NASA CONTRACTOR
REPORT

NASA CR-161663

(NASA-CR-161663) A STATISTICAL DATA ANALYSIS AND PLOTTING PROGRAM FOR CLOUD MICROPHYSICS EXPERIMENTS Final Contractor Report (ESPEE, Inc., Huntsville, Ala.) 145 p HC A07/MF A01 CSCL 04B G3/47

N81-20005

Unclas
41705

A STATISTICAL DATA ANALYSIS AND PLOTTING PROGRAM FOR CLOUD MICROPHYSICS EXPERIMENTS

By Archie J. Jordan
ESPEE, Inc.
500 Wynn Drive
Huntsville, Alabama 35805

Final Report

February 25, 1981

Prepared for

NASA - George C. Marshall Space Flight Center
Marshall Space Flight Center, Alabama 35812

PREFACE

This is the Final Report prepared by ESPEE, Inc., under Contract No. P.O. H-43036B (REEDA Cassette Integration) for the Space Sciences Laboratory of Marshall Space Flight Center. The NASA technical monitor for this contract is Dr. B. J. Anderson.

Prepared By:

Archie J. Jordan

TABLE OF CONTENTS

1.0 INTRODUCTION----------1
2.0 GENERAL----------2
3.0 PROGRAMS----------6
3.1 READM----------6
3.2 CFIGM----------13
3.3 PLOTM----------21
FIGURE 1----------32
FIGURE 2----------34
FIGURE 3----------35
FIGURE 4----------36
FIGURE 5----------37
FIGURE 6----------39
FIGURE 7----------40
FIGURE 8----------41
FIGURE 9----------42
FIGURE 10----------43
FIGURE 11----------44
FIGURE 12----------45
3.4 DUMPM----------46
4.0 ASSUMPTIONS AND LIMITATIONS----------50
5.0 CASSETTE DATA FORMAT----------54
APPENDIX A
APPENDIX B
APPENDIX C
APPENDIX D
APPENDIX E

1.0 INTRODUCTION

During the past several months ESPEE, INC. has been under contract to NASA's Space Sciences Laboratory to design and develop software capable of reading data from cassette tapes on a MEMODYNE model 3765-8BV cassette recorder and formatting and writing the data on a disk on the Space Sciences Lab's REEDA Hewlett Packard (HP)-1000 minicomputer system. In addition, software capability was to be designed, developed, and implemented which could read the aforementioned disk file and generate graphical displays or plots on three of REEDA's HP 1000 graphics devices. These devices are the HP 2647A display terminal, the HP 9872B 4-color pen plotter, and the HP 2608A line printer. This document will describe the hardware and software aspects of the problem, define the use of the programs currently implemented on the REEDA HP 1000 system, list assumptions and limitations of the software documented herein, and provide appendices showing program listings, sample plots, and sample output from some of the utility programs. The purpose of this document is to provide personnel of the Space Sciences Laboratory who will be using the software developed by ESPEE specific information on the utilization of the software.

2.0 GENERAL

There are essentially three (3) distinct steps to generating a plot or sequence of plots from a MEMODYNE cassette tape. They are:

(1) reading a tape on the MEMODYNE recorder and generating disk file(s) on the REEDA HP 1000 minicomputer, and

(2) generating or modifying a disk resident file containing sensor transfer function or engineering unit conversion data, and

(3) generating the plot or sequence of plots desired by interactively running the plot program.

There are, of course, multiple steps to each one of these 3 primary steps and much more to be said later concerning each step. The important thing to remember, however, is the general "sense" of what the user is trying to accomplish so the detail of doing it does not become overwhelming. Since there are three basic steps to get from a cassette tape to a finished plot, there are, not surprisingly, three programs to use to accomplish that purpose . Program READM is the name of the program that reads the cassette tape on the MEMODYNE cassette recorder and generates disk files on the HP 1000 system. Program CFIGM is the name of the program which generates, modifies, and lists the user defined transfer functions or engineering unit conversion scale factors to be used in generating engineering unit plots. Program PLOTM is the program which reads the disk files generated by program READM and produces plots on the HP 2647A display terminal, HP 9872B 4-color pen plotter, or HP 2608A line printer. Becoming familiar and proficient with these three programs is all a user needs to do in order to accomplish the task of converting cassette-

based data to a final, hard copy plot. However, it was recognized early on in the development and check-out phase of this effort that plots of the data would not always be the most desirable end product, so a utility program was developed and implemented which gives the user the ability to read the same disk files generated by READM and to list the data on the line printer in various formats. This program is named DUMPM and its use will be described later, but it will suffice here to say that DUMPM will read a disk file generated by READM and list the contents of that file as octal data (for debugging purposes should the need arise), as a statistical summary of the data (mean, variance, standard deviation, etc.), or as a block by block printout of the voltage for each channel recorded for each time point. In summary, then, there are four (4) programs implemented on the HP 1000 minicomputer system, and their names and functions are:

READM - reads a cassette tape on the MEMODYNE recorder and generates a disk file on the HP 1000 disk for each data set sequence on that cassette.

CFIGM - generates, modifies, and lists user-defined transfer function or engineering unit conversion scale factors used in producing engineering unit plots.

PLOTM - reads disk files generated by program READM and produces plots on the HP 2647A display terminal, HP 9872B 4-color pen plotter, or HP 2608A line printer.

DUMPM - reads disk files generated by program READM and lists data as octal data, generates a statistical summary, or dumps the entire file, record by record, channel by channel, as voltage data on the line printer.

These serve as introductory comments about the programs that have been implemented and checked out on the REEDA HP 1000 minicomputer system. Detailed information follows in the Programs section. There is also a section on general assumptions and limitations related to the programs themselves. It is pointed out here, however, that a basic overriding assumption of the software development process conducted by ESPEE, INC. and the documentation describing this software is that the user is familiar enough with the HP 1000 system and the RTE-IVB operating system that neither the software nor the documentation will attempt to cover every possible RTE action or reaction or interaction with the implemented software. That is, the software has been integrated and checked out on the REEDA HP 1000 minicomputer and works as advertised; however, it is possible (even probable) that an inexperienced user will error in using the software and will have to extract himself by using the appropriate RTE command or method. This document will not and cannot serve as an RTE guide. Similarly, a limitation of the software and documentation relates to future utilization of the software and documentation. The software runs currently on the REEDA HP 1000 system under RTE-IVB, revision 2013, and the software documentation relative to compiling and loading the programs is based on current REEDA hardware configuration constraints and revision 2013 software procedures. ESPEE, INC.

cannot guarantee "upward" compatibility with planned for REEDA hardware enhancements and future HP operating system revisions. At this time there are no known hardware or software revisions planned for the REEDA system which will cause obvious problems for the software; it's just a matter of making a point in this document that future hardware/software changes in the REEDA system can affect the software described in the following sections.

3.0 PROGRAMS

3.1 READM

The function of program READM is to read a cassette tape which has been inserted in a MEMODYNE 3765-8BV recorder and to generate a disk file on REEDA HP 1000 system disk logical unit 37 for each data set on the cassette tape. Program READM is structured to begin reading the cassette tape at the first data set on the tape and to continue reading until a change in sequence number on the cassette tape implies a change in data sets. At this point, READM will close the disk file generated for the previous data set and ask the user to set up to read the next data set. This process continues until all data sets on the cassette have been read and written to disk or until the user terminates the process by exiting program READM. A critical point to remember, then, is that program READM cannot locate a specific data set by sequence number on the cassette tape; that is, the user cannot ask READM to go to the fifth data set on the tape or to locate the data set with sequence number 27, for example. Data sets are read by READM from the beginning of the tape to the end of the last data set on the tape. The mechanism for READM knowing that the last data set on the tape has been read is provided by the test engineer who generates the tapes. It has been agreed upon that the NASA hardware/software responsible for generating the tapes will write a data set sequence number of zero(Ø) on the tape to signify end of data on the tape.

Program READM is comprised of a main program, program READM, and three subroutines, RTAPE, CTAPE, and WAIT. Program listings for &READM, &WAIT, #RTAPE, and #CTAPE are given in Appendix A. The main program serves as the control program, calling subroutine RTAPE to read an 84-byte buffer from the MEMODYNE recorder,

program WAIT to pause a specified period of time, and subroutine CTAPE to control the motion and positioning of the tape.

Subroutines RTAPE and CTAPE deserve special description because of their special nature. RTAPE is the program which communicates with the MEMODYNE recorder to read the data off the tape; CTAPE is the program which communicates with the recorder to issue control commands (load forward, backspace a record, rewind). Both are assembly language programs and both have a special status in the HP 1000 RTE environment - they are called "privileged" routines. The term "privileged" refers to the fact that RTAPE and CTAPE must refer to base page (page Ø) addresses in order to communicate with the MEMODYNE recorder. By RTE convention, the only programs allowed to do this are programs meeting very special RTE requirements and also having been included in a system generation as a module called a "driver", or, as in the implementation described here, simply turning the HP 1000 interrupt system off to address a base page address and turning it back on when base page addressing is complete. So RTAPE and CTAPE both turn the interrupt system off to communicate with the MEMODYNE recorder and turn it back on when thru. The advantage to this "privileged" approach is in time (drivers are extremely laborious to check out since each error correction or driver modification requires a new system generation) and expense (time is money!). The disadvantage of this approach is that it prevents other users from using the system while the interrupt system is disabled. Another disadvantage is that FORTRAN programs using "privileged" subroutines

to acquire data and manipulate it (reformat, write to disk, etc.) often find strange "un-FORTRAN" -like things happening to the data since FORTRAN programs (and FORTRAN programmers) are not used to the interrupt system being disabled during an I/O transfer. Program READM exhibited some of these timing or synchronization problems until subroutine WAIT was developed to give main program READM better timing control. Main program READM issues a call to subroutine WAIT to ensure that disk I/O is complete before issuing the next call to RTAPE to read another 84-byte record. A call of CALL WAIT (1,2) essentially tells RTE to suspend execution of program READM for one second at the point of call. Please see the listing of subroutine WAIT in Appendix A for more detail as it is commented and is basically self-explanatory.

The procedure for running program READM follows in a step-by-step narrative. Some of these steps can occur in a different order, but it is recommended that the user try and follow a pattern like the one outlined in order to minimize the chance of leaving out an important step. The steps to running READM, then, are:

1. Prepare the MEMODYNE recorder for operation with the HP 1000. Disconnect the RS-232 connector from the back of the data phone and connect it to the MEMODYNE cassette recorder. Check and ensure (look at the rear of the recorder) that the recorder is set up for 300 BAUD, 1 stop bit is selected, and that the control codes switch is in the select or up position.
2. Power on cassette recorder. Ensure cassette tape is write protected and place in recorder,

3. Load removable user disk cartridge labeled CLOUDPHYS in HP 7900 disk drive.
4. Log on the REEDA HP 1000. Strike any key on one of the display terminals not in use. RTE will respond with a request to log on:

   PLEASE LOG ON:

   User responds by typing in ANDERSON.CLOUDPHYS CR

   RTE responds with:

   PASSWORD:

   User responds by typing CR (carriage return)

   RTE will now log you on and put you in the File Manager environment. This means you now have the File Manager prompt, ':' (colon). From here on, : is the File Manager prompt in examples.
5. Disable the dial up terminal before you forget it! :CN,13,21B disables dial up terminal.
6. Load program READM.
   This step can be accomplished in any one of several ways, depending on the user's RTE experience and preference. The user can load READM by (1) issuing individual File Manager commands to re-compile (or assemble) all programs required by READM and issuing Loader commands to load READM, or (2) executing a transfer file named READM$ created to load READM automatically, or (3) by dumping a soft key file named HELP!! to LU 1 (if the user is logged on at the HP 2647A terminal), and subsequently selecting the soft key for loading READM.

File Manager Commands to Load READM

```
:RU,FTN4,&READM::37,1,%READM::37
:RU,FTN4,&WAIT::37,1,%WAIT::37
:RU,ASMB,#RTAPE::37,1,%RTAPE::37
:RU,ASMB,#CTAPE::37,1,%CTAPE::37
:RU,LOADR
    RE,%READM::37
    RE,%WAIT::37
    RE,%RTAPE::37
    RE,%CTAPE::37
    END
```

EXECUTING A TRANSFER FILE (READM$) TO LOAD READM

:TR,READM$ File Manager will transfer control to a disk file READM$ containing all of the above commands and they will be executed one at a time.

USING SOFT KEY FILE HELP!!

:DU,HELP!!,1 (works only on HP2647A terminal.)
Press soft key labeled 'RELOAD READM' and an automatic :TR, READM$ will execute.

7. Ensure cassette tape is at beginning of tape. BEOT light should be on. This is importart since READM will execute a load forward command to move the cassette tape off the leader to 1 inch past the BEOT hole.

8. Run program READM by typing :RU,READM<CR>
<CR>means carriage return. Program READM will respond by identifying itself with
READM---MEMODYNE CASSETTE RECORDER DATA REDUCTION
and will then prompt the user with
ENTER DISK FILE NAME:
The user must now enter a 1-6 character alphanumeric name with the first letter of the name being a letter of the alphabet. Once the name is entered, READM prompts the user with
ENTER NUMBER OF 84 BYTE RECORDS TO READ:
Since the user, in general, will not know the number of records (blocks) on tape for a given data set enter a large integer number. The largest integer number one can enter on the HP 1000 in response to this prompt is 32767. Remember, if you enter a number less than the actual number of records on tape for this data set, the next disk file generated by READM will contain part of the current data set. To be safe always enter a number much greater than what is reasonable for the current data set (with a max. being 32767). READM then responds with:
ENTER<CR> TO BEGIN READING TAPE:
Type carriage return when ready to read tape and READM will begin reading the tape. READM will read 84 byte records from the cassette tape until either the number of records input by the user is reached or until a change in data set sequence numbers is detected. As long as the data set sequence number is constant READM lets the user know everything is proceeding normally by printing:

BLOCK "N" READ. SEQUENCE NO. IS "NN".
for every record read and written to disk.
Once a data set sequence number change is detected, READM immediately stops reading the tape and prints the user a message
NEXT SEQUENCE NO. ON THIS TAPE = "NN"
READM then backspaces the cassette tape one record to position the data set to be read next back at block 1 of that data set. After cleaning up internally (closing disk file just generated) READM issues the next user prompt:
ENTER DISK FILE NAME:
and the process starts over again. The program is intended to be run until a sequence number of zero(Ø) is detected, meaning end of data on tape. However, if the user wishes for some reason to gracefully exit READM before end of data is encountered, it is done at this point by entering EX as the disk file name. Program READM will terminate upon reading EX as the file name. Program READM also terminates when a data set sequence number of zero (Ø) is encountered.

9. Enable the dial up terminal when thru by entering:
:CN,13,20B
This is an important step. Be sure and execute it as failure to do so will prevent remote users from acessing the REEDA system via the dial up terminal and make users of READM basically unpopular.
10. Disconnect the RS-232 connector from the rear of the MEMODYNE recorder and re-install at the back of the data phone.

Once program READM has been run and disk files have been written on disk (all disk files are stored on disk LU 37, the CLOUDPHYS removable disk), programs CFIGM, PLOTM, or DUMPM can be invoked to do their things. Program READM ahs been successfully used to generate disk files TEST01 and TEST02 currently residing on LU37 of the REEDA system by reading test cassette tapes generated by personnel of the Space Sciences Laboratory.

3.2 CFIGM

The function of program CFIGM is to allow the user to interactively create, modify, or list a disk resident file hereafter referred to as a "configuration file". This configuration file contains data used by the plotting or graphics program, PLOTM, which defines engineering unit scale factors for each sensor (channel) recorded for a given data set and other information that directs program PLOTM in the plotting and labeling of plots. The user has the option of creating/modifying/or listing a default configuration file (named "TABLEA") or entering a file name of his own to create/modify/or list. CFIGM will interactively prompt the user in a self-explanatory manner to enter the required data to accomplish the user selected function.

Program CFIGM is comprised of a main program, program CFIGM, and two subroutines, LABLE and CNFIG. The main program serves as a control program, prompting the user to enter the file name of the configuration file to be manipulated and simply calling subroutine CNFIG to interact with the user in file generation and maintenance activities. Subroutine LABLE allows "free field" input of hollerith or alphanumeric information and is used to decipher whether or not alphanumeric information input to CFIGM or CNFIG is actual hollerith

string data or carriage return data (signifying default or existing hollerith data). Subroutine CNFIG is the primary software module of program CNFIG and controls the user prompt/input process, writes new or modified configuration data to disk, closes disk files, and lists the selected configuration file. A description of the use of program CFIGM, then, is really a description of subroutine CNFIG. Program listings of &CFIGM and &CNFIG are given in Appendix A. A sample configuration file is given in Appendix C. A default configuration file (TABLEA) has been generated on the REEDA HP 1000 system and resides on LU37 when the CLOUDPHYS disk pack is loaded in the HP 7900 removable disk drive.

The procedure for running program CFIGM follows. The ":" (colon) will serve as the File Manager prompt in the following instructions on running CFIGM and<CR> will denote a user required carriage return.

1. Load user disk pack labeled CLOUDPHYS in the HP7900 disk drive (LU37).
2. Log on the REEDA HP1000.
   Strike any key on one of the display terminals not in use. RTE will respond with a request to log on:
   PLEASE LOG ON:
   user responds by typing<CR>.
   RTE will now log the user on and put the user in the File Manager environment. That is, a File Manager prompt, a ":" (colon) will appear on the display terminal signifying RTE is ready to accept valid File Manager commands.

3. Load program CFIGM.
   As in the case of loading program READM, CFIGM can be loaded by the user many ways (depending on users individual RTE experience) but the following three ways are, in general, recommended.

   File Manager Commands to Load CFIGM

```
:RU,FTN4,&CFIGM::37,1,%CFIGM::37
:RU,FTN4,&CNFIG::37,1,%CNFIG::37
:RU,LOADR
    RE,%CFIGM::37
    RE,%CNFIG::37
    END
```

   Executing A Transfer File (CFIGM$) to Load CFIGM

   :TR,CFIGM$ File Manager will transfer control to a disk file CFIGM$ containing all of the above commands and they will be executed one at a time.

   Using Soft Key File HELP!!

   :DU,HELP!!,1 (works only on the HP2647A terminal)
   Press soft key labeled "RELOAD CFIGM" and an automatic :TR,CFIGM$ will execute.

4. Run program CFIGM by typing
   :RU,CFIGM<CR>
   Program CFIGM will respond by typing
   ENTER FILENAME OF CONFIGURATION TABLE:

The user must now enter a 1-6 character alphanumeric name with the first letter being a letter of the alphabet. This filename is the disk file name of the configuration file the user wants to create, modify, or list. It can be an already existing disk file or a new file the user wants to generate and maintain. If the user types a<CR>in response to this prompt, the default configuration file name, TABLEA, is used. Once the user types in his response to this prompt, subroutine CNFIG is called and controls the rest of the user/program interaction. Subroutine CNFIG first welcomes the user to the file maintenance session with a friendly welcome.

WELCOME TO YOUR BASIC CONFIGURATION PROGRAM!!

The user is then prompted with:

ENTER CONFIGURATION OPTION DESIRED

1 GENERATE NEW CONFIGURATION FILE

2 CHANGE AN ENTRY

3 LIST CONFIGURATION FILE

4 EXIT CONFIGURATION PROGRAM

The user now enters a 1,2,3,or 4 at the keyboard signifying which option he wishes to start this session with. A<CR>yields a number 3 option, or list option as the default case. If 1 is entered the program attempts to create a file name on disk LU37 with the file name entered previously by the user. If a disk file name already exists on LU37 by that exact name, CFIGM will ask the user:

ARE YOU SURE YOU WANT TO DESTROY THE OLD CONFIGURATION FILE?

and name the file by file name.

If the user answers that question by entering a Y<CR> signifying "yes", the program will go ahead and lead the user through a series of data entry questions and generate a new disk file, essentially purging the previously existing disk file by that name. If the user enters a N<CR> signifying "no", the program will go back and prompt the user once again with the "enter configuration option" choices. If the answer is Y<CR>, or if the file name entered does not already exist on disk LU 37, or if the option chosen to begin with was 2 (CHANGE AN ENTRY) then program CFIGM will eventually prompt the user with:

PREPARE TO ENTER SENSOR DATA
TYPE <CR> WHEN READY TO PROCEED

This gives the user an opportunity to gather his thoughts (and his data) before proceeding to the basic mechanism for entering data into the configuration file, a series of questions from CFIGM. When <CR> is entered, CFIGM prompts the user with:

ENTER CHANNEL NO. (<0 OR >31 TO EXIT):

The user must now enter an integer number between 0 and 31 (to continue the process) corresponding to a channel number of interest for a given data set. That is, the number entered in response to this prompt corresponds to the channel number decoded and plotted by program PLOTM. Once this section of code has been entered the only programmed exit is by entering a "false" channel number, an integer less than 0 or greater than 31.

After this entry, CFIGM asks multiple questions which result in data entries which are stored directly into the configuration file. These questions are asked repeatedly for each channel selected by the user and are self-explanatory in nature. Instead of listing each question here, if suffices to list the data by "type" and the FORTRAN arrays the data are stored in order to define what the configuration file is all about. The data entry questions are concerned with prompting the user to input data for 1-32 channels (channels numbered channel 0-channel 31) which will define for each channel:

a). sensor name.
b). sensor linear gain.
c). sensor offset.
d). sensor engineering units.
e). sensor default y-axis minimum (engineering units).
f). sensor default y-axis maximum (engineering units).
g). a comment field for this sensor.

The sensor name for channel I is stored in FORTRAN array ITYPE (6,I+1) and is comprised of 1-12 alphanumeric characters. The calculated y-value in engineering units for channel I is given by:

Y = gain * voltage + offset

Thus, the linear gain which converts voltage for channel I to engineering units is stored in FORTRAN

array GAIN (I+1) where GAIN is a 32-element, floating point array. The offset to be added in the above equation to complete the engineering unit conversion for channel I is stored in FORTRAN array OFFSET (I+1), where OFFSET is a 32-element, floating point array. The engineering units for the sensor defined by channel I is stored in IUNITS (10,I+1) and is used for the y-axis label when engineering unit plots are selected in program PLOTM. This "units" label is comprised of 1-20 alphanumeric characters.

FORTRAN array YMINI (I+1) contains the Y-axis default minimum y-value to be used by program PLOTM when plotting channel I in engineering units. Similarly, FORTRAN array YMAXI (I+1) contains the Y-axis default maximum Y-value to be used by program PLOTM when plotting channel I in engineering units. Both default values can be overriden in program PLOTM at execution time.

Finally, a comment concerning the sensor recorded on Channel I can be included in the configuration file for inclusion on the generated plots. The comment for the sensor recorded in channel I is stored in FORTRAN array ICOMM (10,I+1) and can be comprised of 1-20 alphanumeric characters. The default comment field for all channels is "blank".

Data for these arrays, then, are entered for each channel selected by the user until the user has entered

all the data he desires and terminates the process by entering a channel number less than 0 or greater than 31. When the user does that, CFIGM then asks:
DO YOU WANT TO WRITE TO THE CONFIGURATION FILE?
(Y/[N]) :
An answer of Y<CR> (signifying "yes") results in the previously input data being written to the disk file, the disk file closed and the message:
CONFIGURATION PROCESS COMPLETE DISK FILE ____
WRITTEN AND CLOSED
where ____ identifies the disk file being manipulated by CFIGM. If the answer to this question is N<CR>(signifying "no"), the disk file is closed and subroutine CNFIG returns to CFIGM. Main program CFIGM signifies that the configuration process is finished by writing the message "CFIGM DONE!".

Once program CFIGM has either generated a new configuration file or updated an exsisting one (say, file TABLEA, for example) program PLOTM can be run to generate plots on files generated by READM. The user will find CFIGM easy to use and very self-explanatory in its user/program question and answer sessions. The only way to learn it, however, is to use it.

3.3 PLOTM

Program PLOTM is the program all previous documentation has been leading up to. The sole function of program READM is to read data sets off cassette and generate disk files for PLOTM to plot; similarly, the sole function of program CFIGM is to generate and maintain disk resident configuration files in order for PLOTM to produce engineering unit plots. Program PLOTM reads disk files produced by READM, uses scale factor and plotting information from configuration files produced by CFIGM, and generates plots on the REEDA HP 1000 system's (1) HP 2647A graphics terminal, (2) HP 9872B 4-color pen plotter, and (3) HP 2608A line printer. PLOTM interactively allows the user to choose:

1. Which HP plotting device he wishes to produce the plots on.
2. Voltage or engineering unit plots.
3. The start and end time of the plots.
4. Minimum and maximum Y-axis values, or
5. Auto scaling of the Y-axis.
6. Axes and grid types for the plots.
7. Starting channel number for the plots.
8. Single or multiple channel plotting.

Hard copy, document quality plots are generated on the HP 9872B 4-color pen plotter and HP 2608A line printer. The REEDA HP 1000 system's HP 2647A graphics terminal is not configured with a hard copy device (such as an HP BIG matrix printer/plotter or Tektronix 4631 hard copy unit) so the purpose of utilizing PLOTM with the graphics terminal is to provide "quick look" graphical capability as opposed to another hard copy alternative. Extensive examples of the plotting capabilites of PLOTM are provided under

Appendix D. Hard copy samples of all three plot types are included (2647A plots were produced on another government owned HP 1000 system where the graphics terminal is connected to a Tektronix 4631 hard copy unit). The disk files used for these examples are TEST01 and TEST02, two test case disk files generated by running READM on a test cassette generated by personnel of the Space Sciences Laboratory.

Program PLOTM is comprised of a large main program, program PLOTM, and six subroutines, LABLE, BOUND, GRAF, HCOPY, NGRAF, and a BLOCK DATA subroutine which defines labeled common blocks. PLOTM also refers extensively to numerous HP 1000 graphics library subroutines. These routines will not be documented here and the reader/user is referred to the HP 1000 graphics documentation (see REEDA system manager for help in obtaining this documentation) and the listings of &PLOTM in Appendix A for information on these subroutine calls and their functions. The main program, PLOTM, is the primary module for program PLOTM and serves as a control program, input/output manager, does the decoding of the X-Y data (time-voltage) on the selected disk file, computes the statistical values of the X-Y data, and produces all resultant plots. Subroutine BOUND is called to determine the Y-axis bounds (YMIN and YMAX) for a given channel if the user specifies the Y-axis is to be autoscaled. Subroutine LABLE accomplishes the same function previously described in the document (see section 3.2 on CFIGM), that is, it allows "free field" input of hollerith or alphanumeric strings entered by the user in response to prompts from PLOTM. Subroutines GRAF, HCOPY, and NGRAF are only called by PLOTM when HP 2647A graphics terminal plotting is selected. Subroutine GRAF initializes the graphic mode on a HP 2647A and

turns off the alphanumeric display. Subroutine HCOPY is actually used on the REEDA system to read (input) a<CR>(carriage return) typed by the user to signify he's through viewing the current plot on the terminal and wishes to erase the screen and see the next plot. In practice, 2 lines of code are left in subroutine HCOPY but have been commented out which, when the comment is removed, will allow the HP 2647A to send a hard copy request to a RS 170 composite video hard copy device if connected. Subroutine NGRAF terminates the graphic mode on a HP 2647A terminal and turns back on the alphanumeric display. The BLOCK DATA subroutine is an HP FORTRAN IV requirement which defines any and all FORTRAN labeled common blocks to be shared by routines of program PLOTM. Main program &PLOTM and the aforementioned subroutines are listed in Appendix A. Program PLOTM is heavily commented and should be easy to follow for an experienced FORTRAN programmer. In addition to the subroutines PLOTM calls, there is one other subroutine pertinent to PLOTM which PLOTM does not call. Subroutine DLTBL defines a "device linkage" table which is loaded with PLOTM at program load time and provides identification information for HP 1000 graphics programs referenced by PLOTM. Subroutine #DLTBL is also listed in Appendix A.

The procedure for running program PLOTM follows. It is assumed that programs READM and CFIGM have been run and appropriate disk files have been generated and stored on the REEDA HP 1000 disk subsystem. The steps to running PLOTM, then, are:

1. Load the user disk pack labeled CLOUDPHYS in the HP 7900 disk drive (LU 37).
2. Log on the REEDA HP 1000.
   Strike any key on one of the display terminals not in

use. RTE will respond with a request to log on:
PLEASE LOG ON:
User responds by typing in ANDERSON.CLOUDPHYS CR
RTE responds with:
PASSWORD:
User responds by typing <CR> (carriage return)
RTE will now log on and put the user in the File Manager environment, that is, a File Manager prompt, ":" (colon) will prompt the user at the user terminal. The user can now enter any legal file manager command.

3. Load program PLOTM.
As with the previously documented programs, READM and CFIGM, program PLOTM can be loaded in many ways but one of the following 3 loading approaches is recommended.

File Manager Commands to Load PLOTM

```
:RU,FTN4,&PLOTM::37,1,%PLOTM::37
:RU,FTN4,&WAIT::37,1,%WAIT::37
:RU,LOADR
    OP,LB
    RE,%PLOTM::37
    RE,%WAIT::37
    RE,%DLTBL::37
    SEA,%GPS40
    END
```

Exeucting a Transfer File (PLOTM$) to Load PLOTM

```
:TR,PLOTM$
```

File Manager will transfer control to a disk file, PLOTM$, containing the above commands and they will be executed one at a time.

Using Soft Key File HELP!!

:DU,HELP!!,1 Works only on the HP 2647A terminal. Press soft key labeled "RELOAD PLOTM" and an automatic :TR, PLOTM$ will execute.

4. Run program PLOTM by typing
:RU,PLOTM<CR>
program PLOTM will begin execution by identifying itself by:
PLOTM -HP 1000 GRAPHICS DATA REDUCTION PROGRAM
and will then prompt the user with
ENTER FILENAME OF DATA FILE TO PLOT:
The user responds by entering a 1-6 character alphanumeric name with the first letter of the name being a letter of the alphabet. If the user enters a 2-character name, "EX" or "/E" in response to this prompt, PLOTM will terminate. If the user enters a disk file name, PLOTM will attempt to open a file by that name, read the first (header) record and writes the header information on the terminal and asks the user:
IS THIS THE CORRECT FILE? ([Y] /N):
The "([Y]/N)" part of the above question tells the user the choices for answers to the question. Y implies "yes" and N implies "no". It is a convention of program PLOTM that questions with default answers will have those answers identified to the user by being included in brackets. Thus, [Y] above tells the user "yes" is the default answer in this case. "No" is signified as the answer by entering N CR . If the answer to this question is "no", program PLOTM terminates and writes the message:

PROGRAM PLOTM TERMINATED. CHECK FOR PROPER FILE NAME.
If the answer is "yes", PLOTM next asks the user:
SELECT PLOT TYPE. ([VO] , EN):
Where VO denotes voltage plot and is the default answer. EN denotes engineering unit plot and must be typed in as EN<CR>. PLOTM then prompts the user to enter another disk file name:
ENTER FILENAME FOR CONFIGURATION TABLE [TABLEA]:
where [TABLEA] is the default configuration file name and is selected by entering<CR>. If the user desires to use a different file than TABLEA, he must enter a 1-6 character alphanumeric name with the first letter of the name being a letter of the alphabet. Once this question is answered, PLOTM prompts the user with:
SELECT GRAPHICS LU. ([1],20,28):
The users answer to this prompt will determine the HP plotting device. [1] is the default answer and signifies the HP 2647A graphics terminal. This answer is only valid if the terminal the user is logged on is the HP 2647A graphics terminal. If the user selects 1 as the graphics LU and is not logged on the 2647A terminal, PLOTM will eventually "hang up" and the user will have to abort PLOTM "ungracefully" by entering RTE break mode and aborting PLOTM with some form of OFF command. LU 20 will route the generated plots to the HP 9872B 4-color pen plotter. The user is responsible for loading the paper in the plotter for the first and succeeding plots. LU 28 will route the generated plots to the HP 2608A line printer. Once this choice is

made and entered, PLOTM asks the user:
ENTER START TIME FOR PLOT (ELAPSED SECONDS):
The user now enters the desired start time for the plot in elapsed seconds from the start time of the data set. A<CR> results in a default start time corresponding to the start time of the selected data set. Once the start time has been entered, PLOTM prompts the user to:
ENTER STOP TIME FOR PLOT (ELAPSED SECONDS):
Similarly, the user enters the desired stop time for the plot in elapsed seconds. This time the elapsed seconds are from the start time of the plot, not necessarily the start time of the data set. A<CR> results in a default stop time corresponding to the stop time or final time of the data set. Once the stop time has been entered, a couple of questions concerning the Y-axis scale are put to the user. The first one is:
AUTOSCALE Y-AXIS? (Y/[N]):
The purpose of this option is to allow the user to choose between entering Y-axis minimum and maximum information (next question from PLOTM) or having PLOTM automatically search the Y data over the entire data set and determine the actual Y-minimum and Y-maximum values and use these values as Y-axis bounds when plotting. If the user wishes to autoscale for the current channel, enter a Y<CR> in response to this prompt. The default answer is N for "no". If "yes" is the choice the following question from PLOTM will be skipped; if "no" is the answer the question, then, is:

ENTER MIN AND MAX FOR Y-AXIS VALUES:
Since this question is only asked if the answer to the previous question about autoscaling is "no", it is assumed the user wants to use a standard Y-axis scale or enter a scale of interest for comparison to other plots. The user must enter two floating point numbers seperated by a comma as values to be "read" as YMIN, YMAX or enter a<CR>. A<CR> (carriage return) will result in default YMIN and YMAX values being utilized. For voltage plots (option VO), values of -5.0 and +5.0 are used as YMIN and YMAX. For engineering plots (option EN), YMIN and YMAX default values are those contained in the configuration table for the current channel being plotted. Finally, if the user does enter his own values of YMIN and YMAX, they are entered on the same line seperated by a comma and terminated by a carriage return. Example: ENTER MIN AND MAX FOR Y-AXIS VALUES: -10.,10.<CR>
Once YMIN and YMAX have been entered (or defaulted to), PLOTM prompts the user with:
SELECT AXES AND GRID TYPE
(AXES WITHOUT GRID-[0], AXES WITH GRID- 1):
The default answer is chosen by entering a<CR> and results in labeled axes with major and minor tick marks but without grid lines being drawn inside the axes. An entry of 1 results in the same labeled axes being plotted but grid lines are drawn vertically (X-axis) and horizontally (Y-axis) from the major tic marks to the opposite boundary of the plot frame. Now, PLOTM gets down to business with the next two questions or prompts. The next prompt is:
ENTER CHANNEL NO. TO PLOT:

The user now decides which channel he wishes to plot individually or which channel he wishes to begin a series of plots with. A<CR> signifies a default channel number of zero. Otherwise, the user must enter a legitimate channel number (between 0 and 31) and respond to the next and final question before the plotting begins:

PLOT ALL SUBSEQUENT CHANNELS WITHOUT OPERATOR INTERVENTION? ([Y]/N):

A <CR> results in a default answer of "yes" and plots are generated for each channel starting with the channel entered previously. An answer of N, signifying "no" results in one plot being produced for the channel previously input.

PLOTM now automatically performs the computational and logical steps required to generate plots for the chosen sequence of channels on the selected plotting device. If the HP 2647A graphics terminal was chosen, each plot drawn on the display terminal will remain there for user viewing until a <CR> is entered on the terminal by the user. A <CR> will result in the current plot being erased and the next channel plotted. If the 2608A line printer was chosen as the plotting device, the plots will be automatically generated and plotted on the 2608A line printer one channel at a time without operator intervention (see Section 4.0 for system limitations concerning 2608 line printer plots). If the HP 9872B 4-color pen plotter was selected as the plotting device, much more operator interaction is required. For each plot generated on the HP 9872B plotter it is necessary for the user to load the paper on the plotter (consult REEDA system manager for 9872B plotter documentation). When PLOTM is in the "consecutive

plot" mode, PLOTM completes a plot on the 9872B plotter, stores the pen in the proper holder, raises and moves the pen to the upper-right hand corner of the platten, and informs the user:
CHANGE PAPER ON HP9872B PLOTTER. ENTER<CR>TO CONTINUE.
Thus, the user has all the time he requires to remove the completed plot from the 9872B plotter and load the next paper for the next plot. Entering<CR>causes PLOTM to begin generating the next plot on the 9872B plotter.

