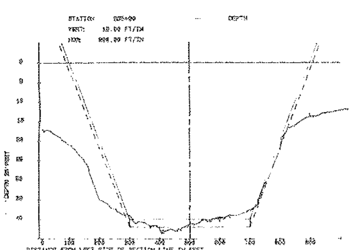
Lillycrop

US Army Corps of Engineers



HYDRAULICS LABORATORY

SURVEYING AND MAPPING RESEARCH AND DEVELOPMENT PROGRAM

TECHNICAL REPORT HL-89-21

SMALL-BOAT SURVEY SYSTEMS

by

Barry W. McCleave

Instrumentation Services Division

and

E. Dale Hart

Hydraulics Laboratory

DEPARTMENT OF THE ARMY
Waterways Experiment Station, Corps of Engineers
3909 Halls Ferry Road, Vicksburg, Mississippi 39180-6199

September 1989

Final Report

Approved For Public Release; Distribution Unlimited

Prepared for DEPARTMENT OF THE ARMY
US Army Corps of Engineers
Washington, DC 20314-1000

Under Work Unit No. 31099

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1a. REPORT SECURITY CLASSIFICATION Unclassified	1b. RESTRICTIVE MARKINGS
2a. SECURITY CLASSIFICATION AUTHORITY	3. DISTRIBUTION/AVAILABILITY OF REPORT Approved for public release; distribution unlimited.
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE	
4. PERFORMING ORGANIZATION REPORT NUMBER(S) Technical Report HL-89-21	5. MONITORING ORGANIZATION REPORT NUMBER(S)
6a. NAME OF PERFORMING ORGANIZATION See reverse 6b. OFFICE SYMBOL (If applicable) See reverse	7a. NAME OF MONITORING ORGANIZATION
6c. ADDRESS (City, State, and ZIP Code) 3909 Halls Ferry Road Vicksburg, MS 39180-6199	7b. ADDRESS (City, State, and ZIP Code)
8a. NAME OF FUNDING/SPONSORING ORGANIZATION US Army Corps of Engineers 8b. OFFICE SYMBOL (If applicable)	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER
8c. ADDRESS (City, State, and ZIP Code) Washington, DC 20314-1000	10. SOURCE OF FUNDING NUMBERS See reverse PROGRAM ELEMENT NO. | PROJECT NO. | TASK NO. | WORK UNIT ACCESSION NO.

11. TITLE (Include Security Classification)
Small-Boat Survey Systems

12. PERSONAL AUTHOR(S)
McCleave, Barry W.; Hart, E. Dale

13a. TYPE OF REPORT	13b. TIME COVERED	14. DATE OF REPORT (Year, Month, Day)	15. PAGE COUNT
Final report	FROM ______ TO ______	September 1989	291

16. SUPPLEMENTARY NOTATION
Available from National Technical Information Service, 5285 Port Royal Road, Springfield, VA 22161.

17. COSATI CODES			18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
FIELD	GROUP	SUB-GROUP	Depth; Dredging; Hydrographic; Range; Surveying

19. ABSTRACT (Continue on reverse if necessary and identify by block number)

The US Army Engineer Waterways Experiment Station has developed a hydrographic survey package. The package, which is of a general nature, includes software that can support a variety of commercially available hardware. Two types of equipment configurations are available with the system: the first configuration places most of the equipment on the shore, whereas the second configuration places most of the equipment on the boat.

The most important part of the development is a complete software package intended to answer most of the needs of US Army Corps of Engineer Districts for small-boat surveys. The package includes presurvey line generation and equipment definition programs, a versatile survey program, and several postsurvey programs for plan-view and cross-section plotting, dredging computations, tide removal, and graphical screen editing. All software is form-driven and is written in FORTRAN.

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT ☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT. ☐ DTIC USERS	21. ABSTRACT SECURITY CLASSIFICATION Unclassified
22a. NAME OF RESPONSIBLE INDIVIDUAL	22b. TELEPHONE (Include Area Code) | 22c. OFFICE SYMBOL

DD Form 1473, JUN 86 — Previous editions are obsolete.

SECURITY CLASSIFICATION OF THIS PAGE
Unclassified

6a and 6b. NAME AND OFFICE SYMBOL OF PERFORMING ORGANIZATION (Continued).

Instrumentation Services Division, CEWES-JO-R

and

Hydraulics Laboratory, CEWES-HS-H

10. WORK UNIT ACCESSION NO. (Continued).

Funding provided by Surveying and Mapping Research and Development Program, Work Unit 31099, sponsored by the Headquarters, US Army Corps of Engineers.

PREFACE

This investigation was performed by personnel of the US Army Engineer Waterways Experiment Station (WES) under the Surveying and Mapping Research and Development Program sponsored by the Headquarters, US Army Corps of Engineers (HQUSACE), under Work Unit No. 31099, "Integrated Hydrographic Surveying Systems."

The study was conducted under the general supervision of Messrs. George P. Bonner, Chief, Instrumentation Services Division; Leiland M. Duke, Chief, Operations Branch; and James L. Ferguson, Chief, Data Reduction and Digital Support Section; and Frank A. Herrmann, Jr., Chief, Hydraulics Laboratory; R. A. Sager, Assistant Chief, Hydraulics Laboratory; Glenn A. Pickering, Chief, Hydraulic Structures Division; and Dr. Bobby J. Brown, Chief, Hydraulic Analysis Branch. This report was written by Messrs. Barry W. McCleave, Data Reduction and Digital Support Section, and E. Dale Hart, Hydraulic Analysis Branch. Mr. M. K. Miles was HQUSACE Technical Monitor. This report was edited by Mrs. Marsha C. Gay, Information Technology Laboratory, WES.

Acting Commander and Director of WES during preparation of this report was LTC Jack R. Stephens, EN. Technical Director was Dr. Robert W. Whalin.

CONTENTS

CONVERSION FACTORS, NON-SI TO SI (METRIC) UNITS OF MEASUREMENT

Non-SI units of measurement used in this report can be converted to SI (metric) units as follows.

Multiply	By	To Obtain
degrees (angle)	0.01745329	radians
feet	0.3048	metres
inches	2.54	centimetres
miles (US statute)	1.609347	kilometres
pounds (mass)	0.4535924	kilograms

SMALL-BOAT SURVEY SYSTEMS

PART I: INTRODUCTION

Background

1. Hydrographic surveys* are a vital part of the US Army Corps of Engineers work effort in maintaining and improving this Nation's waterways. Survey information is needed for dredge operations and payment, new project design, reservoir monitoring, and many other purposes. In all cases, the survey information must go through a sequence of steps from data collection to a final form that is typically a chart of the waterway, showing the depths as a function of longitudinal positions on the chart. Handling and processing the large volume of information collected during hydrographic surveys is a very time-consuming job, if performed manually. This was the standard procedure prior to 1969. In that year, the Headquarters, US Army Corps of Engineers (HQUSACE) directed the US Army Engineer Waterways Experiment Station (WES) to pursue research aimed at automating the hydrographic survey work of the Corps. Part of the WES research effort was to improve communications between Districts already using electronic surveying tools and those using only manual methods. Another part of the WES effort was to improve communication between survey equipment suppliers and District survey personnel. The electronic positioning and data processing equipment available in 1970 was more efficient than existing manual methods. The equipment was successfully applied by many Corps Districts with considerable savings in costs and time. A significant limitation, however, was flexibility as the electronic equipment available in 1970 was large and heavy, and consumed considerable electrical power. This limitation restricted the use of electronic surveying equipment to relatively large survey boats.

2. Corps Districts must survey in a wide variety of waterways, and many of these are impractical to survey using large survey boats. Some isolated waterways must be reached overland by trailered survey boats. Trailering puts an upper limit of approximately 27 ft** on survey boat size, and even this

* Unusual terms are defined in the Glossary (Appendix E).

** A table of factors for converting non-SI units of measurement to SI (metric) units is found on page 3.

size is difficult to handle. A considerably smaller boat is more easily transported. Another reason that small survey boats are necessary is that they are needed to work in small channels where larger boats cannot maneuver and where the water is too shallow for deeper draft boats. This need to use small survey boats conflicted with the need to automate the Corps survey process due to the size of the existing (1970) commercial electronic surveying equipment.

3. In response to Corps District requests for small-boat electronic survey equipment, WES initiated two different equipment development projects. These were for data loggers that recorded survey data on magnetic tape cassettes and permitted the data to be processed by the District automatic data processing center. Results of District use with these small-boat survey data loggers are given for the first project by LaFountain (1976) and for the second project by Thrower (1978).

4. Use of survey data loggers produced a significant savings in data processing time in the Districts, but they did not provide pilot guidance, a capability available on the larger survey boats that had a minicomputer aboard. There also remained a costly lag in time between data collection and the finished chart.

5. Use of data communication links to transfer data from a survey skiff to a larger survey boat was reported (Boone 1980). This paper gave an excellent description of some of the possibilities related to separating the data collection units and the data processing units of a system, but did not discuss communication links to maintain real-time processing speed.

6. By 1983, the availability of smaller and more rugged computers changed the picture of what possibilities were available for developing small-boat survey systems with capabilities almost equal to those possible on a large survey boat. From discussions with District personnel, it became clear that the need for an improved small-boat surveying system was important to many Districts. With HQUSACE backing, WES proposed that a special Corps meeting be held to discuss small-boat survey system development. This meeting was held at the Norfolk District in December 1983. Representatives from 11 Districts attended, which was a good indication of Corps-wide interest in small-boat survey systems.

7. At this meeting, WES personnel presented means by which they felt a practical small-boat survey system could be developed. The WES appraisal of

commercially available components indicated the following:

a. A recently available portable computer by Grid Systems Corporation should be capable of performing on-line in real-time the calculations necessary for pilot guidance, data recording, and data processing.

b. The Grid Systems computer could be programmed in a high-level language, FORTRAN, so that the survey programs, when developed, could be modified by Corps personnel as needed.

c. By using modern communication technology the electronic components of a survey system could be split into groups for optimum use of available boat and land vehicle space. For very small boats the system could be configured so that only a depth sounder and pilot guidance indicator need be aboard the boat and the remainder of the electronic equipment would be in a van or truck on the shore (Figure 1). For larger, but still trailerable, boats, the system could be configured so that most of the equipment would be on the boat and the amount of equipment on the shore would be minimized (Figure 2). In either case, the size, weight, and power consumption of the components would be kept to the smallest available at the time.

8. District response at the 1983 meeting at Norfolk unanimously confirmed the need for a small-boat survey system. Response from the Districts as to the best configuration to use was not clear, for the concepts presented were new and needed time to be understood. There was no question about District need for a small-boat survey system and the HQUSACE representative

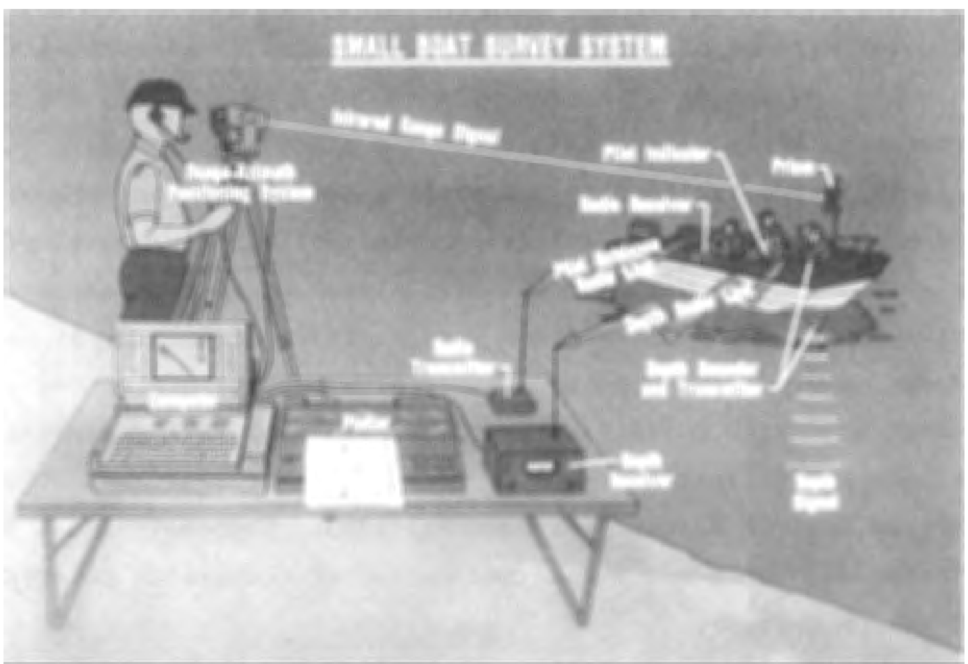


Figure 1. Proposed hardware configuration

Figure 2. Computer hardware with pilot indicator

directed WES to give a high priority to the development of a small-boat survey system.

Purpose

9. The purpose of this study was to develop the software and hardware components that could be combined into a system that would satisfy the requirements for performing Corps hydrographic surveys using a small boat as the survey vehicle. It was recognized that software development would be the primary Corps effort and that the system hardware would be assembled with commercially available components.

Scope

10. This report describes the hardware and software developed by WES for small-boat surveys. Example systems in use by Corps Districts are discussed to assist the reader in determining which combination of components would be most suitable. Each developed software package is discussed in detail. In addition, software packages developed to meet specific District needs are discussed.

PART II: DEVELOPMENT AND PROCUREMENT OF SYSTEM

11. At the meeting of December 1983 at the Norfolk District, the basic system hardware configuration was proposed and adopted. The hardware consisted of a Grid Systems computer for processing, a theodolite-type range-azimuth positioning system, a radio-linked depth system, a printer, a small plotter, a pilot guidance display, a disk drive, and a ruggedized bubble cartridge drive (Figure 1). The development was approved by those in attendance with a decision to proceed with procuring hardware.

Computer

12. The Grid Systems Compass* computer (Figure 2) was selected based on ruggedness and ability to perform real-time, multitasking, closed-loop process control. Since the system was considered Automatic Data Processing (ADP), the requisition and supporting ADP documents were submitted for ADP approval in January 1984. ADP approval was received and the rewritten requisition was submitted to the WES Contracting Division in August 1984; the system was received and a 1-week vendor course on the system was attended in November. The system was demonstrated numerous times including the Corps Surveying Conference at Jacksonville, FL, February 1985, and Savannah, GA, February 1988 (McCleave 1985, 1988).

Positioning System

13. A competitive procurement action for the range-azimuth positioning system (Appendix A discusses the positioning systems) was begun with the submission of a requisition in May 1984. The positioning system was delivered in December 1985. There were some major problems with starting up the positioning system (it required a particular string of characters from the computer) that were not resolved until a few days before its debut at the Corps Surveying Conference in Jacksonville, FL, February 1985 (McCleave 1985). The positioning system was found to have three major shortcomings for hydrographic

---

* At the time testing was conducted, Compass was a registered trademark of Grid Systems Corporation.

work: (a) it had a hardware range break at 1 km, (b) the precise time at which the range measurement was taken was not actually known, and (c) it was impossible for an operator to follow a moving boat and keep it "locked on" for more than a few seconds at a time.

14. The vendor chose not to address these problems, so WES devised a scheme to add a joystick servo-drive control to the device to aid an operator in smoothly following a moving target. This modification was awarded on a low-bid basis in July 1986. Even with the joystick, it was hard for an operator to stay locked on a moving boat for more than 30 sec at a time because the positioning system was too sensitive (having to be exactly on the prism contінously). Therefore, the range-azimuth system procured was determined to be unacceptable. At this time a search of available alternative range-azimuth theodolite-type systems was made.

15. Another vendor, International Measurement and Control (IMC), became very interested in the WES development effort in September 1987. The vendor loaned WES a Hydro 1* unit (Figure 3) for several weeks for testing. The unit was found to work well in all respects but two: (a) vertical tracking was awkward, and (b) the length of time of the range measurement was known only to within 0.8 sec. The vendor voluntarily corrected both problems by (a) adding a vertical movement lever that worked well with the horizontal crank, and (b) rewriting the internal firmware to allow input of a synchronizing character from the computer and returning a data age time allowing the sample time to be known within 0.1 sec. The vendor loaned WES a revised unit to demonstrate the small-boat system at the Corps Surveying Conference in Savannah, GA, in February 1988 (McCleave 1988). An operator could easily follow a boat with the unit, and no shortcomings were noted.

Depth System

16. There was no known commercially available depth system that would allow acquiring depths on a boat and radio-linking them to an IEEE488 (Institute of Electrical and Electronics Engineers (IEEE) 1978) interface on shore. Therefore, a requisition to design one was issued in February 1984, to obtain a system. The system was delivered in June 1984, and found to have

---

* Hydro 1 is a registered trademark of International Measurement and Control.

Figure 3. IMC range-azimuth positioning system

four problems, the worst of which was that the IEEE488 interface was inoperative. The IEEE488 interface was to be the communication link between the depth sounder and the computer; without it, verification of the depth sounder could not proceed. After numerous attempts to find the solution to the problem, including a 3-day trip to the factory, it appeared the vendor was going to be unable to resolve the problem. After waiting to be sure the vendor was not going to respond in a positive manner, a new competitive procurement was begun.

17. A contract to design a radio-linked, IEEE488-interfaced depth sounder was awarded to Innerspace Technology Corporation in July 1986. A completed unit was delivered in October of 1986 (Figure 4). The unit was field tested and no deficiencies were noted. The unit was demonstrated at the February 1988 Savannah conference (McCleave 1988); the vendor also had a unit at its conference booth, as the system was being marketed as a commercially available standard product.

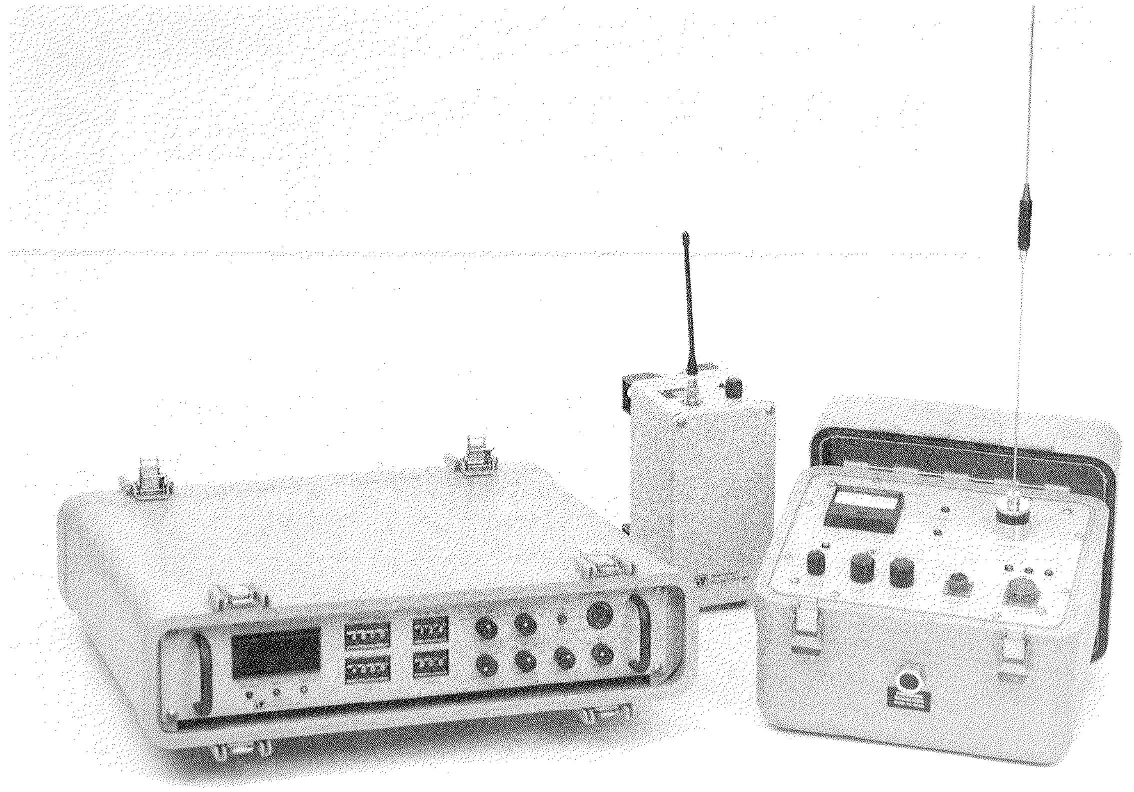


Figure 4. Innerspace Technology depth sounder system

Pilot Guidance Meter

18. In talking with Districts, it was determined that an analog style left/right pilot guidance indicator was the easiest to steer by. Since variable meter scale and distance along the line were also desirable, it was decided to procure a digital graphical display which could emulate an analog meter. This would require either an intelligent display or a display with a high-speed interface so that the computer could provide the graphics. Since one configuration would place the computer on shore and the display on the boat, an intelligent display with a modem interface was decided upon. Radio Shack* was known to market an economical, small, lightweight computer which met these criteria (Figure 5). Therefore, a requisition was issued in September 1984, and the unit was received a few weeks later. The coding was done by WES; the only problem noted was that the unit could operate only a few

---

* Radio Shack is a registered trademark of the Tandy Corporation.

Figure 5. Radio Shack pilot guidance computer

hours on internal batteries and required the external, battery-eliminator module. No problems were noted with the module. The unit was demonstrated at both the 1985 Jacksonville and the 1988 Savannah conferences (McCleave 1985 and 1988, respectively).

Ruggedized Storage

19. The computer procured was capable of taking 130 g's of vibration, but the disk drive was rated only at 3 g's; therefore, a more ruggedized storage media was desired for rough conditions. Any storage media which uses moving parts cannot sustain high shock levels; therefore, a removable cartridge bubble drive was procured. The requisition was issued to Targa Electronics in September 1984, and the drive was delivered in February 1985 (Figure 6). A vendor recall was issued and the unit was returned to the factory in March 1985 to correct a write-protect logic problem. As delivered, the unit was good only for data storage (using specialized software) and could not be used as a standard peripheral (with vendor programs). However, the vendor later developed Grid disk drive emulation firmware (which was installed at no cost

Figure 6. Computer system with Targa bubble memory unit, plotter, and printer

in October 1985) for the IEEE488 interface. This allowed the bubble drive to function as a standard storage device appearing to the computer as an external floppy (or hard) disk. The unit was loaned to the Louisville District in 1986 for field evaluation and found to be intermittent and unusable. The unit was returned to the factory and a revision to the write-protect logic was made. Since then, no problems have been noted; however, it has not been extensively tested. The unit was demonstrated at both the 1985 Jacksonville and the 1988 Savannah conferences (McCleave 1985 and 1988, respectively).

Plotter

20. A small plotter (8.5- by 11-in. paper) was procured with the computer system (Figure 6). However, it soon became apparent that developing a plan view program would require the ability to plot on larger paper. Between 17 January 1986 and 17 February 1986, a plotter was rented to aid in software development. However, there was a recurring need to add large plot enhancements to be able to demonstrate plotting and to help Districts trouble-shoot problems. Therefore, a low-cost D-size (22- by 34-in.) Hewlett-Packard Draftpro* plotter was requisitioned in August 1987 and delivered in September 1987

* Hewlett-Packard and Draftpro are registered trademarks of Hewlett-Packard Company.

(Figure 7). This plotter ran all existing software without modification and no problems have been noticed. It was demonstrated at the Savannah conference in February 1988 (McCleave 1988).

Figure 7. Hewlett-Packard Draftpro plotter

Printer

21. A small printer from Hewlett-Packard was requisitioned in August 1984 and delivered a month later (Figure 6). No problems have been noted with the printer. It was demonstrated at both the 1985 Jacksonville and the 1988 Savannah conferences (McCleave 1985 and 1988).

Radio Link

22. The additional radio link necessary for connecting the land-based computer with the ship-based pilot guidance display was originally ordered in

February 1985. The first supplier was unable to deliver the radio link as specified; and a new requisition was issued to Motorola, and the unit was delivered in August 1986 (Figure 8). The radio link was field tested on several occasions. In March 1987 during a demonstration of the system to HQUSACE and field personnel in Fort Belvoir, VA, an output transistor went out, rendering the unit nonoperational. However, since then no problems have been noted. The link was demonstrated at the 1988 Savannah conference (McCleave 1988).

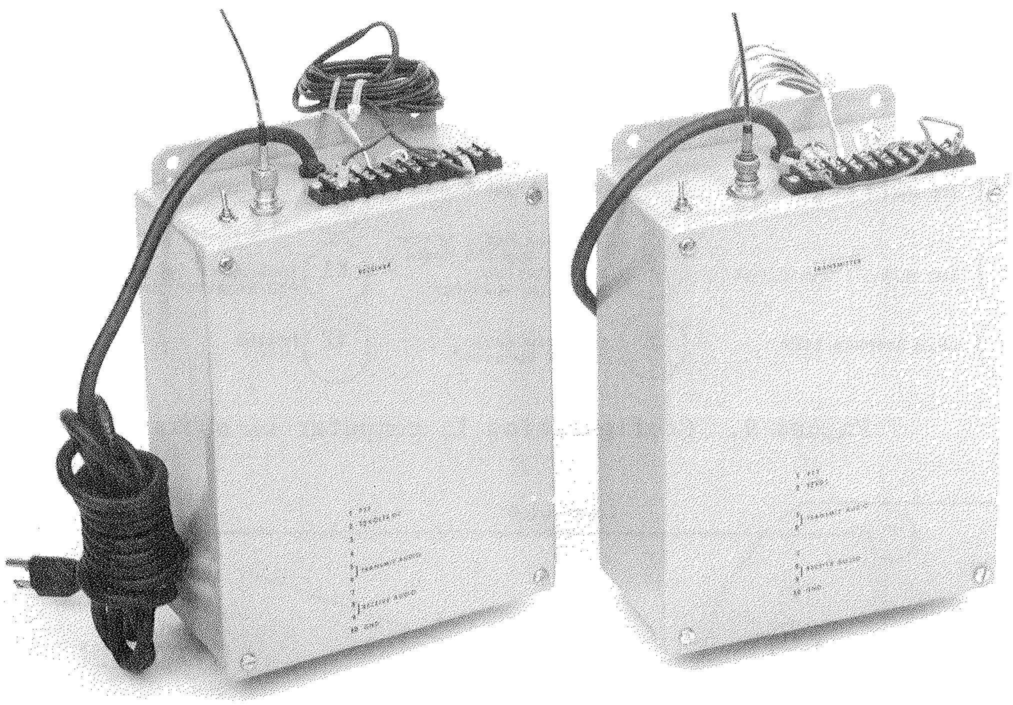


Figure 8. Motorola pilot guidance radio link

Software

23. Software is discussed in Part III.

Configurations

24. The system was designed to function in either of two hardware configurations: (a) configuration 1, computer on shore (Figure 9) or (b) configuration 2, computer on boat (Figure 10). Both configurations can use the same hardware with the only difference being method of connection. In the first configuration, radio links replace wires for communication of depth from

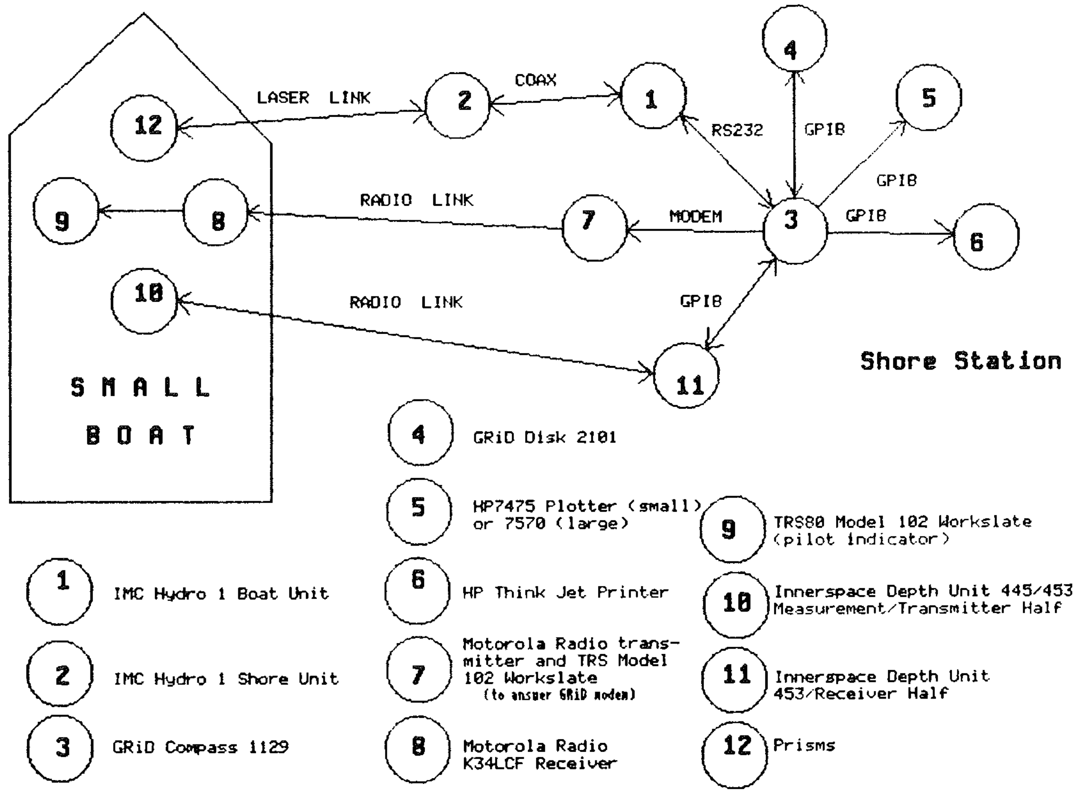


Figure 9. Configuration 1, computer on shore

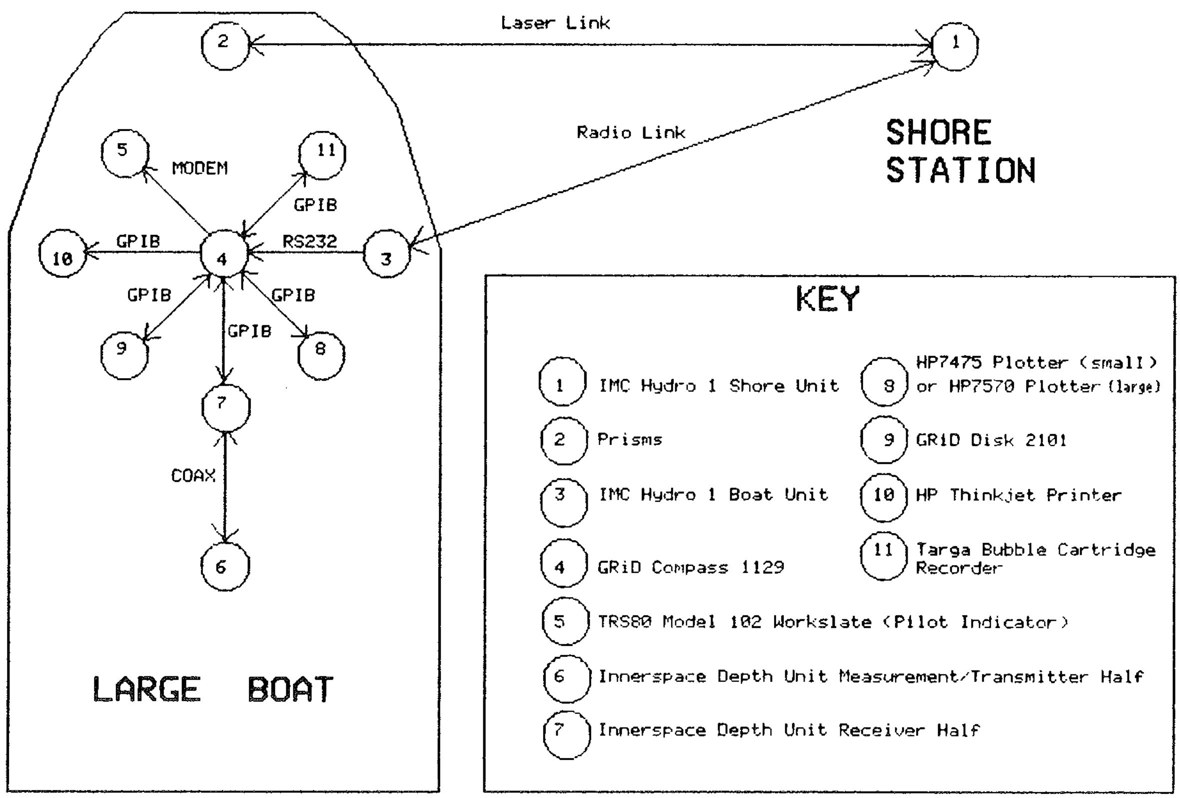


Figure 10. Configuration 2, computer on boat

boat to shore and pilot guidance from shore to boat. In the second configuration, a radio link replaces wires for communication of position from shore to boat. The same software works with either configuration, as the connecting links are transparent to the computer. The configuration diagrams are drawn for range-azimuth equipment. Range-range equipment normally places the computer on the boat as in the second configuration. Global satellite positioning systems would normally be similar to configuration 2. Any additional inputs, such as tide gages, would connect to the computer by IEEE488 with radio links used as needed.

25. The only District-tested systems have been using configuration 2, with the computer on the boat and using an Innerspace digitizer. The presently used field systems are shown as follows:

District	Field System	Registered Trademark of	Figure
Mobile	Del Norte	Del Norte Technology	11
	Autotape	Cubic Precision Corp.	12
Panama City	Mini-Ranger III	Motorola, Inc.	13
	Polarfix	Krupp Atlas	14
Louisville	Del Norte	Del Norte Technology	15
Huntington	Del Norte	Del Norte Technology	16

The WES-demonstrated system of the first configuration (Figure 17) used an IMC range-azimuth system and could function in either configuration. The minimum hardware required is listed in Table 1. Special cabling is diagrammed in Appendix B.

Capabilities

26. A small-boat system capable of performing dredging payment, reconnaissance, and buoy placement surveys was developed (Figure 18 shows a typical small boat and equipment). Ocean surveys and sweep system surveys were outside the scope of this development effort (however, the basic system could be modified to address these and other needs).

27. The hardware and software are modular so that positioning and depth equipment can be easily exchanged in the field (if software modules have been written for the particular hardware). To provide this capability, range and depth information come directly into the computer, not through some vendor box

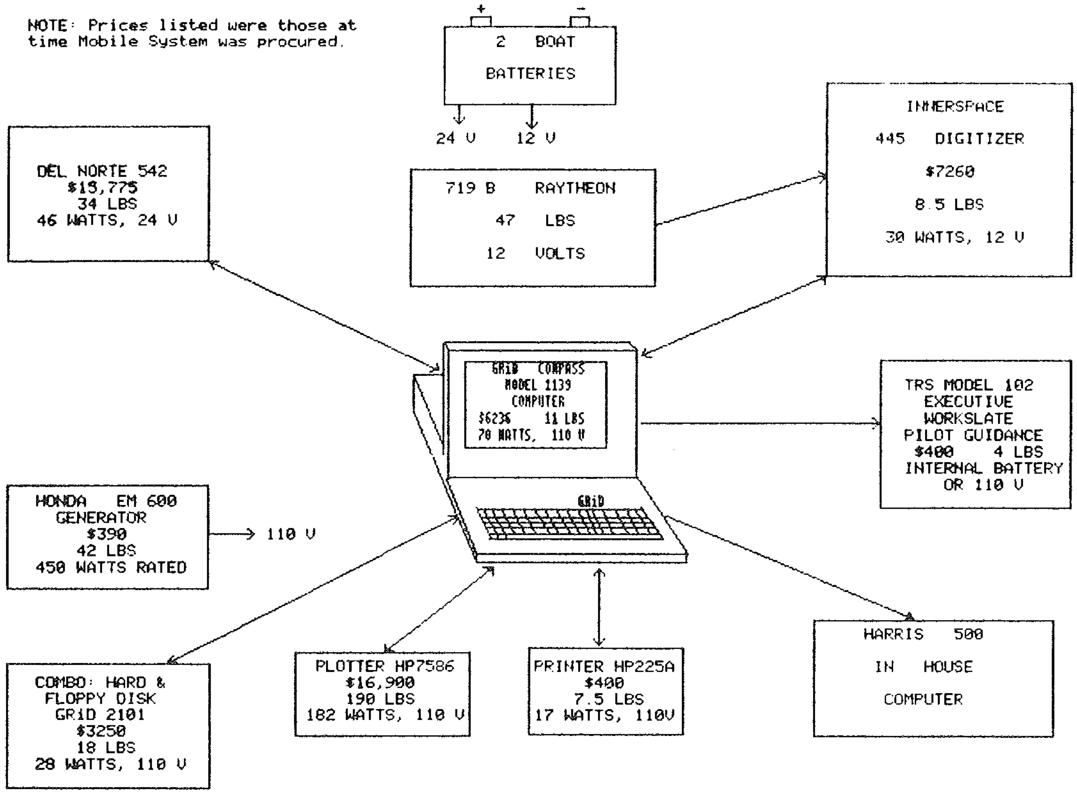


Figure 11. Mobile Del Norte

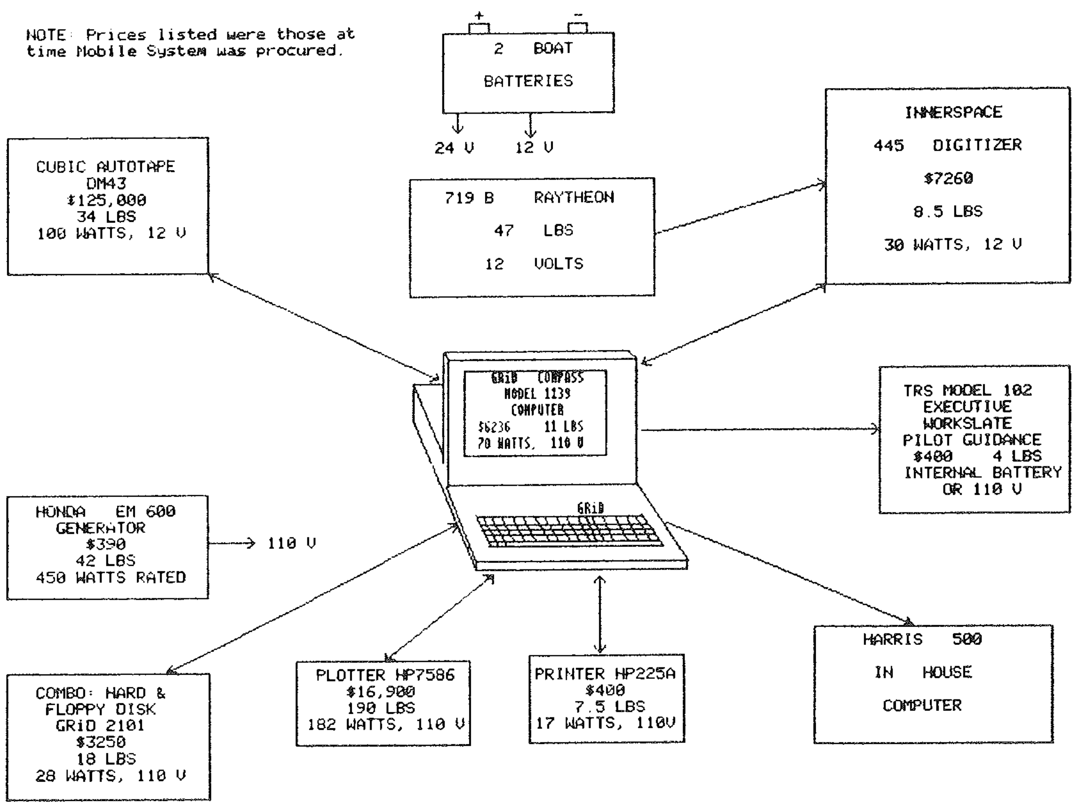


Figure 12. Mobile Cubic Autotape

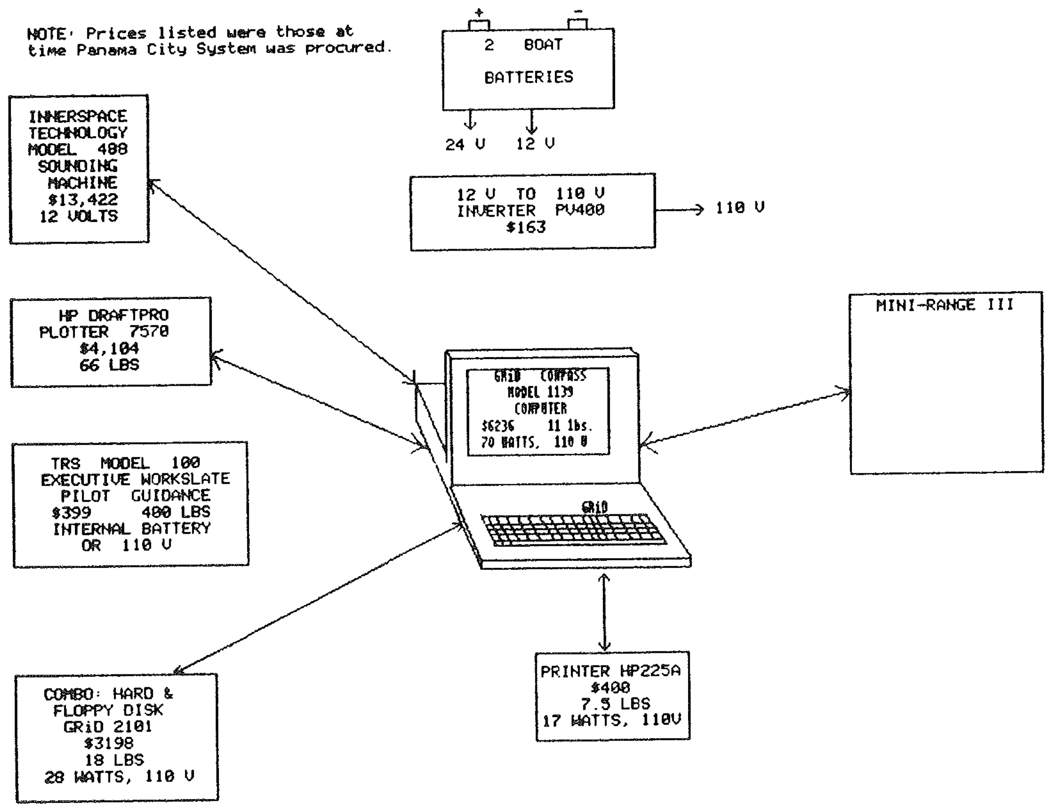


Figure 13. Panama City Mini-Ranger III

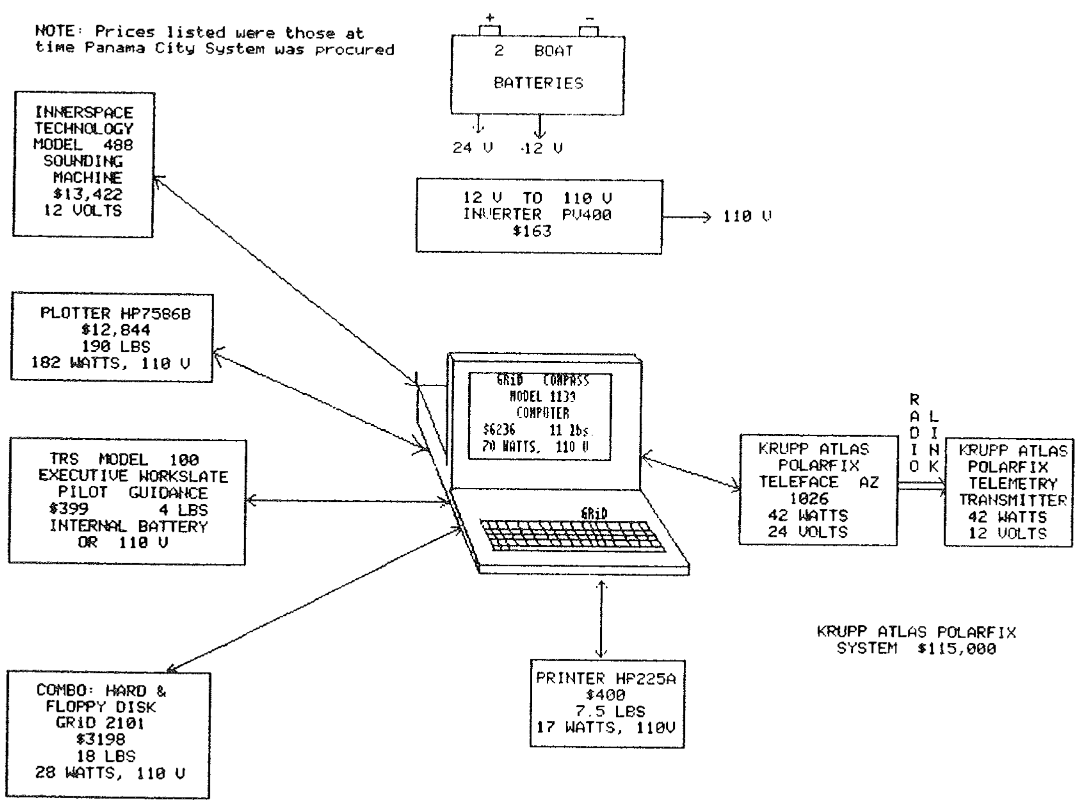


Figure 14. Panama City Krupp Atlas Polarfix

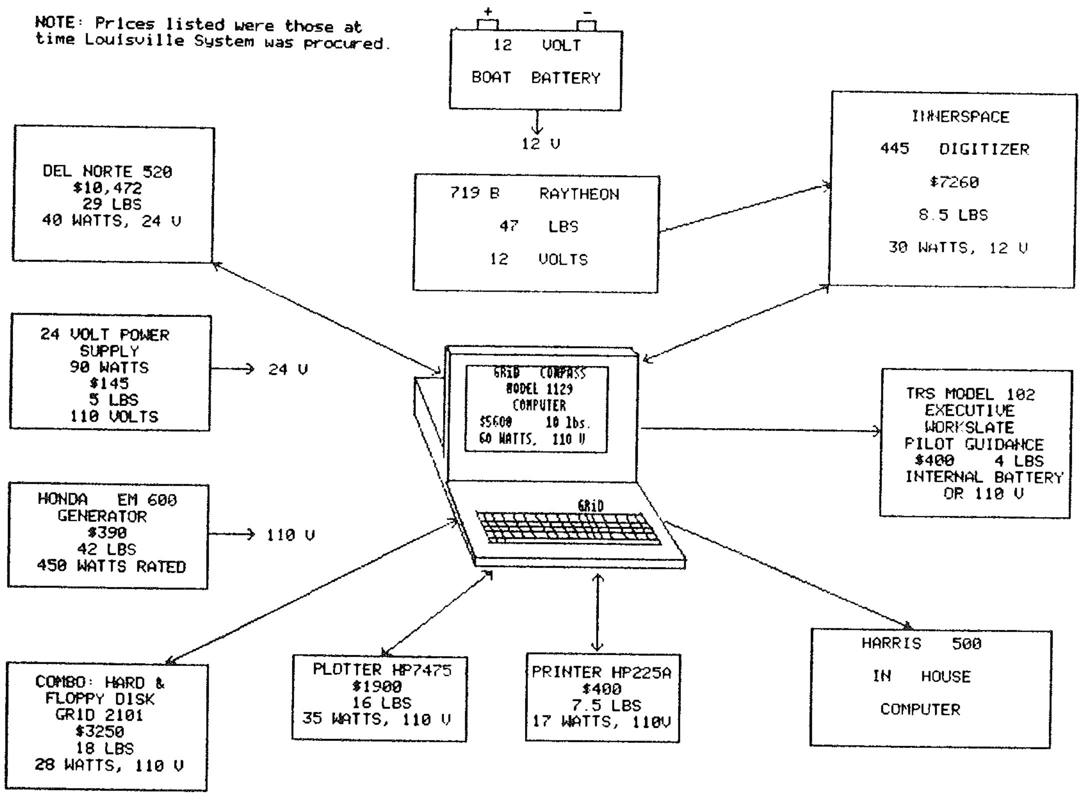


Figure 15. Louisville Del Norte

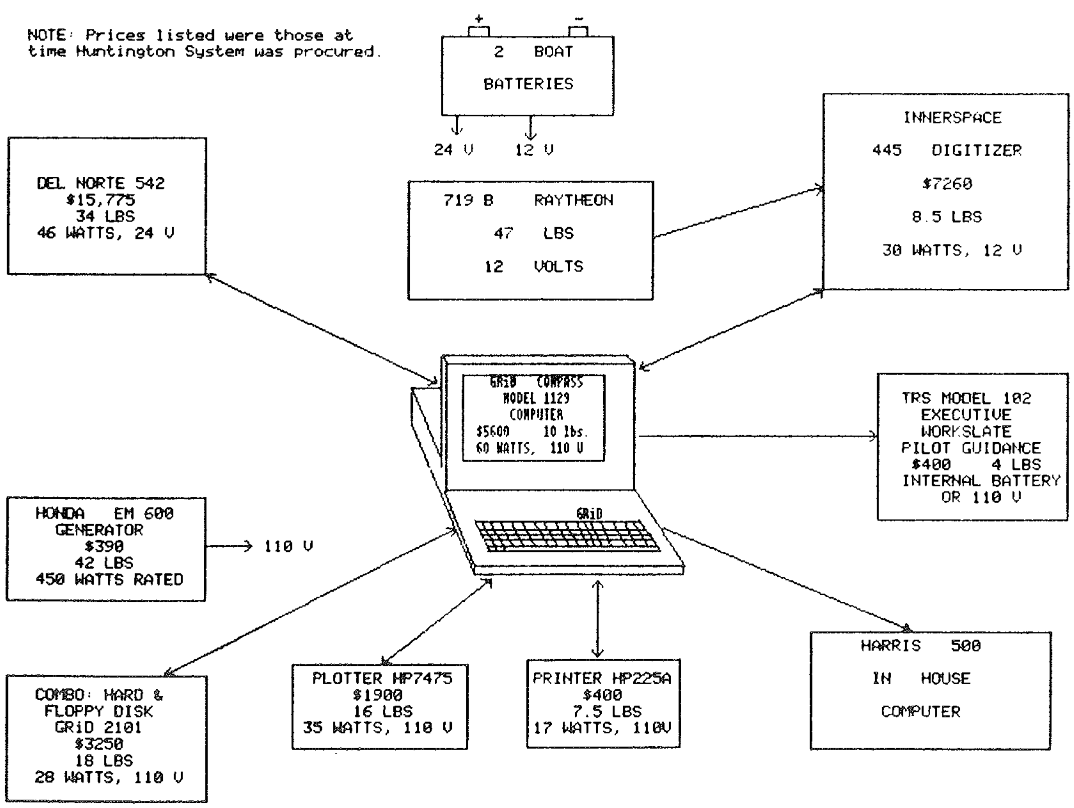


Figure 16. Huntington Del Norte

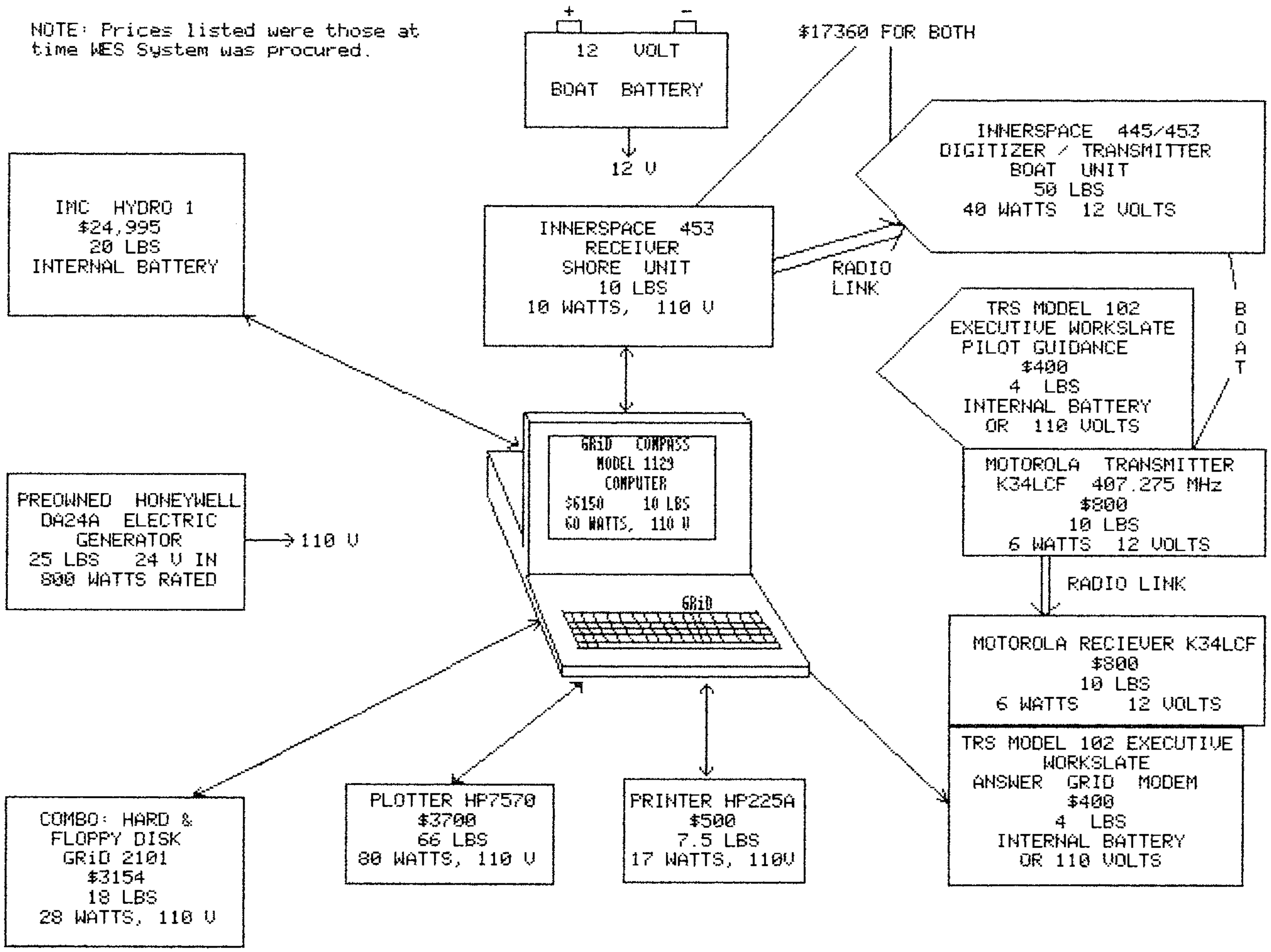


Figure 17. WES IMC

Figure 18. Panama City Area Office boat and equipment

that ties them together through a common interface. This requires that the positioning system have an RS232 (Electronic Industries Association 1969) interface and that the depth measuring system have an IEEE488 interface.

28. Four survey styles are allowed: (a) section line, (b) Louisville, (c) random, and (d) find spot (Figure 19). The survey styles are modular and new styles could be added basically by changing the operator interface software module. The survey styles are discussed in detail in Part III and Appendix C.

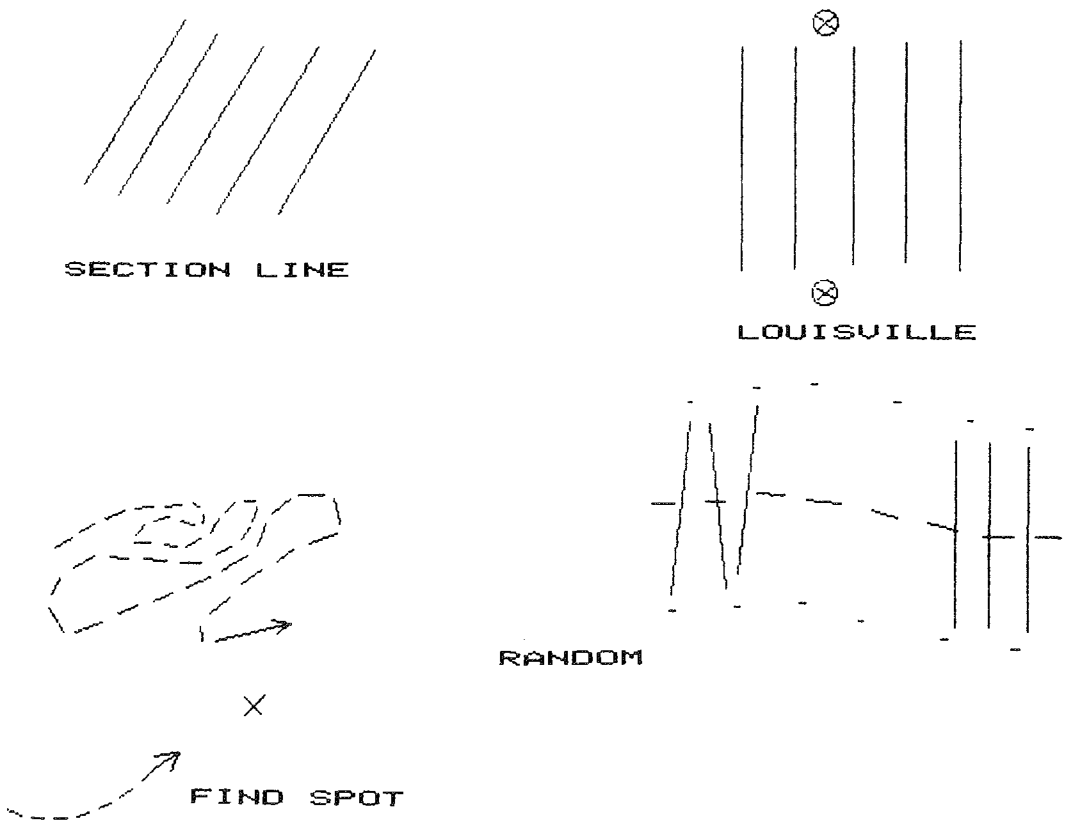


Figure 19. Survey styles

29. Plotting is supported both on-line and afterwards. On-line, the current boat position is shown by pen position at all times; depths are plotted when data saving is enabled. Afterwards three plot types may be produced: (a) plan view with features, labeling, and contours (Figure 20), (b) cross sections (Figure 21), and (c) three dimensionally stacked cross sections (Figure 22). The plot types are discussed in detail in Part III and Appendix C.

30. Printing is supported on-line and afterwards. On-line easting,

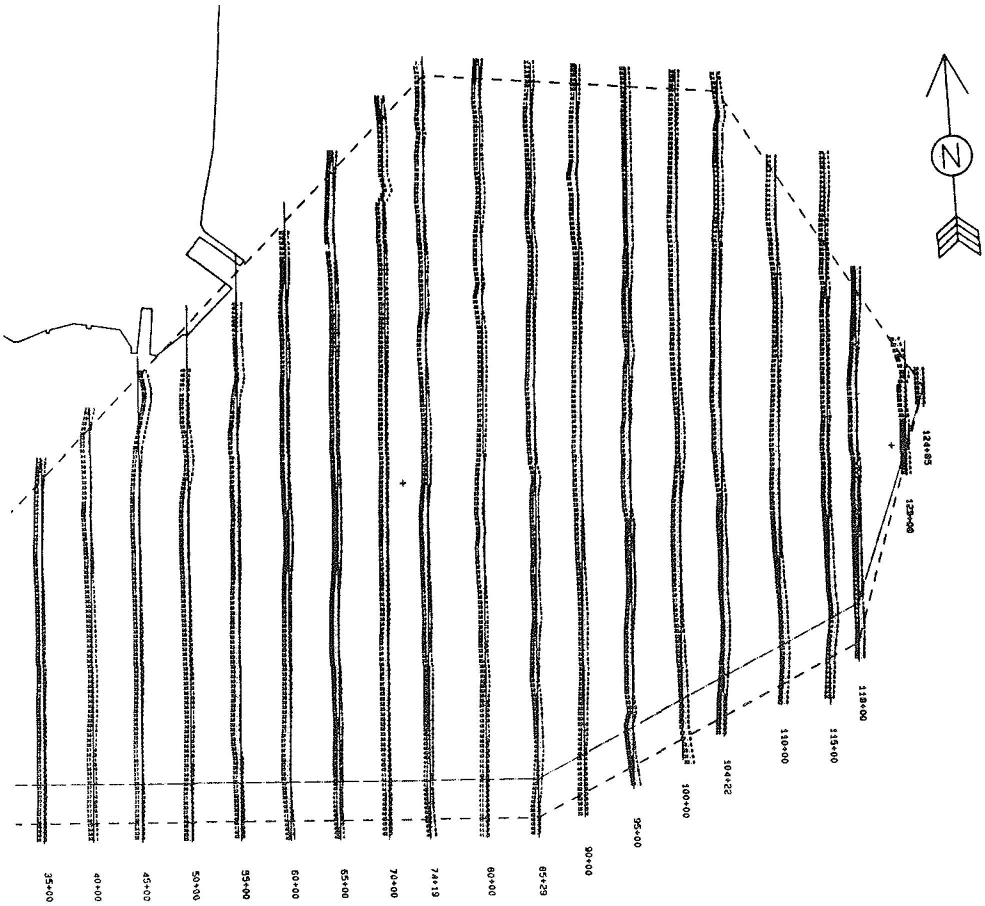


Figure 20. PLAN_VIEW_PLOTTER

northing, and depth may be printed at a predetermined interval. Afterwards, data and dredging quantities may be printed.

31. Since the survey program is written as a group of tasks (modules) which are basically independent, most hardware operations can take place at speeds close to the maximum possible. For example, since depth acquisition is in a separate module from position, depths may be acquired ten times per second while position is acquired only once per second. The data are recorded independently and time marked by a common clock in the computer. Position is then determined by piecewise linear interpolation at the times the depths were acquired (by postprocessing programs). The net effect is that position versus depth is available at 10 times per second for cross-section plots and dredging computations. While this "high-speed" computation is taking place, the operator may leisurely change the display scale, record tides, mark spots, etc., in the background (i.e., at a low priority).

32. The pilot (helmsman) has independent control of his display as

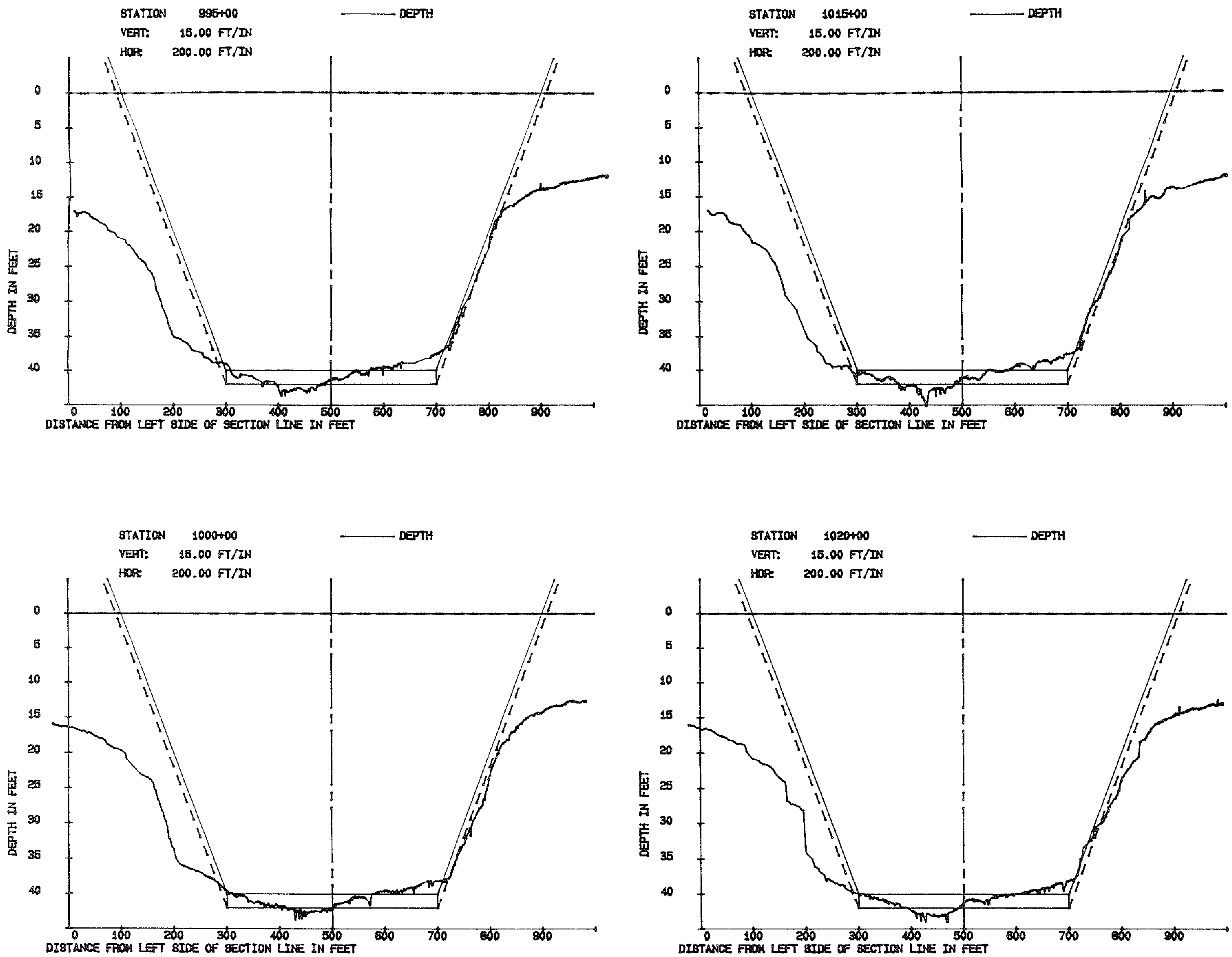


Figure 21. Cross sections for various stations

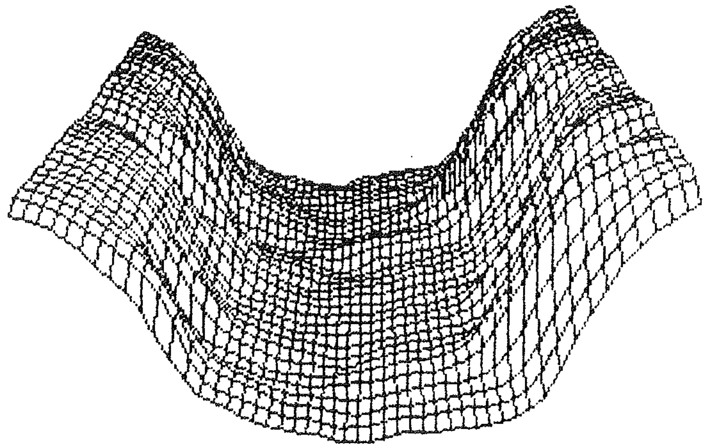


Figure 22. 3D_PLOT

graphics are generated and controlled within the pilot indicator computer (distributed processing). Update information is sent to his display about once every 0.8 sec, as it becomes available. Function keys are used for inputs of scale change and display form (auto scaling may be enabled if desired).

Evaluation

33. The software has been extensively field evaluated by several Districts, primarily Mobile. Problems were corrected as they were discovered and enhancements have been made. Since hydrographic surveying is an evolving field, no software or hardware can ever be considered complete. However, all codes and information generated are available to Districts (i.e., there is nothing proprietary in the package) and consultation by WES personnel is available on a cost per time basis if needed. In general, the system has been proven to be easy to use by surveyors (i.e., computer types are not required), but onsite training is recommended.

34. The Districts, thus far, have all opted for the computer-on-boat hardware configuration, and this configuration has been thoroughly tested. The equipment-on-shore method has not been tested as extensively. The only real difference between the two is radio links; therefore, the main concern which has not been addressed is how susceptible will the system be to local radio interference. However, dedicated-band, Government-only frequencies were selected; so this should not be a problem.

35. Some experimentation was done with prism arrays for targeting. One-inch and two-and-one-half-inch prisms were tested. The smaller prisms may be more tightly spaced but form a weaker target. The best configuration found was the one commonly provided by vendors: two rows of six prisms each in a circular pattern around a pole. The prism rows are offset 30 deg such that a prism occurs at each 30-deg interval around the pole. Reflectors around the prisms (provided by some vendors) are desirable. In addition to the prisms, a vertical foot of reflective tape is wrapped around the pole immediately below the prisms. The tape provides a "close in" target and the prisms provide a long-range target. This enables smooth operation from 100 ft to 2 miles separation between the theodolite and the boat.

PART III: SOFTWARE

36. A number of surveying programs have been developed. These include presurvey, survey, and postsurvey programs. The package, as it is now implemented, is designed to do zigzag surveys, random surveys, and find spot surveys. Cross-section surveys are extensively supported. Profile and random lines can presently be run and plotted in plan view (Figure 23). A program to fit data from profile and random surveys to section lines generated afterwards is planned but is not part of this development. Profile surveys are best run with sweep systems. This package does not presently support sweep systems.

37. Part III provides an overview of most programs in the package. A detail of each program is included in Appendix C. The information layout in memory and in disk files is given in Appendix D.

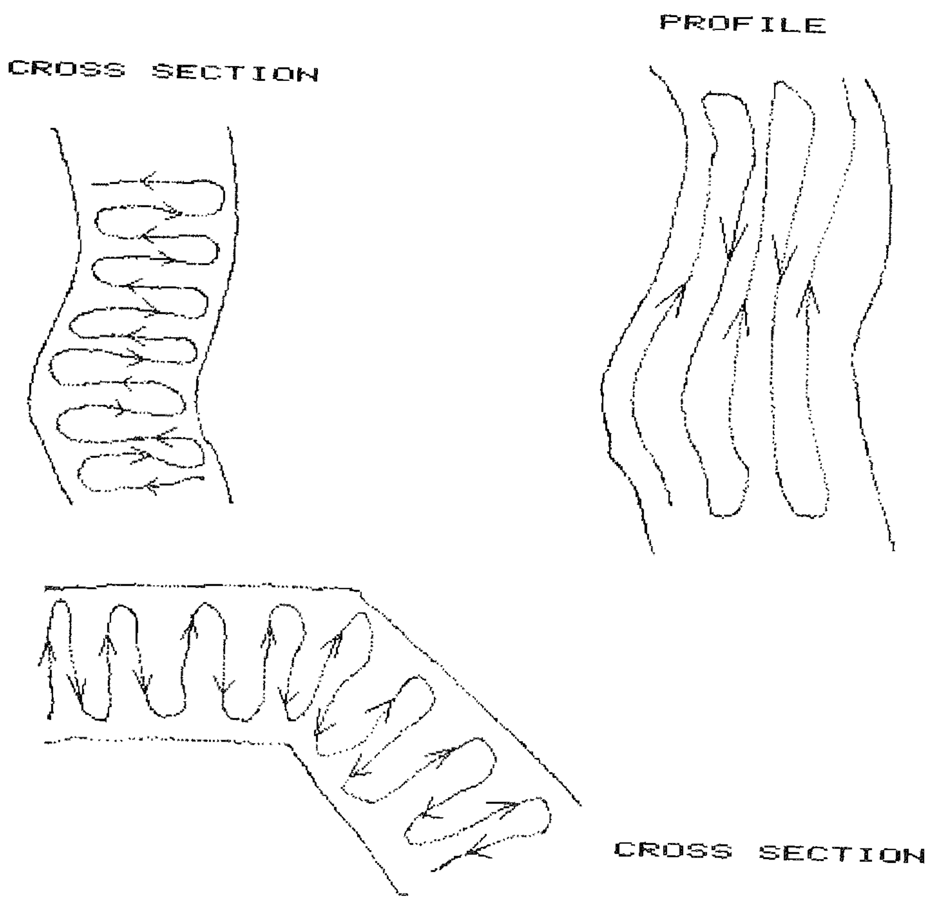


Figure 23. Survey patterns

Design Criteria

38. The design philosophy has been to acquire as much data as possible, as accurately as possible, while providing a user-friendly operator interface and frequent pilot guidance updates. These goals have largely been met.

39. To provide a user-friendly operator environment, the Grid Systems menu/form package has been incorporated in most of the programs. A form (for input of multiple items) is displayed on the screen with default or previous values shown. The operator then uses the cursor to move about and change selected items by either selecting from a list of choices at the desired form position or typing in a value for a particular item. When he is satisfied with the form's contents, he confirms it and the program continues.

40. To collect data at the highest possible rate and provide the pilot with rapid updates, modular, concurrent tasks were used. The survey multiprogram is composed of a set of independent tasks that interact with each other using a common block of memory through which data and semaphores are passed. One task is dedicated to acquiring position information and computing boat position, both absolute and relative to the desired survey line. Another task acquires depth information. A third task sends pilot guidance information to the pilot indicator. A fourth prints current data. A fifth plots current data. And still another program displays graphically the boat's position and allows operator inputs. Since all tasks can effectively take place concurrently and independently, each part does not have to wait for the others to be completed before it can start the next cycle (concurrent processing). Depths and positions can be saved at rates up to 10 times per second. Also, operator intervention such as changing display scales will not affect the data acquisition and pilot indicator update rates. To synchronize the data, an internal clock is read each time a position or depth is acquired and the time is saved along with the acquired data. Analysis programs then linearly interpolate the position readings to get positions at the time depths were acquired, providing position/depth information at 0.1-sec intervals. Multitasking like this operation on the Grid computer is not possible on MS-DOS* computers. This multitasking capability was an important factor in selecting the Grid computer for the small boat system.

---

* MS-DOS is a registered trademark of Microsoft, Inc.

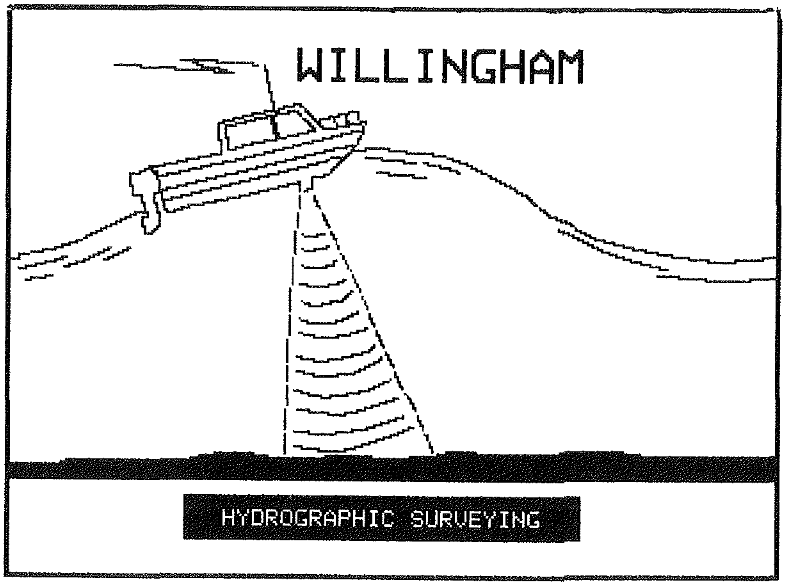


Figure 24. Power-up display

41. To add a personal touch to each survey vessel's system, a picture of a surveying boat (or a picture selected by the survey crew) with the particular vessel's name on the top is displayed on power-up (Figure 24).

High-Level Software

42. To allow Districts that have in-house programmers to better use the system, all programs are written in FORTRAN, and a plot package using industry standard plot commands (PLOT, AXIS, SYMBOL, etc.) is provided. Therefore, analysis and plot programs can be moved to the Grid computer with much less effort than would otherwise be required. This allows District-written software to be run in the field without need of transferring data to the main District computer. Data may be transferred via phone line to the District's main computer if desired.

