

AD-A100 942

NAVAL RESEARCH LAB WASHINGTON DC
A HIGHLY PORTABLE SYSTEM FOR ACQUIRING AND PROCESSING AXBT'S.(U)
JUL 81 D STEIGER
NRL-MR-4536

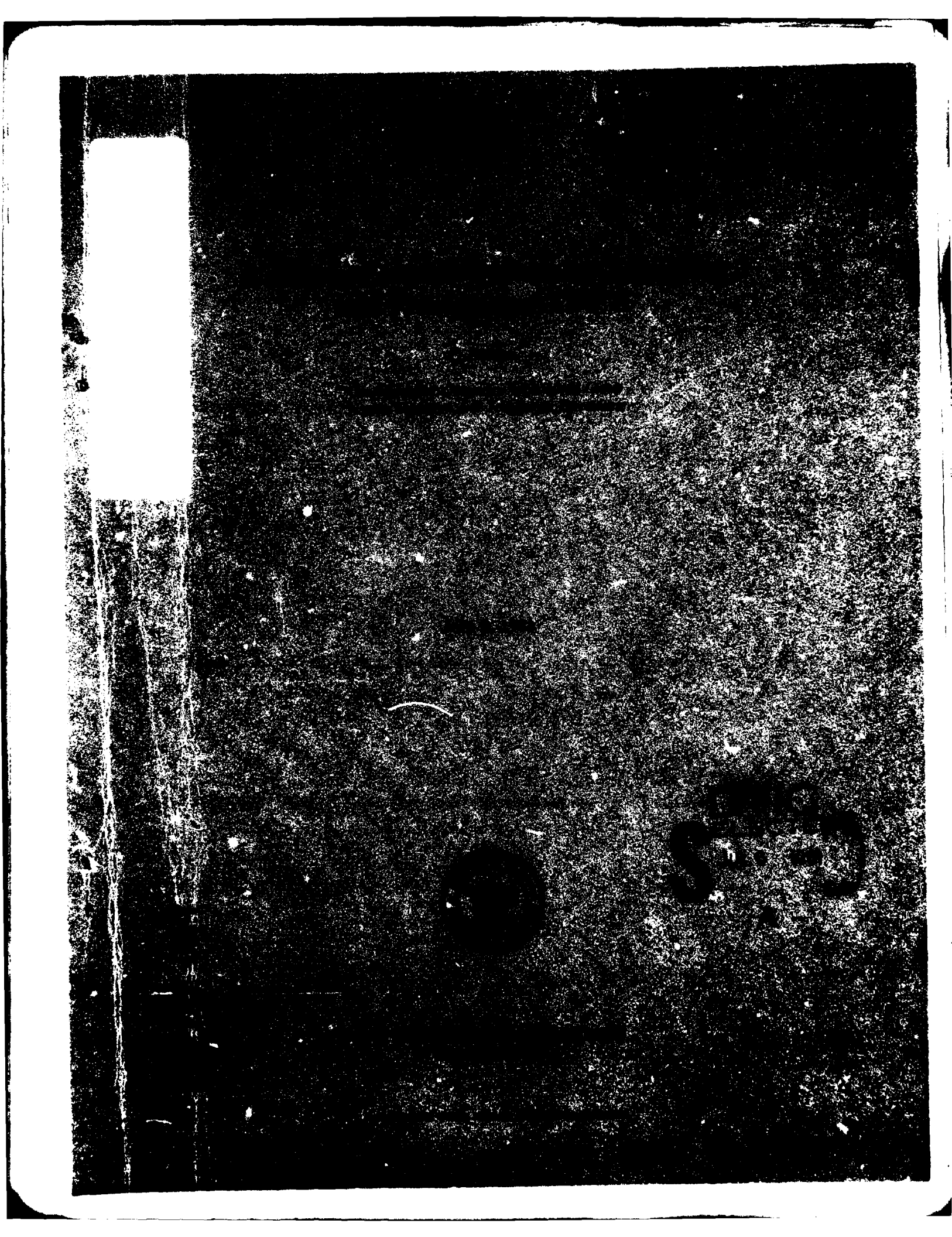
F/6 9/2

UNCLASSIFIED

NL

1 OF 1
48 A
100 942

END
DATE
FILMED
7-81
DTIC



REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER NRL Memorandum Report 4536 ✓	2. GOVT ACCESSION NO. AD-A100 942	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) A HIGHLY PORTABLE SYSTEM FOR ACQUIRING AND PROCESSING AXBT'S.		5. TYPE OF REPORT & PERIOD COVERED Interim report on a continuing NRL problem.
7. AUTHOR(s) D./Steiger		6. PERFORMING ORG. REPORT NUMBER 72-100
9. PERFORMING ORGANIZATION NAME AND ADDRESS Naval Research Laboratory Washington, DC 20375		8. CONTRACT OR GRANT NUMBER(s)
11. CONTROLLING OFFICE NAME AND ADDRESS		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 65056-6013
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)		12. REPORT DATE July 2, 1981
		13. NUMBER OF PAGES 22
		15. SECURITY CLASS. (of this report) UNCLASSIFIED
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE
16. DISTRIBUTION STATEMENT (of this Report) Approved for public release; distribution unlimited.		
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)		
18. SUPPLEMENTARY NOTES		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number) Real-time processing Portable data acquisition system Intelligent computer terminal Airborne Expendable Bathythermograph		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) At the Naval Research Laboratory highly portable intelligent computer terminals are being utilized aboard aircraft to acquire and process data received from Airborne Expendable Bathythermographs (AXBT's). Discussed in the paper is the design and implementation of the data acquisition and processing system which includes a detailed description of the hardware and software utilized.		

CONTENTS

I. INTRODUCTION 1

II. DESIGN OF PORTABLE DATA ACQUISITION AND PROCESSING SYSTEM 1

III. DATA ACQUISITION SOFTWARE DESCRIPTION 3

IV. PROGRAM DEVELOPMENT 5

V. DATA PROCESSING SOFTWARE DESCRIPTION 6

VI. DISCUSSION 6

REFERENCES 13

APPENDIX I ASSEMBLY LANGUAGE PROGRAM FOR AXBT DATA
ACQUISITION SYSTEM 14

APPENDIX II OPERATING INSTRUCTIONS FOR THE TERMINAL AXBT
DATA ACQUISITION SYSTEM 18

APPENDIX III LISTING OF AXBT PROCESSING PROGRAM 20

Accession For	
62-6241	<input checked="" type="checkbox"/>
143	<input type="checkbox"/>
Availability Notes	
Dist	Special
A	

A HIGHLY PORTABLE SYSTEM FOR ACQUIRING AND PROCESSING AXBT'S

I. Introduction

The Acoustics and Environmental Sciences Divisions of the Naval Research Laboratory are engaged in field studies for the Navy in many areas of the world. During these studies AXBT (Aircraft Expendable Bathythermographs) are utilized to obtain the temperature of the ocean from the surface to a depth of approximately 300 meters. The temperature information is used in acoustic propagation models and in studies to characterize the microstructure of the ocean. Also, the AXBT's have been used to locate and delineate ocean eddy's through which acoustic studies have been performed.

