SQT

NASA CR-166689

TITLE OF PROJECT	Advanced Gamma Ray Balloon Experiment Ground Checkout and Data Analysis
PRINCIPAL INVESTIGATOR	E. L. Chupp, Professor of Physics University of New Hampshire Durham, New Hampshire 03824
PERIOD OF PERFORMANCE	19 November 1974 to 30 April 1976
NASA CONTRACT NUMBER	NAS 5-20884
TYPE OF REPORT	Type III Final Report; Operator's Manual; Programmer's Manual
PREPARED FOR	Goddard Space Flight Center Greenbelt, Maryland 20771
TECHNICAL OFFICER	Larry E. Orwig, Code 682 Goddard Space Flight Center
DATE PREPARED	June 1976

PREPARED BY M. Blackstone 22 June 76
M. Blackstone Date

APPROVED BY I. U. Gleske 21 June 76
I. U. Gleske Date

ENDORSED BY E. L. Chupp 21 June 1976
E. L. Chupp Date

(NASA-CR-166689) ADVANCED GAMMA RAY BALLOON EXPERIMENT GROUND CHECKOUT AND DATA ANALYSIS Final Report, 19 Nov. 1974 - 30 Apr. 1976 (New Hampshire Univ.) 40 p HC A03/MF A01 CSCL 03B G3/93

N81-28034

Unclas 30198

Technical Report Standard Title Page

1. Report No. 76-20884-01	2. Gov't Accession No.	3. Recipient's Catalog No.
4. Title and Subtitle Advanced Gamma Ray Balloon Experiment Ground Checkout and Data Analysis		5. Report Date June 1976 6. Performing Organ. Code
7. Author(s) M. Blackstone Approved by: I. U. Gleske and E. L. Chupp		8. Perform. Organ. Rpt. No.
9. Performing Organization Name and Address University of New Hampshire Durham, New Hampshire 03824		10. Work Unit No. 11. Contract or Grant No. NAS 5-20884
12. Sponsoring Agency Name and Address Larry E. Orwig, Code 682 Technical Officer, NAS 5-20884 Goddard Space Flight Center Greenbelt, Maryland 20771		13. Type of Report and Period Covered Type III Final Report; Operator's Manual; Programmer's Manual (19 Nov. 74 - 30 Apr. 76) 14. Sponsoring Agency Code
15. Supplementary Notes		
16. Abstract The objective of this contract was to develop a software programming package to be used in the ground checkout and handling of data from the advanced gamma ray balloon experiment. This document consists of two parts. The first is the Operator's Manual which is intended to permit someone unfamiliar with the inner workings of the software system (called LEO) to operate on the experimental data as it comes from the PCM interface, converting it to a form for later analysis, and monitoring the program of an experiment The second part is a Programmer's Manual, intended to permit someone familiar with the Varian 72 and the requirements of an experimenter to modify the LEO system.		
17. Key Words (selected by author)		18. Distribution Statement
19. Security Classif. (of this report) Unclassified	20. Security Classif. (of this page) Unclassified	21. No. of Pages 35

Table of Contents

Preface

The objective of this contract was to develop a software programming package to be used in the ground checkout and handling of data from the advanced gamma ray balloon experiment.

The programming package developed was named LEO. It consists of programs to accept data from the PCM interface, extract experiment data, save the data on magnetic tape, accumulate spectra and other experiment data, and print and display the accumulated data.

The period of performance was originally six months after the date of contract. The twelve-month delay was necessary in order for the contractor to gain access to the advanced gamma ray experiment ground support computer system. Without access to this computer, program testing at Goddard Space Flight Center (or a place suitable to both parties) was impossible.

This document consists of two parts: Part 1, the Operator's Manual, and Part 2, the Programmer's Manual.

List of Tables

LEO Operator/Programmer Manual

The system called LEO consists of programs to accept data from the PCM interface, extract experiment data, save the data on magnetic tape, accumulate spectra and other experiment data, and print and display the accumulated data.

This document is in two parts. The first is an operator's manual intended to permit someone unfamiliar with the inner workings of LEO to operate on the experimental data as it comes from the PCM interface, converting it to a form for later analysis, and monitoring the program of an experiment. The second part is a programmer's manual, intended to permit someone familiar with the Varian 72 and the requirements of an experimenter to modify the LEO system. Particular changes that may be desired include the formatting of printout and displays, and special attention has been given to the program segments that perform these tasks.

Part 1. Operator's Manual

LEO uses the Tektronix terminal and hard-copy device, the Centronics line printer, the magnetic tape drives and the PCM interface; these should all be ready before starting LEO.

1.0 Loading LEO

LEO is loaded using the AID program. A tape containing the LEO system is mounted on tape unit Ø (or 1) and the AID command RØ, (or R1,) is typed. The system will be loaded and initialized, an asterisk will be typed on the terminal, and the bell will be sounded, indicating that LEO is waiting for operator input. To restart LEO without reloading, the starting address is Ø (i.e., get the computer in AID and type GØ. on the keyboard).