Survey Types

43. The package presently supports four kinds of surveys: section lines, Louisville style, random, and find spot. Typical programs used in the surveys are shown in Figures 25-27. Section line surveys involve presurvey line generation and postsurvey processing in the office. This approach is used by offices that pay dredgers by quantity of material removed. Louisville style surveys require no office preparation as a final plot is produced during

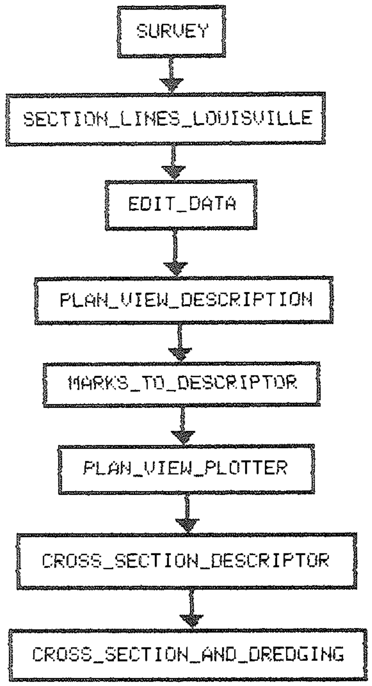


Figure 25. Louisville style survey typical program flow

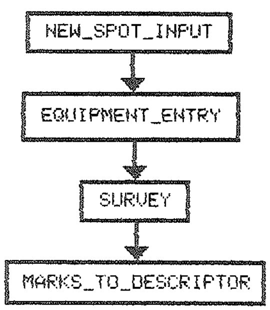


Figure 26. Find spot survey typical program flow

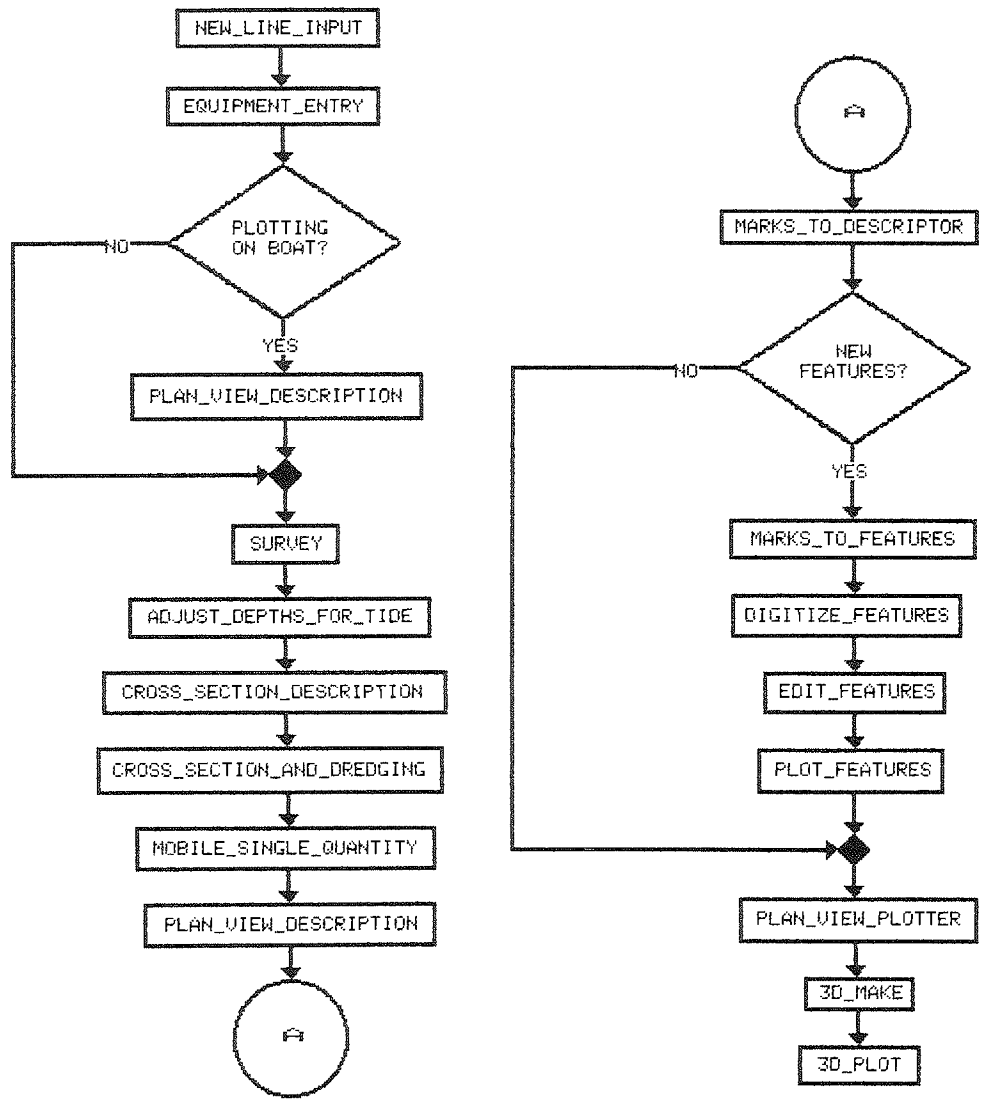


Figure 27. Random or section line survey typical program flow

the survey. This approach is used by offices that pay dredgers by the hour as there are no known control points. Random surveys are a superset of section line surveys in that not only may predefined channel and section line coordinates be used, but also lines may be generated from current boat position at the site. Random surveys are good for general-purpose use as they allow field determination of coordinates referenced from known control points. Find spot allows proceeding to a number of predetermined spots for buoy placement or the like.

Presurvey

44. There are two presurvey programs in the package for section line or random surveys. One accepts site information from the keyboard and produces a file containing lines to be surveyed. The other accepts equipment information from the keyboard. Since numerous files will be associated with each survey site, the files are named by a convention that uses the site name for a base and the type of information in the file for an extension. The operator supplies the site name and the programs supply and assume the extensions.

45. The NEW_LINE_INPUT is a form-driven program that accepts information from the keyboard for a new site or from a previously created site file. If a previously created file is to be modified, the operator enters the site name (such as ARLINGTON) into the first form displayed (Figure 28). The next form has five items of text and one menu item (Figure 29). This form is used

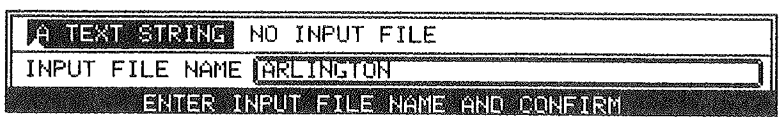


Figure 28. File form: NEW_LINE_INPUT

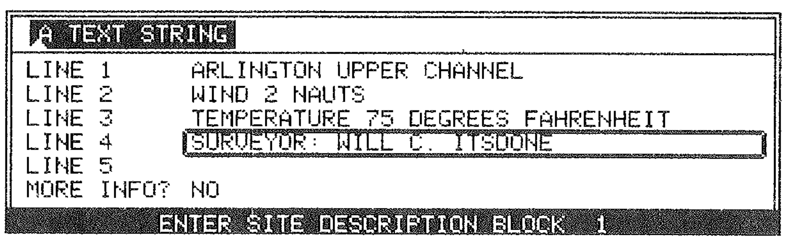
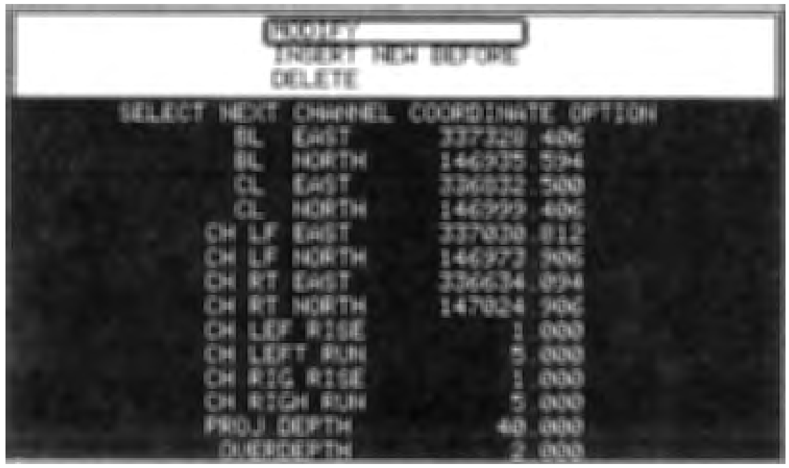


Figure 29. Site description: NEW_LINE_INPUT

to enter descriptive text about the site or survey conditions. The next section of the program allows entering/modifying (Figure 30) channel coordinate information (Figures 31 and 32). The channel coordinate form (Figure 33)

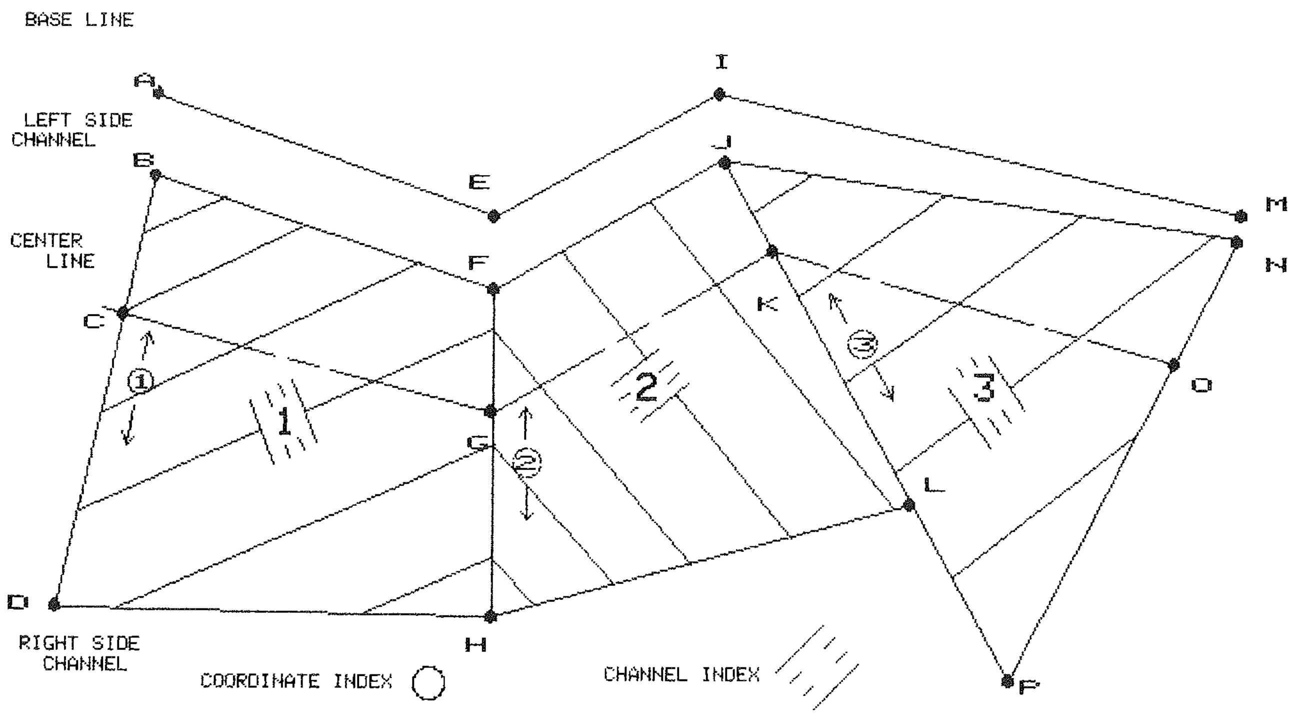


MODIFY
INSERT NEW BEFORE
DELETE

SELECT NEXT CHANNEL COORDINATE OPTION

BL	EAST	337320.406
BL	NORTH	146935.594
CL	EAST	336832.500
CL	NORTH	146999.406
CH LF	EAST	337030.812
CH LF	NORTH	146973.906
CH RT	EAST	336634.094
CH RT	NORTH	147024.906
CH LEF	RISE	1.000
CH LEFT	RUN	5.000
CH RIG	RISE	1.000
CH RIGH	RUN	5.000
PROJ	DEPTH	40.000
OVERDEPTH		2.000

Figure 30. Channel modify form: NEW_LINE_INPUT



COORDINATE INDEX	BL	LS	CL	RS	LEFT SLOPE	RIGHT SLOPE	PROJECT DEPTH
1	A	B	C	D	B ↗	D ↗	BCD
2	E	F	G	H	F ↗	H ↗	EGH
3	I	J	K	L	J ↗	L ↗	JKL
4	M	N	O	P	N ↗	M ↗	NOP

Figure 31. Method of defining channel

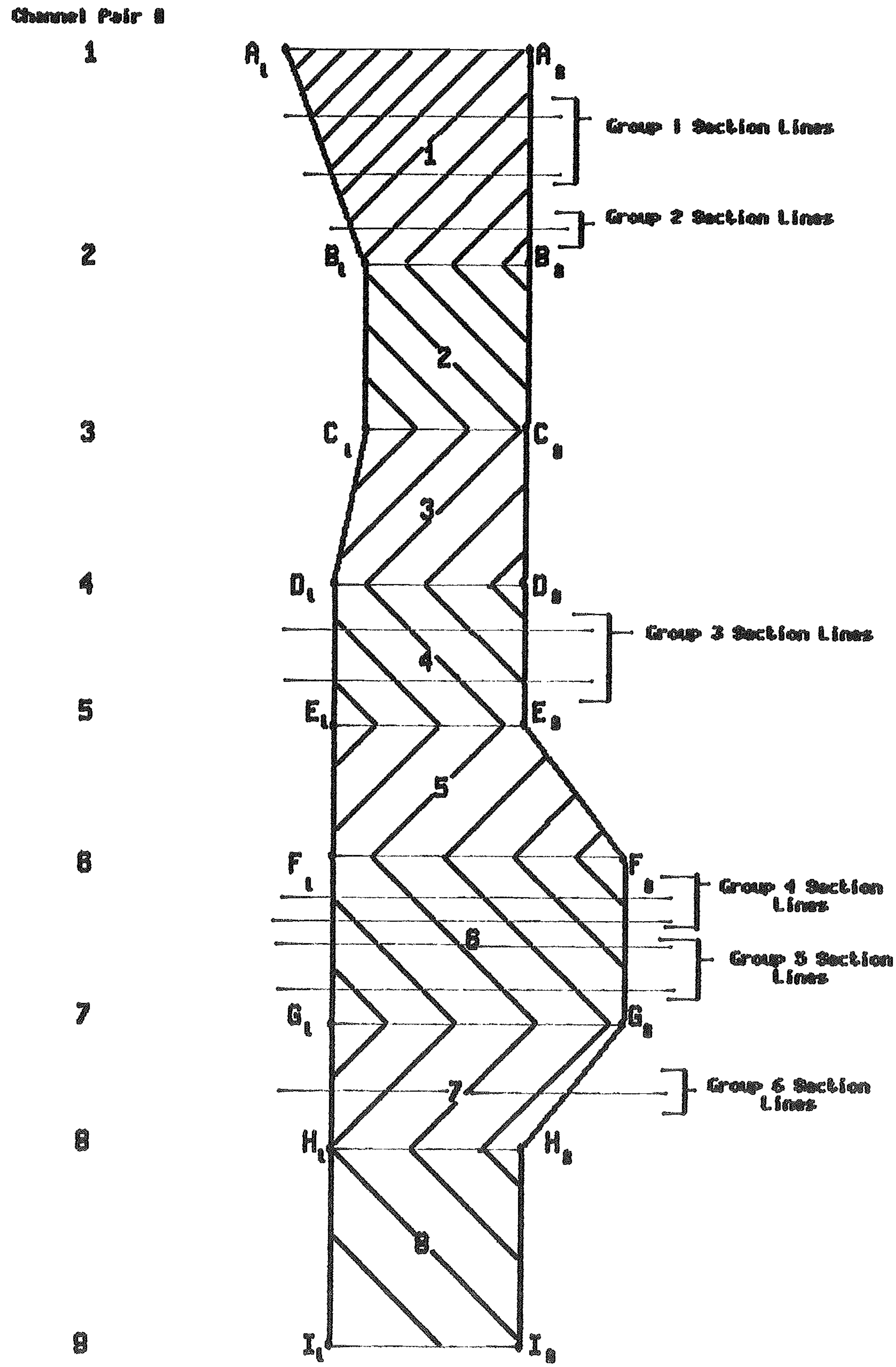


Figure 32. Channel sections (Continued)

NEW_LINE_INPUT

Channel index numbers when using rectangular or nonrectangular input methods are defined by relationship to the channel coordinate pairs.

Channel Coordinates

Assume a survey with channel coordinate pairs

1. A(L),A(R)
2. B(L),B(R)
3. C(L),C(R)
4. D(L),D(R)
5. E(L),E(R)
6. F(L),F(R)
7. G(L),G(R)
8. H(L),H(R)
9. I(L),I(R)

where A(L) is A(LE), A(LN)

A(R) is A(RE), A(RN)

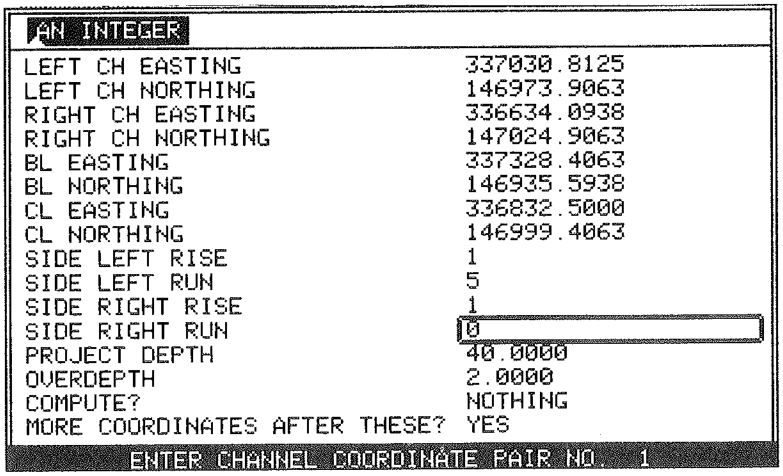
Channel indices are determined by an area of channel plan view with index number of first channel coordinate pair used and are shown as 1-8.

Section Lines

If section lines are superimposed as shown, the group index number should be as follows:

Section Group	No. of Lines	Channel Index No.
1	2	1
2	1	1
3	2	4
4	2	6
5	2	6
6	1	7

Figure 32. (Concluded)



AN INTEGER

LEFT CH EASTING	337030.8125
LEFT CH NORTHING	146973.9063
RIGHT CH EASTING	336634.0938
RIGHT CH NORTHING	147024.9063
BL EASTING	337328.4063
BL NORTHING	146935.5938
CL EASTING	336832.5000
CL NORTHING	146999.4063
SIDE LEFT RISE	1
SIDE LEFT RUN	5
SIDE RIGHT RISE	1
SIDE RIGHT RUN	0
PROJECT DEPTH	40.0000
OVERDEPTH	2.0000
COMPUTE?	NOTHING
MORE COORDINATES AFTER THESE?	YES

ENTER CHANNEL COORDINATE PAIR NO. 1

Figure 33. Coordinate form: NEW_LINE_INPUT

accepts left-side channel easting and northing, right-side channel easting and northing, baseline easting and northing, and center-line easting and northing in district units (feet).

46. Section lines may be defined by one of three methods (Figure 34):

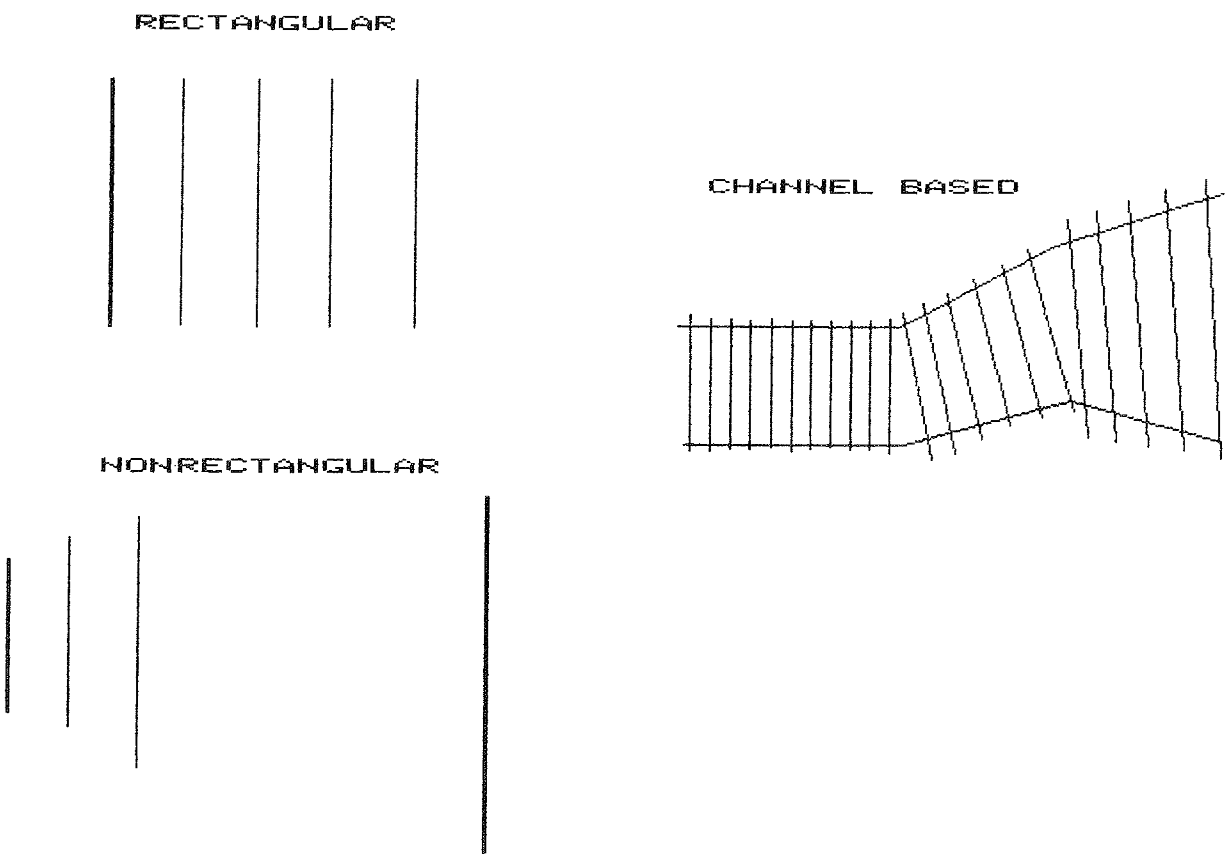


Figure 34. Section line definition methods

rectangular, nonrectangular, or channel based. Groups of parallel lines are created that are either rectangular, trapezoidal, or channel-shaped. The first form (Figure 35) has an item to select the input method and an item to select action on the next group. The rectangular input method (Figure 36) generates lines of fixed length parallel to the first at uniform offset. The nonrectangular input method generates linearly increasing or decreasing length parallel to the first at uniform offset. In addition to all form items required for the rectangular method, a second line (Figure 37) must be defined by start and end coordinates for use as a reference for determining the azimuths of propagation from the ends of the first line. The channel-based input method (Figure 38) uses the channel coordinates as a reference for section line generation. Once all groups are input, a graphical representation of the channel is displayed (Figure 39).

47. The EQUIPMENT_ENTRY program allows entry of transponder easting, northing, and height (Figure 40). The boat antenna height is also entered (Figure 41). An option of using or not using depth chart event marking is allowed. The type of range and depth equipment to be used must be selected. One equipment option is SIMULATION; this allows depth equipment to be exercised while ranges are simulated (vice versa) for diagnostic purposes.

48. The FIND_SPOT style survey requires running the EQUIPMENT_ENTRY program and the NEW_SPOT_INPUT program prior to beginning the survey. The Louisville style requires no presurvey programs.

49. The NEW_SPOT_INPUT program simply allows inputting a series of coordinate pairs (Figure 42) corresponding to the desired spots to be located. The output goes to a file with the site name and the extension ".SPOT."

50. Several programs have been developed to verify system operation. These programs are normally run when there is a problem with the system to determine which component is not functioning properly.

51. Prior to going to the survey site, a simulated run may be done in the office and the simulated data plotted. This is a good check of the survey coordinates and template. The simulated data may be plotted in plan view and cross section, making most bad entry values stand out. It is better to detect and correct input errors at the office rather than the site because there are less pressure and chance for error. This step is not essential as coordinates may be entered on the survey boat as it sails from the dock to the survey site, but is recommended whenever time allows.

```
RECTANGULAR  NONRECTANGULAR  CHANNEL BASED

SECTION METHOD           CHANNEL BASED
ACTION ON NEXT GROUP     MODIFY

     SELECT SECTION INPUT METHOD FOR GROUP  1
```

Figure 35. Section line input method form: NEW_LINE_INPUT

```
A REAL NUMBER
STATION START             965.0000
STATION INCREMENT         5.0000
EASTING START             337328.4063
NORTHING START            146935.5938
EASTING FINISH            336336.5938
NORTHING FINISH           147063.2031
SEPARATION                500.0000
DIRECTION                 LEFT
NUMBER OF LINES           31
CHANNEL INDEX             1
MORE GROUPS AFTER THIS?   NO
         ENTER ALL ITEMS AND CONFIRM
       LEFT OR RIGHT BASED ON STANDING ON
         LEFT BANK FACING THE RIGHT BANK
```

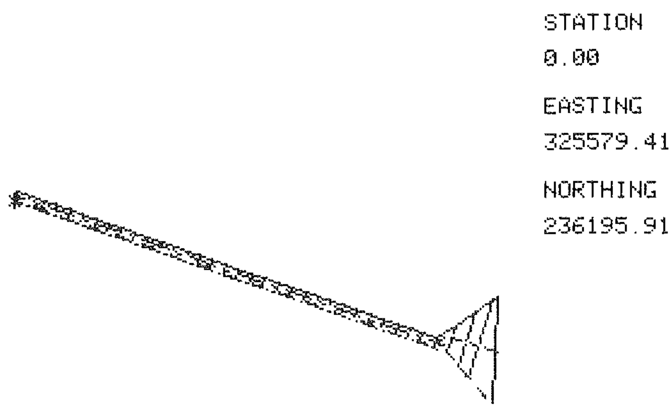
Figure 36. Rectangular input method: NEW_LINE_INPUT

```
A REAL NUMBER
EASTING START     331813.3125
NORTHING START    234010.9063
EASTING FINISH    331945.5938
NORTHING FINISH   234388.4063
       ENTER ANY OTHER LINE IN SECTION
```

Figure 37. Nonrectangular input method: NEW_LINE_INPUT

```
A REAL NUMBER
STATION START                           0.0000
STATION INCREMENT                       2.0000
EXTENSION BEYOND CHANNEL LEFT SIDE      0.0000
EXTENSION BEYOND CHANNEL RIGHT SIDE     0.0000
SEPARATION                              200.0000
STARTING CHANNEL COORDINATE INDEX       0
CHANNEL END LINES TO INCLUDE            BOTH
MORE GROUPS AFTER THIS?                 NO
          ENTER ALL ITEMS AND CONFIRM
```

Figure 38. Channel-based input method: NEW_LINE_INPUT




```
A TEXT STRING   NO OUTPUT FILE
OUTPUT FILE NAME  ARLINGTON
PRINT             NOTHING
        ENTER OUTPUT FILE NAME AND CONFIRM
```

Figure 39. Generated lines: NEW_LINE_INPUT

```
A REAL NUMBER
EASTING                              33456.560
NORTHING                             21342.65
HEIGHT                               10.0000
MORE TRANSPONDERS AFTER THIS?        YES
      ENTER TRANSPONDER TRIORDINATES NO    1
```

Figure 40. Transponder triordinate form: EQUIPMENT_ENTRY

```
← SIMULATE  MINIRANGER  GEODIMETER140  CUBIC  POLARF:→
OUTPUT FILE NAME      ARLINGTON
RANGE EQUIPMENT       CUBIC
DEPTH EQUIPMENT       INNERSPACE 001
BOAT ANTENNA HEIGHT   13.0000
FIX MARKS             FEET
FIX MARK INTERVAL     10.0000
        FILL IN EQUIPMENT FORM AND CONFIRM
```

Figure 41. Equipment selection form: EQUIPMENT_ENTRY

```
*********************************************
CODE RETURN CONFIRMS A FORM
ESCAPE (NOT CODE ESCAPE) ALLOWS ESCAPING
```

```
YES NO
EASTING                          43876.000
NORTHING                         127664.00
MORE SPOTS TO FIND AFTER THIS? YES
      ENTER SPOT TO FIND COORDINATES NO.  1
```

Figure 42. NEW_SPOT_INPUT form

Survey

Section line survey

52. The survey program is actually eight independent tasks that pass information through a common block of memory and gain processor time on a prioritized basis. To initiate the eight, run the program SURVEY. SURVEY will bring the other seven tasks (CONFIGURE, DEPTH, RANGE, PILOT, PLOT_PLOTTER, PRINT, and PLOT_SCREEN) into memory and run them. Starting SURVEY spawns a temporary task called CONFIGURE that reads in the equipment and section line files and displays several short forms that the operator may modify from their default values. The first form displayed allows the operator to select the type of survey (predetermined section line, find spot, random, or Louisville style), whether or not ranges and depths are to be simulated or acquired from transponders, whether or not pilot guidance will be used, whether or not plotting will be used (plan view only at present), whether or not data will be printed during test (if so, time interval between printouts is entered), type of screen display (plan view, cross section, or both), and whether or not all data collected are to be stored (Figure 43). If not all data are to be saved, another form (Figure 44) is brought up in which the operator selects the method by which acquired data are to be decimated for storage.

53. The PLOT_SCREEN task accepts all run-time operator inputs and controls the screen display. The task automatically selects the first line and displays it on the screen (Figure 45). The top of the screen is north, and the beginning of line is further from the edge of the screen than the end. As

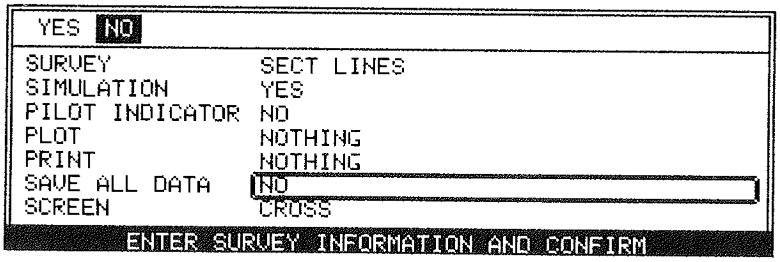


Figure 43. Survey option form: SURVEY (section line)

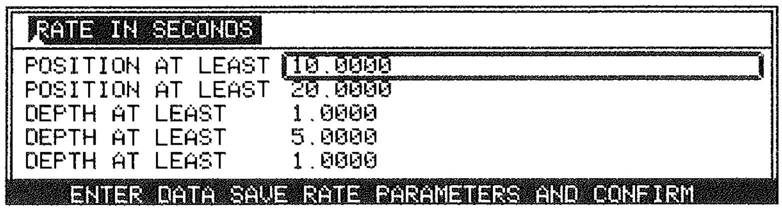


Figure 44. Data rate form: SURVEY (section line)

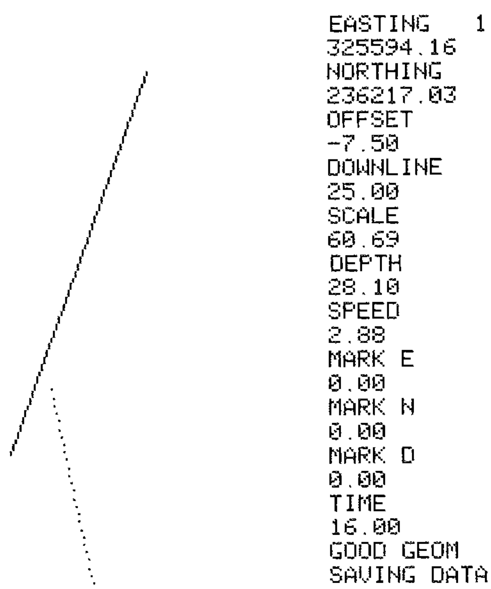


Figure 45. Survey display: SURVEY (Louisville)

each new boat position is read, a dot is displayed on the screen at the proper location. Several "live" values are displayed on the right edge of the screen. Messages are displayed to indicate the geometry of the positioning system and if data are currently being saved. A number of run-time changes may be initiated by the operator. To cause these changes to take place, the operator momentarily depresses a key. *A* aborts the run. *C* allows entering initialization information for Cubic positioning equipment if range signal is lost for extended period and Cubic must be restarted. *D* allows entering a depth gage tide adjustment. *E* allows manual ending of data saving for a line. *F* enables or disables range wild point filtering. *L* allows manually selecting the next survey line. *N* allows changing the next mark number from its next sequential value. *O* overrides automatic "start at beginning of line and finish at end of line" data saving (overriding may be toggled on and off by successively striking *O*). *P* causes data collection to pause until *P* is struck again. *Q* causes the task to quit immediately. *R* reverses the direction in which the current line is to be run. *T* allows new transponder locations to be selected from the transponder menu. *X* exchanges the screen scale.

54. Normally, as the beginning of line is passed, data saving starts and when the end of line is passed, it stops. Data are written to disk in a file with a four-digit extension which is the same as the line number. Then the next sequential line is displayed and the sequence continues. If the boat cannot manuever to the start of line, *S* may be entered to start data collection at the present, down-line position. If the boat cannot reach the end of line, *E* may be entered to terminate the data collection at the present position. *M* causes the current easting, northing, depth, and time to be saved. *B* causes a position offset by some entered distance and angle to be saved. Sometimes it is desirable to enter *R* and reverse the line to make boat steering easier due to currents or wind. If depth chart event marking was selected in the equipment file, event marks are generated.

Louisville style survey

55. Louisville style survey uses a scaled cartesian coordinate system which is not absolutely defined in terms of easting, northing, or rotation. Like the section line survey, the survey task is actually eight independent tasks that pass information through a common block of memory and gain processor time on a prioritized basis. The first form displayed allows the operator to select the type of survey--Louisville style (Figure 46). The task then

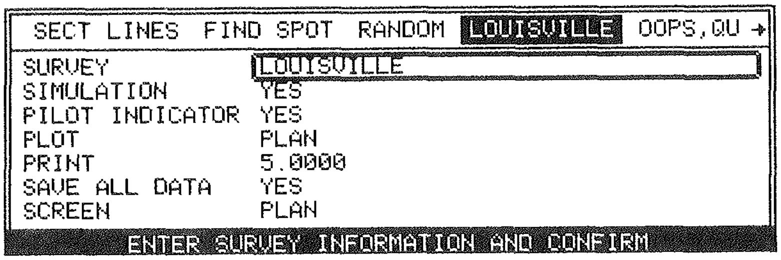


Figure 46. Survey option form: SURVEY (Louisville)

displays a Louisville style input form (Figure 47). Items entered include pool elevation (subtracted from depths during the run), bar identification number, transponder separation, and plot scale.

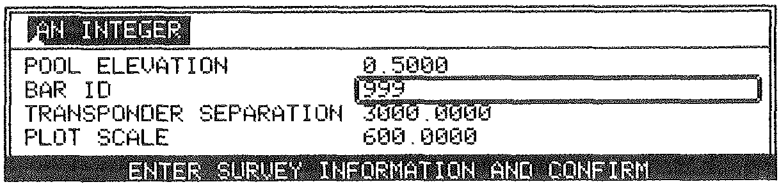


Figure 47. Louisville special form:
SURVEY (Louisville)

56. The PLOT_PLOTTER_LOUISVILLE task first asks the operator to position the plotter's pen at transponder *B*'s relative location and enter a carriage return. It puts the *B* transponder symbol at this point. It then asks the operator to position the pen at transponder *A*'s relative location and enter a carriage return. The task then uses the second operator-selected position to determine an azimuth from transponder *B* to transponder *A*. It then moves to transponder *A*'s plot position along this azimuth a distance equal to the plot relative transponder separation and plots the *A* transponder symbol. Next it generates an easting, northing set of transponder coordinates based on the plotter input transponder coordinates. The PLOT_SCREEN_LOUISVILLE task accepts all run-time operator inputs and controls the screen display. The task automatically creates a line based on present position and displays it on the screen (Figure 48).

57. A program called SECTION_LINES_LOUISVILLE will take the data and generate a pseudo .LINE file to make it resemble a section line survey. It also modifies the distance along line and distance off line in the data files to correspond to the pseudolines it created. This allows the postprocessing programs to be used with data collected in this manner.

Random survey

58. Random survey style is a superset of the section line survey.

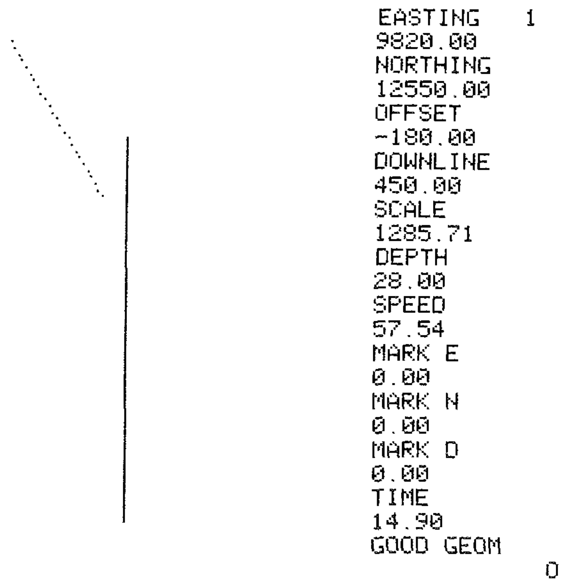


Figure 48. Survey display:
SURVEY (Louisville)

There are two major operational modes: (a) RUN and (b) CHANNEL.

59. RUN mode is the same operationally as the section line survey with the addition of three more key commands (*Y*, *Z*, and *Q*). *Y* and *Z* are used to mark a reference line for use in line generation by the rectangular method previously described. *Y* marks the start of line (left side) and *Z* marks the end of line (right side). *Q* is used to exit run mode and return to the main menu.

60. CHANNEL mode is used to establish new channel coordinates or change old ones. Coordinates are sampled at the current boat position. The coordinates are marked at channel bends in the same manner as in the NEW_LINE_INPUT program. Key commands are pertinent to the present channel coordinate index. *L* marks the left side of the channel; *R* marks the right. *C* marks the center line and *B* the baseline. *N* saves the present channel coordinate pairs and advances to the next index. *J* jumps back to a previous index. *E* ends an index and *Q* returns to the main menu. The main menu allows selecting CHANNEL or RUN mode. Also lines may be generated from the reference line or the most recent line run. When finished, *Q* is entered to end the program and update the .LINE file.

61. The modules are the same as for section line surveys with one change: the module RANDOM replaces PLOT_SCREEN.

Find spot survey

62. A find spot survey is used to locate a number of predetermined coordinate positions. It is similar to a section line survey except that a point rather than a line is displayed on the screen. A subset of the section line key commands is used (*A*, *B*, *C*, *D*, *E*, *L*, *M*, *N*, *T*, *Q*, and *X*). These keys have the same meaning as in the section line survey.

63. The modules are the same as for section line surveys with two changes. The module FIND_SPOT replaces PLOT_SCREEN and the module PILOT_FIND_SPOT replaces PILOT. The change in the latter module is required because pilot guidance uses correction in heading in degrees (-180 to +180) in place of distance off line (left or right).

Data acquired by other means

64. The MANUAL_SURVEY_ENTRY program allows entering depths from keyboard at a fixed distance along the line interval. The operator enters the desired distance along the line interval in feet and the line number to be entered. Next the program prints the distance along the line and asks for depths.

65. The DIGITIZE_DEPTH_CHART program allows digitizing data via a Summagraphics* tablet digitizer from a hard-copy depth plot. (It may be used for depth charts if the boat is travelling at a constant speed or variable speed. Nonlinear distance along the line is supported.) This program automatically rotates the coordinate system as needed.

Postsurvey

66. There are several programs available to postprocess survey data. The programs may be used with data collected by any of the four survey methods, the keyboard entry method, or the digitizer input method. These include list programs, adjustment programs, cross-section and quantity programs, and plan views.

List

67. READ_DATA, READ_MARKS, and READ_TIDE list data, mark, and depth adjustment files, respectively. PRINT_DATA and PRINT_MARKS print data and marks, respectively.

---

* Summagraphics is a registered trademark of Summagraphics Corporation.

Data adjustment

68. There are several data adjustment programs available. ADJUST_DEPTHS_FOR_TIDE allows either a fixed tide, stepped tides, or linearly interpolated tide values to be subtracted from the data. RAW_DATA_TO_TEXT converts the binary raw data to an American Standard Code of Information Interchange (ASCII) form that can be modified by the text editor (file name TEXT).

69. EDIT_DATA is a program to edit surveyed data graphically for bad depth removal and for extension beyond existing data bounds. The program is designed to allow deletion of data on the existing depth versus distance along line curve and addition of data beyond curve limits. The program asks for the site name and whether or not a template is to be displayed. The template should not be displayed on profile lines, only on cross-section lines. The first acquired line is then displayed in cross-section graphical form and the arrow keys can be used to position the crosshairs over the point of interest; when on the curve only, the left-right arrows work and up-down motion follows the curve.

Cross sections and quantities

70. The CROSS_SECTION_DESCRIPTION program accepts inputs from keyboard and writes them out to a file with extension CROSS. It asks what is to be done: screen plots, printer plots, plotter plots, quantities to screen, and quantities to printer. It requests the pen number for the templates, allowed side slopes, center lines, surface, letters, predredge depths, and postdredge depths. Selecting DO NOT PLOT will indicate that this item is to be omitted. The program then asks letter size and scales.

71. The CROSS_SECTION_AND_DREDGING program displays cross sections (Figures 21 and 49) on screen, printer, or plotter. Computations include areas within templates, within the allowed side slope, and between surveys. Volumes and cumulative volumes are computed using these areas. MOBILE_QUANTITY_FORMAT consolidates the data for presurveys and postsurveys and MOBILE_SINGLE_FORMAT for single surveys. Three-dimensional stacked cross sections may be displayed and plotted. 3D_MAKE is run to generate data on a mesh and 3D_PLOT is used to plot it (Figure 22).

72. Quantities may also be computed using the National Bureau of Standards/Engineering Topographical Laboratory algorithm (Bernal 1988). TEMPLATE_MAKER and XYZ generate input data. TOP and DRE use a triangular mesh

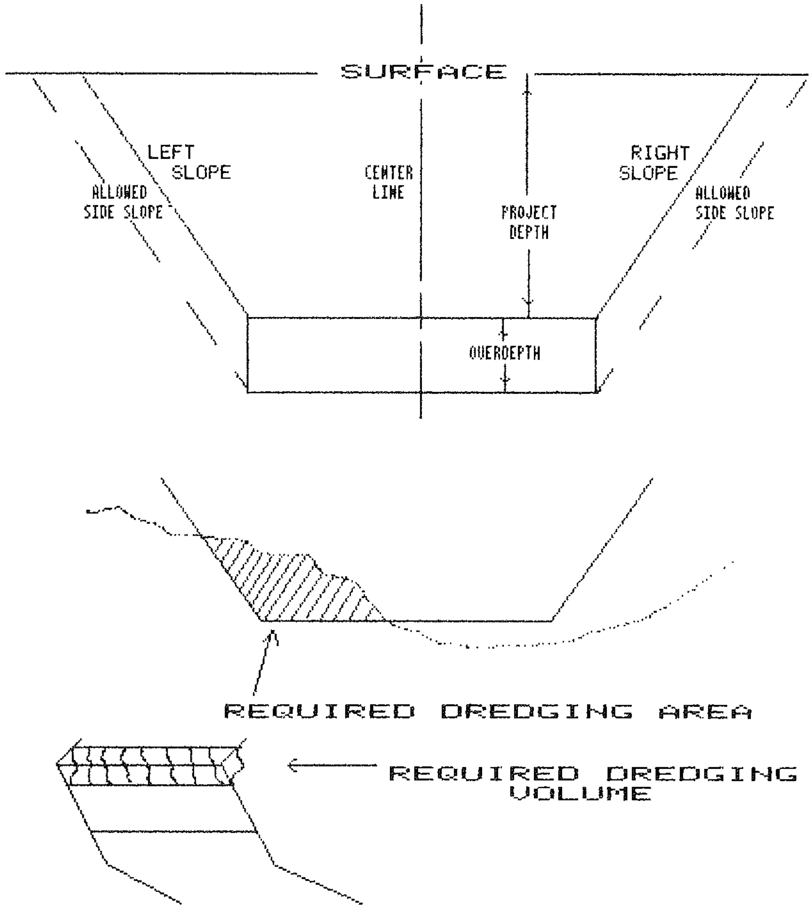


Figure 49. Cross section and dredging volumes

to compute a single volume of material within the channel template.

73. The PLAN_VIEW_PLOT_DESCRIPTION program accepts inputs from keyboard and writes them out to a file with extension PLAN. It requests the pen number for the center lines, channel lines, baselines, depths, legends, section lines, grid marks, north arrow, and transponders. Plot scale, rotation angle, and contouring information are entered.

74. The DIGITIZE_PLOTTER and DIGITIZE_FEATURES programs allow digitizing features via a plotter or digitizing tablet, respectively, from a hard-copy plan view. These programs automatically rotate the coordinate system as needed. The PLAN_VIEW_PLOTTER (Figure 20) program plots channels, lines, data, features, screen image pictures, user-generated labels, and an automatically generated and sized legend. The PLAN_VIEW program (Figure 50) is identical to PLAN_VIEW_PLOTTER except that it additionally produces contours via a triangular mesh (Hamm, Kibler, and Morris 1975) fitted to 300 points of the averaged data and interpolated by a fifth-order bivarient polynomial fit

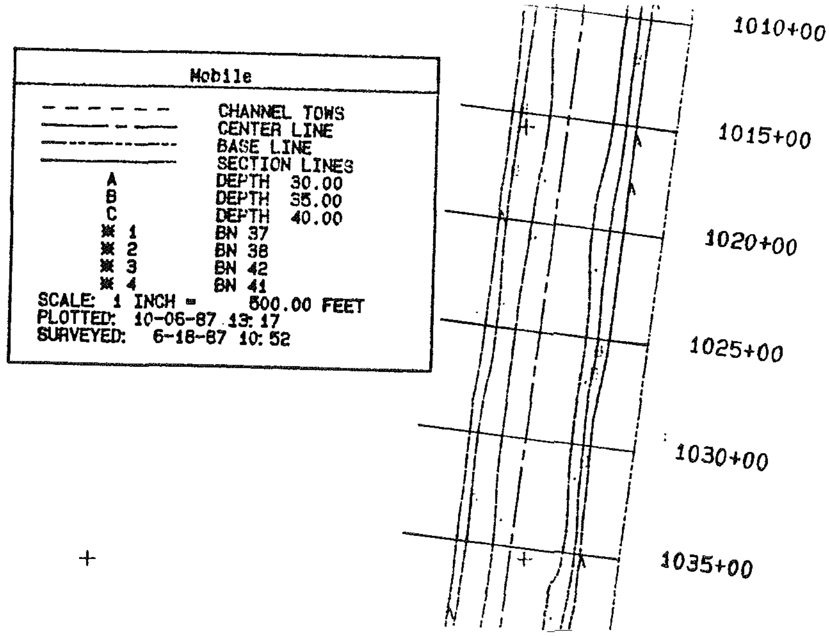


Figure 50. PLAN_VIEW

(Akima 1975). Further information on combined fit algorithm may be obtained in Tracy (1983). There are also programs for plotting features and user-generated labels that do not require a survey site or data; these are used for general-purpose computer-aided map (CAM) generation. EDIT_FEATURES is a program that allows maps to be input and graphically edited using a digitizing tablet and arrow keys (similar to computer-aided design packages; i.e., stamps of lighthouses, Corps emblems, etc., may be saved and restored, rotated, and scaled on new maps).

District Assistance

75. Several Districts funded enhancements to their particular survey systems. This had the effect of broadening the capabilities of the software package developed for Corps-wide use. Their contributions are summarized.

Mobile

76. Since the software package provided under HQUSACE funding was limited, Mobile District chose to fund a number of enhancements and additions. Most of these dealt with input of data from sources other than the survey program. These included (a) a keyboard data input program, (b) a digitizing tablet strip chart data input program, (c) a digitizing tablet features input

program, (d) a Topcon* data logger to data program, and (e) a Topcon data logger to features program. The District also funded inclusion of several position and depth systems, conversion programs to transfer data to existing processing programs on Mobile's Harris** computer enhancements to plotting new programs, CAM enhancements, and new dredging output formats.

77. Mobile Area Office and Panama City Area Office have been instrumental in bringing the software to field readiness. The personnel at these two sites have been very cooperative and understanding of problems associated with the development and are to be commended for their steadfastness in field debugging.

Louisville

78. The Louisville and Huntington Districts were interested in developing a quick survey method by which scaled data without control point reference could be collected. Louisville funded modifications to the existing section line style survey used at Mobile. This included modifications to the survey program, conversion programs to transfer data to existing Harris computer programs, and a program to generate some pseudosection lines so that postprocessing programs could be used. Fortunately, because Mobile had ironed out most of the bugs in the package before Louisville became involved, only minor debugging was required.

---

* Manufactured by the Topcon Instrument Corporation of America.
** Harris is a registered trademark of the Harris Corporation.

PART IV: FINAL DEVELOPMENT CONCLUSIONS

Hardware

79. Two different range-azimuth systems (Hydro 1 manual system by IMC and Polarfix automated system by Krupp Atlas) were found to work well with the rest of the small-boat hydrographic survey system. Both give data age along with position so that time of position sent can be determined within 0.1 sec, and both tracked well in field tests. The Hydro 1 gives position and the age of the data as referenced to the Grid System computer's quiz character; the Polarfix gives a predicted position at the time of the quiz character and data age to show how old the last position was. For the Hydro 1, the age is subtracted from the time the quiz character was sent; for the Polarfix, ages greater than 4 sec indicate the position is questionable and the corresponding position is discarded. In this way the time of position is accurately obtained so that depth and position are correlated accurately. This is important because if a boat running cross sections were moving at 10 fps, a position-time error of 1 sec would effectively move the channel 10 ft to one side. Other range-azimuth systems should work well also, but ease of tracking and time of position should be investigated before purchasing. A number of range-range systems are presently in use. The addition of "space diversity" (i.e., two boat-mounted transponders and electronic control circuitry) to eliminate "range holes" is highly recommended. Pilot guidance over calm waters is smooth with this feature, but highly erratic without it. The position system with space diversity selects the shortest path return signal in a multipath environment. Multirange systems could be added with minor software changes but should also have space diversity, as having multiple shore sites alone will do little to correct for multipath problems.

80. The setup that is easiest to use is configuration 2, equipment on the boat (Figure 10). Therefore, if the boat is large enough to support this configuration, it is recommended. This setup is easier to use because there is less hooking and unhooking of electrical equipment. Most of the equipment is on the boat and can be left cabled together. Also some shore sites are hard to reach and carrying the minimal amount of equipment to the shore is desirable. For shallow water, where a small flat-bottom boat is used, configuration 1 (Figure 9) may be the best solution. Fewer components are

required on the boat, yet pilot guidance and on-line plotting are still available. This is a particularly good configuration for theodolite type positioning systems that require only a prism array on the boat end.

Software

81. Small-boat hydrographic survey software for many survey needs has been developed. Data acquisition, pilot guidance, and all postprocessing software are run on the same computer. The package was designed to be friendly, yet powerful. Good defaults are provided for most plot control entries with form-driven user modification. The package is modular and can be used with a variety of vendor equipment (although each item should be verified for support before purchase). Several cross-section-style survey methods are supported; sweep system surveys are not presently supported but could be added. Presently such corrections for boat motion as yaw, pitch, roll, or vertical displacement are not supported in software. This type of correction would allow correcting depths for wave action and boat squat. Since the software is modular, it would not require a large effort to include these highly desirable corrections. Since small boats are not normally used in rough water, these corrections for boat motion were not explored.

82. Tide adjustment data may presently be time marked at the time the operator enters them. There are a number of automated tide stations capable of sending data to a computer; so automated tide collection and application may be a future enhancement if funded.

Overall

83. The small-boat hydrographic survey system software is available free of charge through the WES Engineering Computer Programs Library* to Government agencies; this software is also available to non-Government agencies, but they must supply the blank diskettes on which the programs are to be recorded. The software is updated regularly to include enhancements and new equipment modules, and to correct any user-discovered bugs or peculiarities.

---

* Engineering Computer Programs Library, Customer Assistance Group, Information Technology Laboratory (CEWES-IM-SC), 3909 Halls Ferry Road, Vicksburg, MS 39180-6199.

Consultation to Government agencies can be provided by WES. Consultation to non-Government agencies is not available unless they are operating under a Government contract.

84. Hardware is available directly from the vendors. An effort will be made by WES to support as many systems as possible; however, since certain criteria must be met by the vendor, it is best to check on compatibility. It was not part of the scope of this report to evaluate vendor equipment; however, systems used are diagrammed. The development was carried out to ensure that at least one system for small survey boats would be available and to demonstrate the Corps' need of these type systems to induce vendor involvement. Several vendors have responded favorably, and more hardware is available or under development.

85. The two biggest successes of this project are (a) development of a powerful, easy-to-use, widely applicable software package and (b) the encouragement of vendor development of hardware suitable for small-boat systems. Side effects include a potential for standardized Corps practices, ability to collect quantities of quality data, and common processing procedures across the Corps. Adoption of Corps-wide standards would reduce grounds for dispute.

REFERENCES

Akima, Hirosha. 1975 (Aug). "A Method of Bivariate Interpolation and Smooth Surface Fitting for Values Given at Irregularly Distributed Points," US Department of Commerce/Institute for Telecommunication Sciences, Washington, DC.

Bernal, Javier. 1988. "An Algorithm for Reconciling a Triangulation with a Line Segment," National Bureau of Standards, Gaithersburg, MD.

Boone, Charles G. 1980. "A Survey Application of a Remote Tracking Duplex Data Link System," Sixth Hydrographic Survey Conference, US Army Engineer Waterways Experiment Station, Vicksburg, MS, p 99.

Electronic Industries Association. 1969. "RS232 Interface, EIA Standard," Washington, DC.

Hamm, Kibler, and Morris. 1975. "A Program for Contouring Randomly Spaced Data," National Aeronautics and Space Administration, Langley Station, VA.

Institute of Electrical and Electronics Engineers Standards Board. 1978. "ANSI/IEEE Standard 488-1978," IEEE Standard Digital Interface for Programmable Instrumentation, 18 July 1978, Institute of Electrical and Electronics Engineers, Inc., and American National Standards Institute, New York.

LaFountain, Jack. 1976 (Mar). "The Buffalo District's Small Boat Integrated Hydrographic Surveying and Positioning System," US Army Corps of Engineers Fourth Hydrographic Surveying Conferecne, 5-6 Feb 1975, US Army Engineer Waterways Experiment Station, Vicksburg, MS, p 11.

McCleave, Barry W. 1985 (Feb). "Small Boat Survey System," Proceedings, US Army Corps of Engineers Surveying Conference, Jacksonville, FL, US Army Engineer Waterways Experiment Station, Vicksburg, MS, p 23.

__________. 1988 (Feb). "Small Boat Survey System," Proceedings, US Army Corps of Engineers Surveying Conference, Savannah, GA, US Army Engineer Topographic Laboratory, Fort Belvoir, VA, pp 30-53.

Thrower, Donald. 1978 (Mar). "Trailerable Automated Survey Boat," Fifth Hydrographic Survey Conference, 8-9 Nov 1977, US Army Engineer Waterways Experiment Station, Vicksburg, MS, p 7.

Tracy, Fred T. 1983 (Oct). "User's Guide: Interactive Graphics Computer Program for Contouring Randomly Spaced Data," US Army Engineer Waterways Experiment Station, Vicksburg, MS.

Table 1

Minimum Hardware Required

Essential:
- (1) Grid* 1129 or 1139 computer**
- (2) Grid 2101 disk drive (with 20-megabyte hard disk** upgrade), double-sided double-density diskettes
- (3) IEEE488 cable to connect (1) and (2) (approximately 3 ft)
- (4) Serial cable Grid number 6100**
- (5) RJ11 phone line (standard modular plugs both ends, approximately 20 ft)
- (6) Grid OS,* Grid Term,* and Grid Paint* software on floppy (optionally FORTRAN, MS-DOS,† etc.**)
- (7) RS232 positioning system with RS232 cable (WES-made cable)
- (8) IEEE488 depth system with IEEE488 cable

For Pilot Guidance:
- (9) TRS80 (Radio Shack) Model 102 Executive Workslate
- (10) AC adapter for (9), Radio Shack 26-3804
- (11) Modem cable for (9), Radio Shack 26-1410

For Plotting:
- (12) Hewlett-Packard (HP) plotter (preferably model Draftmaster for office or Draftpro for field)
- (13) IEEE488 cable for (12) (12 ft approximately) and pens/paper

For Phoning from Hotel Rooms through Operator:
- (14) Grid Handset*,**

For Digitizing Data from Depth Charts, Maps, etc.:
- (15) Summagraphics Bit Pad 1 (small) or Summagraphics Microgrid (large)
- (16) RS232 cable for digitizer (WES made)

For Digitizing Data from Plotter (slow and not recommended):
- (17) Digitizing sight (either straight or angled)

To Aid in Producing Logos, etc., for Inclusion on Plots:
- (18) Grid's version of Mouse System's mouse**,††

For Printing:
- (19) HP Thinkjet, Laserjet II, or many Epson‡ models

For Really Rough Waters:
- (20) Targa Electronics Systems, Inc., bubble (3101) or nonvolatile ram cartridge storage units--specify Grid interface IEEE488**

(Continued)

---

* Registered trademark of Grid Systems Corporation.

** Recently the software was rewritten to run on MS-DOS PC-compatible systems. Minimum computer hardware may now be an MS-DOS computer (e.g., Grid 1535) with Hayes-style modem on COM1:, RS232 serial port COM2:, 640 kbytes memory, CGA graphics capability (most EGA and VGA cards support this), National Instruments PC2A IEEE488 card, and 8087 coprocessor. Software requirements are MS-DOS and Intergrid (with optional programs).

† MS-DOS is a registered trademark of Microsoft, Inc.

†† The Grid version of Mouse System's Mouse is a registered trademark of Grid Systems Corporation.

‡ Epson is a registered trademark of the Seiko Epson Corporation.

Table 1 (Concluded)

For Locating Equipment on Shore:
(21) Motorola (or others) radio links (check with WES)
For Brightly Lit Survey Sites (which is almost everywhere):
(22) Sunscreen tinted mylar for boat cabin windows to reduce glare

APPENDIX A: DEPTH AND RANGE SYSTEMS

Depth Sounders

1. The depth sounders, commonly used for acquiring the distance from the water surface to the bottom, are sonic in nature. A burst of vibratory energy is emitted from a transducer located on the hull of the boat (preferably directly beneath the positioning system antenna). The transit time between the release of this burst and the first returned energy above a certain threshold amplitude is taken to be the time of travel to the bottom and back. This time is then divided by two to get the one-way travel time, and depth is calculated from the equation:

$$\text{DEPTH} = \text{SPEED OF SOUND} \left(\frac{\text{Time}}{2}\right) + \text{DRAFT} - \text{TIDE} \qquad \text{(A1)}$$

The draft of the boat is added to compensate for the depth of the transponder from the surface. The raw depth is tide adjusted to some reference plane by subtracting the local tide for the area surveyed.

2. The particular frequency used, beam angle, and peak selection algorithm affect the depth obtained on anything but smooth, hard riverbeds. The frequency used influences the depth reading on soft bottoms or in areas having suspended sediment. The lower the frequency, the better energy penetrates suspended material and soft bottoms. However, lower frequencies require bigger transducers and subsequently result in larger beam angles. The beam angle is important because it determines the size of the footprint on the bottom. The wider the angle, the bigger the footprint; also, the deeper the bottom, the bigger the footprint. Since most commercial digitizers normally use the first returned peak, they will usually pick up the highest peak on the bottom within the beam footprint. This tends to stretch the peak areas to nearly the width of the footprint. Since the two problems, penetration and footprint size, are diametrically opposite in solution, a compromise must be reached. Most Districts use narrow-beam (2-8 deg) tranducers at about 200 kHz. Interface to this type system was designed into the package.

3. The software expects a digitizer that can measure and transmit depths at about 10 times per second. Rapid updates are necessary on the banks of rivers to properly resolve the rapidly changing side slope. To ensure readings at this rate, digitizers are interfaced by a parallel digital

input/output method, the IEEE488 interface (Institute of Electrical and Electronics Engineers (IEEE) Standards Board 1978).* This provides data transfer at much higher rates (up to 1 million bps) than serial methods (which generally transmit at 120 to 960 bps). Another reason for requiring the depth system to be interfaced via IEEE488 is that the particular computer used in the small-boat system has only one serial interface, which is devoted to the slower position information.

Positioning Systems

4. Position is another critical parameter that must be measured accurately during hydrographic surveys. Obstructions to boat traffic must be identified by location for avoidance and/or removal. There are a number of types of electronic positioning methods in use by surveyors: range-range, range-azimuth, and absolute positioning.

Range-range

5. Range-range systems usually involve radio links between one or two ship-mounted transponders and two or more shore-mounted transponders. In normal range-range calculations only one ship transponder and two shore transponders are used. The ship unit typically broadcasts a pulse of radio frequency energy. The two shore units receive the pulse and after a short delay each returns a pulse of energy. The turnaround delay in the shore unit is constant for a particular shore unit. The adjusted transit time divided by two is used to compute the one-way distance between the boat and the shore station by the following equation:

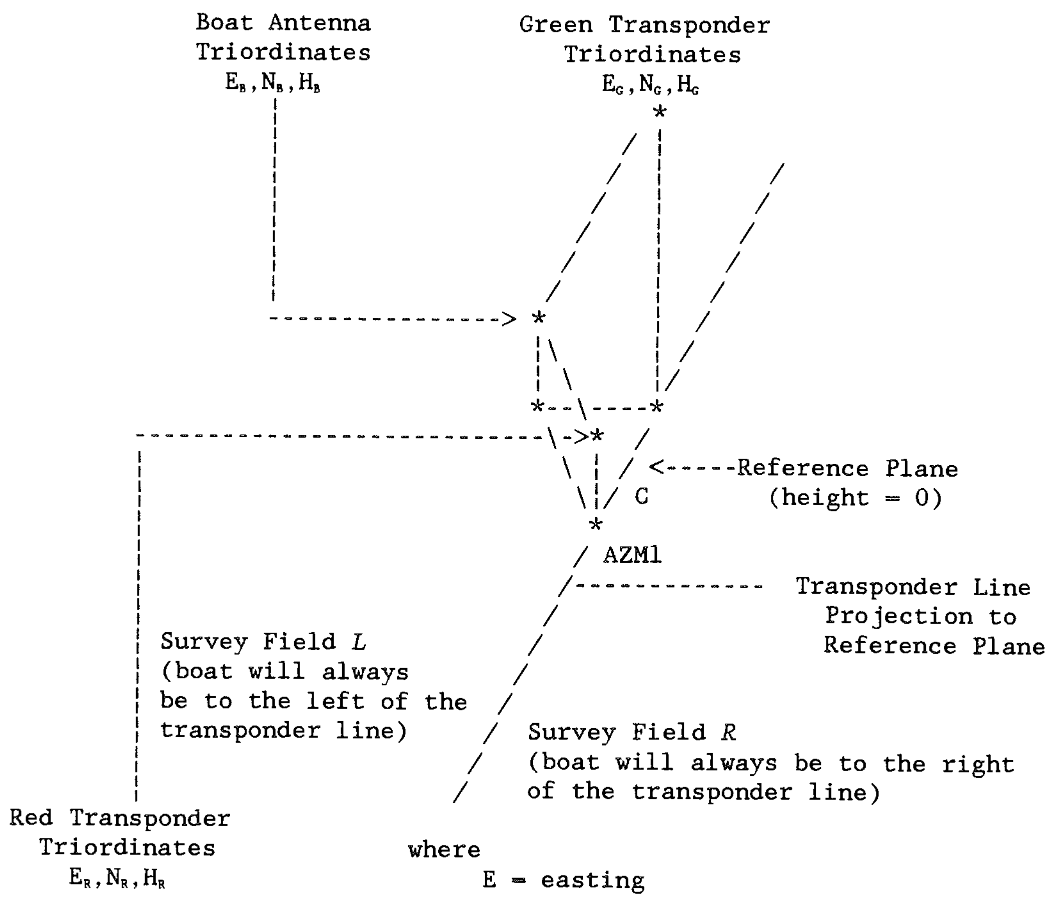
$$\text{DISTANCE} = \text{SPEED OF WAVE PROPAGATION} \times \text{TIME} \qquad \text{(A2)}$$

where

$$\text{TIME} = \frac{(\text{TOTAL TIME} - \text{TURNAROUND DELAY})}{2}$$

The method for range-range position determination is shown in Figure A1. The software uses the distances separating the boat from two shore stations of known position to calculate boat position via triangulation. Two transponders

* All references cited in this Appendix are included in the References at the end of the main text.



where
E = easting
N = northing
H = height

a. Transponder line projection to reference plane

1. Compute the length and angle of the transponder coordinates projected on the zero-height reference plane.

   a. Compute the projected length:

$$E_{RG} = E_G - E_R \qquad (1)$$

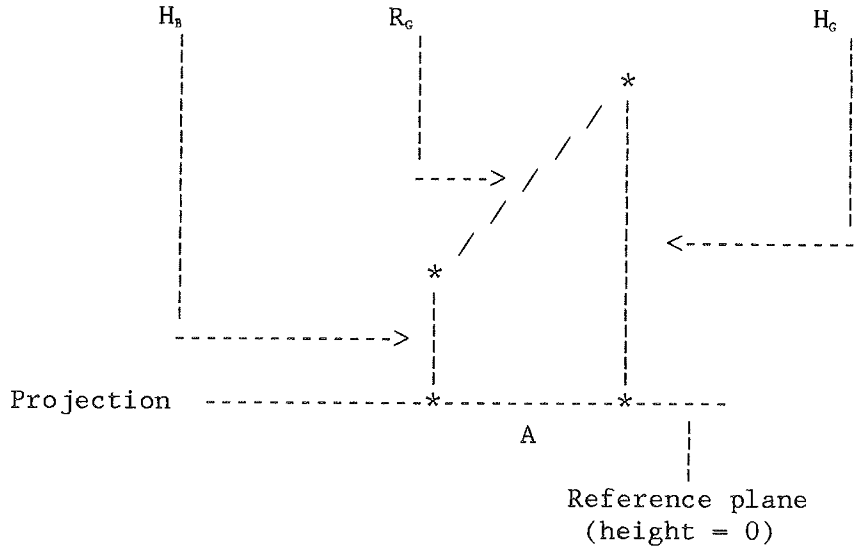
$$N_{RG} = N_G - N_R \qquad (2)$$

$$C = \sqrt{N_{RG}^2 + E_{RG}^2} \qquad (3)$$

   b. Compute the projected angle:

$$AZM1 = \arctan \frac{N_{RG}}{E_{RG}} \qquad (4)$$

Figure A1. Computing boat position given a range-range positioning system (Sheet 1 of 3)



b. Range to reference plane

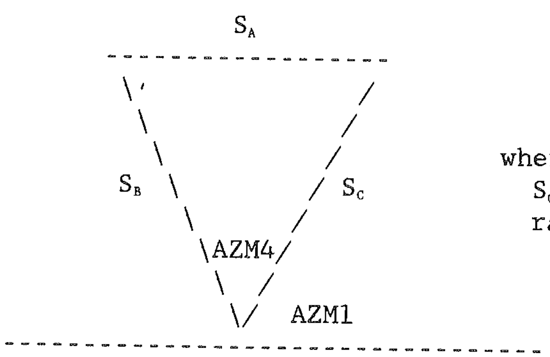
2. Compute the projected length of the acquired ranges on the zero-height reference plane.

   a. Use the Pythagorean theorem to project the reading of the green transponder's range R_G :

$$S_A = \sqrt{R_G^2 - \left(H_G - H_B\right)^2} \tag{5}$$

   b. Use the Pythagorean theorem to project the reading of the red transponder's range R_R :

$$S_B = \sqrt{R_R^2 - \left(H_R - H_B\right)^2} \tag{6}$$



where S_A , S_B , and S_C are sides of the range triangle

   c. Conversion of range-range to range-azimuth

3. Use law of cosines to transform range-range to range-azimuth relative to transponder line angle:

Figure A1. (Sheet 2 of 3)

$$AZM4 = \arc\cos\left(\frac{S_B^2 - S_A^2 + S_C^2}{2\ S_B S_C}\right) \qquad (7)$$

4. Add transponder angle to relative azimuth to get absolute azimuth:

   If survey field = R , then

$$AZM5 = AZM1 - AZM4 \qquad (8)$$

   If survey field = L , then

$$AZM5 = AZM1 + AZM4 \qquad (9)$$

5. Convert from polar to rectangular coordinates to get boat's easting and northing relative to the red transponder:

$$E_0 = A \cos AZM5 \qquad (10)$$

$$N_0 = A \sin AZM5 \qquad (11)$$

   where

   E_0 = eastward offset of boat from red transponder

   N_0 = northward offset of boat from red transponder

6. Add relative coordinates to green's coordinates to get boat's position:

$$E_B = E_0 + E_R \qquad (12)$$

$$N_B = N_0 + N_R \qquad (13)$$

Figure A1. (Sheet 3 of 3)

(space diversity) are used on the boat to lessen the effect of multiple transmission paths. The wave propagation between boat and shore transponders may not necessarily take a direct path between the two points. It may reflect off the water and take a longer path. Using two receivers separated by a wavelength usually causes one of the receivers to get a direct path return. Hardware in the positioning system selects the first return or stronger return, which will correspond to the shorter path, and passes its distance to the computer. The two transponders on the boat will generally provide immunity to range holes and give improved accuracy and stability in pilot guidance.

6. Use of multiple shore stations has the potential for improving position calculations, if properly done. A number of carefully calibrated, properly located shore stations will give confidence due to redundancy, if the positional error is small. When more than two shore stations are used, two methods are commonly used for calculating position. One method uses a least squares fit to all ranges (sometimes weighting them based on accuracy of geometry). This approach uses distances that are known to be inaccurate and will thus give a less than ideal position; however, it does give an accuracy factor that allows the surveyor to have some feel about how much positional error he might be getting. Software should alert the operator if this error exceeds a preset limit. The other method is to let the computer select which of the shore stations gives the best triangulation geometry and use them in a particular area.

7. Since the surveyor is trained in selecting shore positions that will give accurate geometry and has a knowledge of which shore transponders are operating best, the small-boat survey system software uses only two shore stations that are selected by the surveyor, not the computer, at present. Since this requires that only two transponders be calibrated and accurately positioned at known shore coordinates (in practice), the chances of getting accurate positions may actually be improved as the surveyor often has a limited amount of time to calibrate, set the transponders, and acquire the data. Two carefully calibrated, properly located shore stations will give better positional accuracy than a number of poorly located or calibrated redundant shore stations.

Range-azimuth

8. Range-azimuth positioning systems measure distance from a shore site to the boat and the vertical and horizontal angular movement of the shore unit

necessary to align the boat target. Some systems use the same distance-measuring method commonly used with range-range, whereby a shore transponder sends a signal to a boat transponder. After a constant turnaround delay, the signal is returned to the shore station (once again two transponders at the shore station would help eliminate nondirect-path range holes). The other distance-measuring method uses infrared or laser light as the medium and prisms on the boat as a passive target; it uses either phase shift or transit time (from shore to boat and back) to compute distance. Angular movement of the shore station is normally measured by encoders. Encoders may be either electrical or optical and measure degree of rotation from some horizontal and vertical reference. Normally, there is a separate encoder for the vertical and horizontal angle, although some systems measure horizontal angle and use relative distance and height to approximate vertical angle.

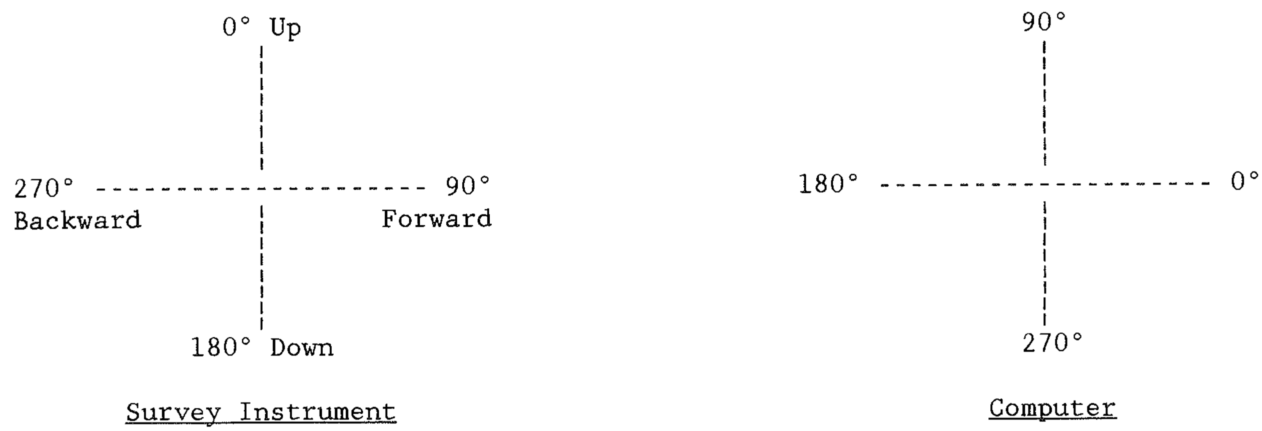
9. Although multiple range-azimuth sites could be used in the same way that multiple range systems are used, normally only one site is used, as the main purpose of using this type system is to avoid having to establish more than one shore station. Range-azimuth positioning geometry offers the considerable advantage over range-range geometry in that there are no poor geometry areas. However, it has the disadvantages of being limited to line of sight. The small-boat system software supports range-azimuth positioning by subtracting the reference angle and using triangulation to determine position. Figure A2 shows the method used in a range-azimuth positioning system.

Absolute position

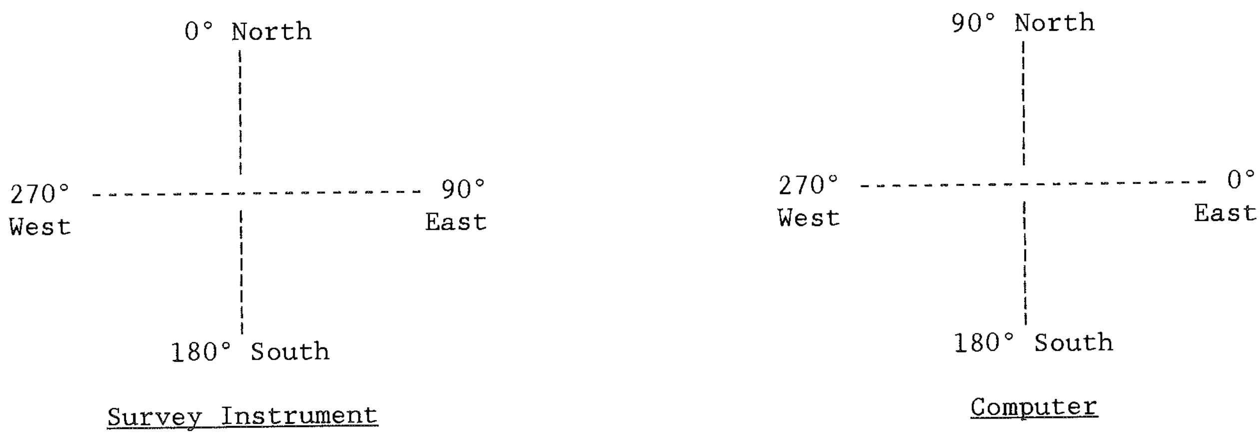
10. Some systems provide either absolute coordinates or relative coordinates. Satellite systems, for example, will provide absolute coordinates in some reference system which will probably have to be mapped onto the surveyor's system. Some range-azimuth systems supply relative coordinates from the shore site. The software then adds the relative coordinate values to the absolute shore site coordinates to determine absolute position.