Once a single plot has been completed (if a single channel plot was chosen) or the final plot of a sequence of plots is complete, PLOTM asks the user:
DO YOU WISH TO PLOT ANOTHER CHANNEL? ([Y]/N):
If the answer is N for "no", PLOTM terminates. If the answer is Y for "yes" or defaults to "yes" (<CR>), PLOTM asks:
SAME DISK FILE? ([Y]/N):
If the answer is N for "no", PLOTM goes back and issues the first prompt in its user interaction series of prompts and questions (ENTER FILENAME OF DATA FILE TO PLOT:) and begins the process anew for the selected disk file. If the answer is Y for "yes" or defaults to "yes" (<CR>), PLOTM rewinds the current disk file, reads the header record, and asks the reader to select the plot option (VO or EN) and begins the previously described input/output operations at that point. When the user eventually answers N ("no") to the "plot another channel" question, PLOTM terminates and writes the message:
PROGRAM PLOTM TERMINATED. HAVE A GOOD DAY!

These steps, then, describe how to set up to run the graphics program, PLOTM, and how to interact with PLOTM'S input/output questions. On the remaining pages of this section are sample

hard copy examples of PLOTM questions and answers for typical runs on sample test case TESTØ2. These samples, along with the step by step descriptions of this section, should provide the user with sufficient instruction on running PLOTM. The sample plots under separate cover are also helpful in outlining PLOTM'S capabilities and utility. As with any computer program, however, the only road to success is familiarity. A good approach to gaining experience with PLOTM is to run it on sample test files TESTØ1 and TESTØ2 and compare results with those contained in the document under separate cover.

The following few pages illustrate sample input to program PLOTM for several different plots generated using test file TESTØ2. These samples are intended to illustrate just a few of the options available to the user and how to enter them; for the purpose of this demonstration only LU 1 (2647A graphics terminal) is selected and an entire sequence of consecutive plots is not generated in order to conserve space in this document. The user is also reminded that for most questions or prompts a<CR>signifies default answers. Thus, in the examples, where no answer is entered for a PLOTM prompt the user can assume a<CR>was entered.

If the user has set up to run PLOTM and enters the run command
:RU,PLOTM<CR>
program PLOTM will begin execution by clearing the display screen at the user terminal and begin asking the user for input. A typical user response might look like: See Figure 1.)
Here, the user has selected default answers to each question where defaults are possible and the resultant plots for channels

```
      PLOTM - HP 1000 GRAPHICS DATA REDUCTION PROGRAM

ENTER FILENAME OF DATA FILE TO PLOT: TEST02
THE HEADER RECORD FOR THIS FILE SHOWS:
SEQUENCE NO. = 77
DATE         = 10/30/80
START TIME   = 13: 1: 2
STOP TIME    = 14:55:39
NO. CHANNELS = 30
IS THIS THE CORRECT FILE ?  ([Y]/N):
SELECT PLOT TYPE.  ([VO],EN):
ENTER FILENAME FOR CONFIGURATION TABLE.  [TABLEA]:
SELECT GRAPHICS LU.  ([1],20,38):
ENTER START TIME FOR PLOT (ELAPSED SECONDS):
ENTER STOP TIME FOR PLOTS (ELAPSED SECONDS):
AUTOSCALE Y-AXIS?  (Y/[N]):
ENTER MIN AND MAX FOR Y-AXIS VALUES:
SELECT AXES AND GRID TYPE
(AXES WITHOUT GRID-[0], AXES WITH GRID-1):
ENTER CHANNEL NO. TO PLOT: 1
PLOT ALL SUBSEQUENT CHANNELS WITHOUT  OPERATOR INTERVENTION ?  ([Y]/N):
```

FIGURE 1.

1 and 2 are shown on the following pages. Note that this response would have generated plots for channels 1-30 but only 2 plots are included here. (See Figure 2 and 3.)

So the user in this case would have produced plots for channels 1-30 which would have included the following characteristics:

a. Voltage plots.
b. Configuration TABLEA data not used but read in.
c. 2647A plots.
d. Default start of test, end of test start & stop times.
e. Default Y-axis minimum and maximum (-5.0,5.0).
f. Default axes & grid type (no grid lines).
g. No default for starting channel. Channel 1 selected.
h. Default answer for consecutive channel plotting & as indicated, channels 1-30 will be plotted.

(See Figure 3.)

The next example illustrates a voltage plot again but points out 3 different user entries. Here, the user selects a start time (600.0 elapsed seconds into the run) and a stop time (4200.0 elapsed seconds after the user selected start time), selects autoscaling and informs PLOTM to plot channel 1 only. The entries look like: (See Figure 4.)

The resultant plot looks like: (See Figure 5.)

Notice the start time and stop time labeled on the plot and compare it with the test start and stop times of the first example plot. Also notice the autoscaled Y-axis of this plot. Since only this plot is produced by the previous user entries, PLOTM is now interested in what the user wants to do next and

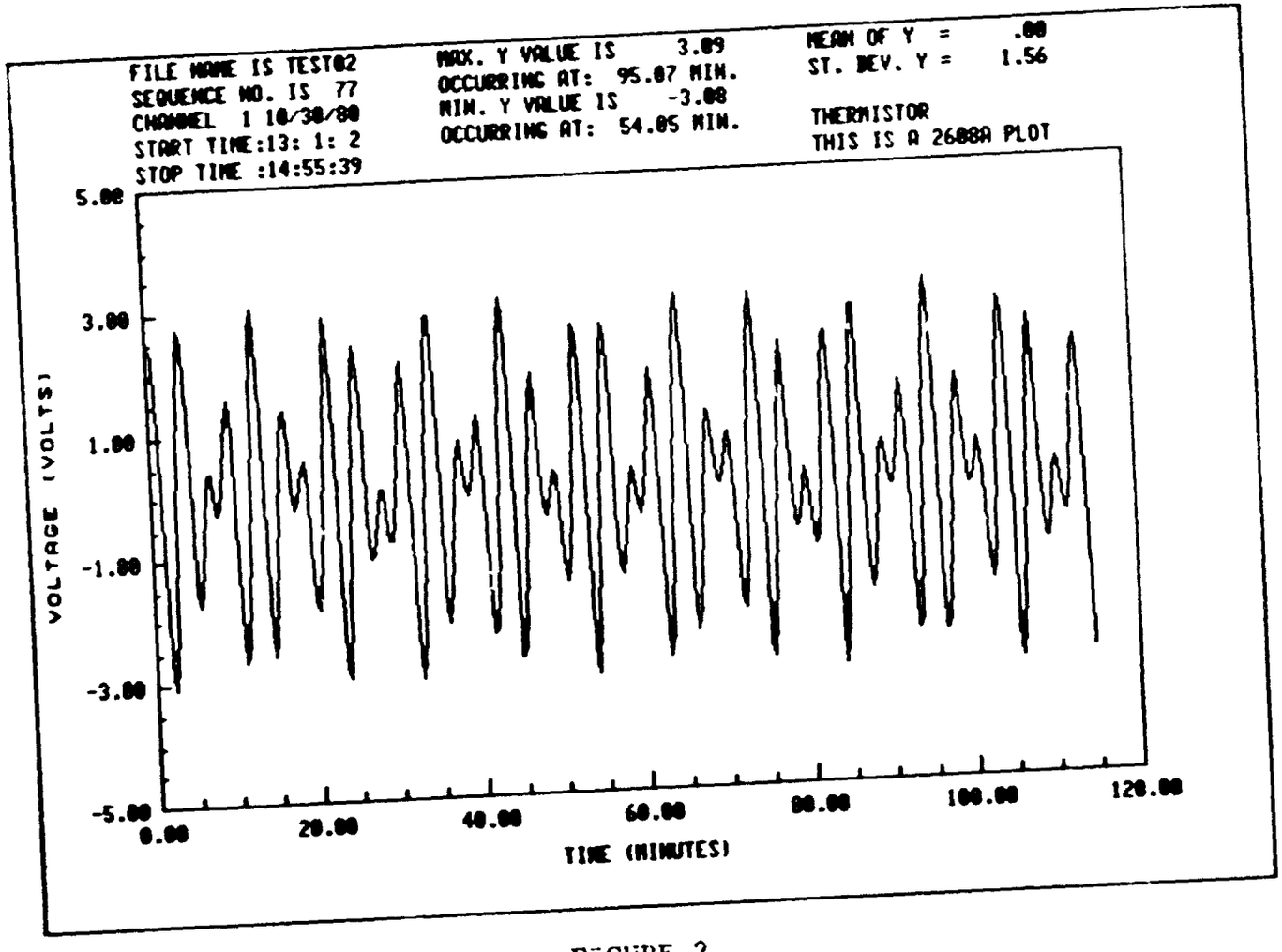

FILE NAME IS TEST02
SEQUENCE NO. IS 77
CHANNEL 1 10/30/80
START TIME:13: 1: 2
STOP TIME :14:55:39
MAX. Y VALUE IS 3.09
OCCURRING AT: 95.07 MIN.
MIN. Y VALUE IS -3.08
OCCURRING AT: 54.05 MIN.
MEAN OF Y = .00
ST. DEV. Y = 1.56
THERMISTOR
THIS IS A 2680A PLOT
5.00
3.00
1.00
-1.00
-3.00
-5.00
VOLTAGE (VOLTS)
0.00
20.00
40.00
60.00
80.00
100.00
120.00
TIME (MINUTES)

FIGURE 2.

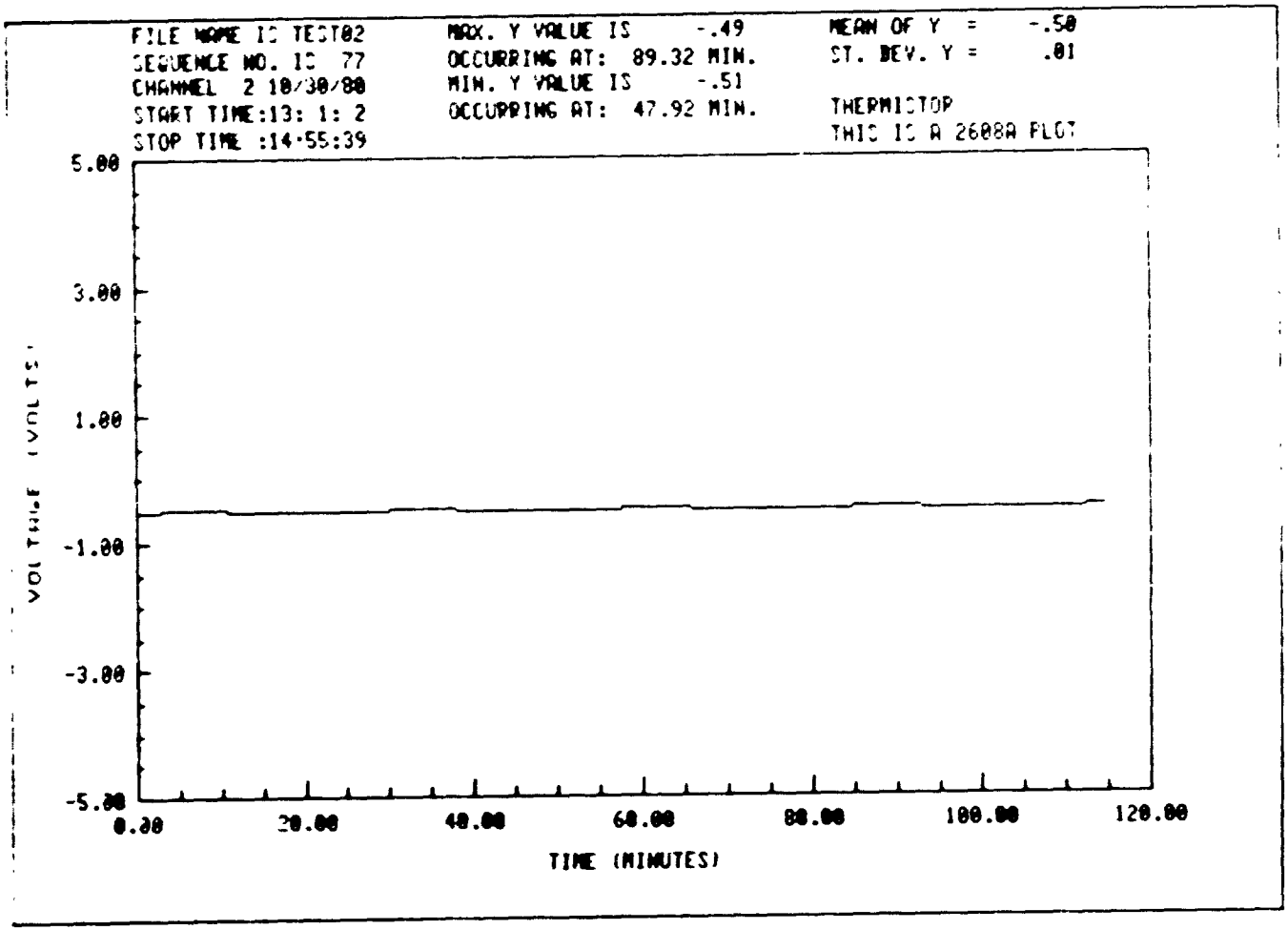

FILE NAME IS TEST02
SEQUENCE NO. IS 77
CHANNEL 2 10/30/80
START TIME:13: 1: 2
STOP TIME :14:55:39
MAX. Y VALUE IS -.49
OCCURRING AT: 89.32 MIN.
MIN. Y VALUE IS -.51
OCCURRING AT: 47.92 MIN.
MEAN OF Y = -.50
ST. DEV. Y = .01
THERMISTOR
THIS IS A 2608A PLOT
5.00
3.00
1.00
-1.00
-3.00
-5.00
0.00
20.00
40.00
60.00
80.00
100.00
120.00
TIME (MINUTES)

Figure 3.

```
      PLOTM - HP 1000 GRAPHICS DATA REDUCTION PROGRAM

ENTER FILENAME OF DATA FILE TO PLOT: TEST02
THE HEADER RECORD FOR THIS FILE SHOWS:
SEQUENCE NO. = 77
DATE         = 10/30/80
START TIME   = 13: 1: 2
STOP TIME    = 14:55:39
NO. CHANNELS = 30
IS THIS THE CORRECT FILE ?  ([Y]/N):
SELECT PLOT TYPE.  ([VO],EN):
ENTER FILENAME FOR CONFIGURATION TABLE.  [TABLEA]:
SELECT GRAPHICS LU.  ([1],20,38):
ENTER START TIME FOR PLOT (ELAPSED SECONDS): 600.0
ENTER STOP TIME FOR PLOTS (ELAPSED SECONDS): 4200.0
AUTOSCALE Y-AXIS?  (Y/[N]): Y
SELECT AXES AND GRID TYPE
(AXES WITHOUT GRID-[0], AXES WITH GRID-1):
ENTER CHANNEL NO. TO PLOT: 1
PLOT ALL SUBSEQUENT CHANNELS WITHOUT  OPERATOR INTERVENTION ?  ([Y]/N): N
```

FIGURE 4.

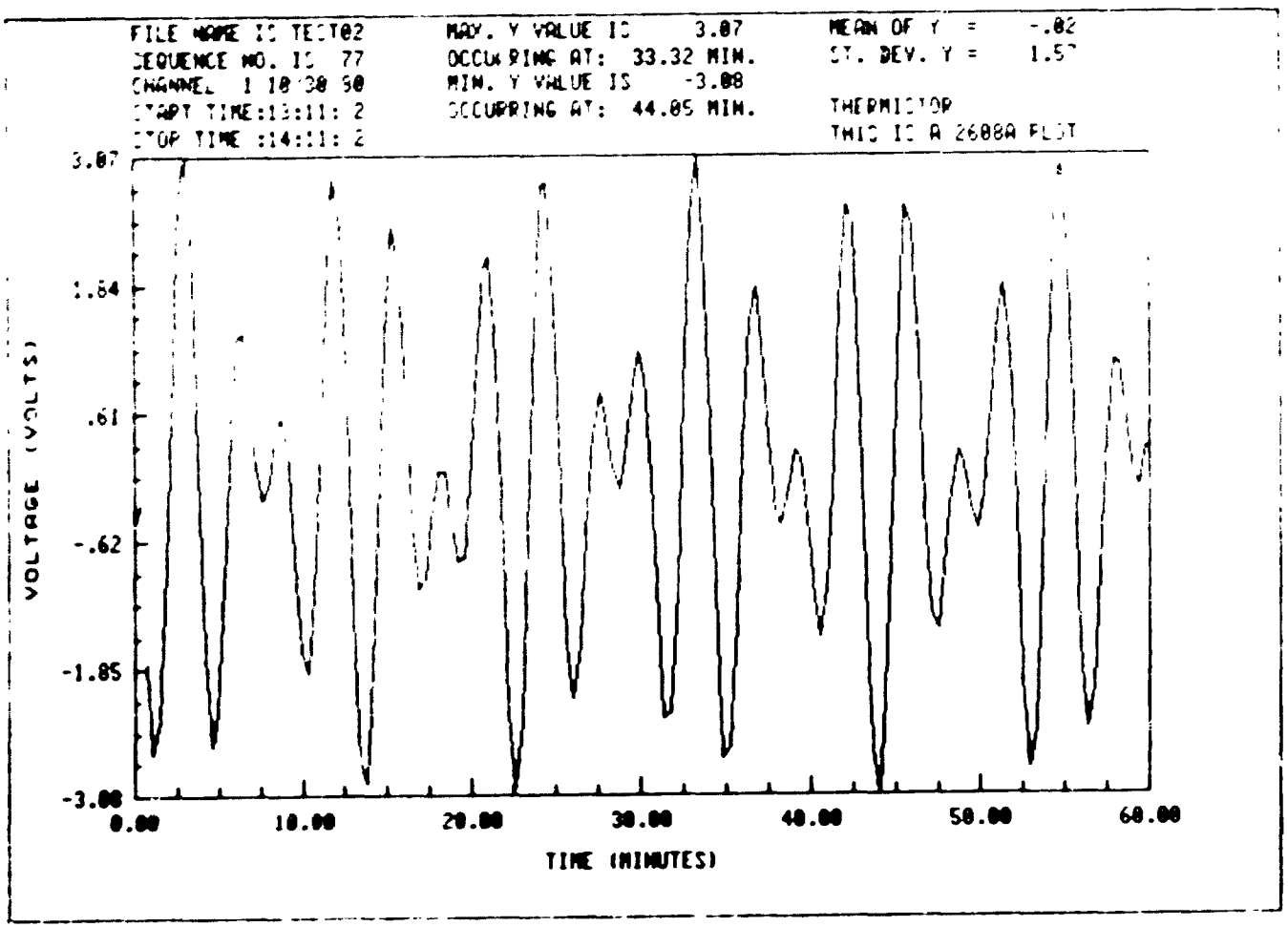

FILE NAME IS TEST02
SEQUENCE NO. IS 77
CHANNEL 1 10:30 50
START TIME:13:11: 2
STOP TIME :14:11: 2
MAX. Y VALUE IS 3.07
OCCURRING AT: 33.32 MIN.
MIN. Y VALUE IS -3.08
OCCURRING AT: 44.05 MIN.
MEAN OF Y = -.02
ST. DEV. Y = 1.57
THERMISTOR
THIS IS A 2608A PLOT
3.07
1.84
.61
-.62
-1.85
-3.08
VOLTAGE (VOLTS)
0.00
10.00
20.00
30.00
40.00
50.00
60.00
TIME (MINUTES)

FIGURE 5.

the following PLOTM/user interaction shows the user asking that another plot be generated using the same disk file, but switching to engineering unit plots (EN), only plotting the first 50 minutes of data (3000.0 elapsed seconds), autoscaling, and selecting the "grid lines" grid format. This discourse is given by. (See Figure 6.)

The resultant plot looks like: (See Figure 7.)

Again, notice start and stop time labeled on plot and the "engineering unit" autoscaled data. Also notice that the comment field for all the examples in this section says "This is a 2608A Plot". This just goes to show you that a computer program does what it is told to, not what you want it to. The comment array of configuration table TABLEA was not updated prior to running these sample plots.

The remaining two plots show channel 1 plotted in engineering units, one showing only 900.0 seconds of data plotted (notice change in X-axis label), the other 1200.0 seconds of data. Also, the first shows the user entering his own Y-axis data (-40.0, 40.0) and the second shows the user selecting default Y-axis data (which is read from configuration file TABLEA). These examples are produced by: (See Figures 8, 9, 10, and 11.)

Finally, the user enters an N ("no") when asked about plotting another channel and PLOTM terminates: (See Figure 12.)

```
DO YOU WISH TO PLOT ANOTHER CHANNEL?  ([Y]/N):
SAME DISK FILE?  ([Y]/N):
SELECT PLOT TYPE.  ([VO],EN): EN
ENTER FILENAME FOR CONFIGURATION TABLE.  [TABLEA]:
SELECT GRAPHICS LU.  ([1],20,38):
ENTER START TIME FOR PLOT (ELAPSED SECONDS):
ENTER STOP TIME FOR PLOTS (ELAPSED SECONDS): 3000.0
AUTOSCALE Y-AXIS?  (Y/[N]): Y
SELECT AXES AND GRID TYPE
(AXES WITHOUT GRID-[0], AXES WITH GRID-1): 1
ENTER CHANNEL NO. TO PLOT: 1
PLOT ALL SUBSEQUENT CHANNELS WITHOUT  OPERATOR INTERVENTION ?  ([Y]/N): N
```

FIGURE 6.

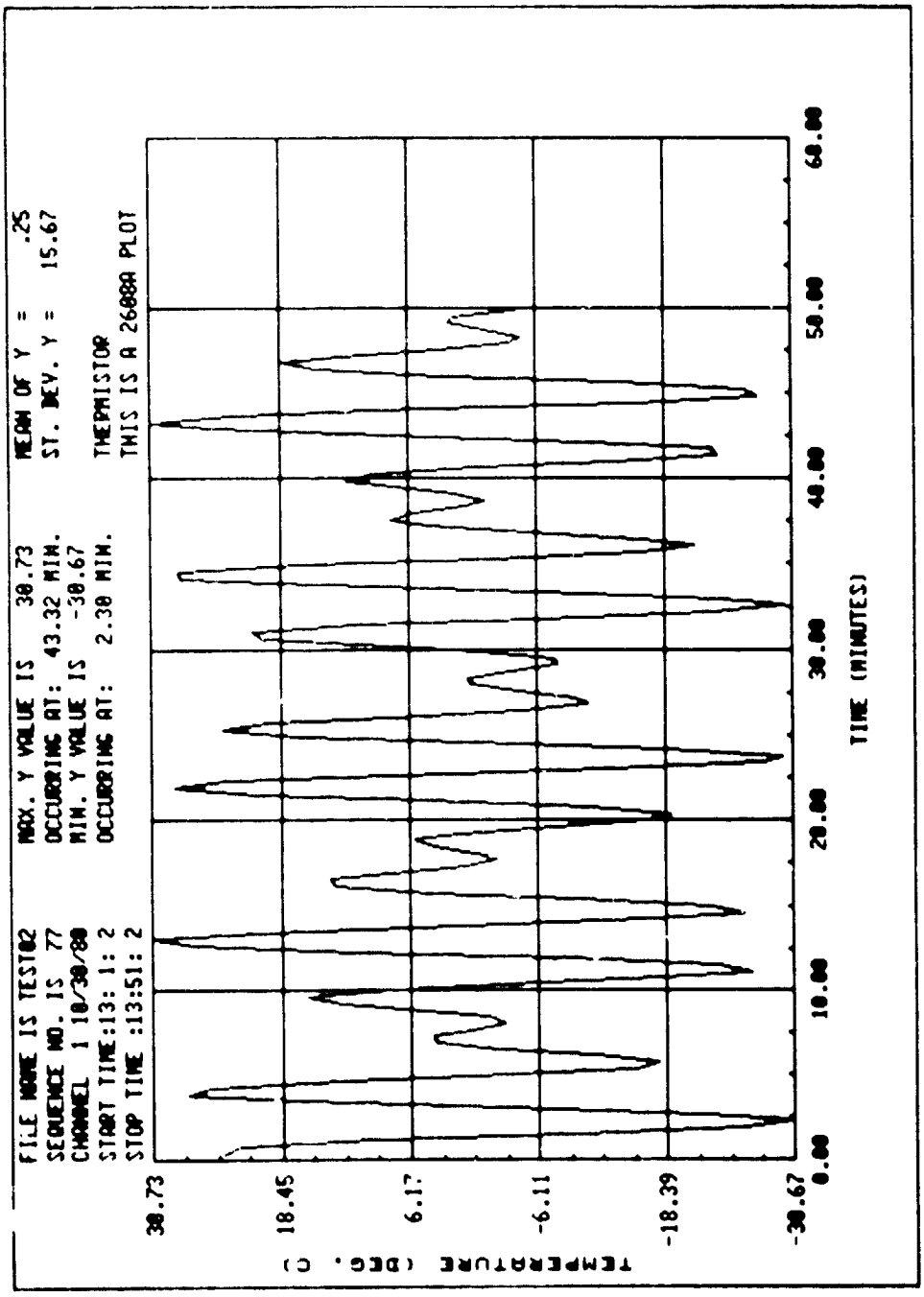

FILE NAME IS TEST02
SEQUENCE NO. IS 77
CHANNEL 1 10/30/80
START TIME:13: 1: 2
STOP TIME :13:51: 2
MAX. Y VALUE IS 30.73
OCCURRING AT: 43.32 MIN.
MIN. Y VALUE IS -30.67
OCCURRING AT: 2.30 MIN.
MEAN OF Y = .25
ST. DEV. Y = 15.67
THERMISTOR
THIS IS A 26000 PLOT
30.73
18.45
6.17
-6.11
-18.39
-30.67
TEMPERATURE (DEG. C)
0.00
10.00
20.00
30.00
40.00
50.00
60.00
TIME (MINUTES)

FIGURE 7.

```
      PLOTM - HP 1000 GRAPHICS DATA REDUCTION PROGRAM

ENTER FILENAME OF DATA FILE TO PLOT: TEST02
THE HEADER RECORD FOR THIS FILE SHOWS:
SEQUENCE NO. = 77
DATE         = 10/30/80
START TIME   = 13: 1: 2
STOP TIME    = 14:55:39
NO. CHANNELS = 30
IS THIS THE CORRECT FILE ?  ([Y]/N):
SELECT PLOT TYPE.  ([VD],EN): EN
ENTER FILENAME FOR CONFIGURATION TABLE.  [TABLEA]:
SELECT GRAPHICS LU.  ([1],20,38):
ENTER START TIME FOR PLOT (ELAPSED SECONDS):
ENTER STOP TIME FOR PLOTS (ELAPSED SECONDS): 900.0
AUTOSCALE Y-AXIS?  (Y/[N]):
ENTER MIN AND MAX FOR Y-AXIS VALUES: -40.,40.
SELECT AXES AND GRID TYPE
(AXES WITHOUT GRID-[0], AXES WITH GRID-1):
ENTER CHANNEL NO. TO PLOT: 1
PLOT ALL SUBSEQUENT CHANNELS WITHOUT  OPERATOR INTERVENTION ?  ([Y]/N): N
```

FIGURE 8.

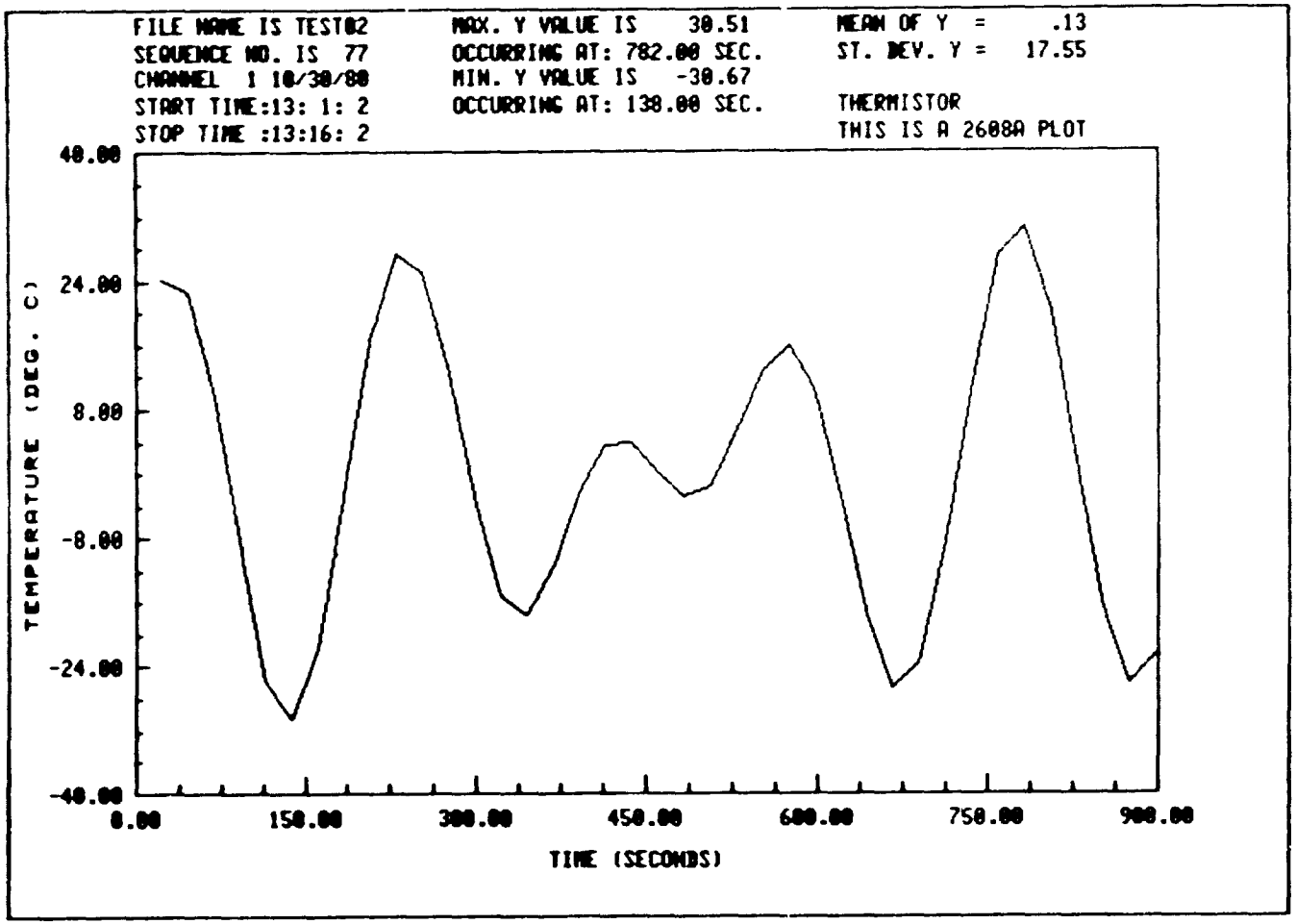

FILE NAME IS TEST02
SEQUENCE NO. IS 77
CHANNEL 1 10/30/88
START TIME:13: 1: 2
STOP TIME :13:16: 2
MAX. Y VALUE IS 30.51
OCCURRING AT: 782.00 SEC.
MIN. Y VALUE IS -30.67
OCCURRING AT: 138.00 SEC.
MEAN OF Y = .13
ST. DEV. Y = 17.55
THERMISTOR
THIS IS A 2608A PLOT
40.00
24.00
8.00
-8.00
-24.00
-40.00
TEMPERATURE (DEG. C)
0.00
150.00
300.00
450.00
600.00
750.00
900.00
TIME (SECONDS)

FIGURE 9.

```
DO YOU WISH TO PLOT ANOTHER CHANNEL?  ([Y]/N):
SAME DISK FILE?  ([Y]/N):
SELECT PLOT TYPE.  ([VO],EN): EN
ENTER FILENAME FOR CONFIGURATION TABLE.  [TABLEA]:
SELECT GRAPHICS LU.  ([1],20,38):
ENTER START TIME FOR PLOT (ELAPSED SECONDS):
ENTER STOP TIME FOR PLOTS (ELAPSED SECONDS): 1200.
AUTOSCALE Y-AXIS?  (Y/[N]):
ENTER MIN AND MAX FOR Y-AXIS VALUES:
SELECT AXES AND GRID TYPE
(AXES WITHOUT GRID-[0], AXES WITH GRID-1):
ENTER CHANNEL NO. TO PLOT: 1
PLOT ALL SUBSEQUENT CHANNELS WITHOUT  OPERATOR INTERVENTION ?  ([Y]/N): N
```

FIGURE 10.

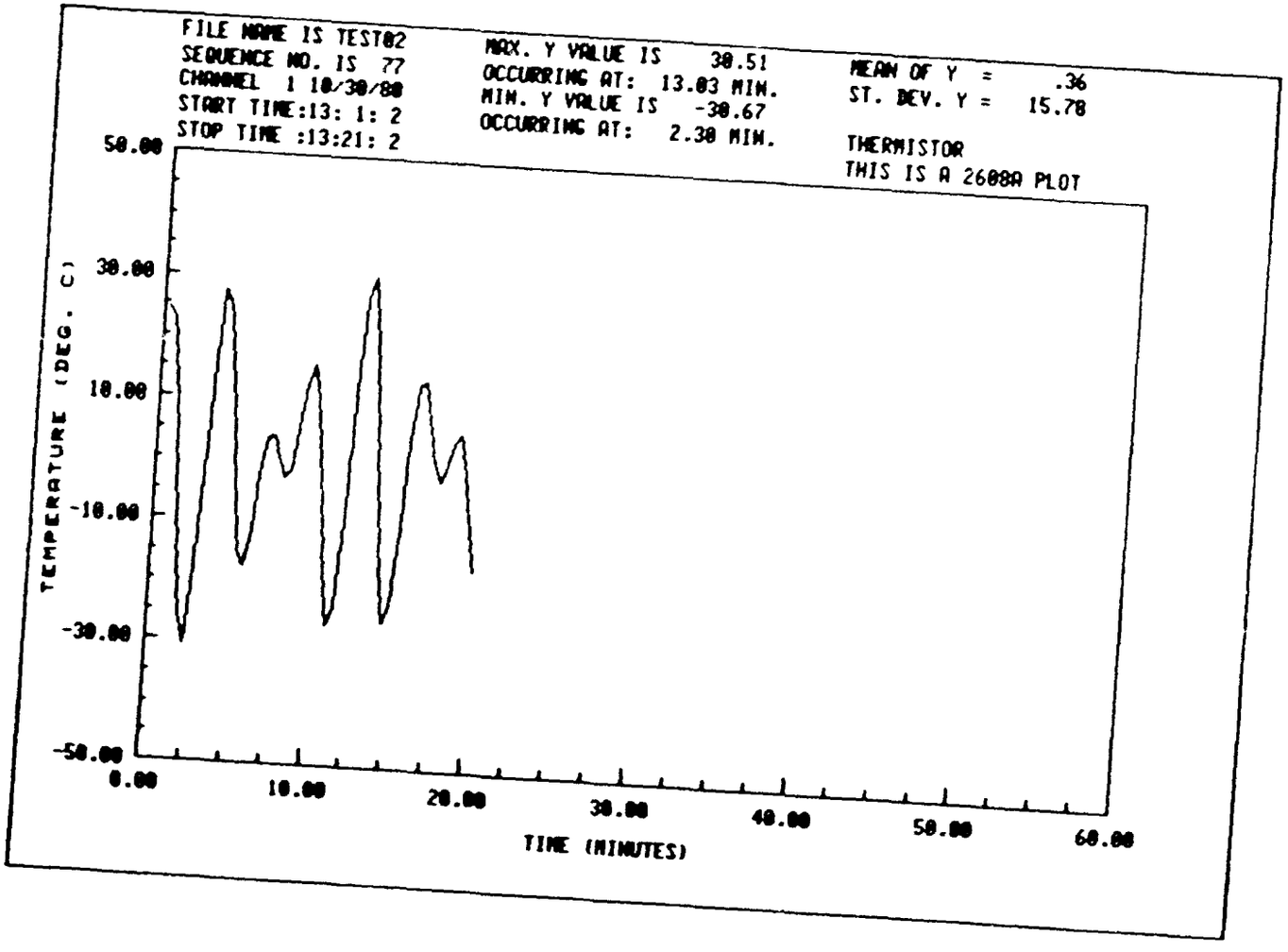

FILE NAME IS TEST02
SEQUENCE NO. IS 77
CHANNEL 1 10/30/88
START TIME:13: 1: 2
STOP TIME :13:21: 2
MAX. Y VALUE IS 30.51
OCCURRING AT: 13.03 MIN.
MIN. Y VALUE IS -30.67
OCCURRING AT: 2.30 MIN.
MEAN OF Y = .36
ST. DEV. Y = 15.78
THERMISTOR
THIS IS A 2608A PLOT
TEMPERATURE (DEG. C)
50.00
30.00
10.00
-10.00
-30.00
-50.00
0.00
10.00
20.00
30.00
40.00
50.00
60.00
TIME (MINUTES)

FIGURE 11.

```
DO YOU WISH TO PLOT ANOTHER CHANNEL?  ([Y]/N): N
PROGRAM PLOTX TERMINATED. HAVE A GOOD DAY!
:
```

FIGURE 12.