The system is capable of performing several tasks, of four types: data input/output, data accumulation/processing, data printout/display, and operator interaction. The printout/display and operator interaction tasks are called slow tasks since they cannot be executed as fast as data flows from the PCM interface; the others are called fast tasks. The tasks which the system is executing at any time are contained on a list (task list) which the operator can manipulate.

1.1 Communicating with LEO

One part of the system called LEO is an operator interaction routine, which for simplicity is also called LEO; to communicate with the system LEO, one must type input to the routine LEO.

Whenever LEO is awaiting input, an asterisk will be typed and the bell will usually sound; whenever another routine of the system is awaiting input, the asterisk is not typed, but the bell will usually sound.

When the operator types input to the LEO system, the following conventions should be observed:

1. Alphabetic input, such as task names, are only checked in their first four characters; any number of characters may be typed, but only the first four are used;

2. Numeric input is interpreted as decimal integers, with a minus sign for negative numbers;

3. A field, alphabetic or numeric, is terminated by one or more spaces, commas, or carriage returns;

4. If a typing mistake is made, and detected before a terminator is typed, the field may be corrected by typing a back-slash (SHIFT-L), and retyping the entire field;

5. When typing a LOG record (in response to the cue LOG:) all characters including blanks and commas, are saved as part of the input line until a carriage return is typed; if a typing mistake is made, one character is deleted for each back-slash typed; this mode is used only with the LOG task to place a comment or identifying record on an output tape (see Section 1.3.5).

1.2 The Task List

In order to cause a task to be executed, it must be placed on the task list. Commands to LEO are the means by which this is accomplished. Tables I & II describe all the commands available, and the parameters required for some commands. The distinction between the two tables is the method of execution of the task. Those tasks in Table I are

placed on the task list; those in Table II are executed directly by the interaction routine, and are more elementary in nature.

The operator should note that typing a command does not necessarily start a task. If the task is a fast task and the PCM is the data input source, then the task is started as soon as it is placed on the task list. If the task is a slow task, or if the CDT (Compact Data Tape) is the data input source, then any slow task (including LEO) already on the list must be executed first, or removed (with the STOP command); the LEO task is removed with the END command.

When data is being processed, LEO is not usually on the list. In order to place LEO on the list, two methods are available.

1. By typing the RUBOUT key, LEO is placed on the task list; because of priorities defined within the system, this will stop any slow task that was running at the time, but will not affect any fast tasks. Once LEO is on the task list, the operator may use the normal commands as described above and in Tables I and II.

2. By typing the BELL key (CTRL-G), LEO is placed on the task list, but with a priority higher than any other task. Thus all other processing is stopped and any PCM data is lost. This method should be used only in "emergency" situations. At times, when CDT input is being used, the BELL can be used without losing data; this would occur when a printing or plotting task is being executed and the input tape is stationary. In this situation, the RUBOUT does not

receive attention until the printing or plotting is finished. The BELL always receives immediate attention.

1.3 LEO Tasks

In order to run certain tasks, the operator must type parameters or responses to queries to accomplish a desired function. The following sections explain the functions which may be accomplished by each task, and the parameters needed by them.

1.3.1 Pulse Height Analyzer

The pulse height analyzer function accumulates the event amplitude and event time spectra, prints the spectra and displays the amplitude spectrum on the terminal.

The analyzer function is started with the PHA command, and requires no parameters; it continues until the end of data or the STOP, PHA command.

The printing of the spectra can be initiated any time with the PHAPRNT command.

The display of the amplitude spectrum can be done in two ways. The PHADSP command requires no parameters and displays all 256 channels; the vertical scale is either the last used scale, or defaults to an initial value of 100 counts. The PHASCALE,m,n,p command requires three parameters: first channel (m), last channel (n), and vertical scale (p).

When the spectrum display program starts, it sounds the terminal's bell and waits for a command input prior to providing a display on the screen. The command input is used to adjust the vertical scale: if Ø is typed, the vertical scale is doubled; if any alphabetic character is typed, the scale is not changed. If any positive number is typed, the display task is terminated. If a new display is desired with updated data this command procedure is repeated to obtain a new plot with a revised scale and using updated pulse height data.

1.3.2 Sub-Com Data

The sub-com data is accumulated for a specified number of major frames by the command SUBCOM with one parameter, the desired number of major frames (SUBCOM,n). If the task is allowed to run to completion and the computer has completed processing the n major frames of data, it displays the words "SUB-COM FINISHED" on the terminal screen.

The accumulated data can be printed at any time by the command SUBPRNT with no parameters. (It is not necessary to wait the total n major frames - partial printouts are allowed.)

The sub-com data can be displayed by the command SUBDSP. The SUBDSP task requires operator interaction to choose the sub-com channels to be displayed. The sub-com channels are two types, normal or compressed, chosen from Table III. For a desired channel the operator types the

proper title , comma, and any single character symbol to identify the line on the plot (e.g. ;LT,A). Any number of channels of the same type may be plotted on a single set of axes. For compressed data, the vertical scale is a log scale.