Pilot guidance

11. It is often desirable to run predetermined section lines. To provide pilot guidance, the absolute boat position in easting and northing must be related to the section line in terms of distance along the line and offset from the line. The method of computing distance along the line and offset from the line is shown in Figure A3.



a. Conversion of vertical azimuth



b. Conversion of horizontal azimuth

1. Convert survey instruments coordinate system to that used by the computer:

$$AZM_H = 90 - AZM_{HS} \quad (1)$$

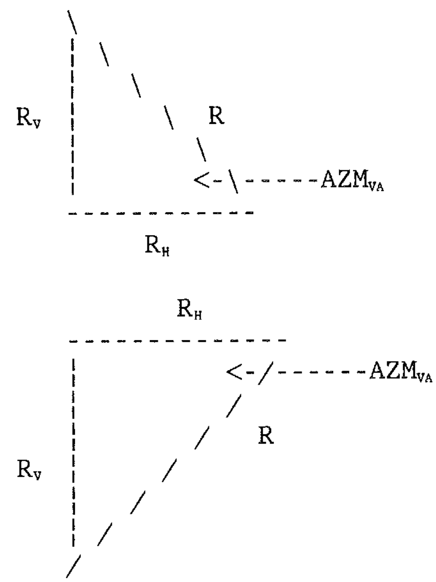
$$AZM_V = 90 - AZM_{VS} \quad (2)$$

$$AZM_{VA} = \left| AZM_V \right| \quad (3)$$

where

Figure A2. Computing boat position given a range-azimuth positioning system (Sheet 1 of 3)

AZM_H = horizontal azimuth in mathematical coordinate system
AZM_{HS} = horizontal azimuth from survey instrument
AZM_V = vertical azimuth in mathematical coordinates system
AZM_{VS} = vertical azimuth from survey instrument
AZM_{VA} = Absolute value of AZM_V



where
R = range
R_H = range horizontal
R_V = range vertical

c. Computing horizontal component of range

2. Determine vertical and horizontal components of range:

$$R_H = R \cos \left(AZM_{VA}\right) \qquad (4)$$

$$R_V = R \sin \left(AZM_{VA}\right) \qquad (5)$$

Set R_V negative if viewing downward.

If AZM_V is less than 0, then

$$R_V = -R_V \qquad (6)$$

3. Adjust vertical height to remove effect of boat antenna height H_B from water:

$$R_V = R_V - H_B \qquad (7)$$

4. Calculate height of boat with respect to zero reference plane:

$$H_R = H_T + R_V \qquad (8)$$

Figure A2. (Sheet 2 of 3)

where

H_R = height of boat measured from reference
H_T = height of Transponder

5. Convert from polar to rectangular coordinates to get boat's easting and northing relative to transponder:

$$E_O = R_H \cos\left(AZM_H\right) \tag{9}$$

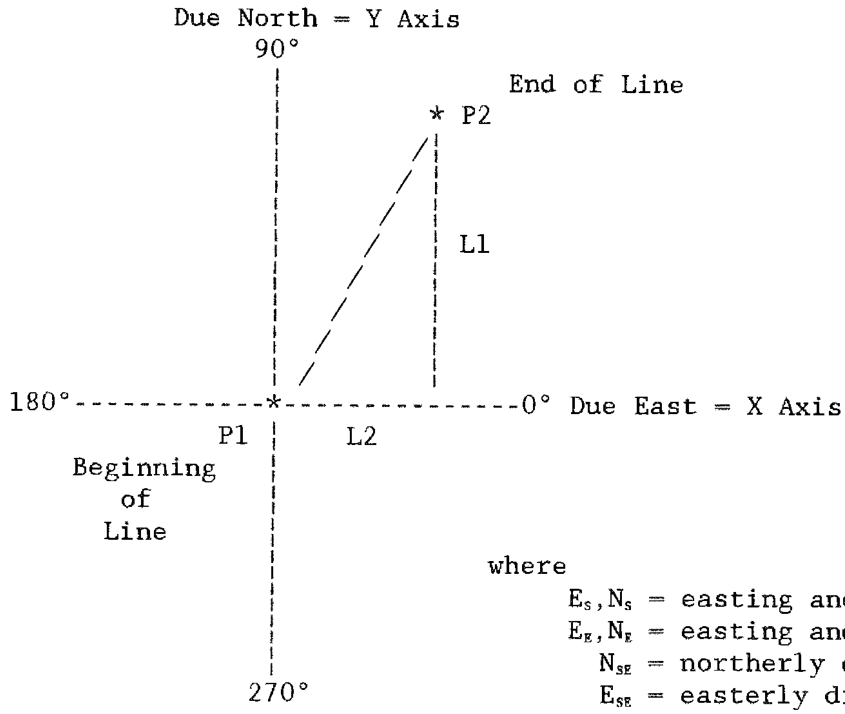
$$N_O = R_H \sin\left(AZM_H\right) \tag{10}$$

6. Add relative coordinates to transponder's coordinates, E_T and N_T, to get boat's position:

$$E_B = E_O + E_T \tag{11}$$

$$N_B = N_O + N_T \tag{12}$$

Figure A2. (Sheet 3 of 3)



where

E_S, N_S = easting and northing start of line

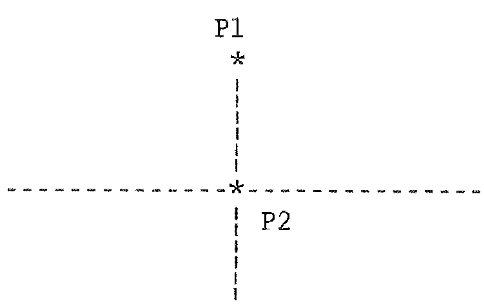
E_E, N_E = easting and northing end of line

N_{SE} = northerly distance from N_S to N_E

E_{SE} = easterly distance from E_S to E_E

P1 = E_S, N_S
P2 = E_E, N_E
L1 = N_{SE}
L2 = E_{SE}

or



P1 = E_S, N_S
P2 = E_E, N_E

a. Line to be surveyed

Figure A3. Converting boat position E_B, N_B to distance D along and offset O from a line (Sheet 1 of 6)

1. Compute equation of line to be surveyed ($AX + BY + C = 0$).

   a. Get projection of line on x- and y-axis (a signed length).

$$N_{ES} = N_E - N_S \tag{1}$$

$$E_{ES} = E_E - E_S \tag{2}$$

   b. Test projection of line on x-axis; if 0, then vertical line of infinite slope.

   c. Compute coefficients of line using

$$A = 1 \tag{3}$$

$$B = 0 \tag{4}$$

$$C = -E_S \tag{5}$$

   if E_{EB} is 0 .

   Or

$$A = \frac{N_{ES}}{E_{ES}} \tag{6}$$

$$B = -1 \tag{7}$$

$$C = N_S - AE_S \tag{8}$$

   if E_{EB} is not 0 .

   The formula of a straight line

$$Y = mX + b \tag{9}$$

   where m is the slope of the line and b is the Y offset when $X = 0$, is used as a basis to get the first-degree equation form

$$AX + BY + C = 0 \tag{10}$$

   where A , B , and C are coefficients of a straight line, by simply rearranging Equation 9 to yield

$$mX - Y + b = 0 \tag{11}$$

Figure A3. (Sheet 2 of 6)

Comparing Equations 10 and 11 suggests

$$A = m \tag{12}$$

$$B = -1 \tag{7}$$

$$C = b \tag{13}$$

Since the slope of the line is simply its y-projection divided by its x-projection,

$$m = \frac{N_{ES}}{E_{ES}} \tag{14}$$

and Equation 12 may be rewritten as

$$A = \frac{N_{ES}}{E_{ES}} \tag{6}$$

The constant b may be computed by either projecting the line back to the point it crosses the y-axis and reading the value of y or by "plugging in" a known value of x and y such as (E_S , N_S) and calculating b from Equation 11. Equation 11 may be combined with Equation 12 to yield

$$AX - Y + b = 0 \tag{15}$$

Plugging in (E_S , N_S) for (x, y) gives

$$AE_S - N_S + b = 0 \tag{16}$$

or rearranging

$$b = N_S - AE_S \tag{17}$$

Combining Equations 13 and 17 gives

$$C = N_S - AE_S \tag{18}$$

Therefore, Equations 6-8 give the coefficients of the straight-line equation for the general case. Equation 6 goes to infinity for the case when the survey line is vertical (i.e., $E_{ES} = 0$) and this special case must be tested for and treated separately.

If the line is vertical, its general equation is

$$X = a \tag{19}$$

Figure A3. (Sheet 3 of 6)

where a is the offset from the y-axis. Rewriting Equation 19 in the form of Equation 10 gives

$$X + (0 * Y) - a = 0 \qquad (20)$$

Comparing Equations 10 and 20 suggests

$$A = 1 \qquad (3)$$

$$B = 0 \qquad (4)$$

$$C = -a \qquad (21)$$

The offset from the y-axis of the vertical line is constant and equal to the offset at the beginning

$$a = E_S \qquad (22)$$

or end

$$a = E_E \qquad (23)$$

of the line.

Substituting Equation 22 into Equation 21 gives

$$C = -E_S \qquad (5)$$

Therefore, Equations 3-5 give the coefficients of the straight-line equation for the special case.

2. Use the coefficients of the straight line (A,B,C) calculated in step 1 and the boat coordinates (E_B,N_B) to calculate the distance off line using the trigonometric equation for the perpendicular distance from a point to a line:

$$\text{Offset} = \frac{A * E_B + B * N_B + C}{\sqrt{A^2 + B^2}} \qquad (24)$$

b. Distance of boat off line

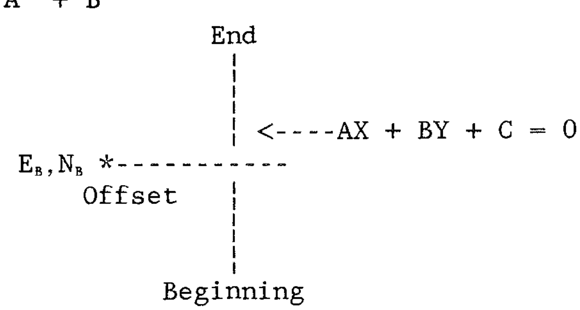


Figure A3. (Sheet 4 of 6)

3. Assign a sign to the offset to signify if the boat is to the left (-) or the right (+) of the line when facing toward the end of line while positioned at beginning of line.

   a. Compute the angle of the boat with respect to the x-axis at the beginning of line:

$$\text{Angle Boat Absolute} = \text{arc tan}\left(\frac{N_B - N_S}{E_B - E_S}\right) \tag{25}$$

   b. Compute the angle of the boat with respect to the survey line:

$$\text{Angle Boat Relative} = \text{Angle Boat Absolute} - \text{Angle Line} \tag{26}$$

   where

$$\text{Angle Line} = \text{arc tan}\left(\frac{N_E - N_B}{E_E - E_B}\right) \tag{27}$$

   c. If the angle of the boat relative to the survey line is between 0 deg and 180 deg, the boat is to the left of the line and

$$\text{Offset} = -\text{Offset} \tag{28}$$

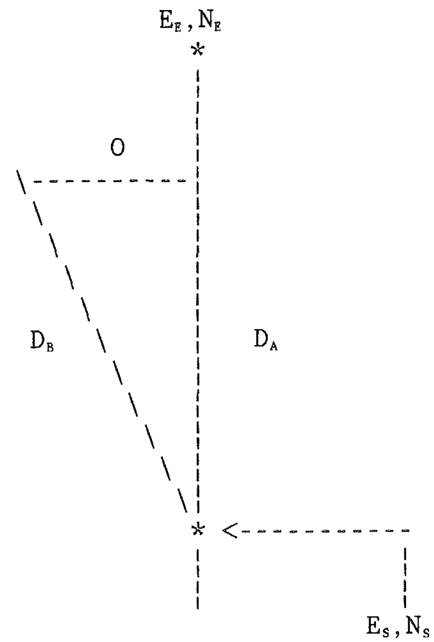
   Otherwise, the boat is to the right of the line and the offset is positive.

4. Finally, the distance along the line is calculated using the Pythagorean theorem.

   a. Compute the distance from the boat to the beginning of the line:

$$D_B = \sqrt{\left[\left(E_B - E_S\right)^2 + \left(N_B - N_S\right)^2\right]} \tag{29}$$

Figure A3. (Sheet 5 of 6)



where

D_A = Distance along line
D_B = Distance from start of line to boat
O = Offset distance of boat from line

c. Boat's distance along the line

b. Compute the boat's distance along the survey line:

$$D_A = \sqrt{\left(D_B^2 - O^2\right)} \tag{30}$$

c. Assign a sign to the distance along line to signify if the boat has not reached (−) the beginning of the line or has proceeded past the beginning of line (+). If 90° < Angle Boat Relative < 270° then

$$D_A = -D_A \tag{31}$$

Figure A3. (Sheet 6 of 6)

APPENDIX B: CIRCUIT AND CABLE DIAGRAMS

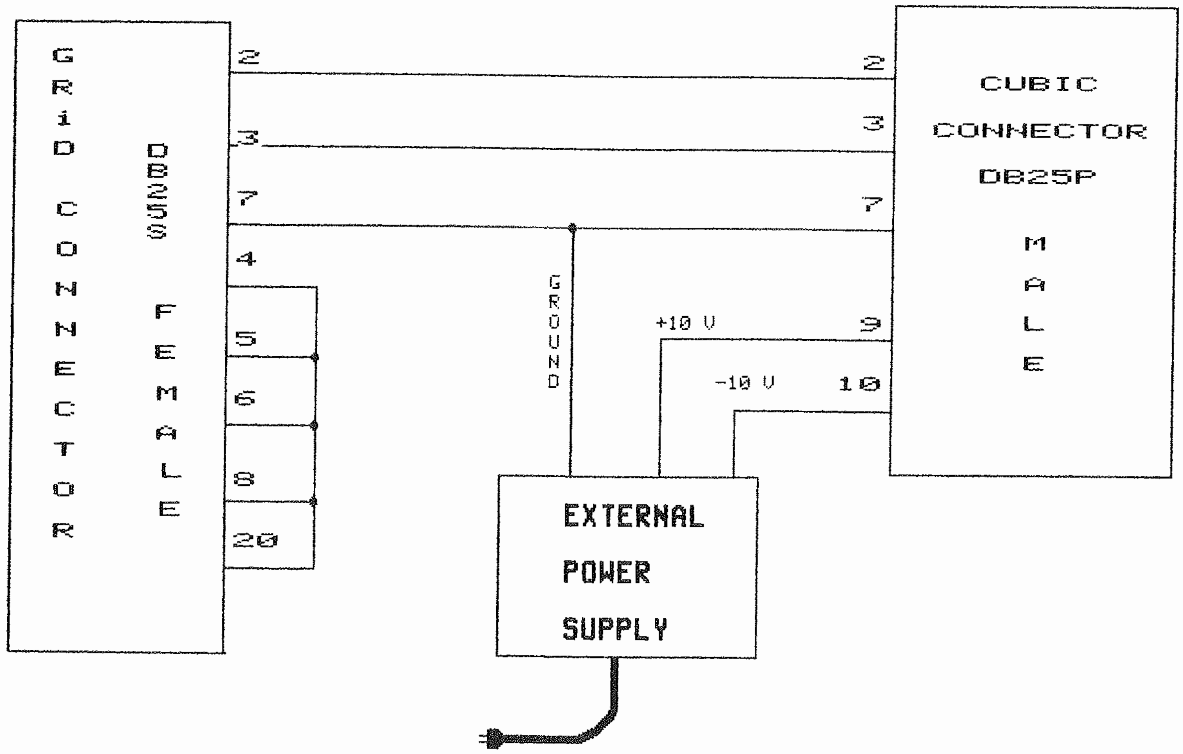


Figure B1. Autotape cable (Autotape is a registered trademark of Cubic Precision Corporation)

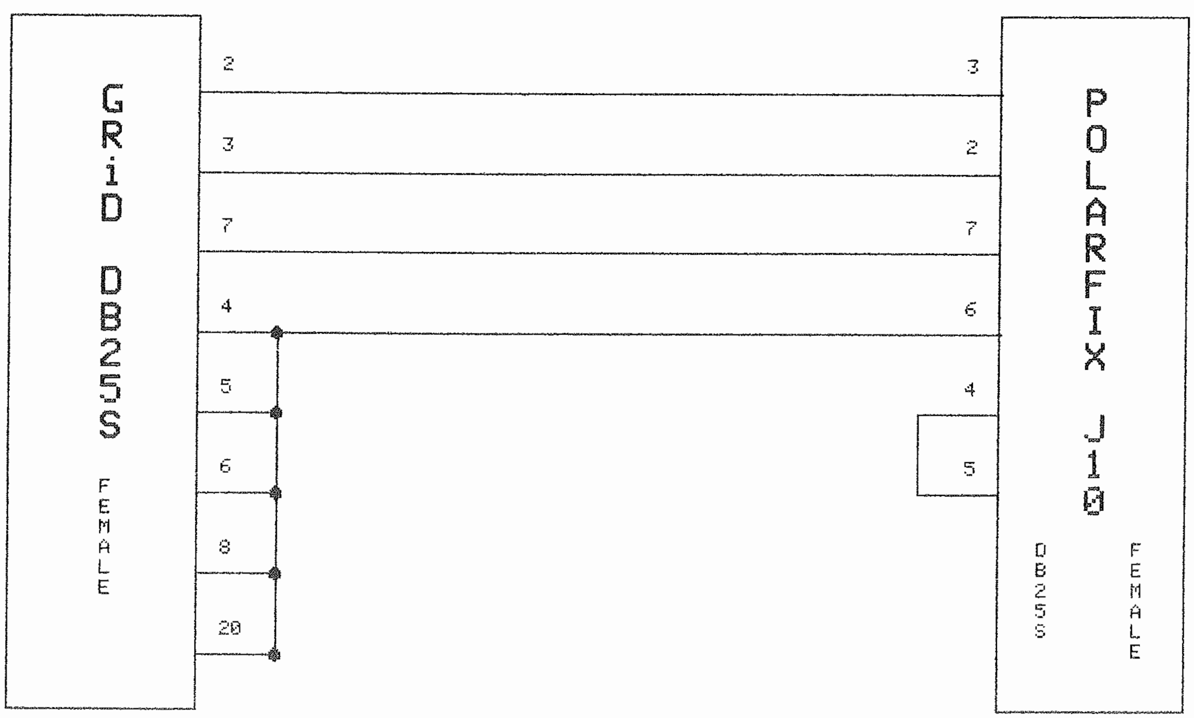


Figure B2. Polarfix cable (Polarfix is a registered trademark of Krupp Atlas)

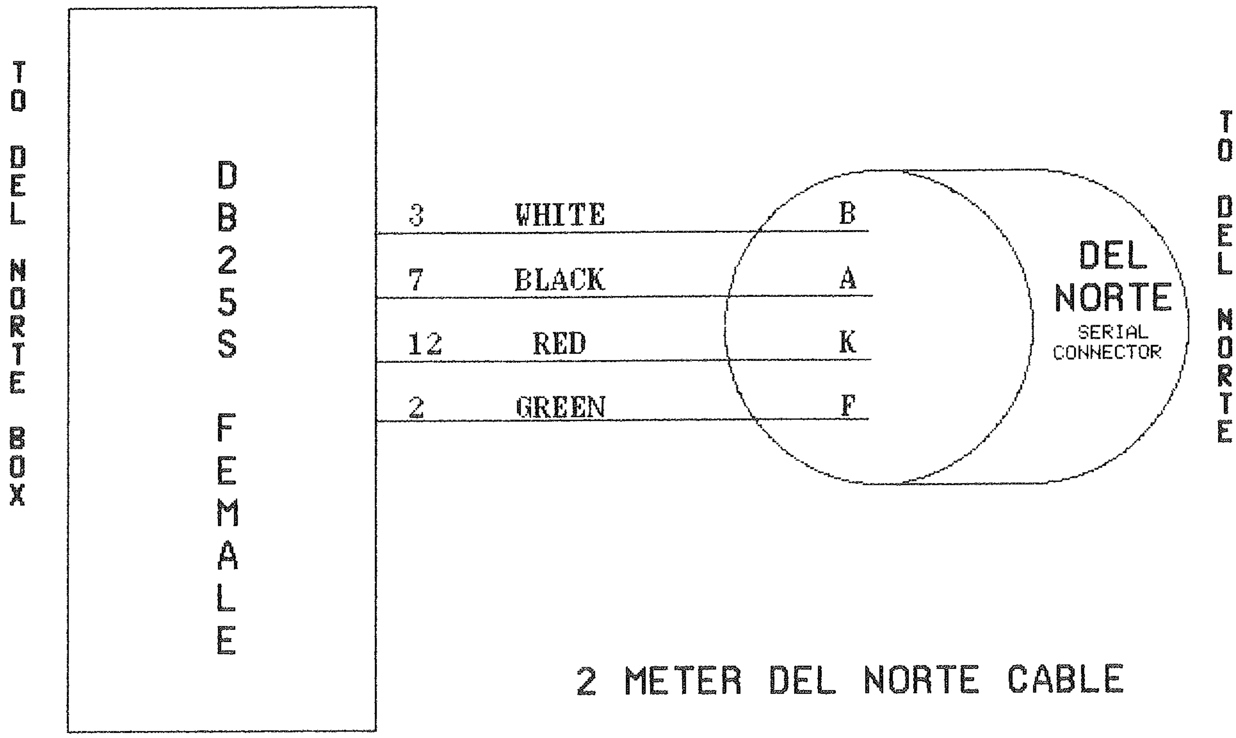


Figure B3. Internal to Del Norte jumper J22-39 to J1-K (serial). (Note: Pin J is used instead of K on older boxes.) (Del Norte is a registered trademark of Del Norte Technology)

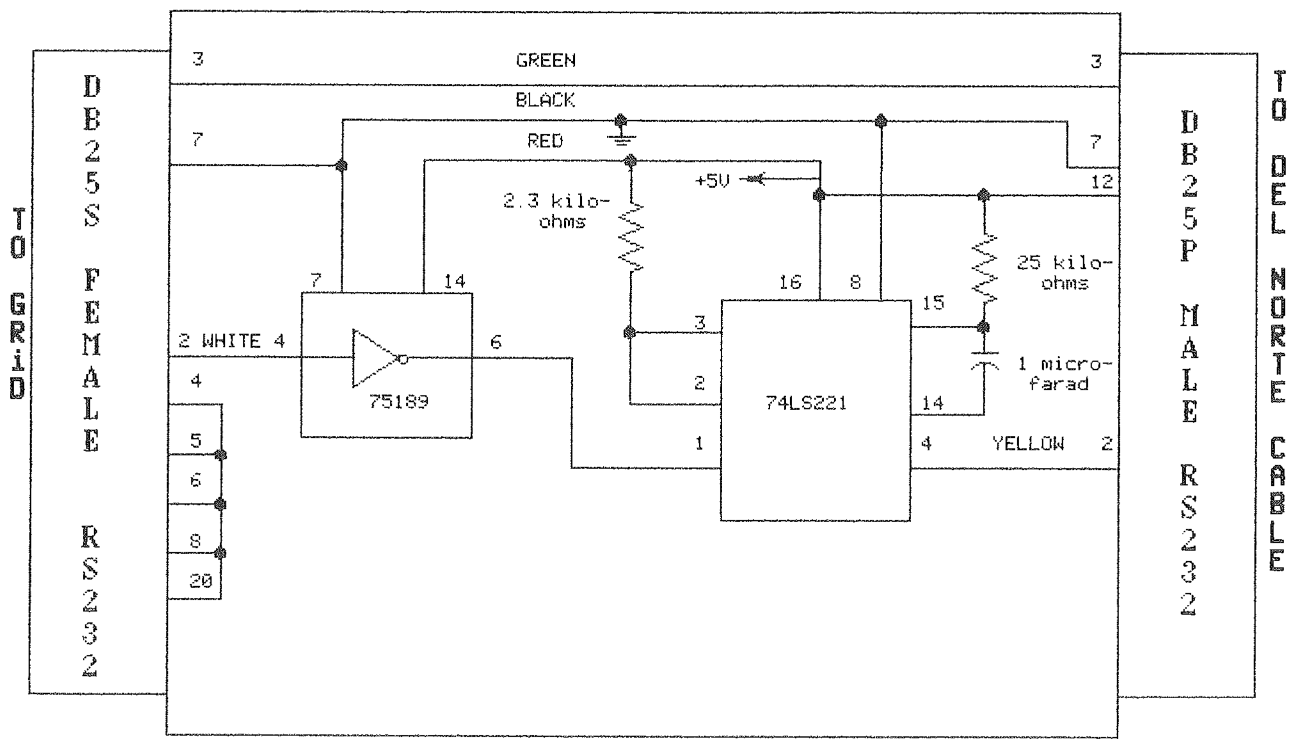


Figure B4. Del Norte RS232 to TTL Box

APPENDIX C: DETAILED PROGRAM DESCRIPTIONS

This appendix presents flowcharts and sample runs for each of the programs along with a description of the program and discussion of the theory. The programs are discussed in the following order:

3D_MAKE
3D_PLOT
ACAD_PLOT
ACAD_XYZ
ADJUST_DEPTHS_FOR_TIDE
CONFIGURE*
CONTOUR_PLOT*
CONVERT_DATA*
CONVERT_DATA_LOUISVILLE*
CROSS_SECTION_AND_DREDGING
CROSS_SECTION_DESCRIPTION
DEPTH_DEVICE*
DEPTH_SIMULATE*
DIGITIZE_DEPTH_CHART
DIGITIZE_FEATURES
DIGITIZE_PLOTTER
EDIT_DATA
EDIT_FEATURES**
EQUIPMENT_ENTRY
FIND_SPOT*
GENERIC_TO_GRID_FORMAT*
INDICATOR*
INTERSECTION_OF_2_LINES
MANUAL_SURVEY_ENTRY
MARKS_HARRIS*
MARKS_TO_DESCRIPTOR*
MARKS_TO_FEATURES*
MARK_DESCRIPTION
MOBILE_QUANTITY_FORMAT
MOBILE_SINGLE_QUANTITY
NEW_LINE_INPUT
NEW_SPOT_INPUT
PILOT*
PLAN_VIEW_DESCRIPTION
PLOT_FEATURES
PLOT_LETTERS
PLOT_LETTERS_DESCRIPTION
PLOT_PLOTTER*
PLOT_PLOTTER_LOUISVILLE*
PLOT_SCREEN*
PLOT_SCREENIMAGE*
PLOT_SCREEN_LOUISVILLE*
PRINT*
PRINT_DATA
PRINT_MARKS
RANDOM*
RANGE_DEVICE*
RANGE_SIMULATE*
SURVEY

---

* Sample run not included.
** Flow chart not included.

3D_MAKE

Purpose

3D_MAKE is a program that transposes the acquired data onto a rectangular grid for input to the three-dimensional (3D) plotting program.

Input files

NAME.LINE, NAME.0001 (where .0001 is line number)

Output files

TEMP3D

External devices

NONE

Modules called

GET_DEPTHS, PROBLEM, INITMENU, DRAWMENU, MESSAGESTACK, DRAWFORM, GET_NAME_OF_FILE

Theory of operation

The progam initializes variables. It then displays the file form for input file (NAME). Then it reads in the data and averages points along the line to produce a point at the desired grid interval. The section lines are used for the fixed grid intervals regardless of separation or azimuth so the distance along channel of the 3D plot will not reflect actual distance along channel, usually. The grid points are saved in TEMP3D for use by 3D_PLOT. The flowchart is shown in Figure C1, and two sample runs are shown in Figure C2.

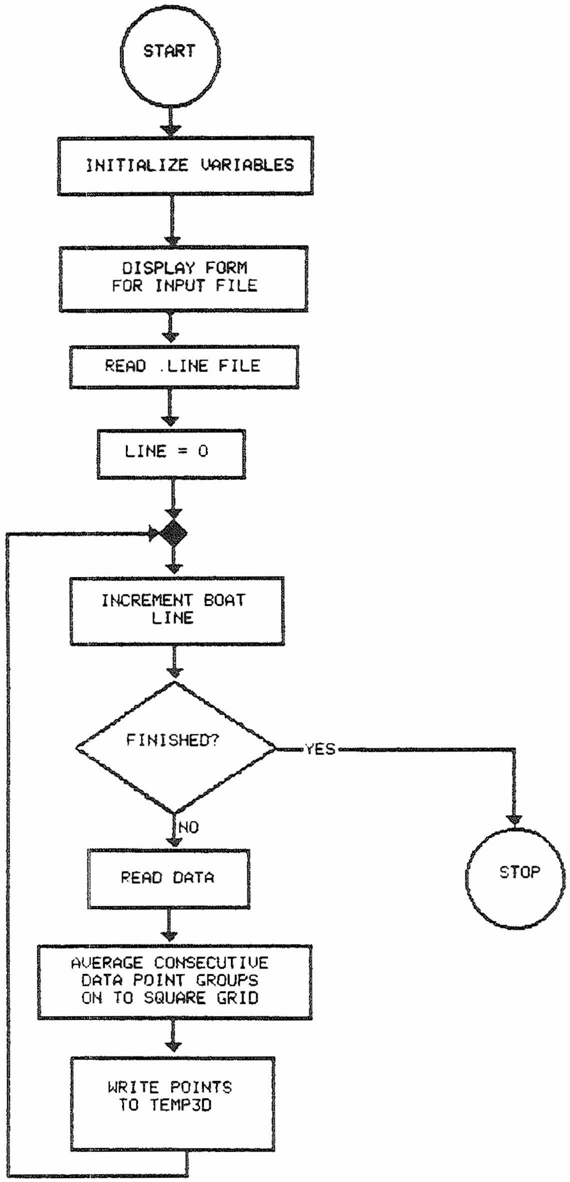


Figure C1. Flowchart, 3D_MAKE

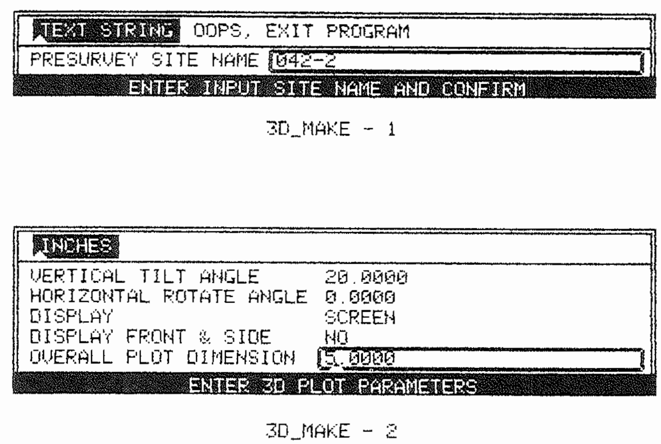


Figure C2. Sample runs, 3D_MAKE

3D_PLOT

Purpose

3D_PLOT is a program that plots data in a rectangular gridded format on the screen, a printer, and/or a plotter.

Input files

TEMP3D

Output files

NONE

External devices

Printer, plotter

Modules called

PROJ, SCREEN_LINE_PLOT, POINT_ALONG_A_LINE, PLOT_SCR, LTS, SCALE_IT_SCREEN, A2, PRINT_SCREEN, PROBLEM, DRAWMENU, MESSAGESTACK, DRAWFORM, INITMENU.

Theory of operation

The program initializes variables. It then displays a file form to accept vertical tilt angle, horizontal rotation, output devices, display of sides enabled, and plot size. The data are then smoothed by a nonrecursive filter and scales. Plotting is done by the subroutine PROJ. The flowchart is shown in Figure C3, and a sample run is shown in Figure C4.

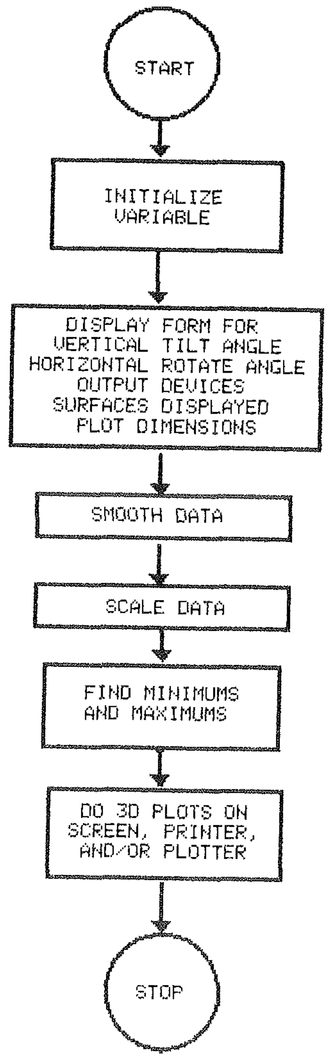
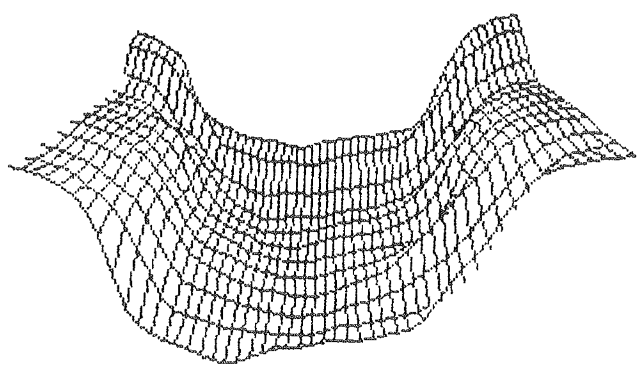


Figure C3. Flowchart,
3D_PLOT



3D_PLOT - 1

Figure C4. Sample run, 3D_PLOT

ACAD_PLOT

Purpose

ACAD_PLOT is a program that plots DXF plot files.

Input files

NAME (entered from keyboard)

Output files

NONE

External devices

Plotter

Modules called

NONE

Theory of operation

The program accepts the input file name and scaling factor. It reads data from the input file, scales it, and plots it. The flowchart is shown in Figure C5, and a sample run is shown in Figure C6.

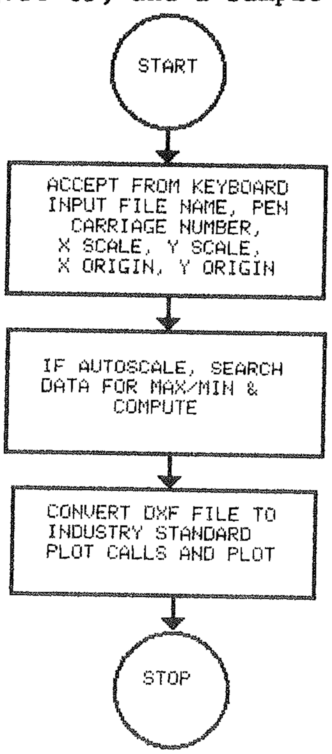


Figure C5. Flowcharts, ACAD_PLOT

```
=▸ACAD_PLOT^RUN^
 ENTER FILE NAME
XYZ
 ENTER PEN NO. (1-6)
3
 ENTER X SCALE FACTOR IN FT/IN (0. FOR AUTO)
0
 ENTER Y SCALE FACTOR IN FT/IN (0. FOR AUTO)
0.
```

ACAD_PLOT-1

Figure C6. Sample run, ACAD_PLOT

ACAD_XYZ

Purpose

ACAD_XYZ is a program to read in the acquired data from a site, reduce the number of points to 500 or less by averaging, and write a DXF file of the points.

Input files

NAME.0001 (where .0001 is the line number)

Output files

XYZ

External devices

NONE

Modules called

INITMENU, DRAWFORM, GET_NAME_OF_FILE, POLREC, A2, PROBLEM

Theory of operation

The program begins by displaying a file form for entry of the input file. It then reads in the .LINE file and determines the number of equally spaced points to extract from each line (500/number of lines). The program reads in the lines and averages the points for output in XYZ format. The flowchart is shown in Figure C7, and a sample run is shown in Figure C8.

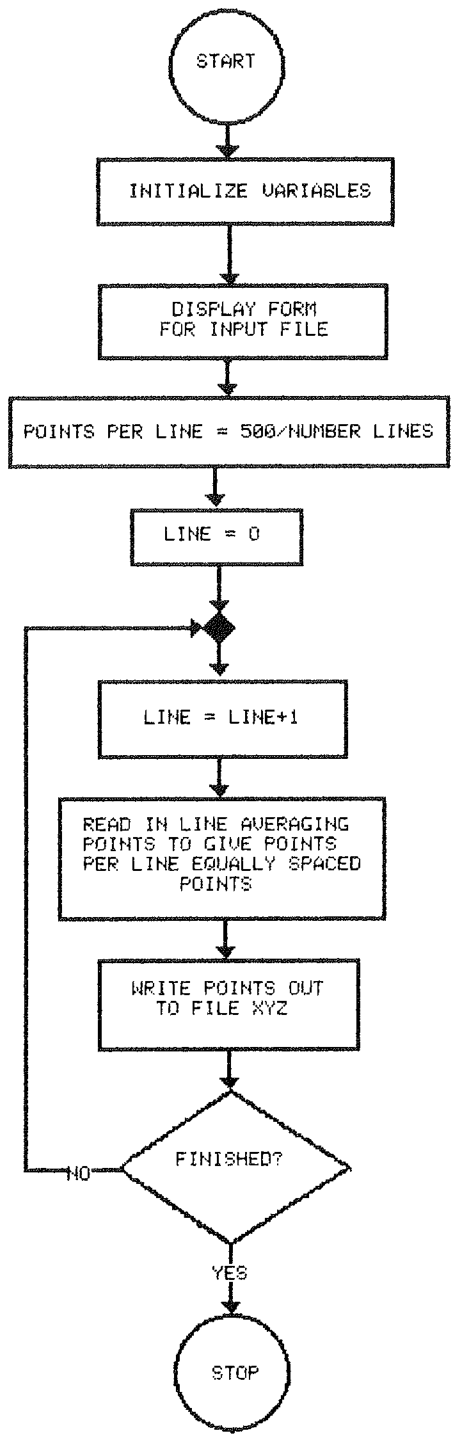


Figure C7. Flowchart,
ACAD_XYZ

TEXT STRING
INPUT FILE NAME (-1 TO STOP) GROUPA
ENTER INPUT FILE NAME AND CONFIRM

ACAD_XYZ-1

LINE 1
LINE 2
LINE 3
LINE 4

ACAD_XYZ-2

724696.31250000	2931076.75000000	-43.29999924
724701.12500000	2931071.75000000	-43.50000000
724707.81250000	2931069.75000000	-42.90000153
724715.31250000	2931067.00000000	-42.50000000
724719.75000000	2931065.50000000	-42.70000076
724731.87500000	2931063.00000000	-44.00000000
724736.31250000	2931061.25000000	-44.29999924
724745.75000000	2931058.25000000	-45.29999924
724755.43750000	2931056.50000000	-44.10000229
724764.31250000	2931055.00000000	-43.79999924
724774.56250000	2931054.50000000	-43.40000153
724777.43750000	2931054.25000000	-44.10000229
724780.25000000	2931054.00000000	-43.90000153

ACAD_XYZ-3

Figure C8. Sample run, ACAD_XYZ

ADJUST_DEPTHS_FOR_TIDE

Purpose

ADJUST_DEPTHS_FOR_TIDE is a program that removes tidal trends from data to reference it to a fixed elevation (such as mean sea level or mean lower low water). It accepts tide values based on time of day and removes the tide value corresponding to the particular data value's time of acquisition.

Input files

NAME.0001 (where .0001 is line number)

Output files

NAME.0001 (where .0001 is line number)

External devices

NONE

Modules called

PROBLEM, GET_NAME_OF_FILE, INITMENU, DRAWMENU, MESSAGESTACK, DRAWFORM

Theory of operation

The program begins by initializing pointers. It then displays the data file input form by calling DRAWFORM with a first argument of INFCOMMANDSTR. Then it asks which of the three tide input methods to use: (1) fixed tide, (2) step tide, or (3) linearly interpolated tide. This is done by calling DRAWMENU with METHOD returned as 1, 2, or 3. Then a tide entry form is produced, if the method was 1, by calling DRAWFORM with the first argument MADCOMMANDSTR. If the method was 2 or 3, a series of tide versus time input forms are produced by calls to DRAWFORM with the first argument of ADCOMMANDSTR. The tide values are stored in READING and the tides in TIMETIDE. The flowchart is shown in Figure C9, and a sample run is shown in Figure C10.

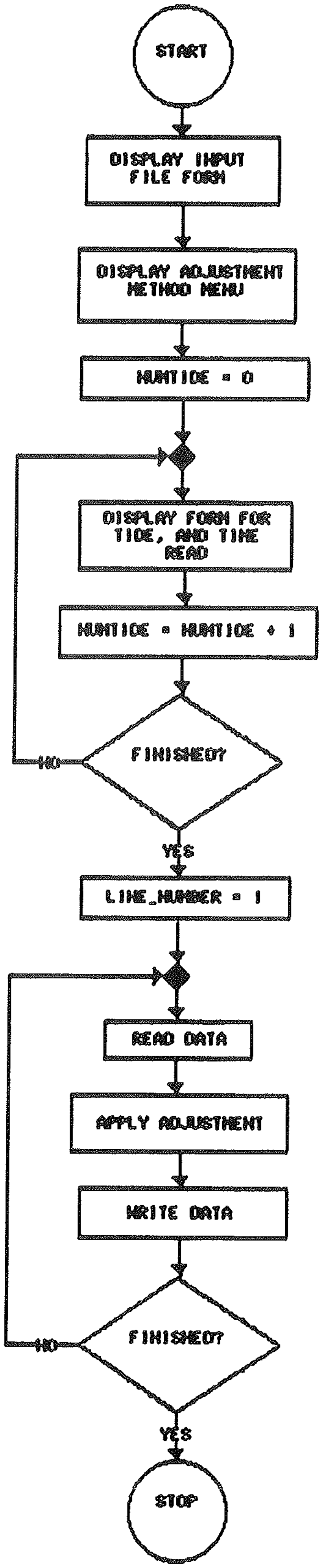


Figure C9. Flowchart, ADJUST_DEPTHS_FOR_TIDE

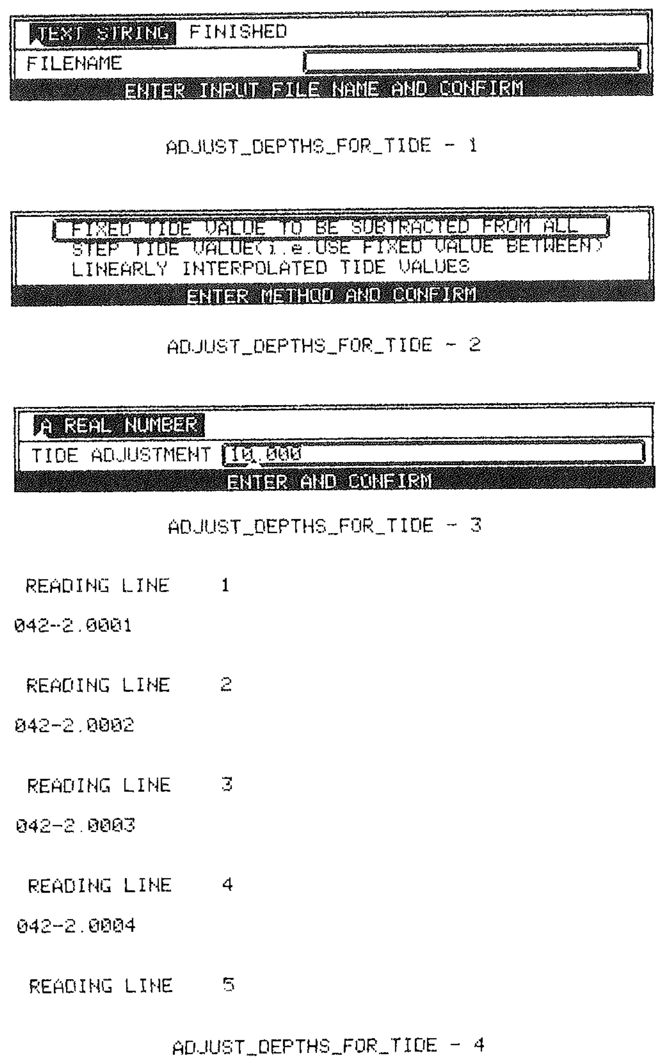


Figure C10. Sample run, ADJUST_DEPTHS_FOR_TIDE

CONFIGURE

Purpose

CONFIGURE is a program that configures the surveying multiprogram operation. It reads in site information from disk file and displays forms for input of test specific information.

Input files

NAME.LINE, NAME.EQUI, DEFAULT, DEFAULT.EQUI, NAME.SPOT, NAME.PLAN, DEFAULT.PLAN, BAR_ID.LIST

Output files

DEFAULT, DEFAULT.EQUI

External devices

NONE

Modules called

LOUISVILLE_STYLE_SURVEY, FIND_SPOT_SURVEY, SECTION_LINE_SURVEY, WORKING_CODE_SL, POLREC, A2, PROBLEM, INITMENU, DRAWMENU, MESSAGESTACK, DRAWFORM, GET_NAME_OF_FILE, GET_BAR_ID

Theory of operation

CONFIGURE is not a stand-alone program. It is a task which is spawned by SURVEY. Its purpose is to come into memory before any of the real-time survey programs are spawned, accept test parameters, then die, thereby freeing memory. It first displays the survey type hardware configuration menu. It then reads in section lines, channel coordinates, plotter type, etc. Depending on the type of survey, it displays applicable forms. All data input is stored in a common block of memory, and the program dies, freeing memory. The flowchart is shown in Figure C11.

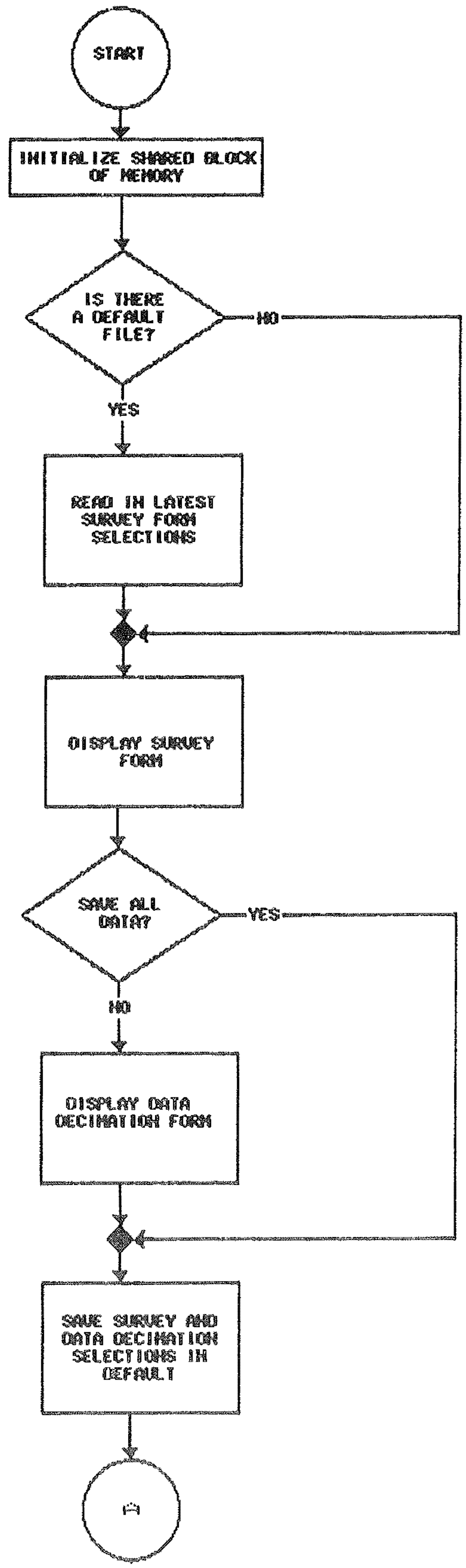


Figure C11. Flowchart, CONFIGURE (Continued)

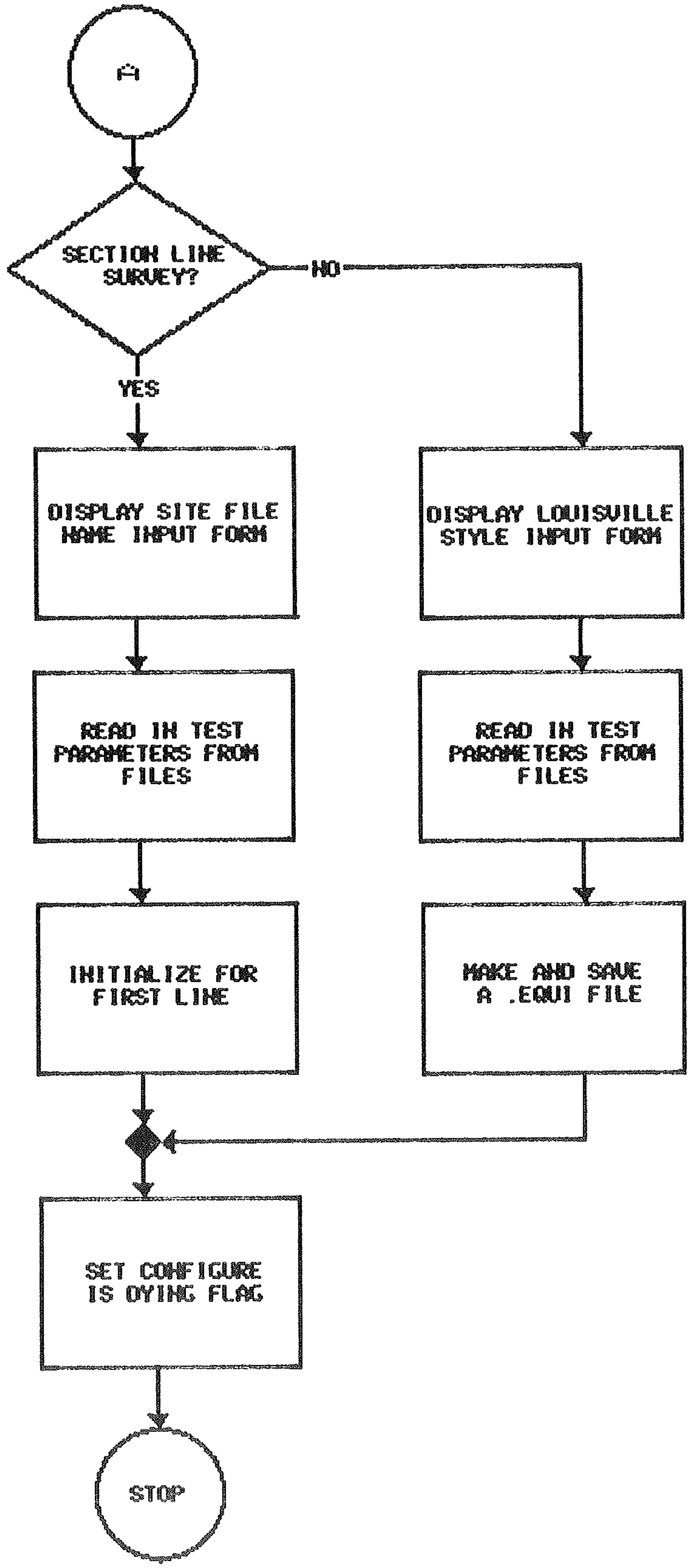


Figure C11. (Concluded)

CONTOUR_PLOT

Purpose

CONTOUR_PLOT runs after PLAN_VIEW to lay contours on top of the plan view.

Input files

TEMPCONTOUR, POSI

Output files

templ, temp2

External devices

Plotter

Modules called

IND, COEF, LTEST, STEST, CONPLT, FLIN, CONZL, DERIV, POLY, REFHUL, READIN, GETXYZ, MINMAX, SHELL, PUTXYZ, TMESH2, TMESH3, PLYTR2, TRIANG, CONHUL, PLOTSC, PLOTROT, TRANSPOSE_ORIGIN, SYMBOL ROTSC, SYMBOLROT, PA, PR, IN, READII

Theory of operation

The program begins by reading in the rotation, scaling, and alignment information from TEMPCONTOUR. Then the number of contour levels and beginning and ending levels are read in. Next the present paper origin from PLAN_VIEW is read in (from file POSI). Then data are read in, contoured, and plotted. Templ and temp2 are used for temporary storage of partial derivatives (Z with respect to X, Y, XX, YY, and XY) and of data. The plot is anotated with a, b, c...to delineate the contours. The flowchart is shown in Figure C12.

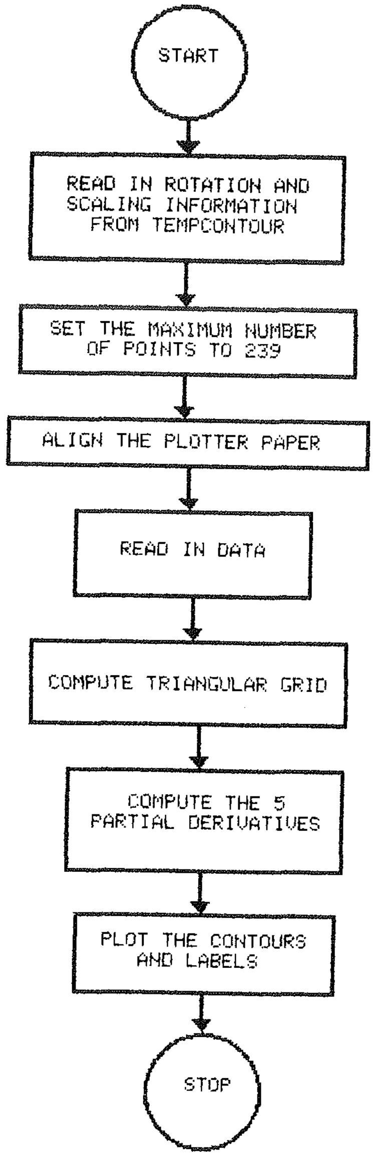


Figure C12. Flowchart, CONTOUR_PLOT

CONVERT_DATA

Purpose

CONVERT_DATA takes acquired depth data from Grid* internal format and converts it to an ASCII distance along the line versus depth format. The file created conforms to the requirements for input to a Harris** program.

Input files

NAME.LINE, NAME.0001 (where .0001 is the line number)

Output files

NAME.HARR

External devices

NONE

Modules called

POLREC, PROBLEM, INITMENU, DRAWMENU, MESSAGESTACK, DRAWFORM, GET_NAME_OF_FILE

Theory of operation

CONVERT_DATA begins by displaying a file form and accepting the input file name. It then accepts a number of Harris specific values from key: (1) three lines of header, (2) elevation, (3) horizontal scale, and (4) vertical scale. It then reads in the data, converts it to distance along the line versus depth, and writes it out to the output file, NAME.HARR. The flowchart is shown in Figure C13.

---

* Grid is a registered trademark of Grid Systems Corporation.
** Harris is a registered trademark of the Harris Corporation.

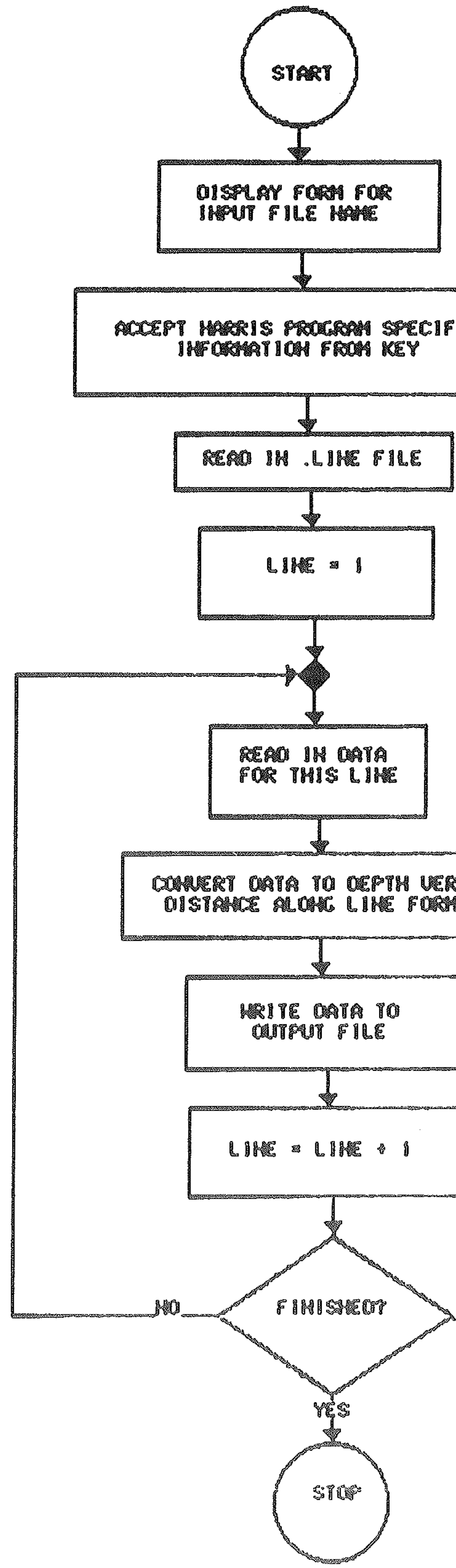


Figure C13. Flowchart, CONVERT_DATA

CONVERT_DATA_LOUISVILLE

Purpose

CONVERT_DATA_LOUISVILLE converts depth data from Grid internal format to easting-northing-depth format for input to a Harris program.

Input files

NAME.EQUI, NAME.0001 (where .0001 is the line number)

Output files

NAME.HARR

External devices

NONE

Modules called

PROBLEM, INITMENU, DRAWMENU, MESSAGESTACK, DRAWFORM, GET_NAME_OF_FILE

Theory of operation

CONVERT_DATA_LOUISVILLE first displays a file form and accepts an input file name. It then reads in the .EQUI file to get the transponder coordinates. Next it reads in the data lines and converts the data to easting-northing-depth format. A file compatible with a Harris program is output. The flowchart is shown in Figure C14.

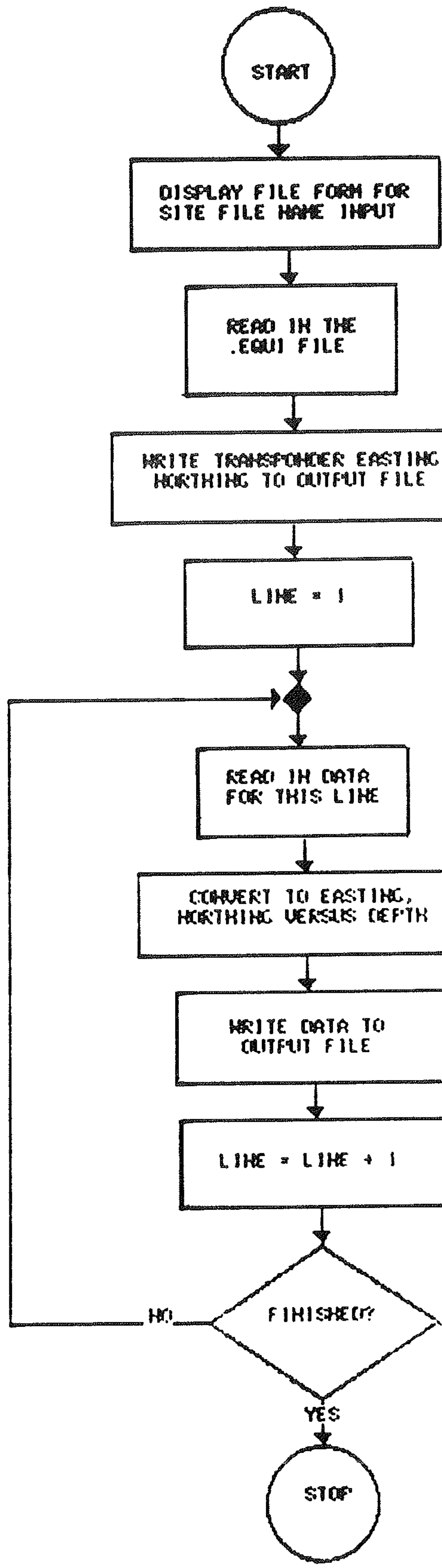


Figure C14. Flowchart,
CONVERT_DATA_LOUISVILLE

CROSS_SECTION_AND_DREDGING

Purpose

CROSS_SECTION_AND_DREDGING displays cross sections and plots them for pre- and postdredge surveys. It computes areas and volumes for each survey with respect to the channel template and with respect to each other.

Input files

DEFAULT.OPTI, NAME.OPTI, NAME.CROS, NAME.0001 (where .0001 is line number)

Output files

QUANTITIES

External devices

Printer, plotter

Modules called

BETW, QUANTITIES, QUANPRNT, ADVANCEPLOT, ENOUGHDATA, PAGE, SL_SLIDE_SLOPE_EQUATION, LAW_OF_COSINES, GET_DEPTHS, DEPTHSCREEN, TEMPLATE_SCREEN, LTS, CHARFEET, SCREENLINE, RDRAW, LTS, DW, SCALE_IT_SCREEN, POLREC, SCALE_IT_PLOT, A2, EQUATION_FROM_SLOPE_AND_1_POINT, EQUATION_PERPENDICULAR_TO_LINE, SCREEN_LINE_PLOT, EQUATION_FROM_2_POINTS, INTERSECTION, POINT_ALONG_A_LINE, PROBLEM, INITMENU, DRAWMENU, MESSAGESTACK, DRAWFORM, GET_NAME_OF_FILE, PRINT_SCREEN

Theory of operation

The program reads in the .LINE file, then begins a hierarchical search for an addition option file. It looks for NAME.OPTI; if this fails it looks for DEFAULT.OPTI. If neither exists, it uses predefined options for stacking top to bottom or bottom to top and predefined axis labels. It then reads in the .CROS file and begins reading in the data lines sequentially from the first requested to the last requested. Templates and block size are computed. Templates are displayed/plotted by calling TEMPLATE_SCREEN; depths are displayed/plotted by calling DEPTHSCREEN. Quantities are computed via QUANTITIES. Quantities are printed and written to a disk file (Quantities) for subsequent processing by calling QUANPRNT. The flowchart is shown in Figure C15, and a sample run is shown in Figure C16.

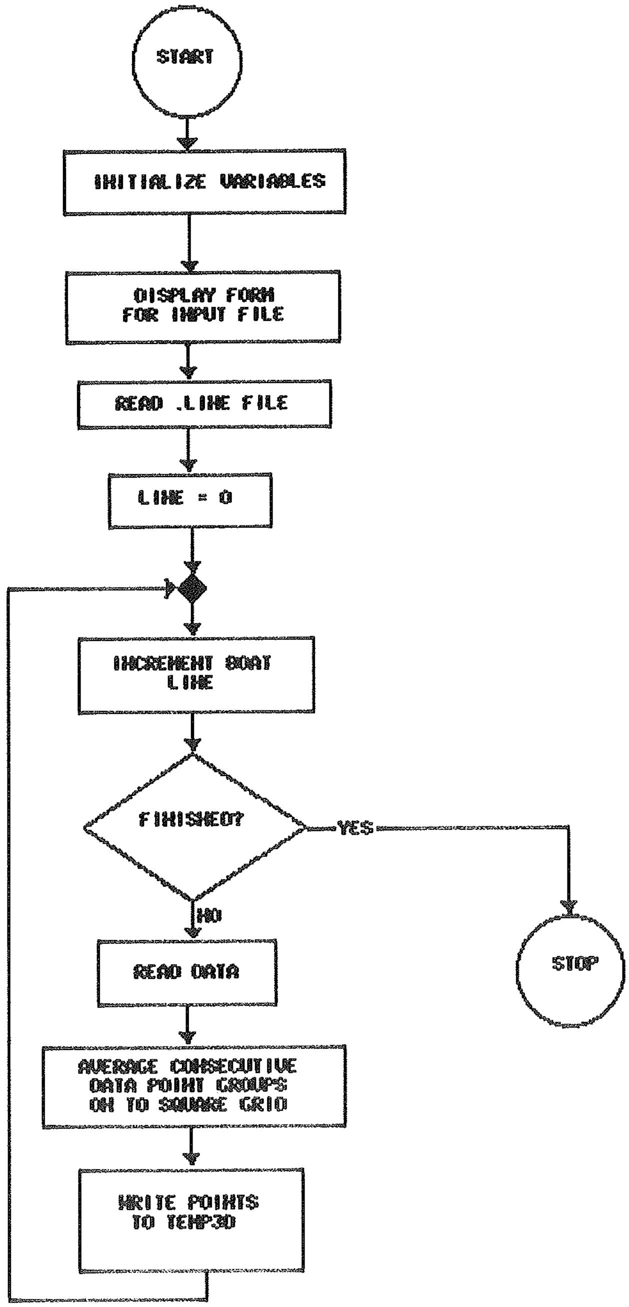
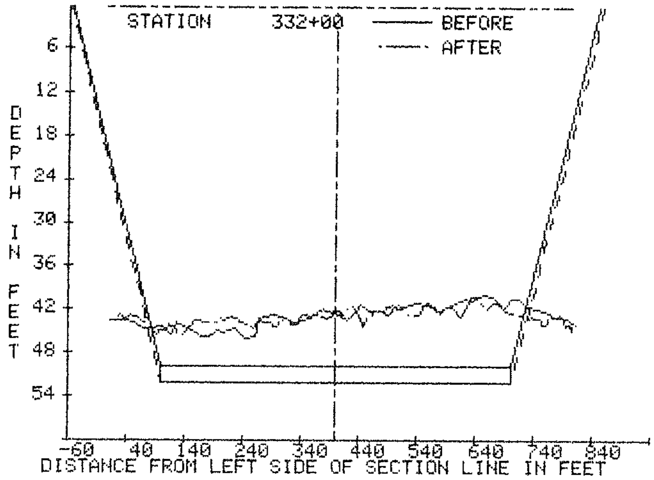
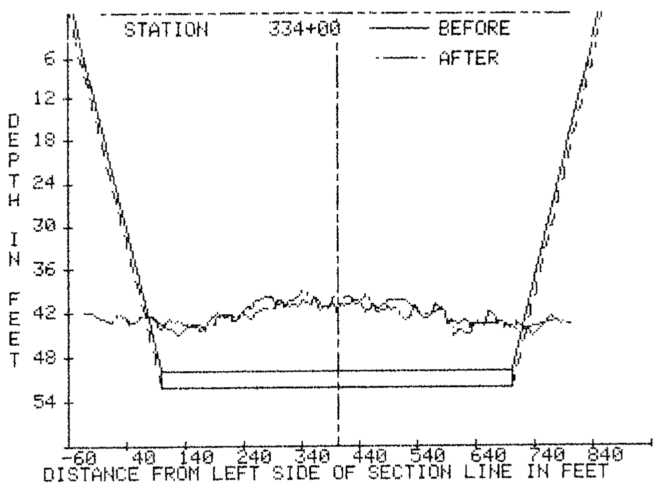


Figure C15. Flowchart,
CROSS_SECTION_AND_DREDGING





CROSS_SECTION_AND_DREDGING-1

Figure C16. Sample run, CROSS_SECTION_AND_DREDGING (Sheet 1 of 5)

BEFORE DREDGE AREA STATIONS 332+00 - 342+00

STATION NUMBER	REQUIRED (SQFEET)	ALLOWABLE SIDE SLOPE (SQFEET)	OVERDEPTH (SQFEET)	REQUIRED+OVERDEP (SQFEET)
332+00	4416.5	93.6	1199.0	5615.5
334+00*	4955.8	68.8	1199.0	6154.8
336+00	3446.6	122.2	1199.0	4645.6
338+00	4447.7	81.0	1199.0	5646.7
340+00	5093.8	121.8	1199.0	6292.8
342+00	6169.3	125.7	1199.0	7368.3

AFTER DREDGE AREA STATIONS 332+00 - 342+00

STATION NUMBER	REQUIRED (SQFEET)	ALLOWABLE SIDE SLOPE (SQFEET)	OVERDEPTH (SQFEET)	REQUIRED+OVERDEP (SQFEET)
332+00	4415.2	91.2	1199.0	5614.2
334+00	4835.1	89.0	1199.0	6034.1
336+00	3603.2	119.5	1199.0	4802.2
338+00	4646.8	91.4	1199.0	5845.8
340+00	5288.9	123.3	1199.0	6487.9
342+00	5969.2	117.2	1199.0	7168.2

CREDIT DREDGE AREA STATIONS 332+00 - 342+00

STATION NUMBER	REQUIRED (SQFEET)	ALLOWABLE SIDE SLOPE (SQFEET)	OVERDEPTH (SQFEET)	REQUIRED+OVERDEP (SQFEET)
332+00	1.3	2.4	0.0	1.3
334+00*	120.7	-20.1	0.0	120.7
336+00	-156.6	2.7	0.0	-156.6
338+00	-199.2	-10.3	0.0	-199.2
340+00	-195.0	-1.6	0.0	-195.0
342+00	200.1	8.5	0.0	200.1

CROSS_SECTION_AND_DREDGING-2

Figure C16. (Sheet 2 of 5)

BEFORE DREDGE VOLUME BY SECTIONS 332+00 - 342+00

STATION NUMBERS		REQUIRED (YARDS)	ALLOWABLE SIDE SLP (YARDS)	OVERDEPTH (YARDS)	REQUI+OVER (YARDS)
332+00-	334+00	34725.7	601.7	8884.9	43610.5
334+00-	336+00	31053.5	706.0	8862.5	39916.0
336+00-	338+00	29249.0	753.1	8884.9	38133.9
338+00-	340+00	35397.6	752.4	8896.3	44293.8
340+00-	342+00	41782.9	917.9	8895.9	50678.8

AFTER DREDGE VOLUME BY SECTIONS 332+00 - 342+00

STATION NUMBERS		REQUIRED (YARDS)	ALLOWABLE SIDE SLP (YARDS)	OVERDEPTH (YARDS)	REQUI+OVER (YARDS)
332+00-	334+00	34273.5	667.6	8884.9	43158.3
334+00-	336+00	31186.1	770.3	8862.5	40048.6
336+00-	338+00	30567.3	781.1	8884.9	39452.2
338+00-	340+00	36860.1	796.5	8896.3	45756.4
340+00-	342+00	41764.1	892.2	8895.9	50659.9

CREDIT DREDGE VOLUME BY SECTIONS 332+00 - 342+00

STATION NUMBERS		REQUIRED (YARDS)	ALLOWABLE SIDE SLP (YARDS)	OVERDEPTH (YARDS)	REQUI+OVER (YARDS)
332+00-	334+00	452.2	-65.9	0.0	452.2
334+00-	336+00	-132.6	-64.3	0.0	-132.6
336+00-	338+00	-1318.3	-28.0	0.0	-1318.3
338+00-	340+00	-1462.6	-44.1	0.0	-1462.6
340+00-	342+00	18.8	25.7	0.0	18.8

CROSS_SECTION_AND_DREDGING-3

Figure C16. (Sheet 3 of 5)

BEFORE DREDGE VOLUME CUMULATIVE 332+00 - 342+00

STATION NUMBERS		REQUIRED (YARDS)	ALLOWABLE SIDE SLP (YARDS)	OVERDEPTH (YARDS)	REQUI+OVER (YARDS)
332+00-	334+00	34725.7	601.7	8884.9	43610.5
332+00-	336+00	65779.2	1307.7	17747.4	83526.5
332+00-	338+00	95028.2	2060.8	26632.2	121660.4
332+00-	340+00	130425.7	2813.2	35528.5	165954.2
332+00-	342+00	172208.6	3731.1	44424.4	216633.0

AFTER DREDGE VOLUME CUMULATIVE 332+00 - 342+00

STATION NUMBERS		REQUIRED (YARDS)	ALLOWABLE SIDE SLP (YARDS)	OVERDEPTH (YARDS)	REQUI+OVER (YARDS)
332+00-	334+00	34273.5	667.6	8884.9	43158.3
332+00-	336+00	65459.6	1437.9	17747.4	83207.0
332+00-	338+00	96026.9	2219.0	26632.2	122659.1
332+00-	340+00	132887.1	3015.5	35528.5	168415.5
332+00-	342+00	174651.2	3907.7	44424.4	219075.5

CREDIT DREDGE VOLUME CUMULATIVE 332+00 - 342+00

STATION NUMBERS		REQUIRED (YARDS)	ALLOWABLE SIDE SLP (YARDS)	OVERDEPTH (YARDS)	REQUI+OVER (YARDS)
332+00-	334+00	452.2	-65.9	0.0	452.2
332+00-	336+00	319.6	-130.2	0.0	319.5
332+00-	338+00	-998.8	-158.2	0.0	-998.8
332+00-	340+00	-2461.3	-202.3	0.0	-2461.3
332+00-	342+00	-2442.5	-176.5	0.0	-2442.5

CROSS_SECTION_AND_DREDGING-4

Figure C16. (Sheet 4 of 5)

PRE- POST DREDGE AREA COMPARISON 332+00 - 342+00

STATION NUMBER	GROSS REMOVED (SQFEET)	GROSS ADDED (SQFEET)	GROSS TOTAL (SQFEET)	LIMITS (FEET->FEET)	
332+00	355.0	328.6	26.4	29.->	784.
334+00	320.7	182.5	138.2	87.->	728.
336+00	166.9	340.2	-173.2	43.->	736.
338+00	111.7	361.2	-249.5	24.->	718.
340+00	230.1	394.6	-164.5	37.->	786.
342+00	401.8	134.2	267.6	63.->	762.

PRE- POST- DREDGE VOLUME COMPARISON 332+00 - 342+00

STATION NUMBERS		GROSS REMOVED (YARDS)	GROSS ADDED (YARDS)	GROSS TOTAL (YARDS)
332+00-	334+00	2503.7	1893.7	610.0
334+00-	336+00	1802.3	1931.6	-129.4
336+00-	338+00	1032.3	2598.7	-1566.4
338+00-	340+00	1267.9	2804.2	-1536.2
340+00-	342+00	2344.0	1961.9	382.1

PRE- POST- DREDGE VOLUME CUMULATIVE 332+00 - 342+00

STATION NUMBERS		GROSS REMOVED (YARDS)	GROSS ADDED (YARDS)	GROSS TOTAL (YARDS)
332+00-	334+00	2503.7	1893.7	610.0
332+00-	336+00	4305.9	3825.3	480.6
332+00-	338+00	5338.3	6424.0	-1085.7
332+00-	340+00	6606.2	9228.2	-2622.0
332+00-	342+00	8950.2	11190.0	-2239.8

CROSS_SECTION_AND_DREDGING-5

DISTANCE FROM 332.00 TO 334.00 IS 200.08
DISTANCE FROM 334.00 TO 336.00 IS 199.57
DISTANCE FROM 336.00 TO 338.00 IS 200.08
DISTANCE FROM 338.00 TO 340.00 IS 200.33
DISTANCE FROM 340.00 TO 342.00 IS 200.32

CROSS_SECTION_AND_DREDGING-6

Figure C16. (Sheet 5 of 5)

CROSS_SECTION_DESCRIPTION

Purpose

CROSS_SECTION_DESCRIPTION provides a friendly environment for entering information necessary for creating cross section plots.

Input files

NAME.CROS

Output files

NAME.CROS

External devices

NONE

Modules called

PROBLEM, INITMENU, DRAWMENU, MESSAGESTACK, DRAWFORM, GET_NAME_OF_FILE

Theory of operation

CROSS_SECTION_DESCRIPTION reads in current plot parameters, fills a form with these values, and displays the form. It then accepts changes and outputs a .CROS file. A flowchart is shown in Figure C17, and a sample run is shown in Figure C18.