During experiments performed aboard an aircraft it is often impractical or inappropriate to install oceanographic computer systems due to the size, weight or resources required to install, maintain and operate the computer system. A terminal data acquisition and processing system is utilized due to its small size, light weight, ease of operation and highly portable nature. Several terminal systems have been developed which can be easily used aboard aircraft. One of these systems is used for acquiring and processing information from AXBT's.

The AXBT data acquisition and processing system was developed using a HP2645 Intelligent Terminal for data acquisition and recording. The AXBT data that is stored on cassette cartridges by the HP2645 terminal is processed by the HP2647 Intelligence Graphics terminal and hard copy numerical and graphical output is obtained by using an HP2631G graphics printer which is interfaced to the terminal. Figure 1 is a photograph of the AXBT data acquisition and processing system installed aboard an NRL aircraft.

II. Design of Portable Data Acquisition and Processing System

A functional diagram of the AXBT Intelligent Terminal Data Acquisition and Processing System is shown in Figure 2. The system consists of an RO-308/SSQ-36 Bathythermograph Data Recorder, CHRONO-LOG Series 70000 Time Code Generator, NRL designed Switch Panel, an HP2645 terminal, an HP2647 terminal and an HP2631G Printer. The components of the system are described in detail below.

1. RO-308/SSQ-36 Bathythermograph Data Recorder

The Bathythermograph Data Recorder is used to record ocean temperature information by converting radio signals transmitted by a temperature sensing sinking probe. The radio frequency signal is converted to a digital output in the form of an eight bit binary data word. The digital output of the Recorder is interfaced to the terminal AXBT data acquisition system.

Manuscript submitted April 7, 1981.

2. AXBT Data Acquisition and Processing System

The AXBT data acquisition system consists of two RO-308/SSQ-36 Data Recorders. The recorders are capable of receiving signals at several different frequencies. This is necessary if there are several AXBT's deployed from the aircraft in the same area. The frequency at which the AXBT will transmit is preset and the AXBT's are dropped in an order that alternates the frequency. The output of the AXBT recorder is in analog form and produced on a strip chart recorder and in a binary 8 bit format on a rear connector. The output is in degrees Fahrenheit with a resolution of one-half degree represented by the least significant bit.

The data acquisition system utilizes an HP2645 intelligent terminal and is shown in Figure 3. The terminal contains a programmable microprocessor, 128K bytes of ROM (Read Only Memory) and RAM (Random Access Memory), keyboard, display, cartridge tape units and control electronics. The terminal has all of the salient features of a microcomputer and can be used as such or as a computer terminal. The major features of the terminal are described below.

a. HP2645 Intelligent Terminal

The interior view of the terminal is shown in Figure 4. There are fifteen available circuit boards that can be inserted into the backplane of the terminal. Ten of these electronic circuit boards are used for control of the terminal leaving five interface locations available. Two of the control boards are high density HP13297A-003 32K Byte RAM boards. The strapping configurations for these boards are documented in Table I. For the AXBT data acquisition system three interfaces are required, namely, the AXBT Recorder, the Time Code Generator and the Switch Panel.

b. I/O Terminal Interfaces

The HP13255 Terminal Duplex Register Board described in Reference 1 was selected for interfacing with the AXBT Recorder, digital clock and switch panel. All of these devices provide binary outputs at TTL logic levels. The Duplex Register Board contains 8 data receiving lines and 8 status lines. It was recognized that the eight status lines could be used for data input as well as the eight data lines resulting in sixteen data lines for input. The polarity of the status lines on the interface is reversed from the data lines except for bits zero and one. By using the status lines the input capacity could be increased from five eight bit words to five sixteen bit words, thereby doubling the data acquisition capacity of the terminal. The problem of polarity can be handled with software by masking the two status bits of opposite polarity complementing the remainder and adding the two bits to the remainder to reform the byte.

c. External Storage

The HP2645 Terminal has two cassette drive units mounted below the display. Each cassette is capable of storing 110K Bytes of information. The information stored on these cassettes are source, object, assembler and debugger programs and the data acquired from the AXBT data acquisition system. The cassettes can be operated using functional keys from the keyboard or under program control. Both ASCII and binary types can be read and written by the

terminal.

3. Time Code Generator

The Time Code Generator provides digital BCD (Binary Coded Decimal) outputs of day of year, hour, minutes and seconds. The depth of the AXBT is determined by its fall rate in the ocean. For the SSQ-36 AXBT the fall rate is five feet per second. Therefore, only seconds need be recorded for determining AXBT depth in the ocean. When a temperature is recorded, seconds of time to determine the depth is recorded as well.

4. Switch Panel

The switch panel is used in the AXBT data acquisition system to record the AXBT number and to inhibit data recording prior to the AXBT sending useful information. Prior to recording AXBT information extraneous interrupts were found to occur resulting in erroneous data recording and the AXBT transmitter would continue to operate long after useful AXBT information was being transmitted.

The AXBT number is set into the least significant eight bits on the switch panel in BCD format. The AXBT number is recorded using the data bits on the terminal interfaces at the same time as a temperature reading is being taken. Bit fifteen on the switch panel is used to start and stop the data acquisition process. The eight most significant bits of the switch panel are interfaced to the status lines of the terminal interface board. Bit fifteen is checked by the software to determine when data recording is required. The AXBT data recorded on these tapes are used by the HP2647 data processing terminal as input.

5. HP2647 Data Processing System

The HP2647 Intelligent Graphics Terminal is constructed in an identical manner to the HP2645. The terminal has additional electronic boards to provide graphics capability and the use of a BASIC interpreter. Two interface slots are available in the terminal. An HP-IB interface was used in the terminal to output to the HP2631G graphics printer. The program in the terminal receives its data from cassettes and processes the data using a program written in BASIC.