1.3.3 Integral Discriminator Count Rates

The Integral Discriminator Count Rates can be accumulated by the command IDC. Since 32 major frames are required to complete a set of spectra, it is normally undesirable to stop the task at an arbitrary time. Instead the command IDCEND will stop the task after the next complete set of spectra is accumulated. When the task is completed the message, "IDC FINISHED" is displayed on the terminal screen.

To print the IDC spectra, use the IDCPRNT command.

To display the IDC spectra, use the IDCDSP command. The task, when started, will sound the terminal's bell and wait for an operator input; if the operator input is a positive number, the task is terminated, otherwise the next spectrum is plotted and the sequence repeats. After all ten spectra have been displayed, the task is terminated. The vertical scale is always a log scale. A printout or display of partially completed IDC spectra can be obtained by commanding IDCPRNT or IDCDSP before the IDC task is completed.

1.3.4 Snap/Dump

In order to get a quick look at the raw data, a dump task is available, called by the command DUMP with two parameters. The parameters may be used in two ways; to see N minor frames, starting at frame number M, the parameters would be M and N (i.e., DUMP,M,N); to see the next N minor frames, the parameters would be N and zero (0) (i.e., DUMP,0,N).

Since the printer cannot print as fast as the data flows from the PCM, the data is saved as it flows in, and printed at the slower rate. Therefore, the maximum number of frames that can be dumped is determined by the amount of memory available to save the data. If the operator requests more than is available, a message is printed and the task is not started.

1.3.5 Data Input/Output

The first method of input is simply the PCM interface. This method is activated by the command PCM. In order to save the compacted data (non-data words stripped out) on tape the Compact Data Tape Output task is used; the command is CDTOUT, and the parameter is the tape unit number, either zero or one.

To read CDT, the command CDTIN is given followed by the unit number.

Since the CDTIN and PCM modes are mutually exclusive, starting either of them automatically stops the other. In the CDT input mode, any non-data records are treated as comments and are printed on the printer and typed on the terminal. In order to place such records on the output tape, the operator must use the LOG command, with the unit number as a parameter. When started, the LOG task cues the operator with the message LOG: and places the input interpreter in a special mode; in this mode, every character typed is saved as part of a log or comment record, until the carriage return is hit. Then the accumulated record is written on the output tape unit and the task is terminated.

1.4 Recommendations and Cautions

The following recommendations are version-dependent, but apply to the version installed on 15 March 1976. It is hoped that maintenance programmers will update this section as changes are made.

The operator should realize that the two forms of input (PCM and CDT) have different effects on the execution of other programs in the system. With PCM input, it is of greatest importance not to lose any data, and to save the data for later analysis. Therefore, slow tasks are executed one at a time and may be interrupted by the PCM interface.

When a CDT is used for input, these interrupts do not occur, and a slow

task, once started, proceeds to completion before more input data is processed. In addition, the RUBOUT command is not recognized immediately when a slow task is running, although it is recognized when the normal task sequence comes to it. In PCM mode, the RUBOUT appears to get immediate attention, whereas it may not in CDT input mode; the BELL always gets immediate attention.

As an orderly procedure, the task list should be completed, and END typed, before the PCM is enabled; the system will wait until data starts to flow from the PCM interface.

Misspelling and incorrect use of names can occasionally be hazardous. When typing commands, the operator should be careful to use enough characters to obtain the desired command. As an example of the effects of improper use: After accumulating a set of IDC spectra and seeing the message IDC FINISHED, the operator typed IDCP (four character abbreviation for IDCPRNT) and obtained a listing of the spectra. Then desiring to see the displays the operator attempted to type IDCD (for IDCDSP); unfortunately, the final D was missed and only IDC was typed. This resulted in the system setting up the IDC task again, allocating a new buffer (cleared to zero) and losing the previous buffer. Since there was no input connected at the time no physical manifestation of the error occurred. The operator noted that no display appeared, realized he had not typed IDCD, and re-typed IDCD; at this time he successfully obtained plots, but of the all-zero spectra in the new buffer.

It is expected that the use of the commands as spelled in Section 1.2 will minimize the risk of this type of accident.

When there is no input task on the task-list, all accumulation tasks should be stopped; otherwise they may repeatedly accumulate the last frame of data, giving erroneous results.

If the operator interrupts and stops tasks, buffers may be allocated and not released automatically; these will waste space in the memory. If space is desired, the operator can obtain a list of currently allocated buffers and their sizes, and may delete buffers not needed to make more space. If the operator deletes a buffer used by a task on the task list, many bad things can happen, including storage of data into the wrong buffer. Caution is advised; the command TASK should be used with the command BUFFER so that the operator has a clear idea of the total system status.