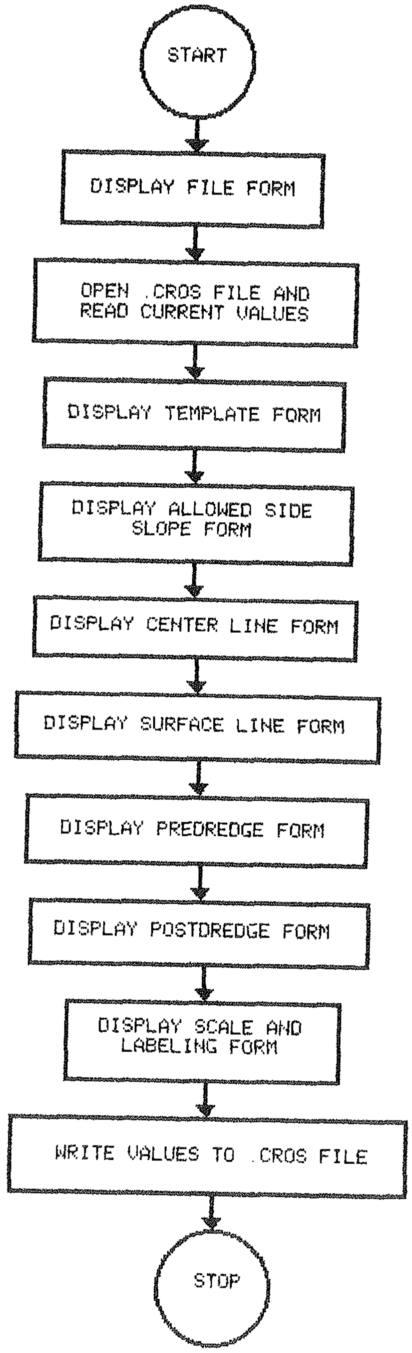


Figure C17. Flowchart, CROSS_SECTION_DESCRIPTION

```
********************************************
CODE RETURN CONFIRMS A FORM
ESCAPE (NOT CODE ESCAPE) ALLOWS ESCAPING
********************************************
IN THE FORMS THAT FOLLOW IF THE
FIRST ITEM IS CHOSEN TO BE
DO NOT PLOT
THEN THE REST OF THE ITEMS MAY
BE IGNORED (NOT FILLED IN)
********************************************
```

```
DO NOT PLOT 1 2 3 4 5 6 7 8
CARRIAGE SLOT FOR PEN  1
LINE TYPE              SOLID
DASH SIZE              MEDIUM
ENTER INFORMATION ABOUT TEMPLATE LINES & CONFIRM
```

CROSS_SECTION_DESCRIPTION - 1

```
DO NOT PLOT 1 2 3 4 5 6 7 8
CARRIAGE SLOT FOR PEN  1
LINE TYPE              DASH
DASH SIZE              SMALL
ENTER INFORMATION ABOUT ALLOWED SIDE SLOPE & CONFIRM
```

CROSS_SECTION_DESCRIPTION - 2

```
DO NOT PLOT 1 2 3 4 5 6 7 8
CARRIAGE SLOT FOR PEN  1
LINE TYPE              CENTER LINE
DASH SIZE              LARGE
ENTER INFORMATION ABOUT CENTER LINE & CONFIRM
```

CROSS_SECTION_DESCRIPTION - 3

```
DO NOT PLOT 1 2 3 4 5 6 7 8
CARRIAGE SLOT FOR PEN  1
LINE TYPE              6
DASH SIZE              MEDIUM
ENTER INFORMATION ABOUT SURFACE LINE & CONFIRM
```

CROSS_SECTION_DESCRIPTION - 4

Figure C18. Sample run, CROSS_SECTION_DESCRIPTION (Continued)

```
DO NOT PLOT  1  2  3  4  5  6  7  8
CARRIAGE SLOT FOR PEN  DO NOT PLOT
LINE TYPE              SOLID
DASH SIZE              MEDIUM
ENTER INFORMATION ABOUT PREDREDGE DEPTH & CONFIRM
```

CROSS_SECTION_DESCRIPTION - 5

```
DO NOT PLOT  1  2  3  4  5  6  7  8
CARRIAGE SLOT FOR PEN  1
LINE TYPE              4
DASH SIZE              MEDIUM
ENTER INFORMATION ABOUT POSTDREDGE DEPTH & CONFIRM
```

CROSS_SECTION_DESCRIPTION - 6

```
SCREEN ONLY  SCREEN & PLOTTER  SCREEN & PRINTER  P→
OUTPUT FILE NAME           042-2
CROSS SECTIONS             SCREEN ONLY
PLOTTER TYPE               LARGE SHEET
PLOTTER PAPER ALIGNMENT    USE DEFAULT
QUANTITIES                 DO NOT COMPUTE
FILL IN CROSS SECTION FORM AND CONFIRM
```

CROSS_SECTION_DESCRIPTION - 7

```
DEPTH IN FEET
VERTICAL SCALE                 15.0000
HORIZONTAL SCALE               200.0000
SMALL SHEET DIMENSION          22.0000
LARGE SHEET DIMENSION          32.0000
CARRIAGE SLOT FOR LETTER PEN   1
LETTER SIZE                    0.0700
FILL IN PLOT DESCRIPTOR FORM & CONFIRM
```

CROSS_SECTION_DESCRIPTION - 8

Figure C18. (Concluded)

DEPTH_DEVICE

Purpose

There are a number of depth equipment programs and one simulation program. One of these programs runs in conjunction with the survey program, determined by the depth sounder used. These programs collect depths, generate event marks on chart paper, and save all desired data.

Input files

NONE (data are passed through a common block of memory)

Output files

NONE (data are passed through a common block of memory)

Equipment required

Supported depth sounder with IEEE488 interface (Institute of Electrical and Electronics (IEEE) Standards Board 1978)*

Modules called

TIME_INTO_TEST, RAYTHEON, INNERSPACE.001 (depth sounder specific), INITDR, READDR, WRITEDR, TERM, PROBLEM

Theory of operation

The depth equipment program is spawned by SURVEY, does some initialization, and waits for PLOT_SCREEN to set a flag before proceeding (PLOT_SCREEN is the last task spawned so all are memory resident at this time). It then begins a tight loop of generating chart event marks, acquiring depths, and putting acquired data in the output buffer. It terminates when PLOT_SCREEN sets the death flag. Flowcharts are shown in Figure C19.

---

* All references cited in this Appendix are included in the References at the end of the main text.

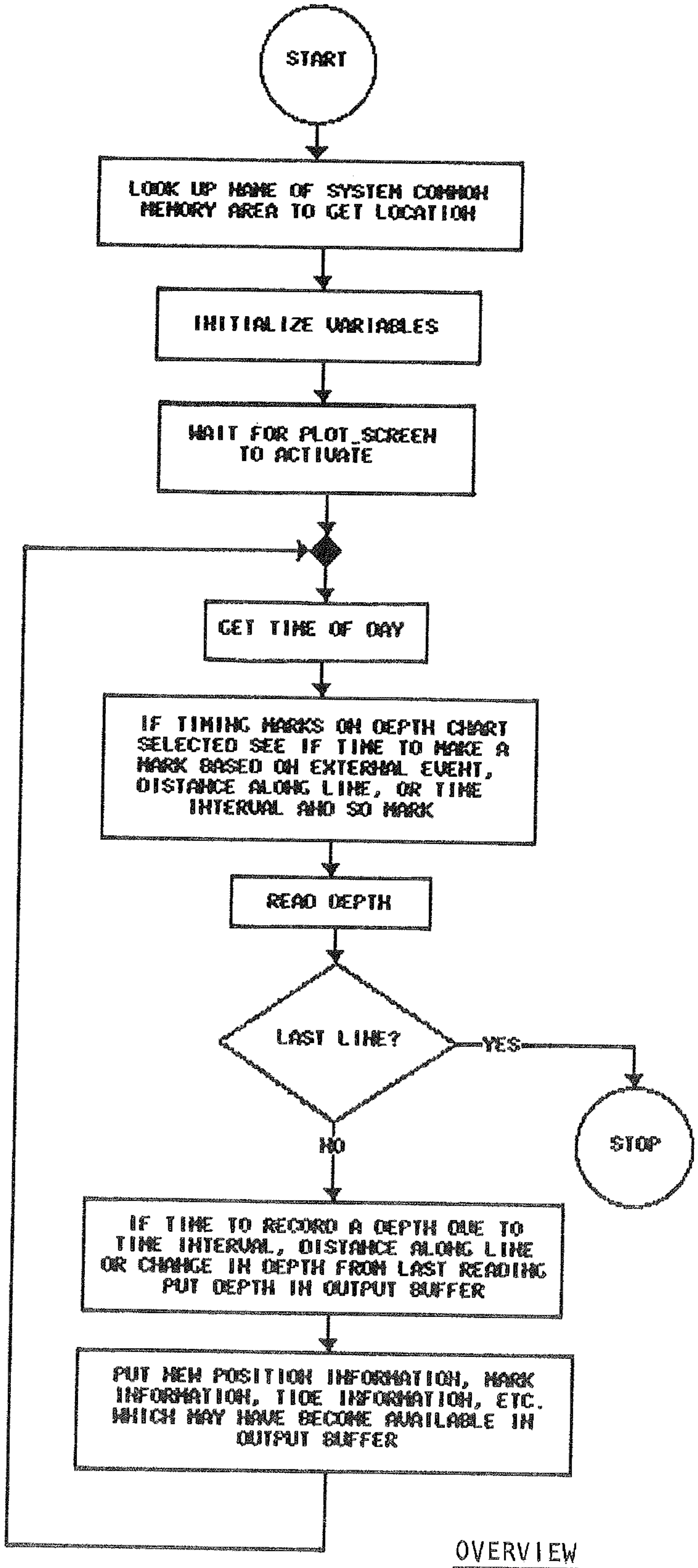


Figure C19. Flowchart, DEPTH_DEVICE
(Sheet 1 of 5)

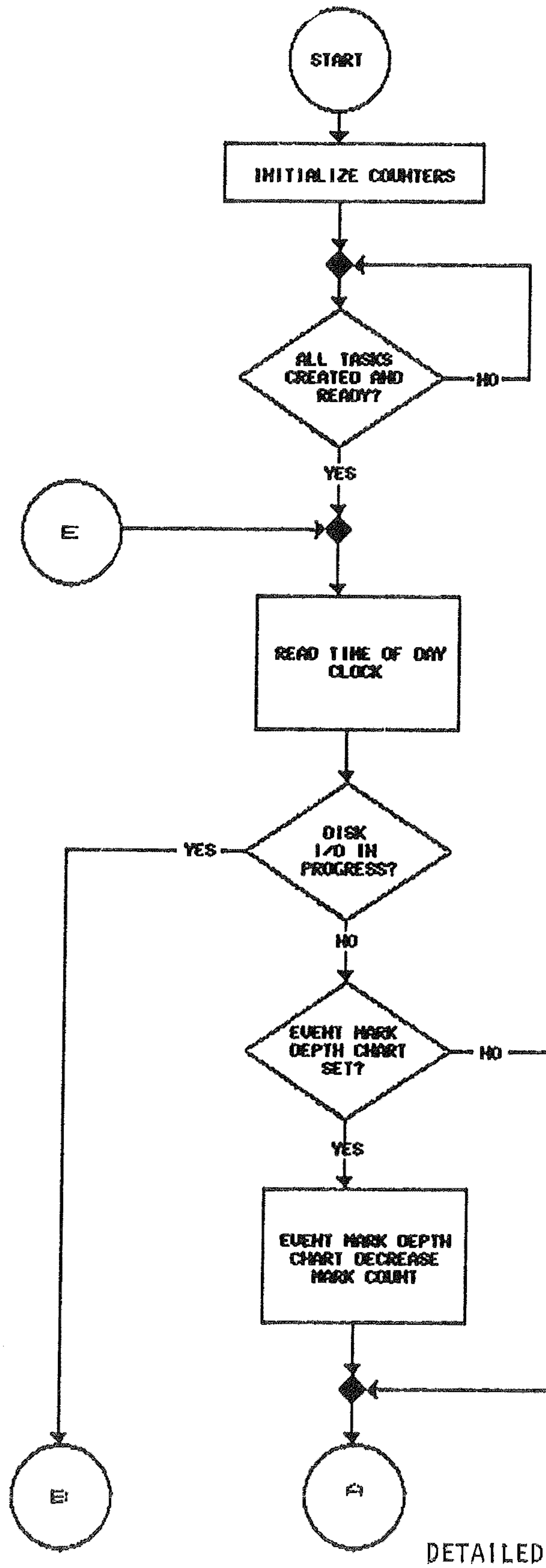


Figure C19. (Sheet 2 of 5)

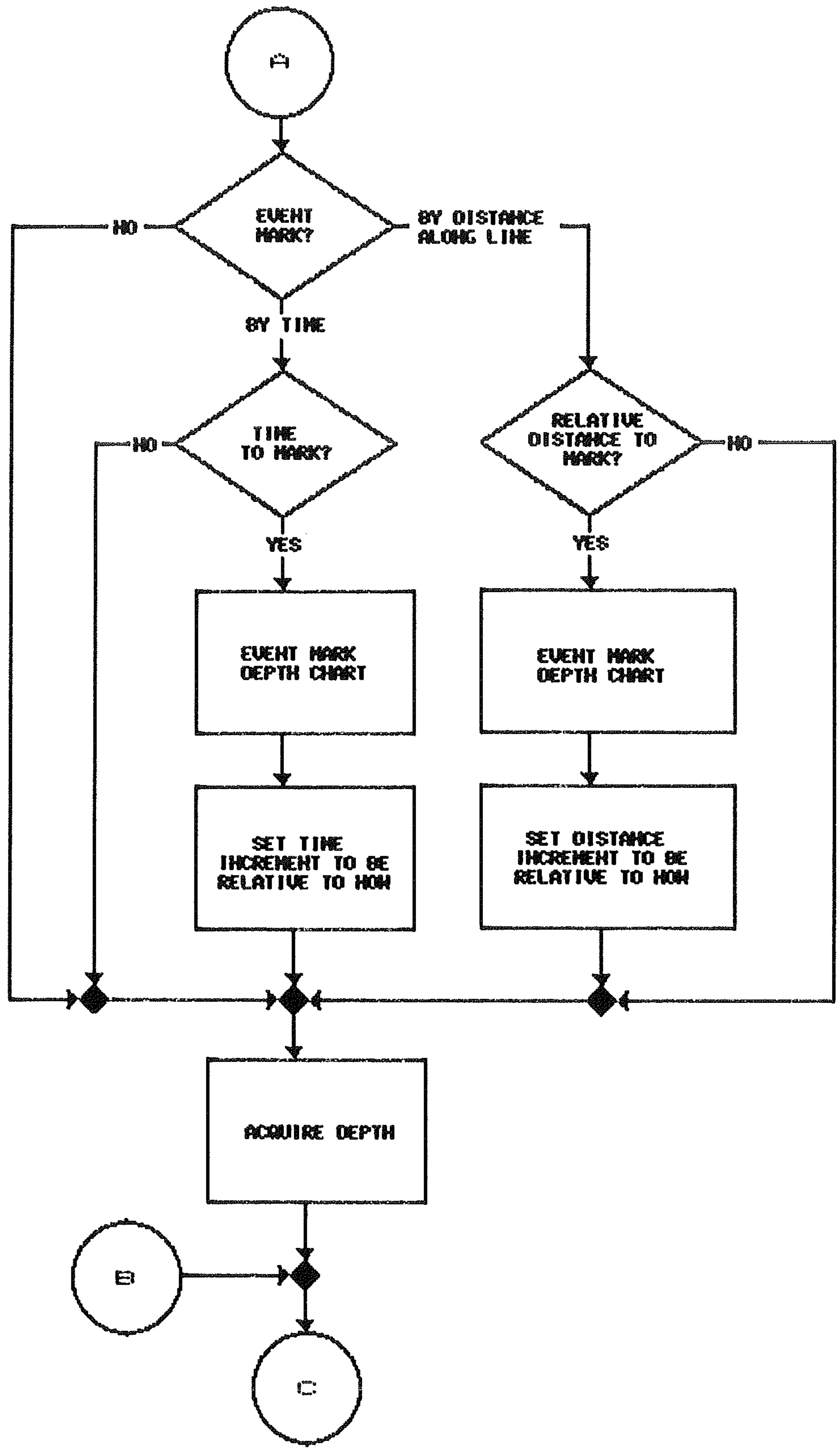


Figure C19. (Sheet 3 of 5)

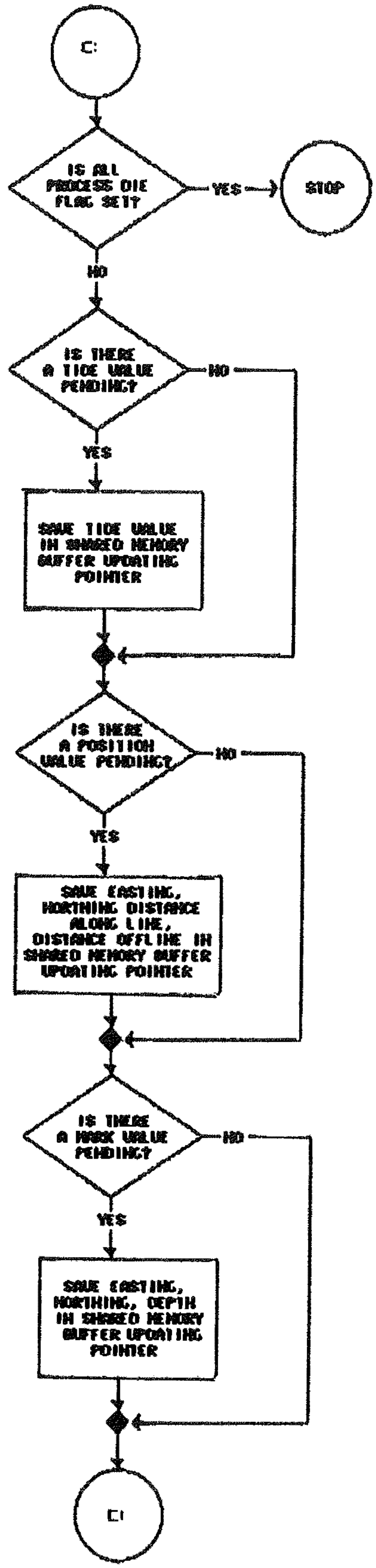


Figure C19. (Sheet 4 of 5)

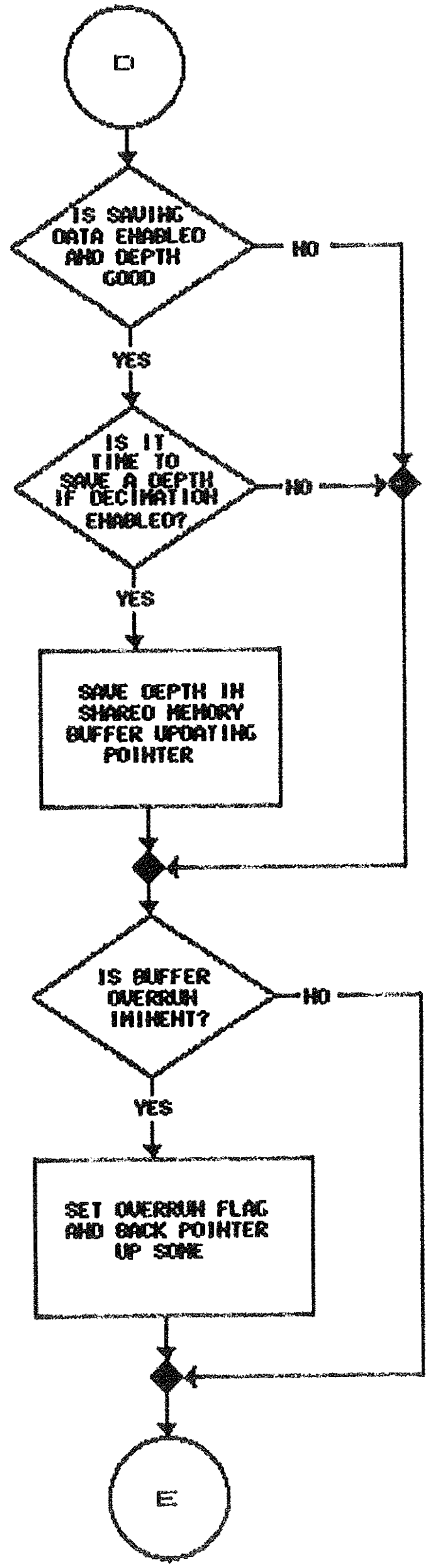


Figure C19. (Sheet 5 of 5)

DEPTH_SIMULATE

Purpose

DEPTH_SIMULATE is a task in the survey package spawned by the task SURVEY. It is used to provide simulated depth data when a depth sounder is not attached.

Input files

NONE (data are passed through system common memory)

Output files

NONE (data are passed through system common memory)

External devices

NONE

Modules called

TIME_INTO_TEST, PROBLEM

Theory of operation

DEPTH_SIMULATE initializes parameters then waits for PLOT_SCREEN to set a flag indicating all tasks are running. It then begins a loop generating a point approximately every 0.1 sec. The program saves any tides that have become available in the common buffer. It then checks to see if it is time to save a depth based on time interval, distance along the line interval, or change in depth since last reading. After saving any depths, it tests to see if any marks have become available and saves them. It then reloops. The flowchart is shown in Figure C20.

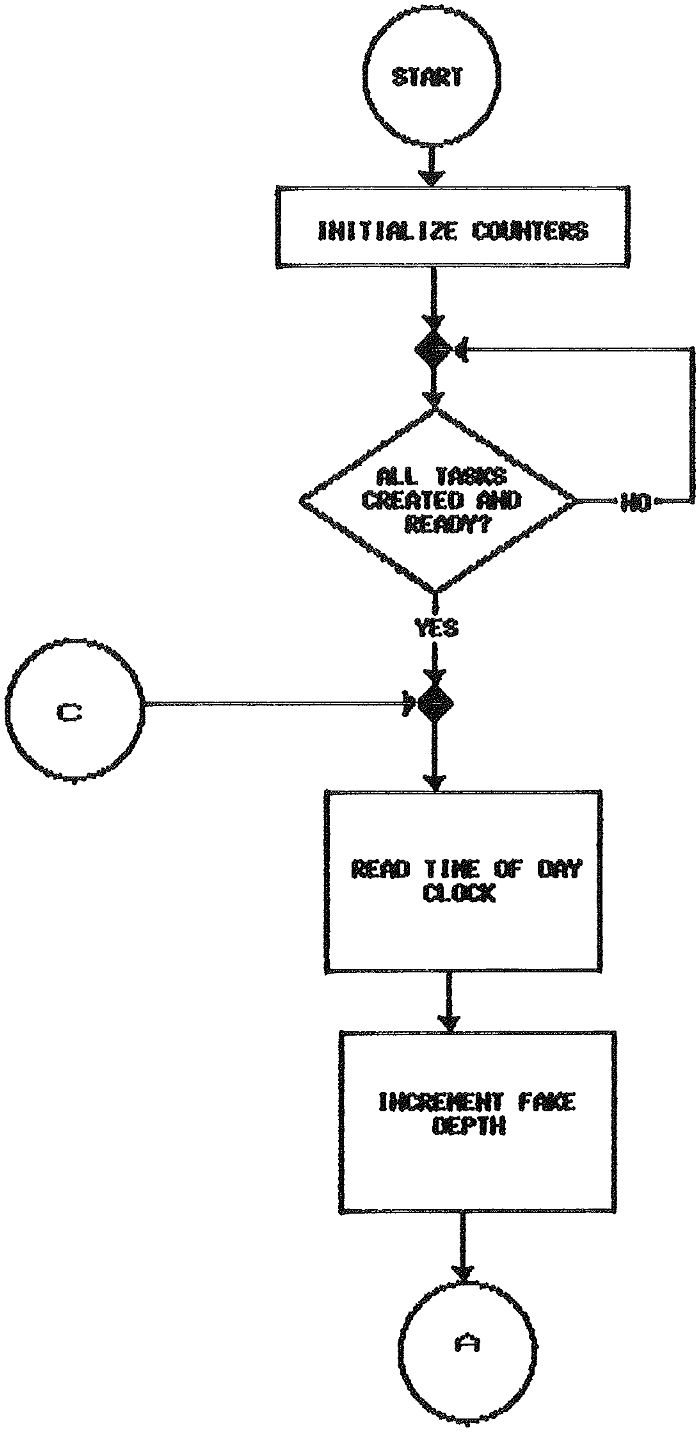


Figure C20. Flowchart, DEPTH_SIMULATE (Sheet 1 of 3)

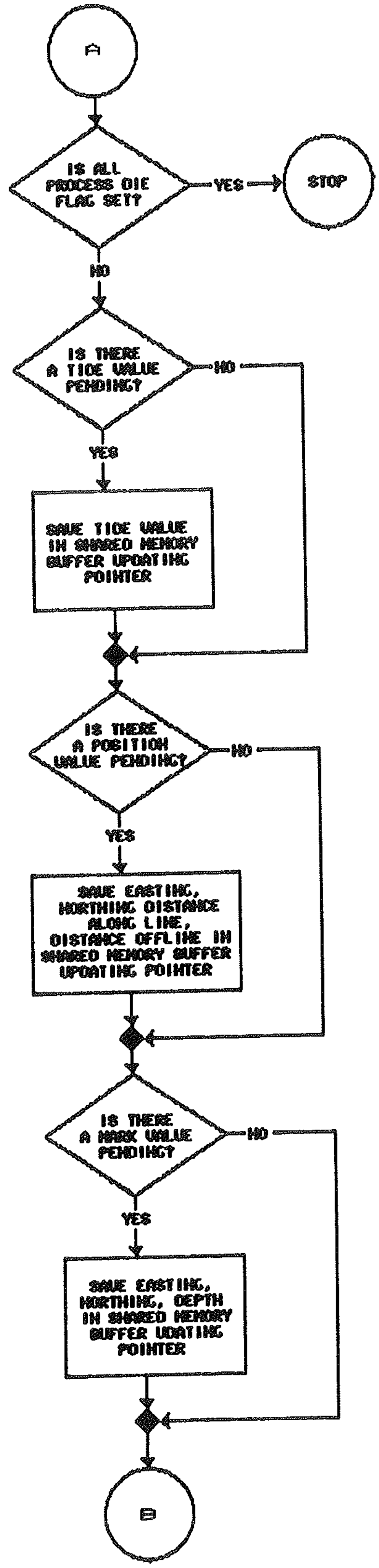


Figure C20. (Sheet 2 of 3)

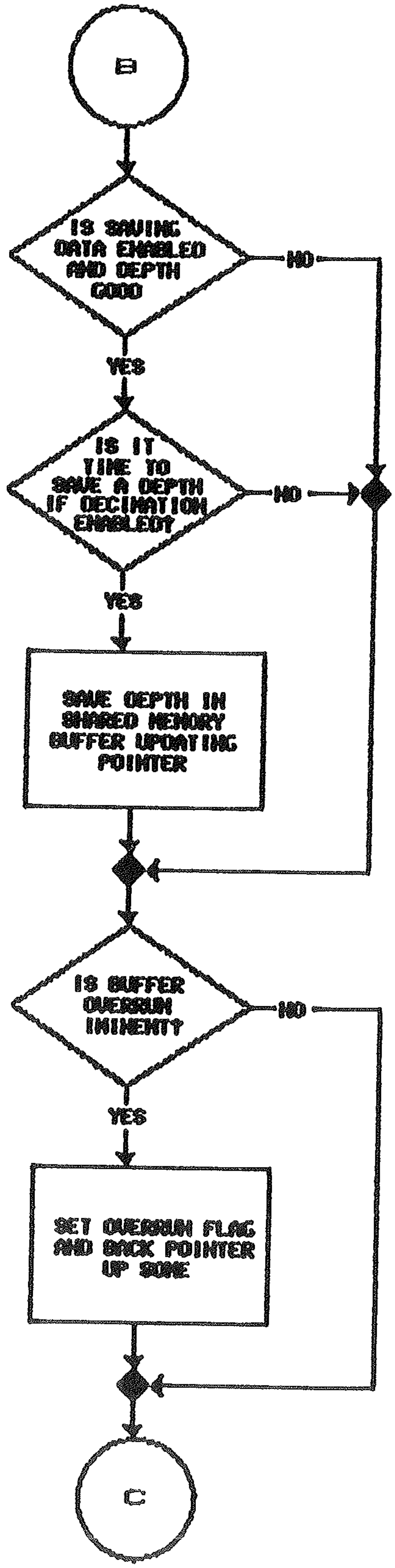


Figure C20. (Sheet 3 of 3)

DIGITIZE_DEPTH_CHART

Purpose

To acquire data from a depth chart using a digitizing tablet. Since the boat may be moving at a nonconstant velocity during the line, there is a mode by which nonlinear x-axis data may be entered.

Input files

NAME.LINE

Output files

NAME.0001

External devices

Summagraphics Bit Pad 1 or Microgrid pencil digitizer*

Modules called

PROBLEM, POLREC, A2, INITMENU, DRAWMENU, MESSAGESTACK, DRAWFORM, DISTOFF, GET_NAME_OF_FILE, DIGITIZER, INISER, RDSERI, WRSERI, TERMSE, ROTATE, INISOUND, WRSOUND, TERMSD

Theory of operation

DIGITIZE_DEPTH_CHART reads in the .LINE file. It then has the operator calibrate the chart in digitizer units. It then asks which paper direction corresponds to the left side of the channel, by looking at the section line file to get the direction, then asking if this is correct or reversed. If it is reversed, it exchanges the start and stop of line coordinates and sets the correction flag. Then it accepts data from the digitizer until the end of line is passed or the *E* is pressed. The flowchart is shown in Figure C21, and a sample run is shown in Figure C22.

---

* Bit Pad 1 and Microgrid are trade names of the Summagraphics Corporation.

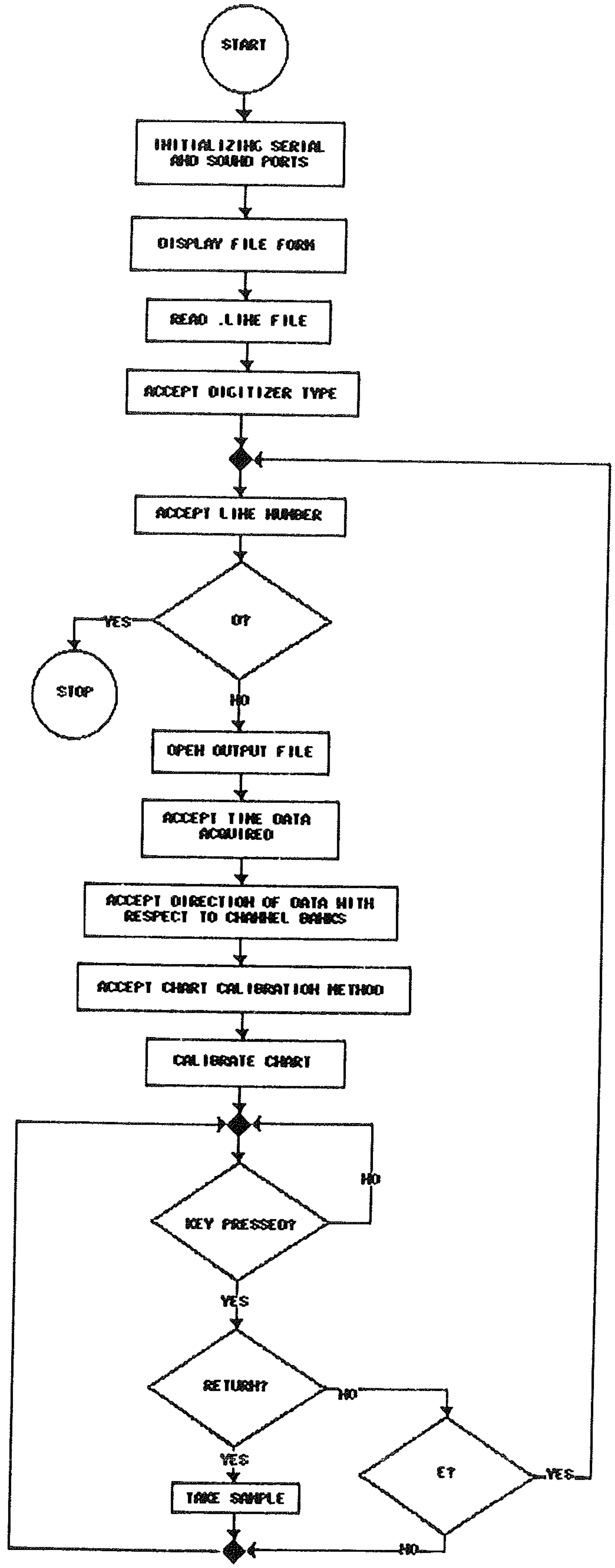


Figure C21. Flowchart,
DIGITIZE_DEPTH_CHART

```
TEXT STRING  FINISHED
FILE NAME          memphis
       ENTER INPUT FILE NAME AND CONFIRM
```

DIGITIZE_DEPTH_CHART-1

```
   SELECT TYPE OF DIGITIZER
 1 FOR SUMMAGRAPHICS BIT PAD 1
 2 FOR SUMMAGRAPHICS MICROGRID
 ENTER 1 OR 2
1

 ENTER LINE NUMBER (0 MEANS FINISHED)
1
memphis.0001
 ENTER DATE THIS LINE WAS ACQUIRED IN FIELD
 USING 2 DIGIT NUMBERS IN FORM MM/DD/YY (EX. 02/03/87
)
11/05/88_
 ENTER TIME THIS LINE WAS ACQUIRED IN FIELD
 USING 2 DIGIT NUMBERS IN FORM HH:MM (EX. 09:07)
12:29
 SECTION LINE FILE SHOWS THIS LINE STARTING ON LEFT S
IDE OF CHANNEL
 DOES LEFT SIDE OF CHANNEL CORRESPOND TO LEFT SIDE OF
 PAPER? Y OR N
Y
```

DIGITIZE_DEPTH_CHART-2

```
 DOES LEFT SIDE OF CHANNEL CORRESPOND TO LEFT SIDE OF
 PAPER? Y OR N
Y
 ENTER METHOD OF DIGITIZING CHART
 1 LINEAR X & Y WITH POSSIBLY TILTED PAPER
 2 NONLINEAR X WITH PAPER ALIGNED FIXED INTERVAL
 3 NONLINEAR X WITH PAPER TILTED FIXED INTERVAL
 4 NONLINEAR X WITH PAPER ALIGNED RANDOM INTERVAL
 5 NONLINEAR X WITH PAPER TILTED RANDOM INTERVAL
2
 ALIGN PAPER UNTIL READING ON THE LEFT
 AND RIGHT SIDES ARE THE SAME.
 WHEN THE READINGS AGREE, ENTER E TO END
    676.00
    727.00
    718.00
    690.00
    670.00
    692.00
    679.00
    741.00
    682.00
    683.00
```

DIGITIZE_DEPTH_CHART-3

Figure C22. Sample run, DIGITIZE_DEPTH_CHART (Continued)

```
    718.00
    690.00
    670.00
    692.00
    679.00
    741.00
    682.00
    683.00
 IN THIS METHOD DIGITIZING OF DATA WILL BEGIN
 AT SOME DISTANCE ALONG THE BASELINE AND WILL PROCEED

 TO THE RIGHT AT A FIXED DISTANCE INTERVAL
 ENTER DISTANCE ALONG THE LINE STARTING POINT
-50
 ENTER DISTANCE ALONG LINE FIXED INTERVAL
50


 TO CALIBRATE THE CHART WE NEED TO DIGITIZE THE CORNE
RS
 THE FIRST POINT SHOULD BE IN THE LOWER LEFT HAND COR
NER
 ENTER DISTANCE ALONG LINE IN FEET FOR FIRST POINT
-200
```

DIGITIZE_DEPTH_CHART-4

```
 ENTER DISTANCE ALONG LINE IN FEET FOR FIRST POINT
-200
 ENTER DEPTH IN FEET FOR FIRST POINT
12
 DIGITIZE FIRST POINT
        312.          718.
 SECOND POINT SHOULD BE IN THE UPPER LEFT
 HAND CORNER AT THE SAME DISTANCE ALONG THE LINE
 AS THE FIRST POINT
 ENTER DEPTH IN FEET FOR SECOND POINT
0
 DIGITIZE POINT
        257.          2328.


 BEGIN DIGITIZING DATA LEFT TO RIGHT
 ENTER E TO END PRIOR TO LINES END
 DIGITIZE ONLY POINTS ON THE CURVE
 AT THE PROPER DISTANCE ALONG THE LINE INTERVAL

 DISTANCE ALONG            DEPTH
 DISTANCE ALONG THE LINE FOR NEXT POINT IS       0.00
```

DIGITIZE_DEPTH_CHART-5

```
 BEGIN DIGITIZING DATA LEFT TO RIGHT
 ENTER E TO END PRIOR TO LINES END
 DIGITIZE ONLY POINTS ON THE CURVE
 AT THE PROPER DISTANCE ALONG THE LINE INTERVAL

 DISTANCE ALONG            DEPTH
 DISTANCE ALONG THE LINE FOR NEXT POINT IS       0.00
          0.00          8.15
 DISTANCE ALONG THE LINE FOR NEXT POINT IS      50.00
         50.00          8.63
 DISTANCE ALONG THE LINE FOR NEXT POINT IS     100.00
        100.00          8.34
 DISTANCE ALONG THE LINE FOR NEXT POINT IS     150.00
        150.00          8.67
 DISTANCE ALONG THE LINE FOR NEXT POINT IS     200.00
        200.00          9.54
 DISTANCE ALONG THE LINE FOR NEXT POINT IS     250.00
        250.00          9.92
 DISTANCE ALONG THE LINE FOR NEXT POINT IS     300.00
        300.00          8.97
 DISTANCE ALONG THE LINE FOR NEXT POINT IS     350.00
```

DIGITIZE_DEPTH_CHART-6

Figure C22. (Concluded)

DIGITIZE_FEATURES

Purpose

DIGITIZE_FEATURES is used to enter coordinate information from maps into .FEAT files on the computer. Shorelines and special features are digitized for later plotting with data.

Input files

NONE

Output files

NAME.FEAT, NAME.LABE

External devices

Summagraphics Bit Pad 1 or Microgrid digitizer

Modules called

PROBLEM, POLREC, A2, INITMENU, DRAWMENU, MESSAGESTACK, DRAWFORM, DISTOFF, GET_NAME_OF_FILE, ROTATE, DIGITIZER, DIGITIZER2, INISOUND, WRSOUND, TERMSO, INISER, RDSERI, WRSERI, TERMSE

Theory of operation

DIGITIZE_FEATURES begins by accepting the output file name and type of digitizer. It then calibrates the map to digitizer units. At this point a loop begins where the user specifies whether the next action will be to enter a feature or a label. Features are saved in the .FEAT file. Labels are saved in the .LABE file. A flowchart is shown in Figure C23, and a sample run is shown in Figure C24.

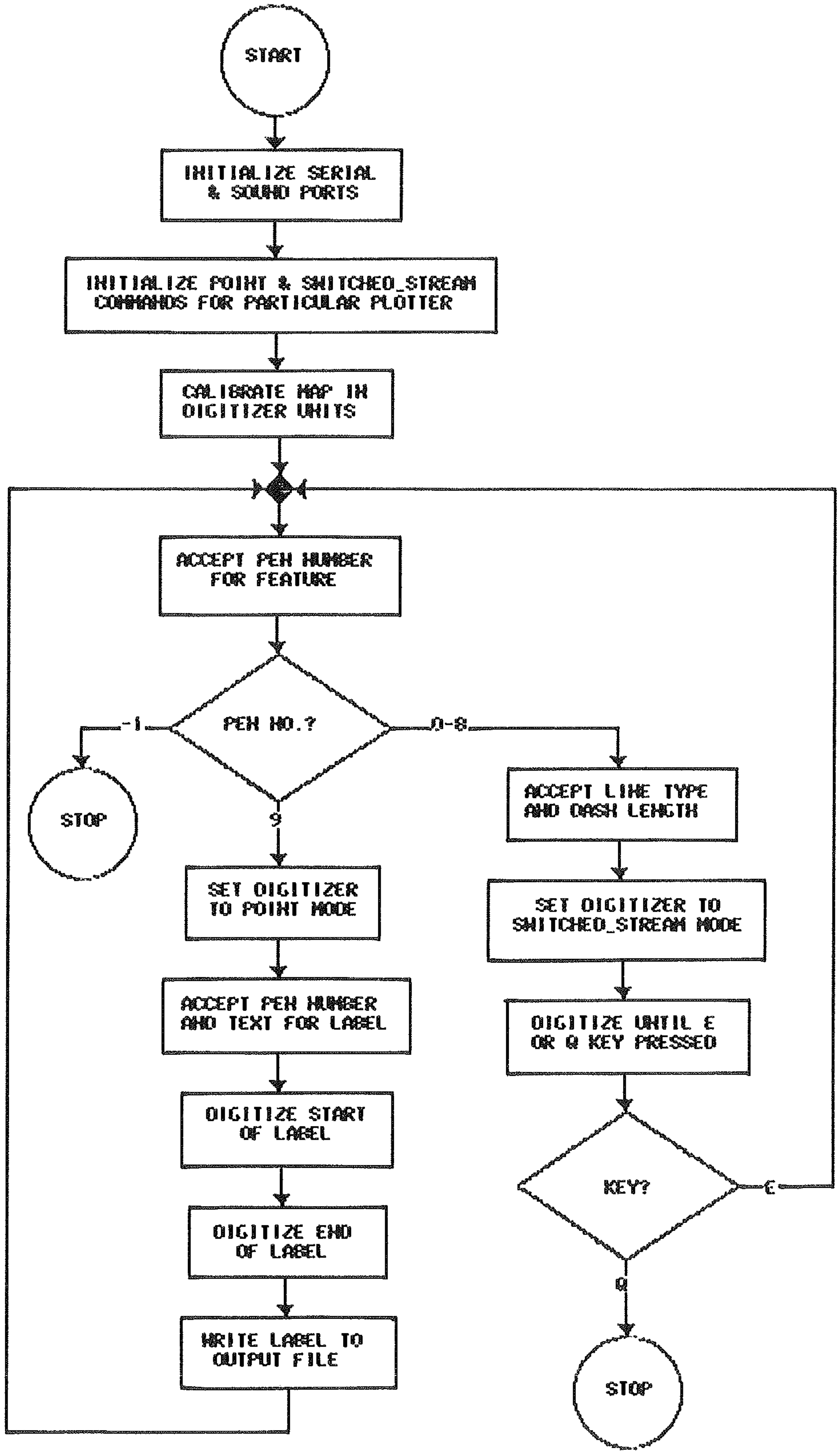


Figure C23. Flowchart, DIGITIZE_FEATURES

```
SUMMAGRAPHICS BIT PAD 1  SUMMAGRAPHICS MICROGRID
DIGITIZER TYPE SUMMAGRAPHICS BIT PAD 1
ENTER AND CONFIRM
```

DIGITIZE_FEATURES-1

```
 ENTER EASTING IN FEET FOR FIRST POINT
750000
 ENTER NORTHING IN FEET FOR FIRST POINT
1000000
 PRESS PEN DOWN TO DIGITIZE FIRST POINT
        348.00          707.00

 SECOND POINT SHOULD BE IN THE LOWER RIGHT HAND CORNE
R
 AT THE SAME NORTHING AS THE FIRST POINT

 ENTER EASTING IN FEET FOR SECOND POINT
780000
 PRESS PEN DOWN TO DIGITIZE SECOND POINT
        2498.00         712.00

 THIRD POINT SHOULD BE IN THE UPPER RIGHT HAND CORNER

 AT THE SAME EASTING AS THE SECOND POINT

 ENTER NORTHING IN FEET FOR THIRD POINT
1500000
```

DIGITIZE_FEATURES-2

```
 AT THE SAME NORTHING AS THE FIRST POINT

 ENTER EASTING IN FEET FOR SECOND POINT
780000
 PRESS PEN DOWN TO DIGITIZE SECOND POINT
        2498.00         712.00

 THIRD POINT SHOULD BE IN THE UPPER RIGHT HAND CORNER

 AT THE SAME EASTING AS THE SECOND POINT

 ENTER NORTHING IN FEET FOR THIRD POINT
1500000
 PRESS PEN DOWN TO DIGITIZE THIRD POINT
        2492.00        2372.00
 ERROR FROM 90 DEGREE ANGLE IS     0.1
 ENTER PEN NUMBER THIS FEATURE (0-8)
OR 9 TO LABEL A FEATURE
OR -1 TO QUIT
1
 ENTER LINE TYPE THIS FEATURE (1-7)
6
 ENTER DASH LENGTH (.5-2.)
1.
```

DIGITIZE_FEATURES-3

```
      761721.94      1138566.88
      761721.88      1137964.38
      760896.19      1115716.88
      760504.44      1105796.88
      760504.44      1105796.88
        -999.00         -999.00
 ENTER PEN NUMBER THIS FEATURE (0-8)
OR 9 TO LABEL A FEATURE
OR -1 TO QUIT
9
ENTER PEN NUMBER FOR LABEL (1-8)
3
ENTER TEXT FOR LABEL
MOBILE & PANAMA CITY SURVEYORS DO IT WITH CLASS
   47
MOBILE & PANAMA CITY SURVEYORS DO IT WITH CLASS
 PRESS PEN DOWN TO DIGITIZE START POINT OF LABEL
      758570.75      1160110.75
 PRESS PEN DOWN TO DIGITIZE FINISH POINT OF LABEL
      767947.56      1290060.50
 ENTER PEN NUMBER THIS FEATURE (0-8)
OR 9 TO LABEL A FEATURE
OR -1 TO QUIT
```

DIGITIZE_FEATURES-4

Figure C24. Sample run,
DIGITIZE_FEATURES

DIGITIZE_PLOTTER

Purpose

DIGITIZE_PLOTTER is a program that digitizes features from the plotter using a digitizing sight instead of a pen. It is a much slower method than using the digitizing tablet with DIGITIZE_FEATURES and should be used only if a digitizing tablet is not available. The digitizing sight is moved to a point on the plot and the space bar is pressed to acquire the point.

Input files

NONE

Output files

NAME.FEAT

External devices

Hewlett-Packard Company plotter with IEEE488 interface

Modules called

PROBLEM, POLREC, A2, INITMENU, DRAWMENU, MESSAGESTACK, DRAWFORM, DISTOFF, GET_NAME_OF_FILE, ROTATE, DIGITIZER

Theory of operation

The program displays a file form for output file name. Then it calibrates the map in plotter units. Next it asks pen number, line type, and dash length for the feature. Finally it accepts points every time the space bar is pressed until *E* or *Q* is pressed. *E* ends this feature, *Q* ends this program.

Special considerations

This program does not allow inputting labels as DIGITIZE_FEATURES does. Labels could easily be added if needed. Figure C25 shows the flowchart, and Figure C26 shows a sample run.

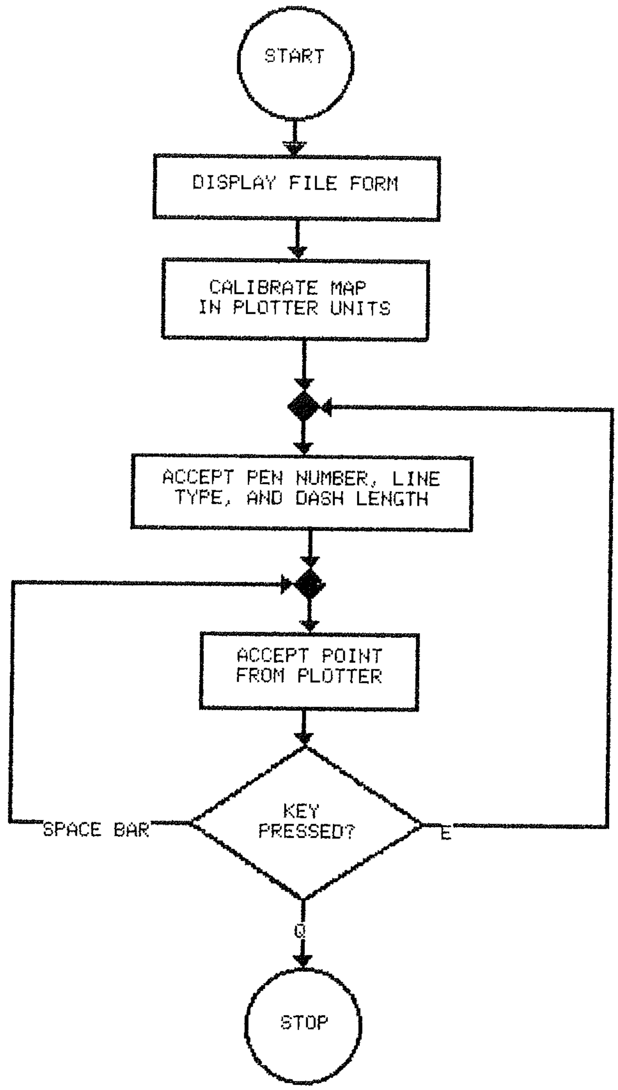


Figure C25. Flowchart,
DIGITIZE_PLOTTER

```
TEXT STRING  FINISHED
FILE NAME              PANAMACITY
       ENTER INPUT FILE NAME AND CONFIRM
```

DIGITIZE_PLOTTER-1

```
 FIRST POINT SHOULD BE IN THE LOWER LEFT HAND CORNER

 ENTER EASTING IN FEET FOR FIRST POINT
850000
 ENTER NORTHING IN FEET FOR FIRST POINT
1200000
 PRESS SPACE BAR TO DIGITIZE FIRST POINT
          21.05           9.02

 SECOND POINT SHOULD BE IN THE LOWER RIGHT HAND CORNE
R
 AT THE SAME NORTHING AS THE FIRST POINT

 ENTER EASTING IN FEET FOR SECOND POINT
860000
 PRESS SPACE BAR TO DIGITIZE SECOND POINT
```

DIGITIZE_PLOTTER-2

```
          21.05           9.02

 SECOND POINT SHOULD BE IN THE LOWER RIGHT HAND CORNE
R
 AT THE SAME NORTHING AS THE FIRST POINT

 ENTER EASTING IN FEET FOR SECOND POINT
860000
 PRESS SPACE BAR TO DIGITIZE SECOND POINT
          15.69           8.99

 THIRD POINT SHOULD BE IN THE UPPER RIGHT HAND CORNER

 AT THE SAME EASTING AS THE SECOND POINT

 ENTER NORTHING IN FEET FOR THIRD POINT
1400000
 PRESS SPACE BAR TO DIGITIZE THIRD POINT
          15.69          13.65
 ERROR FROM 90 DEGREE ANGLE IS -180.3
 THE THREE DIGITIZED POINTS DO NOT FORM A 90 DEGREE A
NGLE
 REDIGITIZE THE THREE POINTS IN ORDER NOW
```

DIGITIZE_PLOTTER-3

Figure C26. Sample run, DIGITIZE_PLOTTER (Continued)

```
 AT THE SAME EASTING AS THE SECOND POINT

 ENTER NORTHING IN FEET FOR THIRD POINT
1400000
 PRESS SPACE BAR TO DIGITIZE THIRD POINT
          15.69           13.65
 ERROR FROM 90 DEGREE ANGLE IS -180.3
 THE THREE DIGITIZED POINTS DO NOT FORM A 90 DEGREE A
NGLE
 REDIGITIZE THE THREE POINTS IN ORDER NOW
          19.16            9.24
          15.20            9.24
          15.20            3.58
 ERROR FROM 90 DEGREE ANGLE IS    0.0



 BEGIN DIGITIZING FEATURE NUMBER     1
 ENTER SPACE TO SAMPLE PLOTTER POSITION
 ENTER E TO END FEATURE
 ENTER Q TO QUIT WHEN FINISHED

 ENTER PEN NUMBER THIS FEATURE (0-8)
1
```

DIGITIZE_PLOTTER-4

Figure C26. (Concluded)

Purpose

EDIT_DATA allows data acquired by survey programs or other means to be graphically edited. Points may be deleted from the existing curve or added beyond the ends (overbank). The program also automatically removes data before the start of line and removes any duplicate times (i.e., the time of data is resolved to the nearest 0.1 sec and it is possible for two depths to be time marked the same as for fast digitizers). It is primarily used to remove irregularities that get through the digitizer's wild point filter (gated mode), such as schools of fish.

Input files

NAME.LINE, NAME.0001 (last four digits are line number)

Output files

NAME.0001 (last four digits are line number)

External devices

NONE

Modules called

SL_SLIDE_SLOPE_EQUATION, LAW_OF_COSINES, SAVEPRESENT, SCREENSAVE, SCREENRESTORE, GET_DEPTHS, DEPTHSCREEN, TEMPLATE_SCREEN, LTS, CHARFEET, SCREENLINE, DW, RDRAW, SCALE_IT_SCREEN, UNSCALE_IT_PIXEL, SCALE_IT_PLOT, POLREC, A2, EQUATION_FROM_SLOPE_AND_1_POINT, EQUATION_PERPENDICULAR_TO_LINE, SCREEN_LINE_PLOT, EQUATION_FROM_2_POINTS, INTERSECTION, POINT_ALONG_A_LINE, INITMENU, DRAWMENU, MESSAGESTACK, DRAWFORM, POINT_ALONG_A_LINE, GET_NAME_OF_FILE, DISTOFF, WINDRAWLINE (Grid Routine)

Theory of operation

The program begins by displaying (via a standard FORTRAN write to device 6--the screen) a list of potential command keys that may be used whenever data is graphically displayed: *A* to add a point, *D* to delete a point, *N* to advance to the next line number, *L* to select any line number, *ESC* to exit the program, and the number and arrow key meanings. The arrow keys are used to move the crosshair left, right, up, and down. The number keys are used to preset the jump increment moved when an arrow key is struck. The jump size formula is 2 (key value 3), where key value is the number printed on the key (except that the 0 key has a value of 10). For example, when beyond the sides of the existing data, key value 1 causes quarter-pixel jumps and key value

8 causes 32 pixel jumps. As the crosshair is moved about, the present depth and distance along the line are numerically displayed; therefore, values of depth and distance along the line may be entered to quarter-pixel length resolution (typically 0.06-ft depth and 1.0-ft distance along the line).

After the key operations are displayed, a form for entering the site name is displayed by a call to DRAWFORM with a first argument of INFCOMMANDSTR. Whether or not a template is to be used is also entered into this form. The NAME.LINE file (where NAME is the site name that was entered) is then opened as device 3 and read.

The line number is set to 1 and the first line is graphically displayed. If template was selected (i.e. ITEMPLATE equals 1) then calls to EQUATION_FROM_2_POINTS and INTERSECTION are made to determine how channel easting, northing coordinates relate to distance along the section line. Next, the data file is opened (NAME.0001) and read. The screen is then scaled based on the template and data; and if ITEMPLATE equals 1, the intersection of the template and the sides of the screen are computed. Then the template and data are displayed. Next, the current value of depth and distance along line are numerically displayed and a crosshair is drawn at this position (by calls to WINDRAWLINE). Then the program waits for a key to be depressed. Execution then branches to the proper code to handle the key value.

If a numeric key is depressed, IJUMP is set to that key's value. If *ESC* is pressed, the program ends. If *N* or *L* is depressed, the value of LINE_NUMBER is updated and the new line is displayed. If *A*, *D*, or an arrow key is depressed, the current mode off operation (1) on the data curve (IONCURVE equals 1) or (2) beyond the ends of the data curve (IONCURVE equals 2) is considered when determining the statement number to which to branch. Up and down arrows and the *A* key have meaning only when beyond the curve.

If a point is to be added (*A*), the direction the line was run (1) left to right (ICORRECT equals 0) or (2) right to left (ICORRECT equals 1) is considered and the present crosshair position is added to the original data. This involves converting the current pixel location to depth versus distance along the line, then assigning time to the depth based on corresponding distance along the line and computing easting and northing corresponding to the distance along the line. The point must be included by moving existing points one position after the time of interest.

The procedure for deleting a point is similar to that of adding, in that direction run must be considered. Otherwise it is simpler in that a point need only be removed from a buffer and the graph replotted.

Cursor keys add the jump index value either to the pixel position or to the buffer index. The data are quickly restored from a saved screen and the crosshairs are superimposed.

Edited screens may be saved or changes may be discarded. Whenever *L*, *N*, or *ESC* is pressed, disposition of the current screen is requested.

Special Considerations

If the profile lines are among those being edited, the template should not be displayed. Since the lines run parallel to the sides of the channel, they will not intersect the sides and forming a template will be impossible (an attempt to do so under these circumstances may crash the system).

Edited files can be distinguished from raw data files by printing their contents. Originally depths and positions are interlaced (raw data). The edit program groups all the depths together and all the ranges together. All programs will work with either layout as time (not order) is used to relate depths to positions.

It is generally best to do any editing desired prior to running other postsurvey programs. This ensures that any bad data points are removed prior to analysis and plotting. The flowchart is shown in Figure C27, and a sample run is shown in Figure C28.

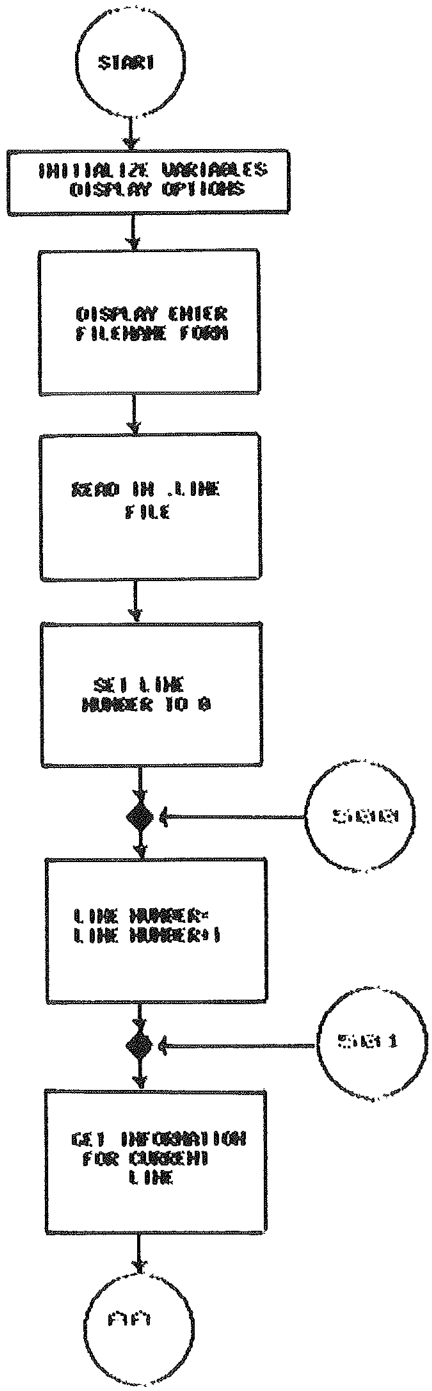


Figure C27. Flowchart,
EDIT_DATA (Sheet 1 of 6)

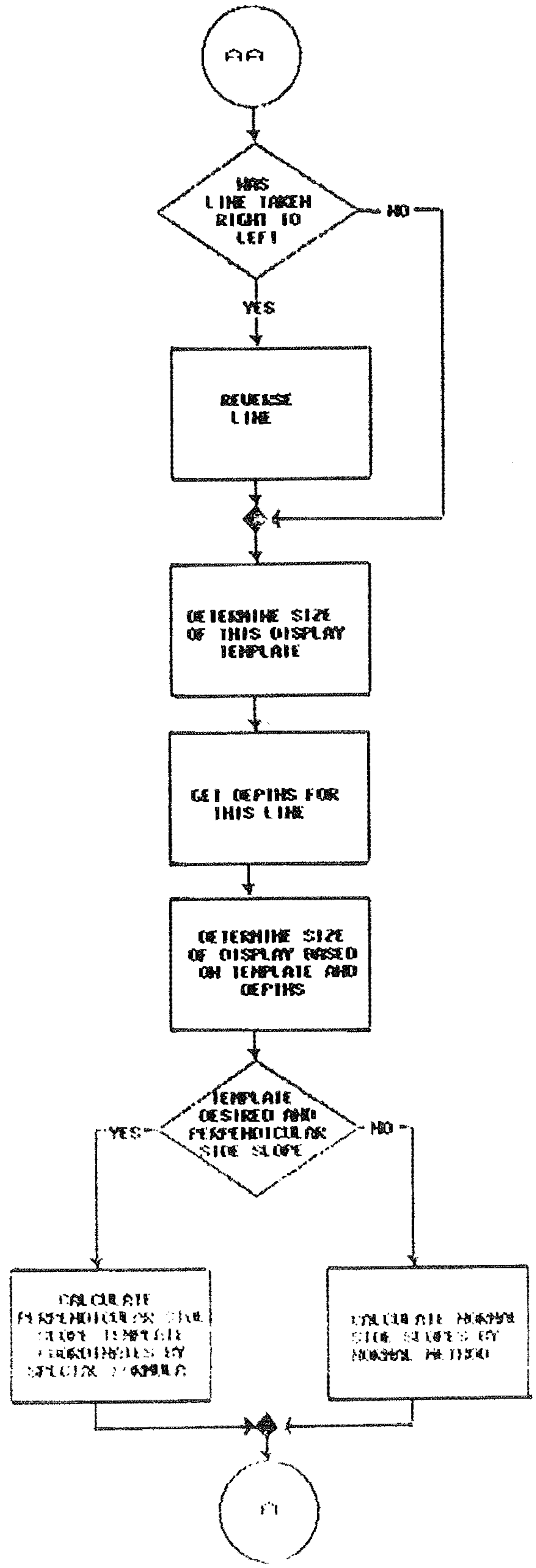


Figure C27. (Sheet 2 of 6)

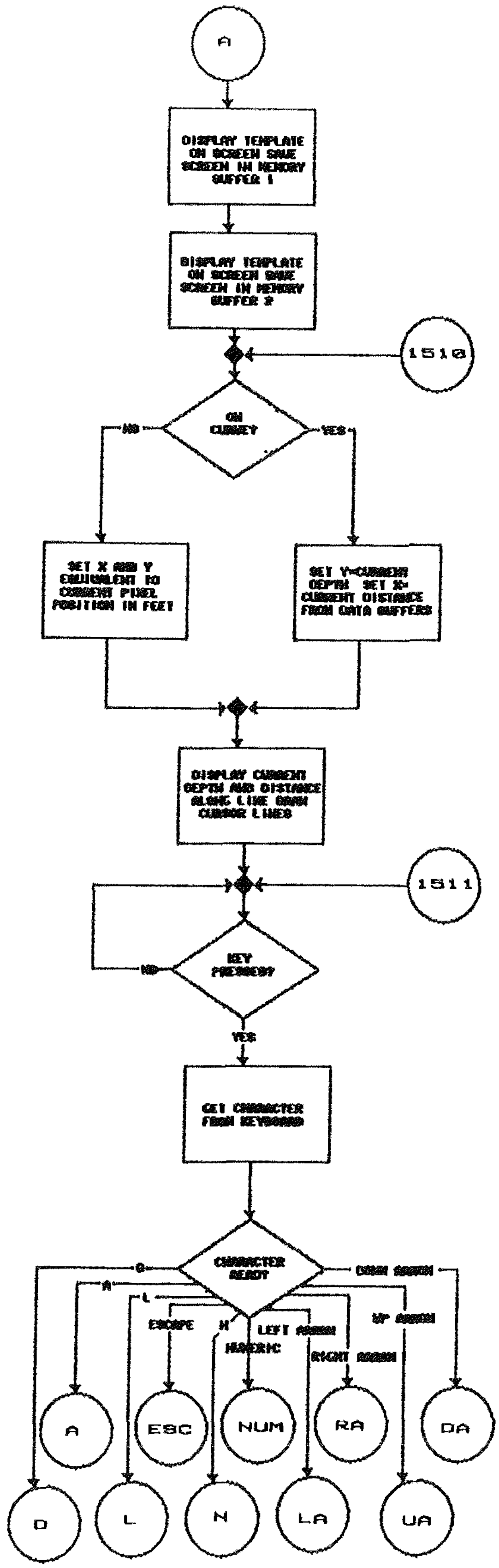


Figure C27. (Sheet 3 of 6)

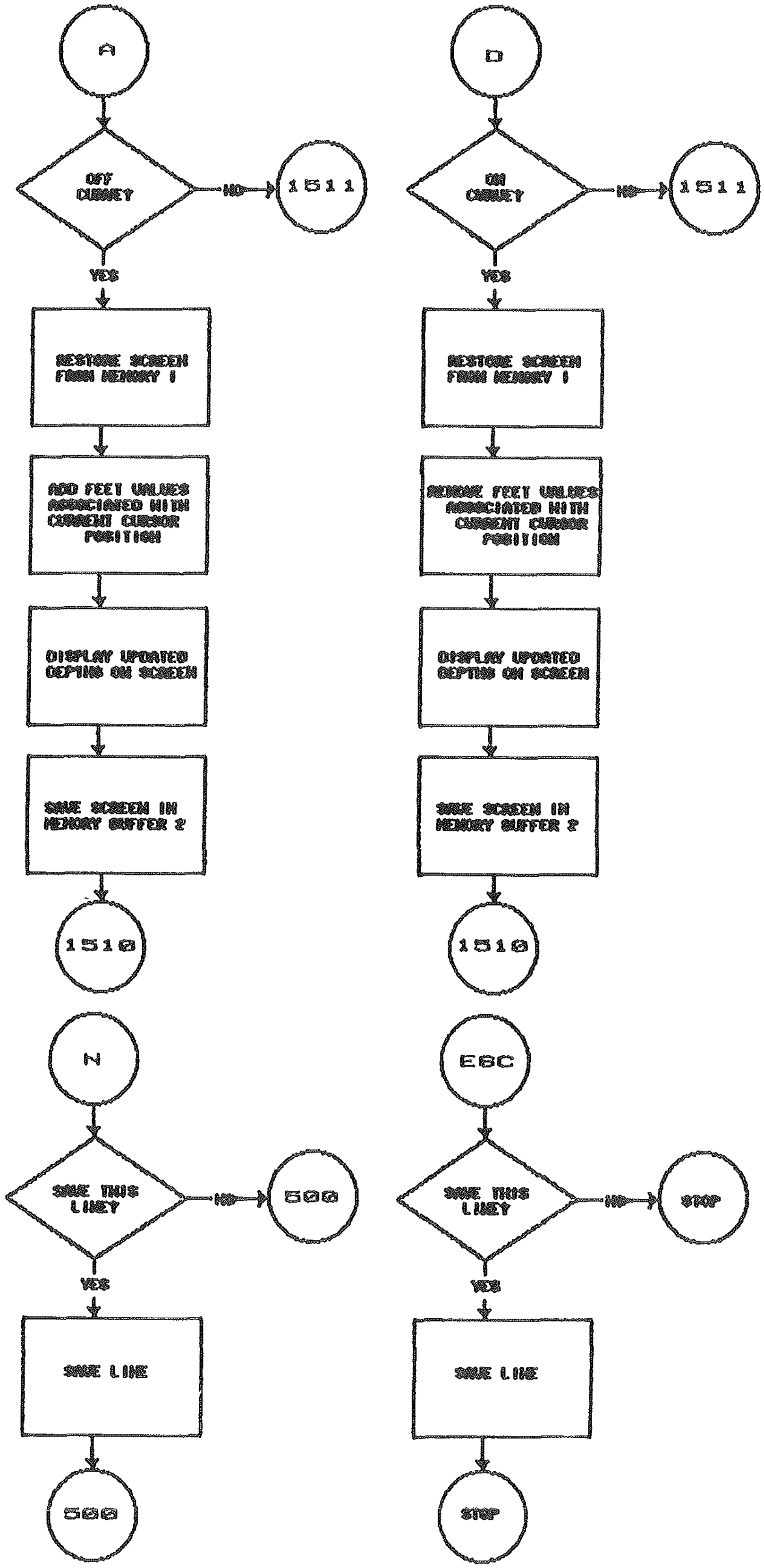


Figure C27. (Sheet 4 of 6)

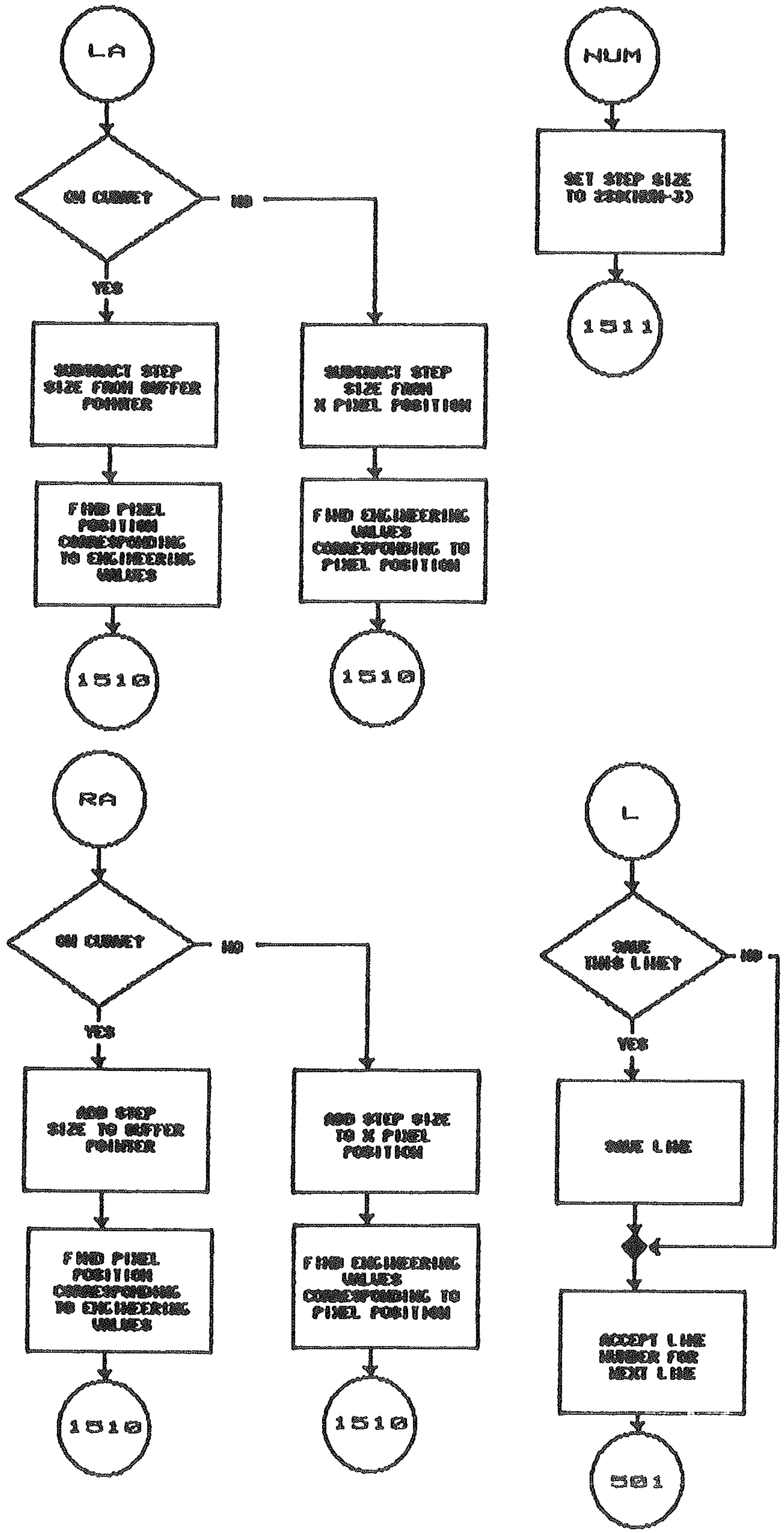


Figure C27. (Sheet 5 of 6)

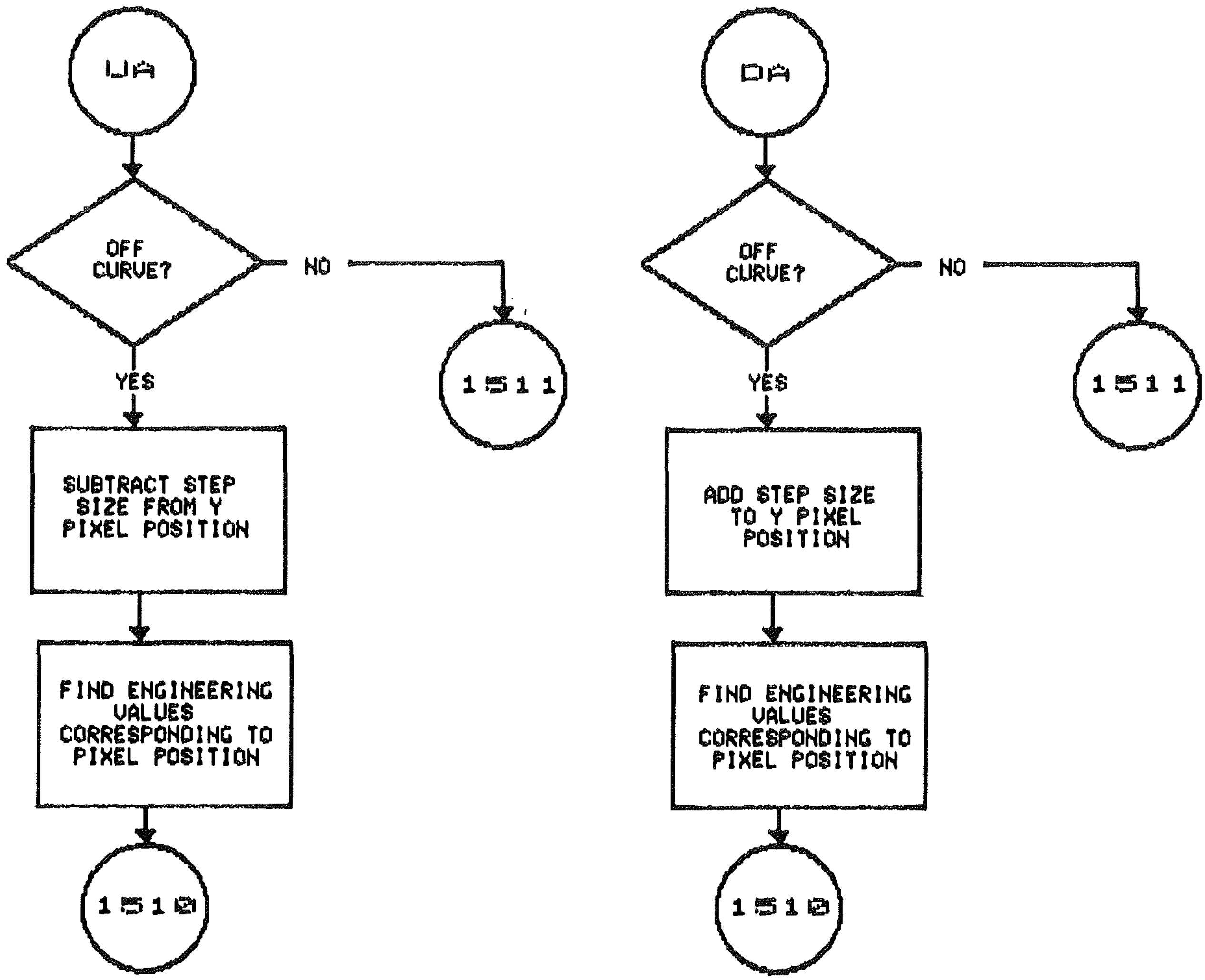
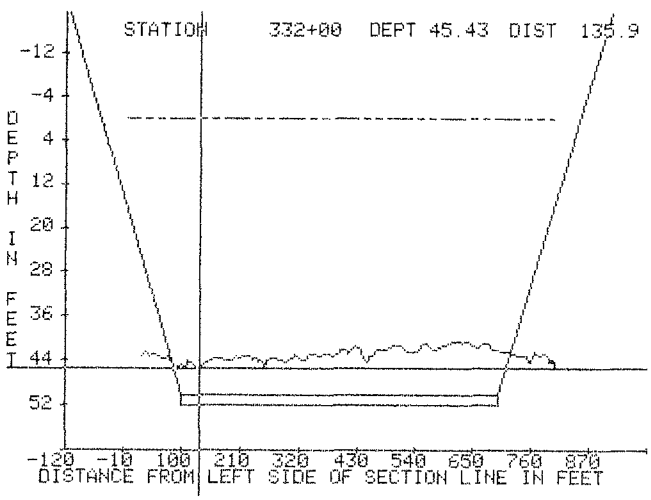


Figure C27. (Sheet 6 of 6)

THE COMMAND KEYS WHEN DATA IS GRAPHICALLY DISPLAYED
ARE
A   ADD A POINT WHEN TO LEFT OR RIGHT OF DATA
D   DELETE A POINT WHEN ON DATA
N   BRING UP NEXT SEQUENTIAL LINE
L   BRING UP A SELECTED LINE
1-0 NUMBERED KEYS SET ARROW KEY JUMP SIZE 2**KEY
<- ARROW KEY MOVE LEFT SELECTED JUMP VALUE
-> ARROW KEY MOVE RIGHT SELECTED JUMP VALUE
^^ ARROW KEY MOVE UP SELECTED JUMP VALUE
vv ARROW KEY MOVE DOWN SELECTED JUMP VALUE
ESC ESCAPE KEY TO EXIT PROGRAM

TEXT STRING OOPS, EXIT PROGRAM
PRESURVEY SITE NAME GROUPB
DISPLAY TEMPLATE   YES
ENTER INPUT SITE NAME AND CONFIRM

EDIT_DATA-1



EDIT_DATA-2

Figure C28. Sample run, EDIT_DATA

EDIT_FEATURES

Purpose

EDIT_FEATURES can be used to edit existing feature and label files or to create new feature and label files from scratch.