III. Data Acquisition Software Description

Programs for the HP2645 intelligent terminal can be developed by preparing the source program and using the assembler available on the terminal or by using an HP1000 mini-computer system to prepare the program and provide a cross assembly for loading into the terminal. Since the debugging of the program can only be performed on the terminal the program for the terminal AXBT data acquisition was developed on the terminal.

The terminal uses a Intel 8080 compatible microprocessor. The microprocessor differences are in the way I/O is managed. Therefore, the program with the exception of I/O is Intel 8080 compatible. The terminal has many software subroutines stored in ROM that can be used by the program by addressing the starting location of the subroutines. These subroutines, since they

are stored in ROMs can not be altered. The routine PUTIO for performing I/O to the terminal display and cartridge tape units was used. This routine will write ASCII records to the display and either tape drive depending upon the device specified. The terminal AXBT program has been programmed to use only the right tape drive to store data.

The terminal has a 10 millisecond internal clock. The clock is used to schedule the AXBT program by storing the number of 10 millisecond intervals required in a location called TIMER which the terminal executive system decrements. Upon decrementing the location to zero the executive system software transfers control to a predetermined location. The starting address of the user program is stored at this location which in turn permits the scheduling of subroutines. The AXBT data acquisition program was scheduled to execute every second. This required the storing of 100 in the location TIMER which equates to one second. It was necessary to schedule the AXBT program to operate once per second because there were many interrupts occurring during this time and there was only a small variation of the temperature data over many seconds.

The terminal data acquisition program is entered by transferring control from the terminal executive program to the program CHTIMO. The function of CHTIMO is to schedule the data acquisition program to run at one second intervals. This is accomplished by checking for the TIMER location to go to zero. When 100 ten millisecond intervals have occurred software control transfers to the main program CNTRL, otherwise a return to the terminal executive program is executed.

The program CNTRL is used to call two subroutines, namely, INIT2, and INPUT. These programs are discussed below.

1. Subroutine INIT2

The program INIT2 stores 100 in the location TIMER which allows the terminal executive system to decrement the location TIMER 100 times, which takes one second before going to zero. The program INIT2 is called every time the program CHTIMO calls the program CNTRL.

2. Subroutine INPUT

The function of subroutine INPUT is to obtain the data from the external sensors and devices. It accomplishes this task by requesting data from the devices using a memory mapped I/O scheme. All of the three interface boards in the terminal have a unique address determined by the strapping configuration on the boards which are given in Table 2. Under program control a request is made of the sensor to send data. The data is buffered into the terminal interface I/O board. By addressing the terminal interface board with its unique address the data can be handled by the microprocessor under program control.

Program INPUT first addresses the status bits of the Switch Panel to determine if bit 15 is set. If bit 15 is set the data acquisition commences otherwise a return to the terminal executive program is executed. When bit 15 is set the program addresses the data bits of the Switch Panel which contains the AXBT number in BCD format. The AXBT number is then converted from

BCD to ASCII and stored in memory.

Temperature from the AXBT Recorder is obtained by setting the IN flip-flop on the terminal interface and waiting for the flip-flop to be reset by an interrupt from the recorder. Upon receiving an interrupt the binary data bits from the interface are converted from binary to ASCII and stored in memory. During this process the least significant bit representing one-half degree is masked off from the binary word before the conversion. If the bit exists an ASCII five is included in the temperature word otherwise an ASCII zero is included.

After obtaining the temperature value, seconds of time in BCD format is obtained by addressing the clock terminal interface. Seconds of time are also stored in memory to determine the depth of the AXBT when the temperature was recorded.

The AXBT number, temperature and seconds of time form one ASCII record. This record is output to the display and right cartridge tape by the program OUTERM. The program OUTERM moves the ASCII record to a system buffer and calls the system program PUTIO. This program records the data on the display and the right cartridge tape.

The assembly language program for AXBT data acquisition is documented in Appendix I.

IV. Program Development

The source program is written in a compatible INTEL 8080 language with the only exception being the I/O operations. These I/O operations are accomplished using programs stored in a terminal ROM, and by using memory mapped instructions to the I/O interfaces.

1. Preparing the Program

For assembling and loading, the source and binary programs must reside on cartridge tape. The source program can be placed on the tape by entering the source code into the terminal display memory through the terminal keyboard. Once in the display memory the source code is transferred to tape using the terminal function keys which provide the capability to transfer data between the terminal and other devices. An alternate method of obtaining the source code on tape is by keying the program into a file using the HP1000 mini-computer system. The file can then be edited and "dumped" to cartridge tape in ASCII format.

2. Assembling the Program

The HP13290B Debugger/Assembler is a commercially available product from Hewlett-Packard, and it resides on cartridge tape. By placing the tape in the left drive of the terminal it is loaded using the function keys on the terminal. Once having loaded the assembler the source program which resides on tape is placed on the left drive and blank tape to receive the assembled code in the right drive. After having successfully completed the assembly the right tape with the assembled code is then placed in the left tape drive and under keyboard command is loaded into the terminal. At this

point the program is ready for execution. Operating instructions for the AXBT data acquisition system are given in Appendix II. An alternate manner of assembling the program is to use the cross-compiler available on the HP1000 mini-computer system. The assembled program is stored on tape in the same format as the assembly on the terminal. Refer to Reference 2 for specific instruction on using the HP13290B Debugger/Assembler.

V. Data Processing Software Description

The HP2647 Intelligent Graphics Terminal is used to process the AXBT data acquired by the HP2645 terminal. The requirements of this processing are to read the data from the cassette tapes, convert the temperature from degrees Fahrenheit to degrees Centigrade, determine the depth from the seconds of time, and provide a hard copy listing and plot of temperature versus depth. The HP2647 terminal was selected as the data processing device because it supports a BASIC interpreter. Computations and plotting are made relatively easy as compared with assembly language programs written for the HP2645 terminal.

The processing program is written in BASIC programming language. The data is read from the right cartridge tape and stored into a buffer in the terminal. Often there are a few data points recorded at the start of an AXBT recording that are erroneous which are caused by spurious interrupts from the AXBT receiver. The program provides the option of deleting these points until the researcher is satisfied that the surface temperature is correct. These erroneous data points are easy to detect since readings are not reasonable values of temperature. After having determined the correct starting point of the data the AXBT records are read, converted to degrees centigrade, checked for reasonableness and stored in a memory buffer. If a reading is not reasonable, that is, if the previous value is more than three degrees different from the current value then the previous value is used since temperature of the water varies slowly. The data processing is concluded when the AXBT number changes.