Table I

Operator Commands Affecting the Task List

Command	Effect
PCM	Starts PCM interface task
CDTIN,n	Starts CDT input task; n = input tape unit (Ø or 1)
CDTOUT,n	Starts CDT output task, n = output tape unit (Ø or 1)
PHA	Starts Pulse Height Analyzer Task
IDC	Starts Integral Discriminator Count Task
SUBCOM,n	Starts Sub-Com data accumulation, for n major frames
LOG , n	Starts task to place log record on output tape n
DUMP,m,n	Starts task to print n minor frame of raw data starting with minor frame m
PHAPRNT	Prints PHA spectrum
PHADSP	Displays spectrum on terminal
PHASCAL,m,n,p	Displays channels m to n with vertical scale of p
SUBPRNT	Print sub-com data
SUBDSP	Displays sub-com data
IDCPRNT	Prints IDC data
IDCDSP	Displays IDC data
STATUS	Determine experiment status
STOP,name	Removes task from task list
END	Removes LEO from task list
RESTART	Re-initializes the system
RESET	Same as RESTART
(RUBOUT)	Starts LEO task, stopping slow tasks
(BELL)	Starts LEO task, stopping all tasks

Table II

Commands Executed Without Task List

EOF,n	Writes file mark on tape unit n (0 or 1)
REWIND,n	Rewinds tape unit n
PAGE	Ejects a page on the printer
SPACE	Determines the amount of buffer space remaining
BUFFER	Types the name and length of buffers currently allocated
RELEASE,name	Releases (de-allocates and loses) buffer named
TASK	Types names of tasks on task list
IDCEND	Sets a flag so accumulation of IDC data will end at end of next complete spectrum

Table III

Sub-Com Channel Titles

Title	Minor Frame	Variable	Type
CCUD	9	Central crystal upper disc level rate	C
CCLD	10	Central crystal lower disc level rate	C
TSUD	11	Top shield upper disc level	C
TSLD	12	Top shield lower disc level	C
CSUD	13	Central shield upper disc level	C
CSLD	14	Central shield lower disc level	C
BSUD	15	Bottom shield upper disc level	C
BSLD	16	Bottom shield lower disc level	C
EUD	17	Ears upper disc level	C
ELD	18	Ears upper disc level	C
PTR	41	Plastic total rate	C
CGER	42	Central good event rate	C
TLT	43	Total live time	C
SLLT	44	Shield lower level live time	C
SULT	45	Shield upper level live time	C
CAL	46	Calibration	C
STAT	47	Experiment status	N
V+15	48	+15 Volts	N
V+10	49	+10 Volts	N
V+5	50	+5 Volts	N
V-5	89	-5 Volts	N
V-10	90	-10 Volts	N
V-15	91	-15 Volts	N
L+5	92	+5 Volt Logic	N
R+28	93	+28 Volt Regulated	N
HV1	94	High Voltage 1	N
HV2	95	High Voltage 2	N
HV3	96	High Voltage 3	N
HV4	97	High Voltage 4	N
DTMP	98	Detector Temperature	N
ETMP	99	Electronics Temperature	N
THRE	76	Variable Threshold	N

Part 2. Programmer's Manual

This manual will describe the various modules of the LEO system. It is intended for the maintenance programer familiar with the Varian computer and the various peripherals used by LEO. In addition, the programmer is expected to have a source listing of LEO for reference, to be familiar with the MOS and DAS, and to have mastered the Operator's Manual for LEO.

The order of the discussion is thought to be fairly logical, though not the same as the order of the listing. The last section contains descriptions of the buffers used in all modules.

2.0 Constraints

LEO is intended to receive data from the PCM interface and decode the data, save the data on magnetic tape, accumulate those parts of the data of interest to the operator and print and display the results. The data from the PCM flows continuously with a time interval of approximately 164 milliseconds per minor frame of data. Therefore that processing which is required to be done every minor frame must be completed in less than 164 milliseconds; and all other processing must tolerate being interrupted for the mandatory processing.

In some uses of LEO, and at some times during the execution of LEO, storage buffers of varying sizes and in varying numbers are required. Therefore the subprograms of LEO do not have fixed buffers; but rather all of the memory space not taken by program code is made available to a dynamic buffer allocation system. On a 32K machine, this leaves about 24K available for buffers.

2.1 Coordination of LEO Tasks

The last module on the listing, COORD, is responsible for the orderly execution of the tasks in LEO; to this end the subprograms in COORD maintain the LEO task list, maintain the execution sequence through the list, monitor the status of the PCM interrupts, and set traps for low priority (slow) tasks.

The task list has room for a maximum of 20 tasks. It is represented by 42 words beginning at TSKTBL. TSKTBL contains the address of the current active word in the table; the next 20 words contain the entry addresses of tasks in LEO; TSKTBL contains the number 10, and the following 20 words contain the priorities of the tasks on the list, ranging from 9 (highest) to Ø.

In order to enter a task in the task list, the subroutine COORD is used; the task address is in the A register and the task priority is X; if the priority is zero, the task is deleted from the list. Tasks are maintained on the list in order of priority.

The subroutine SEQNCE uses the task list and PCM interrupt status to link to the next task; if the next PCM minor frame is ready, then the task list is started from the beginning, otherwise the next task is executed.

The subroutine MONITR is always the last task on the list, with a priority of zero (no task could be added with priority zero). If a CDT is used as input, MONITR restarts the task list; otherwise it waits for the PCM minor frame ready condition, and then restarts the sequence.