Input files

NAME.FEAT, NAME.LABE

Output files

NAME.FEAT, NAME.LABE

External devices

Summagraphics Bit Pad 1 or Microgrid digitizer

Modules called

GETTEXT, PUT_SCREEN, GET_XY, GET_PEN, STORE_PEN, STORE_XY, ERASE_IT, ERASE_WRITE, PLOTSCRN, FIND_2_POINTS, CHOSE_POINTS, SCHAR, PLOTLAB, PROBLEM, POLREC, INITMENU, DRAWMENU, MESSAGESTACK, DRAWFORM, DISTOFF, GET_NAME_OF_FILE, ROTATE, DIGITIZER, DIGITIZER2, INISOUND, WRSOUND, TERMSD, INISER, RDSERI, WRSERI, TERMSE, SCREENLINE, SCALE_IT_SCREEN, UNSCALE_IT_SCREEN, UNSCALE_IT_PIXEL, SCALE_IT_PLOT, SCREEN_LINE, EQUATION_FROM_2_POINTS, INTERSECTION, POINT_ALONG_A_LINE, UPDATESCREEN, ASCI, PRINT_SCREEN, SAVE_IT, AUTO_SCALE, SQUARE_IT_UP, RENAME_IT

Theory of operation

EDIT_FEATURES begins by accepting the file name. If the file name entered exists, the existing file is read; if the file name entered does not exist, the program will create a new file of that name. It then asks for type of digitizer. It then calibrates the map to digitizer units. Next the program asks if it should use autoscale, last window, keyboard entered, or digitizer entered. Then the features file is drawn to the screen by the method chosen. Once drawn, the features and labels file can be edited. A sample run is shown in Figure C29.

Special considerations

Status. The program can be used to edit existing feature and label files or for beginning new feature and label files. Forms and menus have not been fully incorporated at present; some entries are still done by entering a number or letter followed by a carriage return. The current version uses memory resident always.

Beginning a new feature with digitizer. Select TEST and EDIT_FEATURES.

```
C   CUT (WRITE & ERASE)--NOT IMPLEMENTED
D   DIGITIZE A FEATURE
E   ERASE A BLOCK
G   GET A STAMP UNSCALED, SIZE & ROTATE, ADD--NOT IMP
LEMENTED
H   HARD COPY
K   KEY IN A FEATURE
N   NEW OVERALL DATA WINDOW COORDINATES
P   PASTE A BLOCK SCALED--NOT IMPLEMENTED
Q   QUIT
S   SAVE A STAMP UNSCALED--NOT IMPLEMENTED
U   UNZOOM TO OVERALL DATA WINDOW
W   WRITE A BLOCK SCALED--NOT IMPLEMENTED
Z   ZOOM ON BLOCK
1-0 NUMBERED KEYS SET ARROW KEY JUMP SIZE 2**KEY
<-  ARROW KEY MOVE LEFT SELECTED JUMP VALUE
->  ARROW KEY MOVE RIGHT SELECTED JUMP VALUE
^^  ARROW KEY MOVE UP SELECTED JUMP VALUE
vv  ARROW KEY MOVE DOWN SELECTED JUMP VALUE
RET RETURN KEY TO CONFIRM POSITION
 SUMMAGRAPHICS BIT PAD 1  SUMMAGRAPHICS MICROGRID →
 DIGITIZER SUMMAGRAPHICS BIT PAD 1
                 ENTER AND CONFIRM
```

EDIT_FEATURES - 1

```
 TEXT STRING  FINISHED
 FILE NAME            LOCALITY
          ENTER INPUT FILE NAME AND CONFIRM
```

EDIT_FEATURES - 2

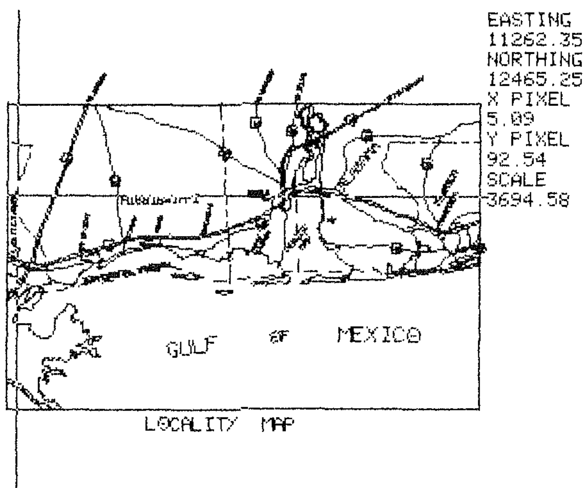
```
 ENTER EASTING IN FEET FOR FIRST POINT
11040
 ENTER NORTHING IN FEET FOR FIRST POINT
7520
 PRESS PEN DOWN TO DIGITIZE FIRST POINT
         360.00         896.00

 SECOND POINT SHOULD BE IN THE LOWER RIGHT HAND CORNE
R
 AT THE SAME NORTHING AS THE FIRST POINT

 ENTER EASTING IN FEET FOR SECOND POINT
21950
 PRESS PEN DOWN TO DIGITIZE SECOND POINT
        2493.00         937.00

 THIRD POINT SHOULD BE IN THE UPPER RIGHT HAND CORNER

 AT THE SAME EASTING AS THE SECOND POINT

 ENTER NORTHING IN FEET FOR THIRD POINT
```

EDIT_FEATURES - 3

Figure C29. Sample run, EDIT_FEATURES (Continued)

```
K
 AT THE SAME NORTHING AS THE FIRST POINT

 ENTER EASTING IN FEET FOR SECOND POINT
21950
 PRESS PEN DOWN TO DIGITIZE SECOND POINT
        2493.00          937.00

 THIRD POINT SHOULD BE IN THE UPPER RIGHT HAND CORNER

 AT THE SAME EASTING AS THE SECOND POINT

 ENTER NORTHING IN FEET FOR THIRD POINT
14580
 PRESS PEN DOWN TO DIGITIZE THIRD POINT
        2469.00          2312.00
 ERROR FROM 90 DEGREE ANGLE IS    -0.1
 NEW WINDOW INPUT METHOD:
A FOR AUTOSCALE
L FOR LAST WINDOW
K FOR KEYBOARD ENTERED
D FOR USE DIGITIZER CALIBRATION POINTS
ENTER A, L, K, OR D
```

EDIT_FEATURES - 4



EDIT_FEATURES - 5

Figure C29. (Concluded)

Position map on digitizer tablet. Enter a nonexistent file name into the file form (e.g., DDOG7) and confirm. Select SUMMAGRAPHICS MICROGRID digitizer from the next form and confirm. Choose calibration points that are outside the area to be digitized (i.e., the edges of the map). Enter the lower left easting, return. Enter the lower left northing, return. Digitize the lower left coordinate. Enter the lower right easting, return. Digitize the lower right coordinate. Enter the the upper right northing, return. Digitize the upper right coordinate. For new window input method, enter *D*, return. Proceed to "Entering changes" section.

Beginning a new feature without digitizer. Select TEST and EDIT_FEATURES. Enter a nonexistent file name into the file form (e.g., DDOG8) and confirm. Select NONE from file form for digitizer and confirm. For new window input method, enter *K*, return. Enter lower left easting, return. Enter upper right easting, return. Enter lower left northing, return. Enter upper right northing, return. Proceed to "Entering changes" section.

Editing an old feature with digitizer. Before beginning, plot existing file with PLOT_FEATURES and mount on tablet. Select TEST and EDIT_FEATURES. Enter existing file name into file form (e.g., 032-3) and confirm. Select and calibrate digitizer as discussed in "Beginning a new feature with digitizer." Choose one of the new window input methods (if unsure, use *A*, return). Proceed to "Entering changes" section.

Editing an old feature without digitizer. Select TEST and EDIT_FEATURES. Enter existing file name into file form (e.g., 032-3) and confirm. Choose one of the new window input methods, most likely *A*, and return. Proceed to "Entering changes" section.

Entering changes. Crosshairs on screen may be positioned either by using the arrow keys on the keyboard or by moving the digitizer pen and depressing the button (last position will be where button released). Arrow key jump size for one pixel is the default; to change this, depress a numbered key. Five is equivalent to one pixel. Four is equivalent to one-half pixel. Six is equivalent to two pixels, and so on.

Functions may be selected by depressing the appropriate key:

Z Zoom

U Unzoom

E Erase a block

B Backup

Q Quit

L Enter a label

D Digitize a feature from the digitizer

K Digitize a feature using the keyboard arrows and screen

N Enter a new set of window coordinate points (note: not normally done)

ESC Aborts present operation

H Hard copy to printer of screen

The functions are described in more detail in the following sections.

Zoom. Position with the arrow keys or digitizer to one corner of the zoom box. Press the *Z* key. Position to the diagonally opposite corner of the zoom box (a box will be formed as the cursor is moved). Press return to zoom or *ESC* to abort zoom. Note that nested zooms are allowed. You may zoom on a zoomed area repeatedly.

Unzoom. Press the *U* key.

Erase a block. To erase a rectangular block from the data on the current screen (screen may be zoomed), position with digitizer or arrow keys to one corner of the erase rectangle and strike the *E* key. Then position to the diagonally opposite corner of the erase box (a box will be formed as the cursor is moved). Press return and the contents of the box will be clipped from the data in memory or *ESC* to abort the erase. If the lower left-hand corner of the first letter of a label is in the box, the entire label will be erased.

Backup. *B* may be entered to save the current feature and label information in memory to disk (or you may wait until finished and the exit routine will ask whether or not to save changes).

Quit. *Q* is entered to end the program. If any unsaved changes have been made, you will be asked whether or not to to save the current memory contents to disk.

Label. *L* is entered to initiate label entry. Labels may be entered at positions selected by either the digitizer or arrow keys or both. When *L* is entered, a dialog box is opened on the lower right-hand corner of the screen. The requested input is displayed along with user responses. The first input is the pen number. A one-digit number (0-8) should be entered followed by a carriage return. The next input is the label. A 1- to 120-character label followed by carriage return should be entered (backspace will remove characters from the back forward--the characters scroll left when more than 8 are entered). Then DIGITIZE FIRST POINT is displayed. The crosshair is moved using the arrow keys or digitizer to the bottom left-hand corner of the first character's desired position. A carriage return is entered to mark the spot. Then DIGITIZE SECOND POINT is displayed. The crosshair is moved to the bottom right-hand corner of the last character's desired position. A carriage return is entered to mark the spot. The properly sized and rotated label is displayed.

Digitize. Digitize works only with a digitizer. When *D* is entered, a dialog box is opened in the lower right-hand corner of the screen. The box requests the pen number and a one-digit number (0-8) is entered. The next request is for a line type: a one-digit number (1-7) is entered (2 = dashed line, 5 = center line, 7 = solid line, etc. See plotter manual for others), followed by a carriage return. The computer then goes into digitize feature mode. The crosshair is blanked and the computer waits for digitized data or key strokes. To input data, position the pen and depress the button, moving the pen in the desired pattern, or move the pen to corners of a polygon and depress the pen momentarily at each corner. The digitized points will be displayed on the screen as they are acquired. Look at the screen display. If it looks CORRECT, enter *E* or return to save the feature in memory. If you do not like it, enter *A* to abort the line. The computer will erase the line from the screen and memory. (Note: if the line crosses another line or label, parts of the line or label may be erased on screen, but not in memory; *U* may be entered to redraw the screen if this is bothersome.)

Key-entered features. *K* may be used to enter a feature from key. Features are entered from key by using the digitizer pen or arrow keys to position and the carriage return to mark each point of the feature. A feature is ended by entering *E* to save in memory or *A* to abort and discard. Entering *K* causes a dialog box to form in the lower right-hand corner. The pen number for the feature 0-8 is entered as a one-digit number followed by a carriage return. Then the line type for the feature is entered as a one-digit number followed by a carriage return (2 = dashed line, 5 = center line, 7 = solid line, etc. See plotter manual for definition of line types.). Then the program goes into digitize mode. Move the crosshairs to the first desired position and hit return. Then move to each successive position hitting returns until finished. If the line on the screen looks correct, hit *E* to save it; if not, enter *A* to abort it. If the line is not cleanly erased when *A* is hit, you may use *U* to redraw the screen (the line has been cleanly erased in memory; only the display is at fault).

NEW_WINDOW. NEW_WINDOW automatically comes up at the beginning of the program and may be selected during the program by depressing *N*. NEW_WINDOW sets the coordinates of the unzoomed screen display. It asks that *A*, *L*, *K*, or *D* be entered.

A (for AUTO_SCALE) searches the .FEAT and .LABE files (on disk or in memory) and finds the minimum and maximum x and y. The lower left-hand corner is set to MINX and MINY, the upper right-hand corner is set to MAXX and MAXY.

L (for LAST_WINDOW) instructs the user to use LAST_WINDOW. Each time a new window is selected, its coordinates are saved in a file called LAST_SCALE. If *L* is selected, the contents of this file are read and used. If desired, LAST_SCALE may be edited with the text editor after running the program to see the coordinates (for future use).

K (for KEYBOARD_ENTERED) requests the user to enter coordinates for upper right and lower left from key. The coordinates from a previous AUTO_SCALE could be used.

D (for DIGITIZER_ENTERED) requests the user to use coordinates from digitizer calibration. If the digitizer was calibrated to include the entire area, this may be the best choice.

If using existing files, *A* or *L* might be best. If using a new map, *D* or *K* might be best.

Escape. The escape key aborts the present operation: (1) zoom, (2) erase, (3) digitize feature, or (4) key in feature. The operation will not be completed and its effects are removed. Once the operation is completed, escape has no function.

Hard copy. The *H* key causes the screen to be dumped to the printer providing a hard copy of the present contents including crosshair reading. Since the crosshair reading is always current, the position of a point can be found and recorded to paper using this feature.

Example run. Copy the files LOCALITY.FEAT and LOCALITY.LABE to hard disk under the MULTITASK subject. Run PLOT_FEATURES to plot the map (to the size of your digitizing tablet if you have one). The map has an outline that may be used for calibrating the digitizer; the coordinates of the left-hand corner are (11040, 7520) and of the upper right-hand corner are (21950, 14580). In the following description, choose nondigitizer options if you do not have one instead of the options listed.

Mount the paper on the digitizer. Run TEST "EDIT_FEATURES" from the MULTITASK DEVELOP file. When the digitizer form is displayed, select MICROGRID as the digitizer using one left-cursor motion (if you have a digitizer; otherwise, select NONE) and confirm. When the file form is displayed, enter LOCALITY and confirm (code-return). The digitizer calibration portion will begin. Follow the instructions by entering:

(1) 11040, carriage return

(2) 7520, carriage return

(3) Locate pen at lower left-hand corner of outline box and press button to digitize first point

(4) 21950, carriage return

(5) Locate pen at lower right-hand corner of outline box and press button to digitize second point

(6) 14580, carriage return

(7) Locate pen at upper right-hand corner of outline box and press button to digitize third point

(8) If the points are accurately entered, the new window option is displayed. Enter *A*, carriage return.

The LOCALITY.FEAT and .LABE files should be drawn on the screen. Wait until the screen is completely drawn and the crosshair is displayed. Move the crosshair using the arrow keys. Depress 8 and the crosshair will move more quickly; depress 3 and it will move by partial pixels, allowing more resolution when entering coordinates from the crosshair. Depress 5 and the crosshair will again move at its normal rate. Use the digitizer pen to position the cursor. Move the pen to a point on the map and press the button. The crosshair will jump to that point on the screen.

Test the zoom feature. Position the crosshair to some point on the screen. Depress *Z*. Expand the box (the crosshair has been replaced) to surround a small section of the screen. Depress return, and the screen will be redrawn; wait until the crosshair reappears. Test nested zooms by positioning to a point on the screen and depressing *Z*. Then expand the box and depress return. The screen will zoom again. Test the hard copy feature by pressing *H* to print the screen. Test the unzoom feature by depressing *U* to return the screen to its initial display.

Test entering labels. Enter L. The label dialog box will appear in the lower left-hand corner of the screen. For pen number enter 1, carriage return. For label, enter "WHAT AN AWEE<(backspace)>SOME PROGRAM," carriage return. Position the crosshair where you want the lower right-hand corner of the label to be and enter a carriage return. The label is drawn on the screen and saved in memory (a flag in the program is set to let it know the file has been modified). Test digitizing features for the digitizer by entering *D*. The DIGITIZE FEATURES dialog box appears. Enter the pen number 1, carriage return. Enter the line type 7 (solid line), carriage return. The program goes into digitize mode, the crosshair is blanked. Move it from one point to another depressing the button at each point. Your pattern is displayed on screen. Depress *E* or carriage return to save the feature (*A* or *ESC* if you do not want to save it).

Test digitizing features from the keyboard by entering *K*. The KEYED FEATURES dialog box appears. Enter the pen number 1, carriage return. Enter the line type 7 (solid line), carriage return. Move the crosshair using the arrow keys to the desired point, and press the carriage return to digitize the first point. Move the crosshair to the next point and press the carriage return to digitize the second point. Continue in this manner until the desired feature is displayed on the screen. Enter *E* to end and save (or *A* or *ESC* to abort the save).

Test erasing a block. Position the crosshair to a corner of the rectangle to be erased. Enter *E*. Enlarge the box using the arrow keys or digitizer to surround the area to be erased and enter a carriage return. The screen should be redrawn with the area within the erase rectangle deleted. Features are cropped at the edges leaving partial features. Labels are removed fully if the lower left-hand corner was within the erase rectangle.

Enter *Q* to quit. Since the files were modified, the computer will display the unsaved method. Normally you would enter *Y* and carriage return to save the modifications, but for now enter *N*, carriage return to exit without saving the changes and the changes will not take effect.

EQUIPMENT_ENTRY

Purpose

EQUIPMENT_ENTRY defines the equipment-specific information for a survey site.

Input files

NAME.EQUI

Output files

NAME.EQUI

External devices

NONE

Modules called

PROBLEM, INITMENU, DRAWMENU, MESSAGESTACK, DRAWFORM, GET_NAME_OF_FILE

Theory of operation

EQUIPMENT_ENTRY displays an input file form and reads information to be modified from the file. It then displays a series of triordinate input forms until the operator sets MORE to NO. It then displays the equipment form and allows input of equipment to be used. The flowchart is shown in Figure C30, and a sample run is shown in Figure C31.

Special considerations

This program must be run prior to a section line, random, or find-spot survey. Simulation of either range or depth device may be specified for checkout purposes. For range azimuth, transponder triordinates are entered in pairs: (1) theodolite position and (2) backsight.

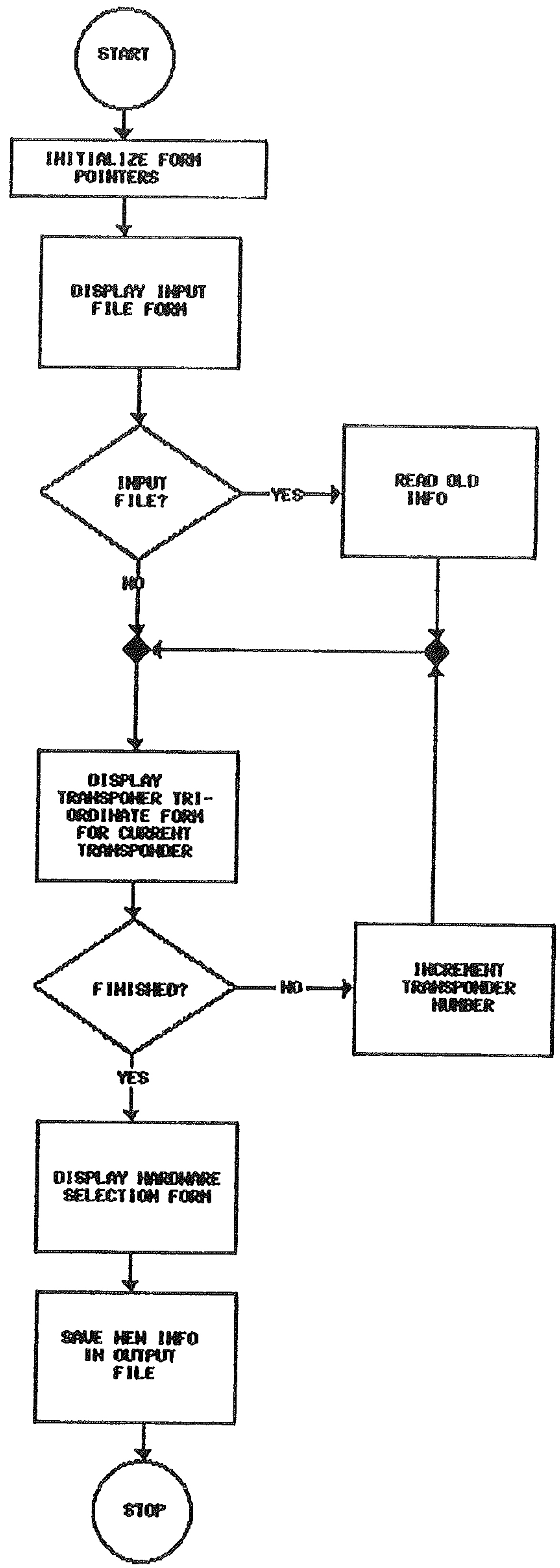


Figure C30. Flowchart,
EQUIPMENT_ENTRY

```
********************************************
CODE RETURN CONFIRMS A FORM
ESCAPE (NOT CODE ESCAPE) ALLOWS ESCAPING
```

```
A TEXT STRING  NO INPUT FILE
INPUT FILE NAME GROUPB
        ENTER INPUT FILE NAME AND CONFIRM
```

EQUIPMENT_ENTRY-1

```
********************************************
CODE RETURN CONFIRMS A FORM
ESCAPE (NOT CODE ESCAPE) ALLOWS ESCAPING
```

```
A REAL NUMBER
EASTING                          724502.1250
NORTHING                         2926944.5000
HEIGHT                           10.0000
MORE TRANSPONDERS AFTER THIS?    YES
        ENTER TRANSPONDER TRIORDINATES NO.  1
```

EQUIPMENT_ENTRY-2

```
********************************************
CODE RETURN CONFIRMS A FORM
ESCAPE (NOT CODE ESCAPE) ALLOWS ESCAPING
```

```
A REAL NUMBER
EASTING                          724638.5000
NORTHING                         2927091.0000
HEIGHT                           10.0000
MORE TRANSPONDERS AFTER THIS?    NO
        ENTER TRANSPONDER TRIORDINATES NO.  2
```

EQUIPMENT_ENTRY-3

```
********************************************
CODE RETURN CONFIRMS A FORM
ESCAPE (NOT CODE ESCAPE) ALLOWS ESCAPING
```

```
← CUBIC  POLARFIXSERIAL  GEODIMET140T  IMC
OUTPUT FILE NAME     GROUPB
RANGE EQUIPMENT      IMC
DEPTH EQUIPMENT      INNERSPACE GET
BOAT ANTENNA HEIGHT  10.0000
FIX MARKS            FEET
FIX MARK INTERVAL    10.0000
        FILL IN EQUIPMENT FORM AND CONFIRM
```

EQUIPMENT_ENTRY-4

Figure C31. Sample run, EQUIPMENT_ENTRY

FIND_SPOT

Purpose

FIND_SPOT is a survey task designed to display spot-type surveys and accept operator run-time inputs. It is intended for guiding the pilot to pre-defined coordinates.

Input files

NONE (uses common block of memory for data passing)

Output files

.M001, .D001 (where 001 is mark or tide entry number)

External Devices

NONE

Modules called

ASCII, UPDATESCREEN, POLREC, A2, PROBLEM, INISOUND, WRSOUND, TERMSD

Theory of operation

FIND_SPOT begins by displaying the first spot on screen. It loops waiting for a key to be pressed and updating the screen. If *E* is pressed, the spot number is incremented and the next spot displayed. The flowchart is shown in Figure C32.

Special considerations

FIND_SPOT does not automatically save any data. The *M* key must be pressed to mark a spot and save data.

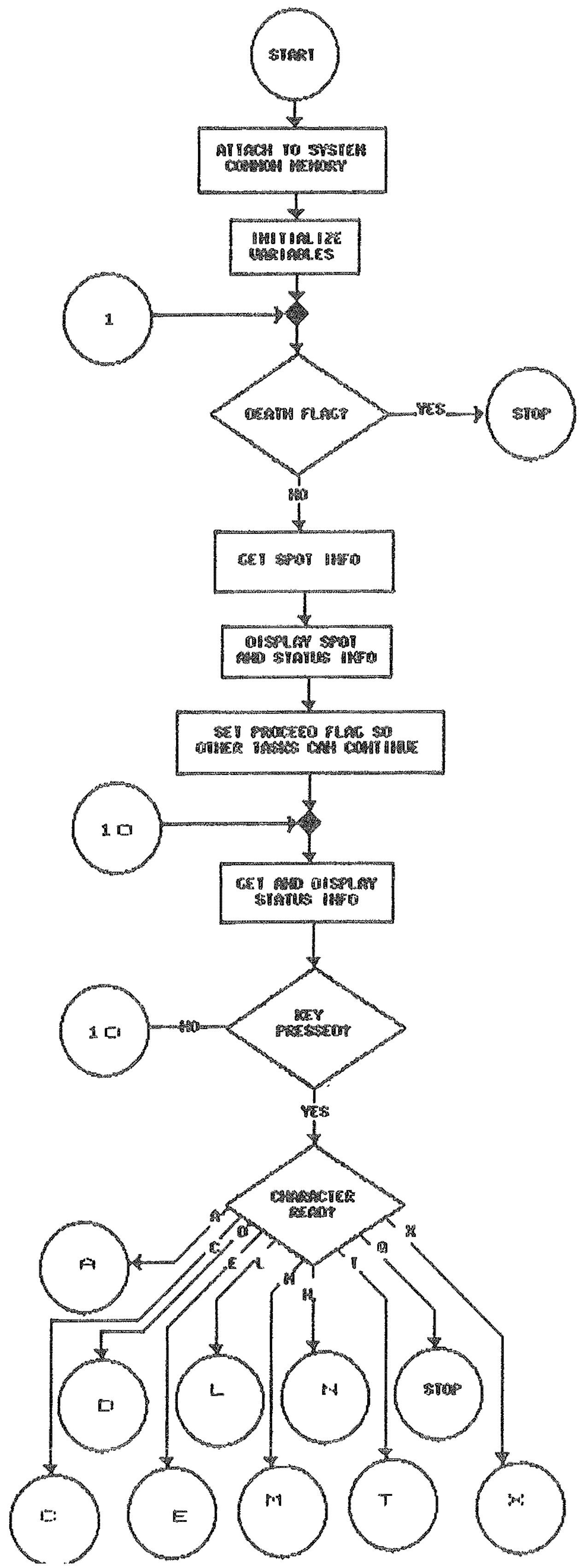


Figure C32. Flowchart, FIND_SPOT
(Continued)

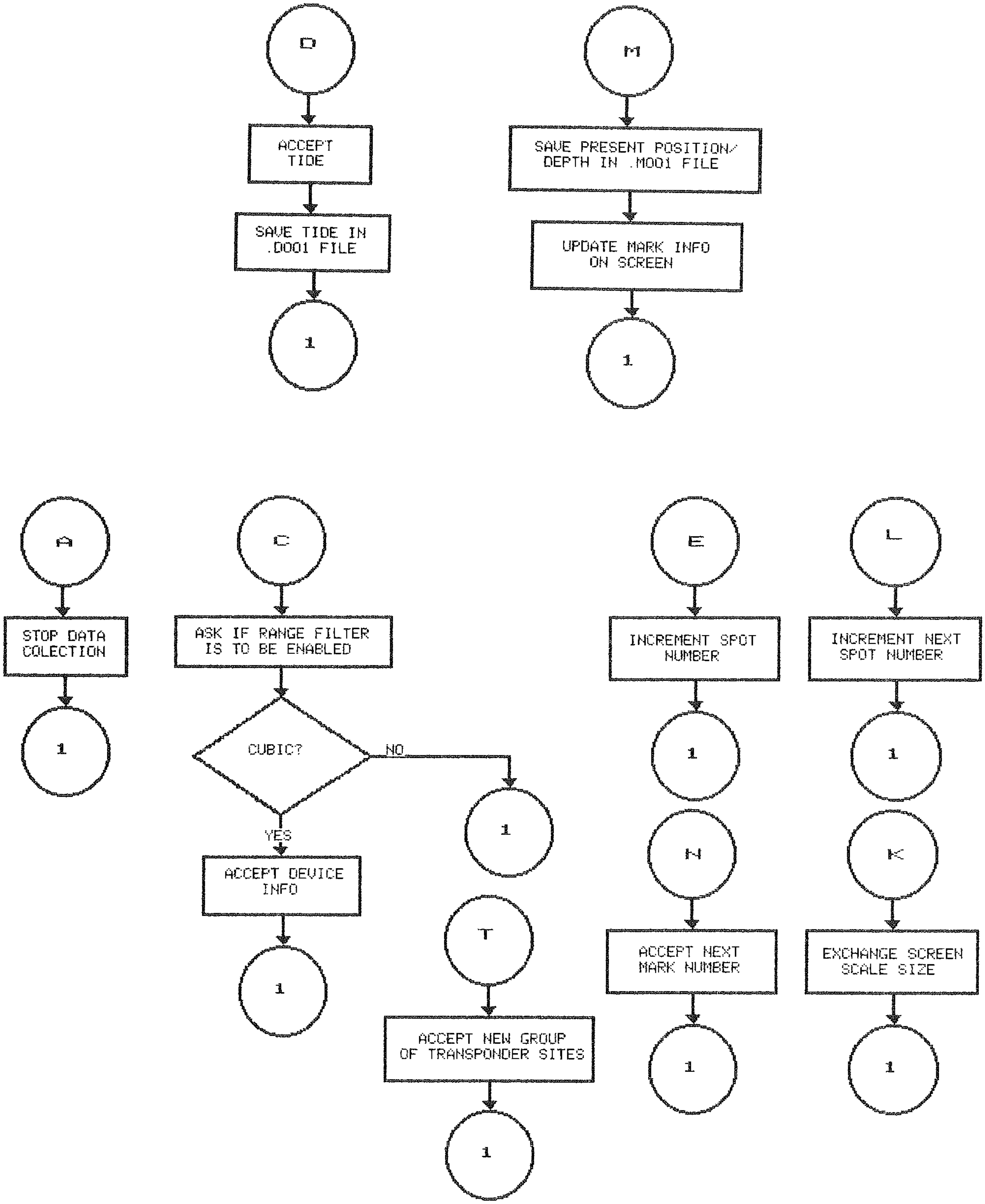


Figure C32. (Concluded)

GENERIC_TO_GRID_FORMAT

Purpose

GENERIC_TO_GRID_FORMAT was developed to ease the import of data collected on other systems into the small-boat postsurvey package. Data files are normally stored in binary format to reduce storage space required; all analysis programs expect this input. Also each line is normally stored in a separate file. To ease import of data, an ASCII file on IBM*-formatted disk is specified for contractor interface. Contractors provide data in this generic format and it is converted to Grid format by the program. Therefore contractors need only be able to provide 5.25-in. 360-kB standard floppy diskettes in the specified data layout.

Input files

INPUT NAME (read from key)

Output files

OUTPUT NAME (read from input file, site,0001)

External devices

NONE

Modules called

PROBLEM, INITMENU, DRAWMENU, MESSAGESTACK, DRAWFORM, GET_NAME_OF_FILE

Theory of operation

Data are read in ASCII format, converted and written out in binary. A flowchart is shown in Figure C33.

---

* IBM is a registered trademark of the International Business Machines Corporation.

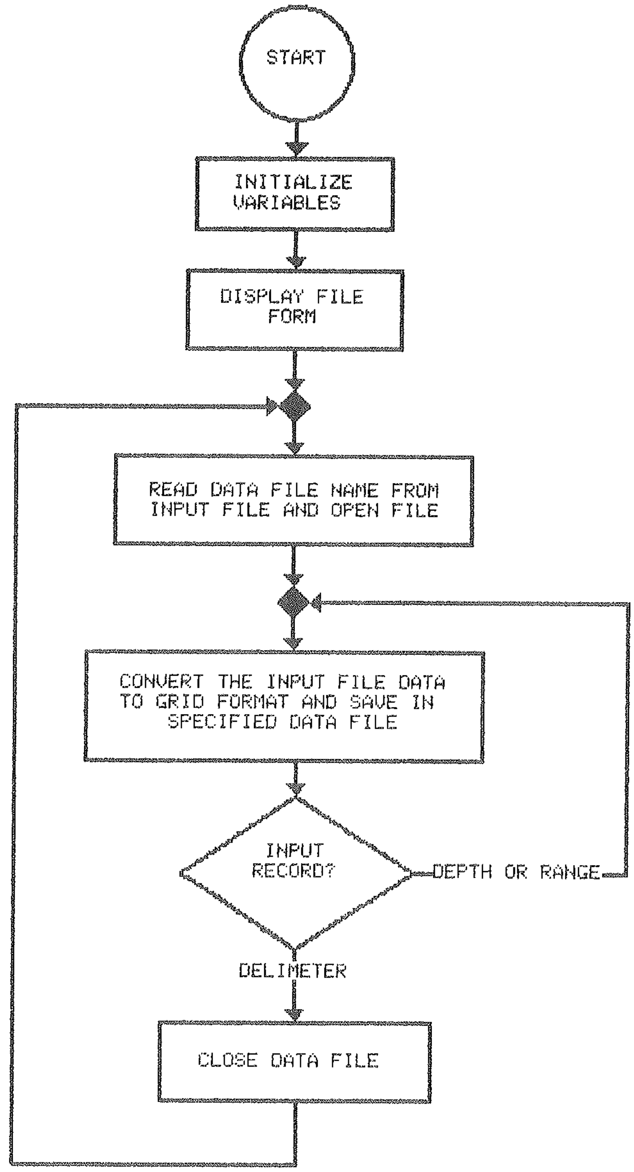


Figure C33. Flowchart, GENERIC_TO_GRID_FORMAT

INDICATOR

Purpose

INDICATOR is a program used to produce the pilot (helmsman) indicator meter display and to accept option changes via function keys from the pilot. It runs on the TRS 80 Model 100 or 102 computer* and normally displays an off-line and distance-along-the-line indicator.

Input files

NONE (data are received via the modem port)

Output files

NONE

External devices

NONE

Modules called

1000, 910, 920, 930, 940, 950, 960, 970, 980, 3600, 3650, 3500, 3550, 3200, 3100, 3150, 3000, 3050, 2800, 2805, 2880

Theory of operation

The TRS 80 pilot indicator program is written in BASIC (all other programs are FORTRAN) and runs on a TRS 80 model 100 or 102 Executive Workslate. The flowchart is shown in Figure C34. Two basic indicator displays are available.

One is a plan view of the section line (called the plan view display), which contains a distance meter along with an offline indicator. This indicator displays a boat that sails toward the center as the survey boat sails toward the survey line and a boat that sails from the center as the survey boat sails away from the line. If the boat's course is close to parallel to the section line or if the boat is on the section line, a boat sailing straight ahead is displayed. The distance meter is a bar graph-type meter that darkens the percent of total line length traversed.

The other indicator display (called the meter display) shows two enlarged meters for distance and offline. The offline meter is a conventional pointer-type meter. The distance meter is a bar graph type similar to the plan view version. Function pushbuttons on the front of the TRS 80 control the scale changes:

---

* TRS 80 is a registered trademark of the Tandy Corporation.

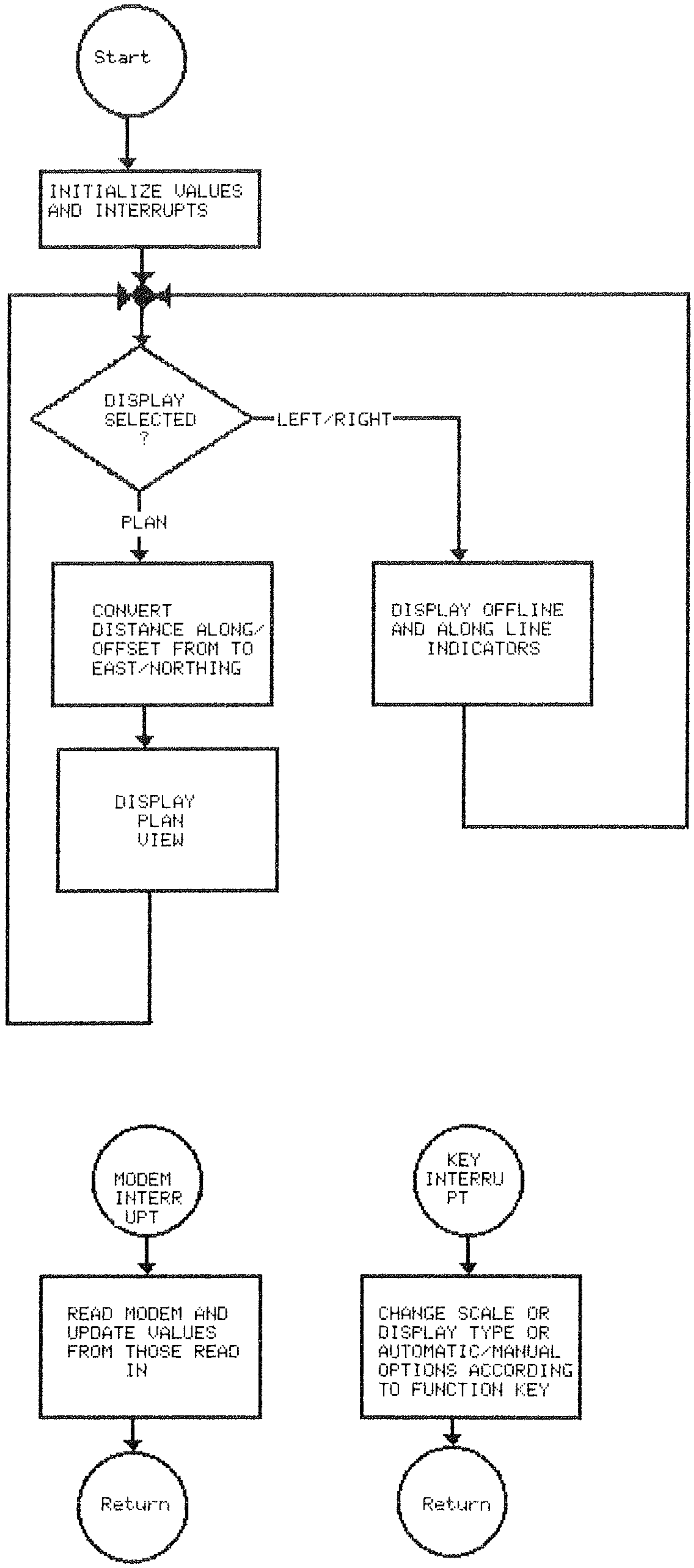


Figure C34. Flowchart, INDICATOR

a. F1 causes the offline meter to drop to a lower range each time it is depressed. For example, if the scale is -50 to 50 before F1 is depressed, it will be -20 to 20 afterwards.

b. F2 causes the offline meter to jump to a higher range each time it is depressed. For example, if the scale is -100 to 100 before F2 is depressed, it will be -200 to 200 afterwards.

c. F3 causes the display to toggle from the plan view display to the meter display, or vice versa, depending on the current display.

d. F4 causes the plan view display to toggle from normal to large area.

e. F5 toggles the enabling or disabling of automatic offset meter scale changing by the program; i.e., if automatic scale changing is enabled and if the meter pegs, the program will effectively push F2. Or if the indicator falls within 15 percent of the meter center, the program will effectively push F1.

f. F6 toggles the enabling or disabling of automatic plan view normal to large area switching; i.e., if the boat position falls within the bounds of the normal size plan view, this view is used; otherwise, the large area is used, if auto switching is enabled.

g. F7 toggles enabling or disabling auto switch from plan to meter display at the beginning and end of line; i.e., if auto switch is enabled, the plan view will be displayed automatically any time new line coordinates are received by the TRS 80 from the Grid Compass,* and the meter view will be displayed automatically when the distance along the line changes from negative to positive.

h. F8 causes the display to be redrawn (in case some unforeseen circumstance causes the display to be garbage).

There are two scales of plan view based on the length of the survey line:

a. A large area display used for initially finding the line.

b. A normal display for use when close to the line.

There are six different offline meter scales:

a. -10 to 10

b. -20 to 20

c. -50 to 50

d. -100 to 100

e. -200 to 200

f. -500 to 500

Scales may be either manually selected by the pilot or automatically selected

* Compass is a trade name of Grid Systems, Inc.

by the TRS 80 program. Letters are displayed on the lower left-hand portion of the TRS 80 display to indicate the current state of automatic feature enables. F5's current state is displayed by the letters *MM* if disabled (i.e., manual meter range) or *AM* if enabled (i.e., automatic meter range). F6's current state is displayed by the letters *MP* if disabled (i.e., manual plan view scale) or *AP* if enabled (i.e., automatic plan view scale). F7's current state is displayed by the letters *MS* if disabled (i.e., manual switch of displays) or *AS* if enabled (i.e., automatic switch of displays). Also the beginning (E1, N2) coordinates of the current line and plan view scale in feet/inches are displayed.

Special considerations

The INDICATOR program normally stays in the memory of the TRS 80 model 100 or 102 computer. This memory is maintained by four AA batteries in the TRS 80. If the batteries need replacing, first power the unit up with the AC adapter and then replace the batteries. If the program is lost, it is stored on the Grid computer in a file called INDICATOR and may be downloaded to the TRS 80. First connect the two with a phone-type modular connector cable. Select BASIC on the TRS 80 and type in the command LOAD "MDM:7E1E"; the TRS 80 should emit the typical modem tone. Select GRIDTERM on the Grid computer (the default options are normally 7 bits even parity as needed--otherwise change them). Set the baud rate to 300, the phone number to 0, and confirm. Then select "Code T" and "Send a file." Choose the file INDICATOR with no special character transmission. The transfer takes several minutes.

INTERSECTION_OF_2_LINES

Purpose

INTERSECTION_OF_2_LINES finds the coordinate where two lines intersect. This is useful for deriving coordinates for channel toes at points where they are not given. NEW_LINE_INPUT requires coordinates of all lines at any bend; sometimes left and right channel bends do not coincide and intermediate points must be calculated.

Input files

TEMPLINE

Output files

TEMPLINE

External devices

NONE

Modules called

INITMENU, ADDRESS, CHANNEL_MESSAGE, DRAWFORM, EQUATION_FROM_2_POINTS, INTERSECTION, A2, MESSAGESTACK

Theory of operation

The program first reads in the last set of coordinates used from a temporary disk file, TEMPLINE. Then a form for line coordinate input is displayed by calling DRAWFORM. If the quit option was selected (i.e., NCHOS1 is returned as 2), the last set of coordinates is saved in TEMPLINE and the program is ended. Otherwise, the line equations ($Ax + By + C = 0$) are generated by calling INTERSECTION. The results are displayed by calling MESSAGESTACK. The process then repeats with redisplay of the form. The flowchart is shown in Figure C35, and a sample run is shown in Figure C36.

Special considerations

NONE

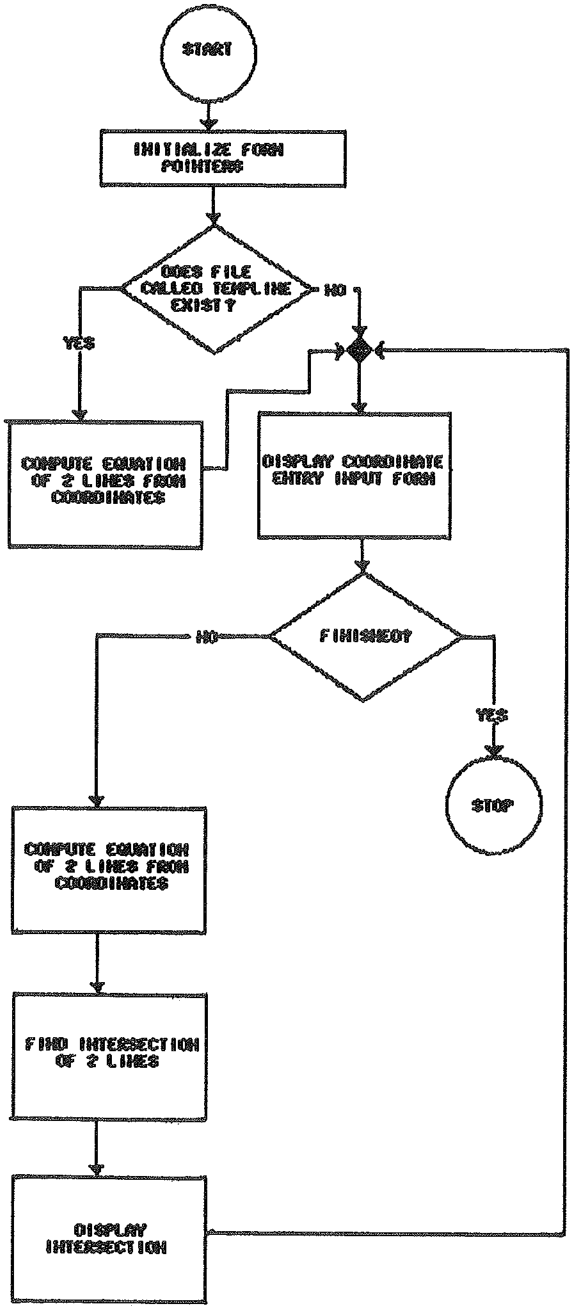
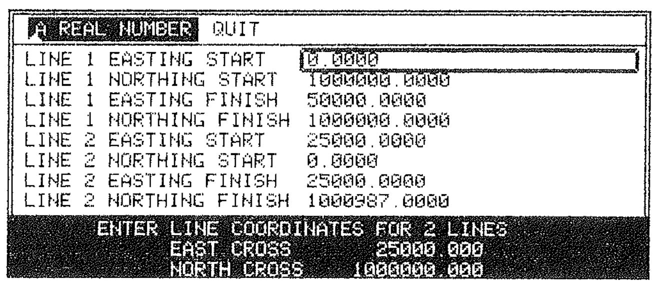


Figure C35. Flowchart, INTERSECTION_OF_2_LINES



INTERSECTION_OF_2_LINES-1

Figure C36. Sample run, INTERSECTION_OF_2_LINES

MANUAL_SURVEY_ENTRY

Purpose

MANUAL_SURVEY_ENTRY accepts survey depths from key. It is used to input data available from no other source than a notebook; i.e., it is not available on disk or in plotted form.

Input files

NAE.LINE

Output files

NAME.0001 (where .0001 is line number)

External devices

NONE

Modules called

PROBLEM, INITMENU, DRAWMENU, MESSAGESTACK, DRAWFORM, GET_NAME_OF_FILE, DISTOFF, POLREC, A2

Theory of operation

MANUAL_SURVEY_ENTRY begins by displaying a file form and reading in the .LINE file. It then displays a line entry form for the line number to be processed. It accepts date and other parameters, then loops through outputting a distance along the line and asking for the corresponding depth. Output is a .0001 file in the same format as lines acquired by automated surveys. The flowchart is shown in Figure C37, and a sample run is shown in Figure C38.

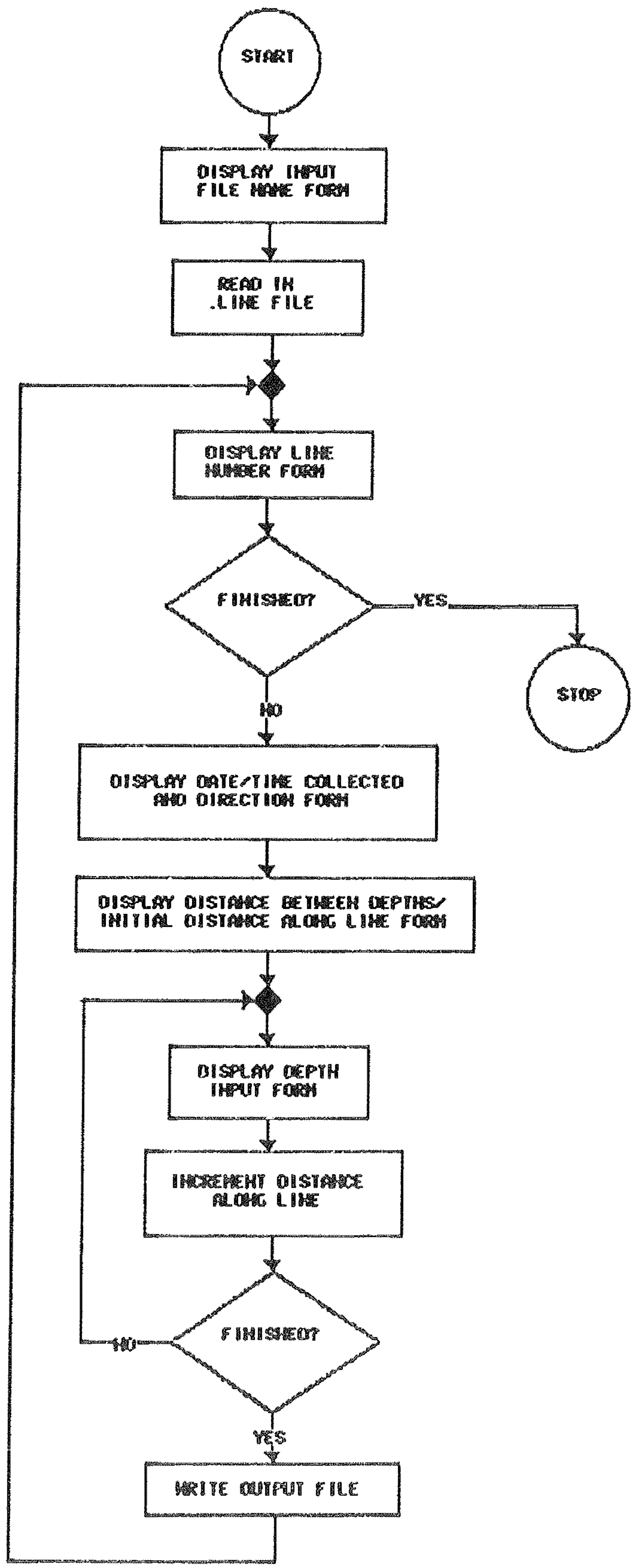


Figure C37. Flowchart, MANUAL_SURVEY_ENTRY

```
TEXT STRING  FINISHED
FILE NAME            groupb
      ENTER INPUT FILE NAME AND CONFIRM
```

MANUAL_SURVEY_ENTRY-1

```
AN INTEGER  FINISHED
LINE NUMBER  1
          ENTER AND CONFIRM
```

MANUAL_SURVEY_ENTRY-2

```
MONTH DATA ACQUIRED                      12
DAY ACQUIRED                             31
YEAR ACQUIRED                            1988
HOUR ACQUIRED                            12
MINUTE ACQUIRED                          59
 LEFT SIDE CORRESPONDS TO FIRST VALUE?   YES
          ENTER AND CONFIRM
```

MANUAL_SURVEY_ENTRY-3

```
IN FEET
DISTANCE BETWEEN DEPTHS        20
INITIAL DISTANCE FROM START    50
          ENTER AND CONFIRM
```

MANUAL_SURVEY_ENTRY-4

```
IN FEET
ENTER DEPTH  30.0000
FINISHED?    NO
      AT       50.00 FEET FROM START
```

MANUAL_SURVEY_ENTRY-5

Figure C38. Sample run, MANUAL_SURVEY_ENTRY

MARKS_HARRIS

Purpose

MARKS_HARRIS converts mark files from binary form to an ASCII format required by a Louisville District program.

Input files

NAME.M001 (where 001 is the mark number)

Output files

MARKS_HARRIS

External devices

NONE

Modules called

PROBLEM, INITMENU, DRAWMENU, MESSAGESTACK, DRAWFORM, GET_NAME_OF_FILE

Theory of operation

MARKS_HARRIS displays a form for site name input. It loops through reading in the .M001 files, converting them to time and easting-northing format, and writing them out. A flowchart is shown in Figure C39.

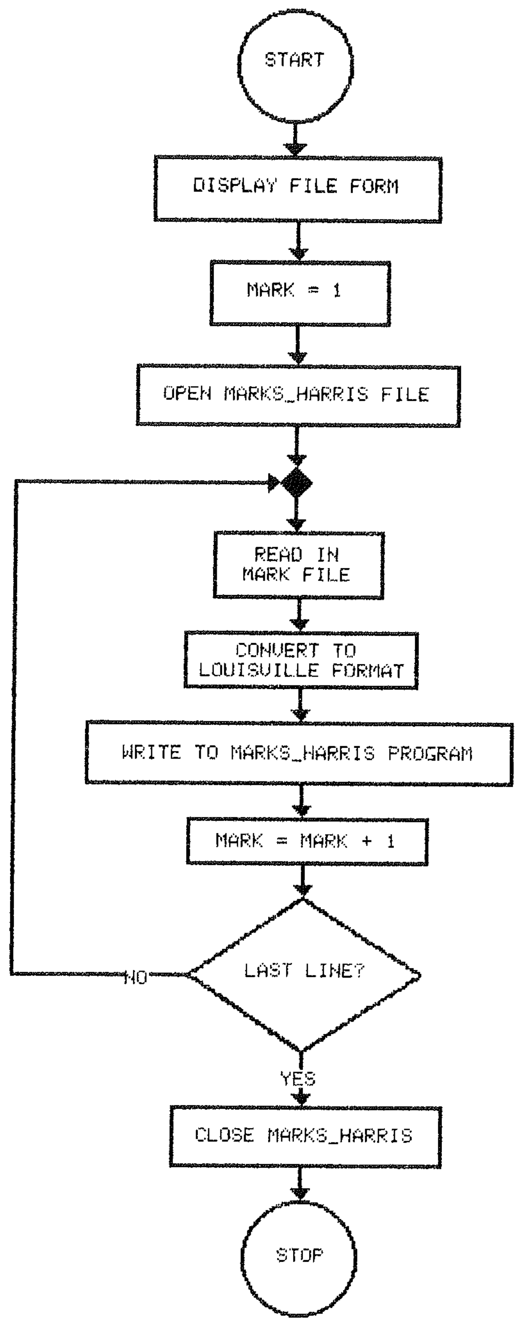


Figure C39. Flowchart,
MARKS_HARRIS

MARKS_TO_DESCRIPTOR

Purpose

MARKS_TO_DESCRIPTOR creates a .MARK file for input to the PLAN_PLOT program so that marks can be plotted using the desired pen, special symbol, and label.

Input files

NAME.M001 (where 001 is the mark number)

Output files

TEMP, NAME.MARK

External equipment

NONE

Modules called

PROBLEM, INITMENU, DRAWMENU, MESSAGESTACK, DRAWFORM, GET_NAME_OF_FILE

Theory of operation

MARKS_TO_DESCRIPTOR displays a file form for the base name of the input and output files. It combines easting-northing-depths from the .M001 files with label information from key and writes them to a file called TEMP. Number of marks is counted and written to the NAME.MARK file followed by all the data in TEMP. A flowchart is shown in Figure C40.

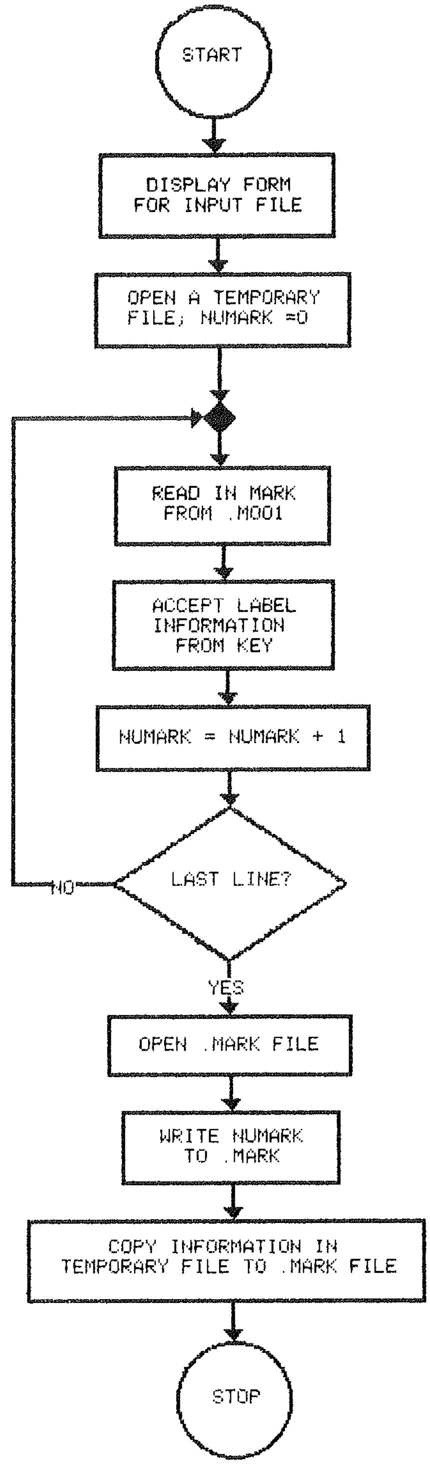


Figure C40. Flowchart, MARKS_TO_DESCRIPTOR

MARKS_TO_FEATURES

Purpose

MARKS_TO_FEATURES allows marks collected with the survey boat to be converted to features (shorelines and other segmented curves).

Input files

NAME.M001 (where 001 is mark number)

Output files

NAME.FEAT

External devices

NONE

Modules called

PROBLEM, INITMENU, DRAWMENU, MESSAGESTACK, DRAWFORM, GET_NAME_OF_FILE

Theory of operation

MARKS_TO_FEATURES accepts the site name and opens a .MARK file. It then processes features as the operator inputs. The operator specifies which marks are to be combined to make a feature and what the feature should look like. The features specified are output to a .FEAT file. The flowchart is shown in Figure C41.

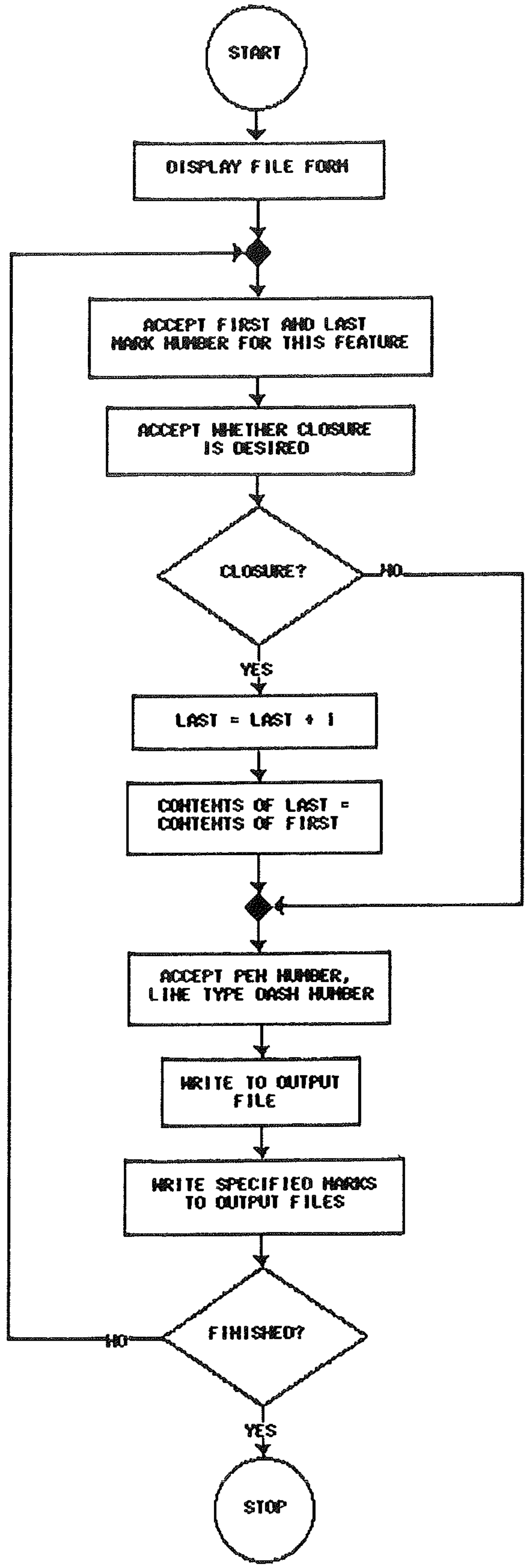


Figure C41. Flowchart, MARKS_TO_FEATURES

MARK_DESCRIPTION

Purpose

MARK_DESCRIPTION accepts mark information from key and builds a .MARK file for use by PLAN_PLOT in determining how to plot marks. It is used for marks not acquired by the survey program.

Input files

NONE

Output files

NAME.MARK

External devices

NONE

Modules called

PROBLEM, INITMENU, DRAWMENU, MESSAGESTACK, DRAWFORM, GET_NAME_OF_FILE

Theory of operation

MARK_DESCRIPTION begins by displaying a file form and accepting the output file name. It then asks for the number of marks to be entered and loops accepting information: (1) plotter pen carriage slot, (2) special symbol pattern, (3) symbol height, (4) easting, (5) northing, and (6) label. The flowchart is shown in Figure C42, and a sample run is shown in Figure C43.

Special considerations

MARK_DESCRIPTION is used to enter marks not collected by the survey program. MARKS_TO_DESCRIPTOR handles automatically collected marks. The text editor can be used to combine mark files (but remember to add the two numbers of marks and put the sum at the beginning while deleting the two addends.

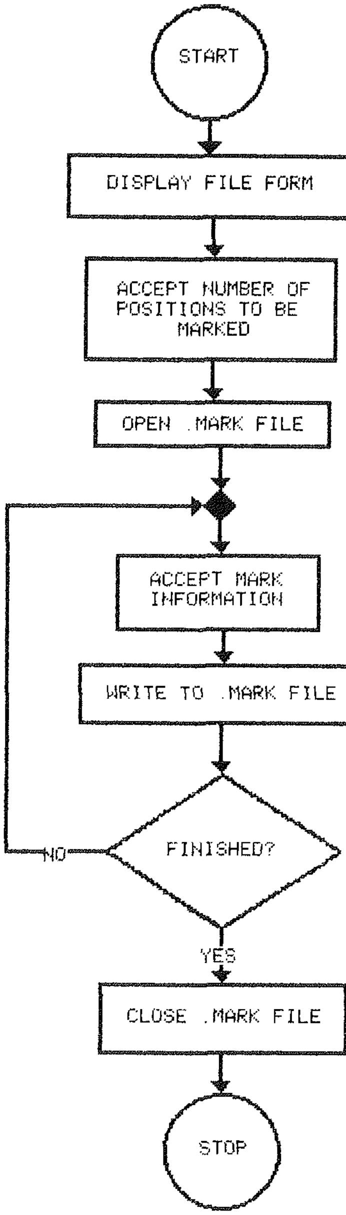


Figure C42. Flowchart, MARK_DESCRIPTION

```
TEXT STRING
OUTPUT FILENAME (-1 TO STOP)      LOUISVILLE
        ENTER INPUT FILE NAME AND CONFIRM
```

MARK_DESCRIPTION-1

```
 ENTER NUMBER OF POSITIONS TO BE MARKED
2
 THE FOLLOWING INFO IS FOR MARK NUMBER    1
 ENTER PEN NUMBER
1
 ENTER SPECIAL SYMBOL NUMBER
4
 ENTER SPECIAL SYMBOL SIZE IN INCHES
.1
 ENTER EASTING
752343.2
 ENTER NORTHING
987676
 ENTER LABEL
HUNTINGTON DISTRICT'S LIGHTHOUSE
 THE FOLLOWING INFO IS FOR MARK NUMBER    2
 ENTER PEN NUMBER
3
 ENTER SPECIAL SYMBOL NUMBER
2
```

MARK_DESCRIPTION-2

Figure C43. Sample run, MARK_DESCRIPTION

MOBILE_QUANTITY_FORMAT

Purpose

The CROSS_SECTION_AND_DREDGING program produces a large quantity of information (approximately 9 pages). MOBILE_QUANTITY_FORMAT grabs portions of this information and prints it on a one-page form. This program works with dual survey results: predredge and postdredge giving required, overdepth, and total removed.

Input files

QUANTITIES

Output files

NONE

Required hardware

Printer

Modules called

SKIP

Theory of operation

MOBILE_QUANTITY_FORMAT begins by getting the date/time from the system clock. It then opens QUANTITIES and gets the date the survey was run. It draws the form and fills it in as it goes using the following information (station number, required area, net area, overdredge area, gross area, net volume, overdredge volume, and gross volume). The flowchart is shown in Figure C44, and a sample run is shown in Figure C45.

Special considerations

CROSS_SECTION_AND_DREDGING must have been run prior to this program with pre- and postsurvey data.

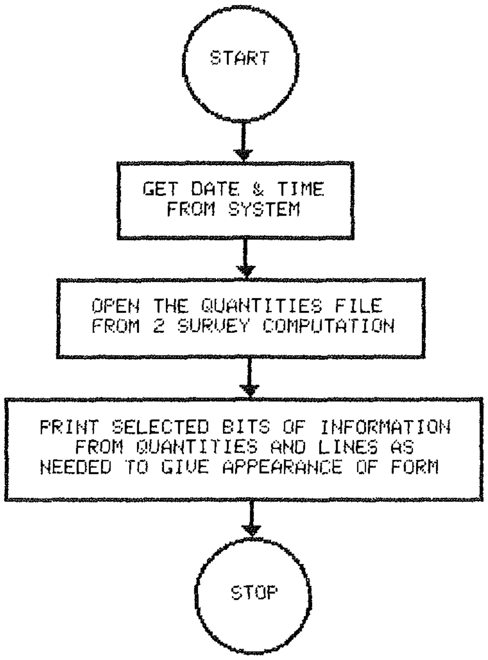


Figure C44. Flowchart, MOBILE_QUANTITY_FORMAT

YARDAGE COMPUTATIONS

LOCATION ____________________ BY ____________________

DATES SURVEYED: 5-24-89, 5-23-89 DATE COMPUTED: 06-06-89 SHEET NO. 1 OF 1

	CREDIT AREAS				CREDIT VOLUMES			
STA NO.	REQUIRE	NET	O.D.	GROSS	REQUIRED	NET	O.D	GROSS
332+00	1.3	1.3	0.0	26.4				
					452.	452.	0.	610.
334+00	120.7	120.7	0.0	138.2				
					-133.	-133.	0.	-129.
336+00	-156.6	-156.6	0.0	-173.2				
					-1318.	-1318.	0.	-1566.
338+00	-199.2	-199.2	0.0	-249.5				
					-1463.	-1463.	0.	-1536.
340+00	-195.0	-195.0	0.0	-164.5				
					19.	19.	0.	382.
342+00	200.1	200.1	0.0	267.6				
				TOTALS	-2442.	-2442.	0.	-2240.

MOBILE_QUANTITY_FORMAT-1

Figure C45. Sample run, MOBILE_QUANTITY_FORMAT

MOBILE_SINGLE_QUANTITY

Purpose

The CROSS_SECTION_AND_DREDGING program produces a large quantity of information (approximately 9 pages). MOBILE_SINGLE_QUANTITY grabs portions of this information and prints it on a one-page form. This program works with a single survey giving required and overdepth dredging quantities.

Input files

QUANTITIES

Output files

NONE

Required hardware

Printer

Modules called

SKIP

Theory of operation

MOBILE_SINGLE_QUANTITY begins by getting the date/time from the system clock. It then opens QUANTITIES and gets the date the survey was run. It draws the form and fills it in as it goes using the following information: station number, required area, net area, overdredge area, net volume, and overdredge volume. The flowchart is shown in Figure C46, and a sample run is shown in Figure C47.

Special considerations

CROSS_SECTION_AND_DREDGING must have been run prior to this program with pre- and postsurvey data.

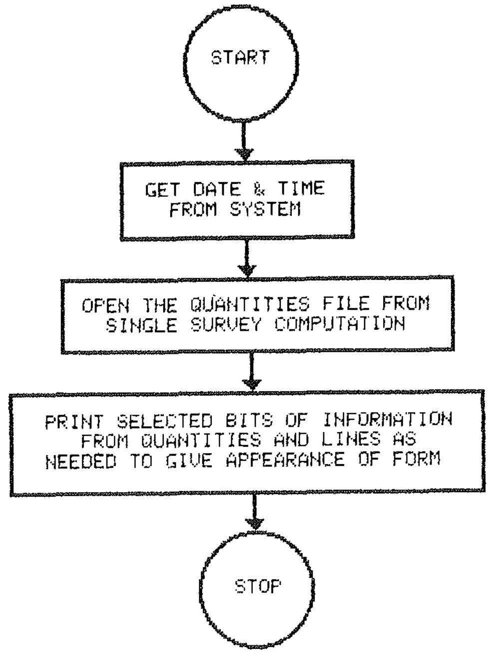


Figure C46. Flowchart, MOBILE_SINGLE_QUANTITY

YARDAGE COMPUTATIONS

LOCATION ______________________ BY ______________________

DATE SURVEYED: 5-24-89 DATE COMPUTED: 06-06-89 SHEET NO. 1 OF 1

	AREAS			VOLUMES		
STA NO.	REQUIRED	NET	O.D.	REQUIRED	NET	O.D.
332+00	4695.2	5894.2	1199.0			
				35521.8	44406.6	8884.9
334+00	4892.0	6091.0	1199.0			
				31114.1	39998.6	8884.5
336+00	3506.0	4705.0	1199.0			
				30374.5	39259.4	8884.9
338+00	4692.0	5891.0	1199.0			
				36142.9	45027.8	8884.9
340+00	5062.9	6261.9	1199.0			
				41361.3	50246.2	8884.9
342+00	6100.4	7299.4	1199.0			
			TOTALS	174514.6	218938.6	44424.0

MOBILE_SINGLE_QUANTITY-1

Figure C47. Sample run, MOBILE_SINGLE_QUANTITY

NEW_LINE_INPUT

Purpose

NEW_LINE_INPUT is the program that generates section line coordinates for use by SURVEY in section line or random mode, MANUAL_DATA_ENTRY, DIGITIZE_DEPTH_CHART, PLAN_PLOT, CROSS_SECTION_AND_DREDGING, TEMPLATE_MAKER, 3D_MAKE, and others. It additionally accepts and records channel coordinates, template information, and general comments.

Input files

NAME.SITE

Output files

NAME.SITE, NAME.LINE

External devices

Printer

Modules called

INITMENU, DRAWMENU, MESSAGESTACK, DRAWFORM, CHANNEL_MESSAGE, POLREC, A2, GENERATE_LINES, SLANTED_LINES, EQUATION_FROM_SLOPE_AND_1_POINT_EQUATION_PERPENDICULAR_TO_LINE, GEN, MINMAX, SCALE_SCREEN, DRAWLINE, FINDCOORDINATE, PLOT_CHANNEL_AND_SECTIONS, DRAWCHANNEL, DRAWCLBL, CENTER_LINE_PLOT, EXTRACT_FROM_SITE_BUFFER, INSERT_IN_SITE_BUFFER, ONE_PORTION_SITE_BUFFER, PATCH_LENGTH, ONE_PART_SITE_BUFFER, GET_NAME_OF_FILE, DISTANCE_POINT_TO_A_LINE, EQUATION_FROM_2_POINTS, CHANNEL_LINES, INTERSECTION, POINT_ALONG_A_LINE, EXTEND_LINE

Theory of operation

The program begins by initializing some of the pointers used by the form package. Then a message about use of the escape key is written to device 6, the screen. Next, the file form is displayed by calling DRAWFORM with a series of arguments beginning with INFCOMMANDSTR. If the operator decides to start with an existing .SITE file and modify it, then INFCHOS would have the value of 1. If this is the case, GET_NAME_OF_FILE is called, the desired file is opened as device 3, and the site information is read in. If the file is not confirmed, but rather escaped, then escape sequence handling is begun by calling DRAWFORM with EXITCOMMANDSTR. A flag, MODIFY_OLD, is set true if a file is read in.

The next section of code deals with entering general text information for archival purposes. If MODIFY_OLD is true, the contents of the form are filled with the information that was read from disk. Then from one to five forms (as determined by inputs) are displayed for user modification. DRAWFORM is called with SITECOMMANDSTR as the first argument. Then the new contents of the form are saved over original contents via a call to INSERT_IN_SITE_BUFFER.

The next section of code deals with entering channel coordinates and template information. If MODIFY_OLD is true, the contents of the forms are filled with the information that was read from disk. Then from 1 to 50 sets of channel coordinates are displayed for user modification. One menu and one form are presented for each coordinate index. The information concerning the present channel index is displayed by calls to CHANNEL_MESSAGE. The menu is displayed by calling DRAWMENU with NEXTCHCOMMANDSTR as the first argument. The menu allows choice of modify, insert before, or delete the present channel index, and values of 1, 2, or 3, respectively, are returned for the argument CHITEMCHOS. If 2 or 3 is chosen, the channel indices from present on are either moved back or forward one index. DRAWFORM is then called with CHANCOMMANDSTR as the first argument. The new form contents are then saved over the old in memory.

Section line generation is the subject of the next section of code. There are three methods of defining section lines and each uses a set of forms. The method and disposition of the current method are selected in a form generated by a call to DRAWFORM with a first argument of COMMANDSTRNLF. ITEMGROUP1 has a value on return of 1, 2, or 3, corresponding to Rectangular Method, Nonrectangular Method, or Channel Based Method, respectively. ITEMGROUP2 has a value on return of 1, 2, or 3, corresponding to modify current index, delete current index, or insert before current index, respectively. As before, the form is preloaded from data read from disk if MODIFY_OLD is true and the new contents replace the old in memory. If ITEMGROUP2 is 2 or 3, the group indices from present on are either moved back or forward one index. If ITEMGROUP1 is 1 (Rectangular), DRAWFORM is called with COMMANDSTR as the first argument, MAINFORMPTR as the second. If ITEMGROUP1 is 2 (Nonrectangular), DRAWFORM is called twice. The first call is with COMMANDSTR as the first argument, MAINFORMPTR as the second; the second call is with SECONDSTR. If ITEMGROUP1 is 3 (Rectangular), DRAWFORM is called with COMMANDSTR as the first argument, CHANGRFORMPTR as the second. Section lines are generated from the information input by calls to subroutines. By ordinal value of ITEMGROUP1, calls are made to (1) GENERATE_LINES,

(2) SLANTED_LINES, or (3) CHANNEL_LINES.

The channel lines entered and section lines computed are then displayed. This is accomplished by calling PLOT_CHANNEL_AND_SECTIONS. A disposition form is displayed on top of the channel by a call to DRAWFORM with a first argument of OUFCOMMANDSTR. This form allows selecting the output file name and what is to be printed. If OUTFCHOS equals one, data are to be saved; if PRINTCHOS is not equal to one, then printing of either site(2), line(3), or both site and line(4) information was selected.

The flowchart is shown in Figure C48, and a sample run is shown in Figure C49.

Special considerations

The left and right side of the channel are arbitrary, but must be consistent throughout. Also, the left channel side coordinate of a section line should be entered first, then the right.

The channel index associated with a section line must correspond to the channel section in which the line falls. A group of section lines generated by a single input cannot span across channel sections, because the channel index is input for the whole group (if this is a problem, the line file may be edited to correct the channel index numbers).

Channel based is the easiest way to generate lines, but it may not associate the desired station number with the line. Often it is desirable to generate lines channel based to get the coordinates at certain points, then go back and regenerate the lines using rectangular or nonrectangular. Channel index numbers when using rectangular or nonrectangular input methods are defined by relationship to the channel coordinate pairs.