These examples of how to interact and enter data with PLOTM will give the user an idea of the flexibility he has at his disposal to preview the data at the HP 2647A graphics terminal and generate the plot or plots of a data set in a manner he wishes to see. A good way of learning to run PLOTM would be to run PLOTM on test file TEST02 on the REEDA HP 1000 and attempt to reproduce these examples.

3.4 DUMPM

Program DUMPM is a utility program used for listing disk files generated by program READM in a format selected by the user. The user selects the disk file to list (dump) and chooses one of three (3) formats:

1. Octal format.
2. Statistical format.
3. Total or block by block dump.

This utility is useful if plots produced by PLOTM are suspicious or in question for some reason and the user wishes to look at the data on disk in some type of "raw" form. The most elementary form to inspect the data in is to dump the data block by block in octal. It is perhaps not a convenient format since most of the data is 4-bit BCD data and not well represented in octal form for visual decoding but it does provide a user or programmer experienced with 16-bit word formats a chance to do some "quick look" visual debugging, if necessary. Most users will not use or require the use of this octal dump facility. The next most basic form to view the raw data in is a block by block dump of the disk file in converted "decimal" form, or a block by block

listing of voltage data for each channel listed by time, channel number, and voltage. This format is easier for the novice user to read but also voluminous in number of pages of hard copy output. Finally, a higher level dump or inspection of the data is a statistical summary in which the mean, variance, and standard deviation of the data is computed for each channel of a selected data file and is printed along with the minimum and maximum voltage values occuring in the data set along with the times they occurred. This is useful to get a quick "feel" for the statistical nature of the data without generating plots and can also be used to verify that the same statistical information is getting printed as title information on the plots.

Program DUMPM is comprised of a main control program, program DUMPM, and these subroutines, ODUMP, SDUMP, and CDUMP. The main program, DUMPM, reads in the name of the disk file to dump, the number of records to read and dump, and calls the proper subroutine. Subroutine ODUMP lists the data contained in the selected disk file in octal format, subroutine SDUMP computes and dumps statistical data about the data in the disk file, and subroutine CDUMP performs the block by block, channel by channel voltage dump.

The procedure for running program DUMPM follows:

1. Load the user disk pack labeled CLOUDPHYS in the HP 7900 disk drive (LU 37).
2. Log on the RTEDA HP 1000.
   Strike any key on one of the display terminals not in use. RTE will respond with a request to log on:
   PLEASE LOG ON:
   User responds by typing in ANDERSON.CLOUDPHYS<CR>
   RTE responds with:
   PASSWORD:

User responds by typing <CR> (carriage return). RTE will now log on and put the user in the File Manager environment, that is, a File Manager prompt, ":" (colon) will prompt the user at the user terminal. The user can now enter any legal file manager command.

3. Load program DUMPM.
   The following three (3) methods can be utilized to load program DUMPM. The user need select only one.

   File Manager Commands to Load DUMPM

```
:RU,FTN4,&DUMPM::37,1,%DUMPM::37
:RU,LOADR
    RE,%DUMPM::37
    END
```

   Executing a Transfer File (DUMP$) to Load DUMPM

```
:TR,DUMPM$   File Manager will transfer control to a
             desk file, DUMPM$, containing the above
             commands and they will be executed one at
             a time.
```

   Using Soft Key file HELP!!

```
:DU,HELP!!   Works only on the HP 2647A terminal.
             Press soft key labeled "RELOAD DUMPM"
             and an automatic :TR,DUMPM will execute.
```

4. Run program DUMPM by typing
   ;RU,DUMPM<CR>
   Program DUMPM will begin execution by clearing the terminal display screen and will prompt the user with:

ENTER FILENAME OF FILE TO DUMP:

The user responds by entering a 1-6 character alphanumeric name with the first letter of the name being a letter of the alphabet. Once done, DUMPM goes on to request:

ENTER NO. OF RECORDS TO DUMP:

User responds by entering an integer number greater than zero and less than 32767. DUMPM will list exactly the number of records input at this point unless an end of file on disk is reached. Next, DUMPM asks the user to select the format of the dump by requesting:

SELECT TYPE DUMP (1-OCTAL, [2-STATISTICS], 3-TOTAL):

An answer of <cr> (default) results in a statistical dump being performed by subroutine SDUMP. A user entry of 1 cr yields an octal dump by subroutine ODUMP, and, finally, a user entry of 3<cr> generates a block by block, channel by channel voltage dump by subroutine CDUMP.

All of these options assume LU 6 is the system line printer and all listings are routed to LU 6. NOTE: (Use 1 page (6 blocks) of octal dump, 1 page stat summar, and 3 pages (6 blocks) of voltage dump or Appendix E.)

4.0 ASSUMPTIONS AND LIMITATIONS

The folowing is a list of assumptions and/or limitations made by ESPEE in the design and development of the previously described software or which occurred in the implementation of the software on the REEDA HP 1000 minicomputer. Most of these are points already mentioned previously in this document; this section provides an easy reference to the basic software limitations in one place.

ASSUMPTIONS

(1) The reader/user is familiar with the REEDA HP 1000 minicomputer and the RTE-IVB operating system. This experience coupled with the instructions contained in this document on logging on the REEDA HP 1000 system, loading the ESPEE implemented software, and running the software are sufficient for the reader/user to successfully utilize the ESPEE software.

(2) The MEMODYNE cassette recorder software (program READM, subroutines RTAPE and CTAPE) has been made to work utilizing the current REEDA HP 1000 hardware, specifically, the MEMODYNE recorder is interfaced to the HP 1000 with an HP 12531D printed circuit assembly (PCA) interface card and is jumpered to operate at 300 BAUD. Furthermore, this interface card and cable are primarily utilized by a dial up terminal or modem and the operating instructions contained in this document for program READM assume a session LU of 13 for the dial up terminal. It is assumed the hardware (12531D interface at 300 BAUD) and the RTE software (LU assignment for dial up terminal) will not change. If it does, program READM will be affected and may not run.

(3) All source programs and files have been implemented on the HP 1000 and stored on the user disk pack labeled CLOUDPHYS. Furthermore, all the software written expects disk files generated or read to be resident on system LU 37 and the user pack CLOUDPHYS to be loaded in LU 37.

(4) Program READM assumes the user places the cassette in the recorder at the beginning of tape (BEOT light should be on). If not, user can count on erronerous results. Program READM assumes the user will read the data one data set at a time, creating a disk file on LU 37 for each data set. READM, finally, assumes that a data set sequence number of zero(Ø) will be present on the tape signifying end of data.

(5) Program PLOTM assumes the existence and maintenance of an HP 1000 graphics library named %GPS40 on the REEDA HP 1000 system. PLOTM uses its own device linkage table at program load time (%DLTBL on LU 37) but an underlying assumption is that the REEDA HP 1000 RTE system will always have the device subroutines and the device command tables loaded (as entries in %GPS40) for the plotting devices PLOTM uses;
HP 2647A graphics terminal - DVG01 & DCT01
HP 9872B 4-color pen plotter - DVG02 & DCT02
HP 2608A line printer - DVG04 & DCT04
The device linkage ID numbers for the above devices are defined in DLTBL to be: 1 for the HP 2647A graphics terminal, 2 for the HP 9872B 4-color pen plotter, and 3 for the HP 2608A line printer.

(6) Program DUMPM expects system line printer to be session LU 6.

(7) Program PLOTM expects (assumes) necessary free disk space on the first disk LU mounted in the system cartridge list (see :CL command in RTE documentation) when HP 2608A plots are being produced. Each plot produced for the HP 2608A line printer results in approximately 360 blocks of data being written to a disk file by HP 1000 graphics software. The user has no "programmatic" control over which disk LU this data is written to (always routed to first disk cargridge in cartridge list) so the user must be aware of how much free space (tracks,blocks,etc.) is on such a disk cartridge. If PLOTM runs out of disk space while running it does not gracefully or ungracefully abort; it will "hang up" and essentially run forever until the user recognizes something is wrong and aborts PLOTM manually. Consult REEDA system management or HP RTE documentation for classification on this point.

LIMITATIONS

(1) The MEMODYNE cassette recorder software (program READM with subroutines RTAPE and CTAPE) runs in the "priviledged" mode on the REEDA HP 1000 system. The interrupt system is alternately turned off and on in order to control and read the recorder and this destroys or invalidates RTE as a multi-user system while running program READM. At 300 BAUD, it takes approximately 1 hour to read 800 8¼-byte blocks on the MEMODYNE recorder.

(2) The cassette recorder software (program READM) cannot locate a specific data set on the cassette tape or be positioned under software control either forward or backward a specific number of blocks.

(3) Since program READM expects a data set sequence number of zero (Ø) on each cassette tape (signifying end of data), failure to generate that data at data acquistion time will result in the cassette tape "running away" when end of data is reached and the user must recognize it and manually halt the tape motion or the tape will run off the end of the reel.

(4) Program PLOTM requires a disk LU mounted first in the REEDA HP 1000 cartridge list with 200 free tracks if all 32 channels of data are to be plotted on the HP 2608A line printer.

(5) Program PLOTM is currently written to use pens 1 and 3 on the HP 9872B 4-color pen plotter. Failure to ensure pens are in those pen holders will result in erroneous plots. Pen 1 is used to draw the axes, label the axes, and write plot title information; pen 3 is used to draw the X-Y points. The user may use whatever color pen in holders 1 and 3 he wishes.

(6) HP 2647A graphics terminal plots can only be selected if logged on at that terminal. Attempts to choose that option at HP 2640 or other terminals will result in program PLOTM hanging up again.

(7) User help file HELP!! can only be run at the HP 2647A graphics terminal.

(8) If LU 6 (line printer) is down or off-line, program DUMPM will not run since it always attempts to list data on LU 6.

5.0 CASSETTE DATA FORMAT

The cassette tapes to be read on the MEMODYNE recorder and written to disk on the REEDA HP 1000 minicomputer consist of data blocks up to 86 bytes in length. These data are considered as 8-bit bytes with no parity. The format of each block (up to 86 bytes) is as follows:

BYTE 1: 2 BCD digits representing data set sequence number.
BYTE 2: 2 BCD digits representing day of month.
BYTE 3: 2 BCD digits representing month of year.
BYTE 4: 2 BCD digits year (last 2 digits)
BYTE 5: 2 BCD digits representing hours
BYTE 6: 2 BCD digits representing minutes
BYTE 7: 2 BCD digits representing seconds
} data acquisition time
BYTE 8: 2 BCD zeroes

Bytes 9 through 84 (possibly only byte 80) contain a BCD representation for the voltage sampled in the recorded channel along with the channel number. This data is contained along with the channel number. This data is contained in four(4) bytes for each channel recorded. Bytes 85 and 86 (possibly 81-84 as well) will always be BCD zeroes.

The 4 bytes of sensor data for each recorded channel are encoded on the cassette tape as follows:

Byte 12,16,20,11,...84	bit 0 bit 1 bit 2 bit 3 bit 4 bit 5 bit 6 bit 7	These 6 bits always Ø. Over-range bit; equals +1 if voltage +3.9999 volts. Sign bit; 0-voltage is negative, 1-voltage is positive.
Byte 11,15,19,...83	bit 0 bit 1 bit 2 bit 3 bit 4 bit 5 bit 6 bit 7	L.S. Digit; BIT 3 is MsBIT. 3rd Digit; Bit 7 is MsBIT.
Byte 10,14,18,...,82	bit 0 bit 1 bit 2 bit 3 bit 4 bit 5 bit 6 bit 7	2nd Digit; bit 3 is MsBIT. Most significant digit (full digit of voltage) Bit 7 is MsBIT.
Byte 9,13,17,...,81	bit 0 bit 1 bit 2 bit 3 bit 4 bit 5 bit 6 bit 7	Channel number; bit Ø is LsBIT. Most significant digit of voltage. Can be 0,1,2,or 3. Bit 7 is MsBIT. (Voltage decimal point after this digit).

If there are more channels of data than will fit in one block, the data will be continued in two consecutive blocks. The second block of a two block set will contain the same sequence number, date, and time as the first block. The second block will be zero filled beyond the last channel recorded. For one block sets, the block will be zero filled beyond the last channel recorded to the end of the block.

Programs PLOTM and DUMPM decode both the header bytes and the sensor data bytes in order to generate their respective plots and/or listings. The following statements illustrate the "decoding" of the header and sensor data blocks in FORTRAN (as accomplished in PLOTM and DUMPM). The 4 byte sensor data are contained in cassette "encoded" format in words IWORD1, IWORD2, IWORD3, and IWORD4, respectively. The seven (7) header bytes are contained in FORTRAN array IBUF(1) through IBUF(7), respectively. The data in FORTRAN by:

```
NSEQ = 10*IAND(IBUF(1),360B)/16+IAND(IBUF(1),17B)
IDAY = 10*IAND(IBUF(2),360B)/16+IAND(IBUF(2),17B)
IMON = 10*IAND(IBUF(3),360B)/16+IAND(IBUF(3),17B)
IYEAR = 10*IAND(IBUF(4),360B)/16+IAND(IBUF(4),17B)
IHR  = 10*IAND(IBUF(5),360B)/16+IAND(IBUF15),17B)
IMIN = 10*IAND(IBUF(6),360B/16+IAND(IBUF(6),17B)
ISEC = 10*IAND(IBUF(7),360B)/16+IAND(IBUF(7),17B)
```

and:

```
ICN = IAND (IWORD1,77B)    (channel number)

VOLTS = FLOAT(IAND(IWORD1,300B)/64)+
        FLOAT(IAND(IWORD2,360B)/16)*0.1 +
        FLOAT(IAND(IWORD2,17B)*0.01+
        FLOAT(IAND(IWORD3,360B)/16)*0.001+
        FLOAT(IAND(IWORD3,17B)*0.0001

ISIGN = IAND(IWORD4,200B)/128

RANGE = IAND(IWORD4,100B)/128

VOLTS = ISIGN*VOLTS

IF(RANGE .EQ. 1)VOLTS = ISIGN *(3.9999)
```

