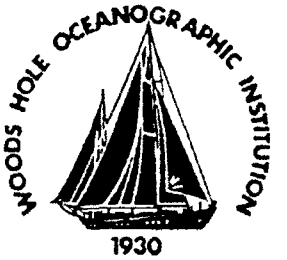
WHOI-92-29

AD-A264 923

Woods Hole Oceanographic Institution



Real-Time Tomography Mooring

by

James Lynch, Daniel Frye, Kenneth Peal, Stephen Liberatore,
Sean Kery, Edward Hobart, Arthur Newhall and Stephen Smith

DTIC
ELECTE
MAY 26 1993
S D
E

June 1992

Technical Report

Funding was provided by the Office of Naval Technology
under Contract No. N000-14-C-90-0098.

Approved for public release; distribution unlimited.

93 5 25 113

93-11675

WHOI-92-29

Real-Time Tomography Mooring

by

James Lynch, Daniel Frye, Kenneth Peal, Stephen Liberatore,
Sean Kery, Edward Hobart, Arthur Newhall and Stephen Smith

Accesion For
NTIS CRA&I
DTIC TAB
Unannounced
Justification
By
Distribution /
Availability Codes
Dist | Avail and / or Special
A-1

Woods Hole Oceanographic Institution
Woods Hole, Massachusetts 02543

June 1992

Technical Report

Funding was provided by the Office of Naval Technology
under Contract No. N000-14-C-90-0098.

Reproduction in whole or in part is permitted for any purpose of the United States Government. This report should be cited as Woods Hole Oceanog. Inst. Tech. Rept., WHOI-92-29.

Approved for public release; distribution unlimited.

Approved for Distribution:

Tim Stanton, Acting Chairman
Department of Applied Ocean Physics and Engineering

ABSTRACT

A real-time tomography system has been developed which combines ocean acoustic tomography with satellite-based time keeping and satellite telemetry. The basis of the system is the acoustic tomography transceiver and its associated acoustic navigation grid. To this basic system, a link to the surface has been added to provide a pathway for telemetry of the tomographic data to shore and a downlink for satellite-derived time which is used to correct the transceiver's clock. The surface buoy contains a GPS receiver, clock comparator, system controller and multiple ID Argos transmitters. Processed tomography signals, transceiver location data, time, time drift and surface buoy engineering data are transmitted to satellite using a total of 32 data buffers transmitted every eight minutes. The report describes the real-time tomography system in detail, with particular emphasis on the modifications implemented to convert the standard tomography instrument to a real-time oceanographic tool.

Key words: Acoustic tomography, real-time, telemetry

ACKNOWLEDGMENTS

The authors would like to acknowledge the valuable contribution to the report's organization and structure made by Wendy Liberatore who collated the contributions of the authors and organized them into a cohesive report. Funding was provided by the Office of Naval Technology under Contract No. N000-14-C-90- 0098.

TABLE OF CONTENTS

1. Introduction 1

1.1 Ocean Acoustic Tomography 1

1.2 Real-Time Tomography 2

1.3 S-Tether Design 4

2. The Tomography System 9

2.1 Controller 10

2.2 Signal Generator 10

2.3 Receiver 11

2.4 Clock 11

2.5 FSK Modem 11

2.6 Navigator 11

2.7 Hydrophone Array 12

2.8 Mechanical Assembly 12

2.9 Instrument Support Programs 12

3. Surface Buoy Electronic System 15

3.1 IBC Hardware 15

3.2 GPS Receiver 15

3.3 Dynamic Motions Measurement System 18

3.4 Other Systems 21

4. Data Processing and Telemetry 24

4.1 Peak Picking 24

4.2 Navigation 26

4.3 Engineering and Ambient Noise 26

4.4 Data Transfer to IBC 27

4.5 In situ Time Check 27

4.6 Application of Time Corrections 27

4.7 Argos Telemetry Formats 28

4.8 Transceiver Record Format Changes 30

5. S-Tether Mooring System 31

5.1 Surface Buoy 31

5.2 Tether Description 33

5.3 Subsurface Buoy 34

5.4 Electromechanical Cables and Terminations 36

5.5 Mooring System Design 36

6. Conclusions 40

6.1 Present Status of Project 40

6.2 Future Directions 41

7. References 44

8. Appendices 44

A. Modifications Necessary for Real-Time Operation 49

B. Environments Used During Development of the Real-Time Tomography Transceiver 51

C. Tomography Transceiver Documentation 54

D. Signal Generator Module 97

E. Test Procedures for Real-Time Tomographic Mooring 106

F. Mechanical Assembly Drawings 112

G. S-Tether Tomography Mooring Analysis 131

Figures:

1. Real-time tomography mooring 5

2. Flowchart of tomography transceiver system operation 14

3. IBC controller block diagram 16

4. IBC controller operation summary 17

5. S-Tether surface electronics data logger 20

6. Surface buoy assembly drawing 23

7. Comparison of sum of complex parts and sum of squares 25

8. Transceiver program flowchart for clock check 29

9. Steel double armor E/M cable 35

10. 3 x 19 torque-balanced wire rope with 3 electrical conductors 37

11a. Current profiles used to design the real-time tomography mooring for deployment at Site D 38

11b. Current profiles used to design a generic version of the real-time tomography mooring 38

List of Tables

	Page
1. Telemetry options for offshore buoys	7
2. FSK modem specifications	11
3. S-Tether surface electronics data logger	19
4. Dynamic motions measurement system specifications	19
5a. Complete Argos buffer (binary)	28
5b. Argos messages (as transmitted)	28
6. Surface buoy specifications	31
7. Results of computer model runs for the real-time tomography mooring	39
8. Comparison of real-time tomography technologies	42

REAL-TIME TOMOGRAPHY MOORING

James Lynch, Daniel Frye, Kenneth Peal, Stephen Liberatore, Sean Kery,
Edward Hobart, Arthur Newhall and Stephen Smith

1. INTRODUCTION

1.1 Ocean Acoustic Tomography

Ocean acoustic tomography was first proposed by Munk and Wunsch in 1978 [1], and was first demonstrated in the ocean in 1981 [2]. We are now entering the second decade of tomography's existence as an oceanographic measurement technique, a decade in which we hope to see tomography go from being an experimental novelty to a first rate oceanographic tool. For those interested in the details of tomography as a technique, we refer to the initial paper by Munk and Wunsch [1], which while slightly dated, is still an excellent overview of the technique, and also to a more recent review by Worcester, Corunelle, and Spindel [3], which covers tomography's status from 1987-1990, i.e., almost to the present. The latter paper contains an excellent reference list, which can further guide the interested reader to the (now quite extensive) literature on tomography.

For the purposes of this report, it will suffice to say that acoustic tomography in its basic form is a technique that uses the time of flight of acoustic pulses between sources (usually moored) and receivers to create fully 3-D maps of the ocean sound speed, temperature, and (given two-way transmissions) current fields. To make these maps, one must resolve and identify the various acoustic multipaths (eigenrays) which connect sources and receivers and clock their arrival times to very high accuracy. Tomography also requires very precise mooring (actually, source/receiver) positioning accuracy, as each 1.5m of distance error between the moorings corresponds to 1 msec of travel time error. Typical oceanographic feature signals are of the order of 10-50 msec, so unaccounted-for mooring tilts of a few tens of meters can mimic large-scale oceanographic features. The first decade of tomography was, to a large part, spent learning how to deal with these experimental measurement issues, which demand higher levels of instrument performance and ocean acoustic expertise than any other oceanographic measurement.

As previously mentioned, the second decade of tomography is being geared towards exploitation of the technique, rather than technique development. Small, less expensive, commercially available instruments such as the transceiver manufactured by Webb Research of Falmouth, MA, are envisioned probing sizable fractions of ocean basins with mesoscale resolution over decadal periods. Larger systems, based on the experience gained from the Heard Island transmissions [4], could be used to monitor

global ocean warming over multi-megameter scales and decadal periods. Assimilation of tomographic data into numerical models for forecasting and model development offers another exciting use of tomography – indeed the oceanography of the future appears to be a synthesis of measurement and modeling techniques, optimized to produce the most information for the least expense and effort.

1.2. Real-Time Tomography

If one wishes to monitor the ocean and incorporate daily information into predictive numerical models, then data must be available in near real-time. Real-time information also allows one to rescue and redeploy faulty instruments in the ocean and to think about the data while it is being collected rather than waiting a year or two for the recovery of internal recording devices. Obtaining real-time ocean acoustic tomography information is not a trivial exercise, mainly due to the large telemetry bandwidth needed to get the data ashore and the very limited power available to moored instruments. The total bit rate attainable and the energy required to send a bit are absolutely critical parameters to any proposed telemetry system. Cost, reliability, and durability are other key factors. To date, a number of possible solutions to the problem of telemetering acoustic tomography data in real time have been proffered.

A brief overview of these proposed systems, their strengths and their weaknesses follows. The first, and perhaps oldest concept for real-time tomography is to use moored sources transmitting to the Navy's SOSUS (Sound Surveillance System) receivers. This technique was considered from the inception of tomography, and was first implemented by Spiesberger [5] and later implemented in real-time by an SIO and APL/UW collaboration [6]. This technique has several advantages. By using existing receivers and inexpensive receiver modifications, it has the advantage of comparatively low cost. The bandwidth of the channel is also very large (effectively infinite for our purposes), so that the data rate required is not a problem. The system also has many receivers, creating many acoustic paths for one source.

A second scheme for implementing real-time tomography is to connect the tomographic systems to existing submarine telephone cables, which are outdated for telephonic purposes, but are potentially very useful for oceanography. This concept would allow very high data transmission bandwidths, many deep ocean locations, transceivers near the SOFAR axis (the ideal spot for tomography), and two-way transmissions for reciprocal tomography. The disadvantages are extremely high installation and maintenance costs (a cable-laying ship is needed), the cost of renting or buying the cable system, and the geographic positions of the cables, which are not optimal for oceanography.

In an effort to retain the submarine cable telemetry concept, but ameliorate some of the difficulties noted above, work on tying oceanographic sensor output into the phone cables via inductive modems is being pursued by WHOI investigators. This

avoids the difficulty of direct electrical attachment of the sensor output to the phone cable. Moreover, high baud rates are attainable by the modem. Long lifetimes (~five years) could be expected for the bottom receiver/inductor, so that maintenance is reduced. The disadvantages are that an ROV deployment is necessary (expensive, but less expensive than a cable ship), and the previously mentioned drawbacks of the cable system, rental (purchase) and location. This is still an experimental technology, but it is a promising one.

The third variant of real-time tomography we will discuss is "moving ship tomography," in which one deploys either the source or receiver from an oceanographic or naval vessel. (Current practice is to deploy the receiving array from the ship.) This technique, which has been tested at sea during both the Greenland Sea and AMODE tomography experiments [7], was originally proposed to allow one to obtain greater spatial resolution of ocean features, i.e., fill in "spatial wave number gaps" left by standard moored arrays. This technique also allows the Navy to map sound speed in areas of fleet activity, a great tactical advantage, using existing systems such as SURTASS (Surveillance Towed Array System).

The disadvantages of this technique, as currently practiced, are as follows. First, due to the slowness of the ship, which typically occupies one receiving station every three to four hours, oceanographic maps made of a large region are temporally undersampled in a peculiar way. Specifically, the area along the most recent 2-D slice is well sampled, but the rest of the area to be covered is not. Spatial maps made from a ship covering a large area with fine spatial resolution will of necessity suffer from the lack of synopticity over the course of the survey, much as happens with CTD work. Use of only one realization per path also means more internal wave noise compared to conventional tomography, where one often averages many transmissions to reduce noise. Another present drawback of this concept is its need for extremely precise measurements of array element locations. High accuracy GPS positioning, short baseline array sensor localization, and precise measurement of hardware positions aboard ship have been needed to get adequate sensor location information, a taxing chore even for a highly trained scientific party. Both the deployment and analysis are very labor intensive in this scheme, also. Finally, one still needs the clock and mooring motion corrections for the source (as with SOSUS) for this technique to be truly real time.

A variant of moving ship tomography has recently been proposed by Duda and Lynch, called "relative arrival structure tomography" [8]. It has been well known to tomographers that the relative structure of the acoustic multipath arrivals contains information about ocean structure, though the ability to resolve features is degraded relative to the careful monitoring of the time changes of individual multipaths. Due to the somewhat poorer information content, tomographers have largely ignored relative structure information. However, recent studies indicate that, at least in some cases, the information content is substantial and as a result, the technique is being pursued. The advantages of relative arrival tomography are as follows. First, it is robust, as

only wristwatch accuracy time and LORAN accuracy position (GPS at worst case) is needed for the measurements. Second, it is cheap; shot sources can be used with sonobuoys as receivers. Third, it's fast; an aircraft can be used to deploy shots and to survey a large area. And finally, it is simple to handle computationally, i.e. the inverses do not have any extra complication due to clock or mooring motion corrections. On the negative side of the ledger, relative arrival tomography gets less information than "standard" tomography, as mentioned. The theory is not yet fully developed for this technique, though preliminary inversion results are encouraging. (Simulations are being performed to find out the range of geometries, conditions, and environments under which the technique works well). Also, as with moving ship tomography, only one realization of a path is used, thus increasing sensitivity to internal wave noise.

The final variant of real-time tomography we will discuss, before moving on to the S-Tether system which is the focus of this report, is multiple convergence zone acoustic telemetry. Recent at-sea tests by Catipovic have demonstrated that one may send data at extremely high bit rates over many convergence zones (4-5) using low frequency acoustic sources [9]. This would allow one to use the tomographic sources themselves to transmit tomographic data (clock times, travel times, mooring motion) to SOSUS station "relays." This also allows the tomographic mooring to be reasonably remote from the SOSUS receiver, thus avoiding classification problems, as well as permitting the transceiver to be near the SOFAR axis, its optimal position.

The weak points of multi-CZ acoustic telemetry are as follows. First, it is a high power consumption technique, using of the order of 1 joule/bit. Due to this power consumption, standard mooring battery capabilities limit total data transmission to ~2-3 Mbyte/yr, the same as is achievable with the Argos system. This rate could improve in the future, however, and work on multi-CZ telemetry is being pursued. This technique is also limited to within a few CZ's of the SOSUS stations (or other suitable relays), so that, again, it is not optimal for mid-ocean coverage.

1.3. S-Tether Design

Developed originally for telemetering standard oceanographic sensor data (currents, temperatures, etc.), the S-Tether system, shown in Figure 1, has been adapted for use with tomographic transceivers under a contract with the Office of Naval Technology. The system consists of a surface telemetry float which sends information to shore via RF (usually satellite) transmissions. It is connected to the tomography transceiver directly by electromechanical cable, permitting high baud rate information transfer from the transceiver to the surface telemetry buoy. By using the S-Tether design, in which a slack tether mechanically decouples the surface buoy motion from the tomography mooring, the effects of surface waves and currents on the mooring, which can be particularly deleterious for tomography, are minimized.

The advantages of using an S-Tether system are numerous. The primary advan-

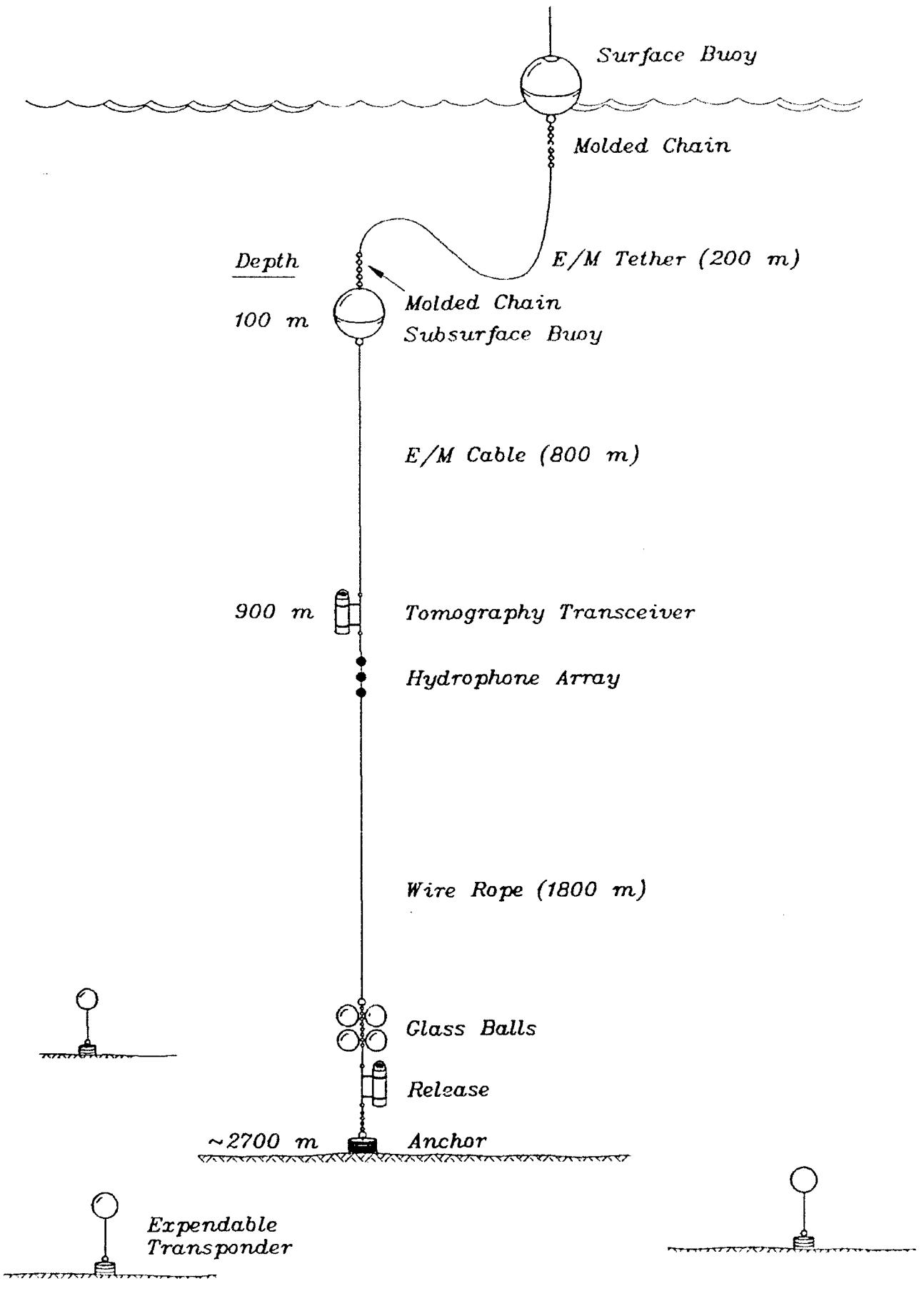


FIGURE 1: Real-time tomography mooring

tage is that its surface expression (which also has disadvantages, to be discussed later) allows use of the RF communication channel. While limited in bandwidth at present, this approach will eventually provide a very high bandwidth, low power transmission path - indeed, only cable communication options can compete in bandwidth. Presently, we are limited in ocean applications to the Argos system bandwidth, which allows us to send about 2.2 Mbyte/yr over four Argos ID's. This does not represent even a fraction of the potential bandwidth of RF channels. In Table 1 (from [10]), some of the present and future options for RF communications from ocean surface buoys are shown. In addition to data telemetry, the surface expression allows accurate time from the GPS system to be obtained. The surface buoy has a GPS receiver which outputs an accurate time tick once per second allowing the tomographic clock on the S-Tether mooring to maintain an accuracy of 20 μsec, far better than the 1 msec accuracy (at best) of the tomographic signal arrival time determination. Moreover, if the surface buoy breaks loose, it is easily tracked by its Argos transmissions and recovered. It should be noted that should the S-Tether system break loose, only the real-time capability is lost in the tomographic system. The transceivers also record internally, in addition to sending a subset of the data in real time via the S-Tether system, so that no valuable oceanographic information is permanently lost.

Another advantage of S-Tether is its ability to operate in most deep ocean locations, which makes it suitable for "regional focus" array systems, as opposed to the telephone cable or SOSUS systems which serve only fixed locations. The transceiver can also be placed at any depth with an S-Tether system. Two-way communications with the buoy are possible, both in the reciprocal tomography (current measurement) sense and the interactive communication sense (e.g. for changing system tasks or schedules). Finally, the S-Tether unit is comparatively cheap - production copies are expected to be less than $100k, and only a few systems are needed to give real-time capabilities to a large tomographic array.

Looking at S-Tether's liabilities, the surface expression and low bandwidth of the Argos link are the main weaknesses. The surface expression has three liabilities: 1) collisions with shipping, 2) breaking of the E/M cable due to dynamic loading by surface waves, and 3) submergence of the float by high current drag. The first liability can largely be avoided by placing the moorings outside of shipping lanes. The second liability has led to the failure of two S-Tether moorings due to cable fatigue. However, a new design using chain at the buoy-cable interface should avoid this problem. The third liability, submergence of the float, will result only in loss of real-time information during the period the float is submerged. The float itself is designed to be capable of submergence to 300m or greater over extended periods with no damage.

The low bandwidth limitation of Argos restricts the data stream to either six transmissions from one remote mooring per day or daily averages from six remote moorings. For testing purposes, the S-Tether is configured in the former mode; in

Telemetry Option	Coverage	Typical Throughput	Power Requirement J/bit	Platform Location	Long-Term Availability	Receive Station
ARGOS	Worldwide	0.02-0.2 bps	1-10	+/- 1 km	Yes	Not required, but available with limited coverage
ATS	Hawaii to Azores but Non-Polar	0.3-1.2 kbps	0.5-1	N/A	No	Not required, but available
GOES/METEOSAT/ GMS/INSAT	Worldwide Except Polar Regions	0.2-2 bps	1-2	N/A	Yes	Not required, but available
VHF[a] (Line-of-sight)	10-50 km from Receive Station	0.3-1.2 kbps	0.02-0.1	N/A	Yes	Required
Meteor-Burst[a]	2000 km from Master Station	1-20 bps	0.2-1	N/A	Uncertain due to licensing	Required
HF Packet[a]	Variable up to Worldwide	1-30 bps	1-10	N/A	Yes	Required.May need several
Inmarsat[a] Standard-C	Worldwide except Polar regions	10-100 bps	0.5-1	N/A	Yes	Not required
Microsats[a/b]	Worldwide	10-100 bps	.01	N/A	Unknown	Unknown
Irridium[b]	Worldwide	Unknown	Unknown	N/A	Unknown	Not required

NOTES: (a) Two way communication
(b) Future systems - estimated throughput and power

TABLE 1: Telemetry options for offshore buoys

actual operations, it will be configured in the latter. Even with the Argos system, a six mooring reception capability is useful, as most tomographic deployments to date have included only about half a dozen systems. Moreover, it is felt that with the advent of new satellite communication systems, and the further development of existing ones that the bandwidth available for ocean data telemetry will soon improve dramatically.

2.0 THE TOMOGRAPHY SYSTEM

The Webb tomography transceiver consists of a high power acoustic source, an acoustic receiver, a stable clock, an acoustic navigation interrogator and a sequencing controller all interconnected with a simple open collector SAIL [11] bus. The system design allows the transceiver to accurately measure and record the arrival times, over many paths, of signals generated by similar instruments moored at ranges of up to 500 km. The recorded travel times are later processed to obtain the intervening sound speed field. Since temperature, salinity, and density are the chief factors influencing sound speed in the ocean, these properties can in theory be derived from the data. In practice, salinity (and thus density) is too weak an effect on sound speed to be measured by tomographic techniques.

Due to attenuation and spreading of the acoustic signal, the source's signal is actually below the ocean ambient noise level by the time it reaches the receiver. For this reason the source generates a signal based on a phase encoded pseudo-random sequence which allows the receiver to employ a simple signal processing technique to enhance reception levels. Signal phase stability must be maintained during receptions and transmissions or a loss in processing gain will occur. Also, the transmit time and the receiver's acquisition time must remain synchronous for travel time to be inferred from arrival time. The transceiver's clock satisfies the first requirement utilizing its temperature-compensated output frequency. However, the second requirement is only satisfied after end-point corrections are applied to the data since the clock's time may be in error by up to 30 ms at the end of a six-month deployment.

Mooring motion due to currents and tides alters the range between the source and the receiver. Even with high tension moorings, movements in excess of 100 meters are not unusual. There is no way (without sacrificing some useful multipath data) to distinguish between altered arrival times which result from mooring motion and those that result from the effects of oceanic variations. For this reason each transceiver is equipped with an acoustic navigation interrogator. This device, prior to each reception, obtains round-trip travel times between the tip of the receiving array and three bottom-mounted transponders. These times define a unique instrument position and allow the data to be corrected for the effects of mooring motion.

The tomography transceiver controller allows communication with an external computer via a 20 mA SAIL loop. This module also contains the sensor interface electronics and the digitizer necessary to monitor all system voltages as well as pressure and temperature both internally and externally. The system controller's primary function is to execute the program which, following a task table downloaded via SAIL, sequences all instrument activity. The current version of this program is capable of handling up to 12 task tables which allows the instrument to alter its behavior up to 12 times during the course of an experiment.

Real-time tomography requires access to the acoustic travel times, mooring lo-

cation data and time correction data in real time. To meet this requirement using the limited Argos bandwidth requires compression of the acoustic data set. This compression is performed by correlation of the demodulates against a replica of the transmitted signal followed by peak picking and sorting. A dramatic reduction in the amount of data to be transmitted is the result of this processing.

2.1 Controller

The transceiver architecture consists of loosely coupled modules, each performing an individual function. The operation of the modules is coordinated by the controller, which operates from task tables to determine when each module should be enabled. The modules are connected to a simple bussed backplane which provides power, communication via open collector SAIL, synchronization and individual reset. A more complete description of the controller is provided in [12].

Most of the modifications required to achieve real-time operation were changes to the controller. The addition of a navigator to the system required no change to the controller since provision for its addition was included in the original design. It is necessary to include navigator information in the task table to enable its use during a deployment.

Telemetry between the tomography transceiver and the surface buoy controller is implemented with a frequency shift keyed (FSK) modem using the E/M cable. This requires mounting the FSK modem adjacent to the controller and performing wiring modifications to existing controller boards. These were needed to control the modem, to provide the required power and clock frequency, and to transfer data to and from the modem.

Other changes to the controller were implemented in software. These changes included modifications to the BIOS to support the low level hardware changes, addition of new virtual devices, and modification to existing device drivers. The new virtual devices are called "time check" and "calculate". Time check instructs the surface unit to enable the GPS receiver and execute a comparison with the internal transceiver clock. The calculate function uses the acoustic data to perform correlation and peak picking; then it formats a message which is sent to the surface unit to be telemetered. Modification to existing drivers is required to extract data from various devices to be used by the calculate function.

2.2 Signal Generator

The signal generator module, on command from the controller, synthesizes a phase encoded pseudo-random sequence. This signal, applied to the module's power amplifier is level shifted and appropriately conditioned to drive the transceiver's outboard ceramic transducers which generate the acoustic signal. For details of the signal generator's hardware/software configuration and for its operation, refer to Appendix D.

2.3 Receiver

The receiver module amplifies, filters, digitizes, and demodulates the very low level signals produced by the hydrophone array. The demodulates are stored on a 200 Mbyte hard disk drive which is a component of the receiver module. For details of the receiver's hardware/software configuration and for its operation, refer to [12].

2.4 Clock

The clock module is the tomography transceiver's primary timing element. Its principle component is a low power microprocessor circuit which implements a digital temperature compensation technique to synthesize a reference frequency from a high quality free-running crystal oscillator. For details of the clock's hardware/software configuration and for its operation refer to [12].

2.5 FSK Modem

The modem used is a modified version of the Instrument Bus Computer (IBC) FSK modem. It operates at 1200 baud, half duplex. Modifications include enhanced front-end noise immunity and an improved dynamic calibration technique. Table 2 shows the modem's detailed specifications.

Table 2: FSK modem specifications

Mark Frequency	4800 Hz
Space Frequency	9040 Hz
Carrier Detect Time	
(capture) Mark	10ms
(capture) Space	7ms
Phase Comparator Output	
Duty Cycle for Mark	21%
Duty Cycle for Space	75%
Band-pass Center Frequency	7200 Hz
Power Consumption	6mW

The FSK modem board is mounted on a single module mount bracket in the position adjacent to the system controller in the transceiver chassis. Power is obtained by a pair of wires to the backplane connector formerly used by the mass store. Control signals come from J4 on the controller I/O interface board. The FSK line is added to the wire bundle to the end cap from the system controller.

2.6 Navigator

The navigation module provides a means of interrogating three bottom-mounted transponders. The three, two-way travel times thus obtained are stored as part of the

navigator's status record and define three slant ranges from which a unique transceiver position may be computed. For details of the navigator's hardware/software configuration and for its operation, refer to [12].

2.7 Hydrophone Array

The hydrophone array, which includes the navigation interrogator's transducer, is a multi-element shaded array designed to be most sensitive at its center. The elements are spaced 1/2 wave length apart at 400 Hz with a shading pattern of 1,3,3,1. This yields a 3dB array gain with a +/- 18 degree horizontal lobe and a null on axis at 400 Hz. The elements are enclosed in a tygon ether-based, polyurethane tube filled with pressure compensating oil. Included in the tubing and attached to a swivel joint at one end and an eye at the other end, is a length of 5/16" galvanized plow steel wire rope. This allows the array to carry the full mooring tension.

2.8 Mechanical Assembly

All electronics modules in the tomography transceiver are mounted on a shock isolated bent aluminum frame. This frame is enclosed in a clear acrylic weather housing which i cludes a check valve at each end to facilitate purging the electronics with dry air.

The battery pack is a series of custom modules composed of alkaline "D" cells wired to provide 45 volts. A single module is rated for 14 amp-hours. As many as 40 modules may be mounted on a supporting rod which is attached to a plate and secured to the pressure housing near the bottom end cap.

The pressure case is 4.6m in length, includes two removable end caps and is rated to 2000m. One end cap has no electrical penetrators and connects to the mooring with a stainless steel lug. The other end cap has two electrical penetrators, one for the hydrophone array and the other for the surface module connection. This end cap also mounts a swivel lug which mates to a similar lug on the end of the array. The mated combination allows the hydrophone freedom of movement in two axes.

For more details regarding the transceiver's mechanical design, refer to the first section of [12].

2.9 Instrument Support Programs

During instrument preparation, tests are performed to verify correct operation. Appendix E outlines a complete plan for this checkout. Part of this work involves running the instrument with test tasks under controlled conditions in the laboratory and evaluating the data recorded. The steps in this process are similar to those in an actual deployment – defining and loading the operating task and evaluating some portion of the recorded data. Three programs described below are used to aid in this process. Figure 2 is a flowchart of the steps in the editing, loading and verifying of the operating task.

ET - task editor: This program reads and analyses task files and communicates with the transceiver to download and check the files in the instrument. Its operation is described in [12]. A modified version of this program was made to deal with differences in the real time operation. These differences relate to the different data structures and the additional virtual devices required.

DM - disk monitor: This program scans the disk to provide a summary of the data recorded during a deployment. It can also read and transfer data from the disk into DOS files for examination. The status files are in ASCII and can be easily listed; the receiver files must be examined with the program PK. Operation of the program is described in [12].

A modified version of this program was made to deal with differences in the real time operation. These differences relate to the different data structures and the additional virtual devices required. The two new virtual devices are treated like physical devices including the generation of local status records which are added to the system status whenever they execute. These records are decoded and displayed by this program.

PK - peak picker: This program reads the receiver disk files written by program DM from the instrument's disk, does the correlation against the correct pseudo random sequence then computes and displays a plot of the intensities. Also displayed are the time of the record, the value of the zero dB level, and the magnitude and location of the largest peak. No change to this program was required for real time operation since the receiver data format was not changed.

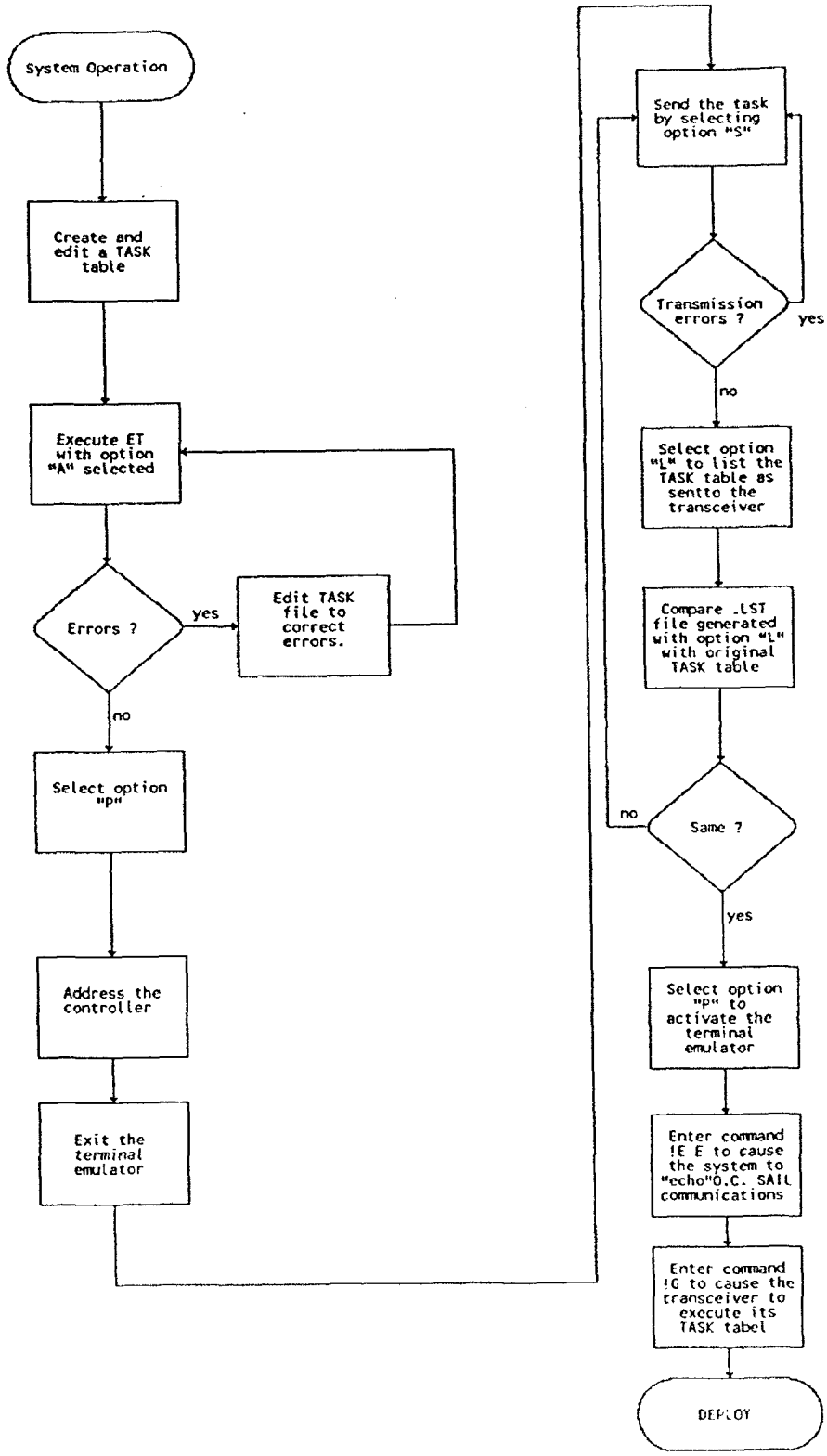


FIGURE 2: Flow-chart of tomography transceiver system operation

3.0 SURFACE BUOY ELECTRONIC SYSTEM

3.1 IBC Hardware

The IBC controller is a two-board set based on an Intel 80C51FC microcontroller. Figure 3 is a block diagram of the hardware and Figure 4 summarizes the system operation. The 80C51 is clocked by a 2.4576 MHz crystal which results in a 4.88 microsecond instruction time. Attached to the 80C51 are three 858 National Semiconductor UARTs. Each of these, along with the UART internal to the 80C51, is attached to a different communications element of the system. Since FSK communications can occur at any time, the 80C51 internal UART is used for FSK communications and is connected directly to the FSK interface board. The other UARTs have the corresponding signal conditioning electronics for their specific needs. SAIL and GPS communications occur infrequently. For this reason, the power to the SAIL clock, GPS receiver and associated UART is under 80C51 control and is only applied when needed. The PTT is always powered but the IBC UART used for its communication is also under separate 80C51 power control.

All communication operations are interrupt driven. This allows the 80C51 to be placed in a low power "idle" mode whenever possible without communications losses. The 80C51 will exit the idle state upon receipt of an enabled interrupt.

Also on the 80C51 bus are a 7170 clock and 32K byte static RAM. The clock's "alarm" function is used to generate a periodic "wake-up" interrupt. The 32K byte RAM is used to buffer the tomography controller data for processing prior to PTT transmission. The 80C51 internal UART and the 7170 clock interrupts are always active. The processor is left in the idle mode for the majority of the time to conserve system power. New activity is then initiated by either the tomography controller or the clock.