The AXBT record in processed form is output to the printer following each reading. The program requests the operator to enter a "9" if ready for a plot, otherwise a "0". The plot is produced on the graphics display and a hardcopy of the plot can be obtained on the graphics printer by the operator.

Appendix III contains a program listing of the AXBT processing system.

VI. Discussion

The major benefits of using the terminal AXBT data acquisition and processing system is its compactness, lightweight and reliability. If final processed data is not required during flight only the data acquisition terminal, Time Code Generator and Switch Panel need be installed in the aircraft. The results obtained by the terminal AXBT system are identical to those that can be obtained by a mini-computer system. Other terminal data acquisition systems for magnetic field surveys and acoustic experiments aboard ship and aircraft have been developed and have been found to operate in the same reliable manner as the AXBT system. The magnetics system has been documented in Reference 5.

TABLE 1

SWITCH POSITIONS FOR HP13297A-003 32K
BYTE RAM

SWITCH	BOARD 1	BOARD 2
INH	OPEN	OPEN
32K	OPEN	CLOSED
16K	OPEN	OPEN
8K	OPEN	OPEN
4K	OPEN	OPEN
INH	OPEN	OPEN
32K	OPEN	CLOSED
16K	CLOSED	CLOSED
8K	OPEN	OPEN
4K	OPEN	OPEN
R.M	OPEN	OPEN
RAM	OPEN	OPEN
R.M	OPEN	OPEN
RAM	OPEN	OPEN
M1	CLOSED	CLOSED
.M2	OPEN	OPEN
.M3	OPEN	OPEN
FST	OPEN	OPEN
RPT	OPEN	OPEN
WPT	OPEN	OPEN

TABLE 2

JUMPER CONNECTIONS FOR HP13255
 TERMINAL DUPLEX BOARDS

DEVICE	A	B	C	D	E	F	G	H	J	K	L	M	N	P	Q	R	ADDRESS
TIME CODE GENERATOR	OUT	IN	OUT	OUT	IN	IN	IN	OUT	IN	OUT	IN	IN	OUT	OUT	IN	OUT	104XXX
SWITCH PANEL	OUT	IN	OUT	OUT	IN	OUT	IN	OUT	IN	OUT	IN	IN	OUT	OUT	IN	OUT	105XXX
AXBT RECEIVER	OUT	IN	OUT	OUT	IN	IN	OUT	OUT	IN	OUT	IN	IN	OUT	OUT	IN	OUT	106XXX



R-032

Fig. 1 — ANBL intelligent terminal data acquisition and processing system installed aboard NRL aircraft

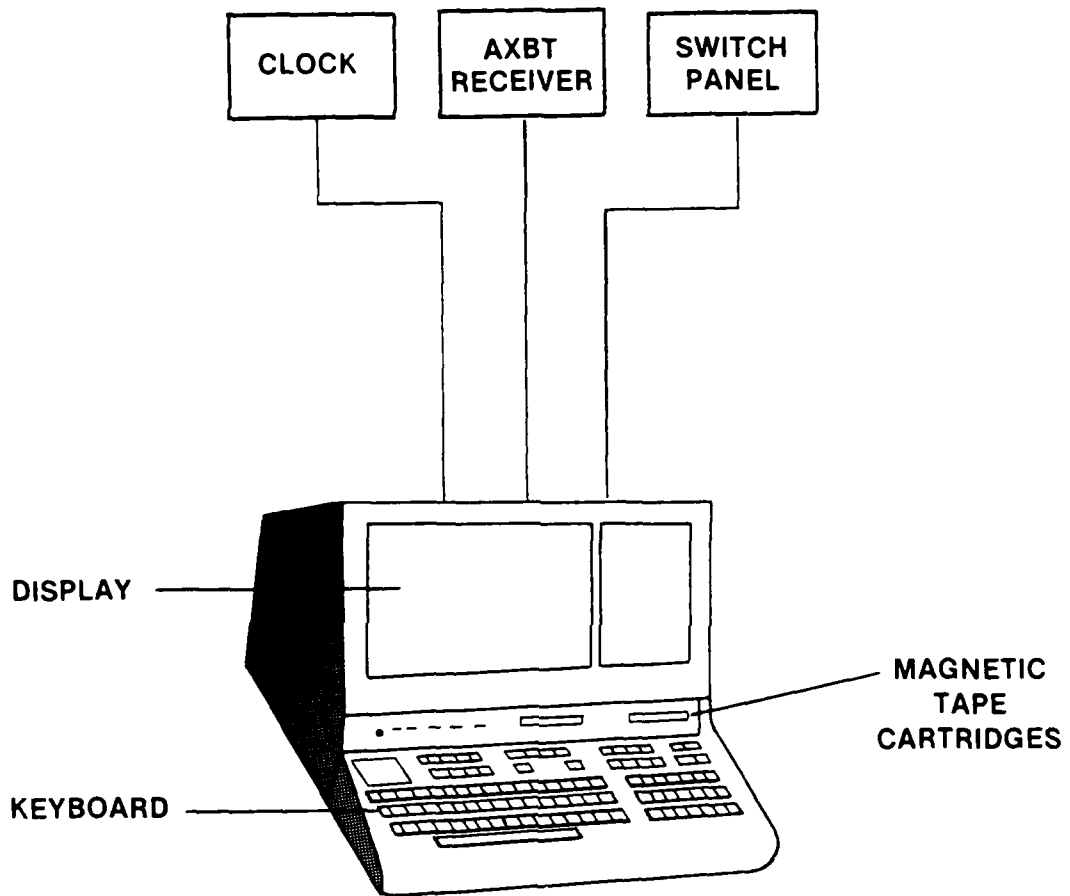


Fig. 2 — Functional block diagram of terminal AXBT system



R-031

Fig. 3 HP 2645 intelligent terminal



R-030

Fig. 4 -- Interior view of HP 2615 intelligent terminal

REFERENCES

1. HP13255 Terminal Duplex Register Module Manual, Hewlett Packard Part Number 13255-91031.
2. HP13290B Debugger/Assembler Reference Manual, Hewlett Packard Part Number: 13290-90009.
3. Clamons, J. D. and Steiger, D., "Can Intelligent Terminals and Modern Calculators Replace Oceanographic Computer Systems?", Woods Hole Oceanographic Institution Proceedings, Second Working Conference on Oceanographic Data Systems (September 1978).
4. Steiger, D., "Using Intelligent Graphics Terminals in Real-Time Processing", NRL Memorandum 4055 (August 24, 1979).

