may be down-loaded via SAIL directly to the storage medium of a suitable computer.

Approved for public release; distribution unlimited.