Finally a small but important piece of the IBC controller hardware is a watchdog timer. This watchdog consists of a pair of 4060 counters that must be reset periodically by the 80C51. A 307.2 kHz crystal is used to clock a third 4060 which provides the time base for the watchdog pair. One output from this 4060 is also used as the 153.6 kHz baud rate clock for the IBC FSK interface. The 4060 watchdog pair are cascaded to provide a signal with a period of approximately two hours which is applied to the reset input of the 80C51. A single port pin from the 80C51 is tied to the reset input of both watchdog 4060's. If they are not reset within a 2 hour period, then the current will reset the 80C51.

3.2 GPS Receiver

The S-Tether Tomography GPS receiver is a Trimble Navigation Limited model Trimble Advanced Navigation Sensor (TANS). This is a multi-channel GPS receiver, capable of automatically providing 2D or 3D position information depending on the satellite coverage. For S-Tether Tomography use, the following TANS operation specifics need to be considered.

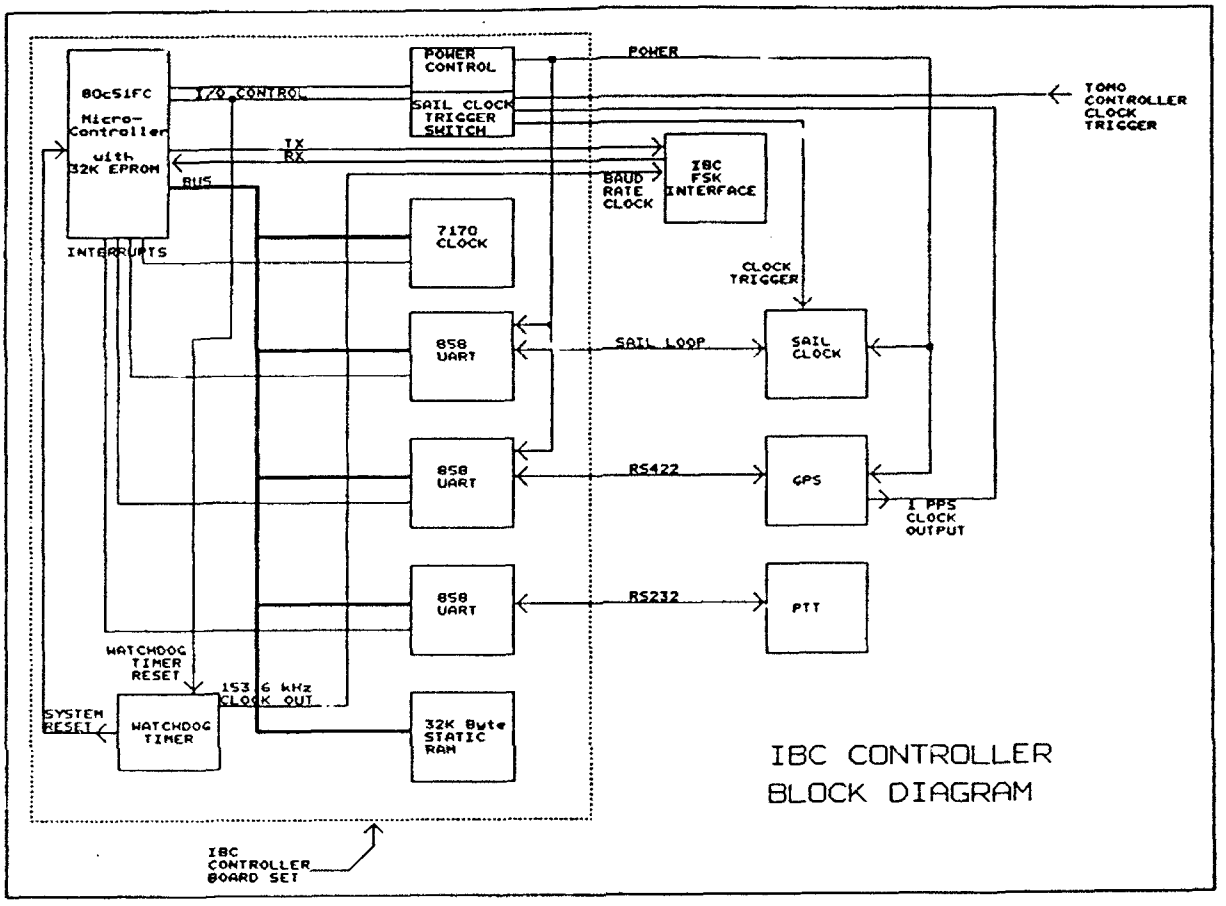
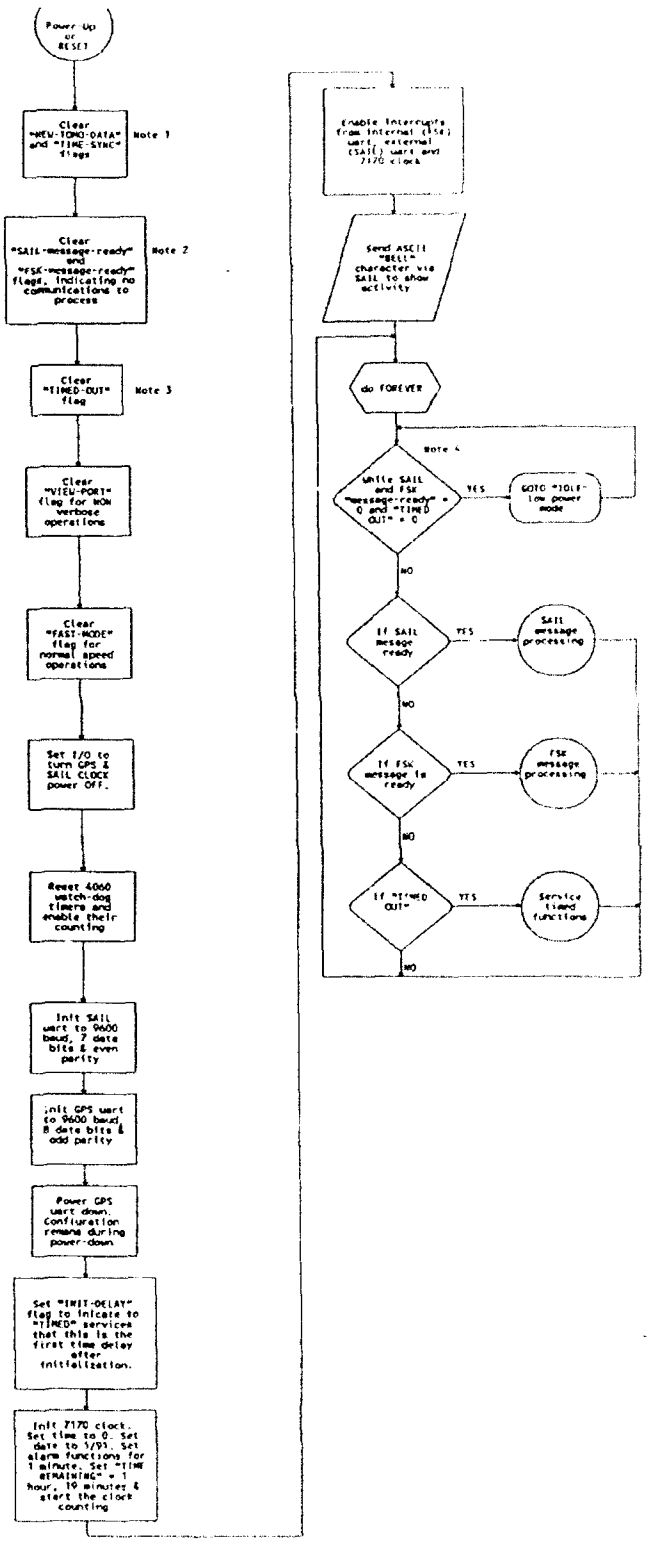


FIGURE 3: IBC controller block diagram

First, the TANS must be programmed to provide the desired information. This is done by first sending it packet 35 with the following data bytes: Byte 0 = 02 (hex) sets the unit for Lat/Lon vs XYZ position data. Byte 1 = 02 (hex) sets it for East/North/Up vs XYZ velocity data. Byte 2 = 03(hex) sets it for UTC time, with position reported automatically and computed on the integer second. Byte 3 = 0 sets raw measurements OFF. (Note that you must adhere to the TANS protocol when communicating. Refer to the TANS manual for communication and packet format information.) Second, to insure obtaining the most rapid position fix, send the TANS packet 22 with its one data byte = 0. This sets the TANS position mode to automatic, which will create a position fix as quickly as possible, depending on the satellite coverage.

The second consideration in the system design is managing the TANS automatic



Note 1:

"NEW TOMO DATA" flag, if TRUE indicates new data has been received from the TOMO controller to be processed. This is a special flag set TRUE by the FSK receive interrupt.

"TIME-SYNC" flag, if TRUE indicates a new GPS time sync is required. Set TRUE by the "TIMED" functions service routine if a request for a GPS time sync is not received from the TIMC CONTROLLER in a 24 hour period.

Note 2:

"SAIL-message-ready" and "FSK-message-ready" flags, when TRUE indicate a new message has been received from their respective port and needs to be processed. Each flag is set TRUE by its corresponding input interrupt routine upon completely receiving a new message.

Note 3:

"TIMED-OUT" flag, if TRUE indicates a timed delay has completed. Set TRUE by the 7170 clock alarm interrupt routine.

Note 4:

"IDLE MODE" is exited by servicing an interrupt. Upon completion of the interrupt, execution continues with the instruction following the one that initiated the "IDLE" state. This "quiet" long requires the system returns to "IDLE" mode until further processing is required.

FIGURE 4: IPC controller operation summary

data transfer. This involves tracking the differences in the data provided along with the variable data rate associated with different phases of operation. These phases are: 1) immediately after power-up, 2) during position acquisition, and 3) when generating fixes. Any device receiving TANS data needs to receive data into a circular buffer by a means independent from the data processing. However, the data must be processed at a rate fast enough to keep up with the data received.

At TANS power-up, four different data packets are sent of which only two are of interest. The first is packet 41, GPS time. This is the time as determined by the TANS internal clock, which is inaccurate and not used for correcting transceiver time. This packet is required because it contains the day of the year information. Day of the year is required by the SAIL clock. The second power-up packet of interest is 4A, LAT/LON position. At power-up and until satellite tracking occurs, this represents the last position prior to power-down. This packet must be ignored until satellite tracking occurs.

During satellite acquisition the TANS sends packet 41 (time), and status packets on a periodic basis. Although the status packets can be processed to determine the state of satellite tracking, by far the easiest way to determine if tracking is active is by the presence of packet 56 (velocity North-East-Up). This packet occurs only when sufficient satellites are being tracked to calculate position fixes.

With packet 56 present in the data, packet 4A represents the actively calculated position and time. The one pulse per second time tick is now produced by the TANS and corresponds to packet 4A time. One note of caution regarding the use of the packet 4A time when correcting the SAIL clock time - packet 4A data lags the time tick it represents. This lag can vary depending on the TANS automatic inclusion of status or time packets in the data stream. Since this delay is difficult to keep track of, once the SAIL clock is adjusted to the GPS time, it must be checked to verify that it is actually on the same second as the GPS time. This check needs to be repeated to verify the SAIL clock has the correct time.

3.3 Dynamic Motions Measurement System

In an effort to gain some insight into the problem of cable fatigue at the surface buoy's cable termination, a data logger and associated sensor suite has been installed on the S-Tether mooring. This logging package collects data on cable tension and two axes of buoy tilt.

An Onset Computer, model VI Tattletale, was selected for use as the data logger. This particular model Tattletale is equipped with 224K bytes of RAM and a 200M byte hard disk, allowing for 928 data files to be stored during a deployment. In addition, the model VI has 14 user accessible, digital I/O lines, a real-time clock/calendar function and a 12 bit, 8 channel A/D converter. Tattletale programming is done using a low level variation of BASIC known as TT Basic (Tattletale Basic). The user

Item	Quantity	Reference	Part
1	2	C1, C2	47C uF
2	1	C3	0.1 uF
3	3	D1, D2, D3	1N4148
4	1	JP1	Header, 2 Pin
5	1	Q1	IRFD 9123
6	1	Q2	2N7000
7	2	R1, R5	69.8 K
8	2	R2, R6	309 K
9	11	R3, R7, R9, R10, R13, R15 R16, R17, R18, R19, R23	100 K
10	2	R4, R8	140 K
11	4	R11, R12, R20, R22	20.0 K
12	1	R14	464 OHMS
13	1	R21	10 K
14	1	R24	1.00 M
15	3	R25, R26, R27	2.00 K
16	2	U1, U2	OP490
17	1	U3	ICL7660

TABLE 3: S-Tether surface electronics data logger

Data Storage:

RAM	224K (7*32768)
EEPROM	128 Bytes
Hard Disk	200M Bytes

Tilt Sensor:

Range	+/- 45 deg.
* Resolution	0.025 deg.
Linearity	+/- 1% of angle
Freq. Response	0.5 Hertz
Time Constant	0.3 sec.

Tension Cell:

Range	11,000N (2500 lbs)
* Resolution	3N (0.7 lbs.)
Linearity	+/- 0.5% F.S.

Supply Requirements:

Input Voltage	9-13 vdc
Min. Compliance	2.5 Amps

Supply Current:

Idle	3.5 mA (2741 min/datafile)
Logging	15 mA (138 min/datafile)
Disk Access	1.5 amps (15 sec/datafile)
** Avg. Current	4.2 mA

* The resolution is a function of the 12 bit A/D converter, not the sensor.

** Based on the present sampling routine. This does not include the power required to offload data.

TABLE 4: Dynamic motions measurement system specifications

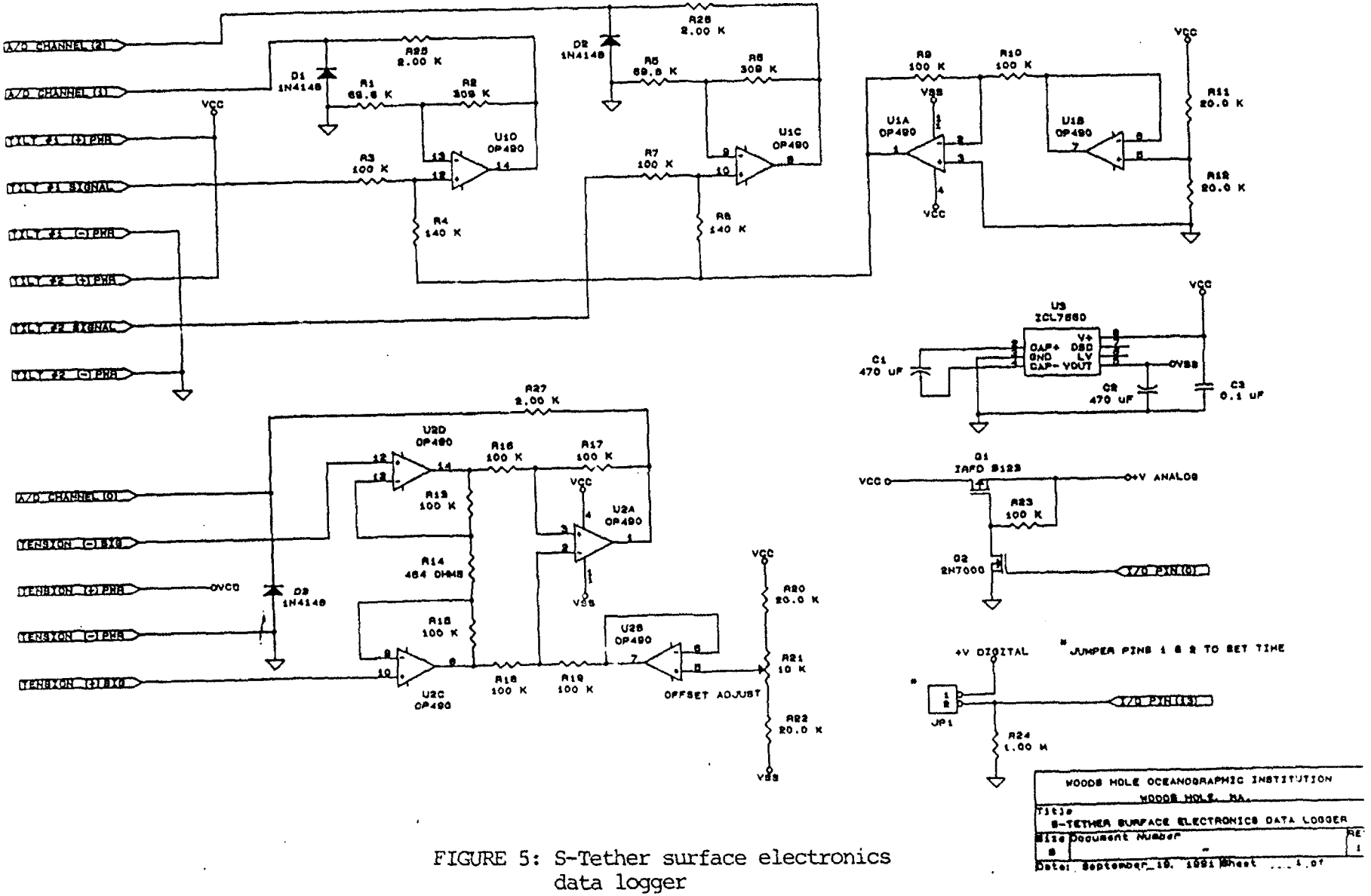


FIGURE 5: S-Tether surface electronics data logger

has the option of storing the application program in RAM, or in EPROM where it is executed at power up.

A Metrox Inc. 11000N (2500 lb.) compression type load cell has been installed at the cable termination point of the surface buoy. This cell is a 700 ohm, foil strain gauge bridge with an output sensitivity of approximately 2 mV/V. When interfaced to the Tattletale, it provides tension data with a resolution of approximately 3N per count. Two axes of tilt are measured with a pair of Lucas Sensing System Inc. clinometers. The tilt data are logged with a least count resolution of approximately 0.025 degrees per count. These sensors have a linear range of $\pm$ 45 degrees and a cross axis error of less than 1%. Dynamically they have a time constant of 0.3 seconds and a frequency response of 0.5 Hertz.

Sensor signal conditioning has been implemented by using standard low power, analog measurement techniques. The power to these analog circuits and sensors is switched by a MOSFET transistor set, Q1 and Q2 to conserve power when the logger is idle (see Figure 5 and Table 3). Load cell signal conditioning utilizes a three OP-AMP (U2A, U2C, U2D) high impedance differential amplifier with a provision for offset adjustment (U2B). Tilt sensor interfacing is done with a straightforward non-inverting amplifier design (U1C, U1D), the sensor output being offset negative to deal with the sensors ratiometric output. U1A and U1B form the negative voltage reference used to offset the sensors. It is designed to track the +5V supply. The Tattletale's lack of a negative supply rail required the addition of U3, a high efficiency switching voltage inverter, used to supply the V(-) rail to the operational amplifiers.

The sampling interval and frequency is set (in software) to 8192 samples per record, measured at four Hertz, twice a day. Each record is preceded by the sampling period's start date and time, and is followed by the period's finish date and time. The total amount of data stored in RAM per record is therefore 49164 bytes. This necessitates transferring the data stored in RAM to the hard disk once every two days. With this sampling schedule the data logger has enough data storage capacity for a five year deployment. System specifications are shown in Table 4.

3.4 Other Systems

Navigation Light

A small xenon strobe light is used to provide night lighting. The strobe unit flashes once every four seconds and draws about 10mA on average at 12V. A day/night switch is used to conserve power. The light is mounted on a mast about 1.5m above the water.

Back-up PTT

A second Argos PTT is installed in the surface buoy along with a second antenna to provide a redundant location mechanism in case the main PTT fails. This second unit also reports the battery voltages from the main and secondary power supplies and tension data from the tension cell. It is programmed to transmit at 90-second intervals.

Power Supplies

There are three battery power supplies in the surface buoy. The main supply consists of four stacks of ninety alkaline D-cells providing 360 amp hours of power at 12V to the IBC and associated electronics. The second supply provides 90 amp hours at 12V to the back-up PTT. The third supply is a rechargeable battery made up of lead-acid batteries (Gates D-cells wired in series) with a capacity of 10 amp hours at 12V. A single 10 Watt solar panel supplies the recharging power for this pack through a Polar Products regulator. The rechargeable battery powers the navigation light. It is mounted in a separate, vented housing outside the main instrument well (see Figure 6).

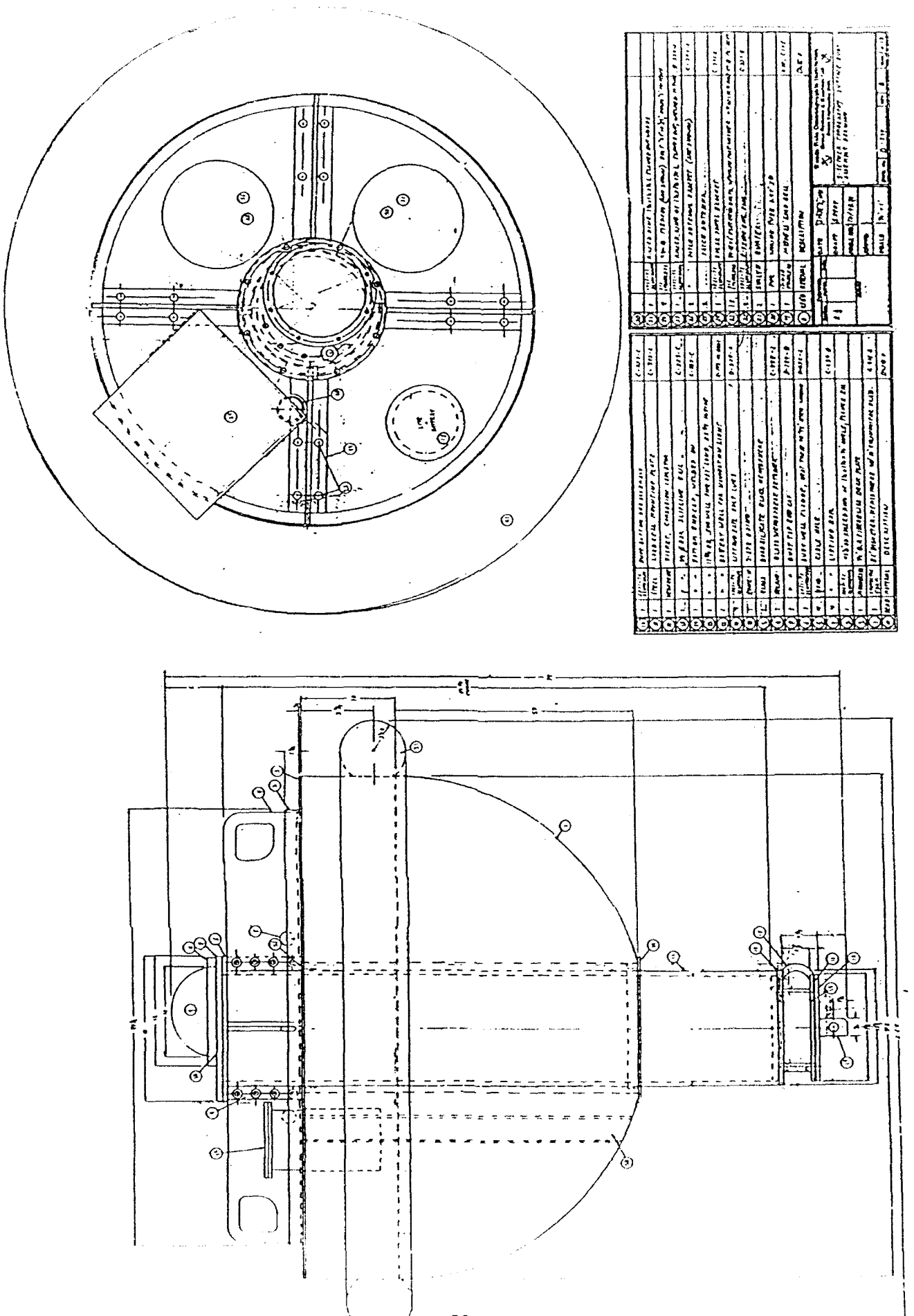


FIGURE 6: Surface buoy assembly drawing

4.0 DATA PROCESSING AND TELEMETRY

4.1 Peak picking

Peak picking is performed as part of the virtual device "calculate" (driver 5). In order for this to take place, it is necessary that the receiver data be collected into a buffer at the time that the receiver device operates (driver 2). When the calculate device operates, it simply uses the current demodulated data buffer.

Peak picking is accomplished by producing a replica of the transmitted code, cross-correlating that replica with the demodulate data, and selecting all the local 3-point maxima from that correlation. The resultant maxima, or peaks, are then sorted to determine a predefined number of the highest peaks. The replica of the transmitted signal is generated using the same sequences that were coded into the source signal: an initialization sequence, a sequence law, and the sequence degree. These must be input into the peak picking code prior to compilation and deployment. A cross-correlation is performed on the imaginary part as well as the real part of the total reception to identify the actual signal within that reception. Once the signal is identified, signal intensity can be computed.

Because of limited data storage and the absence of floating point arithmetic in the processor, the sum of the absolute value of the complex parts of the reception is used to define signal intensity rather than the sum of the squares of the complex parts. This process does not hinder locating peak positions although the level of the intensities will differ (see Figure 7). Absolute intensities can be calculated later once the peak position is located. Each invocation of the peak picking routine results in 32 selected peaks or data points. Each data point consists of an index to the location of that peak in the data buffer and five complex correlates. These five values consist of the largest value or peak with two adjacent sets of correlates on each side of the peak. All 32 peaks are saved in a buffer.

Each complex correlate is two long integers; a total of 8 bytes binary. The index is an unsigned integer which is 2 bytes. The format of the peak in binary is as follows:

Byte	Meaning
1,2	index (n)
3,10	correlate at n-2
11,18	correlate at n-1
19,26	correlate at n (the peak)
27,34	correlate at n+1
35,42	correlate at n+2

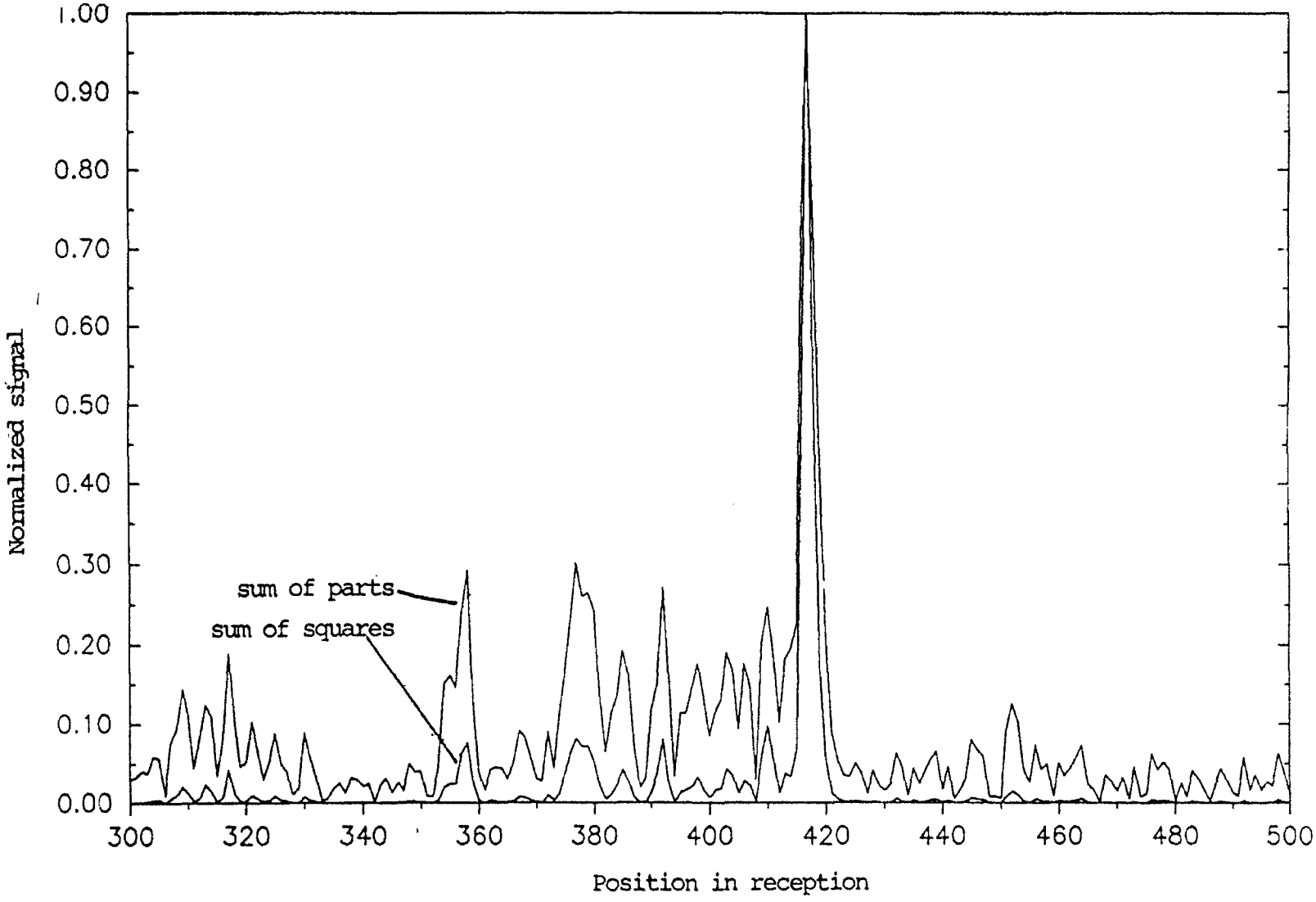


FIGURE 7: Comparison of sum of complex parts and sum of squares

When the peak picking is performed, the time is stored along with the data for inclusion in the message. This is ASCII data formatted as follows:

Char	Meaning
1,4	day
5,6	hour
7,8	minute
9,10	second

4.2 Navigation

Whenever the navigator operates, it stores one or more sets of data in its status subrecord in memory in preparation for being stored on disk. As part of the calculate device, one of these data sets is retrieved and included in the message which is sent to the IBC as part of the Argos message. The data set consists of three integers which represent the round trip travel time to three acoustic transponders. The values are counts of 200 μsec and must be processed as described in the navigator portion of the SeaScan Manual [12]. These values are ASCII data formatted as follows:

Char	Meaning
1,4	ch 1 range
5,8	ch 2 range
9,12	ch 3 range

4.3 Engineering and Ambient Noise

The receiver stores a set of data describing its operation in its status subrecord in memory in preparation for being stored on disk. As part of the calculate device, the receiver gain and ambient noise measurement are retrieved from this record and included in the message which is sent to the IBC as part of the Argos message. When the parameter measurement device operates, it stores the parameter data in its status subrecord in memory in preparation for being stored on disk. As part of the calculate device, these values are retrieved and sent to the IBC. The format of this portion of the data as sent to the IBC in ASCII is as follows:

Char	Meaning
1,2	receiver gain (rx status, byte 31)
3,6	ambient noise (byte 37,38)
7,10	int temp (par status, byte 17,18)
11,14	int pres (byte 21,22)
15,18	ext temp (byte 25,26)
19,22	ext pres (byte 29,30)
23,26	voltage 1 (byte 33,34)
27,30	voltage 2 (byte 37,38)
31,34	voltage 3 (byte 41,42)

4.4 Data Transfer to IBC

When the calculate device has completed peak picking and retrieving data, it addresses the IBC and sends the data using the DB command. The format is as follows:

Byte	Char	Meaning
1,42	1,84	data point 1
43,84	85,168	data point 2
85,126		
...		
1303,1344	2605,2688	data point 32
1345,1349	2689,2698	time
1350,1355	2699,2710	navigation
1356,1372	2711,2744	engineering

4.5 In situ Time Check

The in situ time check is performed as a virtual device, called "time check", which is scheduled in the task table and executed by the system controller. The transceiver is a SAIL master which addresses the IBC and commands it to perform various functions to achieve the time check. The task is performed in the following steps. Initially, the IBC is addressed and commanded to do the time setup. This entails the IBC powering the GPS receiver, then using its time to set the SAIL clock in the IBC. The difference between GPS time and IBC time is termed Set Residual. Next, a time check from the transceiver clock is requested which also causes the SAIL clock to capture its time, thus providing the time Offset between the transceiver clock (Tomo time) and the IBC clock. The IBC then stores the Set Residual and Offset times for inclusion in the Argos message. Figure 8 shows the flowchart for this procedure.

4.6 Application of Time Corrections

The telemetered data is used to perform the time corrections. First the SAIL clock and transceiver clock are compared to ensure that they are on the same second. Then the Offset is computed as described above. These Offset values from successive clock checks give the transceiver clock drift used to correct the acoustic arrivals. The Drift Rate of the SAIL clock oscillator (a TCXO) is calculated as follows:

$$\text{Drift Rate} = \text{SAIL Drift} \times 10^{-6}/\text{Active Period}.$$

If the Active Period is less than about 30 seconds and the Drift Rate is less than about 1×10^{-6}, the time check is working as desired. If either of these increases appreciably, further corrections can be made.

4.7 Argos Telemetry Formats

The Argos telemetry message is made up of the following parts:

- Acoustic correlates and time
- Navigation
- Tomography receiver engineering and ambient noise data
- Time offset measurements (Tomo clock against GPS)
- GPS position
- GPS receiver status

The first three parts are assembled in the transceiver then transmitted to the IBC. The IBC adds the latter three parts to form the complete Argos message - see Table 5a. The Argos buffer is formatted into 48 32-byte buffers. Each 32-byte buffer has an ID number (1 byte) followed by 30 bytes of data and a single byte checksum. The allocation of the data to the 48 buffers is shown in Table 5b (buffer 48 is not full). These data are transmitted as 32 byte records every 15 seconds.

Table 5a: Complete Argos buffer (binary)

Byte	Meaning
1,1372	data buffer
1373, 1392	time check data
1393,1394	time check status
1395,1414	GPS position
1415,1417	GPS status

Table 5b: Argos messages (as transmitted)

Buffer	Bytes	Buffer	Bytes
1	1,30	25	721,750
2	31,60	26	751,780
3	61,90	27	781,810
4	91,120	28	811,840
5	121,150	29	841,870
6	151,180	30	871,900
7	181,210	31	901,930
8	211,240	32	931,960
9	241,270	33	961,990
10	271,300	34	991,1020
11	301,330	35	1021,1050
12	331,360	36	1051,1080
13	361,390	37	1081,1110
14	391,420	38	1111,1140
15	421,450	39	1141,1170
16	451,480	40	1171,1200
17	481,510	41	1201,1230
18	511,540	42	1231,1260
19	541,570	43	1261,1290
20	571,600	44	1291,1320
21	601,630	45	1321,1350
22	631,660	46	1351,1380
23	661,690	47	1381,1410
24	691,720	48	1411,1440

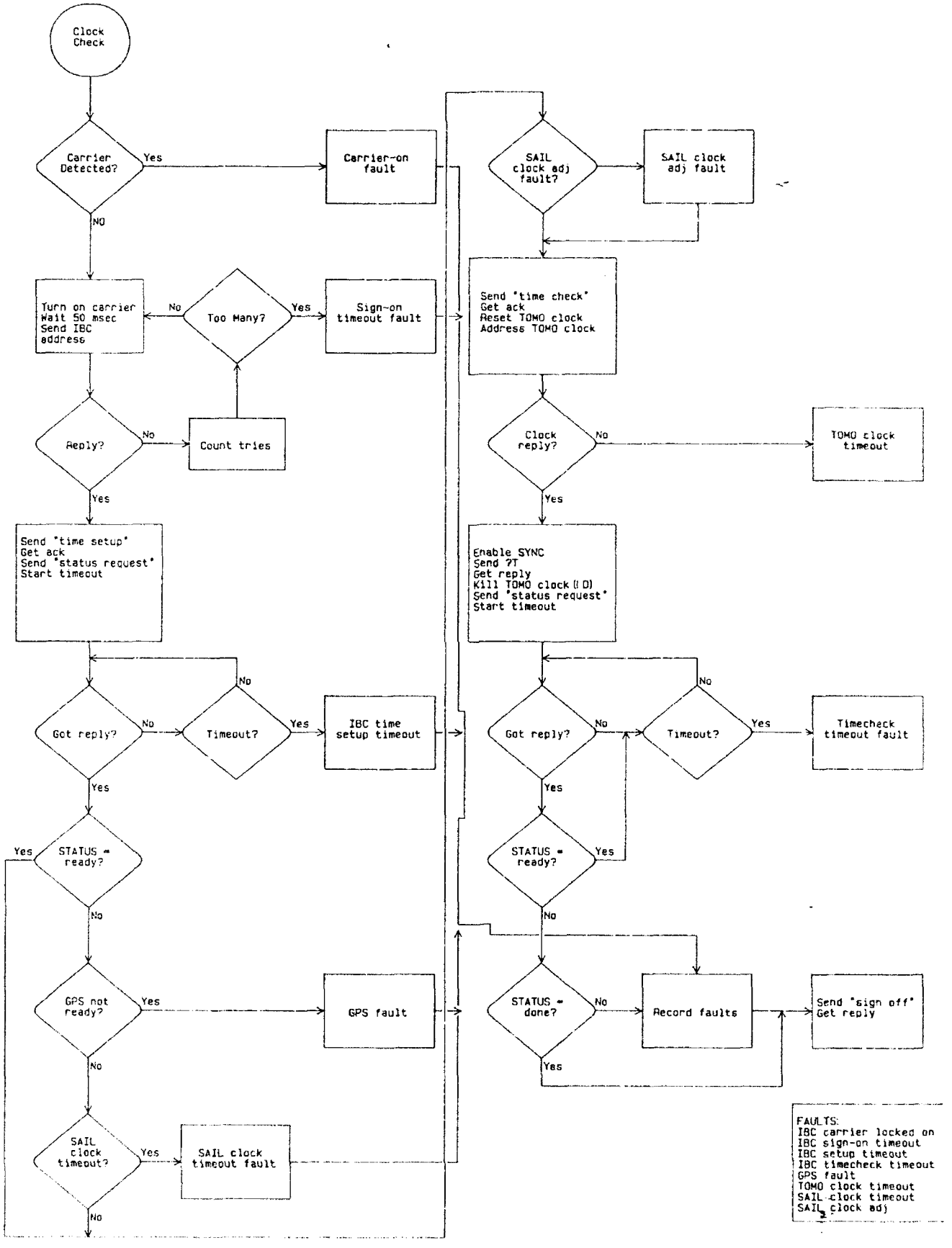


FIGURE 8: Transceiver program for clock check

4.8 Transceiver Record Format Changes

The addition of two virtual devices to the controller was accompanied by two new status types in the disk status subrecord. These are listed below.