The subroutine TRAP is always on the task list with priority 2. TRAP does nothing unless a slow task has previously been interrupted by the PCM; in this case the interrupted task is activated at the point of interruption. All slow tasks have priority 2 or lower.

The subroutine TRAPST is called by a slow task when the PCM is ready, and sets up a trap to the subroutine TRAP.

2.2 Buffer Maintenance

The module BUFSYS maintains the buffer space for LEO. The buffer status is maintained in a table beginning with BFAVAL. BFAVAL contains the address of the next available word in the buffer space. Following BFAVAL, BFLEN contains the number of available words in the buffer space. (The buffer space extends to just below AID, at address 75777.) The next 60 words contain the information for up to 20 buffers in 3-word blocks. These blocks contain: The buffer identifier (the address of the routine that uses the buffer), The address of the first word of the buffer, and the length of the buffer.

When a buffer is required, the subroutine ALOCAT is called, with an identifier in the A register, and the desired length in the B register. ALOCAT releases any buffer with the same identifier and allocates the new buffer; if insufficient space exists for the buffer, a message is typed and the B register is zero on return; otherwise the B register contains the address of the word in the buffer maintenance table which contains the address of the buffer.

When the buffer is no longer needed, the subroutine RELEAS is called, with the identifier in the A register. The identified buffer is released by deleting reference to it in the buffer maintenance table, and by moving all buffers in higher memory locations so that all the available buffer space is one contiguous block. The buffer maintenance table is then updated to effect the changes.

2.3 Terminal Input

The operator's terminal is connected to the Priority Interrupt Module (PIM) of the Varian, so that striking a key causes an interrupt; this frees LEO from scanning the key board periodically. When a key is struck, the interrupt system causes a jump to the subroutine TYINT, in the module TYIN. TYINT treats the input character as an interrupt (to LEO), as a delimiter, or as part of an input field. Since input can be typed independent of the need for it, completed fields are stored in a circular stack, which can hold up to eight fields of two words each. Fields are of two types, numeric (containing numerals with possible leading minus sign) or alphabetic (anything else). Alphabetic fields contain four characters, padded on the right with blanks if necessary; numeric fields contain a number in the first word and a zero in the second.

When a program requires operator input, it calls the subroutine FIELD, which returns the next field from the stack in the A, B registers. If there is no field available, and a carriage return was the last delimiter, the bell is sounded once and FIELD waits for input; since typing is inherently slow, FIELD tests the PCM interrupt status while it waits,

and calls TRAPST (in COORD) if interrupt occurrs.

If the operator types the RUBOUT or BELL key, TYINT calls COORD with a priority of 2 or 9, and the address of OPINT, the operator interaction task (this is called LEO in the Operator's Manual).

TYINT has a special mode, used by the LOG task, in which characters are not interpreted (except for RUBOUT and BELL), but are stored in a buffer for use as a print line. This mode is terminated by the carriage return.

2.4 Operator Interaction (LEO)

LEO executes Operator commands by calling FIELD, calling a search subroutine to scan a command table to match the field, obtaining a subroutine address from the command table, and calling the subroutine. Depending on the command, the called subroutine may do something directly or may in turn call COORD to affect the task list. If the search fails to find a match, an appropriate message is typed.

2.5 PCM Input

When the operator types the command PCM, LEO calls the subroutine PCM. This subroutine removes the CDTIN task from the task list, places the PCMIN task on the list with priority 6, and sets the PIM to generate interrupts when the PCM detects minor frame sync patterns; PCM then returns to LEO.

When an interrupt from the PCM occurs, the subroutine PCMINT is entered, this routine starts the Buffer Interface Controller (BIC) so that the

next minor frame is transmitted into the PCM buffer. A flag (INTFLG) is set and PCMINT returns to the normal program flow.

When a slow task (or MONITR or SEQNCE) detects that INTFLG is set and that the BIC is finished transmitting data, the task list gets restarted and PCMIN is then executed. PCMIN takes the experiment data from the minor frame and copies it into a Compact Data Tape (CDT) buffer, and places the address of the CDT buffer in a pointer called RECPTR. All other tasks in the LEO system obtain their data from the buffer pointed at by RECPTR.

2.6 CDT Input/Output

When the operator types the command CDTOUT, the subroutine CDOUST is called, which calls FIELD to obtain the tape drive number, and calls COORD to set up CDTOUT with priority 4.

CDTOUT counts minor frames, and outputs a record every four frames; this record is 81 words long, 20 words for each frame and one identifier word which is zero.

When the operator types the command CDTIN, the subroutine CDINST is called, which calls FIELD to obtain the tape drive number, calls COORD to delete PCMIN from the task list, clears the CDT buffer with 133 blanks (enough for a printer line) and calls COORD to set up CDTIN with priority 5.

CDTIN increments RECPTR by 20; and if the end of the CDT buffer is reached, reads a record from the input tape drive. If the identifier word is negative, the record is treated as a log record and typed on the terminal and printed.