Channel coordinates

Assume a survey with channel coordiante pairs

1. A(L),A(R)
2. B(L),B(R)
3. C(L),C(R)
4. D(L),D(R)
5. E(L),E(R)
6. F(L),F(R)
7. G(L),G(R)
8. H(L),H(R)
9. I(L),I(R)

where

A(L) is A(LE),A(LN)

A(R) is A(RE),A(RN)

Channel indices are determined by an area of channel plan view with index number of first channel coordinate pair used and are shown as 1-8.

Section lines

If section lines are superimposed as shown, the group index number should be as follows:

Section Group	No. of Lines	Channel Index No.
1	2	1
2	1	1
3	2	4
4	2	6
5	2	6
6	1	7

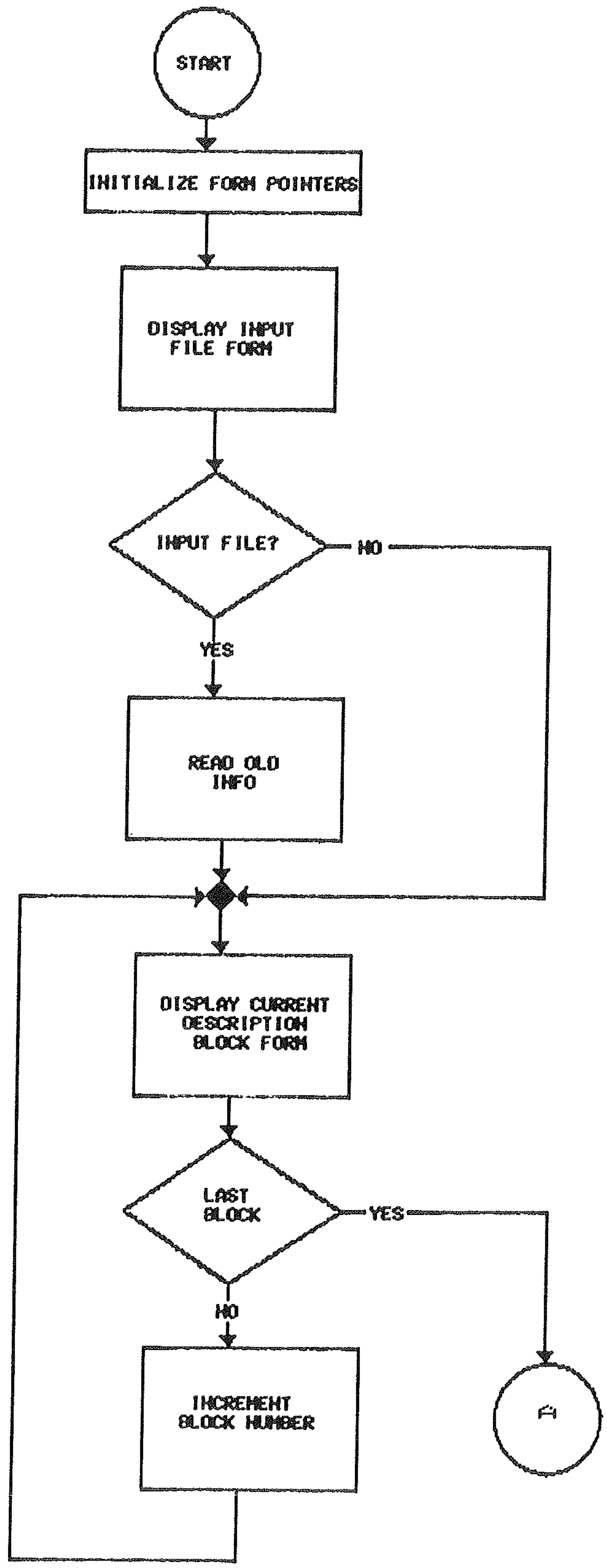


Figure C48. Flowchart, NEW_LINE_INPUT (Sheet 1 of 5)

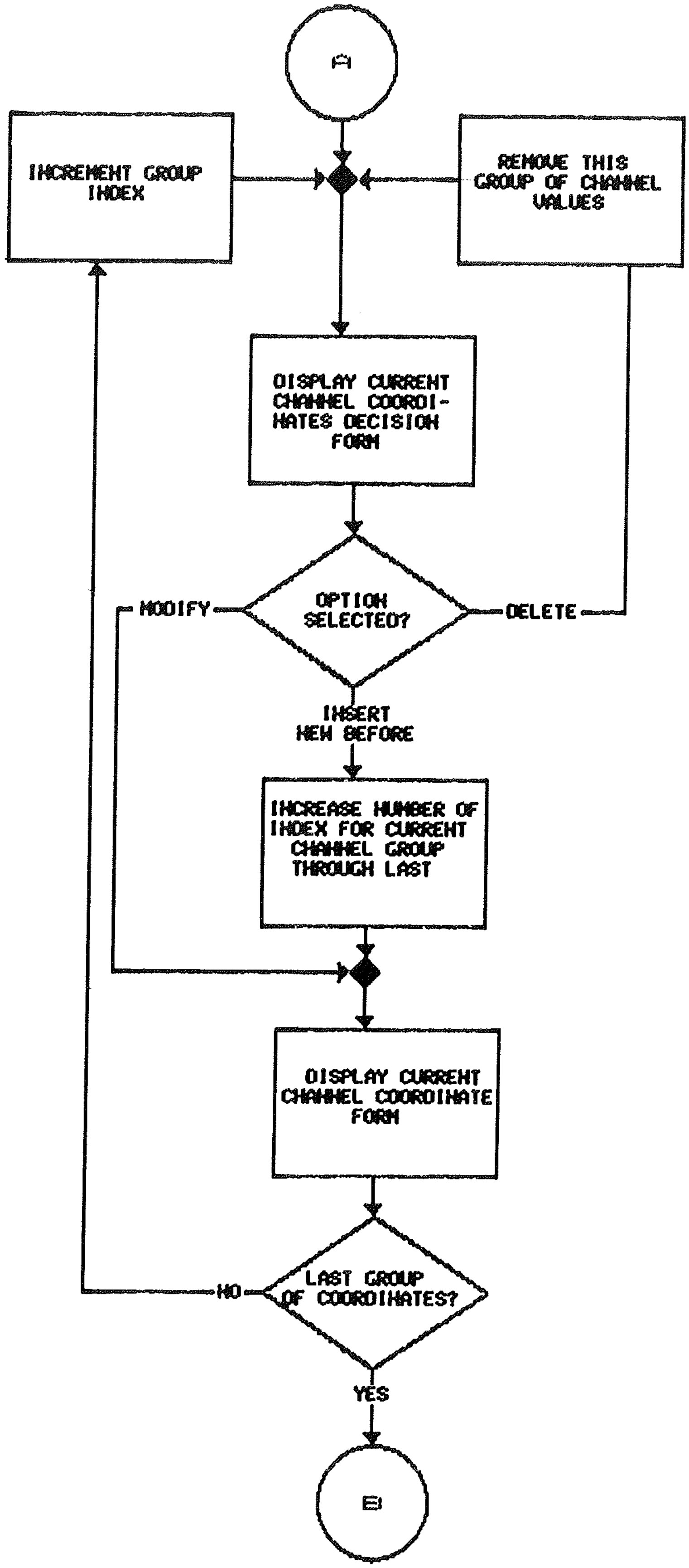
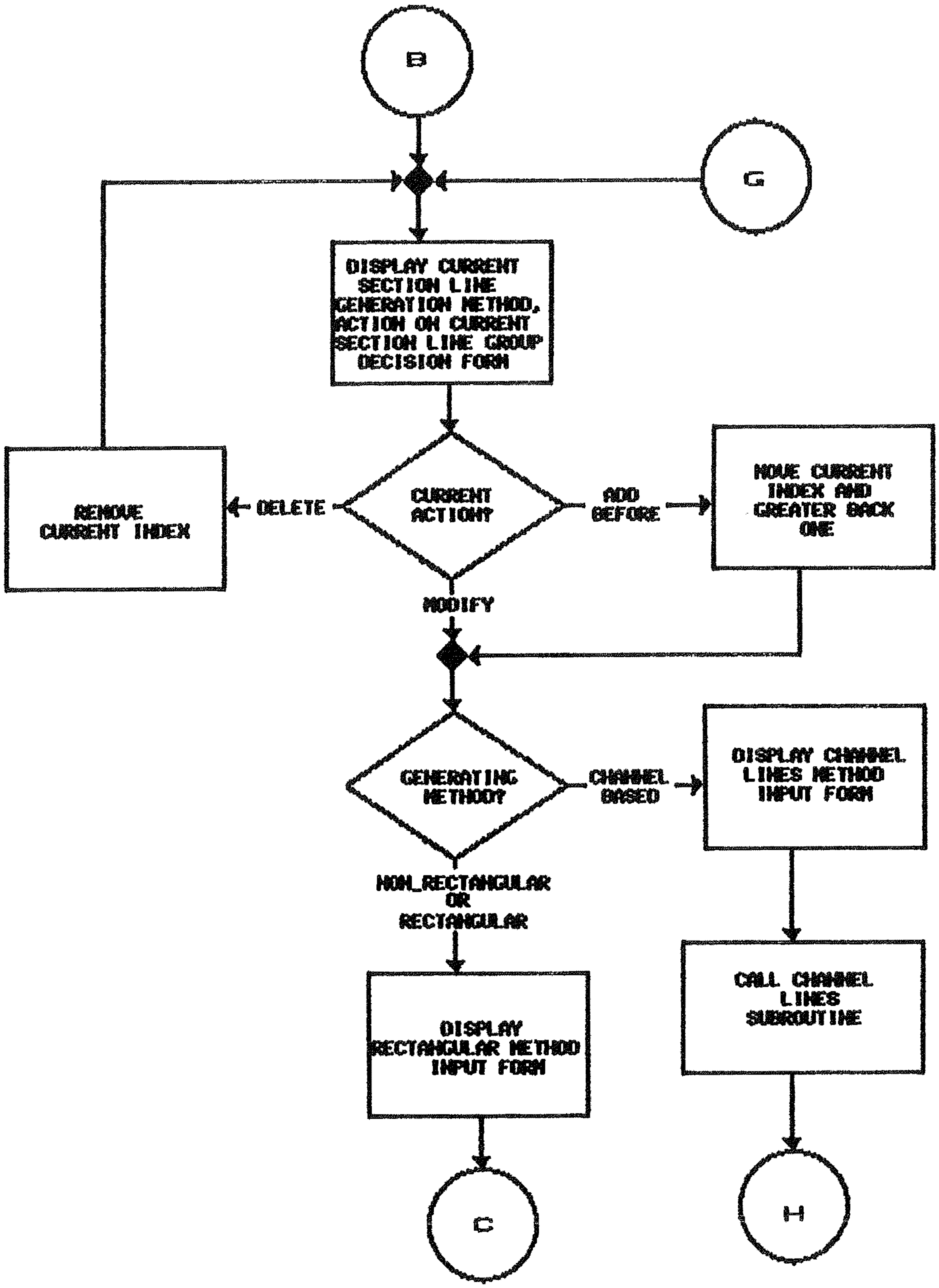


Figure C48. (Sheet 2 of 5)



Sheet C48. (Sheet 3 of 5)

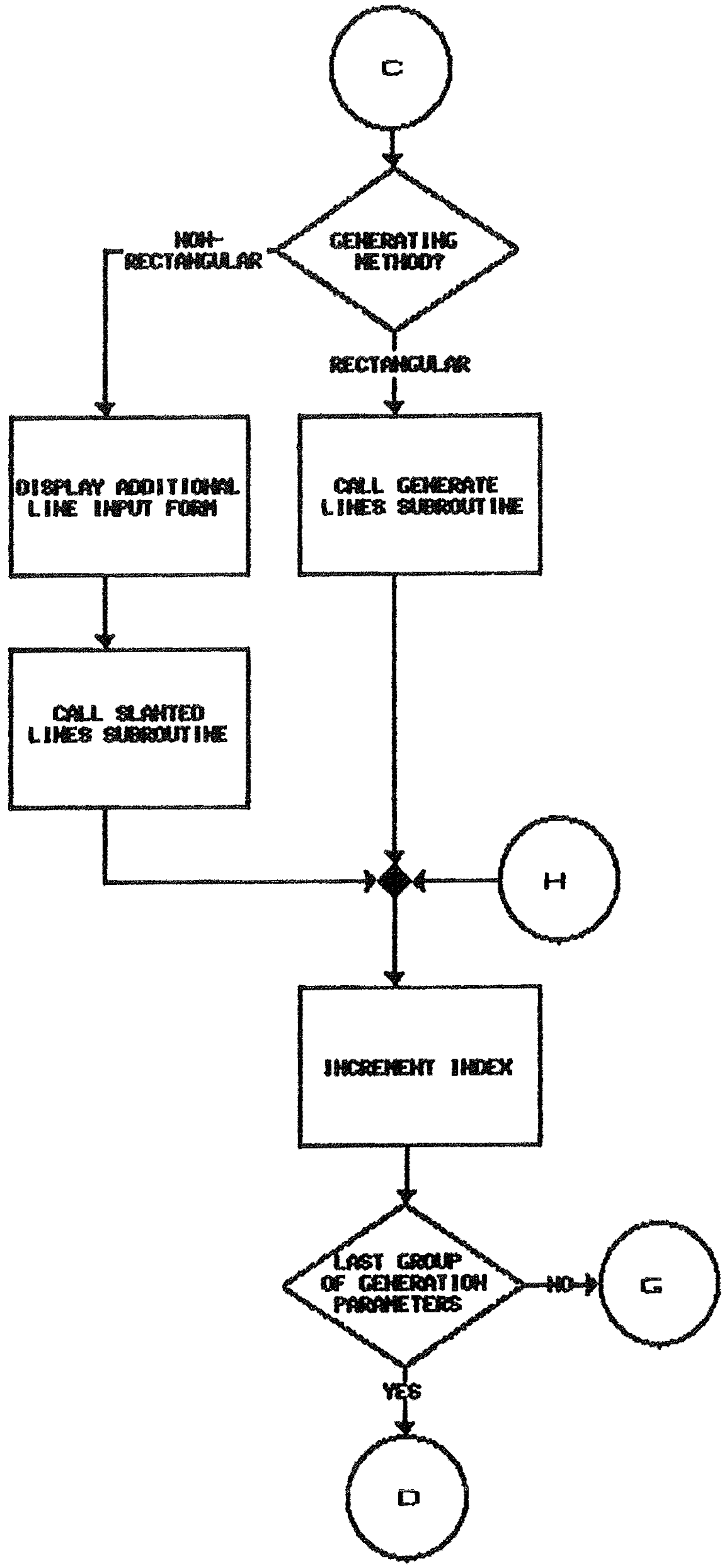


Figure C48. (Sheet 4 of 5)

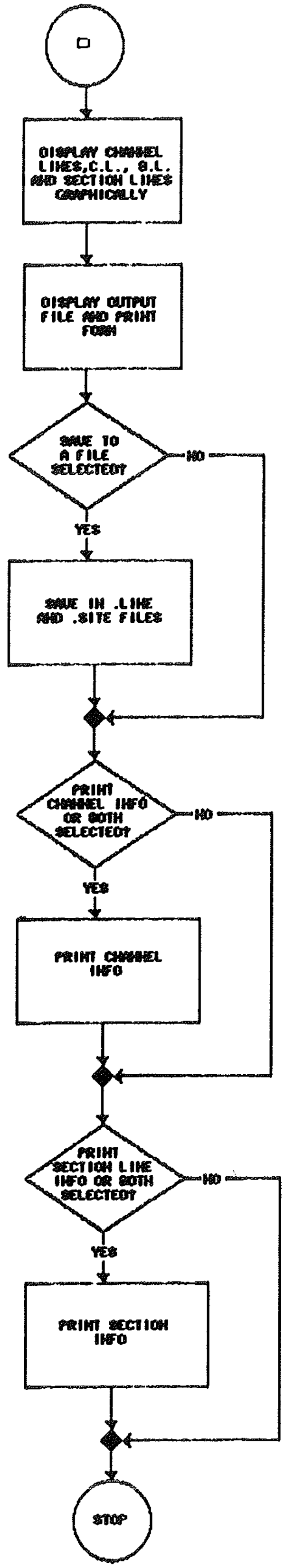


Figure C48. (Sheet 5 of 5)

```
*******************************************
CODE RETURN CONFIRMS A FORM
ESCAPE (NOT CODE ESCAPE) ALLOWS ESCAPING
*******************************************
```

```
A TEXT STRING  NO INPUT FILE
INPUT FILE NAME [MEMPHIS                                  ]
          ENTER INPUT FILE NAME AND CONFIRM
```

NEW_LINE_INPUT-1

```
*******************************************
CODE RETURN CONFIRMS A FORM
ESCAPE (NOT CODE ESCAPE) ALLOWS ESCAPING
*******************************************
```

```
A TEXT STRING
LINE 1      [MEMPHIS, TENNESSEE                          ]
LINE 2       MAY 1989
LINE 3       COE HYDROGRAPHIC SURVEY SCHOOL
LINE 4       IMC RANGE AZIMUTH SMALL BOAT SYSTEM
LINE 5       INNERSPACE DIGITIZER
MORE INFO?  YES
          ENTER SITE DESCRIPTION BLOCK  1
```

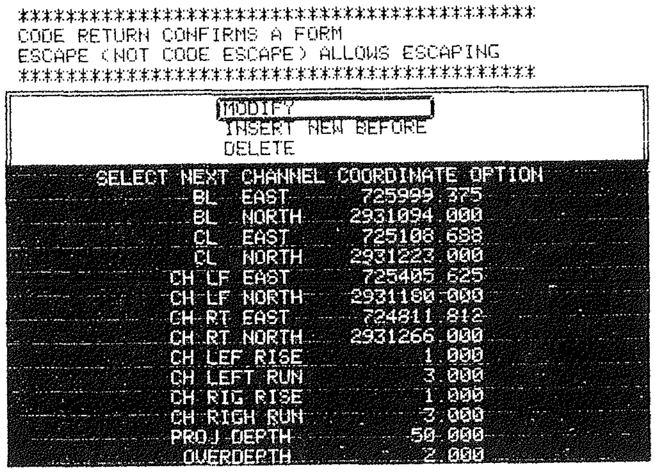
NEW_LINE_INPUT-2

```
*******************************************
CODE RETURN CONFIRMS A FORM
ESCAPE (NOT CODE ESCAPE) ALLOWS ESCAPING
*******************************************
```

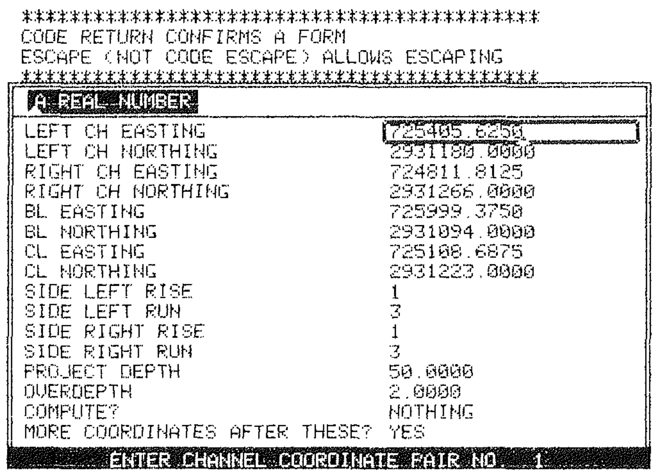
```
A TEXT STRING
LINE 1      [ROGER BUSH FROM MOBILE DISTRICT             ]
LINE 2       PANAMA CITY OFFICE IS SURVEYOR
LINE 3       USING MEMPHIS BOAT AND EMC
LINE 4       THEODOLITE
LINE 5
MORE INFO?  NO
          ENTER SITE DESCRIPTION BLOCK  2
```

NEW_LINE_INPUT-3

Figure C49. Sample run, NEW_LINE_INPUT
(Sheet 1 of 6)



NEW_LINE_INPUT-4



NEW_LINE_INPUT-5

NEW_LINE_INPUT-6

Figure C49. (Sheet 2 of 6)

```
********************************************
CODE RETURN CONFIRMS A FORM
ESCAPE (NOT CODE ESCAPE) ALLOWS ESCAPING
********************************************
A REAL NUMBER
LEFT CH EASTING                725319.3750
LEFT CH NORTHING               2930586.0000
RIGHT CH EASTING               724725.6250
RIGHT CH NORTHING              2930672.0000
BL EASTING                     725913.1875
BL NORTHING                    2930500.0000
CL EASTING                     725022.5000
CL NORTHING                    2930630.0000
SIDE LEFT RISE                 1
SIDE LEFT RUN                  3
SIDE RIGHT RISE                1
SIDE RIGHT RUN                 3
PROJECT DEPTH                  50.0000
OVERDEPTH                      2.0000
COMPUTE?                       NOTHING
MORE COORDINATES AFTER THESE?  YES
        ENTER CHANNEL COORDINATE PAIR NO. 2
```

NEW_LINE_INPUT-7

```
********************************************
CODE RETURN CONFIRMS A FORM
ESCAPE (NOT CODE ESCAPE) ALLOWS ESCAPING
********************************************
                MODIFY
                INSERT NEW BEFORE
                DELETE
      SELECT NEXT CHANNEL COORDINATE OPTION
             BL  EAST        725913.125
             BL  NORTH      2930500.000
             CL  EAST        725022.375
             CL  NORTH      2930628.000
           CH LF EAST        725319.312
           CH LF NORTH      2930586.000
           CH RT EAST        724725.500
           CH RT NORTH      2930672.000
           CH LEF RISE            1.000
           CH LEFT RUN            3.000
           CH RIG RISE            1.000
           CH RIGH RUN            3.000
           PROJ DEPTH            50.000
            OVERDEPTH             2.000
```

NEW_LINE_INPUT-8

```
********************************************
CODE RETURN CONFIRMS A FORM
ESCAPE (NOT CODE ESCAPE) ALLOWS ESCAPING
********************************************
A REAL NUMBER
LEFT CH EASTING                725319.3125
LEFT CH NORTHING               2930586.0000
RIGHT CH EASTING               724725.5000
RIGHT CH NORTHING              2930672.0000
BL EASTING                     725913.1250
BL NORTHING                    2930500.0000
CL EASTING                     725022.3750
CL NORTHING                    2930628.0000
SIDE LEFT RISE                 1
SIDE LEFT RUN                  3
SIDE RIGHT RISE                1
SIDE RIGHT RUN                 3
PROJECT DEPTH                  50.0000
OVERDEPTH                      2.0000
COMPUTE?                       NOTHING
MORE COORDINATES AFTER THESE?  YES
        ENTER CHANNEL COORDINATE PAIR NO. 3
```

NEW_LINE_INPUT-9

Figure C49. (Sheet 3 of 6)

```
*******************************************
CODE RETURN CONFIRMS A FORM
ESCAPE (NOT CODE ESCAPE) ALLOWS ESCAPING
*******************************************

                 MODIFY
                 INSERT NEW BEFORE
                 DELETE
      SELECT NEXT CHANNEL COORDINATE OPTION
              BL  EAST        725855.688
              BL  NORTH      2930104.000
              CL  EAST        724965.000
              CL  NORTH      2930233.000
           CH LF  EAST        725261.875
           CH LF  NORTH      2930190.000
           CH RT  EAST        724668.125
           CH RT  NORTH      2930276.000
           CH LEF RISE             1.000
           CH LEFT RUN             3.000
           CH RIG RISE             1.000
           CH RIGH RUN             3.000
           PROJ DEPTH             50.000
            OVERDEPTH              2.000
```

NEW_LINE_INPUT-10

```
*******************************************
CODE RETURN CONFIRMS A FORM
ESCAPE (NOT CODE ESCAPE) ALLOWS ESCAPING
*******************************************
 A REAL NUMBER
LEFT CH EASTING                    725261.8750
LEFT CH NORTHING                   2930190.0000
RIGHT CH EASTING                   724668.1250
RIGHT CH NORTHING                  2930276.0000
BL EASTING                         725855.6875
BL NORTHING                        2930104.0000
CL EASTING                         724965.0000
CL NORTHING                        2930233.0000
SIDE LEFT RISE                     1
SIDE LEFT RUN                      3
SIDE RIGHT RISE                    1
SIDE RIGHT RUN                     3
PROJECT DEPTH                      50.0000
OVERDEPTH                          2.0000
COMPUTE?                           NOTHING
MORE COORDINATES AFTER THESE?      NO
        ENTER CHANNEL COORDINATE PAIR NO.  4
```

NEW_LINE_INPUT-11

```
*******************************************
CODE RETURN CONFIRMS A FORM
ESCAPE (NOT CODE ESCAPE) ALLOWS ESCAPING
*******************************************


 RECTANGULAR  NONRECTANGULAR  CHANNEL BASED
SECTION METHOD          CHANNEL BASED
ACTION ON NEXT GROUP    MODIFY
      SELECT SECTION INPUT METHOD FOR GROUP  1
```

NEW_LINE_INPUT-12

Figure C49. (Sheet 4 of 6)

```
*******************************************
CODE RETURN CONFIRMS A FORM
ESCAPE (NOT CODE ESCAPE) ALLOWS ESCAPING
*******************************************
```

```
A REAL NUMBER
STATION START                           332.0000
STATION INCREMENT                       2.0000
EXTENSION BEYOND CHANNEL LEFT SIDE      100.0000
EXTENSION BEYOND CHANNEL RIGHT SIDE     100.0000
SEPARATION                              200.0000
STARTING CHANNEL COORDINATE INDEX       1
CHANNEL END LINES TO INCLUDE            START
MORE GROUPS AFTER THIS?                 YES
            ENTER ALL ITEMS AND CONFIRM
```

NEW_LINE_INPUT-13

```
*******************************************
CODE RETURN CONFIRMS A FORM
ESCAPE (NOT CODE ESCAPE) ALLOWS ESCAPING
*******************************************
```

```
RECTANGULAR  NONRECTANGULAR  CHANNEL BASED
SECTION METHOD          CHANNEL BASED
ACTION ON NEXT GROUP    MODIFY
      SELECT SECTION INPUT METHOD FOR GROUP  2
```

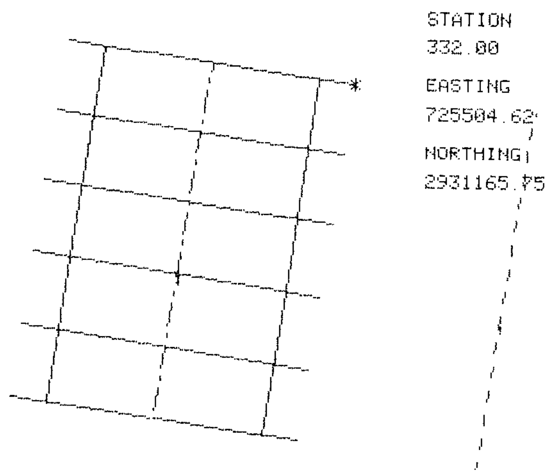
NEW_LINE_INPUT-14

```
*******************************************
CODE RETURN CONFIRMS A FORM
ESCAPE (NOT CODE ESCAPE) ALLOWS ESCAPING
*******************************************
```

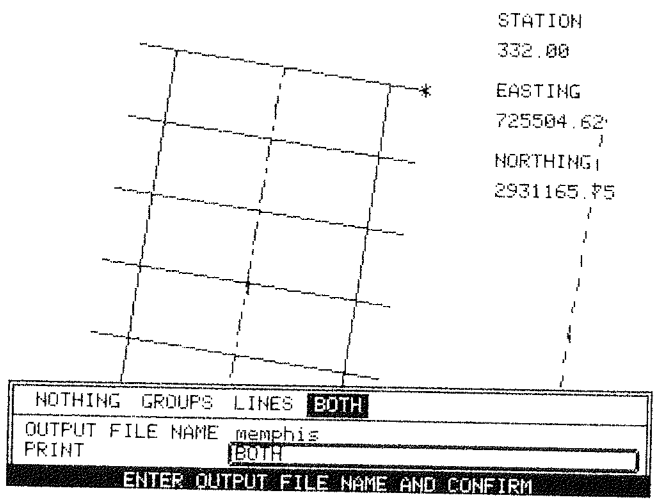
```
A REAL NUMBER
STATION START                           340.0000
STATION INCREMENT                       2.0000
EXTENSION BEYOND CHANNEL LEFT SIDE      100.0000
EXTENSION BEYOND CHANNEL RIGHT SIDE     100.0000
SEPARATION                              200.0000
STARTING CHANNEL COORDINATE INDEX       3
CHANNEL END LINES TO INCLUDE            BOTH
MORE GROUPS AFTER THIS?                 NO
            ENTER ALL ITEMS AND CONFIRM
```

NEW_LINE_INPUT-15

Figure C49. (Sheet 5 of 6)



NEW_LINE_INPUT-16



NEW_LINE_INPUT-17

Figure C49. (Sheet 6 of 6)

NEW_SPOT_INPUT

Purpose

NEW_SPOT_INPUT accepts coordinate inputs and builds a .SPOT file for input to the FIND_SPOT survey program.

Input files

NAME.SPOT

Output files

NAME.SPOT

External devices

NONE

Modules called

PROBLEM, INITMENU, DRAWMENU, MESSAGESTACK, DRAWFORM, GET_NAME_OF_FILE, CHANNEL_MESSAGE

Theory of operation

NEW_SPOT_INPUT begins by displaying an input file form. It then reads in the spot coordinates to be modified (if any) and loops accepting new or changed coordinates. The flowchart is shown in Figure C50, and a sample run is shown in Figure C51.

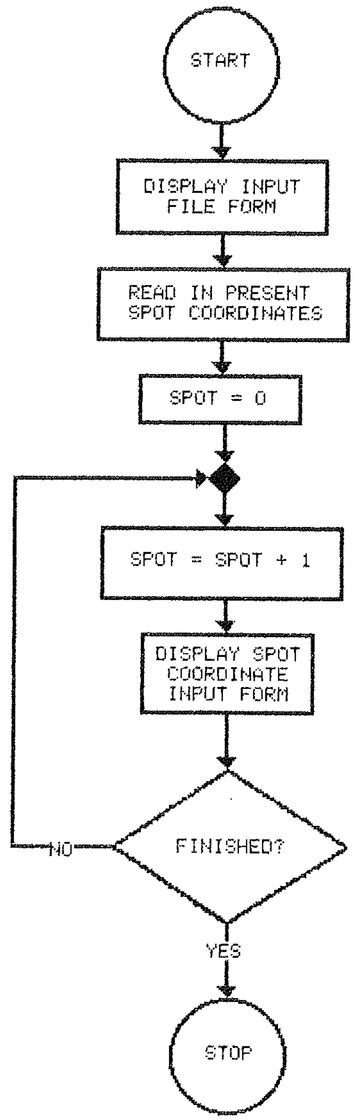


Figure C50. Flowchart,
NEW_SPOT_INPUT

```
*******************************************
CODE RETURN CONFIRMS A FORM
ESCAPE (NOT CODE ESCAPE) ALLOWS ESCAPING
```

```
A TEXT STRING  NO INPUT FILE
INPUT FILE NAME  GROUPD
        ENTER INPUT FILE NAME AND CONFIRM
```

NEW_SPOT_INPUT-1

```
*******************************************
CODE RETURN CONFIRMS A FORM
ESCAPE (NOT CODE ESCAPE) ALLOWS ESCAPING
```

```
                MODIFY
                INSERT NEW BEFORE
                DELETE
     SELECT NEXT SPOT COORDINATE OPTION
          SPOT EAST       726085.500
          SPOT NORTH     2931687.750
```

NEW_SPOT_INPUT-2

```
*******************************************
CODE RETURN CONFIRMS A FORM
ESCAPE (NOT CODE ESCAPE) ALLOWS ESCAPING
```

```
A REAL NUMBER
EASTING                              726085.5000
NORTHING                             2931687.7500
MORE SPOTS TO FIND AFTER THIS?  YES
     ENTER SPOT TO FIND COORDINATES NO.  1
```

NEW_SPOT_INPUT-3

Figure C51. Sample run, NEW_SPOT_INPUT (Continued)

```
********************************************
CODE RETURN CONFIRMS A FORM
ESCAPE (NOT CODE ESCAPE) ALLOWS ESCAPING
```

```
MODIFY
INSERT NEW BEFORE
DELETE
SELECT NEXT SPOT COORDINATE OPTION
SPOT EAST        725913.062
SPOT NORTH      2930499.500
```

NEW_SPOT_INPUT-4

```
********************************************
CODE RETURN CONFIRMS A FORM
ESCAPE (NOT CODE ESCAPE) ALLOWS ESCAPING
```

```
A REAL NUMBER
EASTING                              725913.0625
NORTHING                             2930499.5000
MORE SPOTS TO FIND AFTER THIS?  NO
ENTER SPOT TO FIND COORDINATES NO.  2
```

NEW_SPOT_INPUT-5

```
********************************************
CODE RETURN CONFIRMS A FORM
ESCAPE (NOT CODE ESCAPE) ALLOWS ESCAPING
```

```
A TEXT STRING  NO OUTPUT FILE
OUTPUT FILE NAME  GROUPD
FILL IN OUTPUT FILE FORM AND CONFIRM
```

NEW_SPOT_INPUT-6

Figure C51. (Concluded)

PILOT

Purpose

PILOT is one of the tasks spawned by SURVEY. It takes distance along the line and offset information from the common memory area and sends it out over the modem port to the pilot indicator computer.

Input files

NONE

Output files

NONE

External devices

Pilot Indicator Computer

Modules called

CONVASCI, INIMODEM, WRMODEM, TERMMO, PROBLEM, A2

Theory of operation

PILOT begins by making the modem connection, then waits for a flag from PLOT_SCREEN before proceeding. It then loops until the death flag is detected, sending out line coordinates if the new line flag is set or distance along the line versus depth if not. The flowchart is shown in Figure C52.

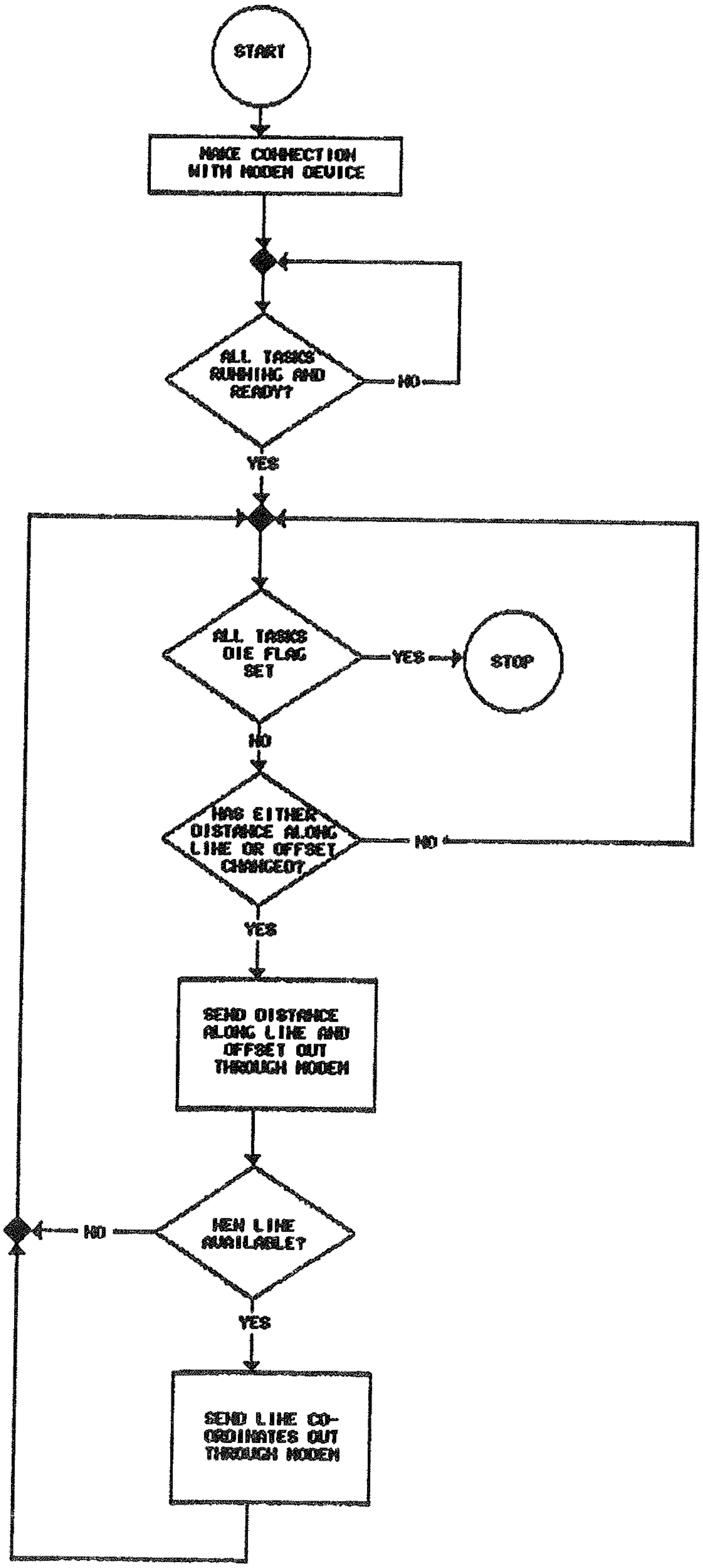


Figure C52. Flowchart, PILOT

PLAN_VIEW_DESCRIPTION

Purpose

PLAN_VIEW_DESCRIPTION is a form-driven program to provide easy input of plotting parameters required by PLAN_PLOT and CONTOUR_PLOT.

Input files

NAME.PLAN

Output files

NAME.PLAN

External devices

NONE

Modules called

PROBLEM, INITMENU, DRAWMENU, MESSAGESTACK, DRAWFORM, SET_LEGEND, GET_LEGEND, GET_NAME_OF_FILE

Theory of operation

PLAN_VIEW_DESCRIPTION displays a file form and reads in the present file description (if any). It then displays a series of forms to allow changing present parameters. Then it writes the changed parameters out to the selected file. The flowchart is shown in Figure C53, and a sample run is shown in Figure C54.

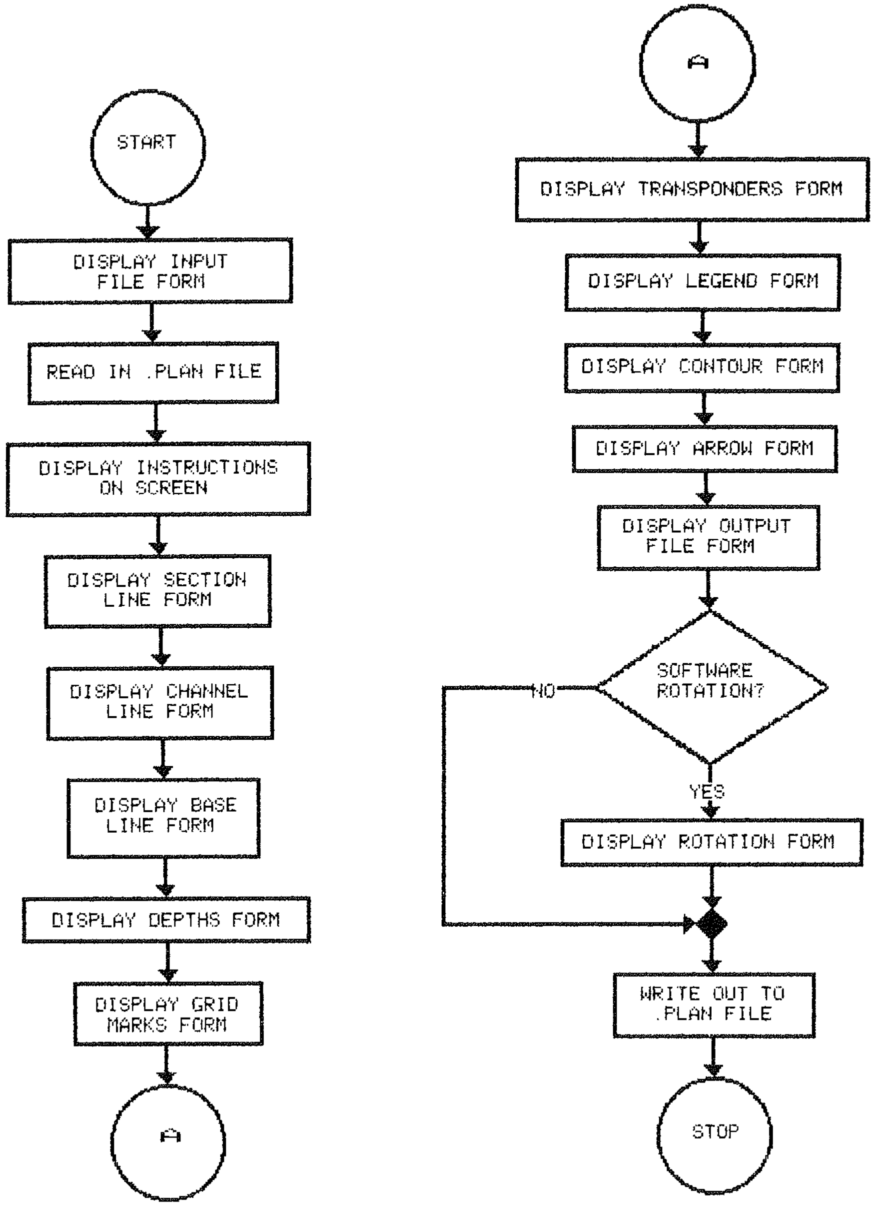


Figure C53. Flowchart
PLAN_VIEW_DESCRIPTION

```
A TEXT STRING  NO INPUT FILE
INPUT FILE NAME GROUPB
ENTER INPUT FILE NAME AND CONFIRM
```

PLAN_VIEW_DESCRIPTION-1

```
  *********************************************
CODE RETURN CONFIRMS A FORM
ESCAPE (NOT CODE ESCAPE) ALLOWS ESCAPING
*********************************************
IN THE FORMS THAT FOLLOW IF THE
FIRST ITEM IS CHOSEN TO BE
DO NOT PLOT
THEN THE REST OF THE ITEMS MAY
BE IGNORED (NOT FILLED IN)
*********************************************
```

```
 DO NOT PLOT  1  2  3  4  5  6  7  8
CARRIAGE SLOT FOR PEN  7
LINE TYPE              SOLID
DASH SIZE              MEDIUM
CHARACTER SIZE         0.1000
ENTER INFORMATION ABOUT SECTION LINES & CONFIRM
```

PLAN_VIEW_DESCRIPTION-2

```
  *********************************************
CODE RETURN CONFIRMS A FORM
ESCAPE (NOT CODE ESCAPE) ALLOWS ESCAPING
*********************************************
IN THE FORMS THAT FOLLOW IF THE
FIRST ITEM IS CHOSEN TO BE
DO NOT PLOT
THEN THE REST OF THE ITEMS MAY
BE IGNORED (NOT FILLED IN)
*********************************************
```

```
 DO NOT PLOT  1  2  3  4  5  6  7  8
CARRIAGE SLOT FOR PEN  4
LINE TYPE              DASH
DASH SIZE              LARGE
ENTER INFORMATION ABOUT CHANNEL LINES & CONFIRM
```

PLAN_VIEW_DESCRIPTION-3

Figure C54. Sample run, PLAN_VIEW_DESCRIPTION (Sheet 1 of 5)

```
 *******************************************
CODE RETURN CONFIRMS A FORM
ESCAPE (NOT CODE ESCAPE) ALLOWS ESCAPING
*******************************************
IN THE FORMS THAT FOLLOW IF THE
FIRST ITEM IS CHOSEN TO BE
DO NOT PLOT
THEN THE REST OF THE ITEMS MAY
BE IGNORED (NOT FILLED IN)
*******************************************
```

```
 SMALL  MEDIUM  LARGE  EXTRA LARGE
CARRIAGE SLOT FOR PEN  4
LINE TYPE              CENTER LINE
DASH SIZE              LARGE
   ENTER INFORMATION ABOUT CENTER LINE & CONFIRM
```

PLAN_VIEW_DESCRIPTION-4

```
 *******************************************
CODE RETURN CONFIRMS A FORM
ESCAPE (NOT CODE ESCAPE) ALLOWS ESCAPING
*******************************************
IN THE FORMS THAT FOLLOW IF THE
FIRST ITEM IS CHOSEN TO BE
DO NOT PLOT
THEN THE REST OF THE ITEMS MAY
BE IGNORED (NOT FILLED IN)
*******************************************
```

```
 1  DASH  3  4  CENTER LINE  6  SOLID
CARRIAGE SLOT FOR PEN  4
LINE TYPE              SOLID
DASH SIZE              MEDIUM
    ENTER INFORMATION ABOUT BASE LINE & CONFIRM
```

PLAN_VIEW_DESCRIPTION-5

```
 *******************************************
CODE RETURN CONFIRMS A FORM
ESCAPE (NOT CODE ESCAPE) ALLOWS ESCAPING
*******************************************
IN THE FORMS THAT FOLLOW IF THE
FIRST ITEM IS CHOSEN TO BE
DO NOT PLOT
THEN THE REST OF THE ITEMS MAY
BE IGNORED (NOT FILLED IN)
*******************************************
```

```
 DO NOT PLOT  1  2  3  4  5  6  7  8
CARRIAGE SLOT FOR PEN  6
CHARACTER SIZE         0.0700
    ENTER INFORMATION ABOUT DEPTHS & CONFIRM
```

PLAN_VIEW_DESCRIPTION-6

Figure C54. (Sheet 2 of 5)

```
********************************************
CODE RETURN CONFIRMS A FORM
ESCAPE (NOT CODE ESCAPE) ALLOWS ESCAPING
********************************************
IN THE FORMS THAT FOLLOW IF THE
FIRST ITEM IS CHOSEN TO BE
DO NOT PLOT
THEN THE REST OF THE ITEMS MAY
BE IGNORED (NOT FILLED IN)
********************************************
```

```
 DO NOT PLOT  1  2  3  4  5  6  7  8
CARRIAGE SLOT FOR PEN  7
CHARACTER SIZE         0.1000
  ENTER INFORMATION ABOUT GRID MARKS & CONFIRM
```

PLAN_VIEW_DESCRIPTION-7

```
********************************************
CODE RETURN CONFIRMS A FORM
ESCAPE (NOT CODE ESCAPE) ALLOWS ESCAPING
********************************************
IN THE FORMS THAT FOLLOW IF THE
FIRST ITEM IS CHOSEN TO BE
DO NOT PLOT
THEN THE REST OF THE ITEMS MAY
BE IGNORED (NOT FILLED IN)
********************************************
```

```
 DO NOT PLOT  1  2  3  4  5  6  7  8
CARRIAGE SLOT FOR PEN  DO NOT PLOT
CHARACTER SIZE         0.1000
  ENTER INFORMATION ABOUT TRANSPONDERS & CONFIRM
```

PLAN_VIEW_DESCRIPTION-8

```
********************************************
CODE RETURN CONFIRMS A FORM
ESCAPE (NOT CODE ESCAPE) ALLOWS ESCAPING
********************************************
IN THE FORMS THAT FOLLOW IF THE
FIRST ITEM IS CHOSEN TO BE
DO NOT PLOT
THEN THE REST OF THE ITEMS MAY
BE IGNORED (NOT FILLED IN)
********************************************
```

```
 LOWER LEFT  UPPER LEFT  LOWER RIGHT  UPPER RIGHT
CARRIAGE SLOT FOR PEN              6
LETTER HEIGHT TITLE                0.1500
TITLE                              GROUP-B
LETTER HEIGHT ENTRIES              0.1000
REFERENCE CORNER OF BOX            LOWER LEFT
X POSITION OF REFERENCE CORNER     0.0000
Y POSITION OF REFERENCE CORNER     0.0000
     FILL IN LEGEND DESCRIPTOR FORM & CONFIRM
```

PLAN_VIEW_DESCRIPTION-9

Figure C54. (Sheet 3 of 5)

```
********************************************
CODE RETURN CONFIRMS A FORM
ESCAPE (NOT CODE ESCAPE) ALLOWS ESCAPING
********************************************
IN THE FORMS THAT FOLLOW IF THE
FIRST ITEM IS CHOSEN TO BE
DO NOT PLOT
THEN THE REST OF THE ITEMS MAY
BE IGNORED (NOT FILLED IN)
********************************************
```

```
DO NOT PLOT  1  2  3  4  5  6  7  8
CARRIAGE SLOT FOR PEN        3
LETTER HEIGHT TITLE          0.1000
SHALLOWEST DEPTH TO CONTOUR  20.0000
DEAPEST DEPTH TO CONTOUR     50.0000
NUMBER OF CONTOUR LEVELS     4
     FILL IN CONTOUR DESCRIPTOR FORM & CONFIRM
```

PLAN_VIEW_DESCRIPTION-10

```
********************************************
CODE RETURN CONFIRMS A FORM
ESCAPE (NOT CODE ESCAPE) ALLOWS ESCAPING
********************************************
IN THE FORMS THAT FOLLOW IF THE
FIRST ITEM IS CHOSEN TO BE
DO NOT PLOT
THEN THE REST OF THE ITEMS MAY
BE IGNORED (NOT FILLED IN)
********************************************
```

```
DO NOT PLOT  1  2  3  4  5  6  7  8
CARRIAGE SLOT FOR PEN  1
LENGTH                 3.0000
X POSITION OF CENTER   18.0000
Y POSITION OF CENTER   16.0000
     FILL IN ARROW PARAMETERS FORM AND CONFIRM
```

PLAN_VIEW_DESCRIPTION-11

```
********************************************
CODE RETURN CONFIRMS A FORM
ESCAPE (NOT CODE ESCAPE) ALLOWS ESCAPING
********************************************
IN THE FORMS THAT FOLLOW IF THE
FIRST ITEM IS CHOSEN TO BE
DO NOT PLOT
THEN THE REST OF THE ITEMS MAY
BE IGNORED (NOT FILLED IN)
********************************************
```

```
DEGREES  0 HARDWARE  90 HARDWARE
OUTPUT FILE NAME          GROUPB
PLOT SCALE                83.3000
SPACING FOR GRID MARKS    250.0000
ROTATION ANGLE            23.7
PLOTTER PAPER ALIGNMENT   USE DEFAULT
     FILL IN PLOT PARAMETERS FORM AND CONFIRM
```

PLAN_VIEW_DESCRIPTION-12

Figure C54. (Sheet 4 of 5)

```
    *******************************************
CODE RETURN CONFIRMS A FORM
ESCAPE (NOT CODE ESCAPE) ALLOWS ESCAPING
*******************************************
IN THE FORMS THAT FOLLOW IF THE
FIRST ITEM IS CHOSEN TO BE
DO NOT PLOT
THEN THE REST OF THE ITEMS MAY
BE IGNORED (NOT FILLED IN)
*******************************************
```

```
IN INCHES
EASTING     789563.0000
NORTHING    2678943.0000
X POSITION  16.0000
Y POSITION  11.0000
      FILL IN POINT OF ROTATION FORM & CONFIRM
```

PLAN_VIEW_DESCRIPTION-13

Figure C54. (Sheet 5 of 5)

PLOT_FEATURES

Purpose

PLOT_FEATURES is a program to plot a particular .FEAT file scaled by the input information.

Input files

NAME.FEAT, NAME.LABE, FEATURES, LABELS

Output files

NONE

External devices

Plotter

Modules called

INITMENU, PLOTSC, DRAWMENU, DRAWFORM, GET_NAME_OF_FILE, POLREC, A2, PLOTROT, TRANSPOSE_ORIGIN, SYMBOLROT, PROBLEM, OA

Theory of operation

PLOT_FEATURES begins by displaying a file form and accepting the .FEAT file name for plotting. It then accepts plot parameters from key. Then it reads in the .FEAT file and plots the features. Next it reads in the the names of other .FEAT files from the file FEATURES and plots the features in the .LABE file and plots the labels. Finally it reads in the names of other .LABE from the file LABELS and plots the labels in these files. The flowchart is shown in Figure C55, and a sample run is shown in Figure C56.

Special considerations

Delete the FEATURES and LABELS files (or clear their contents) when not in use.

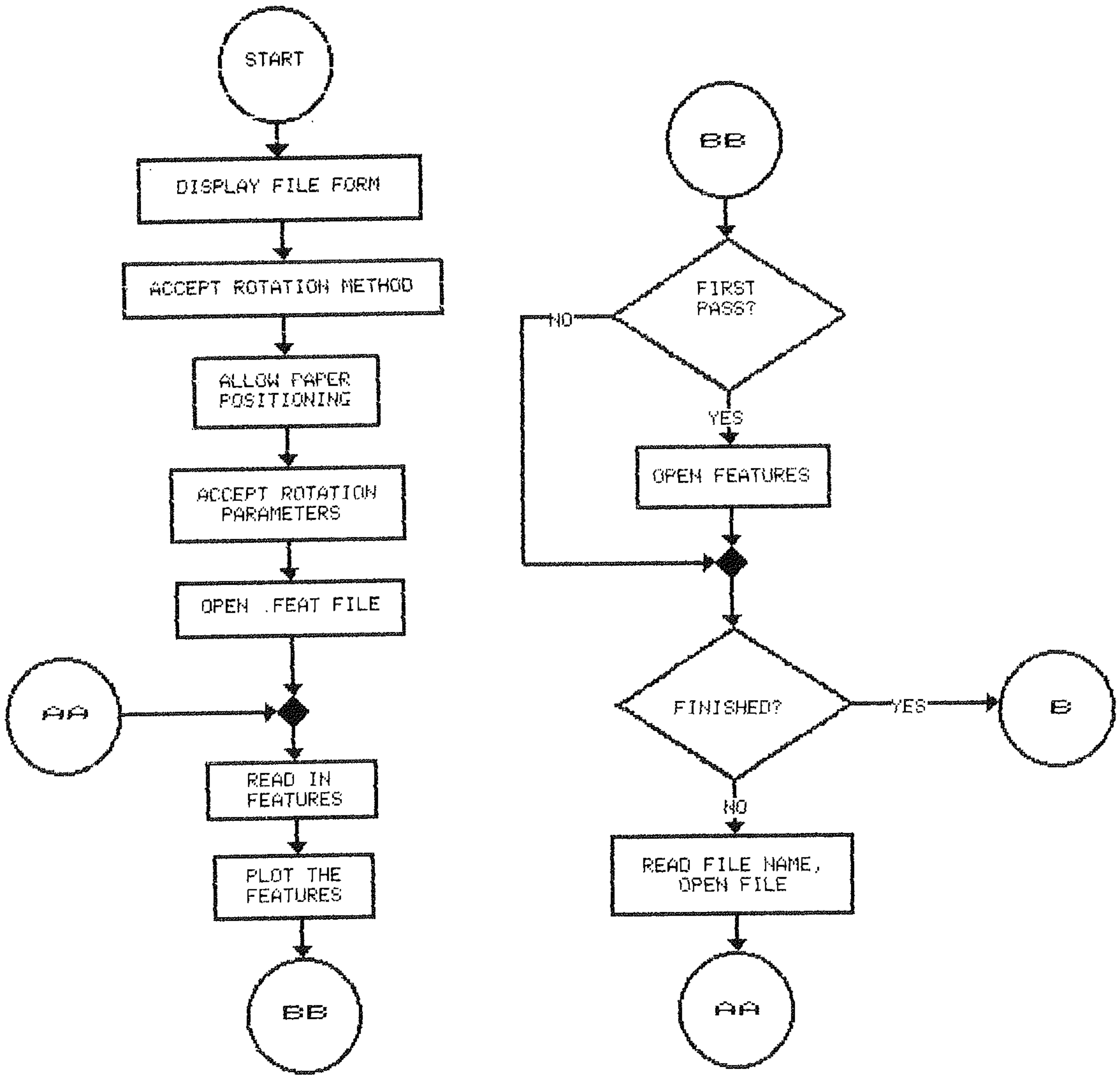


Figure C55. Flowchart,
PLOT_FEATURES
(Continued)

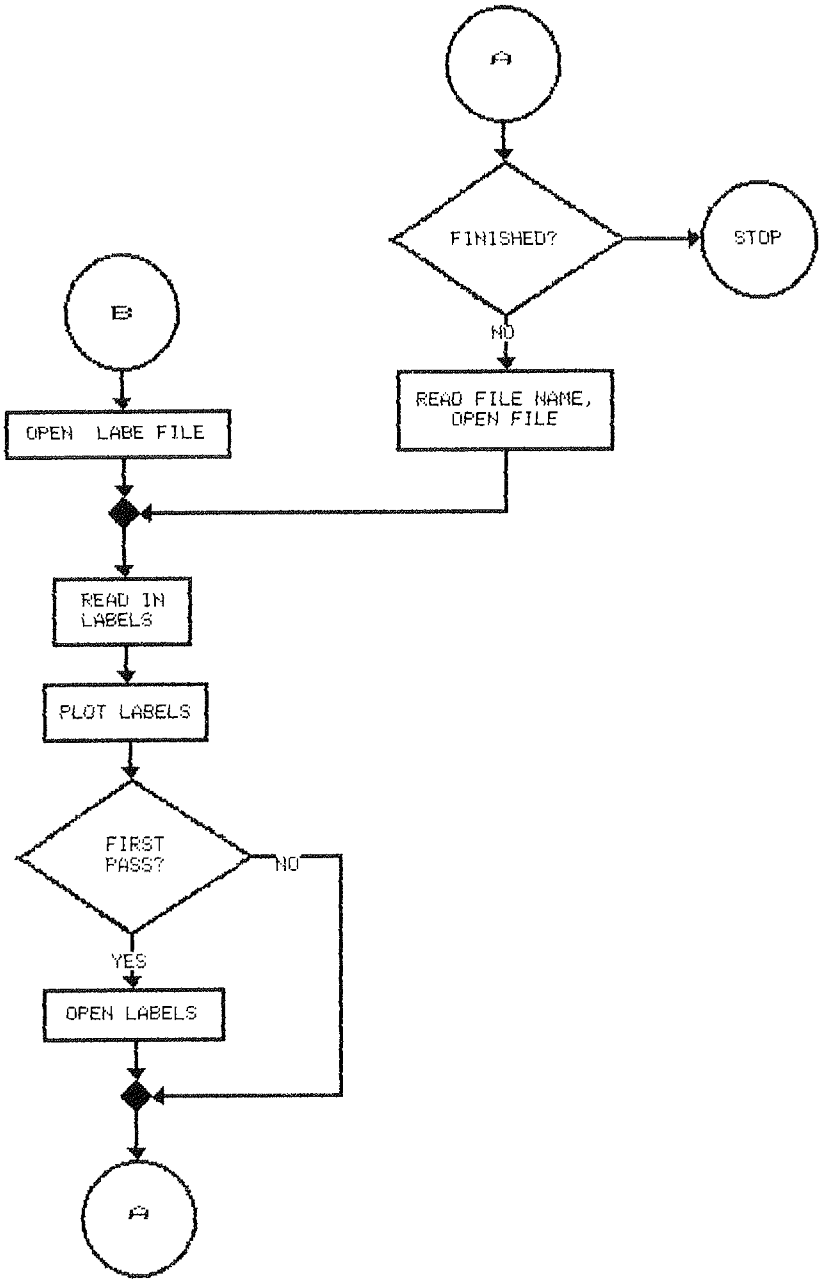


Figure C55. (Concluded)

```
TEXT STRING
INPUT FILE NAME (-1 TO STOP)    GENERAL
ENTER INPUT FILE NAME AND CONFIRM
```

PLOT_FEATURES-1

```
ENTER     0     FOR 0 DEGREE HARDWARE ROTATION
         90     FOR 90 DEGREE HARDWARE ROTATION
         -1     FOR SOFTWARE ROTATION
0
 PEN IS POSITIONED AT LOWER LEFT CORNER
 USE JOYSTICK TO ALIGN PEN ON GRID
 ENTER RETURN WHEN FINISHED

ENTER LOWER LEFT EASTING
750000
ENTER LOWER LEFT NORTHING
2100000
ENTER SCALE IN FEET/INCH
83.3
```

PLOT_FEATURES-2

Figure C56. Sample run, PLOT_FEATURES

PLOT_LETTERS

Purpose

PLOT_LETTERS is a program to plot .LETT files listed in the file LETTERS.

Input files

NAME.LETT, LETTERS, OTHER.LETT files

Output files

NONE

External devices

Plotter

Modules called

INITMENU, DRAWMENU, DRAWFORM, GET_NAME_OF_FILE, PLOTROT, TRANSPOSE_ORIGIN, SYMBOLROT, PROBLEM

Theory of operation

PLOT_LETTERS begins by displaying a file form and accepting the .LETT file name for plotting. Then it reads in the .LETT file and plots the lines and letters defined. Next it reads in the the names of other .LETT files from the file LETTERS and plots the lines and letters in the .LETT file. The flowchart is shown in Figure C57, and a sample run is shown in Figure C58.

Special Considerations

Delete the LETTERS file (or clear its contents) when not in use.

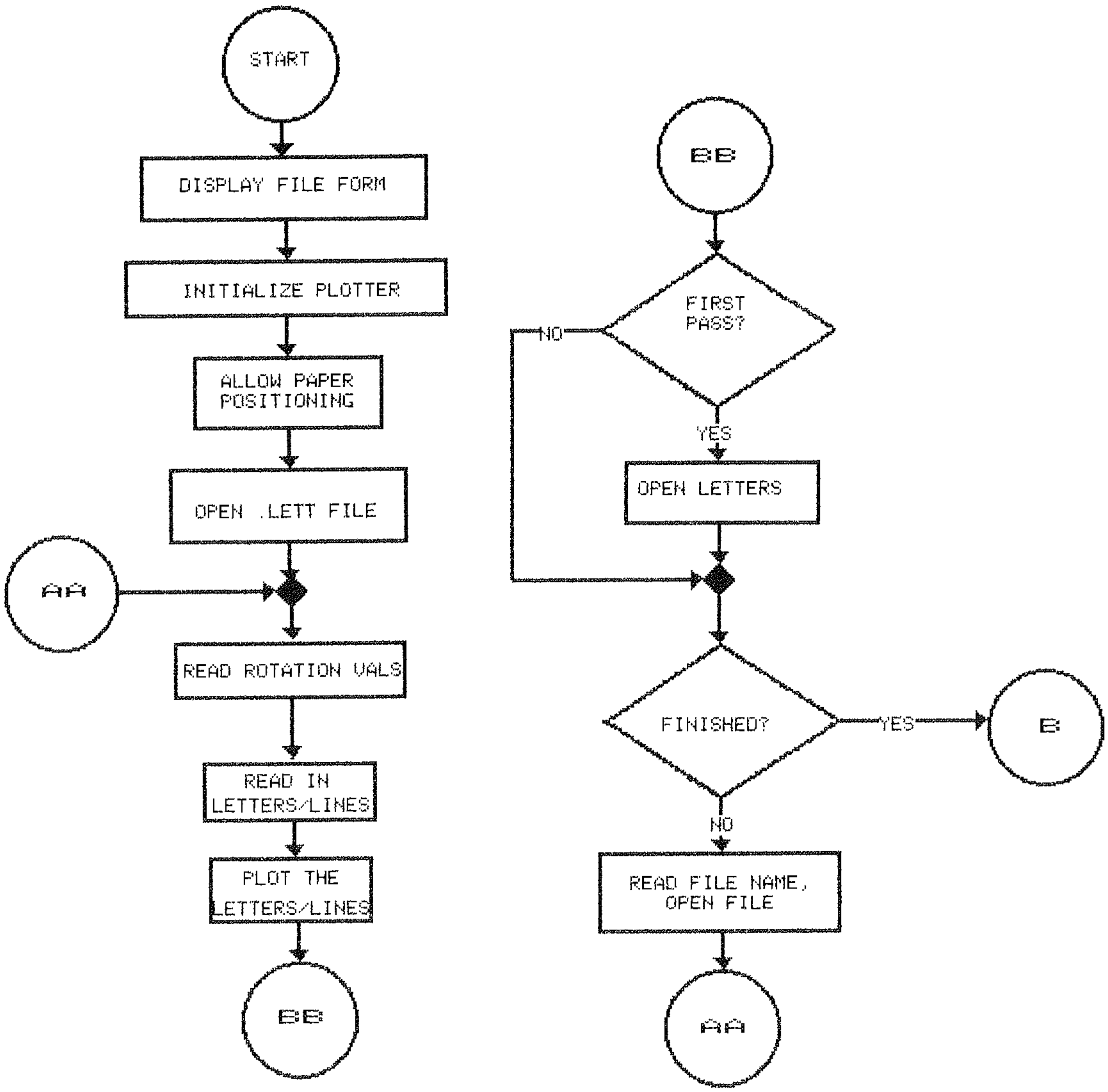


Figure C57. Flowchart, PLOT_LETTERS

```
=▶PLOT_LETTERS"run"
 PEN IS POSITIONED AT LOWER LEFT CORNER
 USE JOYSTICK TO ALIGN PEN ON GRID
 ENTER RETURN WHEN FINISHED
```

PLOT_LETTERS-1

```
 TEXT STRING
 INPUT FILE NAME (-1 TO STOP)     newlett
          ENTER INPUT FILE NAME AND CONFIRM
```

PLOT_LETTERS-2

Figure C58. Sample run, PLOT_LETTERS (Continued)

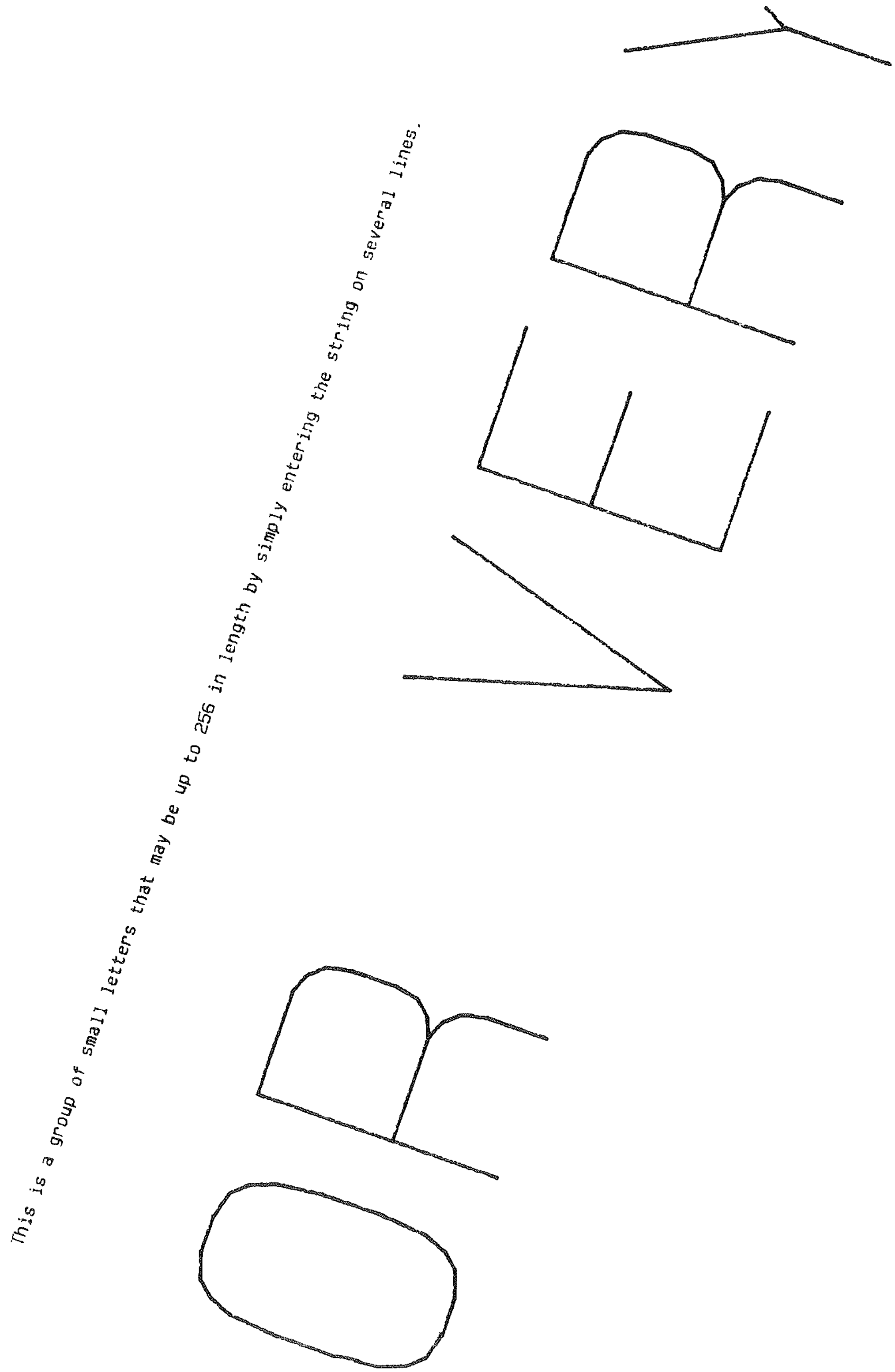


Figure C58. (Concluded)

PLOT_LETTERS_DESCRIPTION

Purpose

PLOT_LETTERS_DESCRIPTION provides a friendly environment for entering information necessary for creating plots of manually entered line or text data. Its output file serves as input to PLOT_LETTERS and to the PLAN_VIEW program. Therefore blocks of lines and letters, such as legends or indices may be created, modified, and plotted by the computer.

Input files

NAME.LETT

Output files

NAME.LETT

External devices

NONE

Modules called

PROBLEM, INITMENU, DRAWMENU, MESSAGESTACK, DRAWFORM, GET_NAME_OF_FILE, INSERT_IN_SITE_BUFFER, ONE_PORTION_SITE_BUFFER, EXTRACT_FROM_SITE_BUFFER, TEXT_MESSAGE, CHANNEL_MESSAGE

Theory of operation

PLOT_LETTERS_DESCRIPTION reads in current line/letter parameters, fills forms with these values, and displays the forms. If a nonexistant file name is entered, a new .LETT file is created. The contents of this new file are defined by filing in forms. To modify the file, the program is run again with the file name entered. It then accepts changes and outputs a .LETT file. A .LETT file would normally define a text/line graphical block. The block is rotated around a point corresponding to 0,0 in. of block units at some x,y position on the paper. The former (0,0) is assumed; the latter (x,y) is entered. Only one line of text may be entered at a time (up to 256 characters long). As each line is confirmed, the position of the default block for the next line is calculated as if a carriage return, line feed had been entered. This simplifies entering multiple lines of text. The intrablock letter rotation angle would normally be 0, 90, 180, or 270 deg (although any angle is acceptable); the block rotation angle might be any angle and sets the overall block rotation. The flowchart is shown in Figure C59, and a sample run is shown in Figure C60.