Local Status for time check device

0	device number (11)
1-4	current time
5	device SAIL status
6-9	nominal start time
10-11	configuration bytes, not used
12-16	TOMO time
17	IBC time check status (x - one alpha character)
18	IBC time check status (f - one ASCII hex 0-F)
19	TT time check status (faults - bit pattern)

Allocations

x	R ready N not ready D done F fault
f	bit 0 GPS receiver not ready 1 SAIL clock timeout 2 SAIL clock set problem 3 TOMO clock timeout
fault	bit 0 GPS receiver not ready 1 SAIL clock timeout 2 SAIL clock set problem 3 TOMO clock timeout 4 IBC carrier stuck on 5 IBC sign-on timeout 6 IBC time setup timeout 7 IBC time check timeout

Local Status for calculate and data transfer device

0	device number (12)
1-4	current time
5	device SAIL status
6-9	nominal start time
10-11	execution time (not used)
12	(not used)
13	receiver to use (not used)
14	data mode - real (0)/test (1)
15-40	partial data point (index and 3 correlates)

5.0 S-TETHER MOORING SYSTEM

The S-Tether mooring system is designed to provide a reliable electrical connectio between instruments deployed on standard subsurface moorings and the surface. Th surface connection then allows the use of various RF techniques to relay data to th instruments and to pass data back from the instruments to the user. In the cas of the tomography mooring, real-time tomography data are telemetered via satellit along with precise timing information derived from the onboard GPS receiver whic allows corrections of the real time data to be made immediately. The mooring desig is shown in Figure 1 and discussed in detail in the following sections.

5.1 Surface Buoy

The surface buoy, shown in Figure 6, is designed to support its instrument payloa and in the case of extreme currents to protect the payload to a depth of 300m. Th surface buoy hull is composed of a syntactic foam hemisphere with a cylindrica extension on top for added buoyancy. A removable axial metal structure sandwiche the foam between the deck superstructure and a lower ring. A tubular aluminum instrument well extends through the foam. Buoy specifications are shown in Table 6

Table 6: Surface buoy specifications

Diameter of foam	142 cm
Diameter with bumper	183 cm
Height overall	188 cm
Height inside well	149 cm
Well Diameter	30.5 cm
Air weight (Empty)	607 kg
Payload	136 kg
Air weight (loaded)	743 kg
Total Buoyancy @ Surface	682 kg
Total buoyancy at depth	523 kg
Depth rating Inst. well*	300 m
Depth rating Syntactic foam*	300 m
Lifting bail rating	1400 kg

* Working depth. Crush depth is approximately 400 meters.

The frame and central well are type 6061 aluminum, heat treated to "T6" condition after fabrication. **[NOTE: Do not weld anything to central well; this would reduce the crush depth to less than 100 meters.]** Rods and fasteners

are type 316 stainless steel with nylon isolation bushings. The buoy is made of syntactic foam with 5 cm plastic balls in aggregate suspension for an average density of 0.377 g/cm. Secondary surface-only flotation of approximately 160 kg is provided by a surlyn foam bumper that encircles the buoy.

One 10-Watt solar panel, two patch style Argos satellite antennas and a self-contained, external, deck mounted, battery cannister to power a mast mounted navigation light are mounted on the buoy deck. An 11,000 N underwater strain gauge shear type load cell provides the mooring attachment point at the bottom of the buoy. A GPS antenna is mounted in a pressure proof housing made from a 25.4 cm diameter glass hemisphere and is located on the top end cap.

The buoy's natural period in roll is 0.7 to 0.9 seconds and its metacentric height is 7.77 cm with no payload. The metacentric height is 13.82 cm with a 136 kg payload. These small values are typical for buoys with hemispherical hull shapes.

The primary flotation for the mooring, a 1360 kg syntactic buoy, is located at 100 meters below the surface. A compliant inverse catenary tether connects the surface buoy to the top of the subsurface buoy. In the event that the mooring is subjected to a period of strong currents, the subsurface mooring may dip to the extent that the surface buoy is pulled under. The surface buoy is designed to survive indefinitely at depths of up to 300 m. The navigation light, and its solar panel and deck mounted battery case are not designed to survive these extreme depth excursions, but the flooding or total loss of this completely independent subsystem will not affect the performance of the rest of the system.

The surface buoy develops a weak positive righting movement at more than 90 degrees from its normal orientation. It follows the waves in heave but is relatively unresponsive in roll due to the small lateral center of buoyancy migration with surface slope and the high damping caused by the lower extension of the central well.

The aluminum central well tube is a 157 cm long length of 32.4 cm, O.D. (30.5 cm ID) pipe. A 19 mm thick plate is welded to the bottom forming the lower end plate. The top of the tube has a 40.6 cm diameter welded flange. A bolt on top end cap mates to the flange with a double O-Ring seal.

The mooring attachment point is a tang with a hole that is an integral part of a shear type tension cell. The 17-4-PH stainless steel tension cell is bolted to a steel plate which acts as an anode. This plate bolts to a flange on the instrument well tube which extends from the bottom of the buoy. The steel plate is isolated from the aluminum structure by a neoprene gasket and nylon bushings on the bolts.

Four lifting bails bolt onto the central well. The strength of the bails is transferred to the well by four 1.0 cm bolts loaded in single shear. Additional bolts go into threaded inserts in the syntactic foam. The bolts in the foam act primarily to key the deck structure against rotation relative to the foam. The foam is fixed to the central well structure by four 1.0 cm threaded rods that join the angle rolled ring welded to the deck structure to a sliding rolled ring that pulls up on the bottom of the foam.

A PVC pipe extends through the foam to act as a cabling conduit. The bottom of this pipe is attached with standard PVC plumbing fittings to the tube that encapsulates the load cell such that there is no exposed wiring below the water line. Assembly drawings for the surface buoy and tether attachment are included as Appendix F.

5.2 Tether Description

Attaching electromechanical cables to surface and subsurface buoys in such a way that the buoy motion does not cause premature cable failure is a difficult problem. Various bending strain relief boots of elastomeric materials have been tried with less than satisfactory results. In most cases the kevlar or steel wire armor has failed in bending and torsional fatigue. Most of the torsion appears to be caused by loss of torque balance as the strength member deteriorates. The bending is caused by wave forces acting on a surface buoy.

The approach that has shown the best results to date has been an encapsulated chain approach (Dwg # B370-A in Appendix F). Several moorings using this design have been deployed to date with no known failures. Encapsulated chains have been used to connect E/M cables to subsurface buoys on the Tethermoor [13] and Abrupt Topography [14] projects. They have connected E/M cables to surface buoys on the STEM [15], STRESS [16] and Inductive Mooring [17] projects.

In these applications the electrical cable is spiraled around a length of chain and then encapsulated in urethane inside a tough abrasion resistant hose that also acts as a mold. The chain can flex within the matrix to allow the buoy to move without fatigue to the cable due to cyclic bending. The spirally wound electrical cable reduces the bending fatigue in the copper. The urethane matrix minimizes relative motion, prevents corrosion and provides a secondary barrier against damage from outside sources such as fishbite. The radiator hose provides an additional barrier against damage and adds to the stiffness of the resulting structure.

The buoy attachment is made with a flat topped clevis butted against the bottom of the buoy that does not allow relative motion. A three meter length of chain is pinned to the bottom of this clevis. The other end of the chain is pinned to a similar fitting ending in a tang or clevis that fits the end of the electromechanical cable, again shaped to prevent any relative motion.

Fabrication of the chain assembly proceeds as follows. The chain, with both ends attached, is suspended in tension from overhead. An electrical cable with connectors on each end is wrapped in a spiral, loosely around the chain. A single cable tie can be used every meter or so to hold it in place. A length of rubber hose is prepared and then slipped over the entire assembly from the top. A hole the size of the electrical cable is made in the hose about 15 cm from the bottom. A knife slot extends the hole to the bottom end. Additional holes are punched at intervals of about 38 cm to allow for the urethane injection. A slot, milled into the top fitting, receives the electrical cable and allows the hose to be slipped over it. When the hose reaches the

bottom fitting, the electrical cable is pulled up through the slot and into the hole. The bottom of the hose is secured to the middle puck of the bottom fitting. Self-vulcanizing rubber tape is used to close the slot. The urethane is injected from the bottom up and each fill hole is packed with butyl tape and wrapped with vulcanizing tape as it is reached. The hole in the center of the top fitting acts as a vent and signals when to stop pouring. An additional pour may be made to create a short tapered bending strain relief boot where each end of the electrical cable exits the hose. It is important to allow enough cable length for an adequate lead on both ends as well as some length for the wraps.

Between the molded chain segments attached below the surface buoy and above the subsurface buoy is the electromechanical tether which provides both the electrical connection to the surface and a compliant mechanical connection. This cable is shown in Figure 9. It has a double armor construction with three conductors in its core. It is jacketed with a thick (3.4cm) rubber coating to prevent hockles from forming when the cable is slack. It has a lead weight (18kg) installed 15m from the surface to minimize the angle at the buoy connection and several small syntactic floats installed just above the subsurface buoy to keep the cable from dipping down below the buoy and getting tangled in the mooring. At each end of the 200m tether, a mechanical termination is made by bird-caging the armor wires and potting them in a specially machined clevis with epoxy (see Dwg. C-217 in Appendix F). The insulated electrical conductors are led through this fitting and spliced to a Brantner pigtail. A water-blocking puck of urethane is used in this process to eliminate water leakage at this critical juncture. After the mechanical and electrical connections have been made in the field, the entire joint is wrapped in self-vulcanizing tape to protect it from chafe and bending strain.

The tether is connected to a 3m molded chain section above the subsurface buoy and from there to the 800m shot of E/M cable that connects with the tomography transceiver. The E/M cable is identical to the cable used in the tether, but without the rubber jacket. Its terminations are made identically to those of the tether.

5.3 Subsurface Buoy

The subsurface buoy is cast of syntactic foam into a spherical shape with a total of 13,350 N of buoyancy. A removable metal structure carries the mooring tension from top to bottom. A 6.4 cm diameter conduit is cast into the foam to facilitate and protect electrical cabling. The removable strength member design is an improvement over past designs where the strength member was cast in place. The new design allows the strength member to be removed and inspected for fatigue or corrosion damage between deployments.

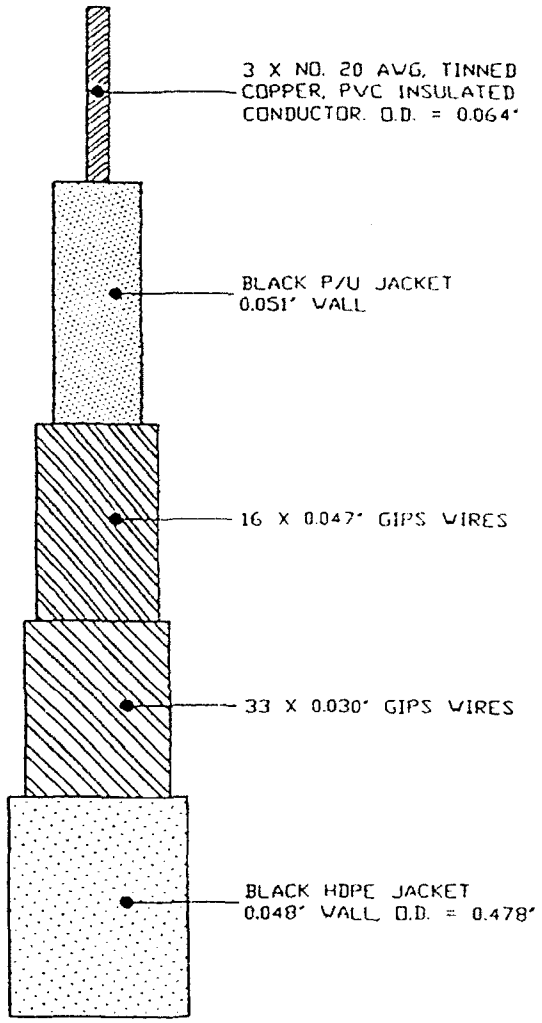


FIGURE 9: Steel double armor E/M cable

5.4 Electromechanical Cables and Terminations

There are two candidate electromechanical cables for use on this type of mooring system. The first candidate is double armor steel. This cable consists of two contra-helically wound layers of GIPS armor wires over an electrical core (see Figure 9). The core consists of three stranded, twisted # 20 copper electrical conductors, each with a color coded PVC jacket. The three conductors are helixed into a bundle and then the bundle is extruded to round with a polyurethane compound. The armor wires surround the core and are in turn extruded to round with high density polyethylene. The tether cable has an additional TPR jacket over this construction.

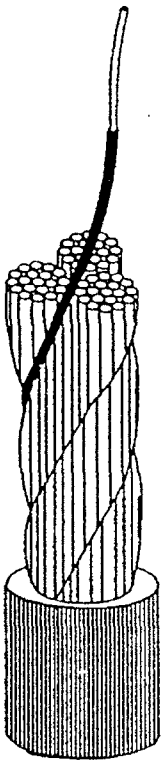
The second candidate is a wire rope construction. The cable strength member is MacWhyte torque balanced 3x19 wire rope (see Figure 10). The three PVC jacketed electrical conductors are helixed into the valleys between the wire rope strands. The cable is then wrapped with mylar tape and extruded to round with a high density polyethylene jacket material. The double-armored cable will be used on the first pilot deployment of the tomography system. The double armored cable is terminated in an epoxy filled socket termination specifically designed for this cable. A polyurethane bending strain relief boot slides over the cable and termination backshell. The boot is designed with a slight interference fit to minimize corrosion by limiting water migration and thereby causing oxygen depletion in the corrosion reaction. The electrical core is inserted into a length of urethane tubing which is then potted with urethane. The ends of the electrical wires are soldered to three brass pins which pass through a urethane puck. An off-the-shelf underwater pigtail, (Brantner, Impulse or Burns), is soldered to the other end of the pins. A small mold is then placed over this electrical splice and filled with urethane. The resulting termination has proven reliable during a number of previous experiments.

The wire rope construction may be used on some future deployments. The 3x19 steel strength member of the cable is terminated with an oversized swage fitting as per standard WHOI practice. The electrical wires are broken out of the HDPE jacket just at the base of the swage fitting and are spliced onto a pigtail with the brass pins as previously described. The whole assembly is then placed into a mold and the bending strain relief boot is potted in place.

5.5 Mooring System Design

The mooring shown in Figure 1 was designed for a short term test of the system at Site D (39° N, 70° W) in 2700m of water south of Cape Cod. The design concept, however, is applicable to anywhere in the deep ocean where extreme currents will not submerge the surface buoy beyond 300m depth. Periods when the buoy is underwater should be minimized since data cannot be telemetered during these times.

Figures 11a and 11b show the current profiles used in developing the mooring design for Site D and for a generic open ocean site in the North Atlantic, respectively.



(3) STRANDED
COPPER CONDUCTORS
AWG #20

INSULATION:
0.015" WALL PVC
0.010" WALL NYLON

5/16" 3x19
TORQUE BALANCED
WIRE ROPE
(MacWhyte)

OUTER JACKET:
Polyethylene
0.054" WALL
O.D. = 0.500"

FIGURE 10: 3x19 torque-balanced wire rope with three electrical conductors

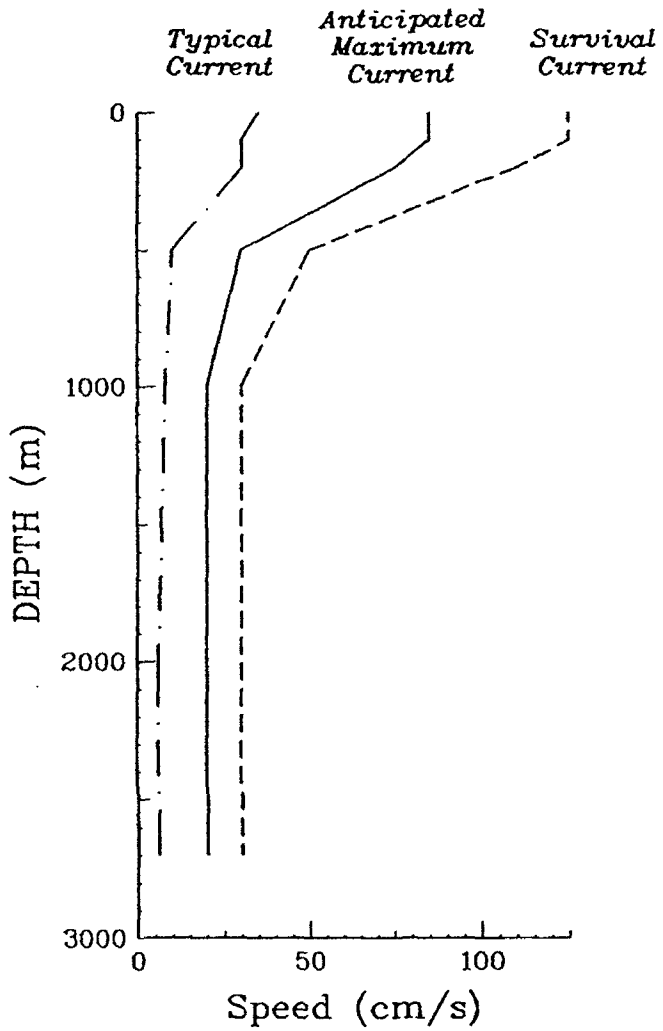


FIGURE 11a: Current profiles used to design the real-time tomography mooring for deployment at Site D.

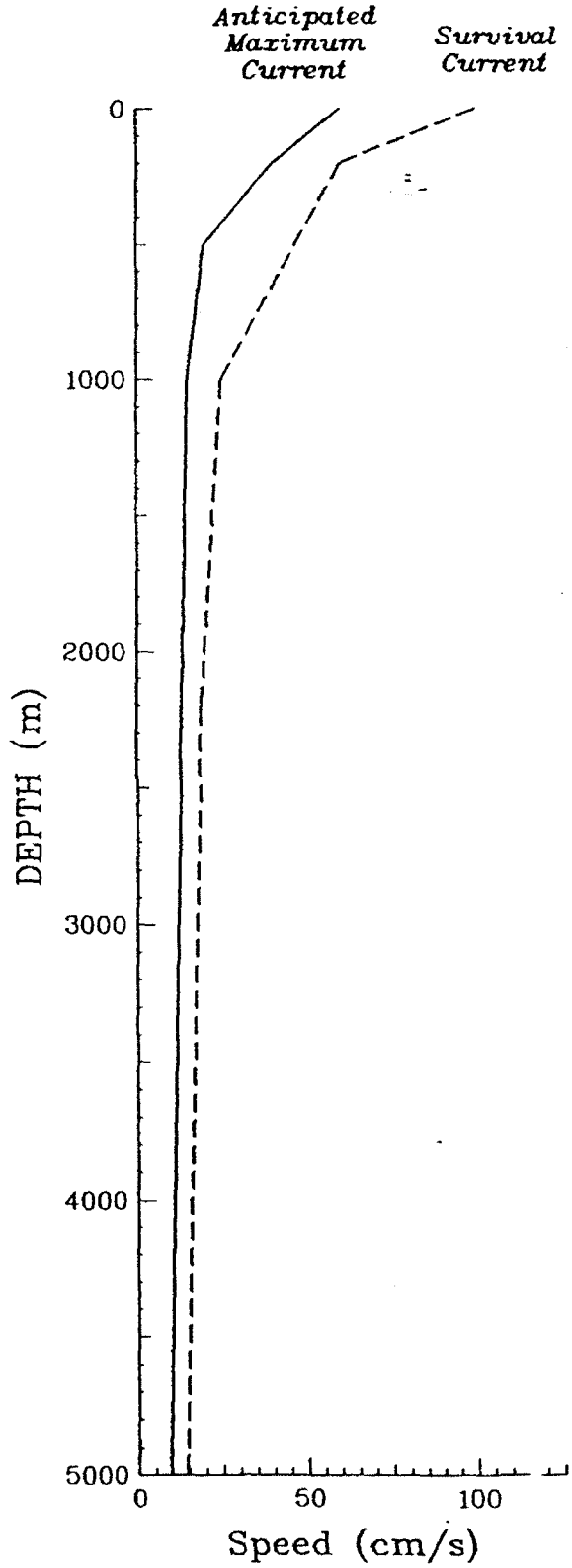


FIGURE 11b: Current profiles used to design a generic version of the real-time tomography mooring.

Table 7 shows the results of computer modeling of the mooring designed for Site D under the design current profiles. It is clear from these results that the surface buoy will be submerged only during periods of extreme currents at Site D, such as when an energetic warm core ring impacts the site. Care should be taken, however, when using these results since they have assumed constant values for the drag coefficient of mooring components such as wire rope, E/M cable, kevlar, etc. No provision has been made for increased drag associated with cable strumming, if present. Current data for the Site D designs is from data collected by WHOI since the late 1960s.

The SURFMOOR [18] mooring design program was used to evaluate the forces on the mooring and its resulting geometry in response to the current profiles. When the currents are sufficient to submerge the surface buoy, a subsurface mooring program, NOYFB, was used to predict the maximum depth of submergence [19].

Current Profiles

Depth (m)	Average Velocity (cm/s)	Expected Maximum Velocity (cm/s)	Survival Velocity (cm/s)
0	35	85	125
100	30	85	125
200	30	75	110
500	10	30	50
1000	8	20	30
2700	6	20	30

Model Results

Parameter	Average	Exp. Max.	Survival
Tension @ surface buoy (lbs)	76	1200	1200
Depth of surface buoy (m)	0	48	181
Instrument watch circle radius (m)	135	618	843
Instrument depth (m)*	902	1046	1149
Angle of transceiver (Deg)	4	19.7	26.6
Surf. buoy watch circle (m)	331	873	1180
Depth of subsurface buoy (m)**	104	234	351

NOTE: Angles are relative to vertical, watch circles are radii about the anchor.

* Instrument depth with no current is 900m.
** Subsurface buoy depth with no current is 100m.

TABLE 7: Results of computer model runs for the real-time tomography mooring

6.0 CONCLUSIONS AND RECOMMENDATIONS

6.1 Present Status of Project

At the date of writing this final section of the report (January 1992), the prototype S-Tether unit for tomography is a reality. The buoy, with its associated hardware and software, exists and is being protectively stored until its first at-sea deployment. The minor modifications to the Webb transceiver needed for interfacing it to the S-Tether buoy have also been made. The documentation of the project exists in the form of this report.

In terms of whether our efforts are useful in the long term, we should be able to affirmatively answer two questions. Specifically: 1) does tomography work well enough to give significantly improved maps of the ocean, which are not obtainable from other sensors at lower cost, and 2) does S-Tether really compete with some of the "newer" telemetry schemes being proposed. The authors of this report insist on answering both of these questions affirmatively before going further.

Two large-scale tomography experiments deployed during 1988-89 were the Gulf Stream and Greenland Sea efforts. The Gulf Stream array, an MIT/WHOI effort using five Webb transceivers, successfully measured the radiation of eddy energy from the Gulf Stream. Temperature, current and vorticity fields were computed from the travel time measurements; these measurements compared favorably to conventional measurements where comparisons could be made, and were superior to conventional measurements for the vorticity field. The Greenland Sea tomography experiment, a joint SIO/WHOI effort, was designed to measure deep water formation and circulation in the Greenland Sea. Inversions of the data show that deep convection was indeed observed by the array. For the first time 3-D time series maps of a deep ocean convective event are being made. Current fields were also measured successfully, even though a mix of two different tomographic instruments (the SIO "Avatar" and Webb transceiver) with different instrument responses and delays were employed. These two experiments give an idea of what tomography is capable of providing to both the oceanographic and operational Navy communities.

As to S-Tether's effectiveness compared to other telemetry technologies, we have already discussed that topic in Section I, so that we will only present a summary in Table 8. S-Tether's major selling points, as far as we are concerned, are: 1) low-cost (compared to **any** other alternative except SOSUS), 2) arbitrary location capabilities, 3) surface RF communications link (especially powerful if microsatellites eventually replace ARGOS), and 4) ease of deployment and recovery. These strengths complement other systems which work better near the continental shelf (SOSUS, acoustic telemetry), in that they allow deployment of a fully telemetered tomography array **anywhere** in the worlds' oceans.

6.2 Future Directions

Our current plans for S-Tether call for using it for the first time in the North Atlantic in 1994-95 as part of a large tomographic experiment to the east of the Grand Banks of Newfoundland which will study the North Atlantic Current. This experiment, which would overlap Lagrangian drifter measurements to be made by T. Rossby, would involve up to sixteen tomographic transceivers and be a joint effort between WHOI, MIT, SIO, APL/UW, NPS, JAMSTEC (a Japanese lab) and perhaps others. Funding for the ocean science is being requested from ONR. It is hoped that funding for the S-Tether part of the project can be acquired through ONT. Another possible experiment in which the S-TETHER technology could be tested and displayed is the "ETAMBOT" effort of the French in conjunction with WOCE.

Future deployment plans for the instruments will obviously depend on the results of the first deployment, and on new developments in technology (i.e. microsatellites) which will become available in time.

Table 8: Comparison of real-time tomography technologies

A. OLDER CONCEPTS

1. MOORED SOURCE TO SOSUS RECEIVERS

Advantages

a. System is in place; improvements to tomography receivers relatively inexpensive
b. Many receivers, so one source can give many paths
c. High bandwidth

Disadvantages

a. Still need to get clock and mooring motion for remote source in real time
b. One way transmissions only
c. Receiver placement often poor for vertical resolution of ocean features
d. Classified system
e. Will system be maintained in future?

2. PHONE CABLE CONNECTION TO MOORING

Advantages

a. High bandwidth
b. Deep ocean locations possible
c. Two way transmissions possible (reciprocal tomography)
d. All transceiver depths available

Disadvantages

a. Expensive to install and maintain link to cable
b. Cable locations may not be desirable or optimal
c. Will system be maintained in future?

3. S-TETHER

Advantages

a. All deep ocean locations possible
b. Two way transmission possible (both in reciprocal tomography sense and buoy communication sense)
c. All transceiver depths available
d. Relatively cheap (after initial engineering of prototype) - $100K or less/unit
e. Surface expression allows links to RF communications
f. Easily deployed/recovered

Disadvantages

a. Low bandwidth (currently, due to using Argos - will be upgraded)
b. Surface expression (vulnerable to extreme waves, ships; current system expected life is one year)

4. MOVING SHIP TOMOGRAPHY

Advantages

a. Gives soundspeed map in area of fleet activity
b. Can use existing resources (i.e. SURTASS)
c. Gives much better spatial resolution of ocean features

Disadvantages

a. Exact measurement of array position extremely involved and high tech
b. Still need to get clock, mooring motion from source to have true real time
c. Still gives temporally undersampled view of a region due to slowness of ship (i.e., 1 station/4 hours)
d. Labor intensive deployment
e. Analysis is very computationally intensive
f. Only one realization per path; internal waves not averaged out

B. NEWER DEVELOPMENTS (Since S-Tether Inception)

1. INDUCTIVE MODEM ON PHONE CABLE (in development)

Advantages

a. Avoids difficulty of direct attachment of (tomography) mooring to phone cable
b. All transceiver depths possible
c. Long lifetime possible (~5 years)
d. No surface expression
e. High baud rate

Disadvantages

a. ROV deployed (modem)
b. Maintenance of phone cables necessary
c. Phone cable locations may not be desirable or optimal

2. MULTI CZ ACOUSTIC TELEMETRY FROM MOORING TO SOSUS (proposed)

Advantages

a. All transceiver depths possible for mooring
b. No surface expression

Disadvantages

a. High power consumption (~1 Joule/bit)
b. Low total data output ($\simeq$ same as Argos) due to power limit
c. Clock accuracy as good as intermittent rubidium (10-25 msec/yr.)
d. Limited range (few CZ)

3. RELATIVE ARRIVAL TOMOGRAPHY USING SONOBUOYS (Lynch/Duda)

Advantages

a. Can quickly survey a large area (no temporal or spatial undersampling)
b. Needs only LORAN accuracy, wristwatch time
c. Inversions simple and works well in cases examined
d. Very cheap

Disadvantages

a. Gets less information than full tomographic inversion
b. Theory not yet fully developed
c. Only one realization per path; internal waves not averaged out

4. DRIFTING BUOY TOMOGRAPHY

Advantages

a. Full power of tomography, i.e., exact clock, GPS accuracy
b. Drifter has long life (years)
c. Engineering already done (i.e., S-Tether)

Disadvantages

a. "Random array"
b. Only one realization per path; internal waves not averaged out

7.0 REFERENCES:

[1] Munk, W.H. and C. Wunsch, "Ocean acoustic tomography: A scheme for large scale monitoring," Deep-Sea Research, 26: 123-161, 1979.

[2] Cornuelle, B.D., "Acoustic tomography," IEEE Trans. Geoscience and Remote Sensing, GE-20: 326-332, 1982.

[3] Worcester, P., B.D. Cornuelle, and R.C. Spindel, "A review of ocean acoustic tomography: 1987-1990," Revs. of Geophys. (Supplement): 557-570, 1991.

[4] Munk, W.H. and A.M.G. Forbes, "Global ocean warming: An acoustic measure?," J. Physical Oceanography, 19: 1765-1778, 1989.

[5] Spiesberger, J.L., P.J. Bushong, K. Metger, Jr., and T.G. Birdsall, "Basin-scale tomography: synoptic measurements of a 4000-km length section in the Pacific," J. Phys. Oceanogr., 19: 1073-1090, 1989.

[6] Anonymous, "The Applied Tomography Experiment of 1990," Applied Physics Laboratory, University of Washington, Technical Report APL-UW TR9106 (February 1991).

[7] Bruce M. Howe, Bruce D. Cornuelle, James A. Mercer, Kurt Metzger, Peter F. Worcester, "Acoustic Mid-Ocean Dynamics Experiment: 1991 Moving Ship Tomography Cruise," Applied Physics Laboratory Technical Memorandum APL-UW TM1818-91, December 1991.

[8] Duda, T.F. and J.F. Lynch, "Smoothly modulated frequency-bounded impulse signals for tomography," WHOI Technical Report WHOI-91-13, 1991.

[9] Catipovic, J., private communication

[10] Briscoe, M.G. and D.E. Frye, "Motivations and methods for ocean data telemetry," Marine Technology Society Journal, 21(2): 42-57, 1987.

[11] "IEEE standard serial ASCII instrument loop (SAIL) shipboard data communications," IEEE Standard 997-1985, 12 pp., Apr. 1985.

[12] "Tomography Transceiver," manual, SeaScan, Inc., N. Falmouth, MA 02556

[13] D. E. Frye and W. B. Owens, "Recent developments in ocean data telemetry at Woods Hole Oceanographic Institution", IEEE Journal of Oceanic Engineering, 16 (4): 350-359, 1991

[14] D. E. Frye, ed., "Advanced Engineering Laboratory Project Summaries, 1990," WHOI Technical Report, WHOI-91-09, May 1991.

[15] Berteaux, H. O., D. E. Frye, P. R. Clay and E. C. Mellinger, "Surface telemetry engineering mooring (STEM)," Proceedings of the Oceans '88 Conference, Oct. 31-Nov. 2, 1988, 670-680.

[16] Williams, A., private communication.

[17] D.E. Frye, A. J. Fougere and S.M. Kery "Prototype expendable surface mooring with inducive telemetry," poster paper presented at AGU Ocean Sciences meeting, New Orleans, Jan. 1992.

[18] Kery, S., private communication.

[19] Moller, D.A., "A computer program for the design and static analysis of single-point subsurface mooring systems: NOYFB," WHOI Technical Report, WHOI-76-59, June 1976.

★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★

APPENDICES

★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★

Appendix A: Modifications Necessary for Real-Time Operation

1. Addition of FSK modem board

The board is mounted on the module mount plate. It is connected to power at the mass store backplane connector. It is connected to the system controller I/O board and to the external SAIL connector.

2. Wiring change

The external power line on the SAIL connector is reassigned as the FSK line.

3. Modifications to the Onset CPU-8088 board.

The 2.4576 MHz signal from the UART is connected to the CPU counter timer.

4. Modifications to the SeaScan tomography controller I/O board

- use fourth position on the SAIL circulator
- reassign function of J4 pins for FSK
- apply power to the backplane pin

5. Instrument BIOS software changes
(1 PROM on the CPU board)

bioscom.asm
- added baud rate control

biosport.asm
- disable two unused reset lines
- reassign PB0 as FSK carrier control output
- reassign PC1 as FSK carrier detect input
- add function (5) to do timer/counter control
- stop counter when defaults set

6. Instrument scheduler software changes
(3 PROM's on memory board)

c0t.asm
- adjust RAM/ROM boundary

define.c
- new structures

setup.c
- new buffer allocations
- initialize circulator mode

monitor.c
- add FSK commands
- add new devices in ?N list

schedule.c
- add new driver calls

system.c
- adjust reset array for new devices
- add SAIL_FSK function
- add SAIL_CONTROL function
- add FINDER function
- add GETDBUF function
- change SET_CONFIG and GET_CONFIG for new BIOS bits
- add SET_COUNTER to control timer counter

driver2.c
- add buffer transfer and decoding
- use separate buffer for reply

driver5.c
- new driver
- do calculations and data transfer

driver6.c
- new driver
- do time check

get_peaks.c
- new function to do correlation and peak picking

7. Deck support software changes

ET

et.c
- announce s-tether version

define.c
- structure changes

analyse.c
- extra devices

sendtask.c
- extra devices

listtask.c
- extra devices

files.c
- larger buffer

DM

dm.c
- announce s-tether version

define.c
- structure changes

stsdata.c
- extra devices

8. Additional program

IBC-EM
- new program to emulate IBC protocol during testing.
- see report in appendix C.