2.7 Log

When the operator types the LOG command, the subroutine SETLOG is called, which obtains a buffer from ALOCAT and clears it to blanks, then calls COORD to place LOG on the task list, with priority 1.

When the task LOG starts, it prints the cue LOG: and sets TYINT in the character mode. LOG then waits for the carriage return, after which the log record is written on the output tape drive and LOG deletes itself from the task list. While waiting for the input to be finished, the task can be interrupted by the PCM.

2.8 Print Utility

The LEO print utility consists of subroutines to format items into a print line and transfer the print line to the line printer. Some other useful subroutines are also included in this module.

The subroutine PRINT is entered with the address of a print buffer in the X register; 132 words are transferred to the printer (one character per word) and replaced by blanks. The line count is incremented and if it goes over 62, the printer page is ejected.

The subroutine EJECT ejects the printer page and resets the line counter. SPACE spaces the printer one print line. LPCHAR sends a character to the printer; while waiting for the printer to be ready, LPCHAR checks the PCM interrupt status and can be interrupted.

The subroutine FORMAT is used to place items into a print buffer. The calling sequence for this subroutine is different in form from other LEO subroutines. (This was so that the subroutine $SE could be used to transfer the parameters; in retrospect the form used by other subroutines, like AXIS or GRAPH, would have been better.) The call is,

```
CALL  FORMAT  NVAR,FIRST,STEP,BUFPSN,BUFSTP,TYPE,DEC
```

The parameters are all integers, FIRST and BUFPSN are addresses, the others are positive parameters. FORMAT is intended to place an array of items of the same type into evenly spaced positions in the print line. The number of items is the parameter NVAR, which must be one or greater. FIRST is the address of the first item; STEP is the address increment between successive items and must be zero or greater. TYPE is an integer code from zero to four and determines the type of items being formatted; these (and their codes) are: floating point (0), compressed (1), integer (2), alpha 2 char/word (3), and alpha 1 char/word (4). BUFPSN is the address in the print buffer (not displacement) of the first item, it is the address of the left-most character for alpha types, and the right-most character for numerical types. BUFSTP is the increment in characters between successive items and must be positive or zero. DEC is the number of digits to the right of the decimal point for compressed or floating point items; it is the number of words per item for alpha types.

FORMAT might be made easier to use if the buffer and/or item parameters were given as displacements and the beginning of the buffers were passed in the registers.

MOVE and CLEAR are used for moving and clearing blocks of memory. In using MOVE, the target address should not lie within the block being moved. In using CLEAR, the block is cleared with the value in the B register, which is usually zero or blank.

2.9 Display Utility

The LEO display utility contains subroutines to provide plots and other functions on the Tektronix terminal.

The subroutine PLOT is used by other subroutines in the display utility, but not directly by other LEO modules; it will initialize the graphic mode and draw a line or point or move the beam.

TYIN and TYOUT are used for transmitting characters to or from the terminal; these subroutines check for the occurrence of PCM interrupts when they must wait.

CURSOR interfaces with the graphic input mode of the terminal, it is not currently used by LEO, but may prove useful later. When called, CURSOR turns on the graphics cross hairs and waits for the operator to hit a key; then CURSOR returns the character and the CURSOR X and Y coordinates in the A, B and X register respectively. The coordinates might be used to position the beginning of a message, or by using the parameters to AXIS and

GRAPH to determine a channel number and value.

DOCU is used to place documentation on plots. The parameters are passed in the registers: A, B gives the X, Y coordinates of the beginning of the message; X contains the address of the message. The first word of the message must contain the number of words of text following. If the A register is negative, default coordinates in the upper right screen are used (with room for 30 characters) if the A register is greater than 1023, the CURSOR is activated and the position designated by the operator is used. On return from DOCU, the A, B registers have coordinates for the next line.

AXIS is used to draw plot axes, with tick marks and labels. The parameters passed to AXIS must be in a table whose starting address is given in the B register. Parameters may be integer or floating point independently for the X and Y axis. The table consists of 20 words, 10 for the X axis and 10 for the Y axis. The 10 words are arranged as:

TYPE	If Ø, MIN, MAX, TICK, LBL are interpreted as floating pt; otherwise first words are taken as integers.
MIN (2 words)	Lowest value (at origin)
MAX (2 words)	Value at end of axis
TICK (2 words)	Interval for tick marck
LBL (2 words)	Interval for long tick mark with numeric label
DOC	Address of axis label (format of label as for DOCU)

GRAPH is used to plot an array within a set of axes; the AXIS subroutine must be called first, since GRAPH uses the scale information computed by AXIS. The parameters for GRAPH must be in a table whose address is passed in the B register.

GRAPH can handle integer, floating point, or compressed arrays; it can also transform the data by one of three methods or by an external subroutine. (This transformation does not affect the data.)