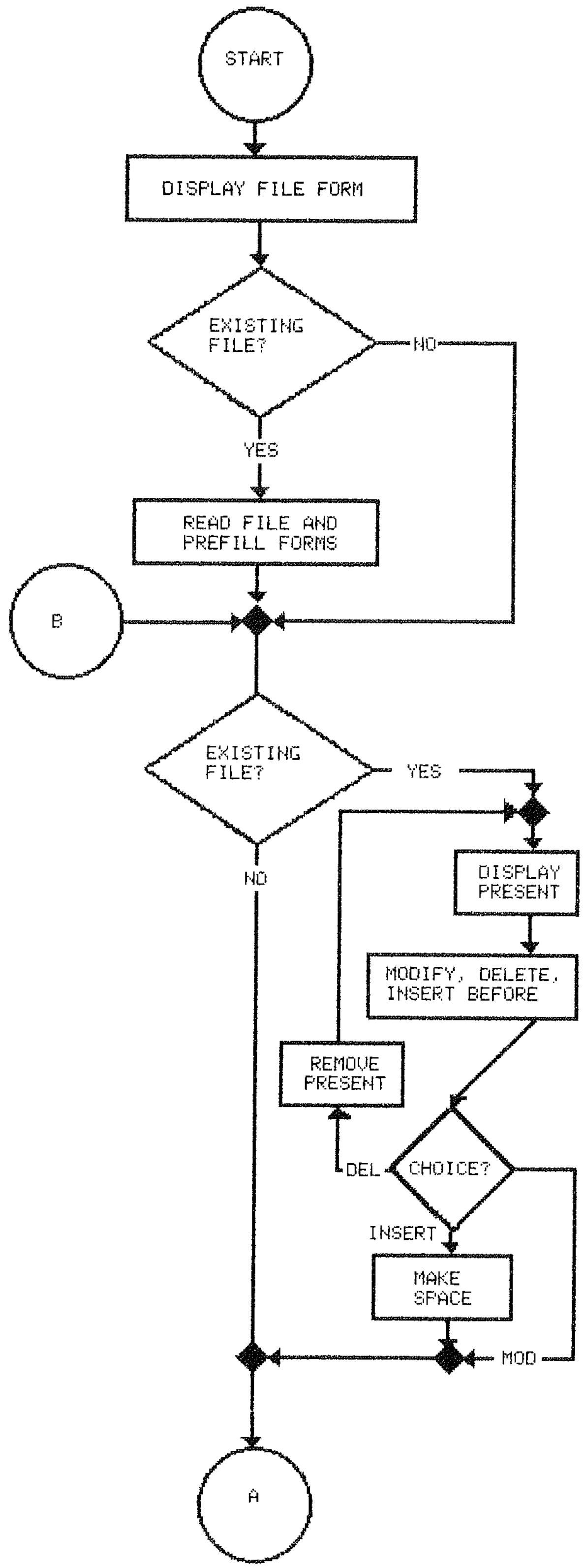


Figure C59. Flowchart, PLOT_LETTERS_DESCRIPTION (Continued)

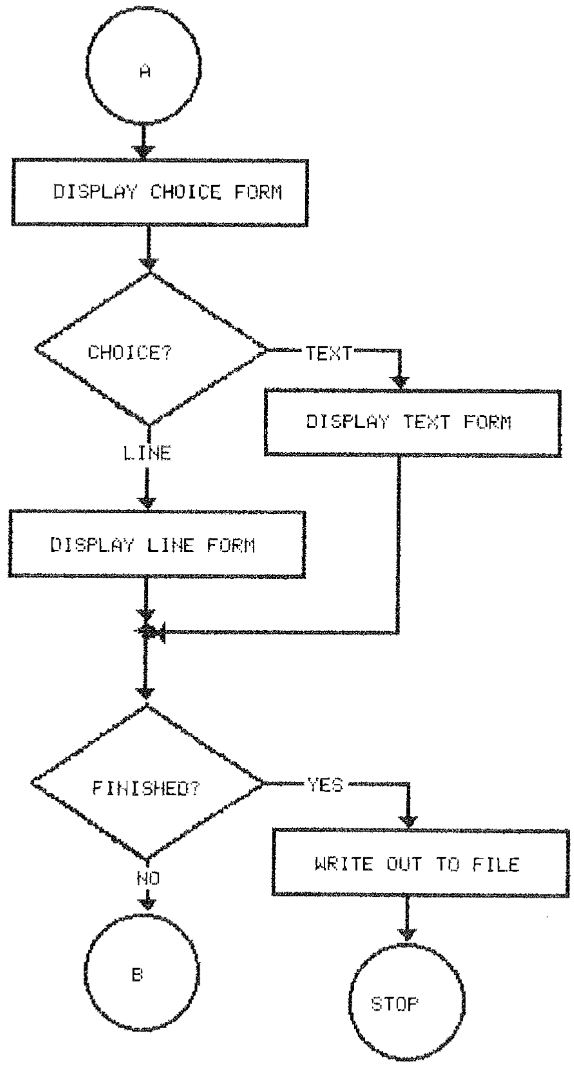


Figure C59. (Concluded)

```
TEXT STRING
INPUT FILE NAME (-1 TO STOP)    NEWLETT
        ENTER INPUT FILE NAME AND CONFIRM
```

PLOT_LETTERS_DESCRIPTION-1

```
****************************************
CODE RETURN CONFIRMS A FORM
ESCAPE (NOT CODE ESCAPE) ALLOWS ESCAPING
****************************************
```

```
LINE TEXT
NEXT OPTION  LINE
             SELECT NEXT PLOT OPTION
```

PLOT_LETTERS_DESCRIPTION-2

```
****************************************
CODE RETURN CONFIRMS A FORM
ESCAPE (NOT CODE ESCAPE) ALLOWS ESCAPING
****************************************
```

```
A REAL NUMBER
ROTATION ANGLE         20
X-ORIGIN OF ROTATION   5
Y-ORIGIN OF ROTATION   5
CARRIAGE SLOT FOR PEN  2
X-START OF LINE        -4
Y-START OF LINE        -2
X-FINISH OF LINE       4
Y-FINISH OF LINE       -2
MORE PLOTS AFTER THIS  YES
        ENTER ALL ITEMS AND CONFIRM  1
```

PLOT_LETTERS_DESCRIPTION-3

Figure C60. Sample run, PLOT_LETTERS_DESCRIPTION
(Sheet 1 of 5)

```
****************************************
CODE RETURN CONFIRMS A FORM
ESCAPE (NOT CODE ESCAPE) ALLOWS ESCAPING
****************************************
```

```
MODIFY
INSERT NEW BEFORE
DELETE
SELECT NEXT PLOT OPTION
```

PLOT_LETTERS_DESCRIPTION-4

```
****************************************
CODE RETURN CONFIRMS A FORM
ESCAPE (NOT CODE ESCAPE) ALLOWS ESCAPING
****************************************
```

```
LINE TEXT
NEXT OPTION  LINE
SELECT NEXT PLOT OPTION
```

PLOT_LETTERS_DESCRIPTION-5

```
****************************************
CODE RETURN CONFIRMS A FORM
ESCAPE (NOT CODE ESCAPE) ALLOWS ESCAPING
****************************************
```

```
A REAL NUMBER
ROTATION ANGLE         20.0000
X-ORIGIN OF ROTATION   5.0000
Y-ORIGIN OF ROTATION   5.0000
CARRIAGE SLOT FOR PEN  2
X-START OF LINE        -4
Y-START OF LINE        2.0000
X-FINISH OF LINE       4.0000
Y-FINISH OF LINE       2.0000
MORE PLOTS AFTER THIS  YES
ENTER ALL ITEMS AND CONFIRM  2
```

PLOT_LETTERS_DESCRIPTION-6

Figure C60. (Sheet 2 of 5)

```
****************************************
CODE RETURN CONFIRMS A FORM
ESCAPE (NOT CODE ESCAPE) ALLOWS ESCAPING
****************************************
```

```
MODIFY
INSERT NEW BEFORE
DELETE
SELECT NEXT PLOT OPTION
```

PLOT_LETTERS_DESCRIPTION-7

```
****************************************
CODE RETURN CONFIRMS A FORM
ESCAPE (NOT CODE ESCAPE) ALLOWS ESCAPING
****************************************
```

```
LINE TEXT
NEXT OPTION  TEXT
SELECT NEXT PLOT OPTION
```

PLOT_LETTERS_DESCRIPTION-8

```
****************************************
CODE RETURN CONFIRMS A FORM
ESCAPE (NOT CODE ESCAPE) ALLOWS ESCAPING
****************************************
A TEXT STRING
ROTAT. ANGLE   20
X-ROT POINT    5.0000
Y-ROT POINT    3
PEN SLOT       3
LETTER SIZE    0.07
LETTER ANGLE   0
X-POS LETTERS  -3
Y-POS LETTERS  1.5
TEXT STRING    This is a group of small letters tha
2              t may be up to 256 in length by simp
3              ly entering the string on several li
4              nes.
5
6
7
MORE PLOTS?    YES
ENTER ALL ITEMS AND CONFIRM  3
```

PLOT_LETTERS_DESCRIPTION-9

Figure C60. (Sheet 3 of 5)

```
****************************************
CODE RETURN CONFIRMS A FORM
ESCAPE (NOT CODE ESCAPE) ALLOWS ESCAPING
****************************************
```

```
MODIFY
INSERT NEW BEFORE
DELETE
SELECT NEXT PLOT OPTION
```

PLOT_LETTERS_DESCRIPTION-10

```
****************************************
CODE RETURN CONFIRMS A FORM
ESCAPE (NOT CODE ESCAPE) ALLOWS ESCAPING
****************************************
```

```
LINE TEXT
NEXT OPTION  TEXT
SELECT NEXT PLOT OPTION
```

PLOT_LETTERS_DESCRIPTION-11

```
****************************************
CODE RETURN CONFIRMS A FORM
ESCAPE (NOT CODE ESCAPE) ALLOWS ESCAPING
****************************************
A TEXT STRING
ROTAT. ANGLE   20.0000
X-ROT POINT    5.0000
Y-ROT POINT    3.0000
PEN SLOT       4
LETTER SIZE    1.5
LETTER ANGLE   0.0000
X-POS LETTERS  -3.0000
Y-POS LETTERS  -1
TEXT STRING    OR VERY LARGE ON 1 LINE
2
3
4
5
6
7
MORE PLOTS?    YES
ENTER ALL ITEMS AND CONFIRM  4
```

PLOT_LETTERS_DESCRIPTION-12

Figure C60. (Sheet 4 of 5)

```
****************************************
CODE RETURN CONFIRMS A FORM
ESCAPE (NOT CODE ESCAPE) ALLOWS ESCAPING
****************************************
```

```
                 MODIFY
                 INSERT NEW BEFORE
                 DELETE
            SELECT NEXT PLOT OPTION
```

PLOT_LETTERS_DESCRIPTION-13

```
****************************************
CODE RETURN CONFIRMS A FORM
ESCAPE (NOT CODE ESCAPE) ALLOWS ESCAPING
****************************************
                 MODIFY
                 INSERT NEW BEFORE
                 DELETE
            SELECT NEXT PLOT OPTION
ROTATION ANGLE          20.000
X-ROT POINT              5.000
Y-ROT POINT              3.000
PEN SLOT                 4.000
LETTER SIZE              1.500
LETTER ANGLE             0.000
X-POS LETTER            -3.000
Y-POS LETTER            -2.950
 TEXT STRING
 2
 3
 4
 5
 6
 7
```

PLOT_LETTERS_DESCRIPTION-14

```
****************************************
CODE RETURN CONFIRMS A FORM
ESCAPE (NOT CODE ESCAPE) ALLOWS ESCAPING
****************************************
```

```
 A REAL NUMBER
ROTATION ANGLE          20.0000
X-ORIGIN OF ROTATION    5.0000
Y-ORIGIN OF ROTATION    5.0000
CARRIAGE SLOT FOR PEN   2
X-START OF LINE         -4.0000
Y-START OF LINE         2.0000
X-FINISH OF LINE        -4.0000
Y-FINISH OF LINE        -2
MORE PLOTS AFTER THIS   NO
        ENTER ALL ITEMS AND CONFIRM  5
```

PLOT_LETTERS_DESCRIPTION-15

Figure C60. (Sheet 5 of 5)

PLOT_PLOTTER

Purpose

PLOT_PLOTTER is a task spawned by the SURVEY program. Its purpose is to display the boat's current position and current depths in plan view on the plotter.

Input files

NONE (parameters from .PLAN are passed through system common)

Output files

NONE

External devices

Plotter

Modules called

CASCII, POLREC, A2, PROBLEM, PLOTROT, TRANSPOSE_ORIGIN, SYMBOLROT

Theory of operation

PLOT_PLOTTER begins by waiting for the operator to align the paper if manual alignment was selected in the .PLAN file. It then does the rotation and scaling selected in the .PLAN file. It plots the channel lines, center line, base line, section lines, station numbers, and north arrow. It then picks up the proper pen for depth plotting and waits for PLOT_SCREEN to set the continue flag. At this point the pen follows the current boat position on the plot. If data are being saved, depths are printed (logic assures depths are not overprinted even though many more depths are saved than plotted).

The flowchart is shown in Figure C61.

Special considerations

A .PLAN file should be created in the office prior to going to the site. Otherwise, PLAN_VIEW_DESCRIPTOR must be run at the site.

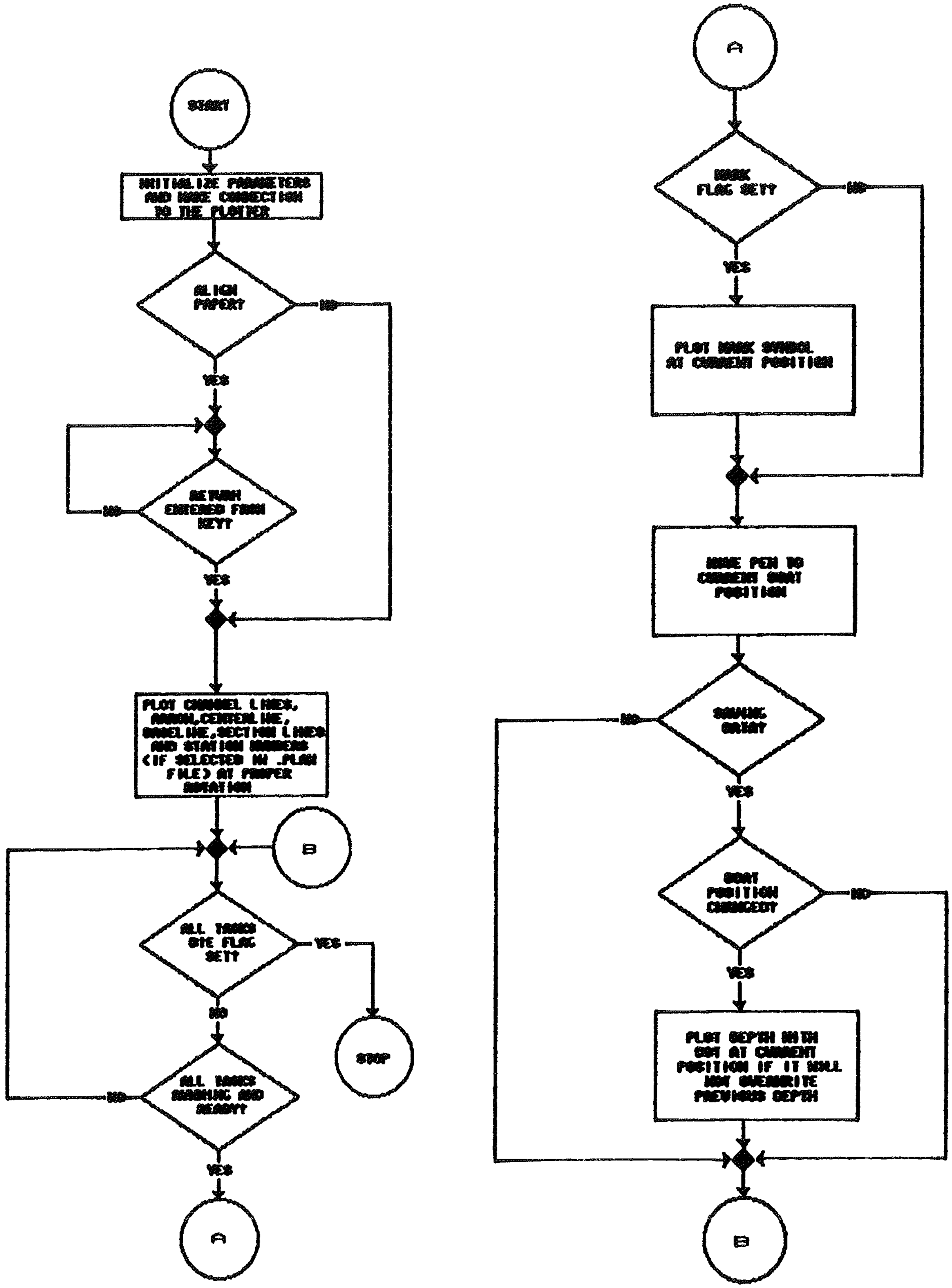


Figure C61. Flowchart, PLOT_PLOTTER

PLOT_PLOTTER_LOUISVILLE

Purpose

PLOT_PLOTTER_LOUISVILLE is a task spawned by the SURVEY program. Its purpose is to display the boat's current positions and current depths in plan view on the plotter.

Input files

NONE (plot parameters are passed through system common)

Output files

BAR_ID.EQUI (where BAR_ID is the bar number)

External devices

Plotter

Modules called

CASCII, POLREC, A2, PROBLEM

Theory of operation

PLOT_PLOTTER_LOUISVILLE begins by plotting legend information about buoy symbol/colors, transponder symbol/colors, pool, bar, etc. It then waits for the operator to position the pen on the paper and select transponder B's location. Transponder B's symbol is then plotted at that point. Separation between transponder A and B was entered in a form displayed by CONFIGURE, but azimuth is presently unknown. Therefore, the operator then positions the pen in the relative azimuth of transponder A and enters a return. The task adjusts the position by the scaled separation and plots the symbol for transponder A. It then recomputes transponder positions and writes these out to the BAR_NO.EQUI file. Then it waits until PLOT_SCREEN_LOUISVILLE sets the continue flag. At this point it loops moving the pen to follow the boat position. If a mark is entered, the appropriate mark symbol is plotted at the current boat position. If saving data is enabled, depths are plotted. Logic assures depths are not overprinted even though many more depths are collected than are plotted. The flowchart is shown in Figure C62.

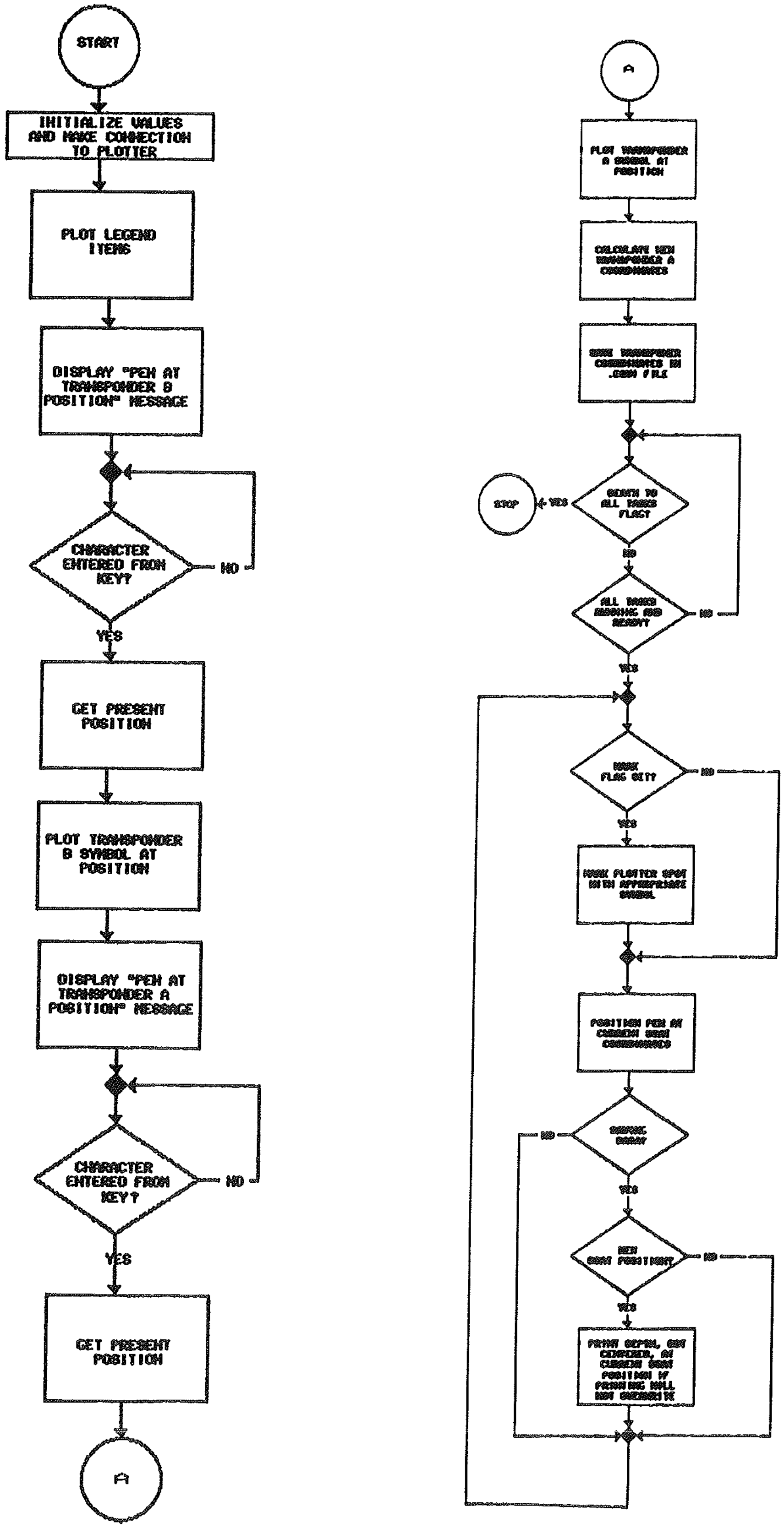


Figure C62. Flowchart, PLOT_PLOTTER_LOUISVILLE

PLOT_SCREEN

Purpose

PLOT_SCREEN is a task spawned by the SURVEY program. It provides the screen and keyboard interface during surveying.

Input files

NONE (parameters are passed through system common)

Output files

NAME.0001 (where .0001 is the line number)

NAME.M001 (where 001 is the mark number)

NAME.D001 (where 001 is the tide number)

External devices

NONE

Modules called

ASCI, UPDATESCREEN, POLREC, A2, PROBLEM, INISOUND, WRSOUND, TERMSOUND

Theory of operation

PLOT_SCREEN begins by mapping to common memory area and initializing to the first line. It then creates the initial screen display of cross and/or plan view and current reading. Next it loops, testing to see if a key has been pressed and updating the screen. If a key has been pressed, it carries out the appropriate action. At the end of a line it updates the screen to the next line, then it writes the data to disk. The flowchart is shown in Figure C63.

Special considerations

Marks or tides entered while saving data to memory are not written to disk until the end of line; so it is faster to collect marks separately from running lines although either way works.

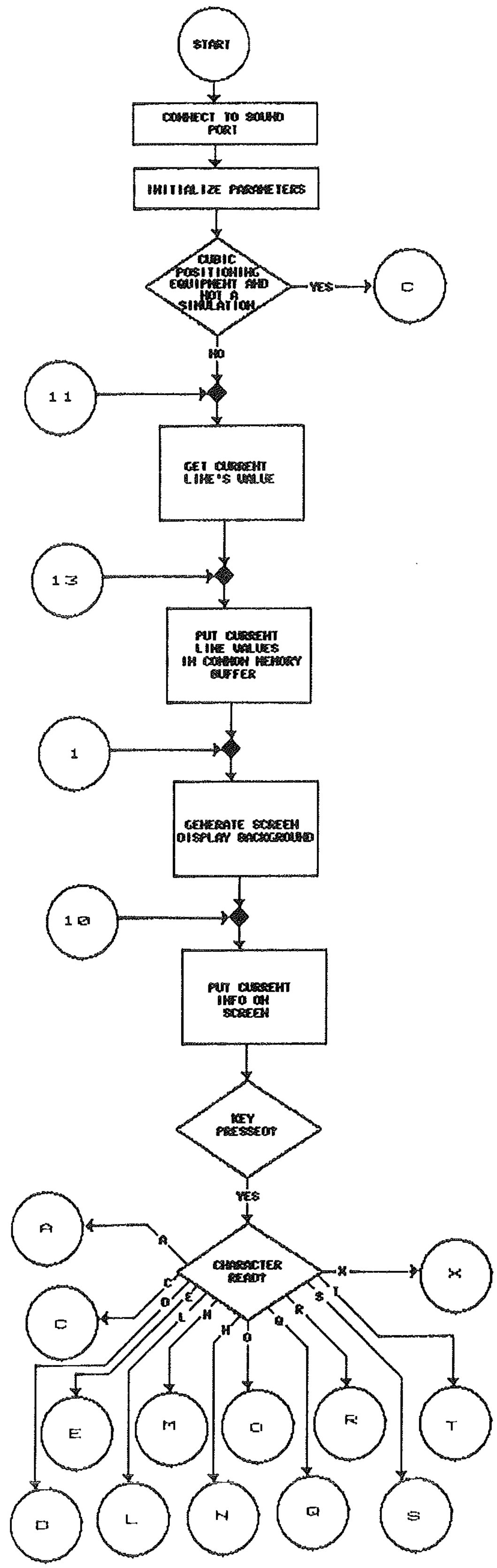


Figure C63. Flowchart, PLOT_SCREEN
(Sheet 1 of 5)

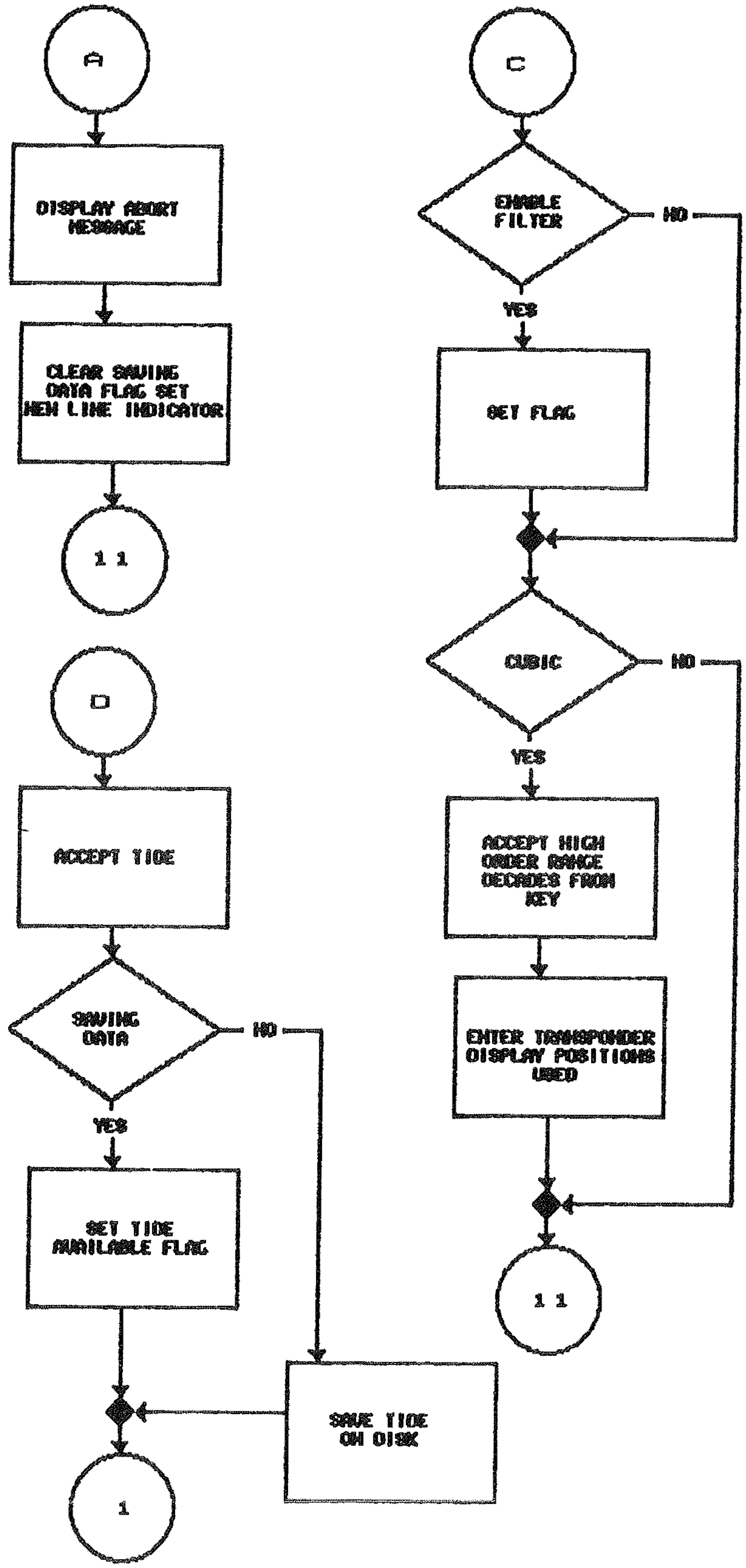


Figure C63. (Sheet 2 of 5)

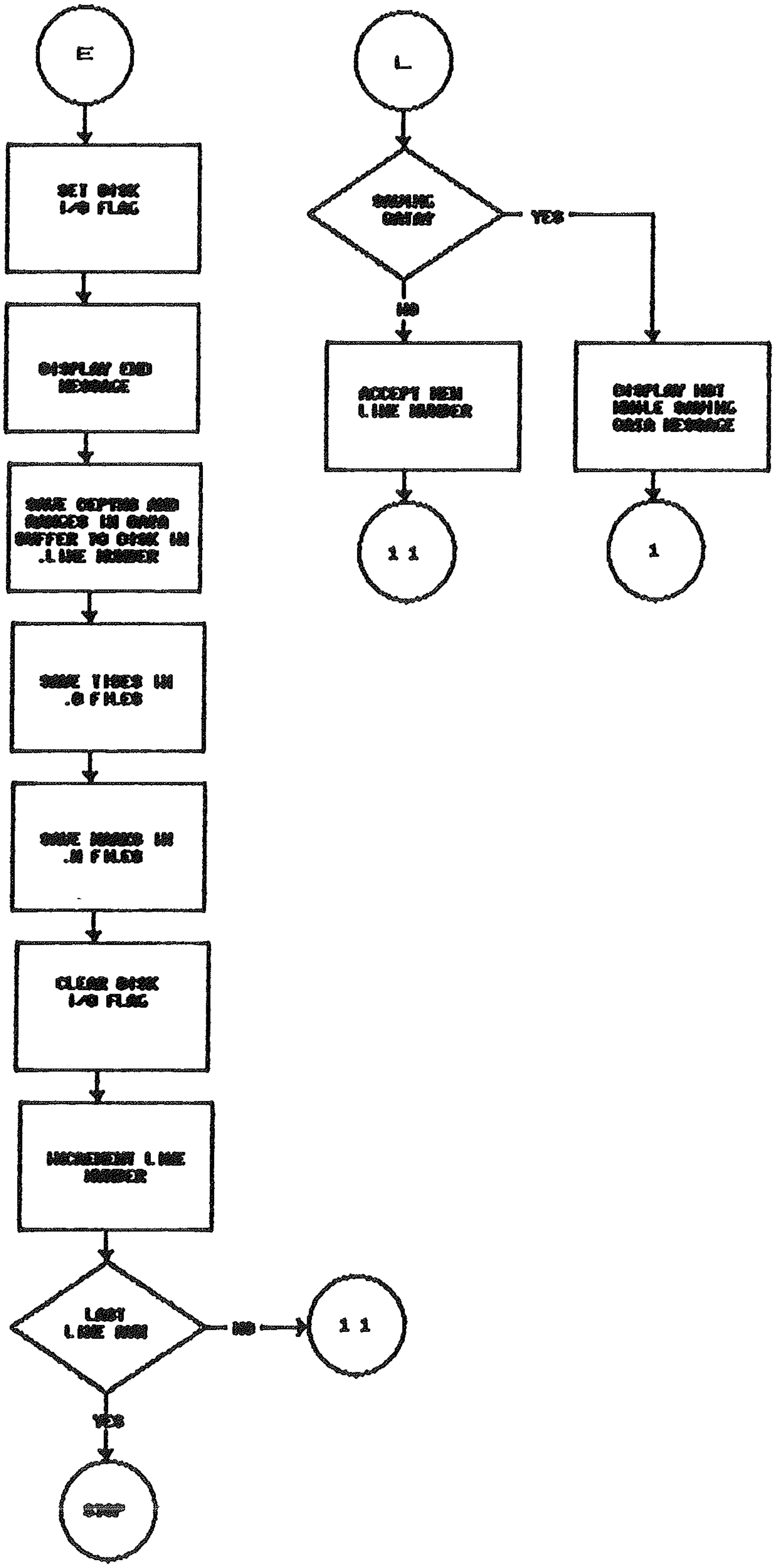


Figure C63. (Sheet 3 of 5)

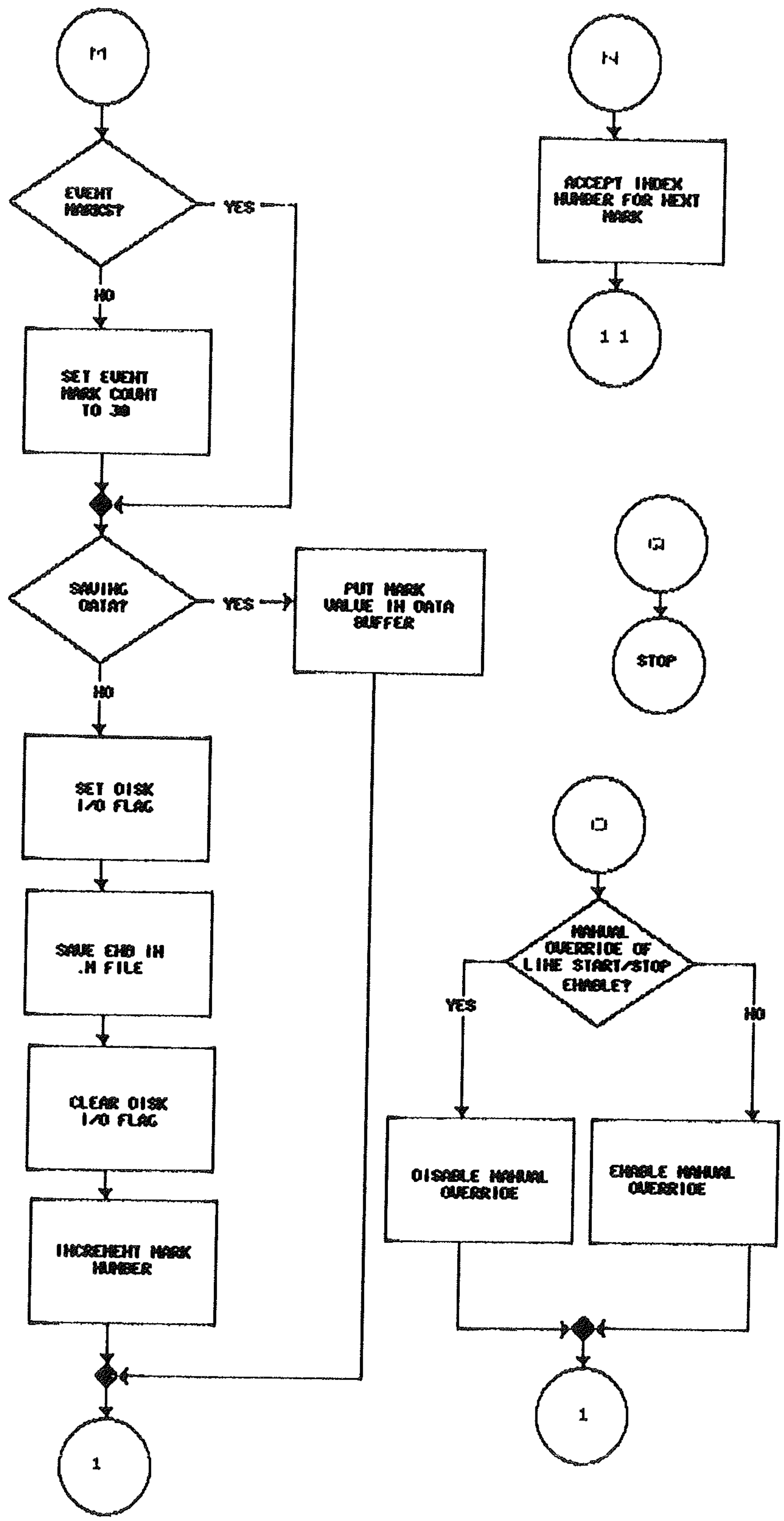


Figure C63. (Sheet 4 of 5)

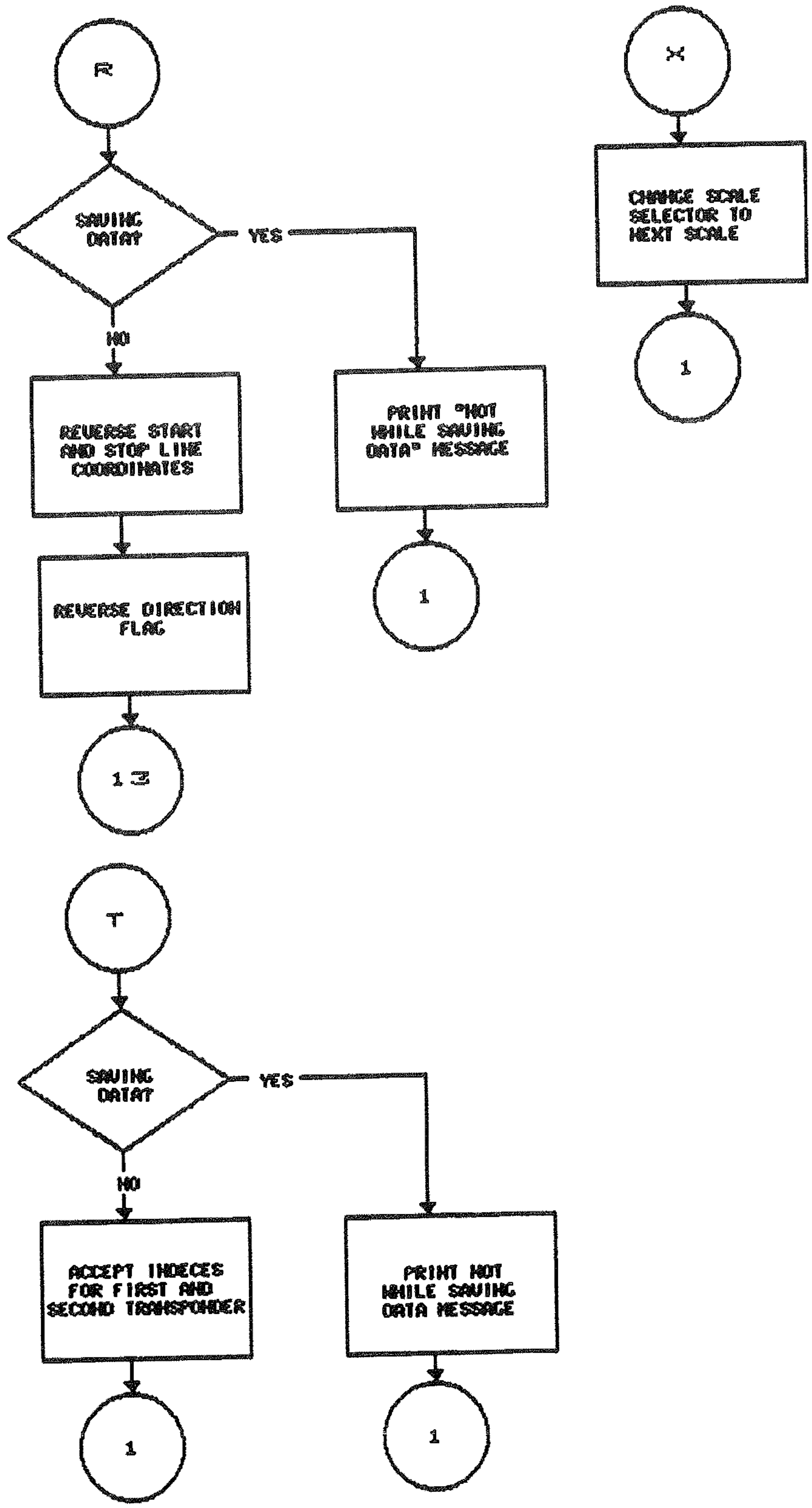


Figure C63. (Sheet 5 of 5)

PLOT_SCREENIMAGE

Purpose

PLOT_SCREENIMAGE reads a list of screen images and plot parameters from SCREENIMAGE and plots the associated images. It is a follow up to PLAN_PLOT and is part of the CAM package.

Input files

SCREENIMAGE, POS2, NAME~SCREENIMAGE~

Output files

NONE

External devices

Plotter

Modules called

PROBLEM, ON, LINEPLOT, SET, PLOTROT, TRANSPOSE_ORIGIN

Theory of operation

PLOT_SCREENIMAGE gets file names from SCREENIMAGE. It takes these raster scan files and converts them to vector plotter commands. Each dot is produced by multiple passes (depending on the plot scale) of short lines. The flowchart is shown in Figure C64.

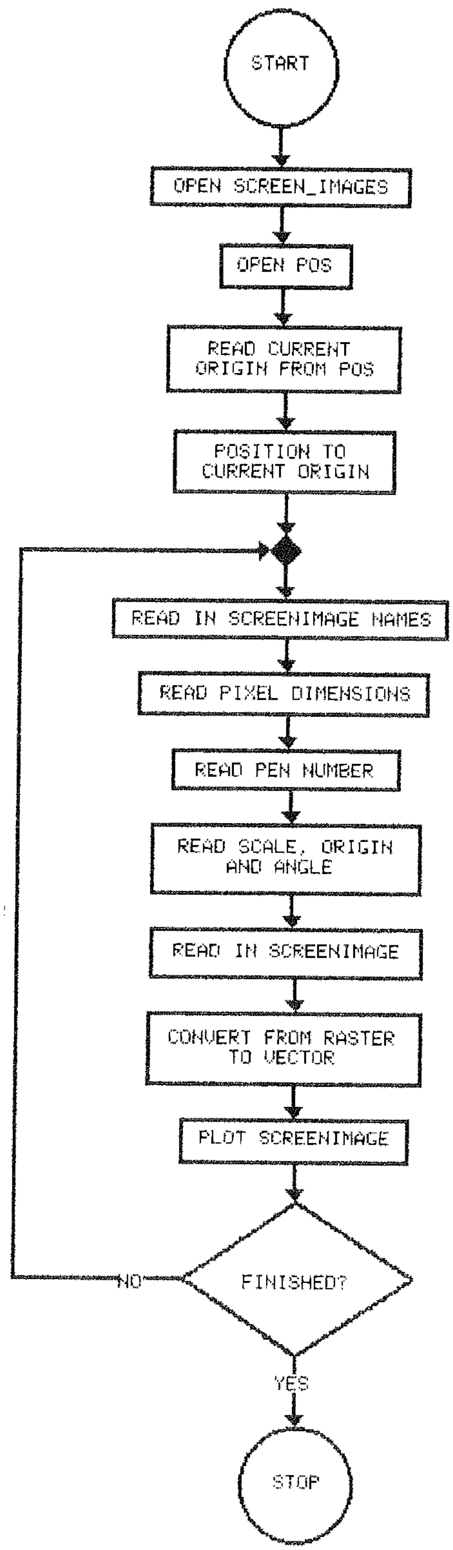


Figure C64. Flowchart, PLOT_SCREENIMAGE

PLOT_SCREEN_LOUISVILLE

Purpose

PLOT_SCREEN_LOUISVILLE is a task spawned by SURVEY that handles the operator interface for the Louisville-style survey.

Input files

NONE (parameters passed through system common)

Output files

NAME.0001 (where .0001 is the line number)

NAME.M001 (where 001 is the mark number)

NAME.D001 (where 001 is the tide number)

External devices

NONE

Modules called

ASCI, POLREC, A2, PROBLEM, INISOUND, WRSOUND, TERMSD, UPDATESCREEN

Theory of operation

PLOT_SCREEN_LOUISVILLE begins by initializing to the first line and displaying that line. It then sets the continue flag so that the other tasks can continue. Then it loops checking to see if a key has been pressed and reacting accordingly while updating the screen. If the line is finished or a mark has been entered, data are saved to disk. The flowchart is shown in Figure C65.

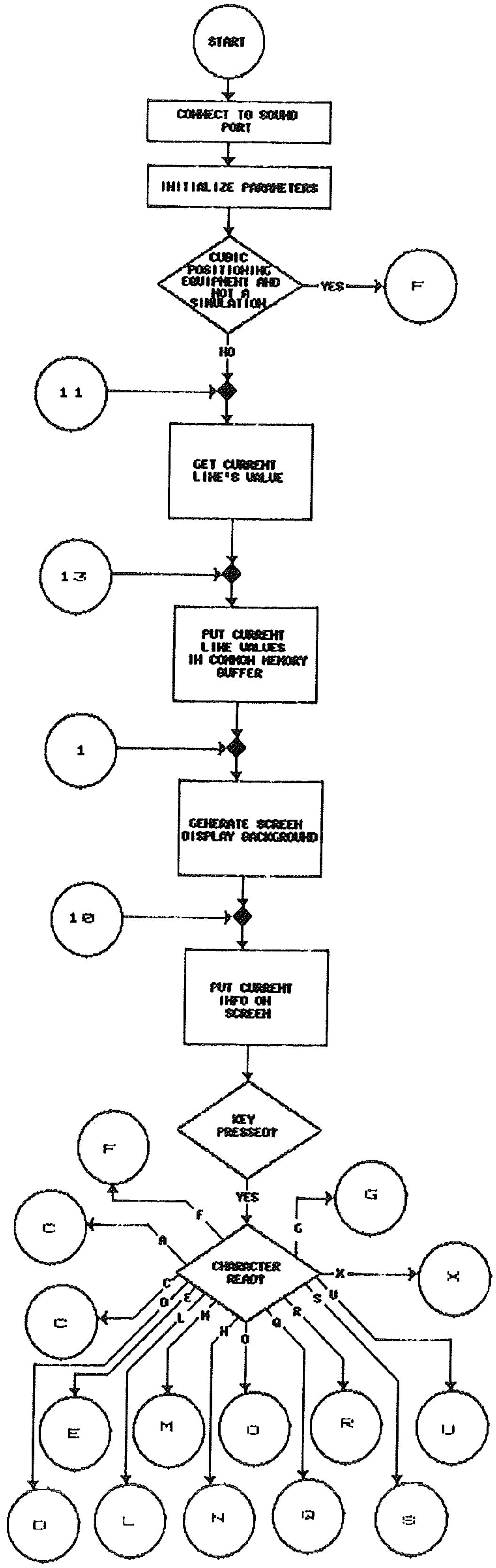


Figure C65. Flowchart, PLOT_SCREEN_LOUISVILLE (Sheet 1 of 5)

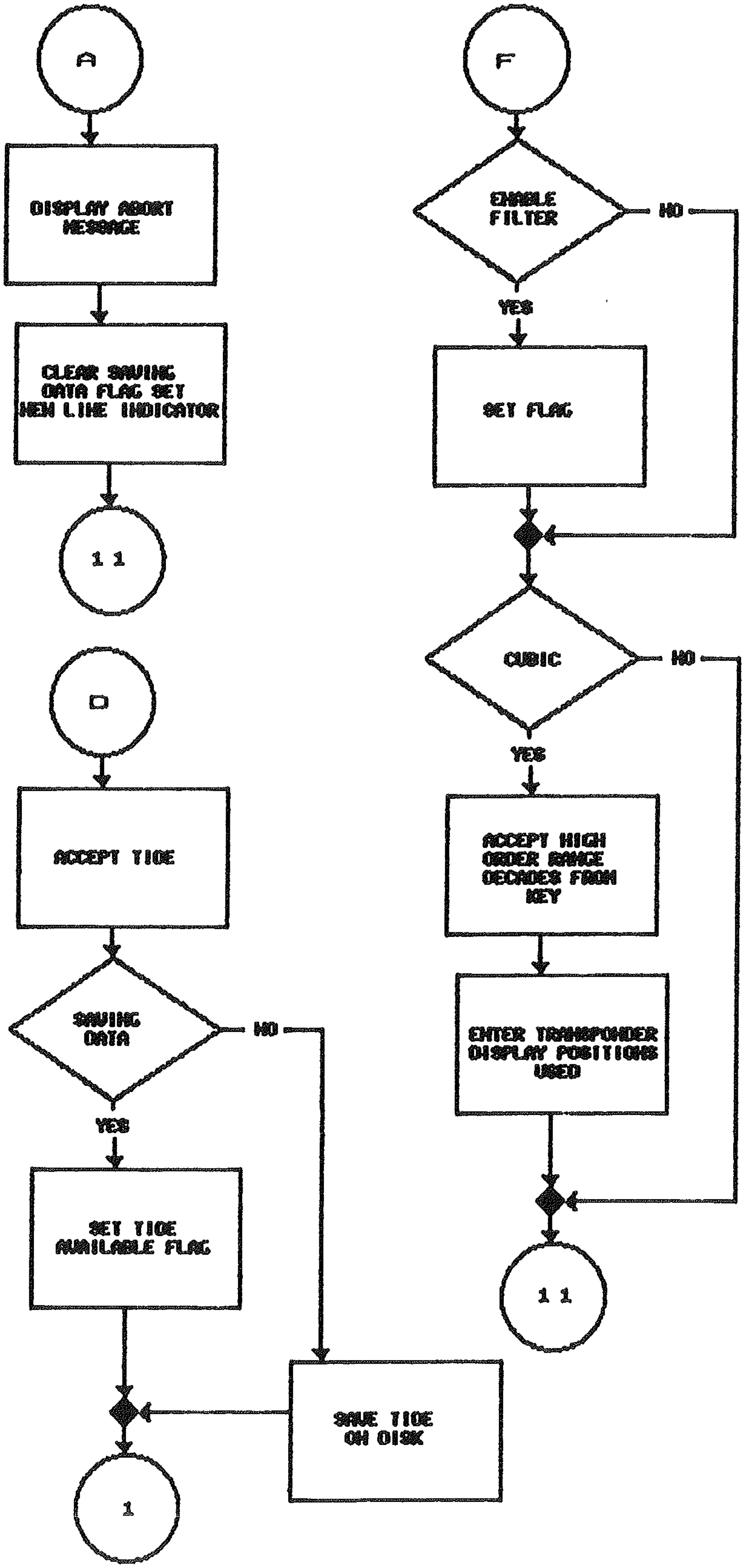


Figure C65. (Sheet 2 of 5)

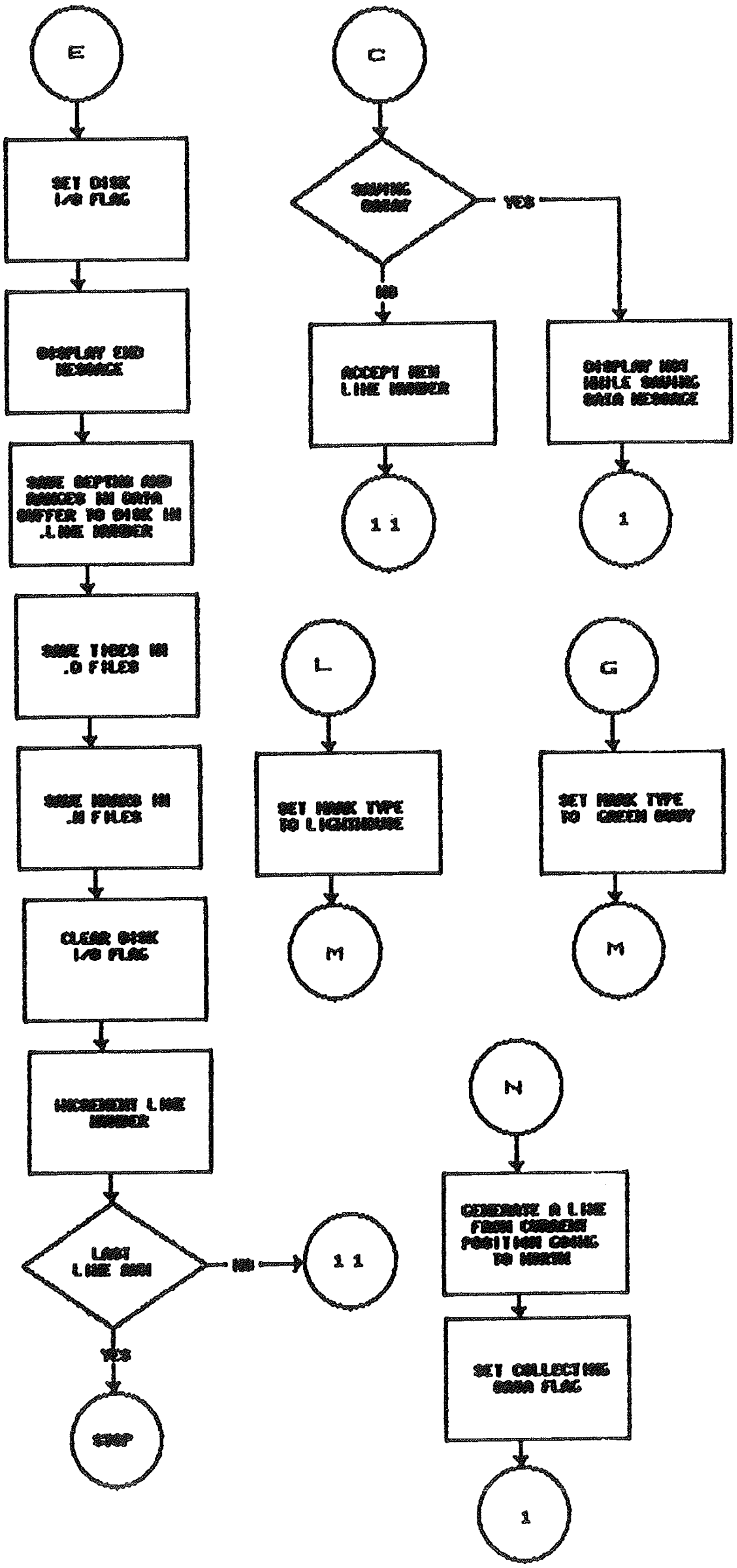


Figure C65. (Sheet 3 of 5)

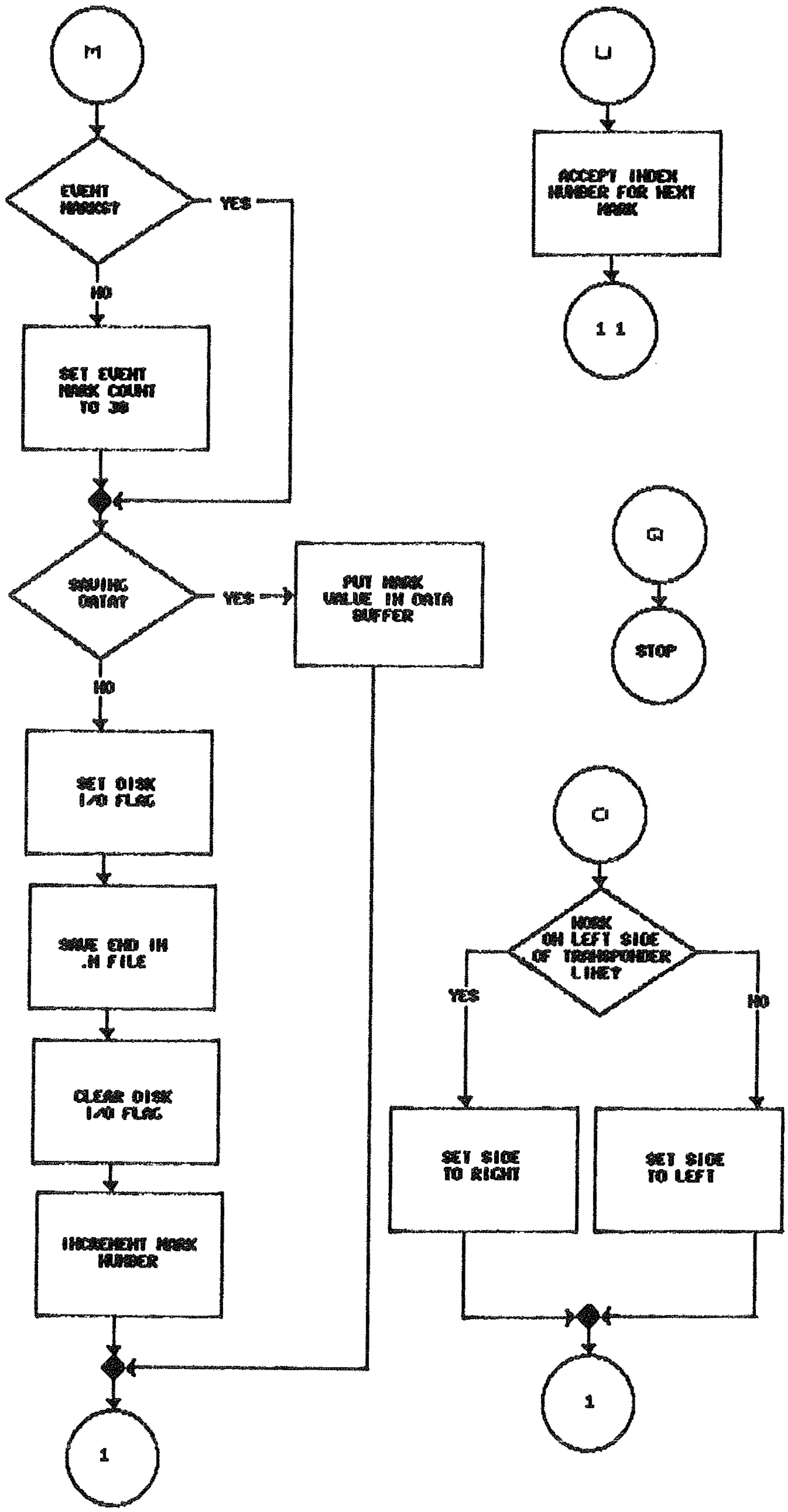


Figure C65. (Sheet 4 of 5)

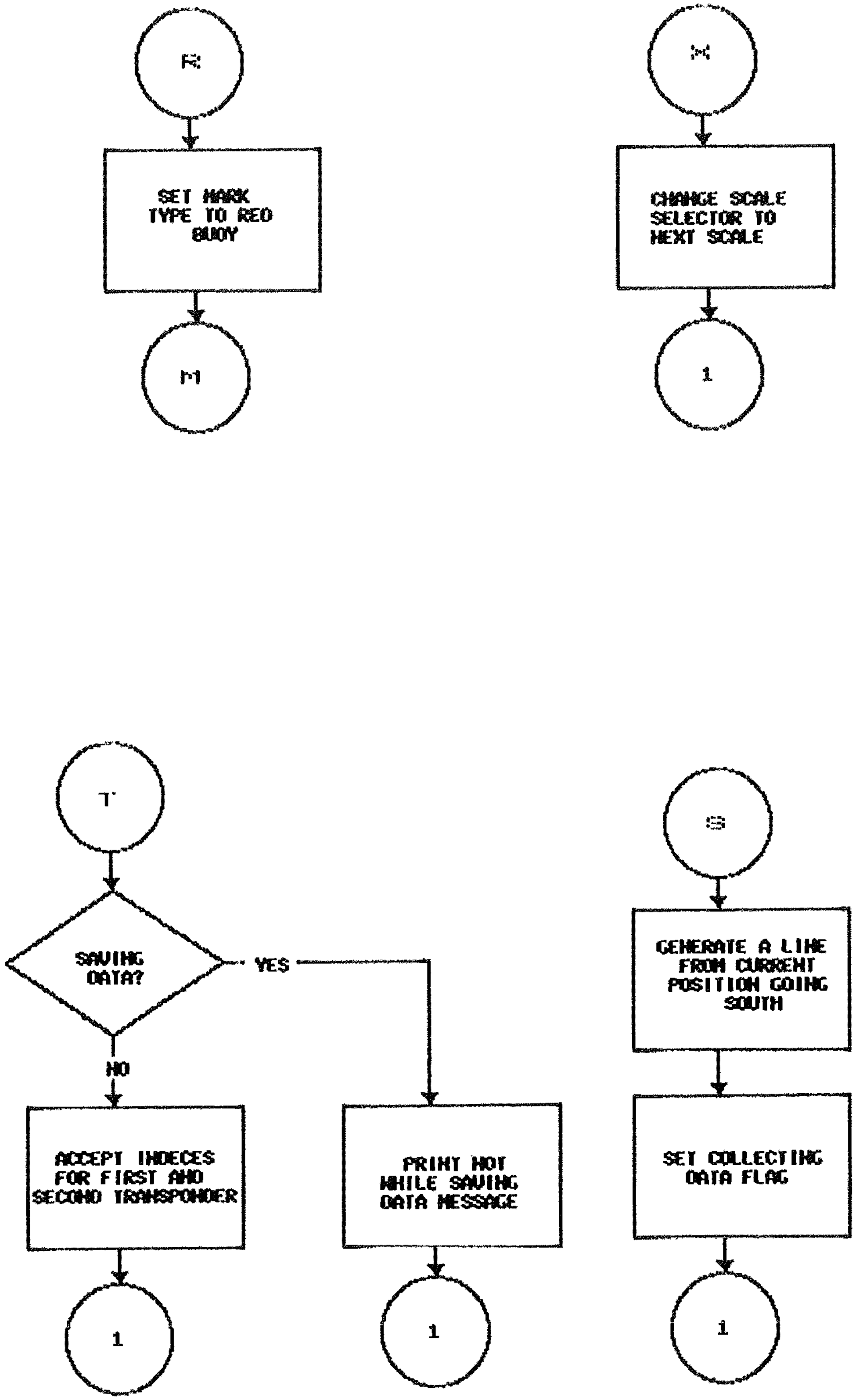


Figure C65. (Sheet 5 of 5)

PRINT

Purpose

PRINT is one of the tasks which may be spawned by the SURVEY program. It prints time, depth, easting, northing, distance along line, and offset at the time interval entered into CONFIGURE's form.

Input Files

NONE (parameters are passed through system common)

Output Files

NONE

External devices

NONE

Modules called

PROBLEM

Theory of operation

PRINT begins by initializing the printer then waits for the continue flag from PLOT_SCREEN. It then loops, printing the latest data acquired. If the line number has changed since the last pass, the line number is printed as a header. The flowchart is shown in Figure C66.

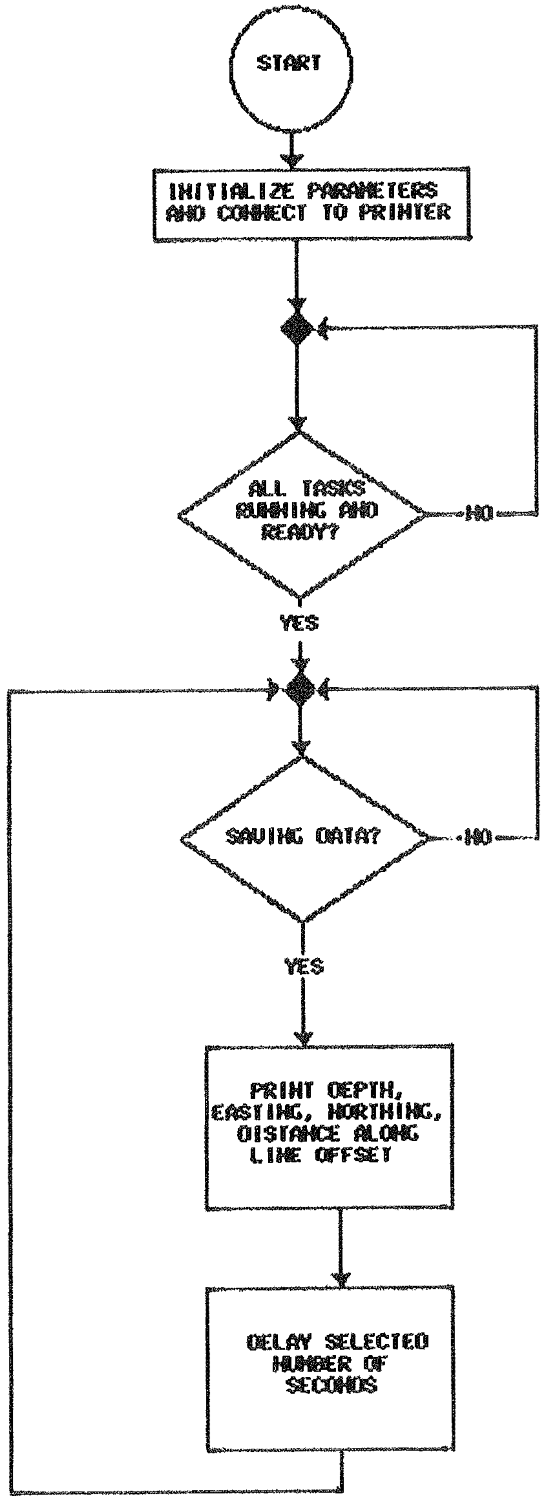


Figure C66. Flowchart, PRINT

PRINT_DATA

Purpose

PRINT_DATA prints the acquired survey data in raw form: (1) time versus depth and (2) time versus position. It is useful for resolving problems and is not normally used, as the graphical presentations are normally preferred.

Input files

NAME.0001 (where .0001 is the line number)

Output files

NONE

External devices

Printer

Modules called

PROBLEM, INITMENU, DRAWMENU, MESSAGESTACK, DRAWFORM, GET_NAME_OF_FILE

Theory of operation

PRINT_DATA displays a file form to get the site name. It then loops through the data lines collected printing the data in raw form: time versus reading. The flowchart is shown in Figure C67, and a sample run is shown in Figure C68.

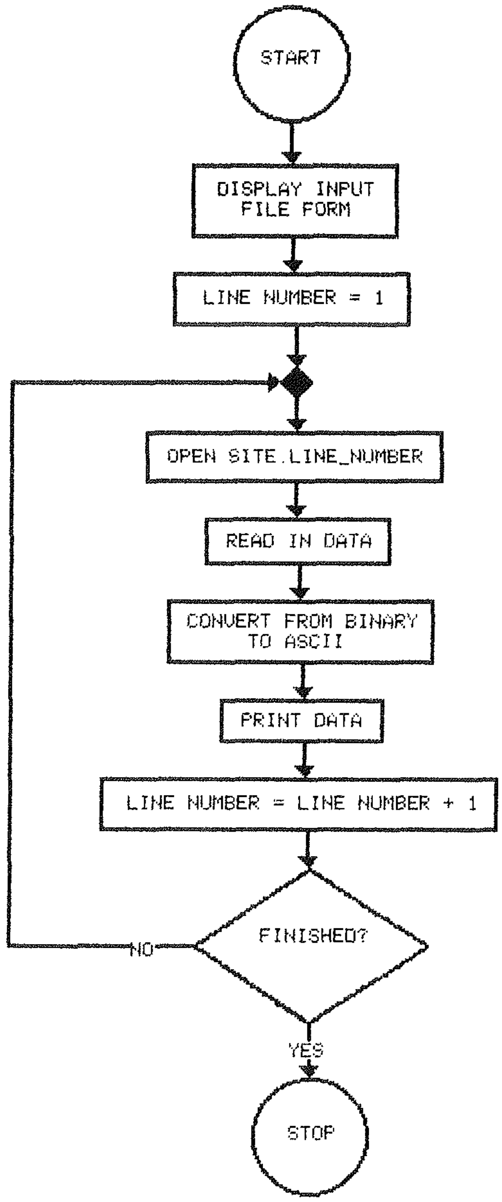


Figure C67. Flowchart, PRINT_DATA

```
TEXT STRING
FILENAME(-1 STOP,-2 SIMULATION) 041-1
ENTER INPUT FILE NAME AND CONFIRM
```

PRINT_DATA-1

```
LINE     1

 7-29-1986  10:56:51.2
RUN LENGTH        1099.94
DIRECTION FLAG  0  CORRECTION FLAG  0
DEPTH    1      82.70       56.40
DEPTH    2      83.70       56.80
DEPTH    3      84.70       56.50
DEPTH    4      85.70       56.80
DEPTH    5      86.70       56.10
DEPTH    6      87.70       56.40
DEPTH    7      88.70       56.80
DEPTH    8      89.70       57.10
DEPTH    9      90.70       56.70
DEPTH   10      91.70       56.40
RANGE    1      91.10   320011.19     46563.70      134.83       6.56
DEPTH   11      92.70       56.60
DEPTH   12      93.70       57.30
DEPTH   13      94.70       56.80
DEPTH   14      95.70       56.40
DEPTH   15      96.70       56.90
DEPTH   16      97.70       56.70
DEPTH   17      98.70       56.70
DEPTH   18      99.70       56.40
DEPTH   19     100.70       56.30
DEPTH   20     101.70       56.80
RANGE    2     101.20   320155.38     46513.58      287.41       1.88
DEPTH   21     102.70       56.80
DEPTH   22     103.70       56.90
DEPTH   23     105.00       56.80
DEPTH   24     106.10       56.90
DEPTH   25     107.10       56.90
DEPTH   26     108.10       57.20
DEPTH   27     109.10       56.60
DEPTH   28     110.10       56.90
DEPTH   29     111.10       56.90
RANGE    3     111.20   320298.94     46464.05      439.19      -3.12
DEPTH   30     112.10       57.20
```

PRINT_DATA-2

Figure C68. Sample run, PRINT_DATA

PRINT_MARKS

Purpose

To print the raw marks acquired by the SURVEY program.

Input files

NAME.M001 (where 001 is the mark number)

Output files

NONE

External devices

Printer

Modules called

PROBLEM, INITMENU, DRAWMENU, MESSAGESTACK, DRAWFORM, GET_NAME_OF_FILE

Theory of operation

PRINT_MARKS begins by displaying a file form for inputting the site file name. It then loops, printing the data in each successive file until no more files are detected. The flowchart is shown in Figure C69, and a sample run is shown in Figure C70.

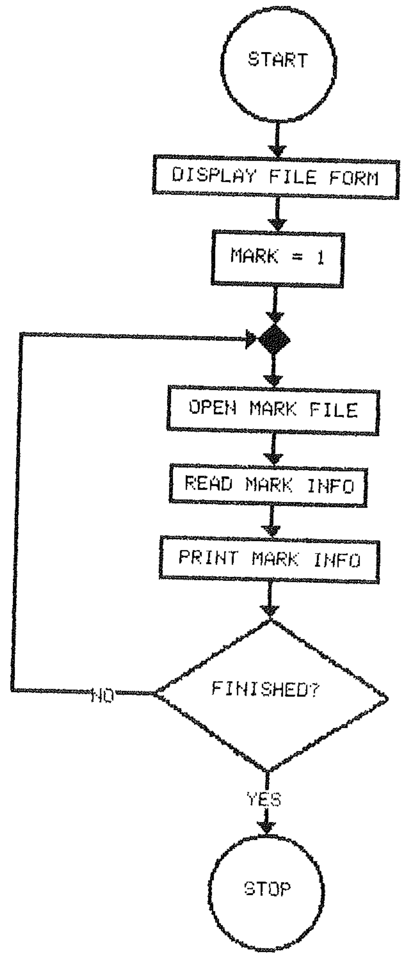


Figure C69. Flowchart,
PRINT_MARKS

```
 TEXT STRING
 FILENAME(-1 STOP,-2 SIMULATION) 041-1
         ENTER INPUT FILE NAME AND CONFIRM
```

```
                    PRINT_MARKS-1


MARK     1

 7-29-1986   9:47:28.5
T  159.8 E   326966.3     N    67686.7     D       43.3

MARK     2

 7-29-1986   9:47:28.5
T  229.8 E   327776.3     N    66991.0     D       31.3

MARK     3

 7-29-1986   9:47:28.5
T 1634.5 E   326833.1     N    63566.5     D       32.3

MARK     4

 7-29-1986   9:47:28.5
T 1743.9 E   325808.5     N    62418.7     D       38.7

MARK     5

 7-29-1986   9:47:28.5
T 2008.5 E   325122.4     N    60048.8     D       36.7

                  PRINT_MARKS-2
```

Figure C70. Sample run, PRINT_MARKS

RANDOM

Purpose

RANDOM is a task spawned by the SURVEY program. It provides the screen and keyboard interface during surveying.

Input files

NONE (parameters are passed through system common)

Output files

NAME.0001 (where .0001 is the line number)

NAME.M001 (where 001 is the mark number)

NAME.D001 (where 001 is the tide number)

External devices

NONE

Modules called

ASCI, UPDATESCREEN, POLREC, A2, PROBLEM, INISOUND, WRSOUND, TERMSOUND

Theory of operation

RANDOM begins by mapping to common memory area and initializing to the first line. It then asks if the present channel and section coordinate information is to be used or discarded. Next it displays a number of options including going into run or channel marking mode. If one of these two options is chosen, it creates the initial screen display of cross and/or plan view and current reading. Next it loops, testing to see if a key has been pressed and updating the screen. If a key has been pressed, it carries out the appropriate action. At the end of a line it updates the screen to the next line, then it writes the data to disk.

RANDOM survey has two major modes: (1) RUN and (2) CHANNEL. RUN mode is used to

a. Establish a reference line using *Y* to mark left side of channel start of line, and *Z* to mark right side of channel end of line

b. Run a generated section line (same as with section line)

c. Run randomly

CHANNEL mode is used to establish new channel coordinates or change old ones. A line will be displayed on the screen only and pilot guidance will be displayed when running a generated line. Channel mode and running randomly must be done by visual surveillance.

Key options for RUN mode are the same as with section line surveys except *Q* returns to the mode select menu rather than quitting and *Y* and *Z* have been added.

Key options for CHANNEL mode are:

L Marks left side of channel

R Marks right side of channel

C Marks center line of channel

B Marks base line of channel

N Saves present channel coordinate pairs and advances to next channel index

J Jumps back to previous channel index and redoes it

Q Returns to mode select menu

E Ends index

When in RUN mode the current section line number is shown in the upper right corner; when in CHANNEL the current channel index number is shown in the upper right corner.

When finished, *Q* must be entered from the mode select menu to quit the program. If the program is not terminated in this manner, the section line file (.LINE) will not be updated.

The flowchart is shown in Figure C71, and a typical run is shown in Figure C72.