Appendix B: Environments Used During Development of the Real-Time Tomography Transceiver

1. Hardware setup - see Figure B1

   End cap pigtail to modified deck box (free "ext pwr" line)
   Deck box SAIL to Wet Systems converter box to PC #1
   Deck box FSK (was ext power) to modem-in-a-box
   Modem-in-a-box RS-232 to special cable to PC #2
   Power via battery connector with current monitor
   Backplane monitor box to Wet Systems converter to dumb terminal
   Main dummy load on power amp output
   Cable wrapped around dummy load lead to receiver input
   Nav dummy load on navigator output (0.1uF // 5000 ohms)

2. Software development

   BIOS
   - MASM
   - make file BIOS
   - burn PROM from BIOS.HEX

   scheduler
   - Turbo C 2.0 integrated environment
   - s3wh.prj in schdlr subdirectory
   - MASM for c0t.asm only
   - dohex.bat to make s3wh.hex
   - download using monitor "L" ("@" to zero, first)
   - run using monitor "G" then "200"
   - upload using monitor "W" to three .HEX files to burn PROM

   et
   - Turbo C 2.0 integrated environment
   - s3wh.prj in ET subdirectory

   dm
   - Turbo C 2.0 integrated environment
   - s3wh.prj in DM subdirectory

   IBC-EM
   - Turbo C 2.0 integrated environment
   - ibc-em.prj in cindy subdirectory

3. Not-to-be-forgotten knowledge

   Turbo C
   - default char type must be unsigned or bytes become integers when sign bit is set
   - on input leading zeros imply octal for scan type "i" but not for "d"
   - floating point and "delay()" very DOS-linked, not usable here

   Serial output from TURBO C using BIOSCOM
   - must assert RTS to get output
   - if RTS not asserted, program completes but no output
   - DTR is asserted during output

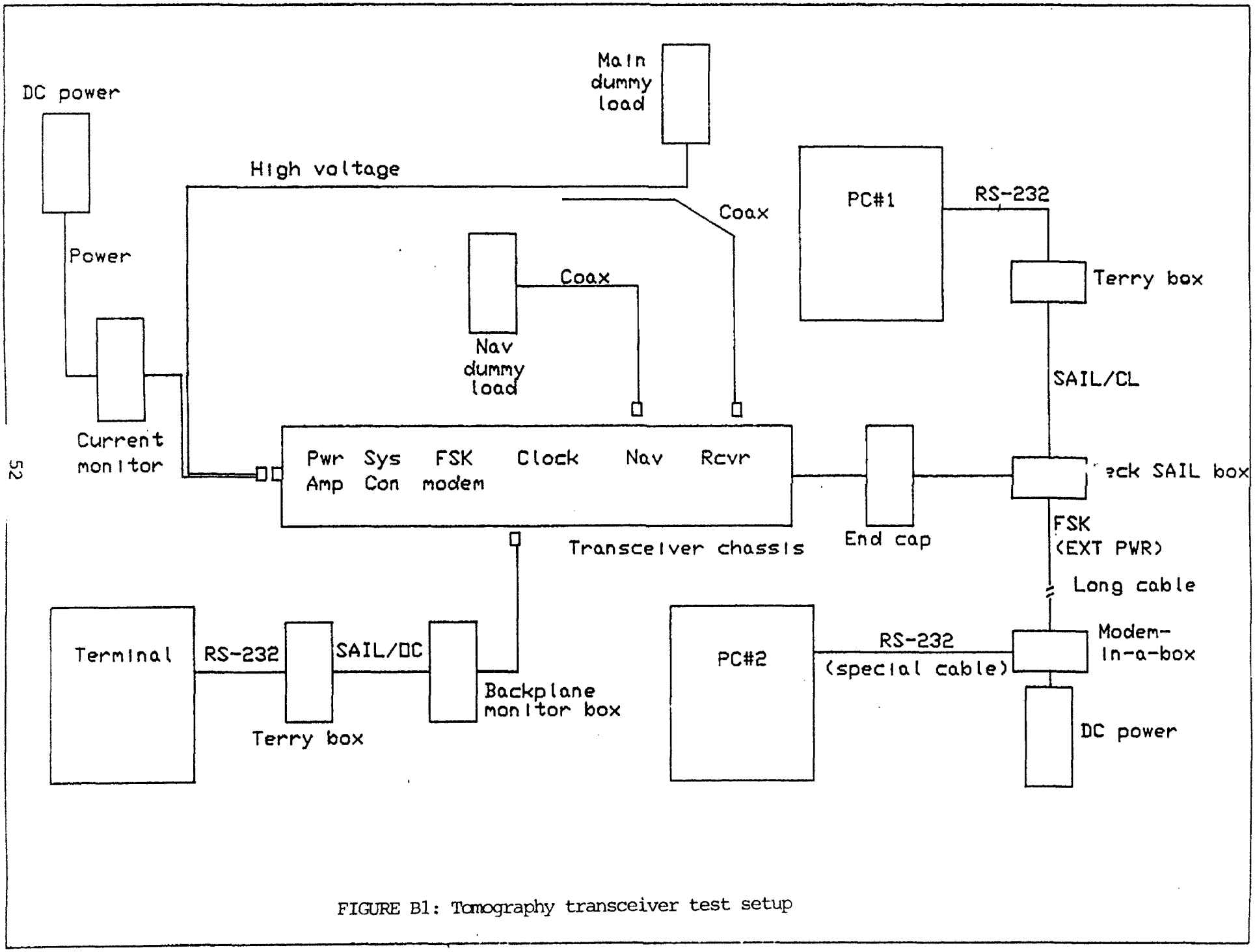


FIGURE B1: Tomography transceiver test setup

```
Adjusting the RAM/PROM boundary
          - examine map for BSS start address
          - change dummy DB in c0t.asm to move to chip boundary
Installing interrupt vectors
          - before burning PROM's, load and execute program
          - exit to monitor (!M)
          - use "W" to upload three .HEX files for burning
```

Appendix C: Tomography Transceiver Documentation

1. Schematics

FSK modem board

Figure C1 is the schematic of the FSK modem. This board is used unmodified. Table C1 is the parts list for the modem.

The FSK signal is generated by dividing a 307.1 KHz crystal controlled clock by either 17 or 32 in U1, and then again by 2 in U2A. This yields symmetric square waves of 9040 and 4800 Hz respectively, that are controlled by a carrier enable line and a Tx data line. When the carrier is not enabled, U2A is held in reset condition, which in turn leaves both drive transistors (Q1 and Q2) in an ''off" state. Switching both transistors off provides for a high impedance input to the E/M cable so as not to load down any inbound signals. Note, this modem is configured such that any transmitted data will be echoed by the receiver.

The modem's FSK receiver is composed of the first order band-pass filter, a zero crossing detector, a CMOS phase locked loop and two threshold detectors. The inbound FSK signal is first passed through a bandpass filter which is centered at 7200 Hz. The purpose of this filter is to attenuate out-of-band signals, specifically the higher order harmonics of the phase locked loop's center frequency. The signal is next passed to a voltage comparator, U4C, configured with 400 mV of hysteresis. This provides the input to the phase locked loop with a jitter free square wave representation of the inbound tone. The phase locked loop (U7) is now used to perform the actual frequency discrimination. It has been configured using its phase comparator #1 and an external, modulus 4 divider. The average output of phase comparator #1 is used to drive a level detector (U4A and U4B). As the phase comparator's duty cycle varies with the input frequency, the average voltage to the threshold detector changes. A feature of this technique of frequency discrimination is that it requires 4 cycles of input signal to ensure reliable operation; hence the baud rate is restricted to 1200 baud to keep the FSK tones at a reasonably low frequency.

SeaScan Tomo I/O board

Figure C2 is the schematic of the Tomo I/O board as modified. The modifications are performed using etch cut and wire jumpers. The functions provided are:

- make use of the forth position of the serial communication circulator formed with U5, U7
- reassign the function of some pins on J4 to carry additional signals between this board and the FSK modem
- carry power to an additional pin on the back plane.

Onset CPU-8088 board

Figure C3 is the schematic of the CPU board as modified. The modification is a wire jumper to take the UART oscillator to the CPU's timer/counter input.

2. Wiring

Modem to I/O board wiring

Figure C4 shows the signals required between the FSK modem and the Tomo I/O board to control the modem and to carry data between the modem and the communications circulator.

Modem to end cap wiring

Figure C5 shows the wiring between the SAIL connector on the end cap and the FSK modem. The FSK signal is carried on a wire formerly assigned as the external power wire. The change required the addition of a wire since the former connection had diodes to control power flow.

Modem to backplane wiring

Figure C6 shows the power wiring used for the modem. It is connected to the power line added to the backplane on the I/O board. The connection is made at the connector assigned to the Mass Store, now not used.

FSK modem to PC cable

Figure C7 shows the special test cable required to connect the modem-in-a-box to PC#2. It uses the RTS line to control the carrier control line of the modem.

External jumper cable

Figure C8 shows the external cable specified to make the connection between the transceiver and the E/M cable to the IBC.

3. Software

This section provides documentation for the operating software required. Section 3.1 covers software which operates in the transceiver itself - the firmware. Section 3.2 covers software which operates on the PC used to support the instrument. Listings are included only for those programs which are new. Listings are not included for those programs which were modified for the S-Tether application. Computer readable sources of all software are archived on floppy disks.

3.1 Transceiver Software

The operating program runs without benefit of system support but does have a rudimentary special purpose I/O system called BIOS in the PROM on the

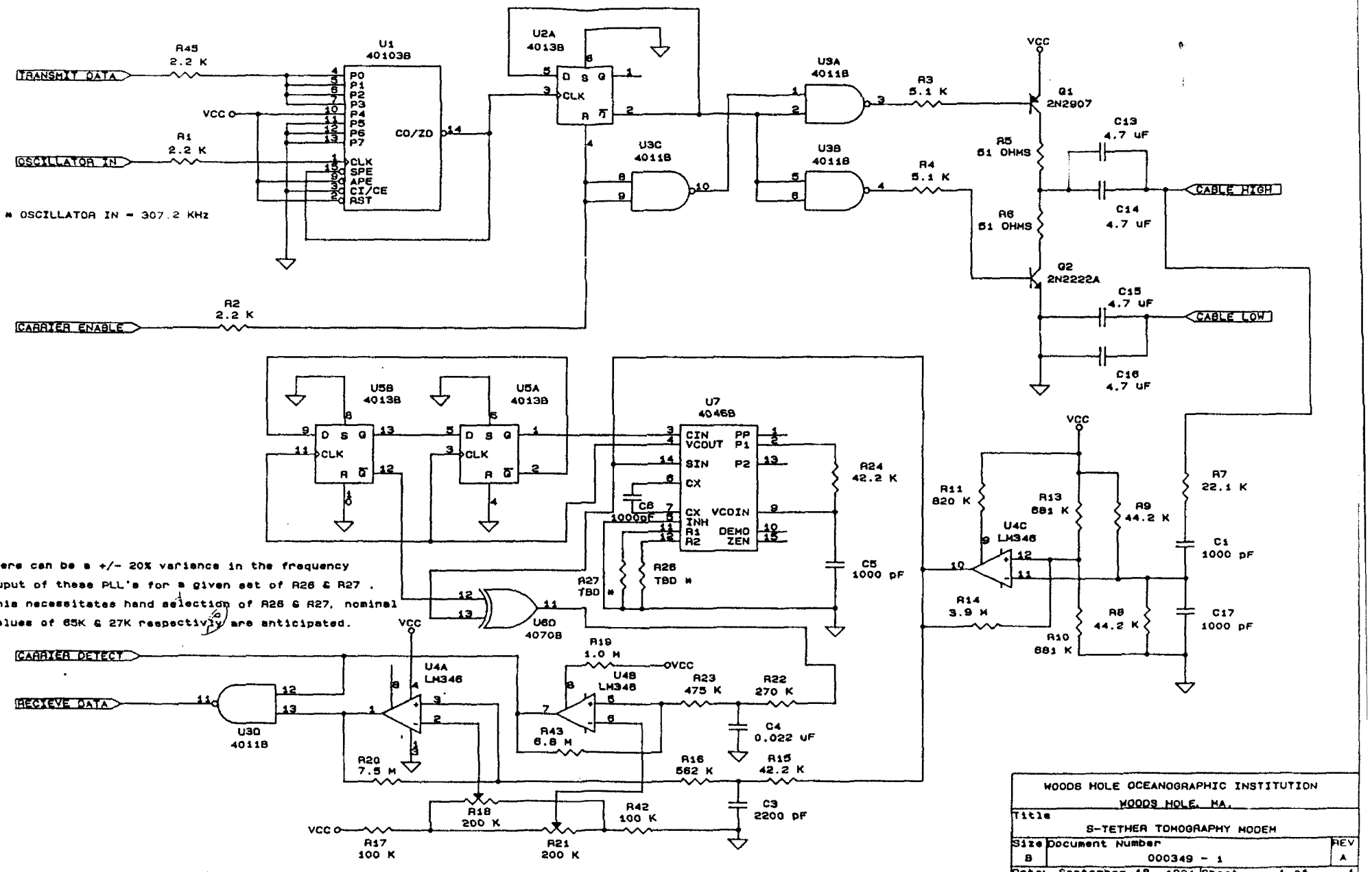


Figure C1: S-Tether Tomography Modem

S-Tether Tomography Modem

Item	Quantity	Reference	Part
1	3	C1, C5, C17	1000 pF
2	1	C3	2200 pF
3	1	C4	0.022 uF
4	1	C6	1000 pF
5	4	C13, C14, C15, C16	4.7 uF
6	1	Q1	2N2907
7	1	Q2	2N2222A
8	3	R1, R2, R45	2.2 K
9	2	R3, R4	5.1 K
10	2	R5, R6	51 ohms
11	1	R7	22.1 K
12	2	R8, R9	44.2 K
13	2	R10, R13	681 K
14	1	R11	820 K
15	1	R14	3.9 M
16	2	R15, R24	42.2 K
17	1	R16	562 K
18	2	R17, R42	100 K
19	2	R18, R21	200 K
20	1	R19	1.0 M
21	1	R20	7.5 M
22	1	R22	270 K
23	1	R23	475 K
24	2	R26, R27	TBD *
25	1	R43	6.8 M
26	1	U1	40103B
27	2	U2, U5	4013B
28	1	U3	4011B
29	1	U4	LM346
30	1	U6	4070B
31	1	U7	4046B

Table C-1

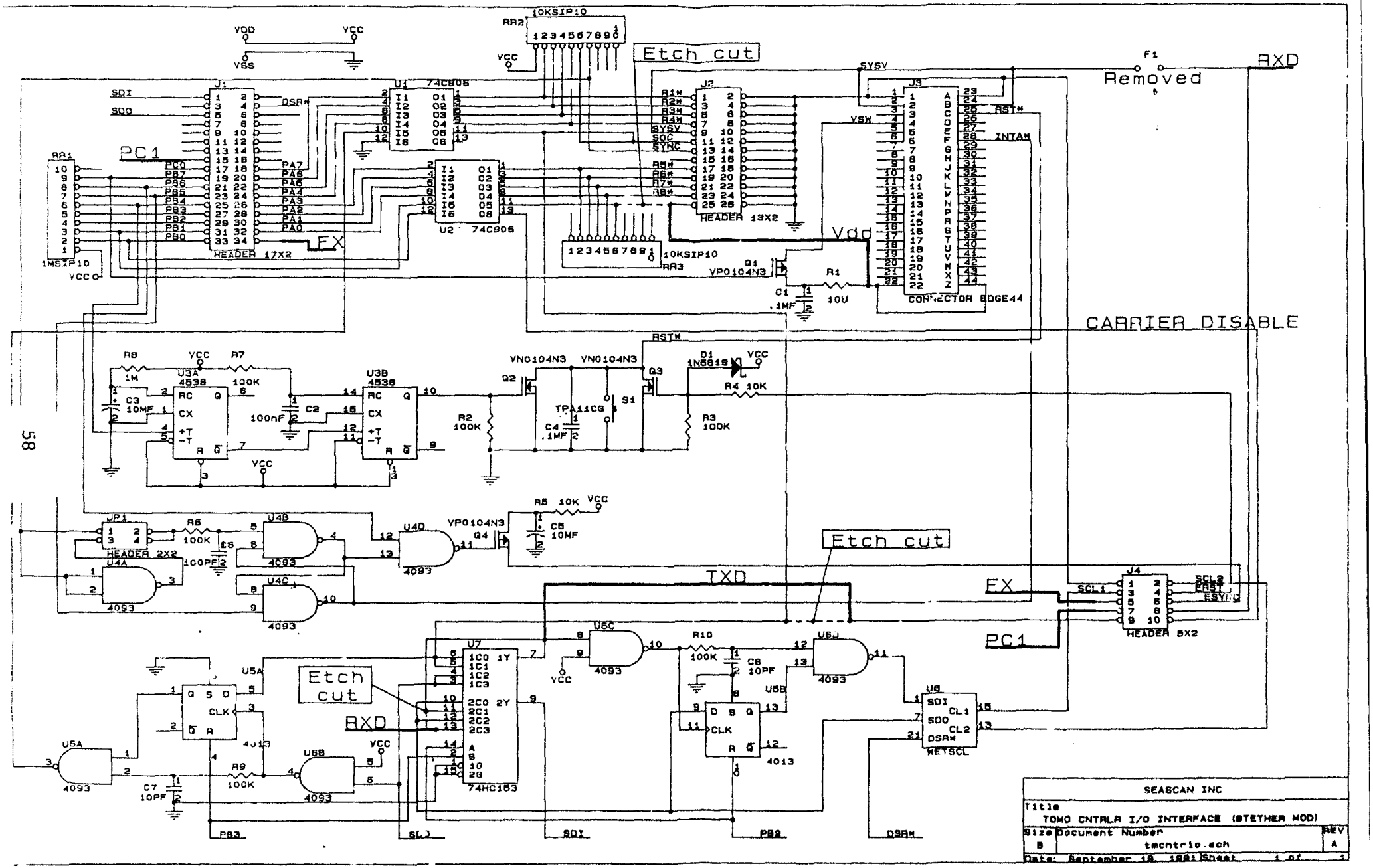


Figure C2: Tomo Controller I/O Interface (S-Tether Modification)

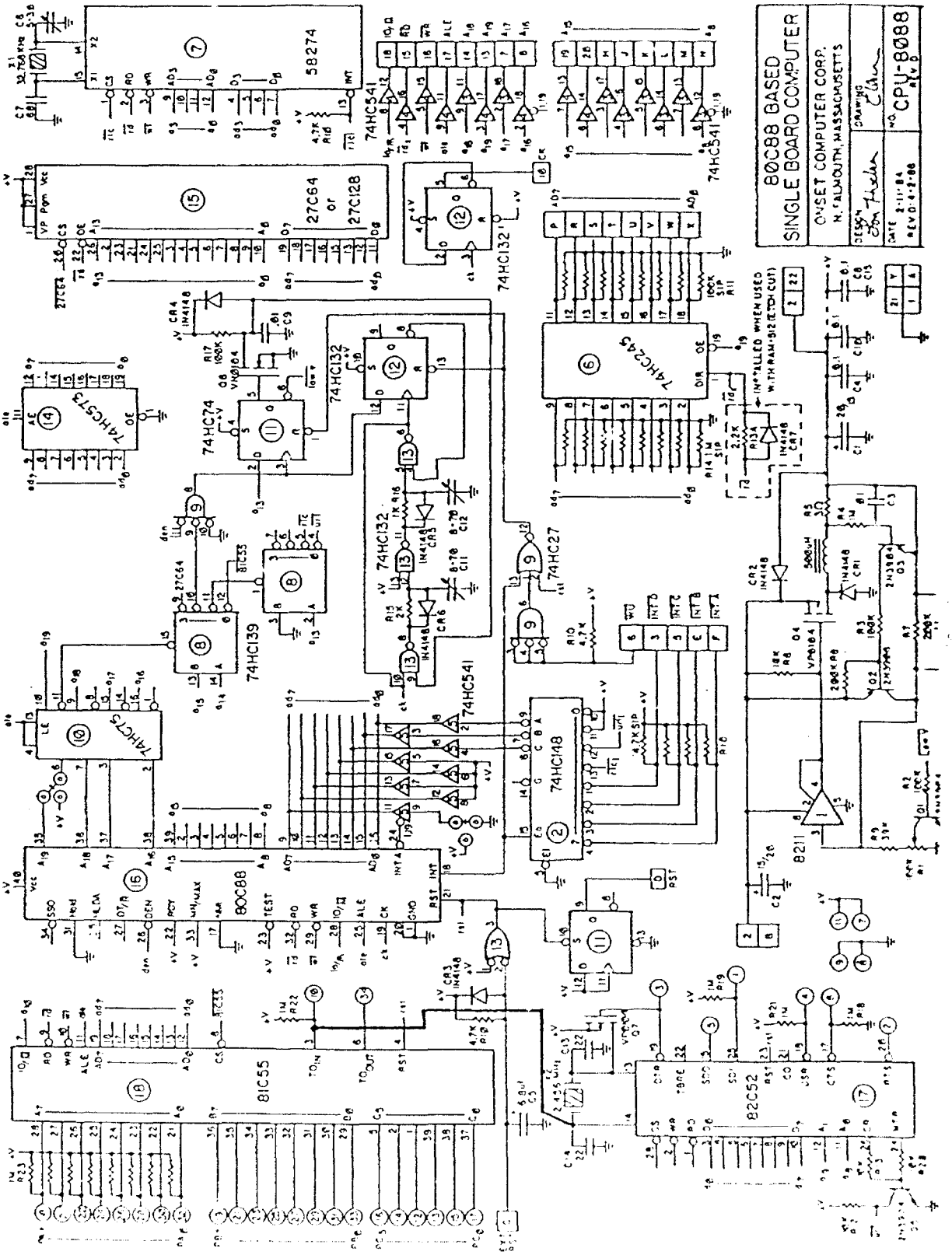


FIGURE C-3: CPU 8088

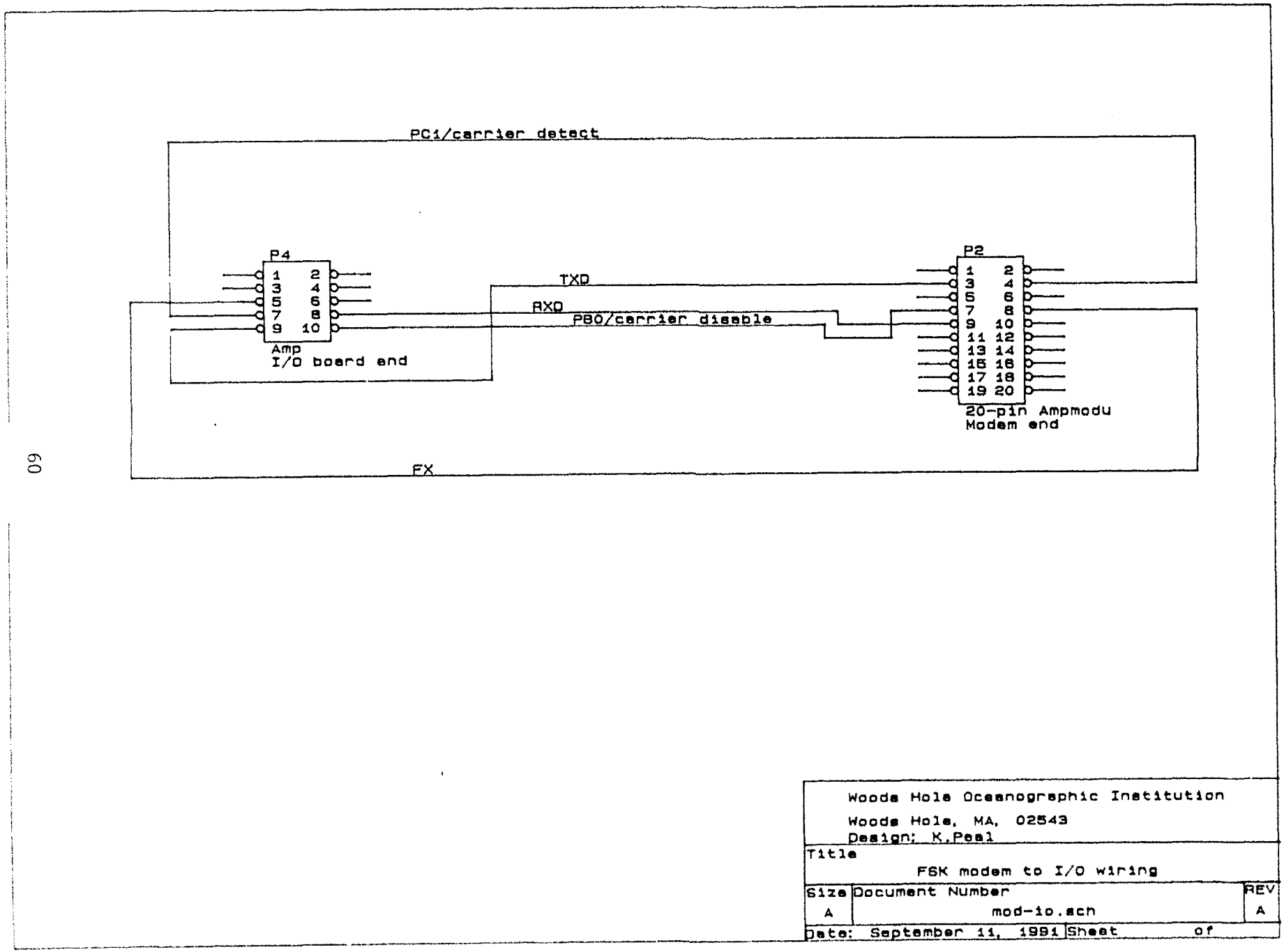
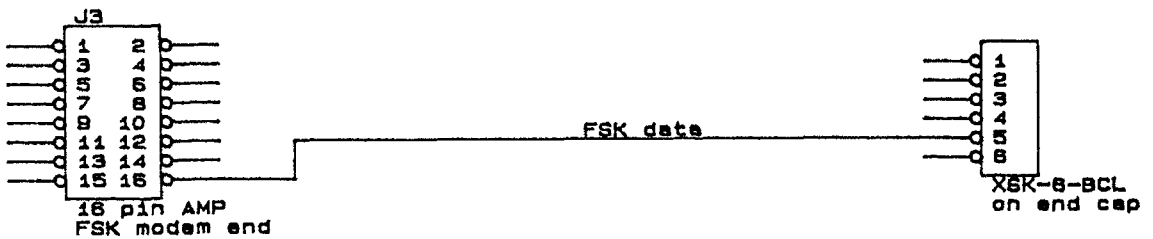


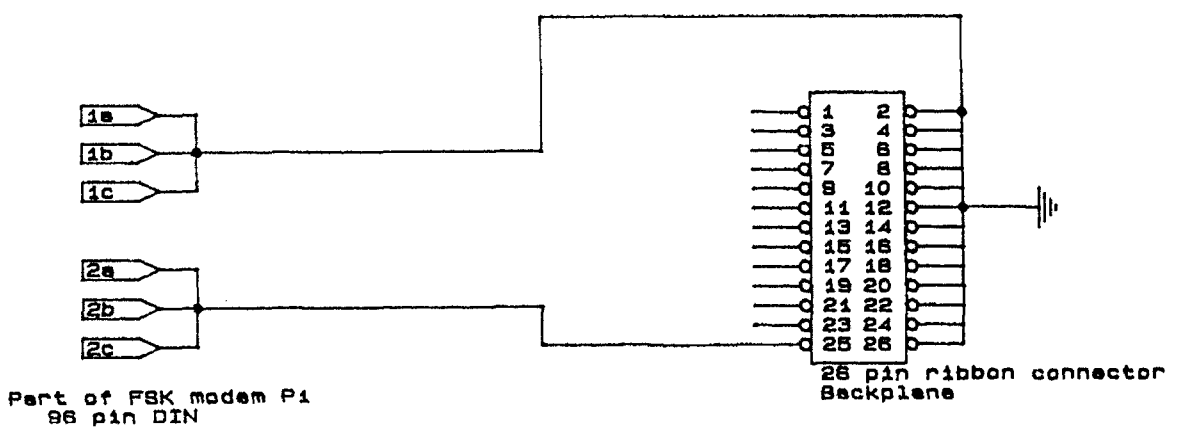
Figure C4: FSK Modem to I/O wiring



Note: wire added to existing harness

Woods Hole Oceanographic Institution Woods Hole, MA, 02543 Design: K.Peal		
Title: FSK modem to end cap wiring		
Size: A	Document Number: mod-end.sch	REV: A
Date: September 11, 1991	Sheet of	

Figure C5: FSK modem to end cap wiring



Woods Hole Oceanographic Institution Woods Hole, MA, 02543 Design: K.Peal		
Title FSK modem to backplane wiring		
Size A	Document Number mod-back.sch	REV A
Date: September 11, 1991	Sheet of	

Figure C6: FSK modem to backplane wiring

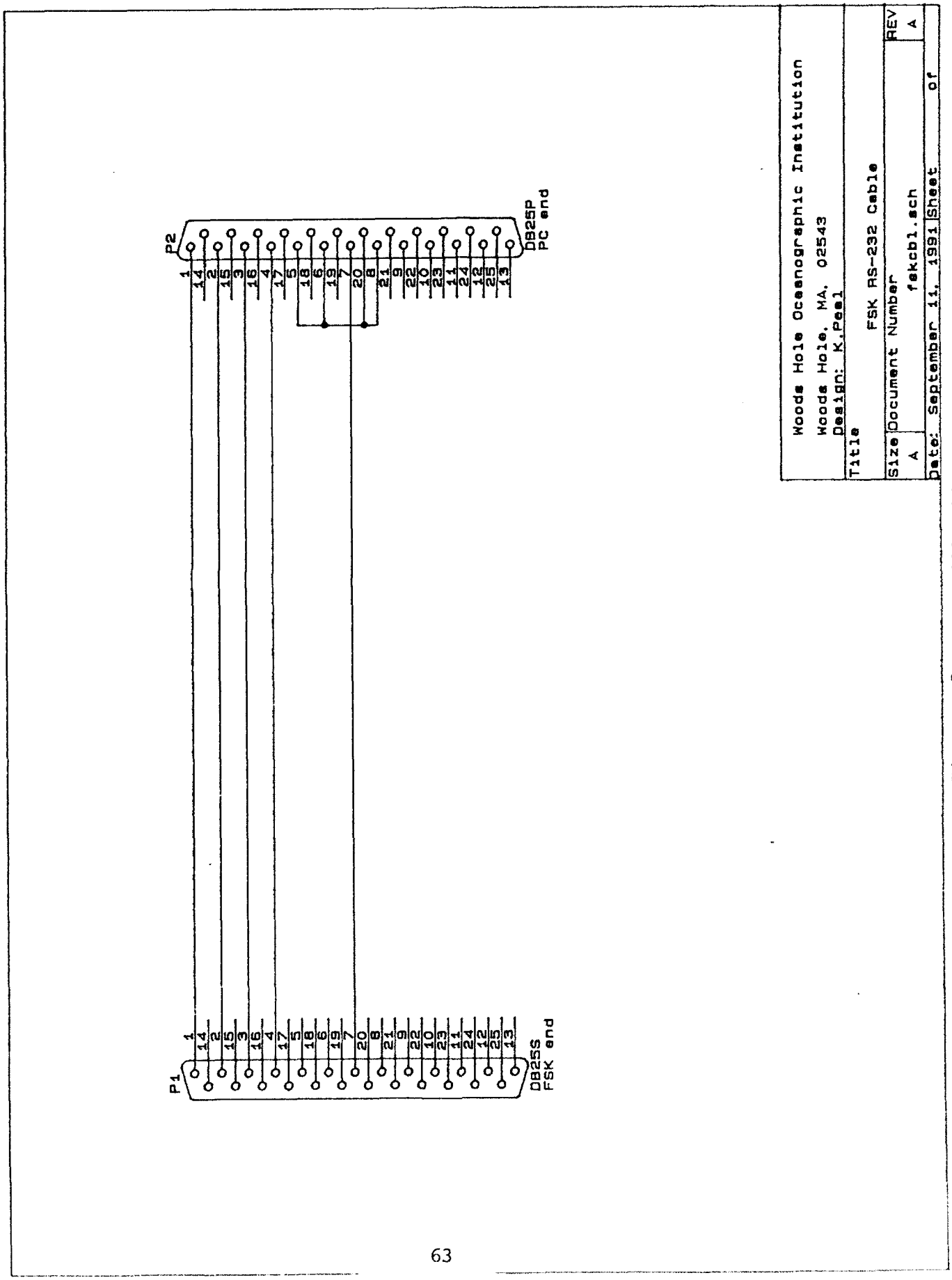


Figure C7: FSK RS-232 Cable

CPU board. This allows the the scheduler program to access system resources through interrupt calls which are serviced by the BIOS.

For the S-Tether application it was necessary to add some functions and capabilities to the BIOS. This necessitated modification of two of the BIOS modules:

- bioscom.asm
- biosport.asm

Scheduler:

The main operating program is written in Turbo C with a simple startup routine in assembly. To achieve the S-Tether functions, several of the program's modules were modified and some new ones were written:

Module	Status	Listing
define.c	modified	-----
setup.c	modified	-----
c0t.asm	modified	-----
monitor.c	modified	-----
schedule.c	modified	-----
system.c	modified	-----
driver2.c	modified	-----
driver5.c	new	C1
driver6.c	new	C2
get_peaks.c	new	C3
s3wh.map	modified	-----

Peak Picker:

To achieve data compression for telemetry, the transceiver performs correlation and peak picking on the fly. Figure C9 shows a flow chart of the process. The program appears as listing C-3.

Task Table: (pkscal.tsk)

The new version of the scheduler program and the edit and check programs require a slightly different format task table. Listing C4 is a test task table used during development of the changes.

et:

The deck support program et (edit task) was modified to deal with the different data structures and the additional virtual devices necessitated by the changes to S-Tether format. A list of the changed modules follows.

et.c
define.c
analyse.c
sendtask.c
listtask.c
files.c

dm:

The deck support program dm (disk monitor) was modified to deal with the different data structures and the additional virtual devices and data records necessitated by the changes to S-Tether format. A list of changed modules follows.

dm.c
define.c
stadata.c

ibc-em:

This is a new program written to run on PC#2 in figure B1. Its function is to emulate the surface computer (the IBC) to allow checkout and testing of the communication protocol and processing functions in the transceiver. It uses one of the PC's COM ports to control a compatible FSK modem mounted in a box near the PC. This allows a long cable to run from this box to the transceiver to test communications functions.

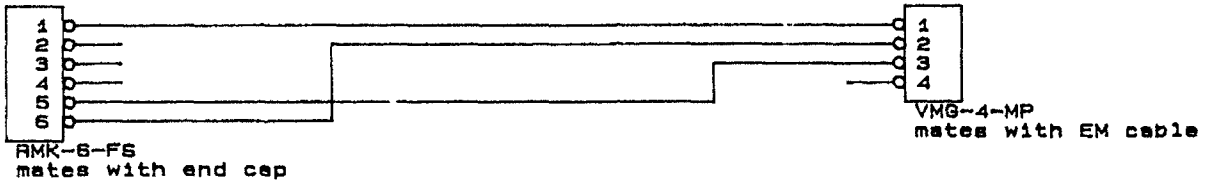
The main features of the program are:

- it follows the SAIL protocol for addressing and communication
- it issues replies that are compatible with the specified protocol
- it deals with the half duplex signal and the requirement to control the FSK carrier when talking
- it can simulate errors and delays in replies from the IBC to the transceiver
- it can display and log all communication between the transceiver and the IBC
- it offers simple operator control to modify parameters during operation thus simulating changing conditions in the IBC.

When the program is started, it operates by displaying all interactions and replying appropriately but by default does not write a log file. To display the command list, hit space followed by <cr>. The operator control items are:

- open or close logging file (IBC-EM.LOG)
- quit program (closes log file if open)
- enter one of the following status characters
  - R - ready
  - N - not ready
  - D - done operation
  - F - fault occurred during operation
- enter a single ASCII hex character coded as follows
  - bit 0 - GPS receiver not ready
  - 1 - SAIL clock timeout
  - 2 - SAIL clock set problem
  - 3 - tomo clock timeout

The program is found in listing C5.