APPENDIX I

ASSEMBLY LANGUAGE PROGRAM FOR AXBT DATA ACQUISITION SYSTEM

***** T=00000 IS ON LU 20

```

0001 ;TAPE#1 TAB VER 13 XBT DATA TIME SET AT ONE SECOND
0002 ALTIO EQU 20B ;DEFINES PROGRAM AS ALT I/O DRIVER
0003 TIMER EQU 176147Q ;TIME OUT COUNTER
0004 PUTIO EQU 4199H ;SUBROUTINE TO OUTPUT DATA IN ASCII
0005 OUTDEV EQU 0FF4DH ;SPECIFIES OUTPUT DEVICE
0006 GTIOB0 EQU 3D1BH ;SYSTEM SUBROUTINE TO GET AN I/O BUFFER
0007 GETPTR EQU 3D46H ;SYSTEM BUFFER ADDRESS
0008 XFRLIM EQU 0FF47H ;SPECIFIES THE NUMBER OF CHAR TO OUTPUT
0009 INMAGD EQU 105001Q ;ADDRESS TO INPUT LS SWITCH PANEL CHARS
0010 INMAGS EQU 105000Q ;ADDRESS TO INPUT MS SWITCH PANEL CHARS
0011 SMINFF EQU 105007Q ;SETS IN FF ON SWITCH PANEL INTERFACE
0012 RMINFF EQU 105005Q ;RESETS IN FF ON SWITCH PANEL INTERFACE
0013 CMINFF EQU 105003Q ;ADDRESS TO READ FLAG ON SWTCH PANEL INT
0014 INDVMD EQU 106001Q ;ADDRESS TO INPUT AXBT READING MS CHARS
0015 INDVMS EQU 106000Q ;ADDRESS TO INPUT AXBT STATUS WORD
0016 SDINFF EQU 106007Q ;SETS AXBT IN FF
0017 RDINFF EQU 106005Q ;RESETS AXBT IN FF
0018 CDINFF EQU 106003Q ;ADDRESS TO READ FLAG ON AXBT INTERFACE
0019 INCLKD EQU 104001Q ;ADDRESS TO INPUT LS CLOCK CHARS(SEC)
0020 INCLKS EQU 104000Q ;ADDRESS TO INPUT MS CLOCK CHARS(MIN)
0021 SCINFF EQU 104007Q ;SETS CLOCK IN FF
0022 RCINFF EQU 104005Q ;RESETS CLOCK IN FF
0023 CCINFF EQU 104003Q ;ADDRESS TO READ FLAG ON CLOCK INT
0024 MASK1 EQU 17Q ;MASKS FOUR LSB,USED IN BCD TO ASCII SUBR
0025 MASK2 EQU 177Q ;MASKS MINUTES ON CLOCK DATA
0026 MASK3 EQU 376Q ;MASKS AXBT VALUE WITHOUT .5 DEG
0027 MASK4 EQU 1Q ;MASKS .5 DEG BIT ON AXBT VALUE
0028 ZERO3 EQU 374Q ;MASKS 6 MSBITS OF STATUS WORD,USED FOR REVSTA
0029 THREE EQU 3Q ;MASKS 2 LSBITS OF STATUS WORD,USED FOR REVSTA
0030 INSFF EQU 80H ;MASK TO CHECK RESET STATUS ON AXBT READING
0031 ;ENTRY VECTORS
0032 ORG 6000H ;ABSOLUTE STARTING ADDRESS IN HEX
0033 DB 50H ;ALTERNATE I/O CODE PRESENT
0034 DB 70H ;CHECK FOR CORRECT LOCATION
0035 JMP INIT1 ;INITIALIZATION FROM RESET
0036 JMP INIT2 ;INITIALIZATION FROM PROGRAM
0037 JMP RETURN ;INTERRUPT LOCATION NOT USED RETURN
0038 JMP MONIT ;MONITOR ROUTINE USED TO DECREMENT INTERVAL TIMER
0039 JMP INPUT ;DATA INPUT ROUTINE
0040 JMP OUTERM ;DATA OUTPUT ROUTINE
0041 JMP CONTRL ;ROUTINE TO CONTROL DATA ACQUISITION
0042 JMP RETURN ;STATUS LOCATION NOT USED RETURN
0043 JMP CHTIMO ;START ADDRESS OF PROGRAM,CHECK FOR TIMEOUT
0044 RETURN EQU $
0045 RET ;RETURN TO TERMINAL EXEC WAIT LOOP
0046 ;INPUT THE DATA
0047 INPUT EQU $
0048 LXI D,DATA01 ;LOAD ADDRESS OF DATA01 INTO REG D
0049 GETM EQU $
0050 LDA SMINFF ;SET IN FF ON SWITCH PANEL
0051 LDA RMINFF ;RESET IN FF ON SWITCH PANEL,GETS CURRENT READING
0052 LDA INMAGS ;INPUT STATUS BYTE OF SWITCH PANEL INTERFACE
0053 ANI INSFF ;(TEST FOR BIT 8 SET,
0054 SUI INSFF ; WHICH IS BIT 15 ON SWITCH PANEL)
0055 JM RETURN ;RETURN IF BIT 8 NOT SET,AXBT NOT ON
0056 LDA INMAGD ;READ DATA BITS ON SWITCH PANEL,AXBT NUMBER
0057 CMA ;COMPLEMENT DATA BITS
0058 PUSH B ;SAVE CONTENTS OF REG B IN STACK,GOOD HOUSEKEEPING
0059 CALL BCD2AS ;CONVERT AXBT NUMBER TO ASCII
0060 STAX D ;SAVE MSCHAR OF AXBT NUMBER
0061 INX D ;INCREMENT THE DATA STORAGE LOCATION
0062 MOV A,B ;GET LSCHAR AXBT NUMBER CNVRTD TO ASCII