The table should be arranged as:

NVAR	Integer number of values in array to be plotted
FIRST	Address of first value
STEP	Address increment for successive values
XSTART (2 words)	X value to be used for first value to plot
XSTEP (2 words)	Increment of X-value for successive values
GXTYPE	Ø if XSTART, XSTEP are floating pt; otherwise the first words are treated as integers
GYTYPE	Type of plot: Ø for point-to-point line, 1 for histogram type, 2 for symbol at beginning of line
XFRMAD	Address of transformation subroutine, Ø if none desired, GLOG, GSQRT, GINVRS for log, square root, or reciprocal. Address of external subroutine for special transformation
SYMBOL	Integer code for symbol to use with GTYPE = 2, otherwise not used.

MPR is a message print subroutine. This call is

```
CALL    MPR,NWORDS,MSGAD
        :
MSGAD   DATA 'TEXT'     . 'TEXT' OCCUPIES NWORDS
```

2.10 Amplitude/Event Time Task

The accumulation task is initialized by the routine SETAT, which obtains and clears the buffer, places ACCAT on the task list and initializes the livetime to zero.

ACCAT is executed once each minor frame. It acccumulates the livetime (in floating point); and then loops 16 times (once for each amp/time word in the minor frame) incrementing the words in the buffer corresponding to the event amplitude and event time; a counter of minor framesis also incremented. (The looping is not managed with the LOOP and LEND macros, but ought to have been; see Section 2.15.)

The printout task is initialized by itself, on operator command. It obtains and clears a print buffer of 132 words, ejects the printer page and forms the heading lines. Then a nested pair of loops forms the print lines of data; these loop 32 times, once for each line; and 7 times for columns in each line, the first and last columns are handled separately.

The display task PHADSP is initialized in two different ways, by ATDST ATSST, depending on the command used; ATSST sets scale and channel parameters. PHADSP plots the amplitude spectra after the operator types something. If zero is typed, the scale is doubled; if a positive number

is typed, the task is terminated; if a negative value (corresponding to any alphabetic character on the keyboard) is typed, the plot is repeated. The AXIS and GRAPH subroutines do all the work, and only the parameters used are changed.

2.11 Sub-Com Data Task

The accumulation of sub-com data is performed by the subroutine ACCHK. This task is initialized by the subroutine SETHK, which calls FIELD to obtain the maximum number of major frames to be accumulated, obtains and clears the buffer and places ACCHK on the task list.

ACCHK is executed once each minor frame. It checks the minor frame number to determine if sub-com data is present; if so, the address in the buffer is computed from the major frame count and sub-com channel (a code from 0 to 31 computed from the minor frame number), and the data is stored. The buffer is arranged as 32 countiguous buffers, each of length specified to SETHK; the first 16 are interpreted as compressed data, and the last 16 as integer data. If the buffer becomes filled, a message is typed and ACCHK is removed from the task list.

The printout task prints the accumulated data in three parts: the 16 integer channels are printed in 16 columns, the 16 compressed channels are printed in two sets of 8 columns. Abbreviations for the data titles are printed at the top of each column.

The display task clears the screen, and types a message. It then positions the beam so that the operator's input appears in the upper right instead of on the left margin. FIELD is called twice, for the abbreviated data title and the plot symbol to be used. Appropriate axes are drawn for integer or compressed type data (log axes for compressed type), and GRAPH is called for the actual plotting.

2.12 Integral Discriminator Count Task

The IDC accumulation task is executed once each minor frame. It checks the minor frame number to determine if the variable threshold value or IDC data is present. If the variable threshold is present, it is compared to the expected value for error detection. If IDC data is present, the current threshold value is used to select a bin in an IDC spectrum for one of five detector sections, the IDC count is then added to the bin. The least significant bit of the threshold indicates the state of the coincidence/anticoincidence circuit, and the higher bits select a value from 0 to 127; however, the buffer is arranged as five interleaved buffers of 256 bins (with 2 words/bin). The displacement in the buffer for bin n and section m is $2*(m + 5 * n)$. Then even numbered bins correspond to coincidence spectra, and odd bins to anticoincidence spectra. When 32 major frames have been accumulated, the flag ENDFLG is checked; if it is non-zero, the task is terminated.

The IDC printing task prints the IDC spectra in two parts: the coincidence spectra are printed, the five sections in five columns; then the anticoincidence spectra are printed.

The IDC display generates 10 plots on semi-log axes. The subroutine waits for operator input before each plot, then calls AXIS, DOCU and GRAPH to generate the plots.

2.13 Dump

The dump task is set up by the routine DMPSET. DMPSET calls FIELD twice for the parameters, and obtains and clears a buffer to save the data; then DMPSAV and DMPRNT are placed on the task list.

DMPSAV is executed every minor frame. It checks the starting minor frame number to determine if saving has previously started. If not, it compares the current minor frame and the starting minor frame; if these match, the frame is saved and a negative flag is placed in the starting minor frame. A frame counter is incremented. When saved, the amplitude and event time are separated into 2 words for ease of printing.

DMPRNT compares the saved frame counter and the printed frame counter; if these are the same, nothing is done. If unprinted frames have been saved, they are placed into a print buffer and printed.