Woods Hole Oceanographic Institution Woods Hole, MA, 02543 Design: K.Peal		
Title External Jumper Cable		
Size A	Document Number jumper.sch	REV A
Date: September 11, 1991	Sheet of	

Figure C8: External Jumper Cable

LISTING C1:
DRIVER 5.C:
Driver for Calculations and Data Transfer

```
/*  DRIVER5.C :  driver for calculations and data transfer */
/*
K.Peal  27 Mar 91
1 July, fixed last memcpy to use only 26 bytes (enough for 3 correlates from
one DATAPT), leaving only two bytes unused on the Status SubRecord.
28 Aug 91, changes for use with PEAKS - GetPeaks() instead of calc_function()
23 Oct 91, correct the saving of the demodulates - first one and save the
second thru fourth shoulder values
*/
#include "define.c"

void driver5 (int Device, int option)
{
        int day,hour,min,sec, mode;
        char *ptr, i,j;

        switch (option)
        {
                case SETUP:
                        break;
                case RUN:
                        mode = (int)Task[ThisTask].Config[Device].Data.fun;
                        GetPeaks (mode);                /* instead of calc_function */
                        ptr = Argos.ascvals;
                        for (i=0;i<PEAKS;i++)
                        {
                                sprintf (ptr,"%04X",values[i].index);
                                ptr+=4;
                                for (j=0;j<5;j++)
                                {
                                        sprintf (ptr,"%08lX%08lX",values[i].correl [j][0],
                                                values[i].correl [j][1]);
                                        ptr+=16;
                                }
                        }
                        ConvertTime (GetLocalTime (),&day,&hour,&min,&sec);
                        sprintf (Argos.Nicetime.dayno,"%04d",day);
                        sprintf (Argos.Nicetime.hr,"%02d",hour);
                        sprintf (Argos.Nicetime.min,"%02d",min);
                        sprintf (Argos.Nicetime.sec,"%02d",sec);
                        ptr = (char *) &Status[INT];
                        for (i=0;i<3;i++)
                        sprintf (Argos.range [i],"%02X%02X",*(ptr+17+2*i),
                          *(ptr+18+2*i));
                        ptr = (char *) &Status[RX1];
                        sprintf (Argos.ibceng.rxgain,"%02X",*(ptr+31));
                        sprintf (Argos.ibceng.ambnoise,"%02X%02X",
                          *(ptr+37),*(ptr+38));
```

```
                        ptr = (char *) &Status[PAR];
                        for (i=0;i<25;i+=4)     /* does temp, pres, .. volt3 */
                                sprintf (Argos.ibceng.inttemp+i,"%02X%02X",
                                  *(ptr+17+i),*(ptr+18+i));
                        Argos.ender = 0x000d;   /* bytes reversed !! */
                        Control_FSK (START);
                        Address (Device);
                        Sail_FSK ("!DB\r",ETX,2);
                        Sail_FSK (Argos.ascvals,ETX,5);
                        Sail_FSK ("!BYE\r",ETX,5);
                        Control_FSK (END);
                        break;
                case FINISH:
                        Status[Device].Device = Device;
                        Status[Device].Time = GetLocalTime ();
                        Status[Device].SailStatus = SailStatus[Device];
                        Status[Device].StartTime = NextStartTime;
                        memcpy ((void *)Status[Device].Data,
                                (void *)&Task[ThisTask].Config[Device],sizeof (CONFIG));
                        memcpy ((void *)(Status[Device].Data+(sizeof (CONFIG))),
                                (void *)(&values[0]),2);
                        memcpy ((void *)(Status[Device].Data+(sizeof (CONFIG))+2),
                                (void *)(&values[0].correl[1][0]),24);
                        Store (&Status[Device]);
        }
}
```

LISTING C2:
DRIVER 6.C:
Driver for Time Check

```
/*  DRIVER6.C :  driver for time check

K.Peal  16 Apr 91

       Status bits:
              1. between IBC and TT
                     0       GPS rx not ready
                     1       SAIL clock timeout
                     2       SAIL clock set problem
                     3       TOMO clock timeout
              2. for local status record
                     0       GPS rx not ready
                     1       SAIL clock timeout
                     2       SAIL clock set problem
                     3       TOMO clock timeout
                     4       IBC carrier stuck on
                     5       IBC sign-on timeout (not there?)
                     6       IBC setup timeout
                     7       IBC check timeout

*/
#include "define.c"
#include <ctype.h>

void driver6 (int Device, int option)
{
       char tstat=0,*ptr, count, safe, ttime [30];
       int found;

       switch (option)
       {
              case SETUP:
                     break;
              case RUN:
                     Control_FSK (START);
                     if (GetConfig () & 0x0200)
                     {
                            tstat |= 0x10;  /* IBC carrier on fault */
                            break;
                     }
                     Address (Device);
                     if (SailStatus [Device] != 0)
                     {
                            tstat |= 0x20;  /* IBC sign-on timeout */
                            break;
                     }
                     count = 0;
                     Wait (1);
```

```
do
{
        if ((found = Sail_FSK ("!TS\r",ETX,3)) != NULL) break;
} while ((*finder (ETX) == '?') && (++count < 5));
if ((count >= 5) || (found != NULL))
{
        tstat |= 0x40;  /* timeout on setup */
        break;
}
count = 0;
Wait (1);
do
{
        if ((found = Sail_FSK ("!SS\r",ETX,3)) != NULL) break;
        ptr = finder ('S');
        if (*ptr == 'S')
        {
                if ((*(ptr+2) == 'R') || (*(ptr+2) == 'F')) break;
                Wait (10);
        }
} while (++count < 30);
if ((count >= 30) || (found != NULL))
{
        tstat |= 0x40;
        break;
}
ptr+=2;
if (*ptr == 'F')
{
        tstat |= *(ptr+1) & 0x0f;
        break;
}
if (*(ptr+1) == '4') tstat |= 0x02;     /* clk adj fault */
count = 0;
Wait (1);
do
{
        if ((found = Sail_FSK ("!TC\r",ETX,3)) != NULL) break;
} while ((*finder (ETX) == '?') && (++count < 5));
if ((count >= 5) || (found != NULL))
{
        tstat |= 0x80;  /* IBC check timeout */
        break;
}
Control_FSK (END);      /* end of FSK, go to clock on OC */

WakeUp (CLOCK);
Wait (2);
```

```
Address (CLOCK);
if (SailStatus [CLOCK] != 0) tstat |= 0x08, /* TOMO ck tm  */
SetConfig (GetConfig() | 0x40); /* set PC6, SYNC enable *,
count = 0;
do
{
        if (Sail ("?T\r",ETX,10) == NULL) tstat |= 0x08;
        safe = 0;
        ptr = buffer;
        while (!isdigit((int)*ptr) && safe++ <= 20) ptr+=
        if (sscanf (ptr,"%d%d%d%d",Status[Device].Data+2,
                Status[Device].Data+4,
                Status[Device].Data+5,
                Status[Device].Data+6) == 4) break;
        Wait (2);
} while (++count < 3);
strncpy (ttime,buffer,20);
SetConfig (GetConfig () & 0x1f);

Control_FSK (START);
count = 0;
do
{       /* send time to IBC */
        strncpy (buffer,ttime,20);
        if ((found = Sail_FSK (buffer,ETX,5)) != NULL) br
} while ((*finder (ETX) == '?') && (++count < 3));
if ((count >= 3) || (found != NULL))
{
        tstat |= 0x80;
        break;
}

count = 0;
do
{
        if ((found = Sail_FSK ("!CS\r",ETX,3)) != NULL) br
        ptr = finder ('S');
        if (*ptr == 'S')
        {
                if ((*(ptr+1) == 'D') || (*(ptr+1) == 'F')  break;
                Wait (10);
        }
} while (++count < 30);

if ((count >= 30) || (found != NULL))
{
        tstat |= 0x80;
        break;
```

```
                        }
                        ptr += 1;
                        if (*ptr == 'F')
                        {
                                tstat |= *(ptr+1) & 0x0f;       /* get fault f */
                                break;
                        }
                        Status[Device].Data[7] = *ptr;  /* status x */
                        ptr++;
                        Status[Device].Data[8] = *ptr;  /* status f */
                        break;
                case FINISH:
                        if (Sail_FSK ("!BYE\r",ETX,3) != NULL) tstat |= 0x80;
                        Control_FSK (END);
                        Status[Device].Device = Device;
                        Status[Device].Time = GetLocalTime ();
                        Status[Device].SailStatus = SailStatus[Device];
                        Status[Device].StartTime = NextStartTime;
                        Status[Device].Data[0] = 0;     /* dummy config values */
                        Status[Device].Data[1] = 0;
                        Status[Device].Data[9] = tstat;
                        Store (&Status[Device]);
        }
}
```

LISTING C.3:
GET PEAKS.C

```
                filename peaks

These routines are used to find peaks in a transceiver
        demodulate array.

All declarations (except those local to these routines) are
        done in the file define.c.


NOTE: Change sequence laws to reflect transmit code.



Modifications to interface with S-Tether Controller
        - remove testing calls (including stdio.h) here
        - change Values to values here
        - change indx to indices here
        - use NDEMOD instead of PTS/2 in controller program
        - add intensT[] to controller program
        - remove dummy arguments in GetPeaks() here
        - change Sort2() argument to unsigned
        K. Peal  Aug 91

*****************************************************************/
/*  #include    <stdio.h>                   Only needed for testing */


#include        "define.c"


#define         SeqLaw    01617  /* Octal */
#define         SeqInit   0400   /* Octal */
#define         SeqDeg    9

void    Correlate();
void    PseudoRandomCode ();
void    GetIntensT();
void    PickPeaks();
void    Sort2();

/* For testing purposes ONLY
void    Write2File();
void    Write2File2();
*/


/*************************************************************
                        GetPeaks

Actual routine to be called. It then will do a correlation
        on the demodulates and pick a specified number of
        peaks that will be passed back to the calling
        routine.

Argument:
        opmode      - used for a flag (not implemented now).
```

```
void    GetPeaks(opmode)

        int     mode;

{
        Correlate();
        GetIntensT();
        PickPeaks();
}

/*************************************************************
                        PseudoRandomCode

Generate the pseudo random code used for the correlation.

Initialization for generating this sequence requires an
        initialization sequence, a sequence law, and
        a sequence degree.

This routine is called as

        PseudoRandomCode(SeqDeg,SeqInit,SeqLaw,code)

        SeqDeg  - sequence degree
        SeqInit - initialization sequence
        SeqLaw  - sequence law
        code    - returned pseudo random sequence


*************************************************************/


void    PseudoRandomCode(degree,ShiftRegister,law,code)

        int degree,ShiftRegister,law, * code;
{
        int NewBit,i,j,imax;
        int tmp;

        imax = (1 << degree) - 2;
        degree--;

        for (i=0;i<=imax;i++)
             {
             *(code+i) = ShiftRegister & 1;
             NewBit = tmp = ShiftRegister & law;
             for (j=1;j<=degree;j++) NewBit += (tmp >> j);
             ShiftRegister = (ShiftRegister >> 1) | ((NewBit & 1) << degree);
             }
}

/*************************************************************
                        Correlate

Correlate the demdulates with a previously generated
        pseudo random sequence.

*************************************************************/

void    Correlate()

{
        int code[DIGITS],i,ii,j;
```

```
        PseudoRandomCode(SeqDeg,SeqInit,SeqLaw,code);
        /*   Write2File("code.dat",code,DIGITS);   */

        for (i=0;i<PTS;i++) space.Correls[i] = 0;


        for (i=0;i<PTS; i+=4)
             {
             for (j=0,ii=i;j<DIGITS;j++,ii+=4)
                 {
                 if(ii >= PTS) ii -= PTS;
                 if (code[j] == 1)
                    {
                     space.Correls[i] += Demods[ii];
                     space.Correls[i+1] += Demods[ii+1];
                     space.Correls[i+2] += Demods[ii+2];
                     space.Correls[i+3] += Demods[ii+3];
                     }
                  }
               }

}


/*************************************************************
                        GetIntensT

Instead of calculating the intensity specifically using
        the sum of the squares, this routine just adds
        the complex parts together to get a "pseudo"
        intensity. This is done because floating point
        arithmetic and storage is not available. Since
        the intensities are calculated ONLY for identifing
        the peaks, this approach saves computation time
        and storage and does not affect the peak picking.


*************************************************************/

void    GetIntensT()
{
        int     i,j;


        for (i=0,j=0;i<NDEMOD;i++,j=i*2) {

                if(space.Correls[j] < 0)
                   if(space.Correls[j+1] < 0)
                        intensT[i] =  -space.Correls[j] - space.Correls[j+1];
                   else
                        intensT[i] =  space.Correls[j+1] - space.Correls[j];
                else
                   if(space.Correls[j+1] < 0)
                      intensT[i] =  space.Correls[j] - space.Correls[j+1];
                   else
                      intensT[i] =  space.Correls[j] + space.Correls[j+1];

                }


/*      Write2File2("peaks.out",intensT,NDEMOD);                        */


}
```

```
/*************************************************************
                        PickPeaks

Find a peak and save its position (add 1 since Demod array
        starts at 0). Sort in decreasing order all the peaks
        that were found (sort MUST start at array index = 1).

Save a specified number of the largest peaks along with
        2 neighbors on each side of each peak.

*************************************************************/

void    PickPeaks()
{
        int     i,j,k,ndx,iml,ict;

        ict = 0;
        for (i=1;i<NDEMOD-1;i++)
            {
             if((intensT[i-1] < intensT[i]) && (intensT[i] >= intensT[i+1]))
                {
                  ict++;
                  indices[ict] = i+1;
                }
            }

        for(i=1;i<=ict;i++) intensT[i] = intensT[indices[i]-1];

        Sort2(ict,intensT,indices);

        for(i=1;i<=PEAKS;i++)
            {
             iml = i-1;
             ict = 0;
             values[iml].index = indices[i];
             for(j=0;j<5;j++)
                {
                for(k=0;k<2;k++)
                        {
                        ndx = (indices[i]-1)*2 - 4 + ict;
                        values[iml].correl[j][k] = space.Correls[ndx];
                        ict++;
                        }

                }

              }

}

/*************************************************************
                        Sort

Heap sorts values in array ra(1,n) in decreasing order. Each
element of rb is switched in accordance with ra.

*************************************************************/
void Sort2(n,ra,rb)


int n;
long ra[];
unsigned rb[];
{
        int l,j,ir,i;
        long rra;
```

```
        unsigned rrb;

        l=(n >> 1)+1;
        ir=n;
        for (;;) {
                if (l > 1) {
                        rra=ra[--l];
                        rrb=rb[l];
                } else {
                        rra=ra[ir];
                        rrb=rb[ir];
                        ra[ir]=ra[1];
                        rb[ir]=rb[1];
                        if (--ir == 1) {
                                ra[1]=rra;
                                rb[1]=rrb;
                                return;
                        }
                }
                i=l;
                j=l << 1;
                while (j <= ir) {
                        if (j < ir && ra[j] > ra[j+1]) ++j;
                        if (rra > ra[j]) {
                                ra[i]=ra[j];
                                rb[i]=rb[j];
                                j += (i=j);
                        }
                        else j=ir+1;
                }
                ra[i]=rra;
                rb[i]=rrb;
        }
}

/*************************************************************
                        Write2File

This function is used for testing purposes only.

*************************************************************/
/*
void    Write2File(string,array, n)

        int n, * array;
        char * string;

{

        FILE    * fp;
        int     i;

        if ((fp = fopen(string,"w")) == NULL)
            {
             printf("\n ERROR!  Cannot open file for WRITE...\n");
             exit (0);
             }

        for (i=0;i<n;i++)
           {
           if ((i%16 == 0) && (i!=0)) fprintf(fp,"\n");
           fprintf(fp,"%d ",array[i]);
           }
        fclose(fp);
}
```

```
 */

/*****************************************************************
                        Write2File2

This function is used for testing purposes only.

*****************************************************************/
/*
void    Write2File2(string,array, n)

        int n;
        long  * array;
        char * string;

{

        FILE    * fp;
        int     i;

        if ((fp = fopen(string,"w")) == NULL)
            {
             printf("\n ERROR!  Cannot openfile for WRITE...\n");
             exit (0);
             }

        for (i=0;i<n;i++)
           {
           fprintf(fp,"%d %d \n",i,array[i]);
           }
        fclose(fp);

}
 */

/*****************************************************************/
```

Flowchart of the peak picking process

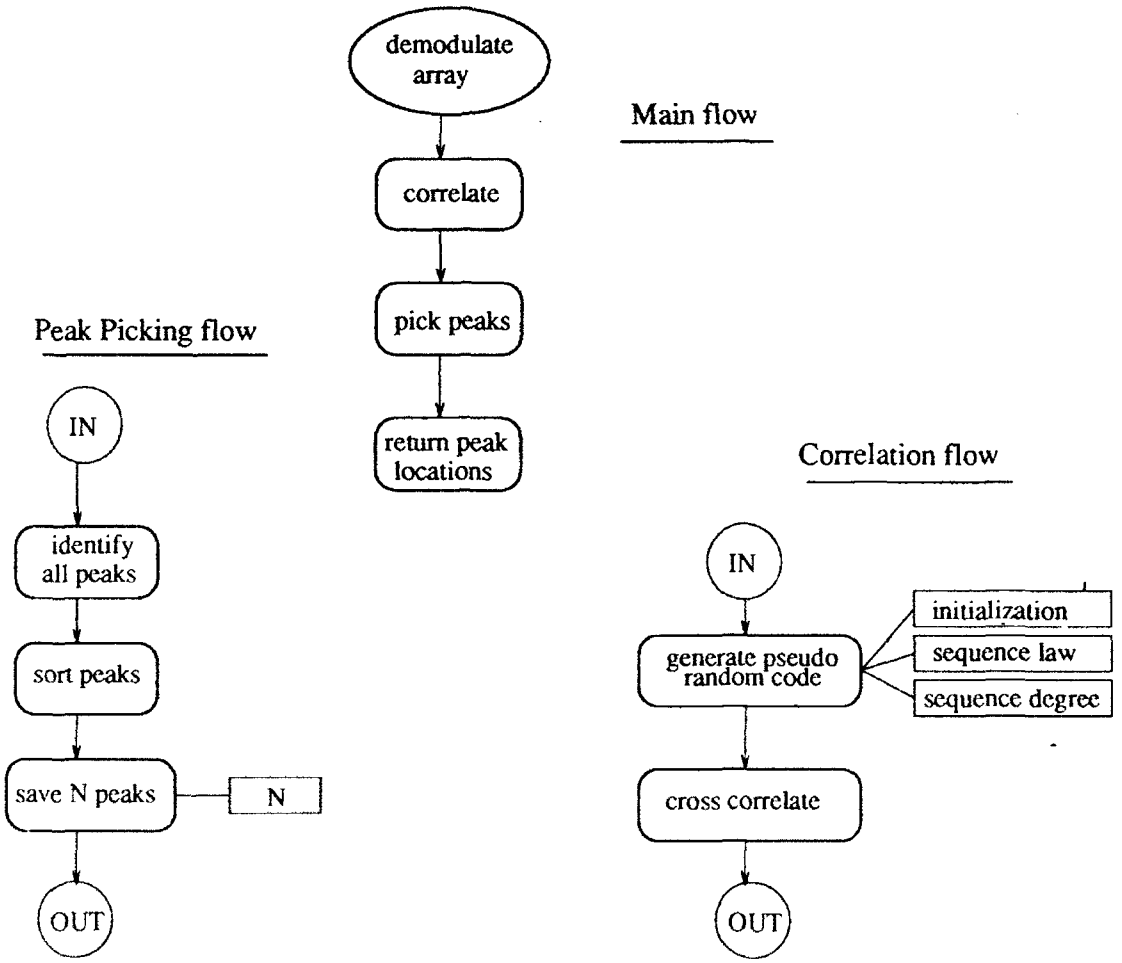


FIGURE C9: Flowchart of the peak picking process

LISTING C.4:
PKSCAL.TSK

```
;  Task  : 01

  0000 00 00 00    ;  start time
  0365 23 59 59    ;  stop time

; Tx   :  Seq  Vref  Pmin
          10    2   200
          01 00 15     0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23
; Rx1  :  Seq  Delay
          08    09
          01 00 15     0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23
; Rx2  :  Seq  Delay
          10    73
          00 00 10     0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23
; Rx3  :  Seq  Delay
          01    74
          00 00 24     0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23
; Rx4  :  Seq  Delay
          01    115
          00 00 24     0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23
; Rx5  :  Seq  Delay
          01    135
          00 00 24     0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23
; Rx6  :  Seq  Delay
          01    163
          00 00 24     0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23
; Rx7  :  Seq  Delay
          01    199
          00 00 24     0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23
; Rx8  :  Seq  Delay
          01    207
          00 00 24     0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23
; Int  :  Seq
          04
          01 10 15     0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23
; Par  :
          00 00 30     0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23
; tim  :
          00 00 30     0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23
; cal  : Rx_no   mode
          2       0
       01 05 15    0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23
```

LISTING C.5:
IBC-EM

```
/* program to emulate IBC portion of STETHER
       Cindy Howitt 4/4/91  version: 5/29/91
       K.Peal  24 June 91, 22 Oct 91
       Changed to IBC-EMD logs data buffer to file ibc-em.dat
*/

#include <stdio.h>
#include <bios.h>
#include <dos.h>
#include <string.h>
#define ETX 3
#define CR 0x0d
#define MAX_CMD_LEN 4
#define MAX_BUF_LEN 4000
#define settings (0x80|0x02|0x18|0x00)
#define RTS_MASK 0x02
#define CTS_MASK 0x10
#define DTR_MASK 0x01
#define UNSET_RTS 0xfd
#define port 1 /* 0 for COM1; 1 for COM2 */

#if port == 0
#define XFE 0x3fe  /* 2 for com2; 3 for com1 */
#define XFC 0x3fc
#endif

#if port == 1
#define XFE 0x2fe  /* 2 for com2; 3 for com1 */
#define XFC 0x2fc
#endif

FILE *num, *ddat;
enum { dc_nothing, check_id, wait_for_bang, wait_for_echo, command_mode,
              fill_buffer, error };
int mode, i, cmdn, buflen, upper, time_delay=50, dbset;
int in_byte;

char me[]="SU01", buffer[MAX_BUF_LEN], cha, X[]="Nbc", F[]="123";
char tsu[]="TS", tck[]="TC", tst1[]="SS", tst2[] = "CS";
char db[]="DB", replystr[20];
char err_str[]=" ?", err_mes[]="Error Message";
char prompt[] = {'\r','\n',':',' ',ETX,'\0'};
char nul[] = " -", bye[] = "BYE";

void prep_to_send(void);
void send_reply(char *string);
void handle_keyboard(void);
char mygetc(void);
```

```
int get_cmd(char *buffer);
void check_for_me(void);
void get_data(void);
void fill_cmd_buf(void);
void check_for_bang(void);
void process_error(void);
void pretty (void);
void main(void);

/* routine to set DTR and send RTS ---------------- */
void prep_to_send(void)
{
/*      while (!(inportb (XFE) & CTS_MASK)) ;*/
        while (1)
        {
        outportb(XFC,RTS_MASK | DTR_MASK);  /* set DTR and turn on carrier */
        delay (time_delay);
        return;
        }
}
/* routine for sending reply out the port and echo to screen  ---------- */
void send_reply(char *string)
{
        delay  (100);
        prep_to_send();
        while (*string != '\0')
        {
                fputchar(*string);
                fputc (*string,num);
                bioscom(1,*string++,port);
        }
        string = prompt;
        while (*string != '\0')
        {
                fputchar (*string);
                fputc (*string,num);
                bioscom (1,*string++,port);
        }
        delay (20);
        outportb(XFC,DTR_MASK);  /* turn off carrier */
}
/* routine to handle keyboard entry ------------------ */
void handle_keyboard(void)
{
        ungetch ();
        cha = getchar ();
                if ( cha == 'F'|| cha == 'f' )
                {
```

```
        F[0] = toupper ((char)getchar());
        getchar ();
        printf ("\n F now set to %c\n",F[0]);
}
else if (cha == 'X' || cha == 'x')
{
        X[0] = toupper ((char)getchar());
        getchar ();
        printf ("\n X now set to %c\n",X[0]);
}
else if (cha == 'T' || cha == 't')
{
        getchar ();
        puts("\n Enter new value for time delay(ms): ");
        scanf("%d",&time_delay);
        printf ("\n Time delay now set to %d\n",time_delay);
        getchar ();
}
else if (cha == 'O' || cha == 'o')
{
        getchar ();
        if (num == NULL)
        {
                num = fopen("ibc-em.log","a+");
                if(num==NULL)
                {
                        puts(" ERROR opening log file");
                        exit(0);
                }
                puts ("Log file opened\n");


                ddat = fopen("ibc-em.dat","a+");
                if(ddat==NULL)
                {
                        puts(" ERROR opening dat file");
                        exit(0);
                }
                puts ("Dat file opened\n");
         }
         else puts ("Log files already open\n");
}
else if (cha == 'C' || cha == 'c')
{
        getchar ();
        fclose (num);
        fclose (ddat);
        num = NULL;
```

```
                puts ("Log files closed \n");
        }
        else if (cha == 'Q' || cha == 'q')
        {
                fclose (num);
                fclose (ddat);
                exit(0);
        }
        else
        {
                getchar ();
                puts("\n Valid entries are: \n\
                Xc<CR> - Change value of X to 'c' (R,N,D,F)\n\
                Fc<CR> - Change value of F to 'c' (0,1,2, - F)\n\
                T<CR>  - To enter new value for time delay \n\
                O<CR>  - To open log file (IBC-EM.LOG)\n\
                C<CR>  - To close log file\n\
                Q<CR>  - Terminate program \n");
        }
}
/* routine to get a character from the port
   while checking for keyboard entries ------------------ */
char mygetc(void)
{
        int dum;
        while (1)
        {
                dum = 0;                        /* to clear the warning */
                outportb(XFC,DTR_MASK);
                while( (upper = bioscom(2,dum,port)) & 0x8000 )
                {
                        if(kbhit()) handle_keyboard ();
                }
                return(upper & 0xff);
        }
}
/* routine to interpret command ----------- */
int get_cmd(char *buffer)
{
        buffer[i] = 0;
        if(!strcmp(buffer,tsu))
        {
                send_reply(tsu);
                mode = wait_for_echo;
                dbset = 0;
                i = 0;
                return(1);
        }
```

```
else if(!strcmp(buffer,tck))
{
        send_reply(tck);
        mode = wait_for_echo;
        dbset = 0;
        i = 0;
        return(2);
}
else if (!strcmp (buffer,bye))
{
        send_reply (bye);
        mode = wait_for_echo;
        dbset = 0;
        i = 0;
        return(5);
}
else if(!strcmp(buffer,tst1))
{
        replystr[0] = '\0';
        strcat(replystr,tst1);
        strncat(replystr,"R\0",1);
        strncat(replystr,F,1);
        send_reply(replystr);
        mode = wait_for_echo;
        dbset = 0;
        i = 0;
        return(3);
}
else if(!strcmp(buffer,tst2))
{
        replystr[0] = '\0';
        strcat(replystr,tst2);
        strncat(replystr,"D\0",1);
        strncat(replystr,F,1);
        send_reply(replystr);
        mode = wait_for_echo;
        dbset = 0;
        i = 0;
        return(3);
}
else if(!strcmp(buffer,db))
{
        send_reply(db);
        dbset = 1;
        mode = wait_for_echo;
        i = 0;
        return(4);
}
```

```
        else
        {
                send_reply(err_str);
                mode = wait_for_echo;
                dbset = 0;
                i = 0;
                return(0);
        }
}
/* -------------------- */
void check_for_me(void)
{
        if(cha != me[i++])
        {
                mode = do_nothing;
                i = 0;
                return;
        } /* if */
        if(me[i] == '\0')
        {
                send_reply(me);
                mode = wait_for_echo;
                dbset = 0;
                i = 0;
        } /* if */
}
/* -------------------- */
void check_for_bang(void)
{
        if(cha == '!')
        {
                mode = command_mode;
                i = 0;
                return;
        }
        if (cha == CR)
        {
                send_reply (nul);
                mode = wait_for_echo;
                i = 0;
                dbset = 0;
                return;
        }
        fputchar (cha);
        fputc (cha,num);
        return;
}
/* ------------------------ */
```

```
void fill_cmd_buf(void)
{
        if(cha != CR)
        {
                buffer[i++] = cha;
                if(i > MAX_CMD_LEN)
                {
                        mode = error;
                        printf("i= %d\n",i);
                        i = 0;
                        return;
                } /* if i>*/
                return;
        } /* if cha */
 /* got here must be CR */
        cmdn = get_cmd(buffer);
        return;
}
/* ------------------------- */
void get_data(void)
{
        if(i > MAX_BUF_LEN-1)
        {
                mode = error;
        return;
        }
        if(cha == CR)
        {
                fputchar ('\n');
                buffer [i++] = cha;
                buffer [i++] = '\0';
                pretty ();      /* write buffer to file and make it pretty */
                send_reply (nul);
                mode = wait_for_echo;
                dbset = 0;
                i = 0;
        }
        else
        {
                buffer[i++] = cha;
                fputchar (cha);
        }
}
/* -------------------- */
void process_error(void)
{
        if(cha == CR)
        {
```

```
                send_reply(err_mes);
                mode = wait_for_echo;
                dbset = 0;
                i = 0;
        }
}
/* ---------------------------- */
void main(void)
{
        int status;

        printf ("\nIBC Emulator program (with data file), using COM port %d\n",port+1);
/* initialize */
        bioscom(0,settings,port);
        num = NULL;
        buffer[0] = 0;
        mode = do_nothing;
        while ( 1==1 )
        {
                cha = mygetc();
                if( cha == 'q')
                {
                        exit(0);
                }
                if( cha == '#' )
                {
                        mode = check_id;
                        i = 0;
                        continue;
                }/* if */
                switch ( mode)
                {
                case check_id:
                        check_for_me();
                        break;
                case wait_for_bang:
                        check_for_bang();
                        break;
                case command_mode:
                        fill_cmd_buf();
                        break;
                case fill_buffer:
                        get_data();
                        break;
                case wait_for_echo:
                        if (i++ >= 0)
                        {
                                if (dbset == 0) mode = wait_for_bang;
```

```
                                else mode = fill_buffer;
                                dbset = 0;
                                i = 0;
                        }
                        break;
                case error:
                        process_error();
                } /* switch */
        } /* while */
}

void pretty (void)
{
        char *point;
        int loop1,loop2,loop3,loop4;

        point = buffer;
        for (loop1=0;loop1<32;loop1++)          /* 32 datapoints */
        {
                for (loop2=0;loop2<4;loop2++)           /* index */
                        fputc ((int)*point++,ddat);
                fputc ('\n',ddat);
                for (loop2=0;loop2<5;loop2++)           /* 5 correlates */
                {
                        for (loop3=0;loop3<2;loop3++)   /* real and imag */
                        {
                                fputc (' ',ddat);
                                for (loop4=0;loop4<8;loop4++) fputc ((int)*point++,ddat);
                        }
                        fputc ('\n',ddat);
                }
                fputc ('\n',ddat);
        }
        for (loop1=0;loop1<4;loop1++)                   /* time  - day no., 4 digits */
                fputc ((int)*point++,ddat);
        fputc (' ',ddat);
        for (loop1=0;loop1<3;loop1++)                   /* time - hour, min, sec */
        {
                for (loop2=0;loop2<2;loop2++)           /* 2 digits each */
                        fputc ((int)*point++,ddat);
                if (loop1 != 2) fputc (':',ddat);
        }
        fputc ('\n',ddat);
        fputc ('\n',ddat);

        for (loop1=0;loop1<3;loop1++)                   /* navigation - 3 travel times */
        {
                for (loop2=0;loop2<4;loop2++)           /* travel time */
```

```
                fputc ((int)*point++,ddat);
        if (loop1 != 2) fputc (',',ddat);
}
fputc ('\n',ddat);
fputc ('\n',ddat);

for (loop1=0;loop1<2;loop1++)                   /* rx gain */
        fputc ((int)*point++,ddat);
fputc (',',ddat);
for (loop1=0;loop1<8;loop1++)                   /* 8 eng parameters */
{
        for (loop2=0;loop2<4;loop2++)           /* 4 characters each */
                fputc ((int)*point++,ddat);
        if (loop1 != 7) fputc (',',ddat);
}
fputc ('\n',ddat);
fputc ('\n',ddat);
```

S-Tether Surface Electronics Data Logger

Item	Quantity	Reference	Part
1	2	C1, C2	470 uF
2	1	C3	0.1 uF
3	3	D1, D2, D3	1N4148
4	1	JP1	Header, 2 Pin
5	1	Q1	IRFD 9123
6	1	Q2	2N7000
7	2	R1, R5	69.8 K
8	2	R2, R6	309 K
9	11	R3, R7, R9, R10, R13, R15 R16, R17, R18, R19, R23	100 K
10	2	R4, R8	140 K
11	4	R11, R12, R20, R22	20.0 K
12	1	R14	464 OHMS
13	1	R21	10 K
14	1	R24	1.00 M
15	3	R25, R26, R27	2.00 K
16	2	U1, U2	OP490
17	1	U3	ICL7660

Table C2

Appendix D: Complete Description of Code Generator

1. Hardware Description

The signal generator module consists of three main components; two of these are PC cards, and the third is the power amplifier. The PC cards conform to the Onset outline but are not C-44 bus compatible. They are mounted in an Onset style three card bucket which, along with the power amplifier, is mounted to a bent aluminum frame. This frame also acts as a heat sink for the four switching transistors and is secured to the main electronics chassis by four 8-32 screws.

PC card number one is the waveform generator's CPU/system interface card. It performs the dual tasks of generating the low level tomographic signal and interfacing the waveform generator with the system bus. The card employs Hitachi's CMOS 6301 with the piggyback EPROM socket and allows signal generation with simultaneous 9600 baud SAIL communication. All outputs from this card are 5 volt CMOS level.

PC card number two accepts the low level signals generated by the CPU/system interface card and converts them to the appropriate levels for driving the gates of the power amplifier's switching transistors. This card also fuses the power amplifier's 50 volt line and acts as the amplifier's power switch.

The power amplifier generates the high voltage signal necessary to drive the two outboard ceramic transducers.

2. Software Description

The Program

The main program is written in 6301 assembly language and implements the standard SAIL protocol along with a pseudo interpreter. These two functions are simultaneous. The interpreter executes a simple set of pseudo-instructions specifically tailored for signal generation. The default baud rate is 9600 but the system will respond to the set baud command and will accept all 2x values between 150 and 19200 baud. Even parity, seven data bits and one stop bit is the expected format.

Operation and SAIL Commands

To address the signal generator, type #SSnn where nn is the module's serial number. Following the address with a space will cause the prompt string to be output. This string is composed of a line feed, carriage return, the : character and the non-printing character ETX. Omitting the space and following the address directly with a command is allowed. A terminating carriage return is not required when entering most commands in this fashion. The exceptions are ?M, !M, !L, !C and !N. A list of other commands is shown in Listing D1.

There are three general purpose monitor commands:

!H	Prints a help file.
?Maaaa_llllcr	Prints the contents of memory beginning at aaa for llll locations.
!Maaaa_ddddcr	Load memory beginning at aaaa with dddd.

There are four commands which deal with the signal generator's output power level.

!I	Increment the power level.
!D	Decrement the power level.
!Lnn	Set the power level to nn which must be between 0 and 62 base 16.
?L	Output the current power level.

The remaining six commands allow direct control of the signal generator primarily by forcing the location of the pseudo program counter.

!Paaaa	Execute pseudo code beginning at address aaaa.
?P	Outputs the current value of the pseudo program counter.
!Cn	Execute pseudo program n. n is any value between 0 and F.
!X	Reinitialize pseudo program counter and execute program at !C0.
!Nnn	Set the variable loop counter to nn.
?S	Output the status character.

3. Signal Specifications

The signal generated by the S-Tether tomography signal generator is produced by clocking a software simulation of the shift register pictured in Figure D1. The preset condition of register #8 is a 1, the remaining stages are preset to 0.

The sequence thus produced is listed in Table D1, qualified and stored as encoded phase transitions each consisting of two bits and represented by the HEX data depicted in Table D2. The second bit of a pair indicates the actual phase to transmit while the first bit flags the requirement for a phase change relative to the previous bit transmitted. The pseudo instructions are interpreted to read this table and produce a series of half periods from which the entire code sequence is constructed. Figure D2 is a flowchart detailing this process and listing D2 is a copy of the program.

The procedure used to calculate the length of the half periods is as follows:

Given...

Center frequency (f) = 400 Hz
Digit length (dl) = 4 cycles
Sequence length (sl) = 511 digits (29 - 1)
Desired modulation angle (θ= 87.46704 degrees ($\arctan(sl^{-.5})$)
Bandwidth (b) = 100 Hz (f/dl)
Sequence law = 1021 (Octal)
Sequence initialization = 400 (Octal)

Calculate the time change required to accomplish the specified phase shift.

t = θ/360 X 1/f = 87.46704/360 X 1/400 = 607.410 uS
Rounding to the nearest uS = 607 uS
Calculate the actual phase shift.
θ = t X f X 360 = 607 uS X 400 X 360 = 87.408 degrees

To calculate carrier transitions proceed as follows:

1) The phase difference between an arbitrary reference and a "1" is +theta.

2) The phase difference between the same reference and a "0" is - theta.

3) The phase shift between a "1" and a "0" is 2 X theta.

4) The phase shift between a "0" and a "1" is accomplished by delaying the first transition of the "1" by 1/f + 2Tp.
Where Tp is the time difference between a quarter cycle of carrier and θ...(90 - θ)/360 X 1/f
Tp = (90 - 87.408/360 X 1/400 = 18.0 uS

5) The phase shift between a "1" and a "0" is accomplished by delaying the first transition of the "0" by 1/f - 2Tp.

6) Since Tp = 18 uS, 2Tp = 36 uS which makes the three possible time intervals between carrier transitions as follows:

```
1/f + Tp = 1/400 + 36 uS = 2,536 uS   (LONG)
1/f - Tp = 1/400 - 36 uS = 2,464 uS   (SHORT)
.5/f = .5/400 = 1,250 uS              (NOM)
```

4. The Constant Power Algorithm

As part of the normal transmit cycle, the controller measures the battery voltage while the signal generator is transmitting. This data is then used during the next transmit cycle to set the signal generator's output power level. The signal generator's output power is proportional to battery voltage times duty cycle. To achieve constant output power this product must remain constant.

In the TASK table there is a parameter designated Vref. This number is defined as the voltage at which the generator's duty cycle should be 100%. The duty cycle will be inversely proportional to the battery voltage, but at battery voltages lower than Vref the duty cycle will remain fixed at 100%.

The desired duty cycle is obtained via the !Lxx command where xx is HEX value between 2 and 62 (2 to 98 decimal).

The algorithm (in decimal) is as follows:

```
Verify that Vref > 0 but < 50
IF Vref < 0 THEN Vref = 1
IF Vref > 50 THEN Vref = 50.

IF (Vbat < Vref)
Duty cycle = 98

IF (Vbat > 50
Duty cycle = (98 X Vref)/50

ELSE
Duty cycle = (98 X Vref)/Vbat
```

The power amplifiers are tuned for maximum output power at a given battery voltage (Vf) with a 100% duty cycle. For operation at various output levels, Vref may be calculated as follows:

Desired power level	Vref (in task table)
1/1	Vf
1/2	Vf / 1.414
1/4	Vf / 2
1/8	Vf / 2.828
1/10	Vf / 3.162
etc.	