APPENDICES

APPENDIX A
PROGRAM LISTINGS

&READM

&WAIT

#RTAPE

#CTAPE

&CFIGM

&CNFIG

&PLOTM

#DLTBL

&DUMPM

```
&READM T=00003 IS ON CR00015 USING 00046 BLKS R=5713

0001  FTN4,L
0002
0003        PROGRAM READM
0004  C
0005  C
0006  C*****************************************************************************
0007  C                                                                            *
0008  C     THIS PROGRAM READS A CASSETTE TAPE ON A MEMODYNE MODEL 3765-8BV         *
0009  C     CASSETTE RECORDER AND WRITES A TYPE 3 (VARIABLE LENGTH RECORD)          *
0010  C     FILE TO DISK. THE PROGRAM USES PRIVILEDGED SUBROUTINE RTAPE TO         *
0011  C     READ THE TAPE. THE NUMBER OF CHARACTERS TO READ PER RECORD IS          *
0012  C     CONTROLLED BY VARIABLE NCHR. THE NUMBER OF RECORDS TO READ BEFORE      *
0013  C     TERMINATING IS CONTROLLED BY VARIABLE NREC. SINCE THE USER HAS NO       *
0014  C     FIRM KNOWLEDGE OF THE EXACT NUMBER OF RECORDS ON TAPE FOR A GIVEN      *
0015  C     DATA SET IT IS RECOMMENDED THAT A LARGE INTEGER NUMBER (LESS THAN      *
0016  C     32767) BE INPUT. READM WILL TERMINATE READING THE DATA SET WHEN A      *
0017  C     CHANGE IN SEQUENCE NUMBERS OCCUR (IF NREC HAS NOT BEEN REACHED).       *
0018  C     PRIVILEDGED SUBROUTINE CTAPE IS CALLED TO SPACE THE CASSETE TAPE       *
0019  C     FORWARD OFF THE BEOT HOLE AND TO BACK SPACE ONE RECORD AFTER A         *
0020  C     CHANGE IN SEQUENCE NUMBER OCCURS. SUBROUTINE WAIT IS UTILIZED TO       *
0021  C     ALLOW THE DISK WRITE OPERATION TO COMPLETE BEFORE RTAPE TURNS THE      *
0022  C     INTERRUPT SYSTEM OFF TO READ THE NEXT RECORD OF DATA. FUNNY THINGS     *
0023  C     HAPPEN IF WAIT IS NOT USED FOR THIS PURPOSE.                           *
0024  C                                                                            *
0025  C     NCHR IS THE NEGATIVE INTEGER VALUE OF THE NUMBER OF 8-BIT CHARACTERS   *
0026  C     TO BE READ.                                                            *
0027  C     IBUF IS THE ARRAY INTO WHICH THE DATA IS READ.                         *
0028  C     JBUF IS THE ARRAY WHICH CONTAINS THE MERGED OR FINAL FORM OF THE       *
0029  C     DATA.                                                                  *
0030  C                                                                            *
0031  C                                                                            *
0032  C     DEVELOPED BY.  ESFEE INC.                                              *
0033  C                    EXECUTIVE PLAZA                                         *
0034  C                    SUITE 305                                               *
0035  C                    205/837-8595                                            *
0036  C                                                                            *
0037  C                                                                            *
0038  C*****************************************************************************
0039  C
0040  C
0041        COMMON NCHR,IBUF(200)
0042        DIMENSION IDCB(144),ISIZE(2),NAME(3)
0043        DIMENSION JBUF(200)
0044        INTEGER HEAD(100)
0045  C
0046  C     GET LU OF USER CONSOLE.
0047  C
0048        CALL RMPAR(IBUF)
0049        LU = IBUF(1)
0050        IF(LU .LT. 1)LU = 1
0051        LP = 6
0052  C
0053  C     CALL CTAPE TO MOVE TAPE OFF BEGINNING OF TAPE.
0054  C
0055        NCHR = 13026B
0056        CALL CTAPE
0057        CALL WAIT(1,2)
0058  C
```

```
0059  C      GET DISK FILE NAME FOR STORED DATA.
0060  C
0061         WRITE(LU,100)
0062    100  FORMAT("ESHEJ")
0063         WRITE(LU,110)
0064    110  FORMAT(" READM --- MEMODYNE CASSETTE RECORDER DATA REDUCTION")
0065  C
0066    120  CONTINUE
0067         WRITE(LU,130)
0068    130  FORMAT(" ENTER DISK FILE NAME: _")
0069         READ(LU,140)(NAME(I),I=1,3)
0070    140  FORMAT(3A2)
0071  C
0072  C      IF DISK FILE NAME IS "EX" TERMINATE PROGRAM READM.
0073  C
0074         IF(NAME(1) .EQ. 2HEX)GO TO 450
0075  C
0076  C      CREATE DISK FILE AS TYPE 3 FILE.
0077  C      USE REST OF CARTRIDGE, TRUNCATE WHEN THRU.
0078  C
0079         ICN = 37
0080         ISIZE(1) = -1
0081         IDCBS = 128
0082         CALL CREAT(IDCB,IERR,NAME,ISIZE,3,0,ICN,IDCBS)
0083         IF(IERR .GE. 0)GO TO 160
0084         WRITE(LU,150)NAME,IERR
0085    150  FORMAT(" ERROR STOP. CREAT ERROR ON FILE ",3A2,". IERR = ",I5)
0086         CALL EXEC(6)
0087    160  CONTINUE
0088  C
0089  C      GET NUMBER OF RECORDS TO READ(NREC).
0090  C      TELL USER TO TYPE <CR> TO INITIATE READING TAPE.
0091  C
0092         WRITE(LU,170)
0093    170  FORMAT(" ENTER NUMBER OF 84 BYTE RECORDS TO READ: _")
0094         READ(LU,*)NREC
0095         WRITE(LU,180)
0096    180  FORMAT(" ENTER <CR> TO BEGIN READING TAPE: _")
0097         READ(LU,*)IK
0098  C
0099  C      INITIALIZE HEADER ARRAY.
0100  C
0101         DO 190 I = 1,100
0102    190  HEAD(I) = 0
0103  C
0104  C      WRITE HEADER OF ZEROES TO DISK. WILL REWIND DISK LATER &
0105  C      UPDATE ONCE TRUE HEADER DATA IS DETERMINED.
0106  C
0107         CALL WRITF(IDCB,IERR,HEAD,84)
0108         IF(IERR .GE. 0)GO TO 210
0109         WRITE(LU,200)NAME,IERR
0110    200  FORMAT(" ERROR STOP. WRITF ERROR ON FILE ",3A2,". IERR = ",I5)
0111         CALL EXEC(6)
0112    210  CONTINUE
0113  C
0114  C      READ NREC RECORDS OF NCHR LENGTH.
0115  C      STORE FINAL DATA IN JBUF & WRITE TO DISK FILE "NAME" .
0116  C
0117         MAXCN = 0
0118         DO 310 L = 1,NREC
```

```
0119          DO 220 J = 1,200
0120          JBUF(J) = 0
0121        220 IBUF(J) = 0
0122    C
0123    C
0124    C
0125          NCHR = -84
0126          CALL RTAPE
0127          DO 230 I = 1,84
0128        230 JBUF(I) = IBUF(I)
0129    C
0130    C     DECODE DATA SET DATA RESIDENT IN ARRAY JBUF.
0131    C     CONTINUE UNTIL A CHANGE IN SEQUENCE NUMBER IS DETERMINED.
0132    C
0133          NSEQ = 10*IAND(JBUF(1),360B)/16 + IAND(JBUF(1),17B)
0134          IF(L .EQ.1)JSEQ = NSEQ
0135    C
0136    C     SEQUENCE NUMBER CHANGE! GO CLOSE THIS DISK FILE.
0137    C
0138          IF(NSEQ .NE. JSEQ)GO TO 320
0139    C
0140          WRITE(LU,240)
0141        240 FORMAT("£H£J")
0142    C
0143    C     WRITE CURRENT BLOCK NUMBER & SEQUENCE NUMBER ON CRT SO USER
0144    C     CAN FOLLOW PROGRESS OF READM READING CASSETE.
0145    C
0146          WRITE(LU,250)L,NSEQ
0147        250 FORMAT(" BLOCK ",I4," READ. SEQUENCE NO. IS ",I3)
0148    C
0149    C     GET DATE & TIME INFO. FOR CURRENT BLOCK.
0150    C
0151          IDAY = 10*IAND(JBUF(2),360B)/16 + IAND(JBUF(2),17B)
0152          IMON = 10*IAND(JBUF(3),360B)/16 + IAND(JBUF(3),17B)
0153          IYEAR = 10*IAND(JBUF(4),360B)/16 + IAND(JBUF(4),17B)
0154          IHR = 10*IAND(JBUF(5),360B)/16 + IAND(JBUF(5),17B)
0155          IMIN = 10*IAND(JBUF(6),360B)/16 + IAND(JBUF(6),17B)
0156          ISEC = 10*IAND(JBUF(7),360B)/16 + IAND(JBUF(7),17B)
0157    C
0158    C     IF NOT FIRST BLOCK FIRST SEVEN HEADER RECORDS ALREADY FOUND.
0159    C
0160          IF(L .GT. 1)GO TO 260
0161    C
0162    C     FIRST BLOCK. GET 7 HEADER RECORDS FOR DISK FILE. THEY INCLUDE
0163    C     DATA SET SEQUENCE NUMBER, START DATE OF DATA SET AND START TIME
0164    C     (TIME OF FIRST DATA BLOCK) OF DATA SET. HEADER USED SUBSEQUENTLY
0165    C     BY PROGRAM PLOTM TO MINIMIZE OPERATOR INPUT AND CONTROL PURPOSES.
0166    C
0167          HEAD(1) = NSEQ
0168          HEAD(2) = IDAY
0169          HEAD(3) = IMON
0170          HEAD(4) = IYEAR
0171          HEAD(5) = IHR
0172          HEAD(6) = IMIN
0173          HEAD(7) = ISEC
0174    C
0175    C     HEADER RECORD WORDS 8-10 UPDATE CURRENT DATA SET BLOCK TIME.
0176    C     LAST BLOCK TIME WILL SERVE AS END OF TEST TIME.
0177    C
0178        260 CONTINUE
```

```
0179          HEAD(8) = IHR
0180          HEAD(9) = IMIN
0181          HEAD(10) = ISEC
0182    C
0183    C     FIND CURRENT CHANNEL NO. & UPDATE COUNTER (MAXCN) FOR MAXIMUM
0184    C     CHANNEL FOUND BY READM FOR THIS DATA SET. WILL BECOME FINAL
0185    C     HEADER RECORD WORD & IS USED BY PROGRAM PLOTM TO CONTROL NO.
0186    C     OF CHANNELS PLOTTED.
0187    C
0188          DO 270 KK = 9,77,4
0189          IWORD1 = JBUF(KK)
0190          ICN = IAND(IWORD1,77B)
0191          IF(ICN .GT. MAXCN)MAXCN = ICN
0192      270 CONTINUE
0193      280 CONTINUE
0194          HEAD(11) = MAXCN
0195    C
0196    C     WRITE DATA TO DISK (84 WORDS) FROM ARRAY JBUF.
0197    C
0198          CALL WRITF(IDCB,IERR,JBUF,84)
0199          IF(IERR. GE. 0)GO TO 300
0200          WRITE(LU,290)NAME,IERR
0201      290 FORMAT(" ERROR STOP. WRITF ERROR ON FILE ",3A2,". IERR = ",I5)
0202          CALL EXEC(6)
0203      300 CONTINUE
0204    C
0205    C     SUSPEND EXECUTION FOR ONE SECOND TO ALLOW DISK WRITE TO
0206    C     COMPLETE BEFORE TURNING OFF INTERRUPT SYSTEM IN RTAPE.
0207    C
0208          CALL WAIT(1,2)
0209    C
0210      310 CONTINUE
0211      320 CONTINUE
0212    C
0213    C     IF SEQUENCE NUMBER JUST READ IS ZERO (0) WARN USER END OF DATA
0214    C     FOUND ON THIS TAPE. NOT WISE TO ATTEMPT TO READ ANOTHER DATA
0215    C     SET ON THIS TAPE.
0216    C
0217          IF(NSEQ .EQ. 0)WRITE(LU,330)NSEQ
0218      330 FORMAT(" WARNING! NEXT SEQUENCE NO. = ",I3,". EOT REACHED!")
0219    C
0220    C     IF SEQUENCE NUMBER NOT ZERO THEN JUST INFORM USER OF NEXT SEQUENCE
0221    C     NUMBER ON TAPE & BACKSPACE TAPE TO BEGINNING OF DATA SET.
0222    C
0223          IF(NSEQ .NE. 0)WRITE(LU,340)NSEQ
0224      340 FORMAT(" NEXT SEQUENCE NO. ON THIS TAPE = ",I3)
0225          NCHR = 13427B
0226          CALL CTAPE
0227          CALL WAIT(1,2)
0228    C
0229    C     TRUNCATE DISK FILE ON CARTRIDGE ICN AND CLOSE FILE.
0230    C
0231          CALL LOCF(IDCB,IERR,IREC,IRB,IOFF,JSEC)
0232          IF(IERR .GE. 0)GO TO 360
0233          WRITE(LU,350)NAME,IERR
0234      350 FORMAT(" ERROR STOP. LOCF ERROR ON FILE ",3A2,". IERR = "I5)
0235          CALL EXEC(6)
0236      360 CONTINUE
0237          IT = JSEC/2-IRB-1
0238          CALL CLOSE(IDCB,IERR,IT)
```

```
0239          IF(IERR  GE  0)GO TO 380
0240          WRITE(LU,370)NAME,IERR
0241      370 FORMAT(" ERROR STOP. CLOSE ERROR ON FILE ",3A2,". IERR = ",I5)
0242          CALL EXEC(6)
0243      380 CONTINUE
0244    C
0245    C     OPEN DISK FILE FOR UPDATE AND WRITE HEADER DATA BACK INTO
0246    C     HEADER RECORD (RECORD ONE).
0247    C
0248          IOPT = 2
0249          CALL OPEN(IDCB,IERR,NAME,IOPT)
0250          IF(IERR .GE. 0)GO TO 400
0251          WRITE(LU,390)NAME,IERR
0252      390 FORMAT(" ERROR STOP. OPEN ERROR ON FILE ",3A2,". IERR = ",I5)
0253          CALL EXEC(6)
0254      400 CONTINUE
0255          CALL RWNDF(IDCB,JERR)
0256          IF(JERR .GE. 0)GO TO 420
0257          WRITE(LU,410)NAME,JERR
0258      410 FORMAT(" ERROR STOP. RWNDF ERROR ON FILE ",3A2,". JERR = ",I5)
0259          CALL EXEC(6)
0260      420 CONTINUE
0261          CALL WRITF(IDCB,IERR,HEAD,84)
0262          IF(IERR .GE. 0)GO TO 440
0263          WRITE(LU,430)NAME,IERR
0264      430 FORMAT(" ERROR STOP. WRITF ERROR ON FILE ",3A2,". IERR = ",I5)
0265          CALL EXEC(6)
0266      440 CONTINUE
0267    C
0268    C     CLOSE DISK FILE.
0269    C
0270          CALL CLOSE(IDCB)
0271    C
0272    C     GO BACK AND READ NEXT DISK FILE NAME
0273    C
0274          GO TO 120
0275    C
0276    C     TERMINATE READM.
0277    C
0278      450 CONTINUE
0279          CALL EXEC(6)
0280          END
```

```
&WAIT  T=00003 IS ON CR00015 USING 00005 BLKS R=5713

0001   FTN4,L
0002         SUBROUTINE WAIT(MULT,IRES,IERR)
0003   C
0004   C         THIS ROUTINE DUPLICATES THE FUNCTION OF THE WAIT SUBROUTINE
0005   C         FOUND IN THE ISA LIBRARY.  THE INPUT VARIABLES ARE DEFINED
0006   C         AS FOLLOWS:
0007   C              MULT - POSITIVE INTEGER INDICATING THE NUMBER OF UNITS
0008   C                     THE CALLING PROGRAM SHOULD BE PUT IN A WAIT STATE
0009   C              IRES - THE RESOLUTION OF MULT, I.E.
0010   C                           0 - 10'S OF MILLISECONDS
0011   C                           1 - MILLISECONDS
0012   C                           2 - SECONDS
0013   C                           3 - MINUTES
0014   C              IERR - RETURNED ERROR FLAG
0015   C                           1 - REQUEST ACCEPTED
0016   C                           3 - ILLEGAL PARAMETER
0017   C
0018         IF(MULT.GT.0 .AND. IRES.GE.0 .AND. IRES.LE.3)GO TO 4
0019         IERR = 3
0020         RETURN
0021       4 IERR = 1
0022         IM = MULT
0023         IF(IRES .EQ. 1)IM = IM/10
0024         IR = IRES
0025         IF(IRES .EQ. 0)IR = 1
0026         CALL EXEC(12,0,IR,0,-IM)
0027         RETURN
0028         END
```

```
#RTAPE T=00004 IS ON CR00015 USING 00018 BLKS R=5713

0001  ASMB,R,L,T
0002        NAM RTAPE,7   12531 PRIVILEDGED SUBROUTINE FOR MEMODYNE RECORDER
0003        ENT RTAPE
0004        EXT $LIBR,$LIBX
0005        COM NCHR IBUF(200)
0006  *
0007  *
0008  *     THIS IS A PRIVILEDGED SUBROUTINE FOR READING RS232 FORMATTED
0009  *     DATA FROM A MEMODYNE MODEL 3765-8BV CASSETTE RECORDER. ITS
0010  *     FUNCTION IS TO DISABLE THE INTERRUPT SYSTEM UPON ENTRY, READ
0011  *     NCHR 8-BIT BYTES FROM THE CASSETTE RECORDER, STORE THE DATA
0012  *     IN ARRAY IBUF, RESTORE THE INTERRUPT SYSTEM, AND RETURN TO
0013  *     THE CALLING PROGRAM.
0014  *
0015  *     NCHR IS A COUNTER CONTROLLING THE NUMBER OF BYTES TO BE READ.
0016  *     IBUF IS A 200 ELEMENT INTEGER ARRAY INTO WHICH THE DATA IS READ.
0017  *
0018  *     THE INTERRUPT SYSTEM IS DISABLED BY A CALL TO $LIBR.
0019  *     THE INTERRUPT SYSTEM IS RESTORED BY A CALL TO $LIBX.
0020  *
0021  *     A READ COMMAND TO THE 12531 INTERFACE CARD IS A 140000 OCTAL
0022  *
0023  *
0024  RTAPE NOP
0025        LDA  BUFZ     GET BUFZ ADDR
0026        STA  BUFF     PUT IT IN WORKING VARIABLE
0027        LDA  NCHR     GET LOOP COUNTER
0028        XAX           PUT IT IN X REG.
0029        CLA           CLEAR A REG
0030  *
0031        JSB  $LIBR    TURN OFF INTERRUPTS
0032        NOP
0033  *
0034        LDA =B133333  PREPARE TO SEND(WRITE) A CONTROL COMMAND TO RECORDER
0035        OTA SC        OUTPUT WRITE COMMAND TO 12531 INTERFACE CARD
0036        LDA =B010421  CONTROL COMMAND IS XON
0037        OTA SC        OUTPUT IT
0038        STC SC,C      START DEVICE
0039        SFS SC        IS OPERATION COMPLETE?
0040        JMP *-1       NO, KEEP CHECKING
0041  *
0042  READ  LDA  =B140000 LOAD COMMAND TO READ CASSETTE IN A-REG
0043        OTA  SC       OUTPUT COMMAND TO 12531 INTERFACE CARD
0044        STC  SC,C     START DEVICE OPERATION
0045        SFS  SC       IS IT THROUGH?
0046        JMP  *-1      NO, CHECK AGAIN
0047        LIA  SC       YES, INPUT IT
0048        AND  B377     MASK OFF UNWANTED BITS
0049        STA  BUFF,I   PUT IT IN BUFFER
0050        OTA  SS       OUTPUT DATA TO SWITCH REGISTER
0051        ISZ  BUFF     INCREMENT BUFFER POINTER
0052        NOP
0053        ISZ           INCREMENT LOOP COUNTER IF ZERO WE ARE THROUGH
0054        JMP  READ     NOT DONE
0055  *
0056        LDA =B133333  PREPARE TO SEND ANOTHER CONTROL CODE TO RECORDER
0057        OTA SC        SET UP WRITE COMMAND TO 12531 INTERFACE
0058        LDA =B011423  CONTROL COMMAND IS XOFF
```

```
0059          OTA SC          OUTPUT IT
0060          STC SC C        START DEVICE
0061    *
0062    EXIT  NOP
0063          ISZ  RTAPE      BUMP THE RETURN ADDRESS
0064          JSB  $LIBX      RESTORE INTERRUPTS
0065          DEF  RTAPE
0066    BUFF  NOP
0067    BUFZ  DEF  IBUF
0068    SC    EQU  26B
0069    SS    EQU  01B
0070    B377  OCT  377
0071    B200  OCT  20000
0072          END
```

```
#CTAPE T=00003 IS ON CR00015 USING 00006 BLKS R=5713

0001  ASMB,R,L,T
0002        NAM CTAPE,7   PRIVILEDGED MEMODYNE CASSETTE CONTROLLER SUBROUTINE
0003        ENT CTAPE
0004        EXT $LIBR,$LIBX
0005        COM IOUT,IBUF(200)
0006  *
0007  *
0008  *     THIS IS A PRIVILEDGED SUBROUTINE FOR ISSUING CONTROL CODES TO
0009  *     A MEMODYNE MODEL 3765-8BV CASSETTE RECORDER. ITS FUNCTION IS TO
0010  *     DISABLE THE INTERRUPT SYSTEM UPON ENTRY, ISSUE THE CONTROL
0011  *     COMMAND TO THE 12531 INTERFACE CARD, RESTORE THE INTERRUPT
0012  *     SYSTEM  AND RETURN TO THE CALLING PROGRAM.
0013  *
0014  *     IOUT IS THE VARIABLE CONTAINING THE CONTROL CODE TO BE ISSUED.
0015  *
0016  *     THE INTERRUPT SYSTEM IS DISABLED BY A CALL TO $LIBR.
0017  *     THE INTERRUPT SYSTEM IS RESTORED BY A CALL TO $LIBX.
0018  *
0019  *     A WRITE CARD TO THE 12531 INTERFACE CARD IS A 133333 OCTAL.
0020  *
0021  *
0022  *
0023  *
0024  CTAPE NOP
0025        JSB  $LIBR      TURN OFF INTERRUPTS
0026        NOP
0027        LDA  =B133333   LOAD WRITE COMMAND IN A-REG
0028        OTA  SC         OUTPUT COMMAND TO 12531 INTERFACE CARD
0029        LDA  IOUT       LOAD CONTROL COMMAND TO 12531 INTERFACE CARD
0030        OTA  SC         OUTPUT COMMAND TO 12531 INTERFACE CARD
0031        STC  SC,C       START DEVICE OPERATION
0032        ISZ  CTAPE      BUMP THE RETURN ADDRESS
0033        JSB  $LIBX      RESTORE INTERRUPTS
0034        DEF  CTAPE      RETURN TO CALLING PROGRAM
0035  *
0036  SC    EQU  2+B
0037  SS    EQU  1B
0038  *
0039        END
```

```
&CFIGM T=00003 IS ON CR00015 USING 00004 BLKS R=0000

0001  FTN4,L
0002        PROGRAM CFIGM
0003  C
0004  C
0005  C*****************************************************************************
0006  C                                                                            *
0007  C     CFIGM IS THE PROGRAM WHICH ALLOWS THE USER TO GENRATE, MODIFY,         *
0008  C     MAINTAIN, AND LIST ONE OR MORE DISK RESIDENT FILES WHICH SERVE AS      *
0009  C     TRANSFER FUNCTION OR ENGINEERING UNIT CONVERSION DATA FILES TO BE      *
0010  C     USED BY PROGRAM PLOTM WHEN GENERATING ENGINEERING UNIT PLOTS. CFIGM    *
0011  C     IS THE MAIN OR CONTROL PROGRAM FOR THIS FUNCTION AND SUBROUTINE        *
0012  C     CNFIG IS THE PROGRAM WHICH ACTUALLY PERFORMS THE CONFIGURATION FILE    *
0013  C     DATA MANAGEMENT FUNCTIONS. THE USER MAY SPECIFY AN EXISTING FILE       *
0014  C     FOR MODIFICATION OR UPDATE PURPOSES OR MAY SPECIFY A NEW FILE NAME     *
0015  C     AND INSTRUCT SUBROUTINE CNFIG TO GENERATE A NEW CONFIGURATION FILE.    *
0016  C     IT IS LEFT TO THE USER(S) TO MANAGE THE UTILIZATION OF MULTIPLE        *
0017  C     CONFIGURATION OR TRANSFER FUNCTION FILES. THE CONFIGURATION FILE(S)    *
0018  C     WILL RESIDE ON REEDA DISK LU 37.                                       *
0019  C                                                                            *
0020  C                                                                            *
0021  C     DEVELOPED BY:  ESPEE INC.                                              *
0022  C                    EXECUTIVE PLAZA                                         *
0023  C                    SUITE 305                                               *
0024  C                    205/837-8585                                            *
0025  C                                                                            *
0026  C                                                                            *
0027  C*****************************************************************************
0028  C
0029  C
0030        INTEGER ISIZE(2),LAB(10)
0031        COMMON/CBLOC/IDCB2(144),NAME2(3),ITYPE(6,32),GAIN(32),OFFSET(32),
0032       .             IUNITS(10,32),YMINI(32),YMAXI(32),ICOMM(10,32)
0033        COMMON/FLAGS/LU,IFLAG
0034  C
0035  C     GET LU OF USER CONSOLE.
0036  C
0037        CALL RMPAR(LAB)
0038        LU = LAB(1)
0039        IF(LU .LT. 1)LU=1
0040  C
0041  C     MAKE DEFAULT CONFIGURATION FILE NAME TABLEAA.
0042  C
0043        NAME2(1) = 2HTA
0044        NAME2(2) = 2HBL
0045        NAME2(3) = 2HEA
0046  C
0047  C     ENTER DISK FILE NAME OF USER CONFIGURATION FILE.
0048  C     <CR> WILL SELECT DEFAULT OF TABLEAA.
0049  C
0050        WRITE(LU,100)
0051    100 FORMAT(" ENTER FILENAME OF CONFIGURATION TABLE: _")
0052        READ(LU,110)(LAB(I),I=1,3)
0053    110 FORMAT(3A2)
0054        CALL LABLE(NAME2(1),LAB,3)
0055  C
0056  C     CALL SUBROUTINE CNFIG TO PERFORM FILE GENERATION & MAINTENANCE
0057  C     OPERATIONS.
0058  C
```

```
0059     120 CALL CNFIG
0060   C
0061   C      CFIGM COMPLETE. TERMINATE CFIGM.
0062   C
0063         WRITE(LU,130)
0064     130 FORMAT(" CFIGM DONE!")
0065         CALL EXEC(6)
0066         END
0067         BLOCK DATA
0068   C
0069   C      BLOCK DATA SUBROUTINE DEFINES COMMON BLOCKS /CBLOC/ AND /FLAGS/.
0070   C
0071         COMMON/CBLOC/IDCB2(144),NAME2(3),ITYPE(6,32),GAIN(32),OFFSET(32)
0072        .              IUNITS(10,32),YMINI(32),YMAXI(32),ICOMM(10,32)
0073         COMMON/FLAGS/LU,IFLAG
0074         END
```

```
&CNFIG T=00003 IS ON CR00015 USING 00032 BLKS R=0000

0001  FTN4,L
0002        SUBROUTINE CNFIG
0003  C
0004  C     THIS SUBROUTINE IS THE PRIMARY MODULE FOR GENERATING, MODIFYING,
0005  C     AND MAINTAINING THE USER DEFINED CONGIGURATION FILES THAT CONTAIN
0006  C     TRANSFER FUNCTION OR ENGINEERING UNIT CONVERSION DATA TO BE USED
0007  C     BY PROGRAM PLOTM IN PRODUCING ENGINEERING UNIT PLOTS. THE USER
0008  C     INTERFACES WITH THIS PROGRAM TO CHOOSE ONE OF THE FOLLOWING
0009  C     FUNCTIONS FOR CNFIG TO PERFORM:
0010  C
0011  C                1. GENERATE A NEW CONFIGURATION FILE.
0012  C                2. MODIFY EXISTING RECORDS IN A CONFIGURATION FILE.
0013  C                3. LIST A CONFIGURATION FILE.
0014  C                4. TERMINATE THE PROCESS & RETURN TO CFIGM.
0015  C
0016  C
0017        INTEGER ISIZE(2) LAB(10)
0018        COMMON/CBLOC/IDCB2(144),NAME2(3),ITYPE(6,32),GAIN(32),OFFSET(32),
0019       .             IUNITS(10,32),YMINI(32),YMAXI(32),ICOMM(10,32)
0020        COMMON/FLAGS/LU,IFLAG
0021  C
0022  C
0023        LFLAG = 1
0024        LP = 6
0025  C$Y
0026        WRITE(LU,100)
0027    100 FORMAT("$H$J")
0028  C$Z
0029  C
0030  C     WRITE WELCOME TO USER TERMINAL.
0031  C
0032        WRITE(LU,110)
0033    110 FORMAT(" WELCOME TO YOUR BASIC CONFIGURATION PROGRAM!!",/)
0034  C
0035  C     PROMPT USER TO ENTER CONFIGURATION.
0036  C
0037    120 CONTINUE
0038        WRITE(LU,130)
0039    130 FORMAT(" ENTER CONFIGURATION OPTION DESIRED ",/
0040       .       "                                    ",/
0041       .       " 1  GENERATE NEW CONFIGURATION FILE  ",/
0042       .       " 2  CHANGE AN ENTRY ",/
0043       .       " 3  LIST CONFIGURATION FILE ",/
0044       .       " 4  EXIT CONFIGURATION PROGRAM ")
0045  C
0046  C     DEFAULT OPTION IS 3 (LIST CONFIGURATION FILE).
0047  C
0048        IFIG = 3
0049  C
0050  C     ENTER OPTION.
0051  C
0052        READ(LU,*)IFIG
0053  C
0054  C     GO TO APPROPRIATE SECTION OF CFIGM.
0055  C
0056        GO TO(140,520,530,690),IFIG
0057  C
0058  C     PREPARE TO GENERATE A NEW CONFIGURATION FILE.
```

```
0059  C
0060    140 CONTINUE
0061        IDCB2(10) = 0
0062        ISIZE(1) = 6
0063        ISIZE(2) = 0
0064  C
0065  C     MAKE FMP CALL TO ENTER A NEW FILE WITH NAME "NAME2" ON LU 37.
0066  C
0067        CALL CREAT(IDCB2,IERR,NAME2,ISIZE,3,0,37)
0068  C
0069  C     IF CREAT RESULTS IN NO ERROR GO ZERO FORTRAN ARRAYS & BEGIN
0070  C     CONFIGURATION PROCESS.
0071  C
0072        IF(IERR .GE. 0)GO TO 190
0073  C
0074  C     IF FILE "NAME2" ALREADY EXISTS ON LU 37 GO ASK USER IF HE REALLY
0075  C     KNOWS WHAT HES DOING!
0076  C
0077        IF(IERR .EQ. -2)GO TO 160
0078  C
0079  C     BASIC PROBLEM! FMP ERROR ON CREAT. TERMINATE CNFIG.
0080  C
0081        WRITE(LU,150)IERR,NAME2
0082    150 FORMAT(" ERROR STOP. CREAT ERROR ON FILE ",3A2,". IERR = ",I6)
0083        CALL EXEC(6)
0084  C
0085  C     ASK USER ABOUT CREATING A FILE "NAME2" WHICH ALREADY EXISTS.
0086  C
0087    160 CONTINUE
0088        WRITE(LU,170)NAME2
0089    170 FORMAT(/,"ARE YOU SURE YOU WANT TO DESTROY THE OLD"
0090       . " CONFIGURATION FILE? '",3A2,"'",
0091       . " (Y/[N]): _")
0092        READ(LU,180) NOYES
0093    180 FORMAT(A1)
0094  C
0095  C     IF USER ANSWERS "N" (NO) GO BACK AND PROMPT HIM AGAIN!
0096  C
0097        IF(NOYES .NE. 1HY) GO TO 120
0098  C
0099  C     USER ANSWERED "Y" (YES)...MAYBE HE KNOWS WHAT HE'S DOING.
0100  C
0101    190 CONTINUE
0102  C
0103  C     INITIALIZE CONFIGURATION ARRAYS TO ZERO & BLANK.
0104  C
0105        DO 200 K = 1,32
0106        GAIN(K) = 0.
0107        OFFSET(K) = 0.
0108        YMINI(K) = 0.
0109        YMAXI(K) = 0.
0110    200 CONTINUE
0111        DO 210 I = 1,6
0112        DO 210 J = 1,32
0113        ITYPE(I,J) = 2H
0114    210 CONTINUE
0115        DO 220 I= 1,10
0116        DO 220 J = 1,32
0117        IUNITS(I,J) = 2H
0118        ICOMM(I,J) = 2H
```

```
  220 CONTINUE
  230 CONTINUE
C
C     CLEAR SCREEN ON CRT.
C
      WRITE(LU,100)
C
C     GET USER READY TO ENTER SENSOR DATA.
C
      WRITE(LU,240)
  240 FORMAT(" PREPARE TO ENTER SENSOR DATA ",/
     .       " TYPE <CR> WHEN READY TO PROCEED ",/)
      READ(LU,*)Z
  250 CONTINUE
C
C     CLEAR CRT AGAIN.
C
      WRITE(LU,100)
C$Z
C     ENTER CHANNEL NUMBER TO BEGIN ENTERING DATA FOR.
C
      WRITE(LU,260)
  260 FORMAT(" ENTER CHANNEL NO. ( < 0  OR  > 31 TO EXIT ): _")
      I = 0
      READ(LU,*)I
C
C     IF CHANNEL NUMBER LESS THAN ZERO (0) OR GREATER THAN 31, EXIT
C     THIS PROCESS AND TERMINATE CNFIG. RETURNING TO MAON PROGRAM
C     CFIGM.
C
      IF(I .LT. 0 .OR. I .GT. 31) GO TO 420
      II = I + 1
C
C     BEGIN ENTERING CONFIGURATION DATA.
C
  270 CONTINUE
C
C     ENTER SENSOR NAME.
C
      WRITE(LU,280)I,(ITYPE(J,II),J=1,6)
  280 FORMAT(" SENSOR NAME FOR CHANNEL ",I2," IS ",6A2)
      WRITE(LU,290)
  290 FORMAT(" ENTER SENSOR NAME:  $&dB$&a+1!C$&a-1!C_")
      READ(LU,300)(LAB(K),K=1,6)
  300 FORMAT(6A2)
      CALL LABLE(ITYPE(1,II),LAB,6)
C
C     ENTER SENSOR LINEAR GAIN.
C
      WRITE(LU,310)GAIN(II)
  310 FORMAT(" SENSOR LINEAR GAIN IS ",G12.4)
      WRITE(LU,320)(ITYPE(J,II),J=1,6)
  320 FORMAT(" ENTER LINEAR GAIN FOR ",6A2," : _")
      READ(LU,*)GAIN(II)
C
C     ENTER SENSOR OFFSET.
C
      WRITE(LU,330)OFFSET(II)
  330 FORMAT(" SENSOR OFFSET FACTOR IS ",G12.4)
      WRITE(LU,340)(ITYPE(J,II),J=1,6)
```

```
0179     340 FORMAT(" ENTER OFFSET FOR ",6A2," : _")
0180         READ(LU,*)OFFSET(II)
0181   C
0182   C     ENTER SENSOR ENGINEERING UNITS.
0183   C
0184         WRITE(LU,350)(IUNITS(K,II),K=1,10)
0185     350 FORMAT(" SENSOR ENG. UNITS ARE: ",10A2)
0186         WRITE(LU 360)(ITYPE(J,II),J=1,6)
0187     360 FORMAT(" ENTER ENG. UNITS FOR ",6A2,":  ξ&dBξ&a+19Cξ&a-19C_")
0188         READ(LU,410)(LAB(K),K=1,10)
0189         CALL LABLE(IUNITS(1,II),LAB,10)
0190   C
0191   C     ENTER SENSOR Y-AXIS DEFAULTS (YMIN & YMAX) FOR PLOTS.
0192   C
0193         WRITE(LU,370)YMINI(II),YMAXI(II)
0194     370 FORMAT(" YMIN, YMAX PLOT LIMITS ARE ",2G12.4)
0195         WRITE(LU,380)
0196     380 FORMAT(" ENTER YMIN, YMAX PLOT LIMITS IN ENG. UNITS: _")
0197         READ(LU,*)YMINI(II),YMAXI(II)
0198   C
0199   C     ENTER SENSOR COMMENT FOR THIS CHANNEL (20 CHARACTERS MAXIMUM).
0200   C
0201         WRITE(LU,390)(ICOMM(K,II),K=1,10)
0202     390 FORMAT(" CURRENT SENSOR COMMENT IS '",10A2,"'")
0203         WRITE(LU,400)
0204     400 FORMAT(" ENTER SENSOR COMMENT:  ξ&dBξ&a+19Cξ&a-19C_")
0205         READ(LU,410)(LAB(K),K=1,10)
0206     410 FORMAT(10A2)
0207         CALL LABLE(ICOMM(1,II),LAB,10)
0208   C
0209   C     GO ENTER ANOTHER CHANNEL.
0210   C
0211         GO TO 250
0212   C
0213   C     OUT OF CHANNEL DATA ENTRY LOOP. PREPARE TO WRITE TO DISK.
0214   C
0215     420 CONTINUE
0216   C
0217   C     ASK USER IF HE WANTS TO LIST CONFIGURATION FILE TO LINE PRINTER.
0218   C
0219         LFLAG = 3
0220         WRITE(LU,600)
0221         READ(LU,180) NOYES
0222   C
0223   C     ANSWER IS YES. GO DO IT.
0224   C
0225         IF(NOYES .EQ. 1HY) GO TO 610
0226   C
0227   C     DOES USER WANT TO WRITE CONFIGURATION DATA TO DISK?
0228   C
0229     430 WRITE(LU,440)
0230     440 FORMAT(" DO YOU WANT TO WRITE TO CONFIGURATION FILE? (Y/[N]): _"
0231         READ(LU,180) NOYES
0232   C
0233   C     ANSWER IS NO. GO CLOSE DISK FILE & RETURN TO CFIGM.
0234   C
0235         IF(NOYES .NE. 1HY) GO TO 680
0236   C
0237   C     ANSWER IS YES. WRITE DATA TO DISK FILE "NAME2".
0238   C
```

```
0239     450 CONTINUE
0240   C
0241         CALL OPEN(IDCB2,IERR,NAME2)
0242         IF(IERR .GE. 0)GO TO 470
0243         WRITE(LU,460)IERR,NAME2
0244     460 FORMAT(" ERROR STOP. OPEN ERROR. IERR = ",I6," ON FILE ",3A2)
0245         CALL EXEC(6)
0246     470 CONTINUE
0247   C
0248         CALL WRITF(IDCB2,IERR,ITYPE,192)
0249         IF(IERR .LT. 0)GO TO 480
0250         CALL WRITF(IDCB2,IERR,GAIN,64)
0251         IF(IERR .LT. 0)GO TO 480
0252         CALL WRITF(IDCB2,IERR,OFFSET,64)
0253         IF(IERR .LT. 0)GO TO 480
0254         CALL WRITF(IDCB2,IERR,IUNITS,320)
0255         IF(IERR .LT. 0)GO TO 480
0256         CALL WRITF(IDCB2,IERR,YMINI,64)
0257         IF(IERR .LT. 0) GO TO 480
0258         CALL WRITF(IDCB2,IERR,YMAXI,64)
0259         IF(IERR .LT. 0)GO TO 480
0260         CALL WRITF(IDCB2,IERR,ICOMM,320)
0261         IF(IERR .GE. 0)GO TO 500
0262     480 CONTINUE
0263   C
0264   C      ERROR OCCUPRED IN A WRITF. TERMINATE CNFIG.
0265   C
0266         WRITE(LU,490)IERR,NAME2
0267     490 FORMAT(" ERROR STOP. WRITF ERROR. IERR = ",I6," ON FILE ",3A2)
0268         CA  EXEC(6)
0269   C
0270   C      INFORM USER DISK FILE "NAME2" HAS BEEN WRITTEN TO DISK.
0271   C
0272     500 CONTINUE
0273         WRITE(LU,510)(NAME2(L),L=1,3)
0274     510 FORMAT(//" CONFIGURATION PROCESS COMPLETE ",/
0275        .           " DISK FILE ",3A2," WRITTEN & CLOSED ",/)
0276   C
0277   C      RETURN TO CFIGM AFTER CLOSING DISK FILE "NAME2".
0278   C
0279         GO TO 680
0280   C
0281   C      OPTION CHOSEN WAS 2 (MODIFY AN ENTRY).
0282   C
0283     520 CONTINUE
0284         LFLAG = 2
0285   C      OPEN FILE "NAME2" AND READ EXISTING CONFIGURATION DATA INTO
0286   C      APPROPRIATE FORTRAN ARRAYS.
0287   C
0288   C
0289     530 CONTINUE
0290         CALL OPEN(IDCB2,IER,NAME2)
0291         IF(IER .GE. 0)GO TO 550
0292         WRITE(LU,540)IER,NAME2
0293     540 FORMAT(" ERROR STOP. OPEN ERROR. IER = ",I6," ON FILE ",3A2)
0294         CALL EXEC(6)
0295     550 CONTINUE
0296   C
0297         CALL READF(IDCB2,IERR,ITYPE,192)
0298         IF(IERR .LT. 0)GO TO 560
```

```
0299          CALL READF(IDCB2,IERR,GAIN,64)
0300          IF(IERR .LT. 0)GO TO 560
0301          CALL READF(IDCB2,IERR,OFFSET,64)
0302          IF(IERR .LT. 0)GO TO 560
0303          CALL READF(IDCB2,IERR,IUNITS,320)
0304          IF(IERR .LT. 0)GO TO 560
0305          CALL READF(IDCB2,IERR,YMINI,64)
0306          IF(IERR .LT. 0) GO TO 560
0307          CALL READF(IDCB2,IERR,YMAXI,64)
0308          IF(IERR .LT. 0)GO TO 560
0309          CALL READF(IDCB2,IERR,ICOMM,320)
0310          IF(IERR .GE. 0)GO TO 580
0311      560 CONTINUE
0312  C
0313  C       ERROR ON A READF. TERMINATE CNFIG.
0314  C
0315          WRITE(LU,570)IERR,NAME2
0316      570 FORMAT(" ERROR STOP. READF ERROR. IERR = ",I6," ON FILE ",3A2)
0317          CALL EXEC(6)
0318      580 CONTINUE
0319  C
0320  C       IF OPTION IS LIST GO DO IT.
0321  C
0322      590 IF(IFIG.EQ.3)GO TO 610
0323  C
0324  C       ASK USER IF HE WANTS CONFIGURATION DATA JUST READ IN FROM FILE
0325  C       "NAME2" LISTED ON PRINTER.
0326  C
0327          WRITE(LU,600)
0328      600 FORMAT(" DO YOU WANT TO PRINT CONFIGURATION FILE?  (Y/[N]): _")
0329          READ(LU,180) NOYES
0330  C
0331  C       ANSWER IS NO. GO BEGIN ENTERING CHANNEL DATA IN ORDER TO
0332  C       ACCOMPLISH THE DATA MODIFICATION FUNCTION.
0333  C
0334          IF(NOYES .NE. 1HY) GO TO 230
0335  C
0336  C       LIST CONFIGURATION DATA ON LINE PRINTER.
0337  C
0338      610 WRITE(LP,620)
0339      620 FORMAT(1H1)
0340          WRITE(LP,630)
0341      630 FORMAT( )
0342          WRITE(LP,640)NAME2
0343      640 FORMAT(49X," CONFIGURATION TABLE FOR FILE ",3A2,//)
0344          WRITE(LP,650)
0345      650 FORMAT("    CN        SENSOR                GAIN          OFFSET",
0346         .       "              UNITS                  YMIN            YMAX",
0347         .       "             COMMENT",/)
0348          DO 670 I =1,32
0349          II = I-1
0350          WRITE(LP,660)II,(ITYPE(K,I),K=1,6),GAIN(I),OFFSET(I),
0351         .              (IUNITS(K,I),K=1,10),YMINI(I),YMAXI(I),
0352         .              (ICOMM(K,I),K=1,10)
0353      660 FORMAT(1X,I5,5X,6A2,2(5X,F10.3),5X,10A2,2(5X,F10.3),5X,10A2)
0354      670 CONTINUE
0355          WRITE(LP,630)
0356          WRITE(LP,620)
0357  C
0358  C       IF OPTION IS 1 (GENERATE NEW FILE) ..... RETURN TO CFIGM.
```

```
C      IF OPTION IS 2 (MODIFY A: ENTRY)    ..... GO ENTER MORE DATA.
C      IF OPTION IS 3 (LIST)               ..... GO ASK USER NEXT OPTION.
C
       GO TO (680,230,430),LFLAG
C
C      CLOSE DISK FILE "NAME2".
C
  680 CONTINUE
      CALL CLOSE(IDCB2)
  690 CONTINUE
C
C      RETURN TO CFIGM.
C
       RETURN
       END
       SUBROUTINE LABLE(LAB1,LAB2,NUM)
C
C      THIS ROUTINE MODIFIES LAB1 BY LAB2
C           IF LAB2 IS BLANK  LAB1 = LAB1
C           IF LAB2 HAS '     LAB1 = BLANK
C           OTHERWISE         LAB1 = LAB2
C
       DIMENSION LAB1(1), LAB2(1)
       DO 100 K=1,NUM
       IF(LAB2(K) .NE. 2H  ) GO TO 110
  100 CONTINUE
       RETURN
  110 IF(LAB2(1) .EQ. 2H' ) GO TO 130
       DO 120 K=1,NUM
  120 LAB1(K) = LAB2(K)
       RETURN
  130 DO 140 K=1,NUM
  140 LAB1(K) = 2H
       RETURN
       END
```

```
&PLOTM T=00003 IS ON CR00015 USING 00163 BLKS R=0000

0001  FTN4,Q
0002        PROGRAM PLOTM
0003  C
0004  C*****************************************************************************
0005  C
0006  C     PROGRAM PLOTM. READS DISK FILES CREATED BY PROGRAM READM AND
0007  C     PRODUCES PLOTS ON THE REEDA SYSTEM'S HP 2647A GRAPHICS TERMINAL,
0008  C     HP 2608A LINE PRINTER, AND HP 9872B 4-COLOR PEN PLOTTER.
0009  C
0010  C     USER CAN CHOOSE:                                                    *
0011  C          1. PLOTTING DEVICE.                                            *
0012  C          2. VOLTAGE OR ENGINEERING UNIT PLOTS.                          *
0013  C          3. START & END TIME OF PLOT.                                   *
0014  C          4. MINIMUM & MAXIMUM Y-AXIS VALUES.                            *
0015  C          5. AUTOSCALING OF Y-AXIS.                                      *
0016  C          6. AXES & GRID TYPE FOR PLOT.                                  *
0017  C          7. STARTING CHANNEL NUMBER FOR PLOTS.                          *
0018  C          8. SINGLE OR MULTIPLE CHANNEL PLOTTING.                        *
0019  C                                                                         *
0020  C                                                                         *
0021  C     DEVELOPED BY:  ESPEE INC.                                           *
0022  C                    EXECUTIVE PLAZA                                      *
0023  C                    SUITE 305                                            *
0024  C                    205/837-8585                                         *
0025  C                                                                         *
0026  C                                                                         *
0027  C*****************************************************************************
0028  C
0029  C
0030        DIMENSION IGCB(192),IOBUF(20)
0031        DIMENSION IBUF(100),LAB(10)
0032        COMMON/CBLOC/IDCB2(144),NAME2(3),ITYPE(6,32),GAIN(32),OFFSET(32),
0033       .             IUNITS(10,32),YMINI(32),YMAXI(32),ICOMM(10,32)
0034        COMMON/DBLOC/IDCB(144),NAME(3)
0035        COMMON/FLAGS/LU,KPLOT,IFLAG,TIME1,TIME2
0036        INTEGER GTYPE,AUTOSC,AUTOPL,RANGE
0037        DATA LU2647/1/
0038  C
0039  C       GET LU NUMBER OF HOST TERMINAL.
0040  C
0041        CALL RMPAR(IBUF)
0042        LU = IBUF(1)
0043        IF(LU .LT. 1)LU = 1
0044  C
0045    100 CONTINUE
0046  C
0047  C       CLEAR TERMINAL SCREEN AND POSITION CURSOR AT HOME POSITION.
0048  C
0049  C$Y
0050        WRITE(LU,110)
0051    110 FORMAT("$H$J")
0052  C$Z
0053  C
0054  C       IDENTIFY PROGRAM AS PLOTM.
0055  C
0056        WRITE(LU,105)
0057    105 FORMAT("       PLOTM - HP 1000 GRAPHICS DATA REDUCTION PROGRAM "
0058       .       ,/)
```

```
0059  C
0060  C       GET FILENAME OF DATA TO BE PLOTTED.
0061  C       IF FILENAME = EX  OR  /E  TERMINATE PROGRAM PLOTM.
0062  C
0063        WRITE(LU,120)
0064    120 FORMAT(" ENTER FILENAME OF DATA FILE TO PLOT: _")
0065        READ(LU,130)(NAME(I),I=1,3)
0066    130 FORMAT(3A2)
0067        IF(NAME(1) .EQ. 2HEX .OR. NAME(1) .EQ. 2H/E)GO TO 820
0068  C
0069  C       OPEN FILE.
0070  C
0071        CALL OPEN(IDCB,IERR,NAME)
0072  C
0073  C       IF OPEN SUCCESSFUL GO READ FIRST BLOCK OF DATA.
0074  C
0075        IF(IERR .GE. 0)GO TO 150
0076  C
0077  C         ERROR ON OPEN , TERMINATE PLOTM.
0078  C
0079        WRITE(LU,140)NAME,IERR
0080    140 FORMAT(" ERROR STOP. OPEN ERROR ON FILE ",3A2,". IERR = ",I5)
0081        CALL EXEC(6)
0082  C
0083    150 CONTINUE
0084  C
0085  C       READ FIRST RECORD OF DISK FILE (HEADER RECORD).
0086  C
0087        CALL READF(IDCB,IERR,IBUF,84)
0088  C
0089  C       IF READF SUCCESSFUL GO WRITE HEADER ON TERMINAL.
0090  C
0091        IF(IERR .GE. 0)GO TO 170
0092  C
0093  C         ERROR ON READF , TERMINATE PLOTM.
0094  C
0095        WRITE(LU,160)NAME,IERR
0096    160 FORMAT(" ERROR STOP. READF ERROR ON FILE ",3A2,". IERR = ",I5)
0097        CALL EXEC(6)
0098  C
0099    170 CONTINUE
0100  C
0101  C       SAVE SOME HEADER PARAMETERS FOR LATER COMPUTATION & CONTROL.
0102  C
0103        NSEQ = IBUF(1)
0104        IDAY = IBUF(2)
0105        IMON = IBUF(3)
0106        IYEAR = IBUF(4)
0107        MAXCN = IBUF(11)
0108  C
0109  C       WRITE HEADER ON TERMINAL.
0110  C
0111        WRITE(LU,180)IBUF(1),IBUF(3),IBUF(2),IBUF(4),(IBUF(M),M=5,11)
0112    180 FORMAT(" THE HEADER RECORD FOR THIS FILE SHOWS: ",/
0113       .       " SEQUENCE NO. = ",I2,/
0114       .       " DATE         = ",I2,"/",I2,"/",I2,/
0115       .       " START TIME   = ",I2,":",I2,":",I2,/
0116       .       " STOP TIME    = ",I2,":",I2,":",I2,/
0117       .       " NO. CHANNELS = ",I2)
0118  C
```

```
0119  C       IS THIS THE CORRECT FILE? IF YES, CONTINUE.
0120  C
0121        WRITE(LU,190)
0122    190 FORMAT(" IS THIS THE CORRECT FILE ?  (Y)/N): _")
0123        IANS = 1HY
0124        READ(LU,200)IANS
0125    200 FORMAT(A1)
0126        IF(IANS .NE. 1HN)GO TO 220
0127  C
0128  C       AN ANSWER OF "N" MEANS THIS IS NOT THE CORRECT FILE. WRITE ERROR
0129  C       MESSAGE & TERMINATE PLOTM.
0130  C
0131        WRITE(LU,210)
0132    210 FORMAT(" PROGRAM PLOTM TERMINATED. CHECK FOR DESIRED FILE NAME.")
0133        CALL EXEC(6)
0134  C
0135    220 CONTINUE
0136  C
0137  C       SELECT PLOT TYPE:  VOLTAGE OR ENGINEERING UNIT.
0138  C                          KPLOT = 1 IMPLIES VOLTAGE PLOT.
0139  C                          KPLOT = 2 IMPLIES ENGINEERING UNIT PLOT.
0140  C
0141        WRITE(LU,230)
0142    230 FORMAT(" SELECT PLOT TYPE.  (VO),EN): _")
0143        KPLOT = 1
0144        READ(LU,240)IOPT
0145    240 FORMAT(A2)
0146        IF(IOPT .EQ. 2HEN)KPLOT = 2
0147  C
0148  C       WHAT IS THE NAME OF THE CONFIGURATION TABLE TO BE USED
0149  C       FOR THIS RUN? DEFAULT FILENAME IS  "TABLEA".
0150  C
0151        WRITE(LU,250)NAME2
0152    250 FORMAT(" ENTER FILENAME FOR CONFIGURATION TABLE.  ","[",3A2,
0153               "]: _")
0154        READ(LU,130)(LAB(I),I=1,3)
0155        CALL LABLE(NAME2(1),LAB,3)
0156  C
0157  C       OPEN CONFIGURATION FILE FOR READING.
0158  C
0159        CALL OPEN(IDCB2,IERR,NAME2)
0160  C
0161  C       IF OPEN SUCCESSFUL GO READ CONFIGURATION DATA.
0162  C
0163        IF(IERR .GE. 0)GO TO 260
0164  C
0165  C       ERROR ON OPEN, TERMINATE PLOTM.
0166  C
0167        WRITE(LU,140)NAME2,IERR
0168        CALL EXEC(6)
0169  C
0170    260 CONTINUE
0171  C
0172  C       READ CONFIGURATION DATA INTO APPROPRIATE ARRAYS.
0173  C
0174        CALL READF(IDCB2,IERR,ITYPE,192)
0175        IF(IERR .LT. 0)GO TO 270
0176        CALL READF(IDCB2,IERR,GAIN,64)
0177        IF(IERR .LT. 0)GO TO 270
0178        CALL READF(IDCB2,IERR,OFFSET,64)
```

```
0179          IF(IERR .LT. 0)GO TO 270
0180          CALL READF(IDCB2,IERR,IUNITS,320)
0181          IF(IERR .LT. 0)GO TO 270
0182          CALL READF(IDCB2,IERR,YMINI,64)
0183          IF(IERR .LT. 0)GO TO 270
0184          CALL READF(IDCB2,IERR,YMAXI,64)
0185          IF(IERR .LT. 0)GO TO 270
0186          CALL READF(IDCB2,IERR,ICOMM,320)
0187    C
0188    C       IF IERR ZERO OR POSITIVE THEN ALL READS HAVE BEEN ERROR FREE.
0189    C       NOW GO & READ THE GRAPHICS LU TO DO THE PLOTTING ON.
0190    C
0191          IF(IERR .GE. 0)GO TO 290
0192    C
0193    C       AN ERROR ON A READF HAS OCCURRED. TERMINATE PLOTM.
0194    C
0195      270 CONTINUE
0196          WRITE(LU,280)NAME2,IERR
0197      280 FORMAT(" ERROR STOP. READF ERROR ON FILE ",3A2,". IERR = ",I5)
0198          CALL EXEC(6)
0199    C
0200      290 CONTINUE
0201    C
0202    C       READ GRAPHICS LU.
0203    C         HP 2647A GRAPHICS TERMINAL     =   LU 1  &  DEVICE ID = 1.
0204    C         HP 9872B 4-COLOR PEN PLOTTER   =   LU 20 &  DEVICE ID = 2.
0205    C         HP 2608A LINE PRINTER          =   LU 28 &  DEVICE ID = 3.
0206    C
0207          WRITE(LU,300)
0208      300 FORMAT(" SELECT GRAPHICS LU.  ([1],20,28): _")
0209          LUG = 1
0210          READ(LU,*)LUG
0211          IF(LUG .EQ. 1)ID = 1
0212          IF(LUG .EQ. 20)ID = 2
0213          IF(LUG .EQ. 28)ID = 3
0214    C
0215    C       READ START & STOP TIMES FOR PLOT. DEFAULT IS TSTART = 0.
0216    C       & TSTOP = END OF RUN.
0217    C
0218          WRITE(LU,310)
0219      310 FORMAT(" ENTER START TIME FOR PLOT (ELAPSED SECONDS): _")
0220          TIMES = 0.0
0221          READ(LU,*)TIMES
0222          WRITE(LU,320)
0223      320 FORMAT(" ENTER STOP TIME FOR PLOTS (ELAPSED SECONDS): _")
0224          TIMEF = 0.0
0225          READ(LU,*)TIMEF
0226    C
0227    C       AUTO SCALE? IF YES, AUTOPL = 0. IF NOT, AUTOPL = 1.
0228    C
0229          AUTOPL = 1
0230          AUTOSC = 1HN
0231          WRITE(LU,330)
0232      330 FORMAT(" AUTOSCALE Y-AXIS?  (Y/[N]): _")
0233          READ(LU,200)AUTOSC
0234          IF(AUTOSC .NE. 1HY)GO TO 340
0235          AUTOPL = 0
0236          GO TO 360
0237      340 CONTINUE
0238    C
```

```
C       READ MINIMUN & MAXIMUM VALUES FOR Y-AXIS.
C       DEFAULT FOR VOLTAGE PLOTS ARE -5.0 & 5.0 VOLTS.
C       DEFAULT FOR ENGINEERING UNIT PLOTS IS +1.0E+37 FOR YMIN & YMAX.
C
      Y1 = -5.0
      Y2 = 5.0
      IF(KPLOT .EQ. 2)Y1 = 1.0E+37
      IF(KPLOT .EQ. 2)Y2 = 1.0E+37
      WRITE(LU,350)
  350 FORMAT(" ENTER MIN AND MAX FOR Y-AXIS VALUES: _")
      READ(LU,*)Y1,Y2
      YMIN = AMIN1(Y1,Y2)
      YMAX = AMAX1(Y1,Y2)
C
C       READ AXIS & GRID TYPE.
C       AXES WITHOUT GRID:  GTYPE=0
C       AXES WITH GRID:     GTYPE=1
C
  360 GTYPE = 0
      WRITE(LU,370)
  370 FORMAT(" SELECT AXES AND GRID TYPE ",/
     .       "(AXES WITHOUT GRID-[0], AXES WITH GRID-1): _")
      READ(LU,*)GTYPE
C
C       ENTER CHANNEL NUMBER TO PLOT. DEFAULT IS CHANNEL 0.
C
      JCN = 0
      WRITE(LU,380)
  380 FORMAT(" ENTER CHANNEL NO. TO PLOT: _")
      READ(LU,*)JCN
C
C       PLOT CHANNEL JCN ONLY OR PLOT REMAINING CHANNELS AS WELL?
C
      WRITE(LU,390)
  390 FORMAT(" PLOT ALL SUBSEQUENT CHANNELS WITHOUT "
     .       " OPERATOR INTERVENTION ?  ([Y]/N): _")
      IGO = 0
      READ(LU,200)IANS
      IF(IANS .EQ. 1HN)IGO = 1
C
  400 CONTINUE
C
C       CONVERT RUN START AND STOP TIMES (FROM HEADER RECORD)
C       TO SECONDS.
C
      IFLAG = 1
      TIME1 = FLOAT(IBUF(5))*3600. + FLOAT(IBUF(6))*60.
     .        + FLOAT(IBUF(7))
      TIME2 = FLOAT(IBUF(8))*3600. + FLOAT(IBUF(9))*60.
     .        + FLOAT(IBUF(10))
C
C       TIMES = 0.0 & TIMEF = 0.0 IMPLIES DEFAULT START & STOP
C       TIMES CHOSEN. DO NOT COMPLETE TIME1 & TIME2 WITH
C       FOLLOWING STATEMENTS.
C
      IF(TIMES .EQ. 0.0 .AND. TIMEF .EQ. 0.0)GO TO 410
      TIME2 = TIME1 + TIMEF
      TIME1 = TIME1 + TIMES
  410 CONTINUE
C
```

```
0299  C       COMPUTE ELAPSED TIME IN SECONDS.
0300  C
0301        ELTIM = TIME2 - TIME1
0302  C
0303  C       CONVERT START & STOP TIMES BACK TO HRS., MIN., & SEC.
0304  C       FOR PLOT LABELING PURPOSES.
0305  C
0306        NHRS = IFIX(TIME1/3600.)
0307        NMIN = IFIX(TIME1-FLOAT(NHRS)*3600.)/60
0308        NSEC = IFIX(TIME1-FLOAT(NHRS)*3600.-FLOAT(NMIN)*60.)
0309        KHRS = IFIX(TIME2/3600.)
0310        KMIN = IFIX(TIME2-FLOAT(KHRS)*3600.)/60
0311        KSEC = IFIX(TIME2-FLOAT(KHRS)*3600.-FLOAT(KMIN)*60.)
0312  C
0313  C       IF ELAPSED TIME IN SECONDS IS GREATER THAN 1000 SECONDS
0314  C       CONVERT ELAPSED TIME ALONG WITH START & STOP TIMES TO
0315  C       MINUTES.
0316  C       IFLAG = 2 WILL DENOTE TIME IS IN MINUTES FROM THIS
0317  C       POINT ON. IFLAG = 1 FROM HERE ON MEANS TIME IS IN SECONDS.
0318  C
0319        IF(ELTIM .GT. 1000)IFLAG = 2
0320        IF(IFLAG .EQ. 2)TIME1 = TIME1/60.0
0321        IF(IFLAG .EQ. 2)TIME2 = TIME2/60.0
0322        IF(IFLAG .EQ. 2)ELTIM = ELTIM/60.
0323  C
0324  C
0325     420 CONTINUE
0326  C
0327  C       IF AUTOSCALING WAS CHOSEN CALL SUBROUTINE BOUND TO
0328  C       READ THE Y-ARRAY FOR CHANNEL JCN AND DETERMINE YMIN
0329  C       AND YMAX.
0330  C
0331        IF(AUTOPL .NE. 0)GO TO 430
0332        CALL BOUND(JCN,YMIN,YMAX)
0333        DELTA = (YMAX - YMIN)
0334        IF(DELTA .GT. 1.0)GO TO 430
0335        Y1 = YMIN - 0.10*YMIN
0336        Y2 = YMAX + 0.10*YMAX
0337        YMIN = AMIN1(Y1,Y2)
0338        YMAX = AMAX1(Y1,Y2)
0339  C
0340     430 CONTINUE
0341  C
0342  C       KPLOT = 1 IMPLIES VOLTAGE PLOT. SKIP CHECK ON KPLOT = 2
0343  C       & YMIN NOT EQUAL YMAX. INTENDED FOR ENGINEERING UNIT PLOTS
0344  C       ONLY.
0345  C
0346        IF(KPLOT .EQ. 1)GO TO 440
0347  C
0348  C       KPLOT = 2  IMPLIES ENGINEERING UNIT PLOT. IF YMIN=YMAX GET
0349  C       YMIN & YMAX FROM CONFIGURATION ARRAYS.
0350  C
0351        IF(KPLOT .EQ. 2 .AND. YMIN .NE. YMAX)GO TO 440
0352        YMIN = YMINI(JCN+1)
0353        YMAX = YMAXI(JCN+1)
0354  C
0355     440 CONTINUE
0356  C
0357  C       CALCULATE XTIC, YTIX, AND TMAX FOR DEFINING TIC MARKS &
0358  C       X, Y LIMITS IN ORDER TO DRAW THE AXES AND/OR GRID.
```

```
0359  C
0360        YTIC = (YMAX-YMIN)/20.0
0361        TMAX = 60*(AINT((ELTIM-1.0)/60.0)+1.0)
0362        XTIC = TMAX/24.0
0363  C
0364  C       DEFINE ASPECT RATIO (AR) FOR THE SELECTED PLOTTING DEVICE.
0365  C
0366        AR = 2.0028
0367        IF(ID .EQ. 2)AR = 1.52
0368        IF(ID .EQ. 3)AR = 1.283
0369  C
0370  C       INITIALIZE THE GRAPHICS TASK.
0371  C
0372        CALL PLOTR(IGCB,ID,1.LUG)
0373  C
0374  C       DEFINE THE VIEW SURFACE ASPECT RATIO.
0375  C
0376        CALL SETAR(IGCB,AR)
0377  C
0378  C       IF PLOT DEVICE IS HP 2647A TERMINAL, INITIALIZE GRAPHICS MODE
0379  C       AND TURN OFF THE ALPHANUMERIC DISPLAY.
0380  C
0381        IF(ID .EQ. 1)CALL GRAF(LU2647)
0382  C
0383  C       IF PLOT DEVICE IS HP 9872B 4-COLOR PEN PLOTTER SELECT PEN #1
0384  C       TO DRAW AXES AND/OR GRID AND TO WRITE INITIAL PLOT TITLE
0385  C       INFORMATION.
0386  C
0387        IF(ID  EQ. 2)CALL PEN(IGCB,1)
0388  C
0389  C       DRAW A FRAME AROUND THE CURRENT CLIPPING LIMITS. AT THIS POINT
0390  C       THE LIMITS ARE THE PHYSICAL LIMITS OF THE SELECTED DEVICE.
0391  C
0392        CALL FRAME(IGCB)
0393  C
0394  C       DEFINE THE VIEWPORT IN THE NORMALIZED COORDINATE SYSTEM.
0395  C
0396        IF(ID .EQ. 1)CALL VIEWP(IGCB,20.,180.,14.,84.)
0397        IF(ID .EQ. 2)CALL VIEWP(IGCB,16.,138.,14.,84.)
0398        IF(ID .EQ. 3)CALL VIEWP(IGCB,14.,118.,14.,84.)
0399  C
0400  C       DRAW A SECOND FRAME AROUND THE REDEFINED CLIPPING LIMITS.
0401  C
0402        CALL FRAME(IGCB)
0403  C
0404  C       DEFINE A WINDOW IN THE WORLD COORDINATE SYSTEM.
0405  C
0406        CALL WINDW(IGCB,0.,TMAX,YMIN,YMAX)
0407  C
0408  C       FOR 2608A LINE PRINTER PLOTS, MAKE THE CHARACTER SIZE HEIGHT
0409  C       SMALLER SO THEY DON'T DOMINATE THE RESULTANT PLOT.
0410  C
0411        IF(ID .EQ. 3)CALL CSIZE(IGCB,2.0)
0412  C
0413  C       DEFINE n IN A FORTRAN F7.n FORMAT FOR AXIS LABELLING.
0414  C
0415        CALL FXD(IGCB,2)
0416        IF(ID .EQ. 3)CALL FXD(IGCB,1)
0417  C
0418  C       IF GTYPE = 0 DRAW LABELLED X-Y AXES WITHIN THE CLIPPING REGION.
```

```
0479  C       LABEL Y-AXIS "VOLTS" I. VOLTAGE PLOT WAS SELECTED.
0480  C
0481        IF(KPLOT .EQ. 1)WRITE(LUG 470)
0482    470 FORMAT("VOLTAGE (VOLTS)")
0483  C
0484  C       LABEL Y-AXIS WITH LABEL FROM CONFIGURATION ARRAY IUNITS
0485  C       IF ENG. UNIT PLOT WAS SELECTED.
0486  C
0487        IF(KPLOT .EQ. 2)WRITE(LUG,480)(IUNITS(JJ,JCN+1),JJ=1,10)
0488    480 FORMAT(10A2)
0489  C
0490  C       RE-ORIENT LABEL DIRECTION TO HORIZONTAL.
0491  C
0492        CALL LDIR(IGCB,0.0)
0493  C
0494  C       FOR HP 9872B 4-COLOR PEN PLOTTER PLOTS, MAKE THE CHARACTER
0495  C       SIZE HEIGHT SMALLER SO THEY DON'T DOMINATE RESULTANT PLOT.
0496  C
0497        IF(ID .EQ. 2)CALL CSIZE(IGCB,2.0)
0498  C
0499  C       DEFINE X NORMALIZED COORDINATE FOR THE SELECTED PLOTTING DEVICE
0500  C       IN PREPARATION FOR WRITING HEADER INFORMATION ON PLOT.
0501  C
0502        XP = 20.0
0503        IF(ID .EQ. 2)XP = 16.0
0504        IF(ID .EQ. 3)XP = 14.0
0505  C
0506  C       MOVE PEN TO X-Y NORMALIZED COORDINATE IN PREAPARATION
0507  C       FOR WRITING DISK FILE NAME ON PLOT. TURN ON TEXT MODE FOR
0508  C       NEXT WRITE STATEMENT.
0509  C
0510        CALL MOVE(IGCB,XP,97.0)
0511        CALL LABEL(IGCB)
0512  C
0513  C       WRITE DISK FILE NAME ON PLOT.
0514  C
0515        WRITE(LUG,490)NAME
0516    490 FORMAT("FILE NAME IS ",3A2)
0517  C
0518  C       MOVE PEN TO NORMALIZED X-Y COORDINATE IN PREPARATION
0519  C       FOR WRITING TEST SEQUENCE NUMBER ON PLOT. TURN ON TEXT
0520  C       MODE FOR NEXT WRITE STATEMENT.
0521  C
0522        CALL MOVE(IGCB,XP,94.0)
0523        CALL LABEL(IGCB)
0524  C
0525  C       WRITE SEQUENCE NUMBER ON PLOT.
0526  C
0527        WRITE(LUG,500)NSEQ
0528    500 FORMAT("SEQUENCE NO. IS ",I3)
0529  C
0530  C       MOVE PEN TO NORMALIZED X-Y COORDINATE IN PREPARATION
0531  C       FOR WRITING CURRENT CHANNEL NUMBER & START DATE ON PLOT.
0532  C       TURN ON TEXT MODE FOR NEXT WRITE STATEMENT.
0533  C
0534        CALL MOVE(IGCB,XP,91.0)
0535        CALL LABEL(IGCB)
0536  C
0537  C       WRITE CHANNEL NUMBER & START DATE ON PLOT.
0538  C
```

```
0539          WRITE(LUG,510)JCN,IMON,IDAY,IYEAR
0540      510 FORMAT("CHANNEL ",I2,1X,I2,"/",I2,"/",I2)
0541    C
0542    C       MOVE PEN TO NORMALIZED X-Y COORDINATE IN PREPARATION
0543    C       FOR WRITING TEST START TIME ON PLOT. TURN ON TEXT MODE
0544    C       FOR NEXT WRITE STATEMENT.
0545    C
0546          CALL MOVE(IGCB,XP,88.0)
0547          CALL LABEL(IGCB)
0548    C
0549    C       WRITE START TIME ON PLOT.
0550    C
0551          WRITE(LUG,520)NHRS,NMIN,NSEC
0552      520 FORMAT("START TIME:",I2,":",I2,":",I2)
0553    C
0554    C       MOVE PEN TO NORMALIZED X-Y COORDINATE IN PREPARATION
0555    C       FOR WRITING TEST STOP TIME ON PLOT. TURN ON TEXT MODE
0556    C       FOR NEXT WRITE STATEMENT.
0557    C
0558          CALL MOVE(IGCB,XP,85.0)
0559          CALL LABEL(IGCB)
0560    C
0561    C       WRITE STOP TIME ON PLOT.
0562    C
0563          WRITE(LUG,530)KHRS,KMIN,KSEC
0564      530 FORMAT("STOP TIME :",I2,":",I2,":",I2)
0565    C
0566    C       REDEFINE THE VIEWPORT BACK TO THE NORMALIZED COORDINATE SYSTEM
0567    C       FOR THE SELECTED PLOTTING DEVICE.
0568    C
0569          IF(ID .EQ. 1)CALL VIEWP(IGCB,20.,180.,14.,84.)
0570          IF(ID .EQ. 2)CALL VIEWP(IGCB,16.,138.,14.,84.)
0571          IF(ID .EQ. 3)CALL VIEWP(IGCB,14.,118.,14.,84.)
0572    C
0573    C       REDEFINE THE WINDOW IN THE WORLD COORDINATE SYSTEM.
0574    C
0575          CALL WINDW(IGCB,0.,TMAX,YMIN,YMAX)
0576    C
0577    C       INITIALIZE IMPORTANT VARIABLES PRIOR TO BEGINNING
0578    C       THE READ DATA & COMPUTE X-Y DATA LOOP.
0579    C
0580    C            IC       = NO. POINTS READ FOR CHANNEL JCN.
0581    C            SUM      = SUM OF Y DATA READ FOR CHANNEL JCN.CN.
0582    C            SUMSQ    = SUM OF Y DATA READ FOR CHANNEL JCN SQUARED.
0583    C            ZMIN     = CURRENT VALUE OF YMIN FOR Y DATA READ FOR CHANNEL JCN.
0584    C            ZMAX     = CURRENT VALUE OF YMAX FOR Y DATA READ FOR CHANNEL JCN.
0585    C
0586          IC = 0
0587          SUM = 0.0
0588          SUMSQ = 0.0
0589          ZMIN = YMAX
0590          ZMAX = YMIN
0591    C
0592    C
0593      540 CONTINUE
0594    C
0595    C       READ ONE RECORD FROM DISK.
0596    C
0597          CALL READF(IDCB,IERR,IBUF,84,ILEN)
0598    C
```

```
0599  C       IF READ SUCCESSFUL GO CHECK ON END OF FILE WORD.
0600  C
0601        IF(IERR .GE. 0)GO TO 560
0602  C
0603  C       ERROR ON READF. TERMINATE PLOTM.
0604  C
0605        WRITE(LU,550)NAME,IERR
0606    550 FORMAT(" ERROR STOP. READF ERROR ON FILE ",3A2,". IERR = ",I5)
0607        CALL EXEC(6)
0608  C
0609    560 CONTINUE
0610  C
0611  C       IF END OF FILE NOT FOUND YET GO TO SECTION
0612  C       OF PLOTM THAT ISOLATES CHANNEL JCN DATA & COMPUTES
0613  C       & PLOTS X-Y DATA.
0614  C
0615        IF(ILEN .GE. 0)GO TO 720
0616  C
0617  C
0618  C       END OF FILE FOUND ON THIS DISK FILE.
0619  C       THIS IMPLIES ALL X-Y DATA FOR CHANNEL JCN HAS
0620  C       BEEN PLOTTED. NOW FINISH TITLE INFORMATION ON
0621  C       PLOT AND DECIDE WHAT TO DO NEXT.
0622  C
0623    570 CONTINUE
0624  C
0625  C       REDFINE VIEWPORT AND WINDOW TO THE PHYSICAL LIMITS OF THE
0626  C       SELECTED PLOTTING DEVICE IN PREPARATION FOR COMPLETING
0627  C       PLOT TITLE AND LABEL INFORMATION.
0628  C
0629        IF(ID .EQ. 1)CALL VIEWP(IGCB,0.0,200.0,0.0,100.0)
0630        IF(ID .EQ. 1)CALL WINDW(IGCB,0.0,200.0,0.0,100.0)
0631        IF(ID  EQ. 2)CALL VIEWP(IGCB,0.0,152.0,0.0,100.0)
0632        IF(ID .EQ. 2)CALL WINDW(IGCB,0.0,152.0,0.0,100.0)
0633        IF(ID .EQ. 3)CALL VIEWP(IGCB,0.0,128.0,0.0,100.0)
0634        IF(ID .EQ. 3)CALL WINDW(IGCB,0.0,128.0,0.0,100.0)
0635  C
0636  C       RE-SELECT PEN 1 FOR THE HP 9872B 4-COLOR PEN PLOTTER.
0637  C
0638        IF(ID .EQ. 2)CALL PEN(IGCB,1)
0639  C
0640  C       DEFINE X NORMALIZED COORDINATE FOR THE SELECTED PLOTTING DEVICE
0641  C       IN PREPARATION FOR WRITING MINIMUM & MAXIMUM Y DATA ON PLOT.
0642  C
0643        XP = 70.0
0644        IF(ID .EQ. 2)XP = 55.0
0645        IF(ID .EQ. 3)XP = 48.0
0646  C
0647  C       MOVE PEN TO NORMALIZED X-Y COORDINATE IN PREPARATION
0648  C       FOR WRITING MAXIMUM Y VALUE ON PLOT. TURN ON TEXT MODE
0649  C       FOR NEXT WRITE STATEMENT.
0650  C
0651        CALL MOVE(IGCB,XP,97.0)
0652        CALL LABEL(IGCB)
0653  C
0654  C       WRITE MAXIMUM Y VALUE FOR CHANNEL JCN ON PLOT.
0655  C
0656        WRITE(LUG,580)ZMAX
0657    580 FORMAT("MAX. Y VALUE IS ",F8.2)
0658  C
```

```
C       MOVE PEN TO NORMALIZED X-Y COORDINATE IN PREPARATION
C       FOR WRITING TIME AT WHICH MAXIMUM Y OCCURRED. TURN ON
C       TEXT MODE FOR NEXT WRITE STATEMENT.
C
      CALL MOVE(IGCB,XP,94.0)
      CALL LABEL(IGCB)
C
C       WRITE TIME AT WHICH YMAX OCCURRED IF TIME IS IN SECONDS.
C
      IF(IFLAG .EQ. 1)WRITE(LUG,590)TIMEX
  590 FORMAT("OCCURRING AT: ",F6.2," SEC.")
C
C       WRITE TIME AT WHICH YMAX OCCURRED IF TIME IS IN MINUTES.
C
      IF(IFLAG .EQ. 2)WRITE(LUG,600)TIMEX
  600 FORMAT("OCCURRING AT: ",F6.2," MIN.")
C
C       MOVE PEN TO NORMALIZED X-Y COORDINATES IN PREPARATION
C       FOR WRITING MINIMUM Y VALUE ON PLOT. TURN ON TEXT MODE
C       FOR NEXT WRITE STATEMENT.
C
      CALL MOVE(IGCB,XP,91.0)
      CALL LABEL(IGCB)
C
C       WRITE MINIMUM Y VALUE FOR CHANNEL JCN ON PLOT.
C
      WRITE(LUG,610)ZMIN
  610 FORMAT("MIN. Y VALUE IS ",F8.2)
C
C       MOVE PEN TO NORMALIZED X-Y COORDINATE IN PREPARATION
C       FOR WRITING TIME AT WHICH MINIMUM Y OCCURRED. TURN TEXT
C       MODE ON FOR NEXT WRITE STATEMENT.
C
      CALL MOVE(IGCB,XP,88.0)
      CALL LABEL(IGCB)
C
C       WRITE TIME AT WHICH YMIN OCCURRED IF TIME IS IN SECONDS.
C
      IF(IFLAG .EQ. 1)WRITE(LUG,620)TIMEN
  620 FORMAT("OCCURRING AT: ",F6.2," SEC.")
C
C       WRITE TIME AT WHICH YMIN OCCURRED IF TIME IS IN MINUTES.
C
      IF(IFLAG .EQ. 2)WRITE(LUG,630)TIMEN
  630 FORMAT("OCCURRING AT: ",F6.2," MIN.")
C
C       CALCULATE THE MEAN AND STANDARD DEVIATION OF Y DATA
C       FOR CHANNEL JCN IN PREPARATION FOR WRITING IT ON PLOT.
C
      YMEAN = SUM/FLOAT(IC)
      YVAR = (FLOAT(IC)*SUMSQ - SUM**2)/(FLOAT(IC)*FLOAT(IC-1))
      YSTD = 0.0
      IF(YVAR .GE. 0.0)YSTD = SQRT(YVAR)
C
C       DEFINE X NORMALIZED COORDINATE FOR THE SELECTED PLOTTING DEVICE
C       IN PREPARATION FOR WRITING MEAN, STANDARD DEVIATION, SENSOR
C       TYPE, AND SENSOR COMMENT DATA ON PLOT.
C
      XP = 130.0
      IF(ID .EQ. 2)XP = 105.0
```

```
      IF(ID .EQ. 3)XP = 87.0
C
C       MOVE PEN TO NORMALIZED X-Y COORDINATE IN PREPARATION
C       FOR WRITING MEAN OF Y DATA ON PLOT. TURN ON TEXT MODE
C       FOR NEXT WRITE STATEMENT.
C
      CALL MOVE(IGCB,XP,97.0)
      CALL LABEL(IGCB)
C
C       WRITE MEAN OF Y DATA ON PLOT.
C
      WRITE(LUG,640)YMEAN
  640 FORMAT("MEAN OF Y  =",F8.2)
C
C       MOVE PEN TO NORMALIZED X-Y COORDINATE IN PREPARATION
C       FOR WRITING STANDARD DEVIATION OF Y DATA ON PLOT.
C       TURN TEXT MODE ON FOR NEXT WRITE STATEMENT.
C
      CALL MOVE(IGCB,XP,94.0)
      CALL LABEL(IGCB)
C
C       WRITE STANDARD DEVIATION OF Y DATA ON PLOT.
C
      WRITE(LUG,650)YSTD
  650 FORMAT("ST. DEV. Y =",F8.2)
C
C       MOVE PEN TO NORMALIZED X-Y COORDINATE IN PREPARATION
C       FOR WRITING SENSOR TYPE ON PLOT. TURN ON TEXT MODE FOR
C       NEXT WRITE STATEMENT.
C
      CALL MOVE(IGCB,XP,88.0)
      CALL LABEL(IGCB)
C
C       WRITE SENSOR NAME FOR CHANNEL JCN ON PLOT.
C
      WRITE(LUG,660)(ITYPE(JJ,JCN+1),JJ=1,6)
  660 FORMAT(6A2)
C
C       MOVE PEN TO NORMALIZED X-Y COORDINATE IN PRAPARATION
C       FOR WRITING A COMMENT ON PLOT. TURN ON TEXT MODE FOR
C       NEXT WRITE STATEMENT.
C
      CALL MOVE(IGCB,XP,85.0)
      CALL LABEL(IGCB)
C
C       WRITE COMMENT FOR CHANNEL JCN ON PLOT.
C
      WRITE(LUG,670)(ICOMM(JJ,JCN+1),JJ=1,10)
  670 FORMAT(10A2)
C
C       SEND PLOT TO DEVICE.
C
      CALL XMIT(IGCB)
C
C       INCREMENT CHANNEL COUNT BY +1.
C
      JCN = JCN+1
C
C       IF THE PLOT DEVICE IS THE HP 2647A GRAPHICS TERMINAL,
C       GET A HARDCOPY FROM THE TEKTRONIX 4631 HARDCOPY UNIT
```

```
0779  C       AND TURN THE GRAPHICS MODE OFF & THE ALPHANUMERIC MODE
0780  C       BACK ON.
0781  C
0782        IF(ID .EQ. 1)CALL HCOPY(LU2647)
0783        IF(ID .EQ. 1)CALL NGRAF(LU2647)
0784  C
0785  C       GO GET NEXT USER DIRECTIVE IF PLOTS ARE NOT TO BE
0786  C       GENERATED CONSECUTIVELY.
0787  C
0788        IF(IGO .EQ. 1)GO TO 770
0789  C
0790  C       GO GET NEXT USER DIRECTIVE IF NEXT CHANNEL NUMBER TO PLOT
0791  C       IS GREATER THAN THE MAXIMUM NUMBER OF CHANNELS DEFINED
0792  C       ACTIVE FOR THIS TEST BY THE VARIABLE MAXCN.
0793  C
0794        IF(JCN .GT. MAXCN)GO TO 770
0795  C
0796  C       PLOT COMPLETE. IF PLOT DEVICE IS HP 9872B 4-COLOR PEN PLOTTER
0797  C       STORE PEN IN HOLDER, THEN RAISE AND MOVE PEN TO UPPER-RIGHT
0798  C       HAND CORNER OF THE PLATTEN.
0799  C
0800        IF(ID .EQ. 2)CALL PEN(IGCB,0)
0801        IF(ID .EQ. 2)CALL PLOTR(IGCB,ID,0)
0802  C
0803  C       IF PLOT DEVICE IS HP 9872B 4-COLOR PEN PLOTTER, ASK USER
0804  C       TO CHANGE PAPER BEFORE CONTINUING.
0805  C
0806        IF(ID .EQ. 2)WRITE(LU,680)
0807    680 FORMAT(" CHANGE PAPER ON HP9872B PLOTTER. ENTER <CR> TO "
0808       .        "CONTINUE.")
0809  C
0810  C       USER MUST TYPE <CR> TO CONTINUE IF DEVICE IS HP 9872B.
0811  C
0812        IF(ID .EQ. 2)READ(LU,*)IGO
0813  C
0814  C       REWIND DISK FILE FOR NEXT PLOT.
0815  C
0816        CALL RWNDF(IDCB,IERR)
0817  C
0818  C       IF REWIND SUCCESSFUL GO READ HEADER RECORD.
0819  C
0820        IF(IERR .GE. 0)GO TO 700
0821  C
0822  C       ERROR ON REWIND, TERMINATE PLOTM.
0823  C
0824        WRITE(LU,690)NAME,IERR
0825    690 FORMAT(" ERROR STOP. RWNDF ERROR ON FILE ",3A2,". IERR = ",I5)
0826        CALL EXEC(6)
0827  C
0828    700 CONTINUE
0829  C
0830  C       READ HEADER RECORD OF DISK FILE IN ORDER TO POSITION DISK
0831  C       AT FIRST RECORD OF DATA FOR NEXT PLOT.
0832  C
0833        CALL READF(IDCB,IERR,IBUF,84)
0834  C
0835  C       IF READF SUCCESSFUL GO START NEXT PLOT.
0836  C
0837        IF(IERR .GE. 0)GO TO 710
0838  C
```

```
0839  C       ERROR ON READF, TERMINATE PLOTM.
0840  C
0841        WRITE(LU,550)NAME,IERR
0842        CALL EXEC(6)
0843  C
0844    710 CONTINUE
0845  C
0846  C       GO START NEXT PLOT. SPECIFICALLY, GO BACK & AUTOSCALE NEXT
0847  C       CHANNEL'S Y DATA OR USE DEFAULT DATA & START PROCESS ALL
0848  C       OVER AGAIN.
0849  C
0850        GO TO 420
0851  C
0852    720 CONTINUE
0853  C
0854  C       PRIMARY X-Y COMPUTATIONAL AND X-Y POINT PLOTTING LOOP
0855  C       BEGINS HERE.
0856  C
0857  C       IF PLOTTING DEVICE IS HP 9872B 4-COLOR PEN PLOTTER
0858  C       SELECT PEN #3 TO DRAW X-Y POINTS.
0859  C
0860        IF(ID .EQ. 2)CALL PEN(IGCB,3)
0861  C
0862  C       LOOP THRU DATA IN IBUF UNTIL CHANNEL JCN IS FOUND (ICN = JCN).
0863  C       USE DATA IN IWORD1, IWORD2, IWORD3, & IWORD4 TO DETERMINE
0864  C       X-Y DATA.
0865  C
0866        DO 740 KK = 9,77,4
0867        IWORD1 = IBUF(KK)
0868        IWORD2 = IBUF(KK+1)
0869        IWORD3 = IBUF(KK+2)
0870        IWORD4 = IBUF(KK+3)
0871  C
0872  C       IF ALL FOUR WORDS = 0 A BASIC ASSUMPTION ABOUT THE DATA
0873  C       HAS BEEN VIOLATED. GO TERMINATE PROGRAM PLOTM.
0874  C
0875        IF(IWORD1 .EQ. 0 .AND. IWORD2 .EQ. 0 .AND. IWORD3 .EQ. 0
0876       +    .AND. IWORD4 .EQ. 0)GO TO 750
0877  C
0878  C       COMPUTE CHANNEL NUMBER FROM CURRENT POINTER TO IBUF.
0879  C
0880        ICN = IAND(IWORD1,77B)
0881  C
0882  C       IF COMPUTED CHANNEL NUMBER GREATER THAN CURRENT CHANNEL NUMBER(JCN)
0883  C       GO READ NEXT RECORD FROM DISK.
0884  C
0885        IF(ICN .GT. JCN)GO TO 540
0886  C
0887  C       IF COMPUTED CHANNEL NUMBER NOT EQUAL TO CURRENT CHANNEL NUMBER(JCN)
0888  C       GO LOOK AT NEXT 4 WORDS OF IBUF.
0889  C
0890        IF(ICN .NE. JCN)GO TO 740
0891  C
0892  C
0893  C       COMPUTE DIGITS FOR VOLTAGE CALCULATION.
0894  C
0895        ID1 = IAND(IWORD1,3008)/64
0896        ID2 = IAND(IWORD2,3608)/16
0897        ID3 = IAND(IWORD2,17B)
0898        ID4 = IAND(IWORD3,3608)/16
```

```
0899          ID5 = IAND(IWORD3,17B)
0900    C
0901    C       DETERMINE SIGN OF VOLTAGE.
0902    C
0903          ISIGN = IAND(IWORD4,200B)/128
0904          IF(ISIGN .EQ. 0)ISIGN = -1
0905    C
0906    C       COMPUTE VOLTAGE FOR THIS TIME POINT. CURRENT CHANNEL IS JCN.
0907    C
0908          VOLTS = FLOAT(ID1) + FLOAT(ID2)*0.1 + FLOAT(ID3)*0.01
0909         .        + FLOAT(ID4)*0.001 + FLOAT(ID5)*0.0001
0910    C
0911    C       DETERMINE IF VOLTAGE IS OUT OF RANGE.
0912    C
0913          RANGE = IAND(IWORD4,10,8)/64
0914          IF(RANGE .EQ. 1)VOLTS = 3.9999
0915    C
0916    C       COMPUTE Y POINT WITH CORRECT SIGN.
0917    C
0918          Y = ISIGN*VOLTS
0919    C
0920    C       UPDATE Y POINT IF ENGINEERING PLOT WAS SELECTED.
0921    C
0922          IF(KPLOT .EQ. 2)Y = GAIN(JCN+1)*Y + OFFSET(JCN+1)
0923    C
0924    C       GET CURRENT HRS., MIN., SEC. TO COMPUTE X POINT.
0925    C
0926          IHR = 10*IAND(IBUF(5),360B)/16 + IAND(IBUF(5),17B)
0927          IMIN = 10*IAND(IBUF(6),360B)/16 + IAND(IBUF(6),17B)
0928          ISEC = 10*IAND(IBUF(7),360B)/16 + IAND(IBUF(7),17B)
0929    C
0930    C       CONVERT CURRENT TIME TO ELAPSED SECONDS.
0931    C
0932          TIMEP = FLOAT(IHR)*3600. + FLOAT(IMIN)*60. + FLOAT(ISEC)
0933          X = TIMEP
0934    C
0935    C       TIME IS IN MINUTES SO CONVERT ELAPSED TIME TO MINUTES.
0936    C
0937          IF(IFLAG .EQ. 2)X = X/60.0
0938    C
0939    C       IF ELAPSED TIME IS LESS THAN USER SELECTED START TIME, DO NOT
0940    C       PLOT OR ACCUMULATE THIS POINT. GO POINT EXT  CORD FROM DISK.
0941    C
0942          IF(X .LT. TIME1)GO TO 540
0943    C
0944    C       IF ELAPSED TIME IS GREATER THAN THE USER SELECTED
0945    C       STOP TIME, DO NOT PLOT OR ACCUMULATE THIS POINT. GO FINISH PLOT.
0946    C
0947          IF(X .GT. TIME2)GO TO 570
0948    C
0949    C       CONVERT ELAPSED TIME TO ELAPSED TIME FROM START OF PLOT.
0950    C
0951          X = X - TIME1
0952    C
0953    C       MOVE PEN TO X-Y NORMALIZED COORDINATE WITHOUT DRAWING A LINE.
0954    C       THIS MOVES PEN TO FIRST X-Y POINT.
0955    C
0956          IF(IC .EQ. 1)CALL MOVE(IGCB,X,Y)
0957    C
0958    C       DRAW A STRAIGHT LINE FROM PREVIOUS X-Y POINT TO CURRENT X-Y POINT.
```

```
0959  C
0960        CALL DRAWK(IGCB,X,Y)
0961  C
0962  C       INCREMENT ACCUMULATED POINT COUNTER BY +1.
0963  C
0964        IC = IC + 1
0965  C
0966  C       ACCUMULATE THE SUM OF Y & THE SUM OF Y SQUARED FOR LATER
0967  C       STATISTICAL CALCULATIONS.
0968  C
0969        SUM = SUM + Y
0970        SUMSQ = SUMSQ + Y**2
0971  C
0972  C       IF CURRENT Y VALUE IS GREATER THAN YMIN SO FAR, KEEP OLD MINIMUM
0973  C       AND GO CHECK ON YMAX.
0974  C
0975        IF(Y .GT. ZMIN)GO TO 730
0976  C
0977  C       UPDATE NEW Y MINIMUM.
0978  C
0979        ZMIN = Y
0980  C
0981  C       UPDATE TIME AT WHICH YMIN OCCURRED.
0982  C
0983        TIMEN = X
0984  C
0985     730 CONTINUE
0986  C
0987  C       IF CURRENT Y VALUE IS LESS THAN YMAX SO FAR, KEEP OLD MAXIMUM
0988  C       AND GO READ NEXT RECORD ON DISK.
0989  C
0990        IF(Y .LT. ZMAX)GO TO 540
0991  C
0992  C       UPDATE NEW YMAX.
0993  C
0994        ZMAX = Y
0995  C
0996  C       UPDATE TIME AT WHICH YMAX OCCURRED.
0997  C
0998        TIMEX = X
0999  C
1000  C       GO READ NEXT RECORD FROM DISK.
1001  C
1002        GO TO 540
1003     740 CONTINUE
1004        GO TO 540
1005  C
1006  C
1007  C       WRITE ERROR MESSAGE AND TERMINATE PROGRAM PLOTM
1008  C
1009     750 CONTINUE
1010        WRITE(LU,760)JCN,ICN
1011     760 FORMAT(" ERROR STOP. DATA DOES NOT CONFORM TO SPECIFICATION. "
1012       .        " JCN, ICN = ",2I6)
1013        CALL EXEC(6)
1014  C
1015     770 CONTINUE
1016  C
1017  C       IF PLOTTING DEVICE IS AN HP 2647A GRAPHICS TERMINAL, TURN
1018  C       OFF THE GRAPHICS MODE AND TURN THE ALPHANUMERIC MODE BACK ON.
```

```
C
      IF(ID .EQ. 1)CALL NGRAF(LU2647)
C
C       WANT TO PLOT ANOTHER CHANNEL?
C
      KANS = 1HY
      WRITE(LU,780)
  780 FORMAT(" DO YOU WISH TO PLOT ANOTHER CHANNEL?  (Y/N): _")
      READ(LU,200)KANS
      IF(KANS .EQ. 1HN)GO TO 820
C
C       SAME DISK FILE?
C
      KANS = 1HY
      WRITE(LU,790)
  790 FORMAT(" SAME DISK FILE?  (Y/N): _")
      READ(LU,200)KANS
C
C       NO, DIFFERENT DISK FILE, GO BACK AND ENTER NEW DISK FILE NAME.
C
      IF(KANS .EQ. 1HN)GO TO 100
C
C       SAME DISK FILE, REWIND IT BEFORE PROCEEDING.
C
      CALL RWNDF(IDCB,IERR)
C
C       IF REWIND SUCCESSFUL GO READ HEADER RECORD.
C
      IF(IERR .GE. 0)GO TO 800
C
C       ERROR ON REWIND  TERMINATE PLOTM.
C
      WRITE(LU,690)NAME,IERR
      CALL EXEC(6)
C
  800 CONTINUE
C
C       READ HEADER RECORD IN ORDER TO POSITION DISK FILE AT FIRST RECORD
C       OF DATA.
C
      CALL READF(IDCB,IERR,IBUF,84)
C
C       IF READF SUCCESSFUL GO GET USER INPUT AND START AGAIN!
C
      IF(IERR .GE. 0)GO TO 220
C
C       ERROR ON READF, TERMINATE PLOTM.
C
      WRITE(LU,550)NAME,IERR
      CALL EXEC(6)
C
  820 CONTINUE
C
C       PROGRAM PLOTM THRU! PLOTM WILL NOW GRACEFULLY EXIT!
C
      WRITE(LU,830)
  830 FORMAT(" PROGRAM PLOTM TERMINATED. HAVE A GOOD DAY! ")
C
C       IF PLOT DEVICE IS HP 98720 4-COLOR PEN PLOTTER STORE PEN IN HOLDER,
C       THEN RAISE AND MOVE PEN TO UPPER-RIGHT HAND CORNER OF THE PLATTEN.
```

```
1079  C
1080        IF(ID .EQ. 2)CALL PEN(IGCB,0)
1081        IF(ID .EQ. 2)CALL PLOTR(IGCB,ID,0)
1082  C
1083  C       RETURN TO RTE!
1084  C
1085        CALL EXEC(6)
1086        END
1087        BLOCK DATA
1088  C
1089  C       THIS ROUTINE DEFINES THE /CBLOC/, /DBLOC/, AND /FLAGS/ LABELED
1090  C       COMMON BLOCKS.
1091  C
1092        COMMON/CBLOC/IDCB2(144),NAME2(3),ITYPE(6,32),GAIN(32),OFFSET(32),
1093       .            IUNITS(10,32),YMINI(32),YMAXI(32),ICOMM(10,32)
1094        COMMON/DBLOC/IDCB(144),NAME(3)
1095        COMMON/FLAGS/LU,KPLOT,IFLAG,TIME1,TIME2
1096        DATA NAME2/2HTA,2HBL,2HEA/
1097        END
1098        SUBROUTINE LABLE(LAB1,LAB2,NUM)
1099  C
1100  C       THIS ROUTINE MODIFIES ARRAY LAB1 BY ARRAY LAB2.
1101  C          IF LAB2 IS BLANK  LAB1 = LAB1
1102  C          IF LAB2 HAS !     LAB1 = BLANK
1103  C          OTHERWISE         LAB1 = LAB2
1104  C
1105        DIMENSION LAB1(1), LAB2(1)
1106  C
1107        DO 100 K=1,NUM
1108        IF(LAB2(K) .NE. 2H  ) GO TO 110
1109    100 CONTINUE
1110        RETURN
1111    110 IF(LAB2(1) .EQ. 2H! ) GO TO 130
1112        DO 120 K=1,NUM
1113    120 LAB1(K) = LAB2(K)
1114        RETURN
1115    130 DO 140 K=1,NUM
1116    140 LAB1(K) = 2H
1117        RETURN
1118        END
1119        SUBROUTINE BOUND(JCN,YMIN,YMAX)
1120  C
1121  C
1122  C       THIS ROUTINE DETERMINES THE Y-AXIS BOUNDS (YMIN & YMAX) FOR
1123  C       CHANNEL JCN. THIS ROUTINE IS CALLED WHEN THE USER SPECIFIES
1124  C       THAT THE Y-AXIS IS TO BE AUTOSCALED.
1125  C
1126  C       IN ADDITION, THE SEARCH FOR YMIN & YMAX IS ONLY CONDUCTED OVER
1127  C       THE TIME FRAME SELECTED BY THE USER AS THE TIME FRAME FOR THE
1128  C       PLOT.
1129  C
1130        DIMENSION IBUF(100)
1131        COMMON/CBLOC/IDCB2(144),NAME2(3),ITYPE(6,32),GAIN(32),OFFSET(32),
1132       .            IUNITS(10,32),YMINI(32),YMAXI(32),ICOMM(10,32)
1133        COMMON/DBLOC/IDCB(144),NAME(3)
1134        COMMON/FLAGS/LU,KPLOT,IFLAG,TIME1,TIME2
1135        INTEGER RANGE
1136  C
1137  C       WRITE THE USER A MESSAGE ON THE USER TERMINAL INFORMING HIM
1138  C       THAT AUTOSCALING TAKES A LITTLE LONGER THAN PLOTTING WITHOUT
```