```

```

0063          STAX D          ;SAVE IT
0064          INX D          ;INCREMENT DATA STORAGE LOCATION
0065          POP B          ;RESTORE REGISTER B
0066 GETD     EQU $          ;ROUTINE TO GET THE AXBT DATA
0067          LDA SDINFF      ;SETS IN FF ON AXBT INTERFACE
0068          EQU $          ;ROUTINE TO WAIT FOR INTERRUPT
0069          LDA CDINFF      ;LOAD REG A WITH FLAG
0070          ANI INSFF       ;(CHECK IF IN FF
0071          SUI INSFF       ; HAS BEEN RESET)
0072          JP WAITD        ; IF RESET HAS NOT OCCURRED WAIT
0073          LDA INDVMD      ; INTERRUPT HAS OCCURRED GET AXBT DATA
0074          STA DATAX       ; SAVE AXBT BINARY VALUE, WILL USE TO OBTAIN LSB
0075          ANI MASK3       ; DELETE LSB FROM AXBT WORD BEFORE ASCII CONVERSION
0076          RRC             ; ROTATE AXBT WORD RIGHT 1 BIT
0077          PUSH D          ; SAVE REG D IN STACK, GOOD HOUSEKEEPING
0078          PUSH B          ; SAVE REG B IN STACK, GOOD HOUSEKEEPING
0079          CALL B2ASC      ; CALL ROUTINE TO CONVERT AXBT BIN WORD TO ASCII
0080          MOV A,B         ; MOVE MS AXBT ASCII CHAR TO REG A
0081          STA ASC1        ; SAVE IT TEMPORARILY
0082          MOV A,C         ; MOVE SECOND AXBT ASCII CHAR TO REG A
0083          STA ASC2        ; SAVE IT TEMPORARILY
0084          MOV A,D         ; MOVE LS AXBT ASCII CHAR TO REG A
0085          STA ASC3        ; SAVE IT TEMPORARILY
0086          POP B          ; RESTORE REG B FROM STACK
0087          POP D          ; RESTORE REG D FROM STACK
0088          LDA ASC1        ; LOAD MS AXBT ASCII CHAR TO REG A
0089          STAX D          ; STORE CHAR IN DATA LOCATION
0090          INX D          ; INCREMENT DATA LOCATION
0091          LDA ASC2        ; LOAD A WITH SECOND AXBT ASCII CHAR
0092          STAX D          ; STORE CHAR IN DATA LOCATION
0093          INX D          ; INCREMENT DATA LOCATION
0094          LDA ASC3        ; LOAD A WITH LS AXBT ASCII CHAR
0095          STAX D          ; STORE IT IN DATA LOCATION
0096          INX D          ; INCREMENT DATA STORAGE LOCATION
0097          LDA DATAX       ; BINARY AXBT VALUE
0098          ANI MASK4       ; MASK OFF LS BIT FOR .5 DEG VALUE
0099          SUI MASK4       ; SUBTRACT BIT FOR TEST OF .5 DEG
0100          JM PUTAZ       ; IF .5 DEG BIT NOT THERE, JUMP TO PUT A ZERO
0101          MVI A,35H      ; MOVE ASCII FIVE TO REG A
0102          STAX D          ; STORE IT IN DATA LOCATION
0103          INX D          ; INCREMENT DATA LOCATION
0104          JMP GETCLK     ; JUMP TO GET TIME
0105 PUTAZ    EQU $          ; PLACE AN ASCII ZERO IN DATA LOC
0106          MVI A,30H      ; MOVE ASCII ZERO TO REG A
0107          STAX D          ; STORE ZERO IN DATA LOCATION
0108          INX D          ; INCREMENT DATA LOCATION
0109 GETCLK   EQU $          ; GET SECONDS OF TIME FROM CLOCK
0110          LDA SCINFF      ; SET IN FF ON CLOCK INTERFACE
0111          LDA RCINFF      ; RESET IN FF ON CLOCK INTERFACE
0112          LDA INCLKD      ; INPUT THE CLOCK DATA
0113          CMA             ; COMPLEMENT THE CLOCK DATA
0114          ANI MASK2       ; ELIMINATE MINUTES BIT ON CLOCK DATA
0115          PUSH B          ; SAVE B REG IN STACK, GOOD HOUSEKEEPING
0116          CALL BCD2AS    ; CONVERT CLOCK SECONDS FROM BCD TO ASCII
0117          STAX D          ; SAVE THE MS ASCII SECONDS CHARACTER
0118          STAX D          ; SAVE THE MS ASCII SECONDS CHARACTER
0119          INX D          ; INCREMENT DATA LOCATION
0120          MOV A,B         ; MOVE THE LS ASCII CHAR SEC TO REG A
0121          STAX D          ; SAVE THE LS ASCII SEC CHAR
0122          INX D          ; INCREMENT THE DATA LOCATION
0123          POP B          ; RESTORE REG B, GOOD HOUSEKEEPING
0124 ;ROUTINE TO OUTPUT 8 DATA CHARACTERS TO DISPLAY AND RT CTU
0125          LXI H,DATA01    ; LOAD IMMEDIATE ADDRESS TO REG H
0126          MVI A,B         ; LOAD REG A WITH 8(NUMBER OF CHARS)
0127          MVI D,6         ; LOAD REG D WITH 6(DISPLAY AND CTU CODE)
0128          CALL OUTERM     ; ROUTINE TO OUTPUT THE DATA