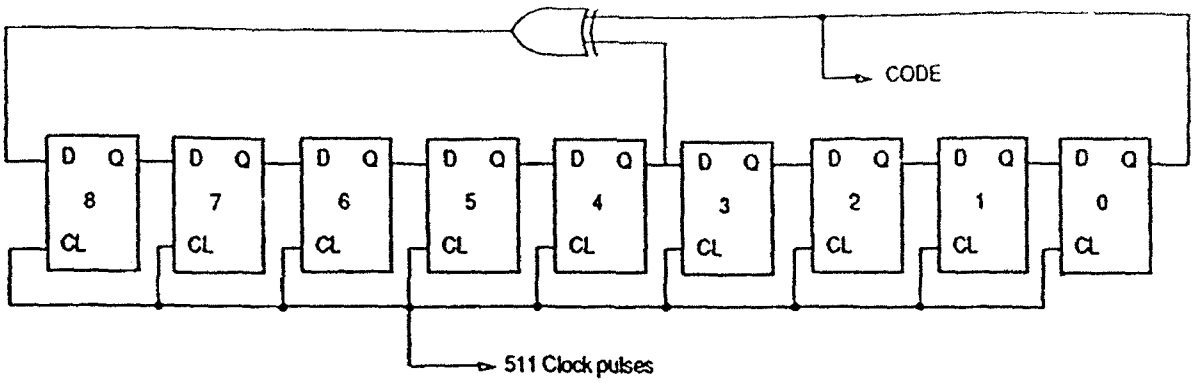


Figure D1 Pseudo-random bit sequence generator.

0000	0000	1000	0100	0110	0001	0011	1001	0101	0110
0001	1011	1101	0011	0111	0010	0010	1000	0101	0110
1001	1111	1011	0010	0100	1011	0111	1110	0100	1101
0100	1100	1100	0000	0110	0011	0010	1000	1101	0010
1111	1110	1000	1011	0001	1101	0110	0101	1001	1110
0011	1110	1110	1000	0011	0101	1011	0111	0110	0000
1011	0101	1111	0101	0101	0000	0010	1001	0101	1110
0101	1101	1100	0000	1110	0111	0100	1001	1110	1011
1010	1000	1001	0000	1100	1110	0001	0111	1011	0110
0110	1000	0111	0111	1000	0111	1111	1100	0001	1110
1111	1000	1011	1001	1001	0000	0100	1010	0111	0110
1000	1111	0011	1110	0110	1100	0101	0100	1000	1110
0011	0110	1010	1110	0010	0110	0010	001		

This is the last digit transmitted.

TABLE D1: Code sequence

0003	80E0	D80E	358E	EED8	0DB5
6E36	D638	3B80	EEDB	8D55	B638
E3B6	D558	E36E	E363	6000	D836
3B83	6E3B	555B	83B6	0D6E	D8ED
8D58	355B	5B80	36ED	B6D6	D803
B6ED	56EE	EE00	3B8E	ED58	ED6D
6003	58D6	E38D	5BB5	BB83	8E03
6358	0ED5	B6D8	DB80	D6D5	80D5
5560	0D5B	5583	B58D	8E00	E3B8
D6DB	8356	3558	DB60	EEE3	8358
36DB	BB58	38D8	3830		

TABLE D2: Hex encoded table of phase transitions

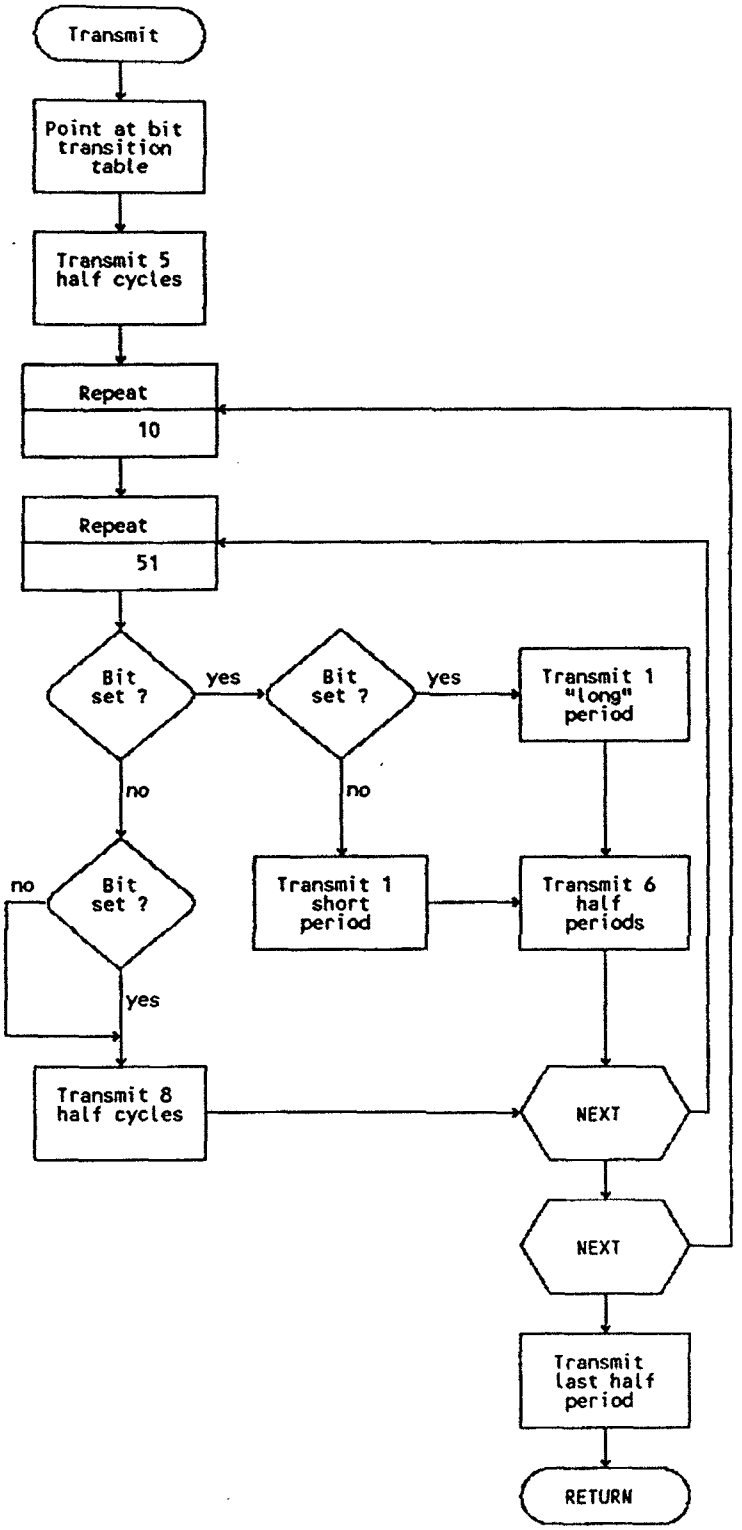


FIGURE D2: A subroutine to synthesize the 511 bit tomographic code

LISTING D2

```
        EJECT
        TITLE       1986 TOMOGRAPHY TRANSCEIVER SIGNAL
        SUBTITLE    PSEUDO INSTRUCTIONS FOR A 511 BIT CODE
        EJECT
;
;*************************************************************
;       FILE    C:\PROGRAMS\ASSEMBLY\6301\TOMGEN\CODE511.PSI
;       BY      STEVE LIBERATORE    REV. 03 3 DEC. 86
;*************************************************************
;
;
;        THIS IS A LINK TABEL TO 16 ENTRY POINTS.
;               >>  DATA TABEL ONLY <<
;        DO NOT ENTER WITH !P COMMAND !!!!!!!!!!
;
        ORG     $E000           ;ASSUMES 64K EPROM
;       ORG     $F000           ;ASSUMES 32K EPROM
;
PLINKS: DW      PSTART0         ;IMMEDIATE SHUT DOWN
        DW      PSTART1         ;UN-MODULATED CARRIER
        DW      PSTART2         ;CODE AFTER EXT. SYNC
        DW      PSTART3         ;CODE AFTER @
        DW      PSTART0         ;ROOM FOR MORE  (4)
        DW      PSTART0         ;               (5)
        DW      PSTART0         ;               (6)
        DW      PSTART0         ;               (7)
        DW      PSTART0         ;               (8)
        DW      PSTART0         ;               (9)
        DW      PSTART0         ;               (A)
        DW      PSTART0         ;               (B)
        DW      PSTART0         ;               (C)
        DW      PSTART0         ;               (D)
        DW      PSTART0         ;               (E)
        DW      PSTART0         ;               (F)
;
POWER_ON:       EQU     $01
POWER_OFF:      EQU     $00
PWR_OFF_FLG_ON  EQU     $04
QUARTER_PWR:    EQU     $18
HALF_PWR:       EQU     $30
THQUTR_PWR:     EQU     $48
FULL_PWR:       EQU     $62
SHORT:  EQU     $09A0           ; 1/F - 36 uS   (2464 uS)
NOM:    EQU     $04E2           ; .5/F          (1250 uS)
LONG:   EQU     $09E8           ; 1/F + 36 uS   (2536 uS)
;
;  TURN OFF XMIT
;
```

```
PSTART0:INITSTK                     ;INITIALIZE THE STACK
        CONSET  POWER_OFF           ;TURN OFF POWER
        STATSET 0                   ;INDICATE '0' FOR IDLE
PLOOP0: WAIT    NOM,$C8             ;USE FREQUENCY ABOVE
        JUMP    PLOOP0              ;LOOP FOREVER
;
;  CW TEST SIGNAL
;
PSTART1:INITSTK                     ;INITIALIZE THE STACK
        CONSET  POWER_ON            ;TURN ON POWER
        PWRSET  QUARTER_PWR         ;SET TO QUARTER POWER
        STATSET C                   ;INDICATE 'C' FOR CARRIER ONLY
PLOOP1: SING    NOM,$C8             ;USE FREQUENCY ABOVE
        JUMP    PLOOP1              ;LOOP FOREVER
;
;
;  ------------------------------------------------------
;  | 1983 SOURCE CODE--511 BIT CODE--400 HZ--           |
;  | 4 CYCLES PER DIGIT--LOPHOLD REPITITIONS--BIT TABLE |
;  | CONTAINS ENCODED PHASE TRANSITIONS TO MODULATE     |
;  | CARRIER +/- 87.408 DEGREES.                        |
;  ------------------------------------------------------
;
PSTART2:INITSTK                     ;INITIALIZE STACK POINTER
        STATSET A                   ;SET STATUS TO 'A' FOR ARMED
        CONSET  POWER_ON            ;TURN ON POWER AMP (?)
        PWRSET  QUARTER_PWR         ;SET POWER LEVEL TO 1/4 FULL
        NOPE                        ;WHY DO WE NEED THIS ?
        SYNCEXT                     ;WAIT FOR EXTERNAL SYNC
CODEOUT:LOOPSTART                   ;INIT VARIABLE LOOP COUNTER
        STATSET C                   ;WRITE 'C' TO SHOW CODING
;
CODECYC:GOSUB   XMIT                ;SEND A CYCLE OF CODE
        DEC_JNZ CODECYC             ;REPEAT LOPHOLD TIMES
;
        CONSET  PWR_OFF_FLG_ON      ;TURN OFF POWER AMP AND SET FLAG
        STATSET D                   ;SET STATUS TO 'D' FOR DONE
;
ENDLOOP:WAIT    $1000,$02           ;LOW POWER WAIT
        JUMP    ENDLOOP
;
;  SUBROUTINE TO GENERATE ONE 5.11 SEC SECTION OF THE CODE
;
XMIT:   SING    SHORT,$01           ;OUTPUT FIRST BIT (A ZERO)
        BITPSET BTABLE2             ;SET BIT TABLE POINTER
        SING    NOM,$05             ;AND SEND 5 FULL HALF CYCLES
;
READBIT:REPEAT  $0A                 ;REPEAT 10 x 51
        REPEAT  $33                 ;FOR 510 BITS
;
        JMPBSET RTEST               ;TEST FIRST BIT OF A BIT PAIR
        JMPBSET RDIGIT              ;IF ITS A ZERO,MOVE PAST NEXT
RDIGIT: SING    NOM,$08             ;BIT AND OUTPUT 8 HALF CYCLES
RBIT:   NEXT
        NEXT
```

```
;
        SING    NOM,$01          ;OUTPUT LAST 1/2 CYCLE
        RETURN                   ;DONE
;
RTEST:  JMPBSET OUTLNG           ;TEST SECOND BIT OF A BIT PAIR
        SING    SHORT,$01        ;IF A ZERO SHIFT PHASE NEGATIVE
        JUMP    NOMX6
;
OUTLNG: SING    LONG,01          ;SECOND BIT OF A BIT PAIR WAS A 1
NOMX6:  SING    NOM,$06          ;SO PHASE SHIFT WILL BE POSITIVE
        JUMP    RBIT
;
BTABLE2:DW        $0003,$80E0,$D80E,$358E,$EED8,$0DB5
        DW        $6E36,$D638,$3B80,$EEDB,$8D55,$B638
        DW        $E3B6,$D558,$E36E,$E363,$6000,$D836
        DW        $3B83,$6E3B,$555B,$83B6,$0D6E,$D8ED
        DW        $8D58,$355B,$5B80,$36ED,$B6D6,$D803
        DW        $B6ED,$56EE,$EE00,$3B8E,$ED58,$ED6D
        DW        $6003,$58D6,$E38D,$5BB5,$BB83,$8E03
        DW        $6358,$0ED5,$B6D8,$DB80,$D6D5,$80D5
        DW        $5560,$0D5B,$5583,$B58D,$8E00,$E3B8
        DW        $D6DB,$8356,$3558,$DB60,$EEE3,$8358
        DW        $36DB,$BB58,$38D8,$3830
;
; THIS IS AN ALTERNATE ENTRY POINT THAT USES THE SAIL @ TO START
; THE CODE.
;
PSTART3:INITSTK                  ;INITIALIZE STACK POINTER
        STATSET E                ;SHOW 'E' FOR ENABLED
        CONSET  POWER_ON         ;TURN ON POWER AMP
        PWRSET  QUARTER_PWR      ;SET LOW POWER
        SYNCAT                   ;AND WAIT FOR SAIL @ SYNC...
        JUMP    CODEOUT          ;AND CARRY ON...
;
```

Appendix E: Test Procedures for Real-Time Tomographic Mooring

A pre-deployment test plan has been developed for the Real Time Tomographic Mooring to ensure that all critical components and subsystems are operating properly prior to launch. This plan, which is divided into subsystem tests and system tests is shown in outline form below:

SUBSYSTEM	TEST
I. Mooring	
A. E/M Cable	-Measure length
	-Check continuity after termination at expected tension
	-Check pinout of terminated cable
	-Pressure test to 1.5 times deployment pressure
B. Urethane Chain (3)	-Check continuity and pinout after termination
	-Pressure test (to 500m)
C. Mooring Design	-Computer analysis at operational and survival currents
	-Check solutions with $\pm$50m depth error (deployment error)
D. Subsurface Buoy	-Measure weight and displacement
E. Mooring Wire	-Measure accurately at tension
F. E/M Tether	-Check continuity underwater and underload
	-Check pinouts
G. Pigtails (2)	-Pressure test and check continuity of pinouts
H. Release	-Test in air
II. Surface Buoy	
A. Buoy	-Measure weight and displacement
	-Measure natural period of roll and stability
	-Perform float test

SUBSYSTEM	TEST
B. Instrument Well	-Pressure test instrument well to 500 psi
C. Antennas	-Pressure test GPS and Argos antenna assemblies to 500 psi
D. Tension Cell	-Pressure test tension cell to 500 psi
	-Calibrate tension cell
E. Light	-Test operation of light and solar charging circuit
III. Surface Buoy Electronics	
A. APIRB (Secondary Argos PTT for location and engineering data)	-Calibrate A/D channels 1. Tension 2. Battery voltage 3. Current drain
	-Test APIRB thru satellite
B. Power Supplies	-Document battery power supplies - voltage, no. of cells, diode protection
	-Test rechargeable battery system in sunlight - voltage and current output
	-Test solar panels into dummy load
C. IBC	-Document power drain quiescent/active
	-Test program operation 1. GPS/SAIL clock interaction 2. PTT uploading and data formats 3. FSK-Tomo interface-use simulator
	-Test IBC for fault tolerance 1. Unplug GPS, SAIL clock, PTT, Tomo unit 2. Simulate errors from GPS, SAIL clock, PTT, Tomo-unit
D. SAIL Clock	-Document power drain-quiescent/active
	-Confirm accuracy of slewed time

SUBSYSTEM	TEST
E. GPS Receiver	-Document power drain-quiescent/active
	-Document accuracy of time tick
	-Document time delays in the IBC-GPS-SAIL clock system
	-Document the time to acquire a time tick
F. Data PTT	-Document power drain-quiescent/active
	-Confirm reliability of test data and checksum thru the satellite
G. TattleTale VI	-Calibrate tilt and tension sensors
	-Document power drain-quiescent/active
	-Confirm data storage operation
	-Test program operation
	-Collect sample data sequence and process data
IV. Tomography Transceiver and Related Electronics	
A. Transceiver	-Document power drain quiescent/active
	-Test acoustic receiver sensitivity
	-Test acoustic transmitter output
	-Pressure test end cap to 1500 psi
B. Clock	-Calibrate clock (determine aging constant)
C. Data Processing	-Exercise self feedback mode of operation
	-Read data via serial link and via Direct 6
	-Check test data for peaks and valid status
	-Check Deck Support Software module

SUBSYSTEM	TEST
D. Communications	-Test FSK-IBC comms -use time task and data task
	-Confirm data integrity - check test pattern buffer through Service Argos
E. Navigation	-Verify transponder frequencies
	-Document output level and pulse width
	-Document receiver sensitivity
	-Check transponder operation
V. Complete System Tests	
A. Bench Tests	-Run Tomography Transceiver with simulated data while interfaced to IBC system including GPS, Argos, SAIL clock. Confirm data integrity. Test fault tolerance by unplugging subsystems and sending erroneous responses.
	-Run Transceiver/IBC system for 2 weeks on the bench
B. Dock Tests	-Install electronics in surface buoy and wire buoy to tomography transceiver using all E/M cables, tether, and pigtails. Test operation in air for 48 hours
	-Deploy complete system in a wire basket in a few meters of water off the dock. Test operation for 48 hours.
C. Pre-Cruise Tests	-Assemble and test all spares

After the hardware and software modifications have been performed, configure the transceiver as in Figure B1. This gives access to necessary parts of the system allowing it to be operated independently of the rest of the system but verifying that its interactions with the IBC are correct. In the figure, PC #1 runs a terminal emulator program and the deck support programs ET, DM and PK. PC #2 runs the IBC emulator program IBC-EM.

Load a task file which enables transmitter, receiver, navigator, parameters, clock check and calculation - see listing C4 in Appendix C. Run the system logging the output received by the IBC emulator computer.

Transcribe and process the disk data and compare the peaks with those in the logged emulator file. Confirm that they are at the correct location and of the correct magnitude. Run the test again with a difference delay in the task file and confirm that the peaks found by both methods move as expected.

Check that all other Argos data in the logged file are formatted correctly and compare with values recorded in the disk status records of the receiver and the parameter (A/D) device.

Equipment and procedures which are needed to perform the pre-deployment tests for the Real-Time Tomography Mooring are listed below:

1. Laboratory equipment

Bench power supply to replace main battery
- sufficient capacity to run transmitter and disk during tests
- clock should never lose power.

Timing reference for clock checks
- ideally SAIL clock run by local flywheel such as Rubidium oscillator,
- storage oscilloscope and radio received time standard acceptable.

SAIL interface and power.

Computer such as IBM-PC or equivalent
- serial port connected to SAIL interface
- with support software.

Dummy load for transceiver and navigator transmitter, simulator for array.

Instrument purge kit.

Other test gear (voltmeters, scopes, etc).

2. Laboratory tests

Instrument connection
- electronics chassis only
- connect dummy load, simulated array and power
- connect SAIL and computer.

Computer interface and control
- instrument sign-on
- passthrough to SAIL/OC devices
- task load and check
- data offload.

Transmitter output
- start transmission
- measure at dummy load -- DANGEROUS VOLTAGES --
- limit output if used with transducer in air instead of dummy load.

Clock setup
- check internal and external backup batteries
- track time against standard
- adjust aging coefficient and check results.

Engineering data
- confirm valid data on all channels.

Run a test schedule
- couple transmitter output back to receiver
- create a task with 1, 2, 4, 8 transmission sequences
- run and record data
- read back data, process, find peaks
- check engineering data.

Power drain
- measure drain when operating and when idle.

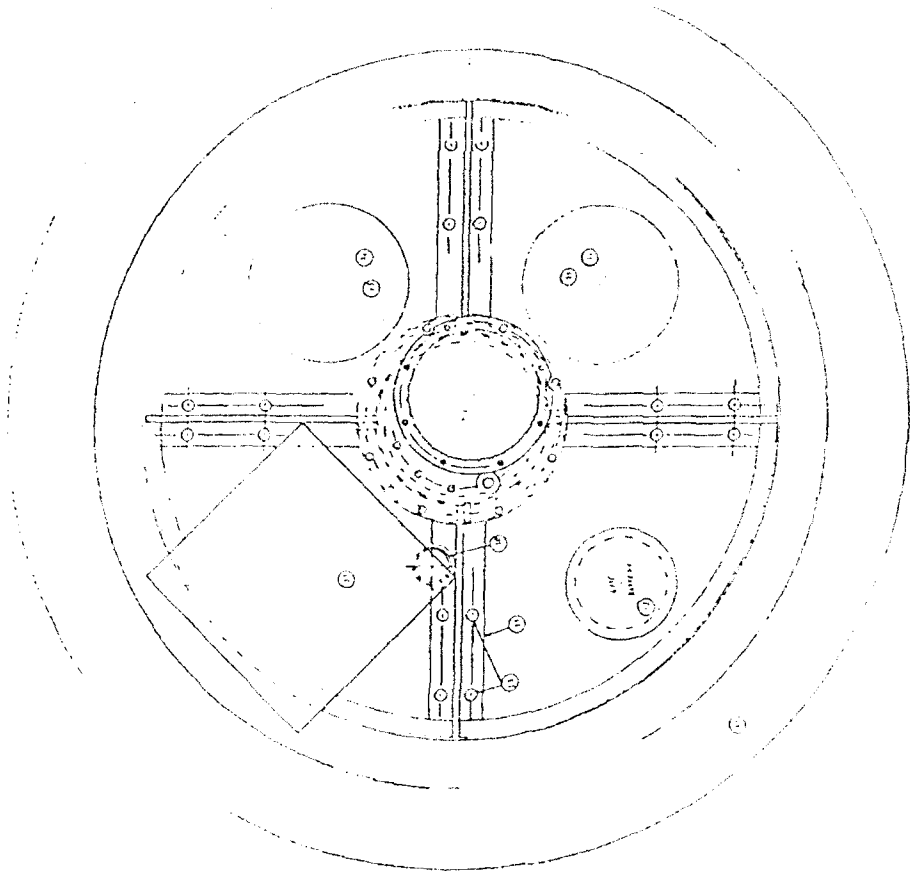
Data recovery
- read back data using Direct 6 board in computer
- confirm good data.

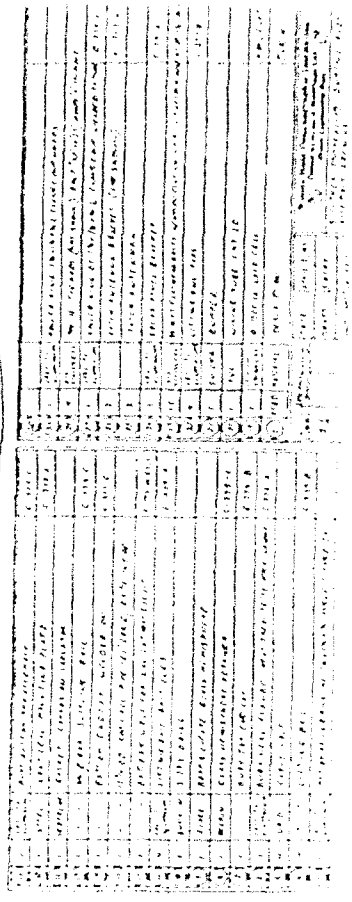
Appendix F: Mechanical Assembly Drawings

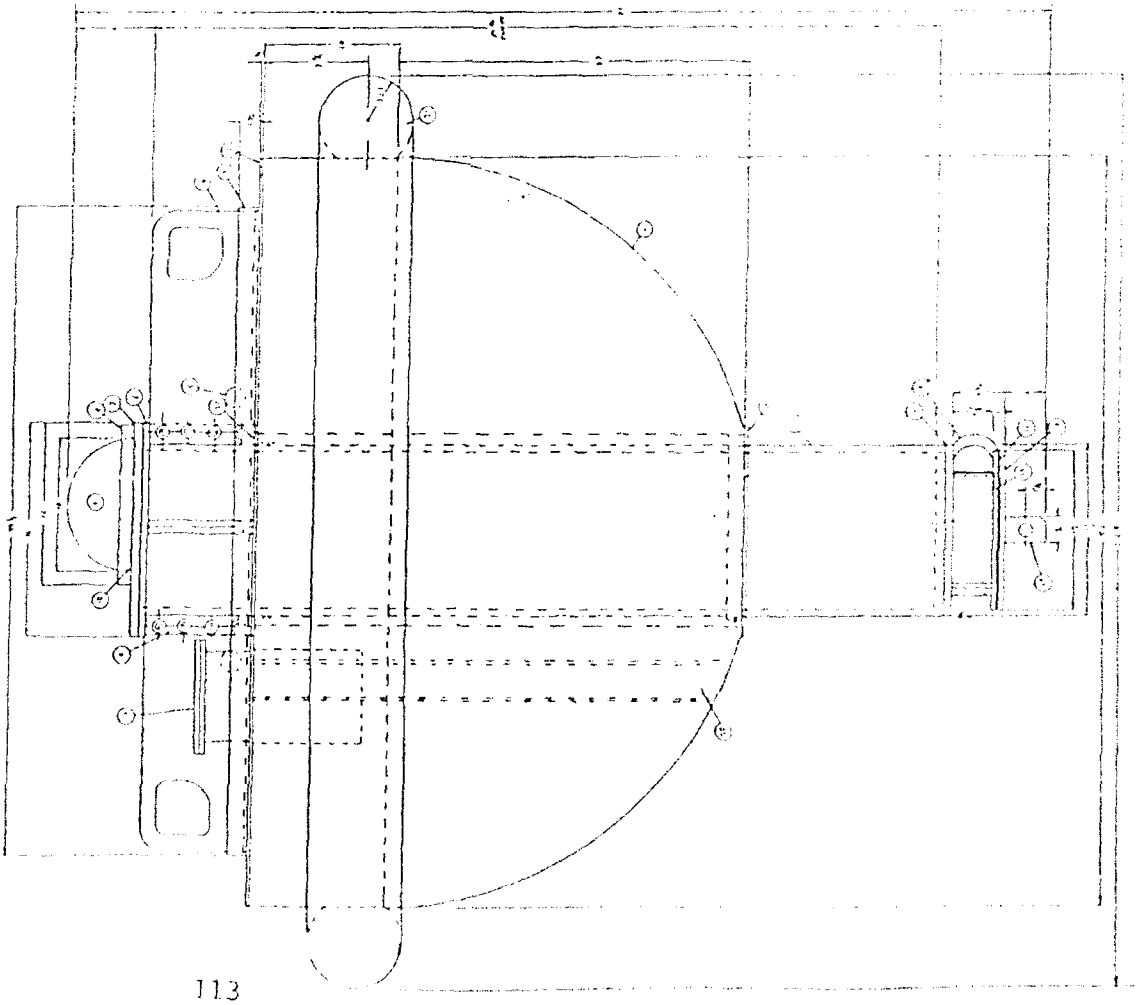
D-234: Surface buoy assembly drawing
C-216: Foam detail
C-239: Lifting bail detail
D-238: Top flange & lifting bail bracket weldment detail
D-224: Top end cap
C-225: 10" glass hemisphere retainer
C-244: Lite battery can
B-372: Plug for Brantner ports
C-223: Buoy bottom weldment
C-242: Load cell mounting plate
C-243: Patch antenna mount
C-241: Solar panel bracket

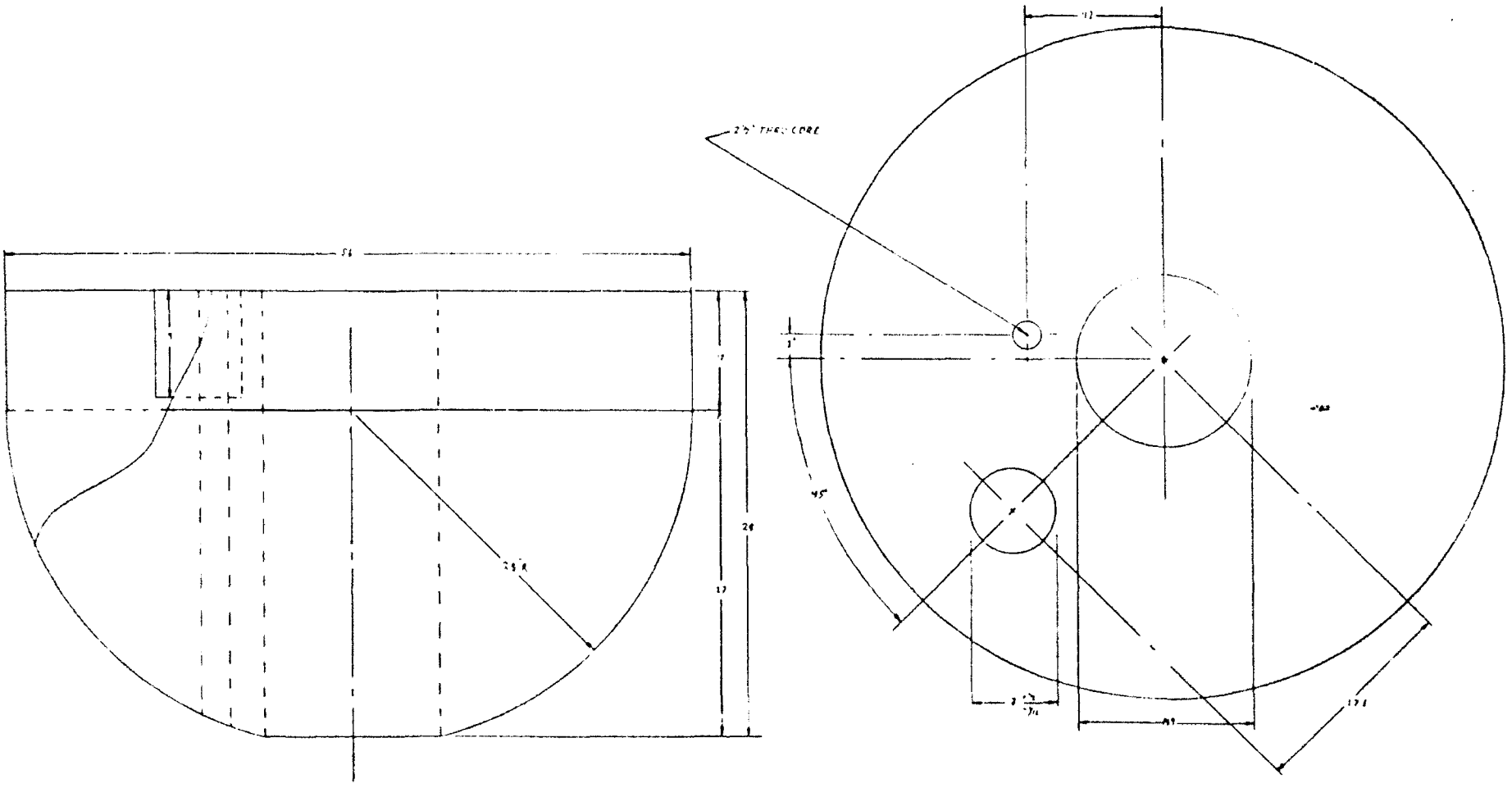
Mooring attachment parts

C-605: Load cell
B-387: Top urethane chain end
B-375: Bottom urethane chain end
C-217: Termination for TPR jacketed cable
D-240: Boot, springs & mold for TPR cable
B-370: U-Chain description