C-2

```
1139  C       AUTOSCALING.
1140  C
1141        WRITE(LU,100)
1142    100 FORMAT("PATIENCE! AUTOSCALING Y ARRAY TAKES TIME.")
1143  C
1144  C       INITIALIZE YMAX AND YMIN TO EXTREMELY SMALL AND LARGE VALUES,
1145  C       RESPECTIVELY.
1146  C
1147        YMAX = -1.0E+37
1148        YMIN = +1.0E+37
1149  C
1150  C       REWIND THE DISK FILE AND READ THE HEADER RECORD TO POSITION THE
1151  C       DISK FILE TO THE FIRST DATA RECORD.
1152  C
1153        CALL RWNDF(IDCB)
1154        CALL READF(IDCB,IERR,IBUF,84)
1155        IF(IERR .GE. 0)GO TO 120
1156        WRITE(LU,110)NAME,IERR
1157    110 FORMAT(" ERROR STOP. BOUND READF ERROR. FILE ",3A2,". IERR =",I6)
1158        CALL EXEC(6)
1159  C
1160  C       BEGIN READING LOOP. READ UNTIL AN END OF FILE IS REACHED OR UNTIL
1161  C       THE STOP TIME IS CALCULATED.
1162  C
1163    120 CONTINUE
1164        CALL READF(IDCB,IERR,IBUF,84,ILEN)
1165        IF(IERR .GE. 0)GO TO 130
1166        WRITE(LU,110)NAME,IERR
1167        CALL EXEC(6)
1168    130 CONTINUE
1169  C
1170  C       END OF FILE REACHED. GO CLEAN UP THIS PROGRAM (REWIND DISK) AND
1171  C       RETURN TO PLOTM.
1172  C
1173        IF(ILEN .LT. 0)GO TO 170
1174  C
1175  C       COMPUTATIONAL LOOP SIMILIAR TO ONE IN PLOTM.
1176  C       IBUF IS UNPACKED TO DETERMINE X(TIME) AND Y(VOLTAGE OR ENG. UNIT)
1177  C       VALUES AND YMIN & YMAX ARE CALCULATED.
1178  C       ONLY REAL DIFFERENCE IN THIS CODE IS THAT X-Y PAIRS ARE NOT
1179  C       PLOTTED.
1180  C
1181        DO 140 KK = 9,77,4
1182        IHR = 10*IAND(IBUF(5),360B)/16 + IAND(IBUF(5),17B)
1183        IMIN = 10*IAND(IBUF(6),360B)/16 + IAND(IBUF(6),17B)
1184        ISEC = 10*IAND(IBUF(7),360B)/16 + IAND(IBUF(7),17B)
1185        X = FLOAT(IHR)*3600. + FLOAT(IMIN)*60.0 + FLOAT(ISEC)
1186        IF(IFLAG .EQ. 2)X = X/60.0
1187  C
1188  C     START TIME NOT YET REACHED. GO READ ANOTHER RECORD.
1189  C
1190        IF(X .LT. TIME1)GO TO 120
1191  C
1192  C     STOP TIME REACHED. CLEAN UP AND RETURN TO PLOTM.
1193  C
1194        IF(X .GT. TIME2)GO TO 170
1195  C
1196        IWORD1 = IBUF(KK)
1197        IWORD2 = IBUF(KK+1)
1198        IWORD3 = IBUF(KK+2)
```

```
1199          IWORD4 = IBUF(KK+3)
1200          IF(IWORD1 .EQ. 0 .AND. IWORD2 .EQ. 0 .AND. IWORD3 .EQ. 0
1201         . .AND. IWORD4 .EQ. 0)GO TO 150
1202    C
1203          ICN = IAND(IWORD1,77B)
1204          IF(ICN .GT. JCN)GO TO 120
1205          IF(ICN .NE. JCN)GO TO 140
1206    C
1207          ID1 = IAND(IWORD1,300B)/64
1208          ID2 = IAND(IWORD2,360B)/16
1209          ID3 = IAND(IWORD2,17B)
1210          ID4 = IAND(IWORD3,360B)/16
1211          ID5 = IAND(IWORD3,17B)
1212    C
1213          ISIGN = IAND(IWORD4,200B)/129
1214          IF(ISIGN .EQ. 0)ISIGN = -1
1215          RANGE = IAND(IWORD4,100B)/64
1216          Y = FLOAT(ID1) + FLOAT(ID2)*0.1 + FLOAT(ID3)*0.01
1217         .     + FLOAT(ID4)*0.001 + FLOAT(ID5)*0.0001
1218          IF(RANGE .EQ. 1)Y = 3.9999
1219          Y = ISIGN*Y
1220          IF(KPLOT .EQ. 2)Y = GAIN(JCN+1)*Y + OFFSET(JCN+1)
1221    C
1222          YMIN = AMIN1(YMIN,Y)
1223          YMAX = AMAX1(YMAX,Y)
1224    C
1225    C       GO READ NEXT RECORD.
1226    C
1227          GO TO 120
1228      140 CONTINUE
1229          GO TO 120
1230    C
1231    C       WRITE ERROR MESSAGE AND TERMINATE PLOTM.
1232    C
1233      150 CONTINUE
1234          WRITE(LU,160)JCN,ICN
1235      160 FORMAT(" ERROR STOP. DATA DOES NOT CONFORM TO SPECIFICATION. "
1236         .       " JCN, ICN = ",2I6)
1237          CALL EXEC(6)
1238    C
1239      170 CONTINUE
1240    C
1241    C       BEFORE RETURNING TO PLOTM, REWIND DISK FILE AND READ HEADER
1242    C       RECORD IN ORDER TO POSITION DISK FILE AT FIRST DATA RECORD.
1243    C
1244          CALL RWNDF(IDCB)
1245          CALL READF(IDCB,IERR,IBUF,84)
1246          IF(IERR .GE. 0)GO TO 180
1247          WRITE(LU,110)
1248          CALL EXEC(6)
1249      180 CONTINUE
1250    C
1251          RETURN
1252          END
1253          SUBROUTINE GRAF(LU2647)
1254    C
1255    C       THIS ROUTINE INITIALIZES THE GRAPHIC MODE ON A 2647A AND
1256    C       TURNS OFF THE ALPHANUMERIC DISPLAY
1257    C
1258    C$Y
```

```
1259     100 FORMAT ("ESC@ESC*dacF")
1260     110 FORMAT(1H ,30(/))
1261   CESCZ
1262         WRITE(LU2647,100)
1263         WRITE(LU2647,110)
1264         RETURN
1265         END
1266         SUBROUTINE HCOPY(LU2647)
1267   C
1268   C        THIS ROUTINE MAKES A HARDCOPY OF A 2647A SCREEN
1269   C
1270   CESCY
1271     100 FORMAT ("ESC&p4u5C")
1272   CESCZ
1273         IF(LU2647 .NE. 1)RETURN
1274   C     WRITE (LU2647,100)
1275   C 110 CALL WAIT(12,2)
1276         READ(LU2647,*)IGO
1277         RETURN
1278         END
1279         SUBROUTINE NGRAF(LU2647)
1280   C
1281   C        THIS ROUTINE TERMINATES GRAPHICS MODE ON A 2647A AND TURNS
1282   C        BACK ON THE ALPHANUMERIC DISPLAY
1283   C
1284   CESCY
1285     100 FORMAT ("ESC*ddE")
1286   CESCZ
1287         WRITE (LU2647,100)
1288         RETURN
1289         END
```

```
#DLTBL T=00003 IS ON CR00015 USING 00004 BLKS R=0000

0001  ASMB,R,L,F
0002        HED < GRAPHICS/1000 DEVICE LINK TABLE -- 01/24/81 >
0003        NAM DLTBL,7 GRAPHICS/1000 DEVICE LINK TABLE -- 01/24/81
0004        ENT DPTR
0005        EXT DVG01,DCT01
0006        EXT DVG04,DCT04
0007        EXT DVG02,DCT02
0008  *
0009  *
0010  *  WRITTEN BY ARCHIE JORDAN ON MARCH 29, 1979
0011  *  LAST MODIFIED ON JANUARY 24, 1981
0012  *
0013  *
0014  *  THIS ROUTINE PROVIDES EXTERNALS FOR THE DEVICE SUBROUTINES (DVGXX)
0015  *  AND THE DEVICE COMMAND TABLES (DCTXX).  THESE EXTERNAL REFERENCES
0016  *  WILL ALLOW THESE MODULES TO BE LINKED FROM THE GPS RELOCATABLE
0017  *  LIBRARY (%GPSLB).
0018  *
0019  *
0020  DPTR  DEC 6          TWICE THE LARGEST ID NUMBER
0021  *
0022        DEF DVG01      ID = 1 FOR THE HP 2648A GRAPHICS
0023        DEF DCT01       TERMINAL
0024  *
0025        DEF DVG02      ID = 2 FOR THE HP 9872A 4-COLOR PLOTTER
0026        DEF DCT02
0027  *
0028        DEF DVG04      ID = 4 FOR THE HP 2608A PRINTER
0029        DEF DCT04
0030  *
0031  *
0032        END
```

```
&DUMPM T=00003 IS ON CR00015 USING 00038 BLKS R=0000

0001  FTN4,L
0002        PROGRAM DUMPM
0003  C
0004  C*****************************************************************************
0005  C                                                                            *
0006  C     PROGRAM DUMPM IS A UTILITY PROGRAM FOR LISTING DATA FILES               *
0007  C     GENERATED BY PROGRAM READM IN ONE OF THREE FORMATS:                     *
0008  C               1. OCTAL FORMAT                                              *
0009  C               2. STATISTICAL FORMAT                                        *
0010  C               3. TOTAL (BLOCK BY BLOCK DUMP) FORMAT                        *
0011  C     SUBROUTINES ODUMP, SDUMP, AND CDUMP ARE CALLED TO GENERATE THE         *
0012  C     RESPECTIVE DUMP FORMATS.                                               *
0013  C                                                                            *
0014  C                                                                            *
0015  C     DEVELOPED BY:  ESPEE INC                                               *
0016  C                    EXECUTIVE PLAZA                                         *
0017  C                    SUITE 305                                               *
0018  C                    205/837-8585                                            *
0019  C                                                                            *
0020  C                                                                            *
0021  C*****************************************************************************
0022  C
0023  C
0024        COMMON/BLOCA/IDCB(272),NAME(3),LU,NREC
0025        DIMENSION IBUF(200)
0026  C
0027  C     GET LU OF USER CONSOLE.
0028  C
0029        CALL RMPAR(IBUF)
0030        LU = IBUF(1)
0031        IF(LU .LT. 1)LU = 1
0032        LP = 6
0033  C
0034        WRITE(LU,100)
0035    100 FORMAT("ęHęJ")
0036  C
0037  C     ENTER FILE NAME OF FILE TO DUMP.
0038  C
0039        WRITE(LU,110)
0040    110 FORMAT(" ENTER FILENAME OF FILE TO DUMP: _")
0041        READ(LU,120)NAME
0042    120 FORMAT(3A2)
0043  C
0044  C     ENTER NUMBER OF RECORDS TO READ & DUMP.
0045  C
0046        WRITE(LU,130)
0047    130 FORMAT(" ENTER NO. RECORDS TO READ: _")
0048        READ(LU,*)NREC
0049  C
0050  C     DEFAULT DUMP TYPE IS 2 (STATISTICAL).
0051  C
0052        ITYPE = 2
0053  C
0054  C     SELECT DUMP TYPE.
0055  C
0056        WRITE(LU,140)
0057    140 FORMAT(" SELECT TYPE DUMP  (1-OCTAL, [2-STATISTICS], 3-TOTAL): _")
0058        READ(LU,*)ITYPE
```

```
0059  C
0060        GO TO(150,160,170),ITYPE
0061  C
0062    150 CALL ODUMP
0063        CALL EXEC(6)
0064  C
0065    160 CALL SDUMP
0066        CALL EXEC(6)
0067  C
0068    170 CALL CDUMP
0069        CALL EXEC(6)
0070  C
0071        END
0072        BLOCK DATA
0073  C
0074  C     BLOCK DATA SUBROUTINE DEFINES COMMON BLOCK /BLOCA/.
0075  C
0076        COMMON/BLOCA/IDCB(272),NAME(3),LU,NREC
0077        END
0078        SUBROUTINE ODUMP
0079  C
0080  C
0081  C     SUBROUTINE ODUMP DUMPS A FILE GENERATED BY READM
0082  C     AND SELECTED BY THE USER IN OCTAL FORMAT.
0083  C
0084        COMMON/BLOCA/IDCB(272),NAME(3),LU,NREC
0085        DIMENSION IBUF(200)
0086  C
0087        LP = 6
0088  C
0089        CALL OPEN(IDCB,IERR,NAME)
0090        IF(IERR .GE. 0)GO TO 110
0091        WRITE(LU,100)NAME,IERR
0092    100 FORMAT(" ERROR STOP. OPEN ERROR ON FILE ",3A2," IERR = ",I6)
0093        CALL EXEC(6)
0094    110 CONTINUE
0095  C
0096        DO 170 K = 1,NREC
0097        CALL READF(IDCB,IERR,IBUF,100,ILEN)
0098        IF(IERR .GE. 0)GO TO 130
0099        WRITE(LU,120)IERR,NAME
0100    120 FORMAT(" ERROR STOP. READF ERROR ON FILE ",3A2," IERR = ",I6)
0101        CALL EXEC(6)
0102    130 IF(ILEN .GE. 0)GO TO 150
0103        WRITE(LU,140)ILEN
0104    140 FORMAT(" ERROR STOP. PREMATUTRE END OF FILE. ILEN = ",I6)
0105        CALL EXEC(6)
0106    150 CONTINUE
0107        WRITE(LP,160)K,(IBUF(I),I=1,84)
0108    160 FORMAT(1H ,I8/(1X,10O8))
0109    170 CONTINUE
0110  C
0111        CALL CLOSE(IDCB)
0112        CALL EXEC(6)
0113        RETURN
0114        END
0115        SUBROUTINE CDUMP
0116  C
0117  C     SUBROUTINE CDUMP DUMPS A FILE GENERATED BY READM AND
0118  C     SELECTED BY THE USER IN A COMPLETE BLOCK BY BLOCK, CHANNEL
```

```
0119  C      BY CHANNEL VOLTAGE DUMP.
0120  C
0121         COMMON/BLOCA/IDCB(272),NAME(3),LU,NREC
0122         DIMENSION IBUF(200)
0123  C
0124         LP = 6
0125  C
0126         CALL OPEN(IDCB,IERR,NAME)
0127         IF(IERR .GE. 0)GO TO 110
0128         WRITE(LU,100)NAME,IERR
0129     100 FORMAT(" ERROR STOP. OPEN ERROR ON FILE ",3A2," IERR = ",I6)
0130         CALL EXEC(6)
0131     110 CONTINUE
0132  C
0133         IC = 1
0134         DO 230 K = 1,NREC
0135         CALL READF(IDCB,IERR,IBUF,100,ILEN)
0136         IF(IERR .GE. 0)GO TO 130
0137         WRITE(LU,120)IERR,NAME
0138     120 FORMAT(" ERROR STOP. READF ERROR ON FILE ",3A2," IERR = ",I6)
0139         CALL EXEC(6)
0140     130 IF(ILEN .GE. 0)GO TO 150
0141         WRITE(LU,140)ILEN
0142     140 FORMAT(" ERROR STOP. PREMATUTRE END OF FILE. ILEN = ",I6)
0143         CALL EXEC(6)
0144     150 CONTINUE
0145  C
0146         IF(K .EQ. 1)WRITE(LP,160)NAME,IBUF(1),IBUF(3),IBUF(2),IBUF(4),
0147        .              (IBUF(MM),MM=5,11)
0148     160 FORMAT(1H1," THE HEADER RECORD FOR FILE ",3A2," SHOWS: ",//
0149        .           " SEQUENCE NO. = ",I2,/
0150        .           " DATE         = ",I2,"/",I2,"/",I2,/
0151        .           " START TIME   = ",I2,":",I2,":",I2,/
0152        .           " STOP TIME    = ",I2,":",I2,":",I2,/
0153        .           " NO. CHANNELS = ",I2)
0154  C
0155         IF(K .EQ. 1)GO TO 230
0156         NSEQ = 10*IAND(IBUF(1),360B)/16 + IAND(IBUF(1),17B)
0157         IF(K .EQ. 2)JSEQ=NSEQ
0158         IF(NSEQ .NE. JSEQ)GO TO 240
0159         IDAY = 10*IAND(IBUF(2),360B)/16 + IAND(IBUF(2),17B)
0160         IMON = 10*IAND(IBUF(3),360B)/16 + IAND(IBUF(3),17B)
0161         IYEAR = 10*IAND(IBUF(4),360B)/16 + IAND(IBUF(4),17B)
0162         IHR = 10*IAND(IBUF(5),360B)/16 + IAND(IBUF(5),17B)
0163         IMIN = 10*IAND(IBUF(6),360B)/16 + IAND(IBUF(6),17B)
0164         ISEC = 10*IAND(IBUF(7),360B)/16 + IAND(IBUF(7),17B)
0165         IF(IC .EQ. 1)WRITE(LP,170)
0166     170 FORMAT(1H1)
0167         IK = K-1
0168         WRITE(LP,180)IK,NSEQ,IDAY,IMON,IYEAR,IHR,IMIN,ISEC
0169     180 FORMAT(1X ," BLOCK NUMBER ",I5/
0170        .         1X," SEQUENCE NO. ",I5/
0171        .         1X," DAY          ",I5/
0172        .         1X," MONTH        ",I5/
0173        .         1X," YEAR         ",I5/
0174        .         1X," HOUR         ",I5/
0175        .         1X," MINUTE       ",I5/
0176        .         1X," SECOND       ",I5//
0177        .         1X," CHANNEL NO.     VOLTAGE "/)
0178         DO 200 KK = 9,77,4
```

```
0179          IWORD1 = IBUF(KK)
0180          IWORD2 = IBUF(KK+1)
0181          IWORD3 = IBUF(KK+2)
0182          IWORD4 = IBUF(KK+3)
0183          IF(IWORD1 .EQ. 0 .AND. IWORD2 .EQ. 0 .AND. IWORD3 .EQ. 0 .AND.
0184         . IWORD4 .EQ. 0)GO TO 210
0185          ICN = IAND(IWORD1,77B)
0186          ID1 = IAND(IWORD1,300B)/64
0187          ID2 = IAND(IWORD2,360B)/16
0188          ID3 = IAND(IWORD2,17B)
0189          ID4 = IAND(IWORD3,360B)/16
0190          ID5 = IAND(IWORD3,17B)
0191          ISIGN = IAND(IWORD4,200B)/128
0192          IF(ISIGN .EQ. 0)ISIGN = -1
0193          VOLTS = FLOAT(ID1) + FLOAT(ID2)*0.1 + FLOAT(ID3)*0.01
0194         .        +FLOAT(ID4)*0.001 + FLOAT(ID5)*0.0001
0195          VOLTS = ISIGN*VOLTS
0196          WRITE(LP,190)ICN,VOLTS
0197      190 FORMAT(5X,I3,5X,F12.5)
0198      200 CONTINUE
0199      210 CONTINUE
0200    C
0201          IC = IC + 1
0202          IF(IC .GT. 2)IC = 1
0203          IF(IC .EQ. 1)GO TO 230
0204          WRITE(LP,220)
0205      220 FORMAT(///)
0206      230 CONTINUE
0207    C
0208      240 CONTINUE
0209          CALL CLOSE(IDCB)
0210          CALL EXEC(6)
0211          RETURN
0212          END
0213          SUBROUTINE SDUMP
0214    C
0215    C     SUBROUTINE SDUMP DUMPS A FILE GENERATED BY PROGRAM READM AND
0216    C     SELECTED BY THE USER IN A STATISTICAL SUMMARY FORMAT INCLUDING
0217    C     THE MEAN, VARIANCE, & STANDARD DEVIATION (VOLTAGE) FOR EACH
0218    C     CHANNEL RECORDED FOR THE SELECTED DATA FILE.
0219    C
0220          COMMON/BLOCA/IDCB(272),NAME(3),LU,NREC
0221          DIMENSION IBUF(200)
0222          DIMENSION SUM(32),SUMSQ(32),YMIN(32),YMAX(32),ETIMA(32),ETIMB(32)
0223          INTEGER TIMA(3,32),TIMB(3,32),IC(32)
0224          INTEGER RANGE
0225    C
0226          LP = 6
0227    C
0228          DO 110 I = 1,32
0229          IC(I) = 0
0230          SUM(I) = 0.0
0231          SUMSQ(I) = 0.0
0232          YMIN(I) = 1.0E+37
0233          YMAX(I) = -1.0E+37
0234          ETIMA(I) = 0.0
0235          ETIMB(I) = 0.0
0236          DO 100 J = 1,3
0237          TIMA(J,I) = 0
0238          TIMB(J,I) = 0
```

```
0239     10C CONTINUE
0240     11C CONTINUE
0241   C
0242   C
0243         CALL OPEN(IDCB,IERR,NAME)
0244         IF(IERR .GE. 0)GO TO 130
0245         WRITE(LU,120)NAME,IERR
0246     120 FORMAT(" ERROR STOP. OPEN ERROR ON FILE ",3A2," IERR = ",I6)
0247         CALL EXEC(6)
0248     130 CONTINUE
0249   C
0250   C
0251         DO 200 K = 1,NREC
0252         CALL READF(IDCB,IERR,IBUF,100,ILEN)
0253         IF(IERR .GE. 0)GO TO 150
0254         WRITE(LU,140)IERR,NAME
0255     140 FORMAT(" ERROR STOP. READF ERROR ON FILE ",3A2," IERR = ",I6)
0256         CALL EXEC(6)
0257     150 IF(ILEN .GE. 0)GO TO 160
0258         GO TO 210
0259     160 CONTINUE
0260         IF(K .EQ. 1)WRITE(LP,170)NAME,IBUF(1),IBUF(3),IBUF(2),IBUF(4),
0261        .              (IBUF(MM),MM=5,11)
0262     170 FORMAT(1H1," THE HEADER RECORD FOR FILE ",3A2," SHOWS: ",//
0263        .           "  SEQUENCE NO. = ",I2,/
0264        .           "  DATE         = ",I2,"/",I2,"/",I2,/
0265        .           "  START TIME   = ",I2,":",I2,":",I2,/
0266        .           "  STOP TIME    = ",I2,":",I2,":",I2,/
0267        .           "  NO. CHANNELS = ",I2,////)
0268         IF(K .EQ. 1)MAXCN = IBUF(11)+1
0269         IF(K .EQ. 1)TSTART = FLOAT(IBUF(5))*3600. + FLOAT(IBUF(6))*60.
0270        .                     + FLOAT(IBUF(7))
0271         IF(K .EQ. 1)GO TO 200
0272         NSEQ = 10*IAND(IBUF(1),360B)/16 + IAND(IBUF(1),17B)
0273         IF(K .EQ. 2)JSEQ=NSEQ
0274         IF(NSEQ .NE. JSEQ)GO TO 210
0275         IDAY = 10*IAND(IBUF(2),360B)/16 + IAND(IBUF(2),17B)
0276         IMON = 10*IAND(IBUF(3),360B)/16 + IAND(IBUF(3),17B)
0277         IYEAR = 10*IAND(IBUF(4),360B)/16 + IAND(IBUF(4),17B)
0278         IHR = 10*IAND(IBUF(5),360B)/16 + IAND(IBUF(5),17B)
0279         IMIN = 10*IAND(IBUF(6),360B)/16 + IAND(IBUF(6),17B)
0280         ISEC = 10*IAND(IBUF(7),360B)/16 + IAND(IBUF(7),17B)
0281         DO 190 KK = 9,77,4
0282         IWORD1 = IBUF(KK)
0283         IWORD2 = IBUF(KK+1)
0284         IWORD3 = IBUF(KK+2)
0285         IWORD4 = IBUF(KK+3)
0286         IF(IWORD1 .EQ. 0 .AND. IWORD2 .EQ. 0 .AND. IWORD3 .EQ. 0 .AND.
0287        . IWORD4 .EQ. 0)GO TO 200
0288         ICN = IAND(IWORD1,77B)
0289         ID1 = IAND(IWORD1,300B)/64
0290         ID2 = IAND(IWORD2,360B)/16
0291         ID3 = IAND(IWORD2,17B)
0292         ID4 = IAND(IWORD3,360B)/16
0293         ID5 = IAND(IWORD3,17B)
0294         ISIGN = IAND(IWORD4,200B)/128
0295         IF(ISIGN .EQ. 0)ISIGN = -1
0296         VOLTS = FLOAT(ID1) + FLOAT(ID2)*0.1 + FLOAT(ID3)*0.01
0297        .       +FLOAT(ID4)*0.001 + FLOAT(ID5)*0.0001
0298         RANGE = IAND(IWORD4,100B)/64
```

```
0299          IF(RANGE .EQ. 1)VOLTS = 3.9999
0300          VOLTS = ISIGN*VOLTS
0301          II = ICN + 1
0302          IC(II) = IC(II) + 1
0303          SUM(II) = SUM(II) + VOLTS
0304          SUMSQ(II) = SUMSQ(II) + VOLTS**2
0305          IF(VOLTS .GT. YMIN(II))GO TO 180
0306          YMIN(II) = VOLTS
0307          TIMA(1,II) = IHR
0308          TIMA(2,II) = IMIN
0309          TIMA(3,II) = ISEC
0310          ETIMA(II) = FLOAT(IHR)*3600. + FLOAT(IMIN)*60. + FLOAT(ISEC)
0311      180 CONTINUE
0312          IF(VOLTS .LT. YMAX(II))GO TO 190
0313          YMAX(II) = VOLTS
0314          TIMB(1,II) = IHR
0315          TIMB(2,II) = IMIN
0316          TIMB(3,II) = ISEC
0317          ETIMB(II) = FLOAT(IHR)*3600. + FLOAT(IMIN)*60. + FLOAT(ISEC)
0318      190 CONTINUE
0319      200 CONTINUE
0320      C
0321      210 CONTINUE
0322          CALL CLOSE(IDCB)
0323      C
0324          WRITE(LP,220)
0325      220 FORMAT(4X,"CN",8X,"MEAN",8X,"VAR.",6X,"STD. DEV.",
0326         .       8X,"YMIN",6X,"TIME",6X,"ETIM(MIN)",10X,"YMAX",
0327         .       6X,"TIME",6X,"ETIM(MIN)")
0328          DO 240 I = 1,MAXCN
0329          IK = I-1
0330          AN = FLOAT(IC(I))
0331          YMEAN = SUM(I)/AN
0332          YVAR = (AN*SUMSQ(I) - SUM(I)**2)/(AN*(AN-1.0))
0333          YSTD = 0.0
0334          IF(YVAR .GE. 0.0)YSTD = SQRT(YVAR)
0335          ETIMA(I) = (ETIMA(I) - TSTART)/60.0
0336          ETIMB(I) = (ETIMB(I) - TSTART)/60.
0337          WRITE(LP,230)IK,YMEAN,YVAR,YSTD,YMIN(I),(TIMA(K,I),K=1,3),
0338         .             ETIMA(I),YMAX(I),(TIMB(K,I),K=1,3),ETIMB(I)
0339      230 FORMAT(1X,I5,3F12.3,5X,F10.3,4X,I2,":",I2,":",I2,4X,F8.3,5X,
0340         .       F10.3,4X,I2,":",I2,":",I2,4X,F8.3)
0341      240 CONTINUE
0342          CALL EXEC(6)
0343          RETURN
0344          END
```