```

```

0129          RET                ;RETURN
0130  DATAX   DB    0            ;TEMPORARY STORAGE OF AXBT DATA
0131  ASC1    DB    0            ;TEMPORARY STORAGE OF FIRST AXBT CHAR
0132  ASC2    DB    0            ;TEMPORARY STORAGE OF SECOND AXBT CHAR
0133  ASC3    DB    0            ;TEMPORARY STORAGE OF THIRD AXBT CHAR
0134  DATA01 DS    8           ;RESERVE 8 WORDS OF DATA STORAGE
0135  ;TAPE#2 TAB VER 13 SINGLE CHANNEL DATA WITH INTERRUPT 4/24/79
0136  INIT1   EQU    $          ;INITIALIZATION ROUTINE
0137  INIT2   EQU    $          ;INITIALIZATION ROUTINE
0138          MVI    A,100       ;MOVE 100 TO REG A(100 TEN MILLSEC)
0139          STA    TIMER       ;STORE 100 IN LOC TIMER(EQUIVALENT TO 1 SEC)
0140          RET                ;RETURN
0141  COUNT4  DB    0H         ;TEMPORARY STORAGE
0142  ;MONITOR ROUTINE FOR TIMING DATA INPUT
0143  MONIT   EQU    $          ;
0144          LXI    H,TIMER     ;LOAD IMMEDIATE ADDRESS OF TIMER
0145          DCR    M           ;DECREMENT THE TIMER LOCATION
0146          RET                ;RETURN
0147  ;ROUTINE TO CHECK FOR TIMEOUT
0148  CHTIMO  EQU    $          ;
0149          LDA    TIMER       ;LOAD REG A WITH VALUE OF LOC TIMER
0150          ORA    A           ;
0151          JP    RETURN       ;JUMP TO RETURN IF NOT TIMED OUT
0152          JMP    CONTRL      ;TIME IS NEGATIVE,ACQUIRE DATA
0153  ;CONTROL ROUTINE TO GET AND PROCESS DATA
0154  CONTRL  EQU    $          ;
0155          CALL  INIT2        ;RESET TIMER
0156          CALL  INPUT        ;ROUTINE TO ACQUIRE THE DATA
0157          RET                ;RETURN
0158  ;BCD TO ASCII CONVERSION ROUTINE
0159  ; INPUT REG A=DATA
0160  ; OUTPUT REG A=MSBYTE,REG B=LSBYTE
0161  BCD2AS  EQU    $          ;
0162          STA    CNBYTE     ;STORE DATA BYTE IN TEMPORARY STORAGE
0163          RRC                ;(SHIFT DATA BYTE FOUR BITS TO
0164          RRC                ;(GET MS BCD CHARACTER
0165          RRC                ;(
0166          RRC                ;(
0167          ANI    MASK1       ;MASK OFF BCD CHARACTER
0168          ACI    30H         ;ADD 30 HEX TO CONVERT TO ASCII
0169          STA    ASMSB       ;STORE MS ASCII CHARACTER
0170          LDA    CNBYTE     ;LOAD REG A WITH DATA BYTE
0171          ANI    MASK1       ;MASK OFF LS BCD CHARACTER
0172          ACI    30H         ;ADD 30 HEX TO CONVERT TO ASCII
0173          MOV    B,A         ;MOVE LS CHAR TO REG B
0174          LDA    ASMSB      ;LOAD MS ASCII CHAR TO REG A
0175          RET                ;RETURN
0176  ASMSB   DB    0            ;TEMPORARY STORAGE MS CHAR
0177  CNBYTE  DB    0            ;TEMPORARY STORAGE DATA BYTE
0178  ;OUTERM  OUTPUTS A RECORD TO THE TERMINAL
0179  ;
0180  ;INPUT  REGISTER  H&L  ADDRESS OF FIRST BYTE
0181  ;          A      NO OF CHAR TO OUTPUT
0182  ;          D      OUTPUT DEVICE
0183  ;          0=LEFT CTU
0184  ;          2=RIGHT CTU
0185  ;          4=DISPLAY
0186  BUFR    DS    2           ;RESERVE 2 LOCATIONS TO SAVE BUFFER ADDRESS
0187  NOCHAR  DS    1           ;RESERVE 1 LOCATION FOR NUMBER OF CHARS
0188  OUTERM  EQU    $          ;OUTPUT ROUTINE
0189          STA    NOCHAR      ;SAVE THE NUMBER OF CHARS FROM REG A
0190          MOV    A,D         ;LOAD REG A WITH OUTPUT DEVICES CODE
0191          STA    OUTDEV      ;SAVE OUTPUT DEVICES CODE
0192          SHLD  BUFR         ;SAVE BUFFER ADDRESS
0193          CALL  GTIOB0       ;GET A SYSTEM BUFFER
0194          MVI    M,200H     ;CLAIM BUFFER WITH BIT

```

```

0195      PUSH   H           ;SAVE STATUS POINTER
0196      DCX   H           ;DECREMENT POINTER
0197      MVI   M,377Q      ;SET UP RECORD TRANSFER(-1)
0198      DCX   H           ;DECREMENT H&L
0199      LDA   NOCHAR      ;LOAD REG A WITH RECORD LENGTH
0200      MOV   M,A         ;SAVE RECORD LENGTH
0201      XCHG          ;SWAP H&L AND D&E
0202      CALL  GETPTR      ;GET BUFFER ADDRESS
0203      CALL  MOVDAT      ;MOVE DATA INTO BUFFER OBTAINED BY GTIOB0
0204      POP   D           ;RESTORE STATUS POINTER
0205      CALL  PUTIO      ;OUTPUT THE RECORD
0206      XCHG          ;SWAP H&L AND D&E REGISTERS
0207      MVI   M,0        ;RELEASE BUFFER
0208      RET                   ;RETURN
0209      MOVDAT EQU   $      ;ROUTINE TO MOVE DATA TO SYSTEM BUFFER
0210      PUSH  D           ;SAVE D&E IN STACK,GOOD HOUSEKEEPING
0211      LDA   NOCHAR      ;NUMBER OF CHARACTERS TO REG A
0212      MOV   D,A         ;SAVE NUMBER OF CHARS IN REG D
0213      PUSH  H           ;SAVE H&L REG IN STACK,GOOD HOUSEKEEPING
0214      LHLD  BUFER      ;LOAD H&L FROM STORAGE ADDRESS
0215      PUSH  H           ;(SWAP H&L AND B&C REGISTERS
0216      POP   B           ;(
0217      POP   H           ;LOAD H&L WITH BUFFER ADDRESS
0218      SAVMOR EQU   $
0219      LDAX  B           ;LOAD A WITH ASCII CHAR
0220      MOV   M,A         ;MOVE ASCII CHAR TO MEMORY
0221      INX   H           ;INCREMENT ASCII SYSTEM BUFFER
0222      INX   B           ;INCREMENT ASCII PROGRAM BUFFER
0223      DCR   D           ;DECREMENT NUMBER OF CHARACTERS
0224      JNZ  SAVMOR      ;JUMP UNTIL B CHARS TRANSFERRED
0225      POP   D           ;RESTORE REGISTER D&E
0226      RET                   ;RETURN
0227      ;SUBROUTINE TO CONVERT BIN TO ASCII
0228      ;ALGORITHM OBTAINED FROM "PRACTICAL MICROCOMPUTER PROGRAMMING",W.J.WELLER
0229      ;INPUT REG A
0230      ;OUTPUT REG B,C,D
0231      B2ASC EQU   $
0232      CNVB  MVI   B,'0'  ;ASCII ZERO TO REG B
0233      MOV   C,B         ;ASCII ZERO TO REG C
0234      CNVBA SUI   100    ;SUBTRACT 100
0235      JC   TENZ         ;SKIP OUT IF CARRY
0236      INR  B           ;INCREMENT 100'S DIGIT
0237      JMP  CNVBA       ;TRY AGAIN
0238      TENZ ADI   100    ;RESTORE NUMBER
0239      CNVBB SUI   10     ;SUBTRACT 10
0240      JC   UNITS       ;SKIP OUT IF CARRY
0241      INR  C           ;INCREMENTS 10 DIGIT
0242      JMP  CNVBB       ;TRY AGAIN
0243      UNITS ADI   10     ;RESTORE NUMBER
0244      ORI  '0'         ;MERGE ASCII CODE BITS WITH UNITS
0245      MOV  D,A         ;MOVE A TO D
0246      RET                   ;RETURN
0247      END