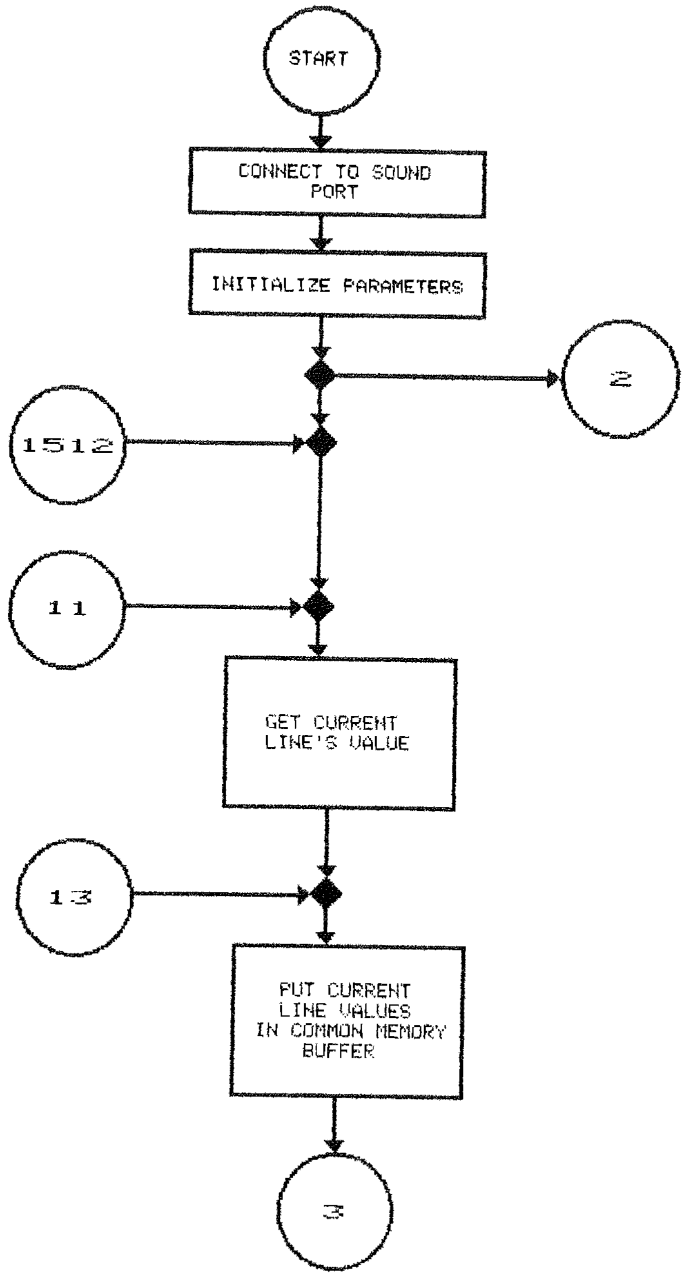


Figure C71. Flowchart, RANDOM (Sheet 1 of 8)

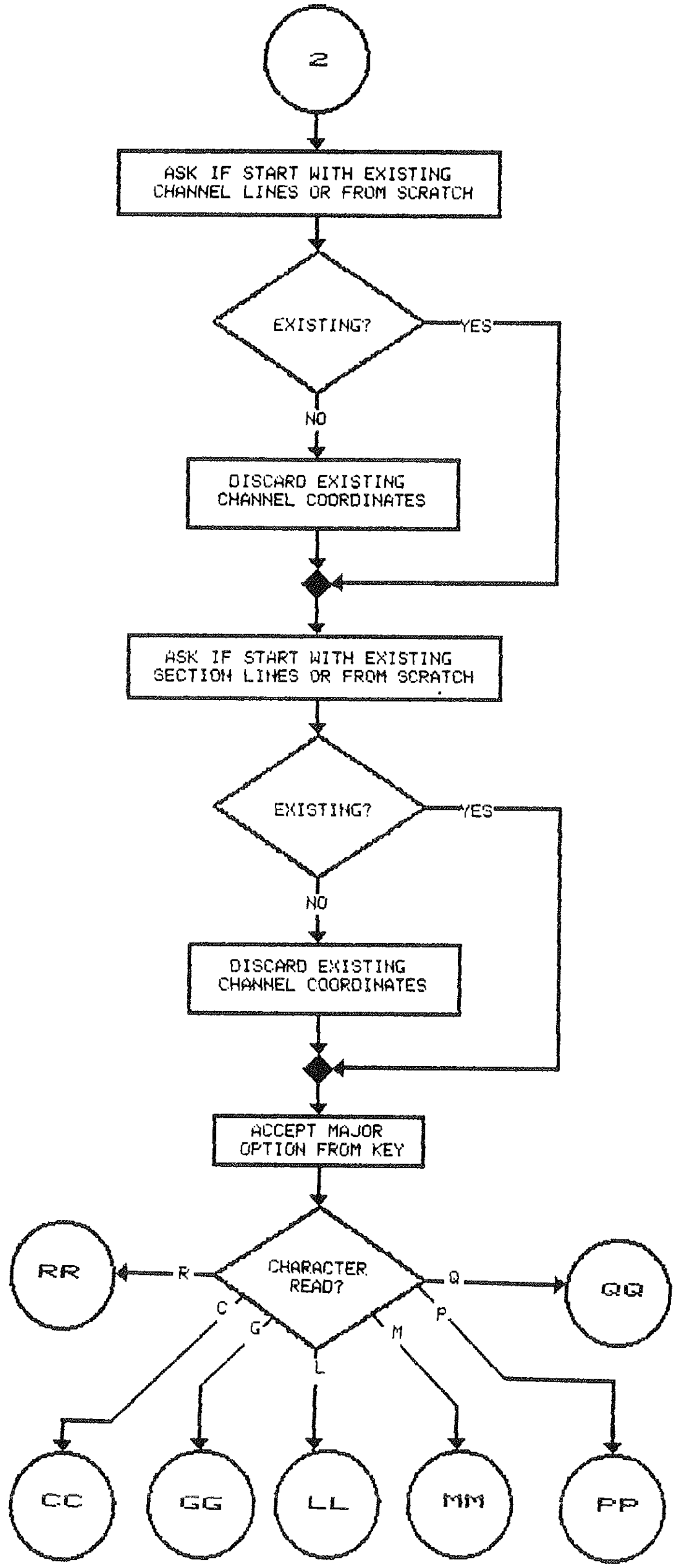


Figure C71. (Sheet 2 of 8)

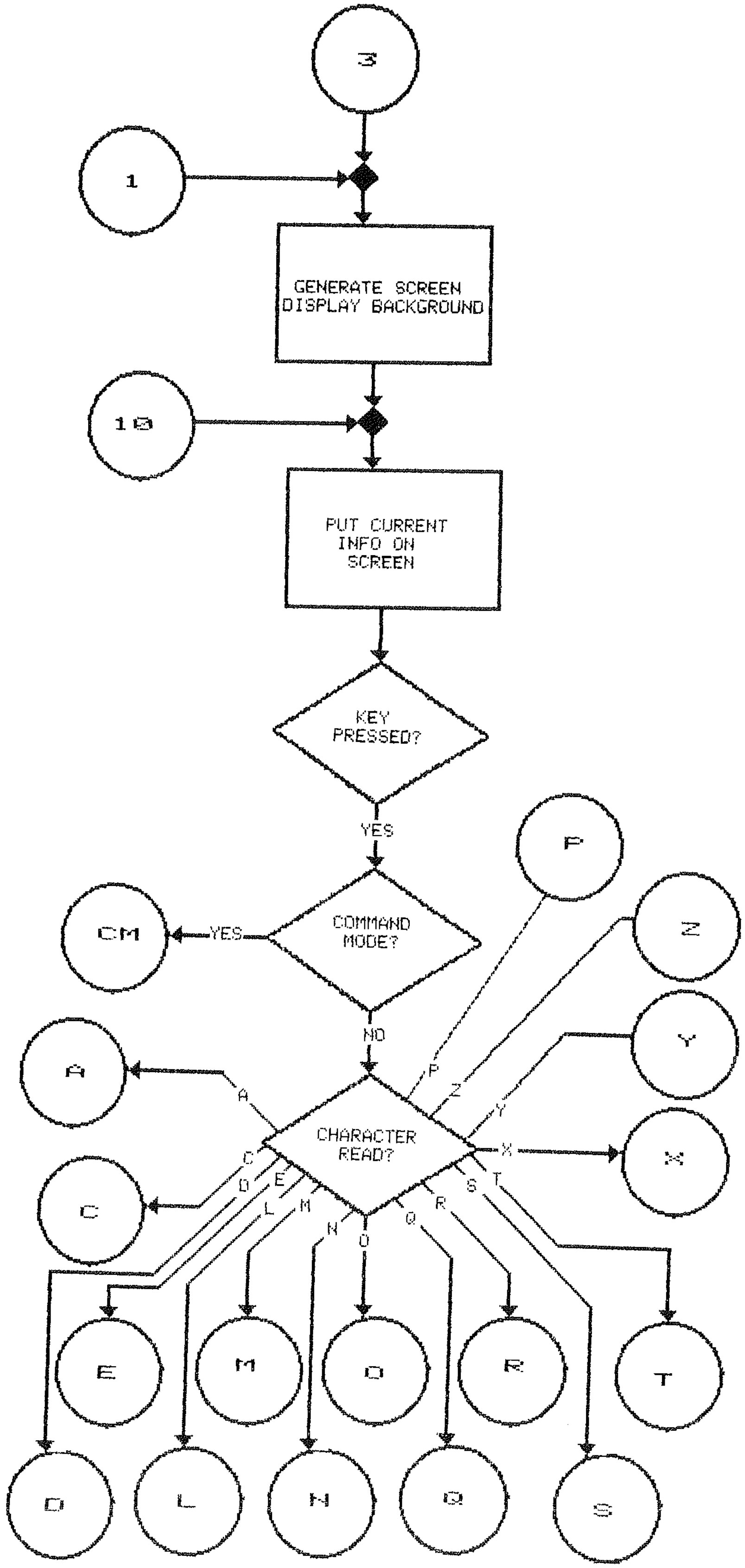


Figure C71. (Sheet 3 of 8)

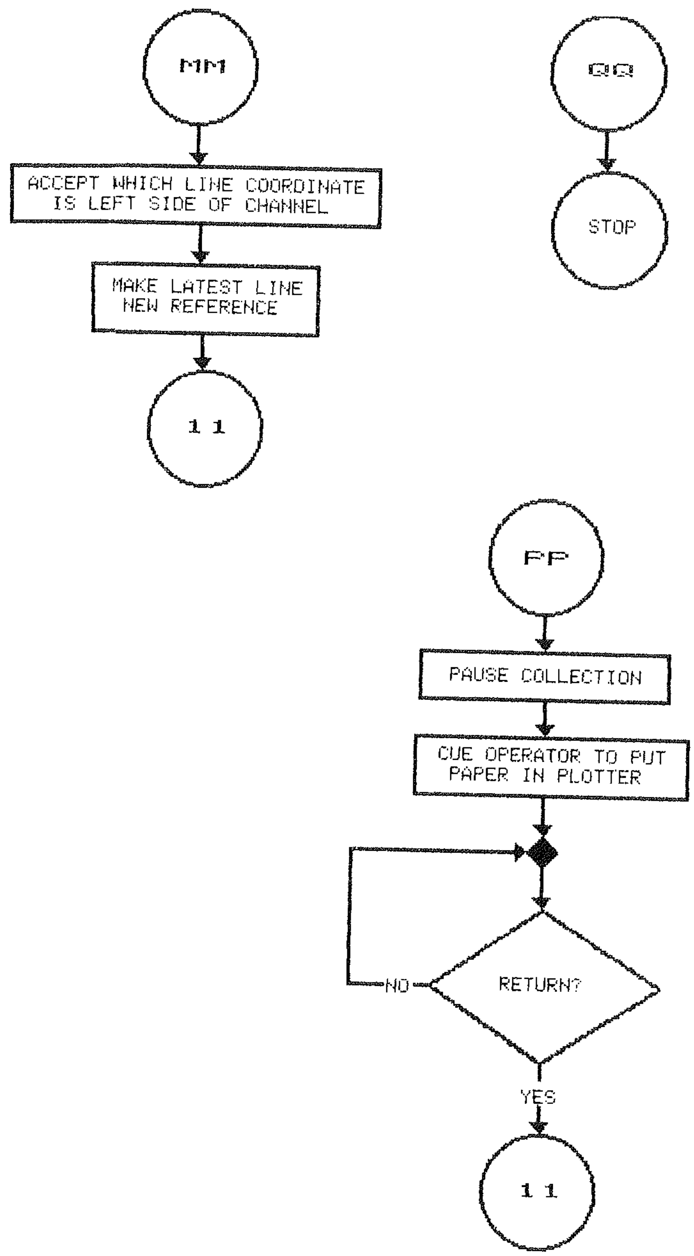


Figure C71. (Sheet 4 of 8)

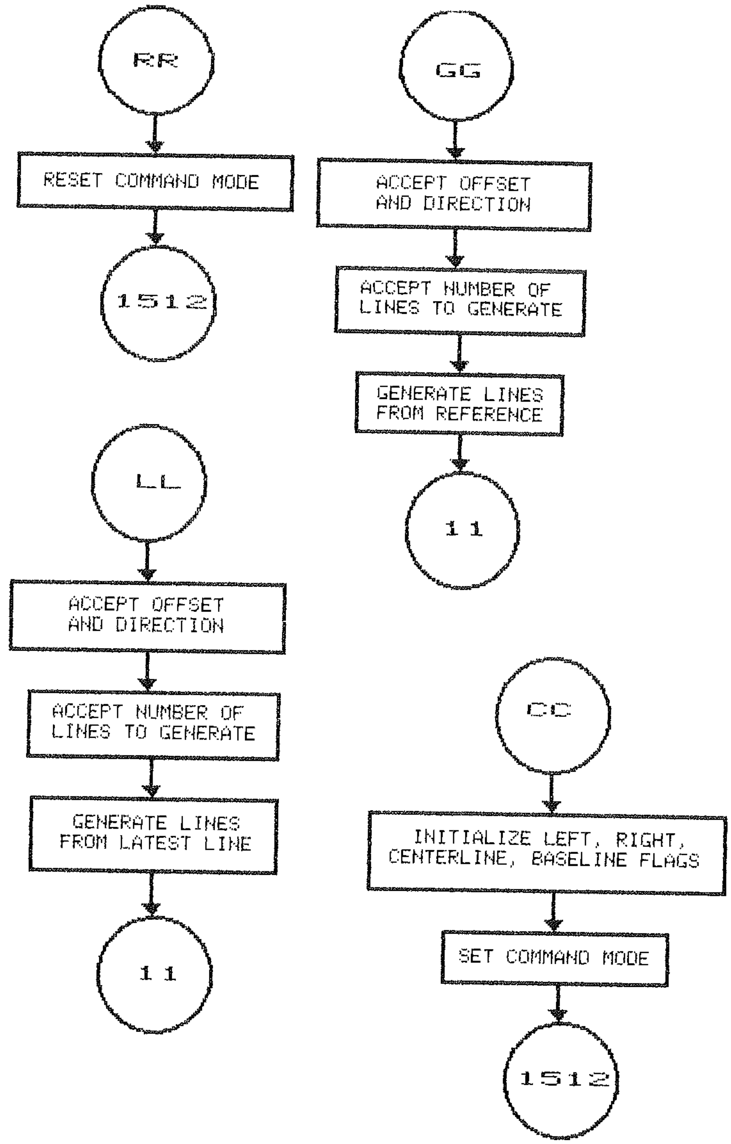


Figure C71. (Sheet 5 of 8)

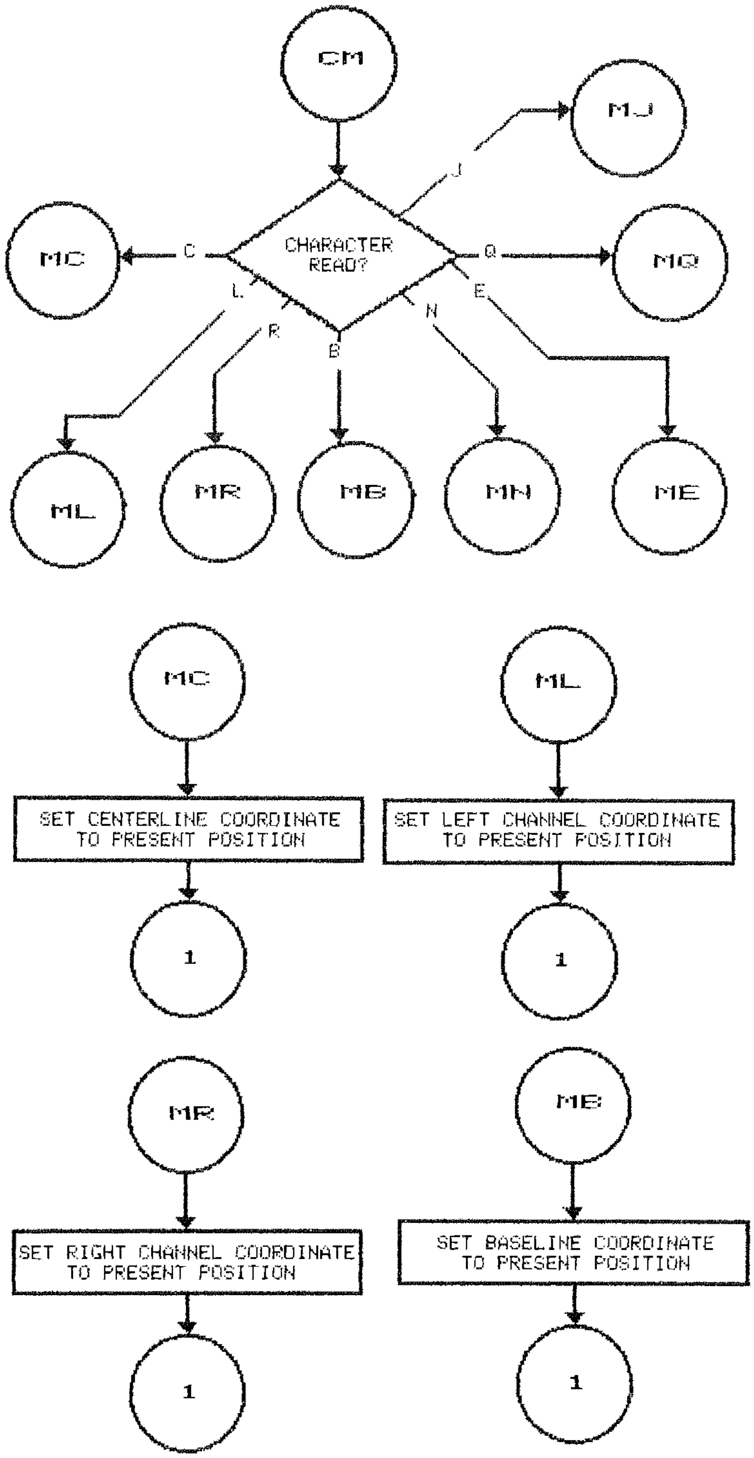


Figure C71. (Sheet 6 of 8)

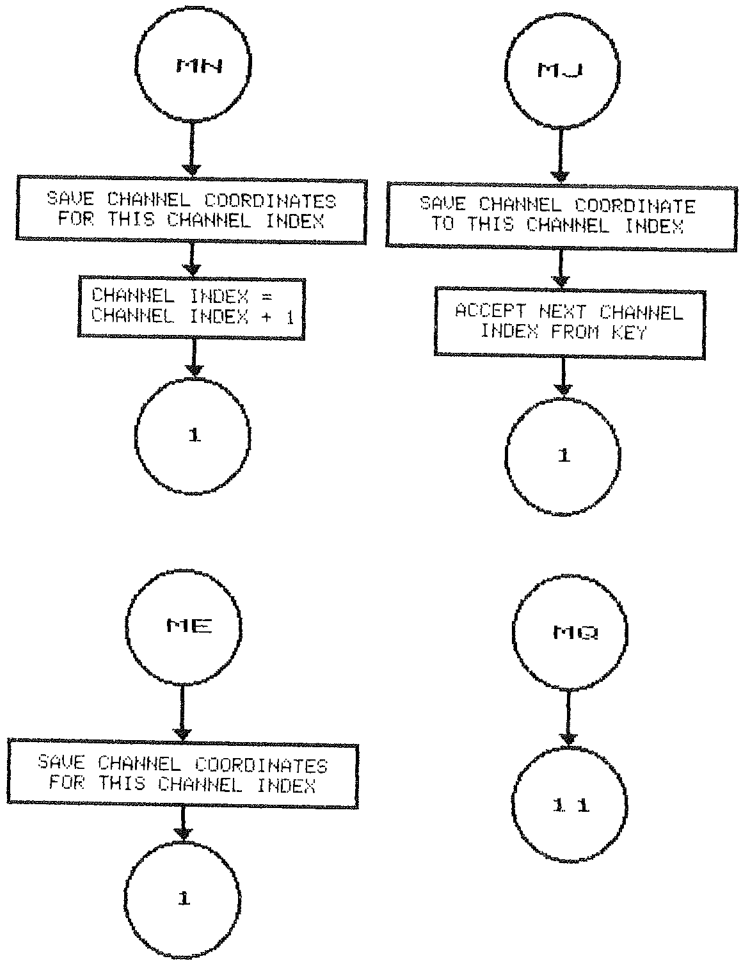


Figure C71. (Sheet 7 of 8)

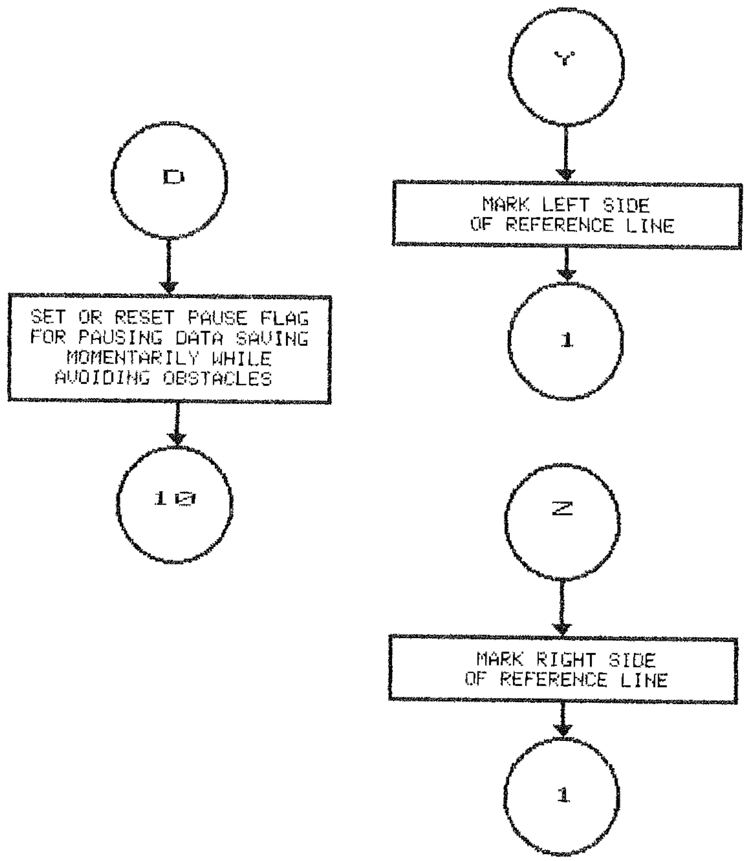


Figure C71. (Sheet 8 of 8)

Typical run?

Enter N

N

to start a new

Go to run mode

Use Y and Z to establish reference

Use Q to quit

Go to CHANNEL mode

Use L and R to establish 1st channel index

Use N to increment

Use L and R to establish 2nd index

Use E to end

Use Q to quit

Generate section lines

Go to RUN mode and run lines

Use Q to quit RUN mode when last line has been completed

Use Q option to end program

Typical pick up where left off?

Enter Y

Y

and continue as before

Typical Random?

Enter N

N

Go to CHANNEL mode and mark channel

Go to RUN mode and use S and E keys to collect data

Use Q to quit RUN mode

Use Q option to quit program

Figure C72. Typical run, RANDOM

RANGE_DEVICE

Purpose

There are a number of range equipment programs and one simulation program. One of these programs runs in conjunction with the survey program, determined by the positioning equipment used. These programs collect ranges and/or azimuths, convert the readings to absolute position and relative-to-the-survey-line position, determine beginning and end of automatic data collection, calculate boat speed, and provide position information to other tasks.

Input files

NONE (data are passed through a common block of memory)

Output files

NONE (data are passed through a common block of memory)

Equipment required

Supported range equipment using RS232 interface

Modules called

TIME_INTO_TEST, OFFDIST, POSITRR OR POSITRA, POLREC, A2, DEL_NORTE (range equipment specific), INISER, RDSERI, WRSERI, TERMSE, PROBLEM

Theory of operation

The range equipment program is spawned by SURVEY, does some initialization, and waits for PLOT_SCREEN to set a flag before proceeding (PLOT_SCREEN is the last task spawned so all are memory resident at this time). It then begins a tight loop of acquiring ranges, calculating position, checking distance-along-line to see if time to trigger an event, and putting acquired data in the output buffer. It terminates when PLOT_SCREEN sets the death flag. The flowchart is shown in Figure C73.

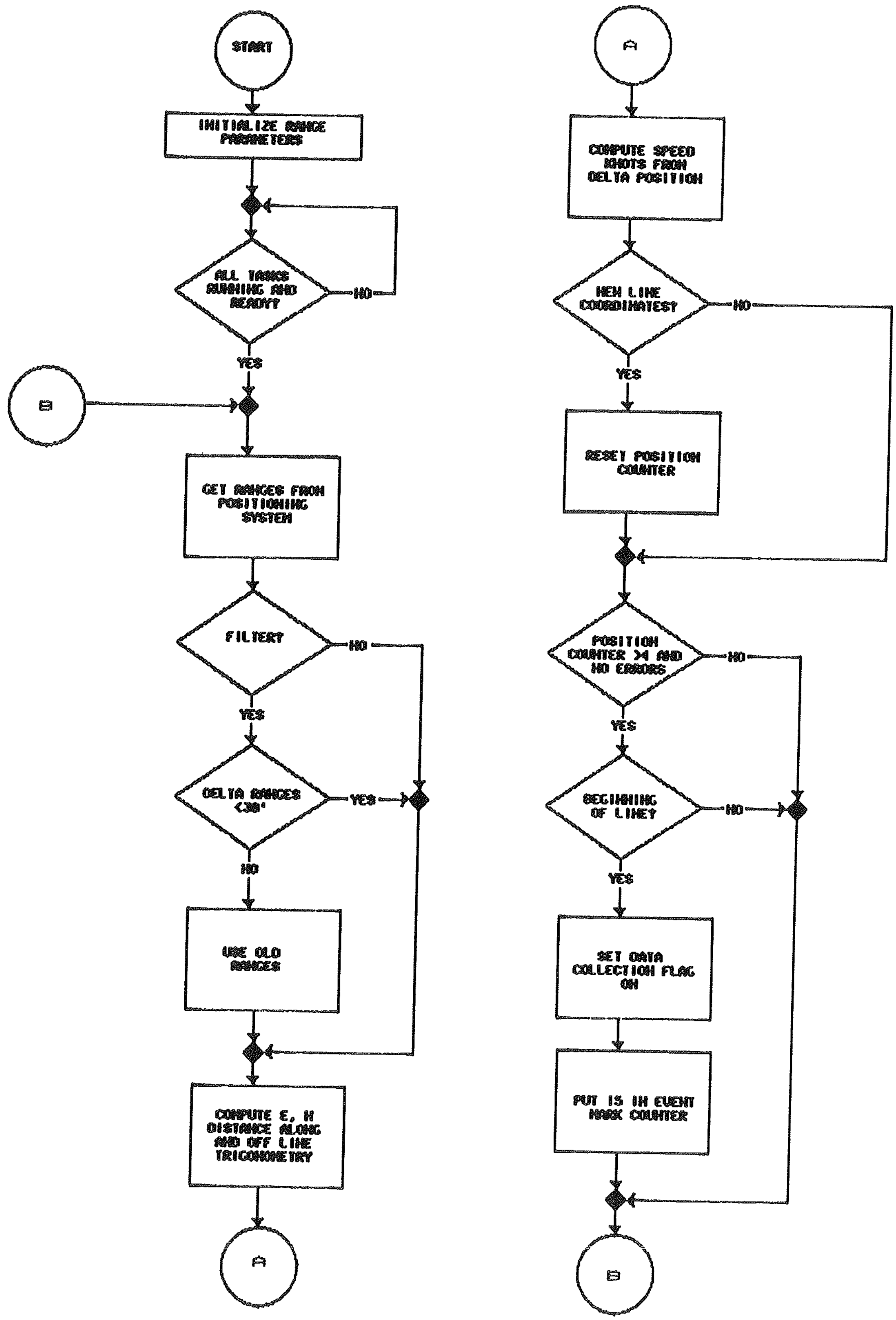


Figure C73. Flowchart,
RANGE_DEVICE

RANGE_SIMULATE

Purpose

RANGE_SIMULATE is a task in the survey package spawned by the task SURVEY. It is used to provide simulated position data when range equipment is not attached.

Input files

NONE (data are passed through system common memory)

Output files

NONE (data are passed through system common memory)

External devices

NONE

Modules called

TIME_INTO_TEST, DISTOFF, POLREC, A2, PROBLEM

Theory of Operation

RANGE_SIMULATE initializes parameters then waits for PLOT_SCREEN to set a flag indicating all tasks are running. It then begins a loop generating a point approximately every 0.8 sec. The program generates pseudo-absolute and relative position, and checks to see if it is time to trigger an event based on distance along the line. It then reloops until the death flag is set. The flowchart is shown in Figure C74.

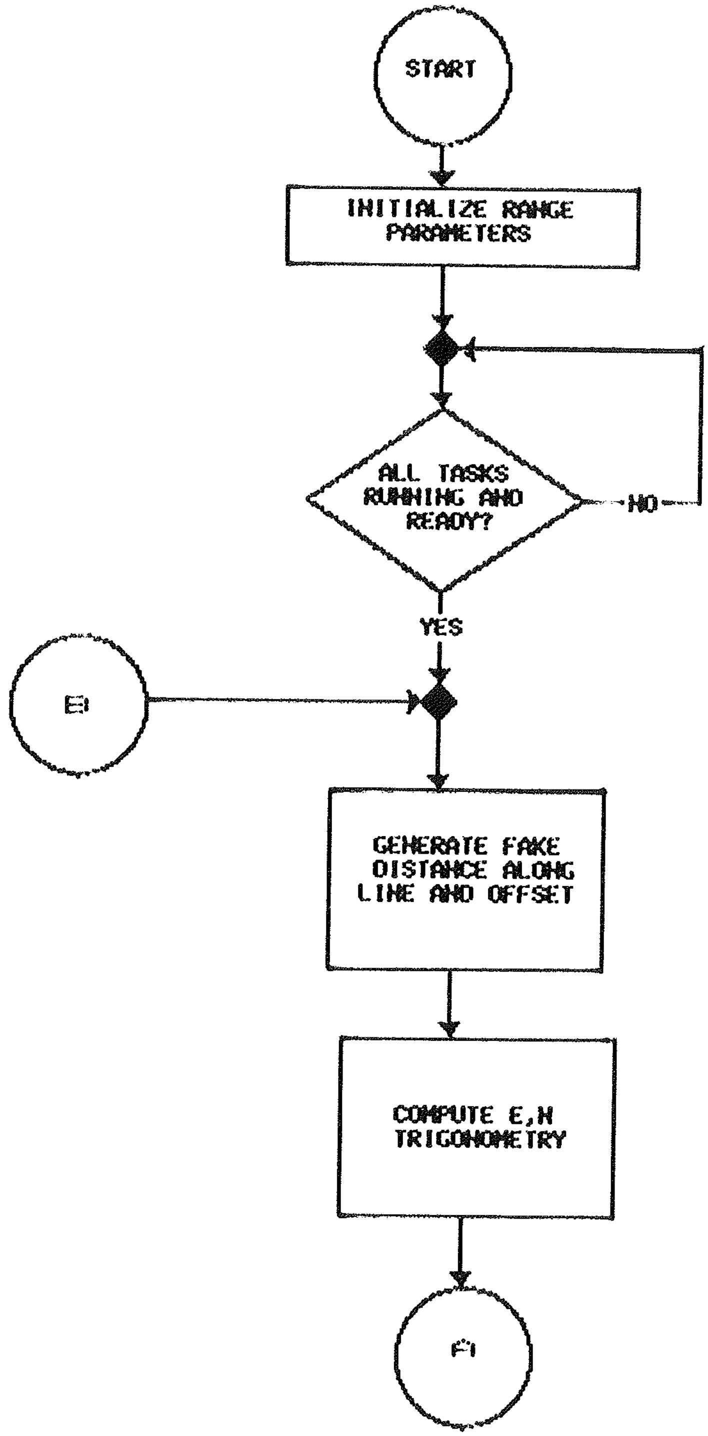


Figure C74. Flowchart, RANGE_SIMULATE
(Continued)

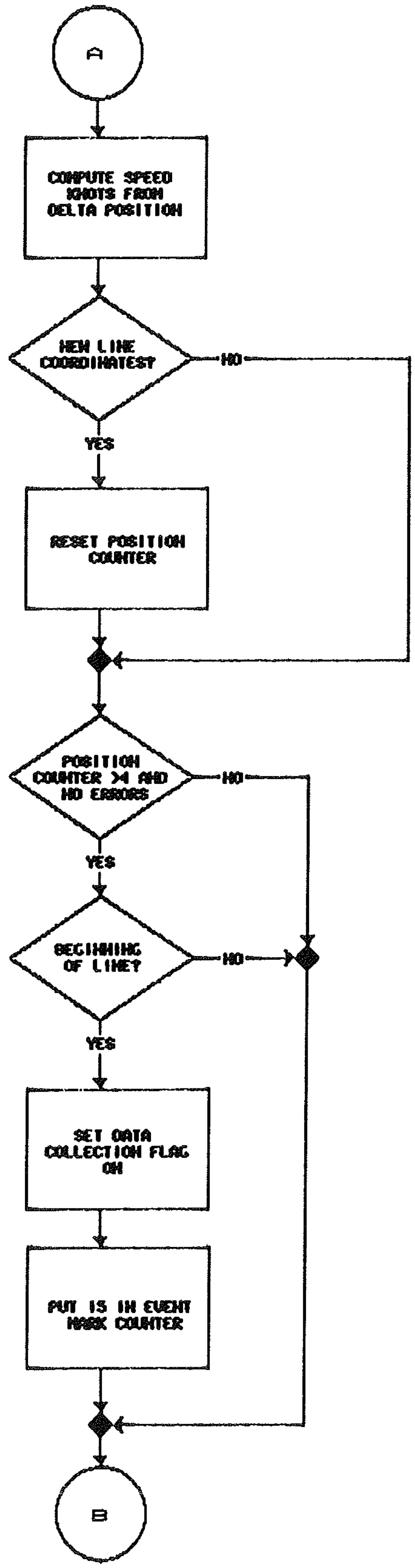


Figure C74. (Concluded)

SURVEY

Purpose

SURVEY is a survey task designed to start up the tasks needed to fulfill the desired survey. It must first determine the survey parameters by spawning a configuration (input/output) task. Then it spawns the needed tasks for the type survey specified.

Input files

NONE (uses common block of memory for data passing)

Output files

NONE (uses common block of memory for data passing)

External devices

NONE

Modules called

PROBLEM

Theory of operation

SURVEY begins by getting its own process identification number from the system. Then it sets its priority to 238. It creates two system common blocks of memory through which tasks can communicate. It then spawns its I/O task CONFIGURE and waits for CONFIGURE to complete its job and die. A definition of the survey to be run is now available in the common blocks of memory. Based on this information various tasks are spawned. A depth task is spawned at priority 239. A range task is spawned at priority 240. The pilot indicator task may be spawned at priority 241. The printing task may be spawned at priority 242. A plotting task may be spawned at priority 244. A screen graphic and operator interface task is spawned at priority 245. The task then loops waiting for the death flag to be set. It then delays for 5 sec before dying as it must be the last task to die since it created the common memory area (otherwise, the system would crash). The flowchart is shown in Figure C75, and a sample run is shown in Figure C76.

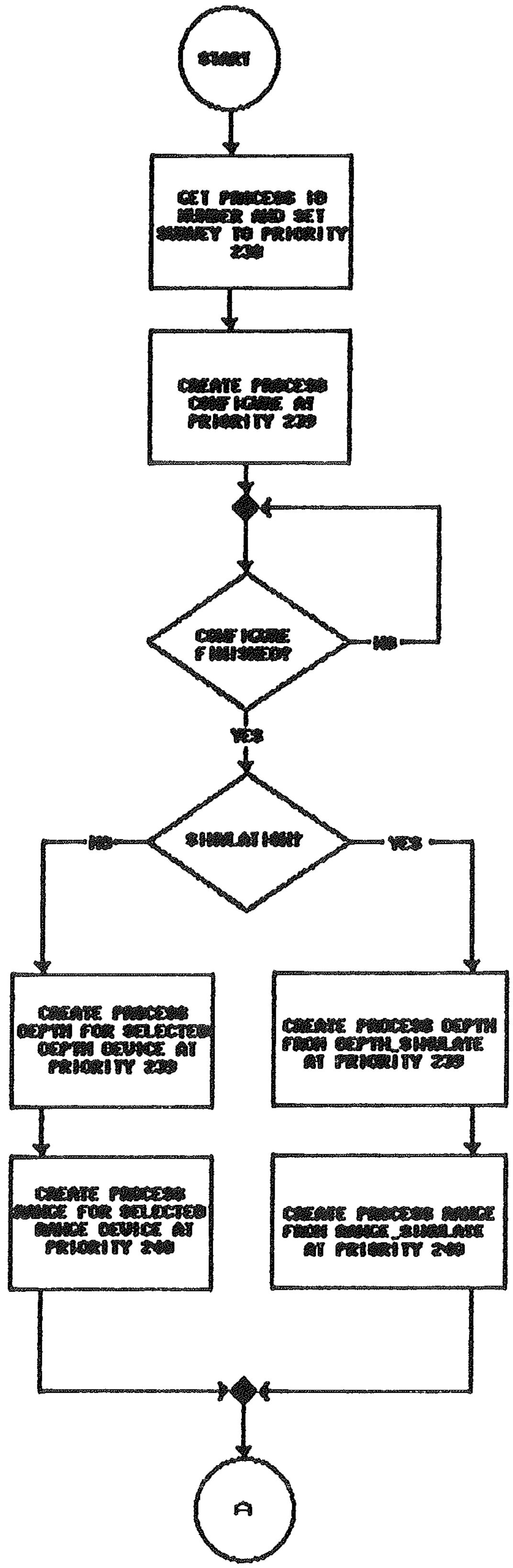


Figure C75. Flowchart, SURVEY
(Sheet 1 of 3)

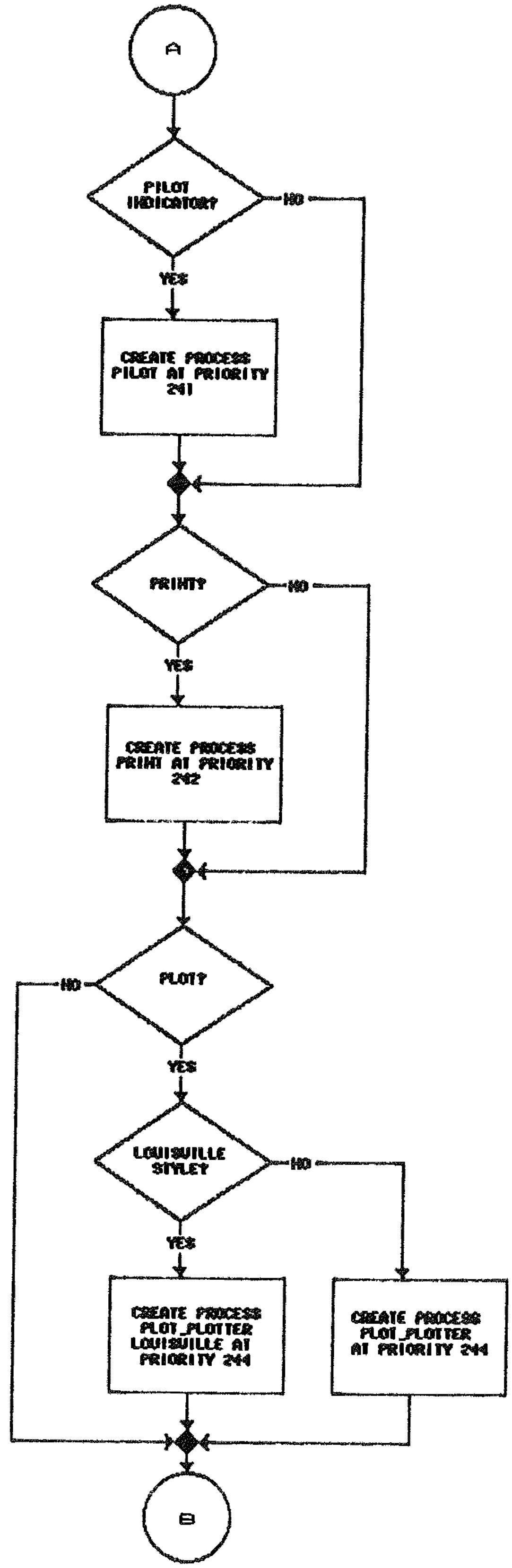


Figure C75. (Sheet 2 of 3)

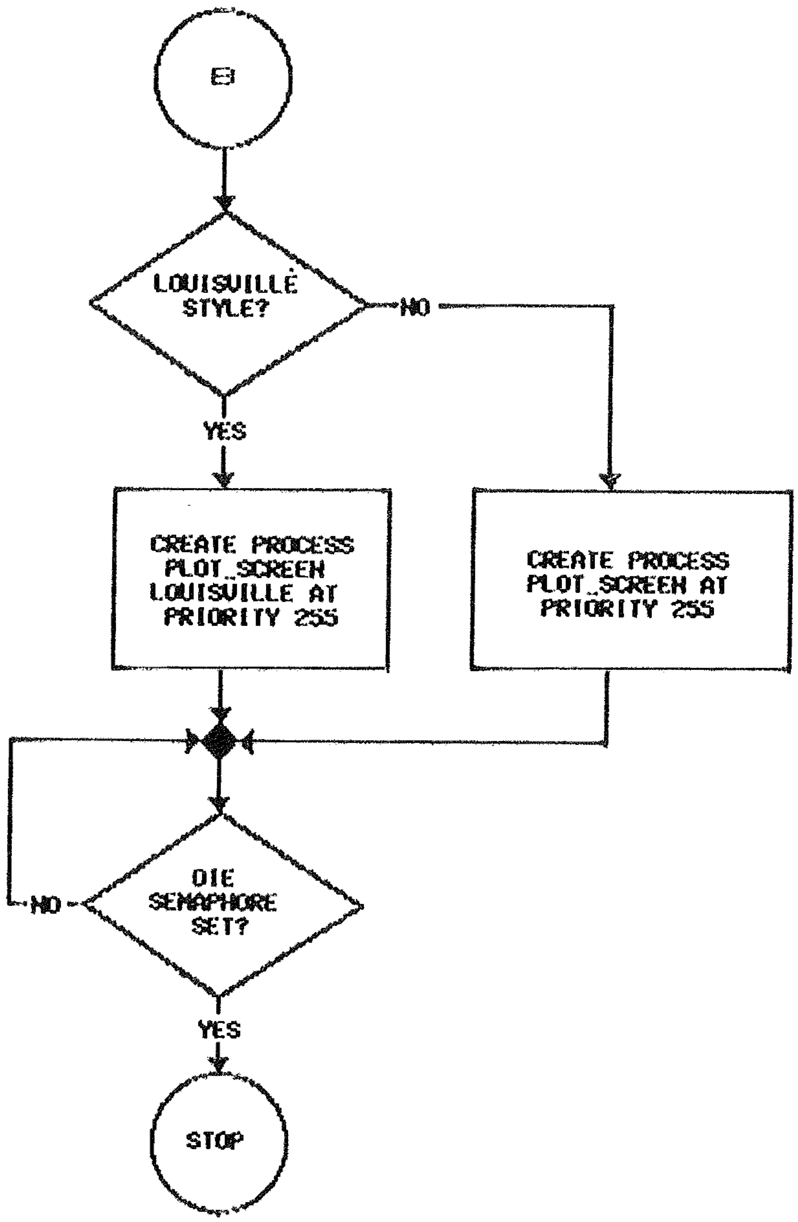
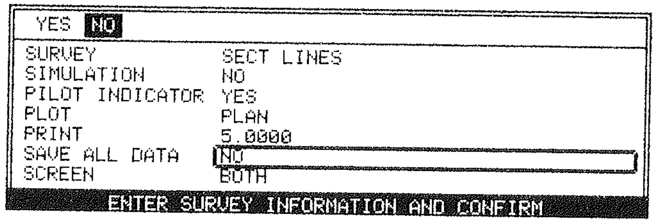
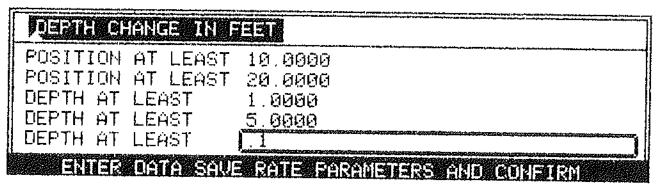


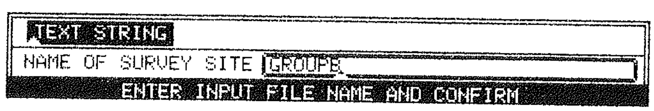
Figure C75. (Sheet 3 of 3)



SURVEY-1



SURVEY-2



SURVEY-3

Figure C76. Sample run, SURVEY

APPENDIX D: FILE AND MEMORY STRUCTURE

Main Run-Time Common Memory Block Layout (Buff)

ENTRIES OF 4 BYTES EACH FOR:

1. LIFE(1)/DEATH(0) FLAG FOR ALL PROCESSES (Tasks)
2. NEW LINE YES(1)/NO(0) FLAG
3. NEED NEW COORDINATE YES(1)/NO(0) FLAG
4. SAVING DATA YES(1)/NO(0) FLAG
5. AUTO START/STOP ENABLE(1)/DISABLE(0) FLAG
6. NOT USED
7. REAL SURVEY(0)/SIMULATION(1) FLAG
8. NOT USED
9. BUFFER POINTER TO NEXT AVAILABLE INDEX (integer)
10. NOT USED
11. LATEST DEPTH (FLOATING POINT)
12. LATEST EASTING (FLOATING POINT)
13. LATEST NORTHING (FLOATING POINT)
14. LATEST DISTANCE ALONG LINE (FLOATING POINT)
15. LATEST OFFSET FROM LINE (FLOATING POINT)
16. LENGTH OF SECTION LINE (FLOATING POINT)
17. PROJECT DEPTH (FLOATING POINT)
18. NOT USED
19. SLOPE OF LEFT SIDE OF CHANNEL (FLOATING POINT)
20. ALLOWED OVERDREDGE (FLOATING POINT)
21. NOT USED
22. NOT USED
23. EASTING OF START OF LINE (FLOATING POINT)
24. NORTHING OF START OF LINE (FLOATING POINT)
25. EASTING OF END OF LINE (FLOATING POINT)
26. NORTHING OF END OF LINE (FLOATING POINT)
27. ANGLE OF SURVEY LINE (FLOATING POINT)
28. NOT USED
29. NOT USED
30. NOT USED
31. BOAT ANTENNA HEIGHT FROM REFERENCE (FLOATING POINT)
32. DATE PART 1 GRID Format
33. DATE PART 2 GRID Format
34. DATE PART 3 GRID Format
35. NOT USED
36. LINE DIRECTION NORMAL(0)/REVERSE(1) FLAG
37. NOT USED
38. BOAT VELOCITY (FLOATING POINT)
39. RELATIVE TIME OF LATEST POSITION SINCE START OF PRESENT LINE (integer)
40. MARK EASTING (FLOATING POINT)
41. MARK NORTHING (FLOATING POINT)
42. MARK DEPTH (FLOATING POINT)
43. RELATIVE TIME SINCE START OF PRESENT LINE INTEGER
44. NOT USED
45. NOT USED
46. NOT USED
47. NOT USED
48. NOT USED

49. NOT USED
50. NOT USED
51. NOT USED
52. SIMULATION POINT COUNTER (integer)
53. SIMULATION DISTANCE ALONG THE LINE INCREMENT (FLOATING POINT)
54. GEOMETRY FLAG GOOD(0)/BAD(1)/POOR(2)
55. HEIGHT RED TRANSPONDER (FLOATING POINT)
56. EASTING RED TRANSPONDER (FLOATING POINT)
57. NORTHING RED TRANSPONDER (FLOATING POINT)
58. HEIGHT GREEN TRANSPONDER (FLOATING POINT)
59. EASTING GREEN TRANSPONDER (FLOATING POINT)
60. NORTHING GREEN TRANSPONDER (FLOATING POINT)
61. SIDE OF LINE BOAT IS ON FLAG
62. ALL TASKS STARTED YES PROCEED(1)/NO WAIT(0) FLAG
63. DEPTH CLOSEST IN TIME TO LATEST POSITION (FLOATING POINT)
64. MARK READY TO BE SAVED NO(0)/YES(1) FLAG
65. RANGE READY TO BE SAVED NO(0)/YES(1) FLAG
66. TIDE ENTRY READY TO BE SAVED NO(0)/YES(1) FLAG
67. TIDE ENTRY (FLOATING POINT)
68. RANGE_TYPE + RELATIVE TIME (Concatenated integer)
69. DISK I/O IN PROGRESS YES(1)/NO(0) FLAG
70. FIX MARKS TO DEPTH CHART YES(1)/NO(0) FLAG
71. FIX MARK INTERVAL (FLOATING POINT)
72. FIX MARK WIDTH IN DEPTH UPDATE INTERVALS (integer)
73. CONFIGURATION PROGRAM RUNNING YES(0)/NO(1) FLAG
74. PILOT INDICATOR YES(0)/NO(1) FLAG
75. RANGE DEVICE IDENTIFIER (integer)
76. DEPTH DEVICE IDENTIFIER (integer)
77. SURVEY TYPE IDENTIFIER (integer)
78. SECONDS/DEPTH READING SAVED (FLOATING POINT)
79. SECONDS/RANGE READING SAVED (FLOATING POINT)
80. SAVE ALL POINTS YES(0)/NO(1) FLAG
81. POOL ELEVATION (FLOATING POINT)
82. PLOT SCALE (FLOATING POINT)
83. DELTA TIME IN SECONDS BETWEEN PRINTING (FLOATING POINT)
84. CURRENT LINE NUMBER (integer)
85. PLOT VIEW NONE (0)/CROSS SECTION(1)/PLAN(2) FLAG
86. 1st CUBIC TRANSPONDER DISPLAY POSITION (integer)
87. 2nd CUBIC TRANSPONDER DISPLAY POSITION (integer)
88. RANGE FILTER DISABLED(0)/ENABLED(1) FLAG
89. CUBIC TEN THOUSANDS DIGIT FIRST TRANSPONDER (integer)
90. CUBIC TEN THOUSANDS DIGIT SECOND TRANSPONDER (integer)
91. CUBIC AUTO 10000 DIGIT CHANGE DISABLED(0)/ENABLED(1) FLAG
92. MARK TYPE NONE(0), GENERAL(1), LIGHT HOUSE(2), GREEN(3), RED(4) FLAG
93. BUFFER OVERRUN NO(0)/YES(1) FLAG
94. SCREEN DISPLAY CROSS SECTION(1)/PLAN VIEW(2)/BOTH(3) FLAG

ASCII Cross-Section Description File Format

"SITE.CROS"

8 BYTES OF FLAGS FOR PRINT CROSS SECTION, DISPLAY QUANTITIES DISPLAY CROSS SECTION, PLOT CROSS SECTION, PRINT QUANTITIES, PLOT QUANTITIES, ALIGN PAPER, PLOTTER TYPE (IN FORM 8A1) FOLLOWED BY CARRIAGE RETURN

21 BYTES OF PEN CARRIAGE NUMBERS FOR TEMPLATE, ALLOWED SIDE SLOPE, CENTER LINE, SURFACE, LETTERING, PREDREDGE DEPTHS, POSTDREDGE DEPTHS (7I3) FOLLOWED BY CARRIAGE RETURN

18 BYTES OF LINE TYPE NUMBERS FOR TEMPLATE, ALLOWED SIDE SLOPE, CENTER LINE, SURFACE, PREDREDGE DEPTHS, POSTDREDGE DEPTHS (6I3) FOLLOWED BY CARRIAGE RETURN

48 BYTES OF DASH LENGTH PERCENTAGES FOR TEMPLATE, ALLOWED SIDE SLOPE, CENTER LINE, SURFACE, PREDREDGE DEPTHS, POSTDREDGE DEPTHS (6F8.2) FOLLOWED BY CARRIAGE RETURN

40 BYTES OF LETTER HEIGHT, SCALE X, SCALE Y, PAPER SIZE X, PAPER SIZE Y (5F8.2) FOLLOWED BY CARRIAGE RETURN

ASCII Data File Format

LINE BLOCK

GROUPS OF LINE BLOCKS
THEN EOF

4 BYTES OF FILE NAME LENGTH (in characters) FOLLOWED BY CARRIAGE RETURN
FILE NAME OF DATA FILE (no. of characters in first record long)
FOLLOWED BY CARRIAGE RETURN
23 BYTES OF DATE/TIME(in form _12-31-1988__13:09:14.6) THEN CARRIAGE RETURN
14 BYTES OF LINE LENGTH (F14.2) THEN CARRIAGE RETURN
38 BYTES OF DIRECTION/CORRECTION FLAGS(_DIRECTION_FLAG__0__CORRECTION_FLAG__1)
THEN CARRIAGE RETURN
1 BYTE OF DELIMETER (ANY CHARACTER EXCEPT R OR D) THEN CARRIAGE RETURN

POSITION ENTRY

10 BYTE POSITION INDEX (IN FORM _RANGE__11) THEN CARRIAGE RETURN
58 BYTE TIME(seconds), Easting, Northing, Distance along line,Distance
off line(F10.2, 4F12.2) THEN CARRIAGE RETURN

DEPTH ENTRY

10 BYTE DEPTH INDEX (IN FORM _DEPTH__46) THEN CARRIAGE RETURN
22 BYTE TIME(seconds),Depth(F10.2, F12.2) THEN CARRIAGE RETURN

NOTE: THE DIRECTION FLAG IS 0 FOR A LINE DEFINED FROM THE LEFT SIDE OF CHANNEL (1 FOR RIGHT). THE CORRECT FLAG IS 0 FOR LINE RUN (i.e. DATA COLLECTED) FROM LEFT TO RIGHT SIDE OF CHANNEL (1 FOR RIGHT TO LEFT)

NOTE: FILE NAMES SHOULD BE IN FORM ARLINGTON.0001, ARLINGTON.0002, ETC. WHERE THE BASE PART IS THE SITE NAME (ARLINGTON) AND THE EXTENSION (.0001) IS THE LINE NUMBER DEFINED IN THE LINE FILE.

ASCII Equipment Description File Format

"SITE.EQUI"
6 BYTES OF RANGE DEVICE NUMBER (I6) FOLLOWED BY CARRIAGE RETURN
6 BYTES OF DEPTH DEVICE NUMBER (I6) FOLLOWED BY CARRIAGE RETURN
11 BYTES OF BOAT ANTENNA HEIGHT (F11.2) FOLLOWED BY CARRIAGE RETURN
6 BYTES OF NUMBER OF TRANSPONDER LOCATIONS (I6) FOLLOWED BY CARRIAGE RETURN
BLOCKS OF TRANSPONDER TRIORDINATES (NUMBER OF BLOCKS GIVEN ABOVE)

TRANSPONDER TRIORDINATE BLOCK

11 BYTES OF TRANSPONDER HEIGHT (F11.2) FOLLOWED BY CARRIAGE RETURN
11 BYTES OF TRANSPONDER EASTING (F11.2) FOLLOWED BY CARRIAGE RETURN
11 BYTES OF TRANSPONDER NORTHING (F11.2) FOLLOWED BY CARRIAGE RETURN

ASCII *Features Index File Format*

"FEATURES"

FILE NAMES OF UP TO 40 CHARACTERS (A40) FOLLOWED BY CARRIAGE RETURN
EOF

ASCII Feature Data File Format

"SITE.FEAT"
BLOCKS OF FEATURE ENTRIES UNTIL END OF FILE

FEATURE ENTRY BLOCK

12 BYTES OF PEN CARRIAGE SLOT, LINE TYPE, DASHLENGTH PERCENT (IN FORM 2I3, F6.2) FOLLOWED BY CARRIAGE RETURN
BLOCKS OF FEATURE COORDINATES UNTIL TERMINATOR
30 BYTE TERMINATOR (EITHER -998., -998. OR -999., -999. IN 2F15.2) THEN CARRIAGE RETURN

FEATURE COORDINATES

30 BYTE EASTING, NORTHING (IN 2F15.2) FOLLOWED BY CARRIAGE RETURN

ASCII Label Data File Format

"SITE.LABE"
BLOCKS OF LABEL INFORMATION
EOF

LABEL BLOCK

3 BYTES OF PEN CARRIAGE NUMBER (I3) FOLLOWED BY CARRIAGE RETURN
6 BYTES OF LABEL LENGTH IN CHARACTERS (I6) FOLLOWED BY CARRIAGE RETURN
LABEL OF UP TO 120 CHARACTERS (120A1) FOLLOWED BY CARRIAGE RETURN
30 BYTES OF EASTING, NORTHING OF LABEL START (2F15.2) FOLLOWED BY CARRIAGE RETURN
30 BYTES OF EASTING, NORTHING OF LABEL END (2F15.2) FOLLOWED BY CARRIAGE RETURN

ASCII Label Index File Format

"LABELS"
FILE NAMES OF UP TO 40 CHARACTERS (A40) FOLLOWED BY CARRIAGE RETURN
EOF

ASCII Letters Index File Format

"LETTERS"
FILE NAMES OF UP TO 40 CHARACTERS (A40) FOLLOWED BY CARRIAGE RETURN
EOF

ASCII Letter Data File Format

"SITE.LETT"
BLOCKS OF LINE OR TEXT ENTRIES UNTIL END OF FILE

LINE ENTRY BLOCK

12 BYTES OF BLOCK ROTATION ANGLE FOLLOWED BY CARRIAGE RETURN
12 BYTES OF BLOCK X POSITION FOLLOWED BY CARRIAGE RETURN
12 BYTES OF BLOCK Y POSITION FOLLOWED BY CARRIAGE RETURN
6 BYTES OF PEN CARRIAGE SLOT FOLLOWED BY CARRIAGE RETURN
12 BYTES OF 0.0 (LINE FLAG) FOLLOWED BY CARRIAGE RETURN
12 BYTES OF LINE START X FOLLOWED BY CARRIAGE RETURN
12 BYTES OF LINE START Y FOLLOWED BY CARRIAGE RETURN
12 BYTES OF LINE FINISH X FOLLOWED BY CARRIAGE RETURN
12 BYTES OF LINE FINISH Y FOLLOWED BY CARRIAGE RETURN

TEXT ENTRY BLOCK

12 BYTES OF BLOCK ROTATION ANGLE FOLLOWED BY CARRIAGE RETURN
12 BYTES OF BLOCK X POSITION FOLLOWED BY CARRIAGE RETURN
12 BYTES OF BLOCK Y POSITION FOLLOWED BY CARRIAGE RETURN
6 BYTES OF PEN CARRIAGE SLOT FOLLOWED BY CARRIAGE RETURN
12 BYTES OF LETTER HEIGHT FOLLOWED BY CARRIAGE RETURN
12 BYTES OF LETTER ROTATION ANGLE FOLLOWED BY CARRIAGE RETURN
12 BYTES OF LETTER START X FOLLOWED BY CARRIAGE RETURN
12 BYTES OF LETTER START Y FOLLOWED BY CARRIAGE RETURN
256 BYTES OF TEXT

ASCII Line File Format

6 BYTES OF NUMBER OF FIVE-LINE-IDENTIFICATION-BLOCKS (I6) FOLLOWED BY CARRIAGE RETURN

BLOCKS OF 5 FORTY CHARACTER LINES OF DESCRIPTION (NO. OF BLOCKS IN RECORD 1), EACH LINE FOLLOWED BY CARRIAGE RETURN

6 BYTES OF NUMBER OF CHANNEL COORDINATE INDICES (I6) FOLLOWED BY CARRIAGE RETURN

BLOCKS OF CHANNEL COORDINATES (SEE FORMAT BELOW)

6 BYTES OF NUMBER OF SECTION LINES (I6) FOLLOWED BY CARRIAGE RETURN

BLOCKS OF SECTION LINES (SEE FORMAT BELOW)

CHANNEL COORDINATE BLOCK

56 CHARACTERS OF CHANNEL LEFT EASTING, CH LEFT NORTHING, CH RT EASTING, CH RT NORTHING (IN FORM 4E14.7) FOLLOWED BY CARRIAGE RETURN

56 CHARACTERS OF BASELINE EASTING, B.L. NORTHING, CENTER LINE EASTING, C.L. NORTHING (IN FORM 4E14.7) FOLLOWED BY CARRIAGE RETURN

47 CHARACTERS OF LEFT RISE, LEFT RUN, PROJECT DEPTH, OVERDEPTH,RT RISE,RT RUN (IN FORM 2I7, 2F13.6, 2I7) FOLLOWED BY CARRIAGE RETURN

SECTION LINE BLOCK

23 CHARACTERS OF STATION NUMBER, SECTION INDEX,CHANNEL SIDE (IN FORM E14.7, I7,A1)

56 CHARACTERS OF SECTION LINE START EASTING, S.L. START NORTHING, S.L. FINISH EASTING, S.L. FINISH NORTHING (IN FORM 4E14.7)

NOTE: SECTION INDEX IS THE CHANNEL SECTION THAT THE SECTION LINE FALLS IN AS DEFINED BY THE CHANNEL COORDINATE BLOCK INDEX FOR THE CHANNEL BREAK IMMEDIATELY PRECEDING THE SECTION LINE. THE CHANNEL SIDE ENTRY IS L FOR SECTION LINE BEGINNING ON THE LEFT SIDE OF THE CHANNEL (R FOR RIGHT SIDE).

ASCII Marks Description File Format

"SITE.MARK"
5 BYTES OF NUMBER OF MARKS IN FILE (I5) FOLLOWED BY A CARRIAGE RETURN
BLOCKS OF MARKS DESCRIPTION (NUMBER OF BLOCKS GIVEN ABOVE)

MARK BLOCK

3 BYTES OF PEN CARRIAGE SLOT (I3) FOLLOWED BY CARRIAGE RETURN
3 BYTES OF SPECIAL SYMBOL NUMBER (I3) FOLLOWED BY CARRIAGE RETURN
13 BYTES OF MARK EASTING (F13.1) FOLLOWED BY CARRIAGE RETURN
13 BYTES OF MARK NORTHING (F13.1) FOLLOWED BY CARRIAGE RETURN
40 BYTES (OR LESS) OF MARK LABEL (40A1) FOLLOWED BY CARRIAGE RETURN

Binary Data File Format

TIDE
"SITE.D001"
12 BYTES OF DATE (GRID FORMAT)
4 BYTES OF DATA IDENTIFIER(250) AND RELATIVE TIME
4 BYTES OF TIDE VALUE IN FLOATING POINT

MARK
"SITE.M001"
12 BYTES OF DATE (GRID FORMAT)
4 BYTES OF DATA IDENTIFIER(251) AND RELATIVE TIME
4 BYTES OF EASTING IN FLOATING POINT
4 BYTES OF NORTHING IN FLOATING POINT
4 BYTES OF DEPTH IN FLOATING POINT

POSITION & DEPTH DATA
"SITE.0001"
12 BYTES OF DATE (GRID FORMAT)

POSITION ENTRY
4 BYTES OF DATA IDENTIFIER(10) AND RELATIVE TIME
4 BYTES OF EASTING IN FLOATING POINT
4 BYTES OF NORTHING IN FLOATING POINT
4 BYTES OF DISTANCE ALONG LINE IN FLOATING POINT
4 BYTES OF DISTANCE OFFLINE IN FLOATING POINT

DEPTH ENTRY
4 BYTES OF DATA IDENTIFIER(9) AND RELATIVE TIME
4 BYTES OF DEPTH IN FLOATING POINT

NOTE: SITE is the site identifier name; the number 001 is the number of the particular tide entry, Mark entry, or Data file and ranges to 999 or 9999.

NOTE: Data Identifier is a 2 byte integer; Relative time is a 2 byte integer representing number of tenths of second since the time in the date record at the beginning of file.

Auxiliary Run-Time Common Block of Memory Layout (BUFF2)

EACH INDEX VALUE CORRESPONDS TO 4 BYTES

1-50 SITE TEXT CHARACTER INFORMATION (5 FORTY CHARACTER LINES)
51-75 NOT USED
76-200 CHANNEL INDEX NUMBER OF LINE (500)
201-250 HEIGHTS OF TRANSPONDERS FLOATING POINT (50)
251-300 NORTHINGS OF TRANSPONDERS FLOATING POINT (50)
301-350 EASTINGS OF TRANSPONDERS FLOATING POINT (50)
351-400 PROJECT DEPTHS OF CHANNEL INDICES FLOATING POINT (50)
401-450 BASELINE EASTINGS OF CHANNEL INDICES FLOATING POINT (50)
451-500 BASELINE NORTHINGS OF CHANNEL INDICES FLOATING POINT (50)
501-550 CENTER LINE EASTING OF CHANNEL INDEX FLOATING POINT (50)
551-600 CENTER LINE NORTHING OF CHANNEL INDEX FLOATING POINT (50)
601-650 CHANNEL TOE LEFT EASTING OF CHANNEL INDEX FLOATING POINT (50)
651-700 CHANNEL TOE LEFT NORTHING OF CHANNEL INDEX FLOATING POINT (50)
701-750 CHANNEL TOE RIGHT EASTING OF CHANNEL INDEX FLOATING POINT (50)
751-800 CHANNEL TOE RIGHT NORTHING OF CHANNEL INDEX FLOATING POINT (50)
801-850 OVERDEPTH OF CHANNEL INDEX FLOATING POINT (50)
851-875 RISE OF LEFT SIDE OF CHANNEL (50)
876-900 RUN OF LEFT SIDE OF CHANNEL (50)
901 NOT USED
902 NUMBER OF CHANNEL INDICES (INTEGER)
903 NUMBER OF SECTION LINES (INTEGER)
904 NUMBER OF TRANSPONDER TRIORDINATES (INTEGER)
905 HEIGHT OF BOAT ANTENNA
906-909 SITE NAME
910 NUMBER OF CHARACTERS IN SITE NAME
911-999 NOT USED
1001-1500 STATION NUMBER FLOATING POINT (500)
1501-2000 SECTION LINE START EASTING (500)
2001-2500 SECTION LINE START NORTHING (500)
2501-3000 SECTION LINE FINISH EASTING (500)
3001-3500 SECTION LINE FINISH NORTHING (500)
3501 CENTER-LINE PLOT/CHANNEL LINE PLOT PEN NUMBERS
3502 BASELINE PLOT/DEPTH PLOT PEN NUMBERS
3503 LEGEND PLOT/SECTION LINE PLOT PEN NUMBERS
3504 ARROW PLOT/TRANSPONDERS PLOT PEN NUMBERS
3505 CENTER-LINE PLOT/CHANNEL LINE PLOT LINE TYPES
3506 BASELINE PLOT/SECTION LINE PLOT LINE TYPES
3507 CENTER-LINE DASH WIDTH PERCENTAGE
3508 CHANNEL LINES DASH WIDTH PERCENTAGE
3509 BASELINE DASH WIDTH PERCENTAGE
3510 SECTION LINE DASH WIDTH PERCENTAGE
3511 DEPTH LETTER HEIGHT PLOT
3512 SECTION LINE LETTER HEIGHT PLOT
3513 ARROW LENGTH PLOT
3514 ARROW X-POSITION PLOT
3515 LEGEND HEADER LETTER HEIGHT
3516 LEGEND ENTRY LETTER HEIGHT
3517 PEN FOR GRID MARKS / NUMBER OF CHARACTERS IN LEGEND TITLE
3518 LETTER HEIGHT OF GRID MARKS

3519	LEGEND CORNER / HARDWARE ROTATION ANGLE
3520	X-POSITION OF LEGEND REFERENCE
3521	Y-POSITION OF LEGEND REFERENCE
3522	TRANSPONDER SYMBOL / ALIGN PAPER FLAG
3523	PLOT LOWER LEFT EASTING
3524	PLOT LOWER LEFT NORTHING
3525	SOFTWARE ROTATION POINT EASTING
3526	SOFTWARE ROTATION POINT NORTHING
3527	SOFTWARE ROTATION ANGLE
3528	SCALE OF PLOT
3529	SOFTWARE ROTATION POINT X-VALUE
3530	SOFTWARE ROTATION POINT Y-VALUE
3531	GRID MARK SPACING
3532-3541	LEGEND TITLE (UP TO 40 CHARACTERS)
3542	Y-POSITION OF ARROW
3543	HEIGHT OF TRANSPONDER SYMBOL
3544-3549	NOT USED
3551-3675	CHANNEL SIDE FLAG (500)

ASCII Site File Format

"SITE.SITE"

6 BYTES OF NUMBER OF FIVE-LINE-IDENTIFICATION-BLOCKS (I6) FOLLOWED BY CARRIAGE RETURN

BLOCKS OF 5 FORTY CHARACTER LINES OF DESCRIPTION (NO. OF BLOCKS IN RECORD 1). EACH LINE FOLLOWED BY CARRIAGE RETURN

6 BYTES OF NUMBER OF CHANNEL COORDINATE INDICES (I6) FOLLOWED BY CARRIAGE RETURN

BLOCKS OF CHANNEL COORDINATES (SEE FORMAT BELOW)

6 BYTES OF NUMBER OF SECTION LINE GENERATION PARAMETER GROUPS TO FOLLOW THEN CARRIAGE RETURN

BLOCKS OF SECTION LINE PARAMETER GROUPS (SEE FORMAT BELOW)

CHANNEL COORDINATE BLOCK

56 CHARACTERS OF CHANNEL LEFT EASTING, CH LEFT NORTHING, CH RT EASTING, CH RT NORTHING (IN FORM 4E14.7) FOLLOWED BY CARRIAGE RETURN

56 CHARACTERS OF BASELINE EASTING, B.L. NORTHING, CENTER-LINE EASTING, C.L. NORTHING (IN FORM 4E14.7) FOLLOWED BY CARRIAGE RETURN

47 CHARACTERS OF LEFT RISE, LEFT RUN, PROJECT DEPTH, OVERDEPTH,RT RISE,RT RUN (IN FORM 2I7, 2F13.6, 2I7) FOLLOWED BY CARRIAGE RETURN

SECTION LINE PARAMETER GROUPS

RECTANGULAR METHOD

6 BYTES OF GROUP TYPE IDENTIFIER (_____1) FOLLOWED BY CARRIAGE RETURN

56 CHARACTERS OF STATION NUMBER, STATION INCREMENT, EASTING START, NORTHING START (IN FORM 4E14.7) FOLLOWED BY CARRIAGE RETURN

54 CHARACTERS OF EASTING FINISH, NORTHING FINISH, OFFSET, NUMBER OF LINES, SECTION INDEX (IN FORM 3E14.7, 2I6) FOLLOWED BY CARRIAGE RETURN

NONRECTANGULAR METHOD

6 BYTES OF GROUP TYPE IDENTIFIER (_____2) FOLLOWED BY CARRIAGE RETURN

56 CHARACTERS OF STATION NUMBER, STATION INCREMENT, EASTING START, NORTHING START (IN FORM 4E14.7) FOLLOWED BY CARRIAGE RETURN

54 CHARACTERS OF EASTING FINISH, NORTHING FINISH, OFFSET, NUMBER OF LINES, SECTION INDEX (IN FORM 3E14.7, 2I6) FOLLOWED BY CARRIAGE RETURN

56 CHARACTERS OF REFERENCE START EASTING, R.S. NORTHING, REFERENCE FINISH EASTING, R.F. NORTHING (IN FORM 4E14.7) FOLLOWED BY CARRIAGE RETURN

CHANNEL BASED METHOD

6 BYTES OF GROUP TYPE IDENTIFIER (_____3) FOLLOWED BY CARRIAGE RETURN

56 CHARACTERS OF STATION NUMBER, STATION INCREMENT, EXTENSION LEFT, EXTENSION RIGHT (IN FORM 4E14.7) FOLLOWED BY CARRIAGE RETURN

26 CHARACTERS OF SECTION INDEX, OFFSET, NUMBER OF LINES (IN FORM I6, E14.7, I6) FOLLOWED BY CARRIAGE RETURN

ASCII Screen Image Index File Format

"SCREEN_IMAGES"
BLOCKS OF SCREEN DESCRIPTORS
EOF

IMAGE DESCRIPTOR BLOCK

FILE NAME AS A CHARACTER STRING (up to 71 characters) FOLLOWED BY CARRIAGE RETURN
6 BYTES OF WIDTH OF IMAGE IN PIXELS (I6) FOLLOWED BY CARRIAGE RETURN
6 BYTES OF HEIGHT OF IMAGE IN PIXELS (I6) FOLLOWED BY CARRIAGE RETURN
6 BYTES OF CARRIAGE SLOT FOR PEN (I6) FOLLOWED BY CARRIAGE RETURN
13 BYTES OF SCALE IN FREE FIELD (F13.0) FOLLOWED BY CARRIAGE RETURN
13 BYTES OF X-ORIGIN IN FREE FIELD (F13.0) FOLLOWED BY CARRIAGE RETURN
13 BYTES OF Y-ORIGIN IN FREE FIELD (F13.0) FOLLOWED BY CARRIAGE RETURN
13 BYTES OF SOFTWARE ROTATION ANGLE IN FREE FIELD (F13.0) FOLLOWED BY CARRIAGE RETURN

ASCII Plan View Description File Format

"SITE.PLAN"

2 BYTES OF PEN CARRIAGE SLOT FOR CENTER LINE (I2) FOLLOWED BY CARRIAGE RETURN

2 BYTES OF PEN CARRIAGE SLOT FOR CHANNEL LINES (I2) FOLLOWED BY CARRIAGE RETURN

2 BYTES OF PEN CARRIAGE SLOT FOR BASELINE (I2) FOLLOWED BY CARRIAGE RETURN

2 BYTES OF PEN CARRIAGE SLOT FOR DEPTHS (I2) FOLLOWED BY CARRIAGE RETURN

2 BYTES OF PEN CARRIAGE SLOT FOR LEGEND (I2) FOLLOWED BY CARRIAGE RETURN

2 BYTES OF PEN CARRIAGE SLOT FOR SECTION LINES (I2) FOLLOWED BY CARRIAGE RETURN

2 BYTES OF PEN CARRIAGE SLOT FOR GRID MARKS (I2) FOLLOWED BY CARRIAGE RETURN

2 BYTES OF PEN CARRIAGE SLOT FOR ARROW (I2) FOLLOWED BY CARRIAGE RETURN

2 BYTES OF PEN CARRIAGE SLOT FOR TRANSPONDERS (I2) FOLLOWED BY CARRIAGE RETURN

OPTIONAL BLOCKS OF LINE INFORMATION (SEE FORMAT BELOW) CL, CH, BL, SL

OPTIONAL BLOCKS OF LETTER HEIGHT INFORMATION (SEE FORMAT BELOW) DEPTH, GRID, SL

OPTIONAL ARROW BLOCK INFORMATION (SEE FORMAT BELOW)

OPTIONAL LEGEND BLOCK INFORMATION (SEE FORMAT BELOW)

OPTIONAL TRANSPONDER BLOCK INFORMATION (SEE FORMAT BELOW)

14 BYTES OF PLOT SCALE (F14.2) FOLLOWED BY CARRIAGE RETURN

14 BYTES OF GRID MARK SEPARATION DISTANCE (F14.2) FOLLOWED BY CARRIAGE RETURN

4 BYTES OF HARDWARE ROTATE ANGLE (0, 90, OR -1 IN I4 FORMAT) FOLLOWED BY CARRIAGE RETURN

EITHER HARDWARE ROTATE BLOCK (IF 0 OR 90) OR SOFTWARE ROTATE BLOCK (IF -1) FOLLOWED BY CARRIAGE RETURN

1 BYTE OF ALIGN PAPER FLAG (A1) FOLLOWED BY CARRIAGE RETURN

4 BYTES OF PEN CARRIAGE SLOT FOR CONTOUR (I4) FOLLOWED BY CARRIAGE RETURN

OPTIONAL CONTOUR BLOCK INFORMATION (SEE FORMAT BELOW)

LINE BLOCK

2 BYTES OF DASH TYPE (I2) FOLLOWED BY CARRIAGE RETURN

OPTIONAL 14 BYTES OF DASH LENGTH PERCENTAGES (F14.2) FOLLOWED BY CARRIAGE RETURN

LETTER HEIGHT BLOCK

14 BYTES OF LETTER HEIGHT (F14.2) FOLLOWED BY CARRIAGE RETURN

ARROW BLOCK

14 BYTES OF ARROW LENGTH (F14.2) FOLLOWED BY CARRIAGE RETURN

14 BYTES OF X PAPER POSITION (F14.2) FOLLOWED BY CARRIAGE RETURN

14 BYTES OF Y PAPER POSITION (F14.2) FOLLOWED BY CARRIAGE RETURN

LEGEND BLOCK

14 BYTES OF LETTER HEIGHT FOR LEGEND HEADER (F14.2) FOLLOWED BY CARRIAGE RETURN

14 BYTES OF LETTER HEIGHT FOR LEGEND ENTRIES (F14.2) FOLLOWED BY CARRIAGE RETURN

2 BYTES OF NUMBER OF LEGEND HEADER CHARACTERS (I2) FOLLOWED BY CARRIAGE RETURN
LEGEND HEADER (NUMBER OF BYTES IN ABOVE RECORD LONG) FOLLOWED BY CARRIAGE RETURN
2 BYTES OF REFERENCE CORNER FOR LEGEND (A2) FOLLOWED BY CARRIAGE RETURN
14 BYTES OF LEGEND REFERENCE X POSITION (F14.2) FOLLOWED BY CARRIAGE RETURN
14 BYTES OF LEGEND REFERENCE Y POSITION (F14.2) FOLLOWED BY CARRIAGE RETURN

TRANSPONDER BLOCK

14 BYTES OF LETTER HEIGHT (F14.2) FOLLOWED BY CARRIAGE RETURN
2 BYTES OF SYMBOL NUMBER (I2) FOLLOWED BY CARRIAGE RETURN

HARDWARE ROTATE BLOCK

14 BYTES OF LOWER LEFT EASTING (F14.2) FOLLOWED BY CARRIAGE RETURN
OPTIONAL 14 BYTES OF LOWER LEFT NORTHING (F14.2) FOLLOWED BY CARRIAGE RETURN

SOFTWARE ROTATE BLOCK

14 BYTES OF ROTATION ANGLE (F14.2) FOLLOWED BY CARRIAGE RETURN
14 BYTES OF ROTATION POINT EASTING (F14.2) FOLLOWED BY CARRIAGE RETURN
14 BYTES OF ROTATION POINT NORTHING (F14.2) FOLLOWED BY CARRIAGE RETURN
14 BYTES OF ROTATION POINT X-POSITION (F14.2) FOLLOWED BY CARRIAGE RETURN
14 BYTES OF ROTATION POINT Y-POSITION (F14.2) FOLLOWED BY CARRIAGE RETURN

CONTOUR BLOCK

14 BYTES OF SHALLOWEST DEPTH TO CONTOUR (F14.2) FOLLOWED BY CARRIAGE RETURN
14 BYTES OF DEEPEST DEPTH TO CONTOUR (F14.2) FOLLOWED BY CARRIAGE RETURN
14 BYTES OF LETTER HEIGHT FOR CONTOUR LABEL (F14.2) FOLLOWED BY CARRIAGE RETURN
2 BYTES OF NUMBER OF CONTOUR LEVELS (I2) FOLLOWED BY CARRIAGE RETURN

APPENDIX E: GLOSSARY

Allowed sideslope - area below channel template sides which is not required to be dredged

Analog style - continuous reading, infinite resolution type device

Channel template - cross section view of desired channel

Communication links (radio links) - radio transmission paths and equipment for sending data from boat to shore or vice versa

Concurrent tasks - programs which interleave in time to give the appearance of simultaneous operation

Cross-section line - line used to indicate desired boat survey path across a channel

Data loggers - storage devices which record gathered data

Depth chart - a graph of depth versus paper motion (i.e. time) produced by some depth sounder systems; it will not be linear with respect to position unless boat speed is constant along the desired line

Depth sounder - electronic depth determining instrument

Disk drive - rotating magnetic media storage device

Electronic positioning - gathering of position data using electronic equipment

Electronic surveying tools (equipment) - electronic equipment used to automate gathering of survey data

Firmware - program stored in read-only memory

Form - a dialog box on the computer screen used for user-friendly data entry

FORTRAN ("FORMULA TRANSLATION") - a programming language designed for scientists and engineers used extensively by the Corps of Engineers

Global satellite positioning - positioning method based on three or four distances.

Hydrographic surveys - surveys conducted on water to gather data such as depth versus position

Joystick - device which changes mechanical motion to electronic energy change

Menu - a list of choices used for user-friendly selection of an option

Modem ("modulator demodulator") - device which allows digital data to be transmitted via telephone or radio links

On-line - during the data collection period

Overdepth - area below channel template bottom which is not required to be dredged

Piecewise linear interpolation - method of determining position at a given depth's sample time based on the relative time from the depth sampling to the positions taken immediately before and after the depth

Pilot guidance - a left/right gage indicating direction to steer the boat to follow a predetermined line

Prioritized multitasking - a system by which the program assigned the lowest "priority" number that is able to run is selected by the operating system to run

Prism - a circular glass target that reflects light back toward its source

Profile line - line used to indicate desired boat survey path along a channel

Quiz character - a method of triggering a device to take a sample and transmit it to the computer

Range-azimuth - positioning method based on a distance and one or more angles

Range-range - positioning method based on two distances

Removable cartridge bubble drive - a storage device using plug-in cassettes which contain magnetic bubble memory modules

Section line - line used to indicate desired boat survey path

Servo-drive - system where a motor turns proportionally to the motion of a joystick

Software package - a program or group of programs designed to perform a particular function

Space diversity - using two transponders at a position to reduce position errors due to reflected rather than direct path radio signals

Spawn - load a program from disk and start it running at a particular priority

Tablet digitizer - a device for converting a map via manual motion of a magnetic coil along the map to computer data

Tasks - programs which interact with other programs

Theodolite - surveying instrument used to determine angles and distances

Tide gages - instruments which measure relative water level