APPENDIX B
TRANSFER FILE LISTINGS

READM$

$READM

CFIGM$

$CFIGM

PLOTM$

$PLOTM

DUMPM$

$DUMPM

```
READM$ T=00004 IS ON CR00015 USING 00002 BLKS R=5713

0001   ;OFF,READM,8
0002   ;RU,FTN4,&READM::37 1,%READM::37
0003   ;RU,FTN4,&WAIT::37,1,%WAIT::37
0004   ;RU,ASMB,*RTAPE::37,1,%RTAPE::37
0005   ;RU,ASMB,*CTAPE::37,1,%CTAPE::37
0006   ;RU,LOADR,$READM
0007   ;TR
```

```
$READM T=00004 IS ON CR00015 USING 00001 BLKS R=5713

0001   ECHO
0002   RE,%READM::37
0003   RE,%WAIT::37
0004   RE,%RTAPE::37
0005   RE,%CTAPE::37
0006   END
```

```
CFIGM$ T=00004 IS ON CR00015 USING 00002 BLKS R=5713

0001    :OFF,CFIGM,8
0002    :RU,FTN4,&CFIGM,1,%CFIGM::37
0003    :RU,FTN4,&CNFIG,1,%CNFIG::37
0004    :RU,LOADR,$CFIGM
0005    :TR
```

```
$CFIGM T=00004 IS ON CR00015 USING 00001 BLKS R=5713

0001   ECHO
0002   RE,XCFIGM::37
0003   RE,XCNFIG::37
0004   END
```

```
PLOTM$ T=00004 IS ON CR00015 USING 00001 BLKS R=5713

0001    :OFF,PLOTM,8
0002    :RU,FTN4,&PLOTM,1,%PLOTM::37
0003    :RU,LOADR,$PLOTM
0004    :TR
```

```
$PLOTM T=00004 IS ON CR00015 USING 00002 BLKS R=5713

0001   ECHO
0002   OP,LB
0003   RE,XPLOTM::37
0004   RE,%WAIT::37
0005   RE,XDLTBL::37
0006   SEA,%GPS40
0007   END
```

```
DUMPM$ T=00004 IS ON CR00015 USING 00002 BLKS R=5713

0001   ;OFF,DUMPM.8
0002   :RU,FTN4.&DUMPM.1,%DUMPM::37
0003   :RU,LOADR.$DUMPM
0004   ;TR
```

```
$DUMPM T=00004 IS ON CR00015 USING 00002 BLKS R=5713

0001   ECHO
0002   RE.XDUMPM:·37
0003   END
```