```

APPENDIX II

OPERATING INSTRUCTIONS FOR THE TERMINAL AXBT DATA ACQUISITION SYSTEM

1. Turn on power to the terminal, Time Code Generator and Switch Panel.
2. Insert cartridge tape marked Debugger/Assembler in left tape drive of terminal.
3. Press the key marked READ on the terminal. Wait for completion.
Explanation: The first record of the Debugger/Assembler tape will be displayed.
4. Press the key marked f2 on the terminal. Wait for completion.
Explanation: By pressing f2 the second record on the Debugger/Assembler tape will be loaded into the terminal memory. The message "OK >" will be displayed on the terminal.
5. Remove the Debugger/Assembler tape from left drive and insert the tape marked AXBT Version 13 Binary.
Explanation: This is the binary AXBT program to be loaded into terminal memory.
6. Type the characters "L" and "CR" (Carriage Return). Wait for completion.
Explanation: This sequence will load the binary program into terminal memory. The message "HP264X ASSEMBLER V2.0" will appear on the terminal display followed by an "OK> " prompt.
7. Place a blank cartridge in the right terminal drive.
Explanation: The data will be recorded on this tape cartridge. The cartridge should be unprotected by moving the protect lever to the left position. The tape cartridge should be labeled by hand. The recommended labeling is day of year and starting hour of tape.
8. Type "/9169" then "CR" on the terminal keyboard.
Explanation: An instruction in location 9169₍₁₆₎ must be modified so that control will be transferred from the terminal executive software to the AXBT program. An "87" will appear on the display.

9. Type "601A" then "CR" on the terminal.

Explanation: The starting location of the magnetics program is 601A(16). A "0" will appear on the display.

10. Type ":" (colon) on the terminal.

Explanation: The ":" will terminate the modification process. An "OK>" will appear on the terminal display.

11. Enter the AXBT number on the Switch Panel in BCD format in bits 7 through 4 and bits 3 through 0.

12. Set Bit 15 on the Switch Panel to the OFF position.

Explanation: Bit 15 in the OFF position inhibits data recording until useful information is being output by the AXBT recorder.

13. Press the RESET button on the terminal only once.

Explanation: Pressing the RESET button once forces a transfer in the terminal executive to the AXBT program. The program will start execution.

14. Wait until an AXBT is deployed from the aircraft and the AXBT recorder receives a valid signal. Turn Bit 15 to the "ON" position on the Switch Panel.

15. Record data for approximately three minutes and thirty seconds. At this time set the AXBT number on the switch panel to zero for approximately five seconds.

Explanation: Changing the AXBT number signals the end of an AXBT recording.

16. Set Bit 15 on the Switch Panel to the "OFF" position to inhibit data recording until the next AXBT is deployed.

17. Four AXBT's can be recorded on the same cassette tape. Replace the Right Cartridge tape if four AXBT's have been recorded. Go back to STEP 14.

APPENDIX III

LISTING OF AXBT PROCESSING PROGRAM

```

10 SET SHORT
20 ASSIGN "RIGHT TAPE" TO #1
30 ASSIGN "SH#1" TO #6
40 PRINT #6;"AXBT","TEMPERATURE","DEPTH"
50 PRINT #6;"NUMBER","(DEG C)","(METERS)"
60 PRINT #6;" "
70 PLOTR
80 GCLR
90 PRINT "ENTER XBT NUMBER"
100 INPUT Xbtno           !AXBT#
110 DIM Temp(400)
120 FOR I=1 TO 400
130 SET LONG
140 READ #1;A$           !GET DATA
150 SET SHORT
160 Xbtn=VAL(A$(1,2))    !GET XBT#
170 IF Xbtno-Xbtn=0 THEN 200 !CHECK IF XBT=#ENTERED
180 IF Xbtsw=1 THEN 390 !CHECK FOR END OF XBT DATA
190 GOTO 120             !NOT YET,FIND XBT DATA
200 Xbtsw=1             !FOUND THE DATA,SET SWTCH
210 Temp=VAL(A$(3,6))/10 !CALC TEMP
220 Time=VAL(A$(7,8))   !GET TIME
230 Lasttime=Time
240 Temp(I)=5/9*(Temp-32)
250 IF Tempedit=77 THEN 320
260 PRINT "TEMPERATURE=",Temp(I)
270 PRINT "TYPE 7 AND CR IF TEMPERATURE OK,OTHERWISE TYPE 0 AND CR"
280 Tempast=Temp(I)
290 INPUT Tempedit
300 IF Tempedit=0 THEN 140
310 Tempedit=77
320 IF Temp(I)-Tempast>3 THEN Temp(I)=Tempast
330 IF Temp(I)-Tempast<-3 THEN Temp(I)=Tempast
340 PRINT Xbtn,Time,Temp(I),(I-1)*5*.3048
350 PRINT #6;Xbtn,Temp(I),(I-1)*5*.3048
360 Tempast=Temp(I)
370 N=I
380 NEXT I
390 S(1)=0               !RESET SWTCH=0
400 PRINT "WHEN READY FOR GRAPH ENTER 9 AND CR"
410 INPUT V
420 PLOTR               !PLOT THE DATA
430 GCLR
440 LOCATE (20,180,20,90)
450 SCALE (0,400,5,25)
460 LGRID (-10,1,0,0,5,5,2)
470 LORG (5)
480 CSIZE (1,1,90)
490 LDIR (1.6)
500 MOVE (-40,15)
510 PRINT #0;"TEMPERATURE DEG C"
520 LDIR (0)
530 MOVE (200,3)
540 PRINT #0;"DEPTH IN METERS"
550 CSIZE (6,1,0)
560 MOVE (20,2)
570 PRINT #0;"AXBT#",Xbtno
580 PENUP
590 MOVE (0,Temp(1))
600 PENDN
610 FOR K=1 TO N
620 DRAW ((K-1)*5*.3048,Temp(K))
630 NEXT K
640 STOP

```

DATE
L MED
-8