2.14 Experiment Status

The experiment status task, EXPSTA, is automatically placed on the task list when the system is initialized. It is executed every minor frame. When minor frame 47 comes in, the experiment status is extracted from the sub-com data and decoded, forming a title which is accessible by other routines in the system. This message is typed on the terminal and EXPSTA deletes itself from the task list. This routine could be easily altered so that it continuously monitors the status, printing the title whenever the status changed; however, the original specification was that the status would not change during an experimental run.

2.15 Conventions and Suggestions

The macro capability of the DAS assembler can be used to eliminate some of the tedium of programming. In particular, the floating point routines are always called by a macro-coded mnemonic, rather than referring to the forgettable names given by VARIAN. These macros are listed at the beginning of those modules which use them.

The management of repetitive loops is often frustrating to the programmer: a loop counter must be allocated and a value incremented or decremented until zero or overflow occurs. A set of three macros has been defined in LEO to eliminate this frustration. LOOP is used for looping a fixed number of times; DO is used for a variable number of times; and LEND is used to terminate a loop. Usage examples follow:

```
LBL1    LOOP    LBL2,23      .LOOP TO LBL2 23 TIMES
          .
          .
          .
LBL2    LEND    LBL1         .END OF LBL1 LOOP
LBL3    DO      LBL4,N       .DO TO LBL4 N TIMES
          .
          .
          .
LBL4    LEND    LBL3         .END OF LBL3 LOOP
          .
          .
          .
N       BSS     1
```

The code for the LOOP and LEND macros is available in the listing of the DUMP module. The code for the DO macro differs from the LOOP macro in one instruction: instead of LDAI,P(2) the instruction LDAE,P(2) is used. The A register is altered at the start of each iteration, and contains a counter value, beginning at N-1 and decrementing to zero, after the DO or LOOP macro. The B and X registers are not disturbed.

2.16 Buffers

The first buffer that data passes through in LEO is the PCM buffer. This consists of 64 words, each holding 2 8-bit bytes. The data is assigned according to the following table:

Word	Data
0-1	sync (not used)
8(high byte)	live time
16	minor frame counter
48(high byte)	major frame counter
49(high byte)	sub-com data (depends on minor frame)
3,7,11,15,19,23,27, 31,35,39,43,47,51, 55,59,63	event amplitude (high byte) and event time (low byte)

The PCM buffer is used only to transfer the data to a simpler buffer for use by the processing programs of LEO. The PCM buffer is fixed, with starting address of 0400.

The next buffer format is that used by the Compact Data Tapes. It consists of 20 words, assigned as follows:

Word	Data
0	major frame count
1	minor frame count
2	sub-com data
3	live time
4-19	amplitude/time bytes

The CDT buffer actually consists of a record identifier (always zero) followed by four buffers as specified above. This buffer starts at address 0140; the address of the buffer currently in use is contained in the word named RECPTR.

Accumulation of event amplitudes and times is done in a buffer of 288 words: 256 for the amplitudes and 32 for the times. This buffer is allocated when needed and is accessed through the pointer whose address is in ATBUF.

Accumulation of sub-com data is done in a buffer of variable length. There are 32 sub-com channels defined in each major frame, and the operator specifies the number of major frames (NFRAME) to be accumulated. The size of the buffer is the product of these: 32*NFRAME. This buffer is logically divided into 32 sub-buffers, each consisting of NFRAME contiguous words. Because of the different types of data in the sub-com channels, the first sixteen are treated as containing compressed data, and the last sixteen as integer data. The sub-com buffer is allocated as needed, and accessed through the pointer whose address in HKBUF.

Accumulation of the shield counting data is done in a buffer of 2560 words. This is 10 buffers of 128 bins with 2 words per bin, since the accumulation is done in floating point arithmetic. These 10 buffers correspond to two types of spectra for each of 5 detector sections. The buffers are interleaved instead of separate. The displacement of the beginning of a spectrum is

given by the expression 4*SECTION + 2*TYPE where SECTION is an integer from 0 to 4, and TYPE is 0 for coincidence and 1 for anticoincidence.

The displacement between successive bins of a particular spectrum is 20 words. The beginning of the entire buffer is accessed through a pointer whose address is in IDCBUF.

The accumulation of minor frames for dumping on the printer is done in a variable length buffer. Since the event amplitudes and times are split, 36 words are required for each minor frame. The beginning of the buffer is accessed through a pointer whose address is in DMPBUF.

All print buffers are 132 words in length, and each word holds 1 character. Although this appears to waste space, since each word is capable of holding 2 characters, it was found that the extra program code to store and retrieve the characters for printing applications was more than 64 words; this arrangement appears optimum since there is not likely to be more than one print buffer allocated at anytime.

New Technology

There have been no reportable items (inventions, discoveries, or innovations) as defined by the New Technology Clause (NPR 9.101-4). Furthermore, there have been no subcontracts awarded under this contract.

Distribution (required by contract)

No. of Copies	Addressee	Code
1	Systems Reliability Directorate, GSFC	300
1	Publications Branch, GSFC	251
1	Patent Counsel	204
1	Contracting Officer, GSFC	286
1	Documentation Branch, GSFC	256
6	L. E. Orwig, GSFC	682