APPENDIX C
SAMPLE CONFIGURATION FILE

CONFIGURATION TABLE FOR FILE TABLE1

CH	SENSOR	GAIN	OFFSET	UNITS	YMIN	YMAX	COMMENT
0	THERMISTOR	10.000	0.000	TEMPERATURE (DEG. C)	-50.000	50.000	THIS IS A 2647A PLOT
1	THERMISTOR	10.000	0.000	TEMPERATURE (DEG. C)	-50.000	50.000	THIS IS A 2647A PLOT
2	THERMISTOR	10.000	0.000	TEMPERATURE (DEG. C)	-50.000	50.000	THIS IS A 2647A PLOT
3	THERMISTOR	10.000	0.000	TEMPERATURE (DEG. C)	-50.000	50.000	THIS IS A 2647A PLOT
4	THERMISTOR	10.000	0.000	TEMPERATURE (DEG. C)	-50.000	50.000	THIS IS A 2647A PLOT
5	THERMISTOR	10.000	0.000	TEMPERATURE (DEG. C)	-50.000	50.000	THIS IS A 2647A PLOT
6	THERMISTOR	10.000	0.000	TEMPERATURE (DEG. C)	-50.000	50.000	THIS IS A 2647A PLOT
7	THERMISTOR	10.000	0.000	TEMPERATURE (DEG. C)	-50.000	50.000	THIS IS A 2647A PLOT
8	THERMISTOR	10.000	0.000	TEMPERATURE (DEG. C)	-50.000	50.000	THIS IS A 2647A PLOT
9	THERMISTOR	10.000	0.000	TEMPERATURE (DEG. C)	-50.000	50.000	THIS IS A 2647A PLOT
10	THERMISTOR	10.000	0.000	TEMPERATURE (DEG. C)	-50.000	50.000	THIS IS A 2647A PLOT
11	THERMISTOR	10.000	0.000	TEMPERATURE (DEG. C)	-50.000	50.000	THIS IS A 2647A PLOT
12	THERMISTOR	10.000	0.000	TEMPERATURE (DEG. C)	-50.000	50.000	THIS IS A 2647A PLOT
13	THERMISTOR	10.000	0.000	TEMPERATURE (DEG. C)	-50.000	50.000	THIS IS A 2647A PLOT
14	THERMISTOR	10.000	0.000	TEMPERATURE (DEG. C)	-50.000	50.000	THIS IS A 2647A PLOT
15	THERMISTOR	10.000	0.000	TEMPERATURE (DEG. C)	-50.000	50.000	THIS IS A 2647A PLOT
16	THERMISTOR	10.000	0.000	TEMPERATURE (DEG. C)	-50.000	50.000	THIS IS A 2647A PLOT
17	THERMISTOR	10.000	0.000	TEMPERATURE (DEG. C)	-50.000	50.000	THIS IS A 2647A PLOT
18	THERMISTOR	10.000	0.000	TEMPERATURE (DEG. C)	-50.000	50.000	THIS IS A 2647A PLOT
19	THERMISTOR	10.000	0.000	TEMPERATURE (DEG. C)	-50.000	50.000	THIS IS A 2647A PLOT
20	THERMISTOR	10.000	0.000	TEMPERATURE (DEG. C)	-50.000	50.000	THIS IS A 2647A PLOT
21	THERMISTOR	10.000	0.000	TEMPERATURE (DEG. C)	-50.000	50.000	THIS IS A 2647A PLOT
22	THERMISTOR	10.000	0.000	TEMPERATURE (DEG. C)	-50.000	50.000	THIS IS A 2647A PLOT
23	THERMISTOR	10.000	0.000	TEMPERATURE (DEG. C)	-50.000	50.000	THIS IS A 2647A PLOT
24	THERMISTOR	10.000	0.000	TEMPERATURE (DEG. C)	-50.000	50.000	THIS IS A 2647A PLOT
25	THERMISTOR	10.000	0.000	TEMPERATURE (DEG. C)	-50.000	50.000	THIS IS A 2647A PLOT
26	THERMISTOR	10.000	0.000	TEMPERATURE (DEG. C)	-50.000	50.000	THIS IS A 2647A PLOT
27	THERMISTOR	10.000	0.000	TEMPERATURE (DEG. C)	-50.000	50.000	THIS IS A 2647A PLOT
28	THERMISTOR	10.000	0.000	TEMPERATURE (DEG. C)	-50.000	50.000	THIS IS A 2647A PLOT
29	THERMISTOR	10.000	0.000	TEMPERATURE (DEG. C)	-50.000	50.000	THIS IS A 2647A PLOT
30	THERMISTOR	10.000	0.000	TEMPERATURE (DEG. C)	-50.000	50.000	THIS IS A 2647A PLOT
31	THERMISTOR	10.000	0.000	TEMPERATURE (DEG. C)	-50.000	50.000	THIS IS A 2647A PLOT

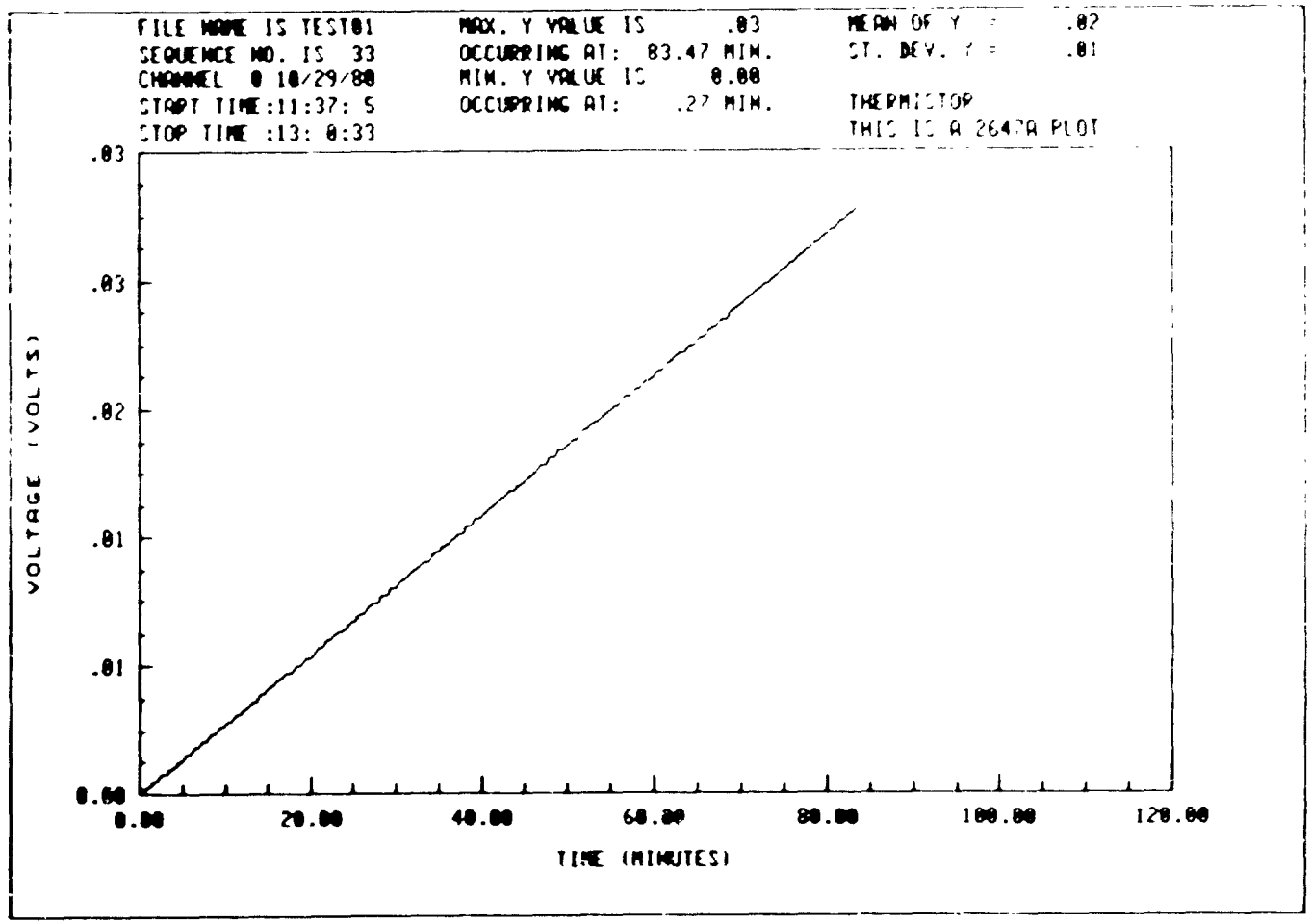
APPENDIX D

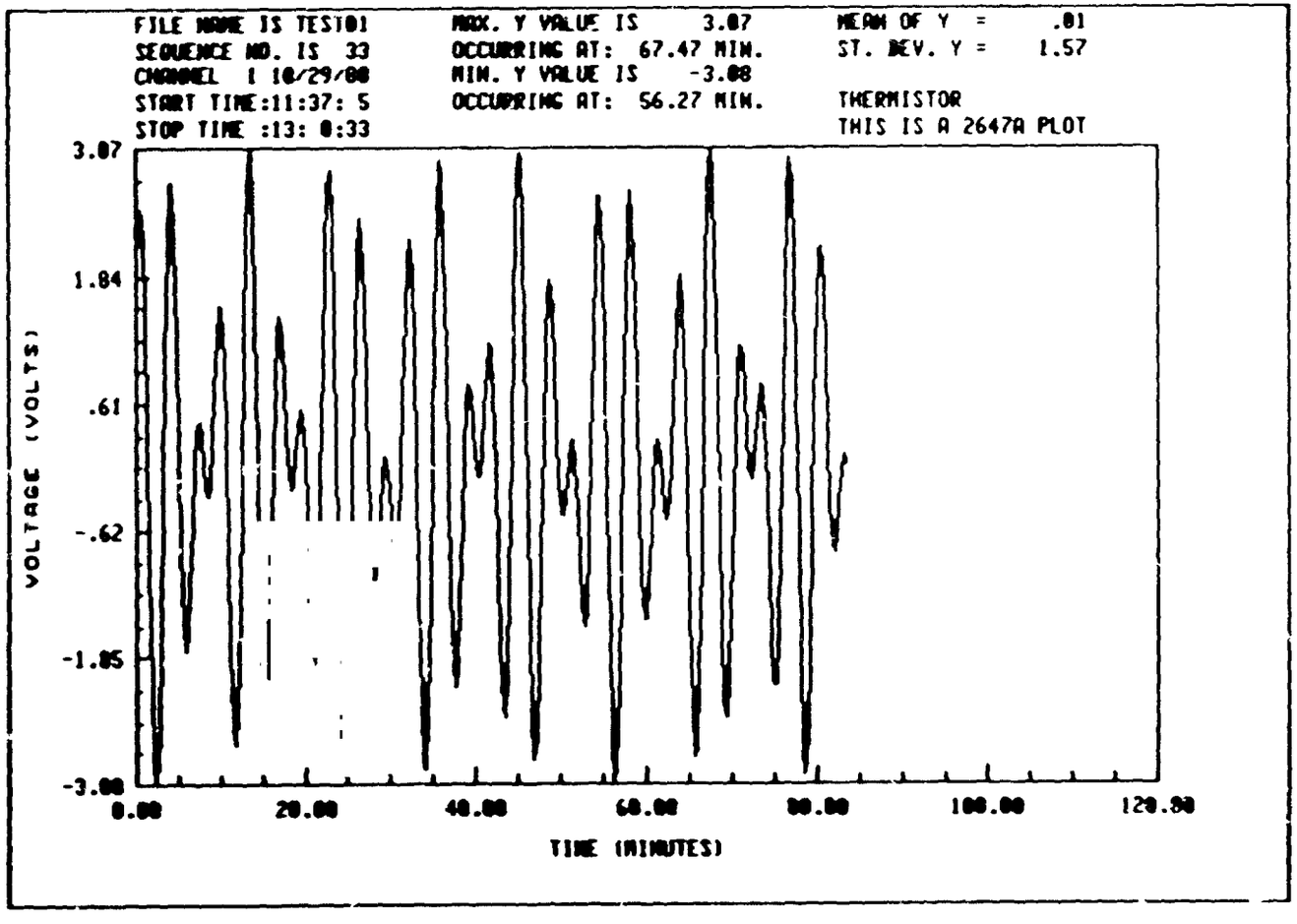
Sample Plots

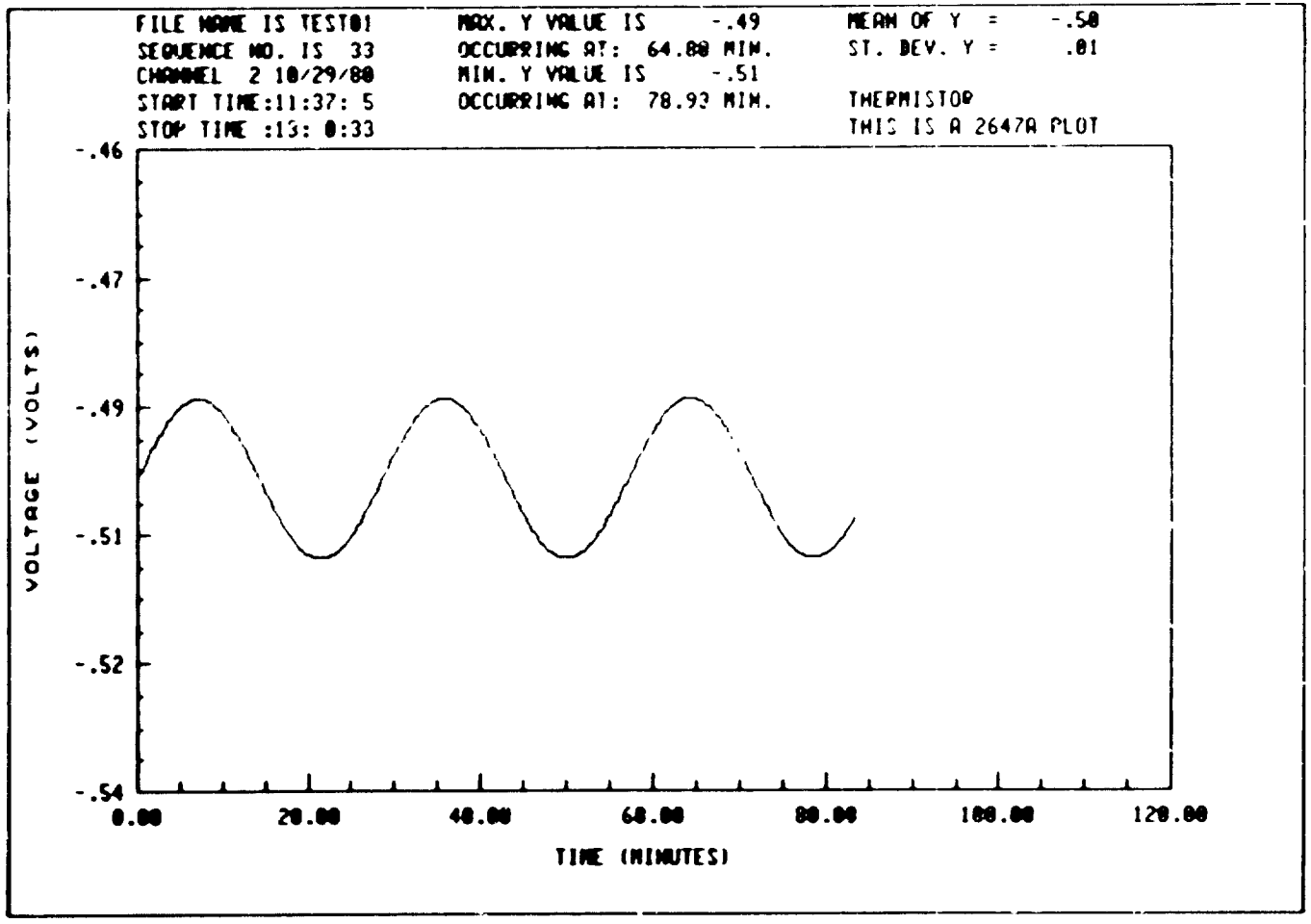
D-1 2647A Plots

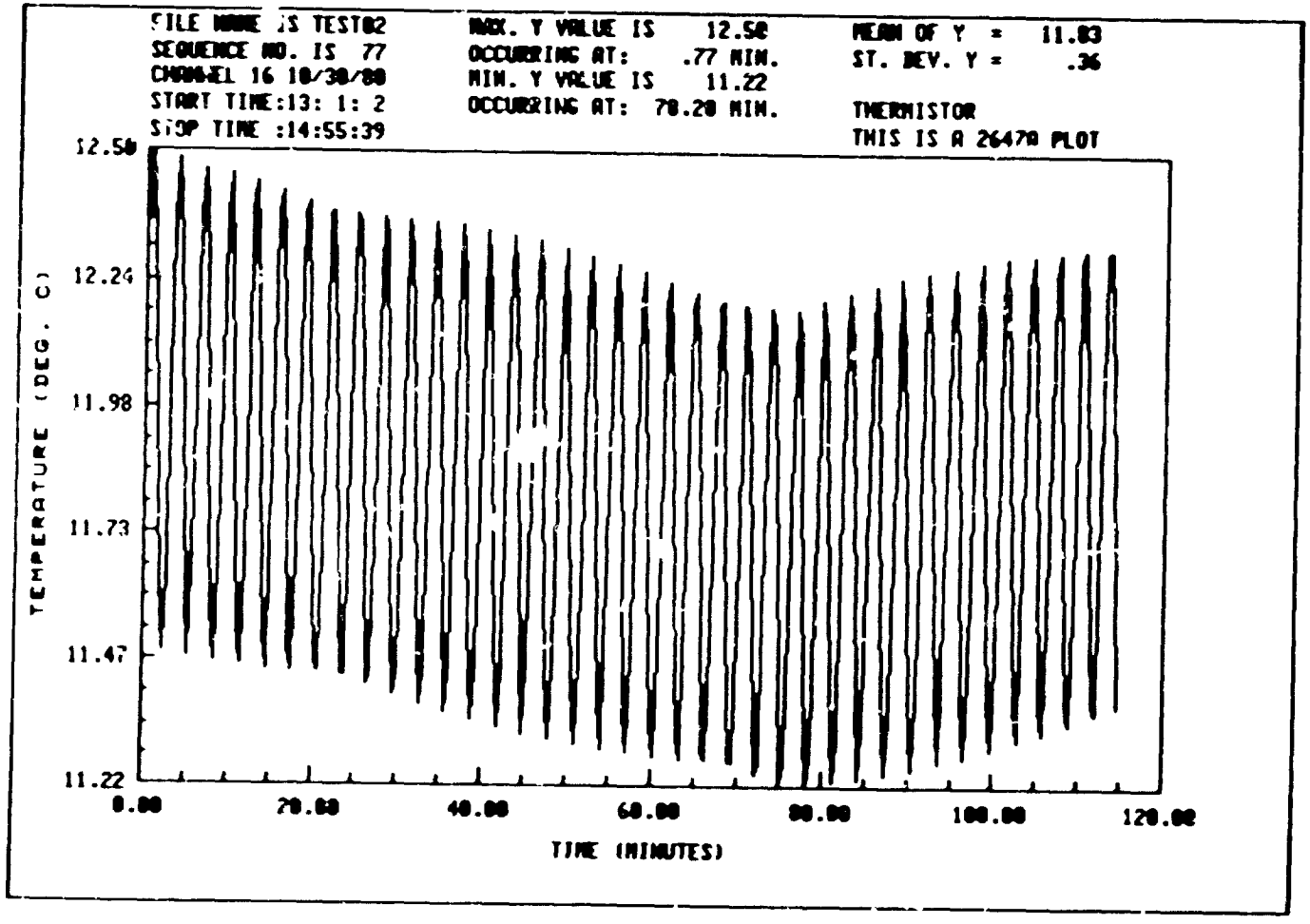
D-2 9872B Plots

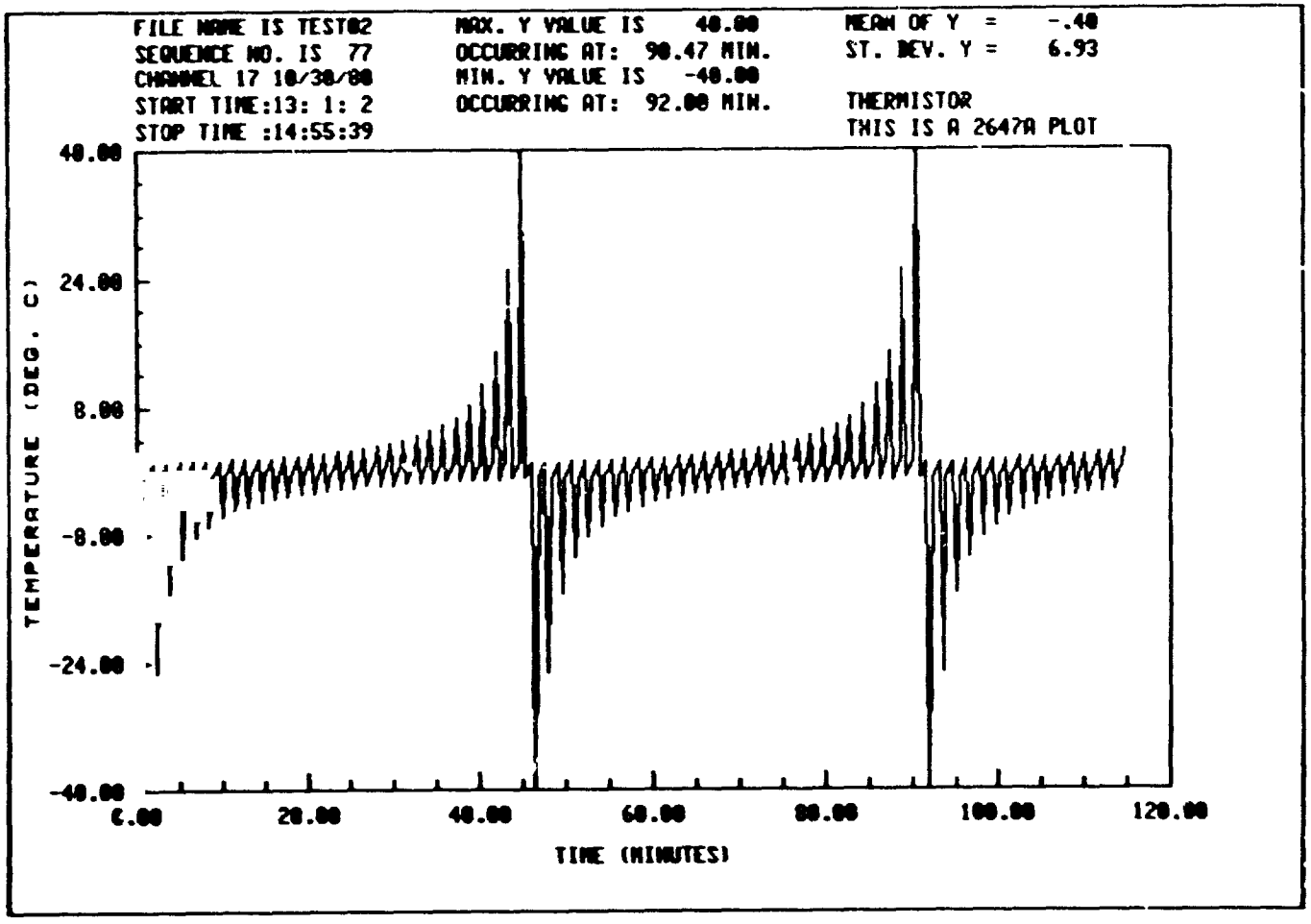
D-3 2608A Plots

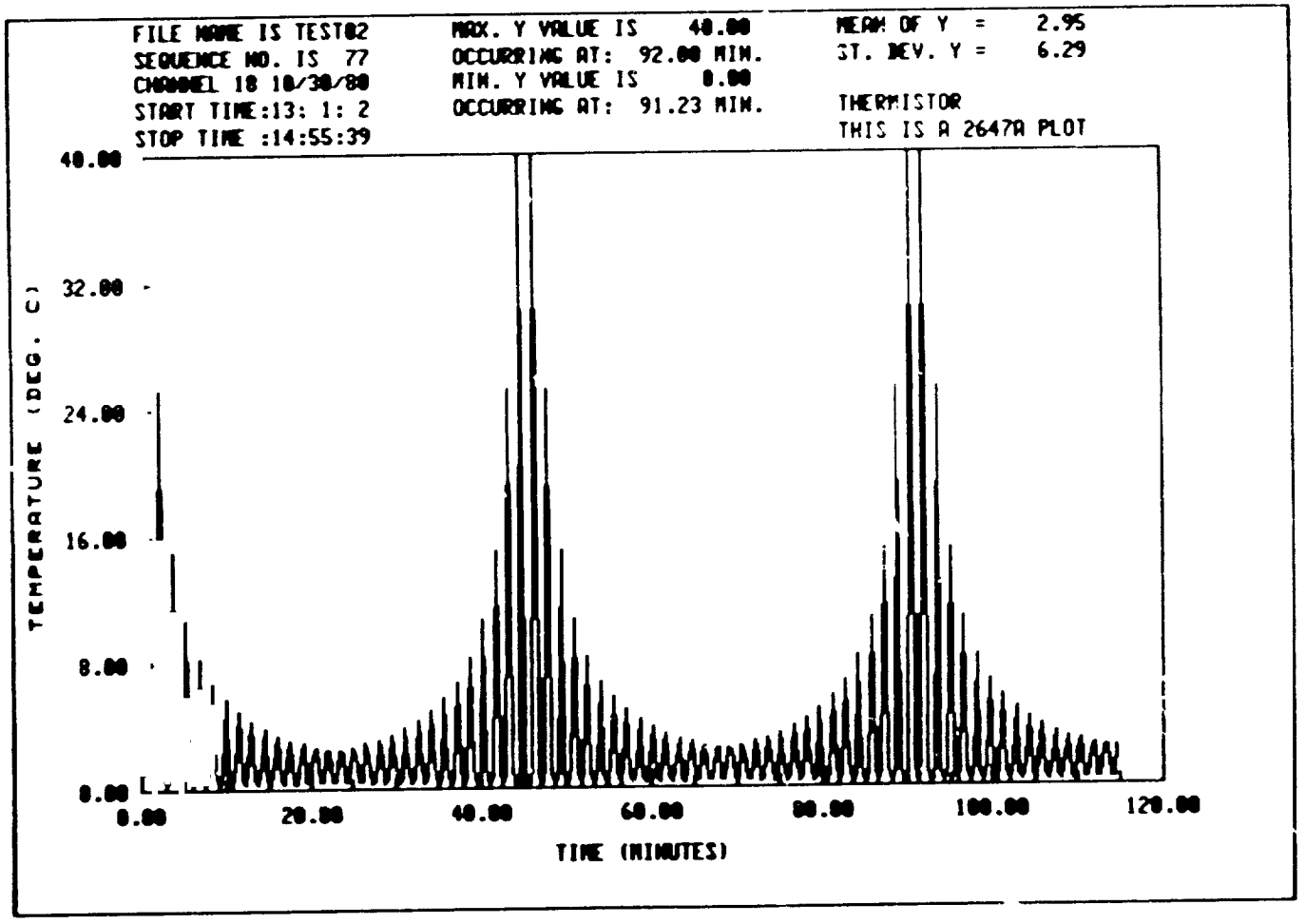

FILE NAME IS TEST01
SEQUENCE NO. IS 33
CHANNEL 0 10/29/80
START TIME:11:37: 5
STOP TIME :13: 0:33
MAX. Y VALUE IS .03
OCCURRING AT: 83.47 MIN.
MIN. Y VALUE IS 0.00
OCCURRING AT: .27 MIN.
MEAN OF Y = .02
ST. DEV. Y = .01
THERMISTOR
THIS IS A 2647A PLOT
VOLTAGE (VOLTS)
.03
.03
.02
.01
.01
0.00
0.00
20.00
40.00
60.00
80.00
100.00
120.00
TIME (MINUTES)

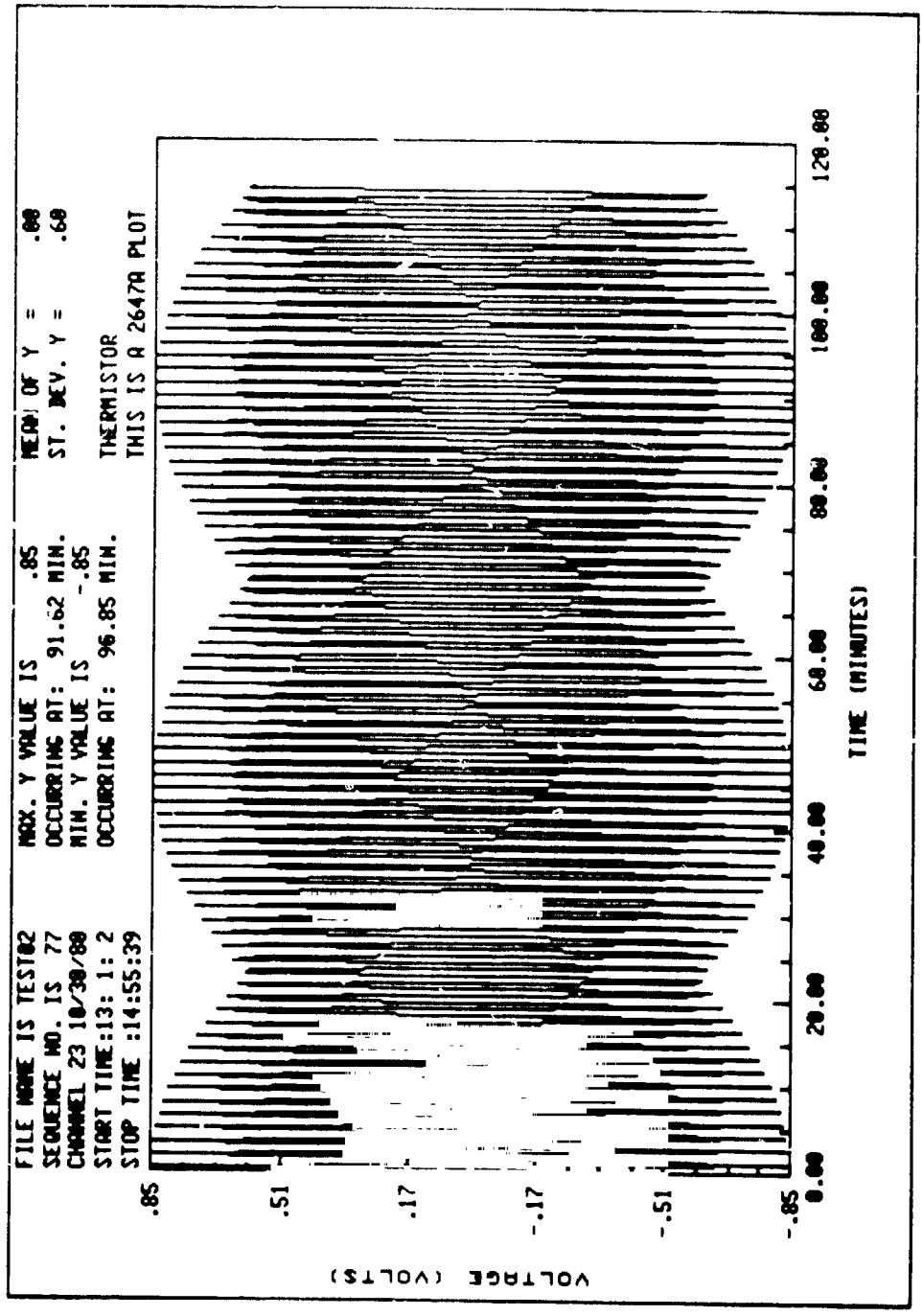

FILE NAME IS TEST01
SEQUENCE NO. IS 33
CHANNEL 1 10/29/80
START TIME:11:37: 5
STOP TIME :13: 0:33
MAX. Y VALUE IS 3.07
OCCURRING AT: 67.47 MIN.
MIN. Y VALUE IS -3.08
OCCURRING AT: 56.27 MIN.
MEAN OF Y = .01
ST. DEV. Y = 1.57
THERMISTOR
THIS IS A 2647A PLOT
3.07
1.84
.61
-.62
-1.85
-3.08
VOLTAGE (VOLTS)
0.00
20.00
40.00
60.00
80.00
100.00
120.00
TIME (MINUTES)

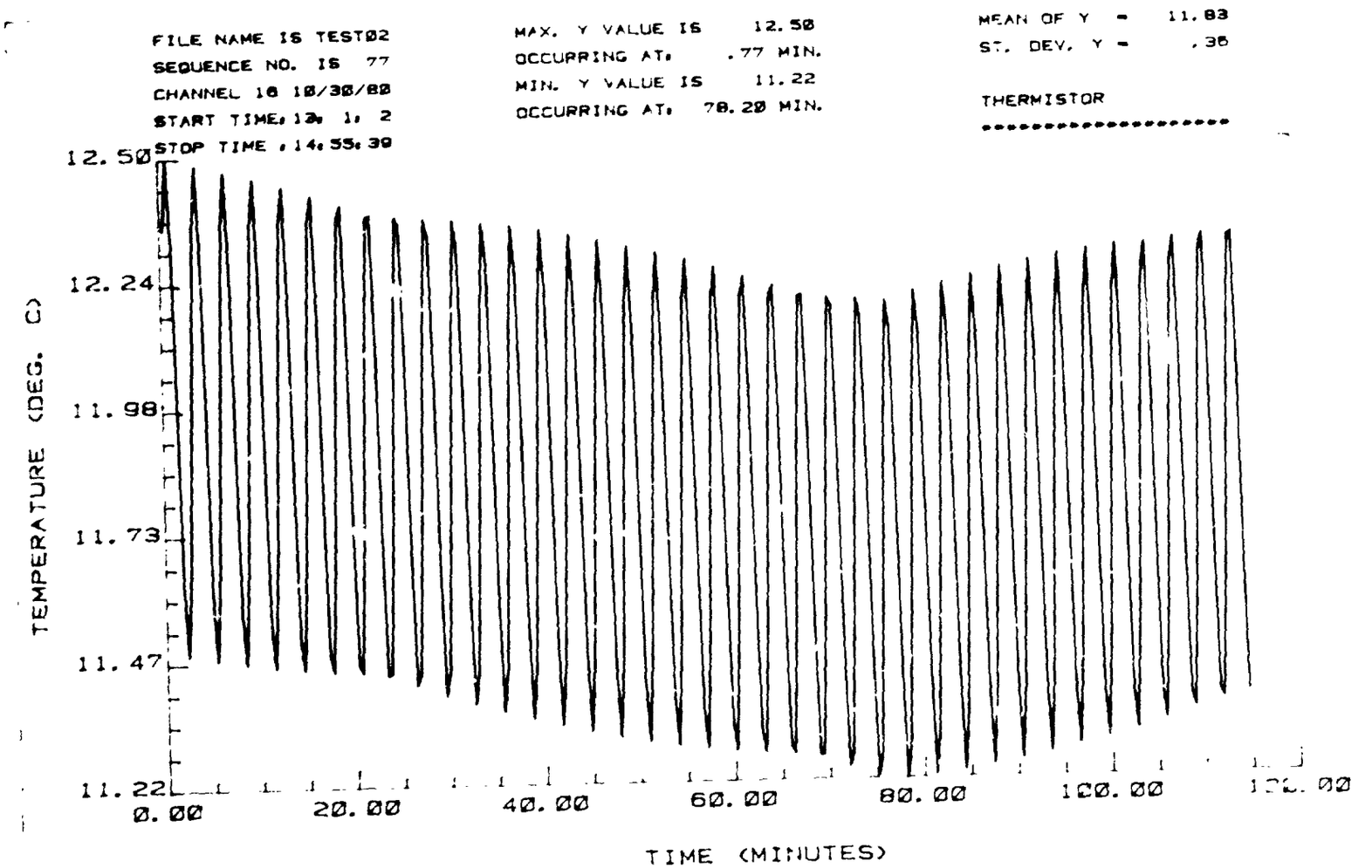

FILE NAME IS TEST01
SEQUENCE NO. IS 33
CHANNEL 2 10/29/80
START TIME:11:37: 5
STOP TIME :13: 0:33
MAX. Y VALUE IS -.49
OCCURRING AT: 64.80 MIN.
MIN. Y VALUE IS -.51
OCCURRING AT: 78.93 MIN.
MEAN OF Y = -.50
ST. DEV. Y = .01
THERMISTOR
THIS IS A 2647A PLOT
-.46
-.47
-.49
-.51
-.52
-.54
VOLTAGE (VOLTS)
0.00
20.00
40.00
60.00
80.00
100.00
120.00
TIME (MINUTES)