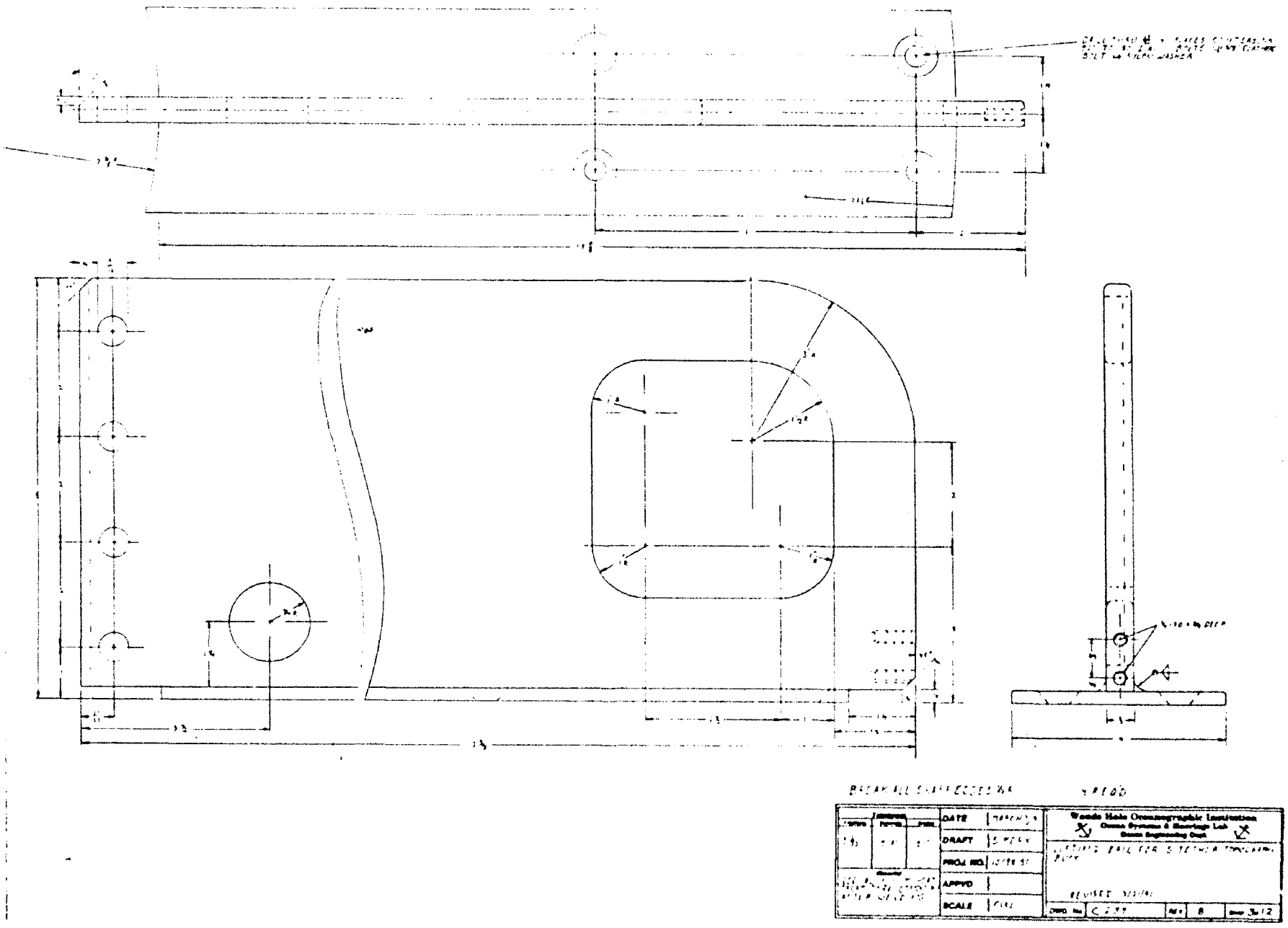




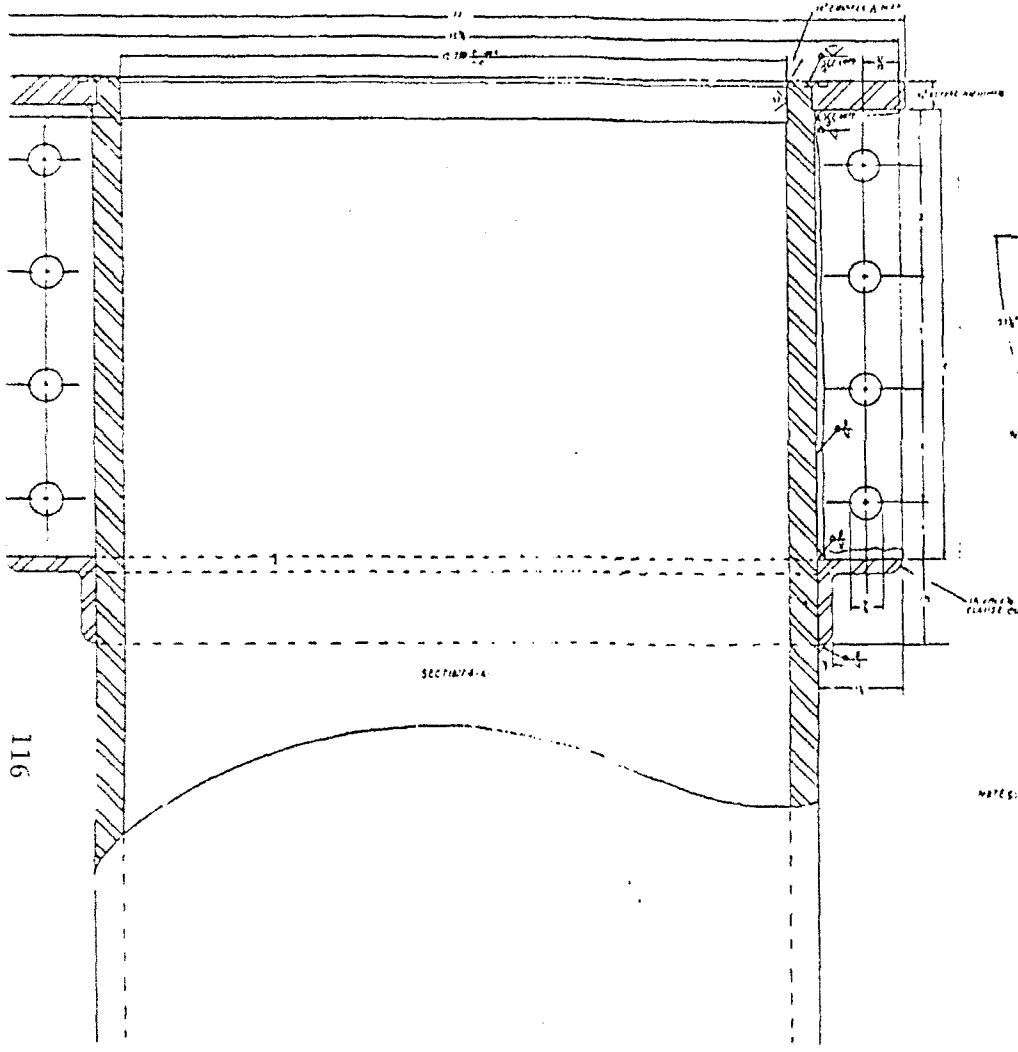
DATE	
DRAFT	
PROJ NO.	
APPVD	
SCALE	

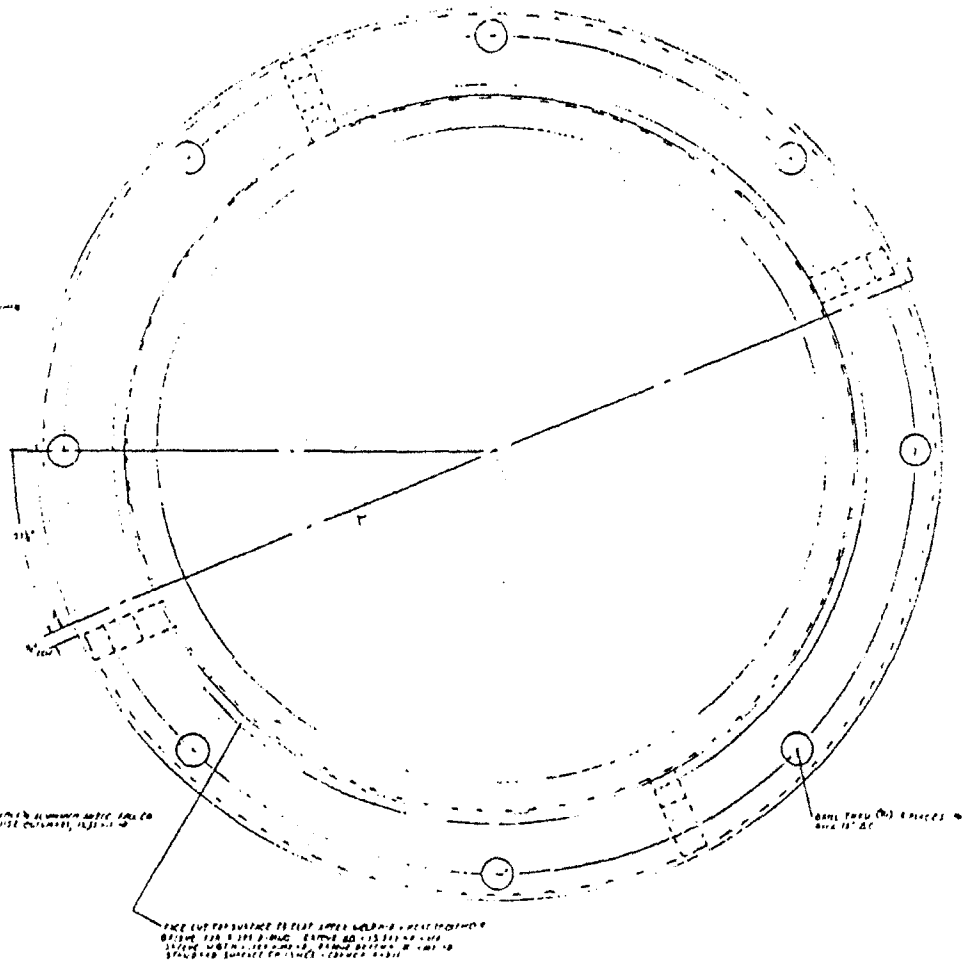
Woods Hole Oceanographic Institution
Ocean Systems & Moorings Lab
Ocean Engineering Dept

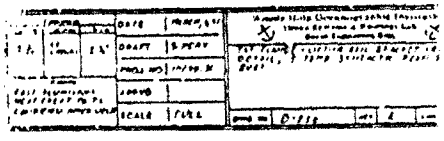
#216


BREAK ALL SHARP EDGES
DATE
DRAFT
PROJ. NO.
APPVD
SCALE
Woods Hole Oceanographic Institution
Ocean Systems & Moorings Lab
Ocean Engineering Dept.
REVISED
REV
B

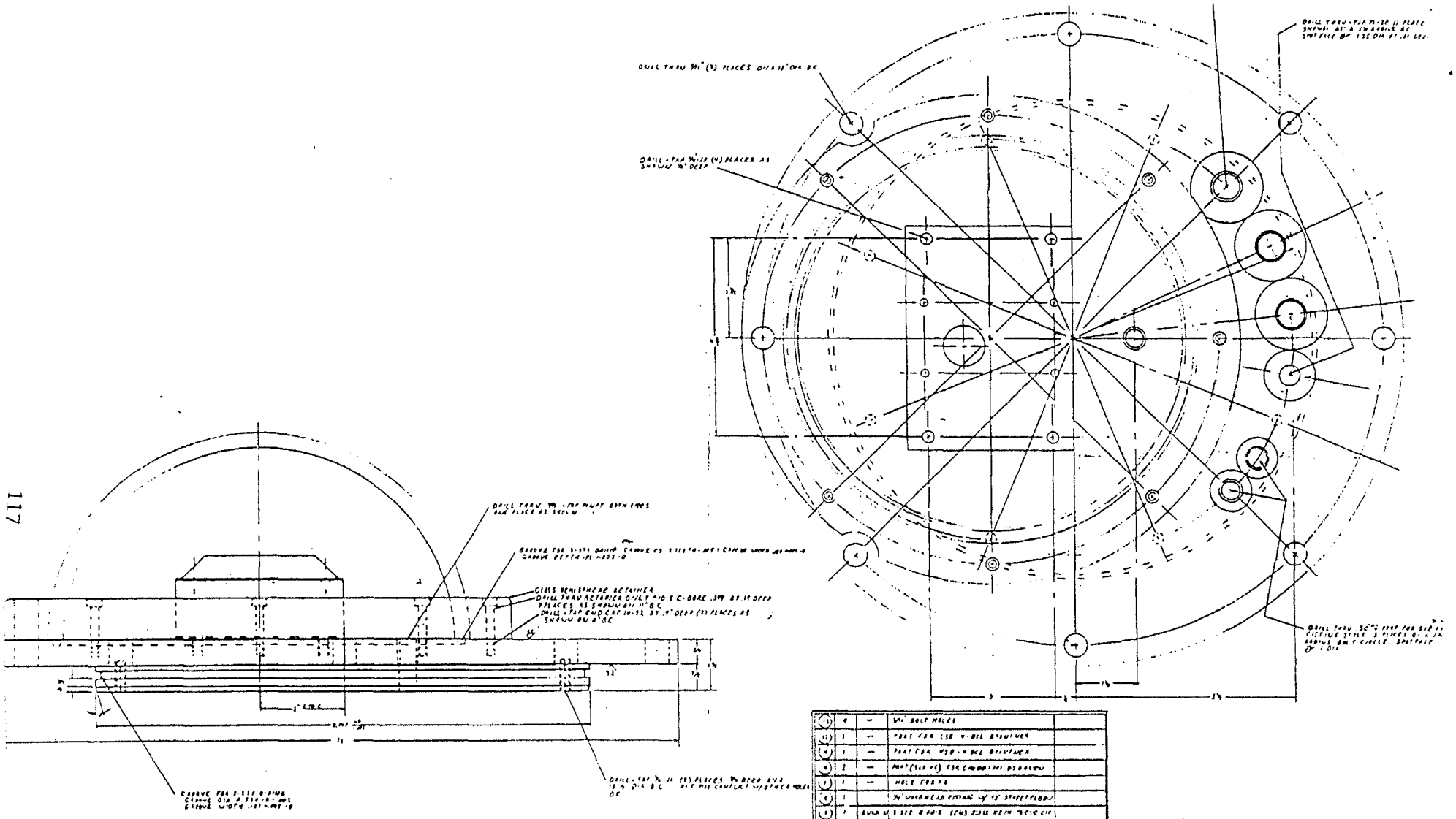
#239



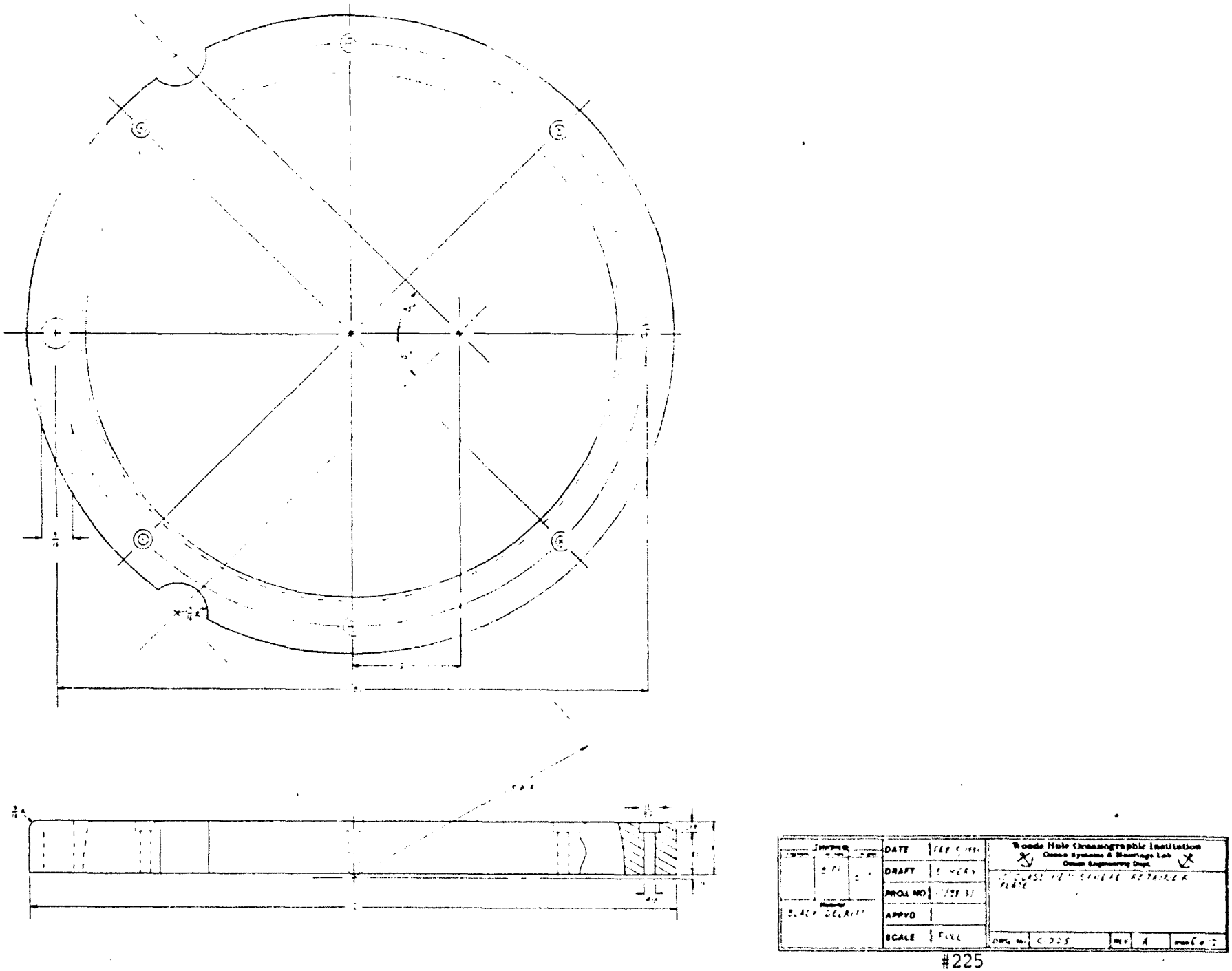




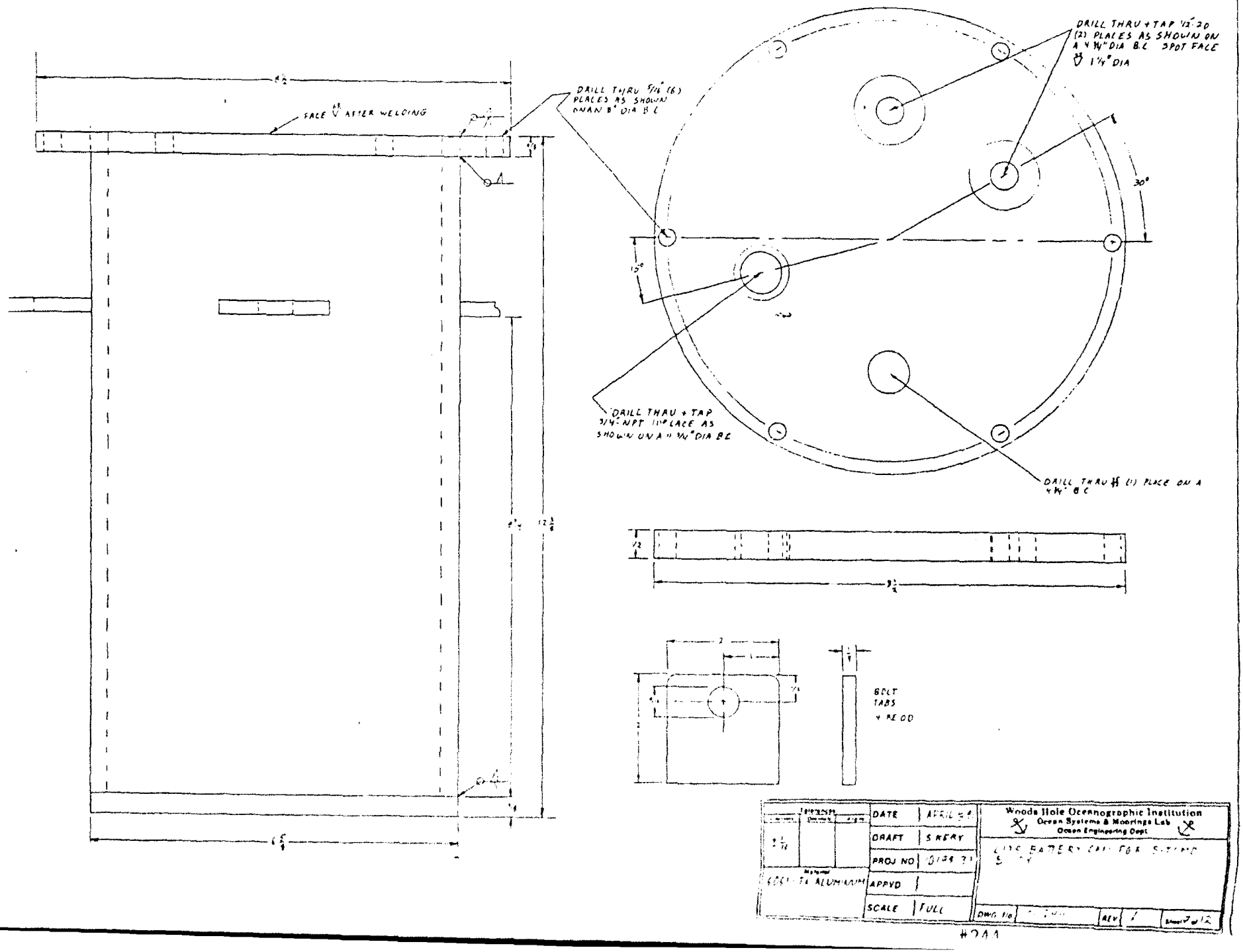
#238

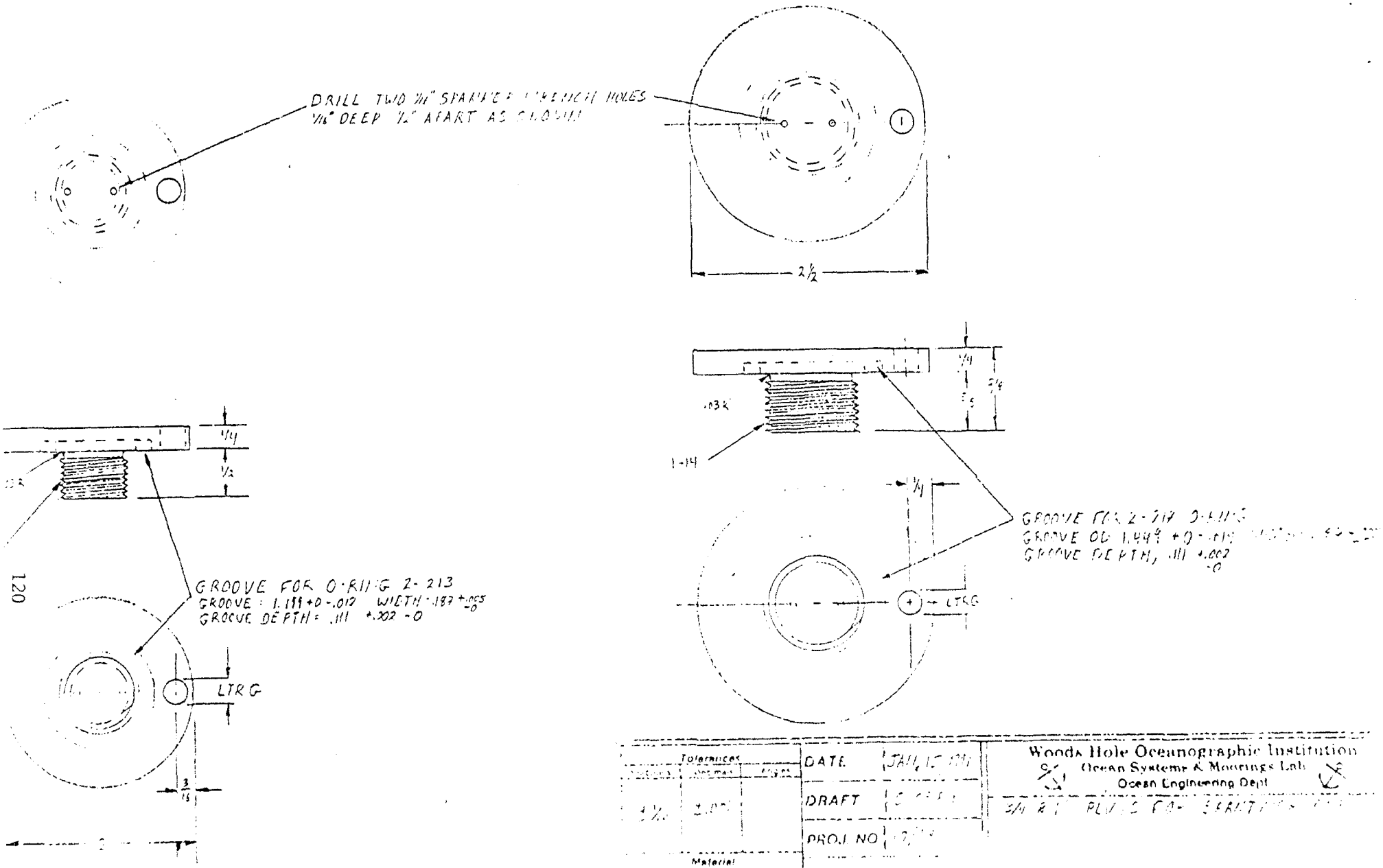


#224


DATE
DRAFT
PROJ. NO
APPVD
SCALE
Woods Hole Oceanographic Institution
Ocean Systems & Moorings Lab
Ocean Engineering Dept.
REV
A

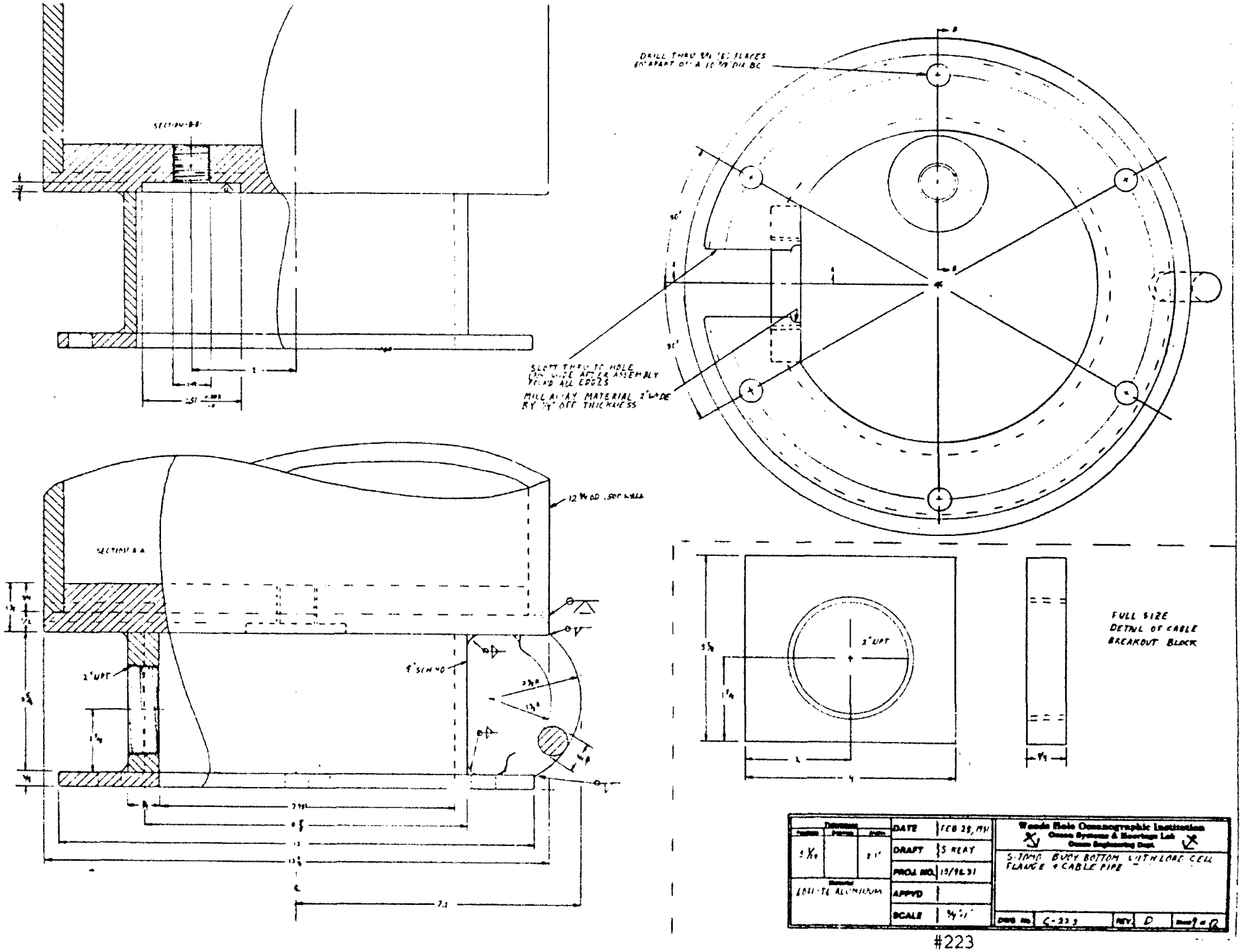
#225

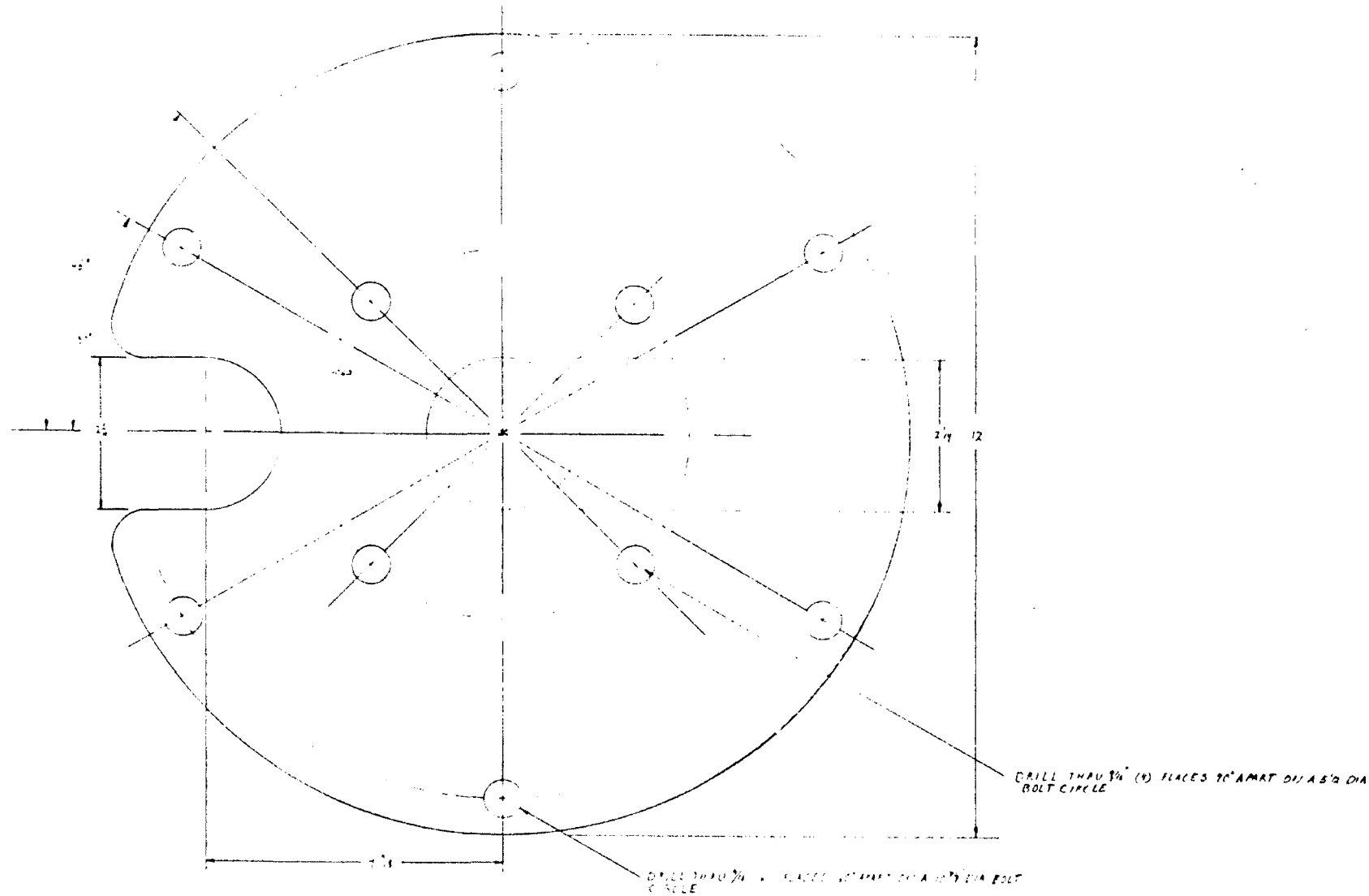

FACE AFTER WELDING
DRILL THRU 5/16" (6) PLACES AS SHOWN ON AN 8" DIA B.C.
DRILL THRU + TAP 1/2"-20 (2) PLACES AS SHOWN ON A 4 3/4" DIA B.C. SPOT FACE 1 1/4" DIA
30°
15°
DRILL THRU + TAP 3/4"-NPT 11° LACE AS SHOWN ON A 11 3/4" DIA B.C.
DRILL THRU (1) PLACE ON A 4 3/4" B.C.
BOLT TABS 4 REQD
DATE
DRAFT
PROJ NO
APPVD
SCALE
FULL
Woods Hole Oceanographic Institution
Ocean Systems & Moorings Lab
Ocean Engineering Dept
#244



Tolerances			DATE	JAN 15 1991	Woods Hole Oceanographic Institution Ocean Systems & Moorings Lab Ocean Engineering Dept
Fractions	Decimals	Angles	DRAFT	[illegible]	3/4 R [illegible] PLUGS FOR [illegible]
±1/16	±.005		PROJ. NO	[illegible]	
Material			APPVD		
[illegible]			SCALE	[illegible]	DWG NO [illegible]

#372

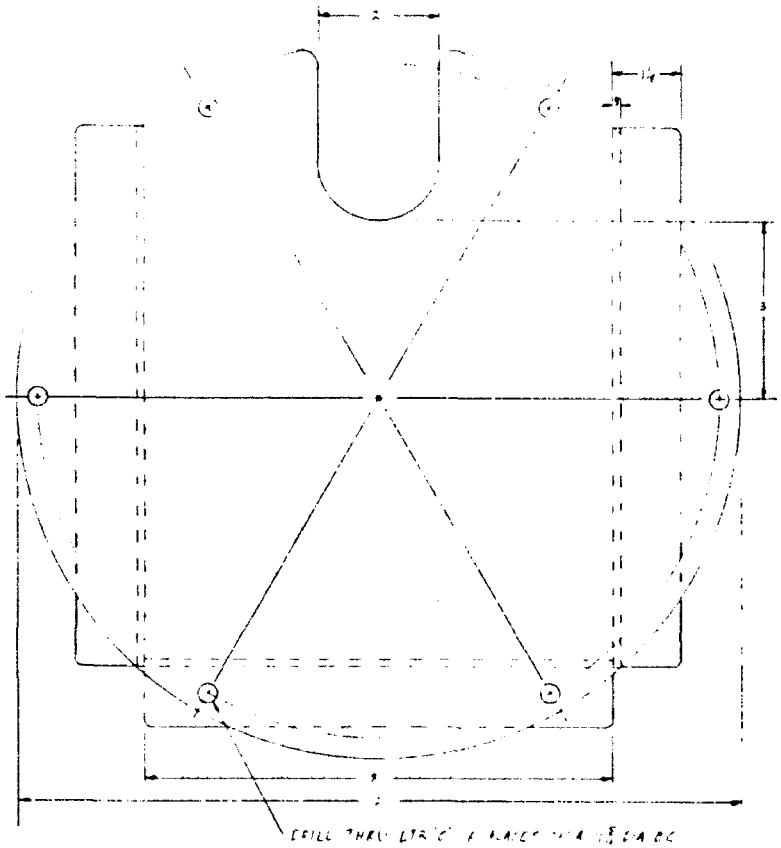

SECTION B-B
SECTION A-A
DRILL THRU 3/4 (6) PLACES 60° APART ON A 10 3/8" DIA BC
SLOTT THRU TO HOLE 1 3/4 WIDE AFTER ASSEMBLY ROUND ALL EDGES
MILL AWAY MATERIAL 2" WIDE BY 1/4" OFF THICKNESS
12 3/4 OD .500 WALL
2" NPT
4" SCH 40
FULL SIZE DETAIL OF CABLE BREAKOUT BLOCK
DATE FEB 28, 1991
DRAFT S KEAY
PROJ NO. 19/96.31
APPVD
SCALE 3/4"=1'
Woods Hole Oceanographic Institution
Ocean Systems & Moorings Lab
Ocean Engineering Dept.
S-TOMO BUOY BOTTOM WITH LOAD CELL FLANGE + CABLE PIPE
DWG No. C-223
REV. D
#223

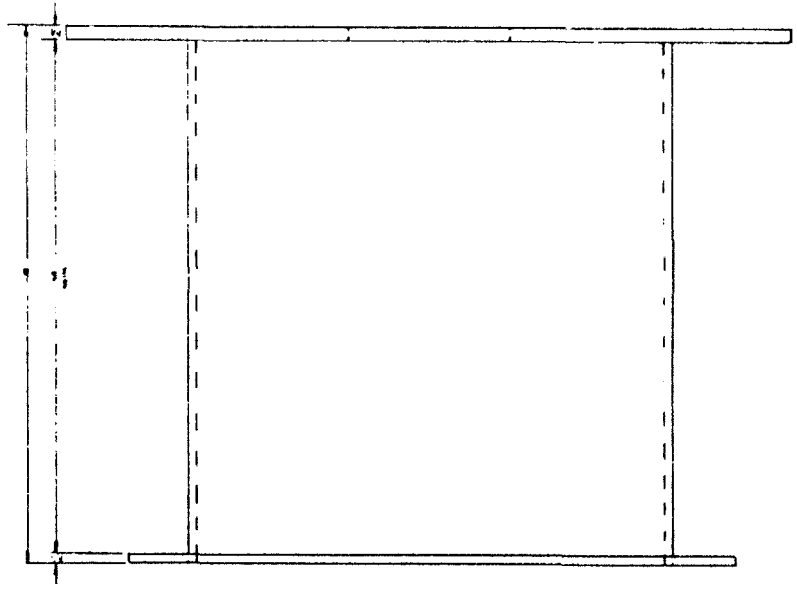


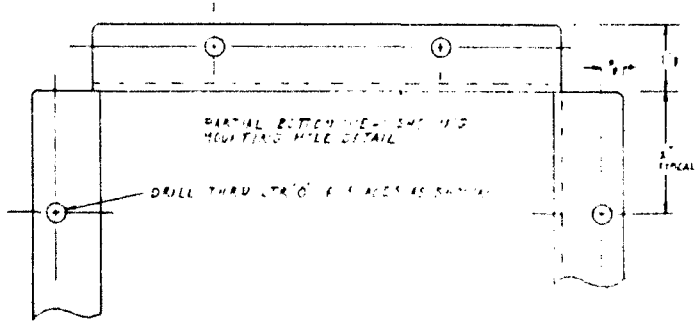
BREAK ALL SHARP EDGES

Tolerances	DATE	APRIL 2, 1991	Woods Hole Oceanographic Institution Ocean Systems & Moorings Lab Ocean Engineering Dept.
± 1/64 — ± 1°	DRAFT	S KERY	LOAD CELL MOUNTING PLATE
	PROJ. NO.	10/98 31	
Material: MILD STEEL 1/2" THICK	APPVD		
	SCALE	FULL	DWG. No. C 242 REV A Sheet 12