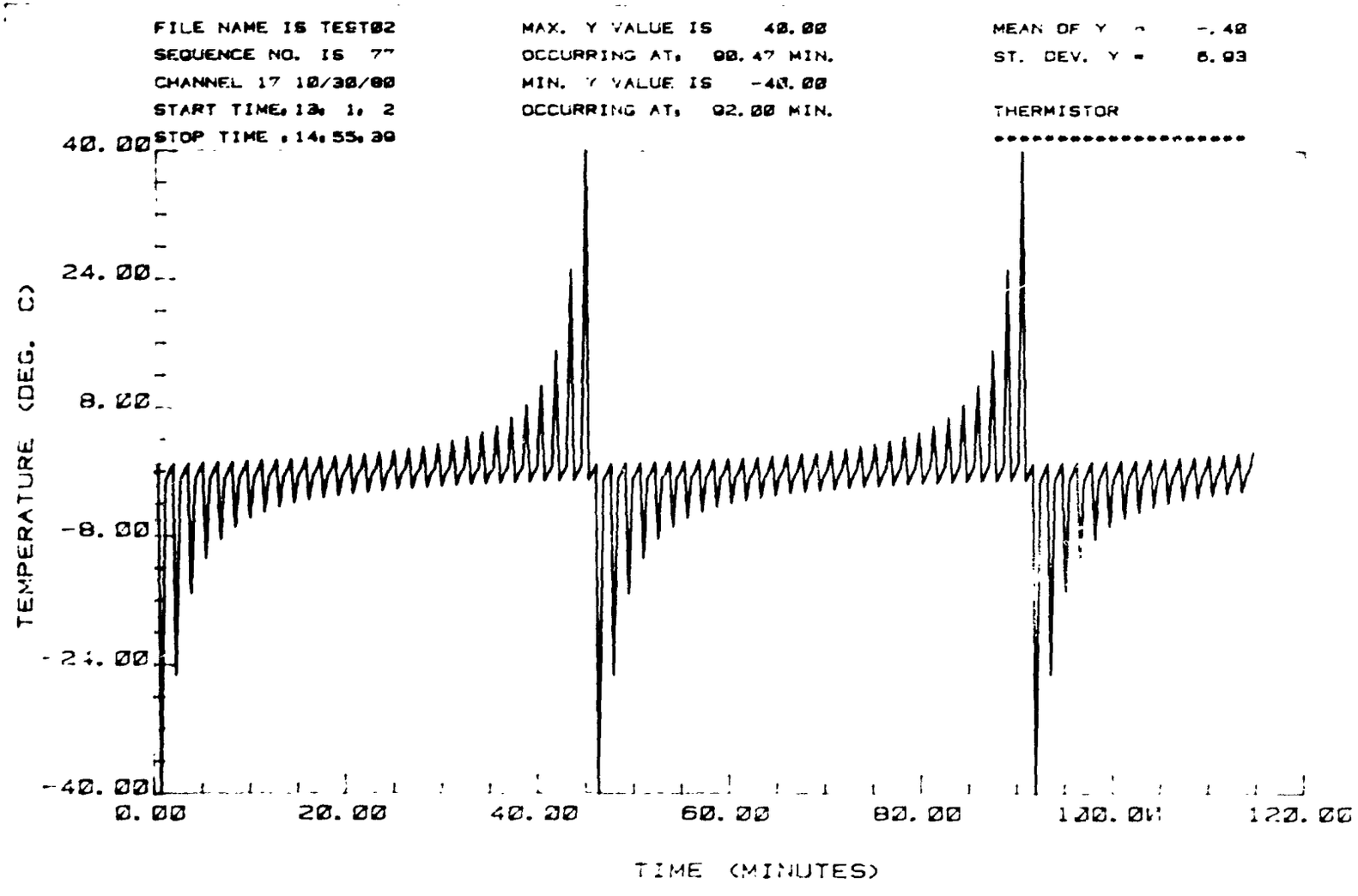

FILE NAME IS TEST02
SEQUENCE NO. IS 77
CHANNEL 16 10/30/88
START TIME:13: 1: 2
STOP TIME :14:55:39
MAX. Y VALUE IS 12.50
OCCURRING AT: .77 MIN.
MIN. Y VALUE IS 11.22
OCCURRING AT: 78.20 MIN.
MEAN OF Y = 11.83
ST. DEV. Y = .36
THERMISTOR
THIS IS A 26470 PLOT
12.50
12.24
11.98
11.73
11.47
11.22
TEMPERATURE (DEG. C)
0.00
20.00
40.00
60.00
80.00
100.00
120.00
TIME (MINUTES)

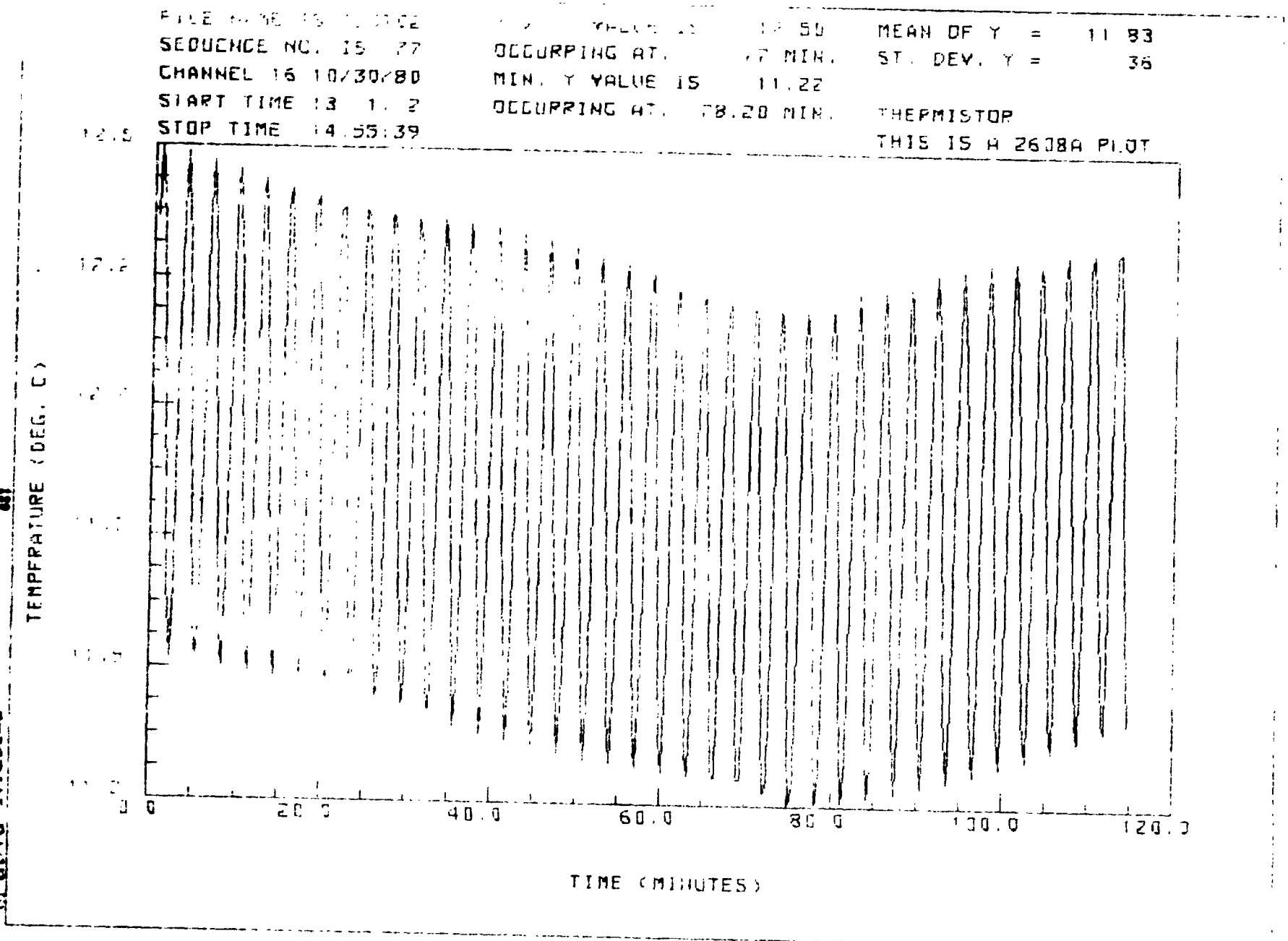

FILE NAME IS TEST02
SEQUENCE NO. IS 77
CHANNEL 17 10/30/88
START TIME:13: 1: 2
STOP TIME :14:55:39
MAX. Y VALUE IS 40.00
OCCURRING AT: 90.47 MIN.
MIN. Y VALUE IS -40.00
OCCURRING AT: 92.00 MIN.
MEAN OF Y = -.40
ST. DEV. Y = 6.93
THERMISTOR
THIS IS A 2647A PLOT
40.00
24.00
8.00
-8.00
-24.00
-40.00
TEMPERATURE (DEG. C)
0.00
20.00
40.00
60.00
80.00
100.00
120.00
TIME (MINUTES)

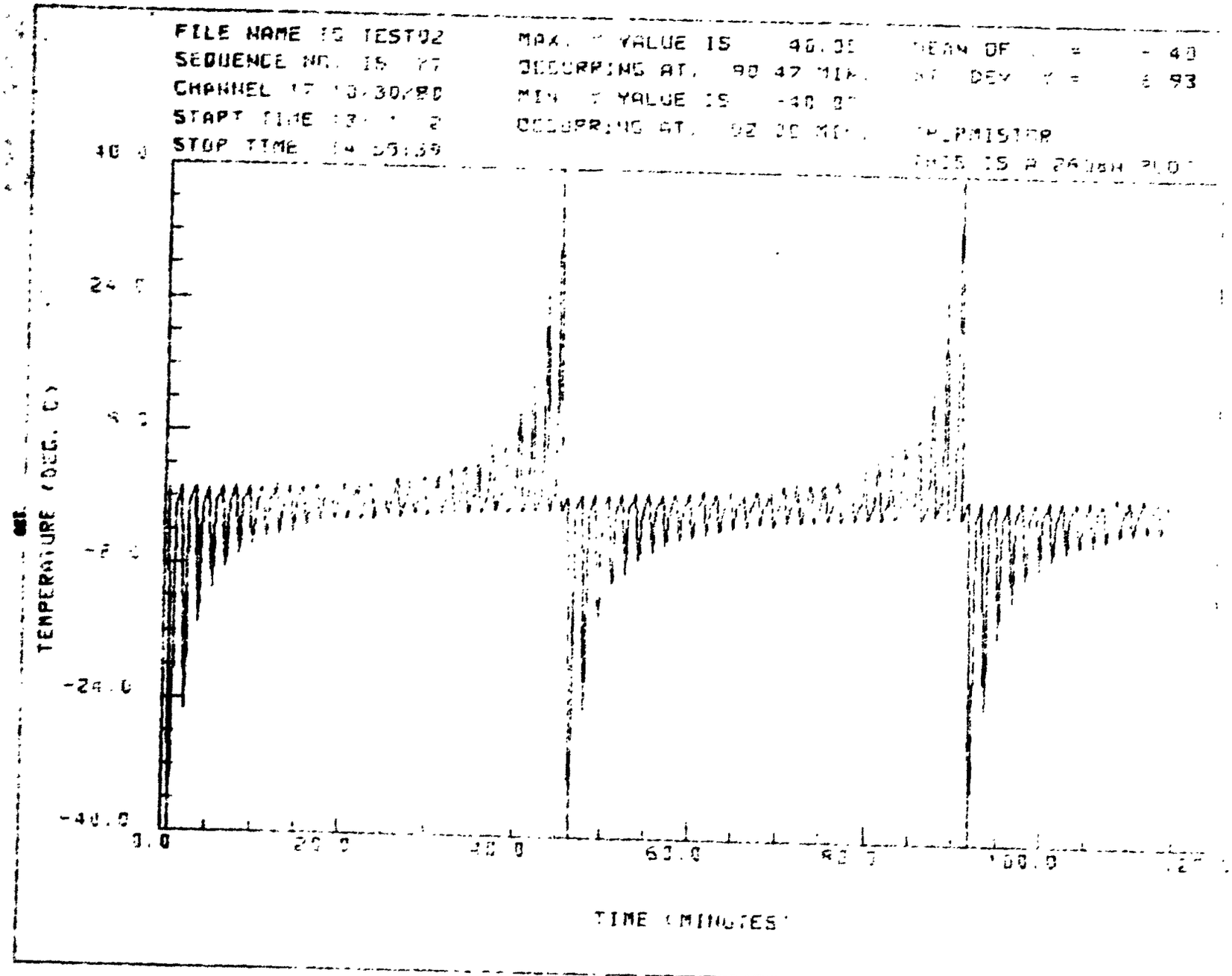

FILE NAME IS TEST02
SEQUENCE NO. IS 77
CHANNEL 18 10/30/80
START TIME:13: 1: 2
STOP TIME :14:55:39
MAX. Y VALUE IS 40.00
OCCURRING AT: 92.00 MIN.
MIN. Y VALUE IS 0.00
OCCURRING AT: 91.23 MIN.
MEAN OF Y = 2.95
ST. DEV. Y = 6.29
THERMISTOR
THIS IS A 2647A PLOT
40.00
32.00
24.00
16.00
8.00
0.00
TEMPERATURE (DEG. C)
0.00
20.00
40.00
60.00
80.00
100.00
120.00
TIME (MINUTES)


FILE NAME IS TEST02
SEQUENCE NO. IS 77
CHANNEL 23 10/30/80
START TIME:13: 1: 2
STOP TIME :14:55:39
MAX. Y VALUE IS .85
OCCURRING AT: 91.62 MIN.
MIN. Y VALUE IS -.85
OCCURRING AT: 96.85 MIN.
MEAN OF Y = .00
ST. DEV. Y = .60
THERMISTOR
THIS IS A 2647A PLOT
.85
.51
.17
-.17
-.51
-.85
VOLTAGE (VOLTS)
0.00
20.00
40.00
60.00
80.00
100.00
120.00
TIME (MINUTES)


FILE NAME IS TEST02
SEQUENCE NO. IS 77
CHANNEL 18 10/30/80
START TIME: 13: 1: 2
STOP TIME : 14: 55: 30
MAX. Y VALUE IS 12.50
OCCURRING AT: .77 MIN.
MIN. Y VALUE IS 11.22
OCCURRING AT: 78.20 MIN.
MEAN OF Y = 11.83
ST. DEV. Y = .36
THERMISTOR
12.50
12.24
11.98
11.73
11.47
11.22
TEMPERATURE (DEG. C)
0.00
20.00
40.00
60.00
80.00
100.00
TIME (MINUTES)


FILE NAME IS TEST02
SEQUENCE NO. IS 77
CHANNEL 17 10/30/80
START TIME: 13: 1: 2
STOP TIME : 14:55:30
MAX. Y VALUE IS 40.00
OCCURRING AT: 90.47 MIN.
MIN. Y VALUE IS -40.00
OCCURRING AT: 92.00 MIN.
MEAN OF Y = -.40
ST. DEV. Y = 6.93
THERMISTOR
40.00
24.00
8.00
-8.00
-24.00
-40.00
TEMPERATURE (DEG. C)
0.00
20.00
40.00
60.00
80.00
100.00
120.00
TIME (MINUTES)




FILE NAME IS TEST02
SEQUENCE NO. IS 27
CHANNEL 17 10:30/80
START TIME 13: : 2
STOP TIME 14 05:39
MAX. Y VALUE IS 40.31
OCCURRING AT 90 47 MIN.
MIN Y VALUE IS -40 87
OCCURRING AT 92 00 MIN.
MEAN OF Y = - 40
ST DEV Y = 6 93
THERMISTOR
THIS IS A 24 HR PLOT
40.0
24.0
-24.0
-40.0
TEMPERATURE (DEG. C)
0.0
20.0
40.0
60.0
80.0
100.0
TIME (MINUTES)

APPENDIX E

Sample Output From DUMPM

E-1 Octal Dump

E-2 Statistical Dump

E-3 Total Dump

```
     1
000041  000035  000012  000120  000013  000045  000005  000015  000000  000041
000017  000000  000000  000000  000000  000000  000000  000000  000000  000000
000000  000000  000000  000000  000000  000000  000000  000000  000000  000000
000000  000000  000000  000000  000000  000000  000000  000000  000000  000000
000000  000000  000000  000000  000000  000000  000000  000000  000000  000000
000000  000000  000000  000000  000000  000000  000000  000000  000000  000000
000000  000000  000000  000000  000000  000000  000000  000000  000000  000000
000000  000000  000000  000000
     2
000063  000051  000020  000200  000021  000067  000005  000000  000000  000000
000000  000200  000101  000111  000231  000200  000002  000120  000000  000000
000003  000111  000020  000000  000004  000000  000000  000200  000005  000000
000000  000200  000106  000165  000124  000000  000007  000000  000000  000200
000010  000000  000000  000200  000011  000000  000000  000200  000012  000000
000000  000200  000313  000231  000230  000000  000314  000231  000230  000200
000115  000040  000000  000200  000116  000040  000000  000200  000017  000000
000000  000200  000000  000000  000000  000000  000000  000000  000000  000000
000015  000012  000000  000000
     3
000063  000051  000020  000200  000021  000067  000041  000000  000000  000000
000000  000200  000201  000040  000066  000200  000002  000111  000224  000000
000003  000111  000020  000000  000004  000026  000124  000200  000005  000000
000000  000200  000106  000165  000124  000000  000007  000200  000140  000200
000010  000124  000111  000200  000011  000162  000223  000200  000012  000111
000060  000200  000313  000231  000230  000000  000314  000231  000230  000200
000115  000042  000121  000200  000116  000044  000047  000200  000017  000004
000050  000200  000000  000000  000000  000000  000000  000000  000000  000000
000015  000012  000000  000000
     4
000063  000051  000020  000200  000021  000067  000067  000000  000000  000000
000002  000200  000201  000111  000147  000200  000002  000111  000210  000000
000003  000111  000020  000000  000004  000061  000111  000200  000005  000000
000002  000200  000106  000165  000124  000000  000107  000071  000046  000200
000110  000001  000123  000200  000111  000024  000001  000200  000012  000203
000023  000200  000313  000231  000230  000000  000314  000231  000230  000200
000115  000044  000065  000200  000116  000047  000142  000200  000017  000007
000144  000200  000000  000000  000000  000000  000000  000000  000000  000000
000015  000012  000000  000000
     5
000063  000051  000020  000200  000021  000067  000123  000000  000000  000000
000003  000200  000201  000045  000226  000200  000002  000111  000202  000000
000003  000111  000021  000000  000004  000104  000225  000200  000005  000000
000003  000200  000106  000165  000124  000000  000107  000131  000230  000200
000110  000064  000161  000200  000111  000030  000122  000200  000012  000231
000171  000200  000313  000231  000230  000000  000314  000231  000230  000200
000115  000044  000231  000200  000116  000051  000064  000200  000017  000011
000067  000200  000000  000000  000000  000000  000000  000000  000000  000000
000015  000012  000000  000000
     6
000063  000051  000020  000200  000021  000070  000011  000000  000000  000000
000004  000200  000101  000111  000146  000200  000002  000111  000166  000000
000003  000111  000022  000000  000004  000127  000004  000200  000005  000000
000004  000200  000106  000165  000124  000000  000107  000067  000023  000200
000110  000111  000107  000200  000011  000221  000222  000200  000112  000000
000031  000200  000313  000231  000230  000000  000314  000231  000230  000200
000115  000044  000050  000200  000116  000051  000020  000200  000017  000011
000024  000200  000000  000000  000000  000000  000000  000000  000000  000000
000015  000012  000000  000000
```

```
    7
000063  000051  000020  000200  000021  000070  000045  000000  000000  000000
000004  000200  000001  000064  000060  000200  000002  000111  000161  000000
000003  000111  000023  000000  000004  000147  000204  000200  000005  000000
000004  000200  000106  000165  000124  000000  000007  000166  000223  000200
000110  000103  000201  000200  000011  000106  000146  000200  000012  000207
000042  000200  000313  000231  000230  000000  000314  000231  000230  000200
000115  000042  000100  000200  000116  000047  000004  000200  000017  000007
000011  000200  000000  000000  000000  000000  000000  000000  000000  000000
000015  000012  000000  000000
    8
000063  000051  000020  000200  000021  000070  000101  000000  000000  000000
000006  000200  000001  000226  000030  000000  000002  000111  000145  000000
000003  000111  000025  000000  000004  000167  000111  000200  000005  000000
000006  000200  000106  000165  000124  000000  000007  000004  000042  000000
000110  000030  000120  000200  000011  000002  000061  000000  000012  000145
000003  000200  000313  000231  000230  000000  000314  000231  000230  000200
000115  000031  000206  000200  000116  000043  000151  000200  000017  000003
000165  000200  000000  000000  000000  000000  000000  000000  000000  000000
000015  000012  000000  000000
    9
000063  000051  000020  000200  000021  000070  000127  000000  000000  000000
000006  000200  000201  000022  000225  000000  000002  000111  000140  000000
000003  000111  000027  000000  000004  000206  000011  000200  000005  000000
000006  000200  000106  000165  000124  000000  000007  000204  000042  000000
000010  000167  000000  000200  000011  000101  000202  000000  000012  000070
000043  000200  000313  000231  000230  000000  000314  000231  000230  000200
000115  000027  000066  000200  000116  000031  000205  000200  000017  000000
000007  000000  000000  000000  000000  000000  000000  000000  000000  000000
000015  000012  000000  000000
   10
000063  000051  000020  000200  000021  000071  000023  000000  000000  000000
000010  000200  000201  000211  000031  000000  000002  000111  000124  000000
000003  000111  000031  000000  000004  000223  000170  000200  000005  000000
000010  000200  000106  000165  000124  000000  000107  000101  000051  000000
000010  000044  000227  000200  000011  000143  000110  000000  000012  000021
000042  000200  000313  000231  000230  000000  000314  000231  000230  000200
000115  000025  000130  000200  000116  000026  000071  000200  000017  000003
000122  000000  000000  000000  000000  000000  000000  000000  000000  000000
000015  000012  000000  000000
   11
000063  000051  000020  000200  000021  000071  000051  000000  000000  000000
000011  000200  000301  000006  000164  000000  000002  000111  000111  000000
000003  000111  000042  000000  000104  000000  000150  000200  000005  000000
000011  000200  000106  000165  000124  000000  000107  000131  000207  000000
000010  000060  000105  000000  000011  000145  000000  000000  000012  000022
000070  000000  000313  000231  000230  000000  000314  000231  000230  000200
000115  000025  000000  000200  000116  000024  000002  000200  000017  000005
000210  000000  000000  000000  000000  000000  000000  000000  000000  000000
000015  000012  000000  000000
   12
000063  000051  000020  000200  000021  000071  000105  000000  000000  000000
000020  000200  000201  000140  000150  000000  000002  000111  000104  000000
000003  000111  000045  000000  000104  000006  000221  000200  000005  000000
000020  000200  000106  000165  000124  000000  000107  000064  000221  000000
000010  000201  000163  000000  000011  000111  000143  000000  000012  000060
000006  000000  000313  000231  000230  000000  000314  000231  000230  000200
000115  000025  000170  000200  000116  000023  000024  000200  000017  000006
000165  000000  000000  000000  000000  000000  000000  000000  000000  000000
000015  000012  000000  000000
```

```
   13
000063 000051 000020 000200 000021 000100 000001 000000 000000 000000
000021 000200 000101 000140 000151 000000 000002 000111 000071 000000
000003 000111 000050 000000 000104 000022 000131 000200 000005 000000
000021 000200 000106 000165 000124 000000 000007 000163 000040 000000
000110 000041 000204 000000 000011 000044 000066 000000 000012 000100
000125 000000 000313 000231 000230 000000 000314 000231 000230 000200
000115 000027 000161 000200 000116 000023 000170 000200 000017 000006
000020 000000 000000 000000 000000 000000 000000 000000 000000 000000
000015 000012 000000 000000
   14
000063 000051 000020 000200 000021 000100 000027 000000 000000 000000
000021 000200 000001 000050 000200 000000 000002 000111 000065 000000
000003 000111 000061 000000 000104 000027 000202 000200 000005 000000
000021 000200 000106 000165 000124 000000 000007 000010 000104 000200
000110 000105 000060 000000 000011 000002 000124 000200 000012 000103
000166 000000 000313 000231 000230 000000 000314 000231 000230 000200
000115 000040 000046 000200 000116 000025 000127 000200 000017 000004
000060 000000 000000 000000 000000 000000 000000 000000 000000 000000
000015 000012 000000 000000
   15
000063 000051 000020 000200 000021 000100 000063 000000 000000 000000
000023 000200 000101 000005 000207 000200 000002 000111 000060 000000
000003 000111 000064 000000 000104 000042 000147 000200 000005 000000
000023 000200 000106 000165 000124 000000 000007 000207 000171 000200
000110 000110 000220 000000 000011 000043 000222 000200 000012 000100
000130 000000 000313 000231 000230 000000 000314 000231 000230 000200
000115 000042 000164 000200 000116 000027 000224 000200 000017 000001
000222 000000 000000 000000 000000 000000 000000 000000 000000 000000
000015 000012 000000 000000
   16
000063 000051 000020 000200 000021 000100 000111 000000 000000 000000
000023 000200 000201 000024 000026 000200 000002 000111 000046 000000
000003 000111 000070 000000 000104 000047 000031 000200 000005 000000
000023 000200 000106 000165 000124 000000 000107 000103 000042 000200
000110 000062 000026 000000 000011 000065 000061 000200 000012 000062
000131 000000 000313 000231 000230 000000 000314 000231 000230 000200
000115 000044 000107 000200 000116 000040 000041 000200 000017 000000
000065 000200 000000 000000 000000 000000 000000 000000 000000 000000
000015 000012 000000 000000
   17
000063 000051 000020 000200 000021 000101 000005 000000 000000 000000
000025 000200 000201 000163 000165 000200 000002 000111 000042 000000
000003 000111 000102 000000 000104 000061 000101 000200 000005 000000
000025 000200 000106 000165 000124 000000 000107 000131 000145 000200
000010 000227 000066 000000 000011 000065 000142 000200 000012 000041
000162 000000 000313 000231 000230 000000 000314 000231 000230 000200
000115 000044 000230 000200 000116 000041 000204 000200 000017 000001
000231 000200 000000 000000 000000 000000 000000 000000 000000 000000
000015 000012 000000 000000
   18
000063 000051 000020 000200 000021 000101 000041 000000 000000 000000
000025 000200 000201 000164 000047 000200 000002 000111 000031 000000
000003 000111 000106 000000 000104 000065 000064 000200 000005 000000
000025 000200 000106 000165 000124 000000 000107 000062 000131 000200
000010 000111 000045 000000 000011 000046 000167 000200 000012 000011
000224 000000 000313 000231 000230 000000 000314 000231 000230 000200
000115 000044 000024 000200 000116 000042 000125 000200 000017 000002
000161 000200 000000 000000 000000 000000 000000 000000 000000 000000
000015 000012 000000 000000
```

```
    19
000063  000051  000020  000200  000021  000101  000067  000000  000000  000000
000027  000200  000201  000031  000061  000200  000002  000111  000025  000000
000003  000111  000121  000000  000104  000070  000230  000200  000005  000000
000027  000200  000106  000165  000124  000000  000007  000151  000102  000200
000010  000005  000127  000200  000011  000022  000150  000200  000012  000001
000001  000200  000313  000231  000230  000000  000314  000231  000230  000200
000115  000042  000026  000200  000116  000042  000064  000200  000017  000002
000121  000200  000000  000000  000000  000000  000000  000000  000000  000000
000015  000012  000000  000000
    20
000063  000051  000020  000200  000021  000101  000123  000000  000000  000000
000030  000200  000101  000044  000227  000200  000002  000111  000022  000000
000003  000111  000125  000000  000104  000102  000051  000200  000005  000000
000030  000200  000106  000165  000124  000000  000007  000022  000145  000000
000010  000131  000144  000200  000011  000002  000011  000000  000012  000011
000205  000200  000313  000231  000230  000000  000314  000231  000230  000200
000115  000031  000140  000200  000116  000041  000122  000200  000017  000001
000160  000200  000000  000000  000000  000000  000000  000000  000000  000000
000015  000012  000000  000000
```

THE HEADER RECORD FOR FILE TEST02 SHOWS:

SEQUENCE NO. = 77
DATE = 10/30/80
START TIME = 13: 1: 2
STOP TIME = 14:55:39
NO. CHANNELS = 30

CN	MEAN	RMP	STD DEV	MIN	TIME	ETIM MIN	MAX	TIME	ETIM MIN
0	.622	.000	.013	0.000	13: 1: 2	0.000	.645	14 55.39	114.617
1	.001	4.141	1.543	-3.077	13:55. 5	54.050	3.092	14 36. 4	95.067
2	-.590	.000	.007	- .510	13:48:57	47.917	- .490	14:30.21	89.217
3	-.568	.000	.006	- .509	14 37:15	96.217	- .491	14.51: 7	110.017
4	1.659	.036	.198	0.000	13: 1: 2	0.000	1.711	14:54:30	113.467
5	.017	.000	.008	0.000	13: 1: 2	0.000	.030	14:17.42	76.667
6	-1.755	.000	.007	-1.755	14 55.39	114.617	-1.755	14:55:39	114.617
7	.006	.282	1.132	-1.000	14 71 76	99.467	1.000	14.33. 2	92.000
8	.017	1.127	1.060	-1.500	14:17:42	76.667	1.500	13 53 56	52.900
9	.006	.014	.117	- .658	13 7 29	2.700	1.155	13. 1 48	.767
10	.006	.012	.106	- .438	13: 4	2.067	.999	13: 1 48	.767
11	4.000	.000	.013	-4.000	14:55 39	114.617	-4	14:55:39	114.617
12	4.000	.000	.013	4.000	14:55 39	114.617	4.000	14:55.39	114.617
13	1.200	.001	.005	1.150	14 34:34	97.533	1.250	14.33: 2	92.000
14	1.201	.002	.040	1.104	14:43:45	102.733	1.296	14:33: 2	92.000
15	.018	.002	.059	- .086	13:12.30	11.500	.121	14:30 2	92.000
16	1.187	.001	.036	1.120	14.19 14	75.200	1.250	13 1:49	.767
17	-.043	.577	.160	-4.667	14:32	92.000	4.667	14:31.30	90.467
18	.306	.485	.696	0.000	14:34 14	91.033	4.667	14 33: 2	92.000
19	-.021	.076	.026	-4.667	13. 1:40	.767	.667	13 1:25	.383
20	.556	- .000	0.000	.555	14 55 39	114.617	.555	14.55:39	114.617
21	0.000	0.000	0.000	.000	11.55 39	114.617	0. .0	14.55:39	114.617
22	-.501	.000	.001	- .500	14 55.39	114.617	-.500	14:55.39	114.617
23	.003	.3e2	.001	.050	14:31:52	90.850	.050	14.32.39	91.617
24	.001	.004	.067	- .541	13: 3:11	1.150	.731	13: 1:25	.383
25	.222	- .000	.000	.222	14 55.39	114.617	.222	14.55:39	114.617
26	.444	.000	.000	.444	14.55.39	114.617	.444	14:55.39	114.617
27	.899	.000	.000	.899	14:55.39	114.617	.899	14 55.39	114.617
28	1.600	.000	.007	1.600	14:55 39	114.617	1.600	14.55:39	114.617
29	1.600	.000	.007	-1.600	14 55.39	114.617	-1.600	14:55:39	114.617
30	-.389	.000	.003	.389	14:55.39	114.617	-.389	14:55 39	114.617

THE HEADER RECORD FOR FILE TEST.. SHOWS

SEQUENCE NO = 33
DATE = 10/29/80
START TIME = 11:37: 5
STOP TIME = 13: 0:33
NO CHANNELS = 15

CH	MEAN	VAR	STD DEV	MIN	TIME	ETIM MIN	MAX	TIME	ETIM MIN
0	.016	.000	.019	[illegible]	11 37 21	.267	.021	13: 0 37	83.467
1	.013	2.455	1.567	-3.062	12 33 21	56.267	3.171	12 44:33	67.467
2	-.500	.000	.007	[illegible]	12 56 1	78.333	-.49	12.41:53	64.800
3	-.500	.020	.096	[illegible]	12:43 5	72.000	-.491	12 34:41	57.600
4	1.643	.047	[illegible]	0.000	11:37 5	0.000	1.710	12.56.25	79.333
5	.016	[illegible]	[illegible]	[illegible]	11 37 21	.267	.036	12.57 5	80.00
6	-1.755	[illegible]	.007	[illegible]	7 33	3.467	-1.755	13 0:33	83.467
7	.015	[illegible]	1.124	[illegible]	12.29 45	2.267	1.600	12.41.21	64.267
8	.025	[illegible]	.04	[illegible]	[illegible]	0.000	1.500	12.32.17	55.200
9	.009	[illegible]	[illegible]	-.65	11 39 29	.400	1.125	11 37 53	.800
10	.012	.017	.129	.43	11 40 17	[illegible]	.002	11.38: 9	1.067
11	-4.000	.000	.017	-4.000	13 0:33	83.467	-4.000	13: 0:33	83.467
12	4.000	.000	.017	4.000	13 [illegible]	83.467	4.000	13: 0:33	83.467
13	.203	[illegible]	[illegible]	[illegible]	12:42 57	65.267	.250	12 41.21	64.267
14	.201	[illegible]	.04	[illegible]	12 55.31	75.467	.298	12 41.21	64.267
15	.017	[illegible]	[illegible]	[illegible]	11 [illegible] 25	21.333	.122	12:41:21	-4.267

```
 THE HEADER RECORD FOR FILE TEST01 SHOWS:

SEQUENCE NO. = 33
DATE         = 10/29/80
START TIME   = 11:37: 5
STOP TIME    = 13: 0:33
NO. CHANNELS = 15
```

BLOCK NUMBER	1
SEQUENCE NO.	33
DAY	29
MONTH	10
YEAR	80
HOUR	11
MINUTE	37
SECOND	5

CHANNEL NO.	VOLTAGE
0	0.00000
1	1.49990
2	-.50000
3	-.49100
4	0.00000
5	0.00000
6	-1.75540
7	0.00000
8	0.00000
9	0.00000
10	0.00000
11	-3.99980
12	3.99980
13	1.20000
14	1.20000
15	0.00000

BLOCK NUMBER	2
SEQUENCE NO.	33
DAY	29
MONTH	10
YEAR	80
HOUR	11
MINUTE	37
SECOND	21

CHANNEL NO.	VOLTAGE
0	0.00000
1	2.20360
2	-.49940
3	-.49100
4	.16540
5	0.00000
6	-1.75540
7	.80600
8	.54490
9	.72930
10	.49300
11	-3.99980
12	3.99980
13	1.22510
14	1.24270
15	.04280

THE HEADER RECORD FOR FILE TEST02 SHOWS:

SEQUENCE NO. = 77
DATE = 10/30/80
START TIME = 13: 1: 2
STOP TIME = 14:55:39
NO. CHANNELS = 30

BLOCK NUMBER	19
SEQUENCE NO.	77
DAY	30
MONTH	10
YEAR	80
HOUR	13
MINUTE	4
SECOND	29

CHANNEL NO.	VOLTAGE
0	.00130
1	1.64940
2	-.49280
3	-.49360
4	1.24970
5	.00130
6	-1.75540
7	1.19650
8	-1.42990
9	.31010
10	-.37070
11	-3.99980
12	3.99980
13	1.23730
14	1.19120
15	-.00730
16	1.23600
17	.11260

BLOCK NUMBER	20
SEQUENCE NO.	77
DAY	30
MONTH	10
YEAR	80
HOUR	13
MINUTE	4
SECOND	29

CHANNEL NO.	VOLTAGE
18	.11260
19	.02920
20	.55550
21	0.00000
22	-.50050
23	.84400
24	.21880
25	.22210
26	.44440
27	.88880
28	1.60000
29	-1.60000
30	-.88880