#242



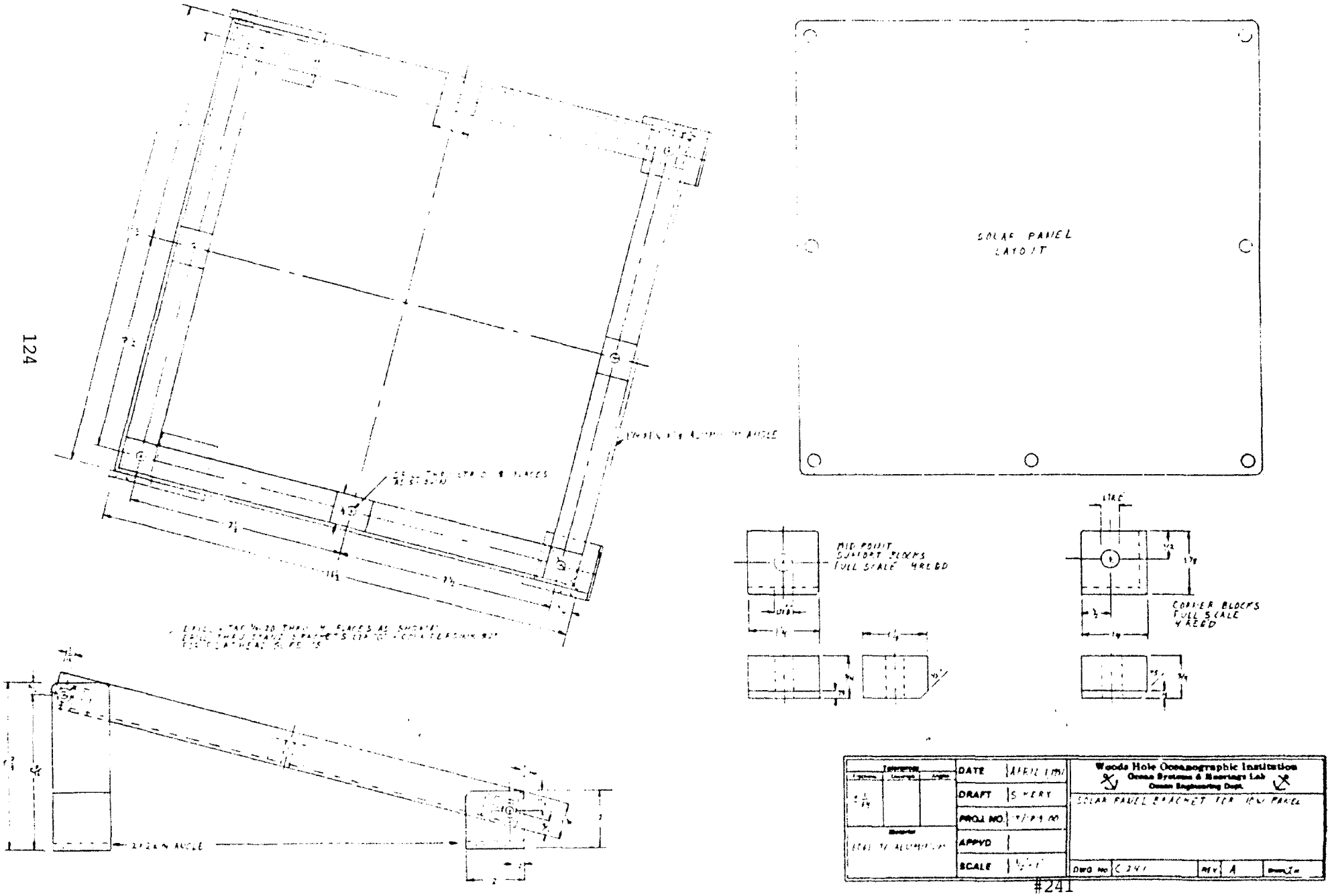




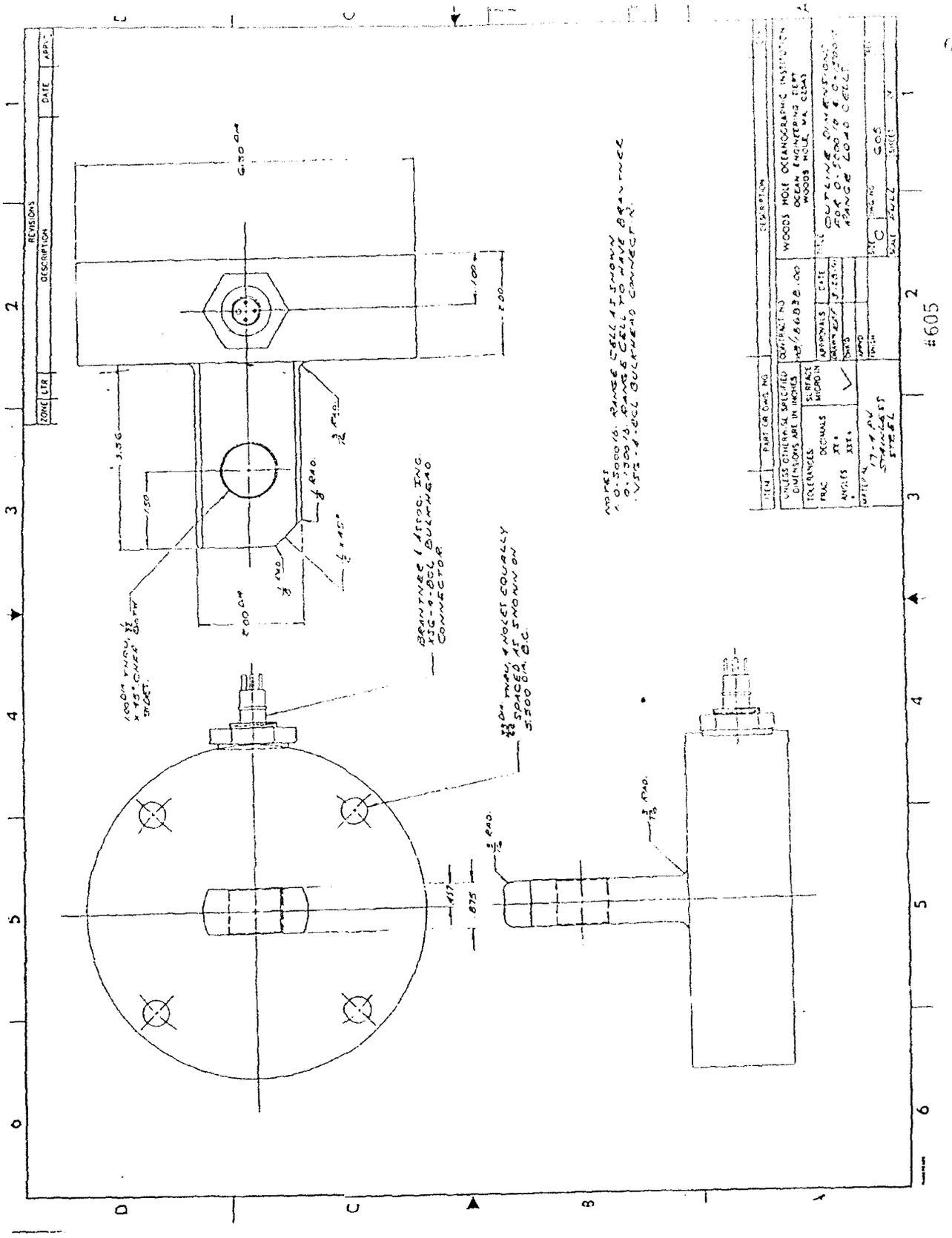
2 REQD

DATE	APRIL 2, 19[illegible]	Woods Hole Oceanographic Institution Ocean Systems & Moorings Lab Ocean Engineering Dept	
DRAFT	S. KERY	ARGOS PATCH ANTENNA BRACKET TALL VERSION SHORT VERSION IS DWG C 1[illegible]	
PROJ. NO.	[illegible]		
APPVD			
SCALE	3/4" = 1'	DWG No. C 243	REV A

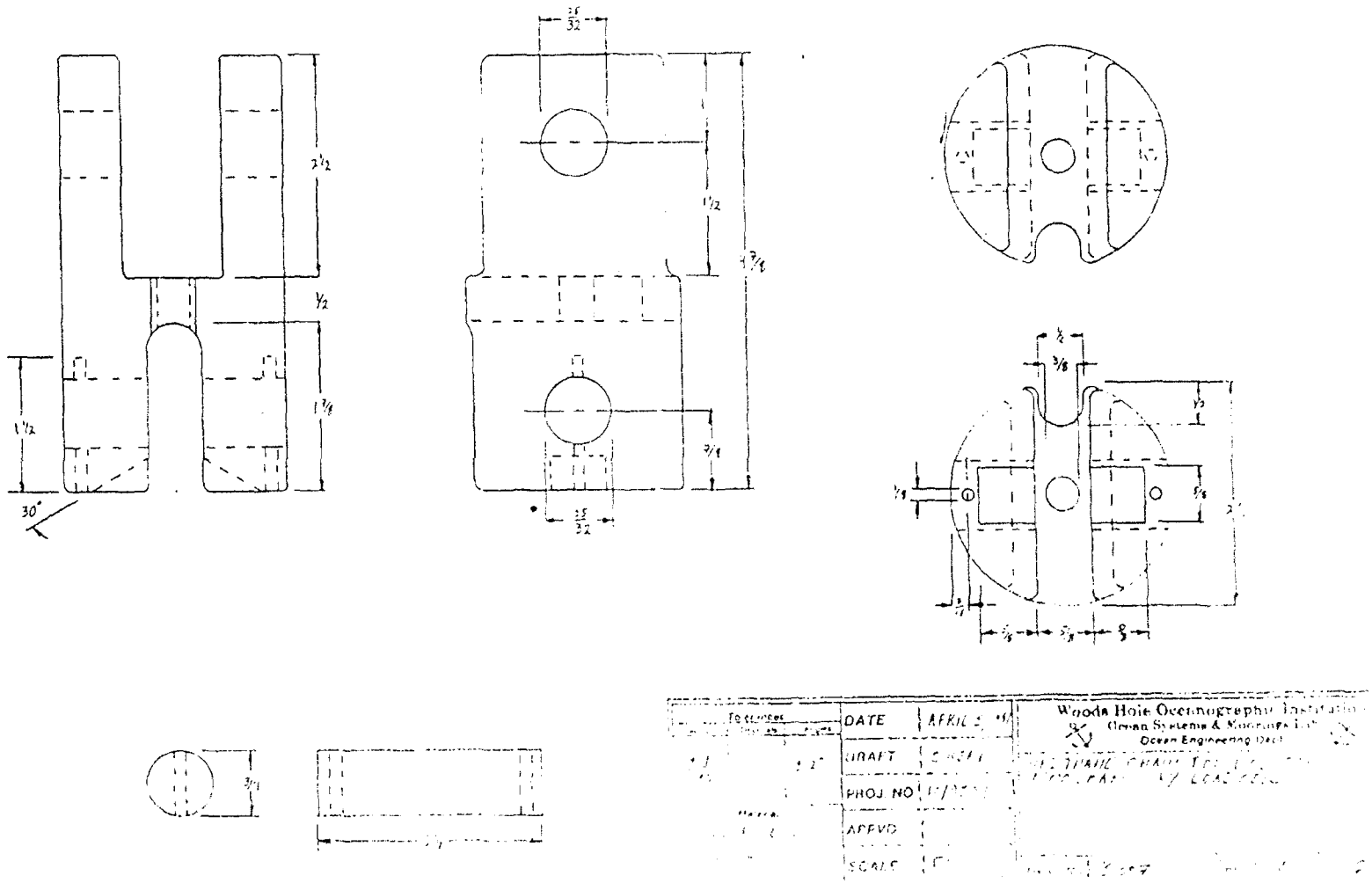
#243


SOLAR PANEL
LAYOUT
MID POINT
SUPPORT BLOCKS
FULL SCALE 4 REQD
CORNER BLOCKS
FULL SCALE
4 REQD
DATE
DRAFT
PROJ. NO.
APPVD
SCALE
Woods Hole Oceanographic Institution
Ocean Systems & Moorings Lab
Ocean Engineering Dept.
SOLAR PANEL BRACKET FOR
DWG NO
REV
A

#241

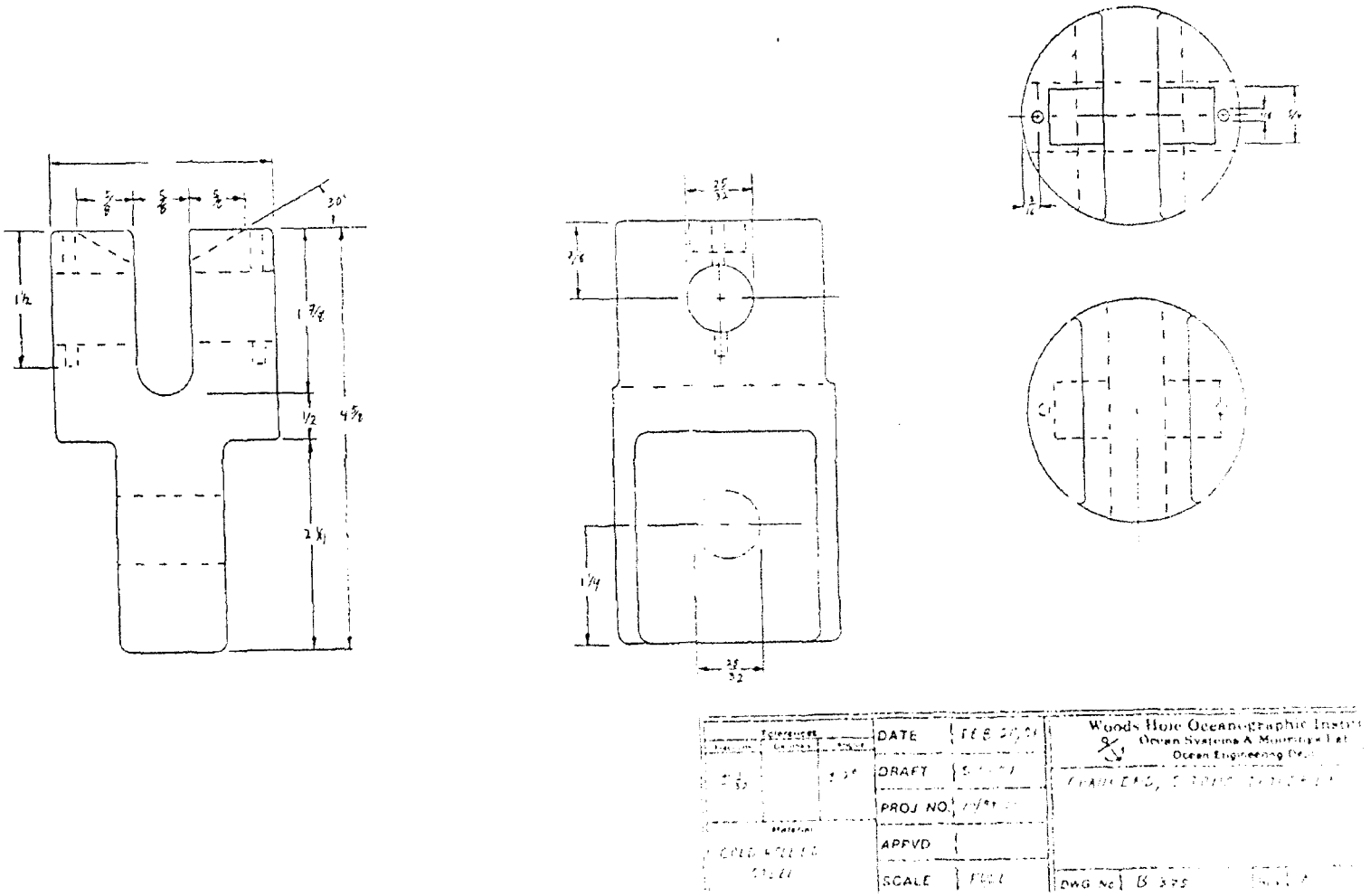


REVISIONS
ZONE
LTR
DESCRIPTION
DATE
APPR
BRANTNER & ASSOC. INC.
XSG-4-BCL BULKHEAD
CONNECTOR
NOTES
1. 0-5000 lb. RANGE CELL AS SHOWN
2. 0-1500 lb. RANGE CELL TO HAVE BRANTNER
3. XSG-4-BCL BULKHEAD CONNECTOR.
WOODS HOLE OCEANOGRAPHIC INSTITUTION
OCEAN ENGINEERING DEPT
WOODS HOLE, MA 02543
OUTLINE DIMENSIONS
FOR 0-5000 lb & 0-1500 lb
RANGE LOAD CELLS
UNLESS OTHERWISE SPECIFIED
DIMENSIONS ARE IN INCHES
TOLERANCES:
FRAC
DECIMALS
ANGLES
SURFACE
MICROIN
APPROVALS
DATE
MATERIAL
17-4 PH
STAINLESS
STEEL
605
#605



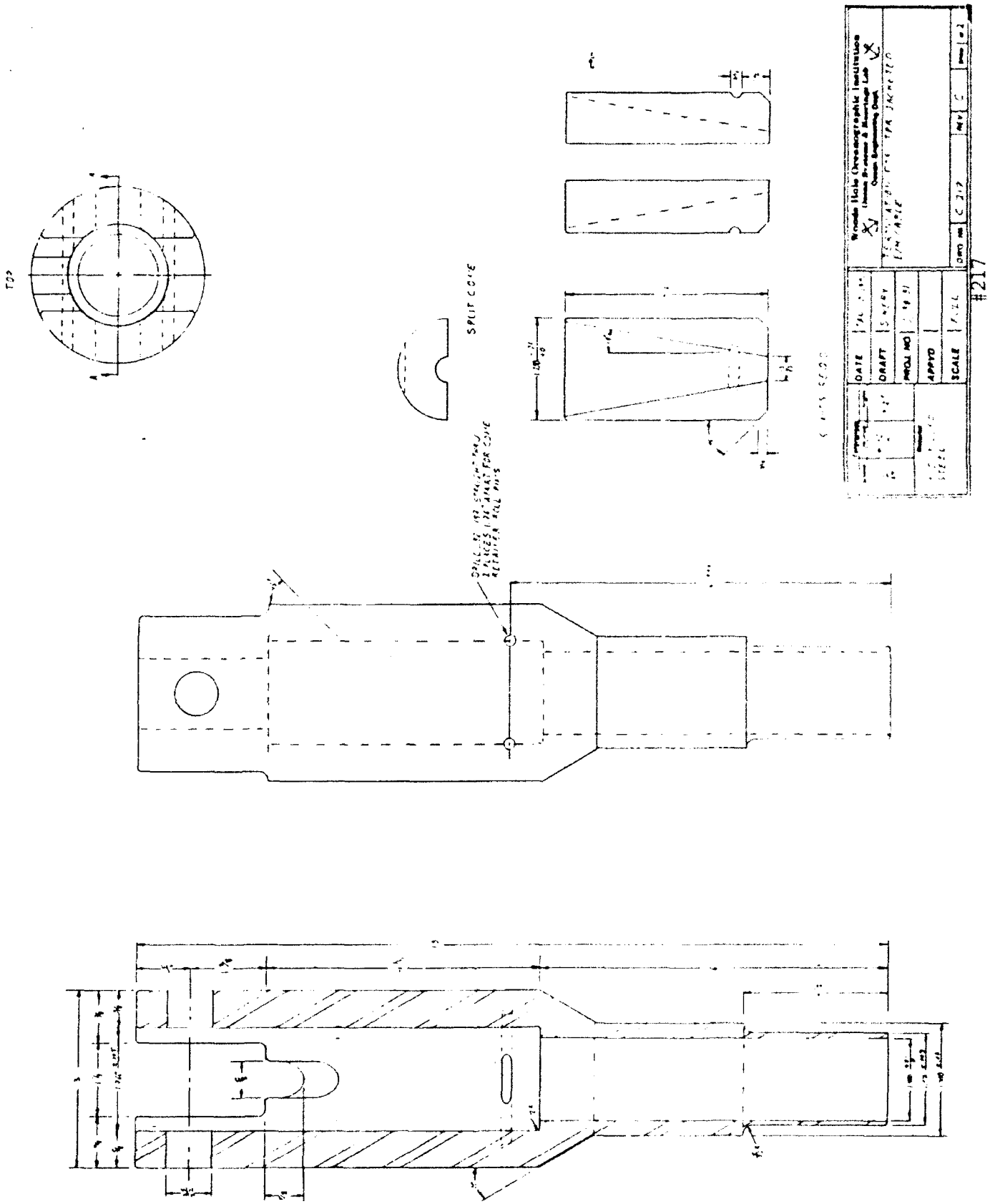
30°
DATE
DRAFT
PROJ. NO
APPVD
SCALE
Woods Hole Oceanographic Institution
Ocean Engineering Dept

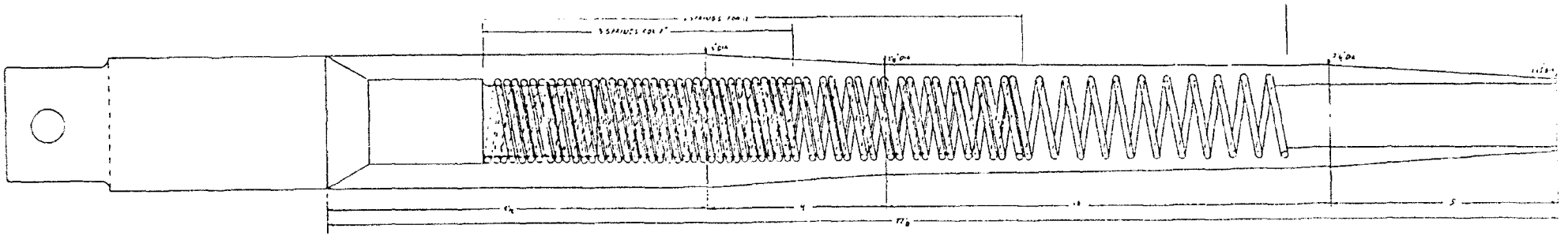
#387

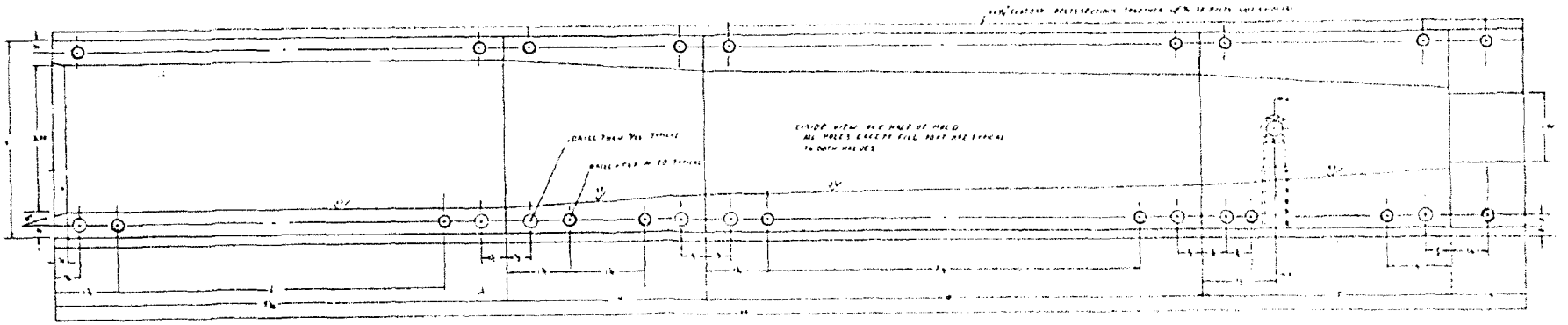


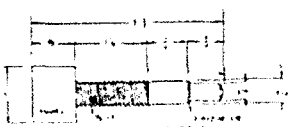
Woods Hole Oceanographic Instit
Ocean Systems & Moorings Lab
Ocean Engineering Dept
DATE
DRAFT
PROJ NO.
APPVD
SCALE
DWG No B 375

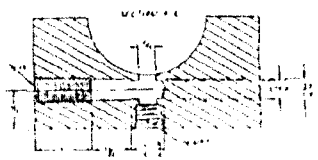
#375


TOP
SPLIT COVE
Woods Hole Oceanographic Institution
DATE
DRAFT
PROJ NO
APPVD
SCALE
#217

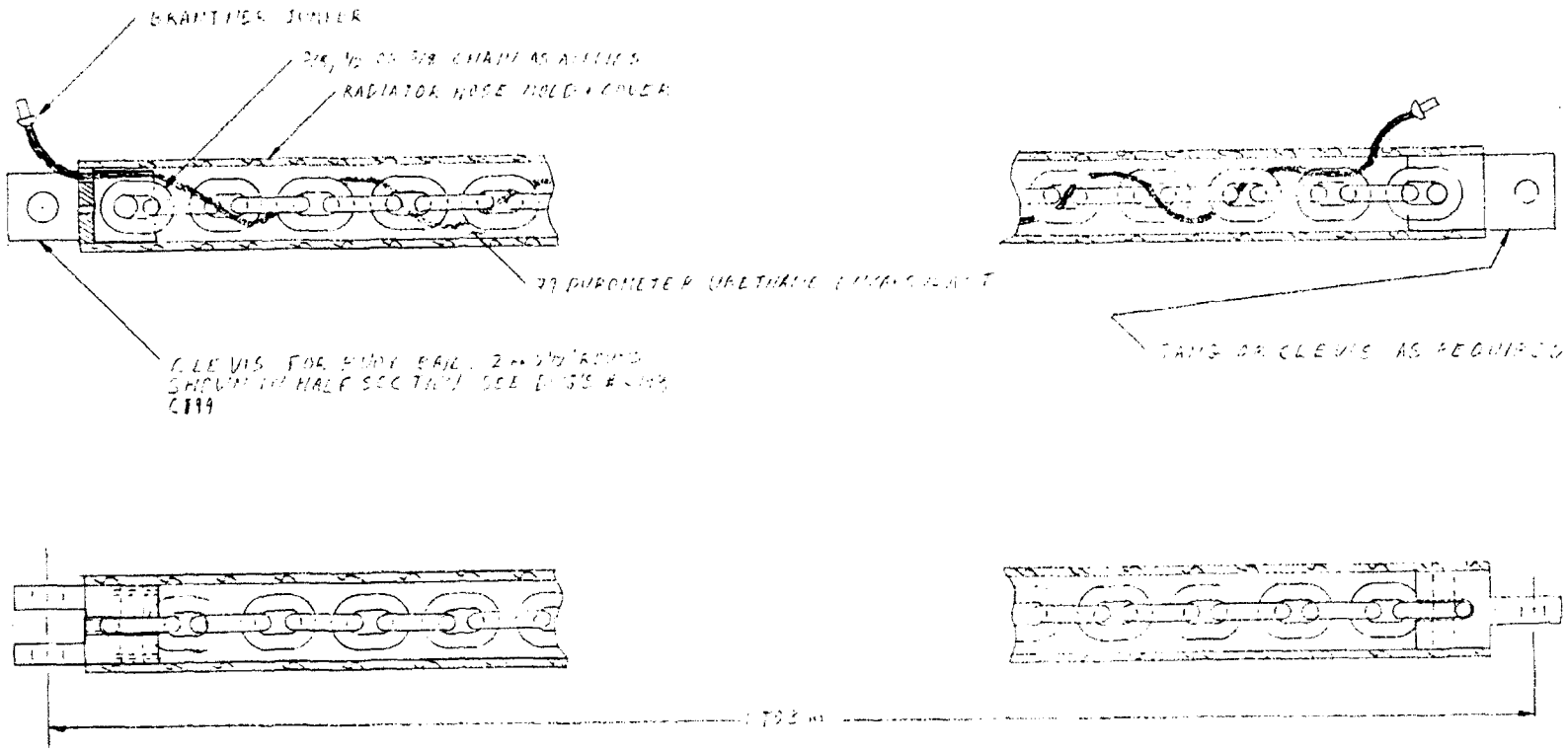








#240



Tolerances			DATE	JAN 5, 1991	Woods Hole Oceanographic Institution Ocean Systems & Moorings Lab Ocean Engineering Dept.
Fractions	Decimals	Angles	DRAFT	[illegible]	POTTED CHAIN METHOD FOR CABLE TO [illegible] TERMINATIONS
Material			PROJ. NO.	[illegible]	#370

Appendix G: S-Tether Tomography Mooring Analysis

The following tables show the results of the analyses performed to determine the effects of currents on the S-Tether Tomography Mooring while deployed at Site D.

G.1 Typical Current Profile

CURRENT PROFILE -	DEPTH(m)	CURRENT(cm/S)	DIR(deg TRUE)	VERT.(cm/s)
1	0.	35.000	0.	.0000
2	100.	30.000	0.	.0000
3	200.	30.000	0.	.0000
4	500.	10.000	0.	.0000
5	1000.	8.000	0.	.0000
6	2000.	6.000	0.	.0000
7	2700.	7.000	0.	.0000

SUMMARY OF RESULTS FOR MOORING S-TOMOGRAPHY 05-29-1991 11:26

			ORIENTATION					FORCES		CURRENT		
#	COMP.	NAME	EXC	AZ	DEP	LEN	INC	TENS	AZ	SPEED	DIR	VER
			m	deg	m	m	deg	lbs	deg	cm/s	deg	mm/s
	104	DISCUS	331	0	0	*	89	76	0	35	0	0
1	853	BOOT	330.8	0	1.0	1 *	4	56	0	35	0	0
2	105	TPR	330.0	0	12.0	11 *	12	54	0	35	0	0
21	105	TPR	161.9	0	100.8	11 *	-95	40	0	30	0	0
22	912	60 SPHERE	161.8	0	104.0	3 *	1	2994	0	30	0	0
23	107	EMS	157.7	0	302.4	198 *	2	2880	0	30	0	0
26	107	EMS	135.3	0	897.1	198 *	2	2535	0	9	0	0
27	TOM	O T	135.0	0	901.9	5 *	4	1534	0	8	0	0
28	ARR	AY	132.6	0	935.4	34 *	4	1533	0	8	0	0
29	904	17 GS SR2	132.3	0	939.6	4 *	4	1731	0	8	0	0
30	103	5/16 WR	125.7	0	1039.9	100 *	4	1690	0	8	0	0
34	103	5/16 WR	96.9	0	1440.7	101 *	4	1524	0	7	0	0
35	402	5/16 KEV	94.3	0	1475.0	34 *	4	1523	0	7	0	0
69	402	5/16 KEV	2.5	0	2640.9	35 *	5	1498	0	7	0	0
70	903	17 GB SR	1.9	0	2650.9	10 *	4	2027	0	7	0	0
74	903	17 GB SR	.3	0	2691.1	10 *	2	4142	0	7	0	0
75	508	1/2 TX	.2	0	2693.3	2 *	2	4127	0	7	0	0
76	812	BENTHOS R	.2	0	2694.5	1 *	2	4070	0	7	0	0
77	508	1/2 TX	.2	0	2695.0	1 *	2	4067	0	7	0	0
86	508	1/2 TX	.0	902	699.7	1 *	2	4035	0	7	0	0

G.2 Design Current Profile

CURRENT PROFILE -	DEPTH(m)	CURRENT(cm/S)	DIR(deg TRUE)	VERT.(cm/s)
1	0.	85.000	0.	.0000
2	100.	85.000	0.	.0000
3	200.	75.000	0.	.0000
4	500.	30.000	0.	.0000
5	1000.	20.000	0.	.0000
6	2700.	20.000	0.	.0000

SUMMARY OF RESULTS FOR MOORING S-TOMOGRAPHY 05-29-1991 11:28

			ORIENTATION				FORCES		CURRENT		
#	COMP.	NAME	EXC	AZ	DEP	LEN	INC	TENS	AZ	SPEED	DIR VER
			m	deg	m	m	deg	lbs	deg	cm/s	deg mm/s
	104	DISCUS	869	0	0	*	89	1695	0	85	0 0
1	853	BOOT	869.2	0	1.0	1 *	1	1675	0	85	0 0
2	105	TPR	869.0	0	12.0	11 *	3	1674	0	85	0 0
21	105	TPR	814.6	0	212.8	11 *	27	1646	0	74	0 0
22	912	60 SPHERE	814.1	0	215.9	3 *	11	4531	0	73	0 0
23	107	EMS	775.4	0	411.3	199 *	13	4420	0	58	0 0
26	107	EMS	632.6	0	991.4	199 *	15	4085	0	22	0 0
27	TOM	O T	631.1	0	995.9	5 *	20	3128	0	20	0 0
28	ARR	AY	619.6	0	1028.2	34 *	20	3128	0	20	0 0
29	904	17 GS SR2	618.2	0	1032.1	4 *	19	3315	0	20	0 0
30	103	5/16 WR	585.6	0	1127.5	101 *	19	3277	0	20	0 0
34	103	5/16 WR	449.2	0	1507.3	101 *	20	3120	0	20	0 0
35	402	5/16 KEV	437.1	0	1539.9	35 *	20	3120	0	20	0 0
69	402	5/16 KEV	14.9	0	2643.4	35 *	21	3096	0	20	0 0
70	903	17 GB SR	11.7	0	2652.9	10 *	19	3596	0	20	0 0
74	903	17 GB SR	1.9	0	2691.9	10 *	13	5645	0	20	0 0
75	508	1/2 TX	1.4	0	2694.0	2 *	13	5631	0	20	0 0
76	812	BENTHOS R	1.1	0	2695.2	1 *	13	5575	0	20	0 0
77	508	1/2 TX	1.0	0	2695.7	1 *	13	5572	0	20	0 0
86	508	1/2 TX	.0	90	2700.3	1 *	13	5541	0	20	0 0

G.3 Survival Current Profile

CURRENT PROFILE

DEPTH	SPEED	DIRECTION
.0	125.0 CM-SEC	.0 DEG
100.0	125.0 CM-SEC	.0 DEG
200.0	110.0 CM-SEC	.0 DEG
500.0	50.0 CM-SEC	.0 DEG
1000.0	30.0 CM-SEC	.0 DEG
2700.0	30.0 CM-SEC	.0 DEG

SUMMARY OF RESULTS FOR MOORING S-TOMOGRAPHY 05-29-1991 11:30

			ORIENTATION					FORCES		CURRENT		
#	COMP.	NAME	EXC	AZ	DEP	LEN	INC	TENS	AZ	SPEED	DIR	VER
			m	deg	m	m	deg	lbs	deg	cm/s	deg	mm/s
	104	DISCUS	988	0	0	*	89	4950	0	125	0	0
1	853	BOOT	988.2	0	1.0	1 *	1	4931	0	125	0	0
2	105	TPR	988.0	0	12.1	11 *	2	4929	0	125	0	0
21	105	TPR	946.1	0	217.8	11 *	20	4906	0	107	0	0
22	912	60 SPHERE	945.4	0	220.9	3 *	14	7802	0	106	0	0
23	107	EMS	894.7	0	415.2	201 *	17	7696	0	86	0	0
26	107	EMS	712.1	0	989.2	201 *	19	7367	0	34	0	0
27	TOM	O T	710.4	0	993.7	5 *	22	6432	0	30	0	0
28	ARR	AY	697.4	0	1026.2	35 *	22	6431	0	30	0	0
29	904	17 GS SR2	695.9	0	1030.1	4 *	21	6617	0	30	0	0
30	103	5/16 WR	658.7	0	1124.8	102 *	22	6579	0	30	0	0
34	103	5/16 WR	506.0	0	1501.8	102 *	23	6425	0	30	0	0
35	402	5/16 KEV	492.4	0	1534.6	35 *	23	6425	0	30	0	0
69	402	5/16 KEV	19.6	0	2644.6	36 *	24	6401	0	30	0	0
70	903	17 GB SR	15.8	0	2653.9	10 *	22	6896	0	30	0	0
74	903	17 GB SR	2.7	0	2691.9	10 *	18	8908	0	30	0	0
75	508	1/2 TX	2.0	0	2694.0	2 *	18	8893	0	30	0	0
76	812	BENTHOS R	1.6	0	2695.1	1 *	18	8839	0	30	0	0
77	508	1/2 TX	1.5	0	2695.6	1 *	18	8836	0	30	0	0
86	508	1/2 TX	.0	90	2700.1	1 *	18	8806	0	30	0	0

DOCUMENT LIBRARY

March 11, 1991

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 5UB
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

50272-101

REPORT DOCUMENTATION PAGE	1. REPORT NO. WHOI-92-29	2.	3. Recipient's Accession No.

4. Title and Subtitle
Real-Time Tomography Mooring

5. Report Date
June 1992

6.

7. Author(s) James Lynch, Daniel Frye, Kenneth Peal, Stephen Liberatore, Sean Kery, Edward Hobart, Arthur Newhall and Stephen Smith

8. Performing Organization Rept. No.
WHOI-92-29

9. Performing Organization Name and Address
Woods Hole Oceanographic Institution
Woods Hole, Massachusetts 02543

10. Project/Task/Work Unit No.

11. Contract(C) or Grant(G) No.
(C) N000-14-C-90-0098
(G)

12. Sponsoring Organization Name and Address
Office of Naval Technology

13. Type of Report & Period Covered
Technical Report

14.

15. Supplementary Notes
This report should be cited as: Woods Hole Oceanog. Inst. Tech. Rept., WHOI-92-29.

16. Abstract (Limit: 200 words)

A real-time tomography system has been developed which combines ocean acoustic tomography with satellite-based time keeping and satellite telemetry. The basis of the system is the acoustic tomography transceiver and its associated acoustic navigation grid. To this basic system, a link to the surface has been added to provide a pathway for telemetry of the tomographic data to shore and a downlink for satellite-derived time which is used to correct the transceiver's clock. The surface buoy contains a GPS receiver, clock comparator, system controller and multiple ID Argos transmitters. Processed tomography signals, transceiver location data, time, time drift and surface buoy engineering data are transmitted to satellite using a total of 32 data buffers transmitted every eight minutes. The report describes the real-time tomography system in detail, with particular emphasis on the modifications implemented to convert the standard tomography instrument to a real-time oceanographic tool.

17. Document Analysis **a. Descriptors**
acoustic tomography
real-time
telemetry

b. Identifiers/Open-Ended Terms

c. COSATI Field/Group

18. Availability Statement	19. Security Class (This Report)	21. No. of Pages
Approved for public release; distribution unlimited.	UNCLASSIFIED	146
	20. Security Class (This Page)	22. Price

(See ANSI-Z39.18) *See Instructions on Reverse*

OPTIONAL FORM 272 (4-77)
(Formerly NTIS-35)
Department of Commerce